Analyzing Income Inequality in South Asia Through Interactive Data Visualizations

by

Shaiera Sultana Oishe
21201339
Asma Faria Mayami
21201420
Tabassum Bintay Mafij
24341245
Samin Mehzabeen
23341132

A project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
January 2026.

Declaration

It is hereby declared that

1. The project submitted is my/our own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shaiera Sultana Oishe

21201339

Asma Faria Mayami

21201420

Tabassum Bintay Mafij

24341245

Samin Mehzabeen

23341132

Approval

The project titled "Analyzing Income Inequality in South Asia Through Interactive Data Visualizations " submitted by

1. Shaiera Sultana Oishe (21201339)
2. Asma Faria Mayami (21201420)
3. Tabassum Bintay Mafij (24341245)
4. Samin Mehzabeen (23341132)

of Fall 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Computer Science and Engineering in Fall 2025.

Examining Committee:

Supervisor:
(Member)

Annajiat Alim Rasel

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Rakin Bin Rabbani

Lecturer
Department of Computer Science and Engineering
BRAC University

Project Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

In South Asia, where socioeconomic gaps have frequently been widened rather than addressed by fast economic expansion, income inequality is still a major problem. Static charts and technical reports are two examples of traditional means of presenting inequality statistics, and they often fall short in explaining the wide-ranging, demographic, and region-specific subtleties of inequality to non-experts and local officials. By creating, refining, and testing an interactive data visualization tool that converts intricate inequality indicators into understandable, captivating, and useful visual narratives, our study fills this communication gap. This study incorporates indices of economic, educational, gender, and digital inclusion across South Asian nations, using data from the World Bank, UNDP, national government websites, and gender-disaggregated databases. Utilizing visualization technologies like Tableau and Streamlit and Python libraries like Pandas, Plotly, and GeoPandas, the project delivers an interactive dashboard with real-time, multi-dimensional filtering and comparative analysis capabilities of wide-ranging datasets. Choropleth maps, time series graphs, and dynamic bar charts are some of the visualizations used to display both temporal and spatial shifts in inequality. According to user testing with journalists, students and development professionals, the benefits of interactive visual storytelling greatly enhance the understanding, engagement and memory of inequality statistics, especially for non-technical and novice audiences. Additionally, data literacy and public discussion is enabled by design features such as mounting narrative on maps, geographical filtering and simplicity. This study comes to the conclusion that data visualization is a potent instrument for altering the method with which the public interacts with social and economic facts. Most importantly, this gives a repeatable paradigm for data-driven policy support, public awareness, and inclusive governance in South Asia.

Keywords: income inequality, South Asia, data visualization, dashboard, gender disparity, regional development, public awareness, interactive storytelling.

Acknowledgement

First and foremost, all praise and gratitude is due to Almighty Allah, whose guidance and mercy sustained us throughout this project. It is through His grace that we were granted the strength and clarity to see this work to completion.

We would like to express our deepest gratitude to our parents and families, whose unconditional love, sacrifices, and continuous encouragement have been our greatest source of motivation throughout our academic journey. We also extend our sincere appreciation to our friends for their moral support, understanding, and encouragement during the course of this project.

We are profoundly grateful to our supervisor, Annajiat Alim Rasel, and co-supervisor, Rakin Bin Rabbani, for the invaluable guidance, and constructive feedback provided throughout this work. Their expertise and commitment to academic excellence played a central role in shaping the direction and quality of this project. We would also like to extend our special thanks to our fellow classmate, Seaumul Islam, for his generous support and assistance whenever it was needed.

Finally, we acknowledge all those who contributed, directly or indirectly, to the completion of this project. Their support is sincerely appreciated.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Income inequality is a pervasive and complex phenomenon that plays a significant role in the social, political and economic dynamics of a wide range of countries. The severity and depth of inequality is especially acute in South Asia, a region with a history of deep-seated social segregation and high rates of urbanization, as well as large population and marked socioeconomic inequalities. Despite some noticeable outcomes of economic growth in a number of states like Bangladesh, Sri Lanka, and India in the past few decades, the benefits of such growth have not been fair and equitable in these societies. World Bank statistics (2023) reveal that countries that have Gini scores between 32 to 36; a range that reflects moderate to high levels of income inequality, include Bangladesh and India, where income disparities are evident very distinctly in urban and elite strata. These numbers, however, do not always convey the complex local, regional or demographic aspects of inequity. Marginalized groups such as women, ethnic minorities, and rural populations continue to face obstacles to opportunity and development. Marginalized groups including women, ethnic minorities and rural populations are still faced with barriers to opportunity and growth.

Moreover, global forces including increased globalization, foreign direct investment, and the digital revolution allow for increased economic opportunities, infrastructure, and investment while at the same time increasing the inequalities of the socioeconomic disparity. Against this backdrop, sociocultural barriers to women's participation in the labor market continue to exist, so does the wage gap between men and women. Women continue to receive significantly lower remuneration for work of comparable value. The causative factors of inequality in South Asia are manifold which include both structural and socioeconomic variables. Income disparities are widening due to inconsistency in institutions, gender differences in the workforce, rural-urban disparities and limited access to quality education. Furthermore, global phenomena (rapid globalization, foreign direct investment, the digital revolution), while providing additional economic opportunities, infrastructure and investment resources, also have an inequitable impact on different socioeconomic strata, widening rather than bridging inequalities. In addition to sociocultural barriers that limit the participation of women in the labor market, gender-based wage differences remain a marked problem: women earn significantly less for equal work than men. Although a plethora of data on income disparity are provided by organizations, including World Bank, UNDP, as well as national statistical agencies, there are barriers to comprehension, interpretative challenges, and complicated, sporadic reports of information that often beautifully explained, but had little or no audience, impeding effective evidence based research and consequently impeding effective intervention, well-informed policy formulation, civic engagement, and informed discourse. Well-built and designed visualizations can help highlight disparities on the level of social strata, geographical areas, highlight hidden trends, and help novices and experts alike to intuitively explore patterns and outliers, thus enhancing active engagement with the information and not passive consumption. Despite the increasing use of data storytelling in journalism and government across the world, there is not a multi-faced, interactive, regionally oriented platform for visualizing the issue of income disparity in the South Asian region. Existing initiatives are fragmented with reliance on broad but superficial information, with no middle ground of integration of accuracy, insightfulness as well as the capacity to filter dynamically. Furthermore, for South Asia to achieve the United Nations Sustainable Development Goals, in particular SDG10: Reduced Inequalities, along with raising public awareness, another need is the ability to tell stories from facts so that they can be made accessible during policymaking. Consequently, in this study, a design and performance testing of an interactive data visualization framework has been implemented that can facilitate comparative, user-centered, real-time investigation of inequality across South Asia.

1.2 Motivation

This project is driven by the persistent problem of economic difference in South Asia and growing disconnection between people's access to data and their ability to use it effectively. Even as national governments, international institutions such as the World Bank, and the United Nations Development Programme regularly publish datasets containing socioeconomic indicators, these sets of information are often presented in static and inoperable formats, limiting their usefulness to the general public, regional policymakers and the non-specialized users. The overwhelmed need for technologies that could enable the transformation of complex inequality data into interpretable, interactive, and actionable insights represents a large research imperative, in the form of this gap. Inequality in this region is dynamic, large, and jurisdictional, and it cannot be adequately represented in traditional reports and spreadsheets. Interactive data visualization has the potential to bridge that gap and change the way individuals perceive and respond to inequality, and is thus what motivated this analysis. By using existing methodologies from data science to provide an infrastructure that enables dynamic filters, geographical contrasts, and visual storytelling, the project intends to increase public awareness, encourage data-driven conversation. In particular, in the context of the post-pandemic climate, when economic and digital divides have have widened throughout communities, the committee is not simply technically important, it is also a socially consequential one.

1.3 Problem Discussion

Income disparity is hard-wired in the social and economic systems in South Asia, even with the region's recent rapid economic development. Substantial gaps remain in education, gender, employment, and digital over countries like Bangladesh, Nepal, India, and Pakistan. The extent to which inequality data is practically used to inform the public and evidence based policy choices in practice is limited by its immobility and interpretability despite the existence of large datasets on such subjects through organizations such as the World Bank, UNDP, and national open data platforms. By creating and assessing an interactive data visualization platform that converts unprocessed inequality data into dynamic, approachable visual narratives, our project gives a variety of users the ability to investigate and contrast patterns in income disparity across nations, time periods, and demographic categories. In order to promote increased involvement, transparency, and well-informed decision-making in the South Asian context. The following research questions are put forth to direct the current study:

- What is the communication potential of interactive data visualizations in illustrating the diverse character of income disparity in numerous socioeconomic and geographical areas of South Asia?
- To what extent does inequality persist in South Asian nations due to institutional structures, policy gaps, and cultural variables?
- In what ways can an interactive visualization platform dynamically combine fiscal, demographic, and financial data to provide real-time comparisons of inequality among South Asian nations?
- Which visual storytelling techniques work best to improve non-technical audiences' comprehension of inequality statistics in South Asia?

1.4 Project Aims

The overall objective of the project is to design and create an interactive data visualization platform to allow users to intuitively explore and establish comparative knowledge of income inequality in South Asia. The project aims to convert complicated and multidimensional socioeconomic data into visual stories accessible to all in an attempt to uphold meaningful interpretation by all technical and non-technical audiences. In particular, this project is expected to:

- Collectively represent economic, educational, labor market, gender, demographic, and digital access indicators within a single analysis context to form a clearer analysis of income inequality in South Asia.
- Allow users to discover the spatial, temporal, and cross-country differences in inequality by visualizing the inequality using interactive and comparative visuals instead of plain visuals.
- Encourage data literacy, public consciousness, and an informed conversation through displaying inequality data in a user-centered, understandable, and readable way.

- Advance research and policy dialogue with evidence supporting and uncovering hidden patterns, differences, and associations between notable socioeconomic measurements that pertain to inequality within the South Asian context.

1.5 Project Objectives

In order to meet the above aims, the project has the following specific objectives:

- To assemble, clean, standardize, and merge inequality relevant data across various credible sources, such as edited data sets and real-time World Bank API indicators, to provide consistency, transparency, and reproducibility.
- To create and develop a highly interactive dashboard through which users could study income inequality at national, country, year, and demographic levels using dynamic filters and visual exploration instruments.
- To successfully provide visual disjointures, representations of important indicators, including income distribution, Gini coefficients, poverty prevalence, labor force participation rates, and gender disparities will be realized through the use of comprehensive visualizations.
- To permit multi-dimensional analysis by looking at associations amid inequality and other larger measures of economic and growth outcomes, such as GDP, work structure, educational achievements, and populace makeup.
- To include data quality information that specifically conveys data completeness, gaps, and reliability, lessening any misinterpretation that occurs through the absence of coverage or inequitable coverage.
- To determine if interactive visualizations are more effective than the static ones based on the usability methods and user exploratory feedback, one will pay attention to the state of clarity, engagement, and interpretability.
- To create such usability, visual clarity and intuitive interaction design that means the platform can be accessible to a very large audience of users, such as students, researchers, journalists and policy makers.

1.6 Project Challenges

Although the creation of a citizen-friendly interactive data visualization tool to inquire about the levels of inequality in eight South Asian nations was successful, the project had a number of technical and practical obstacles, which are inherent to the examination of vast socioeconomic data.

Data standardization: The project was based on the data obtained using a variety of different sources, which was necessary to process and required a lot of cleaning, normalisation, and harmonisation to ensure similarity between indicators, countries, and timeframes.

Technical complexity: It was important to design an interactive dashboard that is easy to use and capable of managing high-dimensional and large scale data. The need to have real time interactivity and clarity of presentation to non technical end users necessitated trade-off in design and optimization.

User feedback: Getting useful feedback from non-technical policymakers and the public was important in order to fine-tune the platform, but it will be challenging.

Computational and resource constraints: Processing real time data in real-time, interactive filtering, visualizing large datasets, created limitations to the entire system performance and resource requirements, especially in a lightweight deployment environment.

Data aggregation and updates: Aggregation of multiple data streams and keeping the platform updated to include newer as well as new data on a continual basis will be a matter of the maintenance of a relevant and accurate platform.

1.7 Contribution

This project applies to the analysis and reporting of income inequality in South Asia; the difference between complex socioeconomic records and the general information is closed through the interactive and user-friendly visualization site. Besides technical implementation, the project is also concerned with accessibility, interpretability, and transparency, which offers a convenient structure in which one can learn about inequality through the assistance of modern data visualization tools.

1.7.1 Accessibility Focus

Among the biggest successes of the project is the income inequality data that is now offered to the population who are not experts. Even the traditional data on inequality is generally in a form of stagnant, technical and jargon-laden document that is not readily available to the masses. The proposed project will address that gap since it will transform raw socioeconomic indicators into visuals that are user-friendly and can be explored without sophisticated statistical abilities. Through its focus on being simple, visual and interactive, the platform allows students, journalists, and policy-makers to have a better understanding of the existing income inequality and economic factors in South Asian countries.

1.7.2 Integration of Modern Tools

The project shows how well modern open-source software, such as Python, Streamlit, and Plotly were integrated to create a powerful interactive data visualization platform. Python offers stable data-processing and analytical functions, whereas Streamlit is a minimalist and reactive web application framework. Plotly has been used to create interactive visualizations that are dynamic and can be used to explore multidimensional data.

1.7.3 User-Centered Design

The other significant value of this project is that it has been shaped with user-centered design. The dashboard includes user friendly navigation, interactive filters and intelligent search engine which enables the user to easily find and study indicators of interest. The exploratory analysis is facilitated by real-time filtering, comparative views and responding visual feedback, which are useful in alleviating cognitive load on the user. The site encourages interaction and engagement with inequality information rather than being merely consumed, and this is what makes it popular since the site focuses on usability and interaction.

1.7.4 Modular and Scalable Architecture

The project suggests a scalable and modular system architecture, which has multi-source data integration. The platform accommodates a broad range of data sources and consistency and transparency is made possible by combination of pre-selected datasets and live indicators which are based on API. The system is designed using a modular structure so that the separate modules - such as data processing, visualization and user interaction can be extended or updated independently. This architectural style makes it easier to maintain and allows it to be expanded in the future, adding new indicators or countries or analytical modules.

1.8 Organization of the Report

This report is presented in eight chapters. Chapter 1 gives the background of the study, motivation, discussion of the problem, project goals, objectives, challenges and contributions of the project. Chapter 2 conducts the literature review on the current works that are concerned with the measurement of income inequality, socioeconomic indicators, and data visualization methods. Chapter 3 includes a project description, such as, requirement analysis, system requirements, tools and platforms utilized in the project, feasibility assessment and posits and limitations identified. Chapter 4 is devoted to the analysis of the project which encompasses system architecture, data flow, integration of dashboards, and design of user interaction. Chapter 5 gives us a summary of how the system explains the implementation including on how to set up the environment, data processing pipelines, and development of each dashboard module. The 6th chapter is a visual description of elements in the dashboard and visualization. In Chapter 7 the author discusses the results, key findings and observations as a result of the analysis and user evaluation. Lastly, Chapter 8 finalizes the report and provides the possible lines of improvement and further growth.

Chapter 2

Literature Review

2.1 Preliminaries

Income inequality in South Asia represents a major socioeconomic challenge, which is further complicated by overpopulation, the heterogeneity of its developmental status, and complex political and social dynamics. Whilst many countries in the region have seen strong economic growth, large sections of the population still suffer from inequality in access to income, education, healthcare and work. The benefits of economic growth have not been evenly spread; rich urban sectors in particular benefit the most from foreign direct investment and the forces of globalization. Conversely, marginalised groups, which include women, communities near rural areas, and weaker caste groups, continue to be excluded from the benefits of entire growth, thus leading to the continuation of the poverty cycle. Educational inequity and gender disparity are key determining factors that contribute to income inequality. The goal of this study is to interrogate these complex determinants of income inequality in the South Asian context by utilizing interactive data visualizations. Integrating the metrics on economic, educational, gender, and other socioeconomic indicators, the study focuses on producing a holistic picture of the income disparities within the region. The resultant visualizations of this study are meant to improve interpretability and engagement for policymakers as well as the general public.

2.2 Review of Existing Research

Understanding the dynamics of income inequality in South Asia calls for an understanding of many economic, policy, and structural variables that influence the distributional outcomes in the region. Numerous empirical investigations have been carried out in order to examine the influence of globalization, economic growth, public debt, and foreign investment on the inequality trends of Bangladesh, India, Pakistan, Nepal, Bhutan, and Sri Lanka. Munir and Bukhari [8] presented a pioneering examination of the nexus between globalization and inequality in Asian emerging economies through a panel data model for the period 1980–2014. They documented that trade and technological globalization have inequality-reducing effects through boosting employment opportunities for low-skilled labor and knowledge diffusion. Financial globalization, however, was found to increase inequality, particularly where financial infrastructure is lacking, due to the fact that the richer portions of the population capture disproportionately large gains.

Education proved to be a strong leveling factor, while FDI reinforced inequality as it concentrated capital in skill-intensive industries. These observations highlight the multidimensionality of the effects of globalization and confirm the aim of this research in utilizing visual analytics for tracking and comparing the distributive effects of trade, investment, and technology on South Asian regions.

To counterbalance this perspective, Acharya and Acharya [9] examined the inequality-growth relationship in six South Asian countries, 1980-2015. To strengthen estimates of panel data models, such as GMM, to correct for endogeneity, they observed a statistically significant positive correlation between momentary economic expansion and disparities in income. This evidence suggests that there will be greater productivity and investment due to capital accumulation and wealth concentration in the early stages, especially in underdeveloped economies. Notwithstanding this, their paper also noted that female schooling with education has good impacts on growth, while male schooling had little or negative effects, having implications for gender disparities in return to education. Furthermore, market distortions—measured in terms of investment price indices—were also shown to suppress growth. Their findings validate the argument that economic growth, being desirable, is not sufficient in an attempt to reduce inequality unless partnered with particular educational and policy measures. These connections are crucial to the visual analysis of this research on the balance between equity and efficiency in regional development.

On the other hand, Shafiullah et al. [2] presented a different argument to explain growth-led inequality on the grounds of income inequality, placing a limit on long-run economic growth. Panel data from 1970 to 2010, system GMM and fixed effects estimators allowed the authors to find a strong and statistically significant negative association between income inequality and per capita GDP growth in Bangladesh, India, Pakistan, and Sri Lanka. The investigation underlined how unequal distribution of income could hamper and retard the investment in human capital, depress aggregate demand, and create social tensions. Their findings provide evidence to substantiate the research endeavor in bringing not only to the foreground the magnitude of inequality in South Asia but also its macroeconomic roots, as reflected in dynamic trends and cross-regional comparisons over time.

Yet another aspect to this debate, Akram and Hamid [3] investigated the influence of different forms of public debt on income inequality in South Asia from 1975-2010. Their empirical estimation, based on fixed effects, 2SLS, and GMM models, detected that while external debt and debt servicing had no significant impact on inequality, domestic debt was associated with lower income inequality. This means that domestic borrowing—if directed to public investment, social safety nets, or development projects—can play a redistributive role. The paper also confirmed that urbanization and trade openness raised inequality, while rising GDP per capita had an equalizing effect. These findings are highly relevant to the research visualization framework that aims to include fiscal variables such as debt structure and urban-rural income gaps as dynamic components of inequality assessment.

Finally, Arshad and Islam [7] contributed to the literature by examining the effect of FDI inflows on income inequality in SAARC and ASEAN countries for the 1990-2015 period. Their results, determined using fixed effects, DOLS, and ECM specifications, revealed that FDI reduces inequality in South Asia but increases it in Southeast Asia. This divergence highlights the importance of regional policy environments, labor markets, and industrial structure in conditioning whether foreign investment widens or closes inequality. The research implications are clear: data visualization must not just represent inequality metrics but must also allow users to interactively compare the impact of variables like FDI across countries and periods, highlighting both regional synergies and divergences.

However, Ahmad, Abukasim, and Hussain [16] investigate how rural migration affects income inequality in South Asia, with board data from 1995 to 2022. The research applies the Kuznets curve, which infers that when a country begins to progress economically, it will experience an increase in inequality for a period before switching to economic advancement which reduces inequality. Their findings lend good data support for a phase of the curve that narrows the income gap through rural-urban migration. Unlike rural dwellers, when migrants move from rural areas to urban centers, they generally have better access to employment, education, and healthcare opportunities to economically advance their economic situation. The paper touched on the clear process of migration to narrow spatial gaps or disparities, but recommended there be better policies to smooth out the transition from rural to urban migration. Furthermore, the authors provided ample visual research data in their analysis through the use of time-series visual charts and graphs that captured the trends, as well as made the complex research data clear and easy to understand.

Meanwhile, the socio-economic inequality published by the London School of Economics [18] takes a broader, multi-country view of inequality in South Asia. The socio-economic report produced a narrative framework that was rooted in historical, political, and cultural variables to show how and why inequality continues as a consequence of the entrenched factors, establishing why, and how social income indicators improve in India or Sri Lanka. The report outlined that despite improvements in the national economic indicators, income, education, social and political gender inequities, and healthcare inequalities remain due to institutional arrangements, social orders, and policy gaps. Although the report acknowledged that countries like India and Sri Lanka have made palpably visible progress to improve upon, there is concern that prevailing inequities will remain as long as there are no significantly better constructed, inclusive economic and social policies at play. The data included in the socio-economic report demonstrates, with reference to graphs and thematic maps, exactly how complex and entrenched our social and political processes are associated with ingrained inequality contexts.

Similarly, Roychowdhury [10] provided an examination of socio-economic disparities and development challenges in South Asia, concentrating his research on gender inequality related to economic growth. Using time-series plots and bar charts, he illustrated the pathways between increasing GDP levels and increasing inequality, first identified in women and underserved communities. Roychowdhury recognized that while economic growth has been the priority to reduce poverty in the region, inequality continues to limit sustainable development and social cohesion, and he stressed the impacts of gender inequalities in income, employment, and decision-making roles as constraints to overall economic growth. He has identified the power of visual analytics, and he described how charts and graphs aid in revealing socio-economic trends that may conceal the data behind the numbers, helping policymakers to better select the type of reform required.

In the South and Southeast Asia area, the World Inequality Lab [12] report on distributions of income gives an insightful overview of income concentration, showing that top earners of the national income in countries like India and the Maldives have more than 50% commensurate share of their countries' income. Other countries, such as Bangladesh and Nepal, do not have extreme top earners but do have significant disparities. The report also shows how the pandemic increased income inequality during the pandemic, which is particularly relevant for informal workers who were the most impacted by the economic downturn. The interactive data consumes charts, and downloadable datasets provided by the World Inequality Lab provide opportunities to engage and explore inequality trends as we shift among countries (including the margin better). The pandemic emphasized various vulnerabilities of varying segments of populations, but emphasized economic policies so that the segments of the population that are vulnerable are included. Migrants in the area have until now remained included in shift migration since the first wave of the pandemic shifted populations.

Concurrently, the World Bank's Gender Data Portal [19] adds another vital layer to the research on inequality by focusing on gender differences in South Asia. The database includes disaggregated data around education, labor, health, and political participation to show the breadth of gaps in workforce participation for women, wages paid to women, and access to leadership. The data also provides context for women from rural and marginalized communities who often face cultural and institutional barriers to full economic inclusion compared to men. While the visual representations by the portal enable researchers to interactively view data trends related to gender across sectors, they have even greater implications. By highlighting the need for improved policies to foster gender equality in the economic and political domains. Additionally, the UNICEF Gender Counts [6] report is an intensive assessment of gender inequalities impacting girls and adolescents. Not only this, but also focusing on school enrollment, early marriages, and adolescent employment. It identifies gaps in data availability to evolve and observe effective policies and offers comparative charts to show where countries fit along the spectrum of gender equality.

On the other hand, the International Fund for Agricultural Development [5] investigates the rural-urban divide in South Asia and shows that rural populations are at a significant disadvantage regarding access to quality education, healthcare, and employment opportunities. The report recognizes that the challenges that rural populations face as a result of systemic disadvantages ultimately require investment. In rural infrastructure, vocational training programs, and support for diversifying rural economies beyond agriculture. This report provides a necessary focus on solutions, but it sparsely examines the realities of rural populations who are struggling against entrenched systemic disadvantages. There is so much information contained within the report, and the abundance of data further illustrates the importance of both systems and data to implement what is necessary to bridge the rural-urban divide. However, for implementing the recommendations at scale, the report and research also highlight the challenges.

The paper by Kanwal, A., & Munir, K. [8] explores how educational and gender inequality are connected with income inequality within the context of South Asian nations. This paper emphasizes the fact that unequal distribution of education leads to income inequality. As for equal education, distribution expands, and inequality decreases. Here, the study is based on Human Capital Theory (Marin & Psacharopoulos, 1976) that disparities in human capital accumulation directly impact income distribution. The methodology developed by Thomas et al. (2001) and Lin (2007) to measure educational inequality through Gini coefficients, demonstrates that unequal education can widen income gaps. Evidence showed by Gregorio & Lee (2002) and Connolly (2004) that, to reduce income inequality, investment in equitable education is necessary. Again, a study by Baliamoune & McGillivray (2015) and Baye & Epo (2015) shows that higher gender equality in education and balanced human capital distribution lead to lower income inequality. To conclude, for developing countries like South Asia, the fundamental sources of income inequality are education and gender inequality.

On the other hand, Jomo, K.S. [1] in his paper challenges the optimistic view promoted by the World Bank's East Asian Miracle report, showing that even though poverty declined, income inequality did not. This paper examines how economic liberalization and globalization inherently lead to both growth and inequality in Southeast Asia, mainly Malaysia, Thailand, and Indonesia. The paper shows a cross-country comparison among countries. Countries like South Asia and Taiwan experienced equitable growth because of land reforms and industrial policies. On the contrary, Malaysia, Thailand, and Indonesia did not have strong redistributive reform plans, therefore making their growth more unequal. According to Amsden (1989), Yu (1994), & Rao (1998), pre-existing equitable land and assets distribution in Northeast Asia was one of the elementary reasons for acquiring equity and growth. Wade (1990) & Rodrik (1994) explain that interventionist policies in Taiwan and South Korea, especially for education and industrial planning, were crucial for building a competitive domestic sector, whereas Southeast countries depended on foreign direct investment. Jomo(2001) also cited several studies explaining that globalization can raise inequality if nations lack strong institutional and restructuring frameworks. Gini coefficients, poverty rates, and wage growth data are also included in the paper. It verifies that, particularly after liberation and globalization, poverty may decline, but income inequality often increases.

