Gen Ai Hub

UCS 503 Software Engineering Project Report

END-Semester Evaluation

Submitted by:

(102303557)Sachin Goyal

(102303559)Vrinda Chhabra

(102303563) Priyal Gupta

(102303580)Raghav Chhabra

BE Third Year, COE

Group No: 2C42

Submitted to:

Prof. Reaya Grewal

Computer Science and Engineering Department

TIET, Patiala

Dec,2025

TABLE OF CONTENTS

S.No.	Assignment	Page No.
1.	**Project Selection Phase**	
1.1	Software Bid	4-5
1.2	Project Overview	6-8
2.	**Analysis Phase**	
2.1	Use Cases	
2.1.1	Use-Case Diagrams	8
2.1.2	Use Case Templates	9-10
2.2	Activity Diagram and Swimlane Diagrams	11
2.3	Data Flow Diagrams (DFDs)	
2.3.1	DFD Level 0	12
2.3.2	DFD Level 1	12
2.3.2	DFD Level 2	13
2.4	Software Requirement Specification in IEEE Format	14-24
2.5	User Stories and Story Cards	25-31
3.	**Design Phase (At least two significant cases of each diagram)**	
3.1	Class Diagram	32

3.2 Sequence Diagram 33-34

3.3 Collaboration Diagram 35

3.4 State Chart Diagrams 36-37

4. Implementation

4.1 Component Diagrams 38

4.2 Deployment Diagrams 39

4.3 Screenshots 40-44

5. Testing

5.1 Test Plan 45-46

5.2 Test Cases 47-57

5.3 Test Reports by Peers 58

1. Project Selection Phase

1.1) Software Bid

Name	Roll No	Project Experience	Programming Language used	Signature
Sachin Goyal	102303557	Personal Ai Health Assistant	Python	Sachin
Vrinda Chhabra	102303559	AI Based Plagiarism Detector	Python	Vrinda
Priyal Gupta	102303563	Email Delivery Event Tracking with SendGrid	JavaScript	Priyal
Raghav Chhabra	102303580	Cloak of Invisibility	Python	Raghav

Programming Language / Environment Experience

1. Python
2. JavaScript
3. C++

Choices of Projects:

	Project Name	Unique Selling Point
First Choice	Gen Ai Hub	An all-in-one Generative AI Hub that goes beyond chat—integrating agentic AI, domain-specific assistants, and database intelligence to provide personalized learning, seamless content creation, and intuitive data interaction—all under a single unified platform.
Second Choice	PredictX	PredictX is a machine learning–powered prediction model that analyzes historical and real-time data to deliver accurate, actionable forecasts. Its unique strength lies in combining high precision with adaptability, continuously learning from

		new data to remain effective across diverse domains like business, e-commerce, and finance.
Third Choice	Hyperlocal Marketplace for College Campus	To build a mini **OLX-type hyperlocal marketplace** for college students where they can buy and sell items like food, dresses, and other goods within the campus. Users can select their hostel or location to connect with nearby buyers and sellers easily. The platform includes secure user verification, image uploads, and a chat system to enable smooth, trustworthy, and convenient transactions within the college community.
Fourth Choice	Smart Adaptive Study Planner (SASP)	It creates personalized study schedules based on a student's learning style, speed, and availability. It breaks down study materials like PPTs and PDFs into manageable daily tasks, balancing workload and suggesting breaks. The app supports file uploads, real-time notifications, and progress tracking.

Additional Remarks/ Inputs

This **GenAI Hub** unifies agentic AI, intelligent assistants, and natural language database interaction into a single platform, eliminating the need to switch between multiple tools. It showcases end-to-end skills in AI, integration, and product design while solving real problems for learners, developers, and professionals. The project demonstrates both strong technical expertise and real-world impact, making it scalable and future-ready.

1.2) Project Overview

Executive Summary

In today's AI ecosystem, users often switch between multiple fragmented tools—one for code assistance, another for content writing, another for learning or data queries. This results in inefficiency and lack of personalization. **GenAI Hub** is a unified web-based platform that integrates **agentic AI applications**, **intelligent assistants**, and **natural language database interaction** into one system. It empowers students, developers, and professionals to learn, create, and analyze effortlessly using AI. The system includes a **Course Generator**, **AI YouTube Blog Generator**, **Code Assistant**, **Math Solver**, **Summarizers**, and **Database Query Chatbot**. Key outcomes include:

(1) reducing context-switching by 70%,
(2) improving productivity by 40%, and
(3) achieving 95% response accuracy.

AI tools today are scattered, requiring users to juggle multiple platforms (ChatGPT, Copilot, Notion AI, etc.)—causing time loss, fragmented learning, and poor workflow continuity.

Proposed Solution:

GenAI Hub offers a single, intelligent interface with modular AI agents:

- **Flowchart Maker: Generates professional flowcharts and process diagrams automatically from user descriptions or text inputs, exporting them as JPG/PNG images.**
- **Course Generator: Builds personalized roadmaps and templates.**
- **AI YouTube Blog Generator: Converts video transcripts into structured blog posts.**
- **Code Assistant: Helps debug, explain, and optimize code.**
- **Math Solver: Provides step-by-step reasoning and formula derivations.**
- **Summarizers: Condenses YouTube and web content for quick understanding.**
- **Database Chatbot: Allows natural language queries without SQL.**
- **PersonaFlow: Interactive Character AI that simulates personalities from books, movies, or history.**

High-Level Architecture:

- **Frontend:** ReactJS , Tailwind CSS, responsive design.
- **Backend:** Python (FastAPI/Flask) orchestrating multiple AI agents.
- **AI Integration:** LangChain, Google Gemini, OpenAI, HuggingFace, Groq.
- **Database:** PostgreSQL / MongoDB.
- **Deployment:** Dockerized app on Render/AWS, CI/CD via GitHub Actions.

Non-Functional Requirements

- Response latency < 5 seconds
- 99.5% uptime
- AES-encrypted communication
- Modular architecture for scalability

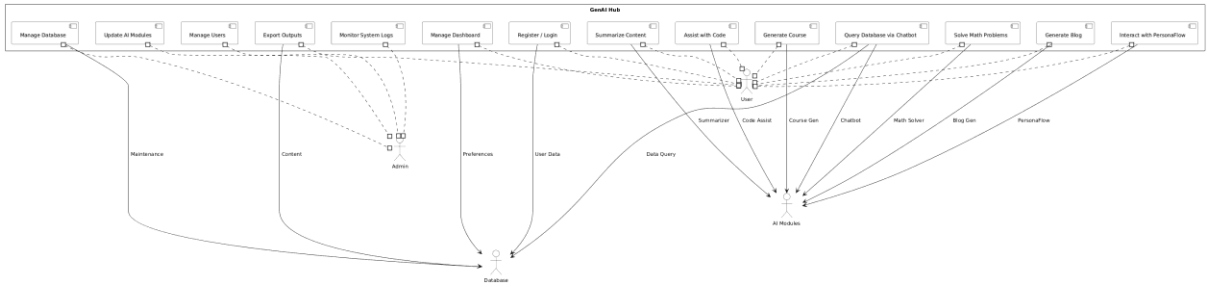
Novelty / Unique Selling Point (USP)

- **All-in-one:** Discovery, registration, attendance, analytics in one platform.
- **Smart recommendations:** Simple content-based suggestions from interests & history (Phase 2).
- **Frictionless check-in:** QR code badges reduce queue time and fake attendance.

  Lightweight: Mobile-first without app-store friction; offline access to tickets.

2) Analysis Phase

2.1.1) Use Case Diagram



2.1.2) Use Case Templates

Use Case Template 1

Section
Project: GenAI Hub – Unified Generative AI Platform

Pre-conditions

- **User has access to the web application (browser + internet).**
- **Backend server (FastAPI/Flask) is running.**
- **AI provider APIs (Gemini,Groq) are active.**
- **Database connection established (MySQL).**
- **User authentication module functional (Signup/Login).**
- **Configuration files, API keys, and routing registered.**
- **Models and agent pipelines initialized (Course Generator, Summarizer, PersonaFlow, SQL Bot,Flowchart Generator).**

Use Case
Multi-Agent Generative AI Assistance via GenAI Hub

Actors

- **Primary:**
  - **End User (Student, Developer, Researcher, Content Creator)**
- **Secondary:**
  - **Authentication System**
  - **GenAI Hub Backend**
  - **AI Model Providers (Gemini / Groq)**
  - **Database (User profiles, chat history,persona profile ,persona chats)**
  - **Tool Components (Course Generator, Blog Generator, Summarizer, SQL Bot, PersonaFlow)**

Purpose
To allow users to access a unified platform for generating courses, summaries, blog posts, AI personas, code help, math solutions, flowcharts,images and natural-language SQL queries—all through a single seamless interface.

Overview

User logs into GenAI Hub → selects an AI tool → enters input → backend constructs prompt → model selection performed → query sent to selected LLM → response parsed → results displayed and stored in the user's workspace.

Type
Primary – End-to-End User Interaction & AI Processing Flow

Main Success Flow

Step	Type	Description
1	SR	System shows Home Dashboard with all AI tools (Course Generator, Summarizer, etc.).
2	AA	User logs in or signs up.
3	SR	System verifies credentials via database.
4	AA	User selects a module (e.g., "Course Generator").
5	SR	Input form appears with required fields (topic, URL, prompt, query).
6	AA	User enters input prompt.
7	SR	Input validation (empty field, invalid URL, unsupported format).
8	SR	Backend constructs enhanced prompt + metadata.
9	SR	Model selection engine routes the query to chosen LLM (Gemini / Groq / Nvidia).
10	SR	Query sent to LLM API; response generated.
11	SR	Response Parser cleans, structures, and formats results.
12	SR	Output displayed on UI with options: "Copy", "Download", "Save to Projects".
13	SR	System stores output linked to user profile/history.
14	AA	User reviews, downloads, or continues another query.

Exception Flows

Case	Description
E1	Invalid Input → System displays error ("Enter a valid URL/topic/query").
E2	API Failure → Fallback to secondary model provider.
E3	Model Timeout → Retry using alternate LLM.
E4	Authentication Failed → Prompt user to retry login/signup.
E5	Database Unavailable → Temporary local session (projects not saved).
E6	Exceeded Rate Limit → Notify user of usage cap (MVP constraint).
E7	Unsupported Persona / Topic → System asks for a refined prompt.

Alternate Flows

Case	Description
A1	User uploads a file (document, transcript) instead of text; system extracts text automatically.
A2	User switches model manually (e.g., Gemini → Groq).

Case	Description
A3	User chooses “Auto-Chain Workflow” (future phase — summarize → convert to blog → save).
A4	User compares outputs across models (side-by-side display).
A5	User selects PersonaFlow → persona traits loaded from predefined database.
A6	SQL Chatbot → executes generated SQL and displays results + query output.

Post-conditions

- **Results generated and displayed.**
- **Output stored in Project History under user’s profile.**
- **Logs updated for analytics and debugging.**
- **(Optional) Downloadable file created (.txt, .csv, .docx depending on module).**

Data Inputs

- **Text prompts**
- **YouTube URLs**
- **Persona selection**
- **Natural language SQL queries**
- **Authentication data (username(unique), password)**

Data Outputs

- **AI-generated responses (roadmaps, summaries, SQL queries, persona chats,flowcharts)**
- **Downloadable documents (DOCX, TXT)**
- **Stored chat/project history**
- **Optional graphs/structured lists**
- **API logs & metadata**

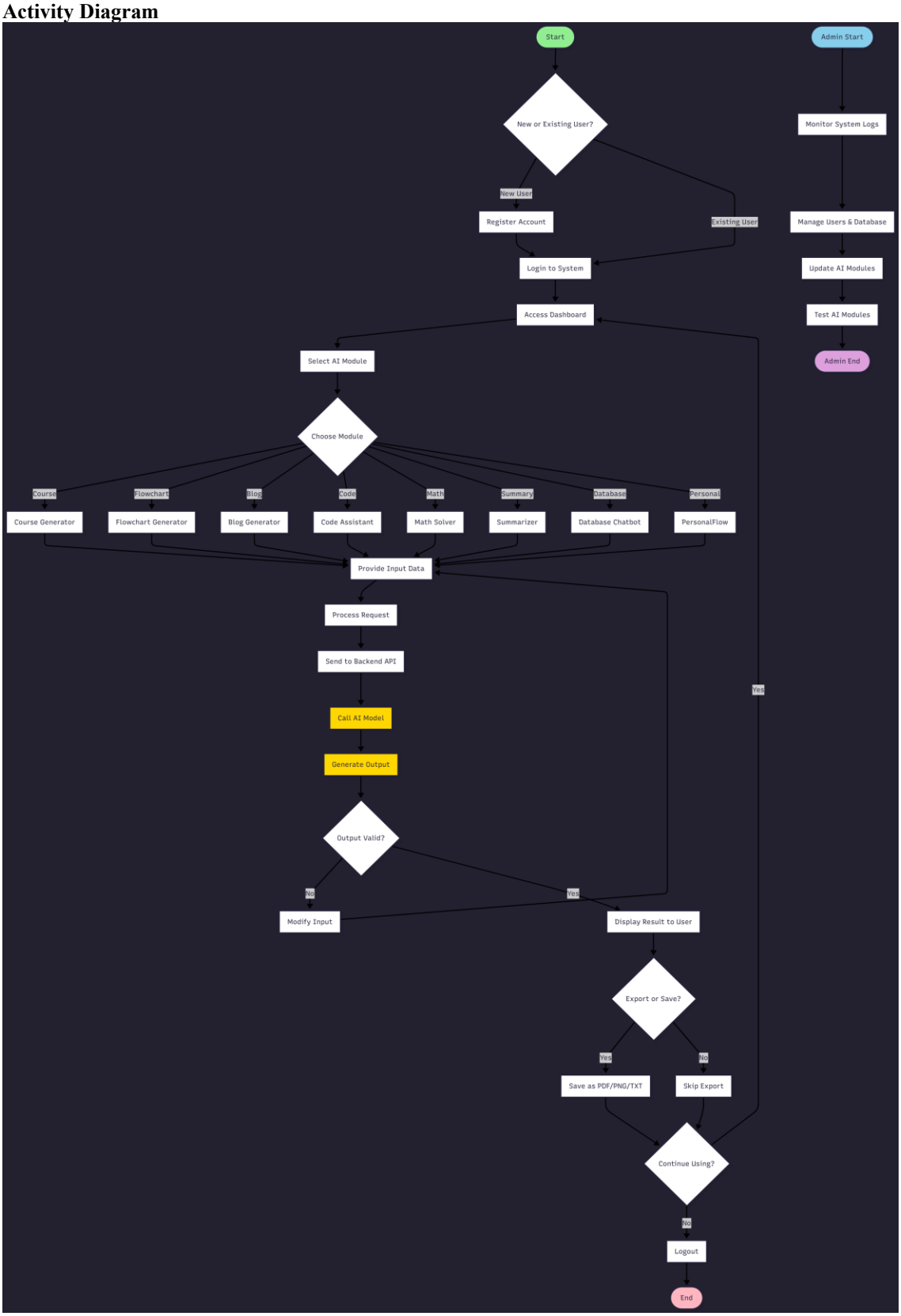
Description (Use Case Summary)

GenAI Hub provides a unified multi-agent framework where all AI tools operate through a common backend pipeline. Using a modular architecture ,the system supports diverse tasks such as content generation, data retrieval, SQL conversion, persona conversations, and code/math assistance.
The user interacts with a clean dashboard, while the backend orchestrates prompt handling, LLM routing, parsing, and storage.

