



INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE

BS(CS) Thesis

STUDENT GUIDE

**THESIS SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE
REQUIREMENT OF THE UNIVERSITY OF SINDH, FOR THE AWARD OF
BACHELOR OF SCIENCE IN COMPUTER SCIENCE DEGREE.**

December 2023



INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE

BS(CS) Thesis

STUDENT GUIDE

By

1. **Labheshwar Sharma (Group Leader)** **2K20/CSM/50**
2. **Shafique Ahmed** **2K20/CSM/89**
3. **Khalilullah** **2K20/CSM/49**

December 2023

INSTITUTE OF MATHEMATICS & COMPUTER SCIENCE



University of Sindh, Jamshoro

CERTIFICATE

This is to certify that the project entitled “**Student Guide**” has been carried out by **Labheshwar, Shafique Ahmed** and **Khalilullah**, during the academic year 2023 as a partial requirement for the degree of Bachelor of Science in Computer Science (BSCS).

Project Supervisor

Dr. Yasir Arfat Malkani
Professor
Institute of Mathematics & Computer Science,
University of Sindh, Jamshoro Pakistan

Director

Prof. Dr. Abdul Wasim Shaikh
Director
Institute of Mathematics & Computer Science,
University of Sindh, Jamshoro Pakistan

DECLARATION

This thesis is our original work and has not been ever submitted, in whole or in part for a degree at this or any other university. Nor it contain, to the best of our knowledge and belief any material published or written by any other person, except as acknowledged in the text.

Labheshwar Sharma

Shafique Ahmed

Khalilullah

COPY RIGHTS

We here by authorize University of Sindh, Jamshoro to supply of our thesis to individual or libraries for study purposes only.

However, no any part of this thesis or the information contained therein may be included in a publication or referred in any publication without prior written permission of the authors. Any reference must be fully acknowledged.

Labheshwar Sharma

Shafique Ahmed

Khalilullah

DEDICATION

**Dedicated to the dreamers and innovators who strive to make a difference
and to all those individuals who believe in the power of technology to bring
people together and expand horizons and make the Earth a better place.**

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and appreciation to our Project Supervisor, Dr. Yasir Arfat Malkani, for his unwavering support and invaluable guidance throughout the development of our project and this thesis. Dr. Malkani's expertise, dedication, and mentorship have been instrumental in shaping the success of our endeavor. His constructive feedback, patience, and constant encouragement have played a pivotal role in ensuring the quality and completion of this project.

We would also like to extend our appreciation to all the individuals, the internet and the resources who and which have contributed to this project in various ways, from providing valuable insights to offering technical assistance. Your collective efforts have been indispensable, and We are truly thankful for your contributions.

ABSTRACT

This thesis serves as a comprehensive blueprint for the design and implementation of a Web App based student portal. The central goal of this project is to create an innovative digital platform that unites university students, teachers, and alumni in a dynamic online community. The student portal has been meticulously crafted to facilitate seamless connectivity, knowledge exchange, and mutual guidance among its users, enhancing the educational experience and fostering a sense of unity within the academic environment.

This thesis is intended for a diverse readership, including academic institutions, software developers, educators, and technology enthusiasts interested in the development and implementation of digital solutions for educational and social purposes. It is recommended for those seeking to understand the technical and functional aspects of a Web App based student portal.

The full document provides an in-depth exploration of the project's requirements, specifications, and the underlying technology used in the creation of the project. It offers valuable insights into the potential benefits of such a platform for university communities, emphasizing the importance of collaborative learning and knowledge sharing in the digital age.

TABLE OF CONTENTS

DECLARATION	III
COPY RIGHTS	IV
DEDICATION	V
ACKNOWLEDGEMENTS.....	VI
ABSTRACT	VII
TABLE OF CONTENTS	VIII - X
LIST OF TABLES	XII
LIST OF FIGURES	XIII
ABBREVIATIONS	XIII
CHAPTER 1.....	1
INTRODUCTION	1
1.1 MOTIVATION	1
1.2 CONTRIBUTIONS OF THE THESIS	2
1.3 PROBLEM STATEMENT	2-3
1.4 SCOPE OF THE THESIS	4
1.5 STRUCTURE OF THE THESIS	4
CHAPTER 2.....	5
2.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIRMENTS	5
2.1.1 FUNCTIONAL REQUIRMENTS	5-6
2.1.2 NON-FUNCTIONAL REQUIREMENTS	7-8
2.2 SOFTWARE PLATFORMS	8
2.2.1 FRONTEND TECHNOLOGIES	8-9
2.2.2 BACKEND TECHNOLOGIES	10-11
2.2.3 DATABASE	11
2.2.4 DEVELOPMENT TOOLS	12
2.3 SERVICES	13
2.4 GOAL	13
2.5 SUMMARY.....	14
CHAPTER 3.....	15
RESEARCH METHODOLOGY	15
3.1 INTRODCUTION	15

3.2 RESEARCH DESIGN.....	16
3.3 DEVELOPMENT METHODOLOGY	17-21
3.4 USER RESEARCH AND ANALYSIS	21-23
3.5 DATA COLLECTION METHODS	23
3.6 APPLICATION FEATURE EVALUATION	24
3.7 LIMITATION AND CHALLENGES	24
3.8 SUMMARY.....	24
CHAPTER 4.....	25
RESULTS AND DISSCUSSION	25
4.1 OVERVIEW OF DEVELOPMENT RESULTS	25
4.1.1 FEATURE IMPLEMENTATION AND USER ENGAGMENT	25
4.1.2 USER INTERFACE AND EXPERIENCE	26
4.2 DISCUSSION OF RESULTS.....	26
4.2.1 ANTICIPATED FEATURE EFFECTIVENESS	26
4.2.2 CHALLENGES AND SOLUTIONS	27
4.3 USER FEEDBACK ANALYSIS.....	27
4.3.1 EXPECTED POSITIVE RESULT	27
4.3.2 POTENTIAL AREAS FOR IMPROVEMENT	27
4.4 FUTURE DIRECTIONS AND POTENTIAL IMPACT	28
4.4.1 BROADER EUDCATIONAL TECHNLOGY IMPLICATIONS	28
4.4.2 PLANNED ENHANCEMENT AND EXPANSION	28
4.5 SUMMARY.....	28
CHAPTER 5.....	29
VISUAL FUNCTIONALITY – A PEEK INTO THE PORTAL	29
5.1 LANDING PAGE	29
5.2 UNIVERSITY REGISTRATION	30
5.3 SUPERADMIN APPROVAL	30-31
5.4 APPROVAL EMAIL	31
5.5 ADMIN PASSWORD CHANGE	32
5.6 USER SIGN-UP	32-33
5.7 USER LOGIN BEFORE APPROVAL.....	33
5.8 ADMIN USER MANAGEMENT	34
5.9 CLASSROOM CREATION.....	34
5.10 CLASSROOM JOINING AND APPROVAL	35
5.11 CLASSROOM ANNOUNCEMENT	35

5.12 ASSIGNMENT CREATION	35
5.13 STUDENT POST	36
5.14 UNIVERSITY PUBLIC POST	36
5.15 COURSE RECOMMENDATION	37
5.16 USER PROFILE	37
5.17 CHAT AND CONVERSATION	38
5.18 JOB CREATION AND LISTING	38-39
5.19 POST REPORT	39
5.20 ADMIN DASHBOARD.....	40
CHAPTER 6.....	41
CONCLUSION AND FUTURE DIRECTIONS	41
6.1 CONCLUSION	41
6.2 REFLECTION ON THE DEVELOPMENT PROCESS	42
6.3 FUTURE DIRECTIONS	42
6.3.1 INTEGRATION OF ADVANCE TECHNOLOGIES	42
6.3.2 EXPANSION OF USER BASE AND PARTNERSHIPS	43
6.3.3 CONTINOUS IMPROVEMENT AND ADAPTATION	43
6.4 FINAL TOUCHES	43
REFRENCES	44-45

LIST OF TABLES.

TABLE 1.1 COMPARISON TABLE.....	3
TABLE 3.1 SURVEY TABLE	22

LIST OF FIGURES

FIGURE 2.1 NEXT.JS AS COMPARED TO THE REACT.JS	8
FIGURE 2.2 NEXT.JS APP WITH AND WITHOUT REDUX.....	9
FIGURE 2.3 NODE.JS ILLUSTRATION	10
FIGURE 3.1 AGILE METHOD LIFE CYCLE	17
FIGURE 2.2 ERD OF THE PROJECT.....	18
FIGURE 5.1 LANDING PAGE	29
FIGURE 5.2 UNIVERSITY REGISTRATION	30
FIGURE 5.3 SUPERADMIN APPROVAL	31
FIGURE 5.4 APPROVAL EMAIL	31
FIGURE 5.5 ADMIN PASSWORD CHANGE	32
FIGURE 5.6 USER SIGN-UP	33
FIGURE 5.7 USER LOGIN BEFORE APPROVAL.....	33
FIGURE 5.8 ADMIN USER MANAGEMENT	34
FIGURE 5.9 CLASSROOM CREATION	34
FIGURE 5.10 CLASSROOM JOINING AND APPROVAL	35
FIGURE 5.11 CLASSROOM ANNOUNCEMENT	35
FIGURE 5.12 ASSIGNMENT CREATION	35
FIGURE 5.13 STUDENT POST	36
FIGURE 5.14 UNIVERSITY PUBLIC POST	36
FIGURE 5.15 OURSE RECOMMENDATION	37
FIGURE 5.16 USER PROFILE	37
FIGURE 5.17 CHAT AND CONVERSATION	38
FIGURE 5.18 JOB CREATION AND LISTING	39
FIGURE 5.19 POST REPORT	39
FIGURE 5.20 ADMIN DASHBOARD	40

ABBREVIATIONS

MERN	MongoDB, Express.js, React.js, Node.js
AWS	Amazon Web Services
SDK	Software Development Kit
WS	Web Socket
HTTP	Hyper-text transfer protocol
S3	Simple Storage Service
MUI	Material User Interface
SDK	Software Development Kit
API	Application Programming Interface
JWT	Json Web Token
JS	JavaScript
CSS	Cascading Style Sheet
HTML	Hyper-text Markup Language
IO	Input Output
CORS	Cross-origin Resource Sharing
ERD	Entity Relation Diagram
IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

In this chapter we provide motivation and contributions of this thesis and at the end of the chapter, we present structure of the rest of the thesis.

1.1 MOTIVATION.

In the fast-paced world of technology, students face increasing competition and the need for mentors and guidance on new technological trends. This ensures they can effectively compete in the real world. In this context, a platform where students can connect with their colleagues and teachers within an organized environment becomes crucial. It's also a place where they can explore courses and find jobs seamlessly. That would truly be a helpful solution.

Moreover, such a platform serves as a bridge between academia and industry, providing students with practical insights and real-world exposure. By creating a space where students can not only learn from their peers and educators but also discover relevant courses and job opportunities, we empower them to make informed decisions about their academic and professional journey. This holistic approach aims to prepare students for the dynamic landscape of the tech industry, fostering collaboration, knowledge-sharing, and career development.