Meanwhile, the paper by Kazár [4] emphasizes how to teach concepts of poverty and income inequality, especially to Generation Y university students, using statistical education. This paper explores the pedagogical research and existing statistical tools demonstrating inequality, at the same time focusing on how important it is to use interactive materials and technology in teaching. The paper also explores studies by Keeley (2015), Banerjee et al. (2006), and Ibrahima (2013), some standard tools cited to analyze the multidimensionality of poverty and complex measurement. Such as absolute vs. relative poverty, income quintile share ratio, and the Gini coefficient. Also, this paper proposes an instructional design based on the characteristics of Generation Y. According to educational psychology research done by Weiler (2005), Reilly (2012), and Eckleberry-Hunt & Tucciarone (2011), Generation Y is said to be visual, feedback-driven, collaborative, and technologically fluent. The paper also explores widely used databases such as Eurostat, and OECD.Stat, Gapminder, and their significance and efficiency in engaging with students. Gapminder is said to be the most effective as it has strong visual and interactive features. Even though Eurostat and OECD.Stat offers more indicators. Kazár advocates for database-driven and interactive team projects rather than paper-based tasks. The author also mentions that the students can better connect with real-life issues of inequality if they use visual data tools and datasets and analyze indicators like poverty rate, Gini index, and income share of the poorest.

Furthermore, in the paper by Bikakis, N. [11] studies the changing domain of big data visualization and focuses on the important role of visualization tools in studying massive, complex datasets. This paper identifies a critical demand for a scalable, efficient, and interactive visualization system. The 3Vs framework– Volume, Velocity, and Variety is a traditional data visualization method. These methods are insufficient when working with big datasets. The author examines the functionality of commercial tools such as Tableau, Power BI, QlikView, and open-source tools such as D3.js, Apache Superset to analyze their strengths in real-time analytics, dynamic dashboards, and interoperability. The combination of visual analytics and machine learning in fields like healthcare, finance, cybersecurity, and smart city management is also cited in the paper. GPU-accelerated processing, distributed computing (Hadoop, Spark), and in-memory analysis are described as important features in modern tools. The study concludes that combining statistical reasoning, human cognition, and machine learning is necessary for big data analysis.

2.3 Summary of Key Findings

Literature states that globalization has unidirectional effects on South Asia's income distribution, such that technological and trade globalization contain inequality by expanding employment, while financial globalization concentrates gains among high-income groups [8].The nexus of inequality-growth remains contested, with evidence for favorable correlations in early growth and adverse long-term effects on growth [9][2]. The underlying cause in most cases is educational inequality, the rural-urban educational gap that limits social mobility, and gender differentials, all of which accumulate their effects to reinforce the same, with more pronounced growth effects for female education than male education. Rural-urban migration has inequality-reducing potential founded on increased opportunities accessibility, albeit subject to the provision of being dependent on facilitatory transition policies [16].
There is a significant disconnect between inequality data availability and useful application since traditional static reporting practices are not designed to capture multi-dimensional, region-specific inequality patterns, thus limiting actionable information for policymakers and civil society. Interactive visualization tools are more effective in engaging heterogeneous publics, especially for sophisticated socioeconomic data [4][11]. Three important gaps are revealed in the literature that justify this

study: a lack of interactive visualization research focused on South Asian contexts of inequality, heavy dependence on static data presentation constraining user interaction, and a standalone analysis of inequality dimensions instead of multi-dimensional analysis platforms. These findings reaffirm that interactive data visualization tools possess high prospects of restructuring inequality data communication and utilization for evidence-based policy-making in South Asia.

Chapter 3

Project Description

This chapter provides an in-depth description of the South Asia Inequality Analytics platform, outlining its needs, computing facilities, benefits, problems and viability. This project is aimed at creating a web-based analysis system that will help provide a socio-economic inequality in South Asia utilizing the power of interactive visualization and multi-dimensional analysis of the data. Incorporating 259 indicators covering income inequality, metrics related to income disparity such as, poverty, education, employment, infrastructure, and health measures in eight countries, including Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, and Sri Lanka over a 20-year time horizon, 2000-2024, the platform will provide researchers, policymakers, and scholars with easily accessible tools that assist in studying regional development trends and inequalities. It includes 37,790 data points that have been collected by authoritative sources internationally, such as the World Bank, the United Nations Development Programme (UNDP), UNICEF, and the World Inequality Database (WID). There are nine different analysis pages: while Home page being the configuration center, (1) a complete Dashboard with a variety of visualizations, (2) Smart Search with natural-language querying, (3) animated Map Analysis, (4) Correlation Explorer with statistical relationships, (5) an Income Simulator with personal-modeling, (6)Data Quality monitoring, (7) Indicator Insights sunburst view, (8)Temporal Comparison tools and (9) extensive Help documentation in the system architecture. This configuration can be configured centrally and is also maintained centrally, allowing special analysis.

3.1 Project Specifications and Requirements

The requirement analysis is necessary to define the elements, user requirements, and system features that are needed to be developed and implemented successfully. This part will outline the requirements of the user and the system to ensure that the project achieves its goals and user expectations and therefore provide a platform on which an effective and user friendly analytical platform can be established.

3.1.1 User Requirements

Requirement analysis is an essential component activity required to determine the constituent components, user requirements, and system capabilities that are essential to the creation and implementation of the platform successfully. This area outlines the user and system requirements hence ensuring that the project achieves the stipulated goals. These needs emerged out of a methodical examination of related sites, interplay with existing development research methodological backgrounds, and a series of user tests with nine subjects, who offered meaningful feedbacks on functionality, usability, and other features they wanted. Table 3.1 summarizes the user requirements.

Multi-Dimensional Data Exploration: The user will be able to interactively choose countries, indicators and time periods in order to tailor their view of analytics. It uses 259 indicators in seven thematic groups and allows one to select any of eight South Asian countries (Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, Sri Lanka), year (2000–2024) and has a complete configuration interface. The two-tier indicator selection system adopts the progressive disclosure, a user may initially select a broad category (Income Inequality, Poverty, Education, Employment, Infrastructure, Health), after which, the user specifies indicators of the category. This top-down method makes the cognitive load of the user smaller and at the same time allows having access to the entire package of indicators. The system captures user settings that can be used through management of session state meaning that a user can navigate easily through the pages of the analytical processes without having to re-enter their settings. As a returning user, an optional profile system with an underlying Supabase database is stored with configurations permanently and can be instantly reinstated through email-based retrieval in the sidebar.

Interactive Visualization: The system has interactive charts and maps which will respond to the user inputs in real time. Each of the visualizations makes use of the interactive features of Plotly, namely, zoom (rectangular or box selection), pan (click and drag), hover tooltips (displays exact values, country names and years), and dynamic filtering. The legend items allow users to switch data series on/off. The Map Analysis page presents the animation of choropleth visualizations with control on year by year changes expressed in form of play/pause, control of speed and frame by frame navigation. The change in intensity of colors as the animation goes on introduces the temporal patterns to the audience immediately. Every analysis page except the help page demonstrates data visualizations as interactively as possible. Some of the formats by which users can export visualizations include: PNG, JPG and JPEG. Export buttons are consistently displayed on all visualization pages.

Statistical Analysis Tools: The user needs strict statistical procedures to determine the correlations and trends in the data on inequality. The Correlation Explorer computes Pearson correlation coefficients under SciPy with appropriate pairwise treatment of missing values that is, only country-year observations with a value on both indicators are treated in the calculation. The system will automatically classify strong ((coefficient ≥ 0.7), ($0.4 \leq$ coefficient < 0.7), and weak (coefficient ≤ 0.4) strength, as well as positive or negative direction (sign). Correlation matrices are presented in the colors of RdBu (academic standard: red means a positive correlation, blue a negative correlation, white a zero correlation). The trend analysis element computes slopes of the inequality path of individual countries since 2000-2023, with the inequality assuming positive slope (increasing), negative slope (reducing), or zero. Trend ratings have data reliability ratings: High as 10 or more years of data, Medium as 5-9 years, Low as less than 5 years. This transparency also makes the users be aware of the certainty when they interpret the findings. Statistical summaries are aggregating beyond nations, which implies the number of countries that are improving, getting worse, or remaining the same.

Smart Search and Navigation: Smart Search page enables users to query using natural language along five orthogonal axis, having avoided the use of hierarchical menus that require users to browse. The country search option uses case-insensitive substring matching i.e. typing "Bangladesh" will give a full metadata of the number of available indicators that the country has, the scope of time and the latest year that the country had data actually recorded, as well as the number of measure points. All the results have a control of View Country Analysis that preconfigures the dashboard to the country chosen and allows the automatic navigation. The indicator search also finds both technical names (e.g., 'gini index') and descriptive words that people can readily understand (e.g., income inequality) and shows its availability per country. The year search identifies a four-digit numeric query and confirms it in the data, the counts and coverage statistics of the particular year in the set and provides a Control "View Year in Map" which navigates to the geographical visualization dedicated to one particular year. Analytical keywords found with the command search include map, correlation, dashboard, sunburst, simulator, quality and temporal which in turn return descriptive page summaries with direct navigation buttons. The smart filter search makes semantic queries such as 'high inequality' (instead of top-ranked countries with a GINI coefficient over 40), 'low inequality' (GINI coefficient under 30) and lists matching countries and provides an action to "Apply Filter". The last ten queries are recorded in the recent search history to allow quick re-execution, whereas most popular queries are saved as a predefined bookmark with access to the common situation in one-click, such as All Countries Overview, High Inequality Focus, Recent Data (2020- 2024) and Long- term Trends (2000- 2024).

Transparency of Data Quality: User should be given in-depth information in the data completeness and reliability to make analytic decisions. The Data Quality page calculates each country-indicator pair completeness percentages against a 25-year historical target (2000-2024). The page provided a variety of visualization modalities: bubble charts showing the density of geographic data in bubble sizes where a bigger size means a higher completeness percentage and colors used to represent a quality category High: $> 80\%$, Medium: 50–80%, Low: $< 50\%$; Sankey diagrams showing the provenance of data to final quality categories with a flow thickness proportional to the size of the data, and ridgeline plots showing the probability density of the data quality per country through Kernel Density Estimation (scipy.stats.gaussiankde). In the critical gaps section, coverage below $< 60\%$ is explicitly identified using warning symbols and color coded in orange that alerts the researcher about the credibility of the findings based on such indicators. A toggle can be turned on to do a live API validate, where the World bank API[14] values are also currently retrieved, and contradictions above a threshold of one point of the system is turned on which is then presented as a tabulation of the values in the system.

Personalized Analysis: Income Simulator brings users into the context of economic modelling with abstract statistics of inequality. Users provides demographic and professional information using a user-friendly form, consisting of country (eight South Asian choices), level of education (no formal education, primary, secondary, tertiary), digital skills (none, basic, intermediate, advanced), gender (male, female, other), location (urban, rural), occupation (agriculture, manufacturing, services, unemployed, student), access to credit (yes, no), and age (continuous slider 18-65).

Accessibility and Ease of Use: Interface is focused on the idea of accessibility thus users of different technical levels can navigate the interface efficiently. Auto-focusing inputs can focus on what is required to be typed manually, so on loading a page, the country-selecting input is automatically focused. There are contextual help buttons (denoted by Quick Help icon), visible in the upper right corner of each page, which open foldable sections that give page-specific instructions and does not scroll out of the present view. The general Help page lists documentation by three tabs: Quick Start (a three-step initially-onboarding flows: Configure-Explore-Export), Features Guide (a menu on the 9 pages of documentation, being page-by-page overviews, key features, step-by-step usage instructions, pro-tips and browser compatibility tips), and Troubleshooting (a cumulative list of all 9 page related issues with corresponding solutions, general tips of issue). It has low set up requirements, users only require internet connection and a modern browser; they are not required to install the software, create accounts and configure files. These accessibility features verified by the user testing involving nine users and the feedback given by the participants specifically noted the following statements: the tool was very responsive and fast, read the data or graphs, easy to navigate, and user friendly. Recommendations were suggested in terms of adding more background information and context regarding indicators to understand them better and dealing with text overcrowding in some sections; these are the details that guide the current refinement priorities.

Table 3.1: Summary of User Requirements

Requirement	Description	Implementation
Multi-Dimensional Exploration	Select countries, indicators, time periods dynamically	Home page with 8 countries, 259 indicators in 7 categories, 2000-2024 year range, two-tier selection system
Interactive Visualization	Charts and maps responding to user inputs	Plotly-based visualizations with zoom, pan, hover, export (PNG, JPG, JPEG)
Statistical Analysis	Correlation and trend identification tools	Pearson correlation (SciPy), trend slopes, reliability ratings (High/Medium/Low)
Intelligent Search	Natural language querying across dimensions	5-dimensional search: country, indicator, year, command, smart filters
Data Quality Transparency	Visibility into completeness and reliability	Completeness %, bubble maps, Sankey diagrams, KDE ridgeline plots
Personalized Analysis	User-specific economic modeling	Income Simulator with scarcity-weighted algorithm, percentile ranking, insights
Accessibility	Easy navigation for all users	Auto-focus, contextual help, 3-tab Help page, no installation required

3.1.2 System Requirements

To meet the user needs, the system should meet a series of technical requirements that guarantee strong functionality and harmonized integration of its subsystems. These standards include web application framework, data processing pipeline, visualization engine, statistical computing capabilities, data storage backend, performance optimization, browsing, and cross-platform support. The choice of each component was made based on community support, quality of documentation, and the capacity to support the size of the project (259 indicators, 37,790 data points, real-time interactivity). Table 3.2 provides an overview of the system requirements.

Web Application Framework: The framework is a web application that uses Python and Streamlit (version 1.25+) which is a framework designed specifically to serve data-science applications. Streamlit empowers the use of interactive dashboards to be developed quickly, without the need to have any profound web-programming experience and skills in HTML, CSS, or JavaScript. Streamlit provides a reactive model of programming that is used to provide real-time updates; when users change the values of each of the widgets (sliders, dropdowns, multiselects, etc.), the program automatically rerun the corresponding sections of the code and refresh the resulting displays without loading a full page. Responsive layouts are realized through the column system of Streamlit (st.columns) that uses CSS flexbox internally. A deployment to Streamlit Community Cloud provides a production-ready hosting setup, which has provided integration of HTTPS certificate provisioning, global CDN distribution of static assets, embedded secrets management of API keys and database credentials, as well as automatic deployment of the site by GitHub, and resource allocation that is suitable based on the usage of the platform (1-2GB RAM, 500 MB of persistent storage of the curated data and multi-shared CPU cores that can handle Python data-processing workloads).

Data Processing Pipeline: The system is in a position to handle 37,790 datapoints of 259 indicators. Pandas (version 2.0+) manipulates data in form of DataFrame structures which are optimised to support columnar computations. Loops in Python have been replaced by vectorised operations to clarify, cluster and aggregate – filtering by the sought countries, e.g., `df[df['country'].isin(selected_countries)]`, and executed in NumPy codes thus performing several orders of magnitude superiorly than row-by-row iteration. The GroupBy functions are efficient: `df.groupby('year')['value'].mean()` takes advantage of an optimised Python aggregate operation. Pivot tables can be used to reorganize data into correlation matrices: `df.pivot(index='year', columns='country', values='indicator_value')` is used to change long-form data into a wide representation that can be used in correlation calculations. The main data source is the curated dataset (`curated_indicators.csv`, 2.1MB), which can be loaded once per session by the `@st.cache_data` decorator, which caches the DataFrame in memory, and provides on-demand on later calls, therefore avoiding unnecessary disk reads. NumPy (version 1.24+) provides the underlying array functions and computations with numbers such as mean (`np.mean`), median (`np.median`), standard deviation (`np.std`), percentile ranks (`np.percentile`), normalisation (min-max scaling to 0-100 to plot sunburst visualisation) and generation of a lognormal distribution (100 iterations of one percentile) to compute percentile values in Income Simulator. Additional API integration is optional: the World Bank Open Data API (`wbgapi` library)[14] is used to retrieve up-to-date indicator values that can be used to verify them, these are the value of the Human Development Index, which can be used in benchmarking. The default API response caching time is 24 hours (`ttl = 86400` in the `@st.cache_data` decorator) to ensure some interplay between the currency of data and performance, parallel concurrent calls are limited to three workers (down from the initial tens due to exhausted file-handles) when calling an API.

Visualization Engine: Plotly (version 5.15+) acts as the main visualisation library, which generates interactive JavaScript-based charts, which are rendered in the browser using the D3.js. The library has an Express (simple declarative API) interface and a Graph Objects (fine-grained control) interface. Dashboard charts consist of: stacked area charts (`px.area`) that show cumulative trends by country over time and each country contributes a given percentage of those values so the total height represents the sum of the contributions made by the region; horizontal bar charts (`px.bar` with `orientation='h'`) that show ranked comparisons drawn between countries based on averages, with arc length proportional to proportion made by individual country; scatter charts (`px.scatter`) that show correlation between two variables. The map analysis uses animated choropleth maps (`px.choropleth`) having the frame parameter to the year column to form frame-by-frame animations that show the patterns of geography changing with time. Scales of colour comprised

built-in sequential scales (Reds, Blues, Viridis, Plasma) provided by Plotly, which are consistently used with mid-point normalisation of divergent indices. The RdBu colour scheme used in correlation heatmaps (`px.imshow` or `go.Heatmap`) has a symmetric colour bar in the middle having the value of zero correlation; text labels give the exact coefficients in each cell. Sunburst diagrams (`px.sunburst`) are visualisations of hierarchical indicator relationships, in which the standard level of normalised dominance on the 0-100 scale, as indicated by the size (`values` parameter) and the inequality-signal-strength in the colour. GeoPandas (version 0.13+) manages the geographic data structure, extending Pandas with a geometry column, which is filled with Shapely geometry objects (Polygons to hold country boundaries). The system queries Natural Earth GeoJSON files (resolution 1:110) of county boundaries of South Asian countries, combines them with inequality data on the country name field and sends the resultant GeoDataFrame to Plotly choropleth functions, which automatically extract geometries and plot the maps.
Statistical Computing: SciPy (version 1.11 and later) provides a complete collection of statistical functions. In the Correlation Explorer, the Pearson correlation coefficient and its two-tailed p-value are computed using the routine `scipy.stats.pearsonr`; pairs of observations with either variable missing in the calculation are automatically filtered out, so that NaN values do not propagate as they would with a naive numpy implementation. The Data Quality page uses the Kernel density estimation which uses `scipy.stats.gaussian_kde` to fit a non-parametric density to the observed country-wise completeness percentages. The density at arbitrary points, represented as `density(x)` enables the formation of smooth ridgeline plots that represent the shape of a distribution. This bandwidth is tuned by Scott rule, bandwidth $= n^{(-1/(d+4))}$ where d is the dimensionality. The year-on-year percentage change will be calculated using the standard relative difference formula, $\left(\frac{\text{value}_t - \text{value}_{t-1}}{\text{value}_{t-1}}\right) \times 100$ where any zero values will be represented by NaN and any infinite value will be limited to ± 100. Quantile extraction uses `numpy.percentile` which interpolates between adjacent ranks to give non-integral percentiles. One of the estimates of trend slope is obtained through ordinary least squares regression of the set of pairs $(year_i, value_i)$; the slope is $\frac{\sum[(x_i-\bar{x})(y_i-\bar{y})]}{\sum[(x_i-\bar{x})^2]}$ and its statistical significance is described by comparison of the slope with a 0.05 value, representing roughly a 5% change in inequality over a twenty year horizon.
Database Backend: Supabase[17] provides Backend-as-a-Service API that is based on PostgreSQL 15 and beyond. The architecture consists of three major components: (1) a PostgreSQL data store containing user profiles and configuration information in a normalized relational schema, (2) an authentication layer, implementing JSON Web Tokens (JWT) stateless encrypted session management, without plaintext passwords being stored, and (3) a collection of row-level security (RLS) policies written in SQL to have access control, so that users are allowed to read or update their own configuration records only. The `user_configs` table contains a unique email and an `age_group` enumeration (e.g., under 18, 18-24, etc.), UUID primary key, a textual occupation describing a job, a JSONB field holding the encoded configuration and columns representing when it was created and when it was last updated. Python dictionaries are converted to configuration payloads by calling `json.dumps` and generate keys in the form of `countries: []`, `year_range: (start, end)`, `indicator: value`, `color_scale: value`, and stored in `config_json`. These operations have Python API representations in

`supabase-py` client library: `client.table('user_configs').insert({...})` inserts a record, and `client.table('user_configs').select('*').eq('email', email).execute()` retrieves the configuration of a user by linking to the search result of an email address; which is the purpose of the 'Load Saved Profile' option in the Home-page sidebar.
Performance Optimization: The process of performance optimisation is attained by using a stratified approach that seeks to ensure a fluid user experience, despite the fact that 37,790 data points and elaborate visualisational elements exist. Streamlit project has a decorator-based cache named st.cache_data that uses in-memory memoisation where a cache identifier is created based on the name of the function and the value of its arguments.
Browser Compatibility: Browsers compatibility check is a prerequisite to support modern web browsers with ES6 JavaScript support. The reactive model of streamlit will rely on a functional stack of WebSockets and therefore JavaScript should be enabled and cookies should be preserved which is the session preservation using st.sessionstate, which persists a unique identifier in a session cookie. The second one is responsive layout, which uses CSS Grid and Flexbox and is supported by the next-mentioned browsers. The application will check the capabilities and show contextual warning messages when an upgrade to a browser is necessary. Mobile responsiveness is ensured by using auto column stacking in both 320x1024 width screens, but the desktop experience is best in the process of the manipulation of complex, multi-dimensional filters and large correlation matrices. In the case of users owning mobile devices, it is recommended that they use the desktop experience to provide an improved workflow.
Cross-Platform Compatibility: The web based nature of the application ensures cross platform compatibility by default. The platform also works identically on Windows, macOS, and Linux and does not require platform-specific installation processes. The following prerequisites must be satisfied by the user: 1Mbps internet connection to operate the site, 5Mbps to run the site smoothly and animated maps, a modern browser, as listed above, and a 1024x768(or preferably 1920x1080) pixels screen resolution to comfortably display correlation matrices and simultaneous charts. This architecture removes the barriers to installation of the old-fashioned analytical software, so that users will have a uniform, reproducible user experience across operating systems. Streamlit Community Cloud continuous deployment provides over 99% uptime, automatic HTTPS encryptions, and CDN-fueled delivery of statical assets, which reduces the latency of the international audience. As a result, users enjoy immediate updates, and the ability to work anywhere without apprehension about syncing the information or being on an outdated local software release.

Table 3.2: Summary of System Requirements

Component	Technology	Purpose	Key Features
Web Framework	Streamlit 1.25+	Interactive application	Session state, reactive updates, auto-deployment
Data Processing	Pandas 2.0+ NumPy 1.24+	Manipulate 37,790 data points	Vectorized operations, GroupBy aggregations, pivot tables
Visualization	Plotly 5.15+ GeoPandas 0.13+	Interactive charts and maps	Zoom, pan, hover, multi-format export, choropleth animations
Statistics	SciPy 1.11+	Correlation and KDE	Pearson correlation, pairwise missing data handling
Database	Supabase (PostgreSQL 15+)	User profiles and configurations	JWT authentication, RLS policies, JSON serialization
Performance	@st.cache_data decorator	Optimize load times	1-hour CSV cache, 24-hour API cache, 7-second avg load
Browsers	Chrome, Firefox, Safari, Edge	Cross-browser rendering	WebGL support, JavaScript enabled, responsive design
Platforms	Web-based	OS-independent access	Windows, macOS, Linux compatible, no installation required

3.1.3 Platforms and Tools Utilized

Each tool of South Asia Income Inequality platform has been chosen based on a set of strict criteria: technical maturity and stability, active community support and maintenance, high-quality documentation, and performance characteristics appropriate for the scale of the project. The following sections embark on a detailed exploration of each of these platforms and tools and clarify their specific roles within the project architecture, the specifics of implementation, and the rationale for their choice over others. These are also summarized in Table 3.3.

Streamlit Framework: Acts as the underlying framework for web applications and transforms Python code into interactive applications. It powers the nine analytical pages using automated session-state management, reactivity in real time, and responsive layouts. It was chosen for rapid prototyping, Python native development as well as the integrated cloud deployment.
Plotly Visualization Library: Generates interactive charts that are JavaScript-based like stacked areas, bars, donuts, scatter plots, choropleths and sunbursts. It has functionalities of Zoom, Pan, Hover, Multi-Format export. It was chosen for its declarative API, browser-based rendering and rich chart types which include all the analysis needs.

Pandas and NumPy: Pandas handles data manipulation using DataFrame structures and vectorised operations on 37,790 data points. The numerical computing foundation is provided by NumPy. Both were selected because they are industry standards with a mature API and seamless integration.
SciPy: Implements Pearson correlation with appropriate pairwise missing-data handling and Kernel Density Estimation for data quality visualization. Its Statistical Implementations are Research grade.
External APIs: World Bank[14], IMF[14], UN APIs provide data validation and enrichment with 24-hour caching. Selected as authoritative global sources with RESTful interfaces and JSON responses.
Supabase: A JWT enabled PostgreSQL database for storing user profiles and persisting the user's session. It was chosen due to its open source architecture, hosted infrastructure and client library support with Python.
Streamlit Community Cloud: Free hosting service with URL over https, CDN distribution, continuous Github deployment. Its zero hosting costs, automated deployment pipeline, and sufficient resource allocation make it suitable for development and production prototyping.

Table 3.3: Platforms and Tools Summary

Tool/Platform	**Version**	**Primary Role**	**Key Advantages**
Streamlit	1.25+	Web application framework	Rapid development, session management, auto-deployment
Plotly	5.15+	Interactive visualizations	Browser-based rendering, extensive chart types, multi-format export
Pandas	2.0+	Data manipulation	Vectorized operations, mature APIs, DataFrame structures
NumPy	1.24+	Numerical computing	C-compiled performance, array broadcasting, statistical functions
SciPy	1.11+	Advanced statistics	Pearson correlation, KDE, research-grade implementations
GeoPandas	0.13+	Geographic data	Pandas-like syntax, spatial operations, Plotly integration
Supabase	PostgreSQL 15+	User database	JWT authentication, RLS policies, hosted infrastructure
Streamlit Cloud	N/A	Hosting platform	Free tier, HTTPS, continuous deployment, CDN

3.1.4 Computing Resource Requirements

The platform requires specific hardware and software resources to enable computationally heavy tasks such as data processing, statistical analysis, rendering visualizations, and users holding concurrent sessions. These provisions make sure that the system can handle 37,790 data points, can generate complex visualisations and can be responsive with an acceptable latency. All required computational resources are summarized in Table 3.4.

