Student Migration Insights

Mini project report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by
Athira Adithyan [FIT22CS046]
Catherine Mary Mathew [FIT22CS057]
Christa Jose [FIT22CS060]

Federal Institute of Science And Technology (FISAT)
Angamaly, Ernakulam

Affiliated to APJ Abdul Kalam Technological University
CET Campus, Thiruvananthapuram
April 2025

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)

Mookkannoor(P.O), Angamaly-683577

CERTIFICATE

This is to certify that the Mini Project report for the project entitled **"Student Migration Insights"** is a bonafide report presented during **VIth semester** (CSD334 - Mini Project) by **Athira Adithyan (FIT22CS046)**, **Catherine Mary Mathew (FIT22CS057)**, and **Christa Jose (FIT22CS060)**, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology (B.Tech)** in **Computer Science & Engineering** during the academic year **2024-25**.

Jismy Mathew	**Ms. Anitha T Nair**	**Dr. Paul P Mathai**
Project Coordinator	Project Guide	Head of the Department

ABSTRACT

Student Migration Insights is a comprehensive web platform designed to assist students in navigating the study abroad process by providing essential resources, peer support, and in-depth university insights. The platform features a community forum, enabling students to connect, share experiences, and seek advice from alumni and fellow aspirants. Its university insights section offers detailed information about institutions worldwide, covering admission requirements, tuition fees, scholarships, and student life. Additionally, the posts section allows users to share personal experiences, study tips, and insights about living in different countries. To address common concerns, the platform includes a dedicated FAQ section covering topics such as visa applications, accommodation, cultural adaptation, and career opportunities. By integrating these features, Student Migration Insights serves as a user-friendly and transparent resource, empowering students with the knowledge and support needed to make informed study abroad decisions.

ACKNOWLEDGMENT

We are grateful to the Almighty, who has blessed us with good health, commitment, and continuous interest throughout our project work. First and foremost, we express our sincere gratitude to **Mr. Jacob Thomas V**, Principal, Federal Institute of Science and Technology, Angamaly, for providing us with all the necessary facilities and a conducive learning environment, which played a crucial role in the successful completion of our project. We would also like to extend our heartfelt appreciation to **Dr. Paul P. Mathai**, Head of the Department, Federal Institute of Science and Technology, for his valuable insights, continuous encouragement, and constructive feedback, which significantly contributed to the refinement and development of our project.

We are deeply grateful to our Project Coordinators,**Ms. Jismy Mathew**, Assistant Professor, and **Ms. Anitha T. Nair**, Assistant Professor (Special Grade), Department of Computer Science and Engineering, for their invaluable guidance, encouragement, and support throughout our project journey. Their expertise and valuable suggestions were instrumental in shaping our work and ensuring its successful completion. Finally, we extend our sincere gratitude to the entire teaching and non-teaching staff of the Computer Science and Engineering Department at the Federal Institute of Science and Technology for their cooperation and assistance throughout our project. Thank you all for your unwavering support, guidance, and dedication to excellence.

Athira Adithyan

Catherine Mary Mathew

Christa Jose

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Student Migration Insights is a comprehensive web platform designed to guide students through every stage of their study abroad journey by offering vital resources, peer support, and expert insights. The platform fosters a community-driven approach, allowing students to connect through a forum where they can engage with alumni, current international students, and fellow aspirants to share experiences, seek advice, and discuss challenges. The university insights section provides extensive details on institutions worldwide, including admission requirements, tuition fees, scholarship opportunities, and insights into student life, helping students make well-informed decisions.

Additionally, the platform features a posts section, where users can share personal experiences, study strategies, and firsthand insights into adapting to different countries. To address common concerns, the FAQ and guidance section offers reliable information on critical aspects such as visa applications, accommodation options, cultural adaptation, and post-graduation career opportunities. By integrating these features into a user-friendly and interactive platform, Student Migration Insights serves as a one-stop resource, simplifying the study abroad process and empowering students with the knowledge and support they need to make confident, informed decisions about their academic and personal journeys abroad.

1.2 Problem Statement

Many students struggle with researching universities and planning their study abroad journey due to the lack of a centralized platform that combines detailed university insights with interactive community support. While some websites provide rankings and admission details, they often lack peer discussions, real-time Q&A, and shared experiences. On the other hand, student forums offer discussions but lack structured university data, forcing students to visit multiple sources, making the process overwhelming. Additionally, choosing the right university and course is challenging due to factors like academic goals, financial constraints, scholarships, and career prospects. Without personalized recommendations and filtering options, students risk making uninformed decisions that may affect their future.

Beyond academics, students also face difficulties understanding visa policies, language proficiency tests, and financial requirements, as this information is often scattered across different sources. Adapting to a new country adds further challenges, with students struggling to navigate housing regulations, work policies, healthcare systems, and cultural differences. Language barriers and unfamiliar social norms can lead to isolation, making the transition stressful. Without a structured guide and peer support, students may encounter unnecessary hurdles in adjusting to their new environment.

1.3 Objectives

Studying abroad is a complex journey that requires careful planning and access to reliable information. Many students struggle to find a single, comprehensive platform that provides all the necessary details about universities, admission processes, scholarships, visa requirements, and cultural adaptation. This project aims to address that gap by centralizing essential resources, reducing the need for students to search through multiple sources. By streamlining this information, the platform will enable students to make well-informed decisions efficiently, ensuring they choose institutions that align with their academic and financial needs. Additionally, detailed university insights, including admission criteria, tuition fees, scholarship opportunities, and deadlines, will empower students to compare options and navigate the application process with confidence.

Beyond academic logistics, this platform will also focus on the social and legal aspects of migration, which are often overlooked. Moving to a new country presents challenges beyond admissions, such as understanding visa requirements, work regulations, housing laws, and healthcare systems. To simplify this transition, the platform will provide structured guidance on visa applications, financial requirements, and language proficiency tests for different countries. Furthermore, a community forum will allow students to connect with peers, share experiences, and seek advice from alumni, creating a supportive network that fosters interaction and knowledge exchange. Additional features like FAQs, study tips, and firsthand experiences from international students will help aspirants adapt to new academic and cultural environments, ensuring a smooth and successful transition into their study-abroad journey.

1.4 Scope of the Project

The Student Migration Insights platform is designed to support students planning to migrate for higher education, catering to both prospective applicants and current international students. The platform offers a comprehensive suite of features to assist students in making informed decisions about their academic journey. Key functionalities include a university and course search, enabling students to explore institutions based on admission criteria, tuition fees, and scholarships. A community forum facilitates peer discussions, allowing students to connect, share experiences, and seek advice from alumni. Additionally, the platform provides country-specific visa guidance, helping students navigate application processes, language proficiency requirements, and financial obligations. To enhance accessibility, an AI-powered chatbot offers instant assistance, addressing common queries related to studying abroad.

The platform initially focuses on major study destinations such as the USA, UK, Canada, Germany, and Australia, with plans for future expansion to include additional countries based on demand. Designed for scalability, the project aims to grow by incorporating mobile applications, API integrations, and advanced features such as personalized recommendations and real-time support systems. By leveraging technological advancements, the platform seeks to become a one-stop solution for aspiring international students, simplifying the migration process and making global education more accessible

1.5 Social Relevance

Empowering students with accurate information is crucial in helping them make thoughtful decisions about migration, education, and career opportunities abroad. By providing reliable insights into universities, courses this platform ensures that students can navigate their study abroad journey with confidence. Access to well-structured data and expert guidance minimizes uncertainty, enabling students to make informed choices that align with their academic and professional goals. Moreover, by offering personalized recommendations based on student preferences and aspirations, the platform simplifies the decision-making process and reduces the stress associated with choosing the right university and program.

Beyond information, the platform fosters community building and global connectivity by creating a supportive environment where students can exchange experiences, seek advice, and share helpful resources through peer-to-peer forums. This sense of belonging helps students overcome challenges such as cultural differences, academic pressures, and adapting to new environments. Additionally, it prioritizes accessibility by offering a responsive and user-friendly design, ensuring that students from diverse backgrounds can fully participate. Through multilingual support, interactive discussions, the platform bridges the gap between students and essential study abroad resources. This inclusive approach promotes equal opportunities, making valuable study abroad resources available to a wider audience while fostering cross-cultural exchange and understanding.

Chapter 2

Literature Review

2.1 Related Works

Several platforms assist students planning to study abroad by offering comprehensive resources and personalized support. These platforms typically provide extensive databases of universities and courses worldwide, allowing students to filter options based on their academic interests, preferred countries, and other criteria. Detailed information about admission requirements, application deadlines, and tuition fees is often available, helping students make informed decisions.

In addition to university listings, these services frequently offer guidance on the application process, including tips on writing personal statements, preparing for interviews, and meeting language proficiency requirements. Some platforms also provide tools to assess eligibility, calculate potential costs, and explore scholarship opportunities, enabling students to plan their educational journey effectively.

Visa application support is another critical feature, with many platforms offering step-by-step guides, document checklists, and updates on visa policies to ensure students comply with all necessary regulations. Accommodation assistance is also common, with information on student housing options, cost comparisons, and advice on securing safe and affordable lodging.

Financial planning resources, such as budgeting tools and information on managing expenses abroad, are often available to help students prepare for the financial aspects of studying overseas. Some platforms also offer pre-departure orientations, covering topics like cultural adaptation, academic expectations, and health and safety considerations, to help students transition smoothly into their new environments.

While these platforms provide valuable information and resources, they may lack real-time interaction features. Incorporating live chat functionalities could enhance user experience by allowing students to receive immediate, personalized assistance, making the process of studying abroad more accessible and less daunting.

References:

[1] React

[2] Firebase

[3] Tailwind CSS

[4] Node.js

2.2 Comparison of Related Works

The Student Migration Website is a comprehensive platform designed to assist students in exploring international educational opportunities by integrating various essential features into a cohesive user experience. It offers a user-friendly interface that enables students to search for universities and courses tailored to their preferences, facilitating informed decision-making. The platform also includes real-time chat functionalities, allowing users to interact with peers, share experiences, and seek guidance, thereby fostering a supportive community. Additionally, it provides access to country-specific migration guidelines, ensuring that students have accurate and structured information to navigate the complexities of studying abroad. Features such as FAQs, discussion forums, and personalized recommendations further enhance the overall user experience, making it a one-stop solution for students considering international education.

