AI-Integrated Learning Management System (LMS)

Demo Video

Github

Overview

This project is a **Learning Management System (LMS)** built using **Flask and Supabase**, designed to streamline the workflow of tutors and students. It supports **student registration, class scheduling, attendance tracking, payment management, and personalized performance reporting**.

A key differentiator of this LMS is the **AI-powered student report generator**, where teachers can feed structured inputs which are processed by the **OpenAI API** to automatically generate personalized student performance reports. Additionally, the system integrates with **n8n workflows** to automate Google Calendar scheduling for classes and reminders.

Features

- **User & Student Management**
  • Secure registration and management of students and classes.
  • Attendance tracking with CRUD operations stored in Supabase.
- **Scheduling & Automation**
  • Class scheduling and rescheduling with automated sync to **Google Calendar** using **n8n webhooks**.
  • Self-hosted n8n setup on Hostinger for cost-effective, 24x7 task automation.
- **Payment & Fee Tracking**
  • Records payments made by students or clients.
  • Automatically calculates pending fees even when payments are irregular or partial.
- **AI-Powered Reports**
  • Teachers can feed minimal structured inputs for each student.
  • **OpenAI API** generates customized performance reports with a balance between personalization and minimal manual work.

System Architecture

(User – Teacher/Student) → **Flask Web App** → Supabase (Database) → Attendance/Payments/Student Data

Flask → **OpenAI API** → AI-Powered Report Generation
Flask → **n8n Webhooks** → Google Calendar Sync → Notifications

---

Tech Stack

- **Backend:** Flask
- **Database:** Supabase Postgres
- **AI Integration:** OpenAI API (for personalized student reports)
- **Automation:** n8n workflows (self-hosted on Hostinger)
- **Frontend:** HTML, CSS, JavaScript (Bootstrap templates)
- **Deployment:** Hostinger (Flask + n8n self-hosted)

Project Structure

lms-flask/
|— app.py: Main Flask application
|— forms.py: Flask-WTF forms for input handling
|— models/: Supabase integration logic
|— templates/: HTML templates (frontend UI)
|— static/: CSS/JS assets
|— requirements.txt: Dependencies

Challenges & Learnings

1. **Database Integration**
    - Initially used **Google Sheets API** for quick prototyping, but it was slow and unreliable for handling multiple students and transactions.
    - Migrated to **Supabase Postgres**, which provided a scalable, fast, and developer-friendly backend.
2. **Designing Teacher Input for AI Reports**
    - Challenge: Teachers needed to feed enough detail for personalized reports without making the process too time-consuming.

- Solution: Created a **structured input format** that balances customization with minimal manual effort, enabling OpenAI to generate accurate reports.

3. **Automation (n8n Hosting Decisions)**
    - Challenge: Cloud n8n was reliable but expensive; local hosting lacked 24x7 availability.
    - Solution: Opted for **self-hosted n8n on Hostinger VPS**, striking a balance between cost and continuous uptime.
4. **Handling Irregular Payments**
    - Challenge: Clients often made **random, partial, or delayed payments**, making fee calculation difficult.
    - Solution: Implemented a **dynamic pending fee calculation system** that adjusts balances based on transaction history.

Author
Akash Gupta
M.Tech Artificial Intelligence, IIT Kharagpur