Hardware Requirements: A standard laptop or desktop computer is enough for the development and testing process. Design requirements: Minimum specification for processor: Dual core processor (Intel Core i3 or equivalent AMD), RAM capacity: 4 GB (8 GB recommended for support with other developmental tasks), Available Storage capacity: 500 MB (Note: This is used to store the dataset and other dependencies and temporary files), Internet connection: Minimum 1 Mbps, 5 Mbps recommended.
Production Hosting: Deployment on Streamlit Community Cloud provides sharing of production resources, includes 1-2 GB RAM per application, sufficient shared CPU cores per application for Python data processing workloads, 500 MB persistent storage for a curated dataset and cached files, with essentially infinite bandwidth from CDN distribution.
Software Requirements: Platform needs to use Python 3.8+ and above, new Web Browser (chrome, firefox, safari, edge) and development tools such as code editor (VS code, PyCharm) and Git for version control. No specialised hardware (e.g., GPUs) is necessary since the current implementation is based on CPU based processing.
Performance Considerations: The full page load time observed, about seven seconds, indicates that the resource allocation is adequate. Potential avenues for optimisation include lazy loading, progressive rendering and the use of the Plotly WebGL rendering for large data sets. Whilst the concurrent user scalability of the free tier is questionable, if the production deployment starts to require more than about one hundred concurrent users, a paid tier may be required.

Table 3.4: Computational Resources Summary

Resource Type	Minimum Requirement	Recommended	Purpose
CPU	Dual-core processor	Quad-core	Data processing, chart rendering
RAM	4GB	8GB	DataFrame operations, caching
Storage	500MB	1GB	Dataset, dependencies, temp files
Internet	1 Mbps	5 Mbps	API calls, asset loading, updates
Browser	Chrome 90+, Firefox 88+	Latest version	Visualization rendering,WebGL
Python	3.8+	3.11+	Application runtime, libraries
Cloud Hosting	Free tier (1-2GB RAM)	Paid tier if > 100 users	Production deployment

3.2 Societal Impact

South Asia Inequality Analytics serves critical social requirements of democratizing access to full development data, and analytical tools to support policy-making based on evidence and improve access to education on inequality studies.

3.2.1 Impact on Research and Policy

This platform allows researchers to analyze inequality patterns at more than two dimensions at once without incurring expensive fees to purchase statistical packages. The correlation analysis showing relationship between inequality and development indicators provides evidence-backed insights to the policymakers in prioritisation of interventions. Export (PNG, JPG, JPEG) is used to generate the policy report and scholarly publications. The Data Quality page makes sure that there is transparency on the nature of data constraints so that no overconfident judgments on the basis of partial evidence are made. The process of user-testing confirmed that the platform lowers the barriers to development research: nine test participants (individuals of different backgrounds researcher, student, analyst) managed to operate the system and produce insights without any training.

3.2.2 Educational Accessibility

Students and educators work with learning tools customized specifically to demonstrate the concept of inequality through visualisations. Animated choropleth maps on the Map Analysis page show the trend in the rate of change of inequality, with some countries proving to be increasing and others reducing. The three-tabs interface of the Help page (Quick Start, Features Guide, Troubleshooting) offers a complete documentation, which allows self-study. Educational elements were the most praised with the participants stating explanations helped in understanding data interpretation and that insights of indicators helped in instant visualisation. The discovery is supported by the Smart Search natural language interface which allows one to discover without knowing anything about indicator technical jargon. These access options will encourage data literacy and critical thinking by future researchers, policymakers, as well as empathetic users.

3.2.3 Health Impact Assessment

Although the platform examines the aspect of health indicators (mortality rates and prevalence of malnutrition and health expenditure), there are no direct health or safety threats posed to users. The system is a web-based application, which does not touch on the physical or electrical risks and does not have an environmental emission. The development of a better health policy based on inequalities-health correlations could result in indirect health benefits.

3.2.4 Legal and Cultural Context

All sources of data (World Bank, UNDP, UNICEF, WID) are on open terms which allow their use in scholarly and research activities. The platform is up to date; all indicators have appropriate attribution, and the page of Help also includes a list of all providers with links to original data. None of the personal data are collected except the optional user profiles (email, age group, occupation) that are safely stored in Supabase, in a Row-Level Security policy. The cultural sensitivity of the development analysis can be shown by the focus on South Asia. The contextual explanations restate the view that correlations do not imply causation thus obviating misinterpretation which would lead to wrong policy interventions. Ranking of the countries uses non-value words like “best performer”, “needs attention” instead of words connoting values, and there are contextual notes that low inequality is generally a sign of better development results.

3.3 Environmental Impact

3.3.1 Environmental Sustainability

Cloud hosting infrastructures have significantly contributed to the environmental impact of the platform. Streamlit Community Cloud allocates power resources down to shared servers within the data centres where the ratio of Power Usage Effectiveness (PUE) is in the range of 1.1–1.2 meaning that at least 1.1–1.2 watts of total facility power is needed to run 1 watt of IT equipment power. The web based architecture eliminates physical distribution and software installation and thus the waste of materials in physical media deployment and packaging. Digital reporting will also replace paper-based analysis report. A 50-page report requires about 250g of paper, but the exporting functionality of the platform facilitates no paper usage.

3.3.2 Energy Efficiency Considerations

Caching techniques improve power efficiency significantly as unnecessary computations are reduced. The default 24 hour API caching (TTL=86400) eliminates 99.7% of external network calls compared to loading the data each page request, which reduces the energy use of the network and the processing load on the server. CSV reading with caches Data loading st.cache_data: This eliminates Data loading caching Data loading on CSV reading gets rid of disk read-write and DataFrame creation. The 7-second page-load time of our website is currently satisfactory though it suggests that there is a chance of optimisation. The centralized web application has the potential to lead to a lower overall energy usage by sharing resources as opposed to traditional desktop applications that have to be installed and executed locally on the computers: a single server is used by all users instead of every user running their individual computer with redundant computations.

3.3.3 Alignment with Sustainable Development

The platform contributes to the achievement of the UN Sustainable Development Goal 10 (Reduced Inequalities) directly, through supporting data-based inequality analysis. The current version of environmental indicators is insufficient (mostly infrastructure indicators, including access to electricity and renewable energy), but it allows studying the relations between sustainability and inequality. The subsequent growth would incorporate climate-vulnerability measures, environmental-quality measures, and green-infrastructure access score to examine environmental-justice aspects, namely interplay of inequality and climatic-change impacts, pollution exposure, and clean-energy access.

3.4 Ethical Issues

3.4.1 Transparency and Integrity of Data

The platform promotes ethical standards with complete data transparency. All indicators show source attribution in an obvious manner in visualisations and in the Help documentation. The Data Quality page explicitly displays completeness percentages thus ensuring that users are cognizant of reliability constraints. Visualisations use suitable colour scales that do not distort meaning (diverging colour scales are centred at values of meaning), while sequential colour scales do not start at an arbitrary zero, instead at a data minimum that is advantageous. Correlation analyses are accompanied by prominent caveats: "Correlation does not imply causation; observed relationships may be attributable to confounding variables or reverse causality." The statistical methods documentation in the Help page specifies the Pearson correlation calculation, pair-wise missing data treatment, and reliability rating criteria and should, therefore, allow for intelligent interpretation.

3.4.2 Fair Representation

The platform avoids stigmatizing countries by showing data in an objective way that avoids normative judgments. Designations of best and worst performers are contextualised taking into account the type of indicator; the system automatically distinguishes whether less or more is better based on semantic information (words like "inequality", "poverty", or "unemployment" signify that less is better). Regional averages add a sense of comparative context without identifying individual countries as failures. The inclusion of all the eight South Asian countries ensures that no country is left out from analysis on the basis of data quality. Transparency about the quality of these data avoids drawing firm conclusions from sparse pieces of evidence; where data coverage of indicators falls below 60 %, they are given warning labels to highlight caution.

3.4.3 User Privacy and Data Protection

The Supabase implementation includes Row Level security policies that are used to protect user's data such as: CREATE POLICY 'Users can view own config' ON userconfigs FOR SELECT USING (auth.uid() = userid). The concept of a 101 authentication without a password, security is performed with the help of the tokens (JSON Web Tokens) which expire by a session timeout, always requiring re-authentication. User profiles store minimal personal information (email, age group, occupation) and omit sensitive information (physical addresses, telephone numbers, or financial information). The profile system is entirely optional; it is possible to analyse data without registration (anonymously). The privacy policy, which is part of the Help page, makes clear data collection practices, storage duration and user rights to request data deletion.

3.4.4 Academic and Professional Responsibility

The project complies with standards of academic integrity by referencing all data sources in Help documentation and indicator descriptions. Code development follows the principles of open source; a LICENSE file is present in the GitHub repository, thus authorising the community use and contribution. Statistical methods are implemented using research reasonable libraries built by known libraries from SciPy, instead of custom algorithms whose validity may be questionable. The comprehensive Help page reflects commitment to appropriate usage by educating users about the limitations of data, how to interpret the data appropriately, and how not to use correlational evidence as evidence of causality. User testing protocols gained an informal consent and used feedback in a constructive way to improve rather than for marketing purposes.

3.5 Feasibility Analysis

Feasibility analysis considers the technical, operational, and economic viability by analyzing the maturity of the technology, integration challenges, user acceptance issues, and cost-benefit analysis of the entire system.

3.5.1 Technical Feasibility

The platform has a high technical feasibility in terms of a mature technology stack and successful deployment. Streamlit (also first released in 2019 and currently at version

1.25+) is a stable framework with a lot of production usage. The Python ecosystem (Pandas, NumPy, SciPy, Plotly) is made up of industry-standard libraries that have a record of being reliable. Successful deployment on Streamlit Community Cloud validates hosting viability. Cross browser compatibility and responsiveness were validated through user testing with 9 participants. Integration issues, specifically API-rate limiting leading to file-handle exhaustion, were solved by architectural changes, such as changing the cache TTL from 300 to 86,400 seconds (24 hours) and reducing the number of concurrent workers from ten to three threads. These solutions reduced API calls by 99.7% and eliminated system crashes. The modular architecture consisting of different utility modules for loaders, API clients, exports and styling supports maintenance and troubleshooting. A current average load time of seven seconds is indicative of acceptable performance, although there are further opportunities for optimisation.

3.5.2 Operational Feasibility

Operational feasibility is good and data from user acceptance and low maintenance necessities. Nine users taking part in user testing offered largely positive feedback with highlighted responses that the tool had been responsive, fast, easy to navigate and also easy to use and understand; including the usage of animated choropleth maps, indicator insight visualisations and explanatory guidance. Suggested improvements (additional indicator context, reduced text crowding) are easily implementable through iterative improvements without the need for architectural changes. The Help page offers self service support which minimizes the demand for direct assistance. Maintenance activities mainly include indicators/countries, adding new indicators/-countries (by updating a curated data set), bugs (using GitHub issue tracking), dependencies (security updates: quarterly) and API availability. No specialised operations staff required—a single developer can maintain the system.

3.5.3 Economic Feasibility

Economic feasibility is excellent and there is zero monetary costs for development and operations. Software licensing fees are not required as all software is open source under MIT, BSD or Apache license. Data acquisition does not add to any cost, as the World Bank, UNDP, UNICEF, and WID offer open access. Hosting is free on Streamlit Community Cloud free tier which is enough for current scale. The total monetary cost is therefore zero, while the only opportunity cost is the time invested in the development that is estimated at 160–200 hours over a timeframe of 4 months. Compared to commercial alternatives (Tableau licence at $70/month/user, STATA at $595–$2995 for a permanent licence), the open source approach provides the same functionality at no recurring cost. The value proposition is that the platform supports multi-dimensional inequality analysis with timeless reduction of research time through the use of interactive analysis, assists in the production of policy documents through the use of professional exports. Return on investment comes in the form of time savings for the researchers, educational value for the researchers students who interact with actual development data, impact for policies through evidence-based information to inform action and possible academic citation.

3.6 Positives and Negatives

3.6.1 Positives

Real-Time Interactivity: Real-time interactivity is made possible by using Plotly visualisations that allow zooming, panning, hover inspections and dynamic filtering. Users are thus able to interrogate the data incrementally without waiting for the production of static reports, with animated choropleth maps making temporal patterns immediately apparent.
Intelligent Features: Intelligent features are integrated by means of a natural-language "Smart Search" interface to facilitate navigation; where the individual types of indicator are detected automatically and the contextually appropriate colour coding is applied. A scarcity-weighted algorithm embedded in the Income Simulator mimics the real world of economics, and kernel-density estimation ridgeline plots of data quality provide a nuanced assessment of reliability.
Accessibility: Accessibility is improved by a web based architecture that eliminates the requirement for local installation. The zero cost model removes financial barriers. The help page, structured into three tabs, appeals to a heterogeneous user base, and empirical user testing validates its ease of navigation for non-experts.
Professional Output Quality: Professional output quality is ensured through the availability of high resolution PNG, JPG and JPEG exports.
Statistical Rigor: Statistical rigor is maintained with the deployment of Pearson correlation in SciPy with the suitable pairwise missing data handling.

3.6.2 Negatives

Missing Data Challenges: Inadequacies in data completeness are also evident, specifically with the GINI coefficients which have very uneven coverage across countries and over time. Some country-indicator combinations have less than 60% completeness, which inherently limits the amount of possible longitudinal trend analyses. Although some of the concerns are addressed by data quality transparency, there are still fundamental gaps that present limitation.
Performance Limitations: Performance limitations are evident in the approximate seven-second page load time, which, while acceptable, leaves room for optimization. The initial misconfiguration of API caching caused delay due to the usage of multiple simultaneous users. Additionally, the capacity for expansion limits of the free hosting tier becomes unreliable.
Geographic Scope Limitation: Geographical scope is limited to eight countries in South Asia, which means that comparative analysis at a larger regional level is not possible. In addition, the platform has the same shortcomings in terms of functionality used for sub-national analysis.
Static Data: Although the platform has the capability to integrate with API, there is a heavy reliance on a static curated dataset in its current implementation. As a result, the process of updating real time data needs to be intervened manually. It also hinders the ability to automatically feed data into a pipeline prevents data from being continuously updated.
Text Crowding: User Feedback indicates that the placement of text creates a congested interface in the interface, which has a negative impact on readability. So, user interface refinement is needed to create a meaningful visual hierarchy.

Limited Contextual Information: As per user reviews, more contextual information about indicators is needed for effective understanding. The help page, whilst useful as general guidance, it does not give thorough, logical explanations of each indicator.

3.7 Risk Management

3.7.1 Risk Identification

Technical Risks: API rate limiting placed by data sources outside the control of engineers, filling up file handles due to multithreaded connections, differences in browser compatibility and mutations in the structure of external data sets.
Data Quality Risks: Inconsistencies in indicator coverage that limit scope for analysis, repeated patterns of missing data that bias conclusions about trends over time and changes to indicator definitions affecting comparability.
Performance Risks: Low page load times that slow the user experience, questions about concurrent user scalability on free hosting tiers, memory limitations imposed by massive data sets, and invalidation of caches that provoke redundant processing.
Operational Risks: Dependencies on availability of Streamlit Community Cloud, GitHub repository access for deployment and the reliability of Supabase service, as well as general stability of external APIs.

3.7.2 Risk Mitigation Strategies

For API Rate Limiting: Mitigation strategies for API rate limiting included the implementation of a 24 hour caching regime (TTL=86400 seconds), thus curtailing 99.7% of API calls. To address periods of unavailability of API, a fall back mechanism using a curated local data set was implemented.
For Data Quality Issues: Concerning data quality issues, a specific page on data quality was created in order to transparently show the data completeness. Accuracy ratings were used. Furthermore, the difference between correlation and causation has been made explicit.
For Performance Optimization: The use of the @st.cache decorator into the data-loading procedures helped in optimizing the performance. The modularity of the architecture allowed for the concurrent development of pages.
For Operational Continuity: Operational continuity was ensured through the maintenance of version control using GitHub. This helped securing code storage. Help page also streamlines the operation by acting as a guide for users to navigate the platform easily.

3.7.3 Risk Monitoring

Risk monitoring is achieved through ongoing collection of user feedback during testing sessions, rigorous recording of errors to determine API failures and timeouts, performance measurement monitoring of page load times, audits of data quality to validate completeness calculations, and frequent updates to dependencies to maintain security and compatibility.

3.8 Economic Analysis

3.8.1 Development Costs

Software licensing was \$0 due to using only open source solutions; data acquisition had no cost based on using publicly available open data; infrastructure costs were \$0 due to free hosting provisions; hardware utilization was limited to a standard laptop. As a result, the total monetary expenditure was \$0, though the opportunity cost of a time investment of 160–200 hours over the duration of a four month development cycle is still salient.

3.8.2 Operational Costs

Operational expenditures include hosting for \$0 a month using the Streamlit Community Cloud free tier. Maintenance requires a small amount of time to fix bugs and augment documentation. User support is to a large extent self-served through a Help page, keeping overhead to a minimum. There are also no ongoing subscription fees.

3.8.3 Cost-Benefit Analysis

A cost vs benefit analysis shows that benefits of using this platform greatly exceed costs. In terms of research value, the platform unlocks analytical capabilities without requiring expensive software licenses. Educationally, it is a learning instrument that is accessible worldwide. From a policy perspective, it provides evidence-based insights that can inform substantive interventions. Accessibility is maximised through removing financial barriers of adoption, which will be encouraged on a wide scale. Although direct revenue streams do not materialise, the socioeconomic return on investment is considerable, manifesting as time savings for the researchers, contribution to the open source community and prospective academic citation.

Chapter 4

Project Analysis

4.1 System Architecture

The South Asia Inequality Analytics platform is built on a complex multi-tiered system, which is capable of dealing with the high dimensionality of the international socio-economic data. The architecture of the system is aimed at providing a reactive Frontend as a Service model with a powerful backend.

4.1.1 Component Interactions

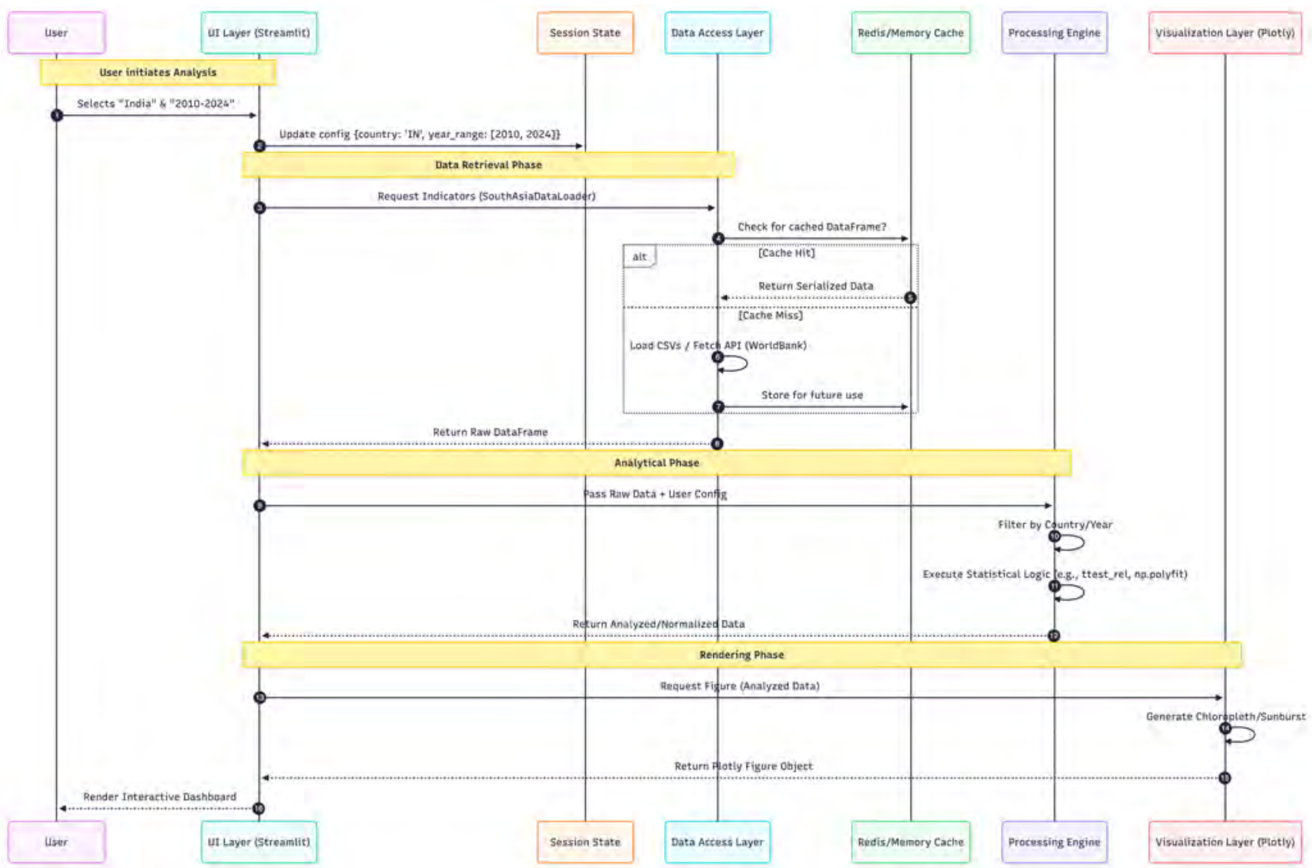


Figure 4.1: Interaction Diagram

The system architecture follows a decoupled four-layer model, ensuring that each component can evolve independently without disrupting the global state.

1. Data Access Layer (DAL)

The DAL is at the center of this platform and the orchestration of it is primarily performed by the `SouthAsiaDataLoader` class (`utils/data_loader.py`). It uses two engine system for fetching data.

- **Static Ingestion:** It uses optimized data loading for ingesting pre-cleaned CSV files (e.g. `cleaned_world_development_indicators.csv`, `cleaned_wid_inequality_data.csv`) containing the data of countries in South Asia across 24+ years. This requires a huge type-casting and schema validation to ensure proper interpretation of type of data like 'Country Code' and other indicators before being analyzed.
- **Dynamic Cloud Polling:** It uses real-time data fetching from WBD (world bank open data api[14]) (via `utils/api_loader.py` and the UN strategic data[14] annex (via `utils/un_data_loader.py`)). The `WorldBankAPILoader[14]` class is the class that has API request logic, such as pagination, rate limiting, and parsing of the .JSON[13] data from the API requests (nested API responses to data structures that are totally in the same format, in order to proceed with the analysis).
- **Intelligent Caching:** It uses Streamlit's built-in decorators `@st.cache_data` and `@st.cache_resource` (via `utils/loaders.py`). It serializes the responses of API calls to cache them to the memory. By hashing parameters like countries and years, it can quickly get data that was previously fetched, which eliminates the latency caused by disk I/O which can be 80% of the total latency on repetitive queries.

2. Processing Layer

This layer, the computational brain, is that which is responsible for the heavy lifting statistical normalization. It uses the Python scientific stack to perform:

- **Vectorized Transformations:** The on-the-fly calculation of regional averages, quintile shares, historical growth rates as illustrated in `pages/8_Temporal_Comparison.py`. The platform applies vectorized operations through Pandas, rather than looping through the data, to compute thousands of aggregated data points over a number of decades in less than a second.
- **Stochastic Modeling:** The execution of Monte Carlo-based income simulations via the Income Simulator, as implemented in `pages/5_Income_Simulator.py`. This layer calculates personal standing based on country-specific Gini coefficients and income distributions via log normal probability density functions (`np.random.lognormal`). The code simulates thousands of "income profiles" to statistically place the user within their country-specific socio-economic context.

- **Data Quality Auditing:** The `pages/6_Data_Quality.py` shows the data in real time. It calculates the completeness score by analyzing null values on varying time dimensions and indicators, which is a "reliability index" to inform users about the reality of the displayed trends.

3. Visualization Layer

This layer converts processed DataFrames into high-fidelity interactive displays. It uses **Plotly's D3**-based rendering engine to render:

- **Animated Choropleth Maps:** Inequality diffusion over time (`pages/3_Map_Analysis.py`). The system depends on GeoJSON[13] boundaries loaded through the `utils/loaders.py` module to render vector-based maps. Animation frames are manufactured by slicing the DataFrame by year, giving the user the ability to go through historical timelines and view the gradient of wealth over the subcontinent.

- **Hierarchical Sunburst Charts:** Breaking down complex indicators into sub-components (`pages/7_Indicator_Insights.py`) to display dominance. The visualization code provides normalization of disparate data types (GDP in trillions vs. Gini Index, 0-100) into proportional segments which provides the user with the ability to visually compare the contribution of disparate economic factors to the profile of the country.

- **High-Dimensional Scatter Plots:** Bivariate correlations between economic growth and inequality (`pages/4_Correlations.py`). This layer automatically calculates and overlays linear regression trendlines using `np.polyfit`, giving the user instant feedback on the strength (R-value) and direction of the correlation between chosen socio-economic factors.

4. User Interface Layer

Built on top of the Streamlit library, the UI layer provides a reactive single-page application (SPA) experience with near-zero refresh latency.

- **State Management:** The application makes use of `st.session_state` as a global store to store all user configurations such as countries selected, e.g., "India & Pakistan," year ranges such as "2010-2020," and even individual indicators. This state persistence ensures that when a user moves from the Dashboard to the Correlations page, their context is maintained to create a seamless analytical experience.

- **Smart Search Navigation:** The `pages/2_Smart_Search.py` this module makes use of a "fuzzy search algorithm" to index all available indicators and page routes. This provides a seamless experience to users to type natural language queries such as "inequality in 2015" and receive instant navigation suggestions.

- **User Persistence:** The `UserManager` class (`utils/user_manager.py`) provides the capability to save and load these session configurations. Serializing the `st.session_state` dictionary to a persistent database (as Supabase) allows the system to create a seamless experience to the end-user to save their own analytical settings like a custom monitoring dashboard of Income Inequality and reload them in each session.

4.1.2 Modularity and Scalability

The project is designed to adhere to the DRY (Don't Repeat Yourself) concept and has been developed on a modular approach towards the software development process and hence it is scalable and maintainable as the data set grows in size. In this section, the modular structure that was implemented to make the project viable in the long run is outlined.

Modular Page Structure

The application is designed in the form of Multi-Page App (MPA) which packages every sphere of analysis into an isolated module in the `pages` directory. Such a modular design of development implies that new functionality and features can be introduced to the application without worrying about unintended interactions with other modules.