1.2 CONTRIBUTIONS OF THE THESIS

In this project, we've created a platform using latest tech trends. It's designed to help university students, teachers and admin persons, to share what they know, manage university and support each other. Our work contributes by making it easier for people in universities to work together and learn from one another. We've used technology to bring people closer and help them make the world a better place through education and cooperation.

1.3 PROBLEM STATEMENT.

In the present educational scenario, students often find themselves juggling between various platforms to meet their essential needs. Job searches are scattered across LinkedIn and Indeed, learning resources are fragmented between Udemy, Coursera, and YouTube, and communication with faculty is maintained through university portals and Google Classroom. These are fundamental requirements for any student. Our vision is to streamline these necessities into one unified platform, eliminating the need for constant platform-switching.

Moreover, we aspire to go beyond the basics. Our platform not only consolidates these features but also provides additional support by connecting students with mentors, offering guidance crucial for their professional journey. We aim to simplify and enhance the student experience, making it a comprehensive and user-friendly space where they can seamlessly navigate through their academic and career endeavors.

A Comparison Table between Students' Guide and other platforms.

Feature	Student Guide	Google Classroom	University LMS	LinkedIn	Udemy
Classroom Management	<input checked="" type="checkbox"/> Create & join classrooms	<input checked="" type="checkbox"/> Create & join classes	<input checked="" type="checkbox"/> Enroll in courses	<input type="checkbox"/> Not applicable	<input type="checkbox"/> Not applicable
Teacher-Student interaction	<input checked="" type="checkbox"/> Teacher-student chat	<input checked="" type="checkbox"/> Teacher-student chat	<input checked="" type="checkbox"/> Forum discussions, chat	<input type="checkbox"/> Interaction outside jobs	<input type="checkbox"/> Q&A within courses
Post Sharing	<input checked="" type="checkbox"/> Share public & classroom posts	<input checked="" type="checkbox"/> No public post sharing	<input type="checkbox"/> file sharing within courses	<input type="checkbox"/> Share job postings & articles	<input type="checkbox"/> Share courses & announcements
Course Recommendation	<input checked="" type="checkbox"/> Course Recommendations	<input type="checkbox"/> Limited course recommendations	<input checked="" type="checkbox"/> Course recommendations within LMS	<input type="checkbox"/> Not applicable	<input checked="" type="checkbox"/> Search based on interests
Student Portal	<input checked="" type="checkbox"/> Dashboard for courses, & activities	<input type="checkbox"/> No dedicated student portal	<input checked="" type="checkbox"/> Course enrollment, comm. portal	<input checked="" type="checkbox"/> Profile, connection, & job recommendations	<input checked="" type="checkbox"/> Manage courses & progress

Figure 1.3: Comparison Table.

1.4 SCOPE OF THE THESIS.

The scope of this thesis encompasses the development and implementation of a comprehensive platform tailored for university students, teachers, and administrators. The platform aims to facilitate seamless communication, knowledge-sharing, and collaboration within university communities. It covers features such as user roles (student, teacher, admin, superadmin), a dashboard providing key insights, university and user management, job announcements, classroom creation, assignment submissions, course recommendations, and integration of various technologies like Next.js, Node.js, Express.js, MongoDB, Socket.io, S3Bucket (AWS), Nodemailer, YouTube API, g2plots, Tailwind CSS, and MUI. The thesis further explores the potential impact of the platform on enhancing educational experiences and fostering a supportive learning environment.

1.5 STRUCTURE OF THE THESIS

The Structure of this thesis is architecture in the given sequence.

CHAPTER 1: INTRODUCTION

CHAPTER 2: BACKGROUND

CHAPTER 3: RESEARCH AND METHODOLOGY

CHAPTER 4: RESULTS AND DISCUSSION

CHAPTER 5: VISUAL FUNCTIONALITY

CHAPTER 6: CONCLUSIONS AND FUTURE DIRECTIONS

CHAPTER 2

BACKGROUND

In this chapter, we present an overview of the Functional and Non-Functional Requirements that define the scope and capabilities of our project. Additionally, we delve into the software platform and tools chosen for the development of our Application.

Furthermore, we introduce the services integrated into our project, emphasizing their role in enhancing user experience and functionality

2.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS.

Functional requirements encompass the specific features and interactions a system must deliver to fulfill user needs, detailing the essential tasks it should perform. They focus on the "what" – outlining the functionalities and behaviors expected. Non-functional requirements, on the other hand, address the broader qualities of a system, including aspects like performance, usability, and security. They emphasize "how" the system should perform and under what conditions. Together, these requirements ensure a comprehensive understanding of both the specific functionalities and the overall characteristics that contribute to the effectiveness and quality of the system.

2.1.1 FUNCTIONAL REQUIREMENTS.

1. User Authentication:

- Users should be able to create accounts and log in securely using modern token-based authentication using JWT.
- Different user roles (students, teachers, admin, superadmin) should be supported.
- Admin manages the University specific users and dashboard while the superadmin manages all the universities and its users.

2. Profile Management:

- Users can create and manage their profiles.
- Ability to update personal information, academic history, and areas of expertise.

3. Knowledge Sharing:

- Users should be able to share educational resources, articles, and study materials.
- A discussion forum for asking and answering questions among users.

4. Announcements:

- Admins can share university wide announcements.

5. Connection Features:

- Users can connect with each other, forming a network within the portal.
- Private messaging/chat functionality to facilitate direct communication.

6. Classroom Management:

- Teachers can create their classrooms within the website and link will be shared among the students using which students can enroll in the classroom.
- Teacher can approve the enrolled users and after which Teacher can share class related announcements and assignments that will notify all the students.

7. Course Recommendations:

- Users will be recommended courses according to their interests and skills.
- Recommendation system for suggesting connections and relevant content.

8. Job Finding:

- All users will be able to create and find jobs within the platform.

2.1.2 NON-FUNCTIONAL REQUIREMENTS.

1. Security:

- User data should be stored securely.
- Authentication and authorization mechanisms must be robust.
- User passwords are hashed using SHA-256 hashing algorithm.
- All the routes are protected by roles, i.e. only admin can access admin routes, any other user will encounter get 401 – unauthorized error while accessing those routes.

2. Scalability:

- The system should be able to handle a growing number of users and content.
- Performance should not significantly degrade as the user base expands as we are using Node.js as backend which can handle multiple requests very easily using its asynchronous nature.

3. User Interface:

- The interface should be user-friendly and intuitive.
- Responsiveness for a seamless experience across different devices.
- Visual attracting design.

4. Reliability:

- The system should be available and reliable, with minimal downtime.
- Regular backups to prevent data loss.

5. Compatibility:

- The portal should be compatible with various web browsers and devices.
- Accessibility features for users with diverse needs.

6. Performance:

- Efficient loading times for pages and resources.
- Optimized use of server resources to ensure a smooth user experience.

9. Compliance:

- Compliance with data protection and privacy regulations.
- Adherence to web standards for interoperability.

2.2 SOFTWARE PLATFORMS.

Our project is developed using a modern and versatile tech stack that encompasses both frontend and backend technologies, along with the modern best practices and essential development tools. The following software, languages, and tools have been employed in the creation of our main Project:

2.2.1 Frontend Technologies:

1. **Next.js:** Utilized as the primary framework for building a fast and efficient frontend framework which is built on the top of React, giving a better user as well as developer experience.

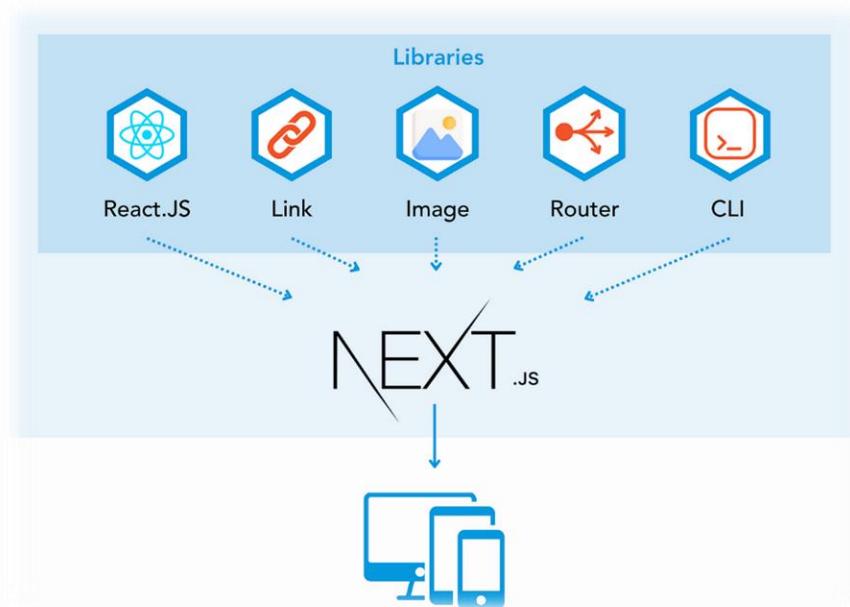


Figure 2.1. Next.js as compared to the React.js

2. TailwindCSS: Employs a utility-first CSS framework for streamlined and responsive styling. Using vanilla CSS has lot of hassle like frequently switching between HTML and CSS file, therefore for that we used TailwindCSS so that we can write out HTML, CSS and JS in one place.

3. MUI (Material-UI): Google's Component Library, used to make the project more scalable, and to follow a divine theme. Implements React components following the Material Design guidelines for a cohesive and user-friendly interface.

4. g2Plots: Integrated for data visualization and charting capabilities.

5. Redux: Utilized for state management, ensuring a predictable and efficient data flow. It helps to have a global store from where the data could be get and set directly instead of moving the data across components.

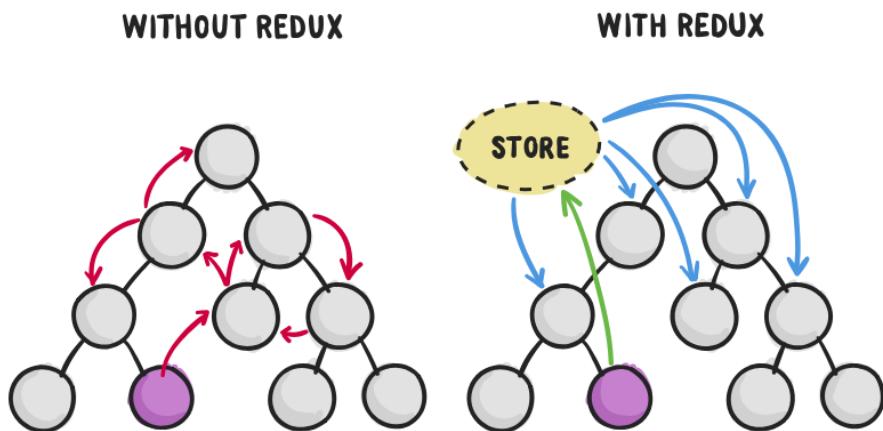


Figure 2.2. Next.js app with and without Redux.

6. Formik: Employed for form management and validation on the frontend by mashing up with Yup.

7. Socket.io-client: Used this library to call web-socket APIs for real-time, bidirectional and event-based communication.

