
Questify FYP Portal

Submitted by

Ateeq ur Rehman (FA21-BSCS-197)

Muhammad Talha (FA21-BSCS-167)

Session 2021-2025

Supervised by

Ms. Rabia Khan



Department of Computer Science

Lahore Garrison University

Lahore

Questify FYP Portal

A project submitted to the

Department of Computer Science

In

Partial Fulfillment of the Requirements for the

Bachelor's Degree in Computer Science

By

Ateeq ur Rehman

Muhammad Talha

Internal Supervisor

Ms. Rabia Khan

Lecturer

Department of Computer Science

External Examiner

Chairperson

Dr. Arfan Nagra

Assistant Professor

Department of Computer Science

COPYRIGHTS

This is to certify that the project titled “**Questify FYP Portal**” is the genuine work carried out by **Ateeq ur Rehman** (FA21-BSCS-197) and **Muhammad Talha**(FA21-BSCS-167) students of BSCS of Computer Science Department, Lahore Garrison University, Lahore during the academic year 2020-24, in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Ateeq ur Rehman _____

Muhammad Talha _____

DECLARATION

This is to declare that the project entitled “**Questify FYP Portal**” is an original work done by undersigned, inpartial fulfillment of the requirements for the degree “Bachelor of Science in Computer Science” at Computer Science Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Group Members

Ateeq ur Rehman _____

Muhammad Talha _____

Supervisor

Ms. Rabia Khan _____

Date: _____

DEDICATION

This project is dedicated wholeheartedly to each of the students and mentors and academic coordinators, believers in the innovation and collaboration and in the power of streamlined learning. Questify FYP Portal pays tribute to a number of dreamers envisioning a smarter, more connected academic world and to a number of doers who tirelessly work for that vision into reality. Sincere gratitude is extended toward supervisor, several evaluators, and more technical teams. A certain amount of their guidance had shaped this very platform. We appreciate the constant encouragement from our families and friends that made this adventure possible. To conclude, May this very work dedicated to Final Year Project students become a bridge for simplifying their academic adventure as well as for inspiring greater solutions.

ACKNOWLEDGEMENTS

On behalf of the entire team that developed Questify FYP Portal, we express our sincere gratitude to all those who were necessary for the project's success. Ms. Rabia Khan, our respected supervisor, offered guidance indispensable and support continuous throughout each phase of the project. We are particularly appreciative of her, for those mindful suggestions of hers. Her technical expertise did help to shape, as well as her constructive feedback did make, our vision into a functional and impactful academic platform.

We would also like to give acknowledgement to and appreciate all of our team members' dedicated efforts and collaboration, and this project was then made into a reality by all of their commitment, creativity, and also synergy. Each member uniquely possessed skills, and relentlessly determined with that helped us overcome technical and conceptual challenges along the way.

Our sincere thanks go to each of the faculty members and all of the peers who provided useful feedback during the reviews and the presentations. This feedback enabled refinement as well as improved the portal for better serving its users. We built up a product along with those understandings that adds up value for the Final Year Project adventure of several students and addresses actual academic needs.

Above all Questify FYP Portal is not just a project it is a vision for better academic, and this sail would not possible without the group effort, trust and support of all mentioned people. We are proud to share this with academic community.

Table of Contents

ACKNOWLEDGEMENTS	v
Table of Contents	vi
List of Tables	xi
List of Figures.....	xii
List of Abbreviation.....	xiii
ABSTRACT.....	xiv
Chapter 1	2
INTRODUCTION	2
1.1. Overview	2
1.2 Aim and Objective	3
1.3 Gantt Chart	4
1.4. Report Organization.....	4
Chapter 2	6
PROBLEM DEFINITION	6
Chapter 3	7
SOFTWARE REQUIREMENT SPECIFICATION.....	7
Introduction.....	7
3.1 Introduction.....	7
3.1.1 Purpose.....	7
3.1.2 Document Conventions.....	8
3.1.3 Intended Audience and Reading Suggestions	8
3.1.4 Product Scope	9
3.2 Overall Description	9
3.2.1 Product Perspective.....	9

3.2.2 Product Functions	10
3.2.3 User Classes and Characteristics.....	11
3.2.4 Operating Environment.....	12
3.2.5 Design and Implementation Constraints	12
3.2.6 User Documentation	13
3.2.7 Assumptions and Dependencies	13
3.3 External Interface Requirements	14
3.3.1 User Interfaces	14
3.3.2 Hardware Interfaces.....	14
3.3.3 Software Interfaces	14
3.3.4 Communications Gap.....	15
3.4 System Features	15
3.4.1 Submission of proposal.....	15
3.4.2 Assignment to the Supervisor	16
3.4.3 Progress Monitoring.....	17
3.4.4 Administration Management	17
3.5 Other Nonfunctional Requirements.....	18
3.5.1 Performance Requirements	18
3.5.2 Safety Conditions	18
3.5.3 Security Requirements	18
3.5.4 Characteristics of Good Software	19
3.5.5 Business Rules	19
Chapter 4.....	20
METHODOLOGY	20
4.1 Research and Analysis.....	20
4.1.1 Requirements Gathering	20

4.2 System design	20
4.3 Development	21
4.4 Testing.....	21
4.5 Deployment.....	21
4.6 Documentation and maintenance	21
4.7 Tools and technologies	22
4.8 Hardware / software requirements	22
4.8.1 Server needs:	22
4.3.2 User needs:.....	22
Chapter 5	23
DETAILED DESIGN AND ARCHITECTURE	23
5.1 System Architecture.....	23
5.1.1 Architecture Design Approach	23
5.1.2 Architecture Design	23
5.1.3 Subsystem Architecture	24
5.2 Detailed System Design.....	24
5.2.1 Classification.....	24
5.2.2 Definition	25
5.2.3 Responsibilities	25
5.2.4 Constraints	25
5.2.5 Composition.....	25
5.2.6 Uses/Interactions.....	25
5.2.7 Resources	25
5.2.8 Processing	26
5.2.9 Interface/Exports.....	26
5.2.10 Detailed Subsystem Design	26

Chapter 6	37
IMPLEMENTATION AND TESTING	37
6.1 Agile Development	37
6.2 Version Control and Collaboration.....	38
6.3 Frameworks and Libraries	38
6.3.1 Backend Technologies	38
6.3.2 Frontend Technologies.....	38
6.4 Database Management	38
6.5 Testing Strategy	39
6.5.1 Functional Testing	39
6.5.2 Performance Testing	39
6.6 Test Execution	39
6.6.1 End-to-End Testing	40
6.6.2 Security Testing	40
6.7 Test Cases	41
6.7.1 Validate Student Login	41
6.7.2 Validate Proposal Submission	41
6.7.3 Validate Supervisor Review	42
6.7.4 Ensure JWT authentication	43
6.7.5 Verify session expiration after inactivity	43
6.7.6 Verify students can submit proposals.	44
6.7.7 Validate document status updates (Pending, Approved, Rejected).	44
6.7.8 Verify supervisor login with valid credentials.	45
6.7.9 Verify supervisor can approve/reject proposals.	45
6.7.10 Verify supervisor can download student documents.	46
6.7.11 Verify supervisors can submit feedback on uploaded documents.	46

Chapter 7	47
RESULTS AND DISCUSSION	47
7.1 Login Page	47
7.2 Student Dashboard	47
7.3 Project Proposal Submission	48
7.4 Document Upload.....	49
7.5 Progress Tracking.....	49
7.6 Comments & Feedback	50
7.7 Chat with Supervisor	50
7.8 AI Chatbot	51
7.9 Supervisor Dashboard	51
Chapter 8.....	53
CONCLUSION AND FUTURE WORK	53
8.1 Conclusion	53
8.2 Future Work.....	53
8.2.1 Automated supervisor allocation	53
8.2.2 Internal & External Collaboration.....	54
8.2.3 AI-Driven project suggestions	54
8.2.4 Additional enhancements	54

List of Tables

Table 1: Used tools and technology	22
Table 2: Test Case for Student Login	41
Table 3: Test Case for Proposal submission	42
Table 4: Test Case for Supervisor review	43
Table 5: Test Case for ensure JWT authentication	43
Table 6: Verify session expiration	44
Table 7: Verify students can submit proposals	44
Table 8: Verify document status updates	45
Table 9: Verify supervisor login	45
Table 10: Ensure supervisors can approve/reject proposals	45
Table 11: Check can download student documents.	46
Table 12: supervisors can submit feedback on uploaded documents	46

List of Figures

Figure 1: Gantt chart of the project.....	4
Figure 2: Use case of all users	27
Figure 3: I-Architectural Diagram	28
Figure 4: II-Architectural Diagram	28
Figure 5: Entities relations Diagram	29
Figure 6: Activity Diagram	30
Figure 7: Sequence Diagram.....	32
Figure 8: Components Diagram.....	33
Figure 9: State Diagram	35
Figure 10: Class Diagram	36
Figure 11: Agile Model.....	37
Figure 12: Login Page.....	47
Figure 13: Student Dashboard	47
Figure 14: Project Proposal Submission	48
Figure 15: Document Upload Dashboard	49
Figure 16: Progress Tracking Dashboard	49
Figure 17: Feedback of documents	50
Figure 18: Student and supervisor communication	50
Figure 19: Chat with Chatbot.....	51
Figure 20: Supervisor Dashboard	51

List of Abbreviation

AI: Artificial Intelligence

TPC: Third person Controller

UI: User Interfaces

IDE: Integrated development environment

ERD: Entity Relation Diagram

HUD: Heads Up Display

VR: Virtual Reality

NPC: Non-Playable Characters

LOD: Level of Detail

SRS: Software Requirements Specifications

RBAC: Role-Based Access Control

MERN: MongoDB Express React Node

FYP: Final year project

LGU: Lahore Garrison University

ABSTRACT

The current FYP management system of LGU is mainly done manually which results in inefficiency and is time-consuming such as creating too much paper work, communication gap, and lack of transparency. Students may struggle with monitoring their progress and receiving timely feedback, and faculty may struggle to manage multiple projects and maintain consistent communication. Questify FYP portal is an advanced web-based system that facilitates FYP management at one window facility at LGU. It streamlines project proposal submission, supervisor allocation, progress tracking, document upload, and evaluation. Through real-time notifications and a shared calendar for deadlines and meetings, the portal improves collaboration between students and faculty.

The project will be developed in MERN which will guarantee a responsive and scalable solution, minimizing manual paper work. By digitizing FYP workflows the Questify FYP Portal aims to improve efficiency, save resources, and promote academic excellence.

INTRODUCTION

1.1. Overview

The rapid technological change had a great impact on education, which led to the development of different tools that improve educational processes. Among these, Final Year Project (FYP) portals have appeared as vital tools for managing complex academic tasks within universities. These portals centralize crucial tasks such as project proposal submission, supervisor allocation, document sharing, and evaluation, saving time and resources while ensuring transparency. As educational institutions try for excellence, the demand for efficient, customizable, and user-friendly FYP portals has grown in experience, making them essential components of modern academia. Lahore Garrison University, like many other institutions, recognizes the need for such a solution to optimize its academic processes and provide a better experience for both students and faculty.

While there are many FYP management systems available, they often don't fully meet the specific needs of universities. Popular platforms like Project-Board and FYP Manager have limitations. For example, they can be hard to use for new users, they often lack real-time progress tracking, and they may not have good communication tools. Many of these systems also have rigid workflows that don't fit well with different university policies. These issues can slow down the FYP process and make it harder for students and supervisors to work together. To address these problems, we need a more comprehensive and flexible solution.

The Questify FYP Portal is designed to solve the problems of current FYP management systems at Lahore Garrison University. It offers a user-friendly platform with features like easy proposal submission, supervisor allocation, progress tracking, and secure documents. Students can easily monitor their progress, receive feedback, and stay updated with deadlines. Faculty can review proposals, track student performance, and manage multiple projects efficiently. The portal promotes transparency and collaboration between students and supervisors to ensure successful project completion.

The Questify FYP Portal is a new and improved way to manage final-year projects (FYPs) at Lahore Garrison University. It's designed to be better than other platforms, with features that fit the specific needs of our university. It can adapt to our university's rules and processes, keeps all project information in one place, and allows students and supervisors to stay connected with instant notifications and a calendar. It simplifies tasks like submitting proposals, tracking progress, and giving feedback. Plus, it promotes transparency by making the progress and evaluation process visible to everyone. By using this portal, we can make the FYP process smoother, more efficient, and more satisfying for everyone involved, which will significantly improve the academic experience at Lahore Garrison University

1.2 Aim and Objective

The Questify FYP Portal is a new solution designed to manage and streamline the Final Year Project (FYP) management process at Lahore Garrison University. This platform will simplify crucial processes such as project proposal submission, supervisor allocation, progress tracking, document submission, and evaluation, enhancing collaboration and transparency at every project stage.

The proposed solution offers an all-in-one system modified to the university's specific needs, featuring real-time notifications, integrated calendars, supervisor allocation, and transparent evaluation tools. Built using MERN, it ensures solid performance, security, and scalability. The front end will be responsive and user-friendliness, while the back end will manage complex workflows efficiently.

1.3 Gantt Chart

Gantt Chart of the project “Questify FYP Portal” is given below figure 1.

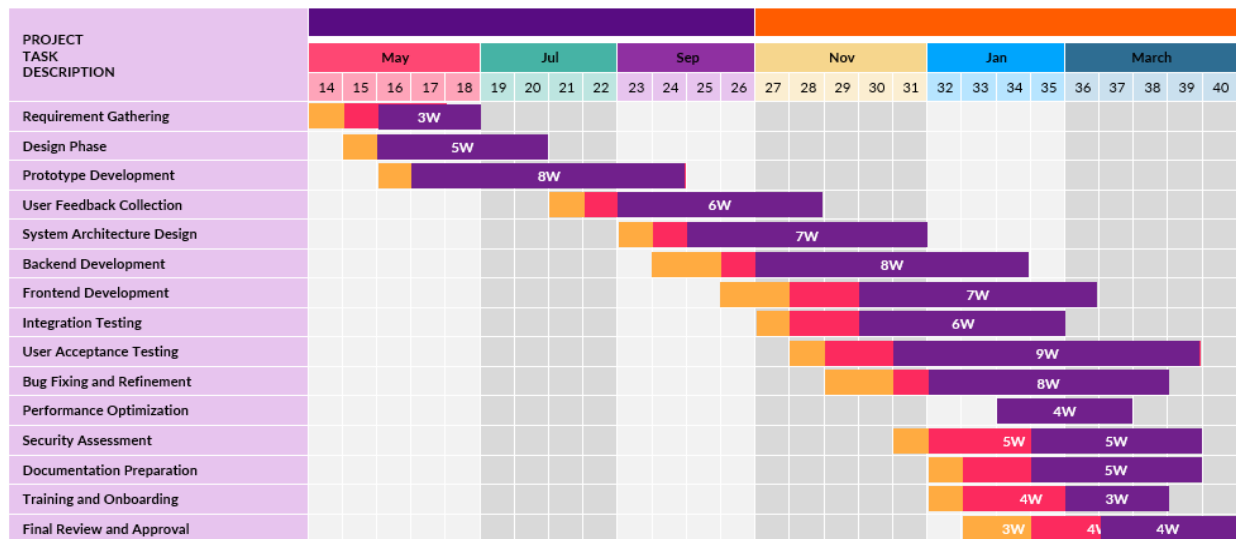


Figure 1: Gantt chart of the project

1.4. Report Organization

This report present Chapter wise organization of project as individual chapter explain tasks which were crucial for the deed of project. List of chapter wise report is given as follow.

