

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### Final Year Project Proposal

For probably the first time in your undergraduate/graduate program, you are required to defend a proposal of a larger project. In teams, you will be working on the common project but individual team members will be required to take on responsibilities for specific work for which each will be held accountable. Interaction, collaboration and assistance are allowed and expected, but each person will receive an individual mark/grade for his/her work performed in the project.

	Day			Month			Year			
DATE	1	1	-	1	0	-	2	0	2	5

<b>PROJECT TITLE:</b>	Advisify-AI Course Advisor
<b>KEY WORDS:</b>	AI Course Advisor, Machine Learning, Academic Recommendation System, Student Registration, Course Prediction, FastAPI, Next.js, Educational Technology
<b>DOMAIN OF THE PROJECT:</b>	Artificial Intelligence, Web Application Development, Educational Technology (EdTech)
<b>SUPERVISOR'S NAME:</b>	Syed Haider Imam Jaffery

STUDENT INFORMATION				
Sr.	Student ID	Name	Email	Mobile
1.	FA22-BSCS-136	ALI RAZA BUGTI	razabugtiali@gmail.com	0315-1345256
2.	FA22-BSCS-0138	ARFAT AYUB	arfat.ayub2002@gmail.com	0316 7200030
3.	FA22-BSCS-0210	MUHAMMAD AHMED	muhammadahmedshykh@gmail.com	0325 6708757

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing*

*Department of Computer Science*

### PROBLEM STATEMENT

University students who fall behind due to course withdrawals, F-grades or semester breaks often struggle to choose the right combination of courses during re-registration. Faculty course advisors face heavy workloads during registration days, leading to delays, confusion, and poor course planning. There is currently no intelligent system to assist students in selecting optimal courses while ensuring academic prerequisites and improved CGPA outcomes.

### EXECUTIVE SUMMARY:

Advisify – AI Course Advisor is an intelligent web-based platform designed to help university students make smarter, data-driven decisions when selecting their semester courses. Many students face challenges after failing subjects, withdrawing from courses, or taking academic breaks. These setbacks make it difficult for them to plan future course registrations, while faculty advisors face high workloads and time pressure during registration periods. Advisify addresses both issues by automating the advising process through artificial intelligence. The system analyzes historical student performance data such as grades, completed subjects, and academic progression to recommend courses that align with a student's past performance and program structure. It also cross-checks each recommendation against the university's course outlines and prerequisites, ensuring that suggestions are both academically valid and beneficial for CGPA improvement. Once students review and accept the suggested courses, their selections are automatically forwarded to their assigned faculty advisor for approval, simplifying the entire registration process. Advisify consists of three main modules: **Student**, **Faculty**, and **Admin**. The Student Module enables learners to upload their academic data and receive AI-generated course recommendations. The Faculty Module allows advisors to view student requests and approve or modify suggested plans. The Admin Module manages batches, courses, and user roles while maintaining the system's overall structure and security. Developed using **Next.js** for the unified web application and **FastAPI** for AI integration, Advisify combines modern web development and machine learning to create a fast, reliable, and intelligent academic advising experience. The system reduces manual workload, minimizes human error, and supports better academic outcomes, representing a meaningful step toward digital transformation in higher education.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing*

*Department of Computer Science*

### INTRODUCTION

In many universities, academic advising plays a crucial role in guiding students toward the successful completion of their degree programs. However, the traditional advising process is often manual, time-consuming, and prone to human error. Students who have fallen behind in their academic progress due to F-grades, course withdrawals, or semester breaks often struggle to decide which courses to take next. At the same time, faculty advisors experience significant workload pressure during registration periods, managing large numbers of students who require personalized guidance. This situation frequently leads to delayed registrations, course selection errors, and overall academic inefficiency. The growing demand for automation and data-driven decision-making in education has led to the integration of artificial intelligence (AI) in student support systems. AI has proven effective in analyzing large academic datasets to predict student performance, identify learning patterns, and recommend suitable academic actions. Systems like **IBM Watson Education**, **Coursera's AI recommendation engine**, and **Edx's personalized learning paths** have shown how AI can be leveraged to enhance academic experiences. Similarly, research studies such as "Predictive Modeling for Student Performance in Higher Education" (Zafra & Ventura, 2019) and "An Intelligent Recommender System for Academic Advising" (Al-Barrak et al., 2020) have demonstrated how machine-learning models can improve student outcomes by suggesting appropriate course sequences. Building on these concepts, **Advisify – AI Course Advisor** aims to apply machine-learning techniques within a university context to recommend courses tailored to each student's academic record and program requirements. Unlike traditional advising systems, Advisify not only predicts suitable courses based on past data but also validates them using course outlines and prerequisite rules. This dual-layer approach ensures that students receive both intelligent and academically valid recommendations, reducing the advising burden on faculty while promoting better academic planning and higher CGPA outcomes.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing

Department of Computer Science

### COMPETITORS/COMPETITIVE ANALYSIS

Several platforms and systems currently exist that offer academic assistance or course recommendation features, but most of them either focus on general learning recommendations or are not tailored to specific university registration processes. The main competitors and related solutions to Advisify are as follows:

1. **IBM Watson Education:**

IBM Watson provides AI-powered educational insights that help institutions understand student learning patterns and predict performance. However, it is a large-scale enterprise solution that focuses on overall learning analytics rather than individualized course registration guidance for university students.

2. **Coursera's Recommendation Engine:**

Coursera uses AI to suggest online courses based on learner interests, completed courses, and skill gaps. While effective for online learning, it is designed for open learning platforms — not for structured university programs or academic advising workflows.

3. **EdX Personalized Learning Paths:**

EdX offers adaptive learning paths that guide users toward courses aligned with their skill goals. However, it focuses on open education rather than institution-specific advising, lacking integration with degree prerequisites or student academic records.

4. **Degree Compass (Austin Peay State University, USA):**

Degree Compass is one of the few systems that predict which courses a student is most likely to succeed in, based on prior academic performance. Though conceptually similar to Advisify, it is a closed university-specific solution and not available for wider implementation.

5. **CAMS (Campus Management Systems):**

Many universities use CAMS or SIS (Student Information Systems) for registration management, but these systems rely on manual advising and lack AI-driven recommendation features.

- **In comparison, Advisify** combines the strengths of these platforms while addressing their limitations. It focuses specifically on **university students**, offering **AI-based personalized course recommendations** tied directly to each institution's **course outlines, prerequisites, and approval process**. This targeted design gives Advisify a unique edge as a lightweight, scalable, and intelligent advising solution tailored for academic institutions.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

### OBJECTIVES

- To design and develop an **AI-powered web system** that assists university students in selecting the most suitable courses for upcoming semesters based on their academic performance and program structure.
- To create a machine-learning model trained on historical student data capable of generating accurate and personalized course recommendations.
- To implement a faculty approval **workflow** where course advisors can review, approve, or modify student course selections efficiently.
- To ensure **validation against course outlines and prerequisites**, so that all recommendations comply with academic rules and semester requirements.
- To develop an **admin panel** for managing users, course data, and AI training datasets while maintaining system transparency and data security.
- To reduce **manual advising workload** and improve the **accuracy and speed** of the student registration process through intelligent automation.
- To build a **scalable and responsive web platform** using **Next.js, FastAPI, and PostgreSQL** that can be deployed on cloud or institutional servers for real-time use.

### MOTIVATION

Every semester, students face significant stress and confusion during the course registration process. Those who have failed subjects, withdrawn from courses, or taken semester breaks often struggle to figure out which courses to take next.

Faculty advisors, on the other hand, experience overwhelming workloads as hundreds of students approach them for manual guidance within a short registration window. This results in delays, long queues, and frequent mistakes in course selection — creating frustration for both students and faculty. Advisify is motivated by the need to eliminate this recurring hassle by automating and simplifying the advising process. By using artificial intelligence to analyze student performance and course prerequisites, the system can suggest suitable course combinations quickly and accurately. This not only reduces the “semester registration chaos” but also helps students make informed academic decisions, improves faculty efficiency, and moves the institution toward smarter, technology-driven education management.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

### FEATURES OF PROJECT

#### 1. AI-Based Course Recommendation

- The core feature of Advisify is its intelligent recommendation system.
- Uses a trained machine-learning model on historical student data (grades, completed courses, CGPA trends).
- Suggests optimal courses for the next semester while validating prerequisite and program rules.
- Aims to improve student CGPA and ensure smooth academic progression.

#### 2. Student Dashboard

- Students can log in using their university credentials.
- Allows uploading of CSV files containing their academic data (course history, grades, etc.).
- Displays AI-generated course suggestions with reasoning and predicted GPA impact.
- Students can accept, modify, or request changes before submitting to their assigned advisor.

