Team Name :Tech Veerangana

COLLEGE NAME : NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING AND TECHNOLOGY, TALEGAON DABHADE,410507

Domain of Project : Health and Wellbeing
Inadequate ambulance services
Problem Statement : In event of medical emergency victims face significant challenges in receiving prompt medical attention due to:

- Inadequate ambulance services
- Lack of basic first aid
- Inefficient transportation
- Traffic congestion

Team Leader: Khushi Chaudhari

Team Members:
Tanishka Kadam
Ujwala Dangat
Preeti Pingale

INTRODCTION

Project Overview:

The **MedaAssist** website is a health assistance platform that aims to provide users with various healthcare-related services in one place. It integrates multiple features to assist users in emergency situations and routine health inquiries:

- **Chatbot**: Provides immediate responses to healthcare-related queries.
- **Hospital Locator**: Helps users find nearby hospitals using location-based services.
- **Ambulance Locator**: Allows users to find the nearest ambulance available.
- **Additional Feature**: (Could be something like medical tips, appointment scheduling, etc.).

The purpose of the project is to create an easily accessible platform for medical assistance, especially in emergencies.

Technologies:

The website is built using the following technologies:

- **HTML**: For the structure and layout of the website.
- **CSS**: For styling and creating a user-friendly interface.
- **JavaScript**: For dynamic functionality like smooth scrolling, chatbot interaction, and API integrations.
- **Node.js (Backend)**: If your website requires server-side functionality for APIs or dynamic content.
- **APIs**: For integrating hospital and ambulance locators (e.g., Google Maps API, custom healthcare APIs).

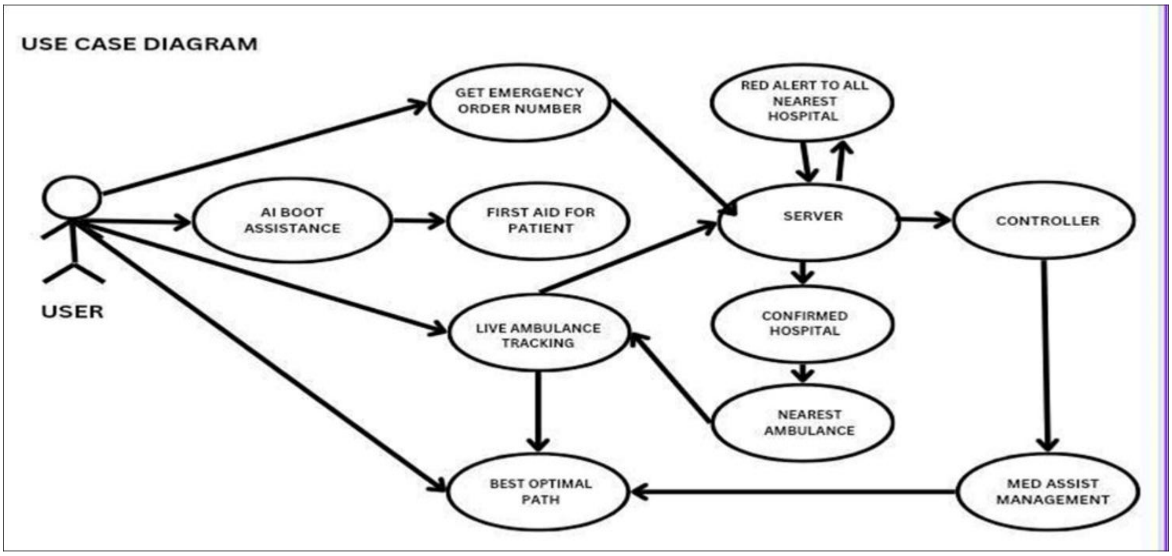
ARCHITECTURE

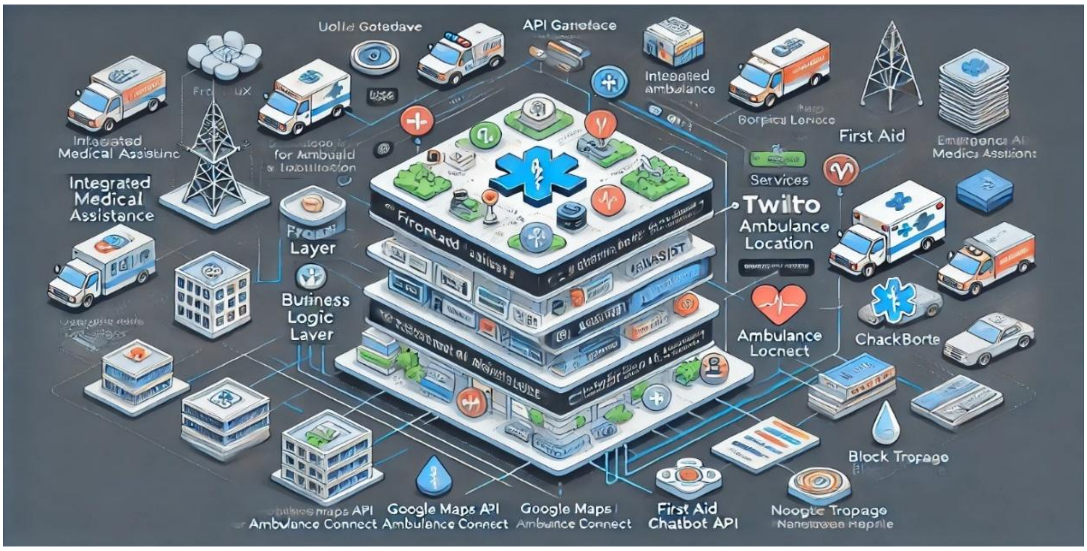
System Architecture:

The MedaAssist website consists of multiple components working together:

- **Frontend (UI)**: The user interface consists of HTML, CSS, and JavaScript files. The UI includes different sections such as a chatbot, hospital locator, ambulance locator, and other features.
- **Backend (Optional)**: If you are using a backend, you will have a server running Node.js or another framework that handles API requests and manages dynamic content like user queries.
- **External APIs**: These are third-party services used to get data such as hospital locations and ambulance routes. They can be integrated using AJAX or RESTful API calls.

