

Software Engineer Intern at Polygon Technology

by

M.A. Ahad Shikder
20301316

Internship Final Report submitted to the Department of Computer Science and
Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
June 2025

© 2025. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The internship report submitted is my own original work while completing degree at Brac University.
2. The report 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 report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

M.A. Ahad Shikder
20301316

Approval

The internship titled “Software Engineer Intern at Polygon Technology” submitted by

1. M.A. Ahad Shikder(20301316)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 20, 2025.

Examining Committee:

Supervisor:
(Member)

Md Sabbir Ahmed
Lecturer
Department of Computer Science and Engineering
Brac University

Program 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 & Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

I would like to state clearly that everything written in this report is based on my own learning and experience during my internship at Polygon Technology. I didn't copy from anyone or use material from outside sources without giving proper credit. Some sections are inspired by what I've learned from team discussions or online materials, but I've always tried to put things in my own words and mention sources when needed.

This report wasn't prepared by anyone else and hasn't been submitted for any other purpose before. I've tried my best to stay honest and sincere throughout the internship and while writing this report. I understand the importance of academic integrity and tried to stick to it as much as possible.

Abstract

The internship report will provide an overview of my experience as a software engineer intern at Polygon Technology. This is a learning-based internship. The internship curriculum covers many areas, starting with C programming basics to provide a solid foundation. After learning C programming, I will move on to Java programming. I will learn data structures and algorithms in Java. The internship also covers HTML and CSS web design, web frameworks, MVC architecture, ORM, and REST API programming. Git version management, including distributed version control, fundamental workflows, branching, merging, and GitFlow, is also part of this curriculum. The internship also covers Docker basics, including images, containers, and Docker Compose. During this internship, I will also learn object-oriented programming, functional programming, package management, and version control. The final report will discuss internship challenges, lessons learned, and growth.

Acknowledgement

First of all, I want to thank Almighty Allah for giving me the strength and patience to go through this internship and finally complete the report.

I'm really thankful to Polygon Technology for welcoming me as an intern and giving me the chance to learn and grow. The experience I had there was truly meaningful. I'm especially grateful to the mentors and team members who were always there to guide me, answer my questions, and help me when I got stuck.

I also want to thank Mr. Md Sabbir Ahmed, my academic supervisor from BRAC University. His guidance, suggestions, and kind support made a big difference throughout the internship.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Acknowledgment	iv
Table of Contents	v
1 Introduction	2
1.1 About Internship	2
1.2 About this Report	2
1.3 Objective	3
1.4 Methodology	3
2 Company Profile	4
2.1 Overview	4
2.2 Vision and Mission	4
2.3 How Does Polygon Work	4
2.4 What Polygon Does	5
2.5 Recruitment process	5
3 Internship Objectives	6
4 Training Phase	7
4.1 Overview	7
4.2 Getting more Familiar with some Programming Languages	7
4.3 Assistance	8
4.4 Communication and Meetings	8
4.5 Work Environment	9
5 My Beginnig in Polygon	10
5.1 How it Started	10
5.2 Getting Started With C	10
5.3 Moving to C++	11
5.4 Group Work	11

6	Learning Linux and Bash Scripting	12
6.1	Introduction to Linux and Bash Scripting	12
6.2	Directory Structure and User Management	12
6.3	Permissions and Access Control	13
6.4	Command-Line Basics	13
6.5	Small Project: Automating Tasks with Bash Scripting	13
7	Learning HTML and CSS	15
7.1	Basics of HTML	15
7.2	Basics of CSS	15
7.3	Designing a Landing Page for Portify	16
7.4	Learning Bootstrap and Tailwind CSS	17
7.5	Redesigning the Portify Landing page with Bootstrap	18
8	Learning JavaScript	20
8.1	Syntax and Programming Basics	20
8.2	Hoisting and Variable Scopes	20
8.3	Objects	20
8.4	Function and First-Class Functions	21
8.5	Asynchronous Programming	21
8.6	Closure and Callback	21
8.7	Promis, Async/Await	21
8.8	DOM Manipulation and Event Handling	21
8.9	ES6 Syntaxes	22
8.10	WebSocket Programming	22
8.11	Project: Building a Live Simple Chatbot Using WebSocket Program- ming	22
8.12	Projects: Rock Paper Scissors and Tic-Tac-Toe	23
	8.12.1 Rock Paper Scissors Game	23
	8.12.2 Tic-Tac-Toe Game	23
9	Backend Development Journey and Tooling	25
9.1	Understanding Web Technology Basics	25
9.2	Database Concepts and Indexing	26
9.3	Mastering Java Syntax and OOP Principles	26
9.4	Learning the Spring Boot Framework	26
9.5	Version Control with Git and GitHub	27
9.6	Introduction to Docker	28
10	Projects	29
10.1	Project 1-Presencia: Human Resource Management and attendance System	29
	10.1.1 Technical Architecture & Design	29
	10.1.2 Core Features	30
	10.1.3 API Endpoint Summary	31
	10.1.4 Database Schema Overview	32
	10.1.5 Application User Interface Showcase	33
10.2	project 2 - ErrorLog Tracker	40
	10.2.1 Project Overview	40

10.2.2	Technical Architecture & Design	41
10.2.3	Architectural Design	41
10.2.4	Core feature	42
10.2.5	API Endpoint Summary	42
10.2.6	Application User Interface Showcase	43
11	Conclusion	47
	References	48

Chapter 1

Introduction

1.1 About Internship

Students and graduates can benefit from gaining real-world experience through internships, which also provide them with the opportunity to expand their practical skills and expertise. This gives them the opportunity to put their academic knowledge into practice on a platform that is provided to them. Through the participation in internships, students are afforded the chance to transition into the working world and acquire practical experience in the subject that is relevant to their chosen curriculum. In addition, students have the opportunity to add this experience to their resume, which will be beneficial in the future when they are looking for employment. In addition, internships give students the opportunity to gain experience working in an office setting. In the past, students were more likely to choose to work on their thesis, but in today's world, students are considering internships to be just as important as their thesis. The concept of them is gaining more and more popularity. In order to graduate from our university, we are required to either finish our thesis or perform an internship. Students are choosing internships in addition to studying for their thesis. Students are required to do an internship of their choosing that lasts for a period of minimum six months. In order to fulfil the requirements for graduating from BRAC University, I made the decision to participate in an internship rather than writing a thesis paper. Polygon Technology is where I am currently completing my internship. A software company that offers its services to customers all over the world, it is a software company. Bangladesh serves as its home base. Providing varied and complete technological solutions that are adapted to the specific requirements of businesses, delivering holistic ways to handle many elements of technology requirements, and assuring optimal outcomes and success are the primary focusses of the organisation.

1.2 About this Report

The following report details my time spent working as an intern in the software engineering department at Polygon Technology. In addition to that, I have written about the information and experience that I have learnt up to this point in my internship. The purpose of this report is to present an overview of my experience as an intern up to this point, as well as the project that I have been engaged on.

1.3 Objective

The objective of this report is to provide readers with the following information: 1 The primary objective of the report is to convey the knowledge and expertise that I have gained through my professional experience.

1.3.2 Specific Goals The goals for this report are as written below:

- A brief summary of the company is provided.
- Describe the atmosphere of the workplace.
- Providing an explanation of the operation of the Polygon Technology
- Writing on the things I've learnt and what experiences I've obtained during my internship up to this point
- The work that I have completed on the project that has been allocated to me up to this point
- A representation of the information that I shall acquire by the time my assignment is finished

1.4 Methodology

The experiences that I have obtained throughout the course of my internship up to this point are the primary topic of this report. In this study, the majority of the information is based on my own personal experiences. In addition to that, I have obtained some information from the website of the company as well as from other specific websites.

I collected data from the following:

Primary Data

- My personal experience
- Work meetings

Secondary Data

- Polygon Technology Website

Chapter 2

Company Profile

2.1 Overview

Polygon Technology is responsible for the entire process, beginning with the inspiration of consumers and continuing through product market fit, user interface and user experience design, and development. A comprehensive market study, iterative design, agile development, and continual improvement based on user input and analytics are all components of their methodology. They provide superior products and user experiences by keeping aware of the latest developments in the business and putting the requirements of their customers first. Making certain that you receive the very best software product available. The main office of Polygon Technology is situated at Banani,Dhaka. Most of the work of the company is being done there. The working environment is very nice and everyone is very cooperative and responsive. All the employees of the company are from well reputed universities and have a lot of experiences in the tech industry.

2.2 Vision and Mission

Polygon Technology empowers businesses with innovative software solutions that boost growth, efficiency, and success.Polygon Technology cares about customers specific needs. Provide high-quality, customisable software development. Polygon Technology builds great collaborations to grow and exceed objectives.Their focus is on customers. They want to become a trusted software development service provider and boost Bangladesh's IT export through industry experience and customer satisfaction.

2.3 How Does Polygon Work

Polygon is a software firm. Polygon makes software solutions as their customer requirements.Firstly, Polygons sales team met with customers and get the idea of what they need. Then sales team briefs the engineering team about the project. Engineers works to make a demo according to the customers requirement. Sales team shares the demo with customer. If the customer satisfied with the demo, project gets locked and Polygons responsible engineering team start working to

deliver the project before deadline. Polygons talented and optimistic engineers can design complex software solutions as they are vastly experienced.

2.4 What Polygon Does

Polygon is one of the country's best software firm. The company provides below services: 1. Software maintenance service: polygon ensures the longevity and optimal performance of customers software. Its experienced team takes care of bug fixes, updates, security enhancements, and performance optimizations. Allowing customers to focus on your core business activities, and enjoy a stable and reliable software solution. 2. UI UX design service: Polygon establishes a seamless design process, meets deadlines, and delivers impeccable results. 3. Software development service: polygon is experienced in making fintech solutions, online marketplaces, agro-tech solutions, machine learning projects etc.

2.5 Recruitment process

Polygons frequently hire engineers and interns. Its website has a section of hiring notices. It also shares the hiring notices on social media platforms. When you drop your CV, it asks you for a written exam. After clearing the written exam a interview board will judge you with knowledge of the interested field and your basics. They especially give importance to problem solving if the job is for software engineers.

Chapter 3

Internship Objectives

My internship is 6 months long and I hope to gain a lot of practical experience throughout this period. I am interning as a Software Engineer. The objectives and goals I aim to achieve during my 6 month internship are as following:

- I want to gain practical experience. I want to put the theoretical knowledge about programming I have gained through my academics to practical use.
- I want to learn how the office environment works and how it feels to to take a step into the real-world.
- I want to acquire and enhance my technical, problem-solving skills that would be helpful in the future.
- I want to improve my communication and interpersonal skills by working in a diverse team.
- I want to be familiarised with the programming languages I'll be working with, so I can be more comfortable and competent in using them.
- I want to improve my problem-solving skills that would help me in dealing with real-word challenges and my critical thinking and analytical skills.
- I aim to develop professional relationships that would help my interpersonal abilities and communication skills.
- I want to gain confidence and professionalism through improving my time management, adaptability and multi-tasking skills. I aim to gain enough confidence and skills necessary to succeed in my field.

Chapter 4

Training Phase

4.1 Overview

Participating in an internship at a company is comparable to taking a step into the real world and entering the realm of employment. Students who participate in internships are better able to transition from the academic world to the working world. An important turning point in a student's life is when they are able to secure an internship with a recognised company. Through training, the employers educate the intern on their work ethic as well as the specific norms and standards of the industry. My internship at Polygon Technology was in the field of software engineering, and I was given the responsibility of working in that department. Due to the fact that it will assist me in acquiring the essential experience and expertise, this chance is quite significant to me. It will also assist me in gaining the practical experience that I need to put the theoretical information that I have obtained via my academics into practice. In order to graduate from BRAC University, it is necessary for me to finish this internship requirement. Prior to beginning my employment here, I had no prior knowledge of how workplaces operate or how interns carry out their duties. Every step of the way, my supervisor at Polygon Technology has been there to assist me and has been teaching me about the importance of work ethics. The official training portion consisted of my supervisor providing me with comprehensive guidance throughout the process.

4.2 Getting more Familiar with some Programming Languages

In order to refresh my memory and become more familiar with several front-end and back-end languages that would be helpful to me in my work, my supervisor informed me that I needed to study some things before I formally started my work. As a result, I was required to take some courses as part of my company's training program. During my time at university, I was not very good at programming, and I had very little experience with learning C and C++ programming. Which is of utmost significance for the field of software engineering. Therefore, I had to begin the process all over again. Before I started working in software development, my supervisor wanted me to strengthen my problem-solving skills as much as there was room for improvement.

4.3 Assistance

It was my supervisor who served as my major mentor and provided me with guidance throughout the duration of the internship up to this point. The time that he set aside for me was used to instruct me on subjects that I did not comprehend as well as the correct way to carry out activities. In addition to that, I visited the internet and conducted independent study on the subjects that I was unable to comprehend. Many of the programming that I found to be puzzling were the subject of some tutorials and articles that I read. There were a few subjects that I was confused about, and my supervisor assisted me in gathering further information on those subjects. Although I did not have a lot of prior experience with this language, he was the one who instructed me in the fundamentals of C and C++. Moreover, he instructed me on the fundamentals of competitive programming in the C++ programming language. These were containers, and iterators.