#### 3. Faculty (Advisor) Dashboard

- Faculty members can view requests only from their assigned batch of students (controlled by admin mapping).
- Provides course suggestions made by the AI for each student.
- Advisors can approve, reject, or modify the course selections with comments or feedback.
- Ensures transparency and reduces manual advising workload.

#### 4. Admin Panel

- Admins manage user roles, batches, and faculty-student assignments.
- Can upload or update course catalogs, prerequisites, and semester offerings.
- Maintains system logs, approvals, and AI recommendation data.
- Has the authority to retrain or update the AI model with new datasets.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

### 5. Course Validation Engine

- Automatically checks AI suggestions against course outlines and prerequisites.
  - Prevents invalid course combinations (e.g., taking Advanced Programming before Data Structures).
  - Ensures that all recommendations follow institutional academic policies.
- 

### 6. Add/Drop Session Handling

- After registration, students can request to add or drop a course.
  - AI re-evaluates and suggests alternative courses based on academic balance.
  - The request is again sent to the respective advisor for quick approval.
- 

### 7. Notifications & Status Tracking

- Students receive real-time updates on advisor decisions (approved/rejected/pending).
  - Advisors and admins can track student progress and view history of approvals.
  - Optional email or in-system notification system to improve communication.
- 

### 8. Data Security and Access Control

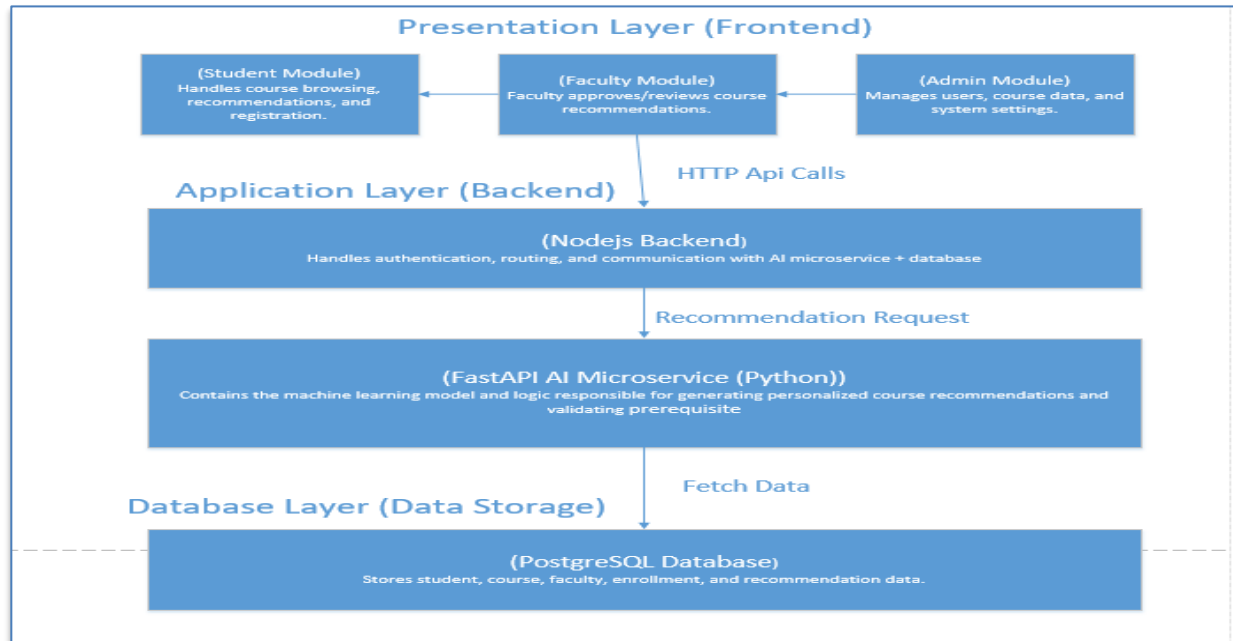
- Implements role-based access (Student, Faculty, Admin).
- Ensures that each user can access only relevant information.
- Data encrypted and securely stored in PostgreSQL with controlled access.
- Uses JWT for authorization.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### ARCHITECTURAL DESIGN



The Advisify system follows a **three-tier architecture** consisting of the **Presentation Layer (Frontend)**, **Application Layer (Backend)**, and **Database Layer (Data Storage)**. This design ensures modularity, scalability, and efficient communication between components.

#### 1. Presentation Layer (Frontend)

- **Technology:** Next.js (React-based framework)
- **Description:** This layer provides the user interface for three modules — **Student**, **Faculty**, and **Admin**.