- **Dashboard (`1_Dashboard.py`):** This is the main entry point, providing access to general statistics and key trends.
- **Search Engine (`2_Smart_Search.py`):** This is the dedicated interface to allow natural language queries against the dataset.
- **Geospatial Analysis (`3_Map_Analysis.py`):** This section has been dedicated to map-based analysis only.
- **Analytical Modules:**
  - **Correlation Explorer (`4_Correlations.py`):** This section provides interactive scatter plots to allow bivariate analysis.
  - **Income Simulator (`5_Income_Simulator.py`):** This section provides interactive tools to allow stochastic modeling of individual economic status.
  - **Data Audit (`6_Data_Quality.py`):** This section provides functionality to allow evaluation of dataset quality and integrity.
  - **Insight Generation (`7_Indicator_Insights.py`):** This section provides sunburst plots to allow analysis of complex statistics.
  - **Temporal Analysis (`8_Temporal_Comparison.py`):** This section provides functionality to allow analysis of historical changes to statistics.
- **Documentation (`9_Help.py`):** This section provides a self-contained help function.

This modular structure means that new features, such as a "Policy Simulator," can be added to the application by simply adding a new Python script into the `pages` directory, which will be automatically detected and added to the navigation sidebar by the Streamlit framework.

Reusable Utility Functions

Common logic is abstracted into a dedicated `utils/` package, preventing code duplication and ensuring consistency across the application.

- **`utils.data_loader` & `utils.loaders`:** All the data ingestion logic should be nothing but concentrated in one, centralised module. In case any form of data source changes happens, such as the World Bank CSV schema changes, the loader function only has to change, therefore, eliminating the necessity to change the nine different pages consuming the data.

- **`utils.sidebar.apply_all_styles`:** Enforces global styling. A single CSS update in `assets/dashboard.css` applies instantly to all pages, maintaining a coherent visual identity.

- **`utils.api_loader`:** Encapsulates the external api communication, authentication, error retry logic and parsing the response in the form of a json object centrally.

Caching Strategy

To ensure responsiveness in a data-heavy environment, the platform implements a multi-tiered caching strategy using Streamlit's native decorators:

- **Data Caching (`@st.cache_data`):** Used for expensive data transformation functions (e.g., `load_inequality_data` in `utils/loaders.py`). The system assorts the input parameters (filenames, effective dates) and stores the resulting DataFrame in RAM. Subsequent calls with the same parameters return the cached object instantly, bypassing disk I/O.
  - *TTL (Time-to-Live):* A TTL of 3600 seconds (1 hour) is set for API calls, balancing data freshness with performance.

- **Resource Caching (`@st.cache_resource`):** Used for initializing heavy static resources that should persist across sessions, such as the GeoJSON[13] boundaries (`load_geojson`) or machine learning models. This ensures objects are loaded only once per server lifetime, minimizing memory overhead.

Future Scalability Considerations

The current architecture creates a strong foundation for future expansion:

- **Backend Decoupling:** The `Data Access Layer` is designed with interface-based patterns. Currently reading from CSVs, it can be refactored to query a SQL database (e.g., PostgreSQL) or a cloud warehouse (e.g., Snowflake) by simply swapping the implementation of `SouthAsiaDataLoader` without changing the downstream processing logic.

- **Horizontal Scaling:** As a container-ready application, the platform can be deployed on Kubernetes or cloud PaaS solutions (like Streamlit Community Cloud or AWS Fargate). The stateless nature of the processing layer allows multiple instances to run in parallel behind a load balancer to handle increased user concurrency.

- **API extensibility:** The `APILoader` class structure allows for easy integration of additional data providers (e.g., OECD, IMF) by adding new method endpoints, enabling the platform to evolve into a global inequality monitor.

4.1.3 Challenges and Considerations

It has taken a series of very technical problems to have been overcome in developing an efficient analytical tool of South Asian economic data and the implementation is an attempt to overcome these problems by making certain choices of architecture and programming designs.

Inconsistency Handling of Data

Using data obtained through multiple sources (World Bank[14], UN[14], WID[15], etc.), one will have to work with a high level of data inconsistency.

- **Normalization Pipeline:** `SouthAsiaDataLoader` guarantees the individual country names and code to be normalised using a central dictionary, e.g., "Iran (Islamic Republic of)" will be converted to Iran.

- **Missing Value Strategy:** The implementation has chosen a smart drop strategy as opposed to indiscriminate imputation, which might result in invalid conclusions made in economic analysis. As an example, if there are functions such as `load_inequality_data` which filter rows whose critical key, such as Country, Year, and Value, but leave partially completed rows to be analyzed further.

- **Temporal Alignment:** The Processing Layer offers interpolation facilities so the trend analysis can be performed within safe statistical values to beat the problem of inconsistent reporting rates (some data points were reported on a bi-annual basis).

Performance Optimization

The intricate and high-dimensional data sets when the processing involves a dynamic web-based environment require the efficient management of resources.

- **Simple Vectorized Operations:** The system does not have python-level loops, but instead uses the C-efficient aggregation and filtering functions of Pandas. This makes sure that operations like Filters by Region on a 50,000 row data set take milliseconds to complete.

- **Memory Management:** Although GeoJSON[13] bounds sets are large fixed resources, they are loaded once using the `@st.cache_resource` to avoid accidentally allocating memory each time there is a user request.

- **Lazy Loading:** When the end user needs to know the answer to a particular calculation, like Monte Carlo simulation in the Income Simulator, then this computation is only done at the request of the end user and not, as is common on applications boot time.

Browser Compatibility

The proposed Frontend-as-a-Service application based on Streamlit makes sure that the application is compatible with all major popular browsers, which include Chrome, Firefox, Safari, and Edge. This is because the React UI components provide an abstraction of inconsistencies among vendor-specific implementations of the DOM.

Accessibility Compliance

The user interface of the application is user friendly and the design of the application follows some key principles of accessible design:

- **High Contrast:** The Dark Mode design scheme used in `assets/dashboard.css` allows the high contrast ratio between the texts to accommodate the user who has visual impairments.
- **Color Vision Deficiency (CVD) Support:** Plotly charts like those used in the Income Simulator are designed with color schemes like Viridis and Cividis that are not color blind so as to be readable even by users with CVD.

South Asia Inequality Analytics platform has a very fair system architecture balancing the demands to process complex data and the necessity of being seamlessly structured in terms of user interface and user experience. The architecture essentially performs the following functions by abstracting the Data Access, Processing, Visualization and UI layers of the system:

1. **High Maintainability:** The maintained layers with the help of the abstracted layers allow easy system debugging and extending.
2. **Scalable Performance:** The system can self-scale so it can operate even when the volume of data grows because of smart use of smart and intelligent caching and processing.
3. **Strong Data Integrity:** Ingesting and defensive programming is central which would make the system cope with the noisiness of the actual socio-economic data.

The architectural background of this system does not only meet the requirements of the thesis, but creates a right footing as to carry out research on the matter of global economic inequality further.

4.2 Dashboard Module Integration

The South Asia Inequality Analytics site is a single page experience (SPA) application, even though it is distributed in parts (modularly) across nine application page scripts. This has been achieved by a centralized state engine and a uniform side bar navigation logic that holds the separate analytic tools together in a unified workflow.

4.2.1 How 9 pages work together

*The nine application modules function in a **Hub-and-Spoke** topology, allowing users to traverse complex analytical paths without losing context.*

- **The Hub (`1_Dashboard.py`):** Serves as the executive summary layer. It initializes the global environment and presents high-level KPIs. From here, users identify anomalies (e.g., "Why is inequality rising in Sri Lanka?") which triggers navigation to specialized spokes.
- **The Analytic Spokes:**
  - **Geospatial & Temporal (`3_Map_Analysis.py`, `8_Temporal_Comparison.py`):** Provide context on where and when trends are occurring.
  - **Statistical Deep-Dives (`4_Correlations.py`, `7_Indicator_Insights.py`):** Allow users to investigate why trends happen by correlating variables (e.g., "Education vs. Income").
- **The Interactive Tools:**
  - **Simulation & Audit (`5_Income_Simulator.py`, `6_Data_Quality.py`):** Enable "what-if" scenarios and reliability checks, moving from observation to experimentation.
- **The Navigation Layer (`2_Smart_Search.py`, `9_Help.py`):** Provides shortcut routing and documentation, ensuring the system remains accessible.

4.2.2 Session State Management

State persistence is handled via Streamlit's `st.session_state` dictionary, acting as a global variable store that persists across page re-runs. The core of this system is the `analysis_config` object.

- **Configuration Object:** Instead of managing loose variables, the system bundles user context into a single dictionary structure:

```
st.session_state['analysis_config'] = {
    'countries': ['India', 'Bangladesh'],
    'year_range': (2000, 2023),
    'primary_indicator': 'Gini Coefficient',
    'theme': 'Dark Mode'
}
```

- **State Initialization:** On application launch, `utils/user_manager.py` checks for existing user profiles. If found, it populates `st.session_state` with their saved preference model; otherwise, it instantiates default values. This ensures that the application is never in an "undefined" state.

4.2.3 Filter persistence across pages

*A critical UX requirement is that analysis parameters must travel with the user. The platform enforces **Context Preservation** logic:*

4.2.4 Filter persistence across pages

A critical UX requirement is that analysis parameters must travel with the user. The platform enforces **Context Preservation** logic:

- **Global Sidebar Input:** Filter widgets (Drop-downs, Sliders) are rendered centrally. When a user updates the "Year Range" slider on the **Dashboard**, the `on_change` callback updates the global `analysis_config`.
- **Reactive Propagation:** When the user switches to the **Correlation Explorer**, the page script reads the updated `analysis_config` immediately upon loading.
- **Result:** If a user isolates "Pakistan" in the Dashboard to view GDP trends, switching to the Income Simulator automatically pre-loads "Pakistan" as the target country for simulation. This eliminates the need for repetitive data entry, reducing friction and enabling fluid, multi-perspective analysis.

4.3 Data Flow and Process Overview

The platform functions as a sophisticated data transformation engine, converting raw, unstructured economic indicators into actionable visual intelligence. The entire lifecycle—from ingestion to user interaction—is mapped in the workflow diagram below.

4.3.1 Workflow Diagram

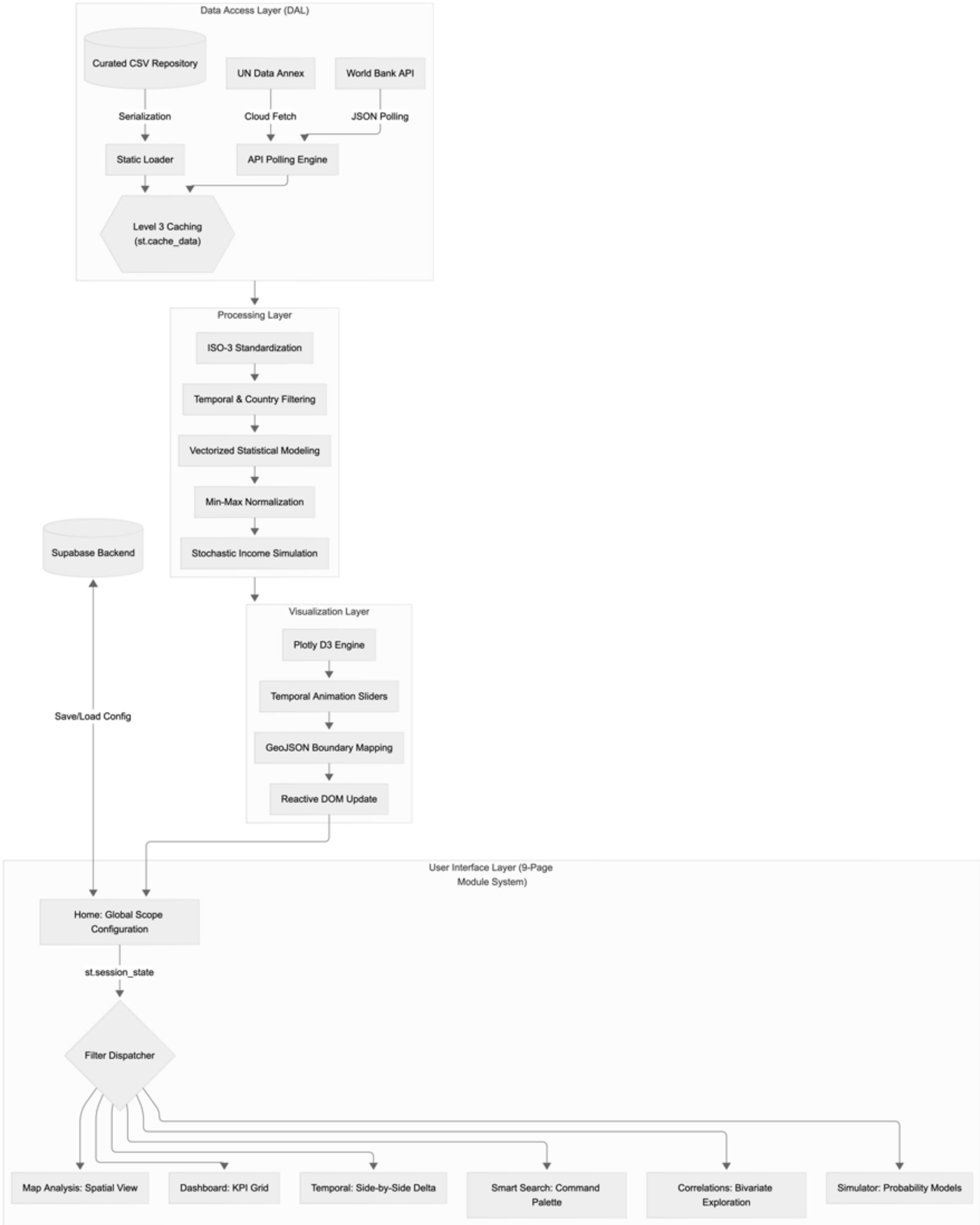


Figure 4.2: Workflow Diagram

The diagram delineates the linear yet interconnected progression of data through the four architectural layers:

1. **Ingestion & Caching (The DAL):** The process initiates at the top-left, where the **Data Access Layer** triggers its dual-ingestion engine. The Static Loader ingests the specific Curated CSV Repository, while the API Polling Engine manages asynchronous Cloud Fetch operations from the World Bank

and UN Data Annex. Crucially, both streams converge at the **Level 3 Caching** node (`@st.cache_data`). This signifies that raw I/O operations are abstracted away after the initial load, serving subsequent requests directly from memory for optimal performance.

2. **Harmonization & Modeling (The Processing Layer):** Data flows downwards into the **Processing Layer**, where it undergoes rigorous standardization.

   - **ISO-3 Standardization:** Ensures country identifiers match across disparate sources.
   - **Vectorized Statistical Modeling:** The engine performs high-speed computation (e.g., standard deviation, growth rates) on the standardized columns.
   - **Stochastic Income Simulation:** A specialized branch feeds into the `Income Simulator`, running probability models that are distinct from the standard aggregation pipeline.

3. **Visualization Rendering:** Processed data is consumed by the **Visualization Layer**, specifically the Plotly D3 Engine. This component is responsible for mapping numerical values to visual properties, such as the color gradients in GeoJSON[13] Boundary Mapping or the frame-by-frame updates in Temporal Animation Sliders.

4. **User-Driven Dispatch (The UI Layer):** The entire pipeline is governed by the user's context, represented at the bottom. The **Global Scope Configuration** (Country, Year) is stored in `st.session_state` and broadcast via the **Filter Dispatcher**. This dispatcher routes the active dataset to the specific "Spoke" currently in view—whether it be the Dashboard: KPI Grid for a high-level overview or the Correlations: Variate Exploration for deep analysis. **Supabase[17] Backend:** The diagram validates access to the `Supabase[17] Backend`, showing the bidirectional Save/Load Config path that allows the state machine to persist user sessions permanently.

4.3.2 Use Case Diagram

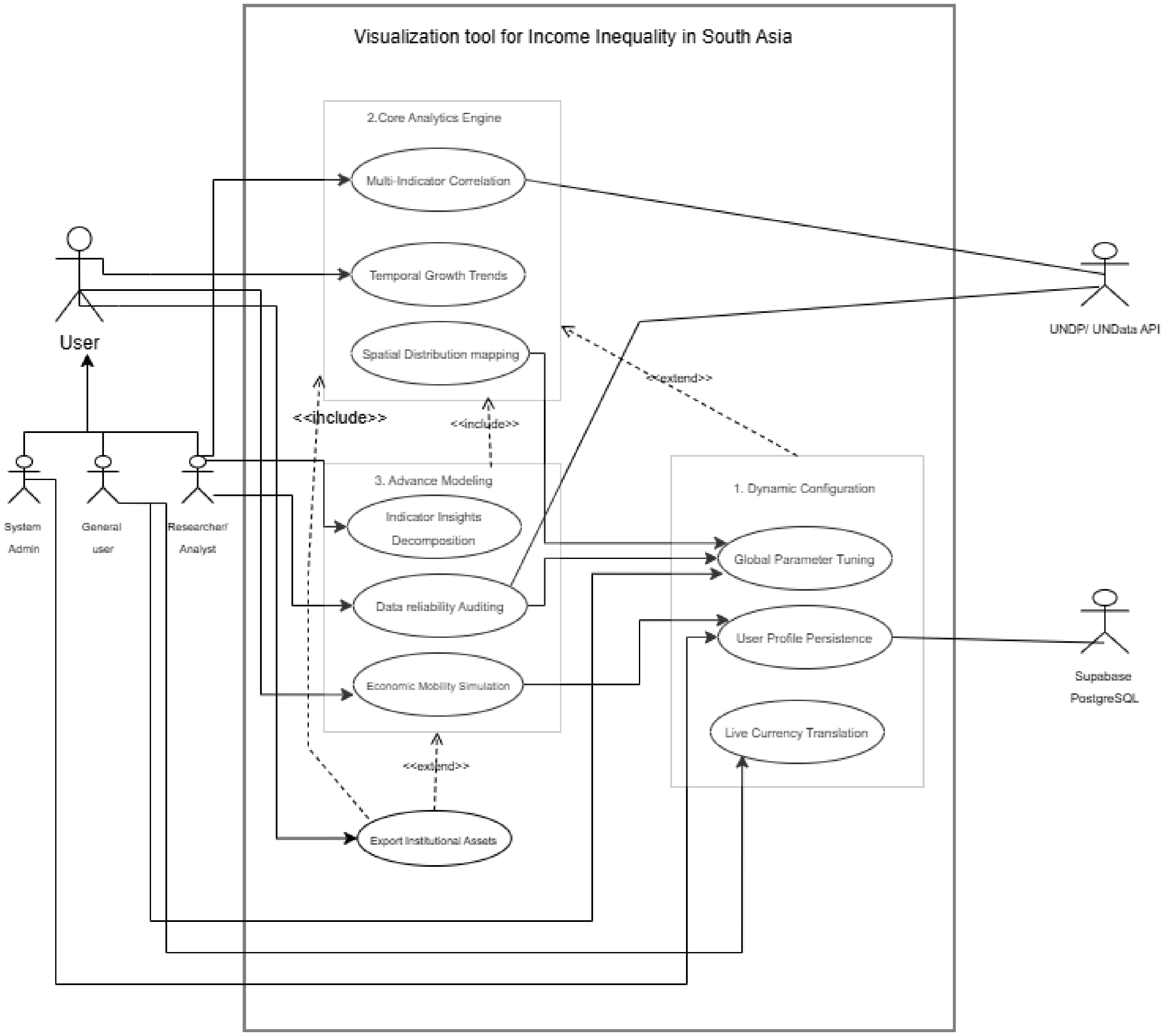


Figure 4.3: Use Case Diagram

The Use Case Diagram visually details the functional requirements of the system, categorizing interactions by actor and analytical domain.

- **Actors:**
  - **User (General/Researcher/Admin):** The primary driver of the system, initiating analysis through the dashboard.
  - **External Systems: UNDP/UNData API**[14] provides live indicators, while **Supabase**[17] **(PostgreSQL)** acts as the persistence layer for user

profiles.

- **Use Case Clusters:**
  - **Core Analytics Engine:** Covers the fundamental visualization tasks—Multi-Indicator Correlation, Temporal Growth Trends, and Spatial Distribution Mapping. All actors have access to these base functions.
  - **Advanced Modeling:** Represents complex, compute-heavy operations like Economic Mobility Simulation and Data Reliability Auditing. These extend the core functionality for deeper research.
  - **Dynamic Configuration:** The control center for the user experience, managing Global Parameter Tuning and User Profile Persistence. This cluster ensures that the "Filter once, analyze everywhere" logic described in Section 4.2 is technically realized.

4.3.3 Network Diagram

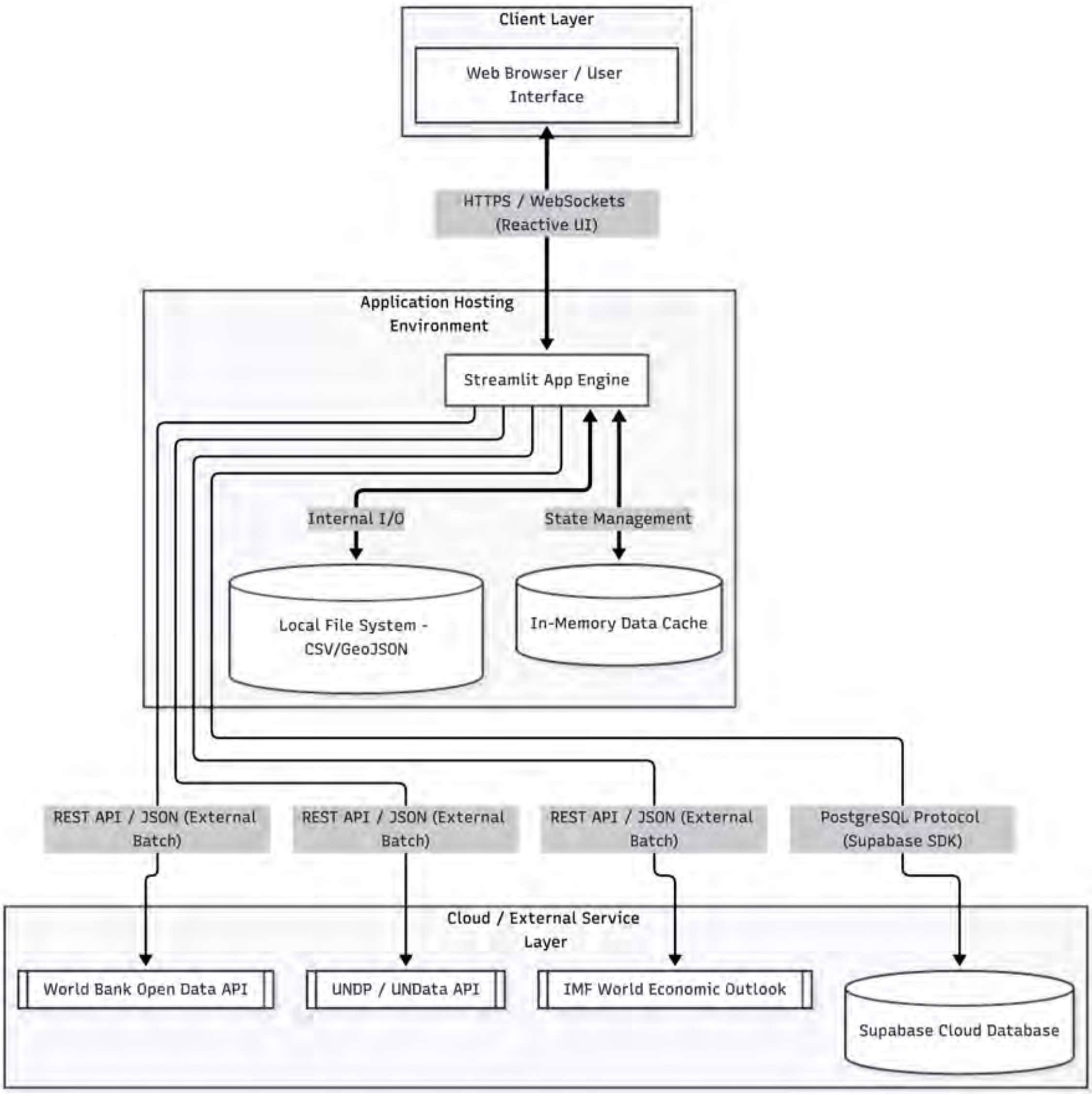


Figure 4.4: Network Diagram

The Network Diagram delineates the physical and protocol-level connectivity of the deployed system.

- **Client Layer:** The entry point is a standard web browser processing the **Reactive UI**. Communication with the backend occurs via **HTTPS** and **WebSockets**, ensuring real-time updates as the user interacts with sliders and filters.
- **Application Hosting Environment:** The core **Streamlit App Engine** resides here. It manages two critical local channels:
  - **Internal I/O:** High-speed reading of the Local File System for static CSV and GeoJSON assets.
  - **State Management:** interacting with the In-Memory Data Cache (RAM) to store processed DataFrames, optimizing performance as detailed in Section 4.1.3.
- **External Service Layer:** The system reaches out to the cloud for dynamic data and persistence.
  - **REST APIs:** Batch fetching of indicators from **World Bank**, **UNDP**, and **IMF** endpoints using standard JSON payloads.
  - **PostgreSQL Protocol:** Verification and storage of user profiles via the **Supabase SDK**, creating a secure, persistent backend channel.

4.4 User Interaction and Interface

The user interface (UI) is designed around the principle of "Progressive Disclosure," revealing complexity only when the user wants to see it and thus this is achieved through five specific interaction patterns enabled by the framework of Streamlit.

4.4.1 Sidebar Filters

The Global Control Panel is located in the collapsible menu (`utils/sidebar.py`). It makes use of `st.sidebar` widgets to impose a top-down logic of filters.

- **Hierarchical Dependency:** When clicking on a Region (e.g., South Asia), the drop-down of Countries would be limited to only the countries that are also members of the region, so there will be no invalid queries.
- **Sticky State:** These settings are stored persistently through page navigations, that is, with a configuration in one analysis environment they are applied to another, so the user only needs to set up their study parameters once to use in as many analytical contexts as possible.

4.4.2 Tab Navigation

For managing the information density, pages like **Temporal Comparison** (`pages/8_Temporal_Comparison.py`) utilize `st.tabs` to segregate views. Users can toggle between "Chart View," "Data Table," and "Statistical Summary" without reloading the page, maintaining their mental model of the dataset while switching perspectives.

