Report on Industrial Application

Development Project

I Team Society Web Based System

From 10th February 2025 - To 30th May 2025

Industrial Application Development Project (ITE5622) for Bachelor of Science in Information Technology Degree Program

Rukman. T.M	122736344
Prashani Thirumal	522726239
S M U H Senanayake	322743795
G T T Dilhani	522943087

Department of Computer Science

Faculty of Natural Sciences

The Open University of Sri Lanka

Acknowledgement

We would also like to thank our mentors, Ms. Wasana C. Uduwela, Mr. Duminda De Silva, Mr. Pasindu and Mr. Nipuna, who have given their time and advice throughout the duration of this project. Your constructive criticism and advice have been greatly appreciated, and we are grateful for the effort and time invested in shaping us.

Finally, we would also like to express our gratitude to our batchmates and friends whose presence and encouragement have made all the struggles pay off. Your companionship and encouragement have kept us going through the challenges and assisted us in celebrating the victories.

To our parents, teachers, mentors, and to our batchmates, thank you for being part of this journey and for making this project a reality.

Preface

This report captures the efforts gone into designing the I-Team Society Web Application as part of the fulfillment of the industrial training requirement under the Bachelor of Science in Information Technology at the Open University of Sri Lanka.

The system was developed to make student society management in the Department of Computer Science better by replacing outdated manual procedures with a web-based, unified platform. The program enables students and personnel to enroll, manage membership, receive reminders, and attend society events in a streamlined process.

The project took advantage of a structured software development lifecycle of analysis of requirements, design, development, and testing. The most recent technologies and tools such as React.js, Tailwind CSS, Vite, Supabase, and Figma were used to develop a responsive and scalable system that complies with the current industry standards.

This report summarizes the entire development process, issues encountered, decisions made, and learnings gained throughout the project. It reflects how theoretical knowledge was converted into real application and emphasizes teamwork, problem-solving, and modern development approaches.

The I-Team Web Application demonstrates the importance of internet-based solutions in school environments and contributes to enhancing the effectiveness of student society management.

Table of Contents

List of Figures

List of Tables

Chapter 01 - Introduction

1.1 Introduction

The I-Team Society Web Application is a centralized, web-based system developed to streamline the membership and event management activities of the I-Team Society, a student-led organization initiated by the Department of Computer Science at the Open University of Sri Lanka. Society was established to foster collaboration, innovation, and skill development among students in the fields of computing and technology.

Managing student and staff memberships in university-affiliated societies often presents significant challenges—ranging from inefficient paper-based systems to fragmented digital tools. Common issues such as lost records, untracked payments, poor communication, and time-consuming manual processes can hinder the effective functioning of student societies. Recognizing this need, the I-Team Society developed a dedicated membership management platform.

The I-Team Web Application was designed to serve members, offering features such as online registration, fee payments, event announcements, member approvals, attendance tracking, and automated report generation. This system not only improves administrative efficiency but also enhances the user experience for students and staff engaging with the society.

The platform emphasizes ease of use, security, and scalability, making it suitable for both small clubs and larger university-wide organizations. This report presents the objectives, development process, tools used, challenges encountered, and outcomes achieved during the creation of the I-Team Society Web Application.

1.2 Aim and Objectives

1.2.1 Aim

This project aims to design and develop a centralized web-based membership management system for the I-Team Society under the Department of Computer Science at the Open University of Sri Lanka. This system is designed to enhance the efficiency and accuracy of managing key societal functions, including member registration, event participation, internal communications, and report generation. By replacing manual and outdated methods with a modern digital platform, the solution aims to provide an intuitive and responsive user experience for both administrators and members, ensuring smoother operations and stronger engagement within the university's student society environment.

1.2.1 Objectives

- To provide a secure and user-friendly platform for student and staff registration.
- To enable society administrators to manage memberships, verify users, and monitor activities efficiently.
- To streamline event planning and attendance tracking for society-organized events.
- To implement notification and announcement features to enhance communication.
- To generate automated reports on member participation, event statistics, and payment records.
- To create a scalable and maintainable solution using modern frontend and backend technologies.
- To ensure responsive design for use across devices including desktops, tablets, and smartphones.

1.3 Proposed Solution

The proposed solution is a comprehensive web-based application designed to centralize all essential functions of a university society. It provides a unified platform for students, staff, and administrators to efficiently manage registrations, events, and communications. Key features include an online registration portal for new and returning members, eliminating paper-based processes. An admin dashboard enables user management, account verification, and event scheduling. Automated notifications ensure timely updates on events and announcements. A secure login system provides role-based access, maintaining data privacy. Real-time tracking of member participation and payments supports instant report generation for monitoring and decision-making. The system's responsive user interface allows seamless access from desktops, tablets, and smartphones, improving usability and engagement. By replacing fragmented, manual processes, the I-Team Web Application enhances operational efficiency and strengthens the society's digital presence.

1.4 System Requirements

The I-Team Society Web Application is designed with a set of core functional requirements to ensure smooth and efficient user experience for both members and administrators.

User Registration and Login:
Users can create accounts and securely log in to the system. Role-based authentication ensures that students, staff, and administrators access features relevant to their roles.

Member Profile Management:
Registered users can update their personal information, view their membership status, and track their participation history within society.

Admin Panel with CRUD Operations:
Administrators have access to a dashboard that enables them to Create, Read, Update, and Delete records related to users, events, and other content. This centralized control panel simplifies data management.

Event Creation and Scheduling:

Admins can schedule new events, edit existing ones, and share event details with members. This feature ensures timely communication and organized event management.

Attendance Tracking:

The system allows admins to record and monitor attendance for each event, providing insights into member engagement and participation trends.

Notification and Messaging Features:

Automated and manual notification options keep members informed about upcoming events, changes, and announcements via in-app messaging or email.

Report Generation:

Admins can generate various reports, such as member lists, event participation records, and payment summaries, which support data-driven decision-making and transparency.

Chapter 02 – Industrial Application Development Project

2.1 System Overview

The I-Team Society Management System is a comprehensive web application designed for The Open University of Sri Lanka's Computer Science Department student society. It provides role-based access for Students, Staff, and Administrators with complete membership management, event organization, and digital identity features.

2.2 Analysis

The analysis phase involved identifying the specific needs of the I-Team Society and understanding the problems faced in the existing manual system. Key challenges included inefficient member registration, lack of centralized event information, and poor communication between society admins and members. A detailed requirement analysis was conducted through discussions with stakeholders such as students, staff advisors, and administrators.

Functional and non-functional requirements were collected and documented. Functional requirements included user registration, event scheduling, attendance tracking, and notification features. Non-functional requirements emphasized system usability, performance, security, and responsiveness. The analysis also included identifying user roles (e.g., student, staff, admin) and their corresponding permissions.

2.3 Design

System design encompasses the structural and architectural blueprint of a product, outlining its key components and interactions to fulfill functional requirements. It serves as a roadmap for development, guiding the design, implementation, and testing phases to ensure that user expectations are met effectively.