.




Uolld Gotedave
API Ganeface
Integrated Medical Assistinc
Integrated Medical Assistance
Intealled ambulance
Borpical Lenoco
First Aid
Emergency AI Medica Assitions
for Ambulild
Front Layer
Business Logic Laver
Services
Twilto Ambulance Location
Ambulance Locnect
ChackBorte
Block Tropage
Google Maps API Ambulance Connect
Google Maps API Ambulance Connect
Google Maps Ambunce Connect
First Aid Chatbot API
Noogte Tropage

FUNCTIONALITY

Chatbot:

The **Chatbot** is designed to provide instant replies to users' queries related to health, emergency contacts, and general medical information. It listens to user inputs and responds accordingly. The chatbot's functionality includes:

- **Message handling**: The user types a query, and the chatbot processes it to provide a response (either predefined or fetched dynamically).
- **API Integration (Optional)**: If you're integrating an AI-based or medical database API, the chatbot can fetch real-time data, like doctor recommendations or symptoms checkers.

Hospital Locator:

The **Hospital Locator** uses an API (e.g., Google Maps API or a healthcare-specific service) to find hospitals based on the user's location. Key features include:

- **Location-based search**: The user allows the website to access their location.
- **API Call**: The frontend sends the location data to the API, which returns a list of nearby hospitals.
- **Display results**: The hospitals are displayed on the map or in a list with details such as distance, contact, and ratings.

Ambulance Locator:

The **Ambulance Locator** operates similarly to the hospital locator. It uses an API to track nearby ambulances:

- **Real-time tracking**: The user can view available ambulances in real-time.
- **Integration with Map APIs**: The ambulance locations are shown on a map, and users can call or request an ambulance.

1. **Start Your Server**: If using Node.js, run node app.js or npm start on the server.
2. **Configure Domain and SSL** (optional): Set up your domain name (e.g., medaassist.com) and secure it with an SSL certificate.

Files

1. chatboot.css

Purpose: Contains styling for a chatbot interface, setting styles for the body, chat container, chat header, chat window, and other UI elements. It defines a light background and positions a chat box at the center of the page with a modern, minimalist look.

2. chatboot.html

Purpose: Provides the HTML structure for the chatbot interface, titled "MedAssist Chatbot." It includes a container for chat messages, an input box for user input, and basic sections for displaying responses.

3. chatboot.js

Purpose: Implements the functionality of the chatbot by defining a basic first aid guidance database. The chatbot responds with guidance for common injuries and symptoms like heart attack, head injury, burns, etc., based on user input.

4. index.html

Purpose: Serves as the main landing page for MedAssist, themed with a pink color scheme. The page has a welcoming appearance with basic information and navigation options, likely linking to other functions or pages within the MedAssist system.

5. index (2).html

Purpose: This page is designed as a "Nearest Hospital Locator" using LocationIQ for map functionality. It includes a map rendered with Leaflet.js and likely displays nearby hospitals using geolocation.

6. index (3).html

Purpose: Provides a "Nearest Ambulance Locator" feature, also implemented using Leaflet.js. It displays a map with an interface for locating ambulances, likely utilizing real-time location or static ambulance locations.

7. RTN.html

Purpose: Implements a "Real-Time Navigation with Traffic-Based Rerouting" feature. The page includes a map and likely supports traffic-based route adjustments, allowing for better real-time navigation.

8. RTN.js

Purpose: JavaScript code for calculating routes, leveraging the OpenRouteService API to retrieve directions with traffic data. This file includes asynchronous functions to fetch traffic-influenced routes, providing rerouting options based on live conditions.

TESTING AND DEBBUGING

Testing Procedures:

1. **Manual Testing**: Verify the functionality of each feature (chatbot, hospital locator, ambulance locator).
2. **User Interaction Testing**: Ensure the site is user-friendly, with proper navigation and error handling.
3. **API Testing**: Use tools like **Postman** to test external APIs and make sure they return the correct data.
4. **Cross-browser testing**: Ensure the website works across all major browsers.

Bug Fixing:

- **Issue**: Chatbot not responding to user queries.
    - **Fix**: Verify API integration and make sure the frontend is correctly sending data to the backend.
- **Issue**: Hospital locator API returning no results.
    - **Fix**: Check the API key and ensure location permissions are enabled in the browser.

CONCLUSION

The **MedaAssist** project achieved its primary goal of providing users with a centralized health assistance platform. Key features such as the **chatbot**, **hospital locator**, and **ambulance locator** were successfully integrated, offering users valuable services in emergency situations. The website was deployed on **Vultr** and is accessible to the public.

Future Improvements:

- Add more features like appointment scheduling or virtual consultations.
- Improve chatbot intelligence by integrating AI-based APIs for better user interactions.

Challenges Encountered:

- API integration issues, especially with real-time location-based services.
- Ensuring smooth UI/UX across different devices and browsers.