

Internship as a Full Stack Developer

by

Mohammed Inzam Ul Azam
ID: 24241258

An internship 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
January 2025

© 2024. Brac University
All rights reserved.

Declaration

It is hereby declared that

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

Mohammed Inzam Ul Azam

ID: 24241258

Approval

The internship report titled “Internship at Full Stack Web Development : Software Engineer Trainee” submitted by

1. Mohammed Inzam Ul Azam (24241258)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 31, 2025.

Examining Committee:

Supervisor:
(Member)

Md. Shahriar Rahman

Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Abdurnur Rafi

Chief Executive Officer
Auracore Technologies Ltd.

Head of Department:
(Chair)

Sadia Hamid kazi

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

Abstract

This internship report highlights the experiences and lessons gained during a comprehensive internship program at Auracore Tech Limited, a startup specializing in digital transformation and software development. The internship focused on full-stack development, particularly front-end technologies such as React JS, HTML, CSS, JavaScript, and backend integration. The report provides an overview of the company's profile, objectives, and operations, emphasizing Auracore's commitment to innovative solutions in web and mobile application development, SaaS products, and digital marketing. The training involved practical exposure to live projects, including the enhancement of the Metro Meal Order Management Software. Key tasks included implementing carousel designs using Daisy UI and Tailwind CSS, leveraging Context API for data flow between components, and adhering to Agile development practices. Mentorship and collaboration with seasoned developers were integral to the program, fostering technical and professional growth. This report documents the technical, professional, and personal development achieved during the internship, underlining the significance of such programs in bridging the gap between academic learning and real-world applications. It concludes with a reflection on the invaluable experience gained and its role in preparing for future endeavors in the technology sector.

Keywords: Auracore Tech, full-stack development, React, HTML, CSS, JavaScript, backend, SaaS, Metro Meal, Daisy UI, Tailwind CSS, Context API, Agile, mentorship, technology.

Table of Contents

Declaration	i
Approval	ii
Abstract	ii
Acknowledgment	iv
Table of Contents	iv
List of Figures	1
1 Introduction	2
1.1 About internship	2
1.2 A strategic approach	2
1.3 Objectives	2
1.3.1 Primary Purpose	2
1.3.2 Specific Objectives	3
1.4 Methodology	3
1.4.1 Primary Data Sources	3
1.4.2 Secondary Data Sources	3
2 Company's Profile	4
2.1 Overview	4
2.2 Vision and Mission	4
2.3 Operational Framework	4
2.3.1 Stakeholders and Partners	4
2.3.2 Program Management	4
2.3.3 Product Ownership	4
2.3.4 Quality Ownership	5
2.3.5 Service Portfolio	5
2.3.6 User Experience Design	5
2.4 Software Development Methodology	5
2.4.1 Requirement Identification	5
2.4.2 Proof or Concept	5
2.4.3 Contextual Research	5
2.4.4 Iterative Releases	5
2.5 Recruitment Process	6

3	Internship Objectives at Auracore Tech	7
3.1	Technical Training	7
3.2	Collaborative Environment	7
3.3	Mentorship Support	7
3.4	Technical Skill Enhancement	7
3.5	Holistic Professional Development	8
3.6	Industry Standards Exposure	8
3.7	Professional Networking	8
4	Training Experience	9
4.1	Training Overview	9
4.2	Front-End Development Fundamentals	9
4.3	React JS Project Folder Structure	10
4.4	Initial Task: Carousel Implementation	11
4.5	Managing Data Flow with Context API	12
4.6	Number varification with OTP system	13
4.7	Implementing Order History Table UI	15
4.8	Implementing User Role Update UI	16
4.9	Backend Development Fundamentals	17
4.10	Connecting Order Data to the Database	17
4.11	Managing User Varification Request	18
4.12	Updating User Role From Backend	19
4.13	Learning Github	19
5	Challenges Faced During Internship	20
5.1	Frontend Development Challenges	20
5.2	Backend Development Challenges	20
5.3	User Identity Verification	21
5.4	Order Management System Challenges	21
5.5	Number Verification System	21
6	Internship Takeaways and Outcomes	22
6.1	Insights into Enterprise Software Development	22
6.2	Practical Experience with Modern Technologies	22
6.3	Developing Problem-Solving and Troubleshooting Skills	23
6.4	Developing Communication and Time Management Skills	23
6.5	Continuous Learning and Personal Development	23
7	Conclusion	24
	Bibliography	26

List of Figures

4.1	Activity Diagram of my Workflow.	9
4.2	Folder Structure in React JS.	10
4.3	Use of Tailwind CSS in React JS file	10
4.4	Carousel Design Implementation	11
4.5	Code for Carousel component	11
4.6	Passing data to other component	12
4.7	Using data from index.js file.	12
4.8	Number input UI	13
4.9	OTP input UI	13
4.10	code for implementing number varification UI	14
4.11	Order history table UI	15
4.12	Code for UI design	15
4.13	User role update UI	16
4.14	Code for User role update UI	16
4.15	Database for getting order history	17
4.16	Database for getting user varification request.	18
4.17	Fetching order verification request.	18
4.18	Code for getting user varification request.	19
6.1	My recent project.	23

Chapter 1

Introduction

1.1 About internship

An internship provides final-year students an opportunity to adapt to real-world workplace procedures and apply academic knowledge practically. This experience enables recent graduates and students to gain industry-specific exposure relevant to their course of study, facilitating a smooth transition from academic environments to professional labor markets. The benefits of internships are multifaceted. Organizations can better understand intern capabilities and strategically assign roles based on organizational needs. For students, internships significantly enhance work experience, enriching professional curricula vitae and expanding potential career opportunities. The popularity of internships continues to grow, with some academic institutions preferring internship experiences over traditional thesis papers. At BRAC University, Auracore Tech was designated as the internship host. Auracore Tech is an enterprise software development company focused on innovative technological solutions, specializing in web and mobile application development and comprehensive marketing strategies. As a local startup in Bangladesh, the organization aims to provide meaningful learning experiences for emerging professionals.