In comparison, several other platforms offer similar services aimed at facilitating the study abroad process for students. For instance, iSchoolConnect leverages artificial intelligence to streamline the application process, reducing the time required from over 200 hours to approximately 24 hours. It provides profile-based recommendations by analyzing a vast database of universities and programs, and offers tools like the Writing Mentor and Video Interview Analyzer to assist students with their applications and interview preparations. Leverage Edu is another platform that provides end-to-end services for students pursuing international education, offering specialized services and tech products such as Univalley and Uniconnect to connect universities and students effectively. Similarly, Leap Scholar serves as a comprehensive study abroad platform, offering products and services related to overseas education, including test preparation, university selection assistance, financial aid planning, and career placement services. Yocket focuses on creating a community-driven online platform that connects students with international universities, providing tools for university research, application tracking, and access to a network of peers and alumni.

While these platforms offer valuable services, the Student Migration Website distinguishes itself by integrating course and university searches, real-time chat, discussion forums, FAQs, and personalized recommendations into a single, cohesive platform. This integration ensures that students have access to all necessary resources and support in one place, streamlining their journey towards studying abroad. The inclusion of real-time chat and community forums fosters a sense of belonging and provides immediate assistance, which is crucial for students navigating the complexities of international education. By combining these features, the Student Migration Website offers a holistic approach that addresses various aspects of the student migration journey, from initial exploration to application and enrollment, providing a seamless and supportive experience for users seeking international education opportunities.

Feature	**Existing Systems**	**Student Migration Website**
Real-Time Chat	Limited or no live chat functionality.	Integrated live chat enabling instant communication among students.
Discussion Forums	Few or no forums available for student interaction.	Comprehensive forums facilitating topic discussions and knowledge sharing.
FAQs	General and often insufficient information.	Detailed FAQs addressing common student migration inquiries.
Migration Guidelines	Broad, non-specific advice.	country-specific guides
User Interface	Basic design with limited user engagement features.	Intuitive and visually appealing interface enhancing user experience.
Community Building	Minimal emphasis on fostering a student community.	Strong focus on building a supportive and interactive student community.

Table 2.1: Feature Comparison Between Existing Systems and the Student Migration Website

Chapter 3

Design Methodologies

3.1 Software Requirement Specification

3.1.1 Purpose

Student Migration Insights is a web-based platform designed to assist students migrating abroad for higher education. It provides university search, course exploration, FAQs, post uploading, and real-time chat support. The platform aims to streamline the study abroad process by offering structured and interactive resources that enhance decision-making and connectivity among students.

3.1.2 Scope

1. University search,location and courses offered: The platform provides a comprehensive university search tool that allows students to explore institutions based on their preferred location, courses, and admission criteria. Users can filter universities according to tuition fees, scholarship availability, and entry requirements, ensuring they find institutions that align with their academic and financial goals. This feature helps students make informed decisions without having to visit multiple websites for scattered information.

2. Real-time chat feature for instant support from the students:A real-time chat system enables students to connect with peers, alumni, and other aspirants instantly.

This feature fosters a supportive community where users can ask questions, share experiences, and get immediate guidance on various aspects of studying abroad. It helps students resolve doubts quickly and gain firsthand insights from others who have gone through similar experiences.

3. Secure and user-friendly interface with personalized recommendations:The platform is designed with a secure and intuitive user interface, ensuring smooth navigation and data protection. It provides personalized recommendations based on students' preferences, such as their academic background, financial constraints, and destination choices. This customization makes the experience more efficient and relevant, helping students find tailored opportunities that suit their needs.

4. AI powered chatbot for queries:An AI-driven chatbot is integrated into the platform to handle common student queries instantly. It provides quick answers on topics like admission processes, visa requirements, scholarships, and accommodation options, reducing the time students spend searching for information. The chatbot is available 24/7, ensuring round-the-clock support for students in different time zones.

5. FAQs for previously asked questions from the students:A dedicated FAQ section compiles frequently asked questions to provide instant solutions to common concerns. It covers a wide range of topics, including visa applications, language requirements, cultural adaptation, and financial planning. This feature helps students find answers quickly without waiting for responses in the forum or chat system.

3.1.3 User Requirements

Users should be able to search for universities and courses based on location, admission criteria, and financial constraints. The platform must include an interactive community support system with a forum and real-time chat feature to facilitate communication among students, alumni, and experts. Secure authentication should be implemented to allow users to register and log in safely, ensuring personalized access to platform features. An AI chatbot should be integrated to provide automated responses to common queries related to admissions, visas, and scholarships. The platform must have a responsive and intuitive user interface for seamless navigation. A comprehensive FAQ section should be available to address common concerns and frequently asked questions. Additionally, users should have the ability to upload posts, share experiences, and contribute study-related insights.

3.1.4 System Requirements

The system requires a minimum of 8GB RAM and a 2.5 GHz processor for smooth server operations. At least 100GB of storage is necessary to manage databases and user-uploaded content. Stable internet connectivity is crucial for enabling real-time interactions and chat functionality.

3.1.5 Software Requirements:

The platform should support Windows, macOS, or Linux-based server environments. The backend should be developed using Node.js for handling server-side logic. MongoDB should be used for structured and efficient data storage. The frontend should be built using React.js for a dynamic and interactive user experience. AI integration should include machine learning APIs for chatbot implementation. Security measures must include SSL encryption, OAuth authentication, and role-based access control.

3.1.6 Functional Requirements

The platform should allow users to register, log in, and securely manage their profiles. It must include a university and course search feature with filters for country and ranking options. A real-time chat system should enable students to connect instantly for support and discussions. Users should also be able to create and upload posts, share experiences, and contribute study-related insights, ensuring a dynamic and engaging community.

3.1.7 Non-Functional Requirements

Performance Requirements

The system should be designed to ensure that the response time for any user action does not exceed 2 seconds, providing a seamless and efficient user experience. Additionally, the system must be capable of handling a minimum of 10,000 concurrent users without performance degradation. This requires robust infrastructure, optimized database queries, efficient load balancing, and scalable architecture to maintain high availability and responsiveness even under heavy user loads.

Security Requirements

The system must implement robust security measures, including data encryption for all user transactions and stored information, ensuring confidentiality and protection against unauthorized access. Additionally, role-based access control (RBAC) should be enforced, categorizing users into distinct roles such as Students and Admins. This approach ensures that each user only has access to the features and data relevant to their role, enhancing security and preventing unauthorized modifications or data breaches.

3.2 Overall Description

3.2.1 Product Perspective

Student Migration Insights is an interactive platform that integrates database-driven content and real-time chat features for user engagement.

3.2.2 User Characteristics

- **Students**: Searching for study-abroad opportunities,maintain a peer to peer interactions.
- **Admin**: Assisting students with migration processes.

Usability Requirements

Simple and intuitive UI/UX design. Fully responsive design for mobile and desktop users. Easy navigation with a well-structured menu system. Clear and concise instructions for users to complete tasks effortlessly. Support for multiple languages to enhance accessibility for international students.

Reliability and Availability

99.9% uptime guarantee. Automatic data backup every 24 hours. Redundant server architecture to prevent single points of failure. Real-time monitoring and alert system for system performance issues. Regular security updates and maintenance to ensure data integrity.

Scalability

The system should scale efficiently with increasing users. Support for cloud-based infrastructure to handle high traffic loads. Modular architecture to allow seamless integration of new features. Load balancing to distribute traffic evenly across multiple servers. Optimized database queries and caching mechanisms for faster performance.

3.2.3 Constraints

Technology Constraints:

1. Internet Dependency:

   A stable internet connection is required for users to access real-time chat, university searches, and downloadable resources. Poor connectivity may hinder functionality, especially in remote areas.

2. Database Management:

   The platform relies on MongoDB Atlas, which requires efficient indexing and optimization to handle increasing amounts of user data efficiently.

3. Security Compliance:

Strong encryption protocols must be implemented to ensure data protection, especially for authentication, chat messages, and document downloads.

4. Third-Party Integrations:

   The system integrates with external services such as authentication providers, analytics tools, and cloud storage, requiring ongoing maintenance to ensure compatibility and functionality.

5. Scalability:

   As the number of users grows, the backend (FastAPI) and frontend (React) must be continuously optimized to maintain performance, reduce response times, and handle large concurrent requests efficiently.

6. Cross-Browser & Cross-Platform Compatibility:

   The website must function properly across different browsers (Chrome, Edge, Firefox) and devices (desktops, tablets, and mobile phones) to provide a seamless user experience.

Hardware Constraints:

1. Server Requirements:

   The backend is hosted on Render, and the database is on MongoDB Atlas. These require scalable cloud infrastructure to efficiently process concurrent user requests and data storage needs.

2. User Devices: Users need smartphones, tablets, or desktops with modern web browsers to access all website features smoothly.

3. Hosting Limitations: The website is hosted on Vercel, which may have bandwidth, storage, or request limitations depending on the hosting plan used.

4. Mobile Optimization:

   Features such as real-time chat and university search require devices with sufficient processing power and memory to ensure smooth and responsive interactions.

3.2.4 Application Architecture Design

Frontend:

Technology: React (Vite)

Description: The frontend application is developed using React with Vite, a modern JavaScript toolchain that offers fast build times and efficient bundling. It enables the creation of interactive and responsive user interfaces with a component-based architecture, enhancing maintainability and scalability.

Backend

Technology: Node.js

Description: The backend of the application is built using Node.js, a JavaScript runtime that allows for scalable and efficient server-side development. Node.js provides a non-blocking, event-driven architecture, making it suitable for handling multiple concurrent requests. It is used with Express.js to create RESTful APIs that manage user authentication, real-time communication, and database operations. The backend integrates with MongoDB Atlas for data storage and supports third-party services for authentication and analytics.

Database

Technology: MongoDB Atlas

Description: The platform utilizes MongoDB Atlas, a cloud-based NoSQL database, to store and manage user data efficiently. It offers automatic scaling, security compliance, and real-time data access, ensuring seamless integration with the backend.