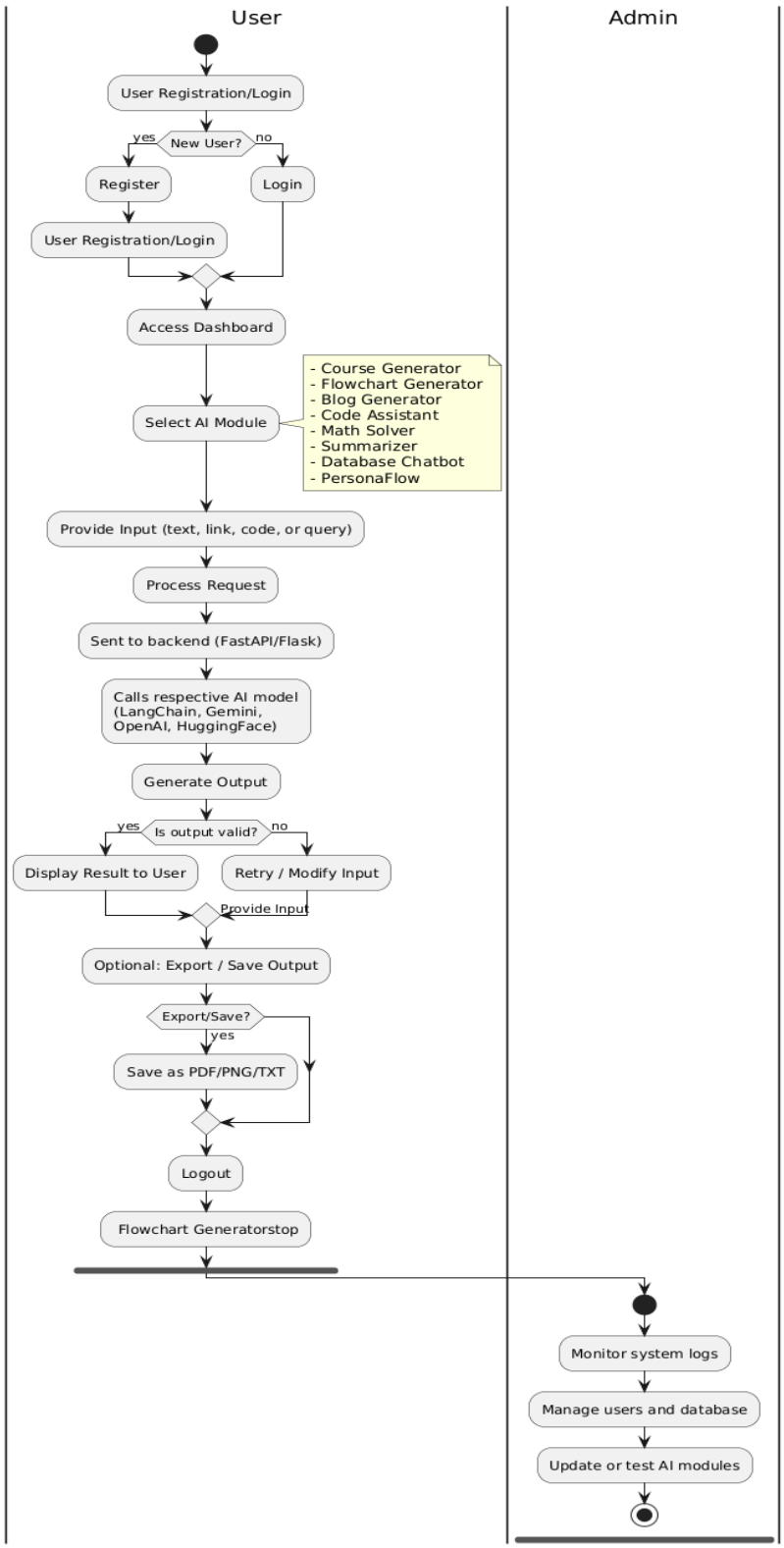
This ensures a fast (≤5 sec), accurate (≈95%), and scalable multi-tool AI experience.

2.2 Activity and Swimlane Diagram

Activity Diagram

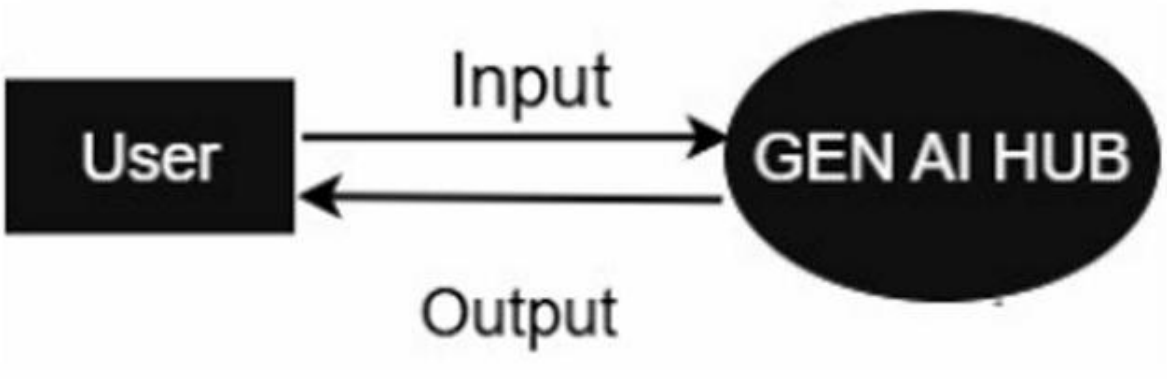


2.2 Swimlane Diagram

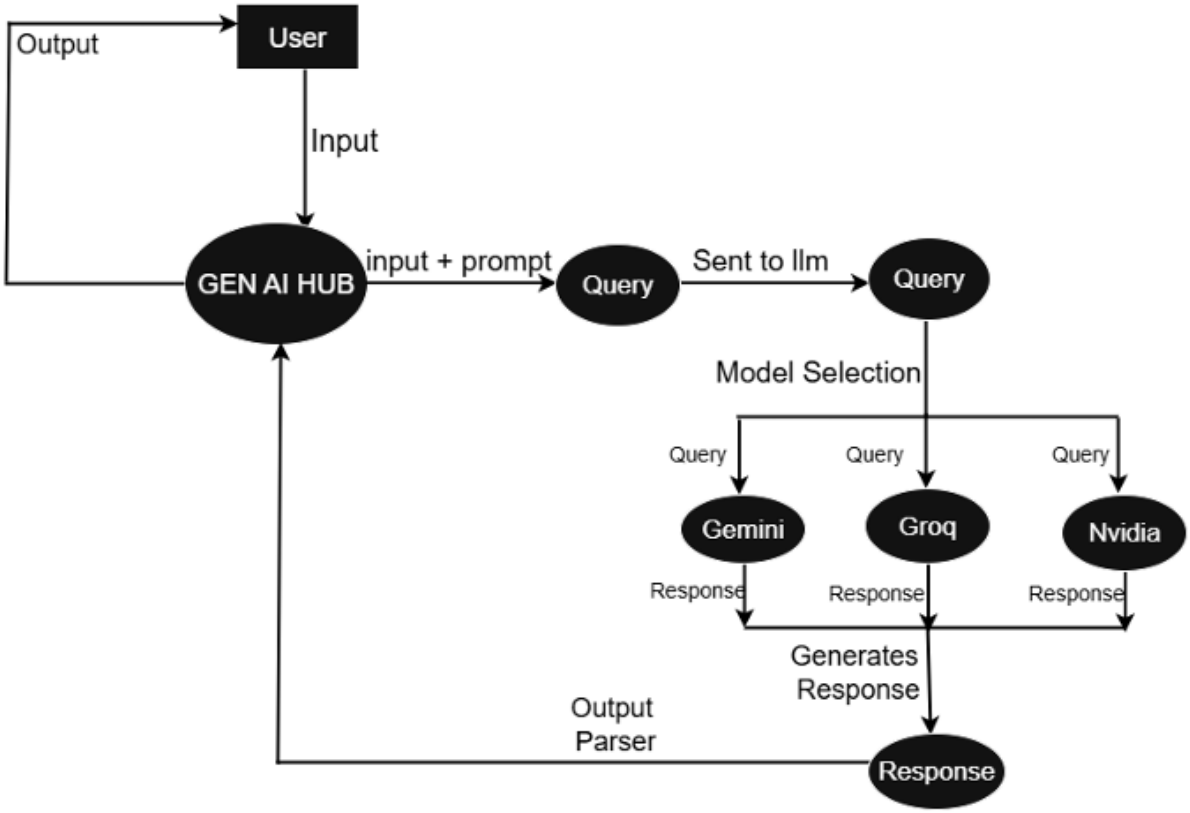


2.3 Data Flow Diagram

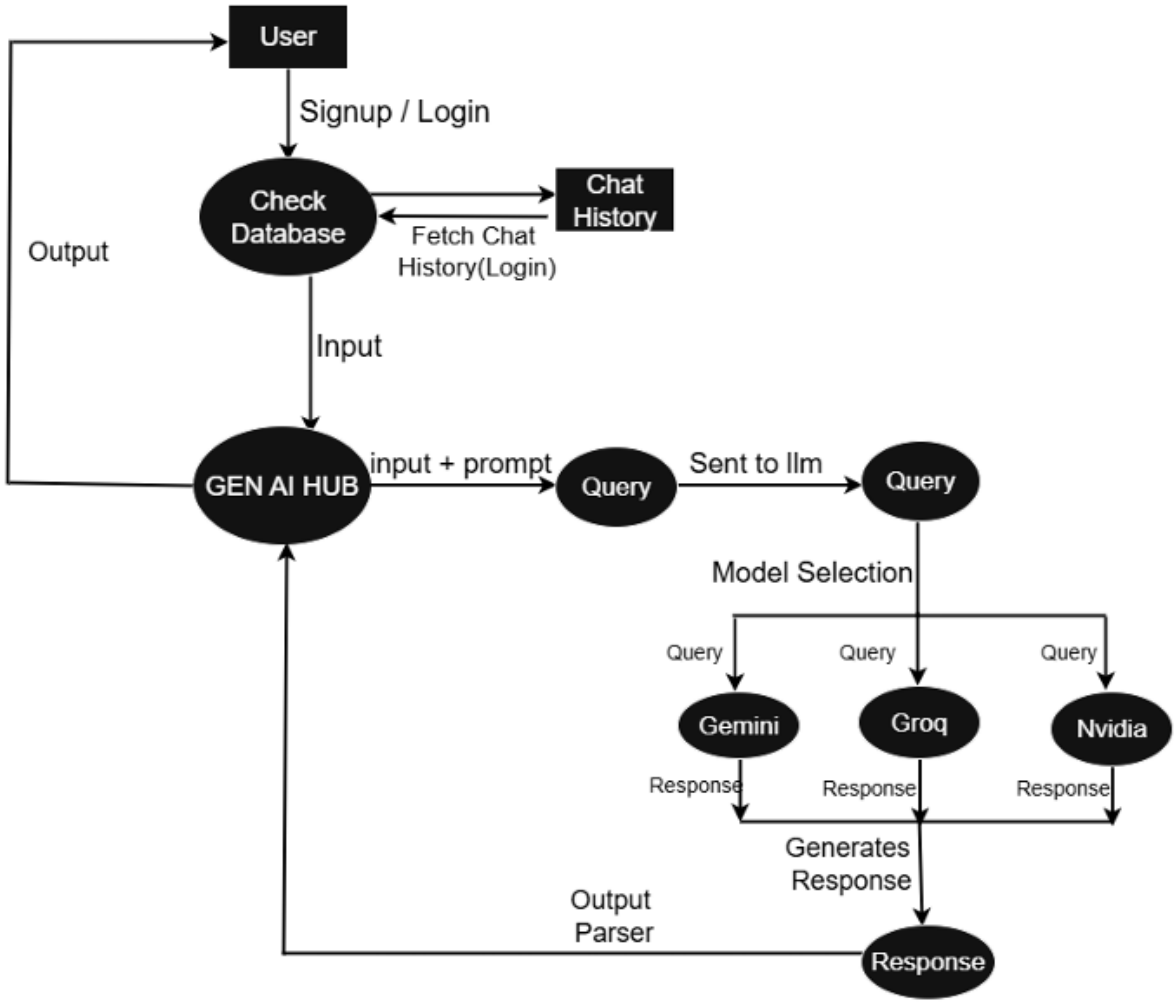
2.3.1) DFD level 0



2.3.2 DFD level 1



2.3.2 DFD level 2



2.4 Software Requirement Specification in IEEE Format

2.4.1 Introduction

2.4.1.1 Purpose of this Document

This Software Requirements Specification (SRS) document provides a detailed overview of the Gen AI Hub software product, its parameters, and its goals. The document describes the project's target audience, user interface, and the functional and non-functional requirements. Its purpose is to align all stakeholders on a shared understanding of the project's scope.

2.4.1.2 Scope of the Development Project

The goal is to design a web-based platform, the Gen AI Hub, which offers a suite of AI tools to users. This system will include a Course Generator, an AI YouTube Blog Generator, PersonaFlow for character conversations, Intelligent Assistants (Code Assistant, Math Solver, Summarizers), and a Database Query Chatbot. The platform will serve various users, including students, developers, content creators, and researchers, by providing efficient tools to automate tasks and enhance productivity.

The software must perform the following core operations:

- Course Generation: Generate educational roadmaps and templates based on user input.
- AI YouTube Blog Generation: Create blog posts from YouTube video content.
- PersonaFlow: Facilitate seamless, natural language conversations with AI characters.
- Intelligent Assistants: Provide specialized assistance for tasks like coding, solving math problems, and summarizing content from web pages or YouTube videos.
- Database Query Chatbot: Translate natural language queries into SQL database commands.

The system will be initially implemented as a pilot phase for a small group of users before a wider, public release.

2.4.1.3 Definitions

Table 1: Definitions for most commonly used terms

S.No.	Term	Definition
1	Gen AI Hub	The Software platform that will host all the Generative AI tools and Agents

2	Course Generator	A tool within the Gen AI Hub that creates educational roadmaps and templates.
3	PersonaFlow	A tool that enables natural language conversations with a variety of AI-driven characters or personas.
S.No.	Term	Definition
4	Intelligent Assistants	A group of specialized tools that provide assistance for specific tasks, such as coding, solving math problems, and summarizing content.
5	Database Query Chatbot	A tool that translates user queries in natural language into SQL commands.

