



SUMMER INTERNSHIP REPORT

on

Frontend Development and Integration of the Adaptive Coding Platform

Submitted by

Gourav Gangwar

B.Tech in Computer Science Engineering,
VIT Bhopal University, Bhopal

Under the Guidance of

Dr. Prabhu Ramachandran

Principal Investigator, FOSSEE Project
Department of Aerospace Engineering,

Indian Institute of Technology Bombay

August 2026

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Prabhu Ramachandran for providing me with the opportunity to be a part of the FOSSEE Internship Program at IIT Bombay and for supporting the development of open-source educational technologies. This opportunity provided me with valuable exposure to real-world software development and the open-source ecosystem.

I would also like to express my heartfelt gratitude to my mentor, Dr. Kushal Shah, for his continuous guidance, valuable feedback, encouragement, and support throughout the internship. His guidance helped me understand project requirements, overcome technical challenges, and improve my contributions to the project.

I am also thankful to my fellow interns and team members for their cooperation, discussions, and support throughout the internship. Working with others in a collaborative environment provided valuable learning experiences and helped me develop a better understanding of professional software development practices.

Finally, I extend my sincere gratitude to everyone associated with the FOSSEE Project at IIT Bombay for providing an environment that promotes open-source development, innovation, collaboration, and continuous learning. This internship has been a valuable experience that strengthened my technical skills, problem-solving abilities, and professional understanding, while motivating me to continue contributing to the open-source community.

Gourav Gangwar
VIT Bhopal University, Bhopal

DECLARATION

This internship has been a highly rewarding learning experience that provided me with the opportunity to contribute to the development of the Adaptive Coding Platform under the FOSSEE Project at IIT Bombay. During the internship, I gained practical experience in frontend development, backend integration, user interface design, accessibility improvements, debugging, and collaborative open-source software development.

The experience provided valuable exposure to professional software development practices and real-world challenges involved in developing and integrating a web-based educational platform. The technical knowledge, problem-solving skills, and hands-on experience gained throughout the internship will serve as a strong foundation for my future academic projects and career objectives.

I would like to extend my sincere appreciation to the FOSSEE Team for their coordination, guidance, and encouragement throughout the internship. The collaborative environment and support provided during the program contributed significantly to my learning and enabled me to complete my assigned responsibilities effectively.

Gourav Gangwar
VIT Bhopal University, Bhopal

INDEX

ACKNOWLEDGEMENT	2
DECLARATION	3
INDEX	4
Chapter 1.....	6
Introduction.....	6
1.1 Project Overview	6
1.2 Internship Background	7
1.3 Problem Statement	7
1.4 Objectives of the Internship	8
1.5 Scope of Work	8
1.6 Report Organization	8
Chapter 2.....	9
ORGANIZATION AND PROJECT OVERVIEW	10
2.1 Introduction to FOSSEE.....	10
2.2 Adaptive Coding Platform (Yaksh).....	10
2.3 Introduction to the Adaptive Coding Platform	11
2.4 Chapter Summary.....	12
Chapter 3.....	13
TECHNOLOGIES AND FRAMEWORKS USED	13
3.1 Frontend Technology Stack	13
3.2 Development Tools and Collaboration	14
3.3 Accessibility and User Experience	15
3.4 Chapter Summary.....	15
Chapter 4.....	16
FRONTEND DEVELOPMENT AND USER INTERFACE IMPLEMENTATION	16
4.1 Frontend Architecture	16
4.2 Development of the Complete Frontend	17
4.3 Backend Integration	18
4.4 Debugging and Critical Issue Resolution.....	18
4.5 User Interface and Accessibility Improvements	19
4.6 Chapter Summary.....	19
Chapter 5.....	20
BACKEND INTEGRATION AND SYSTEM FUNCTIONALITY.....	20
5.1 REST API Integration	20

5.2 Authentication and User Management	20
5.3 Dynamic Data Communication	21
5.4 Critical Functional Issues Identified.....	21
5.5 Testing and Validation	21
5.6 Chapter Summary.....	22
Chapter 6.....	23
USER INTERFACE ENHANCEMENT, ACCESSIBILITY AND APPLICATION OPTIMIZATION	23
6.1 User Interface Refinement	23
6.2 Accessibility Improvements	23
6.3 Responsive Design.....	24
6.4 Application Optimization and Bug Fixes	24
6.5 Manual Testing and Validation	25
6.6 Chapter Summary.....	25
Chapter 7.....	26
RESULTS, CHALLENGES AND FUTURE SCOPE	26
7.1 Technical Challenges and Solutions	26
7.2 Results and Project Impact	27
7.3 Future Scope	27
7.4 Chapter Summary.....	28
Chapter 8.....	29
CONCLUSION	29
Chapter 9.....	30
CONCLUSION AND LEARNING OUTCOMES	30
Learning Outcomes	31

Chapter 1

Introduction

The **Adaptive Coding Platform (Yaksh)** is an open-source, web-based programming education and assessment platform developed under the **FOSSEE (Free/Libre and Open Source Software for Education)** project at the **Indian Institute of Technology Bombay (IIT Bombay)**. It enables students, educators, and institutions to conduct programming courses, coding assignments, laboratory exercises, and coding assessments through a unified online environment. By providing an integrated coding workspace, automated evaluation, user management, and course administration features, the platform simplifies the process of teaching, learning, and assessing programming skills without requiring complex local software installations.

The platform provides a modern web interface where instructors can create courses, manage programming assignments, organize assessments, and monitor student progress, while students can attempt coding problems through an interactive programming workspace and submit solutions for automated evaluation. It is particularly suitable for educational institutions, programming laboratories, coding competitions, and remote learning environments where scalable and accessible programming education is essential. The platform combines role-based access control, automated code evaluation, and centralized course management to create an efficient ecosystem for programming education.

By integrating modern web technologies with an intuitive user interface and open-source development practices, the Adaptive Coding Platform supports FOSSEE's objective of promoting accessible, collaborative, and high-quality software for education. It provides students with practical programming experience while enabling educators to efficiently manage coding-based learning activities through a secure and scalable web platform.

1.1 Project Overview

The **Free/Libre and Open Source Software for Education (FOSSEE)** project, initiated at the **Indian Institute of Technology Bombay (IIT Bombay)**, is a nationally recognized initiative that promotes the adoption, development, and dissemination of Free and Open Source Software (FOSS) across educational institutions in India. The project aims to reduce dependence on proprietary software by providing open-source alternatives for education, engineering, scientific computing, and software development while encouraging collaborative learning and community-driven innovation.

