FLORIDA INTERNATIONAL UNIVERSITY

FIU | Engineering & Computing

Canvas AI Assistant - MCP Server

Students: Nitin Chokkalla, Edgar Velarde, Alberto Abrahantes, Sebastian Rodriguez, Jin Carballosa
Mentor: *Masoud Sadjadi*
Instructor/Faculty: *Masoud Sadjadi,* Florida International University

Table of Contents

INTRODUCTION

Problem

Students struggle to stay organized within Canvas.
Multiple pages for assignments, grades, and announcements
Inconsistent layouts across courses

No single dashboard or unified schedule

Missed deadlines and lost time navigating Canvas

No AI or smart assistant to summarize updates

Solution

An AI-powered system that pulls assignments from the Canvas API, uploads them to Google Calendar, and includes a responsive chatbot solves this, keeping students organized and supported.

CURRENT SYSTEM

Strengths of Canvas LMS:

- Centralized hub for course materials and announcements
- Supports grade tracking and assignment submissions in one place

Limitations:

- Requires navigation through multiple pages manually
- No AI search, automation, or proactive reminders
- Leads to missed deadlines & loss of valuable student time

REQUIREMENTS

Functional:

- Users must be able to link their account without much setup
- System must process data and upload it to a calendar app

Non-Functional:

- UI should be responsive and intuitive
- Backend must process data within a timely matter
- Ai chatbot outputs must be reliable and relevant to academic study materials

SYSTEM DESIGN

Client-server model architecture

Client-Side
Web or chat interface planned.Sends HTTP requests to Flask backend through REST APIs.

Server side

Flask application handles routing, data sync, AI queries, and authentication.

Database

PostgreSQL with SQLAlchemy ORM. Stores courses, assignments, announcements, and grades.

Subsystems

Data Ingestion Layer
Fetches Canvas data using `canvas_sync.py` and updates DB automatically.

Database Manager

Handles CRUD operations and data validation via `manager.py`

AI Query Processor

Uses RAG to answer questions with Gemini/Claude based on stored Canvas data.

CHALLENGES

- Integrating the Canvas API with the Flask backend and PostgreSQL database.
- Managing API authentication tokens, rate limits, and failed requests.
- Designing efficient relational models for courses, assignments, and announcements.
- Handling data synchronization without duplication or loss.
- Ensuring accurate and context-aware AI query responses.
- Maintaining security of user data, credentials, and environment variables.
- Implementing reliable retry logic and scheduling for background syncs.
- Debugging API responses and managing inconsistent Canvas data formats.

FUTURE FEATURES

Real-Time LLM Integration — Connect with Gemini or Claude for intelligent, conversational answers directly from Canvas data.

Smart Notifications — AI-driven alerts for upcoming deadlines, new grades, or missed announcements.

Student Dashboard — Centralized view combining all courses, tasks, and progress analytics.

Voice & Chatbot Interface — Allow students to ask questions using voice or text from any device.

Analytics & Insights — Generate performance trends and personalized study recommendations.

Multi-User Support — Extend platform for multiple students or classroom management tools.

SUMMARY

- Canvas AI Assistant simplifies academic life by integrating Canvas LMS data into one intelligent system.
- Combines Flask backend, PostgreSQL database, and Canvas API for seamless data management.
- Uses AI and LLM integration to answer course-related questions naturally and accurately.
- Provides a foundation for automation syncing data, generating reminders, and powering analytics.
- Future updates will enhance scalability, add smart features, and improve the student experience.
- Overall, this project bridges technology and learning to make studying smarter, faster, and easier.

Q&A Session