2.2.2 Backend Technologies:

1. NodeJS: Serves as the runtime environment for executing server-side JavaScript code. Preferred for its asynchronous and non-blocking architecture, allowing applications to efficiently handle concurrent operations without waiting for each to complete. This event-driven model enhances scalability and responsiveness, making Node.js well-suited for real-time applications and scenarios with a high volume of simultaneous connections.

Non-Blocking I/O

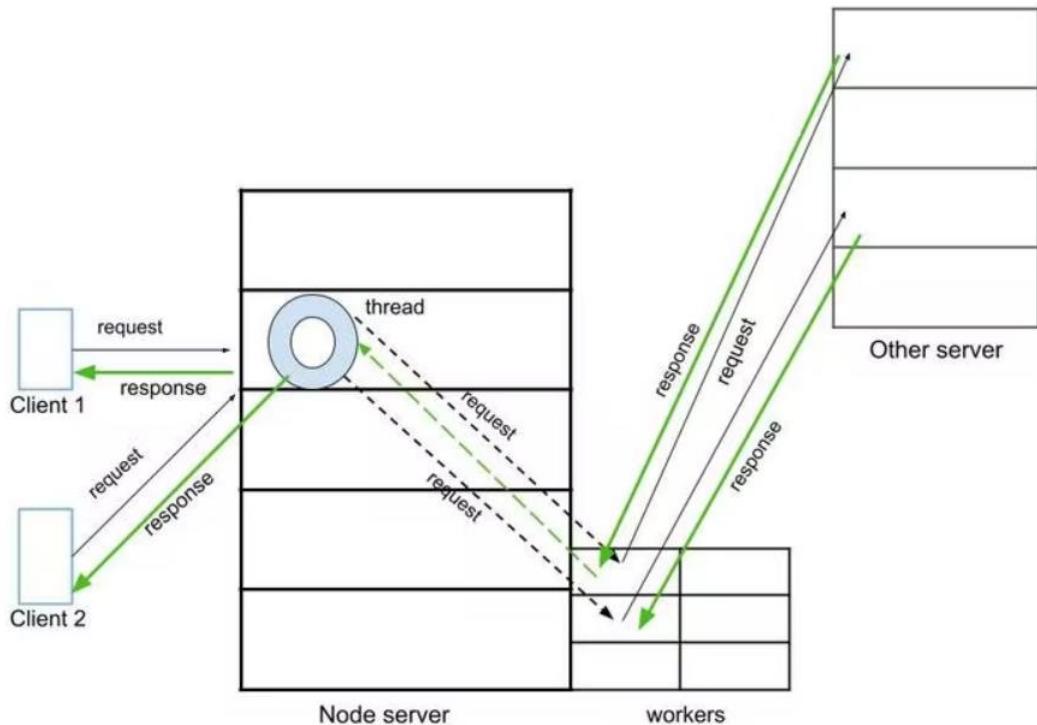


Figure 2.3. Illustration shows Node.js architecture and its non-blocking event handling approach.

2. ExpressJS: Used as the backend framework, ExpressJS simplifies the creation of robust and scalable APIs, streamlining the development of server-side functionalities.

3. AWS-S3 Bucket: Integrated for seamless communication with Amazon Web Services, facilitating cloud-based functionality. AWS-S3 Bucket enhances the

application's capabilities by providing scalable and secure cloud-based storage solutions for Images and Documents.

4. Socket.io: This JavaScript library facilitates real-time, bidirectional communication between the server and the browser. It enables event-based interactions, crucial for applications requiring instant updates and responsiveness.

5. bcryptjs: Employed for secure password hashing, bcryptjs ensures that user authentication remains airtight, protecting sensitive information from unauthorized access.

6. jsonwebtoken: Implementing JSON Web Tokens, this library ensures secure and efficient user authentication and authorization processes, enhancing the overall security of the application.

7. nodemailer: Integrated for effective email communication, nodemailer enables the application to send notifications and messages, providing seamless user interaction.

8. multer: Utilized for efficient handling of file uploads on the backend, multer enhances the application's capability to manage and process various types of user-generated content.

2.2.3 Database:

MongoDB: MongoDB is a No-SQL Database solution for our application following the key-value architecture for storing versatile data in organized form. Its known for flexibility and scalability in handling diverse data types. MongoDB provides a solution to access data from anywhere in the World thanks to its cloud-storage facility (MongoDB Atlas), ensuring accessibility and reliability, while Compass is used for local database management. It is very beneficial when you need to host the website to a remote server.

2.2.4 Development Tools:

- 1. VS Code:** Chosen as the primary integrated development environment (IDE), VS Code offers a versatile platform for coding and project management. With a rich set of extensions, it supports various programming languages and provides powerful tools for debugging, version control, and syntax highlighting.
- 2. Postman:** Integral to the development process, Postman serves as a comprehensive API testing and debugging tool. Its user-friendly interface allows developers to design, test, and document APIs efficiently. Postman also facilitates collaboration among team members through shared workspaces.
- 3. Git/GitHub:** Essential for version control, GitHub enables collaborative development, allowing multiple developers to work on the project simultaneously. It tracks changes, manages branches, and facilitates seamless integration of new features or bug fixes.
- 4. MongoDB Atlas:** A fundamental tool for real-time testing and validation of the web application. Developers often leverage browsers like Chrome, Firefox, or Safari to inspect elements, monitor network activity, and ensure the application's compatibility across different platforms.

This comprehensive set of technologies and tools collectively contributes to the creation of a robust, user-friendly, and scalable project that meets the diverse needs of our academic community.

2.3 SERVICES.

In this project development, we seamlessly integrated several third-party services to enhance its functionality. Using Google's Gmail API in conjunction with Nodemailer facilitated efficient email notifications, ensuring effective communication within the system. AWS S3-Bucket played a pivotal role in providing robust cloud storage solutions, addressing our project's storage needs.

For course recommendations, we harnessed the power of YouTube's Search API, enriching the user experience with personalized suggestions. The robust MongoDB Atlas served as our database storage solution, ensuring secure and scalable data management. All these components seamlessly coalesce in several thousand lines of JavaScript, symbolizing the intricate yet cohesive orchestration required to bring this project to fruition.

2.4 GOAL.

The goal of our project is to create a friendly and helpful online space for university students, teachers, and administrators. We want to make it easy for people to connect, share what they know, and support each other. Through this portal, we aim to encourage collaboration, provide valuable resources, and offer guidance, making the university experience more enjoyable and beneficial for everyone involved. Our ultimate goal is to build a platform that brings people together, expands horizons, and creates a positive impact in the academic community.

2.5 SUMMARY

The student portal project is designed to create an interactive online space for university students, teachers, and administrators, focusing on functional and non-functional requirements. Key functionalities include secure user authentication, profile management, knowledge sharing, announcements, networking features, classroom management, course recommendations, and job finding. Non-functional aspects emphasize security, scalability, user interface design, reliability, compatibility, and performance optimization.

The project's goal is to create a user-friendly and collaborative platform that enhances the university experience by encouraging knowledge-sharing, providing valuable resources, and offering guidance. Ultimately, the aim is to build a positive and impactful online community for the academic domain.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 INTRODUCTION

In this chapter, we'll dive into the methods we used to build "Student Guide," our web application aimed at enhancing the university experience for students, teachers, and administrators. We're covering everything from the nuts and bolts of the app—like user login, classroom management, and job postings—to the behind-the-scenes aspects that make it reliable and user-friendly.

Our approach mixes different technologies and strategies. We've got Next.js and Tailwind CSS making things look good and work smoothly on the front end, while Node.js and Express.js keep everything running efficiently on the server side. MongoDB comes in for handling our data because of its flexibility.

We also integrated some cool services like AWS for storage and the YouTube API for personalized course recommendations. This chapter will explain why we chose these tools and how they all fit together to make "Student Guide" not just functional, but also a great place to hang out and learn.

So, let's get into the details and see how this all came to life!

3.2 RESEARCH DESIGN

So, how did we go about creating "Student Guide"? Think of it like baking a complex cake with lots of layers – each layer needs to be just right for the whole thing to work. Our recipe for success combined a few different research methods.

First up, we chose a mixed-methods approach. Why? Because our app isn't just about lines of code. It's about people using it – students, teachers, admins. We wanted to understand their needs and experiences, which meant collecting both hard data (like how many people click on a feature) and soft data (like how users feel about the app).

For the mains of the app – things like user login and classroom management – we relied on **quantitative methods**. This meant lots of testing, checking, and rechecking to make sure everything worked seamlessly and could handle lots of users at once.

On the flip side, we needed to get into the heads of our users. So, we used **qualitative methods** like surveys and interviews. We asked students and teachers what they wanted in an app, what frustrated them about other apps, and how we could make their lives easier.

Combining these two approaches gave us a full picture of what we needed to do. And, of course, it helped us make sure "Student Guide" was not just a well-oiled machine but also an app that people would love to use.

So, that's our research design in a nutshell – a bit of this, a bit of that, all coming together to create an app that's as useful as it is user-friendly.

3.3 DEVELOPMENT METHODOLOGY

Alright, let's talk about how we actually built "Student Guide". Imagine putting together a big puzzle; you need the right pieces and a good strategy. That's what our development methodology was all about.

We went with Agile development. Why Agile, you ask? Because it's like having a GPS for a road trip. It keeps us on track, lets us adjust our route as needed, and ensures we're always heading towards our destination – a fully functional, user-friendly app. Agile is all about flexibility, rapid iterations, and constant feedback. We broke down the project into smaller tasks, worked on them in short

AGILE METHODOLOGY

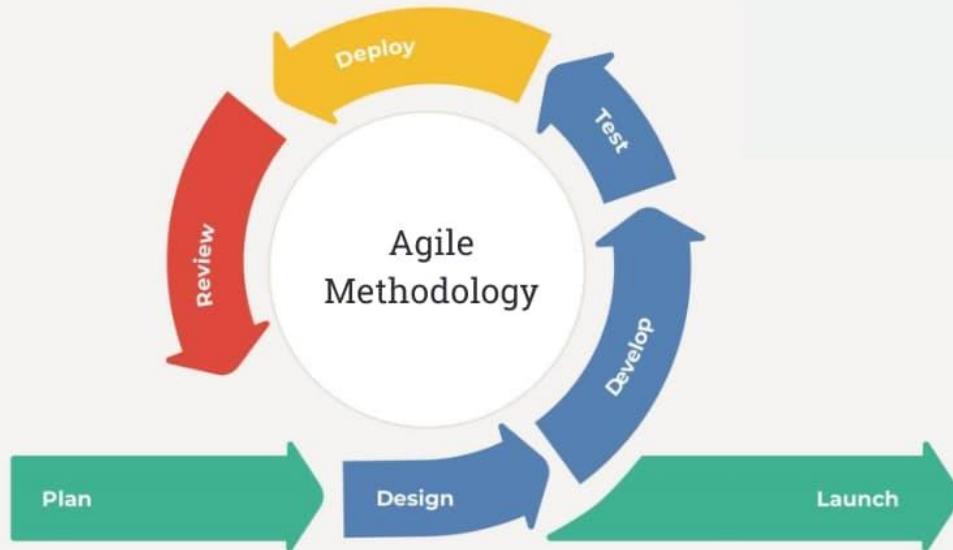


Figure 3.1. Agile Methodology Life Cycle

cycles called sprints, and then reviewed our progress. This way, we could make changes on the fly, adapting to any new challenges or ideas that popped up.