1.2 A strategic approach

This internship report serves as a formal documentation requirement for BRAC University's bachelor's program evaluation. The document comprehensively details the organizational atmosphere, workplace culture, professional experiences, and critical lessons derived from the internship engagement.

1.3 Objectives

1.3.1 Primary Purpose

The primary objective of this report is to present a comprehensive professional overview and document acquired knowledge and insights.

1.3.2 Specific Objectives

The report aims to:

- Provide an in-depth company profile
- Describe the workplace environment
- Outline Auracore Tech's primary operational focus
- Enumerate the comprehensive service portfolio
- Articulate key discoveries throughout the internship
- Deliver a detailed explanation of project-specific tasks

1.4 Methodology

The report primarily focuses on professional experiences and lessons obtained during the Auracore Tech internship. The content is derived from direct professional interactions, colleague consultations, and team meeting insights. Secondary data sources include relevant industry literature and the company's official website.

1.4.1 Primary Data Sources

- Information from the Chief Technical Officer and professional colleagues
- Daily operational meetings and work target assessments

1.4.2 Secondary Data Sources

Auracore Tech Limited Official Website: <https://auracoretech.com/>

Chapter 2

Company's Profile

2.1 Overview

Auracore Tech is a premier technological solutions provider emphasizing digital transformation. Established in 2024, the organization aims to streamline business processes through comprehensive on-demand services, including web development, app development, digital marketing, SaaS products, and cloud services. Operating from a cooperative workspace, Auracore Tech comprises skilled developers and project managers dedicated to delivering innovative, customized client solutions.

2.2 Vision and Mission

Vision: Become the premier technological partner for digital business transformation. Mission: Empower businesses through cutting-edge technological solutions that enhance productivity, stimulate innovation, and optimize customer satisfaction.

2.3 Operational Framework

2.3.1 Stakeholders and Partners

Stakeholder perspectives are critical in determining product direction and ensuring alignment with market demands.

2.3.2 Program Management

Program Managers oversee project execution, serving as primary technical contacts and ensuring quality assurance and customer interface standards meet product owner specifications.

2.3.3 Product Ownership

Passion and expertise make product owners the key personnel for requirement collection and stakeholder relationship development while maintaining investor alignment. The role of product owners plays an essential part in keeping development activities in line with stated strategic objectives.

2.3.4 Quality Ownership

The development of software relies on quality assurance functions for ensuring end products match defined quality criteria. QA engineering assignments are influenced by project complexity together with project-specific parameters.

2.3.5 Service Portfolio

Auracore Tech offers diverse digital transformation services:

- **SaaS Products:** The company partners with Indian companies to deliver automation marketing systems alongside educational management platforms
- **Software Development:** The company delivers web and application development services to create software which responds to users' needs
- **Digital Marketing:** Comprehensive marketing support through specialized departmental expertise
- **UI/UX Design:** The team achieves creative user interface design which delivers optimal user interactions through development processes.

2.3.6 User Experience Design

User experience designers work to create interfaces that users can understand naturally while they collect feedback from both users and project managers and stakeholders.

2.4 Software Development Methodology

Through Agile methodology implementation, Auracore Tech guides developers to perform flexible, iterative work across multiple developmental stages:

2.4.1 Requirement Identification

Through collaborative efforts teams work with stakeholders to identify project needs along with stakeholder expectations and potential solution options.

2.4.2 Proof or Concept

Development teams create prototypes to validate concepts and obtain stakeholder feedback.

2.4.3 Contextual Research

Product managers conduct comprehensive research, including resource allocation, competitive analysis, and market studies to guide development frameworks.

2.4.4 Iterative Releases

Gradual update releases enable continuous testing and user feedback integration.

2.5 Recruitment Process

Auracore Tech implements a rigorous hiring procedure:

- Resume Submission: Candidates provide cover letters and professional profiles
- Initial Screening: Preliminary interview assessing qualifications and candidate interest
- Advanced Screening: Senior management conducts in-depth technical and cultural compatibility interviews
- Final Selection: Comprehensive candidate evaluation based on responses, background, and potential team contributions

Chapter 3

Internship Objectives at Auracore Tech

3.1 Technical Training

An organized internship program in the full stack development department at Auracore Tech provides comprehensive training focused on software development, emphasizing backend and cloud technologies. Participation in real projects bridges the gap between academic knowledge and practical business applications.

3.2 Collaborative Environment

Auracore Tech prioritizes teamwork and collaboration. Interaction with seasoned developers and cross-functional teams improves professional communication skills and collaborative capabilities.

3.3 Mentorship Support

The internship aims to develop technical proficiency across database management systems, programming languages, and frameworks. Interns gain comprehensive insights into best practices for creating reliable, scalable applications, establishing a robust foundation in front-end and back-end development.

3.4 Technical Skill Enhancement

This internship develops the capability to implement entire application distributions that use programming languages and frameworks with database management systems. The internship provides specialists with deep knowledge about creating dependable scalable products and establishes strong front-end and back-end development methods.

