



Marwadi
University
Marwadi Chandarana Group





Major Project-II (01 CE0807), Semester 8

Review 1

(08/11/2025)

SmartCampus: AI-Driven College Assistant with LLM Integration

Team ID: 8PRO_113

Team Member 1: Gopal Gohel (922001 031 53) (8TC5)

Team Member 2: Saksham Kumar Rana (922001 031 43) (8TC5)

Team Member 3: Raj Roy (922001 031 04) (8TC5)

Guided By

Internal Guide Name: Prof. Nitin Varshney

Department of Computer Engineering,

Faculty of Engineering & Technology

- Introduction
- Abstract
- Survey of Existing Systems
- Tools & Technology
- Proposed Approach / Solution / Methodology
- Execution Flow with Planning and Scheduling
- Expected Outcome

- Colleges generate large amounts of data related to students, faculty, and administration.
- Accessing this information is often time-consuming and requires manual searching. Manual data access is slow, repetitive, and time-consuming too.
- With advancements in Artificial Intelligence (AI) and Large Language Models (LLMs), it's possible to create intelligent assistants that understand and respond to human-like queries.
- SmartCampus is designed to simplify data access in educational institutions through an AI-driven chat assistant.
- The system allows users to interact with college data — such as attendance, grades, or faculty performance — using natural language queries, providing quick and accurate results.
- SmartCampus aims to bridge this gap by providing an AI-powered assistant that simplifies how information is retrieved and analyzed within a college environment.

- SmartCampus is an AI-based college assistant that uses LLMs to understand user queries and provide accurate responses from college databases.
- It allows students, teachers, and administrators to access data like attendance, grades, and admissions through a chat interface.
- The system uses FastAPI, PostgreSQL, LangChain, and GPT-4 for backend intelligence and React.js for a user-friendly interface.
- It aims to improve data accessibility, efficiency, and decision-making in educational institutions.
- SmartCampus demonstrates how AI can automate and enhance campus operations through conversational intelligence.

Survey of Existing Systems

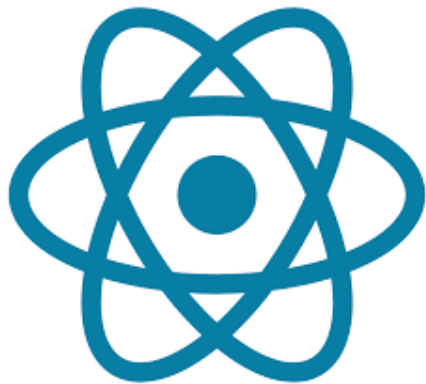


System Type	Examples	Limitation
ERP / SIS Portals	TCS iON, Campus ERP, Fedena, CollPoll	Need manual menu navigation, no conversational access
Static Chatbots / FAQ Bots	Basic university websites	Only pre-defined responses, cannot fetch live DB values
Result / Attendance Dashboards	University portals & PDF downloads	Only show fixed reports; no natural language query support
Admin Desk / Help Desks	Physical counters, faculty requests	Slow & manual human dependency

Most of these systems require the user to **search through UI panels, open multiple pages** or **download PDFs** just to get basic academic information. They **do not** convert a natural English query into SQL, and they **do not** retrieve data dynamically from live academic databases.

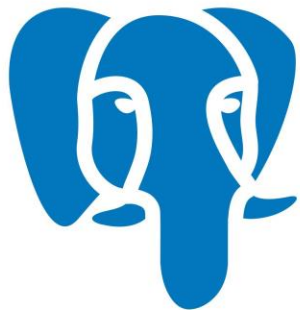
Frontend

- **React.js:** Builds the interactive user interface, including the chat and dashboards for students, faculty, and admins.
- **Tailwind CSS:** Provides a clean, responsive, and modern look for all UI components.
- **Vercel:** Used to deploy the frontend easily with continuous integration and automatic updates.