Database Design:

User Table

FIELDNAME	DATATYPE	CONSTRAINTS
id	ObjectId	Pimary Key(MongoDB Default)
name	String	Not Null
email	String	Unique,Not Null
uid	String	Unique,Not Null
role	String	Not Null,Default: ”user”

Table 3.1

Course Table

FIELDNAME	DATATYPE	CONSTRAINTS
_id	ObjectId	Primary Key (MongoDB Default)
university	String	Not Null
place	String	Not Null
course	String	Not Null
countryCode	String	Unique, Indexed

Table 3.2

Guides Table

FIELDNAME	DATATYPE	CONSTRAINTS
_id	ObjectId	Primary Key (MongoDB Default)
fileName	String	Not Null
fileUrl	String	Not Null
countryCode	String	Primary Key, Unique, Indexed

Table 3.3

Post Table

FIELDNAME	DATATYPE	CONSTRAINTS
_id	ObjectId	Primary Key (MongoDB Default)
userId	String	Foreign Key (References Users Table)
title	String	Not Null
content	String	Not Null

Table 3.4

PrivateChats Table

FIELDNAME	DATATYPE	CONSTRAINTS
_id	ObjectId	Primary Key (MongoDB Default)
participants	Array	Stores user IDs of chat participants (size: 2)
messages	Array	Stores messages exchanged in the chat
lastMessage	Object	Stores the most recent message details

Table 3.5

FAQ Table

FIELDNAME	DATATYPE	CONSTRAINTS
_id	ObjectId	Primary Key (MongoDB Default)
question	String	The text of the FAQ question.
answer	String	The text of the FAQ answer.

Table 3.6

3.3 System Architecture

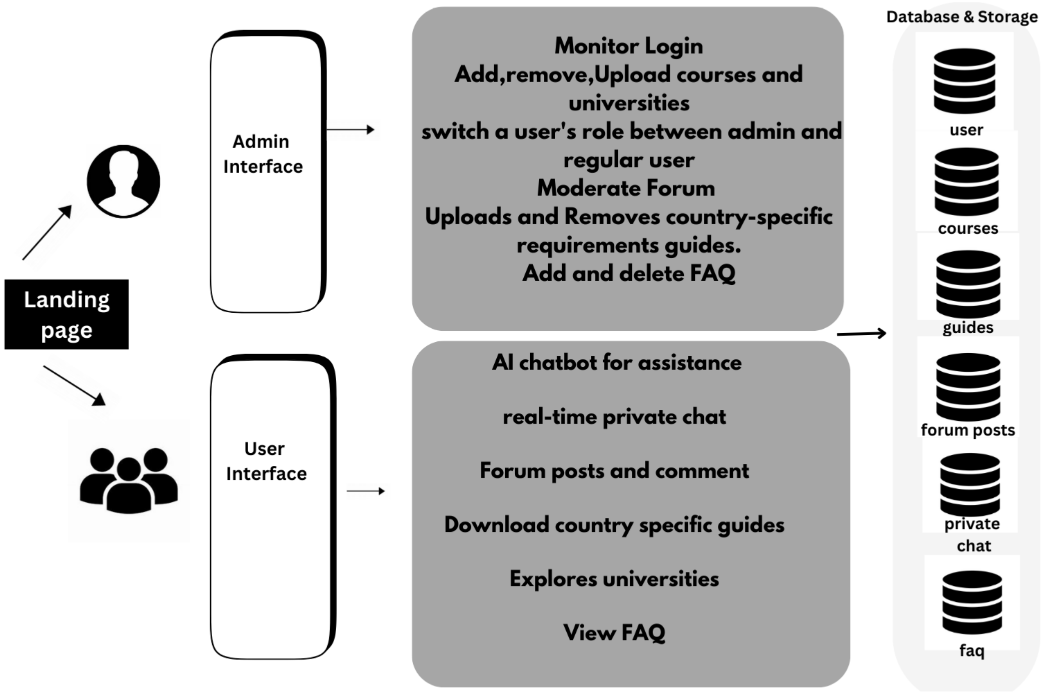


Figure 3.1: System Architecture for Student Migration Insights

The system architecture for the Student Migration Insights platform comprises three key components: the User Interface, the Admin Interface, and Database and Storage. The Landing Page serves as the entry point, directing users either to the User Interface, designed for students, or the Admin Interface, used for platform management. The User Interface provides essential features such as an AI-powered chatbot for assistance, real-time private chat, forum posts and comments, university exploration, downloading country-specific guides, and viewing FAQs. On the other hand, the Admin Interface allows administrators to monitor logins, manage universities and courses, switch user roles, moderate forums, upload and remove country-specific requirement guides, and manage FAQs. All interactions within the platform are stored in a structured Database & Storage system, which maintains user data, courses, guides, forum posts, private chats, and FAQs, ensuring seamless data management and retrieval. This architecture ensures an efficient, user-friendly experience while maintaining robust administrative control over the platform's functionalities.

Chapter 4

Implementation

4.1 Implementation Details

The Student Migration Website is a web-based platform designed to assist students in migrating to different countries for education. It provides an intuitive UI, secure document handling, and an AI-driven advisory system. The platform consists of a React (Vite) + Tailwind CSS frontend, a Node.js (Express) backend, and MongoDB for database storage.

4.1.1 React with Vite 4.2.1

React is a JavaScript library for building interactive user interfaces. The frontend of our project was developed using React with Vite 4.2.1, which ensures fast build times and optimized performance.

Key Components Created:

- **Pages:** Login, Register, Home, About,Country pages, Dashboard, User Details,
- **Reusable UI Components:** Navbar, Footer, Modals, Form Elements.
- **State Management:** Implemented using React Context API and Zustand for efficient state handling.

Key Features Integrated:

- **Hooks:** Utilized useState, useEffect, useContext, and custom hooks for managing application logic.

- **Component-Based Architecture:** Each feature was implemented as a reusable React component.

4.1.2 Tailwind CSS

A utility-first CSS framework used to build a responsive, customizable, and modern UI. It provided prebuilt utility classes, reducing CSS redundancy and improving styling consistency.

4.1.3 React Router

React Router was implemented for navigation and page routing. It supports:

- **Dynamic Routing:** Used for pages like course,post and community forum
- **Protected Routes:** Ensuring only authenticated users can access specific sections.

4.1.4 Firebase Authentication

Firebase Authentication was used for user login and registration, offering secure authentication via:

- Email/Password Login

4.1.5 Axios

Axios was used to handle HTTP requests between the React frontend and the FastAPI backend. It allows:

- Fetching user details, university data, and migration guidelines.
- Efficient API communication with error handling and interceptors.

4.1.6 Zustand (State Management)

A lightweight state management library that ensures efficient global state handling across different components without the complexity of Redux.

4.1.7 React Query

React Query was used for data fetching, caching, and synchronization with the backend, optimizing API requests and reducing redundant network calls.

4.1.8 React Icons

React Icons was used to integrate scalable and customizable icons for better UI/UX design.

4.2 Algorithms

4.2.1 Sign Up

1. **Start**

   Import React, useState, axios, Firebase Authentication methods.

2. **Component Setup**

   Define `Signup` functional component.

   Initialize state variables: `name`, `email`, `password`, `file`, `error`.

3. **Form Submission**

   - Implement `handleSubmit` function.
   - Use Firebase `createUserWithEmailAndPassword(email, password)`.

4. **Send Verification Email**

   - After creating the user, call `sendEmailVerification(user)`.
   - Display a message instructing the user to check their email.

5. **Verify Email Status**

   - On user login, check `user.emailVerified` status.
   - If not verified, show an alert asking the user to verify their email.

- If verified, allow access to the application.

6. **Complete Registration**
    - Once email is verified, navigate to the sign-in page.
    - Store user details (`name`, `email`) in Firestore.
    - Redirect to login with a success message.

7. **Rendering**
    - Render sign-up form with input fields.
    - Display error messages.
    - Add submit button.

4.2.2 Sign In

1. **Start**
    - Import React, useState, Firebase Authentication methods.

2. **Component Setup**
    - Define `SignIn` functional component.
    - Initialize state variables: `email`, `password`, `error`.

3. **Form Submission**
    - Implement `handleSubmit` function.
    - Use Firebase `signInWithEmailAndPassword(email, password)`.

4.2.3 Navigation After Sign In

Check Authentication Status

1. If authentication is successful:
    - Retrieve user details from Firestore.

    - Check if `user.emailVerified` is true.
    - If false, display an alert asking the user to verify their email.

2. If authentication fails, display an error message.

Role-Based Navigation

1. Fetch user role from Firestore.
2. If user role is Admin, navigate to the Admin Dashboard.
3. Else, navigate to the Home Page.

4.2.4 Admin Dashboard User Management

Fetching User Data from Firebase

1. Call Firebase Firestore to retrieve user details.
2. Authenticate admin access before fetching user data.
3. Extract the following details:
    - User Name
    - Email
    - Role (User or Admin)
    - Last Active Status
4. Store retrieved data in the users' state.
5. Handle errors and update the error state in case of failure.

4.2.5 Displaying User Information

1. Map through the users' array.
2. Display user details in a structured table format.
3. Implement search and filtering options for easy management.

4. Allow sorting based on role, country, and last active status.

4.2.6 Modify Country-Specific Guidelines

1. Provide a dropdown to select a country.
2. Fetch existing country-specific guidelines from Firestore.
3. Display the rules in an editable format.
4. Allow the admin to update or delete rules.
5. Save modifications back to Firestore.

4.2.7 User Role Management

1. Provide an option to change user roles (User $\leftrightarrow$ Admin).
2. Restrict role modification to Admins only.
3. Update role changes in Firestore.
4. Notify users about role updates if applicable.

4.2.8 Logout and Active Session Management

1. Display currently active users.
2. Allow the admin to force logout any user if needed.
3. Ensure secure authentication for performing sensitive actions.

4.2.9 Error Handling and Completion

1. Display a loading indicator while fetching data.
2. Show relevant error messages for failures.
3. Ensure proper handling of failed API requests.
4. End execution after successfully displaying and managing users.

5. Ensure data synchronization with Firebase.

4.3 Coding Development

4.3.1 Coding Standards

Code Formatting

- Used **Prettier** for frontend formatting and **ESLint** for linting.

Error Handling

- Centralized error handling in **Express.js** using middleware.

Security Best Practices

- Applied **CORS Policy** to restrict unauthorized access.

Modular Design

- Authentication logic follows **separation of concerns**.
- Reusable React components for authentication flows.

4.3.2 Source Code Control Setup

The project is hosted on **GitHub** and follows best practices for version control.

Repository

The source code is maintained in a GitHub repository at: `https://github.com/`

Version Control Practices

The following version control strategies were adopted:

- **Branching Strategy**: Used **Git Flow** with `main`, `dev`, and feature branches.
- **Commits**: Followed **conventional commit messages** for consistency.

- **Pull Requests & Code Reviews**: All major changes go through PR reviews before merging.