3.5 Holistic Professional Development

Through its internship platform, Auracore Tech delivers a range of training programs that focus on the necessary technical skills, along with the development of communication and critical thinking and the instruction of time management. The program develops essential practical abilities while teaching technical subjects, while also focusing on nurturing critical thinking skills along with effective communication and organized time management techniques. The collective learning methods demonstrate that interns become capable of performing their work with a broad range of skills and improved job readiness.

3.6 Industry Standards Exposure

The internship provides substantial experience in best practices in software engineering to the participants. Participating in this program allows participants to obtain deep knowledge about agile methodologies along with software development lifecycles, code review procedures, and version control systems. Training in this program exposes workers to multiple technology fields that foster their readiness for different roles in technological work.

3.7 Professional Networking

Professional networking serves as the primary cornerstone of the Auracore Tech internship learning module. Business activities, together with interactive peer sessions and industry-professional events, create opportunities to build meaningful professional connections. Networking aims to create potential future employment prospects and provide insightful industry perspectives.

Through these strategic objectives, the internship program is designed to maximize professional growth and establish a strong foundation in the domains of digital transformation and software development.

Chapter 4

Training Experience

4.1 Training Overview

An internship at a well-established company provides an excellent starting point for emerging professionals. Specialized IT internship programs significantly contribute to career development. At Auracore Tech Limited, the intern was assigned to the Metro Meal Order Management Software project, working on a live project under the supervision of the senior developer.

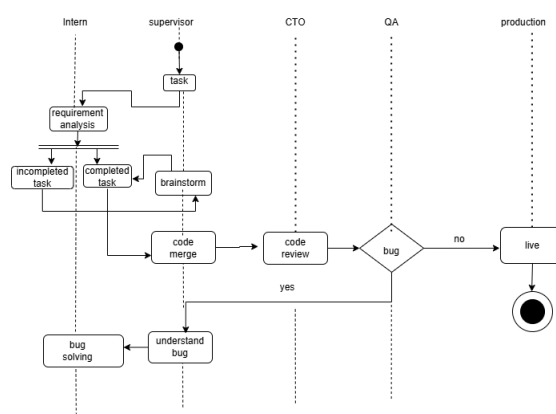


Figure 4.1: Activity Diagram of my Workflow.

4.2 Front-End Development Fundamentals

The initial phase of training emphasized mastering foundational web development technologies, including JavaScript (JS), HTML, CSS, and React JS. During the first week, considerable focus was placed on familiarizing with the organization's workflow, operational patterns, and project setup processes. Subsequently, I was assigned to work on an Integration Management Project. My primary responsibility involved updating and refining design elements in accordance with specific client requirements, ensuring alignment with their expectations and overall project objectives. [1]

4.3 React JS Project Folder Structure

During my internship, I will be working on React.js for the frontend development. In Figure 4.3, the Order folder is highlighted, which contains several files. The primary file in this folder is index.js, serving as the entry point, while other files are utilized to display data through the Context API.[4] This folder structure is considered a best practice for organizing component code in industry-level projects. Additionally,

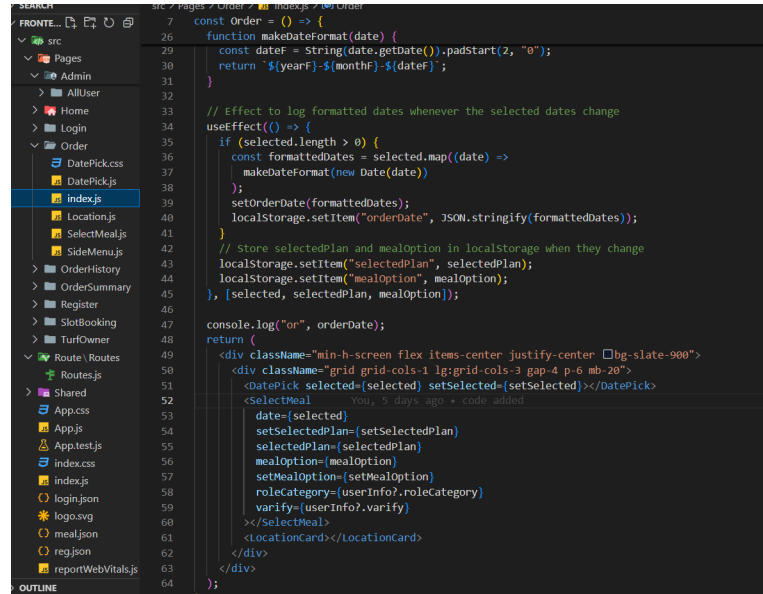


Figure 4.2: Folder Structure in React JS.

Tailwind CSS is used for designing components. Tailwind CSS simplifies the styling process by providing utility-first classes, eliminating the need to create external CSS files in most cases. Exceptions may arise for highly specific design requirements. For further details, the variety of available classes can be referred to in the official Tailwind CSS documentation. [5]

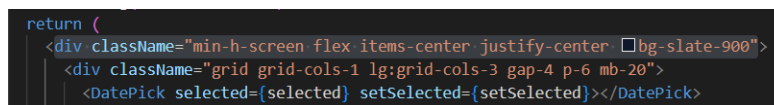


Figure 4.3: Use of Tailwind CSS in React JS file

4.4 Initial Task: Carousel Implementation

After acquiring knowledge of React.js and JavaScript coding conventions, I was assigned a task to make modifications to the frontend design. The specific assignment involved implementing a carousel feature. Initially, the meal types were displayed in a card view, but the requirement was to replace this with a carousel layout.