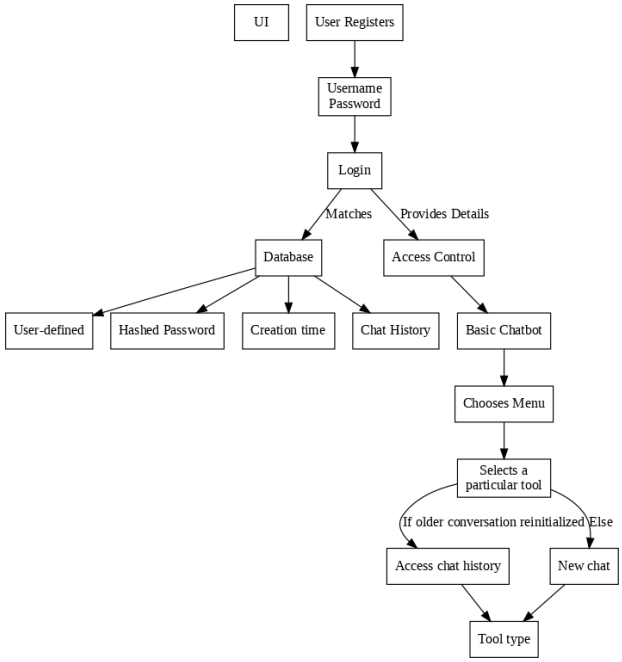
2.4.2 Overall Description

2.4.2.1Product Perspective

The Gen AI Hub will be a cloud-based web application. It will be a self-contained product, but it will interact with external services to function properly. It will require an internet connection, a standard web browser, and user authentication to access its features.

The system will be composed of several modules, as depicted in the following block diagram. This diagram illustrates the interaction between the user and the system, as well as the internal communication between different modules.

Gen AI Hub System Block Diagram



2.4.2.2 Product Functions

The Gen AI Hub will offer a suite of AI tools and will include a complete user management system to ensure a secure and personalized experience.

The product functions are as follows:

1. User Authentication and Management: The system will allow users to sign up for an account and log in to access the tools. This will ensure that all user data and saved projects are private and tied to their account. The system will also have a feature for users to manage their profiles and passwords.
2. Generate a Course Roadmap: A user will be able to input a topic or skill, and the system will generate a structured learning roadmap with suggested resources and templates.
3. Convert a YouTube Video to a Blog Post: A user will be able to input a YouTube video URL, and the system will generate a blog post summary, including key points and a transcript.
4. Engage in Persona-driven Conversation: A user will be able to select an AI persona and engage in a natural language conversation, with the AI maintaining the selected character traits and communication style.
5. Provide Specialized AI Assistance: The system will offer distinct interfaces for:
    - Code Assistance: Answering programming questions and generating code snippets.
    - Math Solving: Providing step-by-step solutions to mathematical problems.
    - Content Summarization: Summarizing content from a provided URL or a block of text.
6. Translate Natural Language to SQL: A user will be able to input a database query in plain English, and the system will generate the corresponding SQL code.
7. Project Management and Saving: A core function of the product will be the ability for users to save and organize their generated content and project history within their profile. This allows them to revisit their work and ensures continuity across sessions.
8. Feedback and Support System: The product will include a basic system where users can report issues, provide feedback, or access a help section to improve their experience and help the development team identify bugs and new features.

2.4.2.3 User Characteristics

The target users for the Gen AI Hub are diverse, each with different expertise levels. The system will be designed to be user-friendly for all, from casual users to technical professionals.

2.4.3 Students: Need help with coursework, research, and learning new skills.

2.4.4 Content Creators: Require tools to automate content generation and repurposing.

2.4.5 Developers & Researchers: Use the platform for coding assistance, data

analysis, and exploring new AI capabilities.

Users are expected to be computer-literate and familiar with basic web applications. They are not required to have a deep understanding of the underlying AI models or how the software functions internally.

2.4.2.4 General Constraints, Assumptions and Dependencies

Constraints:

2.4.3 The software has to be a web-based application, accessible from all major browsers.

2.4.4 The user interface must be simple and intuitive.

2.4.5 Response time for generating content or processing a request must be no longer than ten seconds for 95% of transactions under normal conditions.

2.4.6 All data transmitted must be encrypted to ensure security.

Assumptions:

2.4.7 Users have a stable internet connection.

2.4.8 External services and APIs (like YouTube's API, payment gateways) are functional and accessible.

Dependencies:

2.4.9 The platform depends on the availability and performance of the chosen cloud infrastructure.

2.4.10 The functionality of specific tools (e.g., YouTube Blog Generator) is dependent on the continued availability and policies of external services.

2.4.2.5 Apportioning of Requirements

The Gen AI Hub project is to be implemented in a phased manner to ensure a stable and scalable product. The project will be divided into the following phases:

Phase I: Minimum Viable Product (MVP)

This phase focuses on building the core functionalities and establishing a basic user base. The key requirements to be implemented are:

2.4.3 User Authentication System: The basic login, signup, and user profile management features will be fully functional. This is a foundational requirement to ensure user data privacy and personalized experiences from the start.

2.4.4 Generate a Course Roadmap: This core tool will be launched to allow users to create and save learning roadmaps.

2.4.5 Content Summarization Assistant: As a key utility tool, content summarization will be available for both text blocks and website URLs.

2.4.6 Project Saving and Management: Users will be able to save and retrieve their generated roadmaps and summaries within their profiles.

Phase II: Feature Expansion and Performance Optimization

Following the successful deployment of the MVP, this phase will introduce additional features and enhance the existing ones based on user feedback.

2.4.7 AI YouTube Blog Generator: This tool will be added, allowing users to convert video content into a blog post format.

2.4.8 PersonaFlow: The conversation-driven AI persona tool will be integrated into the platform.

2.4.9 Code and Math Assistants: The specialized AI assistants for programming questions and mathematical problem-solving will be launched.

2.4.10 Scalability and Performance Upgrades: The backend infrastructure will be optimized to handle a larger user load and more complex queries.

Phase III: Advanced Features and Integrations

The final phase will focus on advanced functionalities and a broader market strategy.

2.4.11 Translate Natural Language to SQL: This advanced feature will be implemented, catering to more technical users.

2.4.12 External API Integrations: The system will be enhanced to allow for deeper integrations with external services.

2.4.13 Feedback and Support System: A dedicated in-app feedback mechanism will be fully integrated.

2.4.14 Monetization Strategy: A premium tier with advanced features or higher usage limits will be introduced in this phase, along with a FinOps Dashboard to manage the costs of API calls and model consumption across providers.

Future Updates and Enhancements

After the core features are released in the initial phases, the following updates can be planned to enhance the Gen AI Hub and provide more value to users:

2.4.15 User-Generated Content and Sharing: Implement a feature that allows users to share their generated content, such as course roadmaps or blog posts, publicly or with other users on the platform. This would create a community aspect and allow for collaborative learning.

2.4.16 Orchestration and Automation: Add a feature that allows users to create workflows by chaining different tools together. For example, a user could set up an automated workflow that takes a daily news YouTube video, summarizes it into a blog post, and then posts it to their linked blog. This would streamline complex, multi-step tasks.

2.4.3 Specific Requirements

2.4.3.1 External Interface Requirements

2.4.4 **User Interface:** The system will have a simple graphical user interface (GUI) with text fields for input and a display area for output. The design will be clean and minimal, without the use of complex graphics, sounds, or animations.

2.4.5 **Software Interfaces:** The system will communicate with several external APIs using standard protocols. This includes APIs for YouTube, a potential payment gateway, and any third-party libraries used for specific AI models. The system will also interface with a central database server to store and retrieve data.

2.4.4 Detailed Description of Functional Requirements

The following sections detail the specific functional requirements for each core tool within the Gen AI Hub.

Purpose	Description
Inputs	What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input.
Processing	Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations.
Outputs	The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced as output.

Functional Requirements for the Course Generator

2.4.6 **Purpose:** To create a structured learning roadmap for a user-specified topic.

2.4.7 **Inputs:** A user will type a topic (e.g., "Python for Data Science") into a text field.

2.4.8 **Processing:** The system will use an AI model to generate a multi-step roadmap. It will perform validity checks on the input and handle cases where the topic is too vague or complex by prompting the agents for more specific information, making it a multi-agent system.

2.4.9 **Outputs:** The output will be a display of a structured roadmap, including learning modules, suggested resources (textual descriptions), and a download link for a template.

Functional Requirements for the AI YouTube Blog Generator

2.4.10 **Purpose:** To transform a YouTube video into a textual blog post.

2.4.11 **Inputs:** A user will paste a valid YouTube video URL into a text field.

2.4.12 **Processing:** The system will access the YouTube API to retrieve the video transcript and metadata. It will then process the text to extract key themes and generate a concise blog post. It will handle errors for invalid URLs or videos without transcripts.

2.4.13 **Outputs:** The output will be a full blog post displayed on the screen, with options to copy the text or download it in a .txt or .docx format.

Functional Requirements for the Database Query Chatbot

2.4.14 **Purpose:** To translate a natural language question into an executable SQL query.

2.4.15 **Inputs:** A user will type a question in a chat-like interface (e.g., "Show me the names of all employees in the marketing department").

2.4.16 **Processing:** The system will use a language model to parse the user's intent and generate a syntactically correct SQL query. It will not only execute the query, but will provide the query used also. It will handle invalid or ambiguous requests by asking for clarification.

2.4.17 **Outputs:** The output will be a text containing the generated SQL query and the retrieved result, which the user can then copy.

Functional Requirements for PersonaFlow

2.4.18 **Purpose:** To facilitate a seamless and natural language conversation with a pre- selected AI persona, ensuring the AI maintains its designated character traits and communication style throughout the interaction.

2.4.19 **Inputs:** A user will select a persona from a list (e.g., a historical figure, a fictional character, or a professional guide). The user will then type their conversation prompt or query into a chat interface.

2.4.5 Performance Requirements

- Load Time: The user interface for each tool must load in less than 5 seconds.
- Transaction Speed: 95% of all AI-driven requests (generation, summarization, etc.) must be completed within 10 seconds under normal load.
- Concurrency: The system must be able to support a large number of simultaneous users accessing different tools without significant performance degradation.

2.4.6Quality Attributes

2.4.6 Usability: The product must be intuitive and easy to use for all user types, with a simple, clear interface.

2.4.7 Reliability: The system should have minimal downtime and be able to handle unexpected inputs and errors gracefully.

2.4.8 Security: All user data and communications must be secure. Access to user-specific data should require proper authentication. All administrative actions will be logged.

2.4.9 Scalability: The system must be able to handle an increasing number of users and transactions as the platform grows.

2.4.6 Changes History

30-08-2025	Version 1.0 Initial Release

2.4.7Document Approvers

Prof. Reaya Grewal

Designation : Professor

Date : December 4, 2025

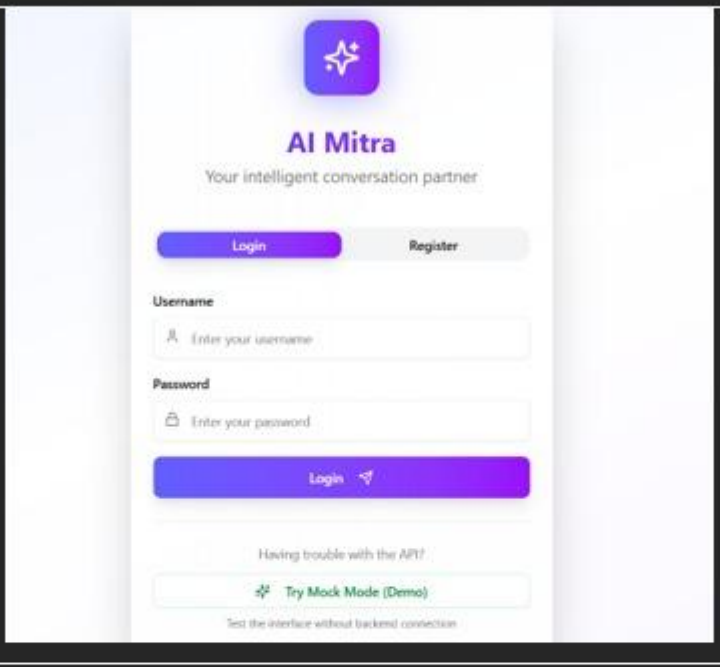
2.5 User stories and cards

2.5.1 User Registration Screen

As a new user, I want to create an account easily so I can start using AI Mitra without confusion or technical difficulty.

#001	User Registration Screen

As a new user, I want to create an account easily so I can start using AI Mitra without confusion or technical difficulty.



Confirmation

1. **Success:**
   When the user enters a valid username & password and clicks ***Register***, the system should show a loading state ("Creating Account"), successfully create the user account and redirect the user to the login page.

The *Mock Mode* button should open the demo version without requiring backend connectivity.

2. **Failure:**

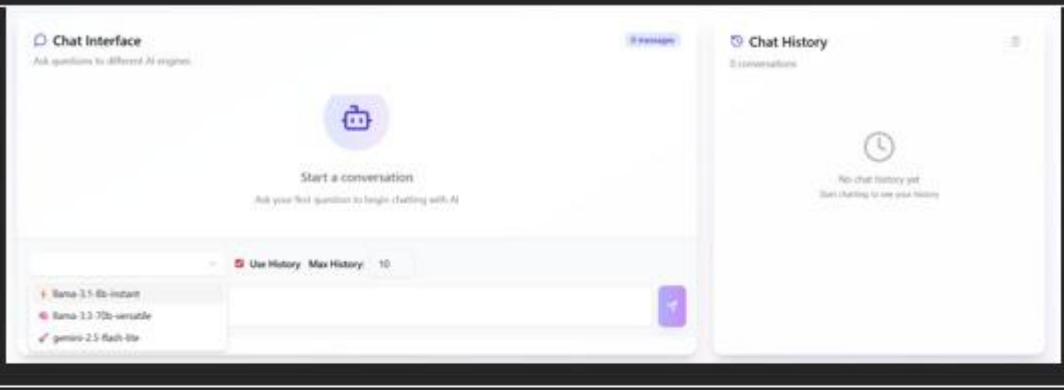
   2.1 If username or password is invalid or missing, the system should display a clear validation error.
   2.2 If account creation fails (API down, backend error), the system should stop the loading animation and show an error.

2.5.2 AI Assistant General Chat Interface

As a user, I want to ask questions and chat with different AI models so I can receive responses tailored to my needs.

#002	AI Assistant General Chat Interface

As a user, I want to ask questions and chat with different AI models so I can receive responses tailored to my needs.



Confirmation

1. Success:

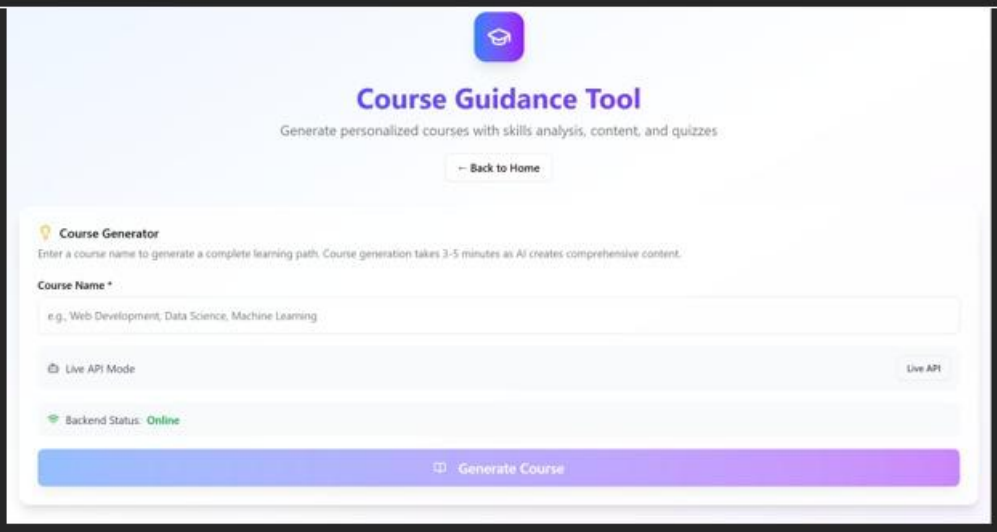
1.1 When the user selects an AI model from the dropdown, the system should switch to that model and prepare it to generate responses.
1.2 When the user types a message and clicks the Send button, the system should display the user's message, send it to the selected AI model, and show the model's response in the chat window.
1.3 If the user enables Use History, the system should include the last N messages (as defined by Max History) in each request.
1.4 After each completed conversation, the Chat History panel should update to include the newly created conversation thread.

