

# Dakhla – AI-Assisted University Admissions Platform

## Project Proposal



**Supervisor**  
**Dr. Umer Farooq**

**Submitted by**  
**Muhammad Nauman Arif**  
{2514-2022 / IT-22-179}  
**Ismail Dad Khan**  
{2927-2022 / IT-22-417}  
**Hamza Ali Khan**  
{2508-2022 / IT-22-212}

**Department of Computer Science,**  
Hamdard University, Karachi.

[26/05/2025]

## 1. Introduction

The university admissions process in Pakistan is fragmented and inefficient. Students are often required to visit multiple university websites to gather information about programs, deadlines, and eligibility criteria, managing everything manually. This lack of a centralized source leads to confusion, missed opportunities, and poor decision-making—especially for students in underserved or rural areas.

**Dakhla** is a web-based, AI-assisted platform that addresses this gap by centralizing admissions data and simplifying the student journey. It allows students to search and compare universities, receive personalized recommendations, interact with an intelligent chatbot (DakhlaBot), and apply directly to multiple universities using a unified application profile. On the institutional side, Dakhla enables universities to manage program listings, view filtered applicant profiles, and engage with prospective students, making the admissions process more transparent, accessible, and efficient for all stakeholders.

## 2. Objective

To design and develop a centralized, AI-assisted web platform that streamlines the university admissions process in Pakistan by providing:

- A unified platform where students can explore detailed, up-to-date information about universities, including available programs, eligibility criteria, deadlines, fee structures, and campus locations.
- An **AI-powered recommendation** system that suggests suitable universities based on each student's academic record, preferred discipline, and location preferences.
- An intelligent chatbot (**DakhlaBot**) that assists users in navigating the platform, understanding admission processes, and receiving instant responses to frequently asked questions.
- A **DirectApply** feature that allows students to submit a unified application profile and supporting documents to multiple universities from within the platform.
- A university-facing dashboard for managing program listings, reviewing applicant profiles, and reaching out to potential candidates based on pre-set filters.

## 3. Problem Description

In Pakistan, the university admissions process is disjointed, inefficient, and heavily manual. Students must visit multiple university websites to

gather essential information such as program offerings, eligibility criteria, deadlines, and fees. This decentralized and inconsistent flow of information often leads to confusion, missed deadlines, and uninformed academic decisions—particularly for students in rural or underrepresented areas who lack access to counseling or digital resources.

At the same time, universities struggle to effectively reach and attract the right applicants. Without a centralized channel to view and filter student profiles, institutions are limited in their outreach and often receive inconsistent or incomplete applications.

**Dakhla** addresses this gap by offering a unified, AI-powered admissions platform. It enables students to explore and compare universities, receive personalized recommendations, and apply to multiple institutions using a single, reusable application profile. For universities, Dakhla provides tools to manage program data, filter applicants, and communicate directly with prospective students. This approach enhances transparency, saves time, and improves the overall quality and accessibility of the admissions process.

## 4. Comprehensive Literature Review

Centralized university admissions platforms have played a transformative role in streamlining higher education enrollment in various countries. Among the most prominent examples is the **UCAS (Universities and Colleges Admissions Service)** used in the United Kingdom. UCAS allows students to apply to multiple universities through a single online application, offering structured features such as deadline management, program comparison, document submissions, and real-time tracking of application statuses [1].

Similarly, the **Common Application (Common App)** in the United States simplifies the admissions process by allowing students to submit one standardized profile to hundreds of colleges. The platform also supports additional institution-specific questions and documents, making it both flexible and scalable [2]. These systems improve accessibility, enhance transparency, and reduce the administrative burden for both students and academic institutions.

In the Pakistani context, platforms such as **Ilmkidunya** and **Eduvision** attempt to support university admissions by providing lists of available programs, admission deadlines, and general guidance [3][4]. However, their scope is limited to content aggregation. They do not offer features like user-specific recommendations, direct application submission, or institutional interaction. Moreover, they lack interactivity, real-time data updates, AI-based assistance, or centralized student profile management.