4.4 Communication and Meetings

The Polygon Technology work culture is based on the principles of open communication and on agile and responsive collaboration, and it combines both formal and informal, real-time interactions to facilitate the alignment of projects and the ongoing updates and learning. My experience as an intern was fully connected with this communication structure.

The most common form of my everyday interaction was based on the organized check-in with my supervisor. Each of these sessions, scheduled most often twice a day, was a small version of the stand-up, during which we would look over the work accomplished, ask about any technical issues or roadblocks I had run into, and define the specific goals of the next period of work. This repetitive feedback loop was a goldmine; it was not just required to provide feedback on the progress so far; but to actively solve problems. We would mostly use pair debugging when discussing some non-trivial issue in such meetings, and my supervisor could offer some insights on how things are usually done better, how to structure code, and how to properly solve a problem quickly.

To establish a real-time, asynchronous communication, common collaboration software, such as Slack (or any messaging system) was used by the team, enabling rapid knowledge sharing and asking questions without breaking the flow of deep work. This was augmented by more structured, calendarized kick-offs of projects to start or team syncs once a week to talk about bigger project objectives and milestones. Much of the documentation and software development took place on git and github. Formal channel of Feedback was the procedure of making pull requests of code changes. My colleagues and seniors (supervisor and other senior engineers) would also perform adequate code reviews leaving constructive comments and suggestions. This practice has helped me get cleaner, easier to read, and efficient code, as well as understand how it is important to write maintainable software when working with the team.

Moreover, the organization provided a healthy cooperative learning environment and a weekly LeetCode session was an outstanding top point of this. During these group meetings, group members would share the solutions of the difficult problems, different methods of approach and complexity of time and space of varied algorithms.

This helped not only in my problem solving skills but also in my in-ability to express technical divulgements and learn through the different visions of my colleagues.

4.5 Work Environment

The work environment in Polygon Technology is very friendly. Everyone is very supportive. And everyone encourages me to work properly. The office campus is very green and open. Polygon technology is a mix of modern design and practicality, making it a great place to work together. The open layout, with comfortable furniture and lots of natural light, helps us communicate and collaborate easily. We have quiet areas for focused work and colorful spaces for brainstorming and relaxing. Plants and artwork make the office feel welcoming and creative. Overall, it's a professional yet comfortable environment that supports both productivity and well-being.

Chapter 5

My Beginnig in Polygon

5.1 How it Started

At Polygon Technology during my first days I started learning C. At first I installed C and started coding in VS code. And started learning the syntaxes of C. Then learned basic variables, control flows , functions after that started solving basic problems of loops and recursions. And then moved onto online problem solving sites. At first I started solving problems in Beecrowd. And then Moved into Codeforces. In Beecrowd I've Solved 150 problems and in Codeforces I've solved 148 problems. I've solved problems In Beecrowd using the C programming language and Solved problems Codeforces using C++.

5.2 Getting Started With C

At first I started learning The basics of C and Syntax by watching several tutorials and documentations on the internet. I've learned variable types such as floats, integers. And data types such as long, double. And then control flows such as loops, nested for loops and if else statements and recursion and solved some basic problems to get familiar with C. Then Solved some pattern problems for better understanding of loops. Also matrix related problems of finding trace of matrix and transpose of matrix. These problem solving helped in better understand of coding and increase my knowledge regarding problem analysis. After building a solid base in C programming I've started solving problems in Beecrowd. An online problem solving platform. In that problem solving site I have solved total 150 problems. My progress bar in Beecrowd is given below:

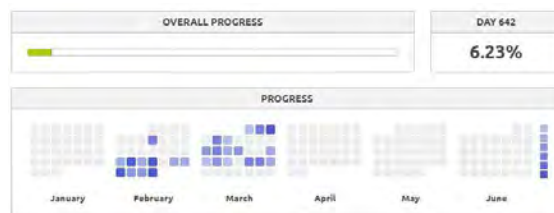


Figure 5.1: Beecrowd Progress



Figure 5.2: Codeforces progress

5.3 Moving to C++

After finishing 150 problems in Beecrowd my supervisor suggested me to move into Codeforces to solve 150 more problems there. He told me to start with 800 level problems and move onto 900 and 1000. Alongside this Problem solving period my supervisor had a session with me regarding STL library which is very helpful for competitive programming and helpful for solving problems in Codeforces. I've learned Containers in those sessions. Which consists of maps and vectors. These data structures are quit helpful in terms of problem solving in Codeforces. Frequency related problems were solved using map .It helped enhance my problem solving skills. My Codeforces progress is given below:

5.4 Group Work

When I completed my internship at Polygon technology, most of the projects that took place involved cross functional teams working in a collaborative environment throughout my internship. Whereas I was in charge of creating the backend of the Presencia project on Spring Boot, another teammate took charge of the frontend programming on the Next.js framework. This involved constant communication and co-ordination to achieve API with consistency and successful data integration.

APIs documentation and endpoint testing according to Postman in general and the debugging process in particular were typically performed in one and the same manner and with a feedback loop developed between frontend and backend developers. There is a collaboration on integration solutions, coordination of feature releases and focus on bug fixes. As an example, I opened REST APIs to the employee attendance and leave management functions, and the frontend team could give feedback on UI and enhance the error handling and formats of responses.

Besides this, there was a team-wide meeting to discuss schema designs, specifications of features and Docker-based deployments. Not only did this experience of teamwork give me a boost to the technical communication skills, but it also allowed me to realize the significance of teamplay in contemporary software development processes.

Chapter 6

Learning Linux and Bash Scripting

6.1 Introduction to Linux and Bash Scripting

In the phase of learning Linux, I focused on the Linux Operating Systems and Bash Scripting on the most basic level. This stage was very important in equipping me with the skills that I needed to handle more challenging tasks and projects in the later stages of the internship. Using Google and other resources, I dived heels first into Bash scripting. Bash scripting is a very useful feature of Linux because it allows for the automation of tasks. I learned the basics of building and executing scripts, including the use of variables, loops, and conditionals to produce functional and reusable pieces of code. My learning also covered command-line parameters and their use, as well as functions aimed at making redirection of input and output that much easier. To cement my knowledge on certain principles, two little tasks were presented at the end of the training. Creating a script to automate a particular task, which helped to further deepen my understanding and showed real-life applications of scripting.

6.2 Directory Structure and User Management

Along with Bash scripting, I also went through the directory structure on Linux and was able to know the rationale behind and the way major directories were organized. For instance, I was taught that important user commands are stored at `/bin` and `/usr/bin` locations, whereas system configuration files are kept at `/etc`. `/home` and `/root` are directories that act as user home directories and `/var` is used for changeable data such as logs. This understanding proved to be very useful because I could easily look for important files and applications and was able to navigate the file system effortlessly. Another vital facet to my study was user administration and the Linux root user. I was able to use commands such as `useradd` for creating users and `userdel` for deleting users as well as manipulate user characteristics through the use of the command `usermod`. Further, I learned about the importance of the root user and also about `sudo` in executing administrative tasks in a safe way. All the learning about user switching using the `su` command helped me in better managing multiple user environments.

6.3 Permissions and Access Control

I learnt about file permissions and access control in Linux, in addition to user administration. I learnt how to use programs like `chmod` to see and adjust file permissions, as well as how to change file ownership with commands like `chown` and `chgrp`. I practiced working with both symbolic and numeric permission modes, which helped me understand how important permissions are for keeping the system secure and managing access to files and folders.

6.4 Command-Line Basics

In an attempt to enhance my command-line skills, I have taken the time to grasp the fundamentals of the command line. Important commands dealing with file operations, namely `ls`, `cd`, `cp`, `mv`, `rm`, `mkdir`, and `touch`, as well as text-processing tool commands such as `cat`, `grep`, `awk`, and `sed`, were practiced. System monitoring commands such as `ps`, `top`, `htop`, `df`, and `du` were also explored. The major functions helped me in system monitoring and resource management.

6.5 Small Project: Automating Tasks with Bash Scripting

Interfacing with the API was one task during my learning phase in linux. Hence, a script was developed for the authentication of users to log in to a system and obtain a token to use for further authenticated requests. A Bash script was thus developed using the `curl` tool to POST to the specified URL (<https://accounts.dev-sheba.xyz/api/v3/profile/login>) mentioned for login purposes. This script takes the user's email and password as the command-line parameters and sends them to the API as form data. I used the 'jq' tool to acquire the message and the token data from the JSON response after the API has answered. The message contains feedback about the login attempt, including whether it was successful. Concurrently, the token was saved to a file called 'token.txt' for future use. This token will henceforth serve as the authentication key for any API calls that would be run in the future. Finally, the script gives a prompt indicating to the user the feedback about the login process. Another key task was to obtain attendance data from an API by using the token that was obtained during the login process. This required creating a second Bash script that reads the token from `token.txt` and uses it to perform an authenticated GET request to the attendance endpoint (<https://api.dev-sheba.xyz/v1/employee/attendances>). The script builds the API URL with query parameters for the year and month, which would allow the attendance data to be filtered for a given time period. The script sends the request with the token contained in the `Authorisation` header by using the `curl` command. The API's response is then processed with `jq` to get the `code` field, which shows if the call was successful (for example, a code of 200 denotes success). If the request was successful, the script will extract the statistics data from the response. This field provides information such as the total number of working days, the number of days off, and the attendance percentage. This data is stored in a file called `statistics.txt` and is also printed to the terminal

for easy viewing. This work showed that I am capable of managing authorised API requests, processing JSON data, and automating data retrieval operations.

Chapter 7

Learning HTML and CSS

7.1 Basics of HTML

After I gained a fair understanding of Linux and Bash programming, I branched into learning something about HTML and CSS[9], which are the primary technologies for developing and styling web pages. The journey for me began from HTML, the HyperText Markup Language, which is the very mark of existence for any web page. I studied how to use very basic elements like ‘<html>’, ‘<head>’, and ‘<body>’ to give a valid basic structure of a web page. I went on to look at how to add metadata like page title ‘<title>’ and associate with other resources such as stylesheets using the ‘<link>’ element in the ‘<head>’ section. I have worked on numerous HTML tags for formatting text in the ‘<body>’ section. For example, I wrote headings ‘(<h1> through <h6>)’ to create a hierarchy of information, used paragraph tags ‘<p>’ to import textual content, and added hyperlinks ‘<a>’ to navigate between pages and resources. Also included were images ‘’, lists ‘,,’, and tables ‘<table>,<tr>,<td>’, which help structure content in an understandable and logical way. I also have learned a lot about semantic HTML, including tags ‘<header>’, ‘<footer>’, ‘<nav>’, and ‘<article>’. This group of elements helps web pages to be indexed by search engines and human perception.

7.2 Basics of CSS

By the time I’d gotten to be efficient in using HTML, my next goal was to fathom CSS (Cascading Style Sheets); the means of styling and visual enhancement of HTML entities. I started straightway by learning how to apply styles using different types of selectors: element selectors (e.g., h1), class selectors (e.g., .class-name), and ID selectors (e.g., #id-name). I explored a variety of CSS properties that could be used to influence certain design aspects of websites; these include color for coloring text, font-size for formatting text size, background-color for formatting background, and border for placing borders around elements. Inline with this, I learned about the box model, which deals with how elements are arranged in CSS concerning their margin, border, padding, and content. This learning helped me view these elements concerning measurement and spacing, to synchronize my layouts with attributes of beauty and functionality. I have worked with various CSS layout techniques to enhance the dynamism and responsiveness of my websites. I have learnt flexible box layouts, a formidable layout model that allows flexible and efficient

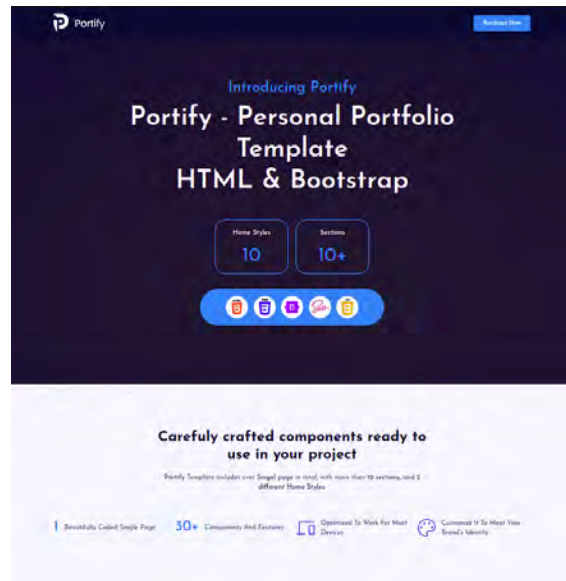


Figure 7.1: A Landing page

alignment of items within a container. Because of flexbox, I could easily build complex layouts including center-justified items, equidistant spacing, and alignment of items in specified directions. I have also learned CSS grid, which is a technique for laying out two-dimensional objects in a row and column format. The grid allows me to create more advanced and responsive web pages that can respond to different screen widths. I was also experimenting with media queries to apply different styles based on the screen size of the device. This way, I could ensure that my web pages looked good on both desktops and mobile devices.

