CHAPTER 5: SYSTEM DESIGN

5.1   SYSTEM DESIGN AND METHODOLOGY

System design and methodology define the architectural framework and processes used to develop and implement a software solution. For an advanced software consulting and development company like TenUp Software, system design involves selecting the right technologies, frameworks, and methodologies to ensure scalability, security, and efficiency. The methodology adopted influences the development lifecycle, from planning and design to deployment and maintenance.

1. Requirement Gathering & Analysis – Understanding user needs and compliance requirements.
2. Design & Architecture – Structuring the system for scalability and security.
3. Development & Implementation – Writing and testing code for various modules.
4. Testing & Quality Assurance – Ensuring software reliability and regulatory adherence.
5. Deployment & Maintenance – Deploying updates and ensuring continuous improvements.

**5.2** **Database Design for Chatbot API Platform**

Our Chatbot API Platform utilizes PostgreSQL as its primary database management system, leveraging a relational model for structured data storage, efficient retrieval, and scalable performance. The database schema is designed to handle user data, chatbot knowledge bases, API configurations, and interaction logs effectively.

**Key Tables in the Database Schema:**

* **User Table**

Stores user credentials, authentication details, and role-based access permissions.

* **Knowledge Base Table**

Maintains user-provided prompts, documents, or structured data to train individual chatbot instances.

* **Chatbot Configuration Table**

Stores settings related to each chatbot instance, including model type, response behavior, and API settings.

* **Interaction Logs Table**

Captures conversations between end-users and the chatbot for monitoring, improvement, and analytics.

* **API Deployment Table**

Stores API endpoints generated for users to integrate chatbots into their websites or applications.

5.3 INPUT / OUTPUT INTERFACE DESIGN

5.3.1 Input Design

###### Input design ensures that user-provided data is accurately captured and processed efficiently within the Chatbot API Platform. The system takes various inputs to generate a customized chatbot API based on user requirements.

###### User Authentication: Secure login and signup using email and password, with optional multi-factor authentication (MFA) for enhanced security.

###### Knowledge Base Upload: Users provide chatbot training data via plain text, PDFs, or structured documents. The system processes and stores this data in a vector database.

###### Prompt-Based Training: Users input specific prompts to define chatbot responses and behavior, allowing customization of responses.

###### API Configuration: Users specify chatbot settings, including response format, tone, and integration preferences for seamless deployment.

###### Query Processing: End-users interact with the chatbot via API calls, sending text-based queries for real-time response generation.

###### **5.3.2 Output Design**

###### The output design defines how processed data is presented to users through interfaces and APIs, ensuring clarity, usability, and integration flexibility.

###### Admin Dashboard: Displays chatbot performance analytics, API usage statistics, and error logs for monitoring.

###### Generated Chatbot API: A custom API endpoint is provided for users to integrate into their websites or applications.

###### Response Logs & Reports: Users can access chat history, response logs, and system-generated reports in downloadable formats (CSV, JSON).

###### Real-time Chat Interface: A test environment where users can interact with their chatbot before deploying it to their platforms.

###### Error Handling & Debugging Tools: Users receive real-time feedback and debugging options if their chatbot does not perform as expected.