This gap indicates a clear opportunity for innovation. **Dakhla** is proposed to address these limitations through the following key enhancements:

- **AI-assisted university recommendations**, tailored to each student's academic profile and preferences.
- **DakhlaBot**, an intelligent chatbot that helps users navigate the platform and answers common admission-related queries in real time.
- **Direct Apply functionality**, enabling students to use a single, reusable profile to apply to multiple universities through one system.
- **University-side dashboards**, allowing institutions to manage listings, view filtered applicants, and communicate directly with prospective students.

## 5. Methodology

Our project follows the **Agile software development methodology**, allowing us to iterate quickly, gather feedback regularly, and refine the platform in manageable sprints. The development will be divided into frontend and backend components, with clear milestones across two semesters.

### Technical Stack & Development Plan

- **Frontend:** Developed using **React.js** in combination with **Tailwind CSS**, ensuring a responsive and user-friendly interface across devices.
- **Backend:** Implemented using **Python** with **Flask** (or optionally Django), providing secure REST APIs to handle data flow, user sessions, form submissions, and role-based access control.
- **Database:** We will use **PostgreSQL** to manage structured data, including university listings, user profiles, application submissions, and institutional records.
- **Authentication:** User login and session management will be handled via **JWT (JSON Web Tokens)** and/or **Firebase Auth** to manage secure login sessions for students and institutional users, with role-specific access and dashboard views..
- **AI Recommendation Engine:** A rule-based system in the first phase that recommends universities based on academic records, preferred programs, and city.
- **DirectApply Module:** Students can fill out one unified application form, attach required documents, and apply to multiple universities through our platform.
- **Chatbot Integration (DakhlaBot):** Integrated using **Dialogflow**, **OpenAI**, or **Gemini API** allowing students to receive instant responses to FAQs, platform navigation help, and admission guidance.
- **Admin and University Dashboards:** Institutions will have access to a dedicated dashboard to publish programs, manage

requirements, review submitted applications, filter applicants, and send updates or messages.

### Automated Data Scraping

To populate university information at scale, we will implement **automated web scraping** using **Python (BeautifulSoup, Scrapy or Selenium)**. The scraper will:

- Periodically crawl verified public university websites
- Extract data such as programs, admission deadlines, fee structures, and eligibility criteria
- Automatically structure and insert this data into the PostgreSQL database
- Include logging and exception handling to monitor scraping performance and avoid data duplication or loss
- Be scheduled to run at defined intervals using cron jobs or background workers

### Supporting Tools

- **Version Control:** GitHub
- **Design:** Figma (UI/UX), Canva
- **API Testing:** Postman
- **Project Management:** GitHub Projects, Slack
- **Hosting/Deployment:** Render, Vercel, or Firebase (for deployment during development)

## 6. Project Scope

The scope of *Dakhla* is defined to ensure the platform delivers its core value without becoming overly complex during the initial development phase.

### Included in Scope

- **Centralized university directory:** A searchable and filterable listing of Pakistani universities, including key information such as program offerings, eligibility criteria, admission deadlines, fees, and campus locations.
- **Student account system:** User registration, login, and a personalized dashboard for managing academic profiles, saved universities, and application tracking.
- **AI-based university recommendation engine:** A rule-based system that recommends universities based on academic background, location preferences, and field of interest.
- **DakhlaBot (AI chatbot):** A conversational assistant that helps students navigate the platform, understand university requirements, and receive quick answers to admission-related queries.

- **DirectApply functionality:** A single reusable application form that can be submitted to multiple universities, with document upload and status tracking support.
- **University dashboard:** A dedicated dashboard for university representatives to manage their listings, receive and review student applications, and communicate with applicants.
- **Automated data scraping pipeline:** Scheduled scraping of publicly available university information to populate and update the directory with minimal manual effort.