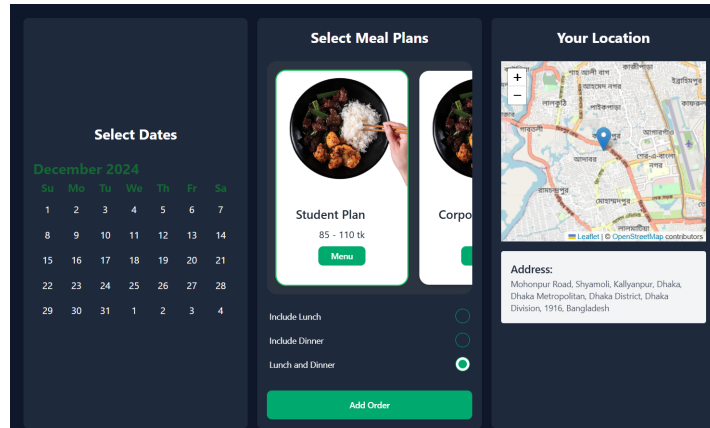


Figure 4.4: Carousel Design Implementation

To achieve this, we utilized Tailwind CSS in combination with DaisyUI, a Tailwind CSS-based framework that simplifies the development of UI components.[2] DaisyUI provided pre-built components, including a carousel, which significantly streamlined the implementation process. The detailed documentation of DaisyUI was instrumental in customizing the carousel according to the client's requirements. As depicted in Figure 4.5, the marked section of code represents the modifications made to display the cards within the carousel component. This involved leveraging DaisyUI's carousel features to create a dynamic and visually appealing layout. Additionally, the carousel component had to be integrated into another folder, which was then accessed and managed using Context API through the index.js file. This approach ensured proper data flow and state management within the application.

```
return (  
  <div className="p-4 bg-slate-800 rounded-lg shadow-lg max-w-full ">  
    <h2 className="text-2xl font-bold text-center text-white mb-6">  
      Select Meal Plans  
    </h2>  
    <div className="carousel carousel-center bg-neutral rounded-box max-w-full space-x-5 p-4">  
      <div className="carousel-item">  
        { /* Regular Plan */ }  
      </div>  
      </div>  
      <div className="carousel-item">  
        { /* Standard Plan */ }  
      </div>  
      <div className="carousel-item">  
        { /* Premium Plan */ }  
      </div>  
    </div>  
  </div>  
)
```

Figure 4.5: Code for Carousel component

This task not only enhanced the user interface but also reinforced best practices in frontend development using modern tools and frameworks.

4.5 Managing Data Flow with Context API

As part of the project, we utilized the Context API, which facilitates the transfer of data between components. While completing my first task my goal was to passing data from the index.js file to other component. Basically retrieving selected meal type and option from the select meal component.

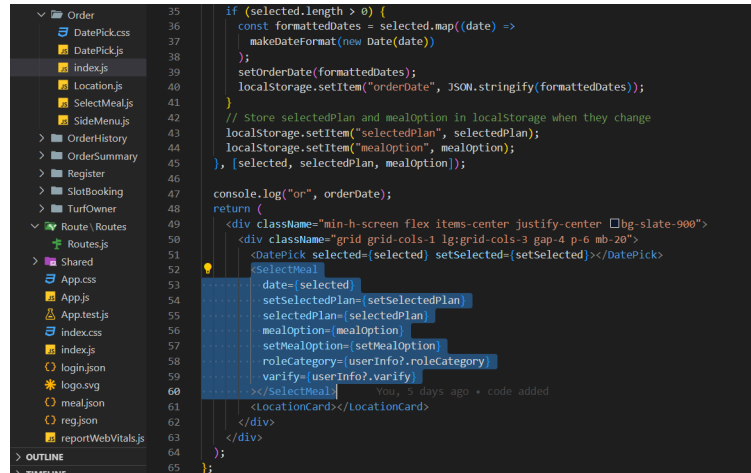


Figure 4.6: Passing data to other component

To achieve this aim, I first created a state variable using React's useState hook, which allows for dynamic storage of information. This state variable was subsequently forwarded to the SelectMeal component for management of the selected meal and option. In the SelectMeal component, I created an onClick function for storing the selected meal and option, thus modifying the state appropriately.

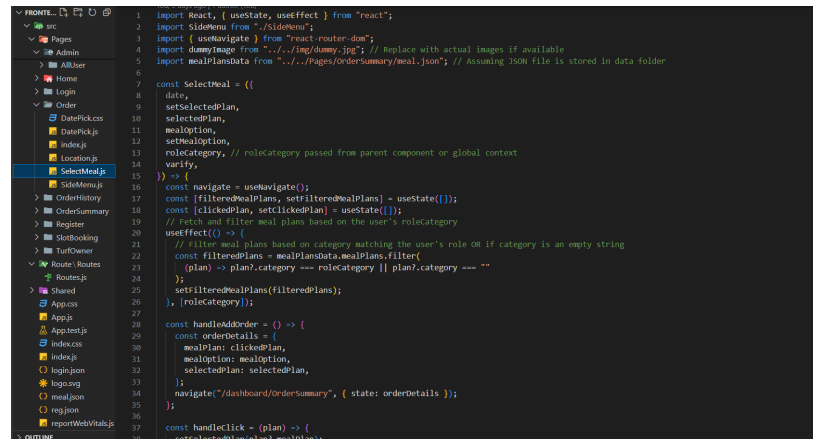


Figure 4.7: Using data from index.js file.