Among the various initiatives under the FOSSEE project, the **Adaptive Coding Platform (Yaksh)** is an open-source learning and assessment platform developed to simplify programming education and coding-based evaluation. The platform enables instructors to create and manage programming courses, organize coding assessments, evaluate submissions automatically, and monitor learner progress through a web-based interface. Students can access coding assignments, write and test solutions

within an integrated coding environment, and submit their programs for evaluation. By combining course management, programming workspaces, automated assessment, and user administration into a single platform, Yaksh serves as an effective solution for academic institutions, programming laboratories, and self-paced programming education.

1.2 Internship Background

During my internship under the FOSSEE project, I contributed to the development of the **complete frontend** of the Adaptive Coding Platform using modern web technologies. The frontend was developed from the ground up, beginning with the implementation of the core user interface and progressively integrating all major modules required by the platform. This included developing responsive pages, implementing reusable React components, integrating backend APIs, and ensuring seamless interaction between different parts of the application.

As development progressed, my responsibilities expanded beyond frontend implementation to include debugging, system integration, accessibility improvements, and application stabilization. I worked on resolving critical functional issues that affected the usability of the platform, improving the overall user experience, refining the interface for better accessibility, and ensuring smooth communication between the frontend and backend services. Throughout the internship, I collaborated closely with mentors and fellow interns while following professional software development practices, including Git-based version control, code reviews, debugging, manual testing, and iterative feature development within an open-source environment.

1.3 Problem Statement

Although the backend architecture of the Adaptive Coding Platform provided the foundation for programming assessment, the application required a complete frontend implementation to transform the backend services into a functional and user-friendly web application. In addition to frontend development, several functional and architectural issues were identified during integration that prevented the platform from operating reliably.

The primary challenges encountered during the internship included:

1. Developing the complete frontend architecture and user interface from scratch while maintaining modularity, scalability, and responsiveness.
2. Integrating frontend modules with backend APIs to ensure reliable communication and seamless data exchange throughout the application.
3. Resolving critical functional blockers, including the absence of question seeding, which prevented programming questions from being available within the system.
4. Addressing the lack of an API endpoint for retrieving question content, preventing the frontend from displaying programming questions to users.
5. Identifying the risk associated with the database initialization script (`init_db.py`), which recreated the database by dropping existing tables before

initialization, posing a potential risk to production data if executed without appropriate safeguards.

6. Improving accessibility, visual consistency, responsiveness, and overall user experience across different modules of the application.

1.4 Objectives of the Internship

The primary objectives of the internship were:

- Develop the complete frontend of the Adaptive Coding Platform using React and modern web development practices.
- Design and implement responsive and reusable user interface components.
- Integrate frontend modules with backend APIs to provide complete application functionality.
- Implement dashboards, authentication screens, coding workspace, and other application interfaces.
- Identify, analyze, and resolve critical functional blockers affecting the platform.
- Improve application accessibility through enhanced color contrast, color-blind-friendly design, and better visual feedback.
- Perform debugging, manual testing, and frontend optimization to improve application stability and usability.
- Contribute to the development of a large-scale open-source educational platform while following collaborative software engineering practices.

1.5 Scope of Work

The scope of this internship extended beyond frontend interface development and involved building a complete web application interface that interacts seamlessly with backend services. The work included designing and implementing the frontend architecture, developing reusable React components, integrating REST APIs, handling authentication workflows, managing application state, debugging frontend issues, and validating application functionality through manual testing.

The internship also involved identifying and resolving high-priority functional issues that affected the overall operation of the platform. Accessibility improvements were implemented by enhancing color contrast, supporting color-blind-friendly design principles, improving interface consistency, and refining the user experience across different modules. The project required continuous collaboration with mentors and fellow developers while following version control, issue tracking, and modern software development workflows to ensure maintainability and code quality.

1.6 Report Organization

This report documents the work carried out during the internship and is organized into the following chapters:

- **Chapter 2** presents the organizational background of the FOSSEE project and provides an overview of the Adaptive Coding Platform.
- **Chapter 3** describes the technologies, frameworks, development tools, and software engineering practices used throughout the internship.
- **Chapter 4** discusses the frontend architecture, user interface development, and implementation of major application modules.
- **Chapter 5** explains the backend integration process, authentication workflow, API communication, and application data flow.
- **Chapter 6** details the accessibility improvements, UI refinements, responsiveness enhancements, and user experience optimizations implemented during development.
- **Chapter 7** presents the debugging process, testing methodology, critical issues resolved, and application stabilization efforts.
- **Chapter 8** summarizes the technical challenges encountered, solutions implemented, project outcomes, and future scope of development.
- **Chapter 9** concludes the report by presenting the internship learning outcomes, professional skills acquired, and overall experience gained during \

Chapter 2

ORGANIZATION AND PROJECT OVERVIEW

2.1 Introduction to FOSSEE

The **Free/Libre and Open Source Software for Education (FOSSEE)** project is a flagship initiative funded by the **Ministry of Education, Government of India**, and coordinated by the **Indian Institute of Technology (IIT) Bombay**. The primary objective of FOSSEE is to improve the quality of education by promoting the adoption and development of **Free and Open Source Software (FOSS)** across educational institutions in India. By encouraging the use of open-source technologies, the project aims to reduce dependence on proprietary software while making high-quality educational tools freely accessible to students, educators, and researchers.

FOSSEE contributes to education by developing open-source software, creating technical documentation, preparing learning resources, organizing workshops, and encouraging students to contribute to real-world open-source projects. These initiatives provide practical exposure to collaborative software development while strengthening the open-source ecosystem in engineering, science, and technology education.

The project encompasses multiple domains, including programming education, scientific computing, electronics, mathematics, computational engineering, and software development. Through collaboration between academic institutions, industry professionals, mentors, and student contributors, FOSSEE continues to promote innovation, knowledge sharing, and accessible education throughout the country.

2.2 Adaptive Coding Platform (Yaksh)

The **Adaptive Coding Platform (Yaksh)** is one of the educational software platforms developed under the FOSSEE project to support programming education and coding-based assessment. The platform provides a web-based environment that enables educators to create programming assignments, conduct coding examinations, evaluate submissions automatically, and monitor student performance through a centralized interface.

Unlike conventional programming laboratories that often require extensive software installation and configuration, Yaksh offers an integrated browser-based environment where students can access programming assignments, write code, submit solutions, and receive automated evaluation. This simplifies the learning process while reducing administrative overhead for educational institutions.