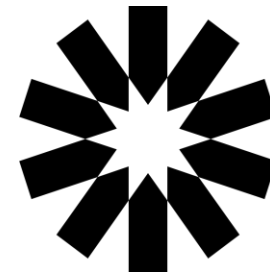
Database & Data Handling

- PostgreSQL: Stores all structured data — including students, faculty, admissions, and performance records.
- Pandas & Scikit-learn: Used for cleaning, analyzing, and generating simple performance insights or analytics.



Backend

- **FastAPI:** Handles all API requests and communication between the LLM, database, and frontend.
- **LangChain:** Integrates the LLaMA model with the backend, allowing the assistant to interpret natural language queries and perform data retrieval.
- **JWT Authentication:** Provides secure login and role-based data access for different user types.



AI & LLM Integration

- **LLaMA**: Core open-source LLM that powers the assistant, fine-tuned or prompted for educational data.
- **ChromaDB**: Vector database that stores and retrieves contextual knowledge, FAQs, and previous interactions for more accurate responses.



Visualization & Deployment

- **Chart.js / Tableau:** Used for visualizing academic performance and analytics.
- **Docker:** Containerizes backend and database for easy deployment and scalability.
- **GitHub:** Enables collaboration, version control, and CI/CD integration.



SmartCampus is designed to simplify access to college data through an AI-powered assistant that understands natural language queries. The system integrates a Large Language Model (LLaMA) with a PostgreSQL database, enabling students, faculty, and administrators to retrieve information quickly and accurately.

1. System Design :

- Identify key user groups — students, faculty, and administrators — and define their main data requirements.
- Design a relational database to store information like attendance, grades, admissions, and faculty performance.
- Plan secure data access and interaction between users, the backend, and the AI model.

2. Data Handling :

- Collect and clean college-related datasets.
- Store all data in PostgreSQL with proper relationships between tables.
- Use Pandas for organizing and analyzing data where required.

3. AI Integration :

- LLaMA model understands user queries in natural language.
- Connected via LangChain to convert questions into database queries.
- Uses ChromaDB to store FAQs and give faster, context-based answers.

4. System Implementation :

- Build a backend with FastAPI for communication between the frontend, database, and AI.
- Develop a simple, chat-based React.js frontend for user interaction.
- Ensure secure logins and role-based permissions using JWT authentication.

5. Output and Results :

- Display responses in a readable format — text, tables, or basic charts.
- Provide quick, accurate answers to queries like:
“Show my attendance,” “List top-performing students,” or “Faculty performance summary.”

SmartCampus will act as a **smart conversational assistant** that saves time, reduces manual data searching, and helps all users easily access the information they need.

Execution Flow with Planning and Scheduling



User Initiates Query

The user types or speaks a plain English question into the SmartCampus interface (web app / mobile / kiosk / WhatsApp / Telegram / campus portal integration).

Query Pre-Processing

The system performs text cleaning, tokenization, and context identification.

This removes noise words and extracts key intent elements such as entity (e.g. subject name) and parameter (e.g. semester, percentage).

LLM Intent Understanding

The Large Language Model processes the cleaned query and determines the exact purpose of the request (example: attendance check, marks analysis, faculty lookup, course schedule retrieval etc.).

Semantic Query Translation

The LLM converts the interpreted user intent into:

Proper SQL query format specific table & column references , required join logic, if data spans multiple tables.



Execution Flow with Planning and Scheduling

Database Execution Layer

The generated SQL query is securely forwarded to the college academic database. The database engine executes the query and retrieves factual structured records.

