Team Name :Tech Veerangana

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

Domain of Project : Health and Wellbeing

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).

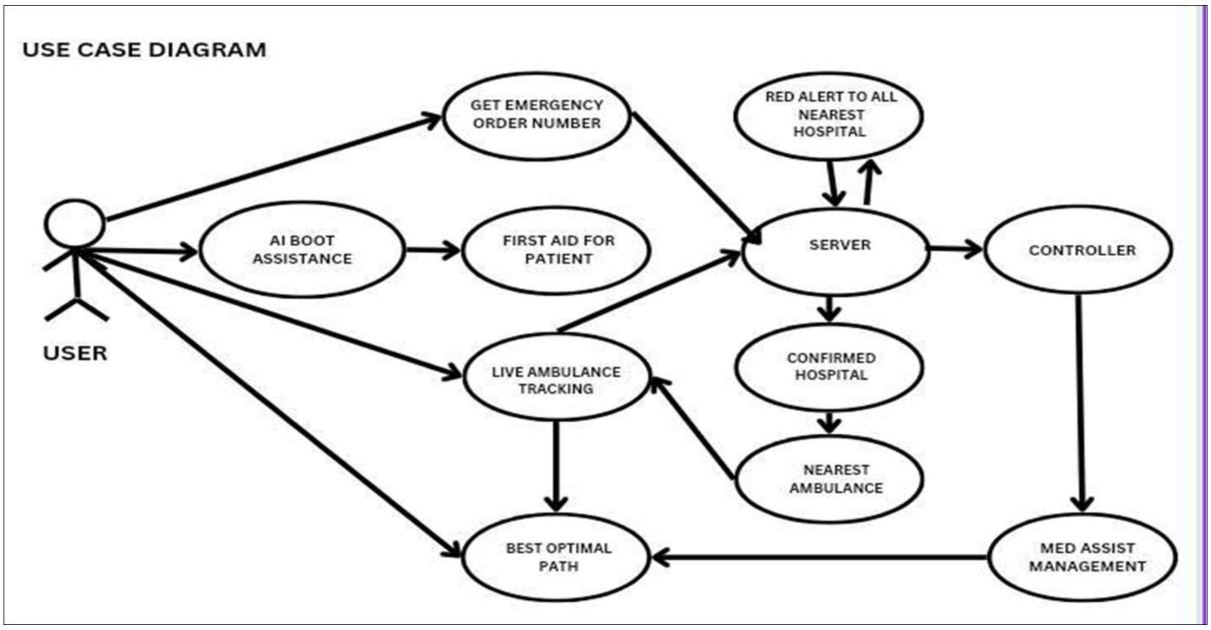
ARCHITECTURE

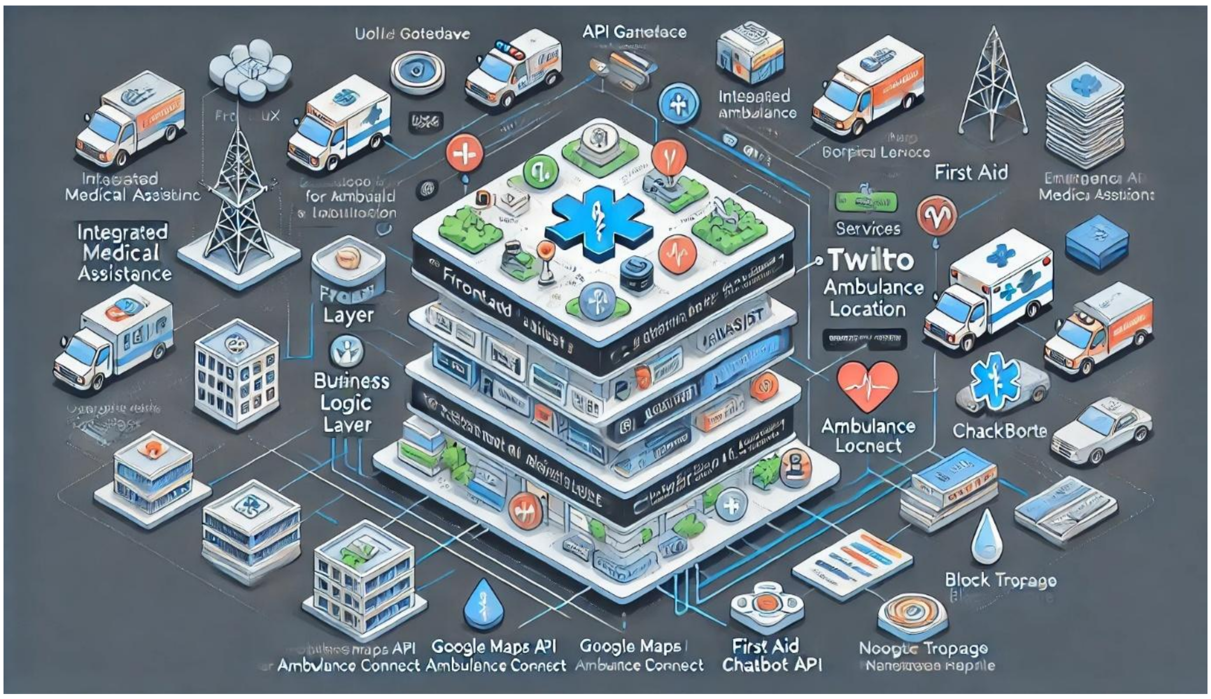
System Architecture:

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.

.


USE CASE DIAGRAM
USER
GET EMERGENCY ORDER NUMBER
AI BOOT ASSISTANCE
FIRST AID FOR PATIENT
LIVE AMBULANCE TRACKING
BEST OPTIMAL PATH
RED ALERT TO ALL NEAREST HOSPITAL
SERVER
CONFIRMED HOSPITAL
NEAREST AMBULANCE
CONTROLLER
MED ASSIST MANAGEMENT


API Gateface
Integrated Medical Assistance
Integrated ambulance
First Aid
Services
Twilto Ambulance Location
Frontend
Layer
Business Logic Layer
Ambulance Locnect
ChackBorte
Block Tropage
Google Maps API Ambulance Connect
Google Maps Ambuince Connect
First Aid Chatbot API
Noogle 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.

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.

Detailed Project Plan

Project Objectives:

- **Develop a health assistance website** that integrates multiple features, including:
  - **Chatbot**: Assists users with healthcare-related queries.
  - **Hospital Locator**: Helps users find nearby hospitals using an API.
  - **Ambulance Locator**: Allows users to find the nearest available ambulance.