2. Failure:

2.1 If no model is selected and the user attempts to send a message, the system should show a warning indicating that a model selection is required.
2.2 If message sending fails due to API or network issues, the system should not show any placeholder response and should instead display an appropriate error message.
2.3 If chat history retrieval fails, the system should continue functioning without history and inform the user that history is unavailable.
2.4 If the system fails to save chat history due to storage issues, it should avoid crashing and show a message indicating that chat history could not be saved.

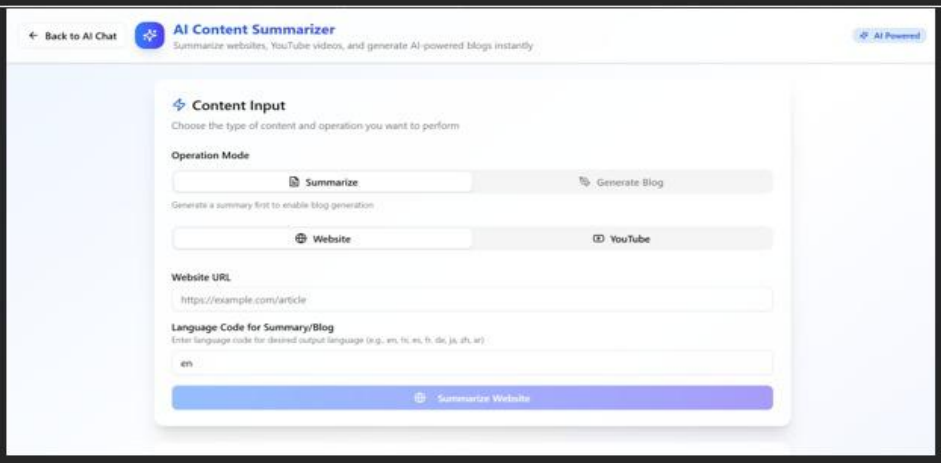
2.5.3 Course Guidance Tool

As a new user, I want to enter a course name easily and generate a personalized learning path, so that I can start using the Course Guidance Tool without confusion or technical difficulty.

#003	Course Guidance Tool
As a new user, I want to enter a course name easily and generate a personalized learning path, so that I can start using the Course Guidance Tool without confusion or technical difficulty.	
	
Success: When the user enters a valid course name and clicks *Generate Course*, the system should display a loading state indicating that the course is being generated. After successful processing, it should present the complete learning path on the results page. If Mock Mode is selected, it should open a demo-generated course without requiring backend connectivity. **Failure:** If the course name is missing or invalid, the system should show a clear validation message and prevent the generation process from starting. If the backend API fails or encounters an error during course generation, the system must stop the loading animation and display an error message, allowing the user to retry without refreshing the page.	

2.5.4AI Content Summarizer

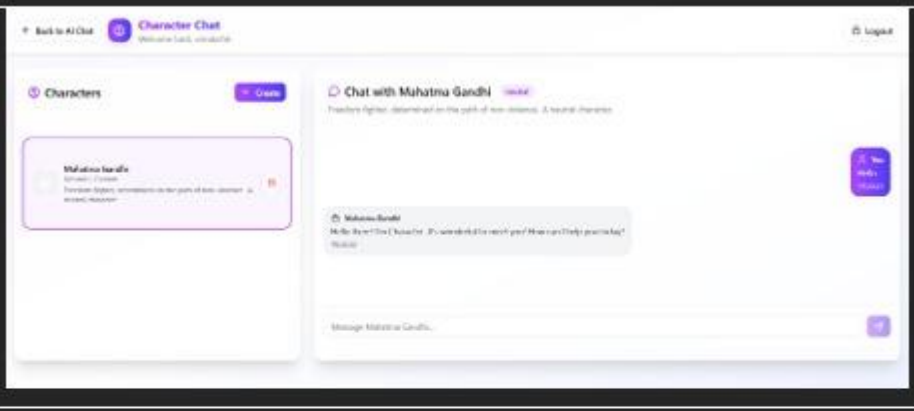
As a new user, I want to summarize website links or YouTube videos easily, so that I can quickly understand long content without confusion or technical difficulty.

#004	AI Content Summarizer
As a new user, I want to summarize website links or YouTube videos easily, so that I can quickly understand long content without confusion or technical difficulty	
	
Success: When the user enters a valid website URL or YouTube link, selects the operation mode, and clicks *Summarize*, the system should show a loading state, process the content, and then display a clear and concise summary. The summarizer should handle both website and YouTube modes as expected. Mock Mode, if enabled, should provide a sample summary without requiring backend connectivity. **Failure:** If the user enters an invalid or missing URL, the system should display an immediate validation message and not begin processing. If the backend API fails, times out, or the YouTube video lacks a transcript in the selected language, the system must stop the loading animation, show an appropriate error message, and allow the user to try again.	

2.5.5 Character Chat (PersonaFlow Tool)

As a user, I want to create and select AI characters so I can chat with customized personas that behave according to my preferences.

#005	Character Chat (PersonaFlow Tool)
As a user, I want to create and select AI characters so I can chat with customized personas that behave according to my preferences.	



Confirmation

1. Success

1.1 When the user clicks the Create button, the system should open the character creation interface so a new persona can be defined.
1.2 When the user selects an existing character from the Characters panel, the system should load that character's details and open the chat interface for that persona.
1.3 When no character is selected, the system should clearly show a placeholder message indicating that no character is selected.
1.4 When the user navigates back to AI Chat, the system should return to the general chat interface without losing any data.

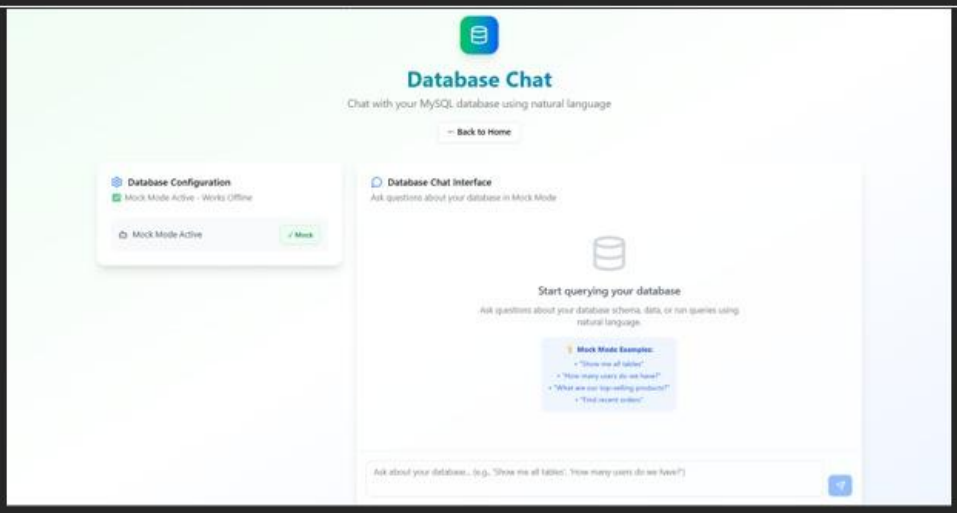
2. Failure

2.1 If character loading fails due to a backend or storage error, the system should show an error message and prevent the user from entering the chat interface.
2.2 If the Create button fails to open the character creation screen, the system should notify the user that character creation could not be initiated.
2.3 If character details cannot be displayed after selection, the system should keep the placeholder message visible and inform the user that the character could not be loaded.
2.4 If logout fails, the system should remain on the current page and show an appropriate error instead of abruptly clearing the session.

2.5.6 Database Chat

As a user, I want to interact with my database using natural language, so that I can retrieve information and understand data without writing SQL queries manually.

#006	Database Chat
As a user, I want to interact with my database using natural language, so that I can retrieve information or understand my data without writing SQL queries manually.	



Success:

When the user opens the Database Chat tool in Mock Mode or Live Mode, the system should allow them to enter a natural-language query and show a loading state while processing. After interpretation, the system should return a clear response based on the database schema or mock data. If Mock Mode is active, all queries should work offline using predefined mock tables, allowing the user to experiment without backend connectivity.

Failure:

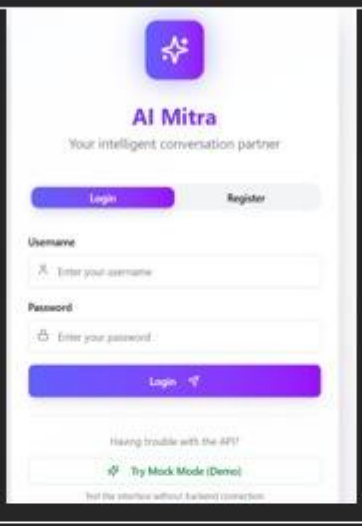
If the user enters an unclear or unsupported question, the system should notify them that the query cannot be processed and suggest valid examples. If backend connectivity fails in Live Mode, the system should stop the loading animation and show a connection error, while still keeping Mock Mode available. In cases where schema information is missing or incomplete, the system should provide guidance instead of failing silently.

2.5.7 User Login Screen

As a returning user, I want to log into my AI Mitra account so I can access my chats, characters, and personalized settings.

#007	User Login Screen

As a returning user, I want to log into my AI Mitra account so I can access my chats, characters, and personalized settings.



Confirmation

1. Success

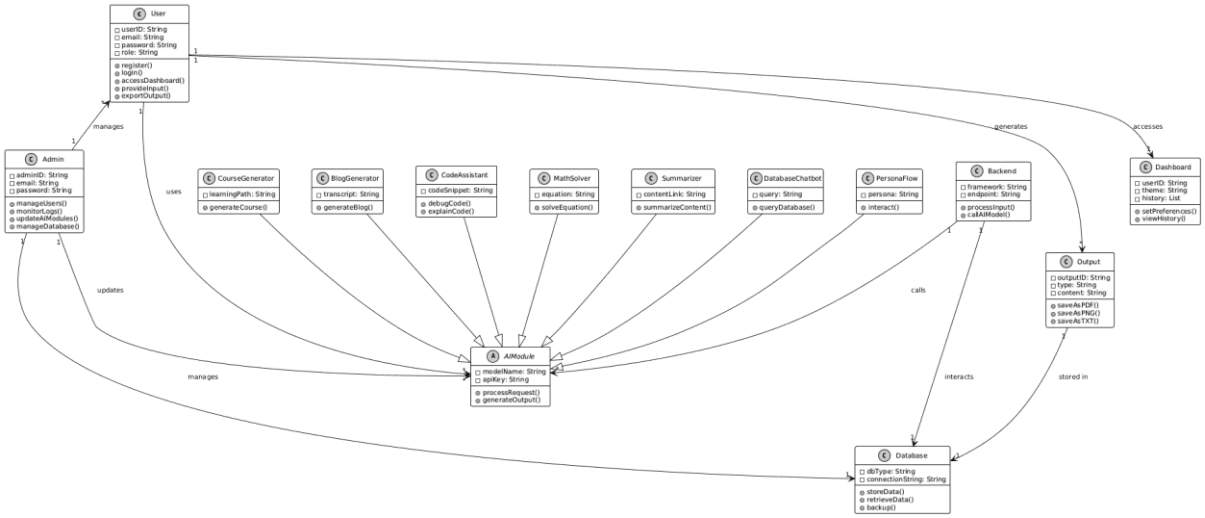
1.1 When the user enters a valid username and password and clicks the Login button, the system should authenticate the credentials and redirect the user to the main chat interface.
1.2 If authentication is successful, the system should load the user's data, including chat history and saved characters.
1.3 When the user selects Mock Mode, the system should allow access to the demo interface without validating credentials or connecting to the backend.

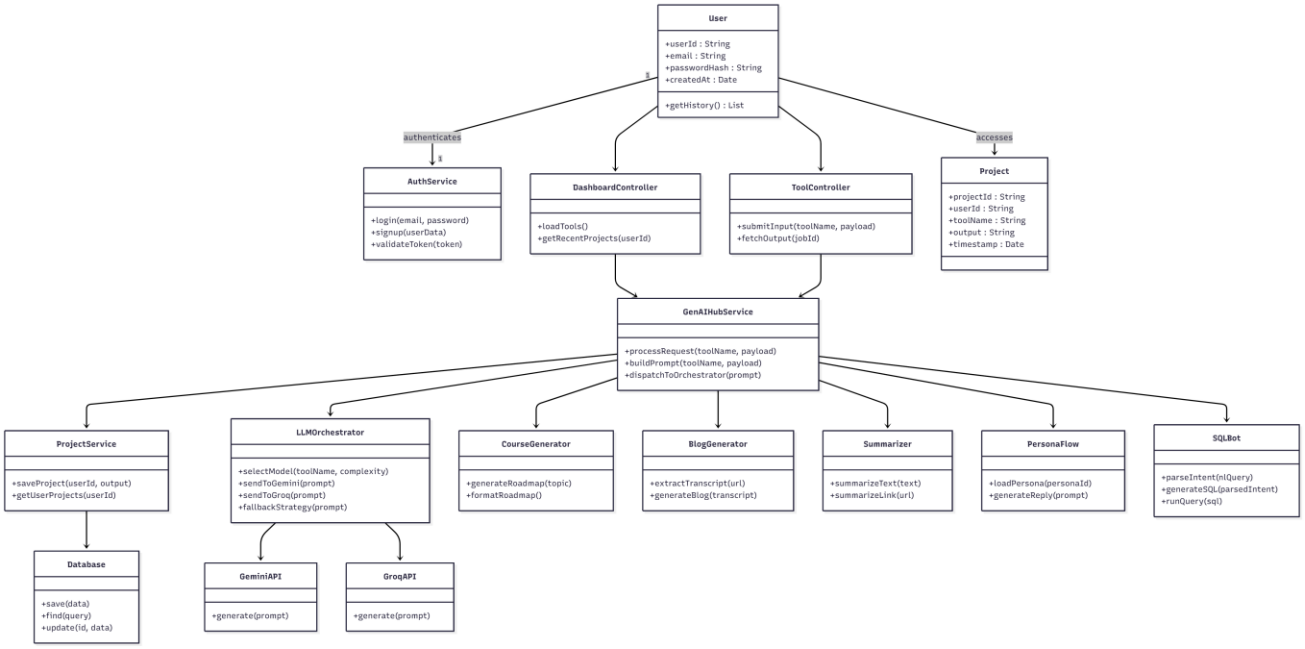
2. Failure

2.1 If the username or password field is left empty, the system should show a message indicating that all fields must be filled before logging in.
2.2 If authentication fails because of incorrect credentials, the system should display an error.
2.3 If the login request fails due to backend issues or API downtime, the system should show an error message..
2.4 If Mock Mode cannot be opened due to internal errors, the system should notify the user that the demo mode is currently unavailable.