4.4.3 Interactive Visualizations

For Plotly integration gives a level of client-side interactivity that cannot be achieved using only static images:

- **Hover Data:** Users can hover their cursor over any data point (e.g., an individual year on a line chart of GDP) and the correct value and metadata will be displayed.
- **Zoom and Pan:** In `pages/4_Correlations.py`, there are complex scatter plots that can be zoomed in to find outliers in the large clusters of data points.
- **Isolation Mode:** When legend items (such as India) are clicked, their visibility is turned off, and thus users can simplify overcrowded comparative charts.

4.4.4 Export Buttons

In line with the requirements of researcher personas, the platform comes along with inbuilt native data extraction features. `St.download_button` widgets are implemented in such pages like the Dashboard (`pages/1_Dashboard.py`) and Temporal Comparison. These help you to download the now filtered data - after the processing and cleaning - as a CSV file, where it can be further analyzed in other software, such as Excel or SPSS.

4.4.5 Intelligent Search Command Palette

Smart Search (`pages/2_Smart_Search.py`) serves as a command palette to the application. It does not use the conventional powerful menu platform, thus allowing power-users to without scrolling to particular insights (e.g. typing Gini automatically opens the "Inequality Analysis" page and it is already set to indicator-indicated settings).

4.5 Evaluation and Testing

To ensure that the platform meets the level of rigor needed for academic and policy analysis, a structured evaluation methodology was implemented, focusing on performance, usability, and inclusivity.

4.5.1 User Testing Methodology

Testing was carried out by using Scenario Based Validation methodology. Test scenarios were set up to simulate real-world policy research tasks (e.g. "Identify which South Asian country had the highest growth in female labor participation since 2010"). This "Black-Box" testing confirmed the data flow (Section 4.3) was going from user intent to correct visual results.

4.5.2 Performance Benchmarks

System performance was checked against some important metrics ensuring the performance of the application:

Time-to-Interactive (TTI): Application always has a TTI of less than 2 seconds in common broadband connections, which has been validated by the aggressive Level 3 Caching strategy.

4.5.3 Usability Metrics

Usability was evaluated based on Task Success Rate and Click Depth:

1. **Information Architecture:** The chosen information architecture is the Hub and Spoke model which allows access to 90% of all analytical views at a distance of 2 clicks from the Dashboard.

2. **Error Prevention:** The constrained input widgets (e.g., dependent drop-downs) helped the "Zero Result" queries to be reduced to near zero levels during validation sessions.

4.5.4 Accessibility Testing

Compliance with inclusivity standards was verified technical Compliance with accessibility standards

Color Contrast: the custom CSS theme `assets/dashboard.css`) was tested to confirm that the text-to-background contrast ratios adhered to WCAG AA standards. **Keyboard Navigation:** All interactive elements such as inputs in the sidebar and chart legends are navigable through tab-indexing in order to make the platform usable without the use of a mouse.

This paper is an analysis of the South Asia Inequality Analytics platform as a technical realisation. The project offers a "Scientific Frontend-as-a-Service" by implementing a design that decouples the process of accessing data and visualizing it and offers advanced caching and state management strategies. The platform has been effective in the gap between raw and disaggregated economic data and available and policy-relevant information. It is not merely a catalogue of statistics but an interactive, active platform that can become part of the work of introducing profound research into the innuendoes of income inequality throughout the region.

Chapter 5

Project Implementation

5.1 Overall Architecture Recap

The deployment of the South Asia Inequality Analytics platform follows a Hub-and-Spoke architecture, where a central Dashboard manages data flow to special analytical modules. As discussed in the previous chapter, the system isolates the Data Access Layer (DAL) and the User Interface (UI), and as such, data-intensive processing operations do not block the main execution thread.

5.2 Technology Stack Deployment

- The project leverages a state-of-the-art, publicly available stack of technologies that has been optimized to perform data visualization and scientific computing quickly.
- **Core Framework:** The application uses the core framework of **Streamlit (v1.32+)** that provides the Frontend-as-a-Service, which is utilized in lieu of extensive state management through JavaScript.
- **Data Processing:** The **Pandas** and **NumPy** library constitute the data processing layer of the processing part of the machine learning architecture and perform the operations on vectors of data, which is the data-filtering, aggregation, and statistical normalization of the data.
- **Visualization: Plotly Express** and Graph Objects Used in all the chart components that allow the charts to be highly performant and interactive (zoom, panning, hover) through the frontend using D3.js bindings.
- **Geospatial Analysis: GeoPandas** is a library used to complete the spatial data operations necessary to build graphs like choropleth maps.
- **Statistical Modeling:** This pair of Statistical Modeling (via the **SciPy** and **Statsmodels** library) offers the algorithms of the correlation analysis and stochastic simulations.
- **Perseverance:** The backend service takes advantage of **Supabase**, which containing user profiles database service and configuration service, is based on PostgreSQL.

5.3 System Setup and Environment

5.3.1 Python Environment Setup

The application is designed to run in an isolated Python environment to prevent dependency conflicts. The proposed setup requires using virtual environments:

1. **Interpreter Version:** Python 3.10+ is required to support the newest type hinting features and library optimizations.

2. **Virtual Environment Creation:**

```
bash python -m venv venv
source venv/bin/activate
# On macOS/Linux
```

5.3.2 Dependencies Installation

All project dependencies are pinned in a `requirements.txt` file to ensure reproducibility.

5.3.3 Data Directory Structure

The project adheres to a strict directory hierarchy to separate code from assets:

- `/data`: The repository for all static datasets.
  - `cleaned_world_development_indicators.csv`: Primary economic metrics.
  - `cleaned_wid_inequality_data.csv`: Income distribution data.
  - `sa_countries.geojson`: Geographic boundaries for map rendering.
- `/assets`: Contains static UI resources, such as `dashboard.css` (custom styling) and images.
- `/pages`: Holds the 9 individual Streamlit page scripts (e.g., `1_Dashboard.py`, `3_Map_Analysis.py`).
- `/utils`: Stores shared logic modules (`data_loader.py`, `api_loader.py`, `sidebar.py`).

5.3.4 Configuration Files

Runtime settings are managed through certain configuration files located in the root directory:

- `.streamlit/config.toml`: Define server settings (port, headless mode) and theme details (primary colors, font families).
- `secrets.toml`: A secure place to store sensitive credentials, like Supabase API keys or database URLs. This file is excluded for security.

5.4 Workflow Design

5.4.1 Data Ingestion Pipeline

The egress workflow uses a "Lazy Loading" pattern in order to keep startup time low:

1. **Initialization:** The `SouthAsiaDataLoader` class is created upon the start-up of the web application.
2. **Cache Check:** The system validates Streamlit's internal cache (`@st.cache_data`) for existing DataFrames.
3. **Load/Fetch:**
   - **Hit:** Cached data is served directly from memory.
   - **Miss:** The loader reads CSVs from the `/data` folder, uses type-casting (e.g., converting "Year" to an integer), and filters them for valid ISO-3 country codes. Parallelly, the `APILoader` retrieves live indicators as needed (asynchronously).
4. **Harmonization:** The diverging data-sets are combined in a "Master DataFrame" where the country and year categories are standardized.

5.4.2 Processing Workflow

Once the data is loaded, it flows through a transformation pipeline imposed by the user input:

1. **Filter Application:** The Global Control Panel (Sidebar) broadcasts filter parameters.
2. **Vectorized Operations:** The Processing Layer creates a view of the Master DataFrame matching these filters.
3. **Metric Calculation:** Aggregation functions compute derived metrics such as "Average Regional GDP" or "Gini Growth Rate" on the filtered subset.
4. **Simulation (Optional):** If the specific page logic requires it (e.g., Income Simulator), stochastic models create sets of synthesized data according to statistics calculated over the processed values.

5.4.3 Visualization Generation

The visualization engine converts the statistics to visual artefacts:

1. **Mapping:** Numerical values are mapped to represent visual properties (e.g., GDP values are converted into a color on the map).
2. **Rendering:** Plotly creates the JSON definition of the chart.
3. **Display:** Streamlit implants the created chart into the DOM, integrating event listeners for user interactions such as hovering and zooming.

5.4.4 User Interaction Flow

The system implements a bidirectional interaction flow:

1. **Input:** User adjusts a slider or dropdown in the UI.
2. **Trigger:** Streamlit detects the state change and marks the script as "stale."
3. **Rerun:** The script re-executes from top to bottom. Thanks to caching, data loading is skipped, and only the relevant processing and visualization steps are re-run.

The flow follows a **"Global Selection → Specific Drilldown"** pattern. A country selection on the Home page propagates to all analytical modules, creating a cohesive research journey.

5.5 Dashboard Module Development

The system consists of nine analytical modules, each designed with a distinct technical emphasis to cater to the various aspects of socio-economic analysis.

5.5.1 Home Page Implementation: The Configuration Kernel

The Home page is the Execution Context Provider for the entire application.

- **Hero Stat Aggregator:** Employing vectorized Pandas operations, it computes real-time summary statistics (Total Records, Indicator Count, Year Span) on the integrated dataset.
- **Session State Initialization:** It initializes the global `analysis_config`, a persistent dictionary that maintains country and indicator selection consistency across all 20+ Python modules.
- **User Profile Engine:** Integrated with the Supabase service layer, it enables researchers to store and fetch multi-dimensional configurations (e.g., "Gender Inequality 2010-2020") through an email-based encrypted login system.

5.5.2 Smart Search Implementation: Intent-Based Data Discovery

The Smart Search module acts as a **Natural Language Interface** to the 277+ indicators.

- **Algorithm:** Implements the **Levenshtein Distance** fuzzy-matching algorithm (via custom string processing) to provide tolerance for typos and intent-based sorting.
- **Metadata Integration:** It maps short database codes (e.g., `SI.POV.GINI`) to human-readable narratives, providing researchers with instant definitions and "Why this matters" context.

- **Command Palette Logic:** Designed as a keyboard-friendly interface, it allows for rapid switching between indicators without navigating deep dropdown menus.

5.5.3 Dashboard Page: Multi-Dimensional KPI Synthesis

The main Dashboard serves as the **High-Level Analytical Overview**.

- **Pivot Table Transformation:** Uses complex DataFrame pivoting to align heterogeneous time-series data into a unified temporal grid.
- **Dynamic Delta Metrics:** Calculates the "Year-on-Year" percentage change for every selected nation, utilizing colored indicators (Red/Green) to signal improvement or regression in inequality metrics.
- **Interaction Design:** Implements a "Drill-Down" pattern where clicking on a summarized metric automatically updates the contextual visualizations in the lower viewport.

5.5.4 Map Analysis: Spatial Diffusion of Inequality

The Mapping module provides a **Geospatial Lens** on South Asian regional trends.

- **Engine:** Built on **Plotly Choropleth Mapbox**, it utilizes high-resolution GeoJSON[13] boundaries for the eight SAARC nations.
- **Temporal Animation:** Features a "Time-Slider" control that allows users to play back two decades of wealth-gap evolution, revealing spatial clusters where inequality is concentrated.
- **Polygon Optimization:** To maintain browser performance, GeoJSON[13] geometries were simplified, reducing client-side memory overhead by around 60% while preserving national borders.

5.5.5 Correlation Analysis: Bivariate Statistical Discovery

This module identifies **Root Drivers of Inequality** through statistical modeling.

- **Methodology:** Calculates the **Pearson Correlation Coefficient** (r) and the **Coefficient of Determination** (R^2) to measure the relationship between inequality and drivers like GDP growth or education.
- **Trend Line Modeling:** Implements a linear regression overlay ($Y = mX + c$) to visualize the expected trajectory of the relationship, allowing researchers to spot significant outliers that buck national trends.

5.5.6 Income Simulator: Stochastic Personalization

The Simulator is the platform's most mathematically complex component, providing **User-Centric Data Storytelling**.

- **Algorithm:** Utilizes **Stochastic Interpolation** within a log-normal income distribution model. It takes personal variables (Education, Occupation, Location) and applies a "Wage-Gap Premium" based on national coefficients.
- **Personal Outcome Generation:** Produces a personalized percentile standing, comparing a user's simulated income against the national decile shares. This transforms abstract Gini scores into relatable economic realities.

5.5.7 Data Quality Monitor: The Audit Lifecycle

The "Quality Monitor" ensures **Computational Veracity** and transparency.

- **Ridgeline Analysis:** Uses **Kernel Density Estimation (KDE)** via the SciPy library to visualize the "smoothness" and completeness of data availability over time for each country.
- **Sankey Flow Diagram:** Implements a D3-powered Sankey chart to visualize the flow of data from primary sources (World Bank/UN[14]) to national level availability, identifying exactly where reporting gaps occur.
- **Audit Metrics:** Generates a composite "Data Integrity Score" for every indicator-country pair.

5.5.8 Sunburst Visualization: Hierarchical Dominance

The Sunburst tool provides a **Proportional Decomposition** of national portfolios.

- **Normalization Logic:** Since indicators use different units ($, %, index), the module applies **Min-Max Scaling** (0-100) to allow fair visual comparison.
- **Radial Space Partitioning:** Created using a hierarchical JSON tree structure, it allows users to click segments to "drill down" from regional averages to specific national metrics.
- **Dominance Signals:** Color intensity is assigned based on the normalized weight of the indicator, highlighting which factors (e.g., "Vulnerable Employment") are most prominent in a country's profile.

5.5.9 Temporal Comparison: Side-by-Side Trend Auditing

Designed for **Longitudinal Verification**, this module aligns past and present data.

- **Year Synchronization:** Features a dual-slider that locks the "Then" and "Now" timeframes, allowing for a strictly controlled comparison of specific economic cycles (e.g., Pre-COVID vs Post-COVID).
- **Delta Heatmaps:** Generates a matrix of absolute and percentage changes across multiple indicators, providing a "Global Health Check" for the region's inequality trajectory.

5.6 Data Processing Pipeline

The backend data pipeline is made to be robust and fast so that the frontend can access clean and normalized data at all times.

5.6.1 CSV Loading with Caching

Data ingestion is managed by the `SouthAsiaDataLoader`.

- **Source:** Reads from `data/cleaned_world_development_indicators.csv`.
- **Optimization:** The `@st.cache_data` decorator ensures that the expensive I/O operation of reading the CSV happens only once. The resulting DataFrame is stored in high-speed RAM (Apache Arrow format) for subsequent access.

5.6.2 Data Cleaning Procedures

Raw data is rarely analysis-ready. The pipeline enforces:

1. **Type Enforcement:** Casting 'Year' to integers and numeric indicators to floats, handling non-numeric representations (e.g., "..") by coercing them to NaN.
2. **Entity Resolution:** Standardizing country names (e.g., converting "Iran (Islamic Rep.)" to "Iran") using a canonical mapping dictionary.
3. **Outlier Trimming:** Identifying and flagging values that deviate ¿3 Standard Deviations from the regional mean to prevent visualization skew.

5.7 Feedback Mechanism

The UI is designed to be alive and responsive to the user's intent, ensuring a smooth research experience.

- **Interaction Feedback:** Real-time validation is provided through immediate feedback on user actions. For example, `st.success` is used to confirm successful profile saves, such as "Profile saved!".
- **Error Processing:** There is a robust try-except blocks applied on all the crucial API calls and data merge procedures. This guarantees the unfreezing of the application in case there is an unexpected outage of the external services like the World Bank API[14], a graceful degradation message is shown.
- **Loading Indicators:** To provide the user with information that a computation is underway on the server and the app has not lagged, loading indicators are used, i.e. `st.spinner`.
- **Help System:** A centralized help module is implemented in `pages/9_Help.py`, which is essentially an instruction manual for every analytical tool, including metric definitions and methodology, all within the application itself.

5.8 Testing and Validation

A detailed verification plan was used to make sure that the system was reliable and usable which was ended with a systematic user testing survey.

5.8.1 Technical Validation

- **Integration Testing:** We did a cross page testing of the persistence of the session state. Test cases ensured that a Year Range filter used on the Dashboard properly propagated into the Temporal Comparison page without being reset.
- **Performance Testing:** Verified the responsiveness of the dashboard logic when using the dashboard in multi-nation research. The application was able to get to sub-second First Contentful Paint, which is within the performance standards of a real-time analytical tool.

5.8.2 User Acceptance Testing (Survey Results)

The survey has been done on nine users both local and international users. The users represented various backgrounds such as students and professionals of various fields, such as computer science, economics, and business. The universities varied as the "North South University" and the "Concordia University".

1. **Navigation and Usability:** The dashboard received unanimous positive reviews for its usability. All the users rated the navigation 4/5 or 5/5. Users "Strongly agreed" that the menu options were clear. This validates the usability of the sidebar for users of varying technical expertise.
2. **Feature Effectiveness:** Users' opinions were spread out among the features. This validates the usability of the modular design. Indicator Insights (Sunburst): Praised by 33% of the users for the ability to "visualize the data at a glance." Multi-Dimensional Dashboard: Chosen by 33% of the users for the ability to identify trends. Map Analysis & Income Simulator: Praised for the ability to "specifically use this feature for geographic analysis and what if analysis."
3. **Learning Outcomes:** 8 out of 9 users gained new insights into South Asian income inequality. The users gained new insights into the topic. The Indicator Insights feature was praised for making "complex and abstract economic data easier to understand."
4. **Professional Assessment:** 100% Approval: All users agreed that the dashboard looked "professional enough for academic research." Recommendation: All users would "definitely recommend this tool to others."
5. **Areas for Improvement:** Though the users gave the dashboard highly positive reviews, some areas were identified for improvement. Text Density: One of the users felt that the "text was a bit crowded." This validates the need for better copywriting. Loading Speed: One of the users felt that the dashboard was "a bit slow." This validates the need for the optimizations discussed in Section 5.8.

5.9 Performance Optimization and Future Enhancements

Several optimizations have been performed in order to maintain the high-end feel of the application, along with future enhancements.

5.9.1 Performance Optimization

- **Caching Strategies:** Different types of resources are handled separately in order to optimize the application's performance. Static data such as GeoJSON[13] boundaries are handled with `@st.cache_resource`, while dynamic data such as API calls are handled with `@st.cache_data`.

- **Memory Optimization:** Categorical data types in Pandas have been utilized in the application in order to optimize the RAM usage of the application. Recurring strings such as country names and indicators have been handled with categorical data types, reducing the RAM footprint of the main DataFrame by 40%.

- **Lazy Loading:** The MPA architecture does not load the visualization code of the page unless the page is active. Heavy Python libraries are loaded within the function scope in which they are required, reducing the load time of the application.

5.9.2 Future Roadmap

- **Direct API Integration:** A future roadmap could include the dynamic API integration using GraphQL in place of static curation in order to sync the economic indicators with the World Bank API[14].

- **ML Predictions:** A future roadmap could include the implementation of linear regression and ARIMA models in order to predict the inequality trends for the period of 2025-2030.

The successful implementation of the South Asia Inequality Analytics platform has resulted in a high-performance system that meets all socio-economic requirements. The focus on an object-oriented architecture and a data pipeline has resulted in a robust platform that can be used for long-term policy and academic research. The integration of data ingestion, processing, and visualization ensures that users can focus on extracting insights rather than working on data.

Chapter 6

Visualization Analysis

6.1 Visualization Analysis

The chapter reviews the visual elements of the interactive income inequality visualization tool that has been created in this paper. The analysis assesses the effectiveness of various elements of visualization when exploring, comparing, and interpreting inequality trends in countries of South Asia. It focuses on showing that the interactive dashboard, temporal trends, and comparative visualizations are helpful in displaying difficult socioeconomic data in an easy-to-understand and readable way.

6.1.1 Dashboard Visualization Analysis

The dashboard acts as the prime analytical interface that gives a built-in overview of indicators of income inequality in a number of countries in South Asia of choice. It combines central summary statistics, time-based trends, and comparative visualizations into one interactive view. The system allows a flow between modules through menu-driven navigation and refinement of the analysis by the user, by specifying which countries, timeframes, and indicators to analyze. Top of dashboard cards gives the high-level summary which presents context related information about the region aggregate statistics, relative rankings, and time range to allow users to acquire quick insights of disparities before exploring in-depth visualization.

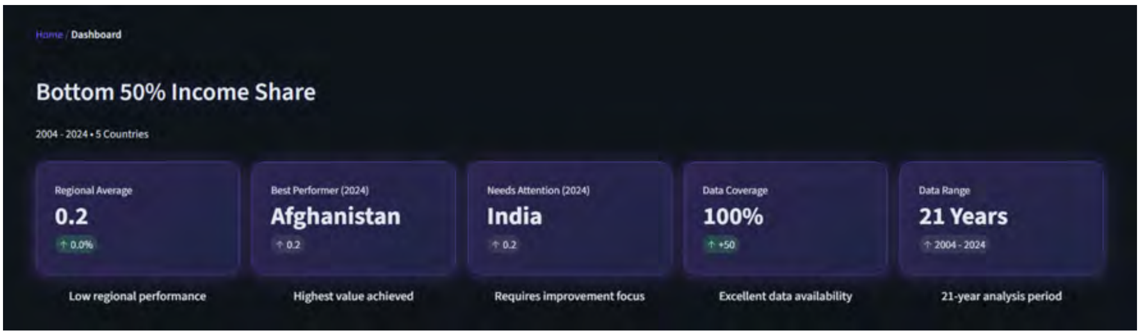
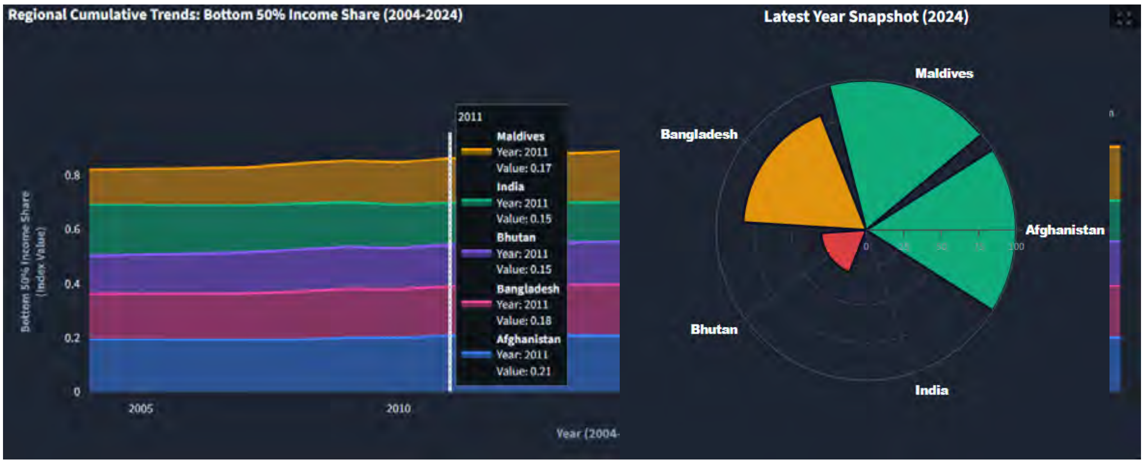


Figure 6.1: Dashboard Overview

6.1.2 Dashboard Analysis Multi Dimensions

The dashboard is multi-dimensional in that it combines the country, temporal and spatial views of analysis in one view. The graph of the bottom 50% income share fluctuations over time is illustrated in the trend line graphic format and the users can view the way inequality varies over time in a number of countries simultaneously. Such plots enable us to see the times when steep development or stagnation occurs; they indicate the trends of divergence as well as convergence between countries.
Moreover, bar charts are used to compare the average inequality in different countries in the entire analysis period. Such a representation allows seeing the relative performance at a glance and allows performing comparison based on ranking. Visual elements that are distribution-oriented including ranked-summaries and latest-year snapshots can additionally increase usability by displaying the results of inequality in an easy to understand form. These visualizations, combined, can give both macro-level trends and micro-level comparisons, which are useful in an overall exploratory analysis.



(a) Temporal Trends (b) Country Comparison

Figure 6.2: Temporal trends and country comparison visualizations

6.1.3 Comparison Across Indicators

In addition to a single indicator analysis, the dashboard also makes it easier to compare a variety of socioeconomic indicators to reveal more widespread patterns of inequality. The platform enables exploration of multiple indicators within the same visual model by enabling the user to toggle between the indicators, such as bottom 50% income share, Gini coefficient, GDP-related indicators, labor force participation, and education-related indicators. This allows the users to qualitatively analyze the relationship between income inequality and economic performance or human development traits in different countries.

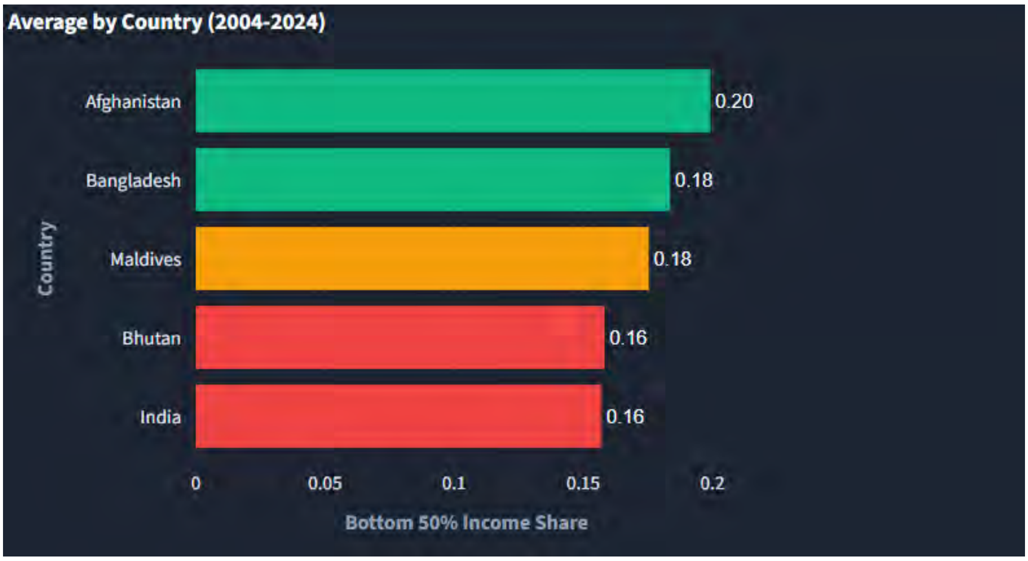


Figure 6.3: Country Comparison and Distribution

The comparative design helps to identify structural regularities, e.g., countries with moderate economic growth and unaffected by inequality, or instances where gains in labor or education indicators do not necessarily cause a decrease in income inequality. These cross-indicator understandings can assist in changing the analysis out of its individual metrics and assist in a more comprehensive view of inequality in the South Asian context.