7.3 Designing a Landing Page for Portify

Once I learned the bare-bones of HTML and CSS, I was assigned the task of working on a landing page called Portify. Portify is a personal portfolio template created in HTML and Bootstrap. The goal was to build an appealing and responsive page that would sell the benefits and features of the template with clarity. The page was segmented into three major portions, namely a navigation bar, a main body area, and a footer. Each section is meant to be clean, modern, and user-friendly.

The navigation bar is at the top of the page, with a logo on the left side and a "Purchase Now" button on the right side. Due to a CSS filter that inverts the logo's colours, it became bright and easy to see against the dark background. The "Purchase Now" button is designed in white font against a cornflower blue background; the hover effect transforms the button's background to orange, thus creating a dialogue with the user. Flexbox was utilized to build that region so as to align the logo and the button horizontally while leaving a balanced space between them. With a dark background and white text, the main body grabbed page attention. A tagline ("Introduction to Portify") was set in a smaller font and lighter colour than the heading. The main headline, "Portify - Personal Portfolio Template HTML & Bootstrap," was set in larger typeface to attract attention. Following below the headline is a banner carrying two boxes that display important information about the template, such as home styles and parts available. These boxes have a unique look

because of rounded borders and a cornflower blue outline. A pill-shaped container I also designed has clickable icons for technologies like HTML, CSS, Bootstrap, Sass, and JavaScript. The hover effect for each icon was an increase in size, creating an energetic feel for the design. The use of Flexbox to centre the main body created visual balance and easy-to-read material. The footer was developed to provide more information about the template. Starting off with a heading empowering the quality and usability of the template, it drives into a description showcasing its features; for example, the number of single pages, sections, and home styles. A grid arrangement was then created below to show the main advantages of the template: beautiful landing single pages, over 30 components, device optimisation, and customisation options. Each benefit was given some embellishment with a little icon or number to make them more attractive. The footer has a background graphic for visual enhancement, and the content was centred using flexbox to keep it neat and professional. To apply the design I had in mind, I used a variety of CSS techniques to create the design I wanted. I relied highly on flexbox for the task of layout; it helped me create flexible and responsive designs. I thought hover effects would work out well with interactable components such as the Purchase Now button and technology symbols to give the page more fun splatters of interest. I also made sure to consider typography and the whole visual palette properly. I set up a consistent colour scheme, using cornflower blue as the major accent colour and white for text so that there is a clear visual contrast on dark backgrounds. I also made use of borders and rounded corners to treat elements such as the banner boxes and pill-shaped container with a nice modern and professional feel. The issues that I came across during the development of this page range from checking responsiveness to ensuring that the design was consistent across various areas. I worked on CSS properties such as flex-wrap, gap, and media queries to overcome certain problems. This helped me learn how to adjust layouts based on different screen sizes and also imparted to me the necessity of responsive design. Here, I gained some hands-on experience in building actual web pages with semantic HTML, designing web pages with CSS, and applying interactive elements to improve user experience. This assignment has helped me strengthen my fundamentals of web development, preparing me for complicated works in the future.

7.4 Learning Bootstrap and Tailwind CSS

Once I understood at a basic level HTML and CSS, I went on to learn Bootstrap [5] and Tailwind CSS [13]. Two of the most common CSS frameworks, these ease the process of designing and building websites that look good and are well-behaved across many devices. They allowed me to work better and to bring life to complicated layouts with very little custom CSS. I will describe their application in the next section: to remodel the Portify landing page using Bootstrap and to create a portfolio using Tailwind CSS.

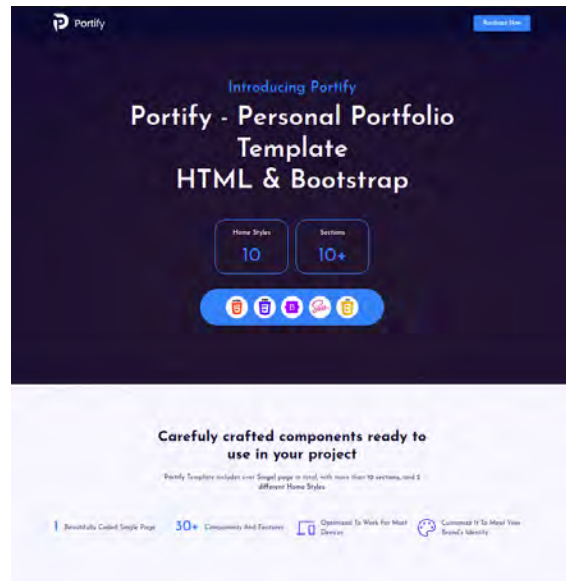


Figure 7.2: A Landing page

7.5 Redesigning the Portify Landing page with Bootstrap

From my learning in Bootstrap, I set out to redesign the Portify landing page, using Bootstrap's ready components and utility classes. This is beneficial for keeping the overall design responsive and fairly consistent visually, with custom CSS written very minimally. Below, I will elaborate on the use of Bootstrap classes across the page pertaining to structure and styling. I started with the basic setup of including Bootstrap's CSS and JS files in an HTML file via a CDN link. This allowed me to take advantage of Bootstrap's grid system, components, and utility classes for a fast page buildup. For the navigation bar, I had made use of the navbar component of Bootstrap. This replaced the custom CSS I was working with; as a result, the navbar became responsive and uniform within the different devices.

- navbar: Defines the navbar container.
- container: Centers the content and provides padding.
- navbar-brand: Styles the logo or brand name.
- btn btn-primary: Styles the "Purchase Now" button with Bootstrap's primary color scheme.

Bootstrap utility classes were applied to center and design the page. The central section comprised a tagline, a main title, a banner with stats, and a pill-shaped box to hold technology icons.

- container-fluid: Makes the container span the full width of the viewport.
- text-center: Centers the text horizontally.
- d-flex flex-column justify-content-center: Uses flexbox to center content vertically and horizontally.

- `border border-primary rounded-3`: Adds a border with rounded corners and the primary color.
- `bg-primary rounded-pill`: Styles the pill-shaped container with a primary background color and rounded edges.
- `rounded-circle`: Makes the technology icons circular.

I came up with the footer to dish out further information regarding implementation. For a responsive layout encompassing the content, I employed the grid system of Bootstrap.

Chapter 8

Learning JavaScript

8.1 Syntax and Programming Basics

Once I became adept in HTML, CSS, Bootstrap, and Tailwind CSS, I proceeded to learning JavaScript [3], a strong programming language aimed at the dynamic and interactive web development. JavaScript stands tall as an undergirding element of any real-time web application, and its good knowledge is vital for creating workable, useable sites. I now give further discussion on the subject matter that I mainly studied during the time of my internship in JavaScript phase. In this software engineering program, I first covered the fundamentals of JavaScript, its syntax, and core programming concepts. This included an understanding of variables(let, const, var), data types(strings, numbers, booleans, objects, arrays), operators(arithmetic, comparison, logical), and control structures(if-else, switch, loops). Learning this way has built a strong foundation for good and efficient JavaScript coding.

8.2 Hoisting and Variable Scopes

Then I read up about hoisting, the JavaScript behaviour that raises variable and function declarations to the top of their scope during compilation. With this, I understood why variables declared with var can be accessed before being declared, whereas those declared with let and const cannot be accessed before their declaration; this gap is termed the temporal dead zone. I also studied variable scopes, like global scope, function scope, and block scope, which are crucial for controlling variable accessibility and obviating clashes in bigger programs.

8.3 Objects

I took quite a bit of time to studying about objects since JavaScript is an object-oriented programming language. I learnt to create and modify objects with object literals, constructors, and the class syntax introduced in ES6. The topic of properties, methods, and prototypes was also quite important for me in organizing and reusing code. I also learnt object destructuring, a handy way to extract properties from objects into variables.

8.4 Function and First-Class Functions

Function has been an essential part of JavaScript, and I have learnt the proper way to define and use them. I researched function declarations, function expressions, and arrow functions, which were all introduced under ES6. Another concept I learnt about was first-class functions: functions that can be assigned to variables, passed as arguments, and returned from other functions. This gives rise to various complex programming patterns like higher-order functions and callbacks.

8.5 Asynchronous Programming

Asynchronous programming is one of the essence of JavaScript, especially when it comes to managing investigating activities such as API calls, file operations, and timers. The very first thing I learnt was that callbacks are functions passed in as parameters to other functions and are called upon the completion of a task. Nevertheless, I also recognised the limitations associated with callbacks, one being callback hell, which led me into studying more advanced mechanisms like Promises and `async/await`.

8.6 Closure and Callback

I spent some time learning about closures in JavaScript that are really useful. A closure is when an inner function still has access to the variables of the outer function even after the outer function is gone. Generally, closures are used for data encapsulation and to create private variables. I also spent some time with callbacks to handle asynchronous functions like retrieving data from an API or processing user input.

8.7 Promis, Async/Await

I have learned about Promises, which are meant to indicate the eventual success or failure of an asynchronous operation, and let the handling of asynchronous code in a more elegant way. Promises provide chaining of `.then()` and `.catch()`, which makes such code cleaner compared with the use of nested callbacks. I also looked at `async/await`, which is syntactic sugar on top of Promises to write asynchronous code in a synchronous way. Thus, my code now looks cleaner and is easier to debug.

8.8 DOM Manipulation and Event Handling

Possibly one of the best features in JavaScript is the ability to communicate with the Document Object Model (DOM). I studied how to use the methods `getElementById`, `querySelector`, `createElement`, and `appendChild` to select, create, edit, and delete HTML elements. I also dove into event handling, where I applied event listeners to respond to user activities such as clicks, mouse movements, and keyboard input. This has enabled me to create dynamic and interactive web pages.

8.9 ES6 Syntaxes

familiarized myself with the ES6 (ECMAScript 2015) features, which brought significant improvements to JavaScript. Key topics included:

- Arrow functions: A concise syntax for writing functions.
- Template literals: Embedding expressions inside strings using backticks.
- Destructuring: Extracting values from arrays or objects into variables.
- Spread and rest operators: Handling arrays and function arguments more efficiently.
- Modules: Organizing code into reusable modules using import and export.

These features made my code more modern, readable, and maintainable.

8.10 WebSocket Programming

Moving on, I studied the programming of WebSockets [2] as a medium for bidirectional real-time communication between a client and a server. I learned to establish a WebSocket connection between the client and server and to send and receive messages. I also learned to handle some events: open, message, and close. This knowledge is important in building real-time applications such as chat applications, live notification, and collaboration tools.

8.11 Project: Building a Live Simple Chatbot Using WebSocket Programming

Basic real-time chat programs were created using some of my knowledge of WebSocket programming in an attempt to learn JavaScript. The project provided wonderful opportunities to experimentally see how real-time and bi-directional communication between a client and a server operates, which are essential in creating interactive and dynamic web applications. The project anticipated the development of a chat program for sending and receiving messages in real time, showcasing the strength of WebSocket programming in providing continuous chatter without sending constant HTTP queries. To handle the operations of the chatbot, I started off by designing a WebSocket server based on Node.js using the ws package. The server accepts WebSocket connections and handles incoming message events. It serves the purpose of creating a unique ID for each user and broadcasting the welcome message to all connected clients upon a client connection. I designed a simple chat client-side interface with JavaScript. The interface includes input fields to type messages, a button to send the messages, and a message area that shows the history of the chat. In the beginning, when a user triggers the chat application, the client uses the JavaScript WebSocket API to set some open communication between a client to a server. The connotation behind having that open connection is the immediate communication between a client and server. Once a user has sent a message, the

client sends the message in the WebSocket connection. The server receives the message, processes it, and publishes it out to all clients connected. The client received the message and updated the chat interface to reflect the new messages in real-time. This ensures that every participant in the chat room gets the messages without problems.

I developed event handlers for dealing with different WebSocket events such as open, message, close, and error. The open-event gets fired when the connection is established; the message-event gets fired on receipt of message; the close-event gets fired when the connection is terminated; and error-event gets fired during an error situation. These event handlers ensure that the program responds appropriately for different situations such as connection issues and disconnection by users.

8.12 Projects: Rock Paper Scissors and Tic-Tac-Toe

My learning route involved using JavaScript to create two interactive games called Rock-Paper-Scissors and Tic-Tac-Toe. I made these projects specifically to strengthen my understanding of DOM manipulation, event management, and basic game logic using vanilla JavaScript, HTML, and CSS. These games serve as a great opportunity to practice coding principles while creating exciting and enjoyable user experiences.

8.12.1 Rock Paper Scissors Game

Rock Paper Scissors is an age-old hand game with the player competing against the computer. This project aimed at developing a simple yet fun game that would test my skills in user-input handling, game logic application, and dynamic user-interface refresh. Therefore, the player will first choose any of the three options: Rock, Paper, or Scissors. The computer's option is randomly generated using the `Math.random()` function in JavaScript. After that, the game compares the player's option with what the computer picked to determine a winner based on the Rock Paper Scissors rules: Rock beats Scissors, Scissors beat Paper, and Paper beats Rock. In case both options are found to be identical, the game will be declared a draw. The user interface was designed in HTML, with the style in CSS. The UI had buttons for the user to make selections and an output area to show the results. The DOM was manipulated to give real-time results. For instance, the user's option, the computer's choice, and the result (win, lose, or draw) were presented in real-time. The advantage was that, in addition, the game kept score for the user and the machine over several rounds for added fun. The UI was designed responsibly, which means the game would run seamlessly on screens of all sizes. This was a fun and dynamic way for me to learn more about JavaScript, and it assisted me greatly in solidifying concepts such as event listeners, conditional expressions, and DOM manipulation.