The workflow in Figure 4.7 helped ensure the efficient management of selected meal data and its smooth flow between components. The use of Context API along with state management enabled the data handling to be improved while ensuring consistency throughout the application.

4.6 Number varification with OTP system

The number verification system was integrated into the registration page, enabling users to verify their phone numbers through an OTP-based process. Bulk SMS BD was utilized as the SMS service provider for sending OTPs.

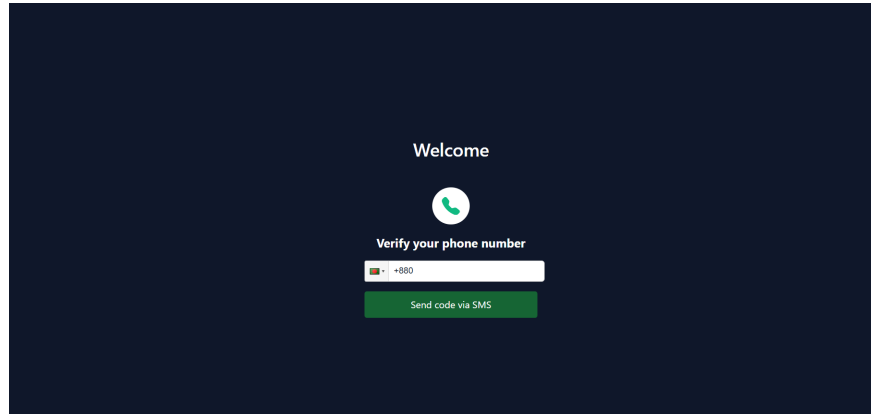


Figure 4.8: Number input UI

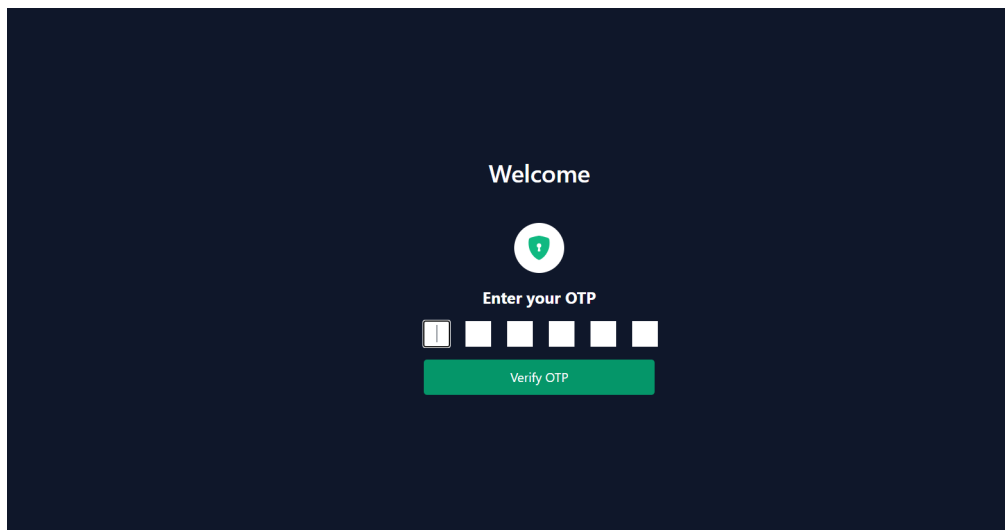


Figure 4.9: OTP input UI

My role involved designing two distinct user interface layouts for this system. The first layout was crafted to allow users to input their phone numbers, initiating the verification process. Upon providing a valid phone number, the second layout was displayed, enabling users to enter the OTP they received. This approach ensured a seamless and user-friendly experience for phone number verification.

```

return (
  <section className="bg-slate-900 flex items-center justify-center h-screen">
    <div>
      <Toaster toastOptions={{ duration: 4000 }} />

      <div className="w-80 flex flex-col gap-4 rounded-lg p-4">
        <h1 className="text-center leading-normal text-white font-medium text-3xl mb-6">
          Welcome {userInfo?.name}
        </h1>
        {showOTP ? (
          <>
            <div className="bg-white text-emerald-500 w-fit mx-auto p-4 rounded-full">
              <BsFillShieldLockFill size={30} />
            </div>
            <label
              htmlFor="otp"
              className="font-bold text-xl text-white text-center">
              Enter your OTP
            </label>
            <OtpInput
              value={otp}
              onChange={setOtp}
              OTPLength={6}
              otpType="number"
              disabled={false}
              autoFocus
              className="opt-container" />
            </>
            <button
              onClick={onOTPVerify}
              className="bg-emerald-600 w-full flex gap-1 items-center justify-center py-2.5 text-white rounded">
              {loading && (
                <CgSpinner size={20} className="mt-1 animate-spin" />
              )}
              <span>Verify OTP</span>
            </button>
          </>
        ) : (
          <>
            <div className="bg-white text-emerald-500 w-fit mx-auto p-4 rounded-full">
              <BsTelephoneFill size={30} />
            </div>
            <label className="font-bold text-xl text-white text-center">
              Verify your phone number
            </label>
            <PhoneInput country={"bd"} value={ph} onChange={setPh} />
            <button
              onClick={sendSms} // Call sendSms instead of onSignup
              className="bg-green-800 w-full flex gap-1 items-center justify-center py-2.5 text-white rounded">
              {loading && (
                <CgSpinner size={20} className="mt-1 animate-spin" />
              )}
              <span>Send code via SMS</span>
            </button>
          </>
        )}
      </div>
    </div>
  </section>
)

```

Figure 4.10: code for implementing number varification UI

4.7 Implementing Order History Table UI

This project primarily focuses on enabling users to place orders and view their order history. To provide a comprehensive view of order details and statuses, I implemented a dynamic table within the user interface. The data for the table is fetched using TanStack React Query, which ensures efficient and real-time data handling.