  - **Additional Feature**: Any extra functionality (e.g., medical tips, appointment scheduling, etc.).
- **Ensure user-friendly navigation** across all components.
- **Deploy the website** to a cloud platform (e.g., Vultr) for public access.
- **Test and debug** the site thoroughly to ensure smooth operation.

Timeline with Milestones:

Phase	**Tasks**	**Timeline**	**Milestone**
Phase 1: Planning	Finalize project requirements and structure.	Week 1	Project plan and requirements document finalized.
Phase 2: Front-End Setup	Develop the basic layout (HTML, CSS), design navigation, and homepage.	Week 2	Basic website structure completed.
Phase 3: Feature Integration	Integrate chatbot, hospital locator, and ambulance locator into the site.	Week 3-4	Core features integrated and functional.
Phase 4: Back-End/API Integration	Integrate external APIs (e.g., hospital locator API).	Week 5	All API integrations working.
Phase 5: Testing	Test functionality, fix bugs, and improve performance.	Week 6	Testing completed, bugs resolved.
Phase 6: Deployment	Deploy to a cloud server (Vultr).	Week 7	Website deployed on Vultr and publicly accessible.
Phase 7: Final Review	Final testing, adjustments, and project handoff.	Week 8	Project completed and handed over.

Deliverables:

1. **Functional Website** with the integrated features.
2. **Technical Documentation** (PDF).
3. **Codebase** (GitHub or ZIP folder with all files).