The **Student Module** allows course browsing, recommendations, and registration.

The **Faculty Module** enables faculty to review and approve course recommendations.

The **Admin Module** manages users, courses, and system settings.

**Communication:** Sends HTTP API calls to the backend for data exchange.



# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### 2. Application Layer (Backend)

- **Technologies:** Node.js (Next.js backend), FastAPI (Python)
- **Description:** The backend acts as the intermediary between the frontend and database.

The **Node.js backend** handles authentication, routing, and communication with the AI microservice and database.

The **FastAPI microservice** contains the trained machine learning model responsible for generating personalized course recommendations and validating prerequisites.

**Interaction:** The backend sends **recommendation requests** to the AI microservice and retrieves processed results for the frontend.

---

### 3. Database Layer (Data Storage)

- **Technology:** PostgreSQL
- **Description:**

Stores and manages structured data including student records, course information, faculty details, enrolment, and recommendation outcomes.

The database ensures data consistency and supports query operations for the backend.

---

#### *Data Flow Summary*

- Users interact with the frontend (Next.js).
- The frontend sends API requests to the Node.js backend.
- The backend communicates with the FastAPI microservice for ML-based recommendations.
- FastAPI fetches necessary data from PostgreSQL, processes it, and returns predictions. The backend sends the final recommendation response to the frontend for display.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### IMPLEMENTATION TOOLS AND TECHNIQUES

The implementation of Advisify – AI Course Advisor will follow a modular and practical development approach to ensure that the system remains scalable, efficient, and easy to maintain. The project will be implemented using modern web technologies and machine learning tools, with development carried out in stages such as design, development, integration, testing, and deployment.

To manage the workflow effectively, the **Agile development methodology** will be used. The project will be divided into smaller milestones (sprints), where each sprint will focus on a specific feature like student registration, AI integration, or faculty approval. This method allows continuous feedback, flexibility, and steady progress throughout the development process.

#### 1. Frontend Development

The frontend will be developed using **Next.js**, which combines React's interactivity with backend API routes in a single framework. It will provide an intuitive, responsive, and user-friendly interface for students, faculty, and administrators. Supporting tools such as **HTML5**, **CSS3**, **JavaScript (ES6)**, and **Tailwind CSS** will be used for styling and layout. **Axios** will handle API communication, while **JWT (JSON Web Token)** or **NextAuth** will be used for secure authentication.

#### 2. Backend Development

The backend will be handled through **Next.js (Node.js)** API routes. This will manage authentication, route handling, and data exchange between the frontend, AI microservice, and the database. The backend will follow an **API-driven architecture** to maintain clear separation between system layers and improve scalability.

#### 3. AI Microservice

The AI module will be developed separately using **FastAPI (Python)**. This service will host the trained machine learning model that analyzes student academic data and generates personalized course recommendations. Libraries such as **Scikit-learn**, **Pandas**, and **NumPy** will be used for data processing and model training, while **Pickle** or **Joblib** will be used to save and load the trained model.

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

### 4. Database Layer

**PostgreSQL** will be used as the database for storing all structured data, including student records, course details, faculty assignments, and approval logs. It ensures data integrity and fast querying. The system will use an ORM (such as **Prisma** or **Sequelize**) to interact with the database securely and efficiently.

### 5. Version Control and Deployment

Version control will be managed through **Git and GitHub** to ensure smooth collaboration and code management among team members. The system will be deployed on reliable cloud hosting platforms such as **Render**, **Railway**, or **Vercel**. Optionally, **Docker** may be used to containerize the system for consistent deployment across environments.

### 6. Security and Techniques

Security will be maintained through **JWT-based authorization**, **HTTPS communication**, and **role-based access control** for Students, Faculty, and Admin users. Data validation and encryption will ensure that all user and academic information remains safe and private.

- In summary, the project will combine **Next.js**, **FastAPI**, and **PostgreSQL** under an agile and modular framework to build a smart, secure, and efficient AI-powered course advising system ready for real-world academic use.

# Muhammad Ali Jinnah University

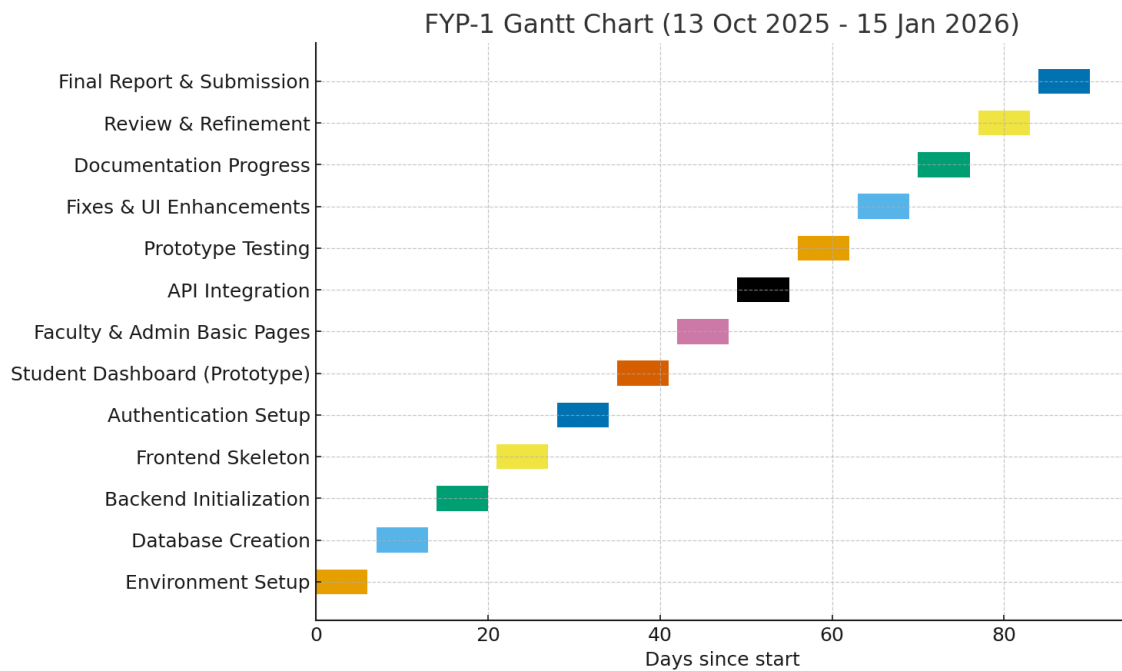
## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### FYP-1 & FYP-2 GANTT CHARTS

This document contains the weekly-wise Gantt charts for FYP-1 and FYP-2 of the Advisify project.

#### 1. FYP-1 (13 Oct 2025 - 15 Jan 2026)



Week	Main Task	Description / Deliverable
Week 1	Environment Setup	Install and configure Next.js, FastAPI, PostgreSQL, and connect GitHub repo
Week 2	Database Creation	Design and create database tables (Student, Faculty, Course)
Week 3	Backend Initialization	Create base API routes for user login and course retrieval

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

Week 4	Frontend Skeleton	Build basic layout and navigation (Home, Login, Dashboard placeholders)
Week 5	Authentication Setup	Implement login/signup functionality with backend
Week 6	Student Dashboard (Prototype)	Display dummy student data and structure main components
Week 7	Faculty & Admin Basic Pages	Create static pages for faculty/admin access
Week 8	API Integration	Connect frontend with backend APIs for login and data flow
Week 9	Prototype Testing	Internal testing of integrated prototype
Week 10	Fixes & UI Enhancements	Resolve issues, adjust layout, and improve visuals
Week 11	Documentation Progress	Write partial documentation (methodology, screenshots)
Week 12	Review & Refinement	Supervisor feedback, code cleanup, minor updates
Week 13–14	Final Report & Submission	Prepare final FYP-1 report and submit presentation