User Profile

Make Order

Order History

Notification

Check Your Order Status

If your order is pending more than 20 mins after order please contact 01855561001

Order ID	Order Dates	Total Price	Container Returned	Status
fUj8h7fll	2024-11-26 (lunch-1, dinner-1)	£ 560	0/4	approved
	2024-11-27 (lunch-1, dinner-1)			
	2024-11-28 (lunch-1, dinner-1)			
lU8grc9	2024-11-29 (lunch-1, dinner-1)	£ 520	0/4	approved
	2024-12-01 (lunch-1, dinner-1)			
	2024-12-02 (lunch-1, dinner-1)			
p\$3RqhNN	2024-12-03 (lunch-1, dinner-1)	£ 1260	0/10	pending
	2024-12-04 (lunch-1, dinner-1)			
	2024-12-05 (lunch-1, dinner-1)			
	2024-12-01 (lunch-1, dinner-1)			
	2024-12-02 (lunch-1, dinner-1)			
DTWkP9%9g	2024-12-03 (lunch-1, dinner-1)	£ 1260	0/10	pending
	2024-12-04 (lunch-1, dinner-1)			
	2024-12-05 (lunch-1, dinner-1)			

Figure 4.11: Order history table UI

[illegible]

Figure 4.12: Code for UI design

To design the table, I leveraged DaisyUI, a CSS framework, as the foundation. I initially utilized pre-defined components and code snippets from DaisyUI's official documentation, customizing them extensively to align with the provided Figma design specifications. These modifications ensured the table's appearance and functionality adhered to the project's design and usability standards, offering users a seamless and visually appealing experience.

4.8 Implementing User Role Update UI

This project categorizes customers into two primary groups: students and corporates. To ensure accurate classification, users are required to confirm their identity through their user profiles, which undergo verification via the admin panel.[6]

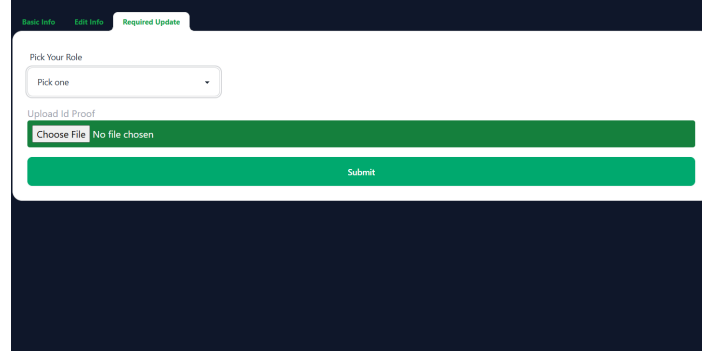


Figure 4.13: User role update UI

```
<input
  type="radio"
  name="my_tabs_2"
  role="tab"
  className="tab text-xs text-green-600 font-bold"
  aria-label="Required Update"
  defaultChecked
/>
<div
  role="tabpanel"
  className="tab-content bg-base-100 border-base-300 rounded-box p-6"
>
  <form
    className="space-y-4"
    onSubmit={handleSubmit(handleSubmitVerify)}
  >
    <div
      className="form-control w-full max-w-xs"
    >
      <div className="label">
        <span className="label-text">Pick Your Role</span>
      </div>
      <select
        className="select select-bordered"
        onChange={(e) => setUserRoleCategory(e.target.value)}
      >
        <option disabled selected>
          Pick one
        </option>
        <option>student</option>
        <option>corporate</option>
      </select>
    </div>
    <div className="form-control">
      <div className="label" block text-gray-400>Upload Id Proof</label>
      <input
        type="file"
        onChange={(e) => generateVerifyLink(e.target.files[0])}
        className="w-full p-2 rounded bg-green-700 text-white"
        required
      />
    </div>
    <button type="submit" className="btn btn-success text-white w-full">
      Submit
    </button>
  </form>
```

Figure 4.14: Code for User role update UI

As part of this process, users must upload a valid photo ID and select their respective user roles (student or corporate). My primary responsibility was to design and implement a user-friendly form to collect the necessary data. This form was developed to facilitate the seamless submission of user details, ensuring that the identity verification process is straightforward and efficient for both users and administrators.

4.9 Backend Development Fundamentals

This project is based on MERN stack therefore we are using database MongoDB moreover using Node js for the backend. MongoDB is a NO SQL database where we do not have to maintain table relationship which was very clear to me from my CSE370 course. To connect database with my backend code we have to just add database table id and pass and then connect each database table with a constant variable. I was given two task to do for the backend those are connecting order data to show in user order history table as well as user varification request from the admin panel. Basic node js fuctions are- get,post,patch/put and delete. Get function helps to get or read all the data from database table. Put/Patch basically work as update a data. But while we are deleting a dtaa and then update newly then we use put. For the insertion we use Post method and lastly for deleting a data we use Delete method. [3]

4.10 Connecting Order Data to the Database

This project is built using the MERN stack, with the backend architecture developed primarily in Node.js. My initial task on the backend involved implementing functionality to retrieve and display user order history.