8.12.2 Tic-Tac-Toe Game

The game of Tic-Tac-Toe is a game for two players in which players take turns marking spots in a grid of three squares wide and three squares tall. The main objective of this project was to implement a working game where two players can



Figure 8.1: Rock Paper Scissors Game



Figure 8.2: Tic-Tac-Toe Game

play against each other, where the game determines if there is a winner or a draw. The logic of the game was coded using JavaScript. Players took turns clicking on the grid to place their markings, either an "X" or an "O." The game checks for a winner after every move by testing its rows, columns, and diagonals. If all spots are filled and there is no winner, then it is a draw. The user interface was created using HTML and stylized using CSS. A 3x3 grid was used for the game board while a text display indicated the current player's move as well as the result of the game. The grid consisted of div components styled via the CSS grid layout, thus ensuring that the layout looks neat and orderly. The game board was updated in real time via the DOM manipulation, allowing a display of players' moves regarding the status of the game. Adding a turn indicator - that indicates the current player's turn (X or O) - as well as a reset button for players to start a new game without refreshing the page definitely enhanced the user experience. The project helped me further understand things such as JavaScript arrays, loops, and DOM manipulation, as well as pushed me to seek an approach to present fairly complicated game logic as intuitively as possible.

Chapter 9

Backend Development Journey and Tooling

Over the course of my internship, my technical growth progressed through a structured path that took me from understanding the web ecosystem to developing and deploying backend applications. This chapter summarizes the stages of my journey in the form of subchapters covering web technologies, databases, Java, Spring Boot, Git, Docker, and Postman.

9.1 Understanding Web Technology Basics

Once I became familiar with JavaScript, I now concentrate on gaining the basics of Web Technology. This stage was very essential to the comprehension of the behind the scenes workings of web applications. I learned all these and discovered the anatomy of requests and responses, as well as how browsers talk to servers through the protocol HTTP. I researched the basis of the web architecture in client-server architecture and the difference between dynamic, data-driven and static websites. I had a clue on the client-server paradigm where by an individual (client, mostly browser) initiates a request to a web server, which processes it and responds. I also learnt different HTTP methods such as GET, POST, PUT and DELETE and the manner in which they correspond to some of the common data operations such as data reading, creation, update and deletion. My experience with learning request/response cycle allowed me to gain the level of understanding of what actually is going on under the hood of a webpage visited by a user and the use of status codes and response payloads.

Moreover, there was the learning on what web hosting is and so there was learning in DNS resolution and translation of domain names into IP addresses. I also got acquainted with REST APIs [1] and the method of exchanging information between the front-end and backend services through requests written in JSON. I performed interaction with the endpoints and studied the structure and content of the responses with a popular API testing tool, Postman. Postman was especially helpful in debugging, testing of the headers and payload as well as automation of workflow on frequently utilized endpoints. This realization was a transition between my frontend development experience into my backend system design.

9.2 Database Concepts and Indexing

Having learned the basics about web technology, I proceeded to databases, which is one of the fundamental elements of web development. The training started with an experience of the relational database, namely MySQL [11] that is used extensively in the industry. This section of the training made me realize how data can be stored, retrieved as well as organized in an effective manner.

My investigation began with database tables and I examined such tables and found that they are composed of rows and columns of the form, entities, and attributes. I got to know how to type SQL commands to manipulate databases in terms of SELECT, INSERT, UPDATE, and DELETE. I also came to understand how to use filtering using WHERE clauses and get related data using JOIN.

Moreover, I read about such constraints as primary and foreign keys and unique conditions to ensure data integrity. I also dwelled into normalization methods to eliminate redundancy and enhance structuring of data. Indexing was an important subject. I came to understand the use of indexes to accelerate the query, which makes the database capable of accessing data quickly. I was also aware of the tradeoffs, that indexes make reads faster but could cause writes to work more slowly because of extra overhead. This made me ready to design backend systems where data accessed patterns should be efficient.

9.3 Mastering Java Syntax and OOP Principles

Having a good understanding of databases, I proceeded to study the Java [10] programming language. I started by learning simple syntax and then learned object-oriented-programming (OOP) which is the crucial building block of large and modular software systems.

I began by learning the basics of Java i.e. variables, data types, operators, control flow structures and arrays. After that I proceeded to define methods, sending parameters and returning values. Another major topic was exception handling that enabled me to create strong programs.

Most noteworthy, I got to know the main principles of the object-oriented programming: abstraction, encapsulation, inheritance, and polymorphism. Such concepts allowed me to develop code that is modular, reusable, maintainable and well structured. I have also done some Java collections framework where I got to learn the usage of Lists, Sets and Maps to effectively manage data. The ease of mastering Java prepared me to learn such enterprise frameworks as Spring Boot.

9.4 Learning the Spring Boot Framework

After familiarizing myself with the Java language, I went further to learn Spring Boot [12], an efficient framework written in the Java language, and it is deployed to develop enterprise-level backend applications. It was one of the milestones in my internship, because Spring Boot is a popular production-grade system.

To start with, I launched my own new Spring boot projects with Spring Initializr and checked a default folder structure. I also got to know how to utilize Model-View-Controller (MVC) pattern to organize web applications and utilized annotations

such as `@RestController`, `@GetMapping`, and `@PostMapping` to specify the API endpoints.

I got the idea of dependency injection and how the `@Autowired` annotation eases the creation of objects and their management. This opened me up to use of Inversion of Control (IoC) container, which is principle and central to Spring, that governs lifecycle and dependencies of components of an application. IoC discourages tight coupling and serves as a scalable concept because the framework dictates how objects are instantiated and dependency is resolved.

I split concerns as well by introducing service and repository layers and making the code more organized. Spring Data JPA helped me to know the basics on mapping of Java classes to schema tables in the database and the handling of relationship e.g. `@OneToMany` and `@ManyToOne`. I used form validation about `@NotNull` and `@Email`.

I applied Maven, which is a popular build automation tool, in order to minimize the dependency of the project and automate the build process. I came to know how to set up `pom.xml` file to handle dependencies, define the plugins and set up the project metadata. Maven made it a lot easier to configure the libraries used in a project and even manage them significantly when dealing with complicated Spring Boot applications.

Another crucial field of attention was security. I applied the JWT [4] (Spring Security) based authentication that allowed me to obtain user authentication/authorisation on endpoints. To secure the communication between client and server I set up the protection to resources: filters and security contexts and access roles.

I also got to know Thymeleaf [14], which is a server-side Java template engine that is applied to rendering HTML views by using Spring applications. Thymeleaf can be used with Spring MVC seamlessly and can encourage development of dynamic websites, with template variables, conditionals, loops. In Thymeleaf, I at least understood how full-stack applications are configured in Spring though it had been not the main part of my concentration of that time.

Postman was used by me to a great extent in order to test these secure REST APIs. It assisted me in transmitting authorization headers, request types testing, simulating errors, and checking the status code and responses. A crucial role was played by Postman which was necessary to make sure that endpoints acted as they should before connecting them to a frontend.

These routines helped me to learn how to create secure and scalable backend systems and maintain them.

9.5 Version Control with Git and GitHub

Once my projects became a little more complex, it became vital that I learn the version control. I enabled change tracking using Git [8], and remote collaboration with GitHub. I also got a thorough understanding of getting the basic of Git, how to initialize repositories, use staging, commit edits, see commit history. I researched how new features branches could be created, merged to the main branch and conflicts which may occur in the process of collaboration overcome. I would host repositories, pull and push changes and collaborate with others using GitHub through pull requests. I also got to understand how to set `.gitignore` files so that they could exclude unwanted files in the version control. I was able to work on the history of the code,

rollback in case of necessity, and become a part of team projects with the help of these skills.

9.6 Introduction to Docker

I studied Docker [6], which is a containerization platform that has been employed in DevOps and in backend deployment to make my applications portable and consistent. My journey started by getting to know the difference between virtual machines and containers. Docker containers are very fast to start and simple to share as they are light-weight and have on them just the application and its dependencies. I understood important Docker CLI commands to work with containers, such as `docker build`, `docker run`, `docker ps`, and `docker exec`. I have containerized my Spring Boot applications by writing Dockerfiles and defining and executing multi-container environments, such as a Spring Boot application linked to a MySQL database with Docker Compose. I also went into data persistence through Docker volumes and to exchange data amongst services with networks. The study of Docker enabled me to create mock production settings on the local machine and gave me the confidence that my applications could be operated reliably everywhere.

This chapter represents my complete backend development journey and the toolset I acquired during the internship, laying the foundation for the two final projects I worked on.

Chapter 10

Projects

10.1 Project 1-Presencia: Human Resource Management and attendance System

Presencia is a fully-featured multi-tenant Human Resource Management (HRM) software, which's backend is made by the spring boot framework by me. It is planned to be used by a number of companies and each of them would have a secure isolated environment where they can manage their employees, attendance as well as do administrative HR tasks. This system has a role-based access control (RBAC) model, whereby a user only access relevancy of the position through the functionalities and information in the system.

Presencias main objectives are to automate and simplify the most important HR tasks, such as hiring of the employee, ensuring his attendance using IP-based policies, managing his leaves by adjustable regimes, and performing the polls in the company. The system has secure, stateless, RESTful API that is expected to be consumed by a frontend application (e.g., a Next.js web app) and authentication and authorization is expected to be implemented with JSON Web Tokens (JWT).

10.1.1 Technical Architecture & Design

The backend of Presencia follows a modular, layered architecture built using the modern Java ecosystem. This approach ensures clear separation of concerns, improves maintainability, and supports future scalability.

Technology Stack:

- **Framework:** Spring Boot 3
- **Language:** Java 17
- **Security:** Spring Security 6 – used for JWT-based authentication and method-level authorization
- **Data Persistence:** Spring Data JPA with Hibernate as the ORM provider
- **Database:** MySQL
- **Build Tool:** Maven

- **Containerization:** Docker and Docker Compose – used to create reproducible development and deployment environments

Presencia backend is comprised of a modern Java stack, and takes the form of a layered architecture model such that separation of concerns, maintainability and scalability can easily be achieved. It makes use of a technology stack that comprises the Java 17 programming language used to implement the security, or Spring Security 6, a core Spring framework known as Spring Boot 3 and data persistence, or Spring Data JPA together with Hibernate as database, as well as a MySQL database. It is controlled and constructed with Maven and is completely containerized with Docker and Docker Compose to enable reproducible conditions of development and deployment.

The architectural layout is already focused on layered model, having a distinction of Controller, Service and Repository layers. In this structure, there is the promotion of clean code and modularity. The system presents its capabilities via stateless RESTful API with the standard HTTP protocol and JSON data transfer. Security stands first, and the model is inclined toward stateless JWTs-based cryptography to authenticate and method-level authorization, which is supported by @PreAuthorize annotation-based Spring Security roles, to enforce RBAC.

10.1.2 Core Features

The Presencia system is composed of several interrelated modules, each designed to address core aspects of human resource management in a scalable and secure manner.

The application is built on a multi-tenant architecture managed through a comprehensive User and Company Management module. It defines three major roles:

- **SUPER_ADMIN** – Responsible for onboarding multiple companies (tenants) and assigning respective company admins.
- **ADMIN** – Company-specific administrator responsible for managing all internal HR operations.
- **USER** – Regular employee with access to their own HR functionalities.

One of the key architectural principles is strict company data isolation. All resources and operations are scoped using a unique `company_id`, ensuring strong data privacy between tenants.

Employee Management Module: This module empowers company admins with full CRUD (Create, Read, Update, Deactivate) control over employee profiles. A distinctive feature is its flexible manager-assignment system, allowing any employee to be assigned as a manager without relying on rigid flags such as `isManager`. The system enforces logical validations to prevent anomalies such as self-assigned managers or circular relationships.

Attendance System: Employees can check in and check out once per day. Their timestamps are automatically evaluated against company-defined schedules (e.g., start/end times, grace periods, and workday rules) to determine statuses such as

PRESENT, LATE_CHECK_IN, or EARLY_CHECK_OUT. When irregularities occur, employees are prompted to provide reasons. Admins can whitelist trusted IP addresses to restrict valid check-in/out requests to office locations. The backend validates the source IP address—even behind proxies—by inspecting headers like **X-Forwarded-For**.

Leave Management System: This highly configurable system supports multiple leave types (e.g., Annual, Sick) defined by company admins, each with custom yearly allowances and activation toggles. Admins can override these defaults for individual users and specific years. The approval workflow routes leave requests to the assigned manager or escalates to the admin when no manager exists. Both managers and employees can monitor request statuses.

Holiday Management: Company admins can define single or multi-day holidays that automatically block leave applications on those dates.