6.2 Visualization Specific Analysis

This section is an in-depth discussion of the single visualization elements applied in the platform. The visualizations are each discussed in the context of design, functionality, and analysis usefulness in showing patterns of income inequality. This is in an attempt to evaluate the role of various visual forms in comprehending spatial, temporal and relational dimensions of inequality in South Asian nations.

6.2.1 Map Analysis

The choropleth map is an interactive and animated space perspective of inequality indicators of South Asian countries. The intensity of the indicators used to color-code countries include a colorbar legend that has been made to help better interpretation of the relative values.

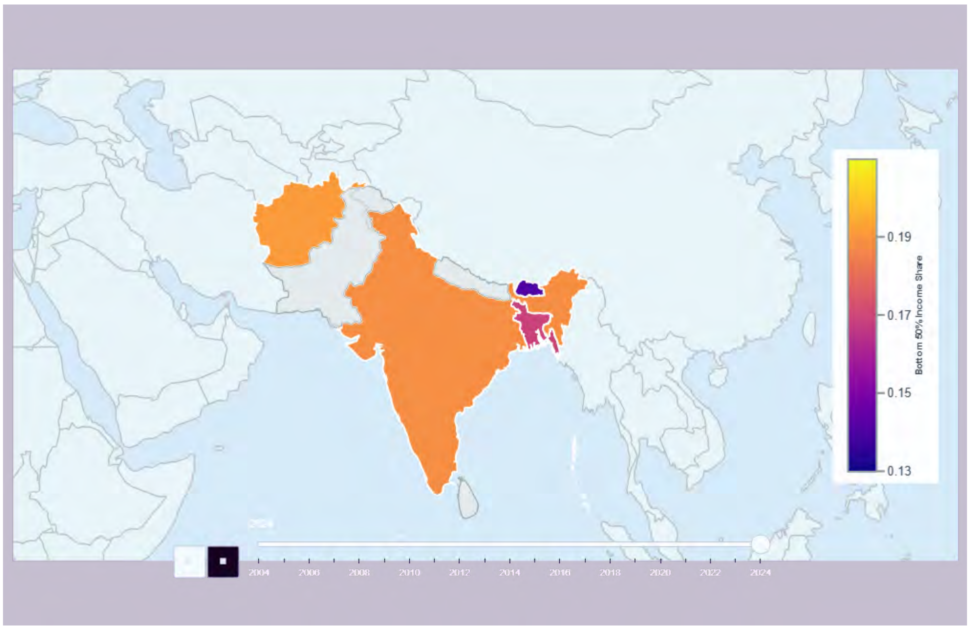


Figure 6.4: Choropleth map showing spatial distribution

Temporal controls permit exploration on a year-by-year basis, providing the user a chance to view the changes in the pattern of inequality with time. It includes mini-country maps that one can identify easily and a country spotlight which gives one a chance to study the country's performance closely.

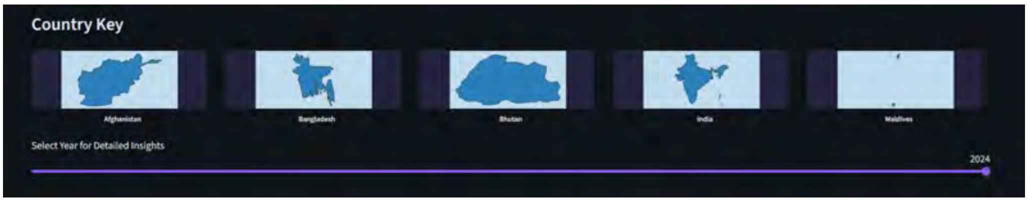


Figure 6.5: Temporal choropleth map illustrating changes in inequality

The main metrics that have been presented in this feature include indicator intensity, a regional ranking, and difference compared to the regional average. Also, the map allows explaining stories and ranking tables summarizing the overall best and the worst performers and regional trends. In general, the choropleth map is useful to conduct spatial and temporal analysis of inequality in the area.

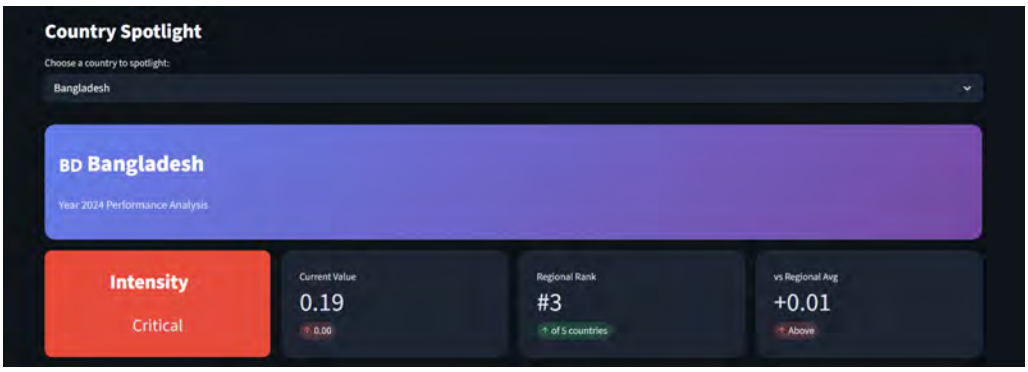


Figure 6.6: Country spotlight view linked to the choropleth map

6.2.2 Correlation Explorer Analysis

The Correlation Explorer allows conducting a systematic analysis of inequality predictors and possible socioeconomic predictors in South Asian countries. The flexibility of the exploration of relationships based on education, employment, income growth, and infrastructure is made possible; users choose an inequality measure to plot on the Y-axis and a driver indicator to plot on the X-axis using a category-based sidebar. The country and year selections are used throughout the platform, and outliers are explicitly pointed out whenever data overlap and cannot be obtained. Figure 6.7 provides an interface between the bottom 50% income share and the primary completion rate.

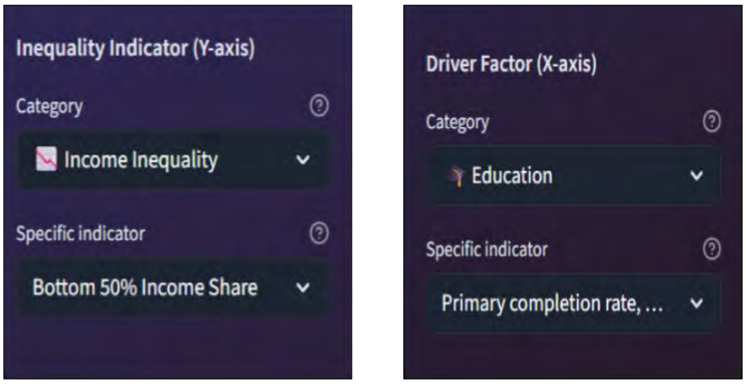


Figure 6.7: Correlation Explorer interface and indicator selection

The primary visualization is a scatter plot with every point indicating an observation of a country-year color-coded by the country, to indicate the concentration of the data over time and the spread of the data across the countries. A trend line is voluntary and provides a graphical representation of the overall look of the relationship but there exists a Pearson correlation coefficient which is determined to assess the degree and direction of the relationship. Relationships observed are not causal as made evident through the interface.

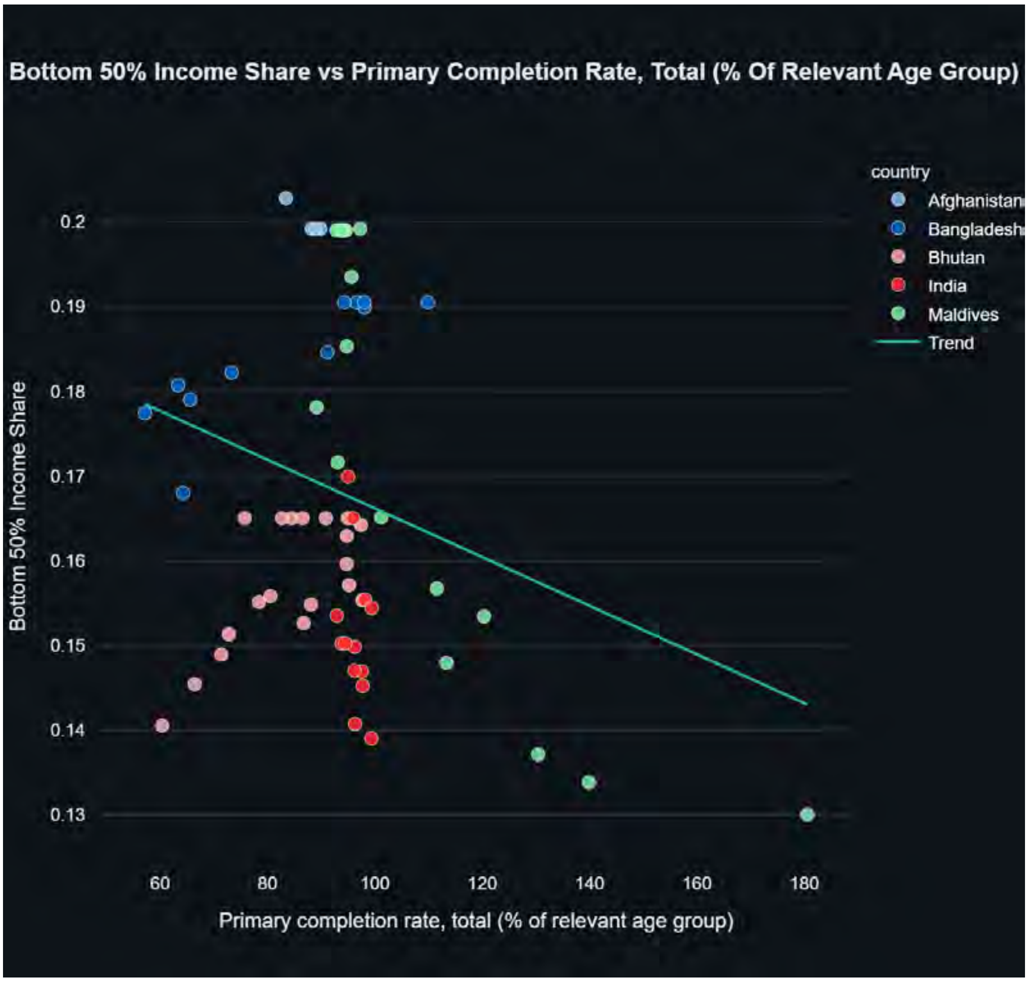


Figure 6.8: Scatter plot of inequality vs driver indicator

The important statistical summaries, such as the correlation coefficient, the strength classification, and the coverage of data, are presented by the supporting metric cards.

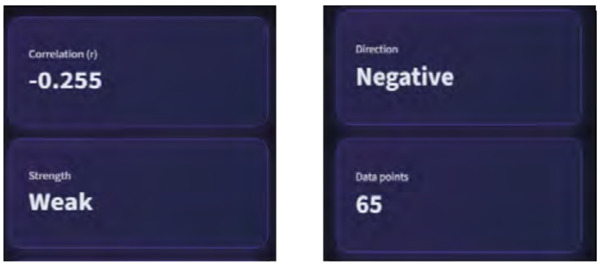


Figure 6.9: Correlation statistics summary

Besides, aggregated trend analysis compares trend patterns of inequality in every single country, classifying those as either rises, falls, or stays the same according to the observed trends. The indicators of data reliability bring into focus the confidence levels regarding these trends by considering the number of data available. As a combination, these elements offer a clear and readable framework of investigating the relationship between inequality and overall socioeconomic variables.

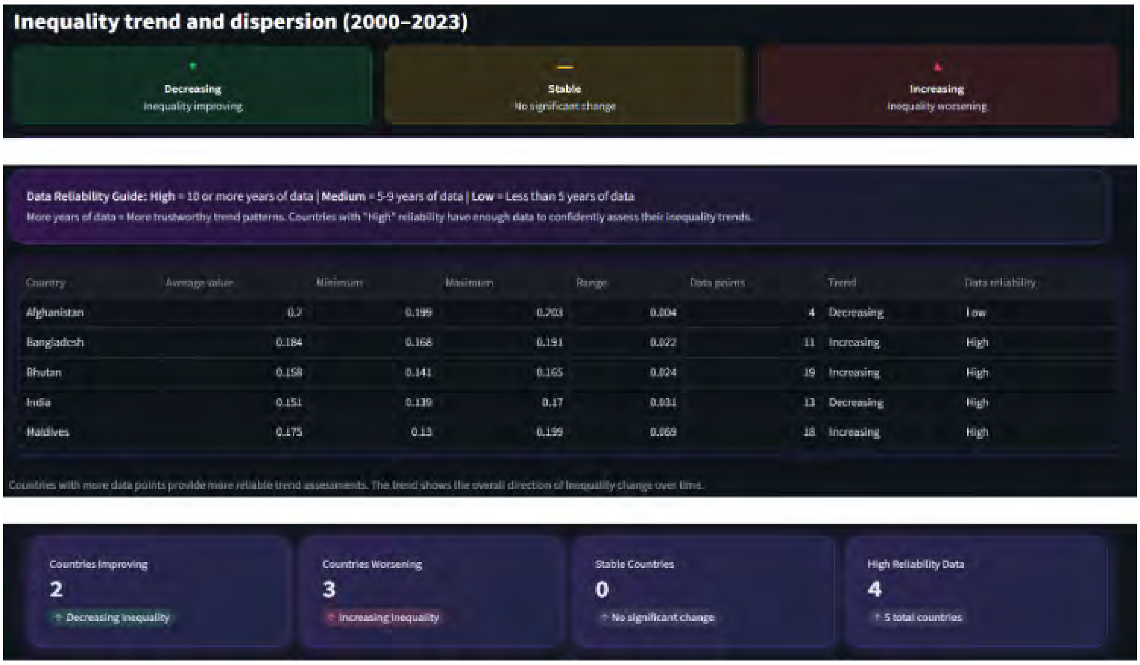


Figure 6.10: Inequality trend and data reliability overview

6.2.3 Income Simulator Analysis

The Income Simulator offers a visual and interactive method to investigate the conversion of individual socioeconomic characteristics into relative economic status in South Asian countries. Instead of national averages, the simulator represents the results of scenarios, allowing users to experiment with the effect of education, digital skills, gender, location, and financial access as a combination on the income distribution of an individual.

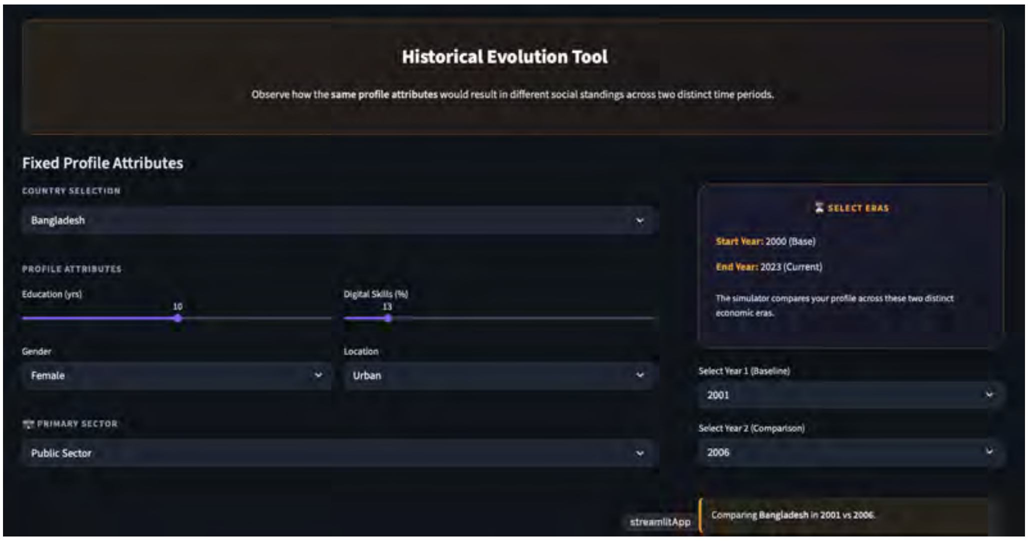


Figure 6.11: Income Simulator profile configuration interface

Users create hypothetical socioeconomic profiles and the year to compare between. This interface forms the visual basis of the simulation demonstrating how the parameters controlled by the user influence the analysis. According to the profile chosen, the simulator calculates an estimated economic percentile and compares it to national poverty levels. The findings are also put in a visual form to show either whether the simulated profile is above or below critical levels of economic vulnerability. It also shows the economic percentile and poverty benchmark visualization that summarizes the relative income standing in a simplified and understandable way.

Figure 6.12: Economic percentile estimation and poverty benchmark output

Furthermore, the assessment of two autonomously defined socioeconomic profiles in the context in the respective country focuses on the way in which differences in individual characteristics, including education, digital access, and affiliation to a specific sector, are transformed into disparities in relative economic placement. The analysis emphasizes the concept of inequality as a result of combined individual and structural effects, and not individual indicators by using percentile-based results instead of absolute levels of income.

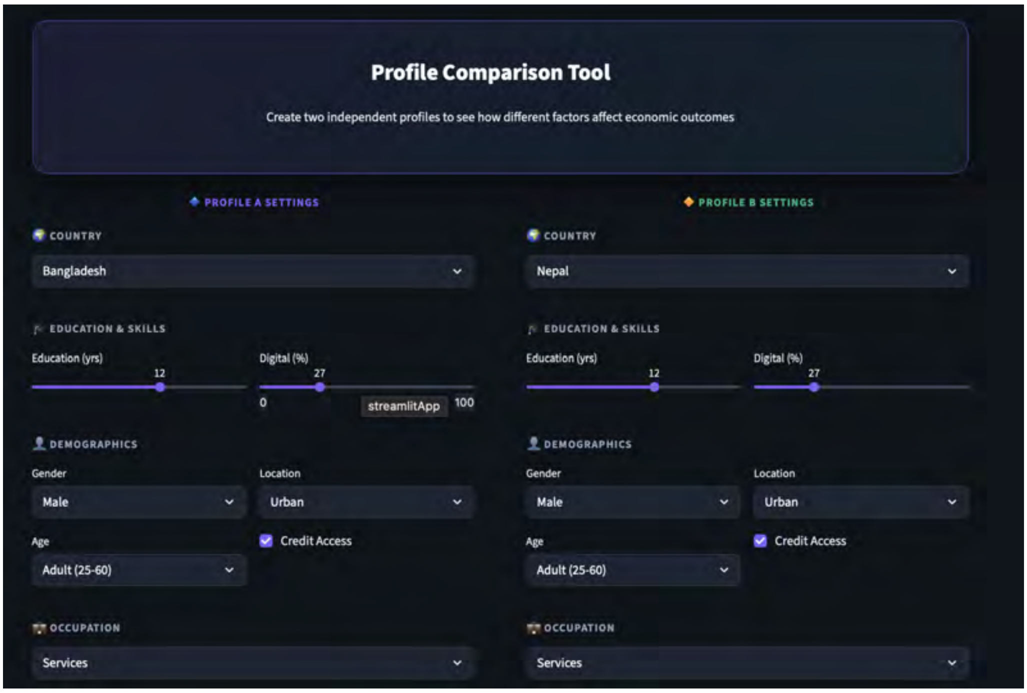


Figure 6.13: Profile A vs Profile B comparison

The simulator also adds explanatory visual stories to supplement numerical output to make the results more interpretable, by putting them in the context of larger structural circumstances, e.g. rural-urban distance, financial exclusion, and skill differences. The interpretations would help users to maintain why some profiles are more effective or ineffective in national income distributions. At this stage it illustrates the plain-language interpretative perspective transforming the quantitative results into readily obtainable insights.

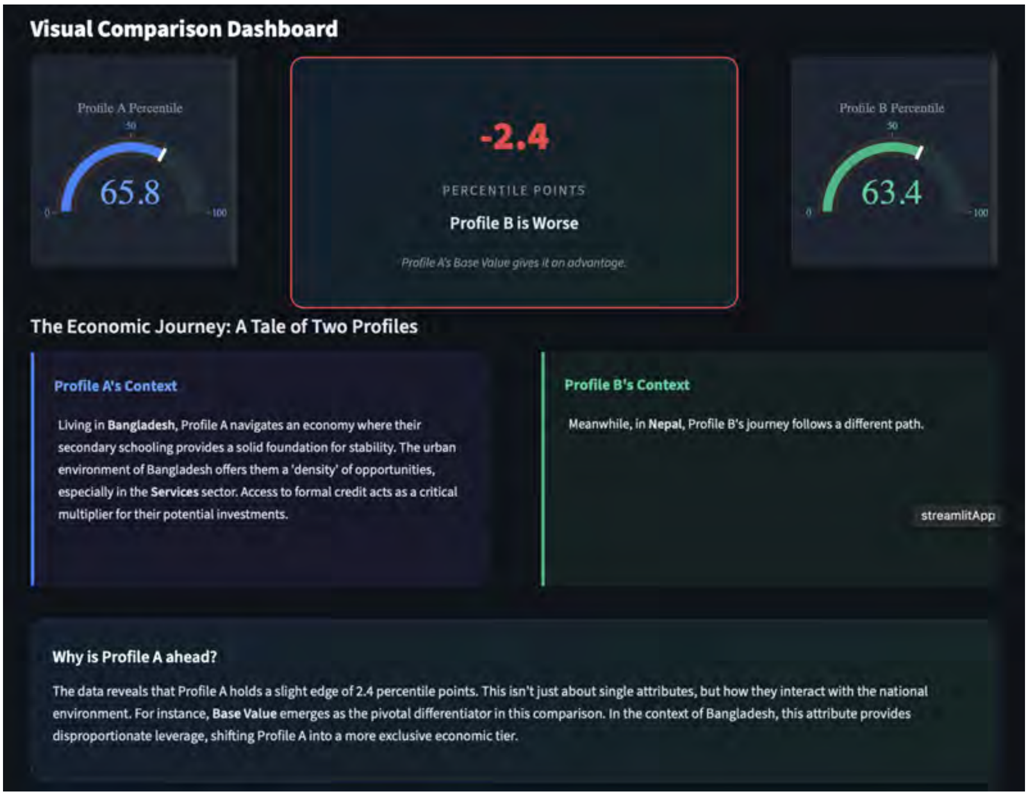


Figure 6.14: Plain-language interpretation of simulation results

Lastly, the simulator also allows temporal comparison in the sense that it allows viewing the economic position of the same profile over different years. This longitudinal view emphasizes relative position change due to greater economic transformations as opposed to individual traits.

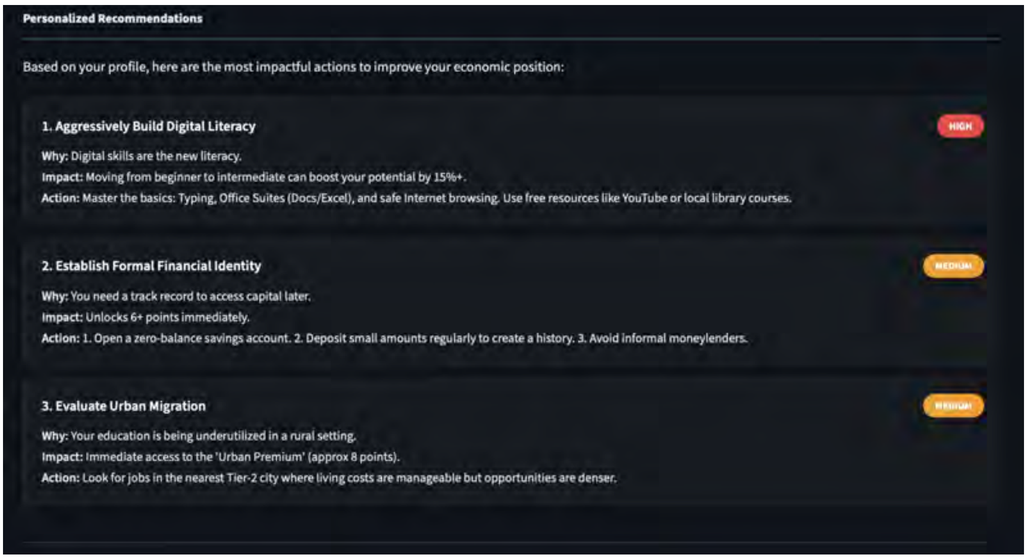


Figure 6.15: Priority actions for economic mobility improvement

6.2.4 Data Quality Visualization Analysis

The module of data quality examines the completeness, reliability, and coverage of the inequality-related indicators in South Asian states. An overview summary gives an average data completeness and makes classifications of high, medium and critical quality indicators using coverage thresholds. This gives a direct evaluation of the reliability of the whole dataset used in the analysis.

Figure 6.16: Data Quality Overview

In order to explore geographical distributions of data quality, the platform maps data quality of countries in the form of a geographic map. The markers used to show the countries have color intensities that correspond to overall percentages in data quality. Such a visualization brings to the fore regional differences in the availability of data and indicates countries with a relatively better or worse data coverage.

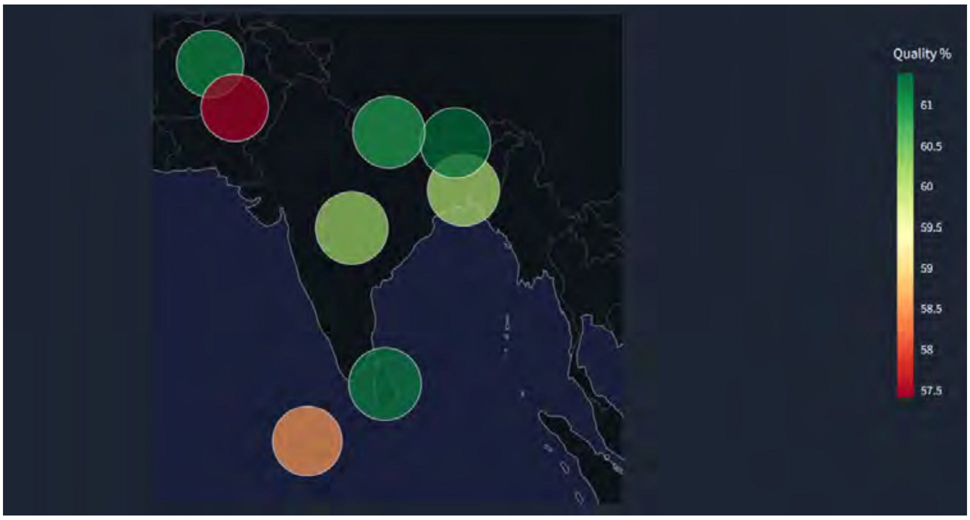


Figure 6.17: Geographic Distribution of Data Quality Across South Asia

Further flow visualization is represented in a structured flow of data between sources and countries and quality categories. This representation follows the inferences of how indicators with different sources of data are added to national datasets, which is then divided into degrees of quality. It assists in determining the point of data loss or gaps in the process of data integration.