Our Agile journey was divided into multiple sprints, each with distinct objectives and deliverables:

Start and Plan:

- First, we decide what app should do. Make list of all things that app need. This includes creating wireframes, ERD, to go for the best one that fills our needs.

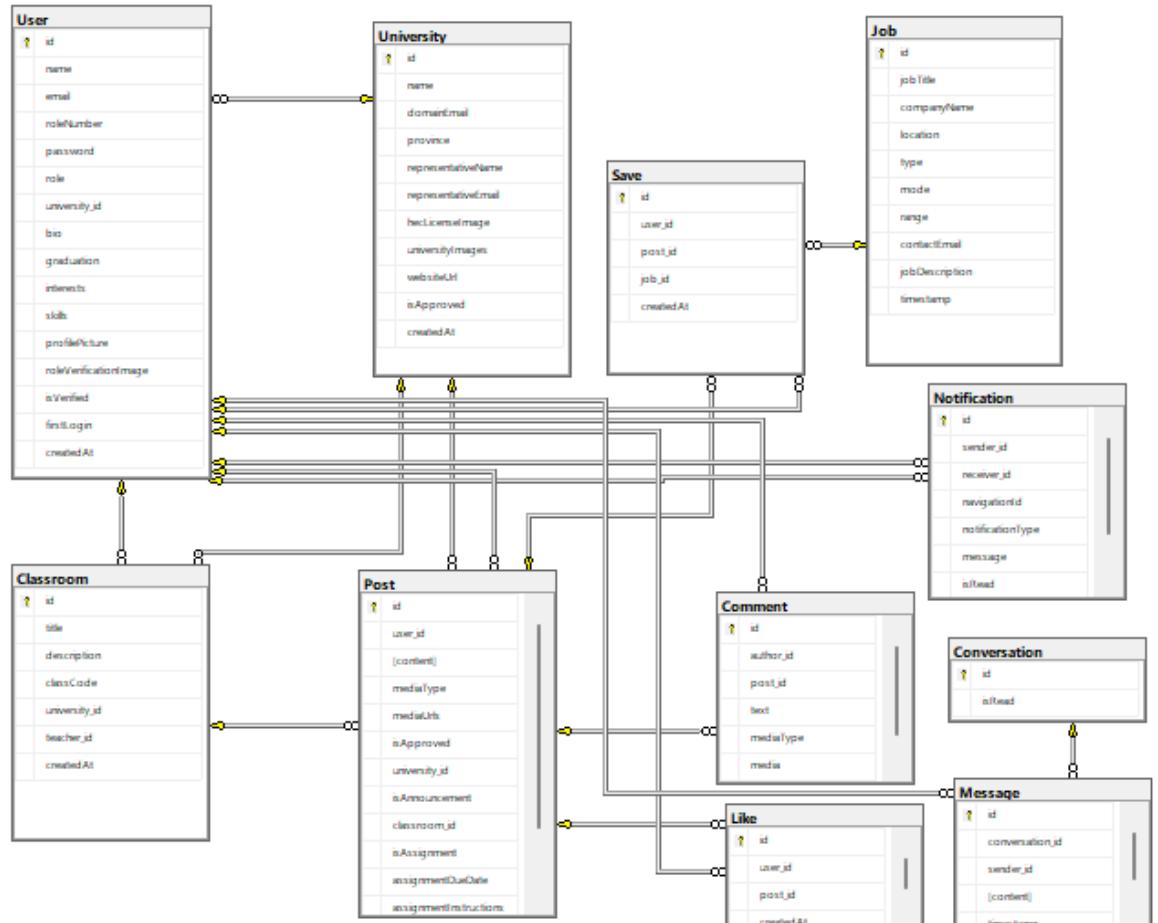


Figure 3.2. ERD (Entity Relation Diagram) of the project

- Setting up the development environment, including tool selection and infrastructure configuration.
 - **VS Code**: We chose VS Code as our IDE as it provides lots of features that increases productivity by miles.
 - **Postman**: Using postman for backend APIs testing since its already ahead of its competition.
 - **Chrome**: Used chrome as default web browser because of its developer tools.

Setting up the project:

- Building the foundational framework of the application using Next.js, ensuring optimal structure and scalability.
- Choosing TechStack for the project and getting started with the basics.
- Setting up the database at MongoDB Atlas and then use the URI in MongoDB Compass to be able to use it locally.
- Initializing git in the project directory.
- Installing basic dependencies for getting started with the backend; i.e. Express, Mongoose, jsonwebtoken, cors.
- Creating next app using using “npx create-next-app” and configuring the basic arguments and setting up the project with the chosen frontend libraries like Redux, Material UI, (TailwindCSS comes default with next).
- Creating basic user authentication APIs using JWT approach.

Feature Implementation: Profile Management and Knowledge Exchange:

- Developing the profile management system, allowing users to maintain personal and academic information.
- Crafting the knowledge exchange module, including resource sharing capabilities and a collaborative discussion forum.

Advanced Functionalities:

- Integrating classroom management tools for educators to create, manage, and engage with students.
- Embedding a course recommendation engine, leveraging the YouTube API to provide personalized learning content.

Job Portal Integration and User Experience Enhancement:

- Incorporating a comprehensive job finding feature, enabling users to post and search for relevant opportunities.
- Iterating on user feedback to refine the user interface and enhance overall user experience.

Testing and User Feedback Incorporation:

- Testing all the Backend endpoints using Postman ensuring that all the routes are protected and only relevant person can access those routes.
- Using frontend testing libraries like Jest to test the Application making sure all the UI components rendering as per requirements.

Finalization:

- Final assessment and refinement of the application, ensuring all functionalities align with project goals.

Technology Stack Rationale:

The technology stack for the project was selected with an emphasis on efficiency, scalability, and user experience:

Frontend: Next.js was chosen for its server-side rendering capabilities, enhancing SEO and performance. Tailwind CSS and MUI were integrated for their extensive customization options and responsive design features.

Backend: Node.js and Express.js provided a robust and efficient server-side architecture, essential for handling high volumes of user traffic and data processing.

Database: MongoDB was selected for its flexibility in handling non-relational data, catering to the diverse data requirements of our application.

Additional Tools: Socket.io enabled real-time communication features, AWS S3 was utilized for scalable cloud storage, and Nodemailer facilitated reliable email communications. These tools were integral in enhancing the application's functionality and user experience.

In summary, the application of Agile methodology, coupled with a strategically chosen technology stack, allowed for a disciplined yet flexible development process. This approach ensured that "Student Guide" was not only built to meet current user needs but also designed to adapt and evolve with future demands.

3.3 USER RESEARCH AND ANALYSIS

To make "Student Guide" the best it can be, it's crucial for me to understand what our users – students, teachers, and admins – really want and need. This aspect of the project is extremely important. I've employed several methods to learn about our users and their preferences.

Understanding Users:

1. Creating User Personas:

- Thought about the different kinds of users, like students, teachers, and admins. Created "personas" – these are like imaginary characters who would use our app. This helped us understand the diverse needs of each user group.

2. Conducting Surveys and Interviews:

- Reached out to real users, conducting surveys and interviews. Asked them about their preferences, the challenges they face, and what they expect from an app like ours. This process gave me a wealth of insights.

Survey Table: University Students' Feedback on Project Features.

Question / Feature	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No	Maybe	Total Responses	% Yes	% No	% Maybe
Need for this Platform.	22	0	0	22	100	0	0
Receive course Recommendations.	22	0	0	22	100	0	0
Job Finding and Announcement	20	0	2	22	90.9	0	9.1
Direct Comm. with Teachers.	19	1	1	21	90.5	4.8	4.8
Share and Upload Files	22	0	0	22	100	0	0
Updates and Announcements	21	0	1	22	95.5	0	4.5
A Modern Dashboard with Analytics.	22	0	0	22	100	0	0
Share Education Related Stuff.	21	0	1	22	95.5	0	4.5
Profile Creation and Modification	21	1	0	22	95.5	4.5	0
Use the Portal Once it's launched?	22	0	0	22	100	0	0
Total	212	2	5		96.8%	0.9%	2.2%

Analysis:

1. Reviewing Data:

- We compiled and analyzed the data from our surveys, interviews, and tests. Looking at this data helped us identify common patterns and user preferences.

2. Making Informed Decisions:

- Using this data, we made informed decisions about which features to include and how to design the app. My focus has always been on creating an application that's both easy to use and genuinely useful for our users.

In conclusion, user research and analysis have been integral to the development of "Student Guide". It has helped me deeply understand our users and create an application that is both practical and appealing to them.

3.4 DATA COLLECTION METHODS

For "Student Guide", we used different ways to collect information. It was important to gather accurate data to understand user needs and app performance.

Surveys: We created surveys and conducted interviews to get direct feedback from users. This helped us understand their expectations and experiences with the app.

Usage Data: By analyzing how users interacted with the app, We could see which features were popular and how they were used. This data was crucial for making improvements.

3.5 APPLICATION FEATURES EVALUATION

Evaluating the features of "Student Guide" was key to ensure they met user needs.

Performance Analysis: I regularly checked the app's performance, looking at loading times and responsiveness to ensure a smooth user experience.

Feature Utilization: By monitoring which features were used most, I could focus on improving popular aspects of the app.

3.6 LIMITATION AND CHALLENGES

Like any project, "Student Guide" had its limitations and challenges.

Resource Constraints: Limited resources meant I had to make tough decisions about which features to prioritize.

Technical Challenges: As a solo developer, I faced technical hurdles that required me to learn new skills and sometimes seek external help.

User Diversity: Catering to the diverse needs of students, teachers, and admins was challenging, as each group had unique requirements.

3.7 SUMMARY

In sum, developing "Student Guide" was a journey filled with learning and growth. Through various data collection methods, I gained deep insights into user needs. Evaluating application features was crucial in refining the app, while ethical considerations ensured respect for user privacy and data security. Despite facing resource and technical limitations, and the challenge of meeting diverse user needs, the project was a valuable experience that taught me much about app development and user engagement.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 OVERVIEW OF DEVELOPMENT RESULTS

The development phase of "Student Guide" culminated in several significant achievements, reflecting both in the functionality and development results of the application.

4.1.1. Feature Implementation and User Engagement:

i) **User Authentication and Profile Management:**

The implementation was successful in a controlled environment, with robust security measures and a user-friendly interface.

ii) **Classroom Management Tools:**

Simulated scenarios demonstrated their functionality, allowing teachers to create and manage classrooms effectively.

iii) **Job Portal and Course Recommendations:**

Preliminary testing indicated these features were operational and ready for real-world user engagement, though their full effectiveness will only be measurable post-launch.

4.1.2. User Interface and Experience:

- i) The application's interface, tested internally and through a small group of beta testers, received positive feedback for its intuitive design and ease of navigation.
- ii) Some minor issues were identified, such as the need for more intuitive access to certain features, which were subsequently addressed.

