System Design Document

# Project Title: Course Planner with AI Assistant

## 1. Project Overview

A full-stack web application that helps students plan academic courses, detect scheduling conflicts, and get AI-powered course recommendations.

Goals:

* - Simplify academic course selection
* - Provide smart AI-based planning suggestions
* - Detect timetable overlaps
* - Export course schedules to calendar tools

## 2. Target Users

- University students (undergraduate and graduate)

- Academic advisors

- Universities (as a SaaS tool)

## 3. Functional Requirements

* - User registration & authentication
* - Search & browse course catalog
* - Add/edit semester plans
* - Detect and notify of schedule conflicts
* - AI assistant for course planning
* - Export to calendar (.ics/Google Calendar)
* - Save/load personal study plans

## 4. Non-Functional Requirements

* - Responsive and mobile-friendly
* - RESTful API design
* - Secure authentication
* - Modular and maintainable codebase
* - Cloud-hosted and scalable (Azure)

## 5. Tech Stack

Frontend: React + TypeScript / Blazor  
Backend API: ASP.NET Core Web API  
AI Assistant: Python (FastAPI) + OpenAI API  
Auth: Firebase Auth / Azure AD B2C  
Database: Azure SQL / PostgreSQL  
Deployment: Azure App Services, Azure Pipelines

## 6. System Architecture Overview

A diagram will be added here.

## 7. Database Design

Tables: Users, Courses, Plans, PlanEntries

## 8. API Design

Examples:

* - GET /api/courses - Get all courses
* - POST /api/plans - Create a new semester plan
* - POST /api/ai/suggest - AI-powered planning suggestion
* - GET /api/user/plans - Get user's saved plans

## 9. AI Prompt Strategy

Prompt template:

"You are an academic advisor. Based on the following course catalog and user's academic goals, suggest a semester-wise plan that includes prerequisites and avoids overlap. Output in JSON."  
  
Course Catalog: [structured input]  
User Goal: [dynamic input]

## 10. Milestones & Timeline

* - Week 1 - Finalize documentation & architecture
* - Week 2 - Scaffold backend + DB schema
* - Week 3 - Build frontend UI + connect API
* - Week 4 - AI assistant + export tools
* - Week 5 - Testing, bug fixes, deployment

## 11. Future Improvements

* - Admin dashboard for uploading course catalogs
* - AI recommendation fine-tuning
* - Support for elective balancing
* - Dark mode / accessibility features