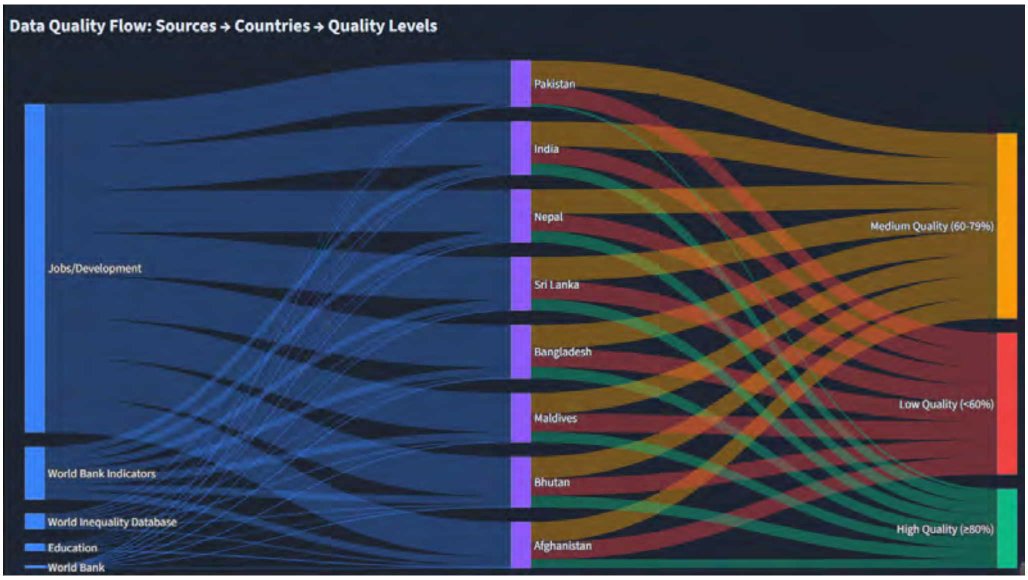


Figure 6.18: Data Quality Flow from Sources to Quality Levels

On a finer level, indicator-specific completeness is studied with the help of a heatmap representation. This plot shows the quality percentage of individual indicators by country and allows one to find missing variables constantly and ones that are well referred to in a short period of time. This transparency helps in interpreting the results of the analytical work carefully in situations where there is a gap in terms of data.

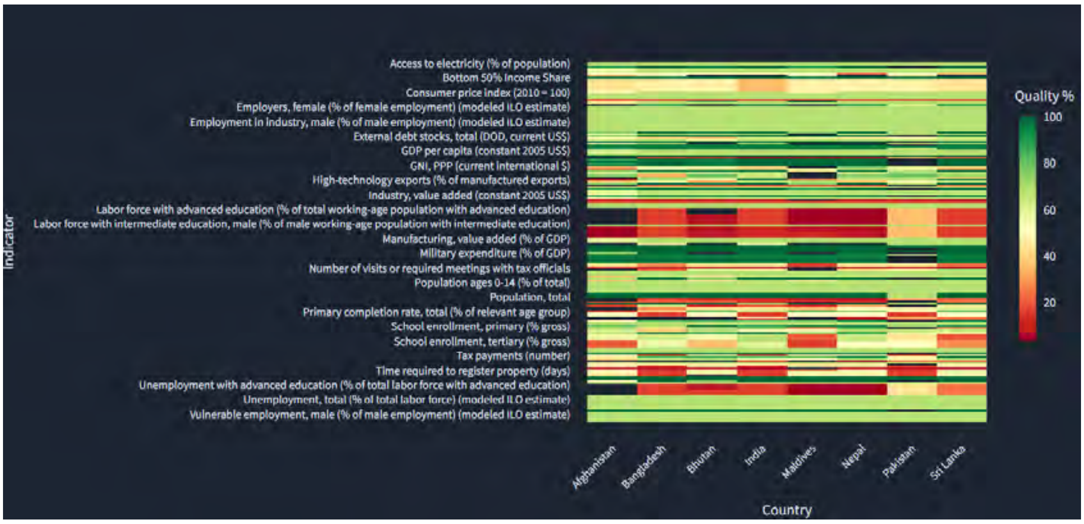


Figure 6.19: Indicator-Level Data Quality Heatmap

6.2.5 Indicator Insights Explorer

Indicating the Insights Explorer offers a perspective of dominance-based socioeconomic indicators to explore the trend of inequality-related patterns in the South Asian countries during a chosen year. Instead of showing raw indicator values, the visualization normalizes the various indicators to the same 0-100 scale to ensure that there is a fair comparison of nations and indicators. It is concerned with finding out which indicators are prominent in the profile of each country and the role that inequality indicators play in larger patterns of dominance. The sunburst visualization contains the central node, which is South Asia, then country-level segments, and finally indicator-level slices. The relative dominance of each slice is presented by the size of the slice and the relative strength of the slice by the intensity of the color.

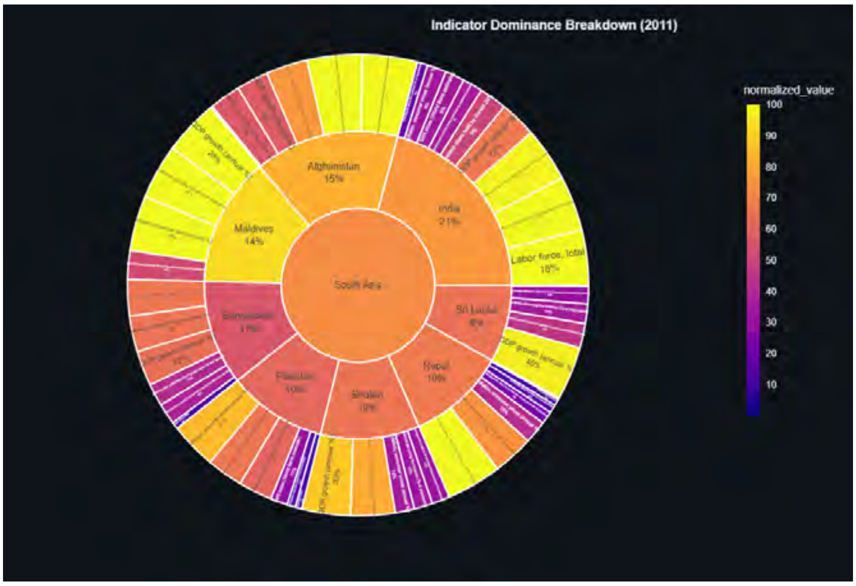


Figure 6.20: Sunburst Indicator Dominance

To help people understand the correct interpretation, the platform has structured advice on the way to interpret dominance patterns and how to differentiate them with the measures of causal inequality. This explanation provides a methodological transparency and avoids the misinterpretation of dominance as the absolute severity of inequality. The interpretation logic of the sunburst visualization is summarized and defines what is meant by circle size, color intensity, normalization, and the descriptive quality of domination patterns.

Figure 6.21: Interpretation Guide of Sunburst

In addition to aggregate visualization, the explorer provides a country-level signal of inequality, which evaluates the amount of dominance that can be ascribed to inequality-related measures including the Gini index, the income shares, the poverty levels, and the unemployment levels. The countries are divided into High, Moderate, or Lower inequality signal groups, which are supplemented by brief narrative descriptions. Figure 6.22 shows the country-based inequality signal summary and the relative inequality trends of the specified year in a descriptive and non-causal way.

Figure 6.22: Inequality Signal Summary Country Wise

To inspect further, the Country Spotlight feature allows indicator-level analysis of individual countries. The view represents a bubble-based dominance profile of indicators that have the strongest relationship to the overall dominance pattern of a country with supporting data tables. Figure 6.23 represents an example of a country's dominance profile, bubble position, and color indicating normalized dominance scores of indicators of choice.

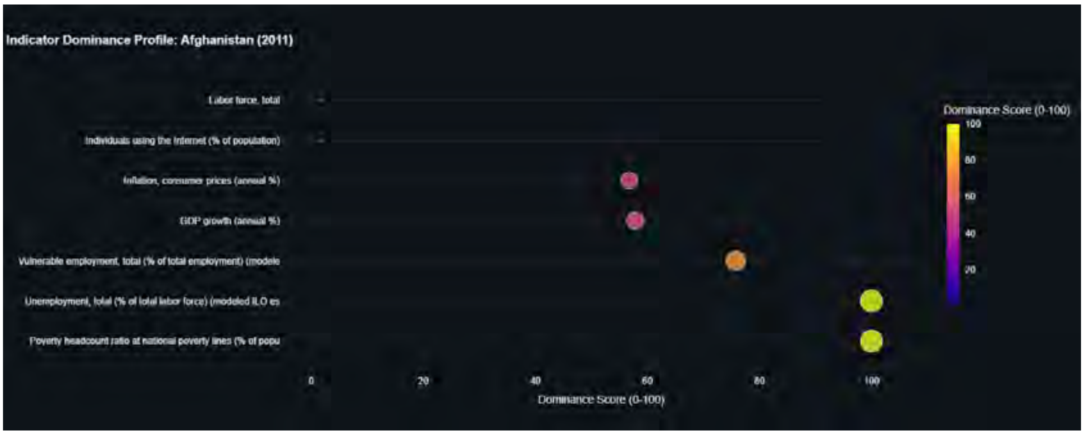


Figure 6.23: Country Indicator Dominance Profile

In order to provide the transparency of the analysis, the platform reveals the raw indicator values and normalized dominance scores. This enables the users to confirm the process of calculating dominance and learn how the visual prominence is related to actual values.

View underlying data (for this country)

Data transparency: Raw values and normalized dominance scores

This table shows the actual indicator values alongside their normalized dominance (0-100 scale). Higher dominance = more prominent in this country's profile relative to others.

Type	Indicator	Actual Value	Dominance Score (0-100)
Inequality	Poverty headcount ratio at national poverty lines (% of population)	38.30	100
Inequality	Unemployment, total (% of total labor force) (modeled ILO estimate)	8.23	100
Inequality	Vulnerable employment, total (% of total employment) (modeled ILO estimate)	66.68	75.8
Economic/Development	GDP growth (annual %)	6.11	57.8
Economic/Development	Inflation, consumer prices (annual %)	10.20	56.8
Economic/Development	Individuals using the Internet (% of population)	5.00	1.7
Economic/Development	Labor force, total	8.13M	1.7

How to read:

- **Dominance Score** = How much this indicator stands out after normalization (100 = maximum dominance for this country)
- **Type** = Whether indicator measures inequality or economic/development factors
- Scores are relative to other countries, not absolute measurements

Figure 6.24: Table of normalised dominance and values of indicators

6.2.6 Temporal Comparison Analysis

The Temporal Comparison module allows comparing inequality indicators systematically between two time points or time periods of choice. The user has the option of selecting a year or time to assess the development of inequality across the years in South Asian countries. The feature allows various types of visualization to emphasize the absolute changes, relative change, and distributional pattern enabling easy identification of the improving and worsening countries. The interface of Temporal Comparison where the user can choose the years of comparison and the types of visualization.

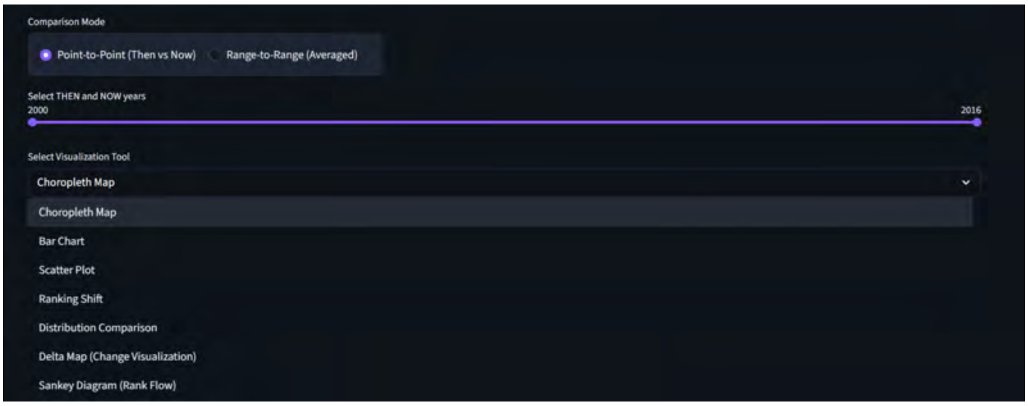


Figure 6.25: Temporal Comparison tool selection interface

The choropleth maps are employed in visualizing spatial variation of inequality that occurs across the periods under selection. The indicators have been color-coded to show the values of the countries so that it is easy to identify regional changes and contrasts. Figure 6.26 shows the choropleth map of the baseline period (THEN) and the choropleth map of comparison period (NOW).

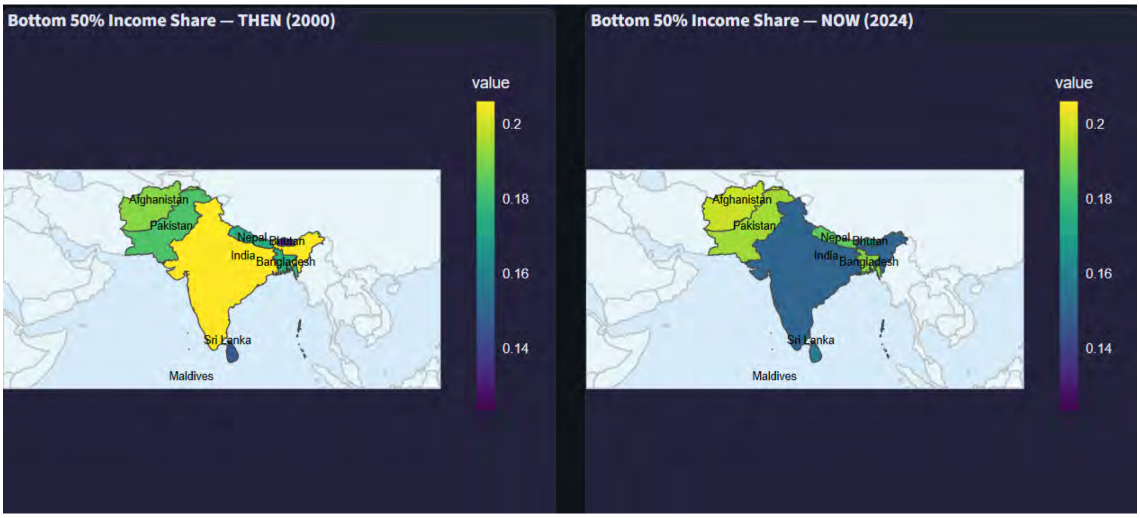


Figure 6.26: Choropleth map of inequality indicator – (2000 vs 2024)

Bar charts are used to make the comparison of the countries across the country by ranking all countries on the basis of this indicator value in each period. This representation shows the variations in relative positions over time. The baseline period is indicated as country rankings.

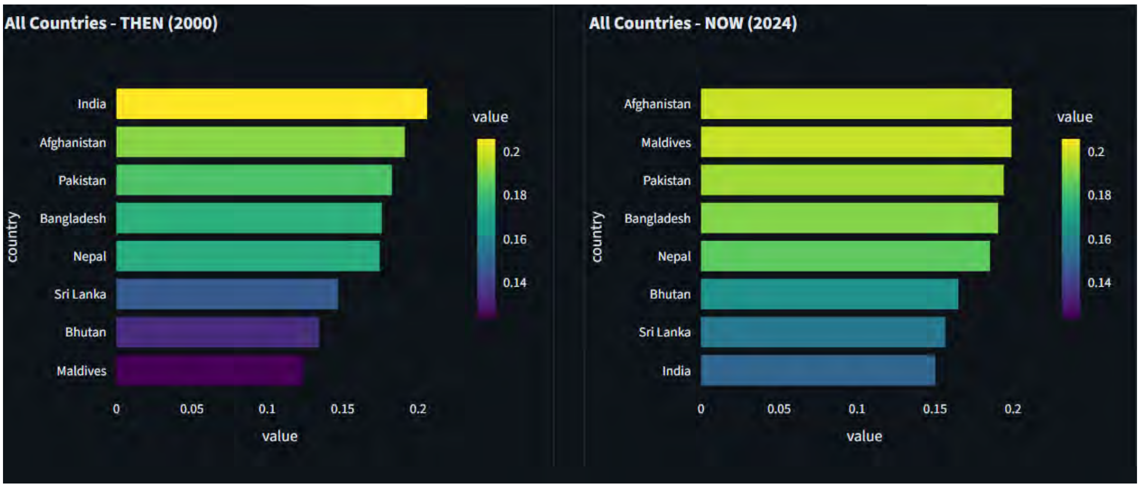


Figure 6.27: Country ranking comparison by inequality indicator (2000 vs 2024)

The distribution comparison represents the variation in the spread of the inequality measure in the entire countries over a period 2000-2024. The variations in median values and dispersion being different will help to show whether the outcomes of inequality will converge or diverge with time. This view offers a regional-scale view over and above the standings of nations.

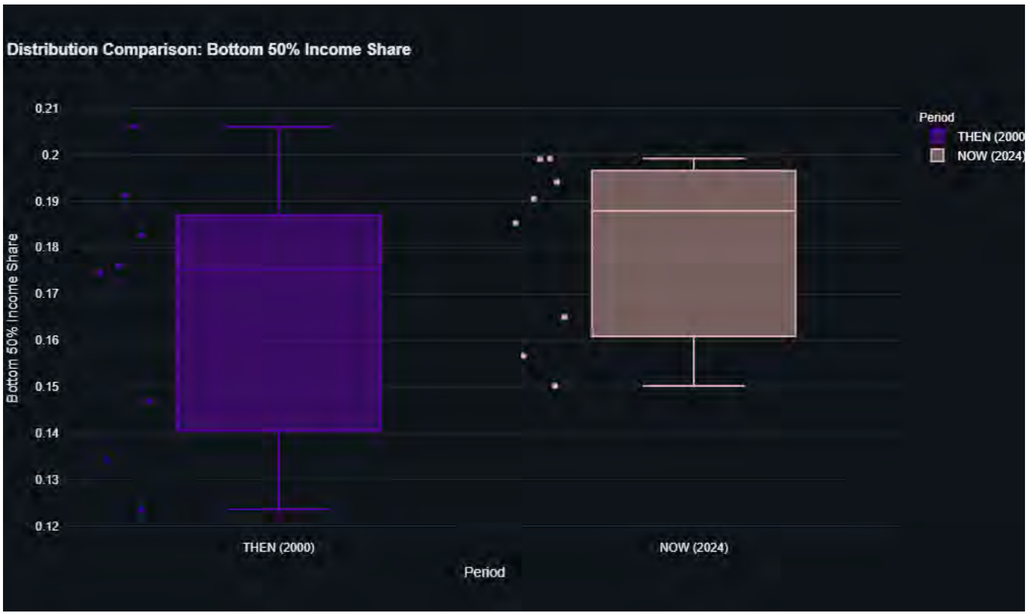


Figure 6.28: Distribution comparison of inequality indicator across countries

These values are put in a summary based on the directional interpretation of each indicator with higher and lower values identified as a more favorable outcome. This would allow the users to properly evaluate whether the change that has taken place in a country over time is an improvement or degradation. As an example, a higher Gini coefficient implies that there is more income disparity and thus is seen as a negative phenomenon whilst higher GDP values represent more economic prosperity and are thus seen as positive. The platform facilitates proper and situational interpretation of cross-country and temporal dynamics by providing a visual analysis with indicator specific meaning.

6.3 Key Visual Insights

In this chapter, the visual elements of the interactive inequality analysis platform are assessed and the discussion is presented on how various methods of visualization aid in the exploration and interpretation of income inequality in South Asia. The platform allows users to analyze inequality in a variety of analytical ways through dashboards, spatial maps, correlation plots, dominance-based indicator views, simulation tools, data quality assessments, and temporal comparisons. All the visualizations demonstrate spatial variations, time variations, correlation between indicators, and structural trends that cannot be easily observed using the traditional reports or tabular data. The system enables both informed and exploratory analysis to technical and non-technical users by utilizing attributes of interactivity, comparative views, and interpretability. Although the visual patterns that are introduced are descriptive and rely on the availability of the data, they offer a solid basis of drawing significant insights and trends.

Chapter 7

Results and Discussion

7.1 Key Findings from Data Analysis

Based on the time-series line chart and trend panels in the dashboard, income inequality in South Asia has always had a gender gap throughout the years between 2000 and 2024. The distribution of male to female income in all the 8 South Asian countries that were reviewed shows that the distribution of male income shares is consistently higher than that of females, thus depicting a hidden structural disparity in the distribution of income resources. Longitudinal plots incorporated in the analytical dashboard show that, despite the overall upward trend in aggregate income levels following 2005, the gains made by the percentage of women in income were relatively low, and hence kept a stable gender income gap across the studied countries during the study period. There is a significant deceleration between the periods 2008 to 2009 and between the years 2020 to 2021, coinciding with the global financial crisis and the COVID-19 pandemic, respectively. These economical decline led to more pronounced income losses among women, as evidenced by steeper declines in female income indicators relative to those for men The heatmap visualization also shows that inequality is higher in countries with higher informal labor dependencies, despite women being the majority in the workforce, especially in India, Pakistan, and Bangladesh. In Bangladesh and Nepal, some minor improvements were witnessed through the years after 2015; the overriding trends indicate that economic progress is not equal to progress in minimizing income inequality. According to the dashboard's analysis, Bangladesh shows a low growth in the contribution of female income starting in the early 2010s, as the country experienced the growth of export-oriented manufacturing, especially in the garment industry. Nevertheless, even with these advances, the problem of gender income inequality is still present because women earn less and are employed in low-paid and stable occupation still face disproportional income gains. India has the best rank of high level of inequality of gender income as observed in the inequality heatmap, over the years. Even though the GDP per capita growth was constantly growing throughout the duration of the study, the growth in income level participation of women remained unchanged or decreased in various intervals, indicating that the economic growth was not progressing evenly across all male and female income earners. The widening gaps in income shares also indicate during which periods the low rates of growth are being translated into gender equal income gains.

In Pakistan, a visual representation reflects that there has been a low-income female share in all years under observation. Gender income inequality is a structural phenomenon, perpetrator of which is, in part, chronically low female labor-force participation and inability to access formal labor. Pakistan is always ranked in the high-intensity categories of inequality by the dashboard, which means that there is little or no progress made over the years. Comparison of longitudinal dashboard trends reveals that Nepal has a relatively smaller gender gap in incomes as compared to other bigger economies of the region. Women who are engaged in agriculture and remittance inflows help in maintaining the stability of income; nevertheless, the benefits are largely concentrated in informal sectors of employment, and hence, the benefits would not be able to make a significant impact in reducing inequality as a whole. Maps and heatmaps consistently show Sri Lanka in the low inequality category, indicating a more equitable distribution of income by gender. Such a relative equality is enabled through improved education levels and access to healthcare and social protection systems; however, performance disparities and inequalities still persist, especially in the private-sector labor force.

The Maldives and Bhutan fall in the middle-low range of inequality group in the visual representation, specifically in the 2015-2018 period. The different employment opportunities created under service-driven and tourism-driven industries have had a relatively improved gender income performance, although occupational segmentation is still evident. The situation with Afghanistan is explicitly defined in the dashboard analysis, where the country illustrates an abrupt decline in the indicators of female income, especially since 2021. The erosion of income earnings by women is a sign of institutional and policy-driven exclusion and not traditional economic relations, which makes Afghanistan stand out from the general trends of the region. At the regional level, the analysis shows that the trend of gendered income inequality in South Asia is systematic. The implications of this trend are the high level of reliance of women in informal jobs, gender occupational division, and lack of access to capital, access to financial services, and formal jobs for women. Also, it is noted that the problem of gender income inequality has a significant correlation with the education level, birth rate, and established societal values, which implies that the imbalance may be tracked both in economic and socio-cultural contexts. In general, the results have proven that gender income inequality in South Asia is not an economic problem but a multidimensional development challenge, which needs a combination of specific policies on gender-sensitive labor, education, and social protection.

7.2 Correlation Analysis Results

The analysis was carried out using correlation analysis to determine the similarity of trends in gender income inequality in South Asian countries between the years 2000 and 2024.The findings are presented in the form of a country-level correlation heatmap and clustering results in the dashboard, describing the co-movement between inequality paths between countries. In general, it is possible to say that the heatmap reflects high levels of positive correlation throughout the area, which means that inequality dynamics are widespread among most members of a population, as well as including the presence of significant structural differences between particular groups of countries.

There is a specific cluster of mainland countries, which includes Bangladesh, Nepal, Pakistan, and Bhutan, and is marked by the stable pattern of a strong positive correlation in trends of gender income inequality. The heatmap indicates a consistently high degree of correlation between the countries under study, which means that the changes in inequality will, in all probability, change simultaneously. Such clustering indicates common structural characteristics, such as a common composition of the labor market, a significant level of informal labor dependency, and similar socio-economic limitations against women's participation in income. Although there are disparities in the size of the economy, the pattern of correlation between the two countries indicates that common structural factors, rather than country-specific shocks, are causing the inequality between the genders' earnings in these countries, hence the idea of treating them as a mainland cluster of countries.

Although India is geographically located in central South Asia, it does not fit well in the correlation structure and thus is considered independent in the analysis. The heatmap also shows that India has a strong positive correlation with some of its neighbouring countries, but the level of association and consistency is more fluctuating as compared to the relatively steady association in the mainland group. This observation calls to the fore the present internal heterogeneity of India, whose main features have large territorial differences, dualistic organization, and unequal female involvement in the workforce and in the labor market in its constituent states. This, in turn, causes the trend in gender income inequality in India not move along the same track as the line of the smaller economies in the mainland, which means that it does not form a statistically compact cluster. However, India is still deeply tied into the regional system, and even the opportunity to include it in a pairwise correlation comparison demonstrates that it is not fully segregated but partially integrated into the grander inequality system, which involves South Asia.