4.2 DISCUSSION OF RESULTS

This section interprets the results from a development perspective and anticipates how they might translate into real-world usage.

4.2.1. Anticipated Feature Effectiveness:

- i) The high engagement levels in beta testing for features like profile management and classroom tools suggest that these will be well-received upon launch.
- ii) The true effectiveness of the job portal and course recommendations will depend on user engagement patterns, which will be closely monitored post-launch.

4.2.2. Challenges and Solutions:

- i) One anticipated challenge is managing server load and performance under real-world user conditions. Plans for scalable cloud solutions and load balancing are in place to address this.
- ii) Catering to the diverse needs of a broader user base will likely require ongoing adjustments and feature enhancements.

4.3 USER FEEDBACK ANALYSIS

Analyzing user feedback is critical for assessing the application's success and areas for improvement.

4.3.1. Expected Positive Responses:

- i) Initial feedback suggests that users will appreciate the seamless integration of educational and career-oriented features.
- ii) The simplicity and user-centric design of the interface are expected to be major positives.

4.3.2. Potential Areas for Improvement:

- i) Based on beta testing, there may be requests for more interactive elements within the classroom management features.
- ii) The adaptability of course recommendations and job alerts to individual user needs may be areas for future refinement.

4.4 FUTURE DIRECTIONS AND POTENTIAL IMPACT

Reflecting on the broader impact and future potential of "Student Guide".

4.4.1. Broader Educational Technology Implications:

- i) The successful integration of various educational tools in "Student Guide" could set a precedent in the educational technology field.
- ii) Insights from post-launch user behavior will be valuable for informing future developments in similar applications.

4.4.2. Planned Enhancements and Expansions:

- i) Post-launch, plans include implementing AI-driven personalization for course recommendations and expanding the platform to more educational institutions.
- ii) Continual updates and feature additions based on user feedback and technological advancements are envisioned.

4.5 SUMMARY

In summary, the development phase of "Student Guide" has been marked by successful feature implementation, positive feedback on the user interface, and anticipation of its effectiveness in a real-world setting. While challenges are anticipated post-launch, plans are in place to address these proactively. The insights gained from this phase are instrumental in setting the stage for a successful launch and future growth, highlighting the application's potential to make a significant impact in the realm of educational technology.

CHAPTER 5

VISUAL FUNCTIONALITY – A PEEK INTO THE PROJECT.

You know how they say a picture is worth a thousand words? Well, that's what we're doing here. Each screenshot is like a snapshot of a specific part of our portal, showing you how it's designed and what you can do there. From the main dashboard to special features, we're showcasing the user-friendly side of our portal. These pictures are like a tour guide, helping you get a feel for how everything fits together.

5.1 LANDING PAGE

Welcome to our portal's vibrant landing page! Experience a refreshing homepage with easy options to explore jobs, add job listings, and a hassle-free login/signup process. It's your one-stop destination for a user-friendly and rewarding journey. Welcome aboard!

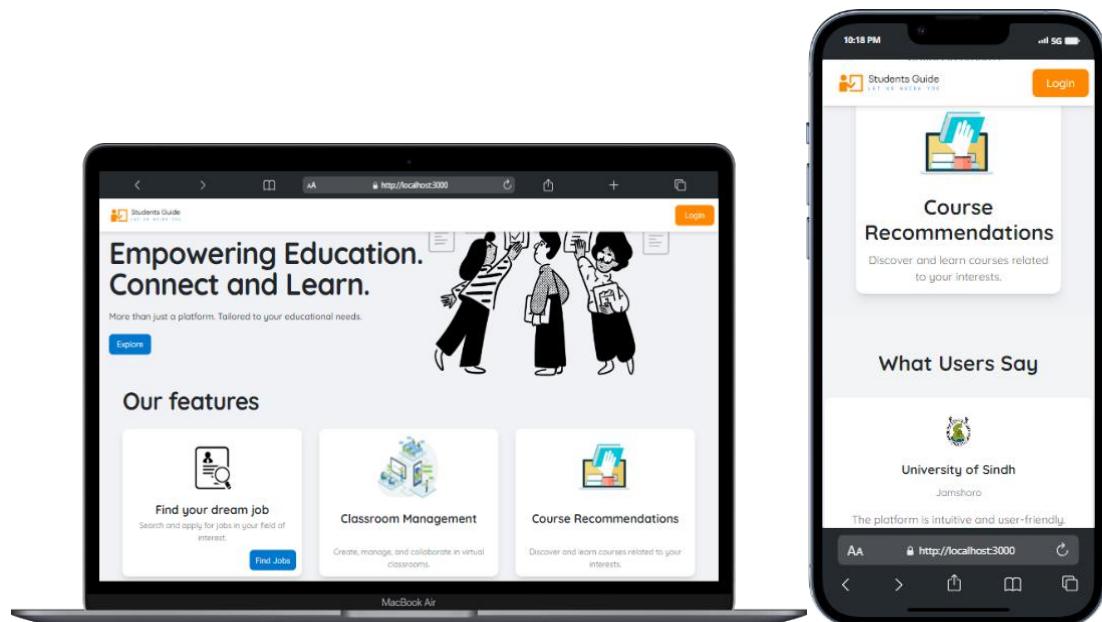


Figure 5.1: Landing page as seen from Desktop and Mobile view.

5.2 UNIVERSITY REGISTRATION

The University Registration Page is where university administrators can sign up quickly and securely. Admins simply need to enter their name and email address, and for added security, provide University registration license for verification that verifies their belonging of the University.

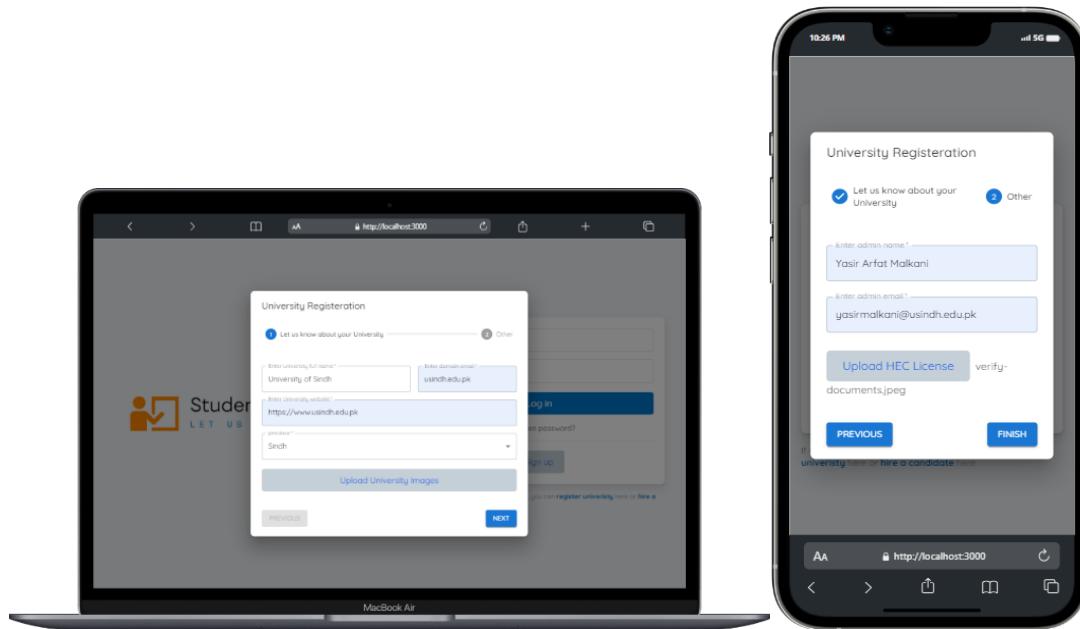


Figure 5.2: University Registration screen as seen from Desktop and Mobile view

5.3 SUPERADMIN APPROVAL

The SuperAdmin Approval Screen is a checkpoint in our registration process. Here, the SuperAdmin carefully verifies university registrations, ensuring accuracy and security. After verification, they have the authority to either approve or decline the registration.

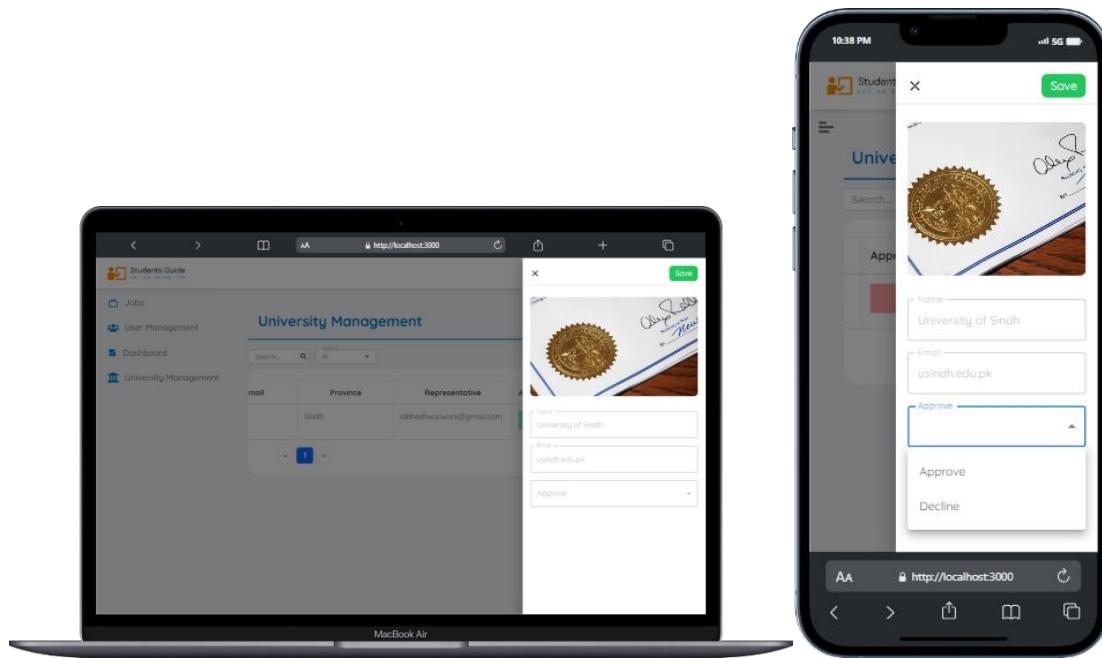


Figure 5.3: SuperAdmin approval screen as seen from Desktop and Mobile view

5.4 APPROVAL EMAIL

Upon approval by the SuperAdmin, the University Admin will receive a confirmation email. This email serves as official acknowledgment, providing assurance that their registration has been successfully accepted. It includes essential details like email and a password

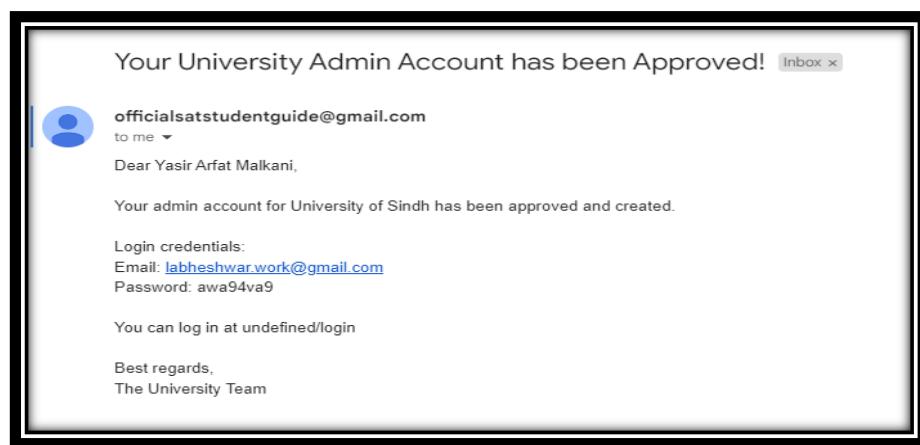


Figure 5.4: User Approval Email Notification.