Poll System: A flexible polling module enables admins to create custom polls for various purposes, such as event participation or lunch preferences. Polls can have configurable options, response limits, default choices, absentee options, and active time windows. Results are compiled into detailed and summary reports available to admins.

Account and Security Module: Presencia offers secure, role-based authentication with password reset features. Users can change their password via verification code, and initiate a secure **Forgot Password** flow, which sends a time-limited OTP to the registered email. All email communication is handled via integrated third-party SMTP services.

10.1.3 API Endpoint Summary

The system’s functionalities are exposed through a comprehensive set of RESTful API endpoints. The following table outlines the key API endpoints used in the Presencia backend.

Feature Area	Method	URL	Role(s)	Description
Authentication	POST	/login, /api/admin/login	Public	Authenticates user and returns JWT.
Password Mgmt	POST	/api/password/request-otp	Public	Sends OTP for password reset.
	POST	/api/password/reset-with-otp	Public	Resets password using OTP.
	PUT	/api/profile/change-password	USER, ADMIN	Change password for logged-in user.
Employee Self-Service	GET	/api/employee/directory	USER	Phonebook of company employees.
	POST	/api/employee/attendance/check-in	USER	Check-in employee via office IP.
	POST	/api/employee/attendance/check-out	USER	Check-out employee via office IP.

Feature Area	Method	URL	Role(s)	Description
	GET	/api/employee/leave/active-leave-types	USER	List of leave types available.
	POST	/api/employee/leave/requests	USER	Submit a leave request.
	GET	/api/employee/leave/requests	USER	View own leave request history.
	GET	/api/employee/leave/balances	USER	View leave balances.
	PUT	/api/employee/polls/id/my-response	USER	Vote on polls.
Manager Actions	GET	/api/manager/leave/requests/pending	USER (as Manager)	View pending requests of direct reports.
	PUT	/api/manager/leave/requests/id/approve	USER (as Manager)	Approve direct report leave.
	PUT	/api/manager/leave/requests/id/reject	USER (as Manager)	Reject direct report leave.
Admin Mgmt	POST	/api/admin/create/employee	ADMIN	Create a new employee.
	GET	/api/admin/employees/potential-managers	ADMIN	List eligible managers.
	PUT	/api/admin/holidays/id	ADMIN	Update a holiday.
	GET	/api/admin/leave/employee-balances-overview	ADMIN	Company-wide leave overview.
	PUT	/api/admin/leave/requests/id/admin-approve	ADMIN	Admin-level leave approval.
	POST	/api/admin/polls	ADMIN	Create polls.
	GET	/api/admin/polls/id/results	ADMIN	View poll results.
	PUT	/api/admin/attendance/company-ip-settings	ADMIN	Set allowed office IPs.
Super Admin Mgmt	GET	/api/admin/dashboard/summary	ADMIN	Dashboard stats.
	POST	/api/superadmin/companies-with-admin	SUPER_ADMIN	Create new company and admin.
	GET	/api/superadmin/companies	SUPER_ADMIN	List all companies.

10.1.4 Database Schema Overview

It has a data model of the system that is constructed by several important entities that have established relationships to ensure that the data integrity is achieved and the specified business logic is represented.

Users: Stores a login (email, password), role and status (enabled). Hyperlinks of the company that an Admin is dealing with.

Company: Stores company information and acts as a key indicator of tenants. Stores company-wide defaults such as regulated attendance time, days when attendance is flexible and permissible IP addresses.

HR data: The pivot point of the HR data of the employee. Has personal information (like name, mobile, department, designation, bloodGroup, joiningDate), a unique business employeeId as well as links to Users (used during login), Company (used during tenancy) and employee (used when creating the manager-subordinate relationship).

AttendanceRecord: This is used to store daily check-in and check-out time stamps, final calculated status and causes of late or early actions of an employee.

LeaveTypeSetting: It is an entity that is configurable by an admin and specifies a type of leave (e.g., Annual), its default allowance per a year, and whether the specified type is active in a particular company.

EmployeeLeaveAllowance: An override record enabling an admin to specify a custom leave allowance day of a particular employee, of a given leave type, on a given year, over-riding the company default.

LeaveRequest: One of the leave requests with one employee. It connects with the one who made the requestEmployee, his or her manager (if any one), desired setting of the LeaveType, and follows its status (PENDING, APPROVED, REJECTED), dates, and comments.

Holiday: Admin can create holidays for a company

Poll, PollOption, EmployeePollResponse: A collection of entities used to handle flexible polls. The description and the rules of poll are defined by Poll. PollOption characterizes the choices that were made especially by the admin related to that poll. EmployeePollResponse keeps track of what PollOption a particular employee voted on.

PasswordResetToken: stores one time secure OTP codes used to enable individuals with a password forget feature tied to a user and an expiry date.

The associations between them (e.g. Cascade family of annotations, many-to-one, one-to-one etc.) are also applied to the classes using JPA annotations to ensure referential integrity via database.

10.1.5 Application User Interface Showcase

This section will present some of the most important screenshots in the web-based Admin Panel as well as on the employee-facing mobile application so that a representation of the Presencia system can be observed. These pictures prove the practical application of main characteristics worked out in the course of internship.

Admin Panel Interface

The online Admin Panel gives the HR overlords of the whole organization all they need to control the HR activities of the organization.

The dashboard: the primary dashboard provides a quick overview of company workforce and attendance per day. It reflects such important data as the number of employees in total and by gender, the number of newly hired people. The attendance pie chart presents a quick overview of the visual representation of occurred

attendance on a specific day between on-time and late check-ins.

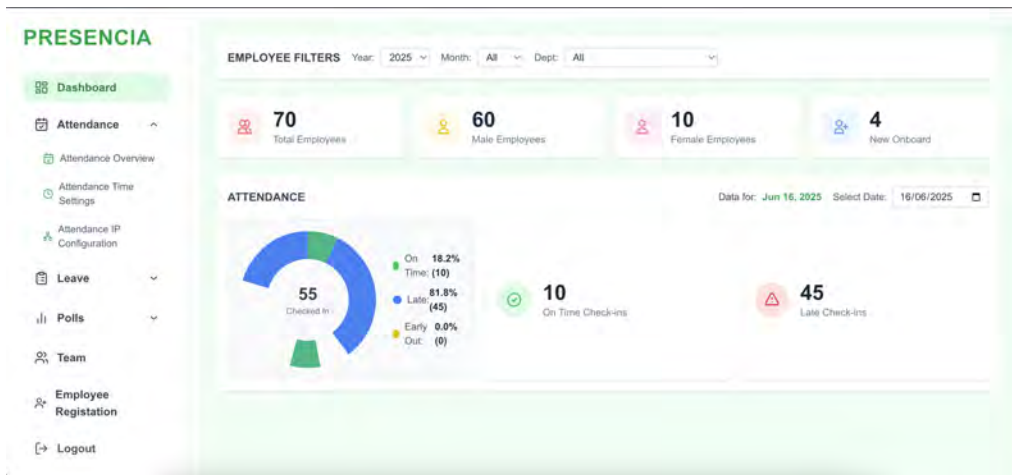


Figure 10.1: Admin Dashboard providing a summary of workforce and daily attendance statistics.

Team Management: Admins will be able to see a list of all employees of their company in detail. Under this view of employees, which is called Employee List, the necessary details about the employee like name, contact, designation, department and the assigned manager is given, where the profile details can be edited of each employee.

The Employee List view displays a table of employees with columns: Avatar, Name, Mobile, Email, Designation, Department, Joining Date, Employee ID, Manager, and Actions. The table contains 6 rows of employee data. The Actions column includes icons for edit, delete, and a green 'Add' button.

Avatar	Name	Mobile	Email	Designation	Department	Joining Date	Employee ID	Manager	Actions
T	Test Employee 1	01521581955	test1@gmail.com	software Engineer I	ENGINEERING	2025-05-10	50	Zakaria Raihan 5	
T	Test Employee 4	01615888102	test2@gmail.com	page	ENGINEERING	2025-05-18	2	Test Employee 1	
T	Test Employee 2	01521581955	test3@gmail.com	sold	DATA & INSIGHT	2025-06-05	3	Zakaria Raihan 5	
Z	Zakaria Raihan 5	01521581954	zakaria@gmail.com	123456	DATA & INSIGHT	2025-06-05	4	Test Employee 1	
R	Raihan	01521581954	raihan@gmail.com	das	FINANCE	2025-06-20	1	Test Employee 1	
R	Rasel	132133	rasel@gmail.com	addias	TECHOPS	2025-06-07	6	Raihan	

Figure 10.2: The Employee List view for team management.

Attendance Settings: The attendance rules can be set with a great number of details, such as regular times of check-ins/outs with grace periods is it late or early, workdays as weekends, or other flexible.

Attendance IP Configuration: In order to provide security the administrators are able to configure a list of IP addresses that are authorized to be used by an employee during the check-in and out of an employee that will maintain activity only within the authorized office environment.

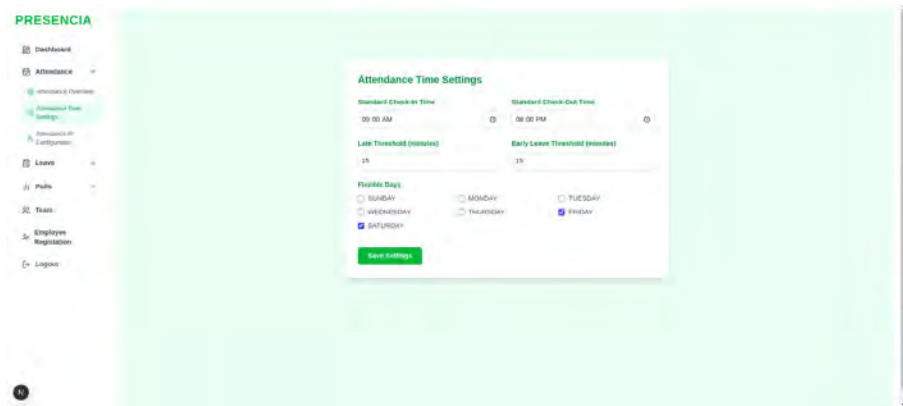


Figure 10.3: Configuration screen for company-wide Attendance Time Settings.

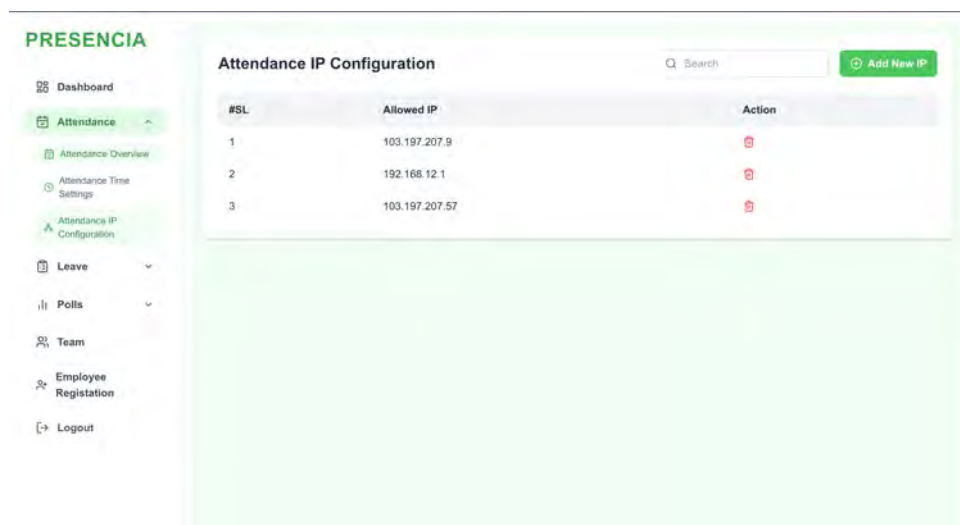


Figure 10.4: Admin interface for managing allowed IP addresses for check-in/out.

Leave Settings: Admins are able to create and maintain different types of leaves of their company (e.g., Annual, Casual, Sick). In regard to each type, they could establish the amount of days permitted per annum and change its active state.

Holiday Settings: At this screen, the admins may configure and design the annual holiday calendar of the company, which will be automatically reflected in leave and attendance modules.

Leave Request Overview: Admin can see the overview of all leave requests at his/her company, the possibility to filter them by date, employee and status. Admin control screen will enable admins to act on the pending requests and act as the hub of centralized management.

Employee Leave Balance Overview: The report gives a detailed grid view of the leave balances of all the employees. Admins can easily access how many days have been utilized and the amount of days left by each employee of each type of leave in the company.