Chapter 1: The Questify FYP Portal functions as a single unified platform which optimizes student project together with supervisor management and administrator processes. Traditional systems exposed their weaknesses in fragmented communication and manual processes, prompting the Questify introduction as a MERN stack solution which delivers comprehensive security alongside scalability and role-based user access features.

Chapter 2: The analysis reveals major difficulties with current FYP management practice. The existing system suffers from the combination of limited real-time teamwork, slow manual supervisor allocation processes, unprotected document distribution and inadequate monitoring of project progress. An integrated system which combines security and usability needs development to fill existing gaps.

Chapter 3: The specification includes requirements for proposal submissions as well as progress tracking with rubric-based assessments alongside RBAC features and requirements for speed under 3 seconds The requirements section includes technical tools such as React and Node.js and

MongoDB while specifying hardware requirements of 8GB RAM operating on either Windows or Linux systems.

Chapter 4: Details the Agile-based development process: Managers provide input through stakeholder interviews to define requirements that developers turn into Figma wireframe designs before building separate interface and server components using React frontend and Node.js backend technologies until users approve the finished solution through iterative testing rounds with Postman APIs and user acceptance trials. This section details the deployment process for Linux servers which utilizes PM2 management alongside manual backup protocols.

Chapter 5: Describes the layered architecture: The project utilizes React.js components to build role-optimized dashboards with Node.js/Express.js APIs serving business functions and MongoDB as the data repository while JWT/RBAC authentication maintains system security. The design describes how the proposal management system works alongside progress tracking while detailing how supervisors are selected manually.

Chapter 6: Covers the development phase: Real-time notification implementation through WebSocket and secure Amazon S3 document upload services joined with a supervisor manual assignment capability form part of the design. The text describes implementation testing through Jest units and Postman integrations and OWASP ZAP security analyses for performance verification and usability assessment.

Chapter 7: Presents outcomes: The system achieved all targets by reducing administrative workloads while improving approval speeds and securing highly positive user assessments. The study evaluates system performance for more than 500 concurrent users while comparing the efficiency of Questify to paper-based traditional workflows.

PROBLEM DEFINITION

Managing Final Year Projects (FYPs) in educational institutions is often unsuitable due to dependency on manual systems, resulting in excessive paperwork, communication gaps, and lack of transparency. Students face challenges in tracking progress, receiving timely feedback, and understanding evaluation benchmarks, while faculty struggle with managing multiple projects and maintaining consistent communication. Existing solutions don't have user-friendliness, flexibility to specific policies, and crucial features like real-time notifications and integrated tools, leading to wasted time and reduced project quality. The Questify FYP Portal addresses these issues by offering a centralized, creative platform modified to well-organized workflows, increasing teamwork, and ensuring transparency in FYP management at Lahore Garrison University.

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

The Questify FYP Portal is a unique and collative system that show the working of FYPs for both students and teachers at Lahore Garrison University. FYP is an important task in student's degree, thus effective and smooth management system play important role in it. Questify is solution to which is one solution for all problems which student and faculty faces during project proposal submissions, supervisor allocation, progress tracking and document sharing. The portal fill-up the communication gap between students and supervisor by notifications, shared calendars, and online meeting. The interface is friendly so student can track their working and get feedback from teacher, while supervisor can manage multiple projects of student with ease. Questify aims to let the user feel best experience of FYP portal, saving money, time and resources by being transparent, scalable, and according to LGU's needs.

3.1.1 Purpose

This Software Requirement Specification (SRS) document related to the “Questify FYP Portal” software product, version 1.0. Questify FYP Portal is new, web-based system which is designed for management of Final Year Projects (FYPs) at Lahore Garrison University. The purpose of this system is to make FYPs project management flow faster and digitized, and for the better facilitation, communication and collaboration between students and faculty, making process more trackable to make sure timely completion of all projects.

The scope of this Software Requirement Specification (SRS) covers the entire Questify FYP Portal, providing a clean and clear overview of the important requirement and functionalities needed for its development, deployment, and maintenance. This document focusses on complete system architecture and its features, which have user authentication, proposal submissions, supervisor allocation, work tracking, document management, feedback, suggestions, notifications

and tools for communication. The SRS aims to make the development process meet the needs of LGU, overall creating a big, scalable and user-friendly system that make the FYPs and management process easy for both students and faculty. Software Requirement Specification (SRS) document present both functional and non-functional requirements for the proposed project so that different stakeholders can get complete insight of the project at a quick glance.

3.1.2 Document Conventions

Document convention for the Questify FYP portal Software requirement Specification (SRS) have been made to things clear and consistent in document. The primary font used is Times New Roman, size 12, which made the text readable and make it look professional. Heading are Bold to identify each section clearly, main headings follow heading 1 styles with font size 16 while subheadings follow heading 2 style in Times New Roman font, size 13. Each heading and subheading have its own priority level. Spacing are most important point for the formatting so spacing for Heading 1 before is 6pt and after is 12pt and for subheadings before 6pt and after 12pt, for the paragraphing with line spacing of 1.5. This hierarchy structured help in understanding the importance of each requirement to overall project. By these methods, the SRS help everyone to understand each and everything clearly. It also works as a guide path during development of the project.

3.1.3 Intended Audience and Reading Suggestions

Developer: The main focus are functional and non-functional requirements, including architecture, database design and coding rules.

Users: The users are students, advisor faculty, administrative staff, who will use FYP portal. Users will know that how the portal meet their needs, how it works, how to access and what functionality are present which can help to enhance their work. Start with user requirement and Designs.

Documentation Writers: Task to create user manual and help by getting information from SRS about features, workflows, and usage scenarios. Focus on sections that give clear info of functionalities and user interactions.

Project Managers: Get to know scope, objectives, deliverable, and timelines. The main interest in sections related to management, resource allocation, risk and budget.

Tester: Understand the expected functionalities of the Questify FYP Portal. Main focus on section like test cases, acceptance criteria, and performance.

Stakeholders: Focus on project's outcomes, goals, benefits of implementing the portal.

3.1.4 Product Scope

The Questify is designed to make a big change in the management of Final Year Projects (FYPs) at Lahore Garrison University. This system is to smooth out the all FYPs process, it helps students and faculty work together and deals with common problems. By bringing a common platform for proposal submissions, supervisor allocation, progress tracking, and document sharing, Questify also help in communication between students and supervisors. The initial goal is making a user-friendly platform that is transparent and organized. It allows students to track their working and get feedback from their advisors, and work on them timely. For Teachers, management of multiple FYPs is simplified, to make sure that, they give clear guidance and support them with ease. In addition to boost project management, Questify aims to save time, resources, and money of students and teachers. That will be done by integrating features such as notifications, shared calendars, and online meetings, that features with fill out the communication gap, and making FYP experience good and effective. At end, Questify FYP portal full fill the goals of Lahore Garrison University by improving the working experience, and developing students for their future careers' effective project management.

3.2 Overall Description

3.2.1 Product Perspective

Questify FYP portal is unrestricted and customized web system that developed to serve the Final Year project management of Lahore Garrison University's requirement. It is designed for Computer science department but can be extended to others, it serves as common platform for students, teachers, admin department, and external examiner. The system has a user-friendly interface for submission, supervisor allocation, work tracking, document checking, communication and grading. It collaborates with university's Student portal system, to verify student's enrollment, categorizing supervisor by their expertise, email notifications, and work tracking. It is a secure online system which accessible from any browser. This removes the need of external software installation. System makes sure the consistency of data access and make the connection between students, faculty and administrator easy.

3.2.1.1 Dependencies on Third-Party Software

- **Node.js:** For server-side development to handle requests.
- **React.js:** For creating a dynamic and responsive user interface.
- **MongoDB:** For database management and secure storage of user data and project-related information.
- **Web Socket Services:** To enable real-time communication features such as notifications.
- **Email Service APIs** (e.g., SendGrid or Node mailer): For sending automated email notifications to users.
- **Cloud Storage Platforms** (e.g., AWS S3, Google Drive): For secure document management and file-sharing capabilities.
- **OAuth 2.0:** For secure user authentication through integration with the university's Single Sign-On (SSO) system.

3.2.1.2 Limitations of the Proposed System

- **Initial Scope Limitation:** Designed for the Computer Science department, requiring additional development for other departments.
- **Internet Dependency:** The system requires a stable internet connection for access and operation.
- **Limited Offline Features:** Offline access is not supported, and all operations depend on real-time connections.
- **Technical Proficiency:** Supervisors and administrators may require initial training.
- **Resource-Intensive:** The system relies on robust university infrastructure, including reliable networks, cloud storage, and security protocols.
- **Dependency on Third-Party Services:** Any downtime or issues with third-party APIs or software (e.g., email services, cloud storage) may affect system performance.

3.2.2 Product Functions

The Portal make the difficult process of management of FYP's projects easier by giving user-friendly and advanced features. The system full-fill the needs of students, supervisor and examiner making sure it serves as a collaborative and working management system. Key feature include:

Supervisor Allocation: supervisor will be assigned with project, that fit better with their expertise and availability, this will balance the load of projects.

Submit and Track Progress: Students are allowed to submit their work and check their work through the portal, and get review on it.

Review, Approve, and Grade: Supervisors and internal faculty can check submissions, approve them, assign grades, and create reports. Which will help in final grading.

Tracking and Notifications: The system track project progress and deadlines using timelines and sends notifications to users.

Document Management: Provides storage and file-sharing for the project related documents.

Communication Tools: Announcements, and meeting scheduling features to allow collaboration and communication between student and supervisor.

Grading System: Track submissions, feedback from internal and external examiners, and automatically generates grades.

3.2.3 User Classes and Characteristics

Questify have multiple roles of users. That's why our system is designed with flexible and have access levels for users.

3.2.3.1 Primary Users

The users who will interact with FYP portal frequently and have major roles in FYP Portal system.

- **Students:** The primary users who work along with the portal to submit proposals, update work, and manage documents. They require little skills to use the system.
- **Supervisors:** Faculty who guide students, grade their work, and track project progress. Supervisors are expected to have the subject's expertise to judge the projects more accurately.
- **Administrator:** Technical person who will maintain functionality if portal, marking issues, checking users' feedbacks and making changing in codes only when needed.

3.2.3.2 Secondary Users

Those users who will interact less with portal.

Administrator: Responsible for the managements of users, generating reports of the work, make sure the smooth process.

Department Head: Keep an eye on overall work, approves serious decisions, and view reports of FYP project overall.

External Examiner: Grading and review project and giving feedbacks to students, they will only interact with system at time of evaluation.

3.2.4 Operating Environment

The Questify FYP portal works in a secure, systematic and accessible environment, which is made for both server and client needs. On server side, system need good hardware with at least 8GB RAM, multi core processors, and 500GB storage with RAID for duplication, Node.js for the backend and MongoDB for database. The frontend is built with React, which will make the system responsive and fast have better user experience. Client can access the portal from any up-to-date device with a stable internet connection, by using browser Chrome, Firefox, Edge or Safari. The system uses secure HTTPS communication, real-time features via Web Sockets, and handles 500 users at same time, average response time 3 seconds. It blends with the university's authentication system, email protocol, and common document template. Testing is done by using Postman, and security is ensured through SSL/TLS encryption, firewalls, role-based access control, and regular backup. The FYP portal make sure a good and user-friendly experience for all the users.

3.2.5 Design and Implementation Constraints

The portal is designed with unique control to meet technical, security, administrative, and performance level. It uses React.js with JavaScript for the frontend, Node.js with Express for the backend, and MongoDB for the database, handle via Git in the university's private repository and implemented on on-premise servers. Security have the top importance, with role-based access, encrypted data, and audit on weekly bases to match data with university's data protection policies and HEC rules. The system is user-friendly. It meets the highest availability level, making it easy for everyone to use, in other cases as their abilities. You can access and use Questify on any device, it can be your laptop, tablet, or smartphone because system is responsive accessible in all devices. Files uploads are limited to 50MB, with a 1GB limit per project daily backups, keeping data for 5 years. Development follow Agile method, with 2 weeks sprint, code reviews, 80% test coverage, and complete documentation, all within 6-month timeline and limited resources. Integrating with

university's SSO and email system is must. The portal can handle up to 500 users. Maintenance is arranged around the calendar, after updates the system will be available from 8AM to 10PM.

3.2.6 User Documentation

The Questify FYP portal provides user friendly documentation for smooth experience for all users. It has detailed manual for students, supervisors, and administrator, covering all functions like proposal submission, work tracking, document management, grading, and communication. An online help system offers assistance, FAQs, and video tutorials. Guide will help the students, teachers, admins, and external examiners. Users can watch video tutorials for common task like submitting proposal, updating progress, and managing document. Training material, consist of slides, templates, PDFs and MP4s. The documentation follows regular updates, version, and user feedback changings to make sure the accurate and helpful information.

3.2.7 Assumptions and Dependencies

The Questify FYP Portal relies on several assumptions and dependencies to function effectively. Technically, it assumes compatibility with OAuth 2.0(specifications for API authentication from university's systems.) for the university's authentication system, stable network bandwidth for concurrent users, and browser support for progressive web applications. Operationally, it expects timely updates to the academic calendar, faculty and student readiness, and a maximum project group size of four. Resource assumptions include IT support availability, sufficient storage for five years of data, a stable development team, and adequate training resources. Key dependencies include integration with the university's SSO system, email servers, cloud storage, and academic calendar. The system also relies on third-party software such as Node.js, React.js, and WebSocket services, supported by robust university infrastructure like reliable networks, backup systems, and security certifications. Content and process dependencies involve department-provided rubrics, templates, policies, supervisor data, and adherence to academic schedules. Risks are mitigated through regular stakeholder validation, dependency monitoring, alternative solutions, and thorough documentation of changes.