The platform is designed to support classroom teaching, programming laboratories, coding competitions, and remote learning by providing a scalable and accessible solution for programming education.

The primary objectives of the Adaptive Coding Platform include:

- Providing an integrated environment for programming education and assessment.

- Supporting automated evaluation of programming assignments.
- Simplifying course, assignment, and user management for instructors.
- Offering students an intuitive coding workspace for solving programming problems.
- Promoting open-source software adoption in academic institutions through an accessible web-based platform.

2.3 Introduction to the Adaptive Coding Platform

The Adaptive Coding Platform is a modern web application that combines course management, programming assessment, user administration, and automated code evaluation into a single integrated system. The platform follows a client-server architecture in which the frontend provides an interactive user experience while the backend manages authentication, data storage, assignment management, and code evaluation.

During this internship, the frontend of the platform was developed and enhanced using modern web technologies. The application provides responsive interfaces for different user roles while communicating with backend services through REST APIs to retrieve, display, and manage application data.

Some of the major features of the platform include:

- **Role-Based User Management:** Separate interfaces and permissions for administrators, instructors, and students.
- **Programming Workspace:** An integrated environment where students can attempt programming assignments and submit solutions.
- **Course and Assignment Management:** Facilities for creating, organizing, and managing programming courses and assessments.
- **Automated Evaluation:** Backend integration for evaluating programming submissions and displaying results.
- **Responsive User Interface:** A modern frontend designed to provide a consistent experience across desktop and mobile devices.
- **Accessibility Enhancements:** Improved color contrast, color-blind-friendly visual design, and enhanced usability for a broader range of users.

Overall System Workflow of the Adaptive Coding Platform

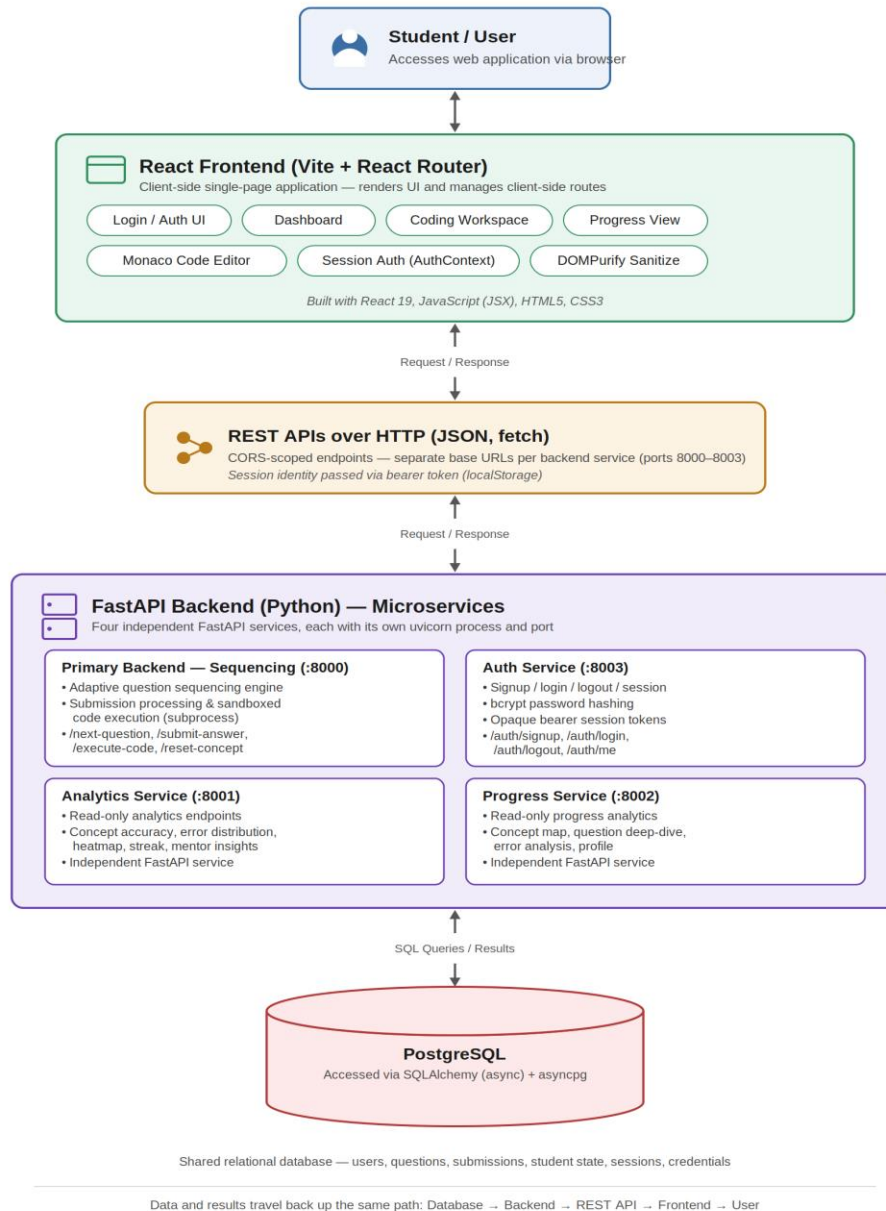


Figure 2.1 – Overall architecture of the Adaptive Coding Platform.

Figure 2.1 – Overall Workflow of the Adaptive Coding Platform

2.4 Chapter Summary

This chapter presented an overview of the **FOSSEE (Free/Libre and Open Source Software for Education)** project and its contribution to promoting open-source software in education. It also introduced the **Adaptive Coding Platform (Yaksh)** as a web-based programming education and assessment system developed under the FOSSEE initiative. Finally, the chapter described the platform's objectives, architecture, and major features, providing the foundation for understanding the technical work carried out during the internship.

Chapter 3

TECHNOLOGIES AND FRAMEWORKS USED

The development of the **Adaptive Coding Platform (Yaksh)** required a modern web technology stack capable of building responsive user interfaces, integrating with backend services, and providing a seamless experience for different categories of users. During the internship, the frontend application was developed using contemporary web technologies and open-source development tools that supported modular development, efficient state management, API integration, and collaborative software engineering.

The selected technology stack enabled rapid frontend development, reusable component architecture, responsive layouts, and smooth communication between the client and server. Version control systems and collaborative workflows further ensured that development could be carried out efficiently within the FOSSEE open-source environment.

3.1 Frontend Technology Stack

- **HTML5**

HTML5 was used to provide the structural foundation of the application. Semantic HTML elements were utilized to improve readability, maintainability, and accessibility while creating responsive web pages for different modules of the platform.