Poll Creation: Admin user can create dynamic polls to the employees, e.g. how many in the office are having lunch on a daily basis. The interface can be used to

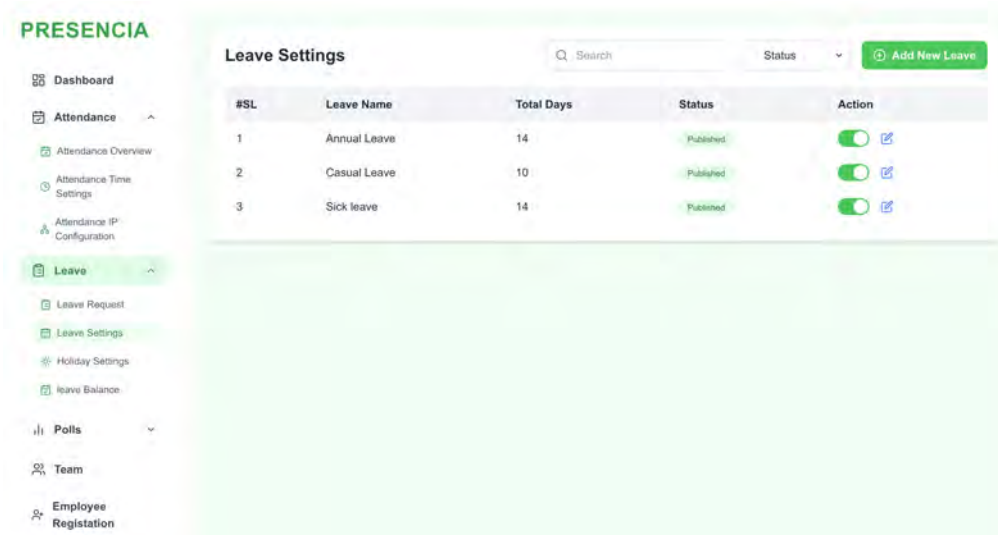


Figure 10.5: Interface for creating and managing different leave types and their yearly allowances.

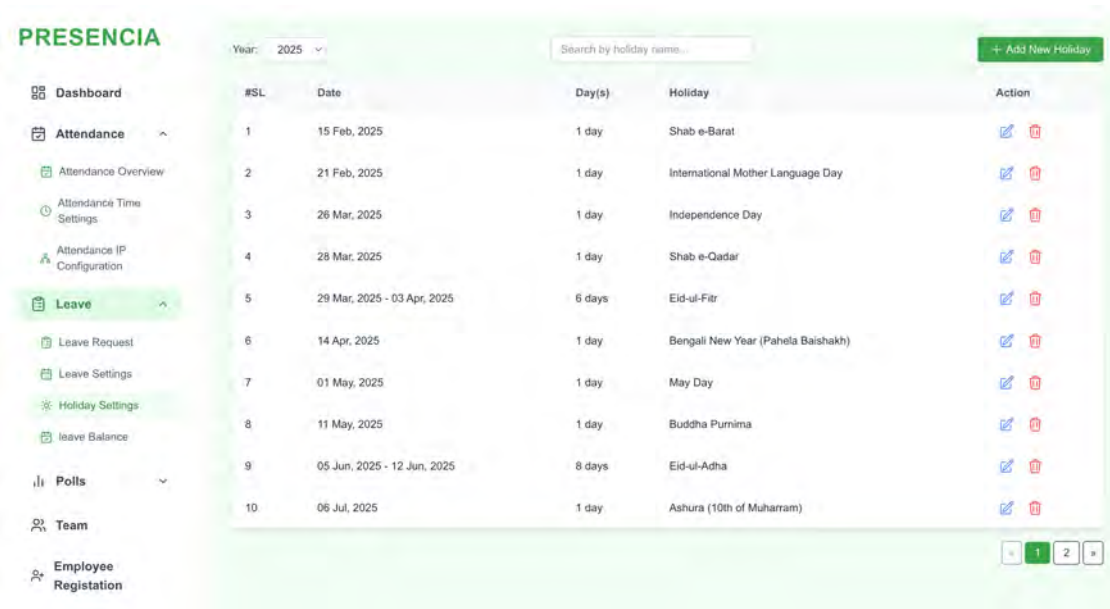


Figure 10.6: The yearly Holiday Settings calendar for the company.

specify the description of the poll, a default value to non voters, a particular voting window and options with respective values.

Employee Mobile Application Interface

The mobile app gives the employees self-service facilities and can control their attendance and leave, can also communicate with the company activities via their mobile devices.

Dashboard and Check-in/Check-out: The home page greets the worker and shows them whether they are currently checked in or out, making it very clear that they need to either check-in or out. The attendance screen is very simple to use; there is a big clear button.

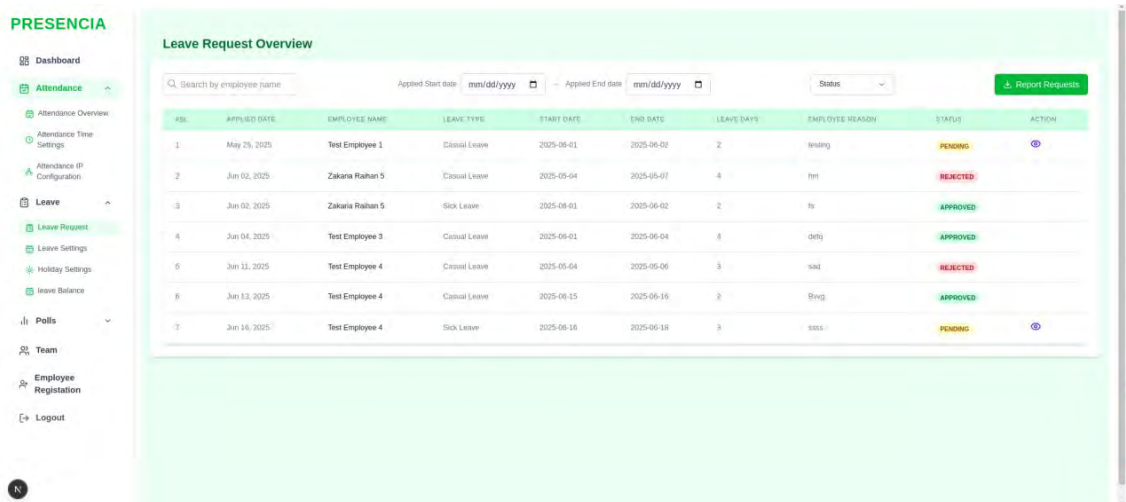


Figure 10.7: Admin's consolidated Leave Request Overview with filtering options.

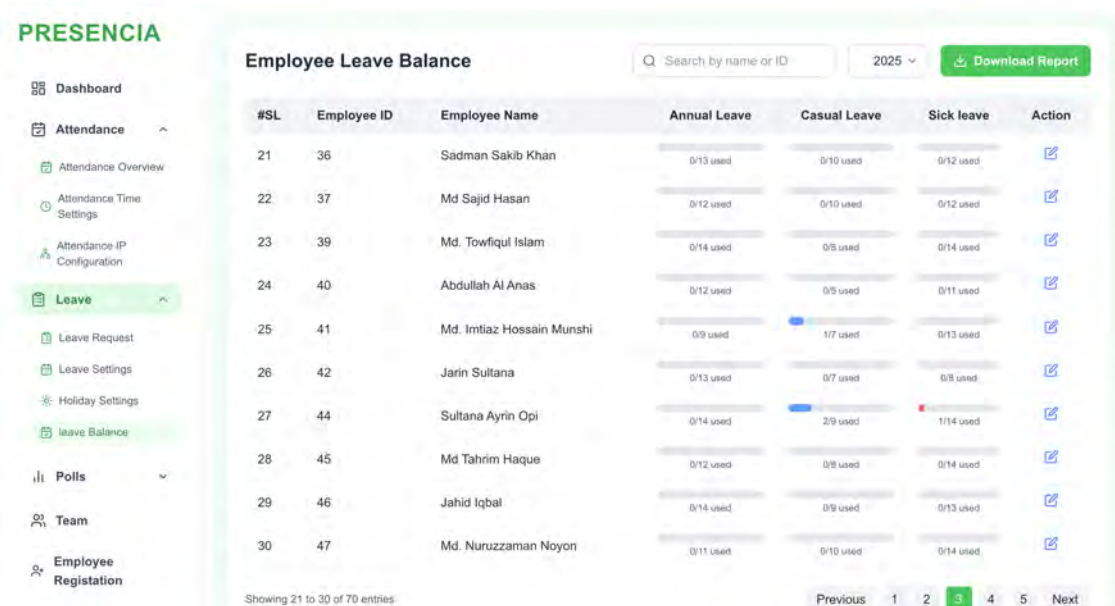


Figure 10.8: Admin view of the leave balance report for all employees.

PRESENCIA

Lunch Poll Admin

Create New Poll

Poll Date: Default Value:

Voting Start Time: Voting End Time:

Poll Description:

Please select if you are not coming or forgoing any guest?

Poll Options:

- Cancel my vote:
- 1 guest:
- 2 guest:
- 3 guest:

[Create Poll](#)

Figure 10.9: The poll creation interface for admins.

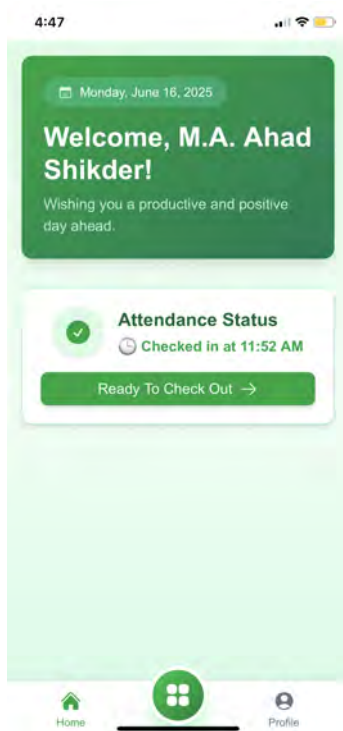


Figure 10.10: Employee mobile dashboard showing attendance status.

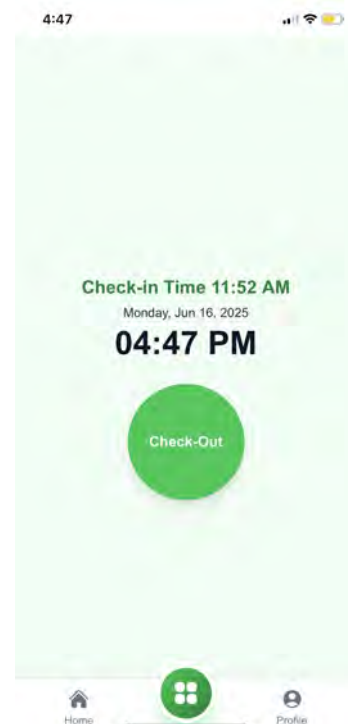


Figure 10.11: Employee check-out screen.

Leave Request Submission: This screen gives the employees a full-fledged option of requesting his/her leave. It shows the balances on each type of leave (Annual, Casual, Sick) available and they show any days taken or still available. Interactive calendar sets a date range of choice available to a given employee, and a real-time counter identifies the number of days chosen.

Manager Approval View: The employees of the company that are assigned with managerial job echelons are provided with an integrated perspective on the dashboard to view the pending leave request of subordinates. They have the possibility to approve or disapprove the requests directly through this interface.

Leave Request History: This screen enables the employees to know the status of all his/her submitted leave requests whether they are in PENDING, APPROVED or REJECTED.

Active Polls: Active polls also appear on the employee dashboard giving the options that can be voted upon and making the employee vote accordingly.

Attendance History: This will also help in showing the employees their attendance history on a particular month as each date is labelled properly as present, late check in, absent, etc. giving a clear picture of the attendance.

Employee Directory: A feature called a phone book allows the employee to quickly access other contact information about their co-workers in the company, and search easily either by name or filter by department.

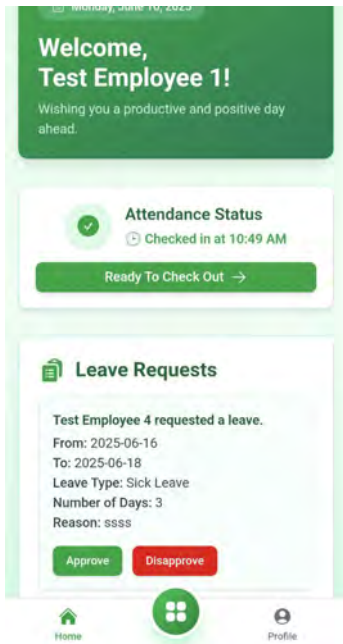


Figure 10.12: Manager's view of a subordinate's pending leave request.

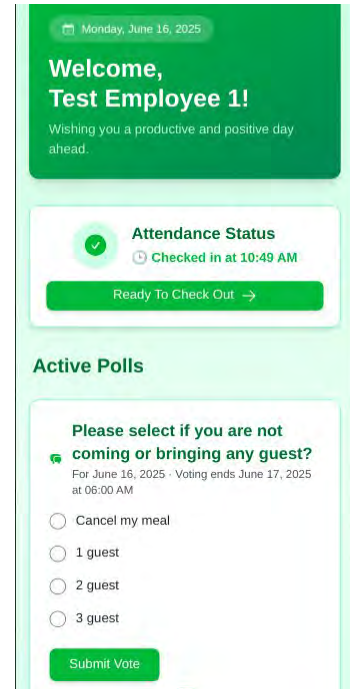


Figure 10.13: Employee view of an active poll.

