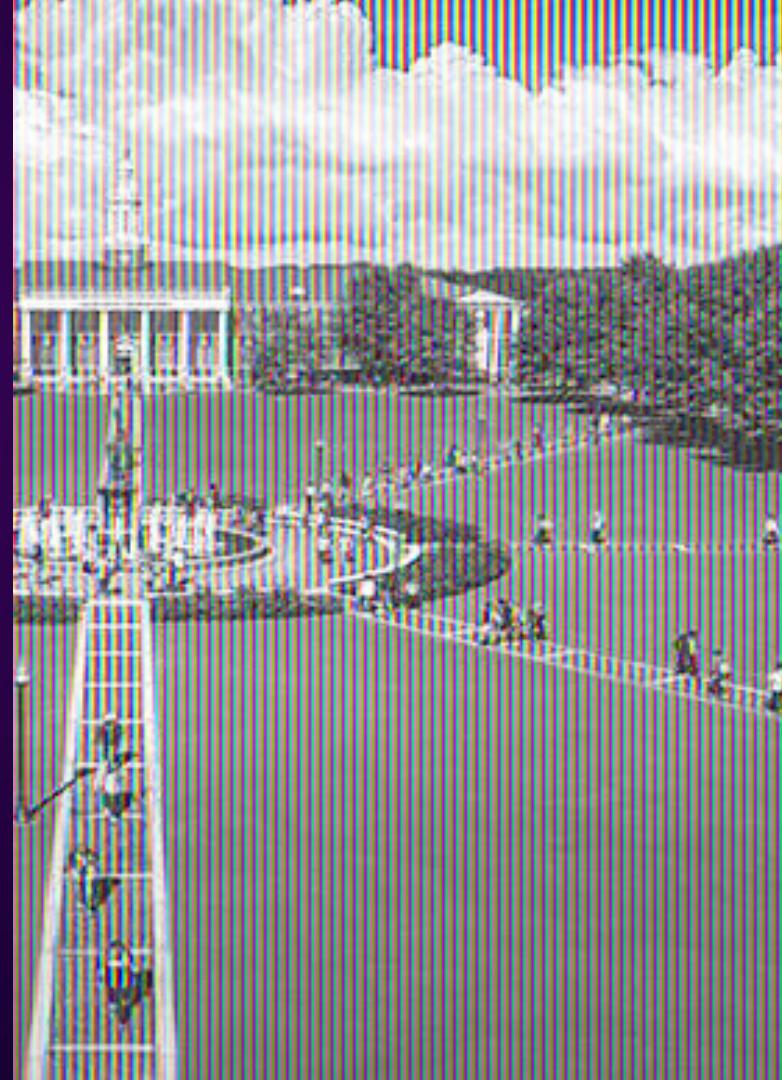


Troy Assist

a smart Gen-AI



Prepared By: Shankar
Bhattarai, Anil Khatiwada,
Bishal Awasthi





Problem Statement

- Students often struggle with getting timely and accurate information on college related queries.
- Common issues: admission process, housing details, deadlines, document requirements.
- Traditional systems (FAQs, email support) are slow or overloaded.

Solution Overview

A Generative AI-Powered college assistant that responds intelligently to student queries.

Use cases include:

- Admission queries
- Housing details
- Course requirements
- College facilities and many more.....