- **CSS3**

CSS3 was used for designing the overall appearance of the application, including layouts, typography, spacing, animations, and responsive behavior. Modern CSS techniques such as **Flexbox** and **CSS Grid** were implemented to create adaptive interfaces that function consistently across different screen sizes and devices.

- **JavaScript (ES6+)**

JavaScript served as the primary programming language for implementing client-side functionality. Modern ES6 features including arrow functions, modules, destructuring, asynchronous programming using **Promises** and **async/await**, and object-oriented programming concepts were extensively used throughout the frontend application.

- **React.js**

React.js was used as the primary frontend framework for building the complete user interface of the Adaptive Coding Platform. The application was developed using reusable components, allowing different modules to share common functionality while maintaining modularity and scalability. React Hooks such as **useState**, **useEffect**, and other built-in features were utilized to manage component state, lifecycle events, and dynamic rendering.

- **React Router**

React Router was used to implement client-side routing, enabling seamless navigation between authentication pages, dashboards, workspace modules, course management pages, and other application screens without requiring full page reloads.

- **REST API Integration**

The frontend communicates with backend services using REST APIs to retrieve and update application data. API integration was implemented to support user authentication, course management, assignment handling, question retrieval, and other platform functionalities while ensuring reliable communication between the client and server.

- **Axios**

Axios was used as the HTTP client for sending API requests and receiving responses from the backend services. It simplified asynchronous communication, request handling, response processing, and error management throughout the application.

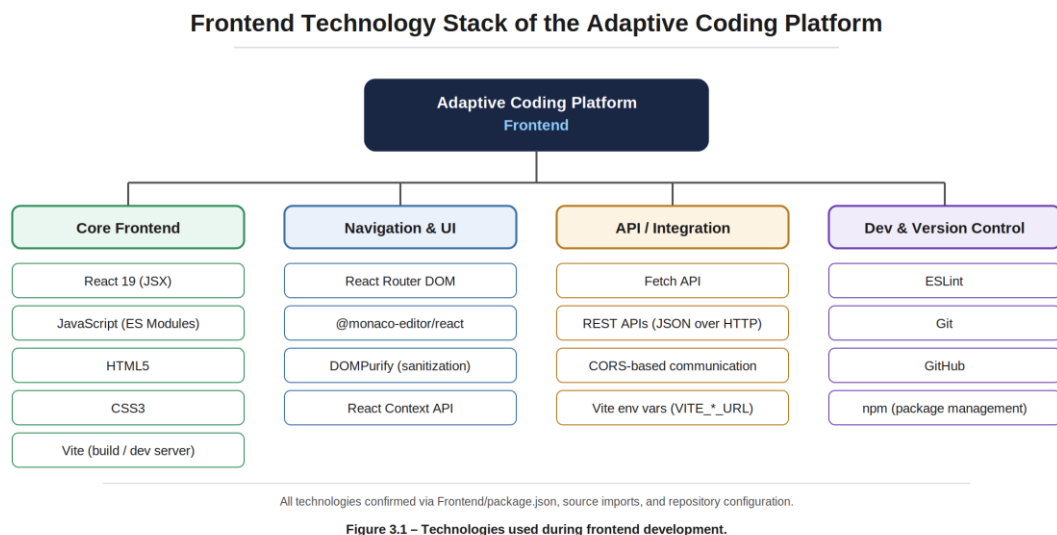


Figure 3.1 – Frontend Technology Stack Used in the Project

3.2 Development Tools and Collaboration

Working within a large open-source project required effective collaboration, version control, and organized development practices.

- **Git**

Git was used as the distributed version control system for managing source code throughout the internship. It enabled tracking code changes, maintaining project history, resolving merge conflicts, and supporting collaborative development through feature branches.

- **GitHub**

GitHub served as the central repository for project collaboration. It was used to maintain the project codebase, share updates, review changes, manage branches, and coordinate development activities among mentors and team members.

- **Visual Studio Code**

Visual Studio Code was used as the primary development environment for writing, debugging, and maintaining the frontend source code. Its integrated debugging tools, extension ecosystem, and Git integration improved development efficiency.

- **Manual Testing and Debugging**

Throughout development, manual testing was performed to verify frontend functionality, validate API integration, identify user interface issues, and ensure that different modules operated correctly. Browser Developer Tools were extensively used to inspect network requests, debug JavaScript code, and analyze application behavior during development.

3.3 Accessibility and User Experience

An important aspect of the internship involved improving the accessibility and usability of the Adaptive Coding Platform.

The frontend interface was enhanced by improving color contrast for both light and dark themes, making the application more suitable for users with color vision deficiencies. Interface consistency was improved by refining layouts, visual indicators, icons, and overall user interaction patterns. Responsive design principles were followed to ensure that the application remained functional across different screen sizes while providing a consistent user experience.

3.4 Chapter Summary

This chapter presented the technologies, frameworks, and development tools used during the internship. It described the use of **React.js**, **HTML5**, **CSS3**, and **JavaScript** for frontend development, along with **REST APIs** and **Axios** for backend integration. The chapter also highlighted the role of **Git**, **GitHub**, and collaborative development practices in managing the project, as well as the accessibility improvements and user experience enhancements implemented throughout the development process.

Chapter 4

FRONTEND DEVELOPMENT AND USER INTERFACE IMPLEMENTATION

The primary responsibility assigned during the internship was the development of the complete frontend of the **Adaptive Coding Platform (Yaksh)**. While the backend services provided the core functionality for authentication, course management, programming assignments, and automated evaluation, the platform required a modern, responsive, and user-friendly interface that could effectively communicate with these services.

The frontend was developed from the ground up using **React.js** and modern web development practices. The application was designed using a modular component-based architecture to improve maintainability, scalability, and code reusability. Throughout the development process, emphasis was placed on creating responsive layouts, seamless navigation, efficient API integration, and an intuitive user experience suitable for students, instructors, and administrators.

4.1 Frontend Architecture

The frontend follows a component-based architecture in which the application is divided into reusable modules responsible for different functionalities. This architecture improves maintainability by allowing independent development and testing of individual components while reducing code duplication.

React Router was used to manage client-side navigation, enabling users to move between different pages without requiring complete page reloads. Each module communicates with backend services through REST APIs, ensuring that application data remains synchronized with the server.

The overall frontend architecture consists of:

- Authentication module
- Dashboard interface
- Programming workspace
- Assignment and course management
- User profile and administration pages
- Shared reusable UI components
- API communication layer

This modular design allows future enhancements to be implemented with minimal impact on existing functionality.