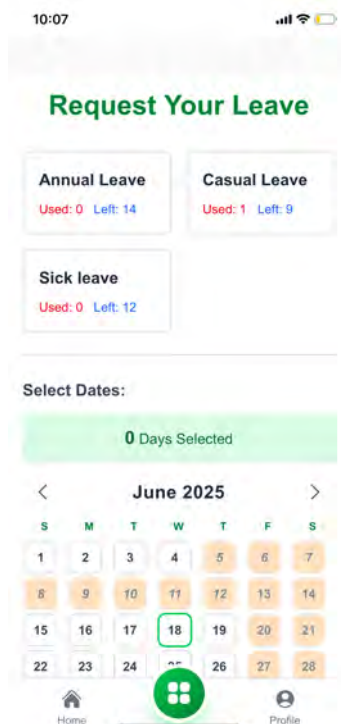


Figure 10.14: The "Request Your Leave" screen showing balances and a calendar.

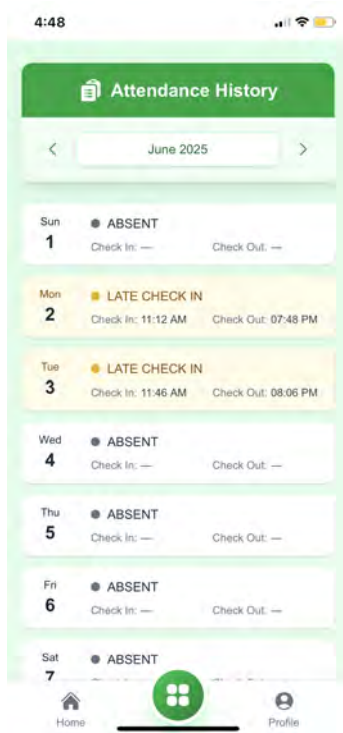


Figure 10.15: Employee's monthly Attendance History view.

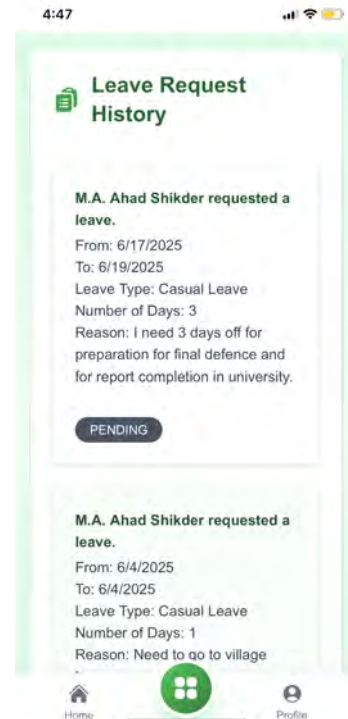


Figure 10.16: Employee's Leave Request History.

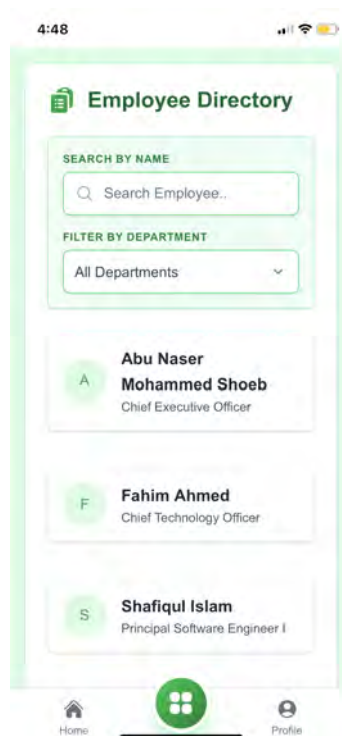


Figure 10.17: The company "phonebook" or Employee Directory.

10.2 project 2 - ErrorLog Tracker

10.2.1 Project Overview

ErrorLog Tracker is a complete web application that was written as a lightweight in-house alternative to commercial error monitoring services such as Sentry [7]. The

main aim of this project is to give a nodal point where application exceptions can be automatically detected, viewed, and analyzed in real time by developers and teams. The system provides a solution of resorting to dedicated space of managing errors and simplifies debugging process being applied to the systems development across numerous software projects.

The software permits authorized users to enroll into their software projects. Every single project will have an identifier, and error logs will be sent through a special REST API endpoint. The system is configured to consume detailed exception information, such as stack traces, HTTP context and other applicable metadata, giving the developers the information they require to address and solve the issue at a relatively fast speed.

10.2.2 Technical Architecture & Design

The ErrorLog Tracker is constructed on a strong signature Java stack, whereby layered architecture is taken into consideration so as to provide a smooth separation of concerns and manageable code.

Technology Stack:

- **Framework:** Spring Boot 3
- **Language:** Java 17
- **Security:** Spring Security 6 – used for JWT-based authentication
- **Data Persistence:** Spring Data JPA with Hibernate as the ORM provider
- **Database:** MySQL
- **Build Tool:** Maven
- **Frontend Templating:** Thymeleaf (Server-Side Rendering)
- **API Testing:** Postman
- **Containerization:** Docker and Docker Compose

10.2.3 Architectural Design

Layered Architecture: The backend would be logically divided in terms of business logic, API requests and data access into a Controller layer, Service layer, and Repository layer respectively. The management of the component lifecycle and component dependencies are achieved using Spring IoC container and Dependency Injection.

Monolithic Full-Stack: Compared to a decoupled frontend, it has Thymeleaf rendering the user interface on the server side to provide a tightly integrated full-stack application in which the backend directly generates and serves the HTML views.

RESTful Ingestion API: The important error submission feature is presented as a RESTful API endpoint with stateless access, so that any outside application (as long as it has the correct credentials) can POST error data.

10.2.4 Core feature

The dashboard is clean and functional, and that is the focus of the user experience of the system. Upon safe login, a developer then gets a summary of all the Projects (Projects) that he/she has registered like Laravel Project or test and has an option to create another project, delete a project. A feed at the bottom of the projects list shows the most recent Error Logs for all their projects in real-time, allowing them to have a quick look at application health.

Its main functionality is error ingestion and analysis capabilities. A specially created REST API endpoint (POST /Errors/add/projectId) allows receiving error data sent by other applications. The API expects a rich JSON payload with lots of context, such as the programming language, the framework version, the class of the thrown exception, the error message, a complete and structured stack trace.

Clicking one of the errors in the dashboard redirects a user to Error Details page. It is significantly based on professional debugging tools such as Sentry, and, therefore, provides detailed Clarity. The left side contains a stack trace that enables the call stack to be navigated easily. The primary exception message is shown in the main view, and a code sample has been shown in the source file and the line containing the exact location where the exception was thrown indicated. To give even more details, it is also important to view metadata critical at the time of the error, placing them on the page, including the full copy of the HTTP Headers, any Database Queries in play, with their execution time and data binding. Such contextual perspective is insurmountably valuable in giving the developers an idea of the application state at the point of error, and to consequently fix the problem fast.

Every page that is accessible to the user as well as every API endpoint concerned with data is secured, and only the project where the user has access to can be viewed and managed by a user. The whole process, starting with the submission of an error through Postman when testing an application, or the possibility of examination of detailed traces on the front-end generated by Thymeleaf, generates an entire and practical error-tracking ecosystem.

10.2.5 API Endpoint Summary

Most application parts are server-side rendered with Thymeleaf, but it has a dead-central architecture regarding the most important RESTful endpoints.

Feature Area	Method	URL	Role(s)	Description
User Management	POST	/register	Public	Registers a new user account.
	POST	/login	Public	Authenticates a user and starts a session.
Project Management	GET	/projects	Authenticated	Lists projects for the logged-in user.
	POST	/projects	Authenticated	Creates a new project.

Feature Area	Method	URL	Role(s)	Description
	DELETE	/projects/{id}	Authenticated	Deletes a project and all its associated error logs.
Error Log Management	POST	/Errors/add/{projectId}	Authenticated	Submits a new error log for a project.
	GET	/Errors/allErrors	Authenticated	Lists all error logs (used by dashboard).
	DELETE	/Errors/{id}	Authenticated	Deletes a specific error log.

10.2.6 Application User Interface Showcase

The ErrorLog Tracker has a developer-friendly interface that allows tracking projects and tracing exceptions. The UI aims to deliver the complex information on errors into an unambiguous form, which can be integrated into the work processes.

Login & Dashboard: Upon a successful log in, one is taken to the dashboard. The page acts as a dashboard, where a list of the projects created by the user (e.g., Laravel Project, test) are listed and each of their Error Logs is displayed in real-time as they are collected. On this screen, the developer can create any new project or delete the old project easily and it has simple project management.

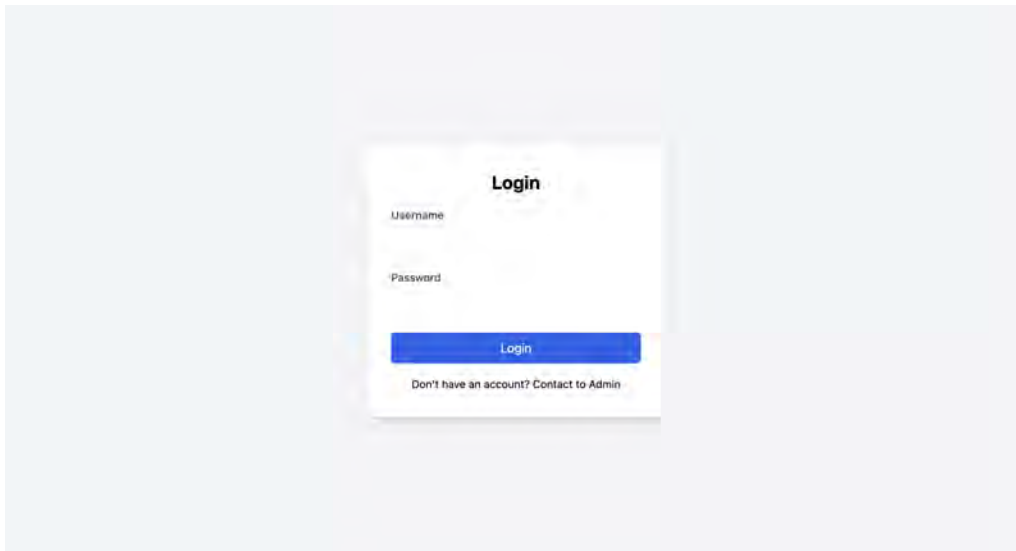


Figure 10.18: The user login screen for ErrorLog Tracker.

Error Details View: The apex of the application is a specified error view that is magnificently influenced by expert monitoring tools such as Sentry. When a developer opens an error in the dashboard, a multi-pane view is opened, which allows them to have an effective debugging experience. The left part gives a navigable stack trace which the developer can click through the steps of the call stack. On the main content area, the exception message (e.g. Test exception) shows prominently and specifies the particular class and method in which the error was generated. Code snippet viewer allows seeing the source code of the file that caused the exception and see exactly the line in which it failed with text marked with red.

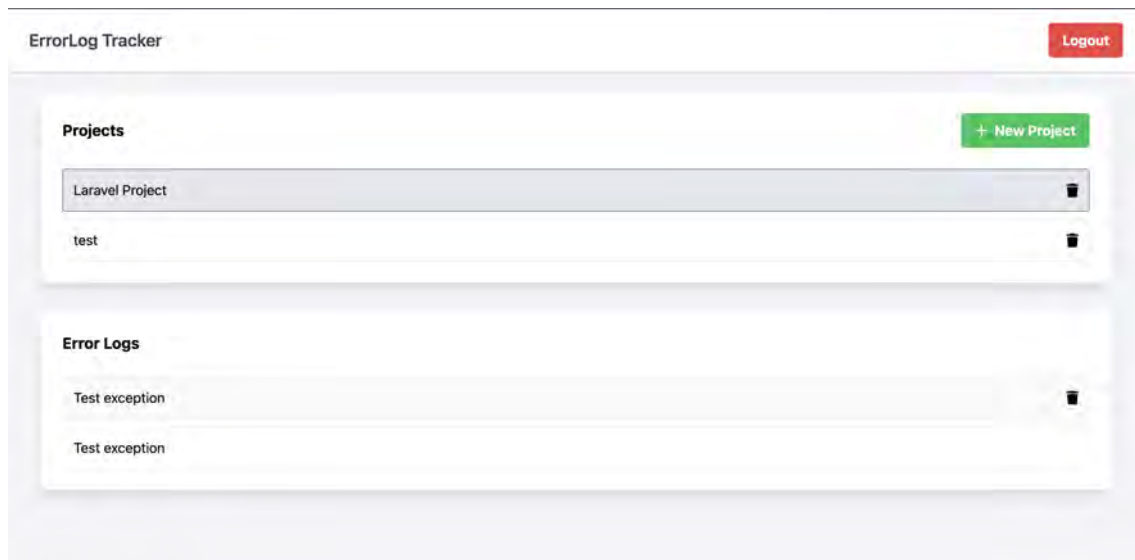


Figure 10.19: The main dashboard showing registered projects and a feed of recent error logs.