#### Excluded from Scope

- **Mobile App:** A mobile-optimized version of the platform will be provided, but native Android/iOS apps will not be developed in the FYP phase.
- **Integrated Online Payments:** Payment gateways for application fees (e.g., JazzCash, Easypaisa) are out of scope for this version.
- **Government or University API Integrations:** Automated connections with official education board databases (e.g., HEC) for document verification.

#### Assumptions

- Initial university data will be collected using automated scraping from public sources and manually validated where needed.
- A limited number of universities will be onboarded manually during pilot testing.
- Users (students and universities) will require internet access to use the platform.
- University participation in the Direct Apply system will be voluntary during the pilot phase.
- Academic records will be self-reported by applicants.

## 7. Feasibility Study

Considering the defined scope, we aim to meet our project schedule. Key aspects to consider include:

#### i. **Risks Involved:**

- **Data Collection Reliability:** University websites may have inconsistent structures or incomplete information, making scraping prone to errors.  
*Mitigation:* Develop flexible scraping scripts with exception handling and validation checks; manually review extracted data via admin panel.
- **University Onboarding:** Getting institutions to actively manage their profiles or accept direct applications may be slow initially.  
*Mitigation:* Begin with a curated list of universities and manually populate their data; initiate outreach with demos and value-driven pitches.

- **User Trust & Adoption:** Students may hesitate to rely on a new platform for important academic decisions.  
*Mitigation:* Build a clean, trustworthy interface; offer chatbot assistance; pilot with known users and get feedback early.
- **Technical Complexity:** Integrating components like scraping, chatbot, AI recommendations, and multi-role dashboards may create architectural challenges.  
*Mitigation:* Use modular, API-based architecture; test components independently; follow Agile sprints with milestones.

## ii. Resource Requirement:

- **Hardware:**
  - Three development laptops
  - Stable internet connection.
- **Software Tools:**
  - **Frontend:** React.js + Tailwind CSS
  - **Backend:** Python (Flask or Django)
  - **Database:** PostgreSQL
  - **AI & Chatbot:** OpenAI API, Dialogflow (optional: Ollama/Gemini)
  - **Scraping:** BeautifulSoup/Scrapy/Selenium
  - **Design & Planning:** Figma, Canva
  - **Testing:** Postman
  - **Version Control:** GitHub
  - **Deployment:** Render, Vercel, or Firebase (free-tier)
  - **External APIs (Optional):** Firebase Authentication (for login/session handling), OpenAI or Dialogflow

## 8. Solution Application Areas

*Dakhla* is designed to serve as a practical, scalable solution for streamlining the university admissions process in Pakistan. It addresses key pain points faced by both students and educational institutions and can be deployed across a wide range of educational and administrative environments.

### For Students (Primary Users)

- **High School Graduates (FSc/FA/A-Level):** Students preparing for undergraduate admissions will use *Dakhla* to research universities, receive AI-driven recommendations, and apply through a centralized profile.
- **Students from Underserved or Rural Areas:** *Dakhla* provides equal access to verified university information and application tools, especially for those lacking counseling or reliable internet search experience.
- **Career Counselors and Parents:** Counselors in schools and coaching centers can use the platform to guide students. Parents can explore and shortlist institutions alongside their children using the same student profile.

### For Universities & Institutions

- **Public and Private Universities:**  
Institutions can attract a more diverse applicant pool, receive organized and filtered student profiles, and manage incoming applications through a dedicated dashboard.
- **Admissions Offices & Marketing Teams:**  
University personnel responsible for recruitment and outreach can monitor student interest, respond to applications, and communicate with high-potential candidates directly via the platform.

### Impact Areas

- **Education Technology (EdTech):**  
Dakhla contributes to digital transformation in the higher education sector of Pakistan.
- **Digital Inclusion:**  
Bridges the digital gap for students with limited resources or technological access.
- **Higher Education Accessibility:**  
Makes the admissions process more transparent, centralized, and student-friendly across provinces and boards.