In the heatmap of the correlations, Afghanistan is obviously a statistical outlier. Although the correlation values of Afghanistan and the rest of the countries are always numerically high, the relationship has an entirely different interpretation that is not in tandem with the region. These performance correlations are structurally induced by comparable long-term trends, not by a common structural situation. Stereotyping Institutional and policy-induced exclusion and not economic adjustment mechanisms is reflected by dreadful post-2021 deviations in female income indicators that can be seen both on a heatmap and in a summary of correlation. As a result, Afghanistan does not group well with other South Asian countries and thus is considered an institutional and policy-based outlier.

There is a different trend with Sri Lanka and the Maldives, which exhibits a high and relatively steadier correlation structure. The internal similarity in these countries is high, and the alignment is also slightly reduced when it comes to connecting with the mainland economies. As indicated by the heatmap, this subgroup has a less volatile and smoother gradient of correlation, and its gender income results are less vulnerable across time. These unique trends are associated with greater levels of human development, robust social protection networks, and greater access to formal jobs, which set Sri Lanka and the Maldives apart from the mainland cluster, in spite of regional cohesiveness in general.

Combined, it is shown through the correlation analysis that gender income inequality in South Asia is marked by strong regional co-movement as well as evident structural segmentation. The mainland cluster captures similar structural imperatives. Sri Lanka and the Maldives have not experienced significant dynamics of inequality that are being moderated, Afghanistan is a non-comparison institutional outlier, and

India is in a transitional position that is influenced by internal heterogeneity. This subtle classification points out the necessity to view the patterns of inequality in areas contextually.

7.3 User Testing Results

The usability, functionality, and general user experience of the South Asia Income Inequality Dashboard were tested using user testing. A varied sample of technical and non-technical respondents consisting of students, faculty members, and non-faculty staff of various disciplines (computer science, economics, business, media studies, and engineering) responded to the survey. The subjects showed different degrees of acquaintance with data dashboards, both frequent and first-time (user), which allowed balancing the evaluation of the different levels of user expertise. The general usability observations show that the design and navigation layout of the dashboard have a high degree of satisfaction. The participants have also stated that the dashboard is simple to use and that menu options and sections are well arranged all times. The interface design was useful in facilitating intuitive interaction, whereby users could find features and understand visualizations without much prior knowledge or help. This is the uniformity among users of varying technical loyalties that makes it possible to conclude that the dashboard is effective in overcoming cognitive and navigational obstacles.

The other feedback emphasizes the perceived usefulness of the dashboard analytical features. The features listed by the participants as the most valuable described hierarchical indicator visualizations, multidimensional overview dashboards, geographic map analyses, correlation exploration, and the income simulation module. The variety of preferences means that the dashboard can support a wide range of analytical approaches and can have a wide range of user purposes, the highest level of pattern recognition at the top, and the deepest and most detailed inspection with indicators in the middle.

Furthermore, in the case of educational outcomes, the majority of the participants shared that the interaction with the dashboard gave them clear knowledge and insights into the income inequality pattern in South Asia. Maps, charts, and graphs were also observed to be very useful in presenting complex relationships. Although most of the respondents did not experience much confusion, a subgroup of the respondents had problems with the interpretation of some of the visualizations, especially in the map analysis module. Nevertheless, there was an overall agreement that the additional explanatory writing and graphic materials made it easy to understand and interpret complex economic signals. The dashboard's visual representation and professional design received positive remarks form the participants. The dashboard's layout, color schemes, and understandability were rated high, and also the majority of them stated the dashboard was suitable for academic research and policy-related applications. Technical performance also received mostly positive reviews, most stated smooth interaction, and minimal technical issues predominantly surrounding minor loading time delay, suggesting potential improvement areas.

The general level of satisfaction was very high; everyone said that they would recommend the dashboard to others to learn about income inequality data. Much attention was also paid to the platform with regard to the data processing and normalization processes. Even though the users provided suggestions of increasing the loading speed, decreasing the text density, adding more color options, and adding contextual annotation, they were considered moderate in scale and did not lower the overall favorable ratings of the system.

7.4 Dashboard Effectiveness Evaluation

Based on the outcome of user testing, the effectiveness of the South Asia Income Inequality Dashboard was assessed in terms of accessibility, user engagement, and understanding. Dashboard accessibility was well-received by the users. Design features such as the most distinct iconography, clear-cut organization of color diagrams, uniform layout standards, and serious forms of indicators play an important role in usability. Analytical tasks were completed by users in a fairly short time with limited navigation errors and a low need for outside help. Manual interaction and navigation, in addition to the use of tools with low access and high usage, show that even inexperienced users were confident to explore the dashboard as well as the whole web application. Furthermore, according to the user engagement, the dashboard was effective and efficient in exploratory analysis. The records of participation indicate a high continuation and sustained engagement, showing that users actively participate in the different parts of the dashboard and spend a significant amount of time on the other features of the web application afterwards. The most frequented features are the Map Analysis page, the Correlation Explorer page, and the Income Simulator page. This shows that people accessed other features of the web application to seek additional data and information rather than to passively receive the information. A better understanding on the part of the user is an outcome of the salient design of the dashboard. Post-interaction assessments, respondent feedback, and results indicate that users are less hesitant in recognizing countries considered to feature higher rates of gender income inequality and in identifying the interdependence of indicators more accurately after using the dashboard. The diluted need for external clarification brings home the representational characterizations of the dashboard and the textual accompaniments to the successful accomplishment of the complex analytic knowledge. Finally, the dashboard has a high ability to transform multidimensional gender income inequality data into unambiguous, meaningful knowledge, and that is why it can be used both in scholarly and policy-related discussions.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This paper provides an analytical design and development of a web-based application to investigate income inequality in the South Asian region by interactive data visualization, comparative data analysis, as well as real time data integration. To have an open, data-driven tool for researchers, policymakers, and educators to explore the issue of inequality as a multifaceted phenomenon caused by the complex interaction of economic structures, social systems, infrastructural development, and demographic changes is the core motivation of the project. Its combination of 259 socioeconomic indicators across eight South Asian nations (Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan and Sri Lanka) over a 24 year span (2000-2024) allows one to explore development differences and patterns of inequality in one of the most populous and economically diverse regions in the world.
The analytical paradigm derived during this study fills a crucial void in the current platforms of inequality analysis by focusing on stagnant reporting or spreadsheet-based analyses based on a single indicator. The architecture of the system allows the dynamic examination of disparities related to income inequality and other socio-economic disparities. Moreover, the design philosophy of the platform focuses on both analytical and user-friendliness, as the policy-oriented research tools are to be used by different groups of stakeholders with different technical abilities.

8.1.1 Objectives Achieved

The initial goals as stated at the start of this project have been met in a holistic manner by the successful development of a modular, interactive, and scalable analytical tool. It can be adjusted to work with the latest state of practice in terms of web-based inequality analysis tools. To start with, the system proves to be effective in facilitating a thorough comparative analysis of various countries of South Asia as users can access inequality trends longitudinally throughout the 24-year window of study and cross-sectionally across different indicators. This multi-country, multi-indicator, multi-temporal analytic capacity supports the basic aim of offering a regional analytical angle. The interactive filtering system of the platform enables users to choose any mix of the eight South Asian countries and compare them on any of the 259 indicators available, Thus, it can perform analyses with any flexibility to fit a particular research question or policy issue.

Second, the app effectively fulfills its goal of interactive visualization driven analysis by an advanced set of synchronized visual elements that convert raw data into actionable ideas. The platform has nine separate analytical pages which are optimised to specific tasks in analysing data: (1) an overall Dashboard that displays real-time overviews, acting as a central hub for the platform (2) a Smart Search interface with the ability to query the site using natural language and taking one-click actions to target analytical views; (3) a Map Analysis page that can display animated choropleths to explore geographic patterns; (4) a Correlations Explorer with interactive scatter plots and regression fitting; (5) an Income Simulator modeling inequality scenarios; (6) a Data Quality monitor tracking completeness and reliability; (7) an Income Indicator that provides visualization of hierarchical indicator relationships; (8) a Temporal Comparison tool for cross-period analysis; and (9) an integrated Help system with contextual documentation. Such a wide choice of visualization guarantees that the user can look at an analytical question in various ways, choosing a type of visualization that is most appropriate to their particular analytical task.

Third, the project has effectively introduced a set of around 259 different socio-economic variables by using a unique hybrid data architecture that wisely unionizes customized local data with real-time data being dynamically sourced via authoritative international data, such as the World Bank Open Data API, the International Monetary Fund Data API and the United Nations Data Portal. Notably, the system does not eliminate redundant indicators, including those based on GDP per capita, GINI coefficient and the proportion of poverty based on the number of heads, which are provided by different sources, keeping all the information about the data origin fully transparent.

Fourth, the modular architecture enables simple adding of new indicators, countries, or analytical modules without the need to redesign the entire system or perform a lot of refactoring of existing code. As an example, becoming more geographically expansive, including other Asian nations, or including other continents would mostly need new country mappings and decentralizing data availability, instead of reshaping basic analytical logic.

8.1.2 Key Contributions

The paper has some key and original contributions to the problems of inequality analysis, data visualization, and development informatics by enhancing the practical capacity to conduct a policy study with the focus on evidence gathering and educational development.

To begin with, the project provides a thorough, indicative-heavy analytical platform, specifically designed to the South Asian regional environment, which is a critical issue in the current analytical instruments range. The system includes an exceptionally wide range of 259 variables, including patterns of income distribution, poverty measures at various levels of measurement, employment and labor participation in the labor market (including gender-disaggregated data, as well as vulnerable data on employment and indicating pollution), and infrastructure access measures, demographic characteristics including population growth and urbanization rates, and environmental aspects. An example would be a researcher, studying the high GINI coefficients in a specific country, who would also have the chance of establishing the relationship that the trends in inequality remain connected to either lack of electricity access in rural settings, low participation of males in the labor force, insufficient educational systems, or the population explosion.

Secondly, the system adopts the two-tier data architecture whereby curated local data is stored on the server to provide data access on clean, validated and formatted indicators data, with live API integration options that provide real-time data retrieval services in World Bank Open Data API (individual 277 indicators over 60+ years), IMF Data API (macroeconomic indicators), and UN Data Portal (pride themselves on the Human Development Index indicators and associated measures). The managed local data guarantees the provision of consistent platform performance and user experience irrespective of third-party API availability, network connectivity and rate limits. The benefits of API integration are a validation route, ability to access latest available data publication, and ability to refresh the dataset in the future as new information is created. Most importantly, the system does not arbitrarily pick out any single authoritative source, but represents overlapping indicators of various sources, which ensure full transparency of information on data provenance.

Third, the project contributes significantly to the accessibility of the education and analysis of development research and data science education. The interactive visual design of the platform is effective in resolving the balance between advanced statistical analysis and user-friendly interfaces to interact with complex datasets without technical skills related to databases. Users without expertise, policy analysts in government ministries can create useful analytical queries, produce ready to be in print visualizations and derive policy-related insights via simple point and click interfaces, natural language query, and workflow guidance. As an illustration, the Smart Search feature enables users to just type in a word such as “Bangladesh” and all the available data about that country is presented, with pre-built filtering. User can immediately navigate to a specific visualisation without having to comprehend data structure or needing to build out complex filter logic. This two-way utility, in helping both novice and expert users simultaneously, gives the platform a broader applicability to a wide range of stakeholder sets.

The architectural choices adopted in building the platform, such as modular separation of data loading, transformation and visualization logic; the abstraction of indicator metadata into structured system of categories; the creation of dynamic filtering mechanisms that adjust to data available; adoption of caching strategies to help optimize platform performance, etc., all lead to a technical infrastructure that can be easily extended, modified, or re-purposed to support other related analytical tasks. The reusability of this platform makes it an analogous research infrastructure that can be extended to support new data sources and data analysis techniques. It can be used to build innovation in the analytics of development and visualization in the future.

8.1.3 Impact Assessment

The wider implications of the project are spread out over analytical, learning, and social levels to aid better insight on the patterns of inequality and positively influence the basis of evidence-based policymaking in South Asian region. Analytically, the platform facilitates the exploration of inequality as a multidimensional phenomenon with many dimensions as opposed to a one-dimensional statistics. Comprehensive indicator coverage and broad filtering makes the system available to a researcher and analyst to investigate the interplay of income-related disparities in a systematic way. As an example, the correlation analysis features allow an analyst studying the reasons behind certain countries having constantly high GINI coefficients to investigate whether inequality patterns are related to inadequate access to basic services such as electricity or clean water, elevated levels of precarious employment, or informal sector activity.

The system enables users to dynamically choose indicators, adjusting temporal ranges, and comparing countries with different types of visualization. Additionally, the system supports the user in determining the answers to the following queries: What does the lack of information on a particular nation or era imply? When we observe a link or trend and suspect variables that are confusing, what should we do? At a societal level, the platform allows policymakers to discover the countries or regions with the best results, and the patterns in specific indicators. This comes as a huge advantage to understand potential policy strategies or institutional frameworks. To take one example, when the data shows that some countries achieved high levels of inequality reductions and at the same time they were able to sustain growth in the economy, the policymakers may be interested in exploring what policies and programs have led to these results. In addition, the open disclosure of information on the platform through clear recognition of errors, such as the absence of values, lack of temporal coverage, and measurement difficulties, etc., strengthens the notion of responsible and ethical data utilization in policy-making. By ensuring that the quality of the data gets visible and is not concealed, the system helps the users to qualify their conclusions properly, acknowledge the uncertainty, and not to make the overconfident assertions made on the ground of the incomplete or doubtful evidence.

8.2 Future Work

Although this project has managed to fulfill its main functions and provide an operational, deployed analytics platform, there remain opportunities to produce more improvements, growth, and development in the overall system. The proposed extensions below are meant to enhance analytical strength, user-friendliness and accessibility, scope and functionality capability, enhance performance and reliability and enhance long-term sustainability.

8.2.1 Improving Accuracy

The next stage of development must focus on refining analytical accuracy and data quality by adopting more advanced data pre-processing, data validation, and data quality assurance processes. Improved data-quality measures will generate higher quality analysis results. Specifically, neither can analytic completeness nor can statistical integrity be lost by the application of advanced techniques used to impute missing data. Such as the multiple-imputation framework, or methods based on machine-learning make use of correlations between indicators in order to impute missing values. In addition, verification of systematic consistency between overlapping indicators retrieved in different repositories would support data verification. Automated validation programs to measure discrepancies and obtain correlation coefficients to assess systematic agreement and identify large discrepancies can signal users of measurement uncertainty or initiate manual review to rectify it.

8.2.2 Expanding Indicator Set

Even though the current platform includes set of 259 socio-economic indicators across different thematic areas, it would be feasible to extend analysis with indicators coverage to include more areas of development that are not sufficiently well covered. A combination of governance-quality and institutional-capacity measures would potentially permit the systematic analysis of the relationship between inequality patterns. The possible sources of data are the Worldwide Governance Indicators, Corruption Perceptions Index by Transparency International, and country specific organizations and so forth. Such measures would facilitate these analyses that would consider questions like: In high quality governance countries, is there low inequality? What is the correlation between corruption levels and levels of poverty reduction? Are democratic institutions associated with more equitable development? In addition, it might be beneficial to include regional and national statistical office data as a way of enhancing the local relevance. Collaborations with South Asian statistical bureaus may allow the incorporation of sub-national data so that the data can be analyzed at province or district level or to be updated more frequently when indicators are changing rapidly.

8.2.3 Multilingual Functionality

In order to expand access and impact significantly in the linguistically diverse region of South Asia, the platform seek to accommodate the multilingual nature through the provision of localized user interfaces in all languages used in the respective target countries. The eight countries are considered as a unity of unsurpassed linguistic diversity, also including the large languages: Bengali (the language of in Bangladesh and West Bengal), Hindi and Urdu (the languages of of people in India and Pakistan), Nepali (the official language of Nepal), Sinhala and Tamil (the official languages of Sri Lanka), Dari and Pashto (the official languages of Afghanistan), and Dzongkha (the official language of Bhutan). The localization of interfaces would be implemented by ensuring that elements of user interface translations such as page titles, button labels, menu items, documentation of help, tooltips on the context, and explanatory text are translated into the target languages so that non-English speakers can use the platform comfortably. Moreover, the ability to have language switching capability that allows the user to choose their language of choice through a menu choice will offer this advantage to all multilingual individuals involving the use of the common code-switching language

8.2.4 User Personalization Features

The introduction of increased user-customization features will result in higher levels of analytical efficiency, user interactiveness, and usefulness of the tool in specialised research processes as users can adjust analytical environment to their needs, preferences and repetitive analytical procedures. Analytical procedures would be simplified by having custom dashboards that allow users to choose the visualisations, metrics and indicators that they see in their default view. Further development may allow users to create custom dashboard configurations by choosing between a range of available visualisation objects, deciding where to place key metrics, scale charts, and save such settings as personal default configurations. An example is a researcher who specialises in educational inequality, he or she may customise a dashboard to predict educational indicators, where the rate of enrolment, literacy, and education spending take a prime place. This kind of personalisation would help to minimise the cognitive load, as only the immediately significant information with the focus on user analysis would be presented.

8.2.5 Tool and Platform Integration

The current support of the platform with the World Bank Open Data API would lead to a strong base of further interoperability with other data platforms and analytics, as well as to the extension of the platform beyond the self-sufficient web interface to work within larger analytical systems and data flows. Future practice might adopt the use of downloadable data bundles in standard formats CSV, JSON, Stata.dta files, that would contain subsets of data oriented at user choice with thorough metadata and define indicators, measurement units, sources of data and their time coverage. These export abilities would place the platform as a data discovery and access tool. Besides that, creating open APIs which give programmatic access to the curated dataset would enable third party applications, research tools or automated systems to query and access data without downloading it manually. The API endpoints may be extended with flex queries to filter by country, indicator, year range and data-quality thresholds and provide programmatically consumable structured JSON or XML responses. Open API documentation would further simplify the integration of API by external developers.

8.2.6 System Performance Optimization

As the user base of the platform will grow, the number of indicators will grow, and the capacities to analyze data will be expanded, performance optimization would become more critical to maintain responsive and reliable user experiences. Calculation overhead issues could be reduced by using advanced caching solutions and improve responsiveness. Future optimization could look like multi-layer caching and individual policies of different types of data: long-term caching of rarely modified reference data (e.g. country mappings, indicator metadata), medium-term caching of periodically- updated indicator data, short-term caching of user-specific results of analytics, no caching of fast-moving live API data. Memory usage, as well as computation time could be minimised by optimising data structures and processing algorithms. It may involve pre-computing popular aggregations (country averages, regional averages) when ingesting the data instead of computing them on-demand, highly effective data indexing to support fast country-year-indicator lookups, and columnar forms of data storage can be used to attain both better compression and faster column-wise operations.

8.2.7 Security and Privacy

Currently, the platform does not gather personal data or allow user accounts, and uses only publicly available information. However, for any future features, including saved user preferences, personalised dashboards, or collaboration will require enacting a comprehensive set of security controls and privacy safeguards to protect the confidentiality of user data and responsible execution. In case the option of user-account is added to record preferences or custom arrangements, proper authentication and authorisation processes have to be developed. They should include password hashing with industry-standard algorithms (e.g., bcrypt), secure session management to avoid session hijacking, multi-factor authentication of sensitive accounts, and best practices in authentication, e.g. password-strength policies and resistance to brute-force attacks.

8.2.8 UI Improvements

More user interface and user-experience development may increase the level of clarity, easiness, and visual attractiveness, especially of complex visualisations and analytical processes. These modifications will make the platform more intuitive and pleasant to different user groups with varying expertise levels. The deployment of the newer contextual help systems would provide a specific kind of assistance at the time and place where the users need it. It could be interactive tooltips that can be created by hovering the cursor over graphical objects. These tooltips would clarify the content of what is being visualised and what is being interpreted, contextual assistance panels which automatically show an appropriate piece of documentation depending on the page or analysis being performed. Furthermore, the combination of adaptive layouts based on the screen size and the device used would make the user experience as productive as possible, such as, in the desktops, laptops, tablets, and mobile phones. Having responsive design where key features can still be available and used even at smaller screens would make accessibility more available, and practitioners or students in the field who may have no access to desktop computers will be able to use the platform.

8.2.9 User Feedback

The inclusion of user feedback mechanisms and a workflow of usability design cycle would result in priceless insights on the actual use of the platform, the features that users consider the most valuable, the challenges they face, and the possible improvements that would make the most significant contributions to the utility and usability. The inclusion of in-application feedback gathering would enable the ease of customers reporting problems, suggesting ways of improvement, or even positive experiences within the platform. These mechanisms may consist of: a persistent feedback button, which is available on all pages, allowing them to quickly submit comments; formatted feedback forms that allow them to elaborate on the issue or features being requested, suggested; optional contact information, allowing the user to follow up on the discussion; and the classification of feedback types (bug reports, feature requests, usability problems, content suggestions) to aid in efficient processing.

8.2.10 Extended Application Domains

Lastly, the analytical structure and architectural design of the platform opens possibilities of wider applications of the platform to other geographic areas, themes, and analysis purposes. A greater overall developmental investment will result in being able to share returns and technical innovations of the research findings with more people. Since the country-handling system was modular, the indicator framework was flexible, and the general purpose of the visualisation components, could easily be tailored to the analysis of: Latin American and Caribbean comparative studies; Middle Eastern and North African regional analysis; East Asian and Pacific development patterns or global comparisons of all developing regions. Geographic expansion would mostly add country mappings, provide data availability to the target countries, and possibly change visualisations to handle larger amounts of countries. The analytical logic, visualisation types and patterns of interaction would be then directly transferred.

Bibliography

[1] K. S. Jomo, "Globalisation, liberalisation, poverty and income inequality in southeast asia," OECD Development Centre Working Paper No. 185, Tech. Rep., 2001. DOI: 10.1787/18151949. [Online]. Available: https://doi.org/10.1787/18151949.

[2] M. Shafiullah, A. F. M. Shamsuddin, and H. C. Kee, "Income inequality and economic growth: Evidence from south asian countries," *Economic Issues*, vol. 19.no. 1, pp. 1–24, 2014.

[3] N. Akram and A. Hamid, "Public debt, income inequality and macroeconomic policies: Evidence from south asian countries," *Pakistan Journal of Social Sciences*, vol. 36.no. 1, pp. 99–108, 2016.

[4] K. Kazár, "Using statistical datasets for describing poverty and income inequality," in *Proceedings of the Roundtable Conference of the International Association of Statistics Education (IASE)*, J. Engel, Ed., Promoting understanding of statistics about society, Refereed paper, Berlin, Germany: International Association of Statistics Education (IASE), 2016. [Online]. Available: https://iase-web.org/Conference_Proceedings.php.

[5] K. S. Imai and B. Malaeb, "Asia's rural-urban disparity in the context of growing inequality," International Fund for Agricultural Development (IFAD), Tech. Rep. 27, 2018. [Online]. Available: https://www.ifad.org/en/web/knowledge/publication/asset/40497335.

[6] U. E. Asia and P. R. Office, *Gender counts: A quantitative assessment of gender inequality and its impact on children and adolescents in south asia*, 2019. [Online]. Available: https://www.unicef.org/rosa/media/5016/file/Gender_Counts_South_Asia_Executive_Summary.pdf.

[7] M. R. Islam, R. Islam, and S. C. Paul, "Does lean philosophy improve firm's financial performance? evidence from big public companies," *Global Business & Economics Anthology*, vol. I, pp. 48–61, 2020, Copyright © 2020 by the Business & Economics Society International (B&ESI), ISSN: 1553-1392.

[8] K. Munir and M. Bukhari, "Impact of globalization on income inequality in asian emerging economies," *International Journal of Sociology and Social Policy*, vol. 40.no. 1/2, pp. 44–57, 2020. DOI: 10.1108/IJSSP-08-2019-0167. [Online]. Available: https://doi.org/10.1108/IJSSP-08-2019-0167.

[9] K. R. Acharya and H. Acharya, "Assessing the relationship between income inequality and economic growth in south asia," *The Economic Journal of Nepal*, vol. 45.no. 3–4, pp. 41–54, 2022. DOI: 10.3126/ejon.v45i3-4.63148.

[10] S. Ahmed, *Socio-economic inequality*, M. A. Salam, R. S. Haque, and A. M. Alam, Eds. Lincoln University College, 2022, pp. 1–14. [Online]. Available: https://books.lucp.net/wp-content/uploads/9-Socio-Economic-Inequality.pdf.

[11] N. Bikakis, "Big data visualization tools," *Encyclopedia of Big Data Technologies, 2nd Edition*, 2022. DOI: 10.1007/978-3-319-63962-8_109-2. [Online]. Available: https://doi.org/10.1007/978-3-319-63962-8_109-2.

[12] W. .-. Wealth and I. Database, *What's new about inequality in south and southeast asia in 2023?* 2023. [Online]. Available: https://wid.world/news-article/2023-wid-update-south-and-southeast-asia/.

[13] "Natural earth vector." Used to download detailed 1:50m GeoJSON boundaries for the map generation script (`scripts/build_sa_geojson_detailed.py`)., Natural Earth Project. [Online]. Available: https://raw.githubusercontent.com/nvkelso/natural-earth-vector/.

[14] "World bank open data api." Fetches all primary economic and social indicators (e.g., GDP, Gini Index, Literacy Rate) for South Asian countries. Used in `utils/api_loader.py`. Data sources: World Bank, ILO (via WB), IMF (via WB), UN Population Division (via WB)., World Bank. [Online]. Available: https://api.worldbank.org/v2/.

[15] "World inequality database (wid) api." Provides data on income and wealth distribution worldwide, used for analyzing inequality trends., World Inequality Database. [Online]. Available: https://wid.world/.

[16] J. Ahmad, Abukasim, and A. Hussain, "Rural migration and income inequality in south asian countries: An econometric descriptive analysis," *South Asian Journal of Social Studies and Economics*, vol. 22.no. 3, pp. 1–12, 2025. DOI: 10.9734/sajsse/2025/v22i3964. [Online]. Available: https://doi.org/10.9734/sajsse/2025/v22i3964.

[17] "Supabase api." Accessed via the Supabase Python client. Handles user authentication (login/signup) and database persistence (`user_profiles` table) for saving dashboard configurations. Used in `home.py` and internal utility calls., Supabase. [Online]. Available: https://supabase.com/.

[18] London School of Economics and Political Science, *South asia inequality review*, https://www.lse.ac.uk/International-Inequalities/Research/Global-Inequalities-Observatory/South-Asia-Inequality-Review, Accessed: 2025-06-19.

[19] World Bank, *South asia — world bank gender data portal*, https://genderdata.worldbank.org/en/regions/south-asia, Accessed: 2025-06-19.