3.3 External Interface Requirements

3.3.1 User Interfaces

The Questify FYP Portal's user interface follows modern, user-friendly design principles, starting with Material Design guidelines and incorporating the university's brand colors. It features a responsive layout adaptable to mobile, tablet, and desktop screens, ensuring accessibility for all users. The interface includes a persistent top navigation bar with a university logo, profile menu, notifications, and help icons, along with a left sidebar for navigation, breadcrumb trails for context, and a footer with useful links.

Role specific dashboards provide good experiences:

- Students can access project statuses, timelines, and quick actions
- Supervisors see project overviews, meeting schedules, and evaluations
- Administrators monitor system status, manage users, and configure settings

Key components include a multi-step project submission form with drag-and-drop uploads, Gantt charts for progress tracking, and rubric-based evaluation tools. Error handling is intuitive, with color-coded messages, real-time validation, and clear recovery instructions.

3.3.2 Hardware Interfaces

It is necessary to have a central server for data storage so that the FYP portal can perform functions of archiving data without hindrances. For editing servers with an architecture of Intel Core i5 processor (or an alternative with similar specifications), a storage space of 500GB (either HDD or SSD), a minimum of 8GB RAM, and stable internet connectivity are recommended. Other types of user devices meet the requirements mobile smartphones, tablets, desktops, and laptops. These devices are expected to have a minimum RAM of 2GB for optimal system performance and faster response.

3.3.3 Software Interfaces

The back-end of the FYP portal will be developed with Node.js and Express.js to handle the server-side features and API calls. The front-end will be developed with React.js, with an interactive and user-friendly design. MongoDB will be used for data management and storage. It provides a scalable and adaptive database alternative. The API of Zoom will be integrated into the system to

allow for meeting planning and execution through the site. For the convenience of every customer, the FYP site will be compatible with various operating systems, such as Windows, macOS, Linux, Android, and iOS. To ensure a seamless experience for viewers on all platforms, the site will also be customized for widely used web browsers, including Chrome, Firefox, and Safari.

3.3.4 Communications Gap

The FYP portal has many features who are meant to make efficient or constant interaction with students, supervisors, along with administrators. For instance, an inbuilt converse feature enables students and instructors to get responses promptly, and thus avoid any kind of delay in conversations and revisions, shared calendars enable all the users to schedule and manage their meetings, and thus it ensures that deadlines along with crucial milestones are evident, the portal also provides file-sharing features securely, enabling students to upload and share project-related files with their instructors. administrators are able to trace all conversation and take suitable action if there is any overlooked issue, and all the communications, both through the application and through emails, automatically notify the users for any new updates, scheduled conferences, and looming due dates, thus maintaining consistent engagement or interaction.

3.4 System Features

3.4.1 Submission of proposal

3.4.1.1. Description and priority

Description: Students may be able to submit their proposals of project through the FYP portal for the approval of their instructors. Since it marks the beginning of the project lifecycle. This process ensures that each proposal is analyzed step by step and published on the site.

3.4.1.2. Stimulus/Response Sequences

Stimulus: The students will upload their project proposal with all details.

Response: The system will check validation of file and notify the supervisor (using email API). Then the supervisor reviews the proposal and give feedback or approve.

3.4.2.3 Functional requirements

- **REQ-1** Student must provide all the details about their project.

- **REQ-2:** System reviews the details and validate that it is in correct format or not.
- **REQ-3:** When the student submits their proposal then supervisors will be notified.
- **REQ-4:** Supervisors are responsible to review and respond to students within a specified period.
- **REQ-5:** The response about the proposal will send to the students (accepted/rejected).

3.4.2 Assignment to the Supervisor

3.4.2.1. Description and priority

Description: The student may choose a supervisor from the website portal. This feature does not bring about disagreements and ensures effective allotment of a supervisor.

Priority: This feature is most important because it is start and main point of our entire project journey. It makes sure we carefully consider proposals, which is a main first step for any successful FYP project.

3.4.2.2. Stimulus/Response Sequences

Stimulus: The student will select supervisor from available list.

Response: The selected supervisor get notification of the request, reviews it, accepts or rejects it. The system updates both the student and supervisor with the status of the request.

3.4.2.3 Functional Requirements

- **REQ-1:** Available supervisors list will be shown with students.
- **REQ-2:** From the available supervisor list student can select the supervisor.
- **REQ-3:** The supervisors receive the notification from students who have sent them a request.
- **REQ-4:** Then supervisor will accept or reject the requests of the students.
- **REQ-5:** The status will be shared with the students through notification.
- **REQ-6:** Administrator can check and ensure that supervisors are performing their duties properly.

3.4.3 Progress Monitoring

3.4.3.1. Description and priority

Description: A development tracking tool can help both supervisors and students monitor the progress of ongoing projects. Students can track their progress by including relevant information, this feature ensures that accountability and openness provide projects with a maintained course and that problems are resolved as quickly as possible.

Priority: This feature is important for both team members and supervisors. It helps us create a system where everyone has balanced work load.

3.4.3.2. Stimulus/Response Sequences

Stimulus: Students updates and complete work and milestones and pending tasks.

Response: Supervisors checks progress, provide feedback, and suggest corrections if needed. Progress and feedback are displayed on a dashboard for students by the supervisors.

3.4.3.3 Functional requirements

- **REQ-1:** Students can track their progress and view their pending tasks.
- **REQ-2:** Supervisor review the progress and also provide the feedback about the project.
- **REQ-3:** Progress and their feedback will be sent to the supervisors and students through the notification.
- **REQ-4:** Dashboard display overall progress of the project.
- **REQ-5:** Administrator can monitor all the work and can give the feedback if required

3.4.4 Administration Management

3.4.4.1. Description and priority

Description: Administrators will be needed for the general working of the FYP portal. They would be responsible for the generation and maintenance of supervisors and students' accounts. This enables easy monitoring and decisions, ensuring a smooth-running portal.

Priority: This feature, can remove stress and avoid last-minute rushes. It helps us to manage our deadlines, making our work more managed and professional.

3.4.4.2. Stimulus/Response Sequences

Stimulus: An admin accesses the portal to manage users, monitor submissions, or resolve conflicts.

Response: The system provides tools for account management, feedback, add or remove students or supervisor, notifications, and analytics.

3.4.4.3 Functional requirement

- **REQ-1:** Administrator can view all the proposals and ensure that there is no miss management.
- **REQ-2:** Administrator manages the notification like status of proposal accept or reject, deadlines.
- **REQ-3:** If there is any problem between student and teacher portals then administrator will resolve the issue.

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

The FYP portal would be able to handle 500 users at the same time without creating any problem. Average response time even during peak hours in page loading is maintained at below three seconds to maintain optimum performance so that the interface appears responsive as well as lag-free.

3.5.2 Safety Conditions

Safety is an integral part of the FYP portal. Data loss because of system crashes will be avoided through planned systematic data backup. The system will also tolerate abrupt system crashes gracefully so that the users' data is saved and recoverable.

3.5.3 Security Requirements

Security in the FYP portal is critical in safeguarding sensitive students and project information. The platform uses role-based visit control, or RBAC (role-based access control), to ensure that only users with permission can access specific features and data. Administrators are granted the highest range of access, allowing them to control user profiles and system policies. All sensitive

details is secured during transit and storage spaces. It protects it from accidental access. Users register through secure access credentials, which are saved properly within the service.

3.5.4 Characteristics of Good Software

It is easy to use for administrators, supervisors, and students through a user-friendly interface. Scalability is another characteristic, which would ensure that it can handle more features or the growth of its user base. The code was written with sustainability in mind; it has plenty of documentation on how to update and enhance later.

3.5.5 Business Rules

There are different business rules governing the FYP portal. It may be allowed for a supervisor to handle multiple five programs in a way that sufficient attention is given to each of the five programs. During these intervals, proposals by the students need to be presented. The sessions between the supervisors and the students need to be scheduled at least 24 hours in advance to ensure that the proper preparation could be done. The needs of students should be met by supervisors within 3 working days (Monday to Friday), ensuring that they have enough time for analysis and progression. Managers can implement these needs and settle disputes that arise during the lifecycle of the progress.

METHODOLOGY

In this chapter, we outline the systematic approach taken to develop the Questify FYP Portal, covering research, design, development, testing and deployment. The portal was built using a modern tech stack (React, Node.js, and MongoDB) to ensure scalability, security and usability for students, supervisor and administrators.

4.1 Research and Analysis

We conducted an in-depth study of existing FYP management systems, university workflows, and pain points in academic project supervision. Through surveys, interviews with faculty and students, and analysis of competing platforms, we identified key requirements: streamlined proposal submissions, supervisor selection real-time progress tracking and secure document management. Academic literature on agile development, role-based access control and secure authentication (OAuth 2.0) further informed our design.

4.1.1 Requirements Gathering

Functional and non-functional requirements were prioritized via stakeholder feedback. Key features included:

- Functional: Proposal submission workflows, supervisor allocation algorithms, progress dashboards and evaluation rubrics.
- Nonfunctional: HTTPS security, < 3-second respond time, 500+concurrent user support and cross browser compatibility.

User stories and UML diagrams mapped interactions between students, supervisor and admins.

4.2 System design

The portal's architecture was designed or modularity:

- Frontend: React.js for dynamic UI and Redux for state management.
- Backend: Node.js/Express Restful APIs handling logic.

- Database: MongoDB for flexible document storage with encrypted sensitive data.
- Security: JWT authentication, RBAC (role-based access control).

4.3 Development

An agile approach ensured iterative improvements:

- Core modules: Proposal submission and approval system with version control.
- Real-time features: Web Socket notifications for deadlines and messages.
- Document manager: secure cloud storage and role-based access.

4.4 Testing

- **Integration Testing:** Postman for API endpoints
- **User acceptance testing (UAT):** Faculty/student trails validated usability.
- **Security testing:** OWASP ZAP for vulnerability scans and penetration tests.

4.5 Deployment

To make Questify accessible to users, we:

- Hosted it locally on a space laptop running windows.
- Using node.js to run the backend server.
- Set up simple proves manager to keep the server running.
- Created manual backups of important data using Google drive.
- Use mongo DB atlas for cloud database storage.

4.6 Documentation and maintenance

We created easy to follow guides for admins and users, including:

- Step by step manuals for managing the portal.
- Video tutorials for common task

-

4.7 Tools and technologies

Category	Tool used	purpose
Frontend	React.js	Build user friendly interfaces
Backend	Node.js, express.js	Handle server logic and APIs
Database	Mongo DB	Store and manage project data securely
Security	JWT , OAuth 2.0	Protect user accounts and data
Testing	Postman	Check for bugs

Table 1: Used tools and technology

4.8 Hardware / software requirements

4.8.1 Server needs:

- Operating system: Window, Linux.
- Memory: 8 GB ram
- Storage 500 GB secure storage (with backups).
- Software: node.js (version 18 or newer).

4.3.2 User needs:

- Device: any computer with 4 GB ram.
- Browser: chrome, Firefox, safari and edge.
- Internet: stable connection for real-time features.

DETAILED DESIGN AND ARCHITECTURE

5.1 System Architecture

The Questify FYP Portal is designed as a comprehensive academic management system that streamlines Final Year Project workflows for students, supervisors and administrators built on the MERN stack (Mongodb, Express.js, React, Node.js) the architecture integrates three core user role: student, supervisor and administrator. Student submit proposals and track progress through the dashboard, supervisor review submissions and evaluate milestones and administrator manages user permissions and system through secure control panels. At its basic, the system leverages React.js for dynamic user interfaces, Node.js/Express for backend logic, and Mongodb for flexible data storage. Real time notifications for deadlines, approvals and messages, while secure document management.

5.1.1 Architecture Design Approach

The Questify portal adopts a modular, role-driven architecture to balance flexibility and security. Students interact with proposal submission forms and progress trackers, while supervisors access evaluation rubrics and communication tools. Admins oversee workflows through centralized dashboards, resolving conflicts and configuring system-wide settings. Modularity ensures independent development of features like document uploads for students, grading tools for supervisors, and audit logs for admins. Security is enforced through role based access control, restricting students to their own projects, supervisors to assigned submissions and admin to administrative controls.

5.1.2 Architecture Design

The Questify architecture is organized into four interconnected layers ensuring seamless functionality while prioritizing security. The presentation layer, built with React.js, provides role-specific interfaces: student submit proposals and track milestones through intuitive forms,

supervisors review submissions and grade work via dashboards and administrator manage user and monitor system through their portal. The application layer, powered by Node.js and Express.js handles critical operations like routing proposal to supervisors, processing evaluations enabling manual supervisor assignment by admins.

The security layer enforces JWT-based authentication to validate user sessions and Role-based access control (RBAC) to restrict permissions: student access only their own project, supervisors view assigned submissions and admin oversee system-wide configurations.

5.1.3 Subsystem Architecture

The portal comprises four core subsystems:

1. Proposal submission subsystem:
 - Allow students to submit proposal in standard formats (PDF/DOCX) through intuitive React.js interface. The subsystem tracks submission statuses. Administrator review submissions for compliance with departmental guidelines and approve or reject proposals.
2. Supervisor Allocation subsystem:
 - Administrator manually assign supervisor to students projects using an interactive dashboard. The subsystem displays supervisor profiles, including expertise areas
3. Progress tracking subsystem:
 - Admin define project milestones and deadlines students update task completion status. Admin monitor overall progress and configure system-wide timelines.
4. Evaluation subsystem:
 - Enables rubric-based grading with predefined criteria
 - Facilitates feedback collection from supervisors.
 - Aggregates final grades for administrative approvals and expert.

5.2 Detailed System Design

5.2.1 Classification

Component: Supervisor Allocation Subsystem

5.2.2 Definition

This subsystem enables administrators to manually assign supervisors to student's projects based on expertise, availability and workload. It provides tools for admins to view proposals, assess supervisor capacity.

5.2.3 Responsibilities

The subsystem displays student proposals alongside supervisor profiles (expertise areas, maximum project capacity). Administrators manually link students to supervisors using add and drop interface.

5.2.4 Constraints

- **Usability:** The interface must allow admins to compare proposals and supervisors efficiently.
- **Data accuracy:** Supervisor's workload and expertise data must be updated regularly.

5.2.5 Composition

- Proposal viewer: Lists student's proposals with basic details.
- Supervisor dashboard: Display supervisor's expertise, active projects and availability.

5.2.6 Uses/Interactions

- **Inputs:** Admin assignment decisions, student proposals, supervisor data.
- **Output:** Assignment logs (student-supervisor pairs), workload reports.
- **Dependencies:** Relies on the proposal submission subsystem for student data and User management for supervisor profiles.