4.4 Datasets

Dataset includes:

- **Universities Dataset**: Contains information about universities in different countries, including:
  - University names, locations, and rankings(opt).
  - Available courses and programs.

4.5 Deployment

The project was deployed with the following setup:

- **Frontend**: Hosted on Vercel.
- **Backend**: Deployed on Render with FastAPI.
- **Database**: MongoDB Atlas cloud storage.

Chapter 5

Testing

5.1 Test Plan

Test Objectives

The primary objectives of testing in **Student Migration Insights** are:

- To verify that all functionalities work as intended, ensuring smooth job search, resume generation, and interview processes.
- To assess system performance under different loads and ensure scalability.
- To identify and fix bugs to enhance user experience and security.

Test Scope

Testing is conducted for:

- **Frontend:** React Vite
- **Backend:** Node js

Test Approach

A variety of test approaches are used, including:

- **Unit Testing:** Testing individual components.

- **Integration Testing:** Ensuring proper communication between frontend and backend.
- **Functional Testing:** Verifying that system functionality meets requirements.
- **Load Testing:** Assessing system performance under load.

Test Environment

Backend Testing

- **Operating System:** Windows/Linux
- **Database:** MongoDB (NoSQL)

Frontend Testing

- **Browsers:** Chrome, Microsoft Edge
- **Operating System:** Windows/Linux

Test Cases

- **Unit Testing:** Individual components in the project are tested.
- **Integration Testing:** Backend and frontend features are integrated and tested.
- **Functional Testing:** The system's overall functionality is tested.
- **Load Testing::** The performance under stress is put to test.

Test Execution

Executing each test case and recording the results.

Test Schedule

- Preparing each test case.
- Executing each test case.

- Bug fixing.
- Regression Testing after bug fixes.

Risks and Contingencies

Changes in evolving needs may make it difficult to adapt to changes.

Conclusion

According to the test objectives, scope, approach, and schedule, a detailed test plan is drawn out for implementation.

5.2 Testing Scenarios

Admin and User Authentication

To verify a user can successfully log in with valid credentials. To verify invalid log-in is reported with an alert message. Verifying token-based authentication ensures that protected pages are accessible only with a valid login. To verify when logged out, the token is destroyed and requires a valid log-in again.

User Details Management

To verify that a user has entered all the required fields.
To verify user details are stored correctly in the database.

Admin Dashboard Management

Verify that all user details are correctly displayed. Manage the community forum.

Post Updates Management

To verify all the uploaded posts are reflected in the frontend.

FAQ Management

FAQ session can be only modified by the admin.
User can only view those FAQs.

User Interface

To verify that all UI elements like buttons, forms, and input fields are working correctly.
To verify error messages are displayed appropriately in case of invalid inputs.

Security

To verify that passwords are protected. To verify that unauthorized accesses are prevented.

Performance

To verify that the system responds to user interactions and requests. To measure and analyze response times for different operations.

5.3 Unit Testing

- **React (Frontend):** Unit testing is performed using the Jest Framework. Jest works with React Testing Library to test individual components.

- **Node.js (Backend):** Unit testing is performed using the Jest or Mocha Framework. These frameworks test backend logic, including API endpoints and business logic.

5.4 Integral Testing

- **React (Frontend):** Integration testing is performed using Jest and React Testing Library. This ensures that different UI components work together correctly.
- **Node.js (Backend):** Integration testing is performed using Jest or Mocha with Supertest. This validates the interaction between API endpoints, business logic, and database queries.

5.5 Functional Testing

Functional Testing is done to verify that all the functionalities specified in software requirements are met. The functional testing can be done on various scenarios like:

User Authentication

- Checking on valid login with correct username and password.
- Displaying an error message for invalid login.
- Verifying logout so that user has to log in again for accessing features.

User Details Management

- Ensuring all required fields are entered properly.
- Verifying that all user details are stored in the database with a unique ID.

Admin Dashboard Management

- Verifying all user details fetched from database are displayed correctly in the dashboard.

- Verifying that changes made in the databases will be automatically displayed in the admin dashboard.
- Verifying admin can manage post,upload courses and add FAQs.

Post Updates Management

- To verify all the uploaded posts are reflected in the frontend.

FAQ Management

- FAQ session can be only modified by the admin.
- User can only view those FAQs.

User Interface

- To verify that all UI elements like buttons, forms, and input fields are working correctly.
- To verify error messages are displayed appropriately in case of invalid inputs.

Security

- To verify that passwords are protected.
- To verify that unauthorized accesses are prevented.

Performance

- To verify that the system responds to user interactions and requests.
- To measure and analyze response times for different operations.

Chapter 6

Results

Our Student Migration Website has proven to be an effective platform for students seeking guidance on international migration. It offers real-time chat functionality that enhances peer-to-peer interaction, fostering a supportive community. The website also provides a powerful country and university search feature, enabling users to explore their migration options with ease. Additionally, students can post information, share experiences, and access downloadable guidelines specific to different countries. The platform's user-friendly design and seamless integration of React, Node.js, and MongoDB ensure a smooth and efficient experience. Looking ahead, future enhancements include expanding support to more countries, introducing AI-driven course recommendations, and adding advanced features to further assist students in their migration journey. These developments will solidify the platform's role as a comprehensive tool for international students.

Chapter 7

Conclusion

The Student Migration Website successfully created a centralized platform where students can explore international study opportunities, interact with peers, and access essential migration-related resources. Key achievements include an efficient university and country search, real-time peer interaction through chat features, and easy access to downloadable country-specific migration guidelines. User feedback highlights a positive experience, emphasizing the platform's user-friendly interface, accessibility, and engagement. While the platform effectively addresses student migration concerns, enhancements such as AI-driven course recommendations, expanded university listings, improved chat features, and performance optimizations can further elevate its impact.

Looking ahead, the project has significant potential for growth. Future developments may include a dedicated mobile application for greater accessibility, integration of visa and immigration assistance through expert consultations, and collaborations with universities to offer exclusive scholarships and direct application processes. Additionally, implementing advanced analytics can provide insights into user behavior and migration trends, enabling continuous improvements. These enhancements will ensure the platform evolves into a more comprehensive and indispensable resource for aspiring international students.

Bibliography

[1] React Documentation:`https://react.dev/`.

[2] Firebase Documentation: `https://firebase.google.com`

[3] Tailwind CSS Documentation: `https://tailwindcss.com/docs`.

