

# MEDICAREBOT(AI/ML PART)

## REQUIREMENT ANALYSIS

### 1) CHATBOT

TOOLS REQUIRED:

fastapi

uvicorn

pydantic

langchain

langchain\_groq

python-dotenv

gunicorn

Editor/Interpreter used – Jupyter Notebook and VS Code

#### 1. Environment Setup

##### Requirement:

- The application requires environment variables to be loaded, specifically the Groq\_api\_key for initializing the Groq LLM.

##### Analysis:

- The .env file must be properly configured with the necessary API key.
- The application needs access to this environment variable for secure and flexible deployment across different environments.

##### Test Case:

- **Test 1:** Ensure that the .env file is present in the project directory with the Groq\_api\_key.
- **Test 2:** Validate that the API key is loaded correctly by the application using os.getenv and that the llm instance is successfully initialized without errors.

---

#### 2. CORS Middleware Configuration

##### Requirement:

- The application needs to handle Cross-Origin Resource Sharing (CORS) to allow interactions from different domains.

##### Analysis:

- The CORS middleware is configured to allow requests from any origin, with any method, and any headers. This is necessary for client applications (like web front-ends) to interact with the API without restrictions.

##### Test Case:

- **Test 1:** Send a request from a different domain and ensure the API responds without CORS errors.
- **Test 2:** Test various HTTP methods (GET, POST, etc.) and headers to ensure they are allowed.

---

#### 3. Prompt Template and Language Model Chain (LLMChain)

##### Requirement:

- The application uses a PromptTemplate to generate responses based on user inputs, integrating a memory system to maintain conversation context.

#### Analysis:

- The PromptTemplate is designed to handle greetings, irrelevant queries, health-related symptom analysis, and serious symptom warnings with recommendations to consult specific doctors.

#### Test Case:

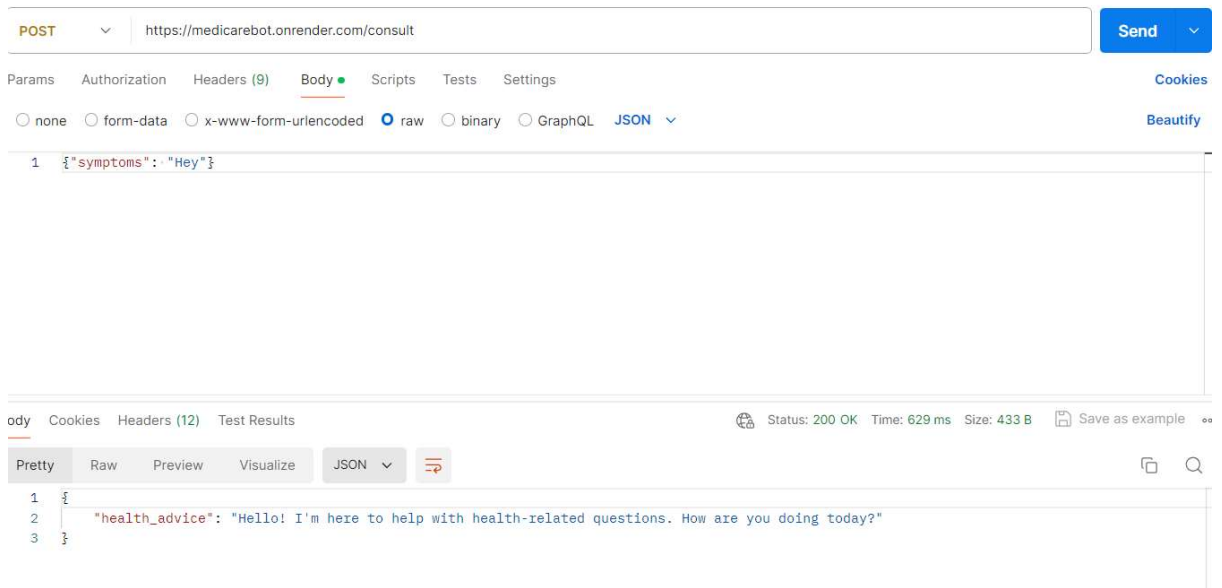
- **Test 1:** Send a greeting message ("Hi") and verify that the response is warm and friendly.
- **Test 2:** Send an irrelevant query and confirm that the response informs the user that the question is outside the chatbot's scope.
- **Test 3:** Provide symptoms like "I have a headache and cold" and check if the chatbot provides appropriate analysis, including potential remedies and doctor recommendations.
- **Test 4:** Introduce severe symptoms and verify if the chatbot recommends consulting a doctor with specific contact details.