5.2.7 Resources

- Memory: Caches supervisor data for quick access.
- Storage: Assignment logs stored in MongoDB.
- CPU: Minimal processing.

5.2.8 Processing

- **Data loading:** Fetch student proposals and supervisor profiles.
- **Manual Assignment:** Admins assign supervisor via dropdown menus.
- **Validation:** system warns if a supervisor's workload exceeds capacity.

5.2.9 Interface/Exports

Side by side assessment of proposals and supervisors. Filter supervisors by their expertise (e.g “cybersecurity”, “Machine learning”). Retrieve student proposals through the Rest APIs (GET /api/proposals).

5.2.10 Detailed Subsystem Design

The subsystem is designed to give administrators full control over supervisor assignments while keeping the process clear and organized. Admins can easily search for proposals and supervisors using filters like the expertise areas to find the best matches. Every assignment is recorded with simple details like the admin's name, the date and the supervisor ID, making it easy to track decisions later. If any proposals remain unassigned after initial review, the system marks them for follow-up, sending weekly email reminders to admins to ensure no student left without a supervisor.

5.2.10.1 Use case Diagram

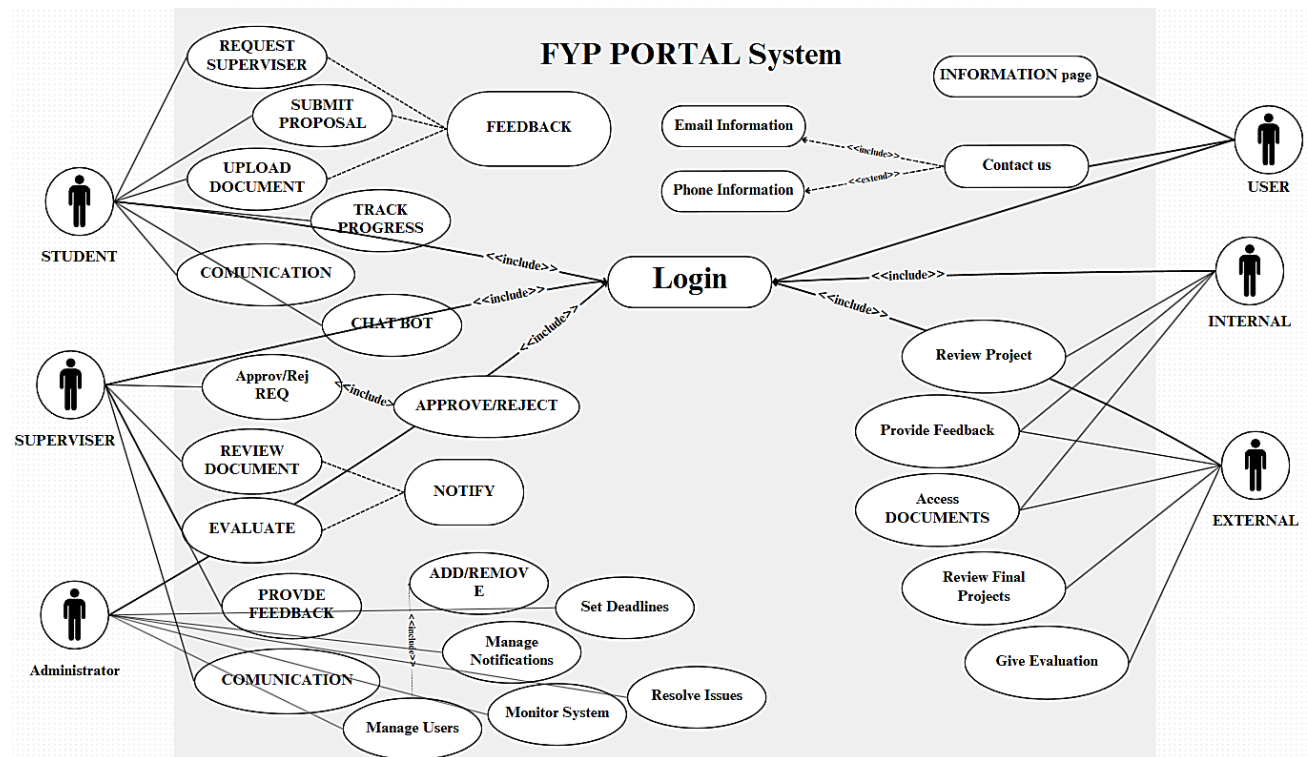


Figure 2: Use case of all users

The Use Case Diagram for the Final Year Project (FYP) Portal is picture of how every user like students, supervisors, admin, and evaluators will interact with the system. Student are the main users, they will perform task like submitting project proposal, uploading documents, tracking their progress, and communicating with supervisors. Supervisors will review proposal, give feedback, approving documents, creating meeting with student to guide them. Admin will make sure the system works properly, managing the account of users, send crucial mails, setting deadlines for the submissions, and fixing overall issues. Evaluator will access the project's documents, evaluating and providing feedback on them and at end grading them. The whole system starts after simple login, showing options that make managing FYP's easier and efficient.

5.2.10.2 Architectural Diagram

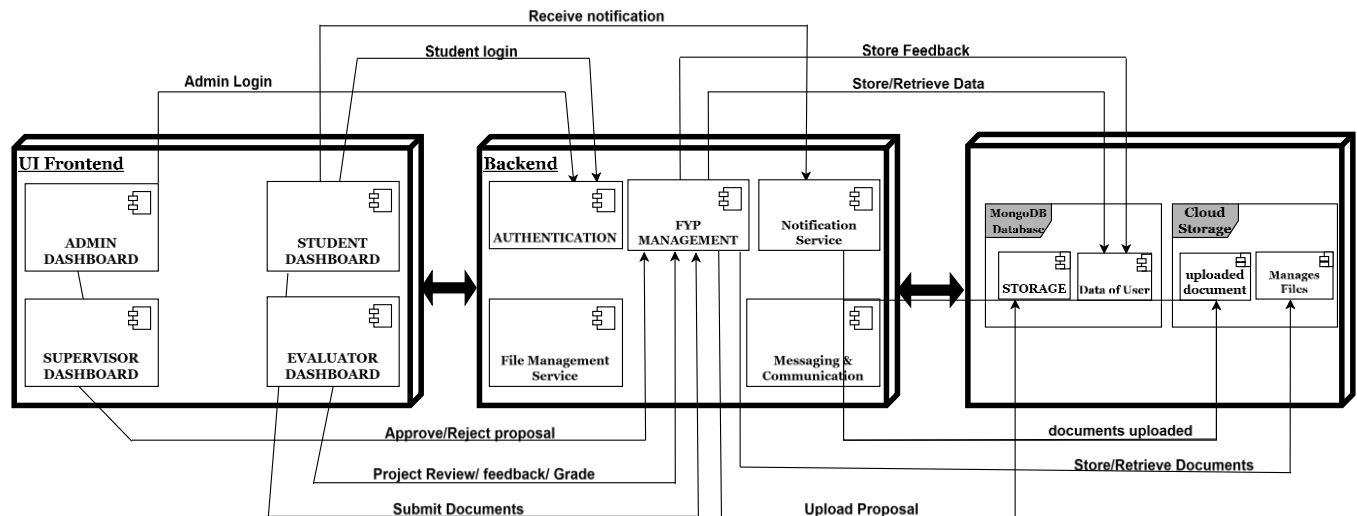


Figure 3: I-Architectural Diagram

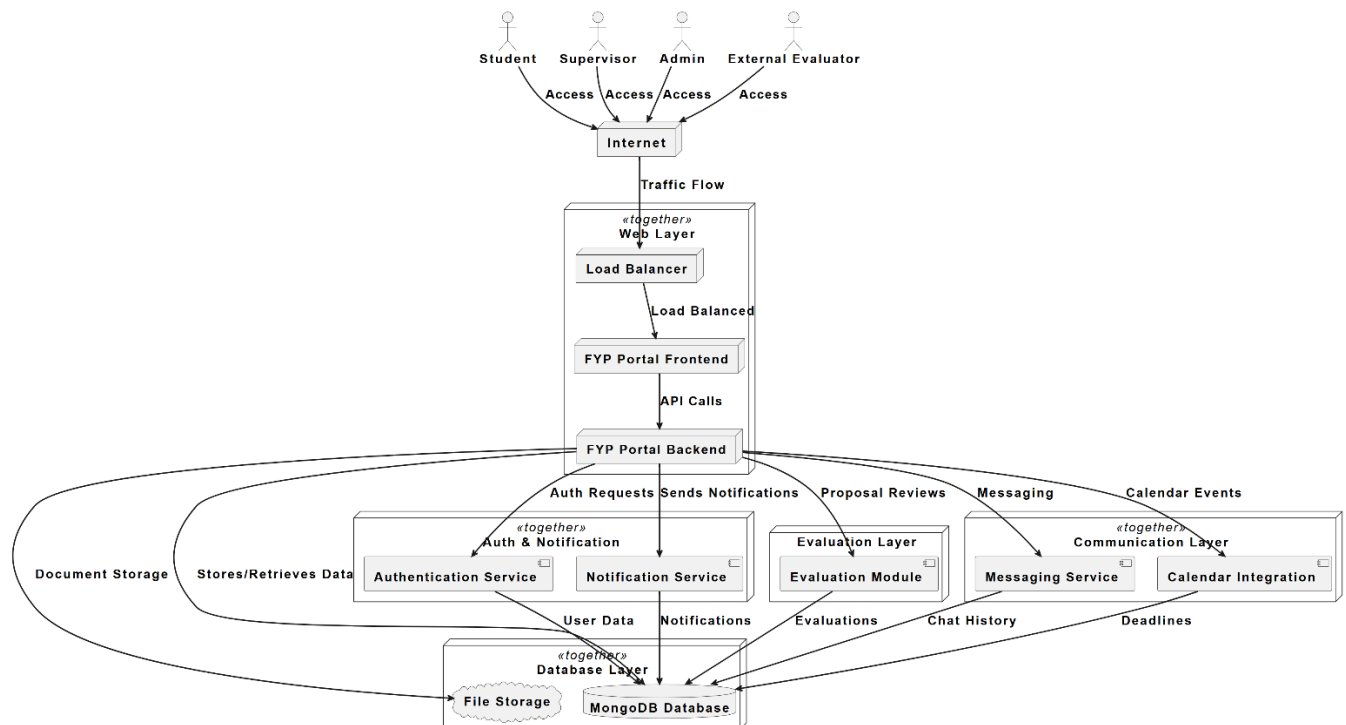


Figure 4: II-Architectural Diagram

The architectural diagram provides a clear understanding of Final year Project (FYP) management system, how it works, a clear view how main four users' students, supervisors, admin and evaluators interact with each other within system. Student, supervisor and admin will use the main FYP portal, whereas the evaluators will only access evaluation tab. This will help in smooth

running of system; a load balancer distributes the user's traffic over the server. The web layer has frontend, which take care of user interaction, and the backend, it will work on logic, API requests, connection of important services. The backend also links with authentication service for users' verification for accessing the portal, and notification services to share all new important messages. All data is stored in MongoDB database, whereas uploaded files are managed separately. The design shows the main features like document storage, reviews, messaging, feedback, calendar integration, map integration, and evaluation workflow, providing a complete scenario of how the system work in FYP management while making the experience smooth, efficient and scalable performance.

5.2.10.3 ER Diagram

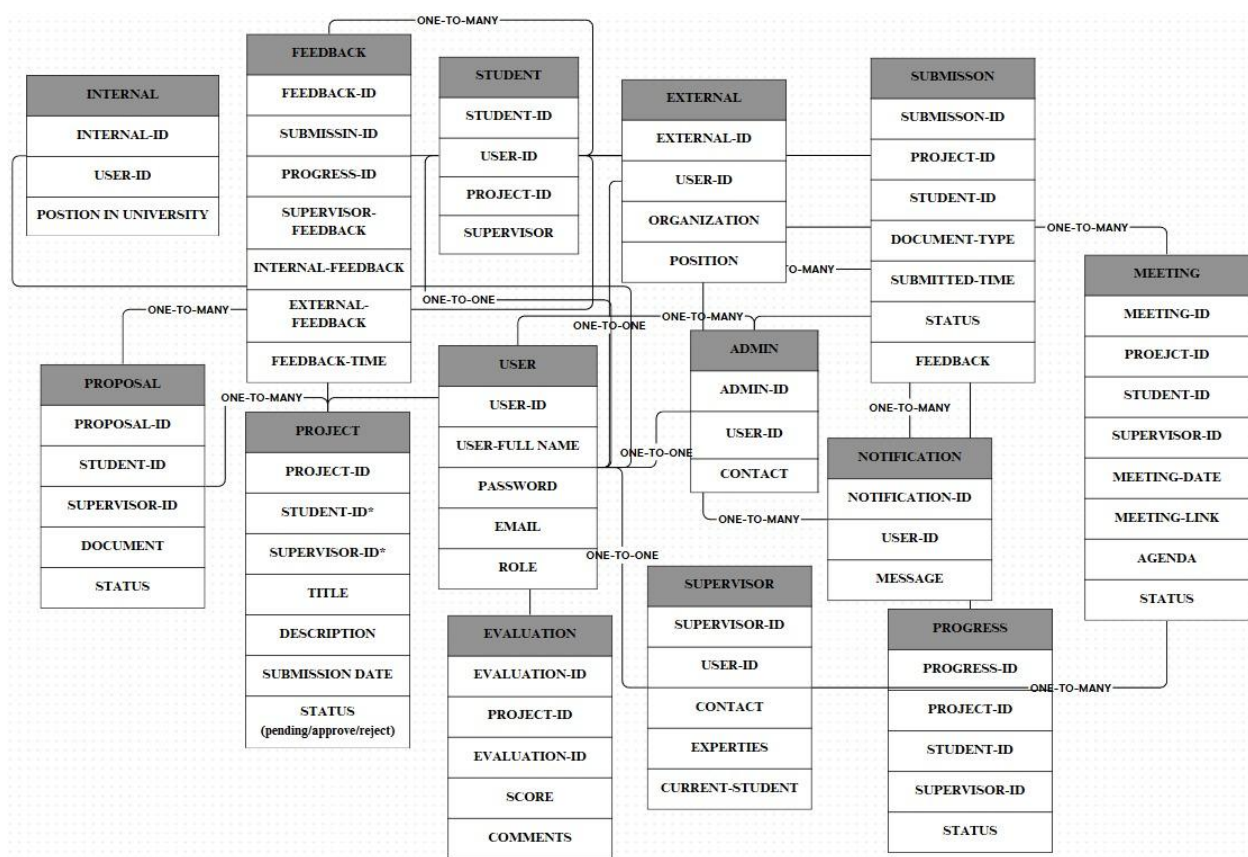


Figure 5: Entities relations Diagram

The structure of the Questify FYP portal telling how the entities of project interact with each other and what are the relation between users.