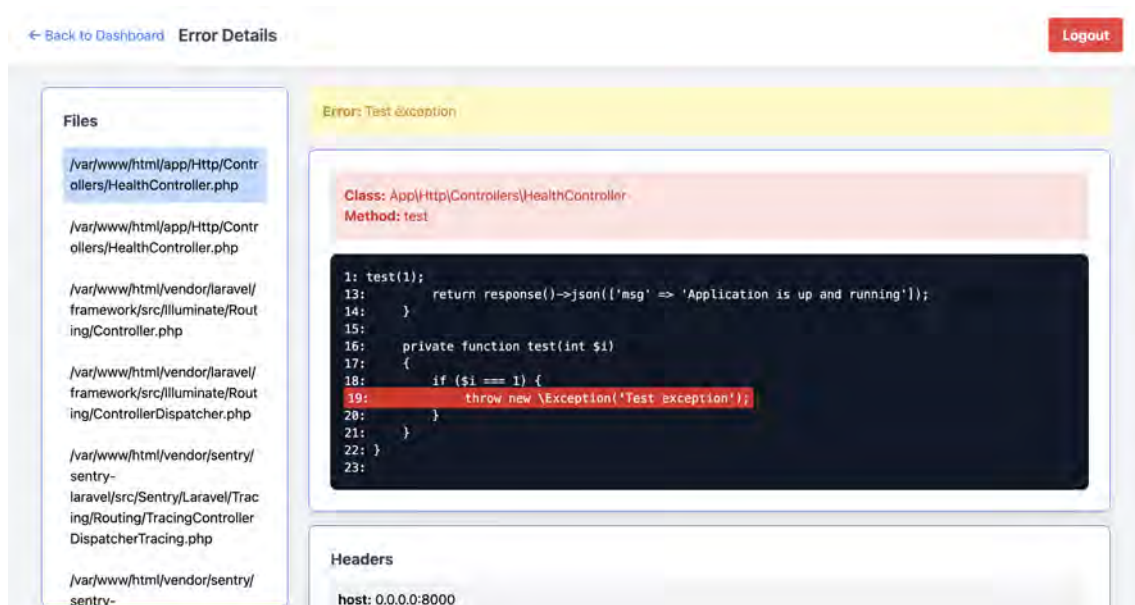


Figure 10.20: The detailed view of an error, showing the stack trace on the left and a code snippet highlighting the exact line of failure.

Contextual Metadata: In a further effort to support the details in debugging, Error Details page contains important metadata that was recorded at the moment of error. This includes:

The complete response of the HTTP Headers. The executed Database Queries, the time of execution, the data bindings. This context related information is of infinite value as far as ascertaining the status of the application at the time of failure is concerned.

API Ingestion via Postman: The REST API endpoint empowers the data collection of the system. The Postman screenshot illustrates how an external application would be able to submit POST request to `/Errors/add/` projectId with a detailed

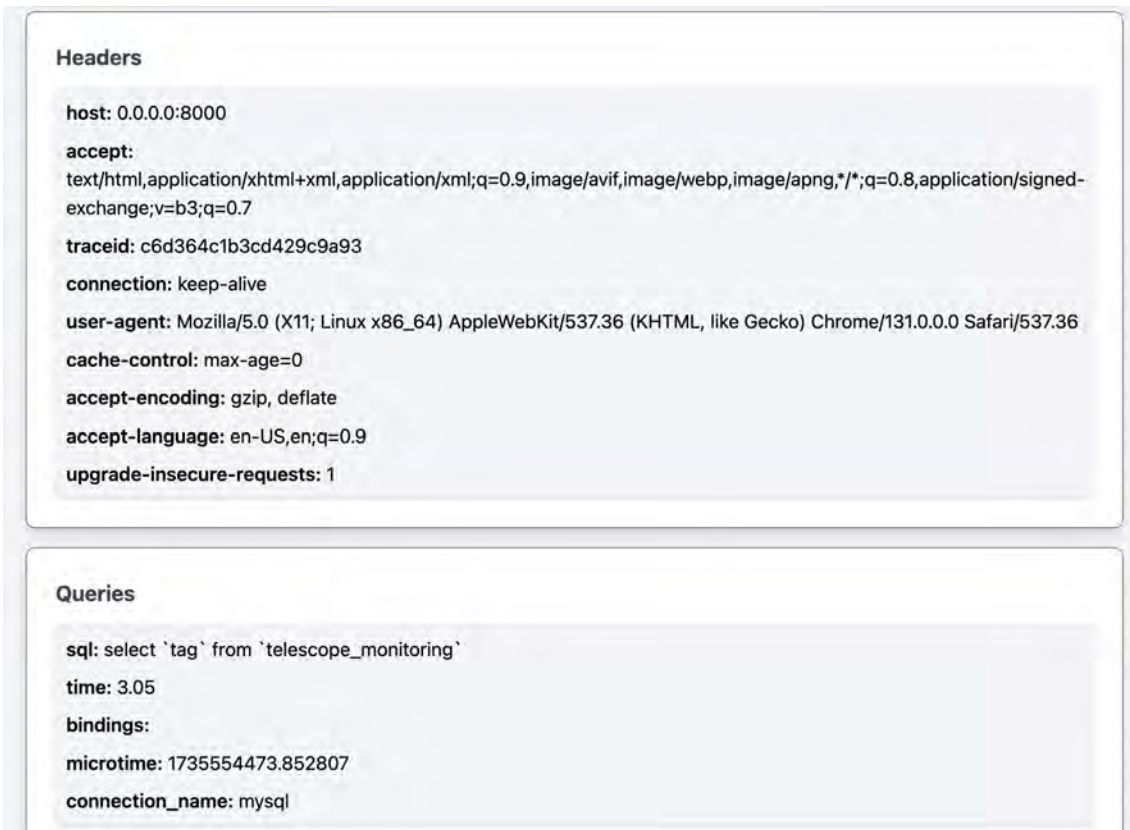


Figure 10.21: Another example of the error detail view, showing a different point in the call stack.

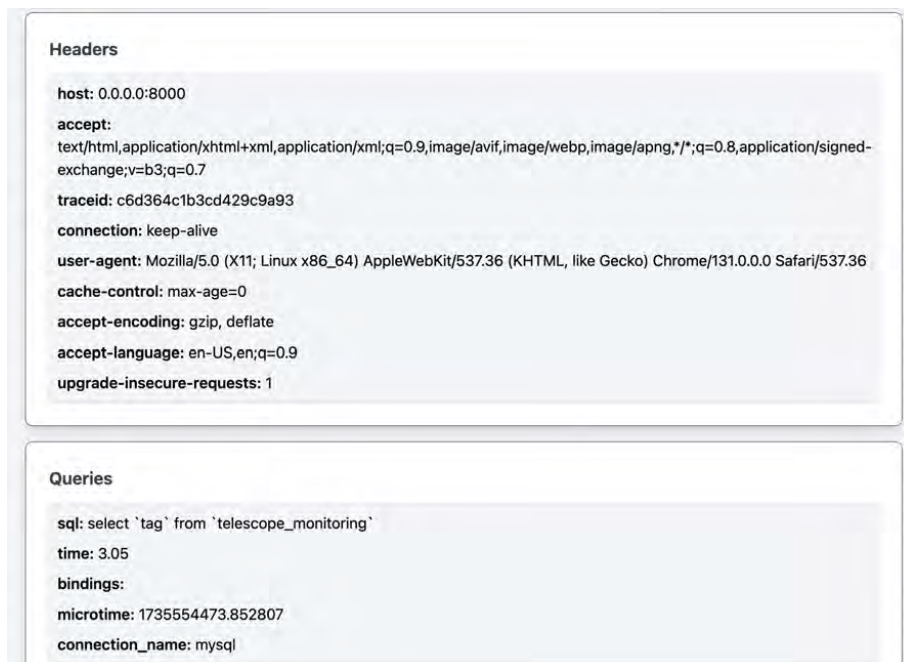


Figure 10.22: Contextual metadata provided with an error, including HTTP Headers and database queries.

JSON payload. This payload also includes the metadata that encapsulates the error it includes, language, version of the framework, exception class, message and a full

stack trace then is deciphered and listed by the ErrorLog Tracker UI.

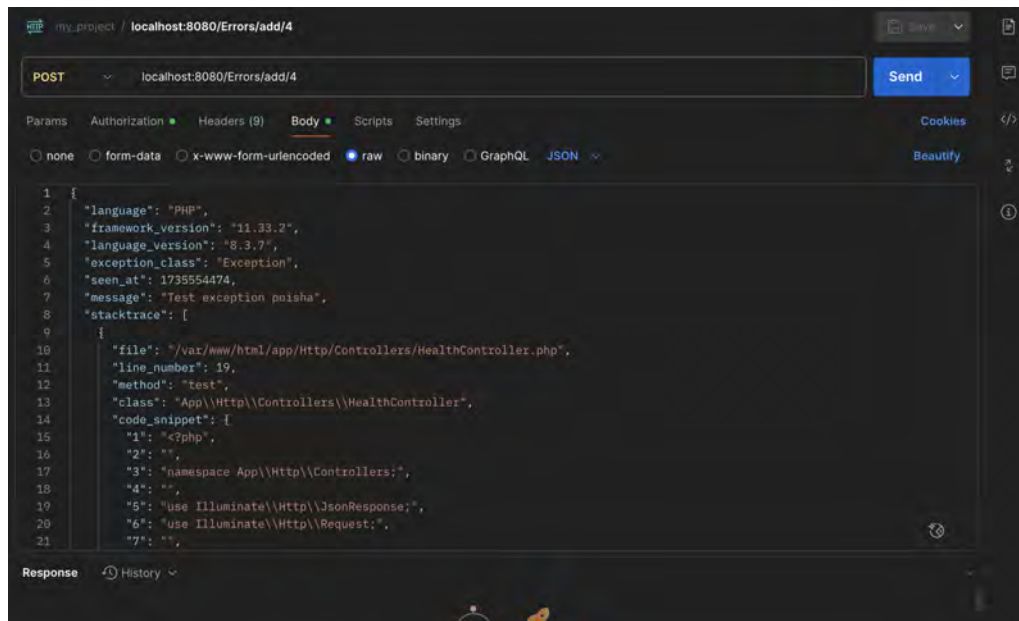


Figure 10.23: A Postman request demonstrating how an external application submits a detailed error log payload to the API.

Chapter 11

Conclusion

Polygon Technology has been a very enriching and life-changing experience as an intern software engineer. It provided me with a well-organized atmosphere not only to gain knowledge on new technology but also use it in resolving practical issues. The internship taught me the key concepts of backend development and system design, API architectures, and recent development tools, and practices.

The experience I have acquired at developing projects with my team, including the Mini Error Tracking System and the Human Resource Management platform People Resource Management Presencia, led me to get a practical understanding of how to design and develop scalable, secure, and modular backend systems in Java and Spring Boot framework. Through these projects, I encountered key ideas of development dependency injection, layered architecture, JWT-based authentication and docker-based containerization, which have become core in contemporary backend application development.

In addition to this, I used Postman a lot to test and debug APIs and make the communication between a frontend system and a backend system robust. Software such as Git and GitHub allowed me to familiarize myself with version control workflows, strategies such as branching, and collaborative development environment that are required in work environments where one is expected to work as a team.

Being an active member of a team strengthened my communication and coordination skills further as well. This involvement of frontend developers to assist in their understanding of the UI requirements and modifying APIs based on their requirements taught me the need to integrate and align the various functions without hassles.

This internship not only sharpened my technical expertise but also how to deal with complexity, solve problems analytically, and do things that were quality-driven and maintainable. I have also become much more confident regarding my ability not only to work on and contribute to real-world software but as well knowing how the process of software development lifecycle actually works in terms of planning and deployment.

To conclude, my future career has been well prepared using this internship. It confirmed my curiosity in backend, enhanced my problem-solving skills and provided me with a better sense of my future directions. I have become a better, more confident, and more eager to develop meaningful software products in the future years following this internship.

Bibliography

- [1] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.
- [2] I. Fette and A. Melnikov, *The websocket protocol*, RFC 6455, 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6455>.
- [3] ECMA International. “Ecmascript 2015 language specification.” Accessed: May 27, 2025. (2015), [Online]. Available: <https://262.ecma-international.org/6.0/>.
- [4] M. Jones, J. Bradley, and N. Sakimura, *Json web token (jwt)*, RFC 7519, 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7519>.
- [5] Bootstrap Team. “Bootstrap.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://getbootstrap.com/>.
- [6] Docker, Inc. “Docker overview.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://docs.docker.com/get-started/overview/>.
- [7] Functional Software, Inc. “Sentry.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://sentry.io/>.
- [8] Git SCM. “Git.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://git-scm.com/>.
- [9] Mozilla Developer Network. “Mdn web docs: Html and css.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://developer.mozilla.org/>.
- [10] Oracle Corporation. “Java platform, standard edition.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://www.oracle.com/java/>.
- [11] Oracle Corporation. “Mysql.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://www.mysql.com/>.
- [12] Pivotal Software, Inc. “Spring boot.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://spring.io/projects/spring-boot>.
- [13] Tailwind Labs Inc. “Tailwind css.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://tailwindcss.com/>.
- [14] Thymeleaf Project. “Thymeleaf.” Accessed: May 27, 2025. (2025), [Online]. Available: <https://www.thymeleaf.org/>.