5.5 ADMIN PASSWORD CHANGE

Upon logging in, the University Admin is greeted by the first screen, where they have the option to change their password. This straightforward feature ensures account security and allows for personalized access to our platform.

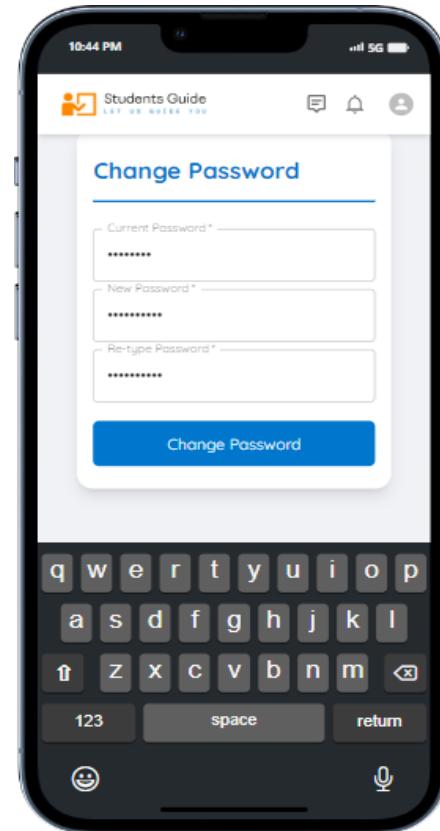


Figure 5.5: Admin password change screen.

5.6 USER SIGN-UP SCREEN

Welcome to our Sign-Up page! Where you have to choose your role as a student or teacher from the drop-down menu. Students, enter your full name, email, set your password, include your roll number, specify your university, and add fields of interest for personalized course recommendations. For teachers, provide your full name, email, set your password, select your university, and upload a document verifying your teaching status.

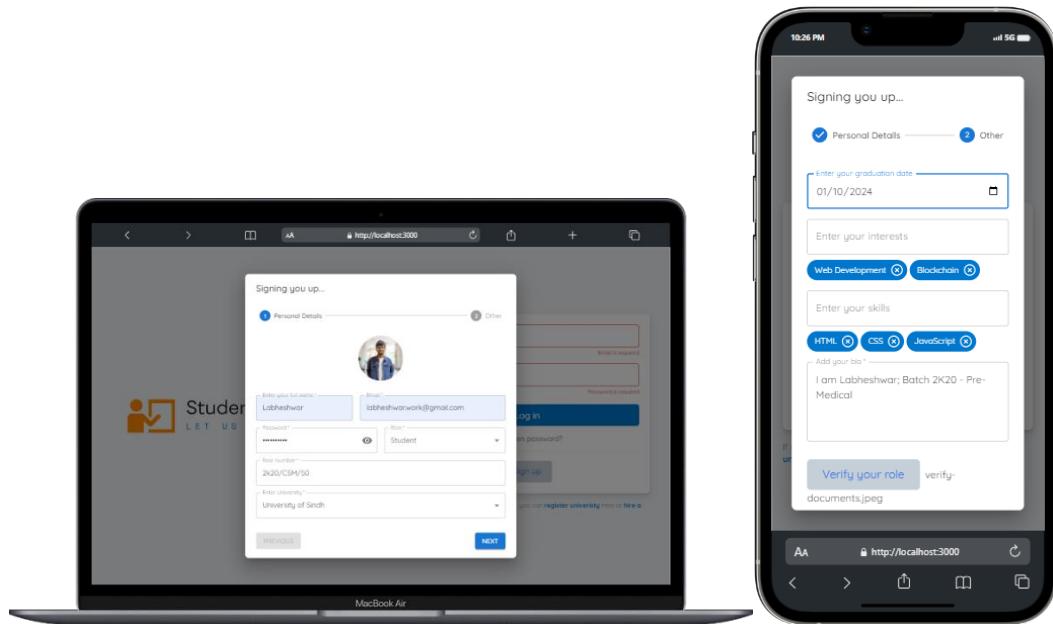


Figure 5.6: Sign-Up screen as seen from Desktop and Mobile view

5.7 USER LOGIN BEFORE APPROVAL

After registration, attempting to log in without account verification leads to an error message: "Your account has not been verified yet." Users must await verification and approval from University Admin (Who was approved by Superadmin) before accessing their accounts, ensuring a secure and controlled login process.

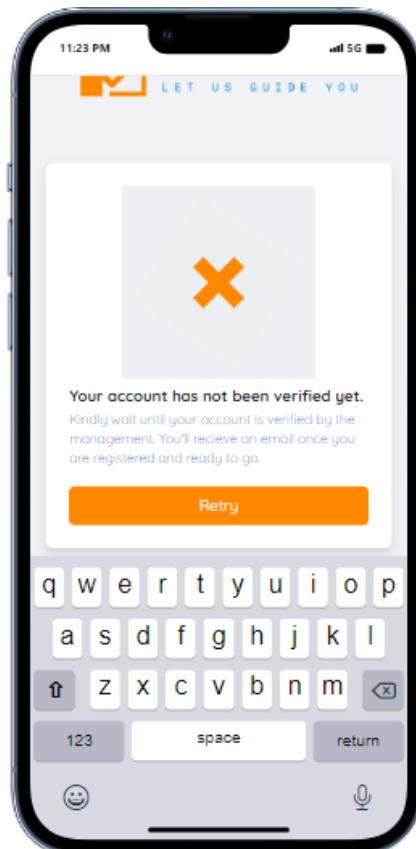


Figure 5.7: User login screen without approval.

5.8 ADMIN USER MANAGEMENT

The User Management Screen empowers the admin to oversee new registrations. Here, admins can thoroughly review details provided by teachers and students. They have the authority to verify and approve new users, ensuring that only legitimate individuals gain access to the platform.

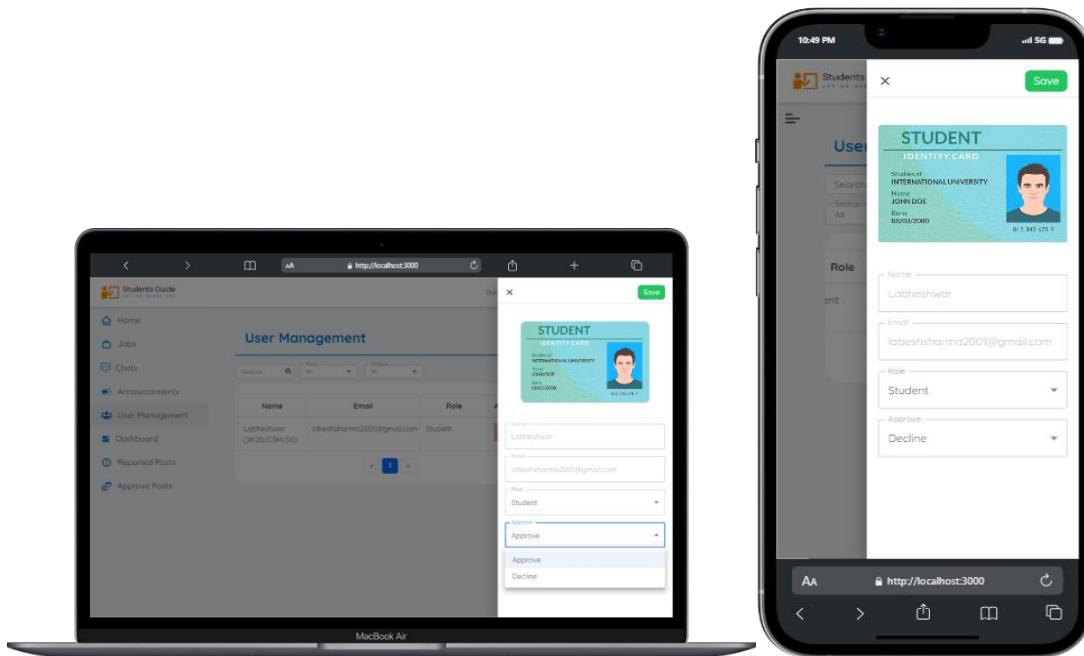


Figure 5.8: User login screen without approval as seen from Desktop and Mobile view

5.9 CLASSROOM CREATION.

Welcome to the Class Creation Screen, which is exclusively for teachers. Here, teachers can easily craft a new classroom by providing a title, description, and a class code according to the subject. It's easy and helps teachers manage their classes smoothly.

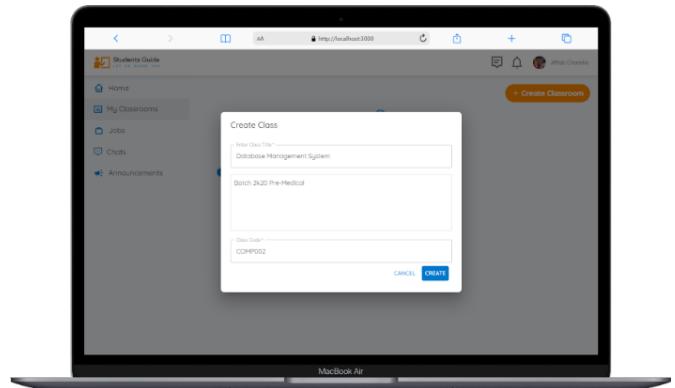


Figure 5.9: Classroom Creation screen.

5.10 CLASSROOM JOINING AND APPROVAL.

After the classroom creation, students can request to join a class by using the classroom link. After sending the request, the teacher can review and approve it. It's an easy way for students to join classes and start learning!

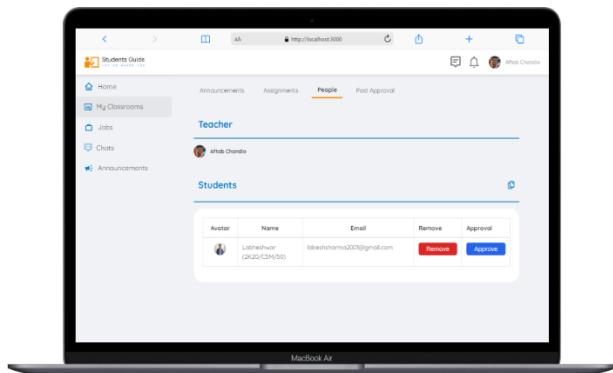


Figure 5.10: Student Approval screen.