[4] Node.js: `https://nodejs.org/api`

APPENDIX

Code

App.tsx

```
import React, { useState, useEffect } from "react";
import { Routes, Route, Navigate } from "react-router-dom";
import axios from "axios";

import SignIn from "./components/SignIn";
import SignUp from "./components/SignUp";
import ForgotPassword from "./components/ForgotPassword";
import Navbar from "./components/Navbar";
import AdminDashboard from "./components/AdminDashboard/
   AdminDashboard";
import Forum from "./components/Forum";
import FAQ from "./components/FAQ";
import HomePage from "./components/Home";
import Usapage from "./components/Usapage";
import Ukpage from "./components/Ukpage";
import Canadapage from "./components/Canadapage";
import Australiapage from "./components/AustraliaPage";
import Germanypage from "./components/Germanypage";
import NetworkAnimation from "./components/NetworkAnimation";
import StartPage from "./components/Startpage";
import { auth } from "./backend/firebase";
import { onAuthStateChanged, User as FirebaseUser } from "
   firebase/auth";
import UsersPage from "./components/ViewUsers";

interface AuthState {
  isAuthenticated: boolean;
  isAdmin: boolean;
  mongoId?: string; // Add MongoDB _id
}

export interface User {
  name: string;
  email: string;
  phone: string;
  gitid?: string;
  countriesChosen: string[];
  courses: string[];
  universities: string[];
  mongoId?: string; // Add MongoDB _id to User
}

const App: React.FC = () => {
  const [authState, setAuthState] = useState<AuthState>({
    isAuthenticated: false,
    isAdmin: false,
  });
  const [activeTab, setActiveTab] = useState<string>("home");
  const [user, setUser] = useState<User | null>(null);

  useEffect(() => {
    const unsubscribe = onAuthStateChanged(
      auth,
      async (firebaseUser: FirebaseUser | null) => {
        if (firebaseUser) {
          const email = firebaseUser.email || "";
          const isAdmin = email === "admin@gmail.com";

          // Check MongoDB for user
          try {
            const response = await axios.post(
              "http://localhost:5000/api/auth/check-user",
              {
                uid: firebaseUser.uid,
              }
            );
            const mongoId = response.data._id;

            setAuthState({ isAuthenticated: true, isAdmin,
               mongoId });
            sessionStorage.setItem("mongoId", mongoId);

            const loggedInUser: User = isAdmin
              ? {
                  name: "Admin Lisa",
                  email,
                  phone: "1234567890",
                  countriesChosen: [],
                  courses: [],
                  universities: [],
                  mongoId,
                }
              : {
                  name: firebaseUser.displayName || "User",
                  email,
                  phone: "",
                  countriesChosen: [],
                  courses: [],
                  universities: [],
                  mongoId,
                };

            setUser(loggedInUser);
          } catch (error) {
            console.error("MongoDB check failed:", error);
            setAuthState({ isAuthenticated: false, isAdmin: false
                });
            setUser(null);
            sessionStorage.removeItem("mongoId");
          }
        } else {
          setAuthState({ isAuthenticated: false, isAdmin: false
              });
          setUser(null);
          sessionStorage.removeItem("mongoId");
        }
      }
    );

    // Check session storage on mount
    const mongoId = sessionStorage.getItem("mongoId");
    if (mongoId && !authState.isAuthenticated) {
      // Optionally verify session with backend if needed
    }

    return () => unsubscribe();
  }, []);
```

```
const handleSignIn = async (email: string, isAdmin: boolean) =>
    {
  const mongoId = sessionStorage.getItem("mongoId") || "";
  setAuthState({ isAuthenticated: true, isAdmin, mongoId });

  const normalUser: User = {
    name: "Alex Johnson",
    email,
    gitid: "alexjohnson",
    countriesChosen: ["UK", "Australia"],
    courses: ["Business Administration", "Marketing"],
    universities: ["University of Oxford", "University of
       Melbourne"],
    phone: "",
    mongoId,
  };

  const adminUser: User = {
    name: "Admin Lisa",
    email,
    phone: "1234567890",
    countriesChosen: [],
    courses: [],
    universities: [],
    mongoId,
  };

  setUser(isAdmin ? adminUser : normalUser);
};

const handleSignUp = (mongoId: string) => {
  const newUser: User = {
    name: "New SignUp User",
    email: "signup.user@example.com",
    phone: "5555555555",
    countriesChosen: ["Canada"],
    courses: ["Computer Science"],
    universities: ["University of Toronto"],
    mongoId,
  };
  setAuthState({ isAuthenticated: true, isAdmin: false, mongoId
      });
  setUser(newUser);
};

const handleLogout = () => {
  setAuthState({ isAuthenticated: false, isAdmin: false });
  setUser(null);
  setActiveTab("home");
  sessionStorage.removeItem("mongoId");
};

const renderContent = () => {
  switch (activeTab) {
    case "forum":
      return <Forum />;
    case "faq":
      return <FAQ />;
    default:
      return <HomePage />;
  }
};
```

```
const ProtectedRoute: React.FC<{
  children: React.ReactNode;
  showNavbar?: boolean;
  requireAdmin?: boolean;
}> = ({ children, showNavbar = true, requireAdmin = false }) =>
    {
  if (!authState.isAuthenticated || !sessionStorage.getItem("
     mongoId")) {
    return <Navigate to="/signin" replace />;
  }
  if (requireAdmin && !authState.isAdmin) {
    return <Navigate to="/dashboard" replace />;
  }

  return (
    <div className="min-h-screen relative z-10">
      {showNavbar && (
        <div className="bg-gradient-to-r from-blue-900/20 to-
           purple-900/20">
          <Navbar
            activeTab={activeTab}
            setActiveTab={setActiveTab}
            onLogout={handleLogout}
            isAdmin={authState.isAdmin}
            user={user}
          />
        </div>
      )}
      {children}
    </div>
  );
};
```

```
return (
  <div>
    <NetworkAnimation />
    <div className="min-h-screen text-gray-900 dark:text-gray
       -100 relative z-10">
      <Routes>
        <Route path="/" element={<StartPage />} />
        <Route
          path="/signin"
          element={
            authState.isAuthenticated ? (
              <Navigate
                to={authState.isAdmin ? "/admin-dashboard" : "/
                   dashboard"}
                replace
              />
            ) : (
              <SignIn onSignIn={handleSignIn} />
            )
          }
        />
        <Route
          path="/signup"
          element={
            authState.isAuthenticated ? (
              <Navigate to="/dashboard" replace />
            ) : (
              <SignUp onSignUp={handleSignUp} />
            )
          }
        />
```

```
          <Route path="/forgot-password" element={<ForgotPassword
              />} />
          <Route
            path="/admin-dashboard/*"
            element={
              <ProtectedRoute requireAdmin={true}>
                <AdminDashboard onLogout={handleLogout} />
              </ProtectedRoute>
            }
          />
          <Route
            path="/dashboard"
            element={
              <ProtectedRoute>
                <main className="container mx-auto px-4 py-8">
                  {renderContent()}
                </main>
              </ProtectedRoute>
            }
          />
          <Route
            path="/dashboard/usa"
            element={
              <ProtectedRoute showNavbar={false}>
                <Usapage />
              </ProtectedRoute>
            }
          />
          <Route
            path="/dashboard/uk"
            element={
              <ProtectedRoute showNavbar={false}>
```

```
        <Ukpage />
      </ProtectedRoute>
    }
  />
  <Route
    path="/dashboard/canada"
    element={
      <ProtectedRoute showNavbar={false}>
        <Canadapage />
      </ProtectedRoute>
    }
  />
  <Route
    path="/dashboard/australia"
    element={
      <ProtectedRoute showNavbar={false}>
        <Australiapage />
      </ProtectedRoute>
    }
  />
  <Route
    path="/dashboard/germany"
    element={
      <ProtectedRoute showNavbar={false}>
        <Germanypage />
      </ProtectedRoute>
    }
  />
  <Route
    path="/dashboard/user"
    element={
      <ProtectedRoute>
```

```
                <UsersPage />
              </ProtectedRoute>
            }
          />
          <Route path="*" element={<Navigate to="/" replace />}
             />
        </Routes>
      </div>
    </div>
  );
};

export default App;
```

Server.js

```
const express = require("express");
const dotenv = require("dotenv");
const cors = require("cors");
const mongoose = require("mongoose");
const http = require("http");
const { Server } = require("socket.io");

const authRoutes = require("./routes/auth");
const postRoutes = require("./routes/Post");
const chatRoutes = require("./routes/chat"); // Add chat routes
const userRoutes = require("./routes/user"); // Add chat routes

dotenv.config();

const app = express();
const server = http.createServer(app);
const io = new Server(server, {
  cors: {
    origin: "http://localhost:5173",
    methods: ["GET", "POST"]
  }
});

app.use(express.json());
app.use(cors());

// Routes
app.use("/api/auth", authRoutes);
app.use("/api/Post", postRoutes);
app.use("/api/chat", chatRoutes(io)); // Pass io to chat routes
app.use("/api/users", userRoutes); // Added users route

// Socket.IO connection handling
io.on("connection", (socket) => {
  console.log("    New client connected:", socket.id);

  // Join user's own room based on their userId
  socket.on('join', (userId) => {
    socket.join(userId.toString());
    console.log(`User ${userId} joined their room`);
  });

  socket.on("disconnect", () => {
    console.log("    Client disconnected:", socket.id);
  });
});
```

```
mongoose
  .connect(process.env.MONGO_URI)
  .then(() => {
    console.log("    MongoDB Connected...");
    server.listen(5000, () => console.log("       Server running
       on port 5000"));
  })
  .catch((error) => console.error("    MongoDB Connection Error:"
     , error));
```

Screenshots

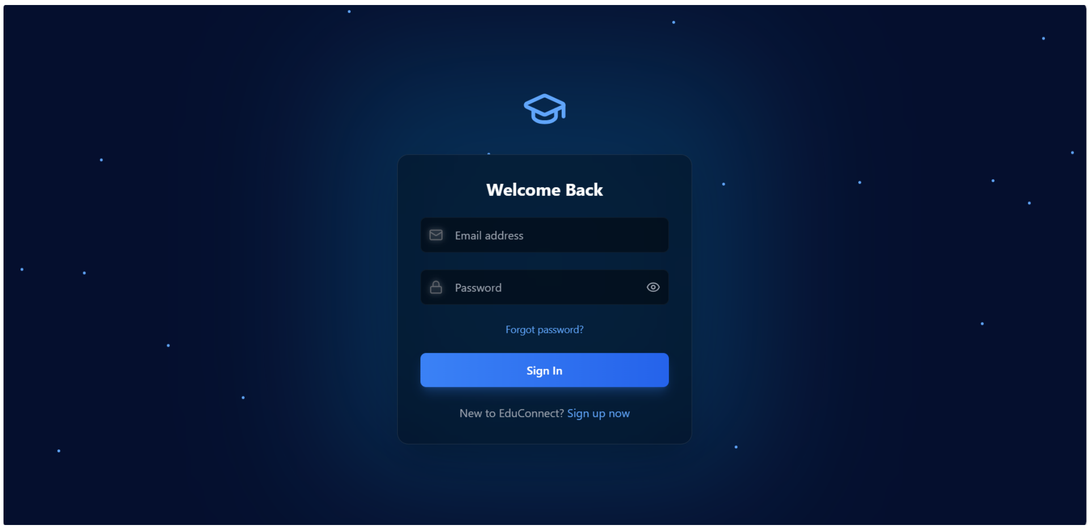


Figure 7.1: Login Page

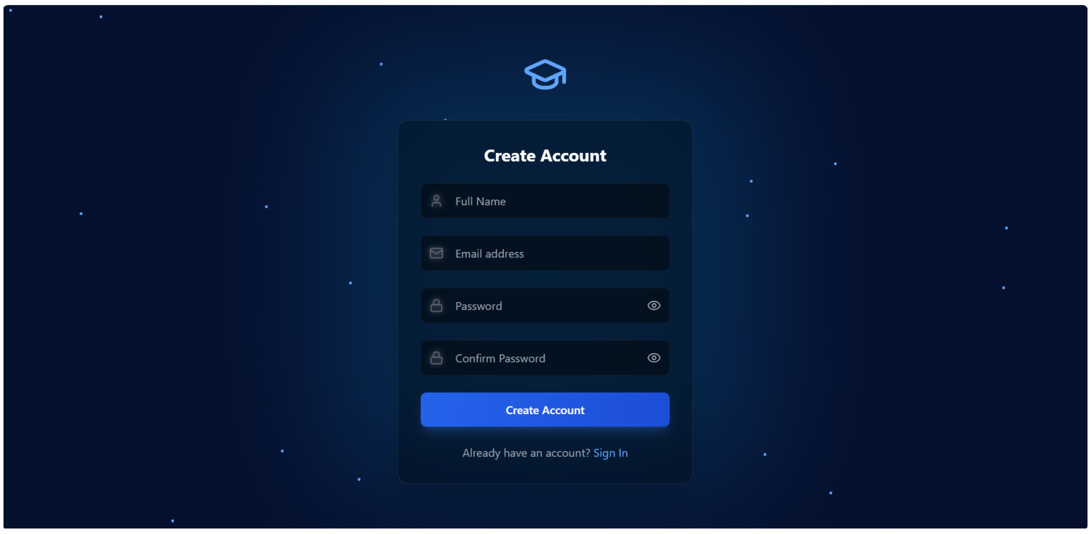
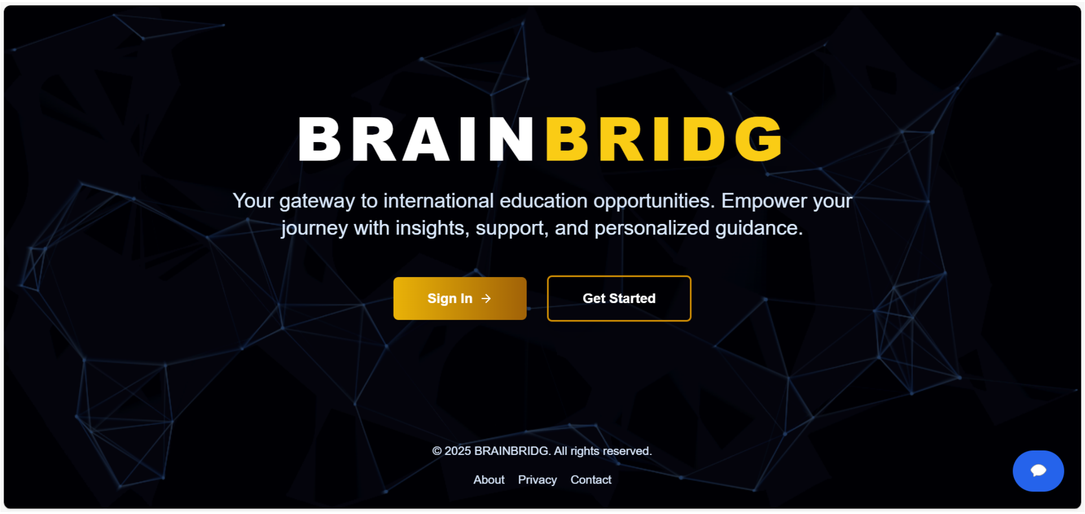