Post-Processing & Analytics

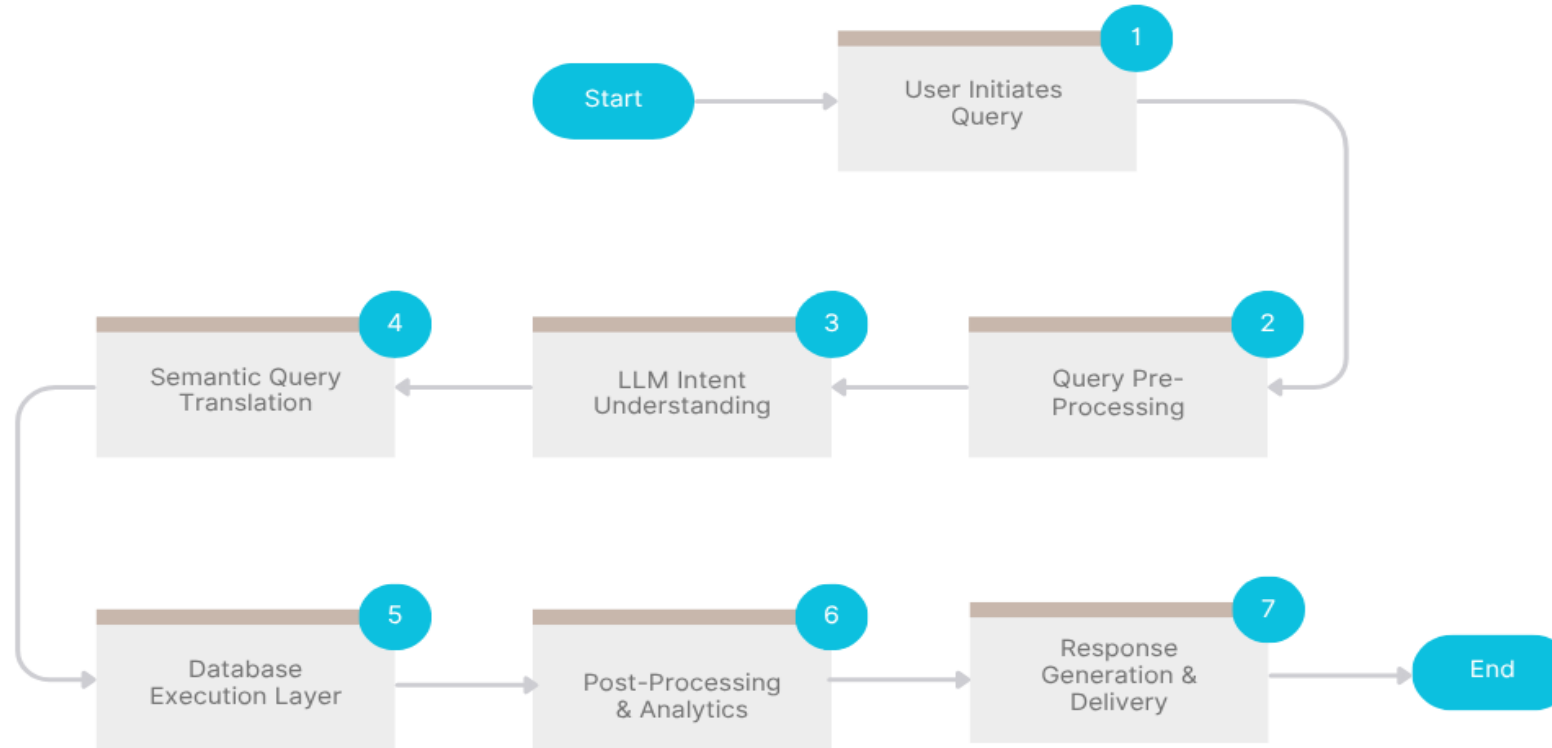
The system reformats fetched data:

- converts to natural sentences
- calculates percentages if required
- prepares charts / tables / comparison metrics if requested

Response Generation & Delivery

The processed output is returned to the user in natural conversational English. If analytics or charts are required, they are attached with the response.

Execution Flow with Planning and Scheduling



- A system where a user can ask academic queries in natural language and receive instant factual answers without browsing ERP portals.
- A method for converting plain English input into structured SQL queries using an LLM based intent understanding engine.
- A database connected AI assistant that fetches real institution tables such as attendance, marks, faculty data and returns cleaned summarized output.
- A system that can generate comparative analytics (e.g. subject wise performance, semester comparisons) using the same conversational interface.
- A campus information retrieval mechanism that eliminates the need for manual portal navigation, PDF lookups or admin contact for basic academic queries.

Expected Outcome



Marwadi
University
Marwadi Chandarana Group





- As per IEEE format



Q&A

THANK YOU