Figure 4.15: Database for getting order history

To achieve this, I structured a query to fetch data from the database by utilizing the user's email address as the search parameter. This approach ensures that only the specific order data associated with the respective user is retrieved. The retrieved information is then seamlessly integrated with the frontend, allowing users to view their order history in an organized and intuitive manner. This task required close co-ordination between the backend and frontend to ensure accurate data representation and a smooth user experience.

4.11 Managing User Varification Request

This task was basically user varification part where admin will varify user varification request with help of identity card image that user has uploaded. Here we get the data from user varification table to show admin.



Figure 4.16: Database for getting user varification request.

```
app.post("/varifyUser", async (req, res) => {
  const data = req.body;
  const result = await varificationCollection.insertOne(data);
  console.log(result);
  res.send(result);
});
app.get("/AllVarifyReq", async (req, res) => {
  let query = {};

  const result = await varificationCollection.find(query).toArray();
  res.send(result);
});
```

Figure 4.17: Fetching order verification request.

To achieve this, I structured a query to fetch data from the database by utilizing the user's email address as the search parameter. This approach ensures that only the specific order data associated with the respective user is retrieved. The retrieved information is then seamlessly integrated with the frontend, allowing users to view their order history in an organized and intuitive manner. This task required close coordination between the backend and frontend to ensure accurate data representation and a smooth user experience.

4.12 Updating User Role From Backend

My initial task helped me to understand get and post method. This task was to update user role while verified by the admin. Here From the frontend I took id of that user to update that user specific field. where user role field will be updated from the backend.**dinu2021automatic**

```
app.patch("/users/verify/:id", async (req, res) => {
  const id = req.params.id;

  let query = {};
  const options = { upsert: true };
  let updatedDoc = {
    $set: {},
  };
  if (req.body) {
    query = { _id: new ObjectId(id) };
    updatedDoc = {
      $set: {
        roleCategory: req.body.role,
      },
    };
  }
  const result = await usersCollection.updateOne(
    query,
    updatedDoc,
    options
  );
  res.send(result);
  console.log("r", result);
});
```

Figure 4.18: Code for getting user varification request.

Here we have applied PATCH method to update only user role field. Snipped line of 129 show updating field as user role.

4.13 Learning Github

My company use github to store the code and also to the team work. **katikalapudi2012associating1** Here as I am new I just pushed the code to my supervisor branch then he checks the code and the merge it to the main branch. I learned git basic while working in this project. I have learned to push the code in a branch and pull the latest code.

Chapter 5

Challenges Faced During Internship

The development of a meal order management system for users in both student and corporate categories presents a range of challenges. These challenges are categorized into frontend development, backend development, user identity verification, order management, and number verification systems, each of which is discussed in detail below.

5.1 Frontend Development Challenges

Creating an interactive and responsive user interface is extremely difficult. The primary features, such as carousels, order history tables, and update user role forms, have to be functionally compliant while at the same time being visually compliant with the design specifications made in Figma. Getting these features responsive on multiple devices adds another layer of complexity. Moreover, using utility-first libraries like DaisyUI entails extensive customization to meet specific needs. throughout the project, ensuring the integrity of a cohesive and lovely aesthetic.

5.2 Backend Development Challenges

Backend development, primarily with Node.js, involves managing efficient database queries and API coordination. For instance, retrieving user-dependent data, such as order history, necessitates optimized queries to ensure smooth performance. Coordinating data flow between backend APIs and frontend components is crucial to providing real-time updates. Additionally, including secure methods to data handling, especially for role confirmation and order history processing, contributes to the complexity of the backend framework.

5.3 User Identity Verification

In order to categorize users under the student and corporate user categories appropriately, identification verification through the user profile must be provided by the system. Users need to upload a valid picture identification and declare their respective roles. The information should then be authenticated through the admin panel. A form with a user-friendly interface should be developed that supports multiple file formats and employs safe data storage measures to ensure the system's reliability.

5.4 Order Management System Challenges

Order management system requires efficient monitoring and representation of orders and their status. The use of a dynamic table to represent this data in a way that is user-friendly is a major challenge. The use of libraries like TanStack React Query in handling real-time data fetching minimizes the number of API calls but makes it harder to efficiently handle data as well as synchronization with frontend components. Tailoring the table to suit particular project requirements adds another level of complexity.

5.5 Number Verification System

The phone number verification system uses an OTP-based process to authenticate users during registration. This involves designing two distinct user interface layouts: one for phone number input and another for OTP submission. Integrating these layouts with the Bulk SMS BD API for OTP delivery and validation requires technical expertise. Ensuring the seamless functioning of this system, along with error-free OTP delivery and validation, is vital for user authentication.

Chapter 6

Internship Takeaways and Outcomes

Internship duration help a student to grow professional and interpersonal skills. During my internship I have faced lots of challenges which helped me to grow my professional skills besides this 6 months long environment helped me to reach my goals.

6.1 Insights into Enterprise Software Development

Another purpose I set for the internship was to become familiar with the concepts and practical aspects of the process of constructing enterprise software solutions. The numerous issues and concerns that relate to and affect the process of educational reform were discussed. Factors that are essential to creating grand, stable, and highly elastic software solutions. In order to learn and realize the concepts of modular architecture and effective data management ultimately, security standards, and interaction with current corporate systems were all part of this.