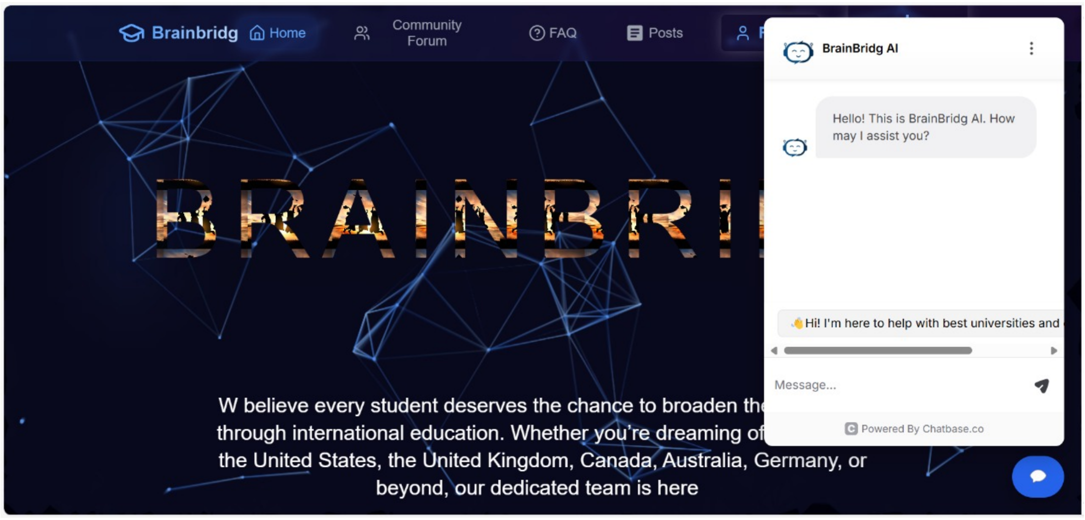
Figure 7.2: Register Page



Your gateway to international education opportunities. Empower your journey with insights, support, and personalized guidance.

© 2025 BRAINBRIDG. All rights reserved.

Figure 7.3: Landing Page

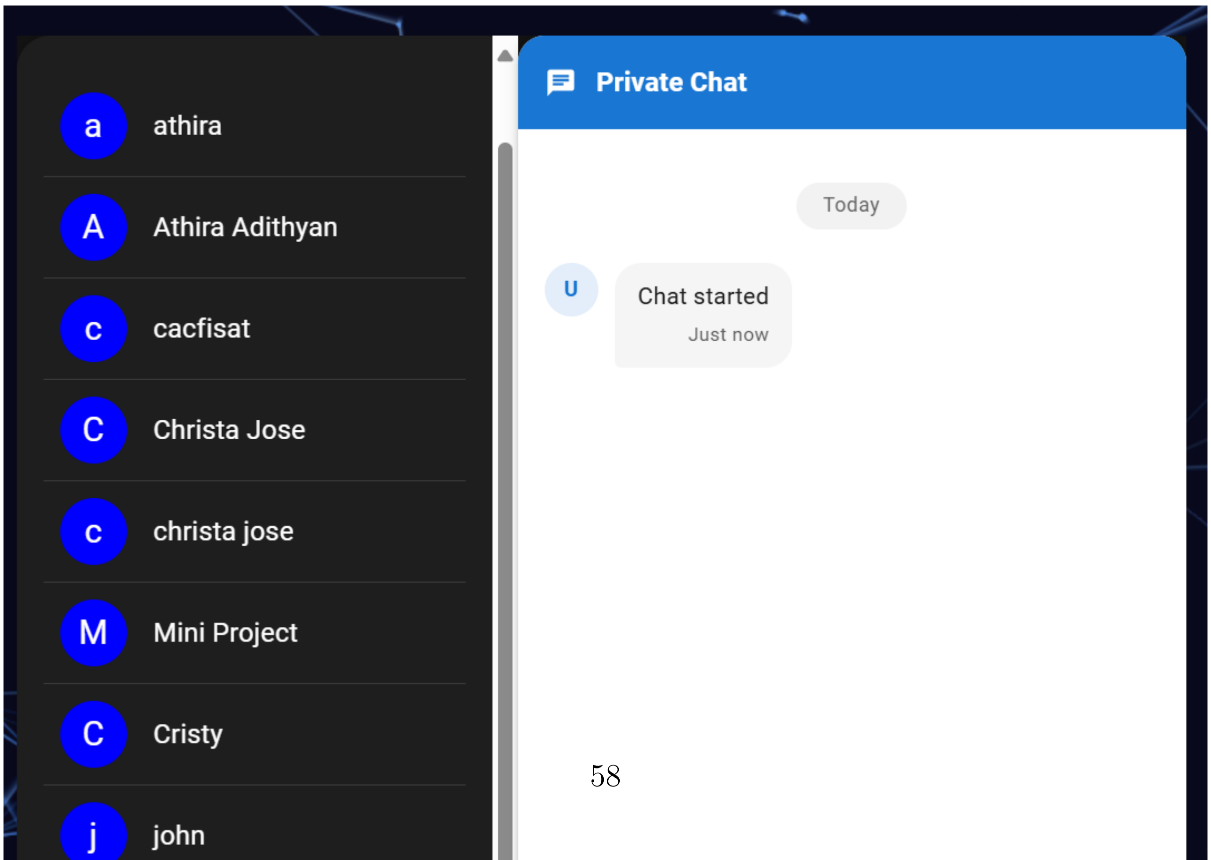


Hello! This is BrainBridg AI. How may I assist you?

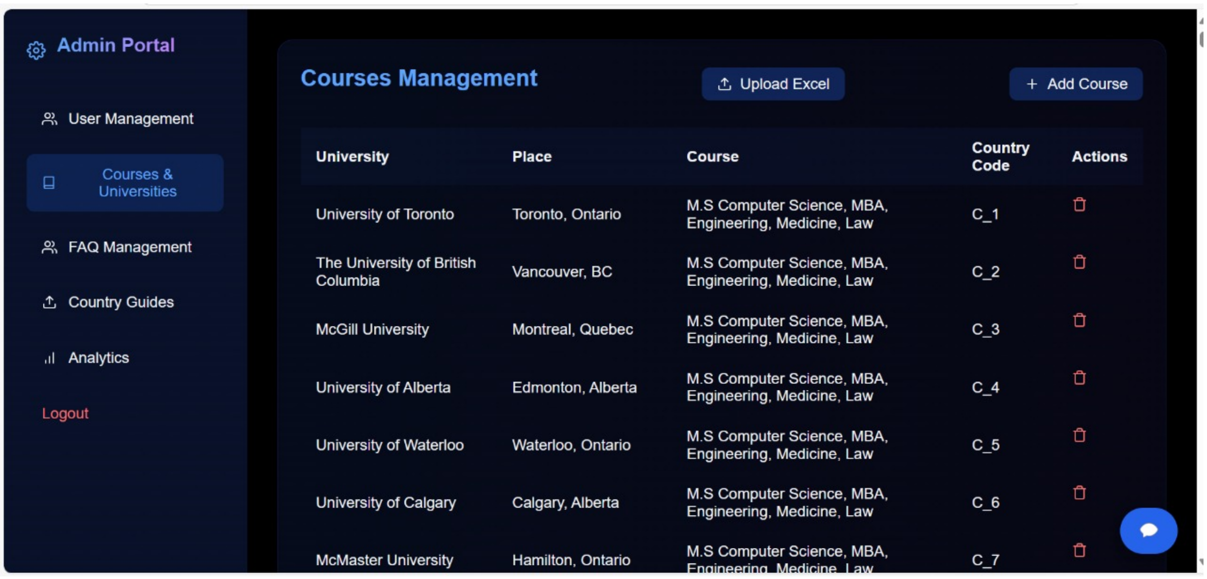
👋Hi! I'm here to help with best universities and

W believe every student deserves the chance to broaden the through international education. Whether you're dreaming of the United States, the United Kingdom, Canada, Australia, Germany, or beyond, our dedicated team is here