## 9. Tools/Technology

The Dakhla platform will be developed using a modern full-stack architecture, combining scalable backend technologies, responsive frontend frameworks, and powerful automation tools. The selected tools ensure reliability, performance, and ease of collaboration across the development lifecycle.

- **Frontend:** React.js, Tailwind CSS / Bootstrap, Figma
- **Backend:** Python (Flask/Django), Node.js, PostgreSQL
- **Automated Data Collection:** BeautifulSoup, Scrapy & Selenium (Python), Schedule / CRON Jobs
- **Chatbot:** OpenAI, Dialogflow, Gemini API, Ollama
- **Authentication:** JWT, Firebase Auth
- **Project Management:** GitHub Projects, Slack
- **Testing:** Postman

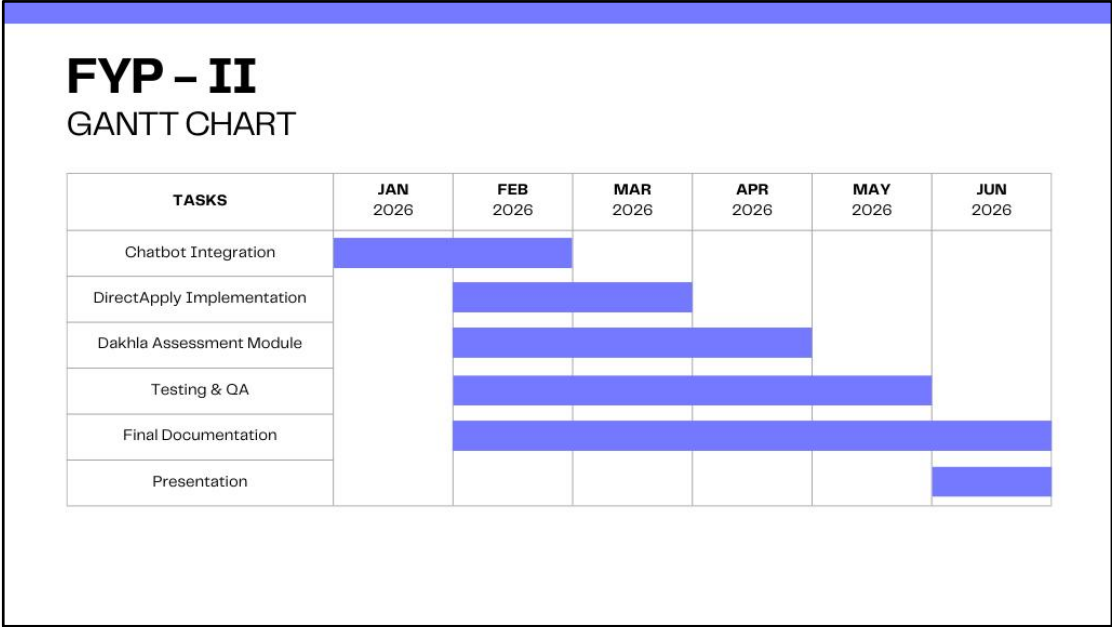
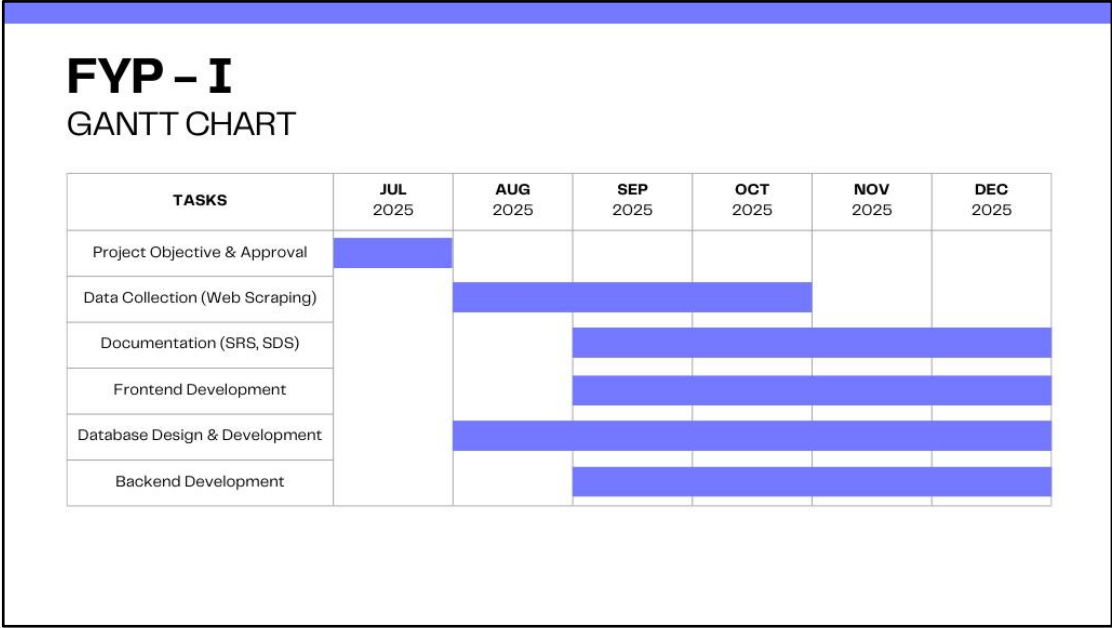