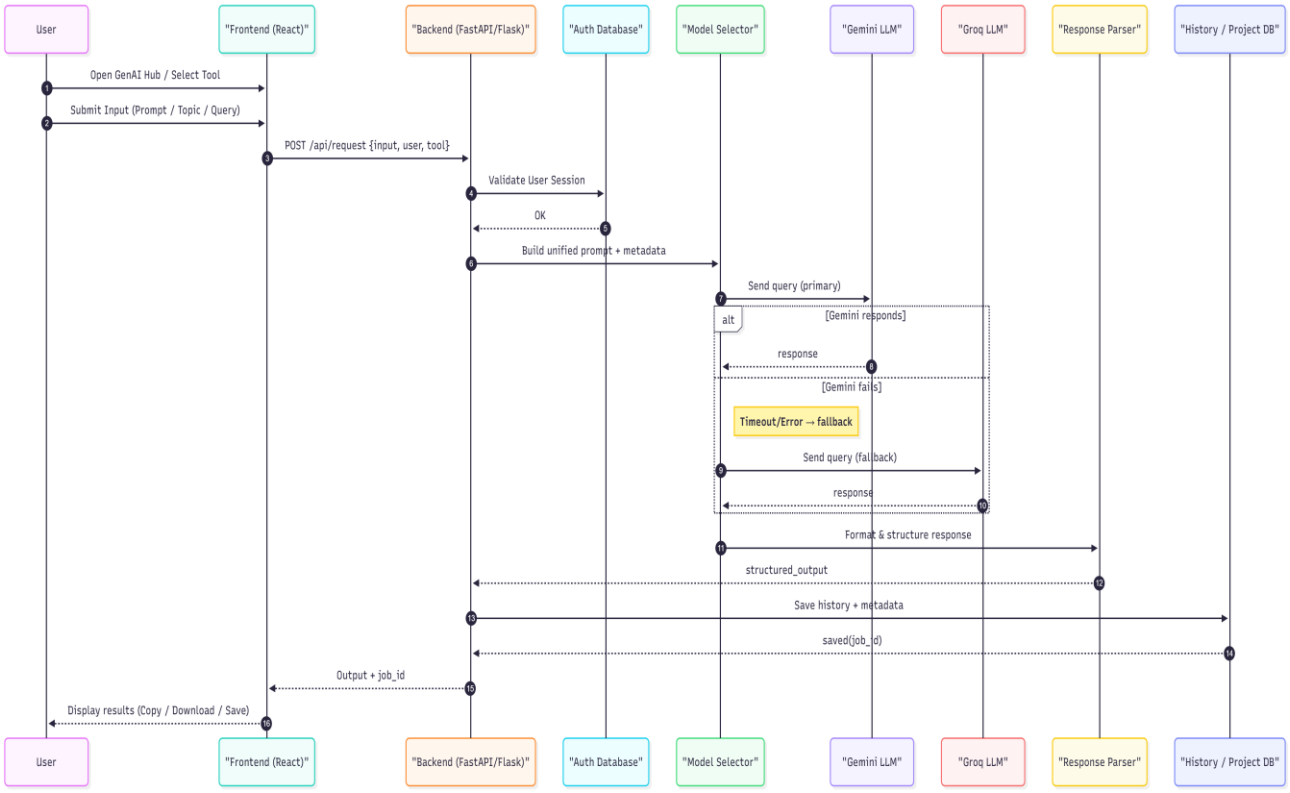
3 Design Phase

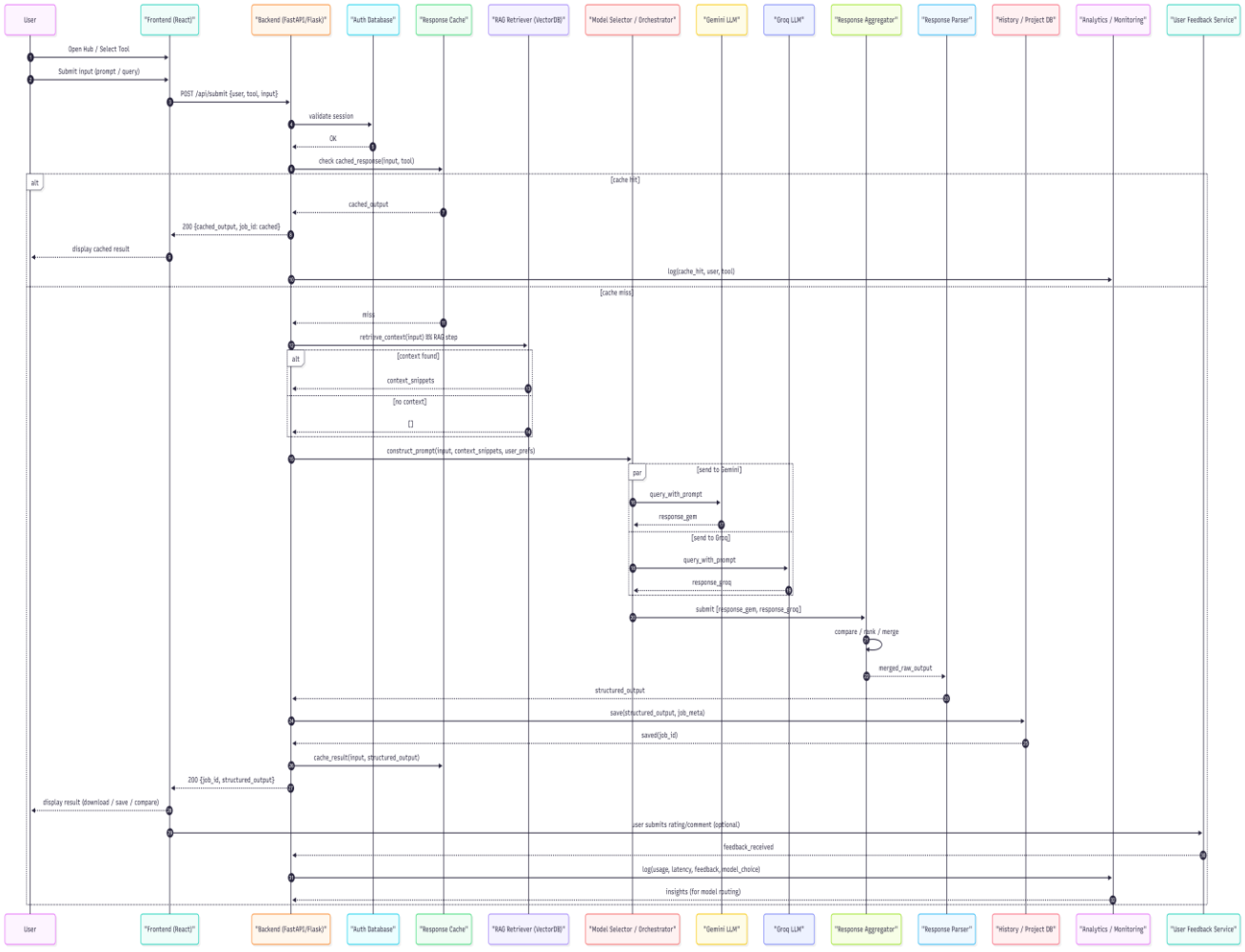
3.1 Class Diagram



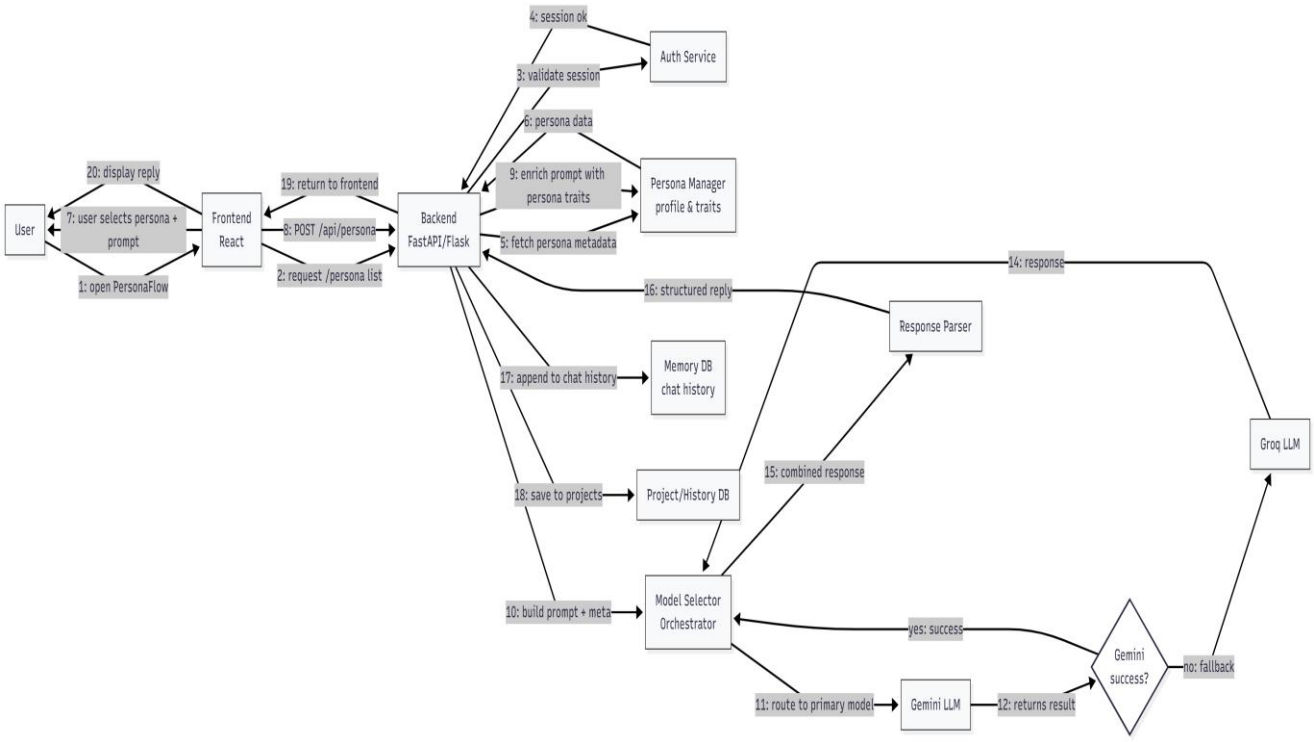


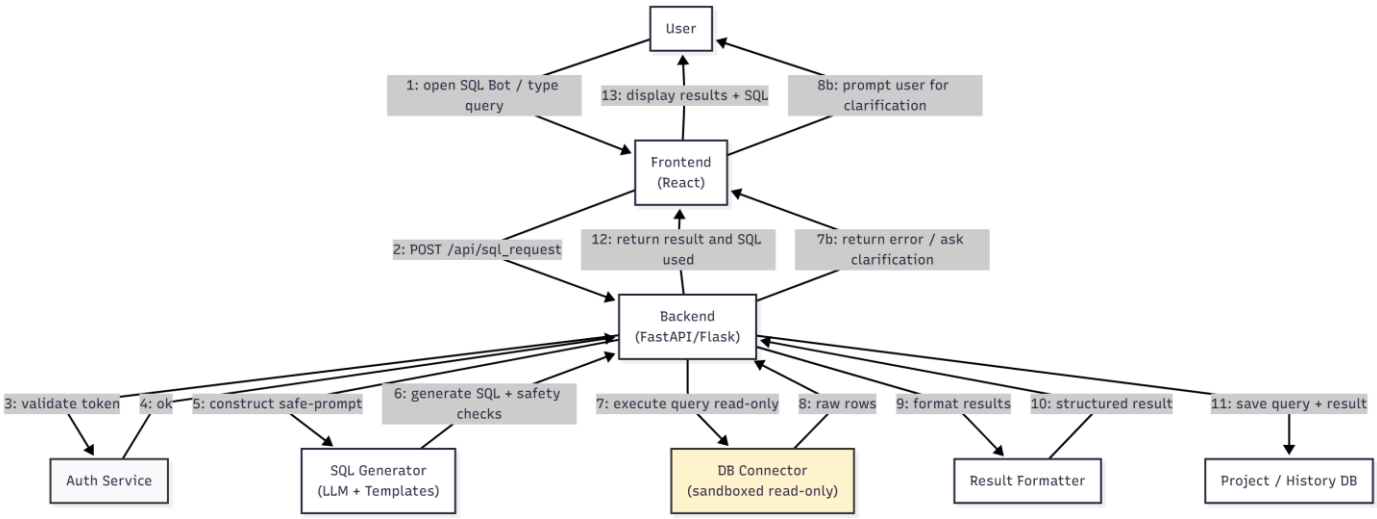
3.2 Sequence Diagram



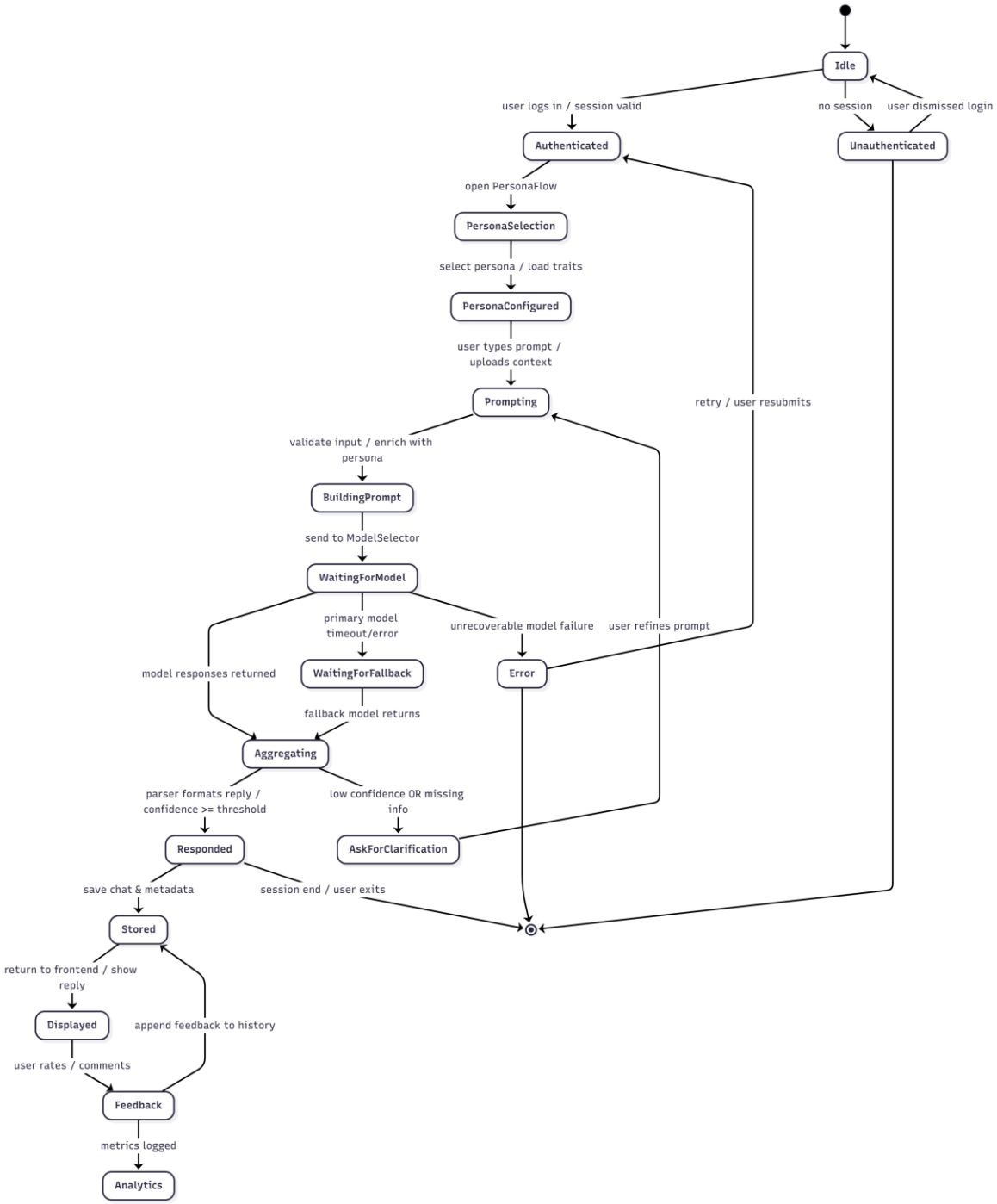

User
"Frontend (React)"
"Backend (FastAPI/Flask)"
"Auth Database"
"Response Cache"
"RAG Retriever (VectorDB)"
"Model Selector / Orchestrator"
"Gemini LLM"
"Groq LLM"
"Response Aggregator"
"Response Parser"
"History / Project DB"
"Analytics / Monitoring"
"User Feedback Service"
Open Hub / Select Tool
Submit input (prompt / query)
POST /api/submit {user, tool, input}
validate session
OK
check cached_response(input, tool)
alt
[cache hit]
cached_output
200 {cached_output, job_id: cached}
display cached result
log(cache_hit, user, tool)
[cache miss]
miss
retrieve_context(input) #6 RAG step
alt
[context found]
context_snippets
[no context]
[]
construct_prompt(input, context_snippets, user_prefs)
par
[send to Gemini]
query_with_prompt
response_gem
[send to Groq]
query_with_prompt
response_groq
submit (response_gem, response_groq)
compare / rank / merge