The screenshot displays a REST client interface with the following details:

- URL:** `https://medicarebot.onrender.com/consult`
- Method:** `POST`
- Body:** `{ "symptoms": "I am feeling Cold and nauseus" }`
- Status:** `200 OK`, **Time:** `55.72 s`, **Size:** `797 B`
- Response Body (JSON):**

```

{
  "health_advice": "Feeling cold and nauseous could be symptoms of a common cold, flu, or stomach bug. However, they can also indicate more serious conditions like food poisoning or appendicitis. If your nausea is severe, accompanied by vomiting or sharp abdominal pain, I recommend consulting a doctor immediately. For milder cases, you can try the following home remedies: 1. Stay hydrated by sipping water, clear broths, or electrolyte-rich drinks. 2. Rest and avoid physical activities until you feel better. 3. Gradually introduce bland, easy-to-digest foods like toast, rice, or bananas. 4. Use a heating pad or take a warm bath to alleviate body chills. Please monitor your symptoms and seek medical attention if they worsen."
}
```



#### 4. Running the Application

##### Requirement:

- The application is intended to run on uvicorn with reloading enabled for development purposes.

##### Analysis:

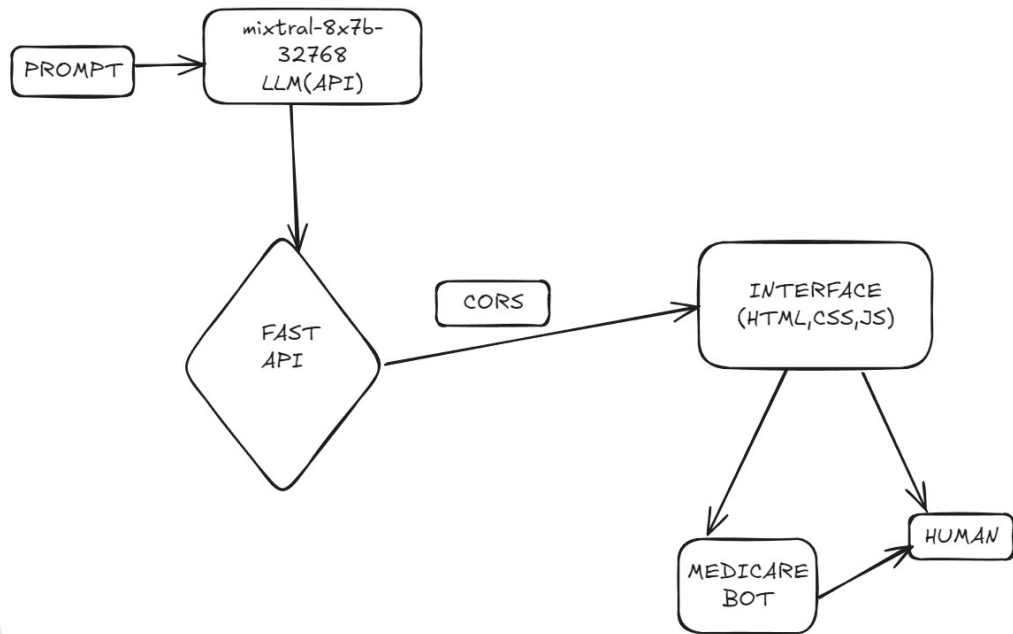
- This setup is ideal for local development, allowing the server to automatically reload when changes are made to the codebase.

##### Test Case:

- **Test 1:** Run the application with uvicorn and verify that it starts correctly and reloads upon code changes.

## CHATBOT DESIGN

TOOL USED-EXCALIDRAW.com



## **WEBSITE(MEDICURE)**

### **REQUIREMENT ANALYSIS**

#### **Frontend:**

1. React.js
  - Requirement: Node.js and npm (Node Package Manager)
  - Description: React is a JavaScript library for building user interfaces. It allows for the creation of reusable UI components and efficiently updates and renders components when data changes.
2. Tailwind CSS
  - Requirement: Node.js and a build process (like webpack or Create React App)
  - Description: Tailwind is a utility-first CSS framework. It provides low-level utility classes to build custom designs without leaving your HTML. It's highly customizable and can lead to faster development times.
3. Font-Awesome (for text styling)
  - Requirement: Can be included via CDN or npm package
  - Description: Font Awesome provides a comprehensive library of icons that can be easily integrated into your project. It's useful for adding visual elements and improving UI/UX.
4. react-icons
  - Requirement: npm package
  - Description: React Icons includes popular icon packs as React components. It's easy to use and can be a good alternative or complement to Font Awesome, especially in React projects.

#### **Backend:**

5. Node.js
  - Requirement: Node.js installation
  - Description: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows you to run JavaScript on the server-side and is the foundation for the backend of this stack.
6. Express.js
  - Requirement: npm package
  - Description: Express is a minimal and flexible Node.js web application framework. It provides a robust set of features for web and mobile applications, making it easier to create API endpoints and handle HTTP requests.
7. MongoDB
  - Requirement: MongoDB installation, MongoDB driver for Node.js
  - Description: MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It's known for its scalability and flexibility, making it a popular choice for modern web applications.

## WEBSITE DESIGN