###### **5.3.3 UI Screenshots**

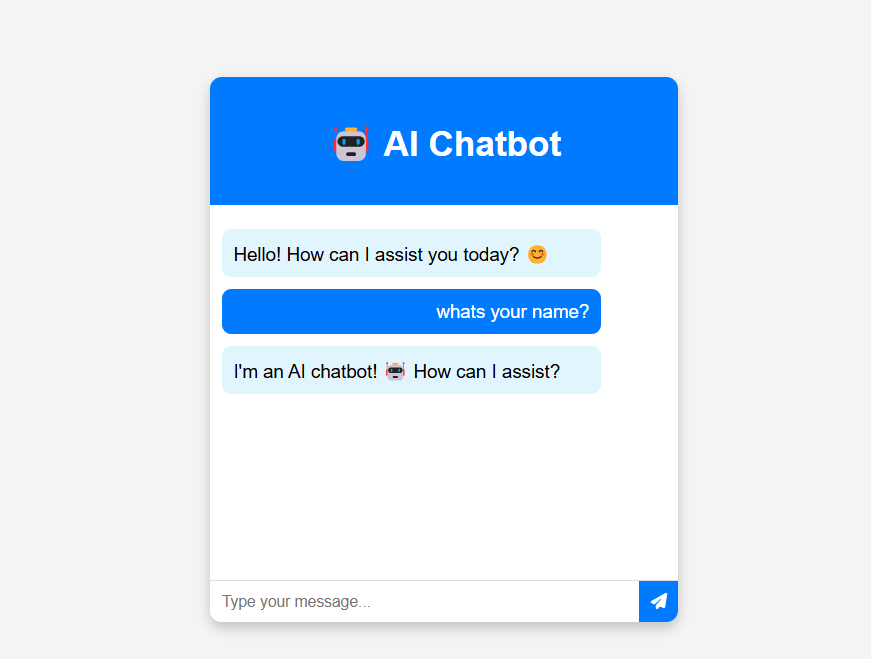
###### Below are the relevant screenshots of ongoing project.