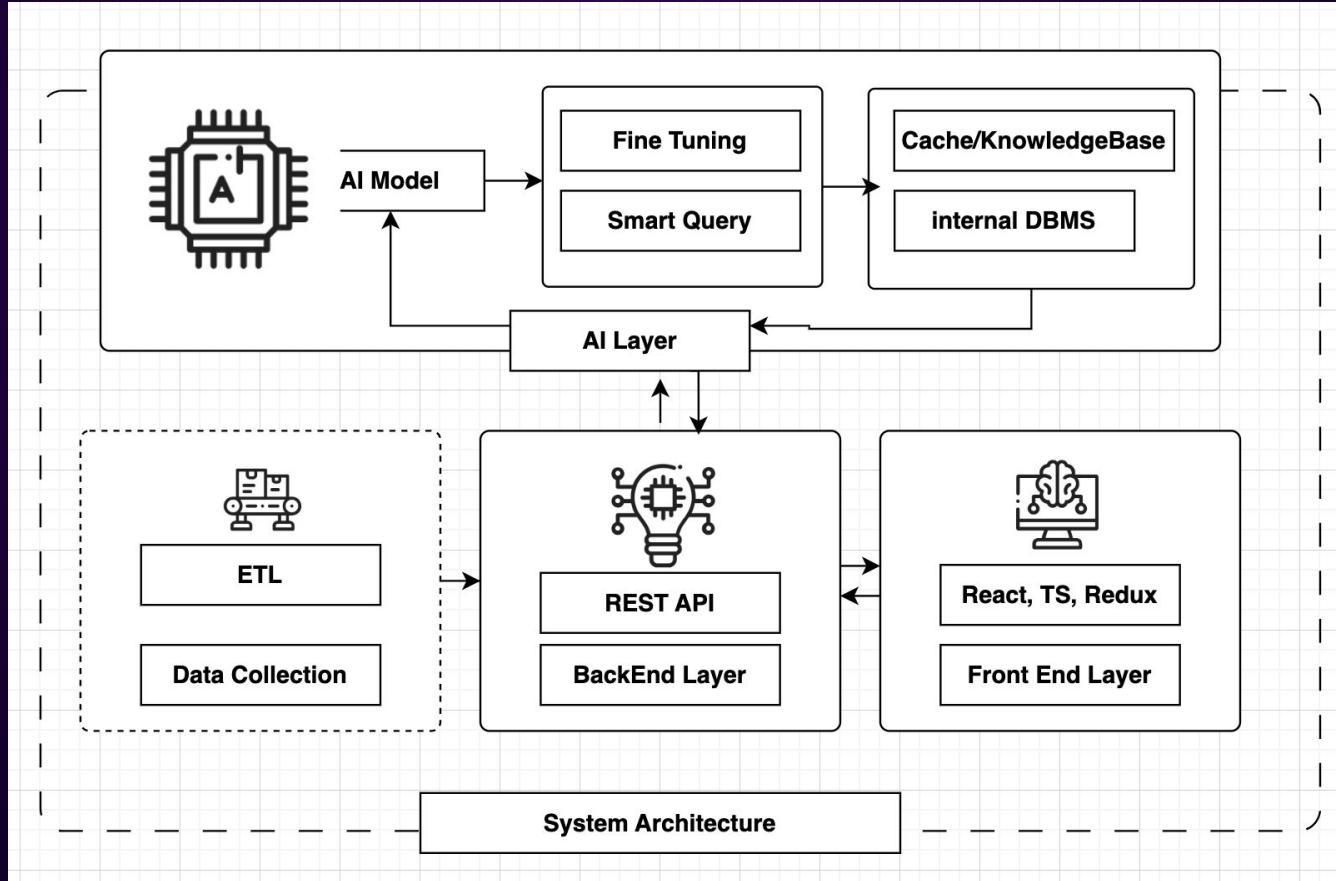


Features

- Conversational interface with real-time responses.
- Context aware answers.
- Dynamic topic coverage (admission, housing, deadlines).
- Clean and modern React-based UI.
- Self-learning based on feedbacks.



System Architecture





Tech Stack

- Backend: Django, Rest API (Python)
- Frontend: React JS, Redux, Typescript
- AI Model: Gemini (Generative AI)
- Database: SQLite
- ETL: Python Script, Beautiful Soup