merged_raw_output
structured_output
save(structured_output, job_meta)
saved(job_id)
cache_result(input, structured_output)
200 {job_id, structured_output}
display result (download / save / compare)
user submits rating/comment (optional)
feedback_received
log(usage, latency, feedback, model_choice)
insights (for model routing)

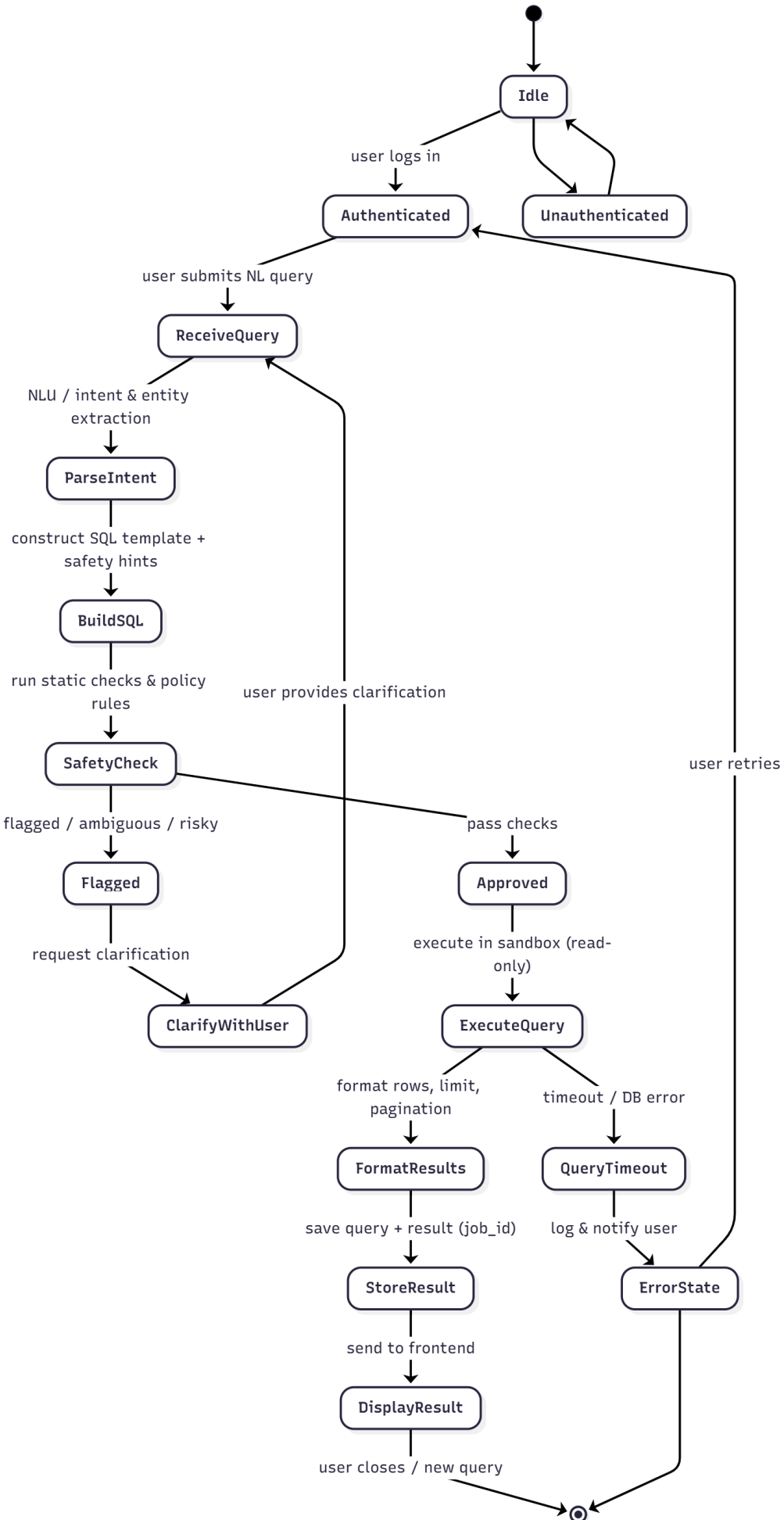
3.3 Collaboration Diagram





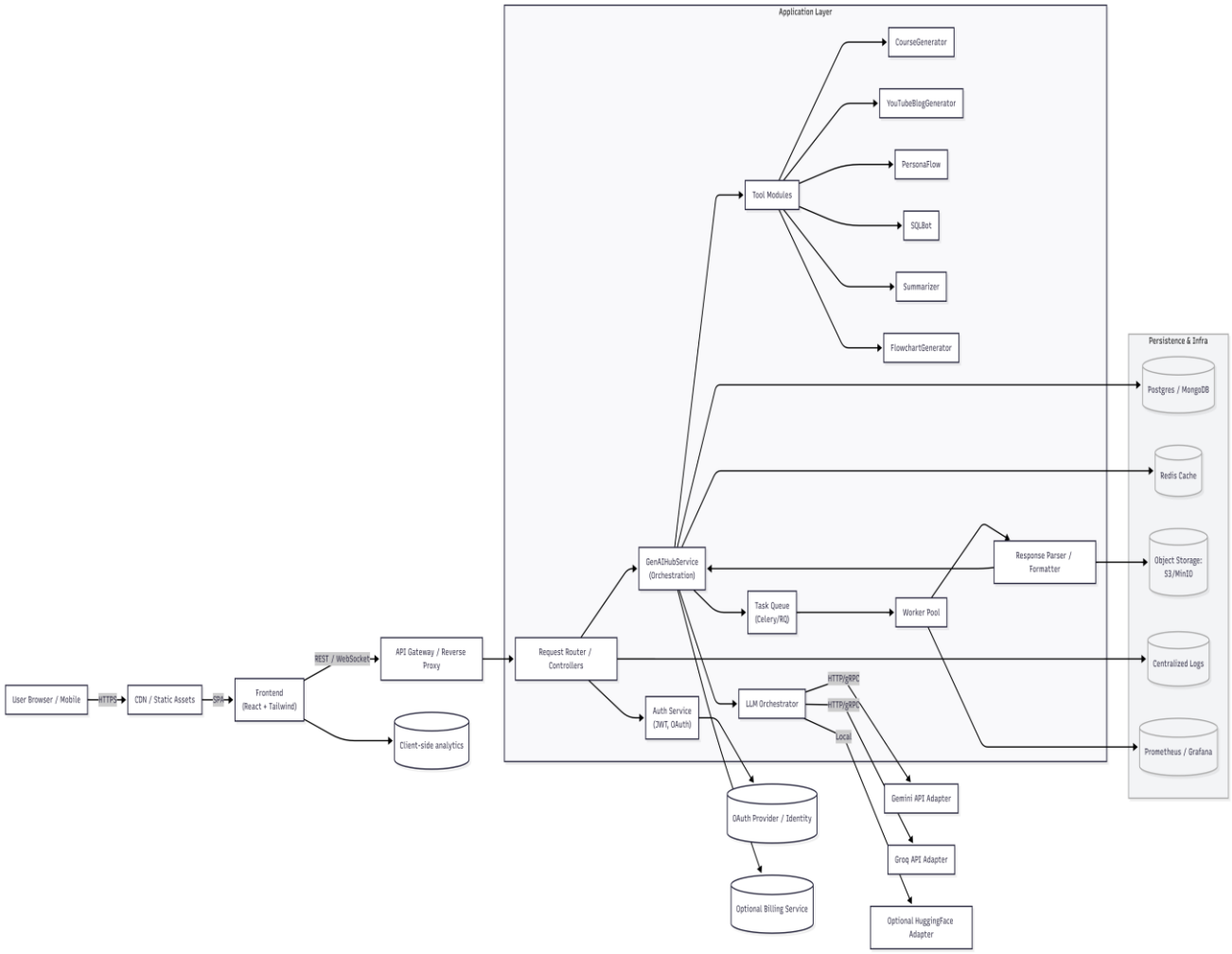
3.4 State Chart Diagrams



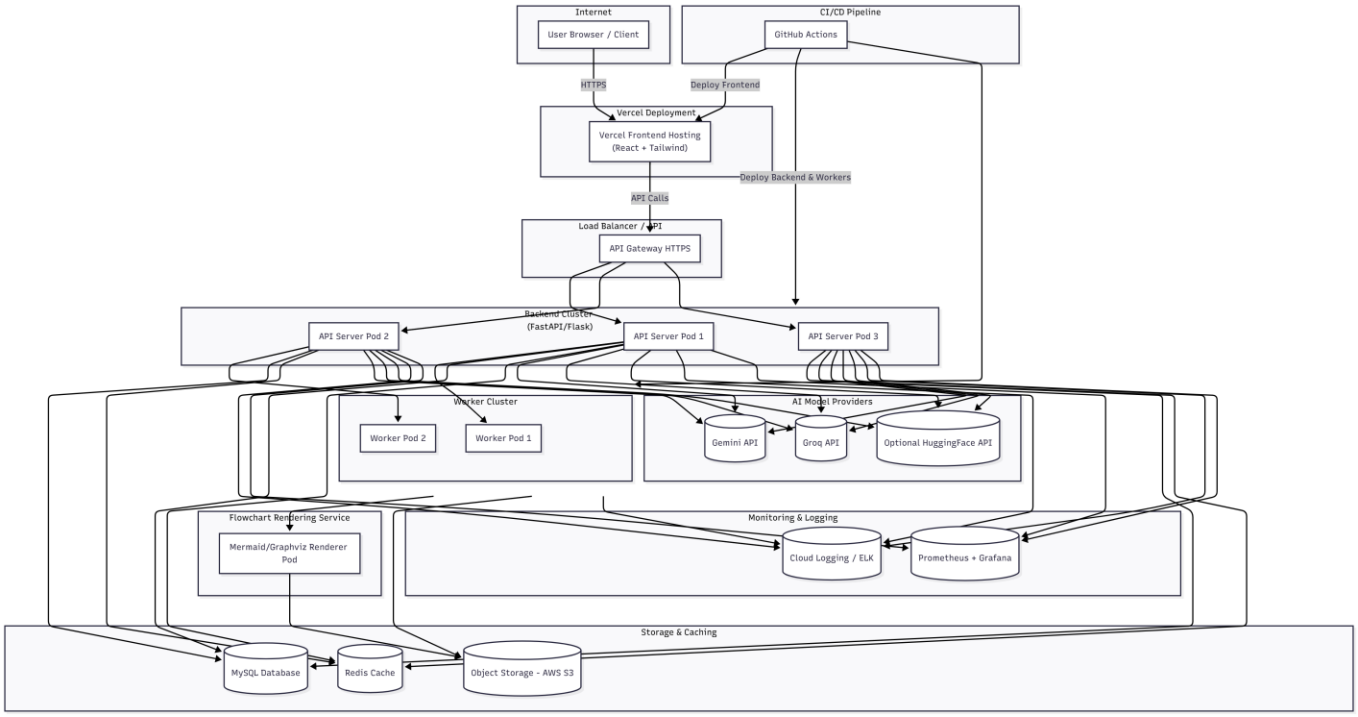

Idle
user logs in
Authenticated
Unauthenticated
user submits NL query
ReceiveQuery
NLU / intent & entity extraction
ParseIntent
construct SQL template + safety hints
BuildSQL
run static checks & policy rules
SafetyCheck
user provides clarification
user retries
flagged / ambiguous / risky
pass checks
Flagged
Approved
request clarification
execute in sandbox (read-only)
ClarifyWithUser
ExecuteQuery
format rows, limit, pagination
timeout / DB error
FormatResults
QueryTimeout
save query + result (job_id)
log & notify user
StoreResult
ErrorState
send to frontend
DisplayResult
user closes / new query