###### 

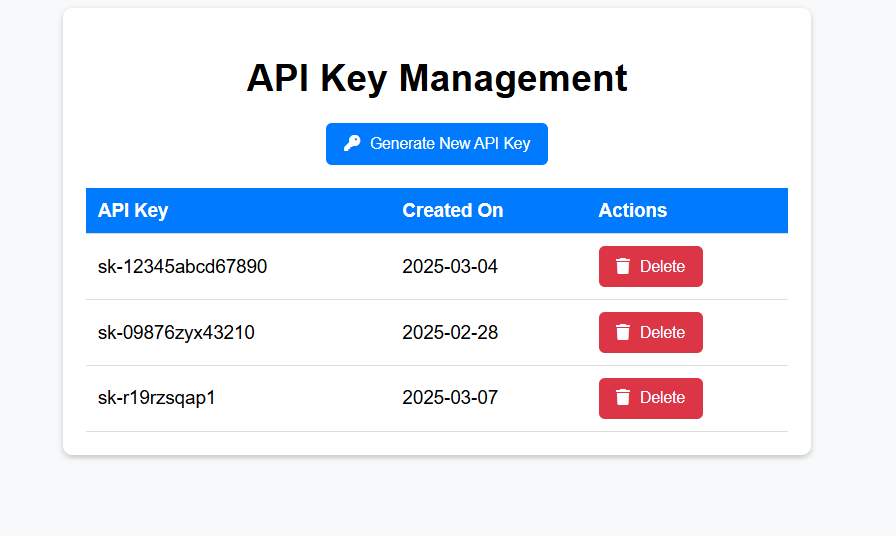
###### **Fig. 5.3.1 Login/Signup page**

###### 

**Fig. 5.3.2** **Users Dashboard**

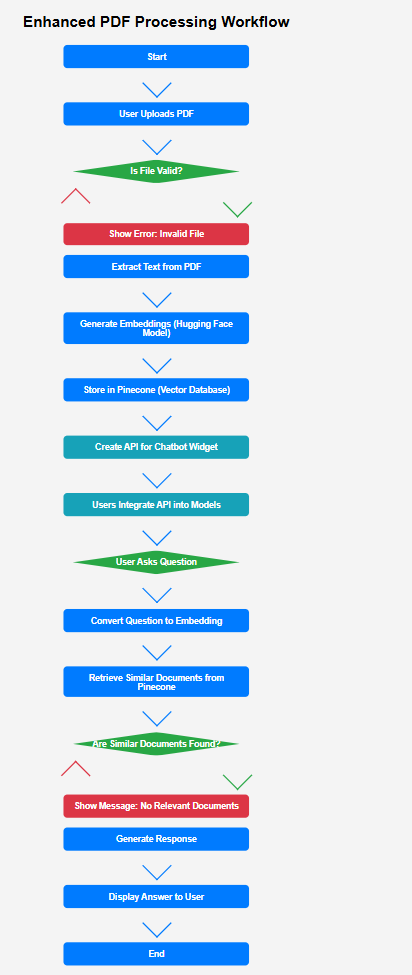


**Fig. 5.3.4 Chat Based on uploaded file**

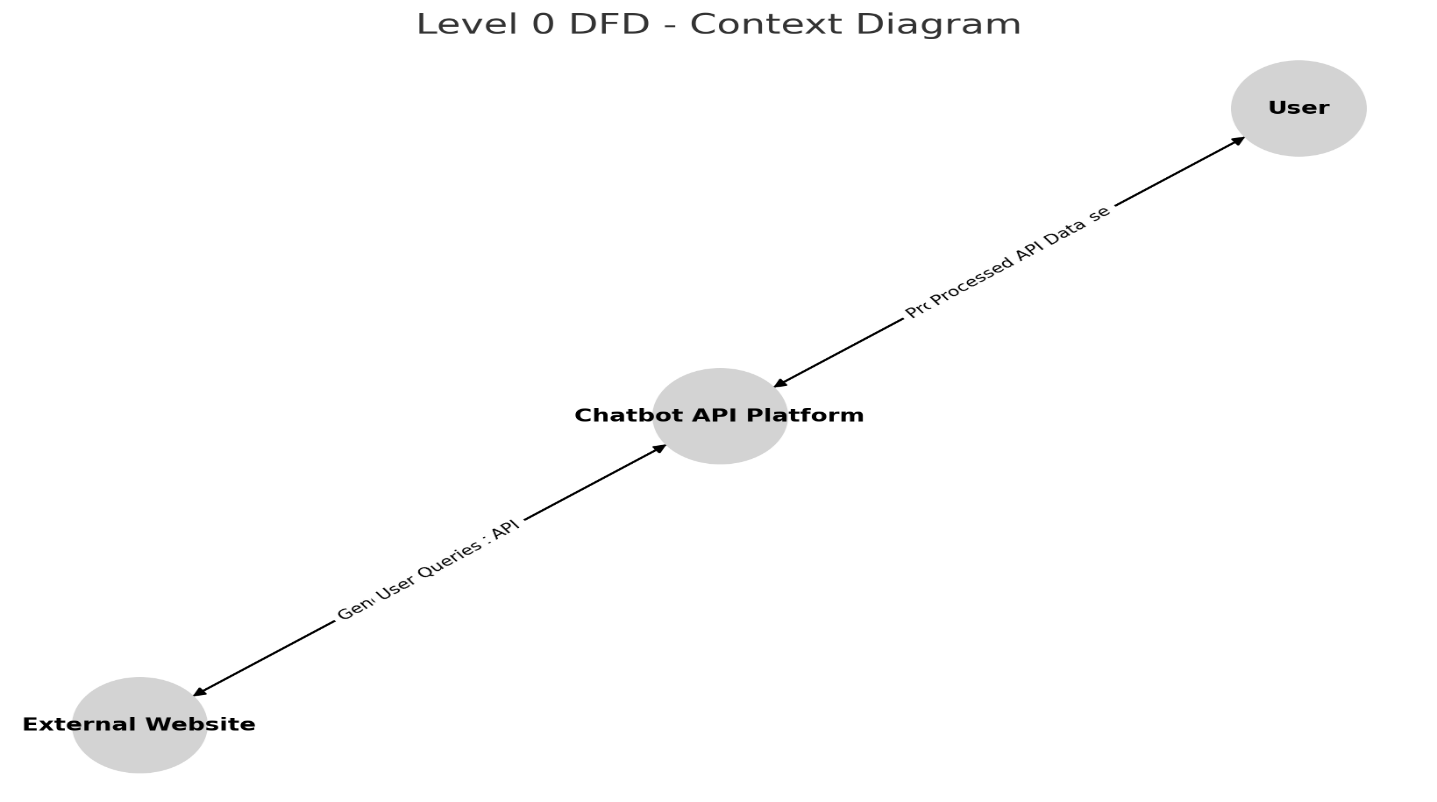