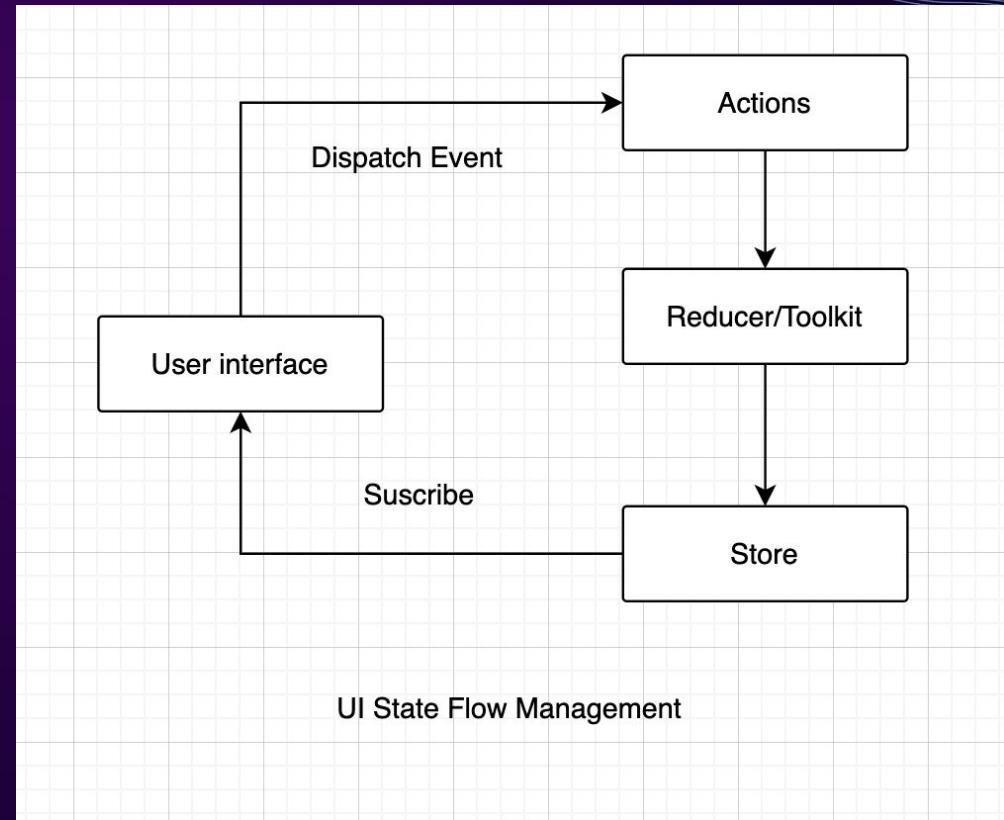
How it works



- User sends a query through frontend.
 - Django backend receives the request.
 - AI finds intent and search the data.
 - Based on found data generates a relevant response.
 - Response is returned and displayed in chat UI.
- 
- 

Structure Overview FrontEnd

```
4 TROY-STUDENT-ASSIST/
5   └── node_modules/
6   └── public/
7   └── src/
8     ├── assets/           ← collapsed
9     ├── environments/
10    └── models/
11    └── portions/         ← collapsed
12    └── rcomponents/
13      └── services/
14        ├── styles/       ← collapsed
15        └── utils/        ← collapsed
16    └── TroyDesktop.tsx
17    └── TroyMobile.tsx
18    └── Store.ts
19    └── vite-env.d.ts
20
21   .env
22   .gitignore
23   eslint.config.js
24   index.html
```



Application UI

The screenshot shows the Troy University website with a dark red header. The header includes the Troy University logo, a search icon, and links for "Apply", "Visit", "Give", and a menu icon. A prominent feature is a white pop-up window for "Troy Student Assist" labeled "BETA". The window contains a message from the AI assistant, a "Feedback" link, and a button to "Explore what Troy Student Assist.. can do". Below the window is a text input field with placeholder text "or, type your question". The main page background shows a night view of the university campus with buildings and trees. A large circular watermark of the Troy University logo is overlaid on the page.

TROY UNIVERSITY

Troy Student Assist

BETA

Clear chat

who are you

Hi! I'm Troy University's AI assistant, still under development by Anil Khatiwada, Shankar Bhattacharai, and Bishal Awasthi. As far as I have information, I can access data on programs, departments, and faculty, but I'm learning more every day! For more detailed information, please visit troy.edu.

Feedback

Explore what Troy Student Assist.. can do >

or, type your question

Welcome to Troy student assist.

General college information only and is not a substitute for official academic advising.

Data Collection

We have systematically gathering information from troy.edu sources to analyze, understand, and use for our decision-making Gen Ai Chatbot.



Data collection Pipeline



Extract



Transform



Load

Build a structured knowledge base to power a Gen AI chatbot for Troy University, supporting academic queries, campus services and more.....

Data Collection Overflow

Web Crawling

- Sources: Public URLs on <https://www.troy.edu>
- Tools used: BeautifulSoup
- Scope: Admissions, Campus Life, Faculty Profiles
- Subpages (e.g., /academics/, /support/).

Text-based Cleaning

- Regex-based Cleaning:
 - Remove HTML tags, boilerplate text
 - Extract Q/A patterns, lists
- Language Normalization:
 - Lowercase conversion, punctuation filtering.
 - Removal duplicate phrases.

