

System Design Document

1. Introduction

This project implements an **AI Syllabus Scheduler Agent** that automates the extraction of coursework deadlines from a syllabus and generates a calendar file (.ics) for easy import into Google Calendar, Outlook, or Apple Calendar.

The goal is to reduce the manual effort students spend in identifying deadlines and entering them into their calendars.

2. System Architecture

High-level flow:

1. User enters *Course Name* and *Syllabus Text* into the UI.
2. AI Tool (Gemini LLM) parses the syllabus and extracts structured deadline data (JSON).
3. Extracted data is shown in a table for monitoring.
4. Custom Tool generates a .ics calendar file from the structured deadlines.
5. User downloads and imports the file into their calendar app.

3. Components

- **AI Tool (Gemini LLM)**
 - Function: Extract deadlines in structured JSON format.
 - Input: Raw syllabus text.
 - Output: JSON array of {assignment_name, due_date, assignment_type}.
- **Custom Tool (ICS Generator)**
 - Function: Convert structured data into .ics events.
 - Libraries: ics, datetime.
- **User Interface (Gradio)**
 - Input form: Course name + syllabus text.
 - Output: Status message, extracted deadlines (table), download link for .ics.
- **Data Representation**
 - Deadlines: JSON array.
 - Monitoring: pandas.DataFrame.
 - Export: .ics format for universal calendar compatibility.

4. Technology Choices

- **Python:** Rapid prototyping, rich ecosystem for AI + data handling.
- **Gradio:** Quick web UI without heavy frontend coding.
- **Gemini API:** Reliable LLM for structured data extraction.
- **Pandas:** Simple tabular visualization of extracted data.
- **ICS library:** Standardized calendar event generation.

5. Reasoning Behind Choices

- Gradio chosen over Flask/Streamlit for simplicity in UI + direct integration with ML workflows.
- Gemini API chosen for strong natural language understanding and JSON-formatted output.
- .ics chosen since it is cross-platform (works on Google, Outlook, Apple).
- Modular design: AI tool + custom tool separation allows future multi-agent expansion.

6. Limitations & Future Work

- Currently depends on Gemini parsing correctly (JSON parsing errors possible).
- No multi-agent orchestration (Planner + Executor) yet.
- Could extend to:
 - Support for multiple courses at once.
 - Editing deadlines before export.
 - Batch scheduling / reminders.

7. Conclusion

This system demonstrates how an **AI Agent prototype** can automate a common student task — syllabus deadline extraction and calendar integration. It balances simplicity with practical utility and is extensible for future improvements.