****

**Fig. 5.3.5 API key Management**

**Fig.5.3.6 workflow Diagram**



5.4.2 DFD (Data Flow Diagram) Representation

1. Level 0 DFD

**Fig 5.1 DFD Level 0**

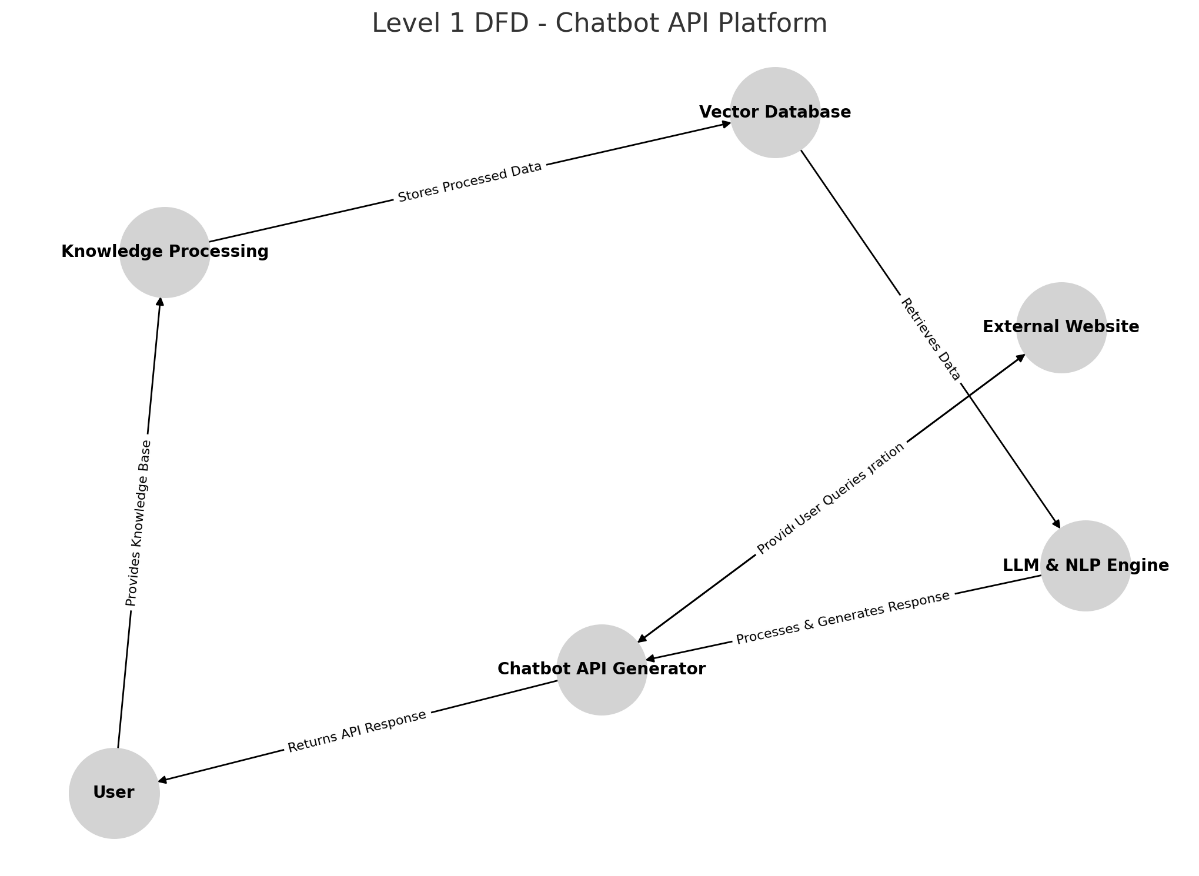
**Fig.5.3.8 DFD level 1**

Fig 5.2 Process Flow Diagram