5.11 CLASSROOM ANNOUNCEMENT.

Introducing the Class Announcement Screen! Where Teacher and Students (after approval by Teacher) can share important updates within the class, and students will get notified instantly. Check out announcements in the classroom section for all the latest info related to classroom.

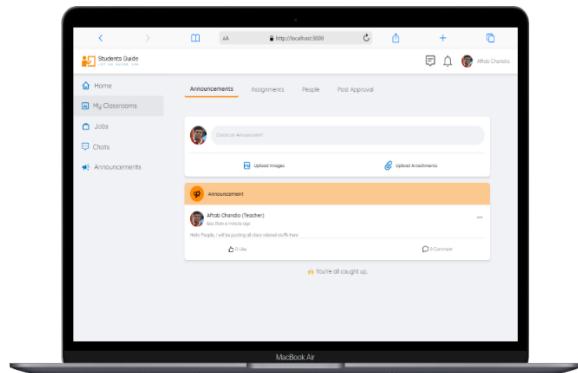


Figure 5.11: Classroom Announcement screen.

5.12 ASSIGNMENT CREATION.

Introducing the Teacher Assignment Creation Feature! Teachers can easily create assignments by adding a description, setting a due date, and attaching the assignment file. Once created, students in the class will receive a notification about it.

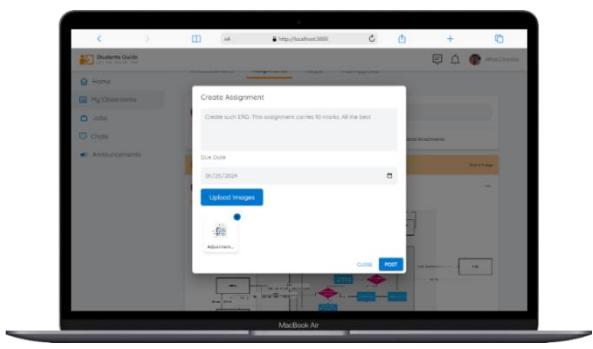


Figure 5.12: Classroom Assignment creation screen.

5.13 STUDENT POST.

Students can create posts to share their thoughts or questions and the teacher will receive a notification. Before the post goes live for everyone, the teacher reviews and approves it.

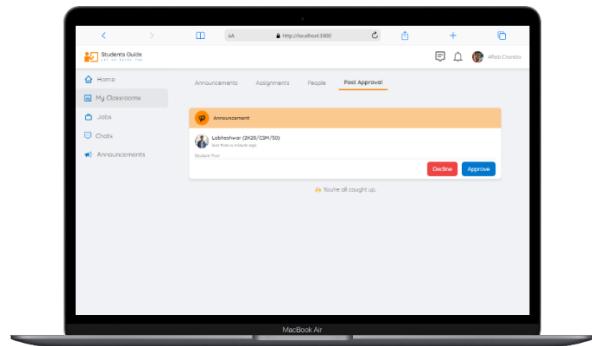


Figure 5.13: Student Post for teacher Approval.

5.14 PUBLIC UNIVERSITY POST.

Here is the screenshot from public university post feature. Where Teachers can share updates visible only to user from their specific university. It's a targeted way to communicate important information, creating a connected learning environment for university students.

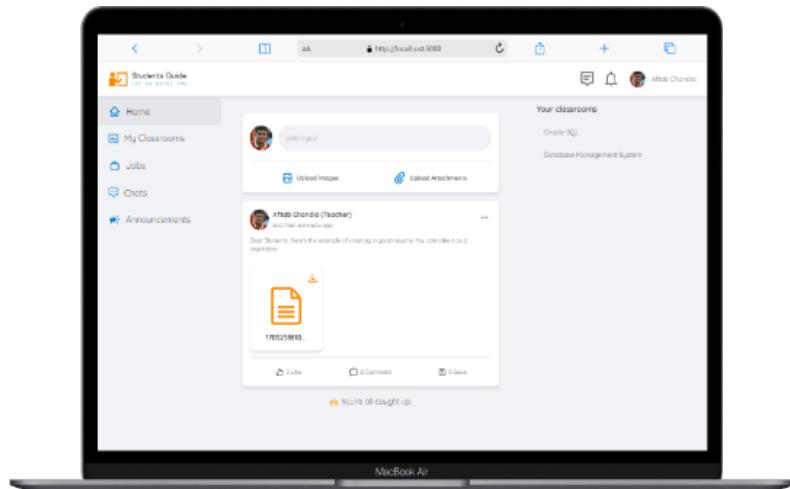


Figure 5.14: Public University Post screen.

5.15 COURSE RECOMMENDATION SYSTEM.

Here is a peek into our Course Recommendation System! Where students input their interests and skills during account creation or update them later. In the Recommendations section, students will discover personalized course suggestions based on their preferences. It's an easy way to find courses that align with their interests and enhance their learning journey.

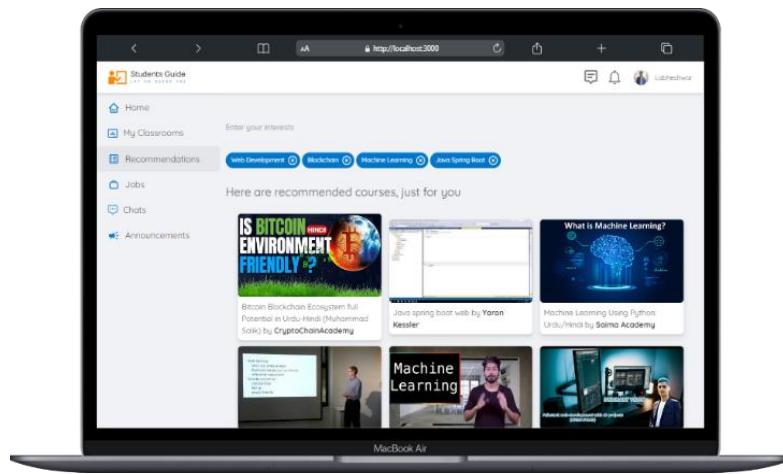


Figure 5.15: Course Recommendation screen.

5.16 USER PROFILE.

This is User Profile Settings. Where students can personalize their profile by editing and updating the name, bio, profile picture, and interests/skills. It's an easy way for students to showcase their personality, skills, and interests within our community and keep their profile fresh and engaging.

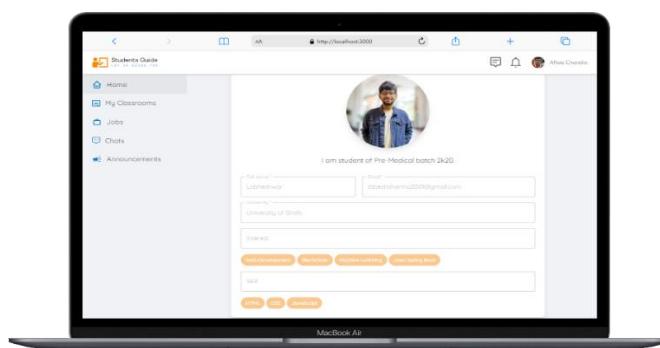


Figure 5.16: User Profile Settings screen.

5.17 CHAT AND CONVERSATION.

Explore our End-to-End Chat Feature! Students can directly and privately message teachers, initiating one-on-one conversations. It's a seamless way to connect, ask questions, and get personalized support, fostering a dynamic and interactive learning experience.

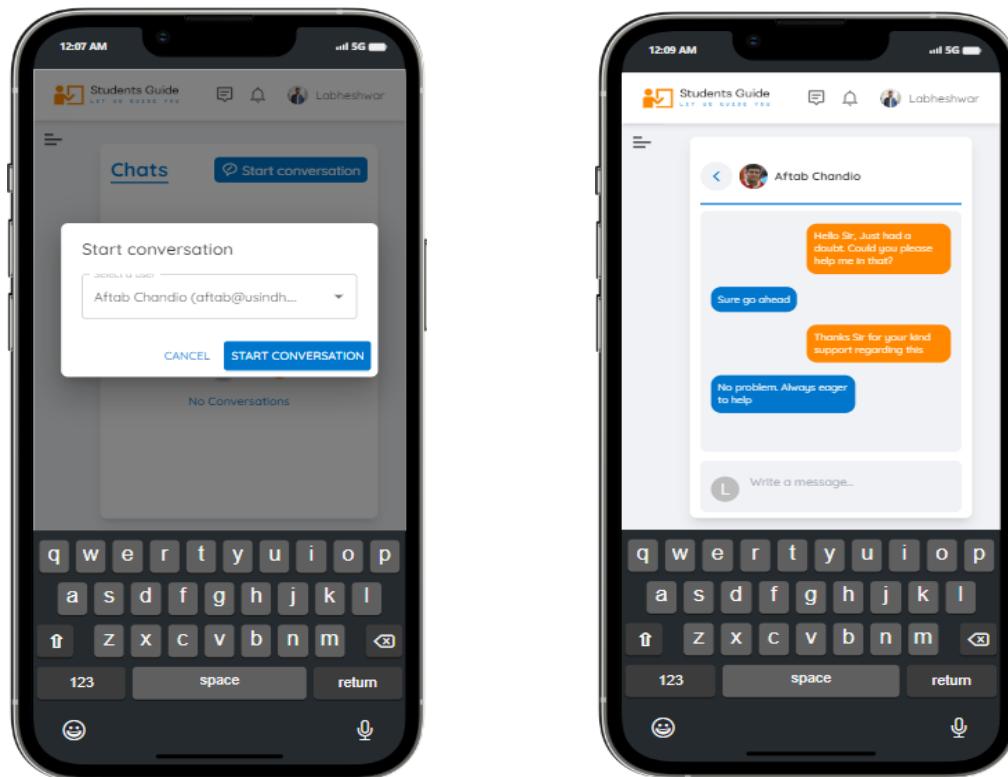


Figure 5.17 Chat and Conversation screen.

5.18 JOB CREATION AND LISTING.

Here is our Job Creation and Listing Feature! Employers can easily create job listings by providing essential details like job title, company name, location, job type (onsite, remote, or hybrid), company size, and contact email. It's a straightforward process to connect employers with potential candidates, streamlining the job posting experience. Users can find jobs with location, job-type, job mode, and posting date filters.

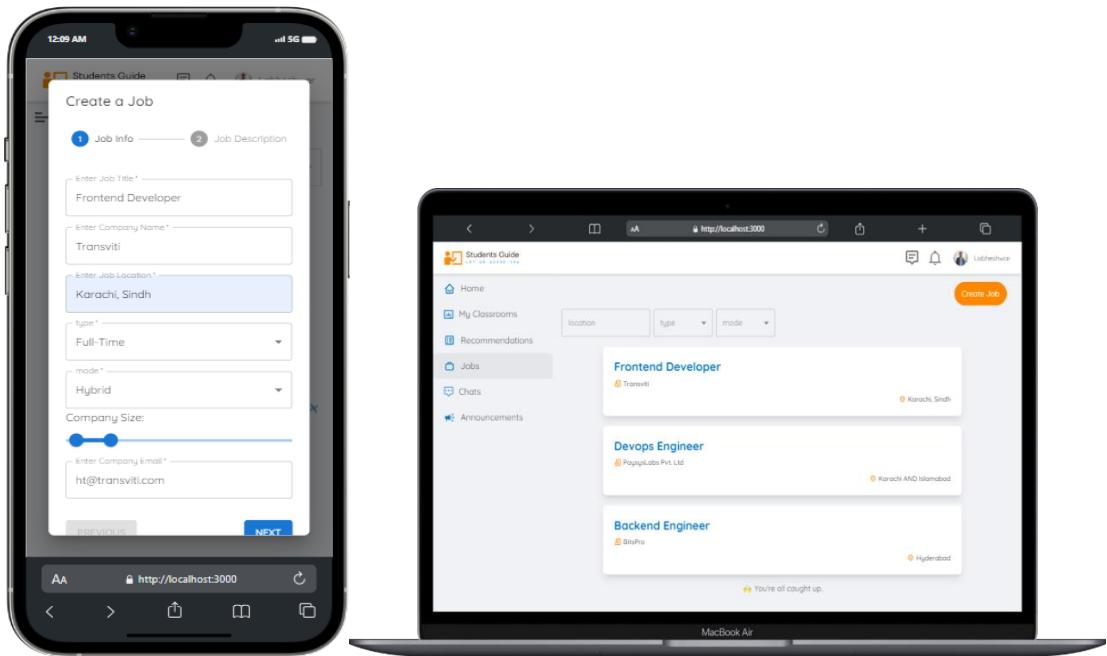


Figure 5.18: Job Creation and Listing screen

5.19 POST REPORT.

Here's our Post Reporting System. If a user or a student comes across a post that seems spammy, false, or goes against the rules, then they can report it to the admin for review and potential deletion. Reported post will be notified the admin, who can then take action by either deleting the post or choosing to ignore it by clicking the ignore button in case of a false report.

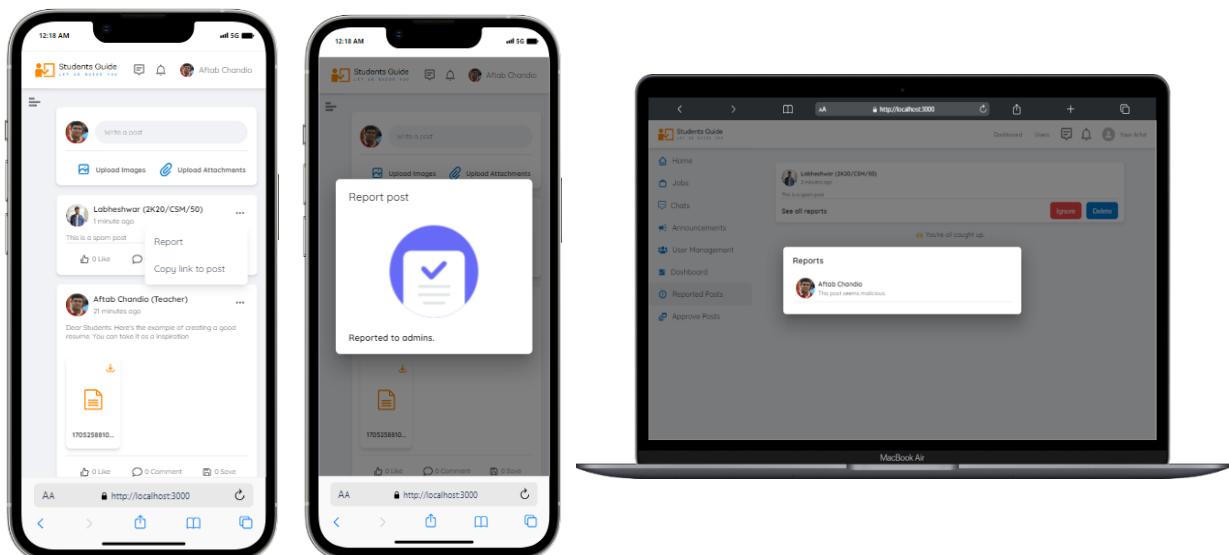


Figure 5.19 Post Reporting Flow.

5.20 ADMIN DASHBOARD.

Here is the refreshing and modern Admin Dashboard - the control center for overseeing portal activities! Here, admins can effortlessly monitor the platform's pulse, checking the number of registered users, teachers, created classrooms and reported posts. The dashboard provides a user-friendly overview, displaying counts for teachers, admins, posts, and announcements, along with a breakdown of verified and non-verified users in donut and pie charts.

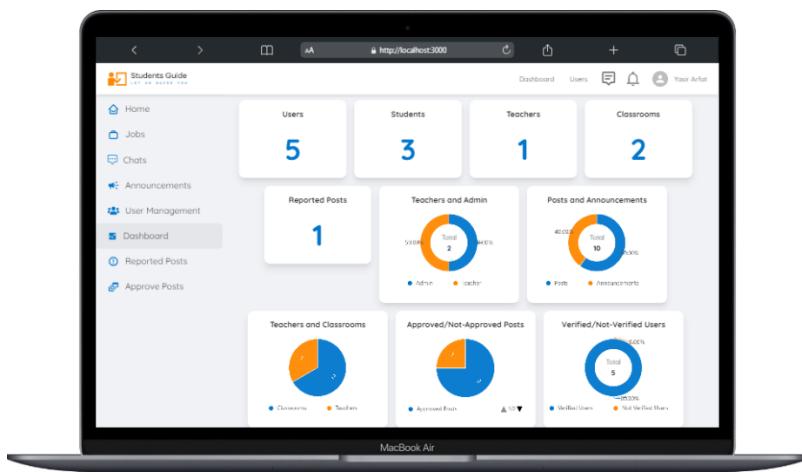


Figure 4.20: Admin Dashboard screen.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

6.1 CONCLUSION

The journey of developing "Student Guide" has been both challenging and rewarding, marking a significant step in the realm of educational technology. This project was conceived with the aim of enhancing the university experience for students, teachers, and administrators by providing a comprehensive platform for classroom management, job finding, and course discovery. The development process, grounded in thorough research and a user-centered approach, has led to the creation of an application that not only meets its initial objectives but also sets new standards for usability and functionality in educational tools.

Throughout the development phase, key features such as user authentication, profile management, classroom management tools, job portal integration, and course recommendation systems were successfully implemented. The application's interface was designed with a focus on user-friendliness and aesthetic appeal, taking into account the diverse needs and technical proficiencies of its intended user base. Beta testing and internal reviews have shown promising results, indicating the application's potential for positive reception and high user engagement upon launch.

6.2 REFLECTIONS ON THE DEVELOPMENT PROCESS

Reflecting on the development process, several key learnings stand out. The importance of flexibility and adaptability in software development was a recurring theme, especially in an agile environment. Regular feedback loops and iterative development were crucial in refining the application and aligning it more closely with user expectations. The project also highlighted the significance of balancing technical innovation with practical usability, ensuring that the technology serves to enhance, rather than complicate, the user experience.

6.3 FUTURE DIRECTIONS

Looking ahead, "Student Guide" is poised for a series of expansions and enhancements:

6.3.1 Integration of Advanced Technologies:

- i) Plans are in place to incorporate AI and machine learning for personalized course recommendations and job alerts, providing users with more tailored and relevant content.
- ii) Want to add an AI chatbot integrated with OpenAI's GPT-3.5 Model to help Students with the advances queries and flow.
- iii) Giving user option to start a group chat, synced with their classroom or university and optimizing the overall chat experience such that it can compete other platforms in terms of chat.
- iv) Integrating firebase for push notifications and adding socket on notification so that the user can have instant notification.

6.3.2 Expansion of User Base and Partnerships:

- i) A key focus will be on expanding the user base, including reaching out to more universities and educational institutions, making them familiar with the app.
- ii) Forming partnerships with educational content providers and industry professionals to enrich the resources available on the platform.
- iii) Do social media marketing of the app on different platforms such that more users get to know about this platform.

6.3.3 Continuous Improvement and Adaptation:

- i) The application will undergo continuous improvement based on user feedback post-launch. This will include regular updates to enhance existing features and potentially introducing new functionalities.
- ii) Keeping up with technological advancements and educational trends will be crucial to ensure that "Student Guide" remains a relevant and valuable tool for its users.

6.4 FINAL THOUGHTS

In conclusion, "Student Guide" represents a significant achievement in educational technology, with the potential to positively impact the way students, teachers, and administrators interact and access resources. The project's journey has been a testament to the power of user-centered design, agile development, and continuous learning. As we look to the future, "Student Guide" is well-positioned to grow, evolve, and continue making meaningful contributions to the educational community.

REFERENCES

- [1] Agile Alliance. What is Agile? Agile Alliance. Available at: <https://www.agilealliance.org/agile101/>
- [2] Cronin, D. The Interaction Design Foundation: Interaction Design Best Practices. Interaction Design Foundation. Available at: <https://www.interaction-design.org/literature/topics/interaction-design>
- [3] Lewis, J. R., & Sauro, J. The System Usability Scale: Past, Present, and Future. MeasuringU. Available at: <https://measuringu.com/sus/>
- [4] MongoDB. MongoDB Atlas: the Global Cloud Database. MongoDB, Inc. Available at: <https://www.mongodb.com/cloud/atlas>
- [5] Next.js. The React Framework for the Web. Next.js, Inc. Available at: <https://nextjs.org/>
- [6] Node.js Foundation. Node.js – JavaScript runtime. Available at: <https://nodejs.org/en/>
- [7] Express.js. Express.js: Node.js web application framework. Express.js, Inc. Available at: <https://expressjs.com/>
- [8] Google Console. Google Developers API. Available at: <https://console.cloud.google.com/apis>
- [9] Material UI. React Components that implements Material Design - MUI. Available at: <https://mui.com/material-ui/>

[10] YouTube API Services. YouTube Data API Overview. Google Developers. Available at: <https://developers.google.com/youtube/v3/getting-started>

[11] Tailwind CSS. Utility-first CSS Framework. Available at: <https://tailwindcss.com/>

[12] G2Plot. Visual graphs library. Available at: <https://g2plot.antv.antdgroup.com/>

[13] React Icons. Icons for React. Available at: <https://react-icons.github.io/>

[14] Redux. A predictable state container. Available at: <https://redux.js.org/>

[15] Formik. Build forms in React, without the tears. Available at: <https://formik.org/>

[16] Amazon S3 Bucket. Cloud Object Storage – Amazon S3 - AWS. Available at: <https://aws.amazon.com/s3/>

[17] JWT. JSON Web Tokens – JWT.io. Available at: <https://jwt.io/>

[18] Nodemailer. Nodemailer. Available at: <https://nodemailer.com/>

[19] GitHub. Student Guide Repository. Available at: <https://github.com/labheshwar/students-guide>

[20] API Documentation. Student Guide API Documentation on Postman. Available at: <https://www.postman.com/bold-star-463736/workspace/students-guide/collection/24369399-42e872cd-744d-48f8-926e-ccb2b23b4e3e?action=share&creator=24369399>