

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

PROBLEM

Students struggle to track deadlines across multiple courses, and existing tools don't sync learning platforms with personal schedules. 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.

SYSTEM DESIGN

Data Ingestion: Runs scheduled sync jobs to pull courses, assignments, announcements, and grades.

Backend & Database: A Python MCP server connects to a PostgreSQL database via SQLAlchemy ORM and exposes 7 MCP tools plus REST-style endpoints for client apps.

Client & Integration Layer: An MCP Client (Claude Desktop) serves as the main chat interface. Planned integrations include Google Calendar deadline sync, notifications, and an LTI app directly inside Canvas.

VERIFICATION

Verification combines unit, integration, and system-level tests. A MockCanvasClient with realistic data (2 students, 3 courses, 7 assignments) is used to exercise all API paths without requiring a live Canvas instance. Pytest unit tests target core logic and achieve high coverage, including validation, status calculation, and error handling. Integration tests verify MCP tools end-to-end with Claude Desktop, confirming that user queries correctly trigger database reads and formatted responses. Additional checks validate database CRUD operations, connection health, and startup scripts.

CURRENT SYSTEM

Canvas LMS offers structured access to courses, grades, and assignments, but Requires users to navigate multiple pages Lacks AI-based search or automation. Offers no conversational or proactive information retrieval. So students waste valuable time and often miss critical updates or deadlines.

OBJECT DESIGN

- The backend uses SQLAlchemy to model Canvas entities:
- Course: Identified by Canvas ID; stores name, code, term; links to assignments, announcements, and grades.
 - Assignment: Identified by Canvas ID, linked to a course; stores title, description, due date, points, and a status enum (e.g., is_overdue).
 - Announcement: Linked to a course; stores title, content, and posted date for course messages.
 - Grade: Links course and assignment to a student's score, letter grade, feedback, and graded time.

MCP Tools: Seven asynchronous tools manage Canvas operations, including get_student_courses, get_upcoming_assignments, get_missing_assignments, get_course_grades, get_assignment_grades, get_announcements, and get_course_assignments. All tools validate JSON input and return structured, human-readable results with clear error messages.

REQUIREMENTS

- Functional Requirements:**
 - Users must be able to link their account without much setup work.
 - The system must process data and upload it to a calendar app.
- Non-Functional Requirements:**
 - The UI should be responsive and intuitive.
 - Backend should process data within a timely manner.
 - AI chatbot outputs must be reliable and relevant to academic study materials.

IMPLEMENTATION

The system is implemented as a Python MCP server backed by PostgreSQL. On startup, the server initializes database tables, loads configuration from environment variables, and connects to the Canvas API or a MockCanvasClient for testing. Synchronization scripts pull Canvas data. Each MCP tool is implemented as a separate module that: (1) validates inputs, (2) queries the database using ORM models, and (3) formats results into concise text for Claude Desktop.

SUMMARY

The Canvas AI Assistant automates data collection from Canvas LMS and allows students to interact through conversational AI. It centralizes assignments, Announcements and grades into a single backend system powered by intelligent query understanding. By integrating Large Language Models, this assistant improves accessibility, reduces missed deadlines and enhances the overall learning experience.

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Professor Masoud Sadjadi. We thank Professor Masoud Sadjadi for his/her/their assistance and mentorship that we received throughout the senior design project.