5.2.10.4 Activity Diagram

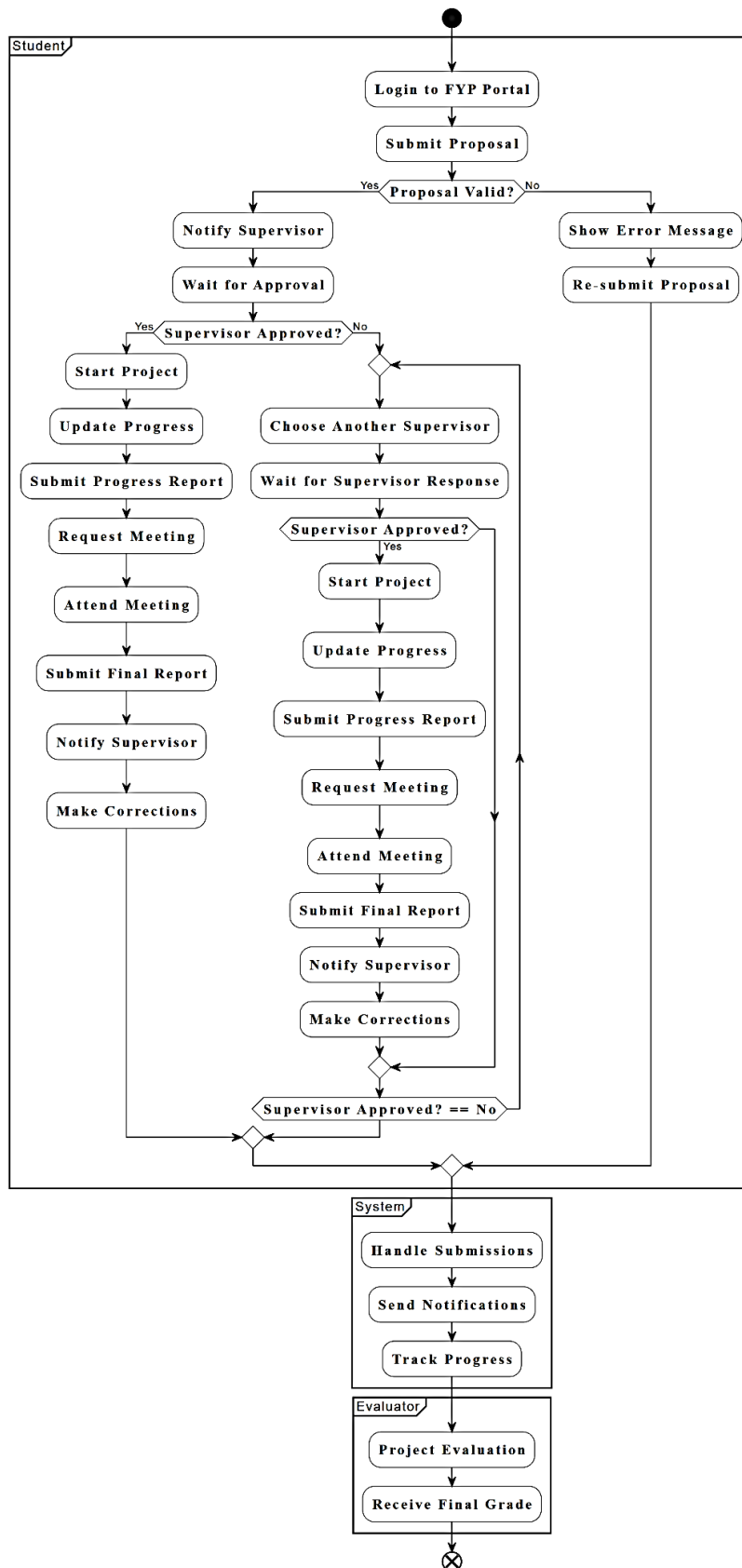


Figure 6: Activity Diagram

This activity diagram shows the interaction of all users with the Questify FYP portal system, starting from the simple login by the student, follow up by submitting proposal, system check validation, select supervisor, system notify supervisor about proposal, then accept or again for updating after feedback, update progress, system store every changing and progress in project, meeting requests, notifications, when almost near to completion of project, submit final report, get supervisor review, then evaluation, where evaluators access document and review project then provide feedback and grade them. At admin level he manages users, deadlines, notifications, and resolve conflicts and finally generate grade, and the system marks the project as completed.

5.2.10.5 Sequences Diagram

The image depicts various sequence diagrams for the Final Year Project (FYP) System, illustrating the interactions between different actors (Student, Supervisor, Admin, Internal, External) and the system itself.

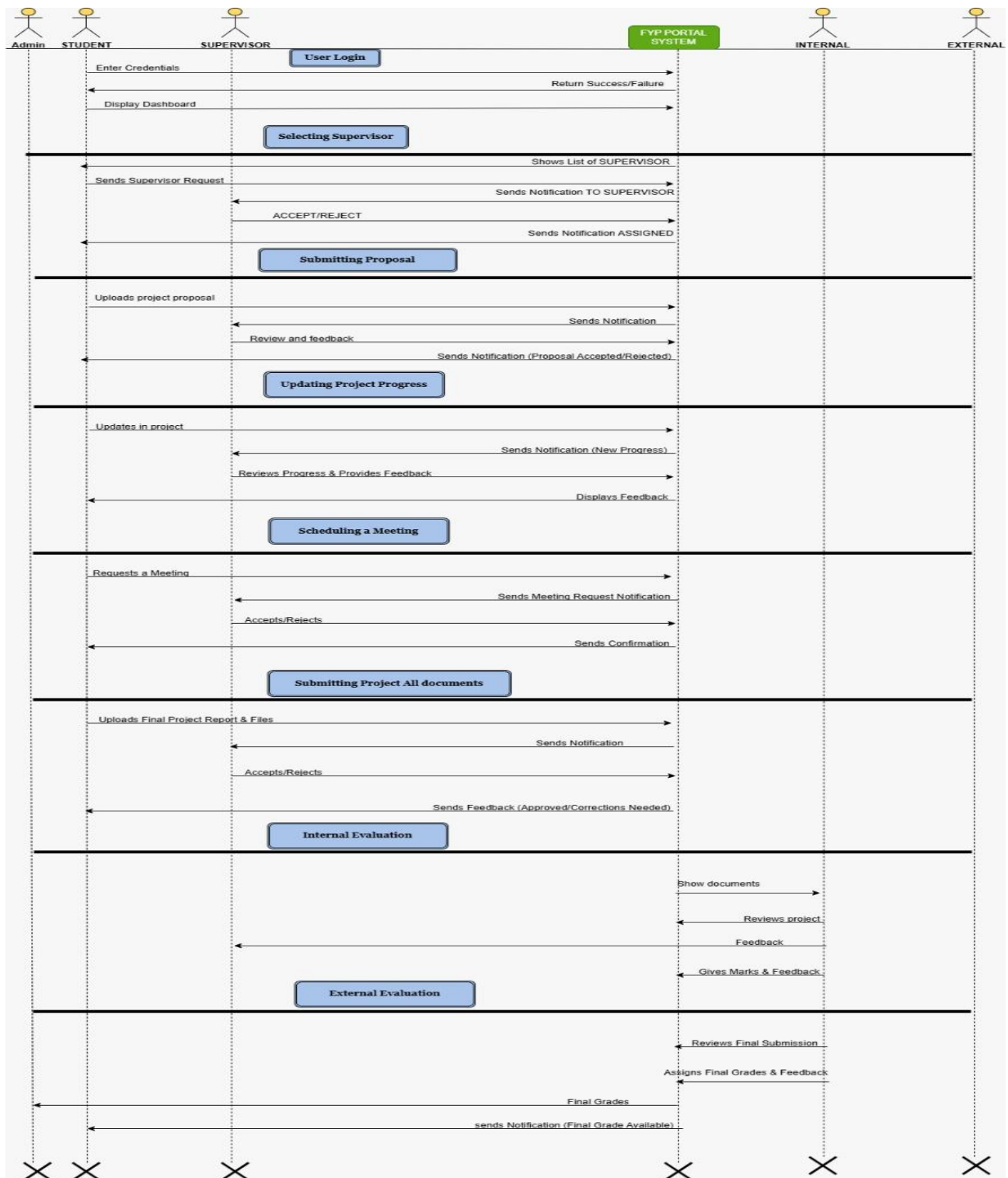


Figure 7: Sequence Diagram

The diagram shows the student journey throughout the Questify FYP portal, start with login that allow to access portal according to their role. As student first project proposal, then selecting supervisor, send request, when supervisor assigned, review proposal then approve or return with feedback, Further student progress and update project, constantly get feedback from supervisor. Creating meeting for communication, with request and confirmation handle directly. After document submission face the project enter in evaluation process, evaluator access final project and provide grades and feedback on project, that complete the FYP journey.

5.2.10.6 Component's Diagram

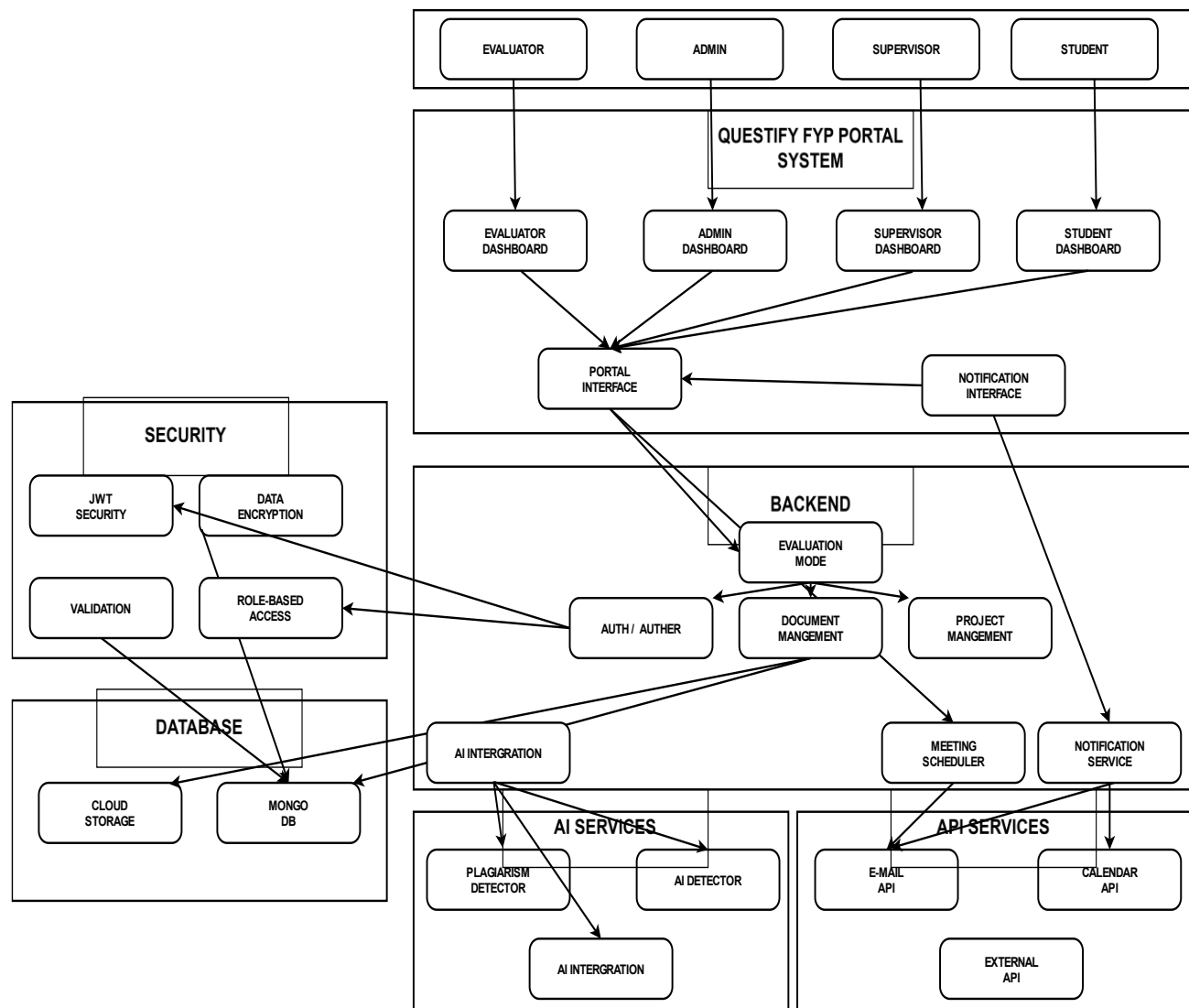


Figure 8: Components Diagram

The component diagram of Questify FYP portal shows the clear parts of the system that how it is structured, visualizing main components, how they connect and work together. All users have their personal dashboard according to their roles. All users connect with a portal interface, which

manages user interaction, navigation and all frontend experience. Whereas in backend, the system have two parts like authentication and authorization, Document management, project management, evaluation, meeting arrangement, and notification service, all work in backend to make a smooth and functional system. Security is core have feature, JWT authentication, data encryption, validation, and role-based access control. Data is stored and managed by MongoDB and cloud storage for documents. The system has AI services for plagiarism and AI detection, more services via API requests, including email API, calendar API and other third-party integrations. The diagram highlights the modules, secure and scalable nature of the Questify FYP portal.

5.2.10.7 State Diagram

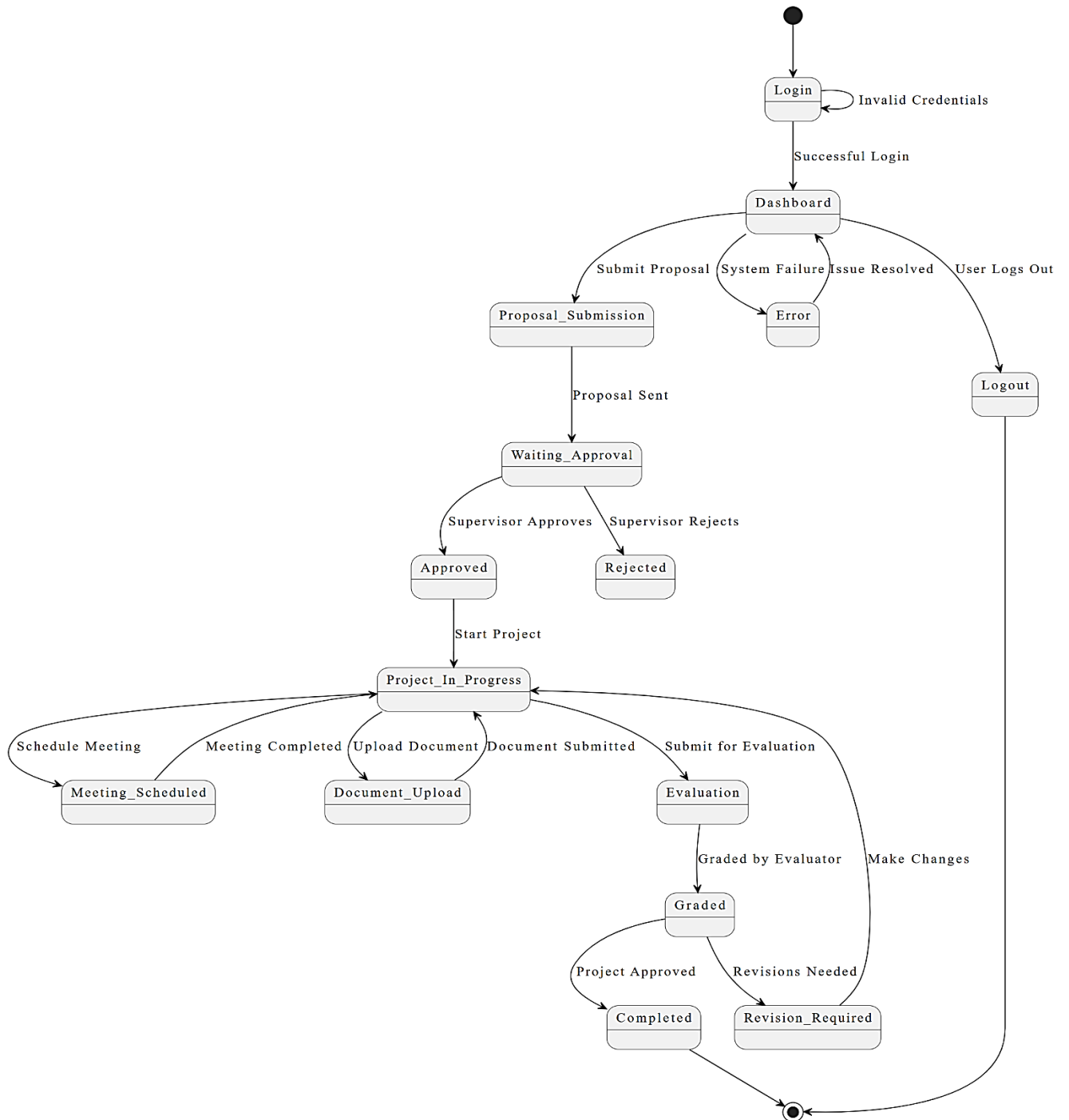


Figure 9: State Diagram

The state diagram of Questify FYP portal refers to the user interaction with system from login to project completion. Users' login, submit proposal, and move one by one through states like waiting for review, approval, feedback, project in progress and evaluation. Successful projects reach the completed state.

5.2.10.8 Class Diagram

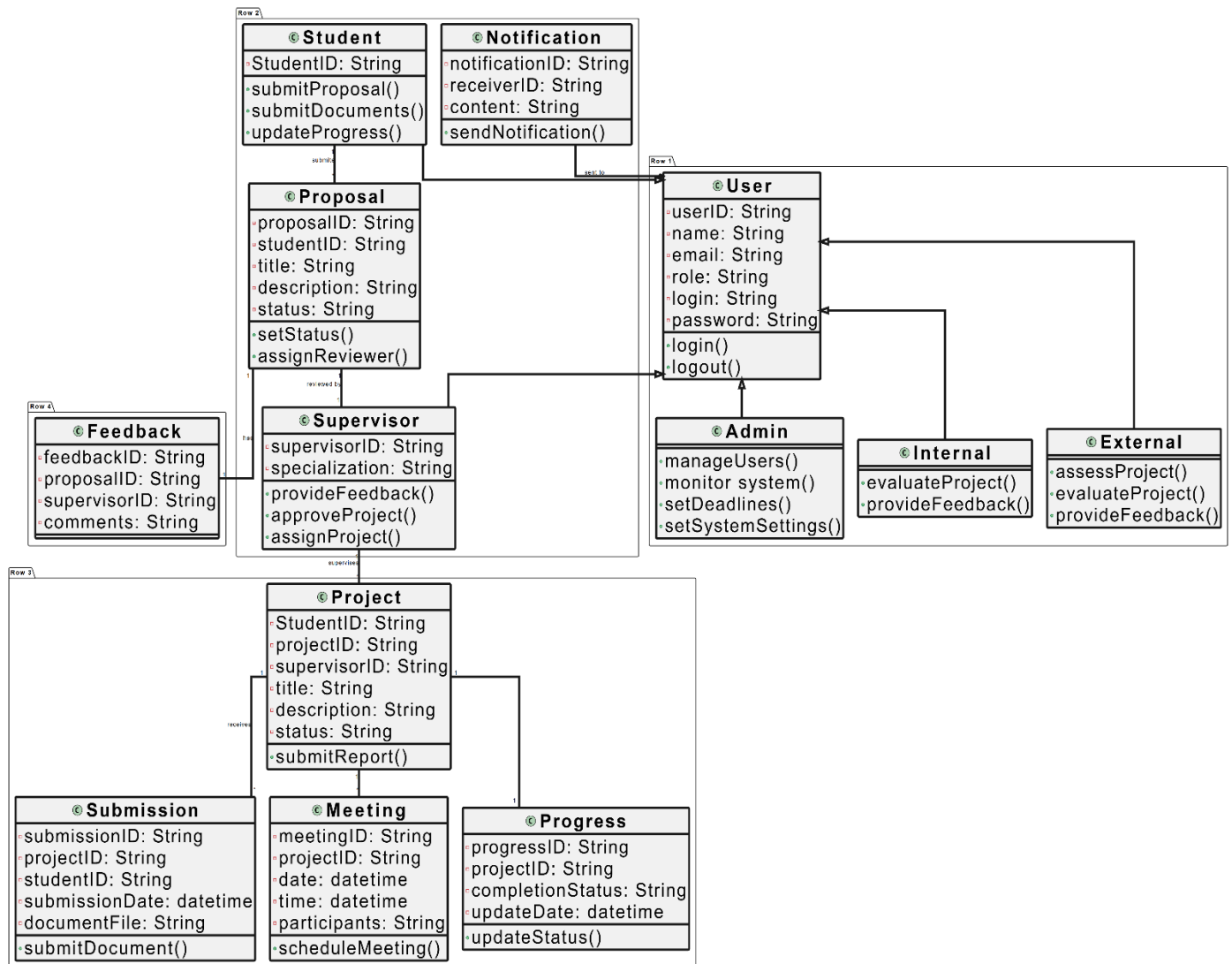


Figure 10: Class Diagram

The class diagram for Questify FYP portal shows how the system is built and how the FYP process works at Lahore Garrison University. It includes classes with key details like user ID, name, email, role, and login credentials. From this start, specific roles such as students, supervisors, Admins, and evaluators each have their own functions. The diagram shows the interactions between these classes, highlighting how users connect in an organized way for managing FYPs efficiently.

IMPLEMENTATION AND TESTING

6.1 Agile Development

To ensure an efficient and flexible development process, the **Scrum** framework was used with bi-weekly sprints. The development cycle followed an iterative approach, focusing on rapid feedback and continuous improvement. The key activities in the Agile workflow included:

- **Sprint Planning** – Defining goals and tasks for each sprint.
- **Daily Stand-ups** – Short meetings to discuss progress and roadblocks.
- **Sprint Reviews** – Evaluating completed work at the end of each sprint.
- **Sprint Retrospectives** – Identifying improvements for future sprints.

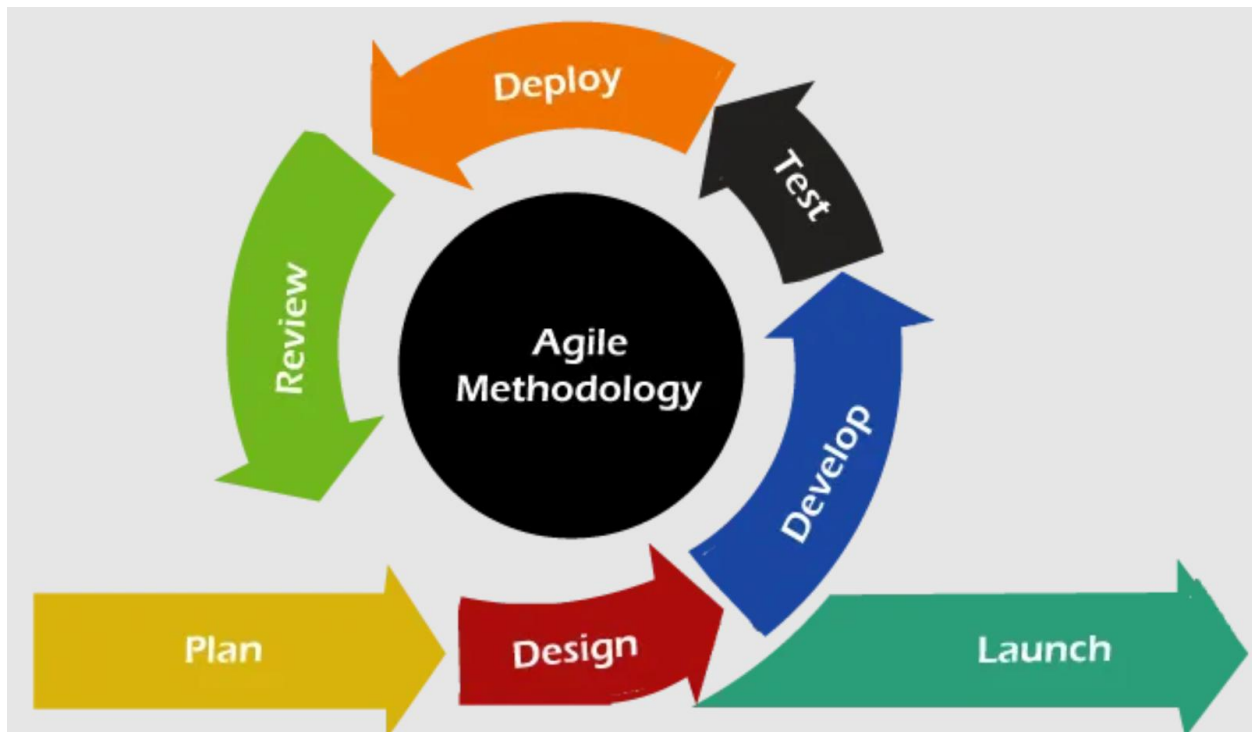


Figure 11: Agile Model

6.2 Version Control and Collaboration

- **Git** was used for version control, ensuring code integrity and collaboration among developers.
- **GitHub/GitLab** repositories facilitated issue tracking, pull requests, and collaboration.
- **Branching Strategies** like **Git-Flow** were used for efficient code management.
- Code reviews were performed before merging into the main branch to maintain code quality.

6.3 Frameworks and Libraries

6.3.1 Backend Technologies

- **Node.js** and **Express.js** were used for building a robust server-side architecture.
- **Middleware** integration for authentication, logging, and error handling to ensure a secure and smooth request/response cycle.
- **RESTful API** design to facilitate seamless communication between the frontend and backend.
- **Role-based** access control (RBAC) using JWT for secure authentication and authorization of students, supervisors, and admins.

6.3.2 Frontend Technologies

- **React.js** was chosen for its **server-side rendering (SSR)** capabilities and optimized client-side routing.
- **React.js** provided a component-based architecture, improving code reusability.
- **Bootstrap** was implemented for utility-first styling, enabling rapid UI development and easy customization.

6.4 Database Management

- **MongoDB** was selected as the NoSQL database due to its flexibility in handling unstructured data.
- **Mongoose** was used for schema validation and interaction with the database, ensuring data consistency.

- **JWT Authentication** was implemented to manage secure user authentication across student, supervisor, and admin roles.

6.5 Testing Strategy

To validate the functionality and security of the Questify FYP Portal, the following testing methodologies were adopted:

6.5.1 Functional Testing

Ensuring that each feature of the portal works as expected. Key areas tested:

- **Student Functionalities:** Login, Dashboard, Proposal Submission, Document Upload, Progress Tracking, Supervisor Selection, AI Chatbot, and Supervisor Chat.
- **Supervisor Functionalities:** Login, Dashboard, Document Review, Proposal Evaluation, and Student Feedback.

6.5.2 Performance Testing

Evaluating the system's responsiveness and stability under various conditions:

- **Load Testing:** Ensuring system stability under peak usage periods, such as when multiple students submit project proposals, upload documents, and supervisors evaluate submissions simultaneously.
- **Stress Testing:** Evaluating system performance under extreme conditions, such as a surge in concurrent logins and multiple document uploads close to submission deadlines, ensuring the system remains responsive.
- **Response Time Analysis:** Measuring API response times to ensure optimal performance.

6.6 Test Execution

Test cases were executed using manual and automated testing approaches:

- **Postman** was used to test API endpoints and responses.
- **React Testing Library** were utilized for unit testing.

6.6.1 End-to-End Testing

- Verified application behavior across different browsers and environments.
- Automated tests for login, project proposal submission, document upload, and supervisor review processes.

6.6.2 Security Testing

- Penetration testing and vulnerability scanning conducted as part of the security audit.
- Focus on securing authentication, preventing unauthorized access, and protecting stored data.
- Ensured JWT token security, proper user role access control, and input validation to mitigate injection attacks.

6.7 Test Cases

6.7.1 Validate Student Login

Test Case ID	TC-01
Test Case Title	Validate Student Login
Test Case Description	This test checks if a student can log in using valid credentials.
Pre-conditions	The student must be registered in the system.
Post-conditions	Student is redirected to the student dashboard upon successful login.
Test Steps	1. Navigate to the Questify FYP Portal login page. 2. Enter valid email and password. 3. Click the login button.
Test Case Input	Valid email and password.
Test Case Output	Success: Student is logged in and redirected to the dashboard. Failure: Error message displayed for invalid credentials.

Table 2: Test Case for Student Login

6.7.2 Validate Proposal Submission

Test Case ID	TC-02
Test Case Title	Validate Proposal Submission
Test Case Description	This test checks if students can successfully submit their project proposals.
Pre-conditions	Student must be logged in and on the proposal submission page.
Post-conditions	Proposal is stored in the database and marked as "Pending Review."

Test Steps	<ol style="list-style-type: none"> 1. Navigate to the proposal submission page. 2. Upload a valid project proposal file (PDF/DOCX/PPT). 3. Click the submit button.
Test Case Input	Valid proposal file.
Test Case Output	<p>Success: Proposal successfully submitted and pending supervisor review.</p> <p>Failure: System displays an error message for missing or invalid file format.</p>

Table 3: Test Case for Proposal submission

6.7.3 Validate Supervisor Review

Test Case ID	TC-03
Test Case Title	Validate Supervisor Review
Test Case Description	This test checks if supervisors can review and provide feedback on submitted proposals.
Pre-conditions	Supervisor must be logged in and assigned to a student's proposal.
Post-conditions	Proposal status is updated to "Approved" or "Rejected" with feedback.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the supervisor dashboard. 2. Select a pending proposal. 3. Add comments and approve or reject the proposal.

Test Case Input	Review comments and approval/rejection status.
Test Case Output	<p>Success: Proposal status updated, and student notified.</p> <p>Failure: Error message displayed if action fails.</p>

Table 4: Test Case for Supervisor review

6.7.4 Ensure JWT authentication

Test Case ID	TC04
Title	Ensure JWT authentication is working correctly
Description	Verify that JWT tokens are issued upon successful login
Pre-Conditions	Student login is required
Test Steps	<ol style="list-style-type: none"> 1. Login as a student 2. Inspect API request headers 3. Verify the JWT token is present
Expected Result	JWT token is issued and stored in the browser/local storage

Table 5: Test Case for ensure JWT authentication

6.7.5 Verify session expiration after inactivity

Test Case ID	TC05
Title	Verify session expiration after inactivity
Description	Ensure that student sessions expire after a period of inactivity
Pre-Conditions	Logged-in student session
Test Steps	<ol style="list-style-type: none"> 1. Log in as a student 2. Leave the session inactive for 60 minutes 3. Attempt to perform an action

Expected Result	Student is logged out and redirected to the login page
------------------------	--

Table 6: Verify session expiration

6.7.6 Verify students can submit proposals.

Test Case ID	TC06
Title	Verify students can submit proposals in PDF/DOCX/PPT format
Description	Ensure students can upload valid proposal documents
Pre-Conditions	Student must be logged in
Test Steps	<ol style="list-style-type: none"> 1. Navigate to "Submit Proposal" page 2. Upload a valid PDF/DOCX/PPT file 3. Click "Submit"
Expected Result	File is uploaded successfully and status is updated

Table 7: Verify students can submit proposals

6.7.7 Validate document status updates (Pending, Approved, Rejected).

Test Case ID	TC07
Title	Validate document status updates (Pending, Approved, Rejected)
Description	Ensure document status reflects supervisor actions
Pre-Conditions	Document must be submitted
Test Steps	<ol style="list-style-type: none"> 1. Upload a document 2. Wait for supervisor review 3. Check status updates

Expected Result	Status changes correctly based on supervisor action
------------------------	---

Table 8: Verify document status updates

6.7.8 Verify supervisor login with valid credentials.

Test Case ID	TC08
Title	Verify supervisor login with valid credentials
Description	Ensure that a supervisor can log in successfully using the correct credentials.
Pre-Conditions	A supervisor account must exist in the system.
Test Steps	<ol style="list-style-type: none"> 1. Open the Questify FYP Portal login page. 2. Enter a valid supervisor email. 3. Enter the correct password. 4. Click on the "Login" button.
Expected Result	The supervisor is successfully logged in and redirected to the supervisor

Table 9: Verify supervisor login

6.7.9 Verify supervisor can approve/reject proposals.

Test Case ID	TC09
Title	Ensure supervisors can approve/reject proposals
Description	Verify approval/rejection workflow
Pre-Conditions	Supervisor must be assigned proposals
Test Steps	<ol style="list-style-type: none"> 1. Open a proposal 2. Approve or reject it 3. Verify status updates
Expected Result	Status updates to "Approved or Rejected"

Table 10: Ensure supervisors can approve/reject proposals

6.7.10 Verify supervisor can download student documents.

Test Case ID	TC10
Title	Verify supervisors can view all assigned student proposals
Description	Ensure supervisors can access uploaded documents
Pre-Conditions	Students must have uploaded documents
Test Steps	1. Log in as a supervisor 2. Go to “Documents” 3. Download a file
Expected Result	File downloads successfully

Table 11: Check can download student documents.

6.7.11 Verify supervisors can submit feedback on uploaded documents.

Test Case ID	TC11
Title	Ensure supervisors can submit feedback on uploaded documents
Description	Verify document feedback workflow
Pre-Conditions	Document must be uploaded
Test Steps	1. Open a document 2. Enter feedback 3. Click "Submit"
Expected Result	Feedback is saved and visible to students

Table 12: supervisors can submit feedback on uploaded documents

RESULTS AND DISCUSSION

7.1 Login Page

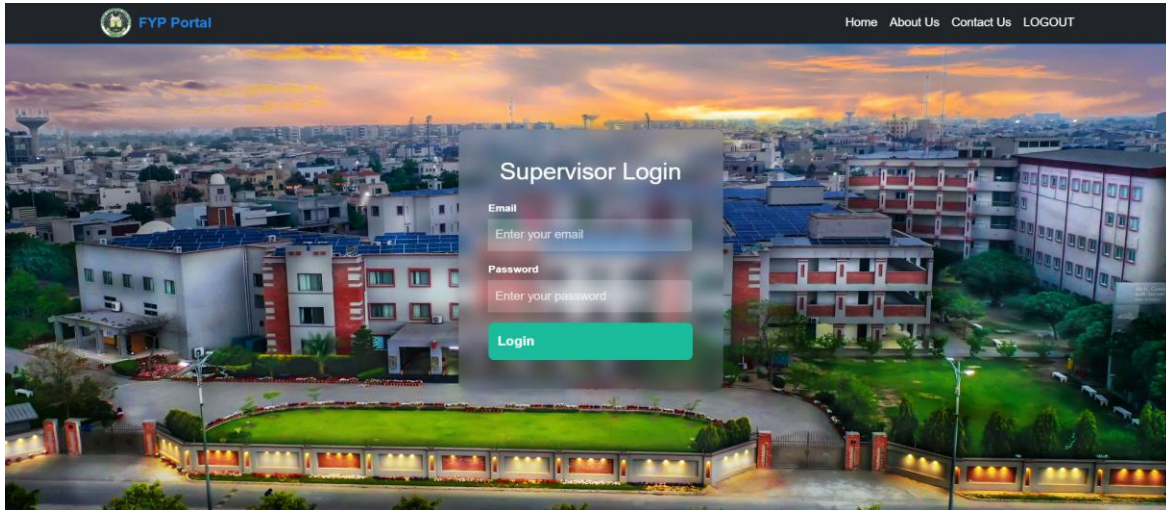


Figure 12: Login Page

The given page has functionalities to let the user's login. When the credentials are entered, then by clicking proceed system will check the if credentials are correct or not. Then they will redirect their dashboard according to their role after authorization by the system.

7.2 Student Dashboard

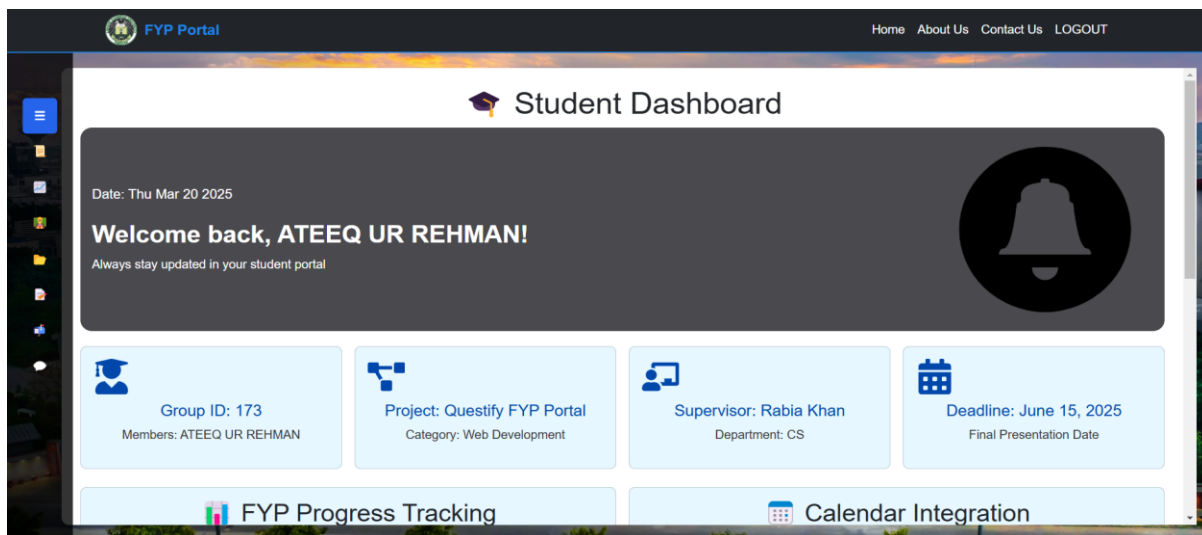


Figure 13: Student Dashboard

The Student Dashboard in the Questify FYP Portal is the main page where students can easily manage their final year projects. Once login, students see a clear dashboard which show details of their project, including the Group ID, Project Title, Supervisor's name, and the project Deadline. A sidebar menu allow access to options like Submit Proposal, Track Progress, Request Supervisor, Upload Document, Feedback/Comments, and a Communication section that includes a Chat feature for direct interactions. There's also Calendar, so students keep track of important dates and deadlines. A progress visually shows how the project progress is. The "FYP-Labs Map" highlights available labs Lab 1: 79 LAB, Lab 2: AI LAB, and Lab 3: GAMING LAB. Altogether, this dashboard is designed to help students stay organized, communicate effectively, and keep their projects on track.

7.3 Project Proposal Submission

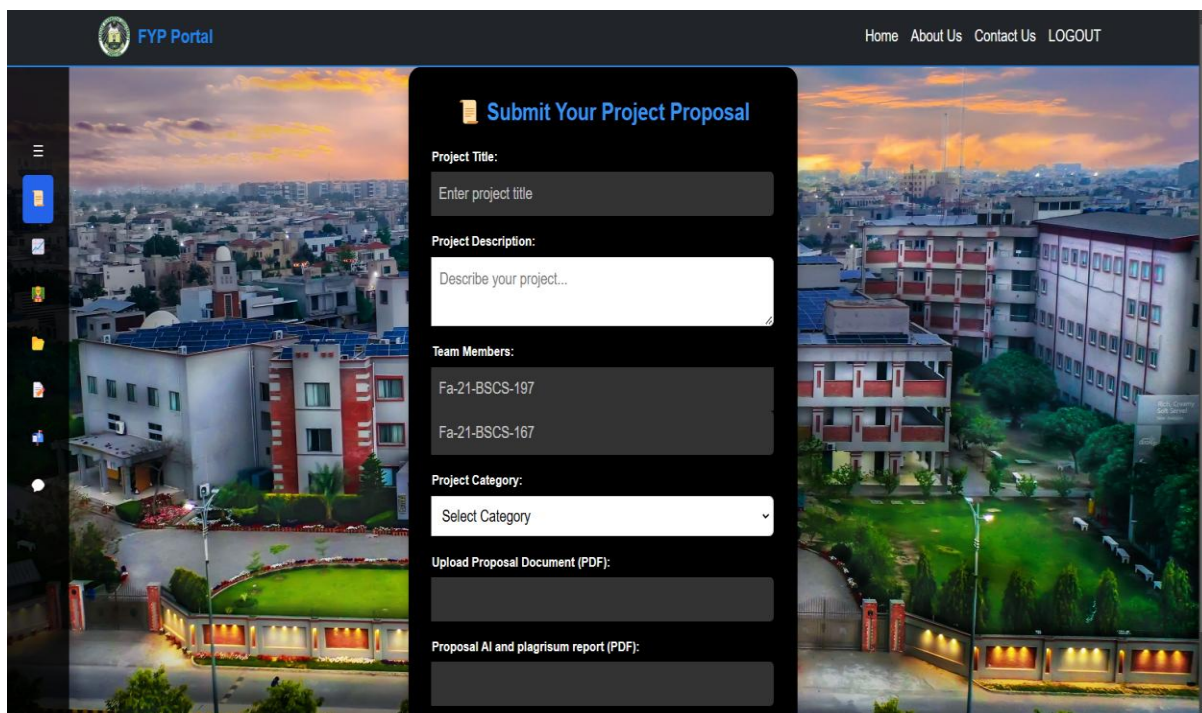


Figure 14: Project Proposal Submission

The "Submit Your Project Proposal" page on the Questify FYP Portal gives students a simple way to submit their FYP proposals. Students can fill in their Project Title, write a detailed Project Description, and list their Team Members by separating names with commas. They also select the appropriate Project Category from a dropdown menu to classify their project correctly. To complete the submission, students upload their proposal document in PDF format. When everything is entered, then by clicking Submit proposal button will upload proposal to the

system. This page makes sure that all needed information is entered about the project to work further on it.

7.4 Document Upload

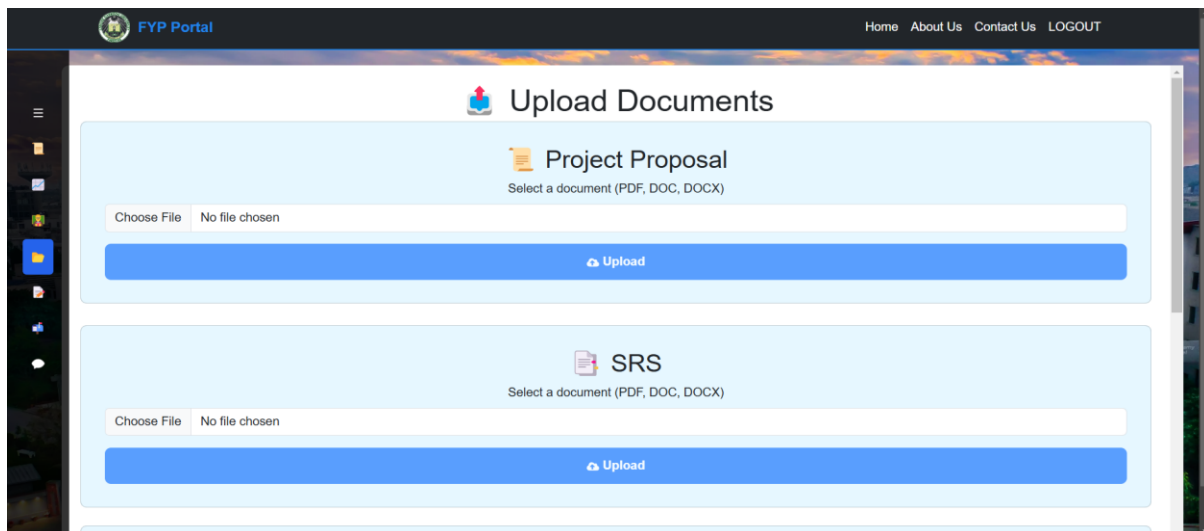


Figure 15: Document Upload Dashboard

The "Upload Document" page on the Questify FYP Portal gives students a simple way to upload their FYP proposal and SRS documents. Students can upload the files pdf, doc, and docx. When the student clicks on the choose file then the student can select file from their personal computer easily.

7.5 Progress Tracking

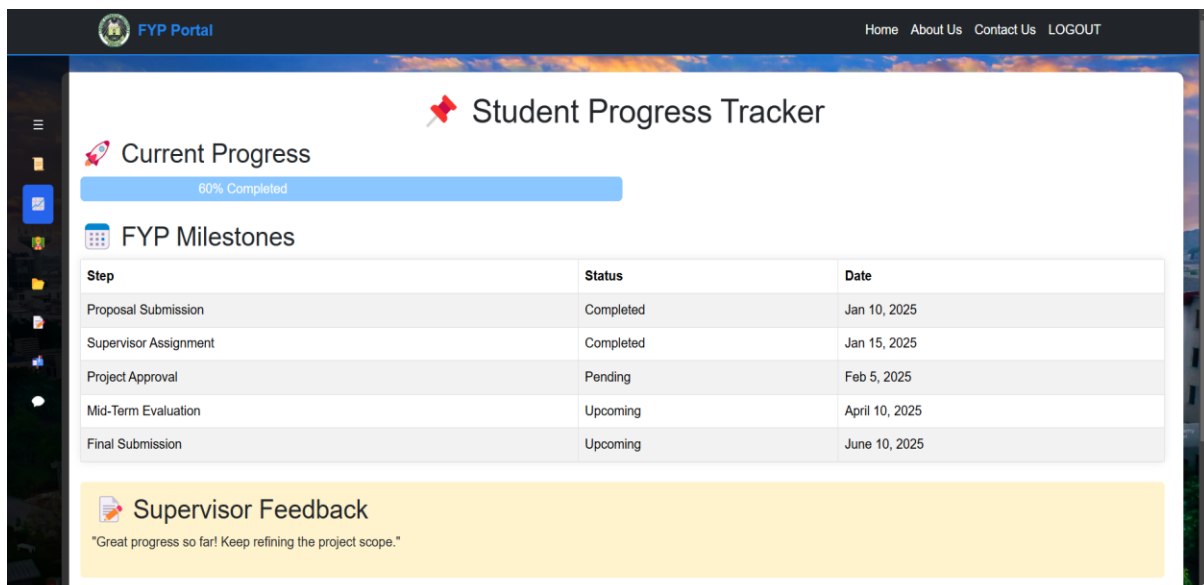


Figure 16: Progress Tracking Dashboard

After uploading documents student can track their progress easily. “Student Progress Tracker” page help the student to check the upcoming milestones. And also track their documents progress. In this way student can easily complete their work on time without any delay. Supervisor also give feedback about any document that student have uploaded.

7.6 Comments & Feedback

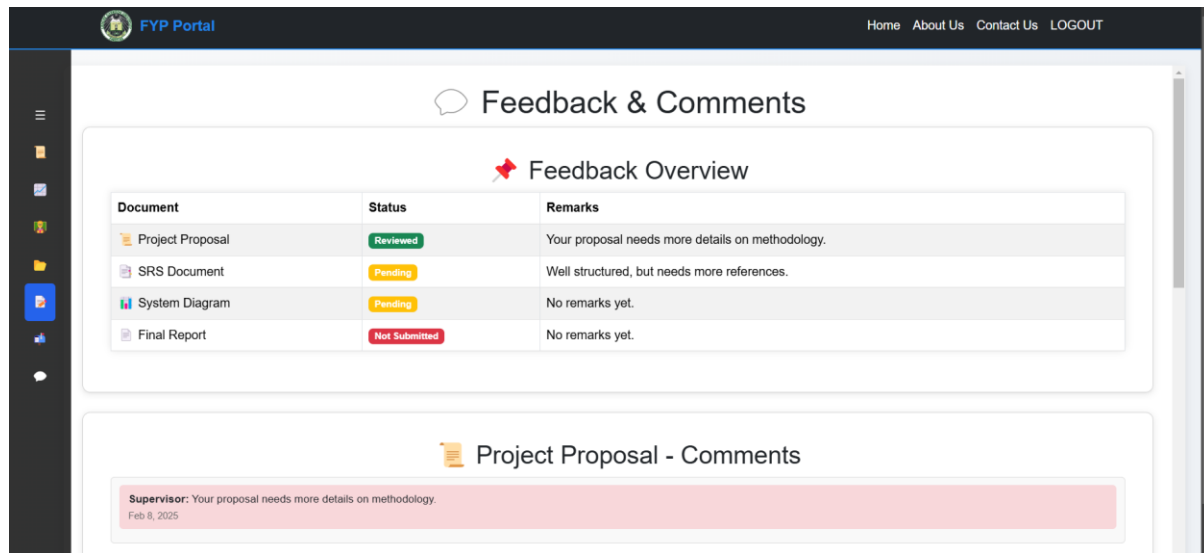


Figure 17: Feedback of documents

The "Feedback & Comments" page on the Questify FYP Portal gives students response about their documents. Student can see the status of their uploaded document. Through the remarks student can see what changing is required in their proposal for approval of their document.

7.7 Chat with Supervisor

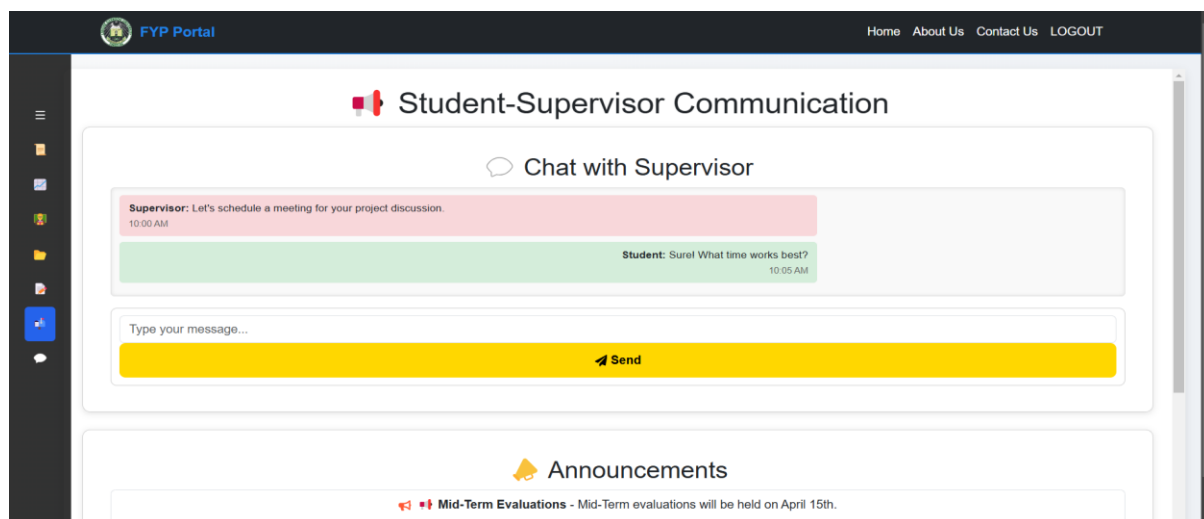


Figure 18: Student and supervisor communication

The “Student-Supervisor Communication” page help the student to discuss any problem with their supervisor. If student can’t get the point of supervisor through the remarks, then student can easily communicate with the supervisor. Also, supervisor can send message to the student.

7.8 AI Chatbot

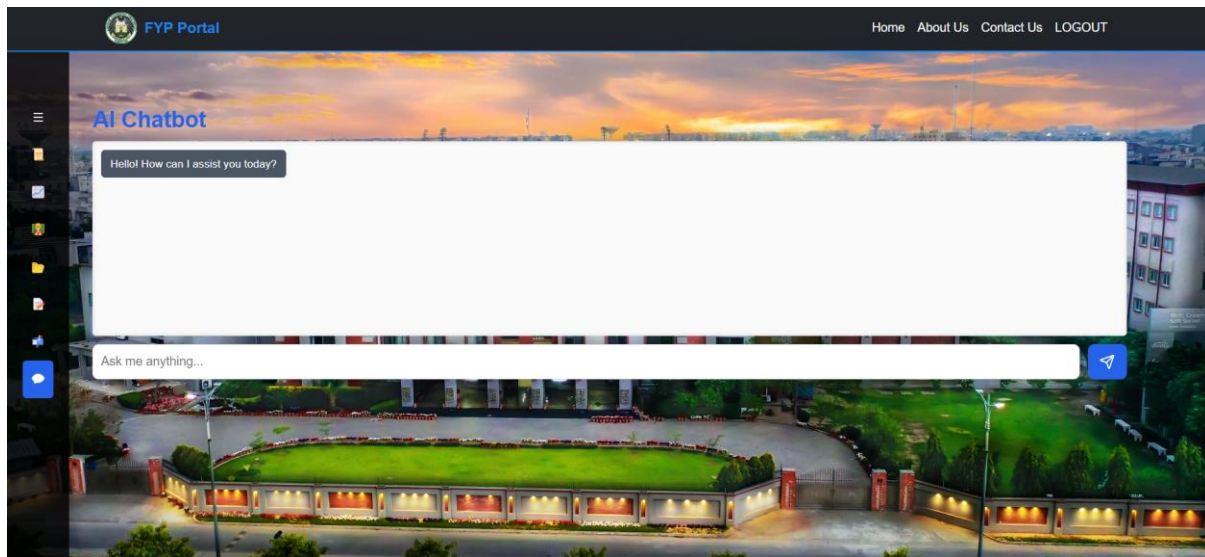


Figure 19: Chat with Chatbot

This page is given in student dashboard, it allows students to clear their short and little quires about their FYP’s, this bot will answer student by get help from different Ais like Chatgpt, Gemini, Lama, Deepseek, Blackbox, and some others. This feature will help students to clear their problem as soon as possible, direct question direct answer.

7.9 Supervisor Dashboard

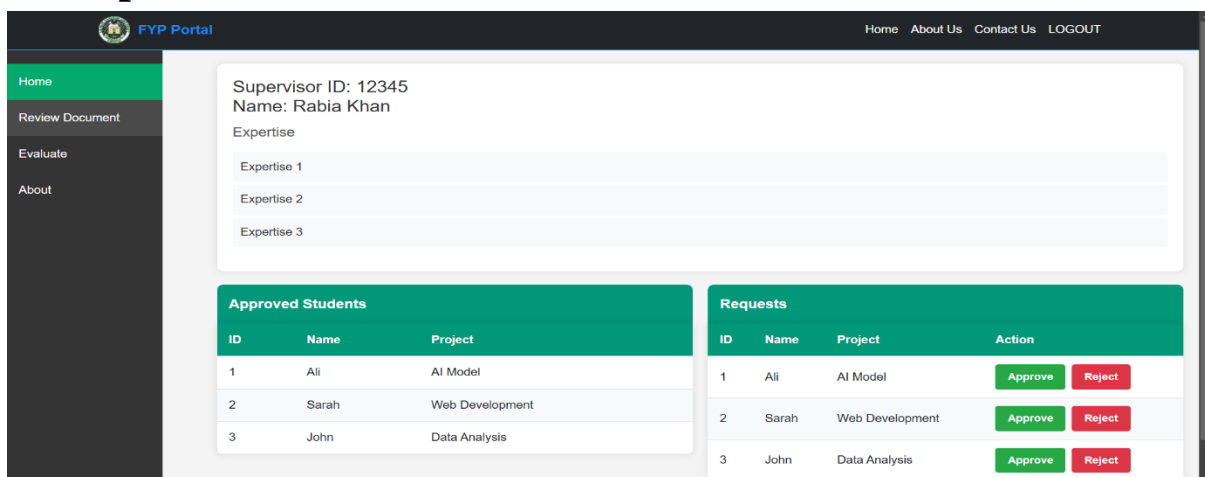


Figure 20: Supervisor Dashboard

The “Supervisor Dashboard” contains the supervisor id, name and their expertise. Supervisor can see the requests of students and approve or reject the requests. And this page also contains the list of approved students. Supervisor also evaluate the students through their portal.

CONCLUSION AND FUTURE WORK

8.1 Conclusion

The Questify FYP Portal addresses critical challenges in Final Year Project management by providing a unified, secure, and user-centric platform for students, supervisors and administrators. Built on the MERN stack, the system streamlines proposal submissions, manual supervisor assignments, progress tracking and rubric-based evaluations while enforcing robust security protocols like JWT authentication. By prioritizing modularity, scalability and role-based workflows, the portal eliminates fragments communication, reduce administrative overhead and ensures transparency across academic processes.

The current implementation successfully bridges gaps in traditional FYP systems, offering students intuitive tools for proposal drafting and milestone tracking, supervisor's centralized dashboards for evaluations and administrators oversight capabilities for conflict resolution and reporting.

Ultimately, the Questify FYP Portal is more than a tool, it is a catalyst for academic excellence, empowering students to focus on creativity, supervisors to mentor effectively and institutions to streamline governance.

8.2 Future Work

8.2.1 Automated supervisor allocation

To streamline administrator tasks, the portal could introduce automated supervisor allocation, matching students with supervisors based on expertise, workload and student preferences this system would analyze proposal keywords (e.g., "AI" or "cyber security") using machine learning algorithms. Tools like python's scikit-learn and NLP libraries could power this feature, reducing administrator effort while ensuring fairness in assignment.

8.2.2 Internal & External Collaboration

For internal users, dashboards could be added for department heads to monitor FYP trends and resource allocation, integration with university ERP/LMS system for real-time data. External collaboration could involve inviting industry experts as evaluators via a secure portal with time-bound access, enriching feedback quality and industry relevance.

8.2.3 AI-Driven project suggestions

An AI-powered recommendation engine could suggest project ideas aligned with student interest and skills. By training models on past projects, research papers, and trends. The system could spark creativity and reduce proposal drafting time. Tools like TensorFlow and OpenAI API could drive this feature.

8.2.4 Additional enhancements

Mobile app integration: A native app (built with React Native or Flutter) could offer notifications, document uploads and milestones tracking.

AI Plagiarism checker: Detect copied content using Turnitin API or custom NLP models.