4.2 Development of the Complete Frontend

One of the major contributions during the internship was the development of the complete frontend interface of the Adaptive Coding Platform. The user interface was built from scratch by translating project requirements into functional React components integrated with backend services.

Several major modules were developed during this process, including authentication screens, dashboards, programming workspace interfaces, navigation components, and administrative pages. Particular attention was given to maintaining a consistent design language throughout the application by using reusable components and standardized layouts.

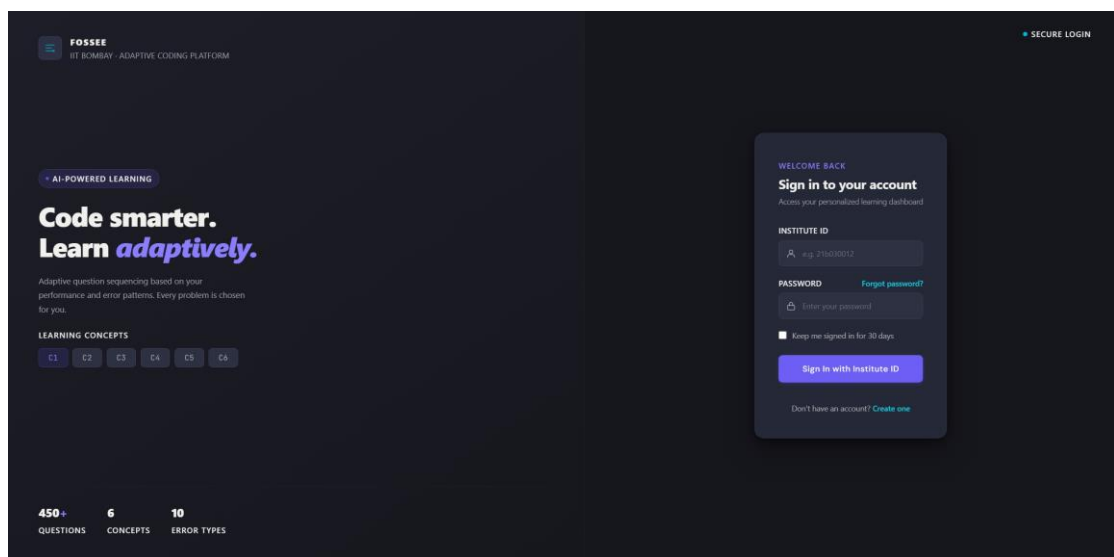


Figure 4.1 – Login Interface of the Adaptive Coding Platform

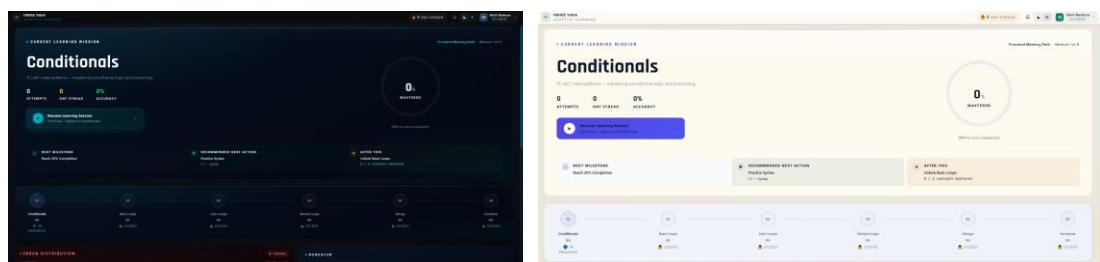


Figure 4.2 – Adaptive Coding Platform Dashboard in Dark and Light Modes

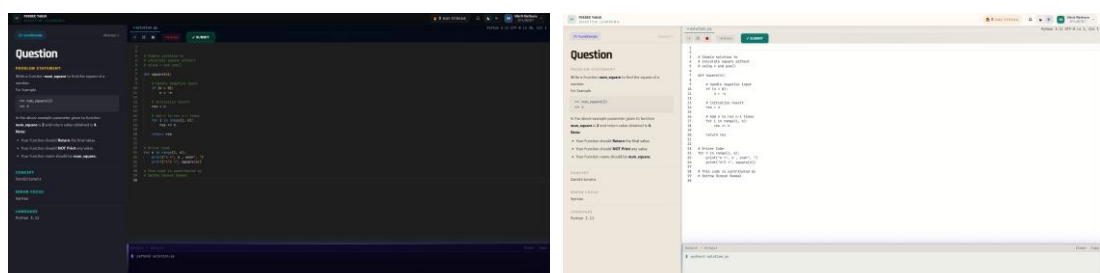


Figure 4.3 – Programming Workspace in Dark and Light Modes

The frontend implementation focused on:

- Building responsive user interfaces for different user roles.
- Creating reusable React components to reduce development time.
- Maintaining consistency across different pages.
- Ensuring smooth navigation throughout the platform.
- Implementing clean layouts for improved usability.
- Supporting desktop and mobile screen resolutions.

The completed frontend provides an organized environment where users can efficiently interact with different platform features while maintaining a consistent user experience.

4.3 Backend Integration

A significant part of the internship involved integrating the frontend with backend REST APIs. The frontend communicates with backend services to retrieve, display, update, and manage application data dynamically.

API integration was implemented for multiple platform functionalities, including user authentication, dashboard information, programming assignments, question management, and user-specific data. Axios was used to send HTTP requests and process server responses while handling asynchronous operations efficiently.

Proper error handling mechanisms were incorporated to manage communication failures and unexpected responses gracefully. This ensured that users received meaningful feedback whenever network or server-related issues occurred.

4.4 Debugging and Critical Issue Resolution

As development progressed, several functional issues affecting the operation of the platform were identified and resolved.

Some of the major issues addressed during the internship included:

- **Frontend Integration Issues:** Multiple frontend modules were corrected to ensure proper communication with backend APIs and accurate rendering of application data.
- **Question Seeding Issue:** A critical issue was identified where programming questions were not being seeded into the database. Without question seeding, the platform could not provide coding problems to users, preventing the system from functioning as intended.
- **Missing Question Retrieval Endpoint:** During integration, it was observed that the backend lacked an API endpoint for retrieving question content required by the frontend. This prevented programming questions from being displayed within the coding workspace. The issue was documented and communicated for backend implementation.
- **Database Initialization Risk:** The database initialization script (`init_db.py`) recreated the database by dropping existing tables before initialization. This behavior was identified as potentially dangerous because executing the script

on an existing deployment could result in complete loss of application data. The issue was reported as a critical architectural concern.

- **Application Stability:** Various frontend issues identified during testing were resolved to improve application reliability, consistency, and overall user experience.

Resolving these issues significantly improved the stability of the platform and enabled successful frontend-backend integration.

4.5 User Interface and Accessibility Improvements

Beyond frontend implementation, additional effort was dedicated to improving the overall usability and accessibility of the platform.

Several enhancements were introduced to create a more inclusive and visually consistent interface, including:

- Improving color contrast for both light and dark themes.
- Applying color-blind-friendly design principles wherever applicable.
- Refining layouts to improve readability.
- Adding appropriate visual indicators and icons to improve usability.
- Maintaining consistency across different pages and components.
- Improving responsiveness for various screen sizes.

These improvements enhanced the overall user experience while making the platform more accessible to a broader range of users.

4.6 Chapter Summary

This chapter described the development of the complete frontend of the Adaptive Coding Platform during the internship. It discussed the frontend architecture, implementation of major user interface modules, integration with backend services, and the resolution of critical functional issues affecting platform operation. The chapter also highlighted the accessibility improvements and user interface enhancements that contributed to a more reliable, responsive, and user-friendly programming education platform.

Chapter 5

BACKEND INTEGRATION AND SYSTEM FUNCTIONALITY

After completing the development of the frontend interface, the next major phase of the internship involved integrating the application with backend services. Although the backend already contained the core business logic, successful operation of the Adaptive Coding Platform depended on establishing reliable communication between the frontend and backend. This integration enabled the application to authenticate users, retrieve programming assignments, display dynamic content, and manage user interactions in real time.

The integration process required careful coordination between frontend components and backend APIs to ensure accurate data exchange, efficient error handling, and consistent application behavior. During this phase, several functional issues affecting system operation were identified, analyzed, and resolved to improve the overall reliability of the platform.

5.1 REST API Integration

The Adaptive Coding Platform follows a client-server architecture in which the React frontend communicates with backend services through RESTful APIs. Every user interaction requiring server-side processing is performed by sending HTTP requests and processing the corresponding responses.

The frontend was integrated with backend services to perform operations such as:

- User authentication and session management.
- Retrieving dashboard information.
- Accessing programming assignments.
- Displaying coding questions.
- Managing user-specific information.
- Updating application data based on user actions.

API requests were implemented using Axios, enabling asynchronous communication between the client and server while maintaining a responsive user interface.

5.2 Authentication and User Management

Authentication forms one of the fundamental components of the Adaptive Coding Platform. During integration, the frontend was connected with backend authentication services to validate user credentials and control access to different modules of the application.

Role-based access mechanisms ensured that users were presented with interfaces appropriate to their responsibilities. Administrators, instructors, and students accessed

different sections of the application according to their permissions, providing a secure and organized workflow throughout the platform.

Proper validation and error handling mechanisms were implemented to improve the login experience and provide meaningful feedback whenever authentication requests failed.

5.3 Dynamic Data Communication

The frontend was designed to display dynamic information retrieved directly from backend services instead of relying on static data.

Whenever users navigated through different sections of the application, API requests were generated to obtain the latest available information. The received responses were processed and rendered within the user interface, ensuring that assignments, questions, and user-specific information remained synchronized with the backend database.

This dynamic communication architecture improves scalability by separating presentation logic from backend business logic while allowing future system enhancements with minimal frontend modifications.

5.4 Critical Functional Issues Identified

During frontend-backend integration, several critical functional issues were identified that significantly affected the operation of the Adaptive Coding Platform.

One of the most significant issues was the absence of **question seeding** within the database. Since programming questions were not being populated into the system, users could not access coding problems, preventing the platform from performing its primary educational function.

Another major issue involved the absence of a dedicated backend endpoint responsible for retrieving question content. Although the frontend interface was capable of displaying programming questions, the required API endpoint was unavailable, preventing successful data retrieval. This issue represented an important gap in backend functionality and was reported during development.

A further concern was identified within the database initialization process. The `init_db.py` script recreated the database by dropping all existing tables before initialization. While appropriate during initial development, executing this script on an existing deployment could permanently remove application data. This behavior was recognized as a critical architectural concern requiring careful handling before production deployment.

Identifying these issues at an early stage contributed significantly to improving system reliability and preventing future deployment problems.

5.5 Testing and Validation

Following backend integration, extensive manual testing was performed to verify application functionality and ensure successful communication between frontend components and backend services.

Testing activities included:

- Verifying authentication workflows.
- Validating API responses.
- Confirming successful rendering of dynamic content.
- Checking navigation across application modules.
- Identifying integration-related issues.
- Testing error handling under different scenarios.
- Verifying frontend behavior after bug fixes.

These testing activities improved confidence in the stability of the application and ensured that implemented features operated according to project requirements.

5.6 Chapter Summary

This chapter presented the backend integration process carried out during the internship. It discussed REST API communication, authentication mechanisms, dynamic data handling, and the integration of frontend components with backend services. The chapter also highlighted several critical functional issues discovered during development, including missing question seeding, the absence of a question retrieval API, and the database initialization risk associated with `init_db.py`. Finally, it described the testing and validation activities performed to ensure the stability and reliability of the Adaptive Coding Platform.

Chapter 6

USER INTERFACE ENHANCEMENT, ACCESSIBILITY AND APPLICATION OPTIMIZATION

Developing a functional application is only one aspect of modern software engineering. Equally important is ensuring that the application is intuitive, responsive, accessible, and capable of providing a consistent user experience across different devices and user groups. After completing the frontend implementation and backend integration, the next phase of the internship focused on refining the overall user interface, improving accessibility, and optimizing the application for better usability.

These improvements were implemented by analyzing the existing interface, identifying usability issues, and incorporating accessibility principles to make the Adaptive Coding Platform more inclusive and user-friendly.

6.1 User Interface Refinement

The frontend interface was continuously refined to improve visual consistency and simplify user interaction. Several screens were redesigned to provide a cleaner layout, better alignment, and improved navigation throughout the application.

Common interface elements were standardized to maintain consistency across different modules. Buttons, cards, navigation menus, forms, and page layouts were adjusted to provide a uniform design language that enhances both readability and usability.

The major UI improvements included:

- Standardizing page layouts across different modules.
- Improving spacing, alignment, and typography.
- Maintaining consistent styling for buttons, forms, and navigation components.
- Enhancing dashboard presentation for improved readability.
- Refining the coding workspace interface for a cleaner user experience.

These modifications resulted in a more organized and professional user interface while reducing visual complexity for users.

6.2 Accessibility Improvements

One of the significant contributions during the internship involved improving the accessibility of the Adaptive Coding Platform.

Accessibility enhancements were implemented to ensure that users with different visual abilities could comfortably interact with the application. The interface was reviewed to identify areas where accessibility could be improved without affecting the existing design.

The implemented improvements included:

- Improving color contrast in both light and dark themes.
- Applying color-blind-friendly color combinations wherever appropriate.
- Improving the visibility of interface elements through better contrast.
- Using icons and visual indicators to complement color-based information.
- Enhancing readability by maintaining consistent typography and spacing.
- Improving focus visibility and interactive element clarity.

These improvements increased the overall accessibility of the platform and aligned the interface more closely with modern web accessibility practices.

6.3 Responsive Design

Since users may access the platform using devices with different screen sizes, responsive design played an important role in frontend development.

Layouts were designed to adapt dynamically across desktop and mobile devices while preserving functionality and usability. Flexible layout techniques using modern CSS ensured that interface components resized and repositioned appropriately without affecting application performance.

The responsive design improvements included:

- Flexible page layouts.
- Adaptive navigation components.
- Responsive dashboard elements.
- Improved spacing for smaller displays.
- Better content organization across different screen resolutions.

These enhancements ensured that users experienced a consistent interface regardless of the device being used.

6.4 Application Optimization and Bug Fixes

During testing and frontend integration, multiple usability and functional issues were identified and resolved to improve the stability of the application.

Optimization efforts focused on improving interface reliability rather than introducing additional features. Existing components were refined to eliminate inconsistencies, improve rendering behavior, and ensure smoother interaction between frontend modules and backend services.

Several issues affecting application functionality were corrected, including frontend integration errors, interface inconsistencies, and user interaction problems identified during manual testing.

These improvements contributed to a more stable and reliable application while reducing unexpected behavior during normal usage.

6.5 Manual Testing and Validation

Manual testing formed an essential part of the development process. After implementing new features and resolving identified issues, the application was thoroughly tested to verify correct functionality and ensure that recent changes did not introduce additional problems.

Testing activities included:

- Verifying frontend navigation.
- Testing authentication workflows.
- Validating API integration.
- Checking dashboard functionality.
- Testing responsive layouts.
- Verifying accessibility improvements.
- Identifying interface inconsistencies.
- Confirming successful bug fixes across different application modules.

Manual testing helped identify edge cases and ensured that implemented changes behaved as expected before being shared with mentors for review.

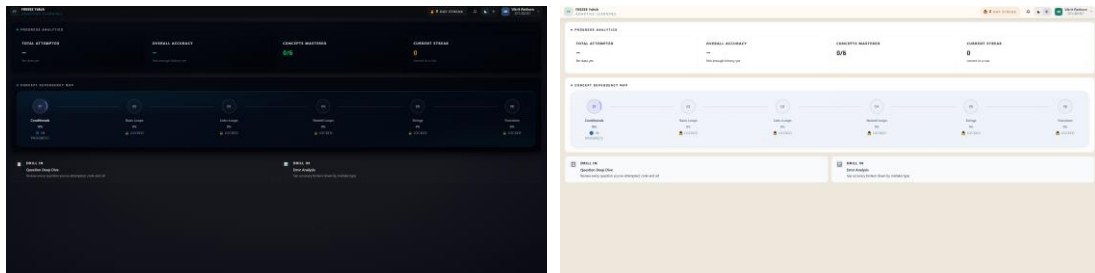


Figure 6.1 – Progress Analytics Interface in Dark and Light Modes

6.6 Chapter Summary

This chapter discussed the improvements made to the user interface, accessibility, responsiveness, and overall usability of the Adaptive Coding Platform. It highlighted the refinement of frontend components, implementation of accessibility enhancements such as improved color contrast and color-blind-friendly design, responsive layout improvements, application optimization, and manual testing activities. Together, these enhancements significantly improved the quality, usability, and reliability of the platform while providing a more accessible programming environment for users.

Chapter 7

RESULTS, CHALLENGES AND FUTURE SCOPE

Throughout the internship, the development of the Adaptive Coding Platform involved addressing several technical and functional challenges while building a complete frontend application and integrating it with the existing backend services. The project progressed from developing user interface components to resolving integration issues, improving accessibility, and stabilizing the application for practical use. The work carried out during the internship contributed significantly to improving the usability, functionality, and overall reliability of the platform.

7.1 Technical Challenges and Solutions

Developing the frontend of a large-scale educational platform required overcoming several challenges related to application architecture, backend integration, and overall system functionality. Many of these issues affected the core operation of the platform and therefore required immediate attention.

The major challenges encountered during the internship included:

Complete Frontend Development

The platform required a fully functional frontend capable of supporting multiple user roles and communicating efficiently with backend services. A modular React-based architecture was implemented to create reusable components, maintain scalability, and simplify future development.

Frontend–Backend Integration

Integrating the frontend with backend APIs required careful handling of asynchronous requests, response validation, and error management. Proper API communication ensured that users could interact seamlessly with different modules of the application.

Question Seeding Issue

A critical issue identified during development was the absence of question seeding within the database. Without seeded programming questions, the platform could not provide coding assignments to users, directly affecting its primary functionality. The issue was identified, documented, and reported for resolution.

Missing Question Retrieval Endpoint

During API integration, it was observed that the backend lacked an endpoint for retrieving programming question content. Although the frontend interface was prepared to display questions, the absence of this endpoint prevented successful data retrieval. This gap was identified during testing and communicated as a high-priority backend requirement.

Database Initialization Risk

Another significant concern involved the database initialization process. The `init_db.py` script recreated the database by dropping existing tables before initialization. While suitable during development, executing the script on an active deployment could result in permanent data loss. This behavior was recognized as a critical architectural risk and reported accordingly.

Accessibility Improvements

The user interface was enhanced by improving color contrast, refining layouts, supporting color-blind-friendly design principles, and introducing clearer visual indicators. These improvements increased the usability of the platform while making it more accessible to a broader range of users.

7.2 Results and Project Impact

The work completed during the internship significantly improved the overall quality of the Adaptive Coding Platform.

The complete frontend implementation transformed backend services into an interactive web application capable of supporting programming education through an intuitive interface. Backend integration enabled dynamic communication between different application modules, while debugging activities resolved several functional issues that affected platform stability.

Accessibility improvements enhanced the visual experience by providing improved readability, better contrast, and more consistent interface behavior. Manual testing and continuous debugging further increased application reliability by identifying and resolving issues before deployment.

The overall outcome of the internship included:

- Successful development of the complete frontend application.
- Integration of frontend components with backend services.
- Improved responsiveness and user experience.
- Enhanced accessibility through improved visual design.
- Identification and reporting of critical backend issues affecting application functionality.
- Improved application stability through debugging and manual testing.

These contributions established a solid foundation for future development while improving the platform's readiness for academic use.

7.3 Future Scope

Although the Adaptive Coding Platform provides a comprehensive environment for programming education, several enhancements can further improve its capabilities.

Potential future improvements include:

- Expanding the programming workspace with additional development features.
- Supporting more programming languages and execution environments.

- Implementing comprehensive automated frontend and integration testing.
- Enhancing analytics for instructors to monitor student performance more effectively.
- Improving mobile responsiveness for smaller devices.
- Strengthening accessibility by incorporating additional Web Content Accessibility Guidelines (WCAG).
- Optimizing application performance for larger datasets and increased user traffic.
- Expanding collaborative development through additional open-source contributions.

These enhancements would further strengthen the platform as a scalable solution for programming education.

7.4 Chapter Summary

This chapter discussed the major technical challenges encountered during the internship, the solutions implemented, and their overall impact on the Adaptive Coding Platform. It highlighted the successful development of the complete frontend, backend integration, identification of critical functional issues, accessibility improvements, and application stabilization. Finally, the chapter outlined potential future enhancements that can further improve the platform's functionality, performance, and educational value.

Chapter 8

CONCLUSION

The internship under the **FOSSEE (Free/Libre and Open Source Software for Education)** project at the **Indian Institute of Technology Bombay (IIT Bombay)** provided an excellent opportunity to apply theoretical knowledge to the development of a real-world open-source software project. Working on the **Adaptive Coding Platform (Yaksh)** enabled me to gain practical experience in modern frontend development, backend integration, debugging, accessibility enhancement, and collaborative software engineering.

A major achievement during the internship was the successful development of the complete frontend of the Adaptive Coding Platform using React.js and related web technologies. The frontend was designed with a modular architecture that supports maintainability, scalability, and responsiveness while providing an intuitive user experience for students, instructors, and administrators. Integrating the frontend with backend services allowed the application to deliver dynamic content and support the platform's core functionalities.

Beyond frontend implementation, the internship involved identifying and resolving several critical functional issues affecting the application. Important observations included the absence of question seeding, the lack of an API endpoint for retrieving question content, and the potential risk associated with the database initialization process. Reporting and addressing these issues contributed to improving the overall stability and reliability of the platform.

Another important aspect of the internship was improving the usability and accessibility of the application. Enhancements such as improved color contrast, color-blind-friendly interface design, responsive layouts, and better visual consistency made the platform more accessible to a wider range of users while improving the overall user experience.

Throughout the internship, I also strengthened my understanding of professional software development practices, including component-based application development, REST API integration, version control using Git and GitHub, debugging techniques, manual testing, collaborative development, and open-source contribution workflows. Working under experienced mentors and collaborating with fellow developers provided valuable insight into large-scale software engineering and team-based project development.

Overall, this internship significantly enhanced both my technical knowledge and professional skills. It provided practical exposure to building and maintaining production-level web applications while reinforcing the importance of clean code, collaboration, accessibility, and systematic problem-solving. The experience has prepared me for future opportunities in software development and has strengthened my confidence in contributing to complex open-source projects and modern web application development.

Chapter 9

CONCLUSION AND LEARNING OUTCOMES

The internship under the **FOSSEE (Free/Libre and Open Source Software for Education)** project at the **Indian Institute of Technology Bombay (IIT Bombay)** provided invaluable practical experience in modern web application development and open-source software engineering. Working on the **Adaptive Coding Platform (Yaksh)** enabled me to apply theoretical concepts acquired during my academic studies to the development of a real-world software system used for programming education and assessment.

One of the most significant achievements of the internship was the successful development of the complete frontend of the Adaptive Coding Platform. Starting from the initial user interface implementation, the project progressed into a fully functional web application integrated with backend services. The experience strengthened my understanding of component-based architecture, responsive web design, REST API integration, state management, and modern frontend development using React.js.

The internship also involved identifying, analyzing, and resolving several critical functional issues that affected the platform. These included frontend integration challenges, the absence of question seeding, the lack of an API endpoint for retrieving question content, and identifying the potential risk associated with the database initialization process. Addressing and documenting these issues improved my analytical thinking, debugging abilities, and understanding of large-scale software systems.

Another important aspect of the internship was enhancing the overall user experience of the platform. Improvements in accessibility, including better color contrast, color-blind-friendly interface design, responsive layouts, and improved visual consistency, reinforced the importance of designing software that is both functional and inclusive. These enhancements contributed to creating a more user-friendly application suitable for a diverse range of users.

Throughout the internship, I gained practical experience in collaborative software development by working with mentors and fellow developers in an open-source environment. Regular code integration, version control using Git and GitHub, manual testing, issue tracking, debugging, and iterative development provided valuable exposure to professional software engineering workflows followed in real-world projects.

The internship significantly strengthened both my technical and professional competencies. In addition to improving my programming and development skills, it enhanced my ability to analyze problems systematically, communicate technical findings effectively, and collaborate within a development team. The experience also increased my confidence in contributing to complex open-source projects and understanding software development from planning to deployment.

Learning Outcomes

The internship helped me achieve the following learning outcomes:

- Developed a strong understanding of React.js and component-based frontend development.
- Learned to integrate frontend applications with backend REST APIs.
- Improved debugging and problem-solving skills through the resolution of real-world software issues.
- Gained practical experience in Git, GitHub, and collaborative software development workflows.
- Enhanced knowledge of responsive web design and modern UI/UX principles.
- Understood the importance of web accessibility and inclusive application design.
- Improved manual testing techniques and application validation practices.
- Acquired experience in contributing to a large-scale open-source educational software project.
- Strengthened communication, teamwork, and professional development skills.

In conclusion, the internship served as an excellent bridge between academic learning and professional software development. It provided practical exposure to building scalable web applications, solving real-world engineering challenges, and contributing to an impactful open-source project. The knowledge, experience, and skills gained during this internship will serve as a strong foundation for my future career in software engineering and full-stack web development.