6.2 Practical Experience with Modern Technologies

The internship gave me an opportunity to be acquainted with several new technologies. The principles and processes used in establishing corporate software. I was able to work with current frameworks, database, and development solutions. This hands-on experience I developed the necessary skills and gained amongst other competency in Information technology including JavaScript, TypeScript, Node technologies of enterprise software development. JavaScript, ReactJS, AngularJS and others, are the frameworks that help in making robust web applications.

6.3 Developing Problem-Solving and Troubleshooting Skills

This internship program helped me to understand and debug a problem. I have gained experience of inspecting an element to understand the problem of my problem. Moreover, using console helps me to identify data that coming from backend and also make changes according the data.

6.4 Developing Communication and Time Management Skills

During my internship I always have to report my supervisor and my office seniors. Moreover I have to attend client meeting with my seniors to gain communication skills to know client demands and convince with requirements. Besides I always had to maintain 7 hours of time on the working day for the internship. Sometimes I left with free time during office hours where I learn different skills based on my job requirement. Therefore this internship helped me to manage time effectively.

6.5 Continuous Learning and Personal Development

After successfully completing my internship from Auracore Tech I have learned there is no ending of learning about modern frameworks. My supervisor also advised to learn modern frameworks, therefore I have starter learning Next JS which is a powerful version of React Js. Moreover I am practicing TypeScript for developing my skills.

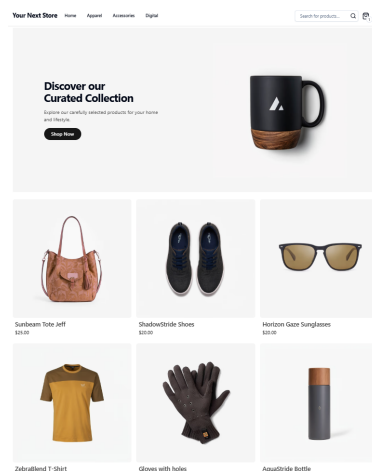


Figure 6.1: My recent project.

Chapter 7

Conclusion

The internship at Auracore Tech Limited has been an enriching learning experience, providing a hands-on exposure to full-stack development as well as merging theoretical knowledge with real-world software engineering practice. Direct exposure to modern technologies, development processes, and basic problem-solving methodologies to achieve success in the technology industry was provided by the internship through active involvement in the Metro Meal Order Management Software project.

The majority of the internship involved frontend development, where the intern had extensive interaction with React.js, Tailwind CSS, DaisyUI, and Context API to design interactive and user-oriented interfaces. The primary activities in frontend development involved the implementation of a carousel layout for different meal types using DaisyUI and Tailwind CSS, thus enabling a contemporary and responsive user interface. Implementation of Context API in managing data flow among components played a key role in improving state management and code effectiveness. Additionally, the intern implemented and used a number verification system that integrated an OTP-based authentication system via Bulk SMS BD, thus greatly improving security and user authentication. The second most important activity within the frontend development stage was developing an order history table user interface in which real-time data fetching was carried out using TanStack React Query for data visualization. The internship also included the development of a user role update interface that allowed administrators to authenticate users through the processing of identity verification requests and instant updating of user roles. These tasks included careful coordination with designers and backend developers both, adhering to best practices in user interface and user experience design, performance optimization, and data handling.

On the backend front, the intern worked hands-on with Node.js, Express.js, and MongoDB, thus enhancing his knowledge of server-side development and database management. One of the major backend tasks included associating order data with the database, which allowed easy retrieval and display of user order histories. Moreover, the intern was also in charge of handling user verification requests, whereby the backend handled ID verification requests from users and thereby provided administrators with a organized way of assessing and approving them. The other significant backend operation was the dynamic modification of user roles using PATCH requests to modify access levels within the database. Throughout the internship, GitHub was a key part of the development process, allowing the intern to participate in collab-

orative development, commit code, resolve conflicts on merge, and work effectively with version control. These backend tasks cemented basic principles that included RESTful API design, database management, authentication frameworks, and security best practices.

Apart from the acquisition of technical abilities, the internship also enabled the acquisition of professional competencies, including communication, collaboration, time management, and flexibility. Working in an environment that was collaborative and dynamic allowed the intern to gain a strong work ethic and the ability to address real-world problems in an efficient manner. Exposure to industry-standard software and practices, including version control, API integration, and database optimization, greatly aided the intern in preparation for subsequent software development projects.

Overall, the internship with Auracore Tech Limited not only satisfied academic requirements but also provided a strong foundation for a potential career in the tech industry. The learning experience in frontend and backend development, software design, and agile principles will serve as a vital point of reference to overcome future obstacles and succeed in the ever-evolving environment of software engineering.

Bibliography

- [1] K. Beck, M. Beedle, A. V. Bennekum, *et al.*, *Manifesto for agile software development*, Retrieved from <https://agilemanifesto.org>, 2001.
- [2] R. Wieruch, *The Road to React: Your Journey to Master Plain Yet Pragmatic React.js*. Independently Published, 2021.
- [3] *Mongodb manual: The complete documentation for mongodb*, Retrieved from <https://www.mongodb.com/docs/>, MongoDB, Inc., 2024.
- [4] *React documentation: A javascript library for building user interfaces*, Retrieved from <https://react.dev>, React Team, 2024.
- [5] *React query documentation: Manage server-state effectively in your react applications*, Retrieved from <https://tanstack.com/query/latest/docs/react/overview>, TanStack, 2024.
- [6] *Tailwind css documentation: Rapidly build modern websites without ever leaving your html*, Retrieved from <https://tailwindcss.com/docs>, Tailwind Labs, 2024.