## 10. Responsibilities of the Team Members

### RACI Matrix

|                                      | M. Nauman Arif | Ismail Dad Khan | Hamza Ali Khan | Supervisor |
|--------------------------------------|----------------|-----------------|----------------|------------|
| Workflow Development                 | Responsible    | Accountable     | Responsible    | Consulted  |
| Frontend Development                 | Accountable    | Responsible     | Responsible    | Informed   |
| Backend Development                  | Responsible    | Accountable     | Responsible    | Consulted  |
| Database Design & Development        | Responsible    | Accountable     | Responsible    | Consulted  |
| Data Collection (Automated Scraping) | Responsible    | Consulted       | Accountable    | Informed   |
| Chatbot Development                  | Consulted      | Accountable     | Responsible    | Consulted  |
| Testing & QA                         | Responsible    | Responsible     | Accountable    | Consulted  |
| GitHub Repository Management         | Responsible    | Consulted       | Accountable    | Informed   |
| Documentation                        | Accountable    | Responsible     | Responsible    | Informed   |
| Presentation                         | Consulted      | Responsible     | Responsible    | Informed   |

11. Planning



## 12. References

1. **UCAS.** "How to apply through UCAS." 2024. UCAS Ltd. Accessed May 2025. <https://www.ucas.com>
2. **Common App.** "Apply to college with Common App." 2024. Common Application Inc. Accessed May 2025. <https://www.commonapp.org>
3. **Ilmkidunya.** "Admissions in Pakistani universities." 2024. Accessed May 2025. <https://www.ilmkidunya.com>
4. **Eduvision.** "Career planning and university guidance." 2024. Eduvision Pakistan. Accessed May 2025. <https://www.eduvision.edu.pk>
5. **OpenAI.** "ChatGPT API documentation." OpenAI. Accessed May 2025. <https://platform.openai.com/docs>
6. **BeautifulSoup Documentation.** "Web scraping library in Python." Accessed May 2025. <https://www.crummy.com/software/BeautifulSoup/>
7. **Flask Documentation.** "Flask Web Framework." Pallets Projects. Accessed May 2025. <https://flask.palletsprojects.com>
8. **PostgreSQL Documentation.** "PostgreSQL: The World's Most Advanced Open Source Relational Database." Accessed May 2025. <https://www.postgresql.org>
9. **Dialogflow.** "Build conversational experiences." Google Cloud. Accessed May 2025. <https://cloud.google.com/dialogflow>
10. **React.js Documentation.** "A JavaScript library for building user interfaces." Meta. Accessed May 2025. <https://react.dev>