2.3.1 Use case Diagram

Figure 1: Use Case Diagram

2.3.2 Login Activity Diagram

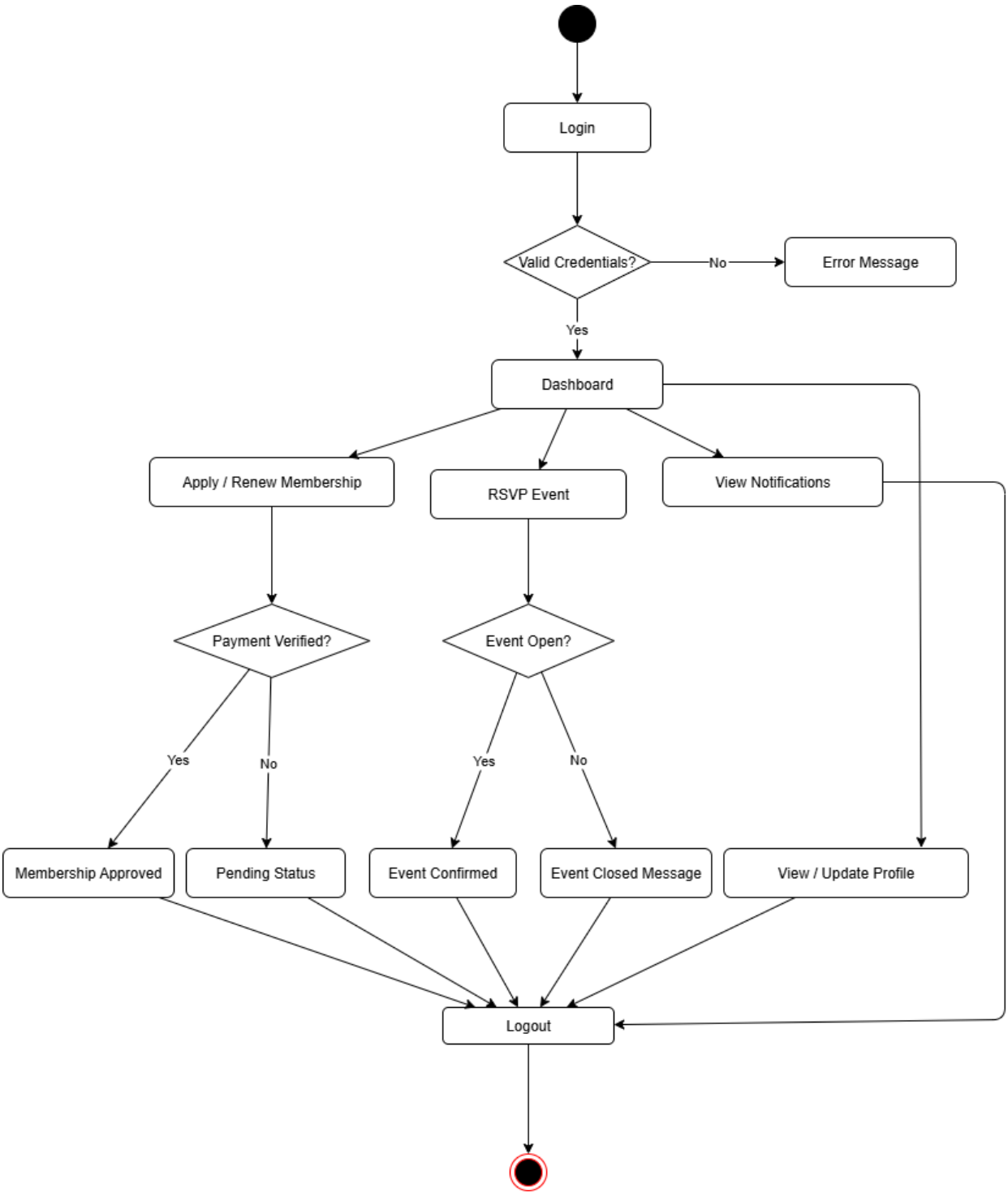

Figure 2: Login Activity Diagram

2.3.3 Login Sequence Diagram

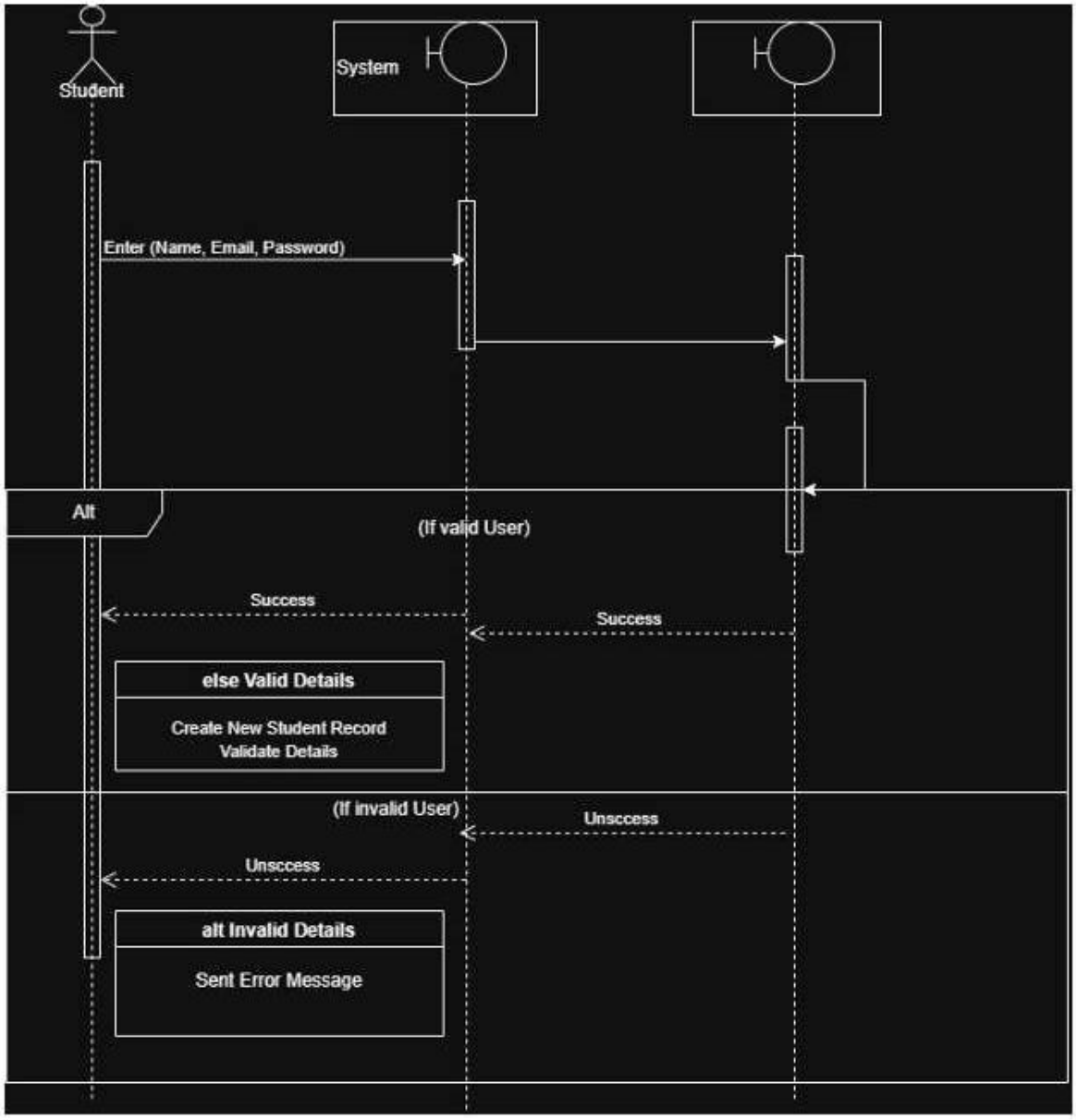

Figure 3: Login Sequence Diagram

2.3.4 ER Diagram

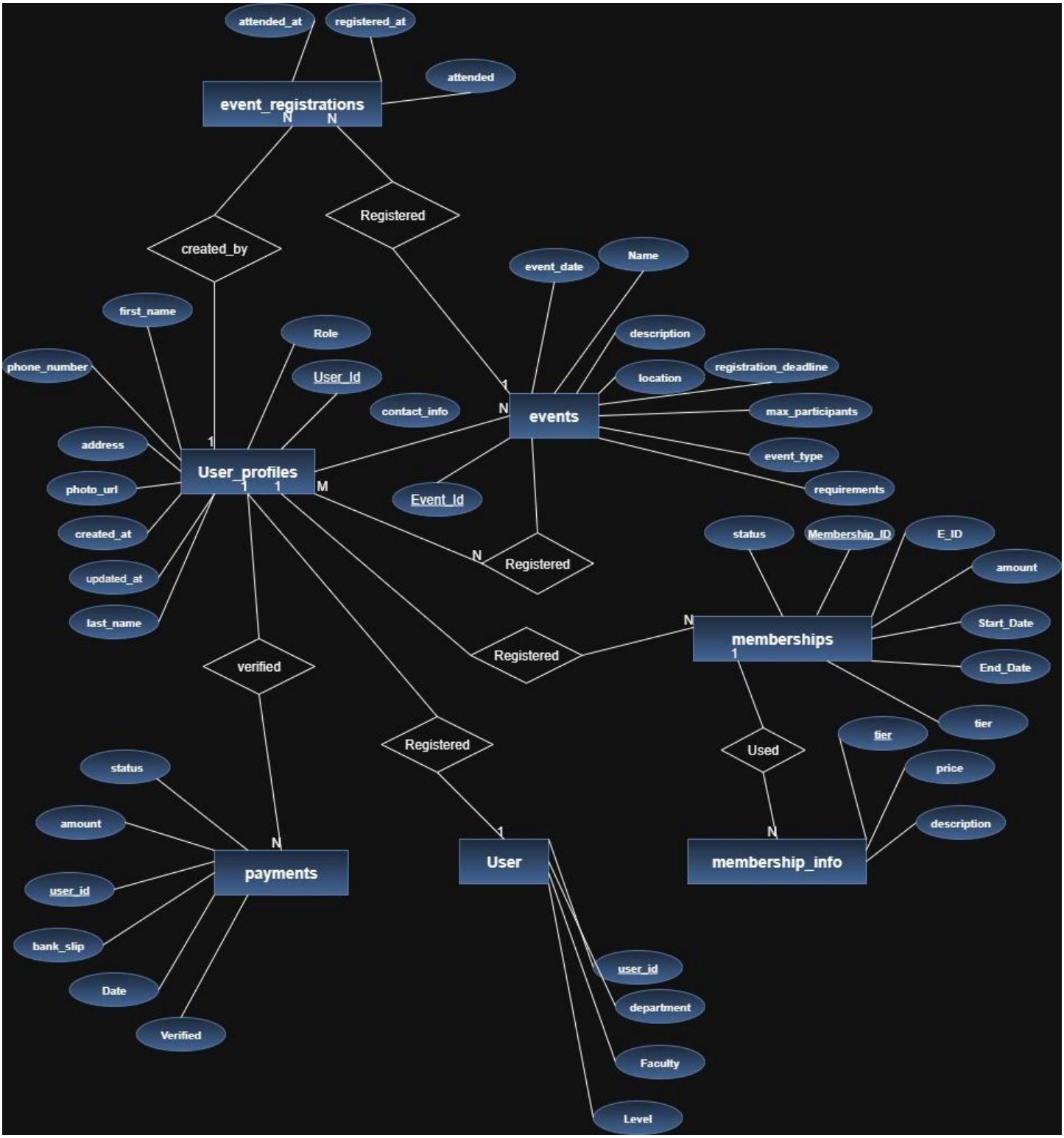

Figure 4: ER Diagram

2.3.5 Project Structure

src/

```
├── components/          # Reusable UI components
│   ├── auth/            # Authentication components
│   ├── dashboard/       # Dashboard-specific components
│   ├── events/          # Event management components
│   ├── layout/          # Layout components
│   ├── membership/      #  Membership-related components
│   ├── notifications/   # Notification system
│   └── ui/              # Base UI components
├── context/             # React context providers
├── integrations/        # External service integrations
├── pages/               # Page components
│   └── dashboard/       # Dashboard pages
├── services/            # Business logic and API calls
└── types/               # TypeScript type definitions
```

2.4 Technology Adapted

2.4.1 Frontend

React 18

The frontend of the I-Team Society Web Application was built using a modern and efficient technology stack to deliver a high-performance, responsive, and user-friendly interface. At the core of the frontend is React 18, a powerful JavaScript library widely used for building component-based user interfaces. React allows developers to break down the UI into reusable components, improving development speed and maintainability. React 18 introduced concurrent rendering and automatic batching, which enhanced the responsiveness and fluidity of the user interface.

TypeScript

To bring type safety and better development tooling to the project, TypeScript was integrated with React. TypeScript adds static typing to JavaScript, which helps catch errors during development and ensures code quality and readability. It also improves collaboration in team environments by making the codebase easier to understand and refactor.

Tailwind CSS

For styling, the application uses Tailwind CSS, a utility-first CSS framework that provides low-level utility classes for building responsive and customizable interfaces. Tailwind enables rapid prototyping and eliminates the need to write custom CSS for most layout and design tasks. It also ensures consistency across the entire application, reducing design discrepancies.

2.4.2 Backend

Supabase

The backend of the I-Team Society Web Application was developed using Supabase**,** an open-source backend-as-a-service (BaaS) platform that provides powerful tools for building modern applications. Supabase serves as a comprehensive solution by integrating database management, authentication, file storage, and real-time capabilities — all essential for a full-stack web application.

PostgreSQL

At its core, Supabase uses PostgreSQL**,** a robust and scalable relational database system that ensures data consistency, reliability, and security. PostgreSQL was used to store all application data, including user information, event details, attendance records, and payment statuses. Its support for relational data and SQL queries made it easy to manage complex relationships and generate reports efficiently.

Auth

For user authentication and access control, Supabase Auth was implemented. It supports secure sign-up, login, and session management, offering role-based access (e.g., student, staff, admin) with minimal setup. This ensured that sensitive operations, such as user verification and event management, were only accessible to authorized users.