Figure 7.4: Home Page with Chatbot

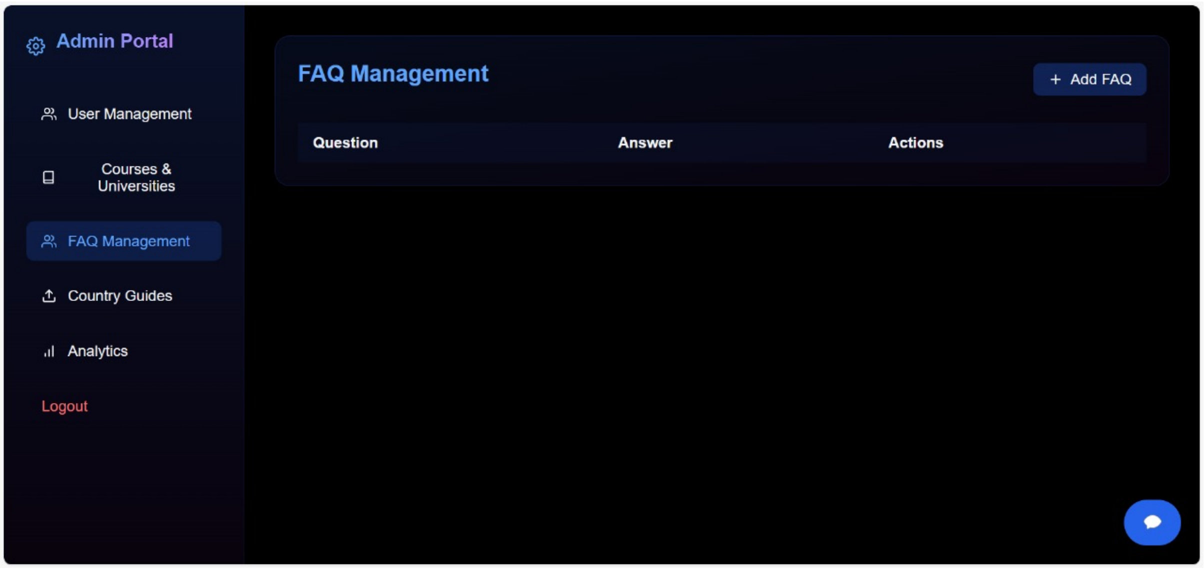


Chat started
Just now



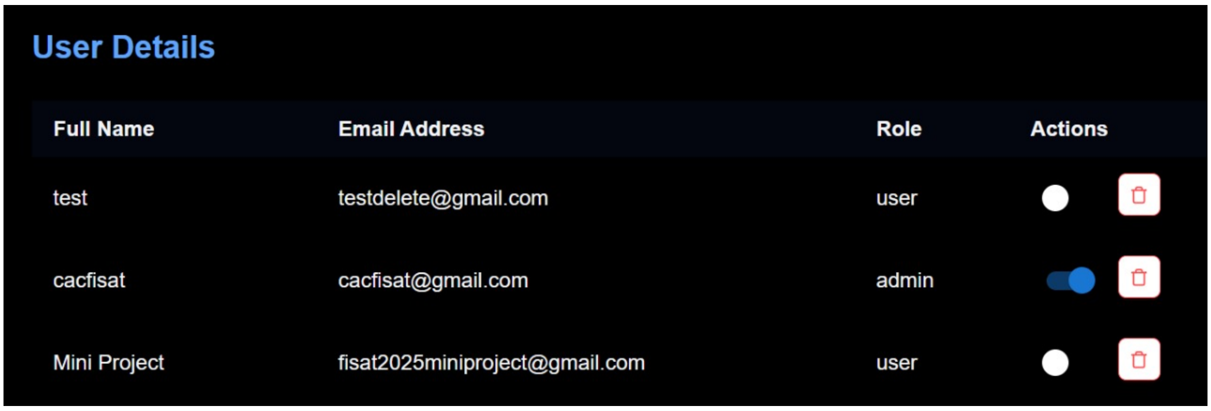
University	Place	Course	Country Code	Actions
University of Toronto	Toronto, Ontario	M.S Computer Science, MBA, Engineering, Medicine, Law	C_1	🗑
The University of British Columbia	Vancouver, BC	M.S Computer Science, MBA, Engineering, Medicine, Law	C_2	🗑
McGill University	Montreal, Quebec	M.S Computer Science, MBA, Engineering, Medicine, Law	C_3	🗑
University of Alberta	Edmonton, Alberta	M.S Computer Science, MBA, Engineering, Medicine, Law	C_4	🗑
University of Waterloo	Waterloo, Ontario	M.S Computer Science, MBA, Engineering, Medicine, Law	C_5	🗑
University of Calgary	Calgary, Alberta	M.S Computer Science, MBA, Engineering, Medicine, Law	C_6	🗑
McMaster University	Hamilton, Ontario	M.S Computer Science, MBA, Engineering, Medicine, Law	C_7	🗑

Figure 7.5: Admin Course View



Question	Answer	Actions

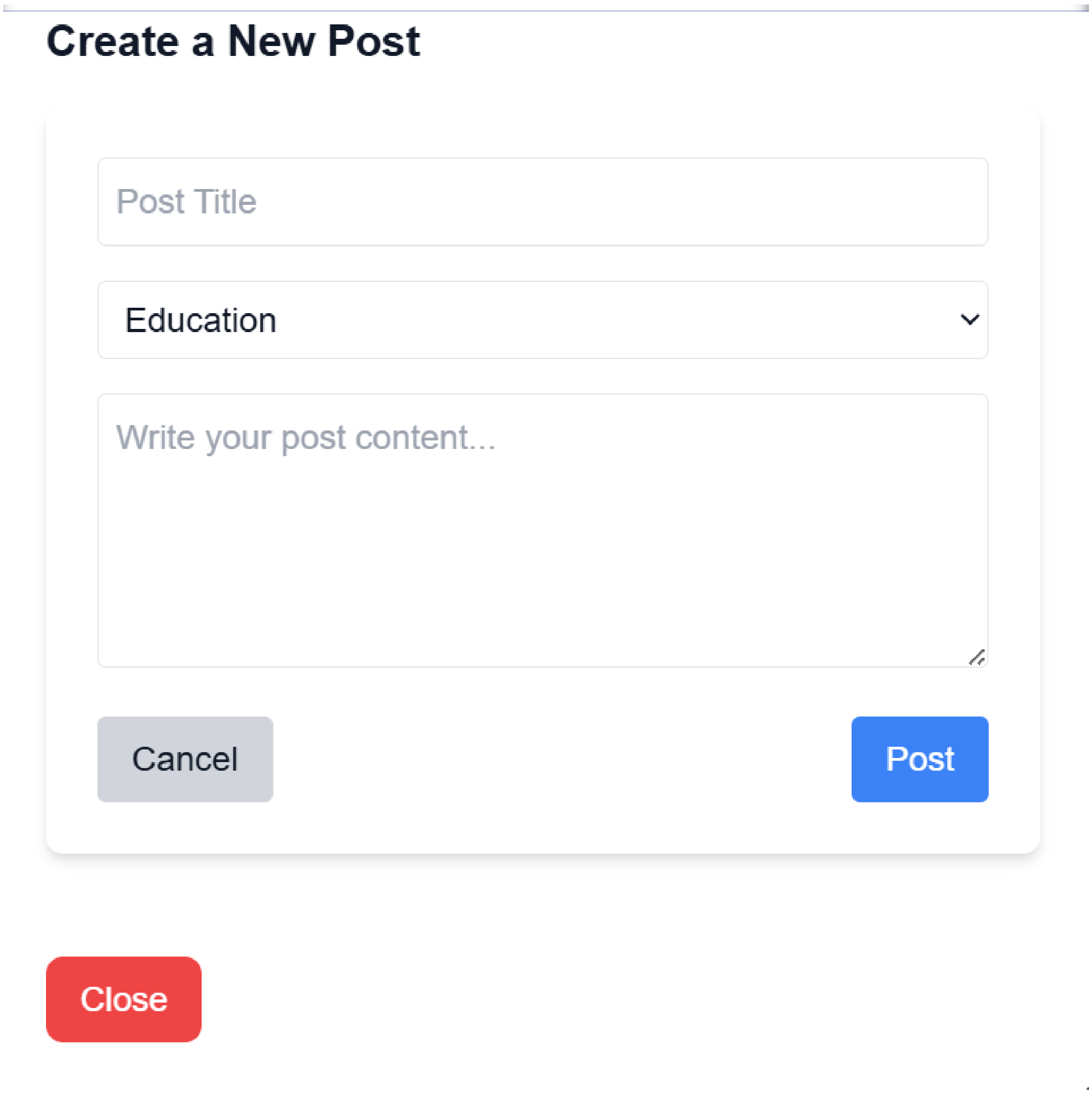
Figure 7.6: Admin FAQ



User Details

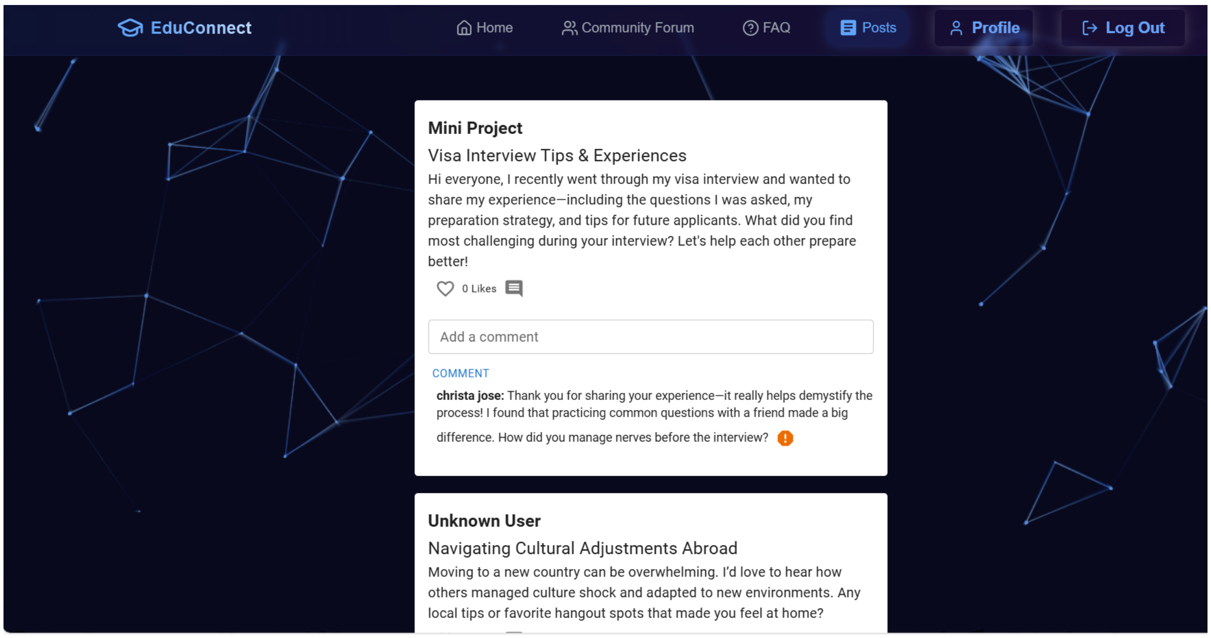
Full Name	Email Address	Role	Actions
test	testdelete@gmail.com	user	
cacfisat	cacfisat@gmail.com	admin	
Mini Project	fisat2025miniproject@gmail.com	user	

Figure 7.7: Admin User



Create a New Post

Figure 7.8: Post in Forum



Mini Project

Visa Interview Tips & Experiences

Hi everyone, I recently went through my visa interview and wanted to share my experience—including the questions I was asked, my preparation strategy, and tips for future applicants. What did you find most challenging during your interview? Let's help each other prepare better!