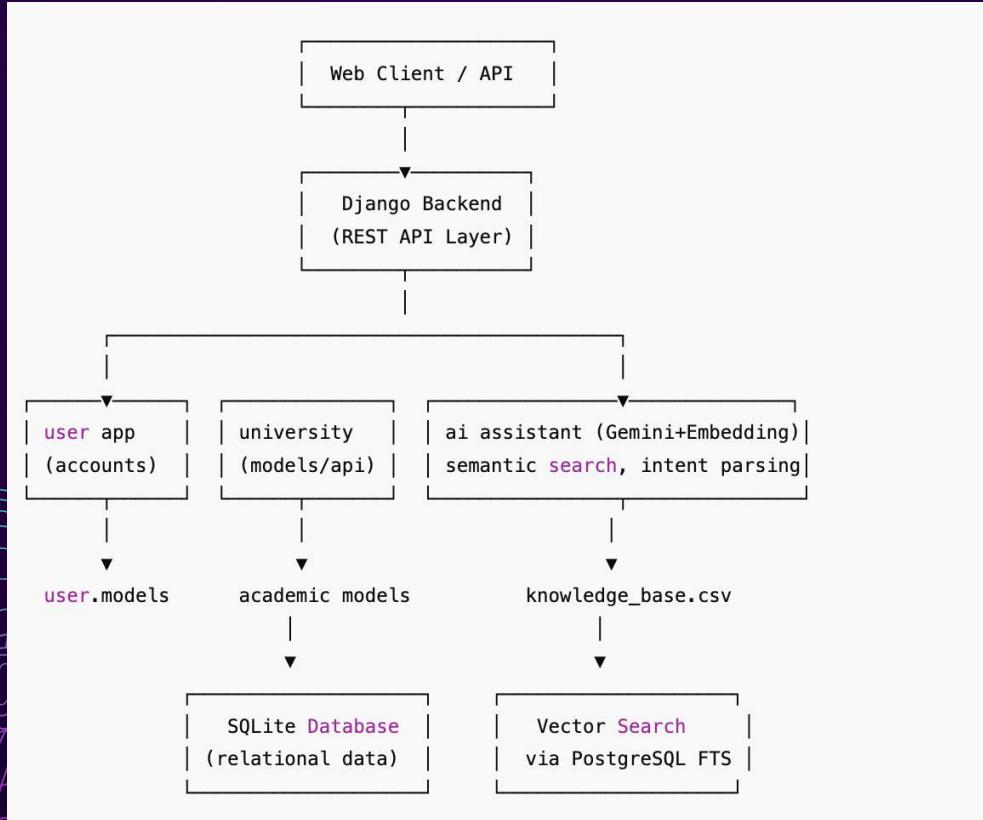
CSV Structuring Format

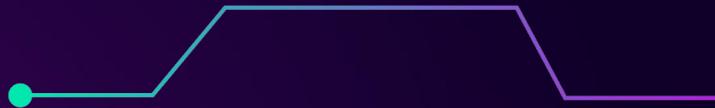
Format: csv
CopyEdit: Question, Answer, Source.

Refinement Storage

Stored in SQLite.

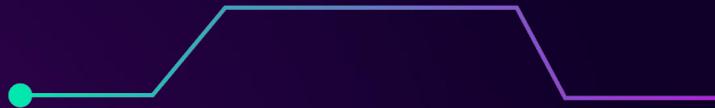
Structure Overview: BackEnd





Query Flow & Intent Detection

1. User submits a query (e.g., "Show me computer science faculty").
2. IP-based user context is retrieved or initialized from cache.
3. Gemini AI is used to:
 - a. Extract important keywords for knowledge base search.
 - b. Analyze the query to detect intent (e.g., faculty_info) and entities (e.g., department name).
4. Context and conversation history are used to enrich understanding.



Data Retrieval Pipeline

1. Based on detected intent, query routes to appropriate Django ORM model:
Examples: Department, Faculty, Course, Student, etc.
2. Supports dynamic filtering using extracted entities (e.g., filter by department name or rank).
3. If no relevant data found in database:
 - a. Fallback to semantic keyword search on KnowledgeBaseEntry using Django SearchVector and similarity ranking.

Gemini-Powered Response Generation

Relevant data is passed into Gemini prompt with a natural language response template.

Response is:

Short, informative, and branded as "Troy University AI assistant".

Language-sensitive: responds in user's input language.

Suggestions for follow-up questions generated based on intent (e.g., related courses or departments).

Frontend Integration & Output

Returns structured JSON for frontend use:

- > text response.

- >articles, options, knowledge_base blocks.

- > suggested questions for continued interaction.

Context is updated after each response to maintain history across sessions.

Errors handled gracefully with informative fallback messages.

Thank you :)

**We appreciate your time and attention.
We're excited about the future of smart,
student-centered support — and we
hope Troy Assist is just the beginning.**

**By: Shankar Bhattarai, Anil Khatiwada,
Bishal Awasti**