Storage

The Storage module of Supabase was used to handle file uploads such as profile images or documents. Files are securely stored and can be accessed or managed easily through Supabase's API.

Real-time

One of the key features that enhanced the interactivity of the system was Supabase's real-time capability**,** which allowed live updates for actions such as event participation, profile changes, or admin approvals. This feature improved the user experience by keeping data updated without requiring manual page refreshes.

2.4.3 UI Components

Shadcn/ui

The I-Team Society Web Application utilizes the Shadcn/ui component library to build a clean, modern, and accessible user interface. Shadcn/ui is a flexible and developer-friendly UI toolkit built on top of Radix UI and Tailwind CSS, offering a collection of pre-designed components that follow best practices in terms of accessibility, responsiveness, and design consistency.

2.4.4 Build Tool

Vite

The I-Team Society Web Application uses Vite as its build tool to optimize and streamline the development and production workflows. Vite is a modern, fast, and lightweight frontend build tool created to improve performance during both development and deployment.One of the major advantages of Vite is its instant server start and lightning-fast hot module replacement (HMR)**.** This allows developers to see code changes reflected in the browser immediately without needing a full reload, significantly boosting development speed and productivity.

2.4.5 Deployment

Vercel

The I-Team Society Web Application is fully optimized for deployment on Vercel**,** a modern cloud platform ideal for frontend frameworks like React and Vite. Vercel enables continuous deployment by integrating directly with GitHub, so any code changes pushed to the repository are automatically deployed. This simplifies the release process and ensures the application is always up to date.

Vercel offers fast build times, global CDN delivery, SSL security, custom domains, and environment variable support**,** making it secure and user-friendly. It also supports preview deployments**,** allowing developers and stakeholders to test features before production release. With its speed, scalability, and ease of use, Vercel is an excellent choice for hosting the I-Team application efficiently and reliably.

2.5 Tools

2.5.1 Figma

Figma was used as the primary UI/UX design tool. It enabled collaborative wireframing and prototyping, making it easier to design user interfaces with real-time feedback from the team. Its browser-based interface allowed seamless access and sharing among group members and supervisors.

2.5.2 Draw.io

Draw.io was used to create system diagrams such as ER diagrams, flowcharts, and component diagrams. It supported clear visualization of the system's architecture and relationships between modules, helping in both planning and documentation.

2.5.3 GitHub

GitHub was the version control platform for managing source code. It helped the team collaborate efficiently, track code changes, and resolve merge conflicts. Branching and pull requests allowed structured development and peer review before integrating new features.

2.5.4 VsCode

VsCode served as the main code editor. With rich extension support for React, Tailwind CSS, and TypeScript, it provided a productive coding environment. Features like IntelliSense, terminal integration, and Git support made development efficient and error-free.

2.6 Implementation

2.6.1 Login/Sign up

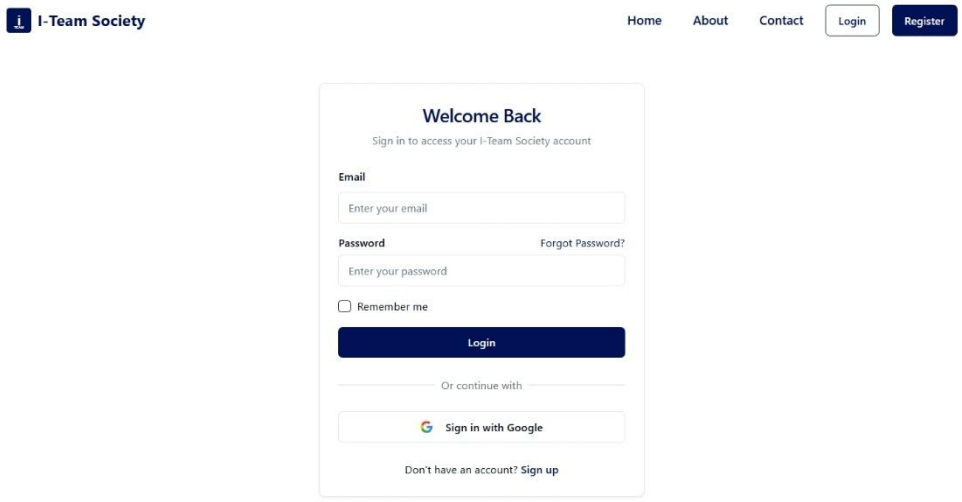

Figure 5: Login

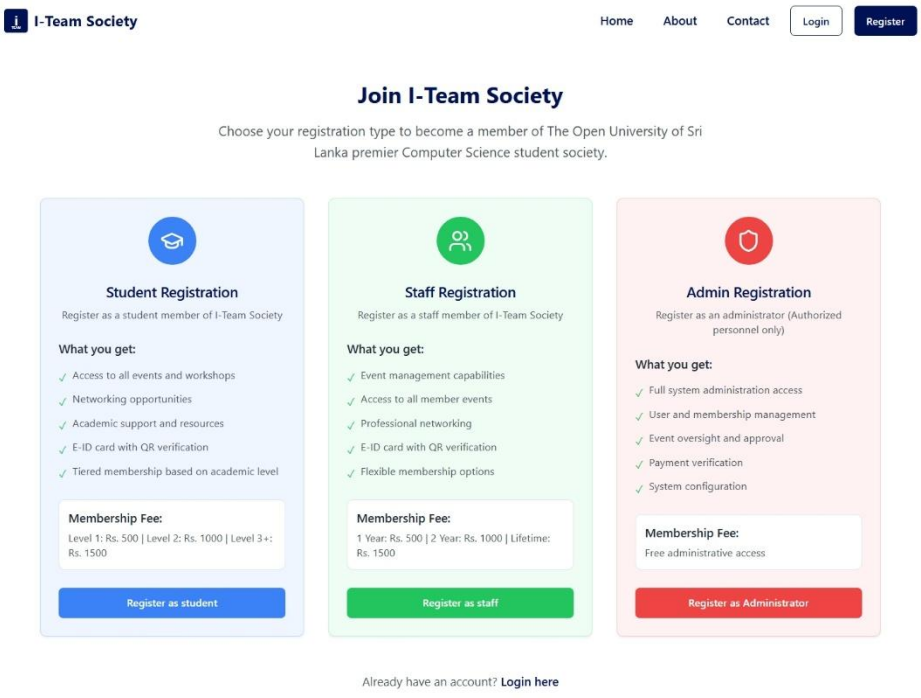

Figure 6: Register

2.6.2 Home

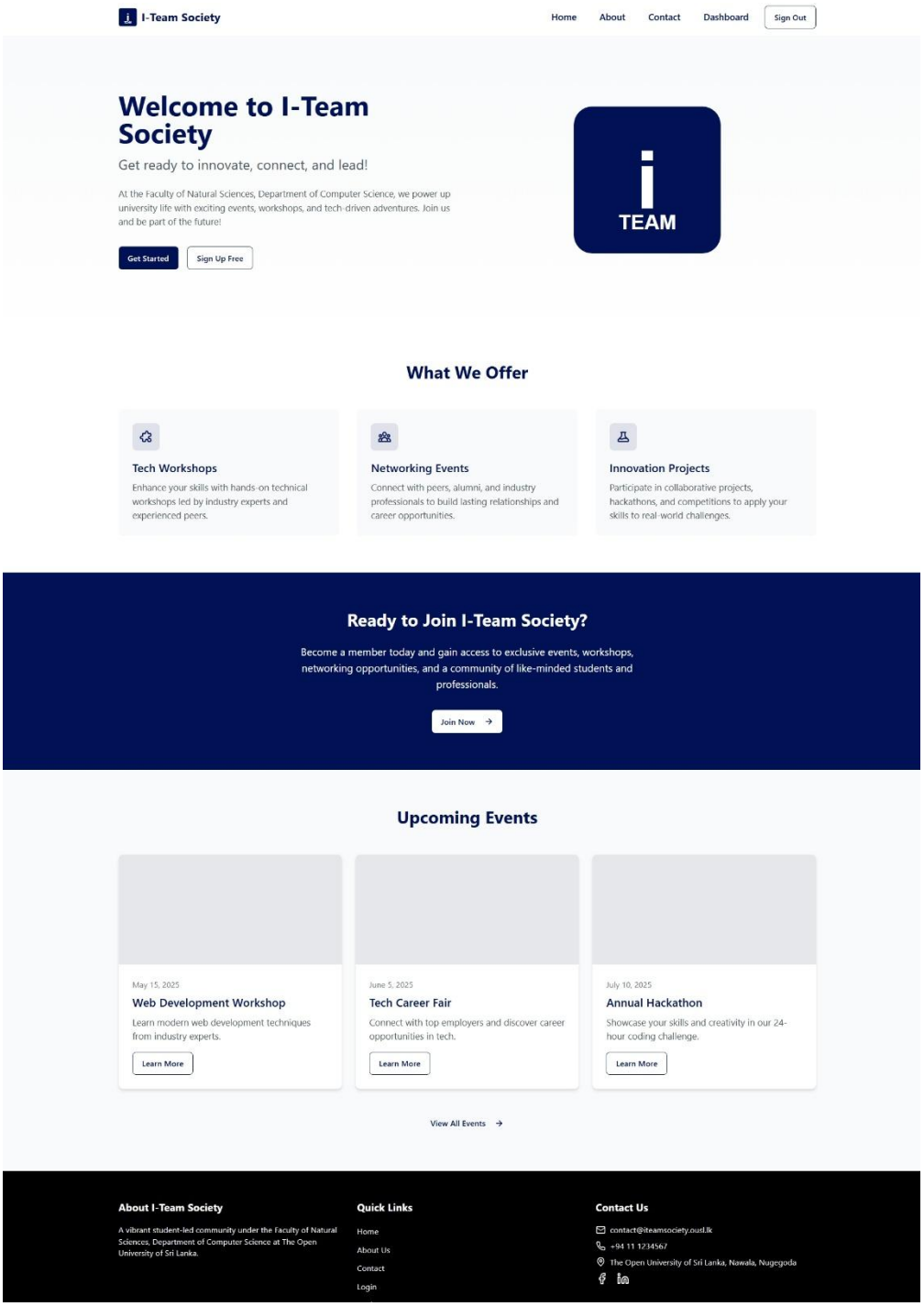

Figure 7: Home

2.6.3 Staff Dashboard

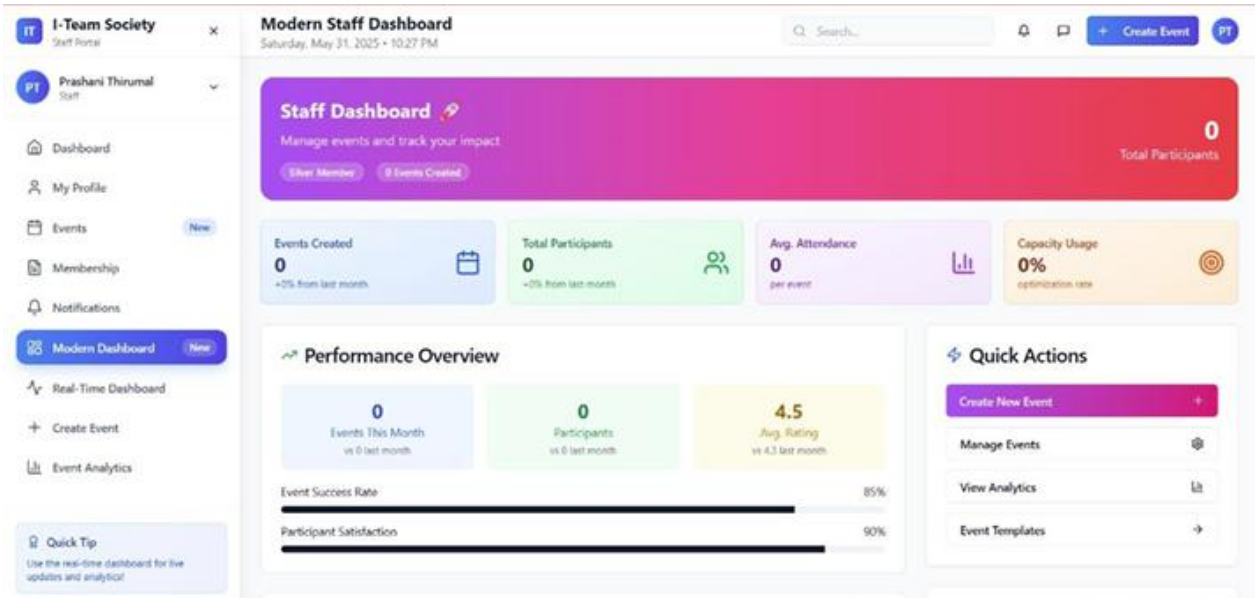

Figure 8: Dashboard

2.6.4 Membership

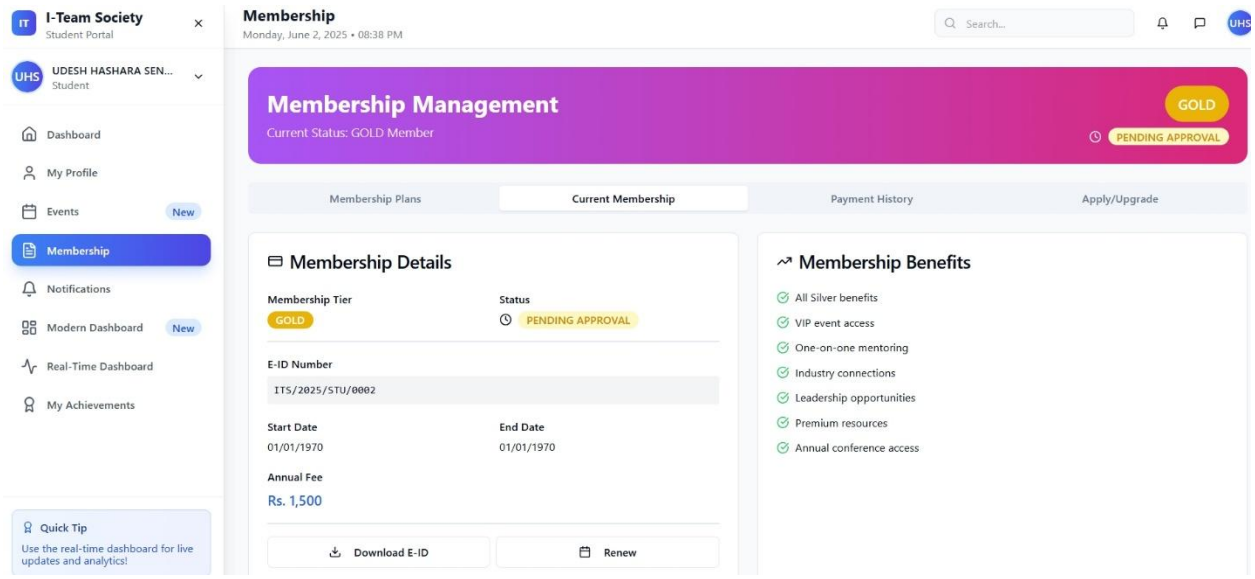

Figure 9: Membership

2.6.5 Student Dashboard

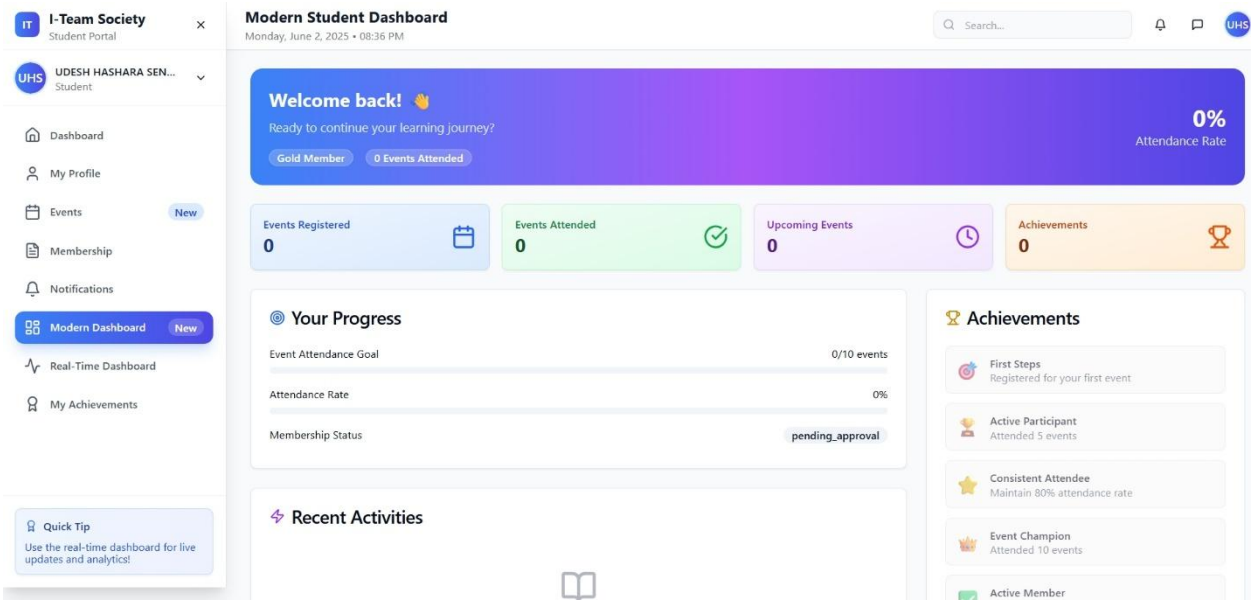

Figure 10: Student Dashboard

2.6.6 Event

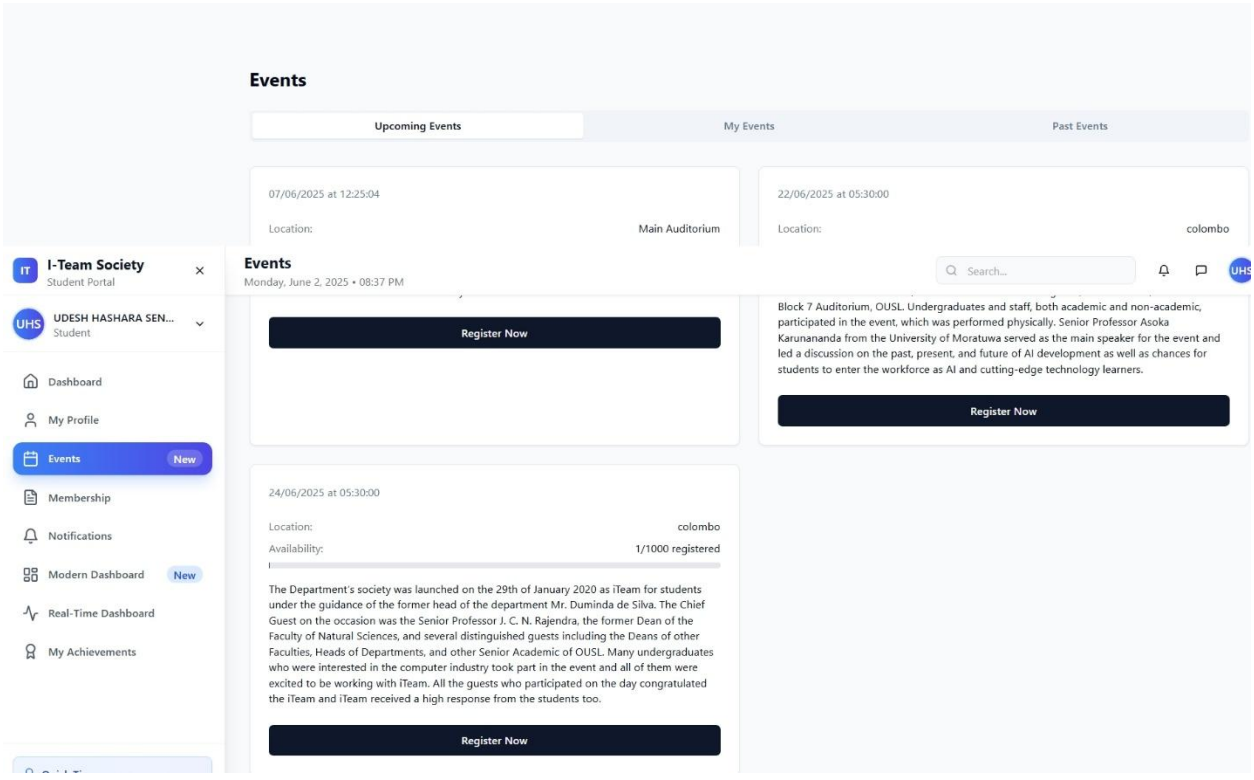

Figure 11: Event

2.6.7 Profile

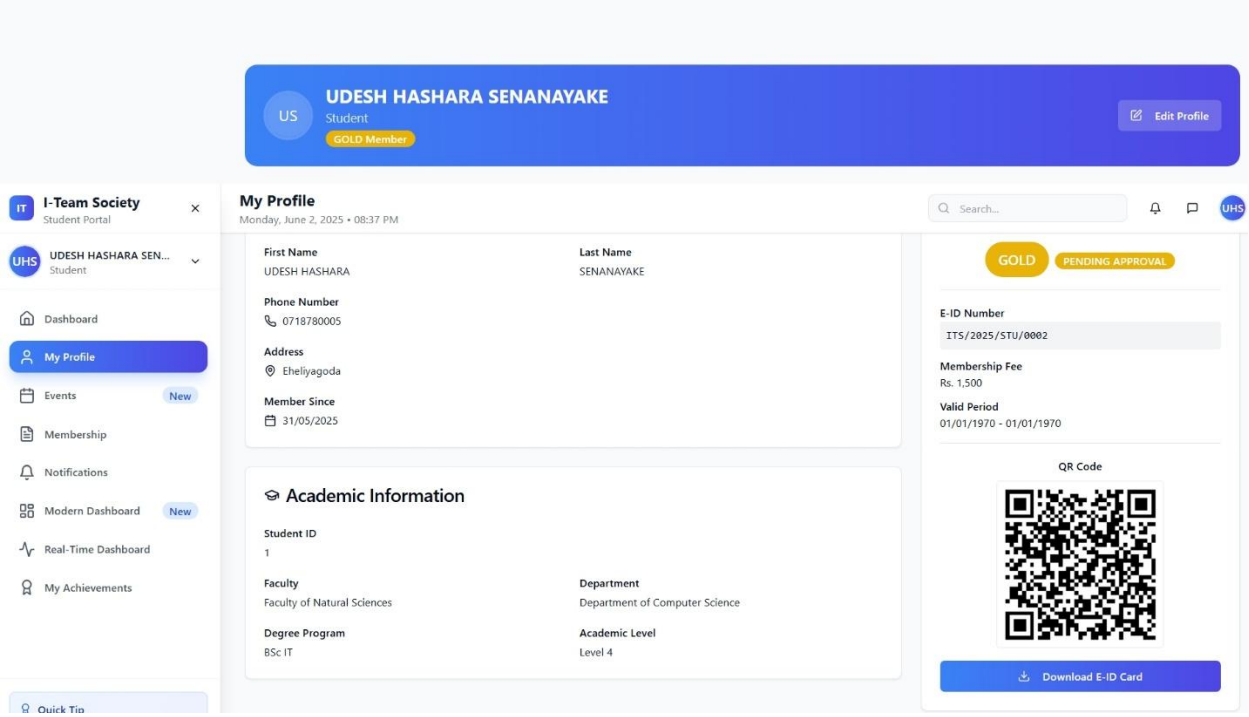

Figure 12: Profile

2.6.8 Admin Dashboard

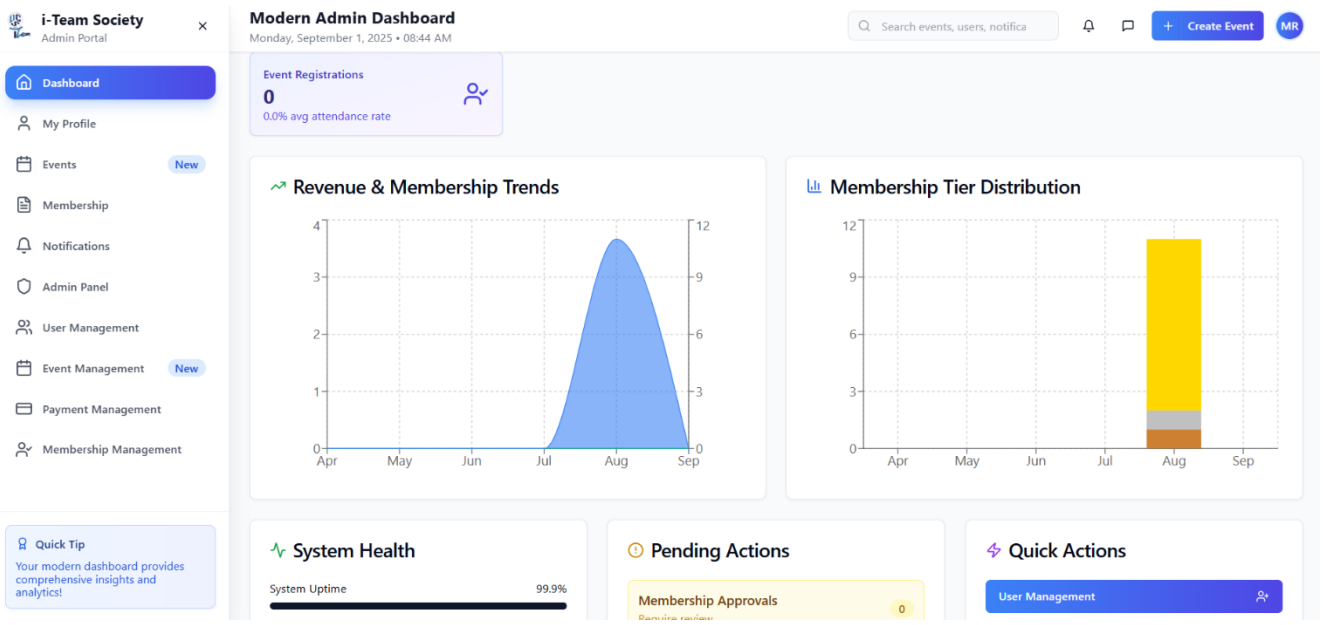

Figure 13: Admin Dashboard

2.6.9 User Management

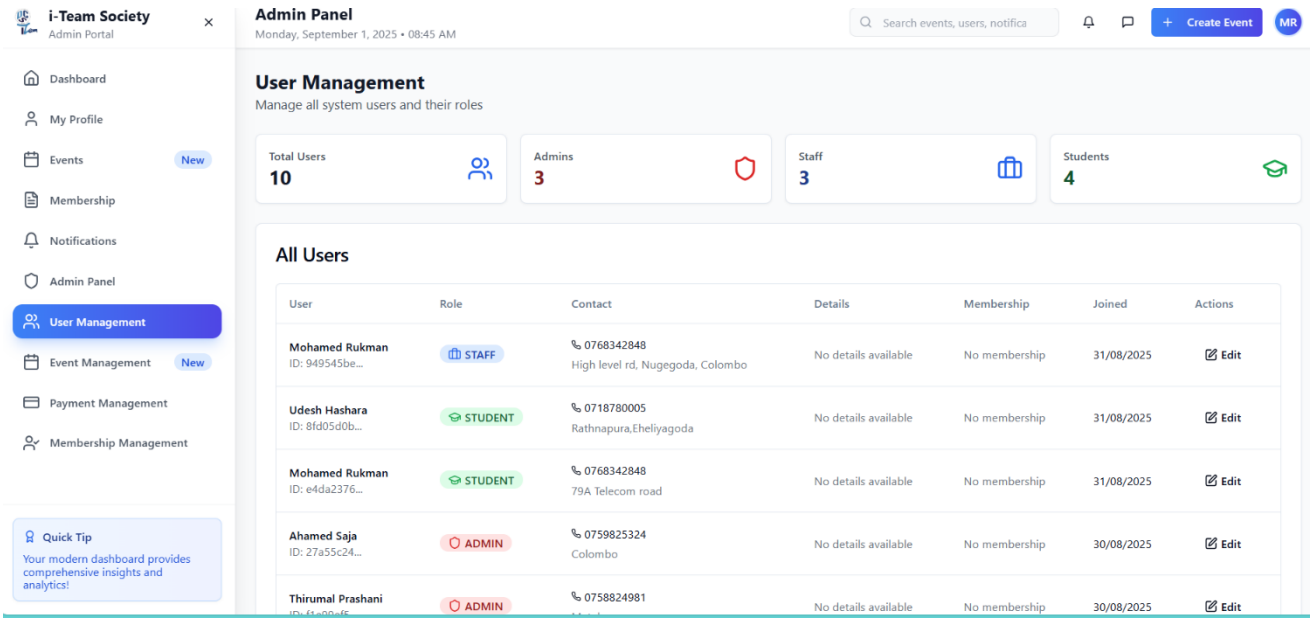

Figure 14: User Management

2.6.10 Payment Management

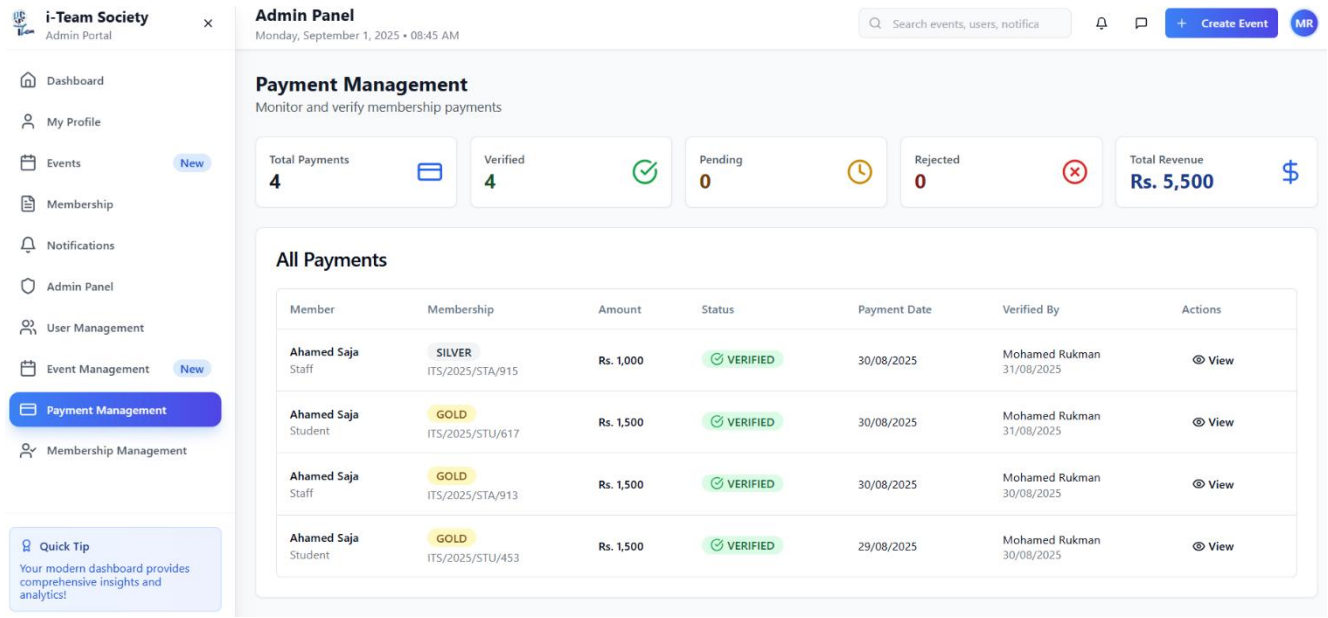

Figure 15: Payment Management

2.6.10 Membership Management

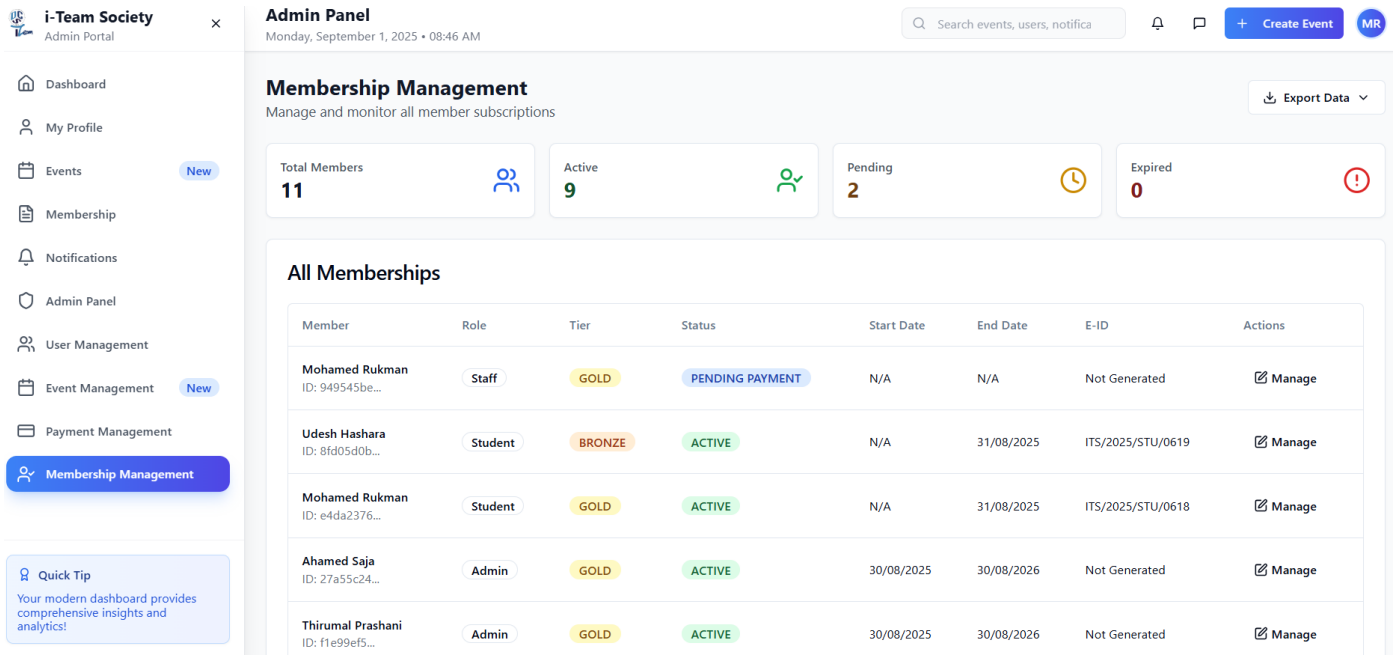

Figure 16: Membership Management

Chapter 03 – Conclusion & Further Work

3.1 Introduction

This chapter summarizes the overall outcomes of the I-Team Society Web Application project. It outlines key achievements, challenges faced, lessons learned, and future enhancements that could be made to improve the system further**.**

3.2 Discussion

3.2.1 Key Achievement

• Successfully designed and developed a centralized web-based membership management system for the I-Team Society at the Open University of Sri Lanka.

• Implemented role-based authentication and secure login using Supabase, ensuring privacy and controlled access for students, staff, and administrators.

• Built a PostgreSQL database schema for storing members, events, announcements, and payment records with optimized queries for performance

. • Developed real-time features such as attendance tracking, notifications, and report generation, improving communication and decision-making.

• Delivered a responsive frontend with React.js, TypeScript, and Tailwind CSS, ensuring accessibility across desktops, tablets, and smartphones.

• Adopted Agile methodology, version control (GitHub), and CI/CD deployment pipelines, enhancing collaboration, productivity, and project quality.

3.2.2 Challenges and Solutions

Challenge	Solution
Role-based redirection after login	Implemented custom routing logic using React Router and Supa basesession
Designing a professional yet simple UI	Used MUI components with Tailwind for custom styling
Integration with Supa base auth and database	Carefully followed Supabase documentation; created helper functions
Syncing frontend state with backend data	Used use Effect () and use State () to manage real-time state updates
UI issues on smaller screens	Performed manual responsive testing and adjusted grid breakpoints

Table 01

3.2.3 Lessons Learned

This project provided a wealth of hands-on experience and revealed many essential insights that go beyond what is typically taught in academic settings. As a backend developer, I was able to deepen both my technical and professional skill sets in meaningful ways.

• Practical Experience with Real-World Backend Systems

Working on a live project gave me exposure to the full development cycle of backend services— from setting up database schemas to deploying authentication workflows— enhancing my understanding of scalable architecture and production-level coding standards.

• Supabase Mastery

I gained significant experience in utilizing Supabase as a Backend-as-a-Service (BaaS). This included:

o Configuring PostgreSQL tables and relationships

o Managing row-level security policies

o Implementing role-based authentication

o Leveraging Supabase Auth for secure login, sign-up, and session management

o Utilizing Supabase Realtime features for syncing frontend and backend data

• Improved API Integration

I became proficient in creating and consuming RESTful APIs, handling asynchronous data, managing secure headers and tokens, and debugging API call issues—skills critical for modern full-stack development.

• Database Design and Query Optimization

I learned how to design relational database schemas to support features such as event management, user roles, and announcement systems. Writing optimized queries and using filters efficiently also helped in maintaining performance.

• Security Awareness

By working with authentication and data access layers, I developed a better understanding of backend security principles including password hashing, role-based access control, and safe data handling practices.

• Error Handling and Logging

I gained experience in implementing structured error handling and logging mechanisms to diagnose issues in real-time and maintain system reliability.

• Version Control & Collaboration

Through GitHub-based collaboration, I became more disciplined in backend code management— creating isolated feature branches, writing clear commit messages, and using pull requests with code reviews.

• Deployment and Environment Management

I became familiar with configuring environment variables securely and preparing backend services for deployment in cloud environments.

3.3 Conclusion

This industrial project has served as a significant steppingstone in my professional and academic development. The opportunity to work on the I-Team Society Web Application allowed me to bridge the gap between classroom theory and real-world application, providing practical experience in software development under actual project constraints and requirements. Throughout the course of this project, I was actively involved in developing a modern, responsive, and functional web application tailored to the unique needs of a student-led society. The system's implementation—featuring secure authentication, role-based access control, and dynamic event and announcement modules—provided hands-on exposure to building scalable digital solutions. In addition to honing my frontend development skills using React.js and Material UI, I also deepened my understanding of Supabase as a Backend-as-a-Service platform, learning how to integrate real-time databases, manage authentication workflows, and deploy a full-stack application in a structured development environment. Moreover, working in a team setting sharpened my collaboration, problem-solving, and version control practices. I learned the value of iterative development, user feedback, and testing in delivering a high-quality final product. Overall, this project has not only enriched my technical capabilities but also boosted my confidence in contributing to real-world software projects. I believe the experience and knowledge gained through this endeavor will significantly contribute to my future career in the IT industry and help me approach more complex challenges with greater confidence and professionalism.

3.4 Further Work

3.4.1 Technical Enhancement

• Mobile Application Development:

Extending the system into a dedicated mobile app (using React Native or Flutter) to provide offline capabilities and push notifications for instant communication.

• Advanced Security Features:

Incorporating multi-factor authentication (MFA), biometric logins, and encrypted data storage to further strengthen privacy and protect sensitive member information.

• Cloud Scalability:

Migrating backend services to microservices or serverless architecture for better performance under heavy traffic and future-proofing the application as membership grows.

• Integration with University Systems:

Linking the platform with existing student portals, Learning Management Systems (LMS), or email servers to streamline access and provide a unified digital ecosystem.

• Automated Payment Gateways:

Enabling integration with online payment systems (Stripe, PayPal, or local payment providers) to handle membership fees and event registrations securely and efficiently.

• Analytics & Dashboards:

Adding data visualization tools to track participation, event success rates, and financial summaries, empowering administrators with actionable insights.

• Testing & CI/CD Improvements:

Enhancing automated testing, error logging, and continuous integration pipelines for improved system reliability and maintainability.

3.4.2 User Experience Expansion

• Personalized Dashboards:

Providing role-specific dashboards (student, staff, admin) with quick access to relevant actions, summaries, and notifications.

• Community Engagement Features:

Adding interactive modules such as chat forums, discussion boards, or Q&A sections to encourage collaboration among members.

• Gamification Elements:

Introducing badges, participation scores, or leaderboards to motivate students and increase engagement in society activities.

• Multilingual Support:

Offering content in Sinhala, Tamil, and English to make the platform inclusive and accessible to the wider university community.

• Accessibility Enhancements:

Improving compliance with WCAG (Web Content Accessibility Guidelines) to support students with disabilities.

• AI-Powered Recommendations:

Using machine learning to suggest events, activities, or resources to members based on their interests and past participation.

• Enhanced Notification System: Expanding the notification system to include SMS alerts and mobile push notifications for time-sensitive announcements.

• Event Media Galleries:

Allowing photo and video sharing from past events to create a richer, more engaging digital archive for society.

3.5 References

• Supabase Documentation: https://supabase.com/docs

• React Official Documentation: https://react.dev/

• TypeScript Official Documentation: https://www.typescriptlang.org/docs/

• Tailwind CSS Documentation: https://tailwindcss.com/docs

• Material UI (MUI) Documentation: https://mui.com/

• PostgreSQL Official Documentation: https://www.postgresql.org/docs/

• Node.js Official Documentation: https://nodejs.org/en/docs

• GitHub Guides: https://docs.github.com/

• Figma Design Resources: https://www.figma.com/resources/

• MDN Web Docs (JavaScript, HTML, CSS): https://developer.mozilla.org/ [27]

• W3C Web Accessibility Initiative (WAI): https://www.w3.org/WAI/

• Agile Alliance – Principles of Agile Development: https://www.agilealliance.org/agile101/

• Netlify Documentation: https://docs.netlify.com/

• Vercel Documentation: https://vercel.com/docs

• Firebase Documentation (for comparison with Supabase): https://firebase.google.com/docs

• ResearchGate – Academic Papers on Web-Based Systems: https://www.researchgate.net/

Additional Platforms & Tools Used

• YouTube – Tutorial Resources: https://www.youtube.com/

• Google Chrome – Development & Testing Browser: https://www.google.com/chrome/

• Facebook – The Open University I-Team Society Page