0 Likes

Add a comment

COMMENT

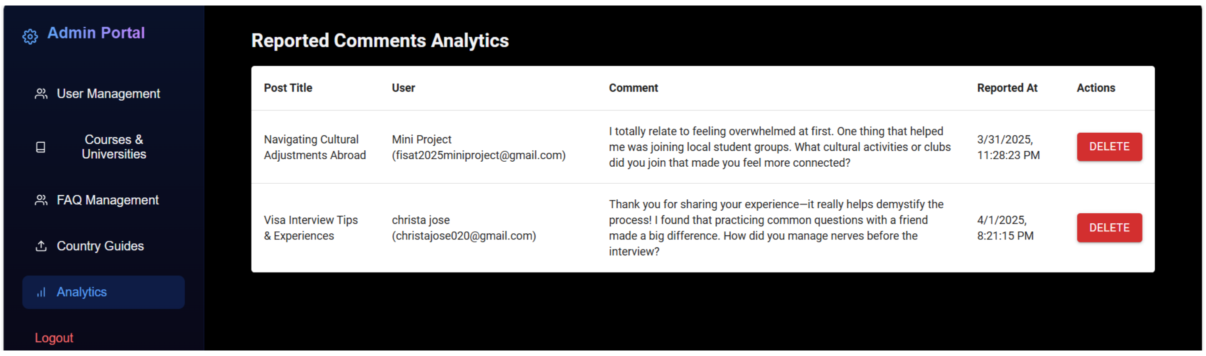
christa jose: Thank you for sharing your experience—it really helps demystify the process! I found that practicing common questions with a friend made a big difference. How did you manage nerves before the interview?

Unknown User

Navigating Cultural Adjustments Abroad

Moving to a new country can be overwhelming. I'd love to hear how others managed culture shock and adapted to new environments. Any local tips or favorite hangout spots that made you feel at home?

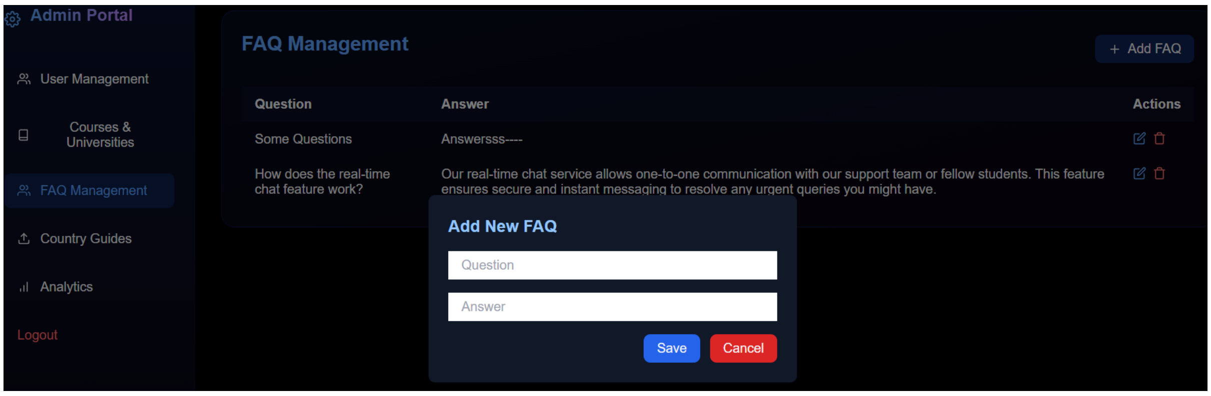
Figure 7.9: User Posts



Reported Comments Analytics

Post Title	User	Comment	Reported At	Actions
Navigating Cultural Adjustments Abroad	Mini Project (fisat2025miniproject@gmail.com)	I totally relate to feeling overwhelmed at first. One thing that helped me was joining local student groups. What cultural activities or clubs did you join that made you feel more connected?	3/31/2025, 11:28:23 PM	DELETE
Visa Interview Tips & Experiences	christa jose (christajose020@gmail.com)	Thank you for sharing your experience—it really helps demystify the process! I found that practicing common questions with a friend made a big difference. How did you manage nerves before the interview?	4/1/2025, 8:21:15 PM	DELETE

Figure 7.10: Admin Forum Post Management


Admin Portal

User Management
Courses & Universities
FAQ Management
Country Guides
Analytics
Logout

FAQ Management

+ Add FAQ

Question	Answer	Actions
Some Questions	Answersss----	
How does the real-time chat feature work?	Our real-time chat service allows one-to-one communication with our support team or fellow students. This feature ensures secure and instant messaging to resolve any urgent queries you might have.	

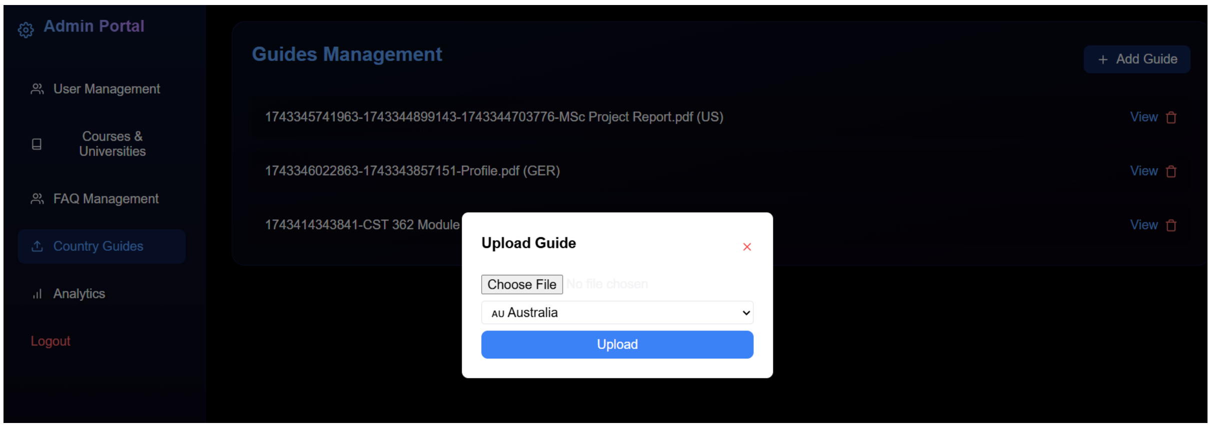
Add New FAQ

Question

Answer

Save Cancel

Figure 7.11: FAQ Management


Admin Portal

User Management
Courses & Universities
FAQ Management
Country Guides
Analytics
Logout

Guides Management

+ Add Guide

1743345741963-1743344899143-1743344703776-MSc Project Report.pdf (US) View

1743346022863-1743343857151-Profile.pdf (GER) View

1743414343841-CST 362 Module View

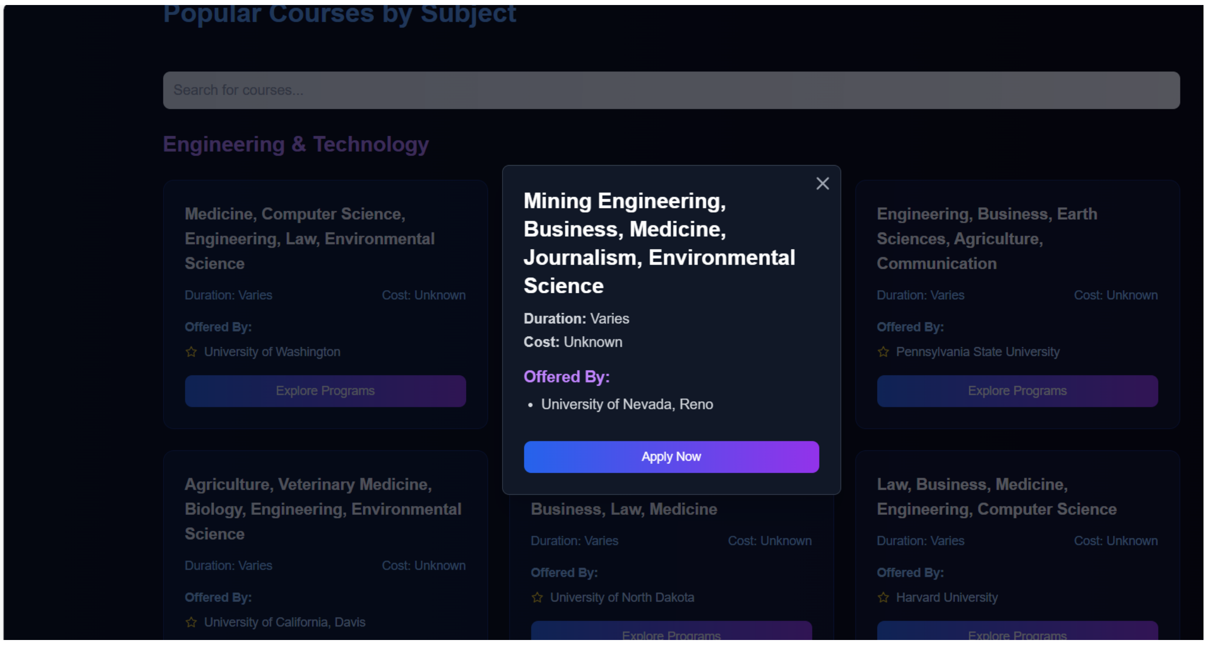
Upload Guide ×

Choose File No file chosen

AU Australia

Upload

Figure 7.12: Country guides


Popular Courses by Subject

Search for courses...

Engineering & Technology

Medicine, Computer Science, Engineering, Law, Environmental Science
Duration: Varies Cost: Unknown
Offered By:
☆ University of Washington
Explore Programs

Mining Engineering, Business, Medicine, Journalism, Environmental Science
Duration: Varies
Cost: Unknown
Offered By:
- University of Nevada, Reno

Apply Now

Engineering, Business, Earth Sciences, Agriculture, Communication
Duration: Varies Cost: Unknown
Offered By:
☆ Pennsylvania State University
Explore Programs

Agriculture, Veterinary Medicine, Biology, Engineering, Environmental Science
Duration: Varies Cost: Unknown
Offered By:
☆ University of California, Davis

Business, Law, Medicine
Duration: Varies Cost: Unknown
Offered By:
☆ University of North Dakota
Explore Programs

Law, Business, Medicine, Engineering, Computer Science
Duration: Varies Cost: Unknown
Offered By:
☆ Harvard University
Explore Programs

Figure 7.13: Country Page:Courses & Universities