4 Implementation

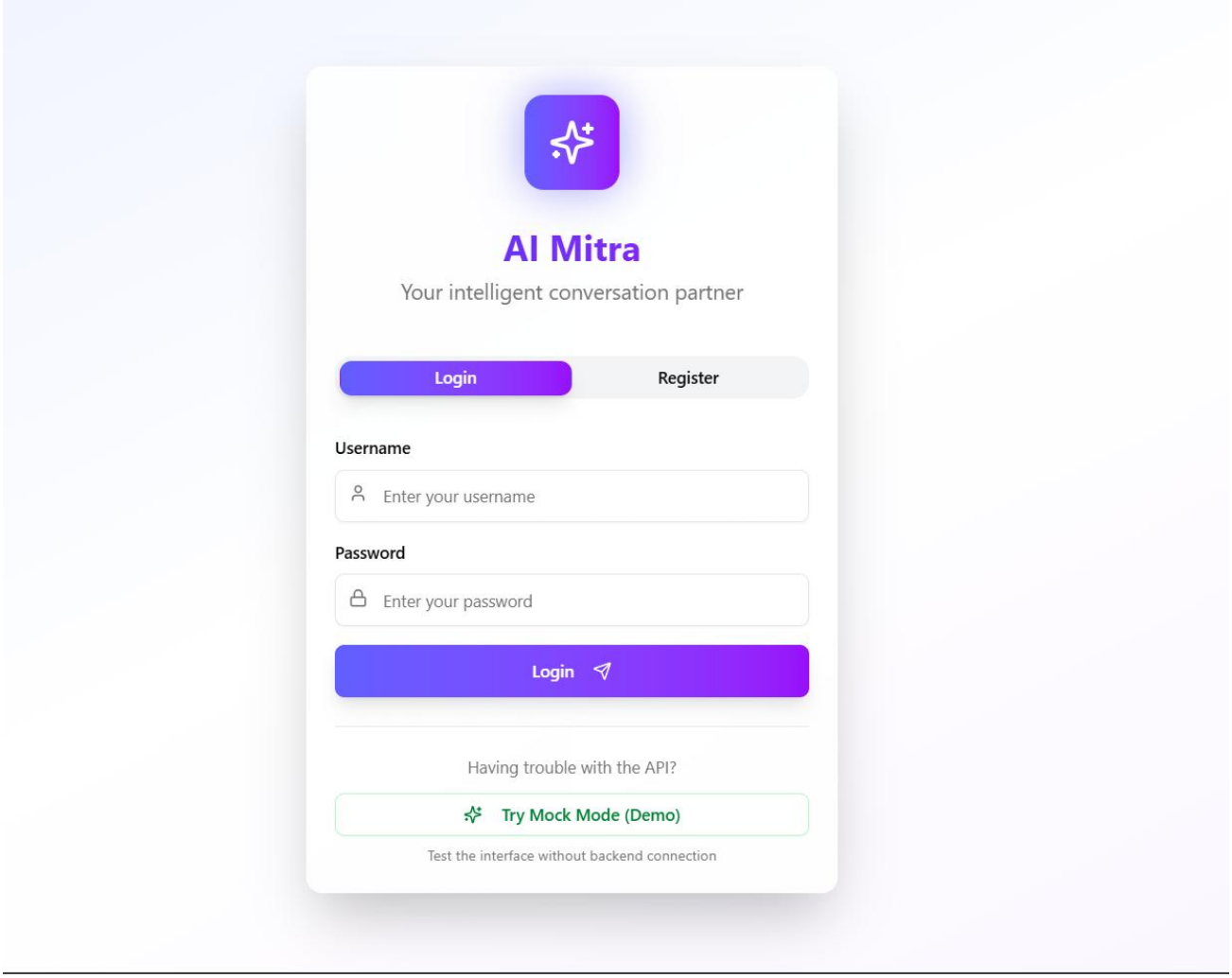
4.1 Component Diagram

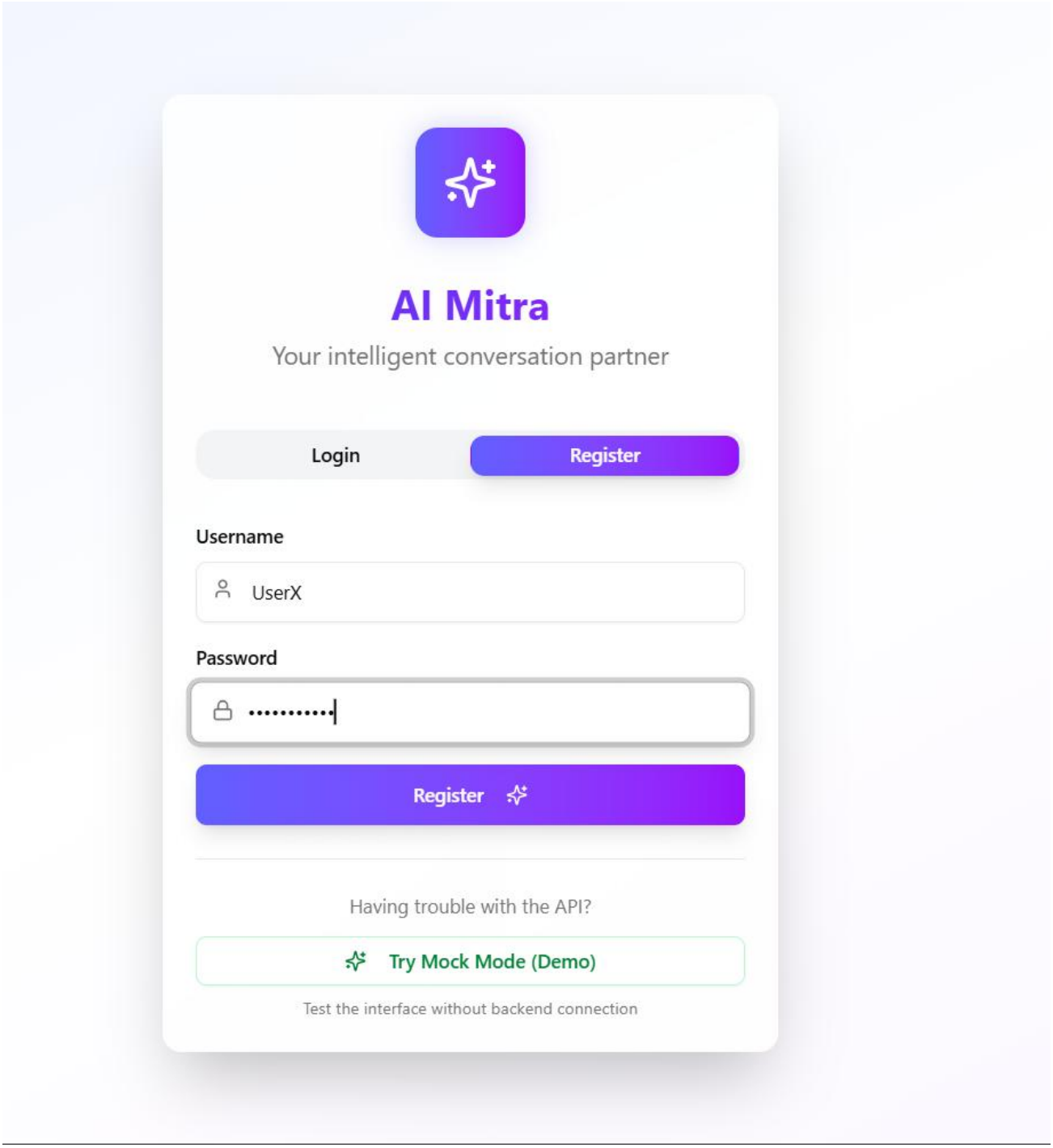


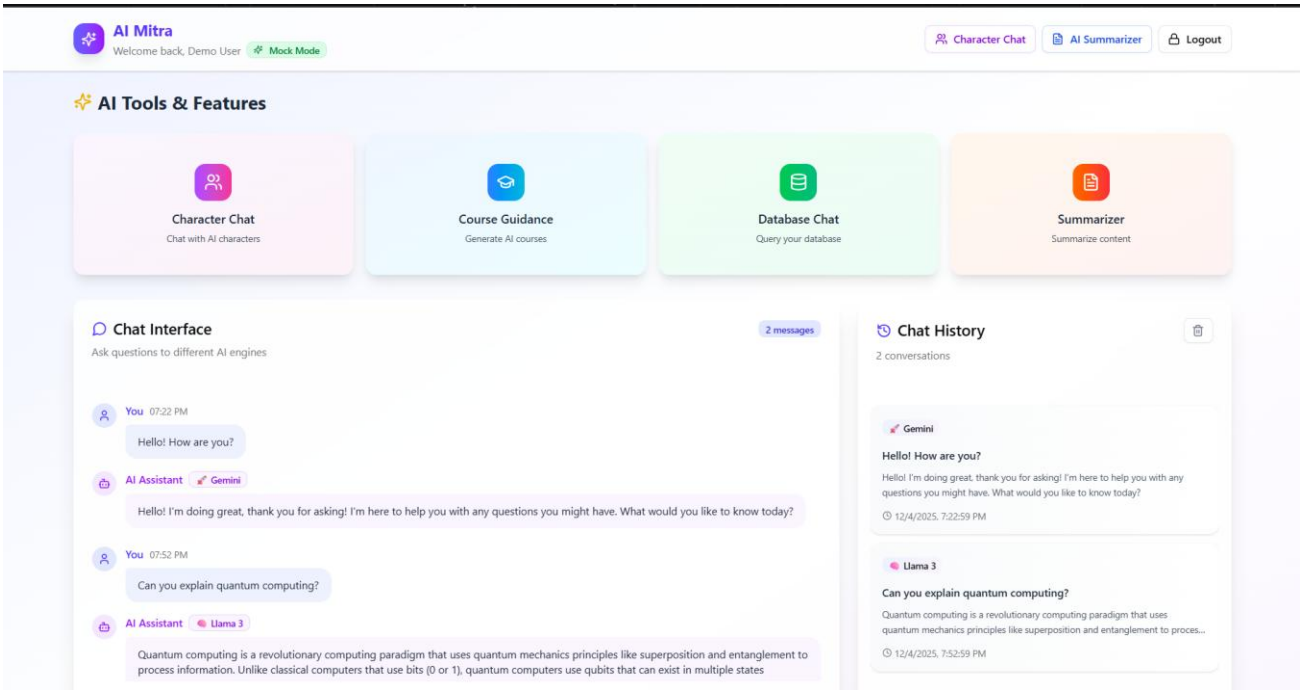
4.2 Deployment Diagram

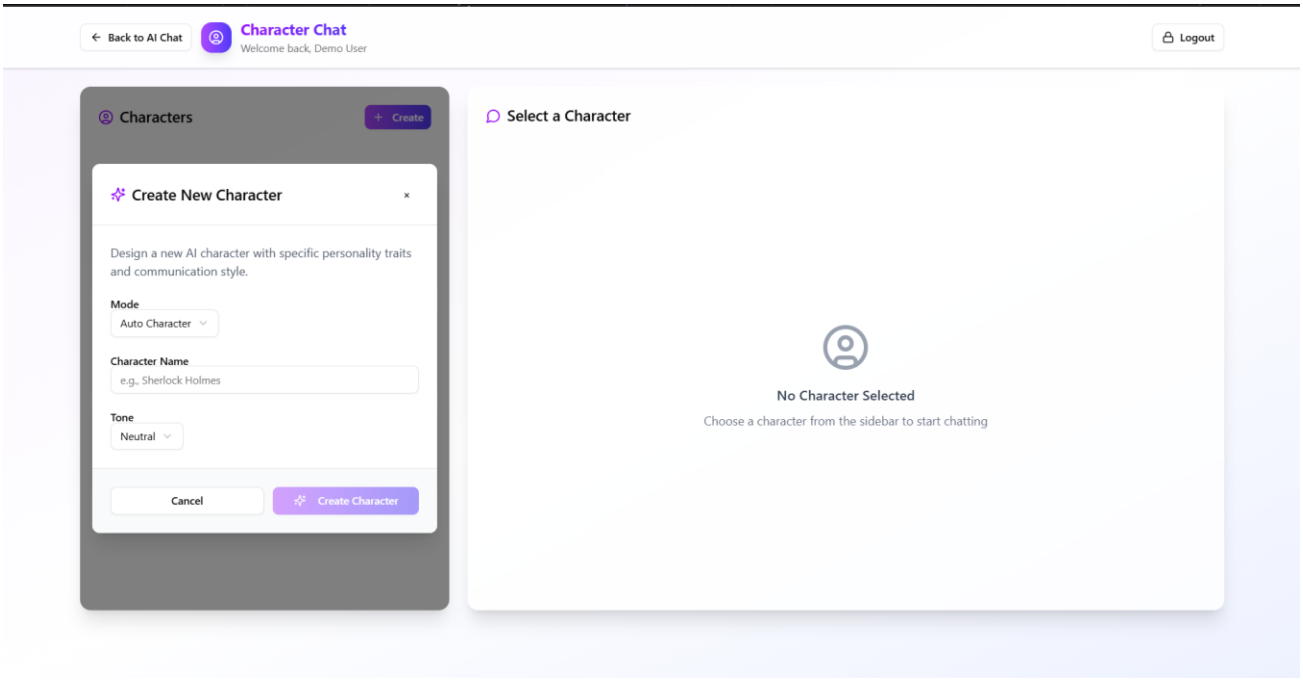


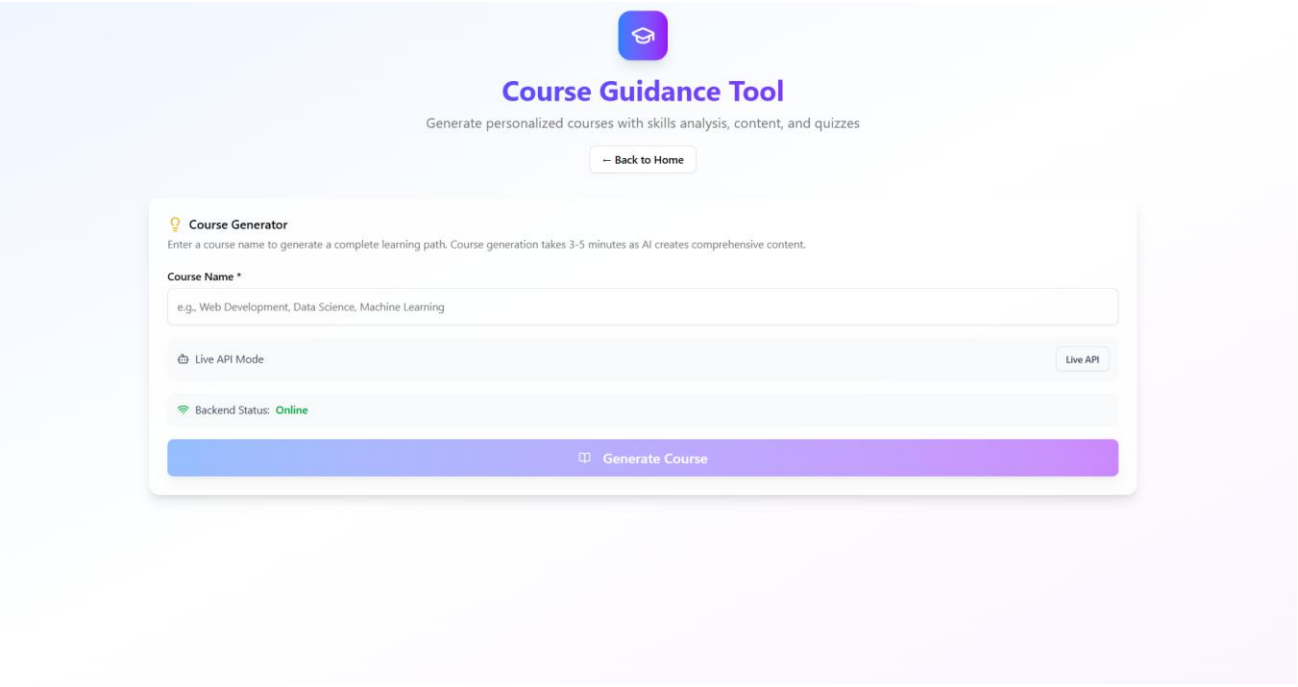
4.3 ScreenShots

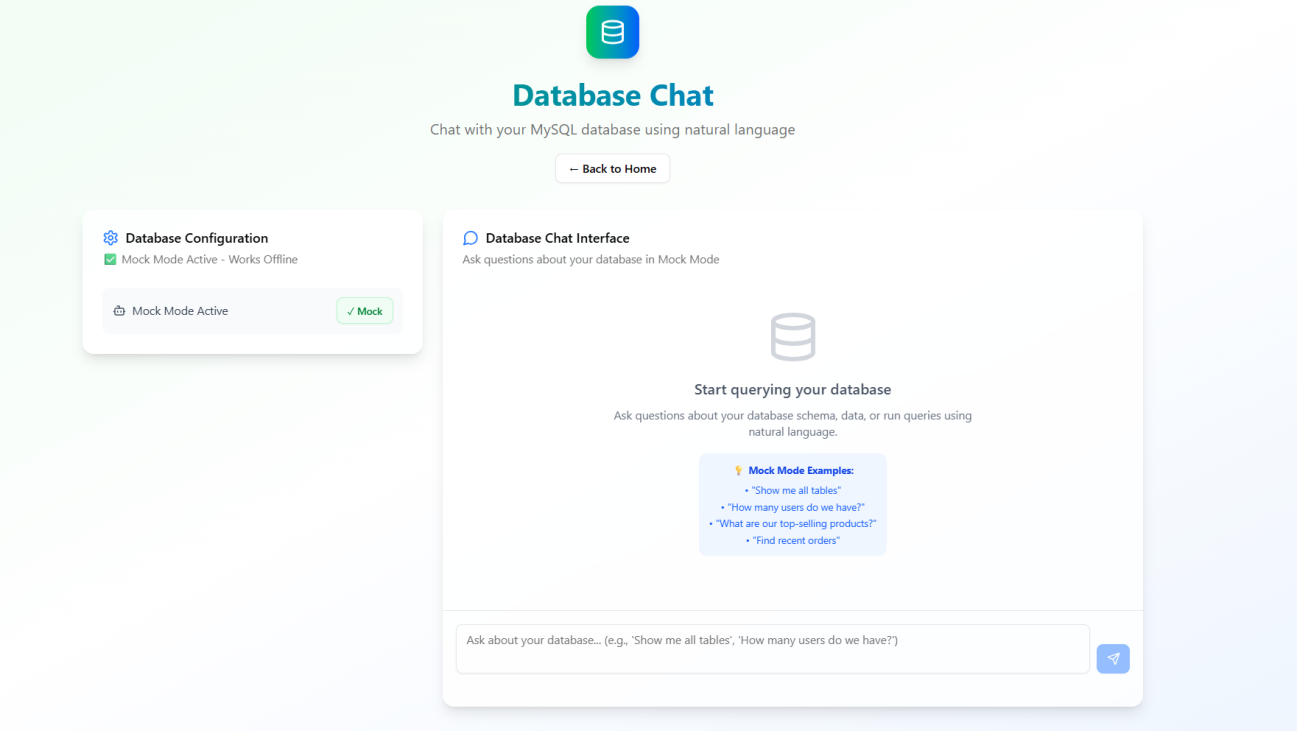


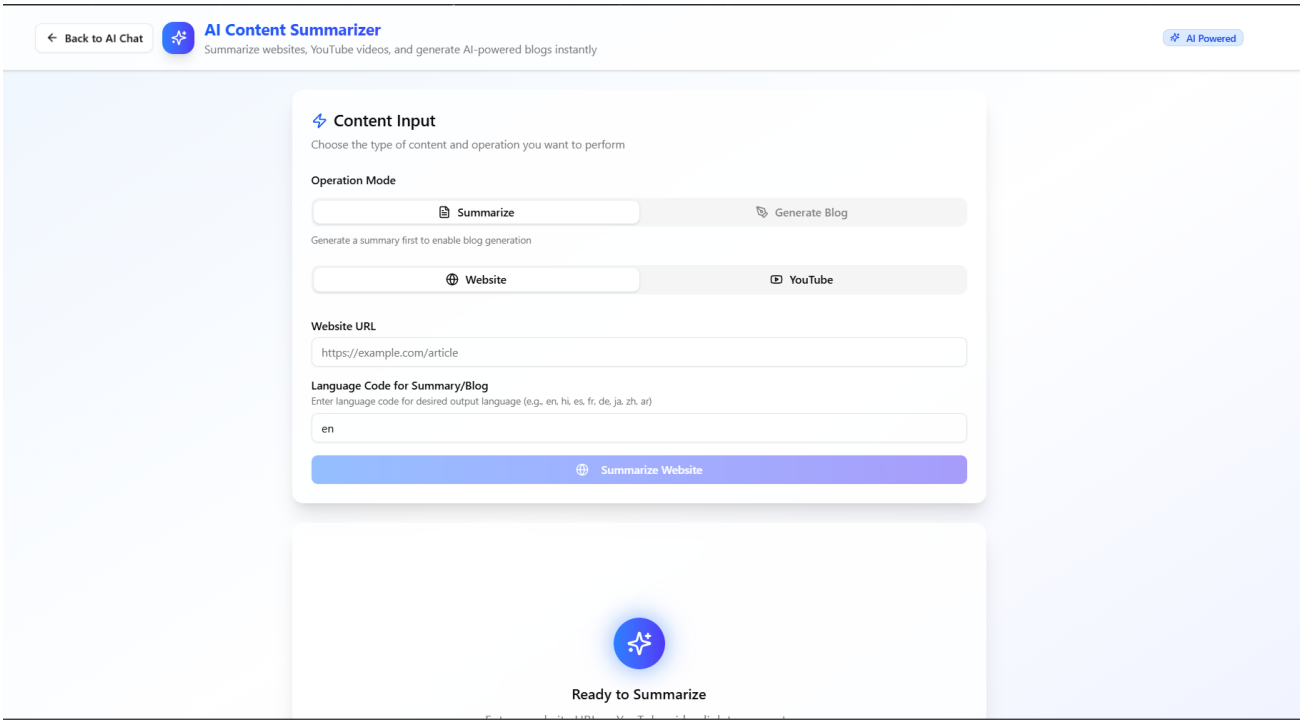

AI Mitra
Your intelligent conversation partner
Login
Register
Username
UserX
Password
Register
Having trouble with the API?
Try Mock Mode (Demo)
Test the interface without backend connection


AI Mitra
Welcome back, Demo User
Mock Mode
Character Chat
AI Summarizer
Logout
AI Tools & Features
Character Chat
Chat with AI characters
Course Guidance
Generate AI courses
Database Chat
Query your database
Summarizer
Summarize content
Chat Interface
Ask questions to different AI engines
2 messages
You 07:22 PM
Hello! How are you?
AI Assistant
Gemini
Hello! I'm doing great, thank you for asking! I'm here to help you with any questions you might have. What would you like to know today?
You 07:52 PM
Can you explain quantum computing?
AI Assistant
Llama 3
Quantum computing is a revolutionary computing paradigm that uses quantum mechanics principles like superposition and entanglement to process information. Unlike classical computers that use bits (0 or 1), quantum computers use qubits that can exist in multiple states
Chat History
2 conversations
Gemini
Hello! How are you?
Hello! I'm doing great, thank you for asking! I'm here to help you with any questions you might have. What would you like to know today?
12/4/2025, 7:22:59 PM
Llama 3
Can you explain quantum computing?
Quantum computing is a revolutionary computing paradigm that uses quantum mechanics principles like superposition and entanglement to proces...
12/4/2025, 7:52:59 PM


Back to AI Chat
Character Chat
Welcome back, Demo User
Logout
Characters
Create
Create New Character
Design a new AI character with specific personality traits and communication style.
Mode
Auto Character
Character Name
e.g., Sherlock Holmes
Tone
Neutral
Cancel
Create Character
Select a Character
No Character Selected
Choose a character from the sidebar to start chatting


Course Guidance Tool
Generate personalized courses with skills analysis, content, and quizzes
← Back to Home
Course Generator
Enter a course name to generate a complete learning path. Course generation takes 3-5 minutes as AI creates comprehensive content.
Course Name *
e.g., Web Development, Data Science, Machine Learning
Live API Mode
Live API
Backend Status: Online
Generate Course


Database Chat
Chat with your MySQL database using natural language
← Back to Home
Database Configuration
Mock Mode Active - Works Offline
Mock Mode Active
✓ Mock
Database Chat Interface
Ask questions about your database in Mock Mode
Start querying your database
Ask questions about your database schema, data, or run queries using natural language.
Mock Mode Examples:
• "Show me all tables"
• "How many users do we have?"
• "What are our top-selling products?"
• "Find recent orders"
Ask about your database... (e.g., 'Show me all tables', 'How many users do we have?')


Back to AI Chat
AI Content Summarizer
Summarize websites, YouTube videos, and generate AI-powered blogs instantly
AI Powered
Content Input
Choose the type of content and operation you want to perform
Operation Mode
Summarize
Generate Blog
Generate a summary first to enable blog generation
Website
YouTube
Website URL
https://example.com/article
Language Code for Summary/Blog
Enter language code for desired output language (e.g., en, hi, es, fr, de, ja, zh, ar)
en
Summarize Website
Ready to Summarize

5.0 Testing

5.1 Test Plan

A manual black box testing approach will be used to verify whether all functional and non-functional specifications in the system are met.

The following attributes define the scope of GenAI Hub's testing:

A. Authentication & Access Rules

a. Login Access

i. User must log in using a valid username and password.
ii. Password must meet security constraints .
iii. Incorrect credentials return an appropriate error.

b. Chatbot & Model Access

i. AI Chatbot is accessible only after successful login.
ii. User can switch between Gemini and Groq models manually.
iii. Chat history must persist per session and be retrievable from Dashboard.

c. Database Module Access

i. User must enter:

- DB Host
- Port
- Username
- Password
- SQL Query

ii. Invalid credentials must return connection error.
iii. Only SELECT (read-only) queries are allowed; destructive commands should be blocked.

B. Persona Module

a. Persona Access

i. User can view Persona List after login.
ii. User may choose:

- Auto Mode (persona decides style/tone)
- Custom Mode (user selects tone: friendly, formal, emotional, etc.)

iii. Persona chat history must be saved separately from general chat.

b. Persona Behavior

i. Persona must maintain defined traits consistently.
ii. Summary of conversation must be generated when requested.

C. Summarizer + Blog Generator

a. YouTube/Website Summarizer

i. Accepts:

- YouTube URL
- Website URL
- Long text

ii. If transcript unavailable → error displayed.

b. Blog Generator

i. Converts transcript to a formatted blog.
ii. Must produce readable HTML/text output.

D. Flowchart Generator

i. User inputs steps or process text.
ii. System converts text → Mermaid/HTML Flowchart.
iii. Output must be downloadable (SVG/PNG).
iv. Malformed text must prompt refinement.

E. Course Guidance Generator

i. User inputs learning topic.
ii. System generates:

- Step-by-step roadmap
- Resources
- Optional downloadable template

iii. Vague topics require clarification.

5.2 Test Cases

Test Case #01 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Invalid Login Attempt **Subsystem:** Authentication **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025
Short Description: Login should fail when incorrect password is used.	
Pre-conditions	
User exists; active internet connection.	

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open Login page	Login form displayed		
2	Enter username + wrong password	Input accepted		
3	Click Login	Error: "Invalid Credentials"		
Post-conditions				
User remains unauthenticated.				

Test Case #02 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Successful Login **Subsystem:** Authentication **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate successful login using correct credentials.

Pre-conditions

Valid username and password exist.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Enter valid credentials	Credentials accepted		
2	Click Login	Redirect to Dashboard		

Post-conditions

User is logged in and redirected.

Test Case #03 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Model Switching **Subsystem:** AI Chatbot **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Ensure chatbot switches between Gemini and Groq.

Pre-conditions

User logged in; chatbot accessible.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open chatbot	Chat interface loads		
2	Select "Groq"	Groq highlighted as active		
3	Enter prompt	Input accepted		
4	Submit prompt	Response generated using Groq		

Post-conditions

Chat entry stored in chat history.

Test Case #04 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Chat History Availability **Subsystem:** AI Chatbot **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025
Short Description: Ensure previous messages load correctly.	
Pre-conditions	
User has past chat history.	

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open Chatbot	Previous messages displayed		
2	Scroll chat window	Older messages load		

Post-conditions
History persists across sessions.

Test Case #05 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Invalid Database Connection **Subsystem:** Database Tool **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Ensure DB rejects invalid credentials.

Pre-conditions

User logged in; incorrect DB details available.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open DB Tool	Connection form displayed		
2	Enter invalid host/port/username	Input accepted		
3	Click Connect	Message: "Connection Failed"		

Post-conditions

No connection established.

Test Case #06 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Database Query Execution **Subsystem:** Database Tool **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate SELECT query execution.

Pre-conditions

Valid DB credentials.

Step	Action	Expected System Response	Pass/Fail	Commen
1	Enter correct DB credentials	Successful connection		
2	Enter SELECT query	Input accepted		
3	Click Execute	Results displayed in table format		

Post-conditions

Query saved to history.

Test Case #07 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Persona Auto Mode **Subsystem:** Persona Module **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate persona selection and auto mode reply.

Pre-conditions

User logged in; persona list available.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open Persona Module	Persona list displayed		
2	Select persona	Persona profile loads		
3	Choose Auto Mode	Tone auto-selected		
4	Send message	Persona-style response returned		

Post-conditions

Chat stored under persona's history.

Test Case #08 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Summarizer Output **Subsystem:** Summarizer **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate summarizer output for valid URLs.

Pre-conditions

Valid YouTube or Website URL.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Open Summarizer	Module loads		
2	Enter valid URL	Input accepted		
3	Click Summarize	Summary generated		

Post-conditions

Summary downloadable.

Test Case #09 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Flowchart Generation **Subsystem:** Flowchart Tool **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate flowchart creation from text.

Pre-conditions

Flowchart tool accessible.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Enter flow/process text	Accepted		
2	Click Generate	HTML/Mermaid diagram displayed		
3	Click Download	File saved (SVG/HTML)		

Post-conditions

Diagram added to Flowchart History.

Test Case #10 **System:** GenAI Hub **Designed by:** Team GenAI Hub **Executed by:** Team GenAI Hub	**Test Case Name:** Course Guidance Generation **Subsystem:** Course Guidance **Design Date:** 03/12/2025 **Execution Date:** 03/12/2025

Short Description: Validate study roadmap generation.

Pre-conditions

User logged in.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Enter topic	Accepted		
2	Click Generate	Roadmap displayed		
3	Save roadmap	Added to history		

Post-conditions

Roadmap accessible from user dashboard.

5.3 Test Reports

GenAI Hub - Test Results Summary

Test Case ID	Description	Result
TC_01	Login must fail when incorrect password is used.	PASS
TC_02	Login must succeed when valid credentials are entered.	PASS
TC_03	User must be able to switch between Gemini and Groq models.	PASS
TC_04	Chat history must load consistently when reopening the chatbot.	PASS
TC_05	Database connection must fail when invalid credentials are provided.	PASS
TC_06	Database SELECT queries must execute successfully with valid credentials.	PASS
TC_07	Persona module must load persona list and generate persona-style responses.	PASS
TC_08	Summarizer must generate summary for valid YouTube/website URLs.	PASS
TC_09	Flowchart generator must convert text into a valid diagram (HTML/Mermaid).	PASS
TC_10	Course guidance generator must produce a roadmap for a valid topic.	PASS

Project Links :

Live Website : https://my-ai-mitra.vercel.app/

Github Repository : https://github.com/SachinGoyal94/NexaGen