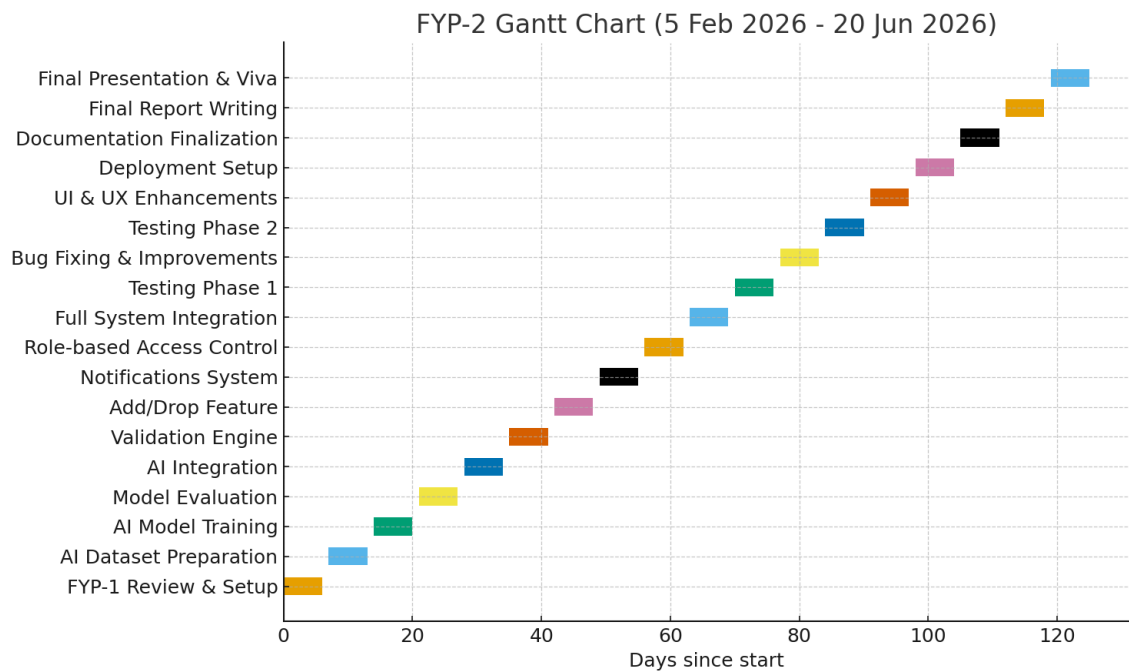
Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing

Department of Computer Science

2. FYP-2 (5 Feb 2026 - 20 Jun 2026)



Week	Main Task	Description / Deliverable
Week 1	FYP-1 Review & Setup	Review previous work and set up for continuation
Week 2	AI Dataset Preparation	Collect and clean data for recommendation model
Week 3	AI Model Training	Train the machine learning model for course prediction
Week 4	Model Evaluation	Test and tune model performance
Week 5	AI Integration	Integrate AI with FastAPI microservice
Week 6	Validation Engine	Develop course prerequisite validation logic

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

*Faculty of Computing  
Department of Computer Science*

Week 7	Add/Drop Feature	Implement add/drop functionality with backend
Week 8	Notifications System	Add email and in-app notifications
Week 9	Role-based Access Control	Add student/faculty/admin authentication roles
Week 10	Full System Integration	Connect all modules and ensure data flow
Week 11	Testing Phase 1	Conduct unit and integration testing
Week 12	Bug Fixing & Improvements	Fix reported issues and optimize performance
Week 13	Testing Phase 2	Perform user acceptance testing
Week 14	UI & UX Enhancements	Refine interface design for usability
Week 15	Deployment Setup	Deploy backend (Render), frontend (Vercel), and database (Supabase)
Week 16	Documentation Finalization	Complete final documentation and screenshots
Week 17	Final Report Writing	Compile results, achievements, and conclusions
Week 18–19	Final Presentation & Viva	Prepare and deliver final presentation

# Muhammad Ali Jinnah University

## Final Year Project (FYP)

Faculty of Computing  
Department of Computer Science

### REFERENCES

- Zafra, A., & Ventura, S. (2019). Predictive Modeling for Student Performance in Higher Education. *Educational Data Mining Journal*, 11(1), 1–15.
- Al-Barrak, M. A., & Al-Razgan, M. (2020). An Intelligent Recommender System for Academic Advising Using Machine Learning. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(5), 54–62.
- IBM Corporation. (2021). IBM Watson Education Overview. Retrieved from <https://www.ibm.com/watson-education>
- Coursera. (2022). AI-Powered Course Recommendation System. Retrieved from <https://www.coursera.org>
- edX. (2022). Personalized Learning and Course Pathways. Retrieved from <https://www.edx.org>
- Austin Peay State University. (2018). Degree Compass: Predictive Course Recommendation System. Retrieved from <https://www.apsu.edu/degreecompass>
- Scikit-learn Developers. (2024). Machine Learning in Python. Retrieved from <https://scikit-learn.org>
- FastAPI Documentation. (2024). Modern Web Framework for Building APIs with Python. Retrieved from <https://fastapi.tiangolo.com>
- Next.js Documentation. (2024). React Framework for Building Web Applications. Retrieved from <https://nextjs.org/docs>
- PostgreSQL Global Development Group. (2024). PostgreSQL: Open Source Relational Database. Retrieved from <https://www.postgresql.org>

Supervisor's Signature: -

FYP-Coordinator's Signature: