

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

PROJECT TITLE:	Advisify - AI-Based Automated Course Registration System
KEY WORDS:	AI Registration System, Automation, Advisory Detection, Course Allocation, Timetable Generation, Multi-Agent System, FastAPI, Next.js, Educational Technology
DOMAIN OF THE PROJECT:	Artificial Intelligence, Web Application Development, Academic Automation, Educational Technology (EdTech)
SUPERVISOR'S NAME:	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

Course registration remains one of the most time-consuming administrative tasks in universities. Students who have failed, withdrawn, or taken semester breaks must go through a manual advisory process, while regular students are still registered individually by departmental staff. This outdated approach creates long queues, scheduling errors, and unnecessary workload during registration periods. There is a growing need for an intelligent, automated system that can handle both advisory and regular cases simultaneously detecting academic exceptions, assigning appropriate courses, and completing registration accurately with minimal human involvement.

EXECUTIVE SUMMARY:

Advisify – AI-Based Automated Course Registration System is an intelligent web-based platform designed to automate the entire university course registration process for both advisory and regular students. During registration periods, departments face heavy workloads as staff manually process enrollments, and students with advisory cases such as F-grades, withdrawals, or semester freezes require additional guidance. This often results in delays, human errors, and inconsistent course allocations.

Advisify eliminates these challenges by introducing an AI-driven automation workflow powered by multiple intelligent agents. The system analyzes student academic data to automatically detect advisory cases, assign suitable courses, and generate a clash-free timetable for the department. Regular students are registered in their standard semester courses, while advisory students receive course allocations based on their remaining academic requirements.

The platform operates through a single **Admin Module**, allowing authorized users to upload student and course CSV files, trigger automated registration, and download final schedules all with one click. Developed using **Next.js** for the frontend and **Node.js + FastAPI** for backend automation, the system integrates AI logic, rule-based algorithms, and scheduling techniques to ensure accuracy and efficiency. Advisify minimizes manual effort, reduces administrative pressure, and enables universities to achieve fully digital, error-free course registration representing a major step toward smarter academic management.

Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing

Department of Computer Science

INTRODUCTION

In most universities, course registration and advisory processes still depend heavily on manual work. Students who have fallen behind due to F-grades, course withdrawals, or semester freezes often struggle to identify which courses they should take next, while regular students still rely on staff to manually register their courses each semester. This traditional approach causes long queues, delays, and frequent errors, especially during registration weeks when departments handle large volumes of students simultaneously.

The increasing need for automation in academic management has led to the integration of artificial intelligence (AI) and multi-agent systems in higher education. These technologies can analyze student data, detect advisory cases, and automate complex administrative operations efficiently. Systems like IBM Watson Education and AI-driven scheduling tools have demonstrated how automation can enhance decision-making and improve institutional workflows.

Building on these advancements, **Advisify – AI-Based Automated Course Registration System** introduces a new approach to university registration. It uses autonomous agents to identify advisory students, assign courses intelligently, and generate a clash-free timetable without manual intervention. Unlike traditional advising or registration systems, **Advisify** focuses on full automation reducing administrative workload, preventing registration conflicts, and promoting a seamless, one-click registration experience for universities.

Muhammad Ali Jinnah University

Final Year Project (FYP)

*Faculty of Computing
Department of Computer Science*

COMPETITORS/COMPETITIVE ANALYSIS

- Several educational platforms and registration systems currently exist that support student enrollment or course planning, but most rely on partial automation or are not designed to handle both advisory and regular student registrations. The main competitors and related systems to **Advisify** are as follows:
1. **IBM Watson Education:**
IBM Watson offers AI-powered analytics for predicting student performance and optimizing learning outcomes. However, it focuses on academic insights rather than automating operational tasks such as registration or course allocation.
 2. **Ellucian Banner & PeopleSoft Campus Solutions:**
These enterprise-level university management systems handle student data, enrollment, and course scheduling but depend heavily on manual configuration and lack adaptive AI automation for advisory cases.
 3. **Degree Compass (Austin Peay State University, USA):**
Degree Compass predicts which courses a student is most likely to succeed in but is limited to recommendation, not automated registration or scheduling.
 4. **CAMS / SIS (Student Information Systems):**
Commonly used in universities for course registration and data management, these systems require staff involvement for each registration and do not provide intelligent advisory detection or conflict-free scheduling.
 5. **Timetable Generator Tools:**
Standalone tools that create class timetables, but they are not integrated with student academic data or automated registration systems.

In comparison, **Advisify** bridges these gaps by combining automation, advisory detection, and AI-driven scheduling within a single system. It allows departments to register all students advisory and regular automatically through an agent-based workflow, making it a **lightweight, scalable, and intelligent automation solution** 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 automates the entire course registration process for both advisory and regular university students.
- To build an intelligent multi-agent framework capable of detecting advisory students and assigning appropriate courses based on academic history and program requirements.
- To integrate a scheduling engine that generates clash-free timetables automatically for all registered students.
- To create a centralized admin interface for uploading student and course data, triggering automated registration, and managing institutional configurations.
- To ensure rule-based validation of course prerequisites and credit limits for accurate and compliant registration outcomes.
- To minimize manual workload, human errors, and processing time during registration through complete automation.
- To develop a scalable and responsive web platform using **Next.js (Node.js)**, **FastAPI**, and **PostgreSQL**, deployable on cloud or institutional servers for real-time academic management.

MOTIVATION

Every semester, university departments face overwhelming pressure during the course registration period. Students, especially those with advisory cases such as F-grades, course withdrawals, or semester freezes, experience confusion and delays in getting enrolled. At the same time, administrative staff and faculty must manually process hundreds of registrations within a limited time, leading to queues, data errors, and unnecessary workload.

Advisify is motivated by the need to eliminate these recurring challenges by introducing automation into the registration process. By leveraging artificial intelligence and rule-based algorithms, the system can automatically detect advisory students, assign courses intelligently, and generate clash-free schedules for all students. This not only simplifies registration but also enhances accuracy, saves time, and helps universities move toward efficient, technology-driven academic management.

Muhammad Ali Jinnah University

Final Year Project (FYP)

*Faculty of Computing
Department of Computer Science*

FEATURES OF PROJECT

- **AI-Based Automated Registration**

- The core feature of AutoRegAI is its intelligent automation engine.
- Uses trained AI models and rule-based logic to detect advisory students and register them automatically.
- Allocates courses based on academic history, prerequisites, and program structure.
- Ensures accurate, clash-free scheduling and balanced course load for all students.

- **Admin Dashboard**

- A single control center for all registration activities.
- Allows the admin to upload student and course CSV files for automated processing.
- Generates complete registration schedules for advisory and regular students.
- Displays final timetables and reports for review or download.

- **Multi-Agent Workflow**

- Employs multiple intelligent agents for specialized tasks such as advisory detection, course allocation, and timetable generation.
- Each agent communicates with others to ensure smooth and coordinated automation.
- Improves system efficiency and scalability for handling large datasets.

- **Automated Course Validation Engine**

- Validates each course assignment against prerequisite rules and degree requirements.
- Prevents invalid combinations (e.g., enrolling in an advanced course before its prerequisite).
- Guarantees that all course allocations comply with institutional academic policies.

- **AI-Powered Timetable Generation**

- Automatically generates a clash-free timetable for all (CS) batches and students.
- Balances session times to avoid overlap between advisory and regular registrations.
- Supports adjustment and regeneration in case of changes or late additions.

- **Notification and Reporting System**

- Provides instant notifications for registration completion or detected conflicts.
- Generates summary reports for departments, highlighting total enrollments and advisory cases.
- Offers exportable CSV or PDF reports for record-keeping.

- **Data Security and Access Control**

- Implements role-based access with restricted administrative privileges.
- All data encrypted and securely stored in PostgreSQL.
- Uses JWT for secure authentication and HTTPS for data transfer protection.

Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing
Department of Computer Science

ARCHITECTURAL DESIGN

Presentation Layer (Frontend)

(Admin Module)

Manages CSV uploads, triggers automation, and views results.

Application Layer (Backend)

(Nodejs Backend)

Handles auth & routing, sends data to FastAPI, updates database.

HTTP Api Calls

(FastAPI AI Microservice (Python))

Performs advisory detection, course allocation, timetable generation, and rule validation.

Fetch,Store Data

Database Layer (Data Storage)

(PostgreSQL Database)

Stores student, course, registration, timetable, and configuration data.

Muhammad Ali Jinnah University

Final Year Project (FYP)

*Faculty of Computing
Department of Computer Science*

The Advisify system follows a three-tier architecture consisting of the **Presentation Layer (Frontend)**, **Application Layer (Backend)**, and **Database Layer (Data Storage)**.

1. Presentation Layer (Frontend)

Technology: Next.js (React-based framework)

Description: This layer provides a single **Admin Module** interface where authorized staff upload student and course CSV files, configure rules, trigger automated registration, and view generated timetables.

Communication: Sends HTTP API requests to the backend for data processing and result retrieval.

2. Application Layer (Backend)

- **Technologies:** Node.js (API & routing), FastAPI (Python – AI Microservice)

- **Description:** The **Node.js backend** handles authentication, routing, and communication between the frontend, database, and AI service.

The **FastAPI microservice** executes core automation tasks including advisory detection, course allocation, timetable generation, and rule validation.

- **Interaction:** Node.js sends CSV and configuration data to FastAPI, which processes it using AI agents and returns results for database update and admin display.

3. Database Layer (Data Storage)

- **Technology:** PostgreSQL

- **Description:**

Stores student, course, registration, timetable, and configuration data.

Ensures data integrity and supports efficient read/write operations for backend services.

Data Flow Summary

1. Admin uploads CSVs and triggers automation via the Next.js interface.
2. Node.js backend receives data and forwards it to the FastAPI microservice.
3. FastAPI performs advisory detection, course allocation, and schedule generation using stored data.

Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing
Department of Computer Science

IMPLEMENTATION TOOLS AND TECHNIQUES

The implementation of **Advisify – Automated AI Course Registration System** will follow a modular and practical development approach to ensure scalability, performance, and easy maintenance. Development will be carried out in stages - design, development, integration, testing, and deployment under the **Agile methodology**, using short sprints for continuous improvement and feedback.

1. Frontend Development

The frontend will be developed using **Next.js**, providing a unified and responsive interface for the **Admin Module**. The admin can upload student and course CSVs, trigger automation, and view generated timetables.

Supporting tools include **HTML5**, **CSS3**, **JavaScript (ES6)**, and **Tailwind CSS** for styling. **Axios** will handle API communication, while **JWT** or **NextAuth** will manage secure authentication.

2. Backend Development

The backend will be powered by **Node.js** (Next.js API routes).

It handles authentication, routing, and communication between the frontend, AI microservice, and database. An **API-driven architecture** ensures clean separation between layers and high scalability.

3. AI Microservice

The **AI automation module** will be implemented using **FastAPI (Python)**.

It handles core logic including **advisory detection**, **course allocation**, **timetable generation**, and **rule validation**.

Libraries like **Pandas**, **NumPy**, and **Scikit-learn** will be used for data processing and model logic, with **Joblib/Pickle** for model persistence.

4. Database Layer

PostgreSQL will serve as the primary database to store student, course, registration, timetable, and configuration data.

Data interaction will be handled through an ORM such as **Prisma** for security and maintainability.

Muhammad Ali Jinnah University

Final Year Project (FYP)

*Faculty of Computing
Department of Computer Science*

5. Version Control and Deployment

Git and **GitHub** will manage version control and collaboration.

Deployment will be handled via **Vercel**, **Railway**, or **Render**, with optional **Docker** containers to ensure consistent environments.

6. Security and Techniques

Security will be ensured through **JWT-based authentication**, **HTTPS**, and **role-based access control** (for Admin users).

Data validation, encryption, and controlled file uploads will maintain integrity and privacy.

- **Summary:** Advisify integrates **Next.js**, **Node.js**, **FastAPI**, and **PostgreSQL** in a modular, AI-driven system - providing an automated, secure, and scalable course registration solution for academic institutions.

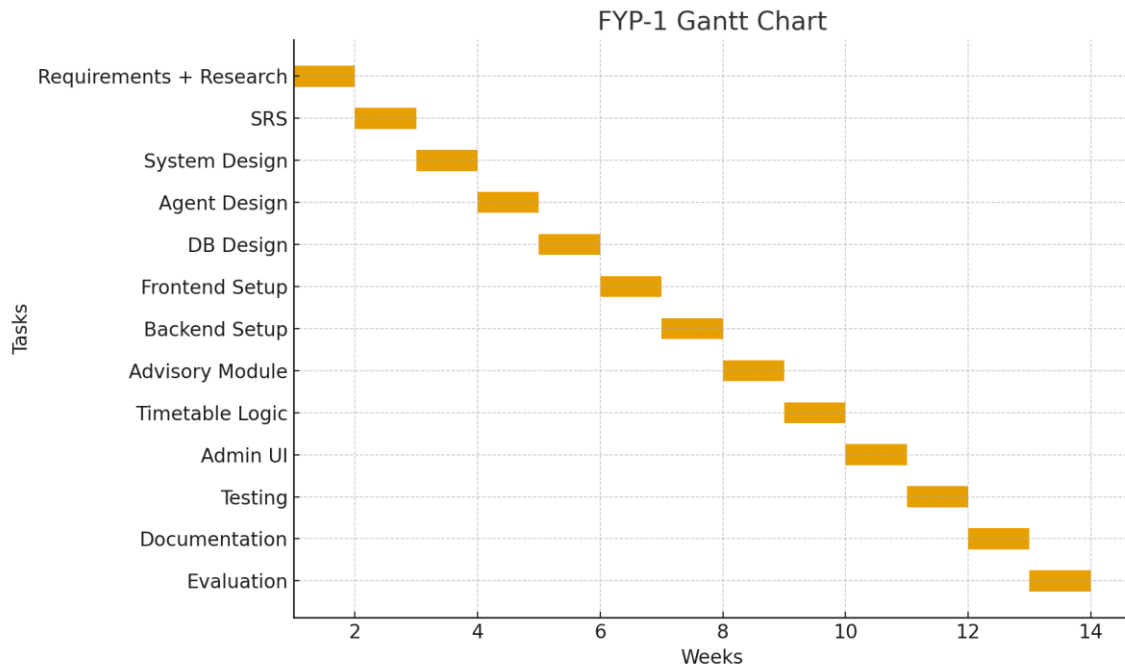
Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing
Department of Computer Science

FYP-1 & FYP-2 GANTT CHARTS

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



Week	Task
Week 1	Understand problem and requirements
Week 2	Research and study other systems
Week 3	Make final list of requirements (SRS)
Week 4	Make system design and diagrams
Week 5	Design how agents will work in system
Week 6	Design database structure (tables + ERD)

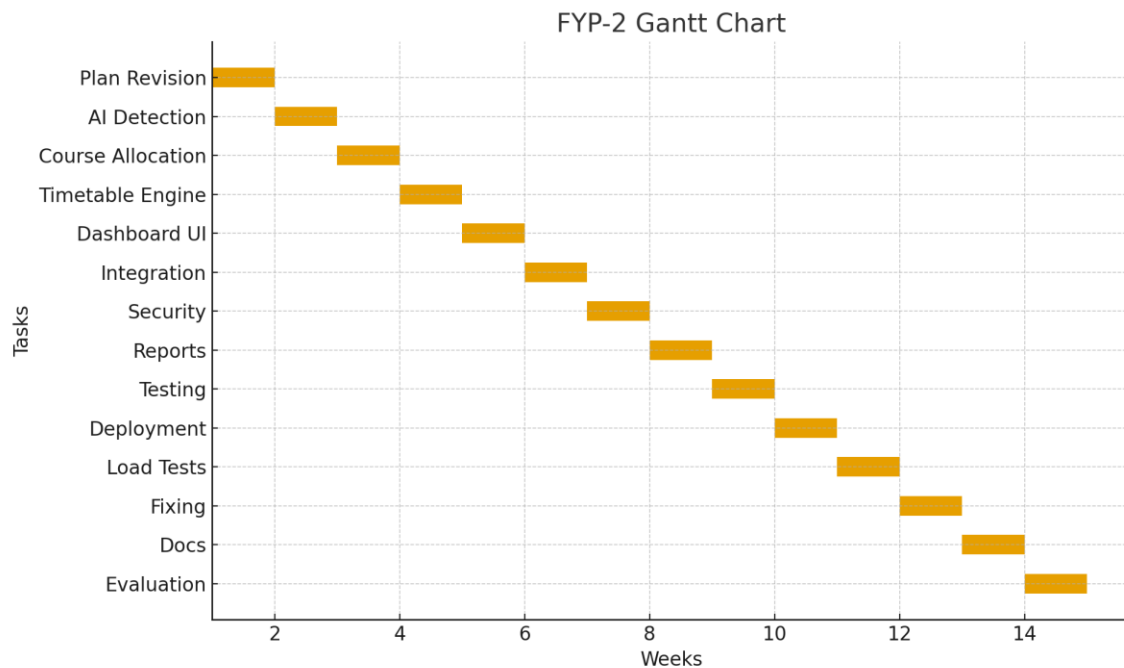
Muhammad Ali Jinnah University

Final Year Project (FYP)

Faculty of Computing
Department of Computer Science

Week 7	Start frontend setup (Admin Panel)
Week 8	Start backend setup (Node + FastAPI)
Week 9	Make advisory detection rules first version
Week 10	Research and test timetable generation logic
Week 11	Admin Dashboard UI part 1
Week 12	Connect system and do basic testing
Week 13	Documentation writing + prepare for mid
Week 14	Final updates and FYP-1 evaluation

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



Muhammad Ali Jinnah University

Final Year Project (FYP)

*Faculty of Computing
Department of Computer Science*

Week	Task
Week 1	Improve plan after supervisor feedback
Week 2	Make advisory detection smarter with AI
Week 3	Make automatic course enrollment working
Week 4	Make clash-free timetable engine working
Week 5	Admin dashboard with timetable View
Week 6	Connect all agents together, full working flow
Week 7	Improve security: database safety and login
Week 8	Add alerts and downloadable reports
Week 9	Full testing of complete system
Week 10	Deploy system on cloud
Week 11	Load test with many students
Week 12	Fix bugs, improve UI
Week 13	Write final documentation
Week 14	Make presentation and poster
Week 15	Internal evaluation
Week 16	Fix things recommended by internal panel
Week 17	Final polishing and improvements
Week 18	Prep & final show for external evaluation

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: