**MEDMAP**

## **A PROJECT REPORT**

***Submitted by,***

**Team-G84**

**Mr Thushar R - 20201CSE0527**

**MS. Ashwadhi s– 20201CSE0520**

**Mr. Narendra Kumar Reddy – 20201CSE0535**

**Mr. H S Uthkarsh – 20211LCS0011**

### *Under the guidance of,*

**Dr./Mr. Jothish C Sir**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING, COMPUTER ENGINEERING, INFORMATION SCIENCE AND ENGINEERING Etc.**

**At**

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE**

**CERTIFICATE**

This is to certify that the Project report **“MedMap”** being submitted by “Thushar R, Ashwadhi S, Narendra Kumar Reddy, H S Uthkarsh” bearing roll number(s) “20201CSE0527, 20201CSE520, 20201CSE0535, 20211LCS0011” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bona fide work carried out under my supervision.

|  |  |
| --- | --- |
| **Dr Jothish C Sir**  Asst Professor  School of CSE&IS  Presidency University | **Dr. Pallavi**  DESIGNATION & HoD  School of CSE&IS  Presidency University |

|  |  |  |
| --- | --- | --- |
| **Dr. C. KALAIARASAN**  Associate Dean  School of CSE&IS  Presidency University | **Dr. L. SHAKKEERA**  Associate Dean  School of CSE&IS  Presidency University | **Dr. SAMEERUDDIN KHAN** Dean  School of CSE&IS  Presidency University |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **MedMap** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Jothish Sir, Asst Professor,** **School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Mr Thushar R - 20201CSE0527

MS. Ashwadhi s– 20201CSE0520

Mr. Narendra Kumar Reddy – 20201CSE0535

Mr. H S Uthkarsh – 20211LCS0011

|  |  |
| --- | --- |
|  |  |

**ABSTRACT**

The remarkable progress in technology has significantly impacted various aspects of human life, and the rapid evolution of smartphones stands out as a pivotal contributor to our daily efficiency. In the realm of this technological advancement, our project, MedMap, endeavors to create a web application that simplifies the process of locating healthcare facilities based on specific specializations. Users have the flexibility to choose the distance range from their location, and this search can be refined according to the type of healthcare unit, whether it be a hospital, clinic, or health center. MedMap not only provides concise information about the identified healthcare units but also offers a visual representation by displaying their locations on an interactive map. This initiative aims to harness the power of technology to streamline and enhance the accessibility of healthcare services for users.

The proposed application is developed that locates the 5 nearest hospitals with the desired medical speciality. The nearest position of hospitals is calculated with a built-in feature of Global Positioning System in React Geo Location. With the help of this application, a patient can find the nearest hospital. A comprehensive profile hospital is available in the application including the website, mailing addresses and contact numbers. The Web application is a user friendly app which provides the required details to the patients in an efficient manner.

**ACKNOWLEDGEMENT**

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and Dr. “Pallavi”, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr./Mr Jothish C Sir, Asst Prof**, School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators “NAME”.

**Thushar R**  **Ashwadhi S**

**Narendra Kumar Reddy**

**Uthakarsh HS**

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Figure Name** |  | **Page No.** |
| 1 | Figure 6.0.1 |  | 19 |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Figure 6.0.2  Figure 6.0.3  Figure 6.0.4  Figure 6.0.5  Figure 7.0.1  Figure 7.0.2  Figure c.0.1  Figure c.0.2  Figure c.0.3  Figure c.0.4  Figure c.0.5  Figure c.o.6  Figure c.0.7  Figure c.0.8 |  | 23  25  28  32  34  34  44  44  45  45  46  46  47  47 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **ACKNOWLEDGMENT**  **…** | **5**  **6**  **…** |
| **2.**    **3.** | **INTRODUCTION**  **1.1 Key components of MedMap**  **LITERATURE SURVEY**  **RESEARCH GAPS OF EXISTING METHODS** | 9  12 |
| 4.  4.1  4.2  4.3  4.5  4.6  4.7  4.8  5  6  6.1  6.2  6.3  6.4  6.5  6.6  6.7  7  8  9  10 | PROPOSED MOTHODOLOGY  HARDWARE AND SOFTWARE  USER AUTHENTICATION  HOSPITAL DETAILS PAGE  MAP INTEGRATION  SEACRCH AND FILTERING  OBJECTIVES  SYSTEM DESIGN AND IMPLEMENATION  MODEL  VIEW  CONTROLLER  ROUTES  DATA FETCHING  COORDINATE TARNSFORMATION FUNCTION  HAVERSINE DISTANCE CALCULATION  TIMELINE FOR EXCEUTION OF PROJECT  OUTCOMES  RESULTS AND DISCUSSIONS  REFRENCES  PESUDOCODE  APPENDIX B |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER-1**

**INTRODUCTION**

MedMap has been meticulously crafted to deliver patients specialized treatment, featuring near by loactions within 5km radius. While the conventional general hospital, different department, is widely recognized, the pivotal role of district hospitals in serving as major healthcare facilities cannot be overstated. MedMap's adaptability is a cornerstone, extending its reach to various specialized hospitals, encompassing rehabilitation facilities, children's hospitals, geriatric facilities for seniors, and those specifically addressing medical needs such as orthopedic problems. This diverse spectrum of services aspires to contribute significantly to reducing healthcare costs when compared to conventional general hospitals.

The intricate structure of MedMap is characterized by various departments, including Multispecialty, Urgent Care, and Intensive Care, complemented by specialized units like cardiology. Some hospitals within the network feature outpatient departments and chronic treatment units. Common support units such as pharmacy, pathology, and radiology are integral components that contribute to the overall functionality and efficiency of MedMap.

An exceptional feature of MedMap lies in its provision of hospital directions and user reviews, presenting users with a comprehensive perspective on healthcare facilities. The primary objective of this application is to elevate the user experience through additional features. The user interface has been intentionally streamlined, presenting precisely the right amount of information. MedMap seamlessly displays the nearest hospitals within a specified radius, providing patient details such as ratings, contact numbers, and navigational routes. By breaking down traditional barriers, MedMap establishes fluid connections between patients and an extensive network of healthcare providers, specialists, and medical facilities, ultimately fostering a more efficient and user-friendly healthcare experience.

**1.1 Key Components of MedMap**

**1.1.1 User Authentication:**

* Allow users to create accounts and log in securely.
* Implement secure authentication methods to protect user data.

**1.1.2 Intuitive User Interface (UI):**

* Design a user-friendly interface for easy navigation.
* Ensure a seamless and intuitive user experience.

**1.1.3 Hospital Database:**

* Establish a comprehensive database of hospitals with up-to-date information.
* Include details such as location, contact information, specialties, services offered, and operational hours.

**1.1.4 Geolocation Services:**

* Integrate geolocation features to identify the user's current location.
* Utilize GPS to provide accurate distances to nearby hospitals.

**1.1.5Mapping Integration:**

* Incorporate mapping functionalities for visualizing hospital locations.
* Allow users to view hospitals on a map and get directions.

**1.1.6 Search Engine:**

* Develop a robust search engine with filters for location, specialty, services, and accreditation status.
* Enable users to find hospitals based on specific criteria.

**1.1.7 Hospital Profiles:**

* Create detailed profiles for each hospital, including reviews, ratings, available facilities, and any unique features.
* Provide a comprehensive overview to assist users in making informed decisions.

**1.1.8 User Reviews and Ratings:**

* Implement a rating system for hospitals and their services.
* Allow users to leave reviews to share their experiences.

**1.1.9 Security Measures:**

* Implemented hashed security protocols to protect user data, including personal and health-related information.
* Comply with privacy regulations and standards.

**CHAPTER-2**

**LITERATURE SURVEY**

V.S.P Vidanapathirana, K.H.M.R Peiris and Dhishan Dhammearatchi has discussed about the utilization of web applications in healthcare settings, specifically those operating on the Web platform, to facilitate the location and direction-finding within hospitals. The system use Bluetooth BLE for its implementation process with the aid of iBeacons.The primary purpose of this system is to assist users in navigating hospital facilities, helping them locate various critical areas such as wards, patient rooms, and the pharmacy.

Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites.

Muhammad Wasim Munir, Syed Muhammad Omair and M. Zeeshan Ul Haque has discussed about the development of web applications operating on the Web platform that facilitate the identification and location of hospitals with specific medical specialists within a 5 km radius in Karachi employing GPS Technology.[2] Central to the application's functionality is its capacity to provide comprehensive profiles of both doctors and hospitals,

Ajay Kumar G R, Akash Aman, Avinash Kumar, Harshith L has discussed about the development of Web applications tailored to locate hospitals using GPS and also provide services to patients, such as providing hospital information and predicting diseases. This application allows patients to register and access crucial information about hospitals while also leveraging symptom-based disease prediction to guide users to the most suitable medical facilities.

Pratima Panneer Selvam had discussed about the development of healthcare applications that deliver comprehensive hospital information online and through text-based services.The application is designed to provide users with vital information about various hospitals, along with other healthcare-related features. User can access the nearest hospitals by using GPS feature and also can send the SMS request and receive the required information.

Devayani.G, HariPriya.R, Sruthi.S, C.SenthilKumarhas discussed about the application that enable patients to search for hospitals based on their specific healthcare needs and subsequently provide detailed hospital and doctor information in response to patient requests is a significant area of interest.[5] The application is used to find the hospital within five km radius with built-in feature of GPS and find route from their current location through Google map application program interfaces(API).

Shivam Bajpai, Tushar Modi, Vatsalya Vinay Sinha, Vidhi Jaiswal had developed the application that assist users in locating the nearest hospital within a five-kilometer radius, specifically catering to their desired medical specialist. The nearest hospitals are located using GPS. This application also provides additional information like the opening-closing time of the hospital, availability of doctors and beds in the hospital.

Leila GHOLAMHOSSEINI, Farahnaz SADOUGHI, Aliasghar SAFAEI had developed Hospital real-time location system(HRTLS) which is an application based ubiquitous computing. The application is also used to track the movement of patients and ensure their safety. Technologies like IOT and cloud computing or a combination of these two are used to design the real-time location system and RFID technology is used to locate patients in different areas.

Ganapathi Shankar, DR. D Subba Rao has discussed about development of an Web-based tracing system for Emergency Medical Services (EMS) on the cloud, aimed at guiding emergency patients to the nearest appropriate hospitals.This application is helpful for senior people and chronic patients as it uses technologies like GPS to track patient in-case of emergency and the streamlining data management is done using cloud computing.

H. Gamboa, F. Silva, H. Silva has discussed about a wireless wearable sensing system which is designed to monitor patients within the building in real-time on a continuous basis. This is designed for indoor usage and it is an integrated system, including a set of end-devices, tracking routers and a base station.

Syed Farzana, Kanakam Sasikalyan, Jasti Manikanta, Kommalapati Manoj, Choppara Prasanth has discussed about the development of a system for locating hospitals in close proximity to a patient's location by leveraging the Google Maps API.[10] Identifying nearby hospitals with the minimum travel distance and assessing bed availability is the main aim. Application includes utilization of computer simulation forecasting for reliable predictions, especially regarding bed availability in hospitals.

**CHAPTER-3**

**RESEARCH GAPS OF EXISTING METHODS**

DocAppointments.com.au originated from Australia. It is the first fully integrated, automatic and real time online appointment booking system in Australia. The system available on a website and on IOS and Web phones. Users will have a 24hoursaday access to the appointments online and on their IOS devices and Web devices. Also, the system allows users to choose their own appointment time from a wider range of available time nominate by the administrator [2]. Other than that, users can also choose the doctors that are available to make an appointment with. Users can select the ‘current location’ option to find the nearest Doctor’s Practice or search the list to locate the local Practices.

Furthermore, the system also allows users to choose their preferred doctor that is available. User has the access to cancel their appointment. Once the appointment is made, the details will be saved automatically on users’ calendar which includes the practice’s address and phone number [3]. The disadvantage of the system is it needs the medical practices to download it software which will cost the practices money as it is not free.

Health Engine is Australia’s largest online booking system and health marketplace. Health Engine helps people to find and book health appointment anywhere and anytime. The application helps user to find available doctor, dentist, physiotherapist and other medical practices and help them to book the appointment online [4]. Health Engine available on a website and Web and IOS application. It provides a quick easy search service for health practitioners and health practices in Australia. The application also provides user with the function of finding the nearby medical practices and the directions to them. The drawback of this application is it does not require user registration; thus, user have to fill in their information for every appointment scheduling based on above details create

The selected qualitative case study methodology aligns strategically with the study's focus on subjective assessment, capturing attitudes, opinions, and behaviours. Grounded in the principles outlined by Khotari (2004) and the characteristics elucidated by Butina, Campbell, and Miller, this approach places the researcher as a vital instrument in both data collection and analysis. By adopting an inductive stance, the study seeks to illuminate the subjective interpretations and meaning attributions of individuals, aiming to construct a theoretical framework that addresses the research questions.

The heavy reliance on desk research and document analysis, while valuable, may benefit from the incorporation of diverse data collection methods. A more nuanced approach that encompasses alternative methodologies could enrich the study's depth and breadth, ensuring a more robust research design.

**CHAPTER-4**

**PROPOSED MOTHODOLOGY**

**HARDWARE & SOFTWARE**

The development of a MedMap Portal requires a careful integration of both hardware and software components to ensure a seamless and secure user experience while also allowing scalability for future demands.

**Hardware Requirements:**

1. PC or Laptop: Users need standard desktops or laptops to access the Medical Appointment Application.

2. Operating System: Compatibility with Windows ensures flexibility based on user preferences.

3. Web Browsers: Compatibility with major browsers (e.g., Chrome, Edge) ensures a smooth user experience.

**Software Requirements:**

Developing a Healthcare Access Portal involves a robust stack of technologies for a dynamic and secure user interface along with a sturdy backend functionality:

Vscode

Postman

Mongo

Github

**Frontend Technologies:**

Tailwind: Handles styling and visual presentation, enhancing the portal's aesthetic appeal and user experience.

React JS: React Native is an exciting framework that enables web developers to create robust web applications using their existing JavaScript knowledge.

Figma: Figma is an interactive application focused towards the design of web pages and helps add functionality.

**Backend Technologies:**

MongoDB : MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need.

Express : Web Applications Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and web applications.

React Js : Its is useful when you want to specify that a prop or variable can hold any type of content that can be rendered by React, but you don't want to specify a more specific type.

Node Js: As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.

The integration of frontend technologies like native react, CSS ensures a user friendly and visually appealing interface. On the backend, we used node js because node is single language for both front end and backend , hence it consumes time and also offer a powerful and scalable framework, while JSON facilitates seamless data exchange. The MongoDB database system ensures robust data management, crucial for healthcare applications' compliance and security standards.

**User Authentication:**

To establish a secure login system, users will first need to register with the hospital finder application, providing essential details such as username, email, and a securely hashed password. During registration, input validation and secure password storage practices should be implemented to enhance security. Subsequently, a user authentication system, such as JSON Web Tokens (JWT) or OAuth, should be integrated for secure session management. For user profile management, users can edit their personal information, change passwords, and manage preferences through a dedicated profile page. Encryption and secure protocols, such as HTTPS, should be enforced to protect sensitive data during transmission. Additionally, incorporating multi-factor authentication adds an extra layer of security. Regularly updating and patching security vulnerabilities is crucial to maintaining the integrity of the login system.

**Hospital Details Page**:

The detailed page for each hospital serves as a comprehensive source of information for users, offering a holistic view of the healthcare facility. Contact details such as the hospital's address, phone number, and email address are prominently featured. To enhance user decision-making, the page incorporates a rating system. This detailed approach aims to empower users in making informed choices when selecting a healthcare provider.

**Map Integration:**

We leveraged geolocation services and implemented the Haversine formula to ascertain the distance between the user's current location and potential hospitals. Utilizing latitude and longitude data stored in the database, the application dynamically marks the five nearest hospitals on the map. This approach ensures an accurate representation of nearby healthcare facilities, enhancing user convenience and facilitating efficient navigation to the closest medical services. Through the combination of geolocation technologies and the Haversine formula, the application provides users with real-time information, optimizing the user experience and aiding in prompt decision-making when seeking medical assistance.

**Search and Filtering:**

The search functionality in the hospital finder web application enables users to locate hospitals based on various criteria, including name, location, and services offered. Users can input specific hospital names, explore hospitals within a particular geographic area, or filter results based on the services they require. Additionally, the application allows users to refine and enhance their search experience by providing sorting options for search results. This feature enables users to organize and prioritize hospitals based on relevant parameters, ensuring a more tailored and efficient discovery process. The combination of search, filtering, and sorting functionalities contributes to a user-friendly experience, allowing individuals to quickly identify and access the most relevant healthcare facilities based on their specific needs.

**CHAPTER-5**

**OBJECTIVES**

1. Up-to-Date Hospital Information: Accurately gather and continually update data on hospitals, covering essential details like location, contact information, specialties, ensuring users access the most current information.

1. Geolocation and Mapping Integration: Seamlessly integrate geolocation features into the app, aiding users in finding nearby hospitals based on their current position. Incorporate mapping functionalities for a simplified navigation experience.

1. Search Functionality: A search engine that empowers users to locate hospitals using diverse criteria such as location, specialty, provided services, and accreditation status, enhancing the precision of their search.

**CHAPTER-6**

**SYSTEM DESIGN & IMPLEMENTATION**

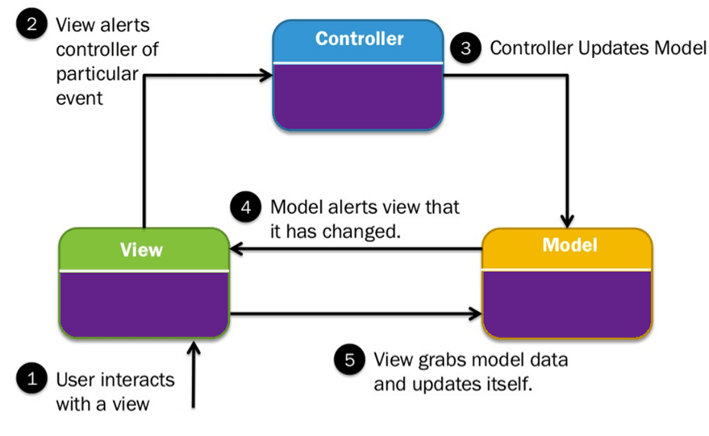


Fig 6.0.1

The MedMap project embraces a comprehensive Model-View-Controller (MVC) architecture, a design pattern that fosters a structured and modular approach to web development. This pattern is instrumental in enhancing code organization, maintainability, and scalability. Let's delve into each component of the MVC structure in the context of MedMap, highlighting the distinctive roles they play.

The Model component of MedMap serves as the backbone for managing data. In the User model, user-related information such as registration details and authentication credentials are stored. MongoDB, a NoSQL database, is employed to house these models due to its schema-less nature, enabling flexibility in handling dynamic user data. Similarly, the Hospitals model governs healthcare facility data, encompassing information such as names and geographical coordinates. This organization of data into distinct models ensures clarity and separation of concerns, contributing to the overall maintainability and scalability of the application. MongoDB, being a document-based database, aligns seamlessly with the JSON-like structure commonly used in JavaScript applications.

In the View component of MedMap, the frontend interface is meticulously crafted using React, a declarative and component-based JavaScript library renowned for its efficiency in building modern and interactive user interfaces. The Home Page, a dynamic centerpiece of the application, skillfully utilizes React to render a map showcasing the user's location and the closest hospitals in real-time. This not only underscores React's capacity for seamless integration of geolocation data and hospital information but also highlights its ability to create visually engaging and responsive interfaces.

React's component-based architecture is a key factor in optimizing the maintainability and scalability of the frontend. Each UI element, encapsulated within a React component, represents a modular and reusable building block. This modular approach ensures that code can be efficiently organized, tested, and maintained, contributing to the overall robustness of the frontend. As a result, the Home Page's dynamic rendering of the map is achieved through the composition of modular components, each responsible for a specific aspect of the user interface.

The Hospitals Page, another facet of the View component, exemplifies React's prowess in creating interactive and user-friendly interfaces. Equipped with a search bar, this page allows users to interactively query hospitals in the database. React's ability to manage state efficiently ensures that the interface remains responsive, providing users with a seamless and intuitive search experience. The modular components on the Hospitals Page contribute to the overall responsiveness, enabling the page to adapt dynamically to user interactions.

In addition to React, MedMap leverages Tailwind CSS to style and design the frontend components. Tailwind is a utility-first CSS framework that streamlines the styling process by providing a set of pre-defined utility classes. This approach allows developers to rapidly style components without the need for custom CSS, facilitating a consistent and visually appealing design across the entire application. Tailwind's utility-first methodology aligns well with the component-based structure of React, offering a pragmatic and efficient way to manage styles and layout.

In summary, the View component in MedMap harmoniously integrates React and Tailwind CSS to craft a frontend interface that is not only visually compelling but also highly functional. React's component-based structure facilitates the creation of modular and reusable UI elements, optimizing maintainability and scalability. Tailwind CSS complements this approach by simplifying the styling process, ensuring a consistent and aesthetically pleasing design throughout the application. Together, React and Tailwind contribute to the creation of an engaging and responsive user interface in MedMap.

The Controller component acts as a mediator between the Model and View, orchestrating the flow of data and business logic. The Auth controller manages user authentication processes, ensuring secure access to the application. Admin controller oversees administrative tasks, such as hospital management and user account deletion. The Hospital controller governs core logic related to hospital-finding functionality. Express, a minimalist and flexible Node.js web application framework, is utilized to build these controllers. Express simplifies routing and middleware usage, facilitating the separation of concerns and promoting an organized codebase. Each controller encapsulates specific functionalities, ensuring a modular and scalable architecture.

In the realm of Routes, Express is used to establish mappings between incoming HTTP requests and corresponding Controller actions. Auth routes handle user authentication-related requests, Admin routes facilitate administrative actions, and Hospital routes manage requests related to hospital information retrieval. This separation of routes aligns with the principle of distinct responsibilities for each component, ensuring a clean and logical flow of control within the application. Express routing facilitates the smooth handling of diverse functionalities and API calls.

Furthermore, to enable seamless communication between the frontend and backend, the Axios library is employed for making HTTP requests. Axios simplifies data fetching and sending processes, enhancing the efficiency of data exchange between the View and Controller components. Its promise-based architecture aligns seamlessly with the asynchronous nature of web development, ensuring smooth and responsive data transaction

In conclusion, the meticulous adoption of the MVC architecture in MedMap, coupled with the integration of Express, MongoDB, React, and the Axios library, establishes a robust foundation for a scalable, maintainable, and user-centric hospital finder application. The careful orchestration of Models, Views, and Controllers, along with the efficient data communication facilitated by the Axios library, contributes to the overall success and efficiency of MedMap in delivering a seamless and reliable healthcare location service.

**Model: User and Hospitals**

In MedMap, the Models encapsulate the data-handling logic. The User model manages user-related data, such as registration details and authentication credentials, ensuring a secure and personalized experience for users. Simultaneously, the Hospitals model governs information related to healthcare facilities, including names, geographical coordinates, and other relevant details. MongoDB, a NoSQL database, is employed to store these models, providing flexibility and scalability. The schema-less nature of MongoDB aligns seamlessly with the dynamic requirements of user and hospital data in MedMap.

The User model in MedMap encapsulates the data-handling logic for user-related information, ensuring a secure and personalized experience for each individual interacting with the application. The schema for the User model includes essential details such as:

Name: Capturing the user's full name for personalization.

Phone: Storing the user's contact number for communication purposes.

Email: Facilitating unique user identification and communication.

Gender: Recording the gender of the user for personalization and statistical analysis.

DOB (Date of Birth): Providing a critical piece of information for personalization and, potentially, age-related services.

Password: Ensuring the security of user accounts through encrypted password storage.

This schema enables the User model to efficiently manage registration details and authentication credentials, contributing to a secure and user-centric experience.

**Hospital Model:**

Concurrently, the Hospital model governs the data associated with healthcare facilities in MedMap. The schema for the Hospital model includes a comprehensive set of information, covering:

Name: Identifying the healthcare facility by its name.

Type: to identify the type of hospital(general, multispecialty, eyecare etc.)

Location: Capturing the general location of the hospital.

Latitude and Longitude: Precisely pinpointing the geographical coordinates of the hospital for accurate mapping.

Phone: Providing contact information for users seeking medical assistance.

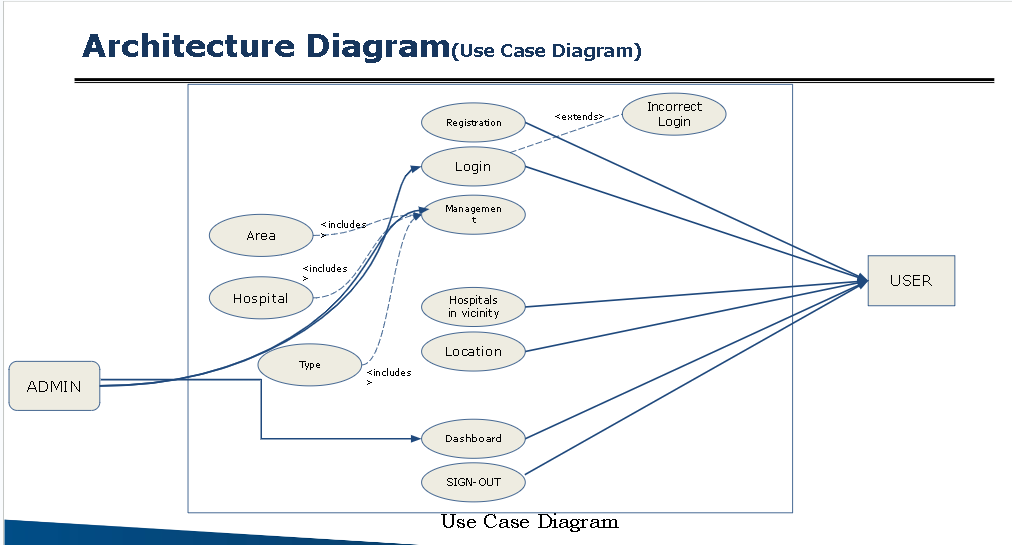
Address: Detailing the physical location of the hospital for navigation and accessibility.

Image Links: Incorporating visual elements by linking to images, potentially aiding users in recognizing the hospital.

The schema for the Hospital model aligns with the diverse set of information needed to facilitate efficient hospital-finding functionality. Storing latitude and longitude coordinates allows for seamless integration with mapping features, and including image links enhances the visual appeal and recognition of healthcare facilities.

**View: Frontend Components – (Home/Hospitals/Locations)**

The View component within MedMap serves as the user interface, facilitating interactive engagement with the application's features. The Home Page, a focal point of the frontend, dynamically renders a map of Bangalore. Leveraging geolocation, it captures the user's location, converting the geographical coordinates into pixel values. This pixel-to-coordinate transformation ensures accurate mapping on the displayed image, offering users an intuitive representation of their position on the map.

 Fig 6.0.2

The dynamic rendering of the Home Page is further enriched by the incorporation of information retrieved from the Models and manipulated by the Controllers. Utilizing the React library, the Home Page adopts a declarative and component-based structure, enabling the creation of modular and reusable UI elements. This approach enhances the overall responsiveness of the interface, providing users with a seamless and visually engaging experience.

Additionally, the Home Page displays the five nearest hospitals based on the user's location. This functionality is powered by a distance calculator that employs the Haversine distance calculation theory. The utilization of Haversine ensures accurate distance measurements on the Earth's surface, enhancing the precision of hospital proximity information presented to the user.

Furthermore, each area on the map of Bangalore is mapped using area tags. Clicking on a specific area redirects users to a dedicated page showcasing all the hospitals located in that particular region. This geographical segmentation enhances the user experience by allowing focused exploration of healthcare facilities in specific areas of interest.

The frontend also includes essential user interaction features such as a login page and a registration page, facilitating secure access to personalized services. A user profile page offers a snapshot of basic information, enhancing user engagement and personalization. The hospital page acts as a comprehensive repository of hospital data retrieved from the database, and a search bar enables users to query hospitals based on area names or types, ensuring a user-friendly and efficient navigation experience.

For detailed exploration, a hospitals detail page is provided, offering comprehensive information such as images, an embedded Google Map through an iframe, hospital name, address, and contact details. Clicking on a specific hospital initiates a seamless transition to this detailed view, providing users with in-depth insights into their chosen healthcare facility.

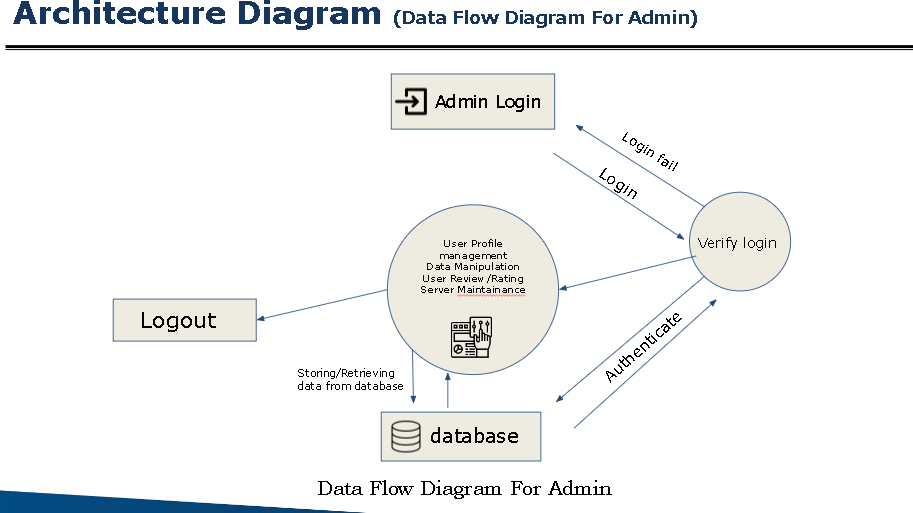
In summary, the View component in MedMap, exemplified by the Home Page and other user-centric pages, showcases a sophisticated and responsive frontend interface. The seamless integration of geolocation, pixel mapping, and distance calculations enhances the user's ability to find and explore healthcare options efficiently. The use of React's component-based structure ensures modularity and responsiveness throughout the application, contributing to an overall positive and user-friendly experience.

**Controller: Auth, Admin, and Hospital Controllers**

The Controllers act as intermediaries between the Models and Views, orchestrating the flow of data and business logic. The Auth controller manages user authentication processes, ensuring secure access to the application. Admin controller oversees administrative tasks, facilitating the addition or removal of hospitals from the database and user account management. The Hospital controller governs the core logic related to hospital-finding functionality, including distance calculations and retrieval of nearest hospitals. Express, a minimalist and flexible Node.js web application framework, is employed to build these controllers, providing an efficient routing system and middleware support.  
   
The Controllers in MedMap play a pivotal role in orchestrating the seamless flow of data and business logic between the Models and Views. Leveraging the Express framework, a minimalist and flexible Node.js web application framework, the Controllers efficiently manage routing and middleware support. Let's delve into the distinctive functionalities of each controller based on the provided description.

**Auth Controller:**

The Auth controller assumes a crucial role in managing user authentication processes, guaranteeing secure access to the MedMap application. It employs middleware for both user and admin authentication, ensuring a robust security layer. The admin middleware authenticates administrative users, granting them privileged access to certain functionalities. Additionally, the Auth controller handles tasks such as obtaining the current user's information, user registration, and login processes. This multifaceted controller forms a crucial part of MedMap's security infrastructure, ensuring that user and admin authentication processes are executed effectively.

 Fig 6.0.3

**Admin Controller:**

The Admin controller takes charge of administrative tasks, providing functionalities crucial for managing the database and user accounts. It includes functions to add and remove hospital data, allowing administrators to update the database dynamically. Furthermore, the Admin controller facilitates the viewing of user data, offering administrators valuable insights into user activities. Admins also possess the capability to remove users, ensuring a streamlined user management process. This controller is an integral part of MedMap's administrative functionalities, enabling efficient oversight and control over the application's data and user base.

**Hospital Controller:**

The Hospital controller governs the core logic related to hospital-finding functionality, specifically handling the retrieval of detailed hospital information. It ensures that all relevant hospital details are efficiently delivered to the client side, enhancing the user's experience when exploring healthcare options. This controller plays a pivotal role in the dynamic rendering of hospital information on the frontend, ensuring that users have access to comprehensive data about healthcare facilities. Through efficient distance calculations and data retrieval, the Hospital controller contributes to the accurate and responsive delivery of hospital details.

In conclusion, the Controllers in MedMap, including Auth, Admin, and Hospital controllers, form an integral part of the application's architecture. They serve as intermediaries between the Models and Views, facilitating smooth data flow and implementing critical business logic. The use of Express ensures an efficient routing system and middleware support, contributing to the overall effectiveness and security of the MedMap application.

**Routes: Auth, Admin, and Hospital Routes**

Routes establish the mapping between incoming HTTP requests and corresponding Controller actions. MedMap defines three route modules to handle API calls. Auth routes manage user authentication-related requests, Admin routes facilitate administrative actions, and Hospital routes handle requests related to hospital information retrieval. Express routing, integrated into the backend, ensures a logical flow of control within the application, facilitating efficient handling of diverse functionalities.

**Data Fetching: Axios Library**

In the MedMap project, the seamless interaction between the frontend and backend is made possible through the incorporation of the Axios library. Axios serves as a pivotal tool for facilitating HTTP requests, effectively bridging the gap between the View component, responsible for the user interface, and the Controller components handling the application's logic on the server side.

Axios plays a crucial role in simplifying the complex process of data exchange between the frontend and backend. It operates as a robust and user-friendly HTTP client, providing a streamlined interface for making asynchronous requests to the server. This capability proves essential for fetching data from the server and sending relevant information from the frontend to the backend, forming a cohesive and dynamic interaction between the two components.

**Middleware – authentication**

In the MedMap project, the implementation of JWT (JSON Web Token) authentication is crucial for securing access to the application's resources. The Auth middleware, where JWT authentication is applied, plays a central role in verifying the authenticity and authorization of users and administrators interacting with the application.

**JWT Authentication Overview:**

JWT is a compact, URL-safe means of representing claims to be transferred between two parties. In the context of MedMap, JWTs are utilized to securely transmit information between the frontend and backend during user authentication. When a user successfully logs in or an administrator authenticates, the server issues a JWT containing a digitally signed payload with specific user details and permissions. This token is then sent to the client, typically stored in the browser's local storage, and subsequently included in the headers of HTTP requests to the server.

**Auth Middleware in MedMap:**

The Auth middleware is a crucial segment of MedMap's server-side logic that intercepts incoming requests and verifies the attached JWT before allowing access to protected routes or functionalities. Let's elaborate on its functionalities:

**1. User Authentication:**

For user authentication, the Auth middleware ensures that the JWT received from the client is valid and has not expired. It decodes the token using a secret key known only to the server, retrieving the user's identity and any associated permissions. If the token is valid, the middleware allows the request to proceed, granting access to protected resources. If the token is invalid or has expired, the middleware denies access, prompting the client to re-authenticate.

**2. Admin Authentication:**

Admin authentication follows a similar process. The Auth middleware checks the validity of the JWT for administrators, confirming their identity and permissions. This ensures that only authorized administrators can perform certain actions, such as managing hospitals or viewing user data.

**4. Secure Routes and Actions:**

The Auth middleware is integrated into routes and actions that require authentication and authorization. For instance, accessing user profiles, making changes to user accounts, or performing administrative tasks would trigger the middleware. This guarantees that only authenticated users and administrators with the appropriate permissions can execute these actions.

**Benefits of JWT Authentication:**

Stateless and Scalable: JWTs are stateless, meaning the server doesn't need to store session information. This contributes to a more scalable architecture.

Efficient Communication: JWTs can be easily included in the headers of HTTP requests, ensuring efficient and secure communication between the frontend and backend.

**Considerations:**

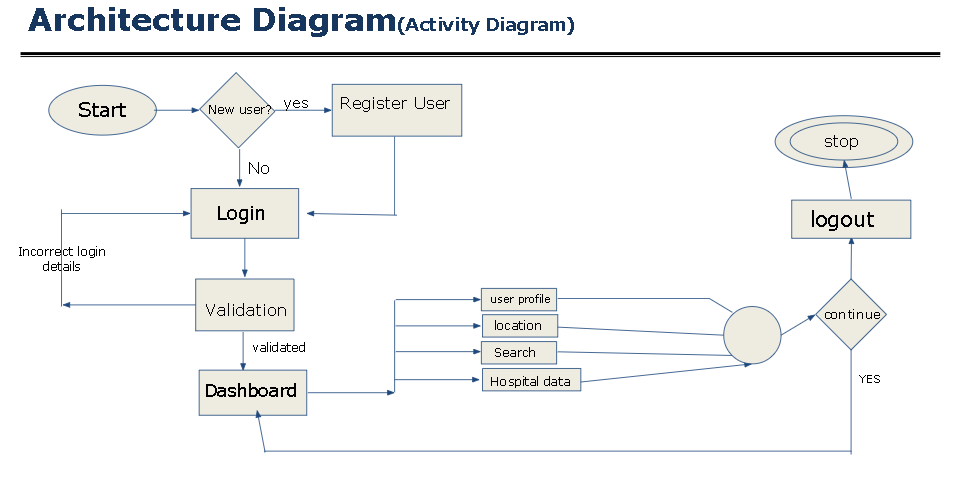
-Token Security:\*The security of JWTs relies on the secret key used for signing. Ensuring the confidentiality of this key is paramount to prevent unauthorized parties from generating fake tokens.

In summary, the Auth middleware in MedMap, powered by JWT authentication, acts as a gatekeeper, ensuring that only authenticated and authorized users and administrators can access protected resources and perform sensitive actions within the application. It plays a vital role in enhancing the security, reliability, and integrity of MedMap's user authentication processes.

**Express, MongoDB, and React Integration**

Express, MongoDB, and React form a powerful trio in MedMap's architecture. Express provides a streamlined backend infrastructure, MongoDB offers a flexible and scalable database, and React delivers an interactive and responsive frontend. The integration of these technologies results in a cohesive and efficient system where data flows seamlessly through the MVC structure, enabling MedMap to provide users with a reliable and user-friendly hospital-finding experience.

In conclusion, the MedMap project's MVC architecture, coupled with the integration of Express, MongoDB, and React, establishes a robust foundation for a scalable, maintainable, and user-centric hospital finder application. The careful orchestration of Models, Views, and Controllers, along with the efficient data communication facilitated by the Axios library, contributes to the overall success and efficiency of MedMap.

 Fig 6.0.4

**Coordinate Transformation Function:**

In the MedMap project, the convertCoordinates function plays a crucial role in translating real-world geographical coordinates (latitude and longitude) into pixel values on the digital map image. This function is essential for accurately placing the user's current location on the map interface, ensuring a seamless and visually intuitive experience.

The convertCoordinates function holds pivotal significance within the MedMap project as it serves the fundamental purpose of bridging the gap between real-world geographical coordinates and the pixel-based representation on the digital map image. This transformation is indispensable for creating a user-friendly and contextually relevant map interface. In essence, the function acts as a cartographic translator, allowing seamless integration of real-world locations into the digital realm, thereby enhancing the overall user experience.

In geographical terms, latitude and longitude define precise points on the Earth's surface. However, when translating these coordinates to a digital map, the process involves transforming continuous geographical data into discrete pixel values that make up the visual representation. The conversion facilitated by convertCoordinates ensures that the user's current location is accurately positioned within the specified digital map bounds.

One compelling reason for this conversion lies in the disparity between the scale and units of measurement in the real world versus the digital space. Geographical coordinates are continuous and measured in degrees, whereas pixel values on a digital map are discrete and measured in units such as pixels. The conversion process harmonizes these differences, allowing for a cohesive and visually coherent mapping of real-world locations onto the digital canvas.

Moreover, the pixel-based representation is integral for rendering a visually intuitive map interface. By accurately placing the user's current location on the map, the function facilitates a seamless understanding of spatial relationships and distances within the digital environment. Users can easily interpret their surroundings and navigate the map with confidence, fostering a user experience that is both informative and engaging.

In conclusion, the conversion of real-world coordinates by the convertCoordinates function is not merely a technical necessity but a strategic decision aimed at providing users with a map interface that seamlessly merges the precision of geographical coordinates with the visual clarity of digital mapping. This transformation is the cornerstone of MedMap's commitment to delivering an intuitive and user-centric hospital-finding experience.

**Parameters**:

actualMapCoordinates: An array representing the latitude and longitude of a location in the real world.

actualMapBounds: An array specifying the geographical bounds of the actual map coordinates [latitude1, longitude1, latitude2, longitude2].

targetMapBounds: An array indicating the pixel bounds of the target map coordinates [x1, y1, x2, y2].

**Description:**

The function begins by extracting the individual coordinates from the provided inputs i.e. latitude and longitude retrieved from the geolocation of the user and is stored in a variable to be converted. It then proceeds to perform a linear transformation on these coordinates, effectively mapping them from the actual geographical bounds to the pixel values of the target map image. The function holds two formulas shown below and uses both the bounds of the actual map and digital one to normalize actual coordinate values to digital map pixel coordinates.

**Linear Transformation:**

The linear transformation is executed using the following formulas:

let targetX = (x - ax1) / (ax2 - ax1) \* (tx2 - tx1) + tx1;

let targetY = (y - ay1) / (ay2 - ay1) \* (ty2 - ty1) + ty1;

Where:

x, y: Current latitude and longitude of the location.

ax1, ay1, ax2, ay2: Actual geographical bounds.

tx1, ty1, tx2, ty2: Target pixel bounds on the map image.

These formulas ensure a proportional and accurate conversion, effectively placing the real-world coordinates onto the digital map image.

**Return Value:**

The function returns an array containing the transformed coordinates [targetX, targetY], representing the pixel values on the map image. This returned array is then used to locate and visually represent the person’s location on the map on the home page of this webapp.

**Usage:**

let actualCoordinates = [userLatitude, userLongitude];

let transformedCoordinates = convertCoordinates(actualCoordinates, actualMapBounds, targetMapBounds);

In summary, the convertCoordinates function serves as a crucial component in the MedMap project, enabling the seamless integration of real-world locations onto the digital map interface, providing users with an accurate representation of their surroundings. It also helps in further features of the webapp where we have to represent hospital locations on the map when we are given the real world coordinates.

**Hospital Proximity Calculation: closestFive**

The closestFive function in the MedMap project plays a pivotal role in enhancing the user experience by precisely determining and presenting the five closest hospitals relative to the user's current location. This functionality is particularly crucial in the context of a hospital finder application, as it empowers users to quickly access critical healthcare services based on their proximity.

The core strategy employed by closestFive involves leveraging the Haversine formula, a mathematical formula that accounts for the curvature of the Earth when calculating distances between two sets of geographical coordinates. This formula is especially well-suited for applications that require accurate distance measurements over the Earth's surface. By employing the Haversine formula, MedMap ensures that the distances calculated are not only precise but also account for the spherical nature of the Earth, providing users with reliable information about the proximity of hospitals.

The function initiates by creating an array, distances, which serves as a repository for the calculated distances between the user's location and each hospital in the provided dataset. This information is not only used for identifying the closest hospitals but is also instrumental in offering users a ranked list based on proximity.

As the function iterates through the dataset of hospitals, it invokes the getDistance function for each hospital, generating accurate distance measurements. The gathered information is then structured into an array of objects, each containing essential details about the hospital, including its ID, name, latitude, longitude, and the calculated distance from the user.

Furthermore, the distances array is sorted in ascending order based on the calculated distances. This sorting mechanism ensures that the user is presented with a clear and concise list of the five nearest hospitals, facilitating quick decision-making and minimizing search time.

The utilization of the Haversine formula, combined with the thoughtful organization of data, underscores the commitment of MedMap to providing users with a highly effective hospital-finding experience. By presenting the closest hospitals based on accurate distance calculations, MedMap not only optimizes the user's ability to access healthcare promptly but also contributes to an overall positive and reliable application experience.

**Description:**

The function initializes an array, distances, to store calculated distances between the user's location and each hospital in the provided dataset. It iterates through the hospitals, invoking the getDistance function for each, and populates the distances array with relevant information such as hospital ID, name, latitude, longitude, and distance from the user.

**Distance Calculation and Sorting:**

The distances are computed using the Haversine formula, which takes into account the curvature of the Earth for more accurate results. Subsequently, the array is sorted in ascending order based on the calculated distances, ensuring the nearest hospitals appear first.

**Return Value:**

The function returns an array containing information about the five closest hospitals, including their ID, name, latitude, longitude, and distance from the user's location.

**Haversine Distance Calculation**

The getDistance function serves as the core component for calculating distances between geographical coordinates using the Haversine formula.

The Haversine formula is a mathematical equation used to determine the great-circle distance between two points on the surface of a sphere, such as the Earth. The formula is expressed as follows:

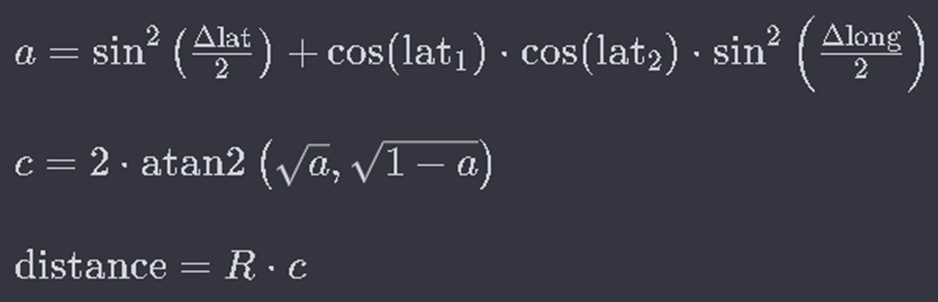


Fig 6.0.5

1. Δlat is the difference in latitude between the two points,
2. Δlong is the difference in longitude between the two points,
3. lat1 and lat2 are the latitudes of the two points,
4. a is the square of half the chord length between the points on the unit circle,
5. c is the angular distance in radians,
6. R is the Earth's radius (mean radius = 6,371 kilometers).

**Parameters**:

lat1, long1, lat2, long2: Latitude and longitude coordinates for two locations.

**Description**:

This function converts the latitude and longitude from degrees to radians and then employs the Haversine formula to compute the distance between two points on the Earth's surface. The result is a precise distance value in kilometers.

**Return Value:**

The function returns the calculated distance between the two sets of coordinates.

In summary, the closestFive and getDistance functions collectively provide a robust solution for determining the five nearest hospitals to the user's location. This functionality enhances the MedMap project by offering users an efficient and personalized means of accessing vital healthcare services based on their proximity.

**Dynamic Rendering of User's Position**

Upon accessing the MedMap application, the user's current location is obtained using the React Geolocation feature. The latitude and longitude coordinates obtained from the user's device are then fed into the convertCoordinates function, transforming these real-world geographical coordinates into pixel values that can be accurately plotted on the digital map image. This conversion ensures a precise mapping of the user's position onto the map, allowing for an accurate representation of their location in the digital space.

The dynamically rendered user position is not static but updates in real-time as the user moves. This constant synchronization between the user's movements and the digital map provides an accurate and up-to-date reflection of their current location, enhancing the overall usability of the application.

**Dynamic Rendering of Closest Hospitals**

Simultaneously, the closestFive function is invoked, calculating the distances between the user's location and available hospitals using the Haversine formula. The five hospitals with the shortest distances are then dynamically rendered on the map interface.

Each hospital's geographical coordinates, obtained from the dataset, undergo the same convertCoordinates transformation to ensure consistent and accurate positioning on the digital map. The dynamically rendered hospital markers are strategically placed to visually communicate their proximity to the user's position.

To further enhance user engagement and understanding, additional information about these hospitals, such as their names and distances, can be displayed alongside the corresponding map markers. This approach not only aids in quick identification of nearby healthcare facilities but also provides users with vital details to make informed decisions about their healthcare options.

**CHAPTER-7**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**

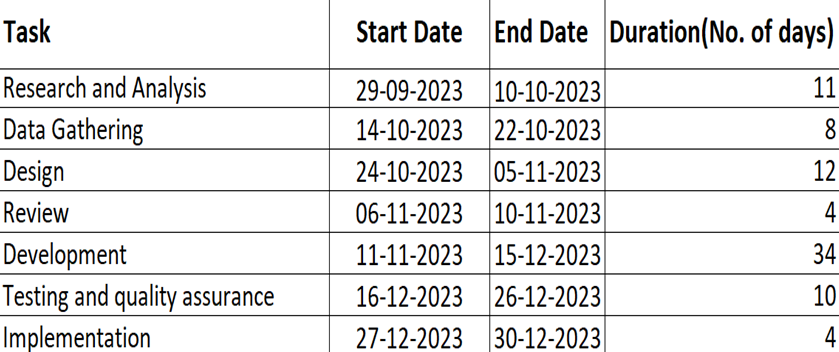


Fig 7.0.1

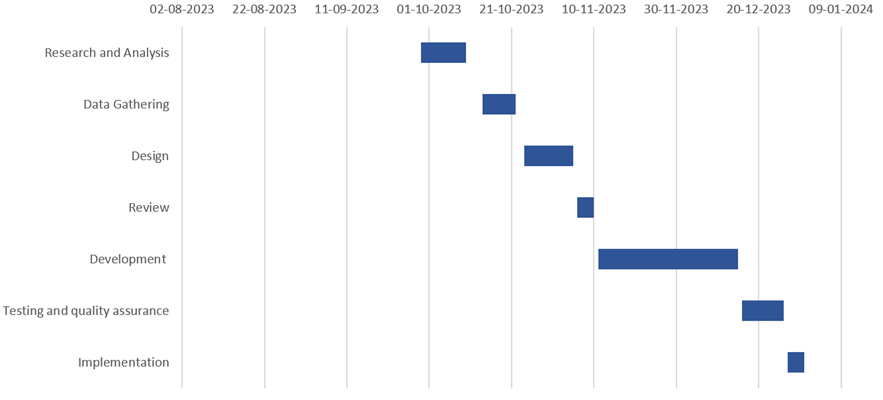


Fig 7.0.2

**CHAPTER-8**

**OUTCOMES**

**Improved Informed Decision-Making:**

MedMap serves as a valuable tool for users, providing comprehensive details about various healthcare facilities to empower them in making well-informed decisions about their healthcare choices. The platform facilitates a thorough understanding of the services offered by different facilities, enabling users to assess and compare options based on their specific needs.

**Streamlined Information Retrieval:**

The platform facilitates quick and efficient retrieval of information regarding hospitals, clinics, and medical centers. Users can easily access contact details, addresses, and relevant information based on their geographical location, ensuring seamless connectivity to medical services.

**Contribution to Healthcare Transformation:**

MedMap sets a role for ongoing transformation of healthcare by harnessing technological innovations to elevate accessibility and streamline patient interactions within the healthcare ecosystem. MedMap aligns seamlessly with the overarching objective of enhancing user-friendliness, operational efficiency, and responsiveness to individual needs within healthcare services.

**Comprehensive Medical Records Management:**

The Medical Records feature offers users a secure platform to access and manage their comprehensive medical records online. This functionality allows users to securely store test results, prescriptions, and health history, enabling easy reference to health-related information. Moreover, users can securely share these records with healthcare professionals as needed, thereby facilitating better-informed consultations and treatments for improved care delivery.

**CHAPTER-9**

**RESULTS AND DISCUSSIONS**

**Results:**

The ‘Find a Hospital’ function plays a pivotal role in improving healthcare accessibility, enabling users to easily find nearby hospitals, clinics, and medical facilities. This feature not only furnishes comprehensive details about the services offered by these healthcare providers but also empowers users to make informed decisions regarding their healthcare choices. Furthermore, it facilitates seamless access to contact details and addresses based on the user's geographical location, simplifying the process of reaching out for medical assistance.

An integral aspect of the application providing immediate access to crucial contact information during critical situations. Users can efficiently pinpoint nearby emergency departments or urgent care facilities, obtaining real-time information and guidance for seeking urgent medical attention. This functionality ensures that individuals in emergency situations can swiftly receive vital medical assistance, potentially mitigating adverse health outcomes and saving lives.

**DISCUSSIONS**

The MedMap web application presents a promising solution in the healthcare sector, demonstrating both practical functionality and a user-centric experience. In the following discussion, we will delve into key aspects of MedMap, including its innovative features, potential benefits, and considerations for future enhancements.

MedMap's standout feature lies in its ability to swiftly locate the nearest hospital within a specific specialization, offering a critical advantage in emergency situations. The integration of Smartphone GPS and Google Maps enhances the precision and usability of the application, providing users with detailed information such as hospital names, addresses, and routes. The real-time display of distances and routes contributes to efficient decision-making during critical moments.

Medmap not only delivers outcomes in improving healthcare accessibility but also sparks conversations about the future of patient-centric care, ethical considerations in healthcare technology, and the democratization of fundamental healthcare services. This creates its impact not just in practical terms but also in shaping broader discussions and considerations within the healthcare landscape.

**CHAPTER-10**

**CONCLUSION**

The proposed web application, MedMap, stands as a comprehensive, open, and freely accessible platform designed to deliver powerful functionality and an excellent user experience. This introduction provides an in-depth overview of the web application framework and outlines the fundamental principles guiding its operation. The emphasis is placed on its capability to identify nearby locations through the web platform, elucidating the underlying mechanism.

The research presented in this work introduces a novel service-based design approach, focusing on the implementation of two popular consumer applications on the web. Notably, one of these applications is dedicated to utility search places for clients.

This study primarily aims to design a basic and up-to-date medical category application, specifically tailored to assist patients and caregivers in identifying the nearest hospital with a particular specialization. Leveraging the geolocation service within the application dynamically determines hospital names, addresses, and routes. Integration with Google Maps enables the display of distance and routes to each hospital, proving particularly beneficial in emergency situations and for non-residents in unfamiliar cities. The future trajectory of this application involves real-time tracking of specialist doctor availability near the patient's location, with a prospective feature for online appointment booking, optimizing time for patients.

**REFERENCES**

1. Yeo, S., Sankaranarayanan, S., & Sait, S. (2013). "Application of Smart Technologies for Mobile Patient Appointment System." \*International Journal of Advanced Trends in Computer Science and Engineering\*, August.

2. Deceuninck, M., Fiems, D., & De Vuyst, S. (2018). "Outpatient scheduling with unpunctual patients and no-shows." \*European Journal of Operational Research\*, 265(1), 195–207.

3. Benzaid, M., Lahrichi, N., & Rousseau, L. M. (2020). "Chemotherapy appointment scheduling and daily outpatient nurse assignment." \*Health Care Management Science\*, 23(1), 34–50.

4. Khan, M. A., Muhammad, K., Sharif, M., Akram, T., & Kadry, S. (2021). "Intelligent fusion-assisted skin lesion localization and classification for smart healthcare." \*Neural Computing and Applications\*, 8, 1–16.

5. Hylton III, A., & Sankaranarayanan, S. (2012). "Application of Intelligent Agents in Hospital Appointment Scheduling System." \*International Journal of Computer Theory and Engineering\*, 4, 625-630.

6. Diamant, A., Milner, J., & Quereshy, F. (2018). "Dynamic patient scheduling for multi-appointment healthcare programs." \*Production and Operations Management\*, 27(1), 58–79.

7. Garaix, T., Rostami, S., & Xie, X. (2020). "Daily outpatient chemotherapy appointment scheduling with random deferrals." \*Flexible Services and Manufacturing Journal\*, 32(1), 129–153.

8. Guido, R., Ielpa, G., & Conforti, D. (2020). "Scheduling outpatient day service operations for rheumatology diseases." \*Flexible Services and Manufacturing Journal\*, 32(1), 102–128.

9. Shehadeh, K. S., Cohn, A. E. M., & Jiang, R. (2021). "Using stochastic programming to solve an outpatient appointment scheduling problem with random service and arrival times." \*Naval Research Logistics\*, 68(1), 89–111.

10. Almomani, I., & Alsarheed, A. (2016). "Enhancing outpatient clinics management software by reducing patients’ waiting time." \*Journal of infection and public health\*, 9(6), 734–743.

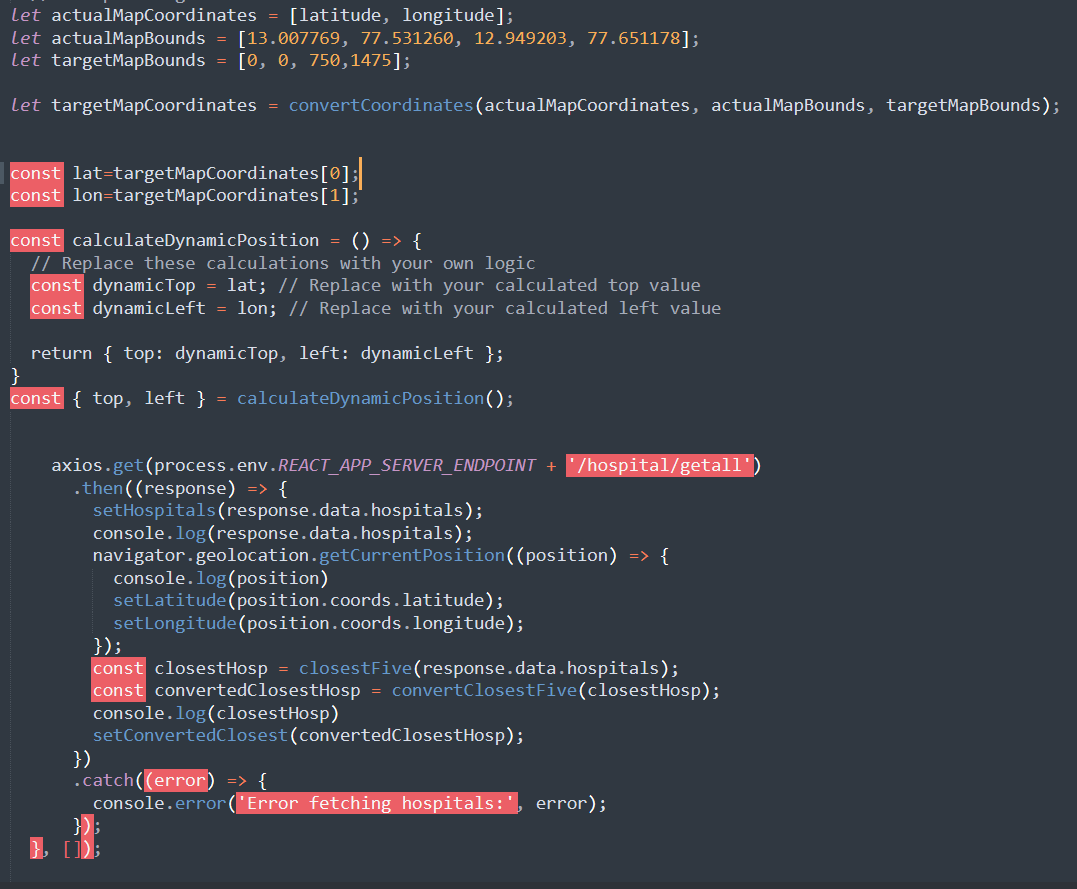
11. Aghav, J., Sonawane, S., & Bhambhlani, H. (2014). "HealthTrack: Health Monitoring and Prognosis System using Wearable Sensors." \*IEEE International Conference on Advances in Engineering & Technology Research\*, 15.

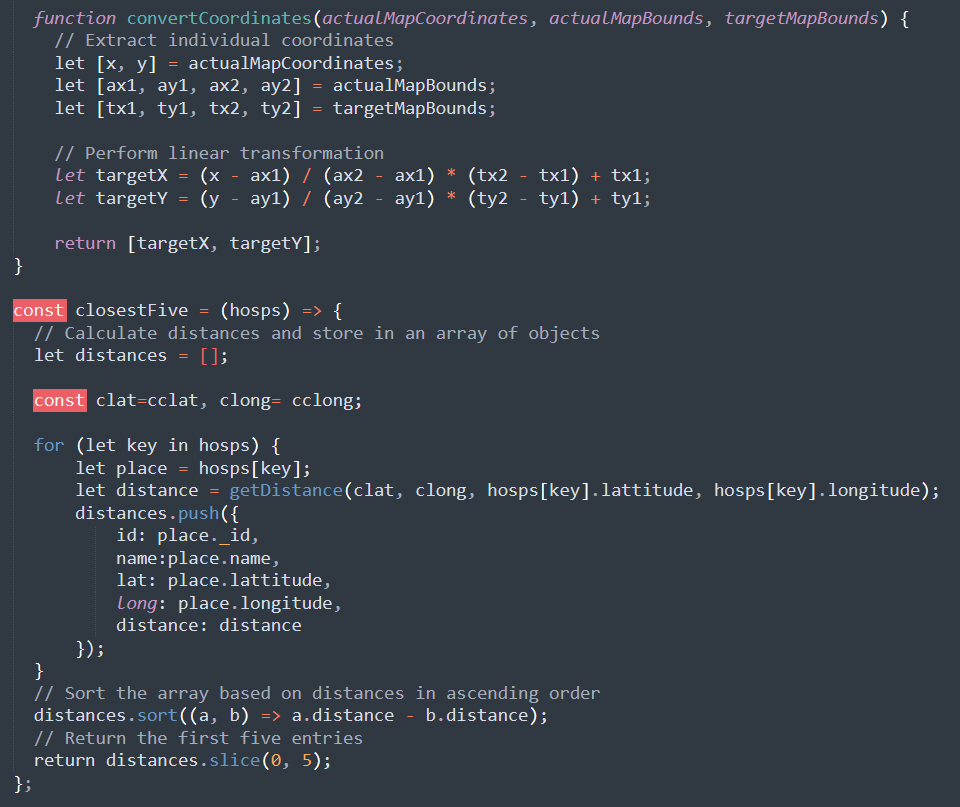
12. Radman, M., & Eshghi, K. (2018). "Designing a multiservice healthcare network based on the impact of patients’ flow among medical services." \*Spectrum\*, 40(3), 637–678.

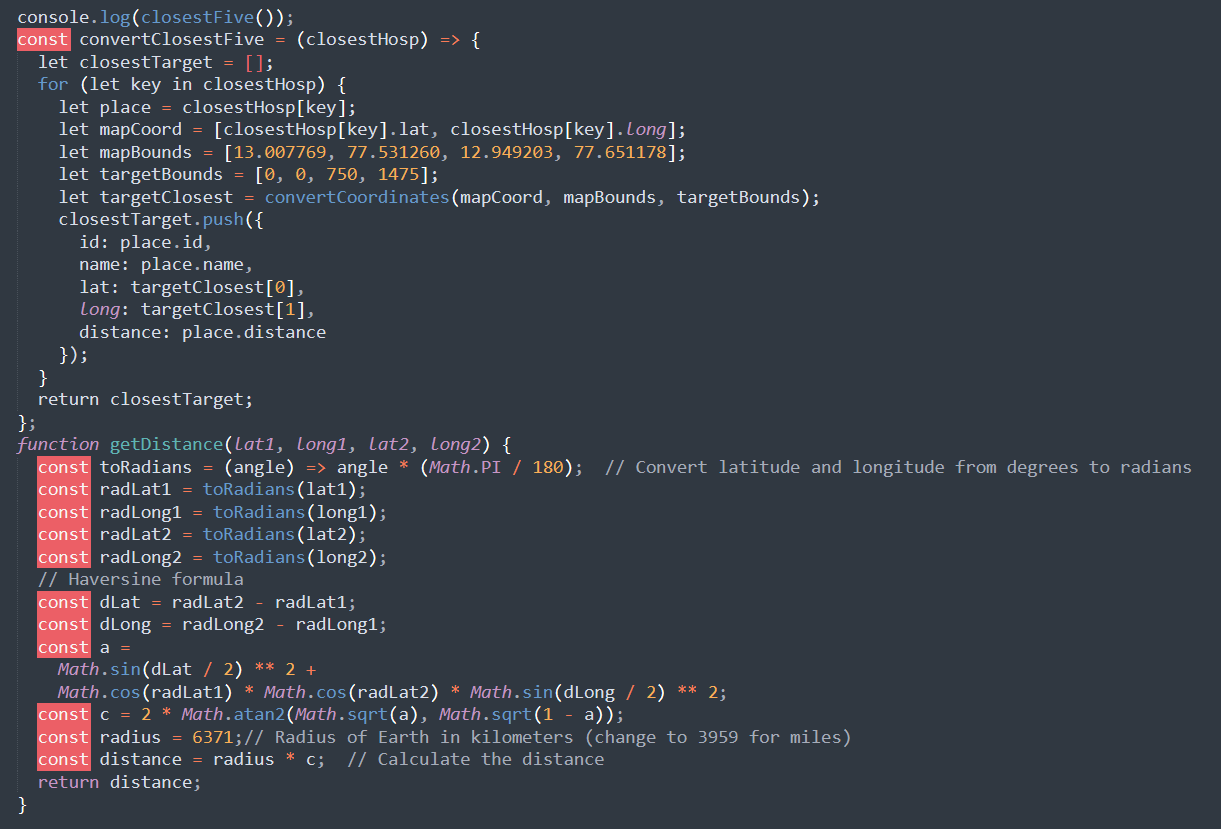
13. Konrad, R., Ficarra, S., Danko, C., Wallace, R., & Archambeault, C. (2017). "A decision support approach for provider scheduling in a patient-centered medical home." \*Journal of Healthcare Management\*, 62(1), 46–59.

**APPENDIX-A**

**PSUEDOCODE**

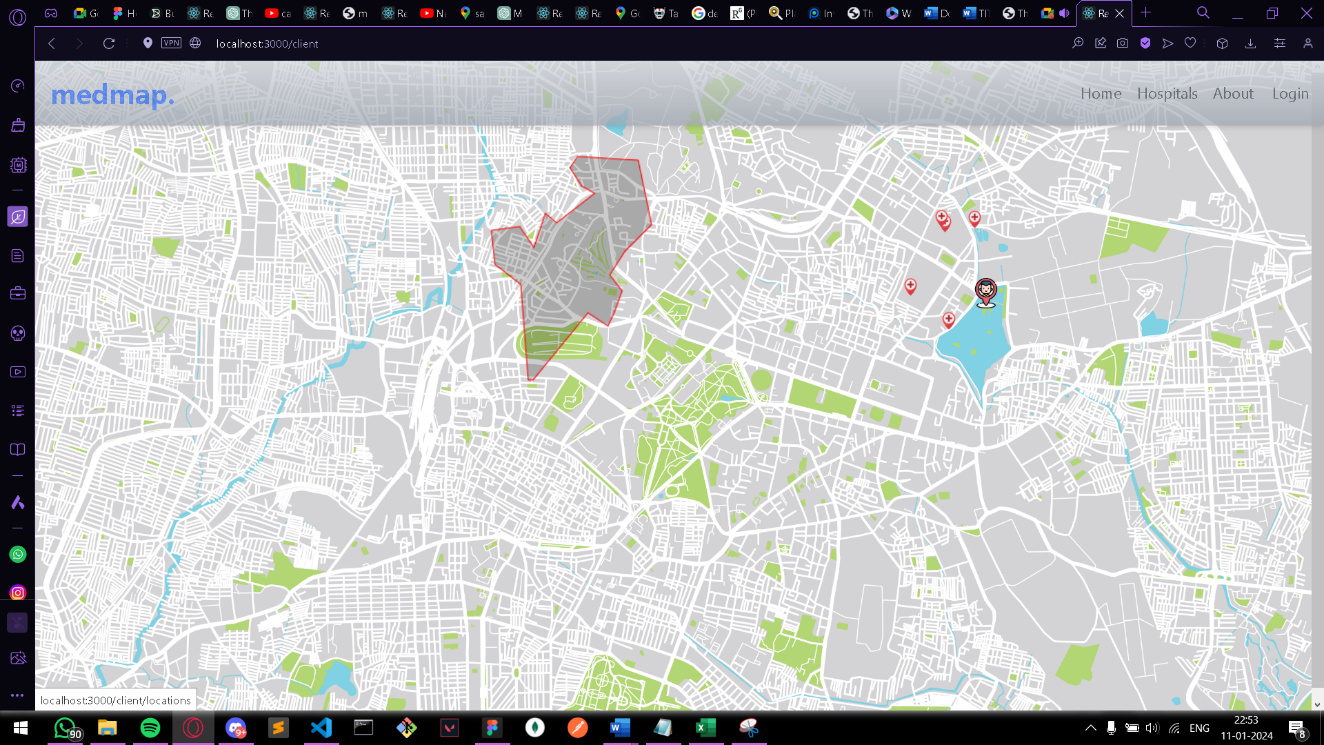


Fig



**APPENDIX-B**

**SCREENSHOTS**

 Fig c.0.1

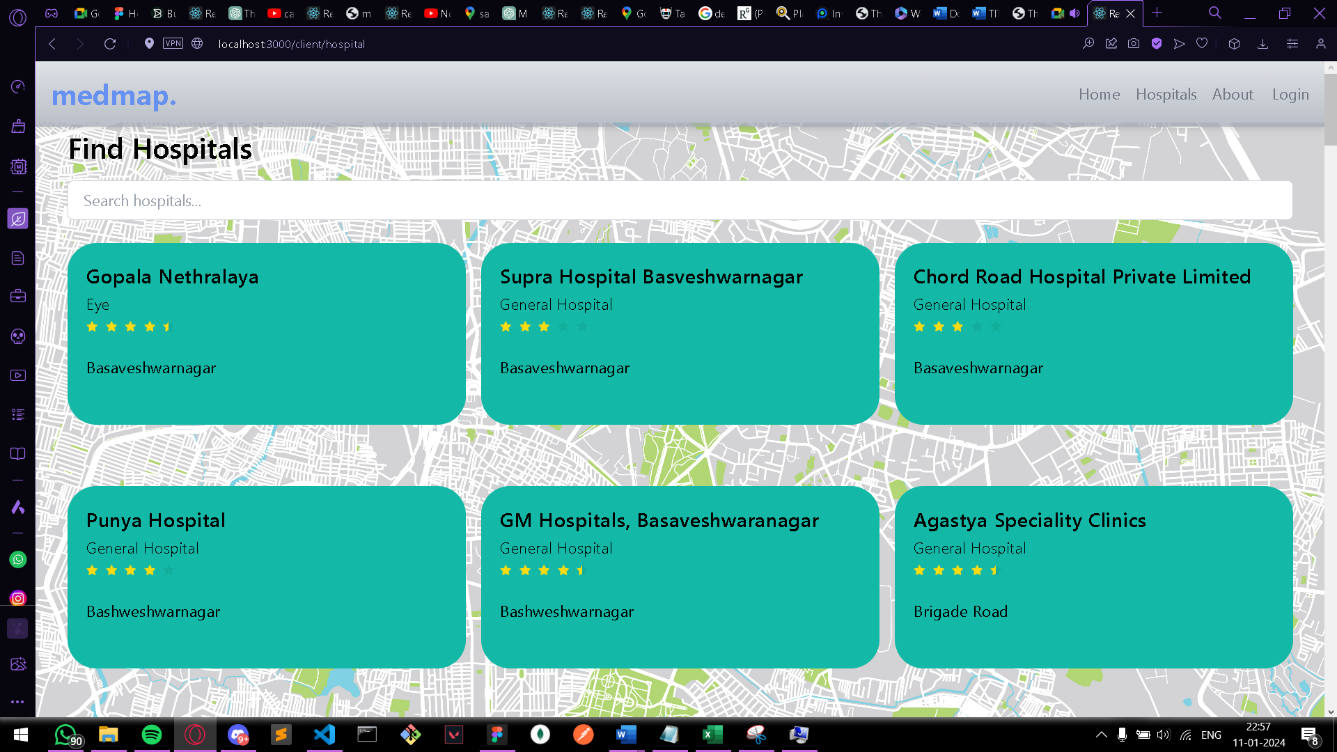


Fig c.0.2

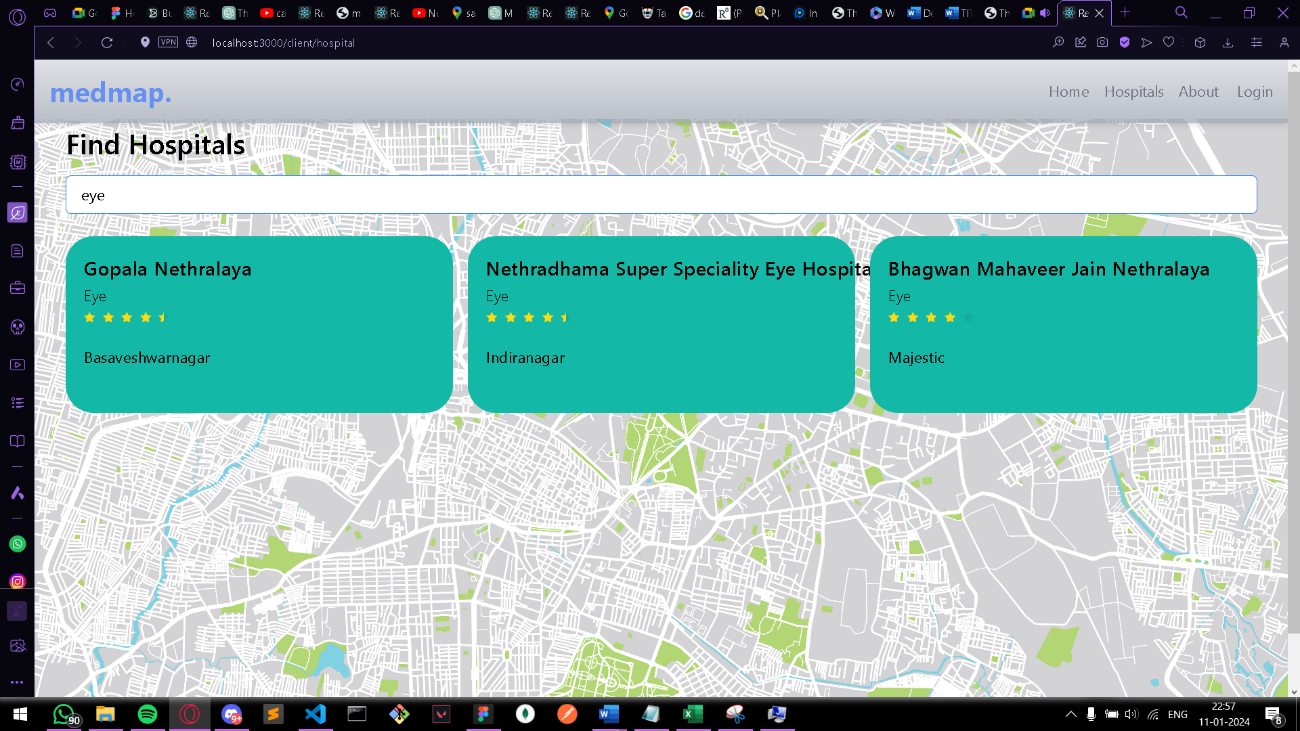


Fig c.0.3



Fig c.0.4

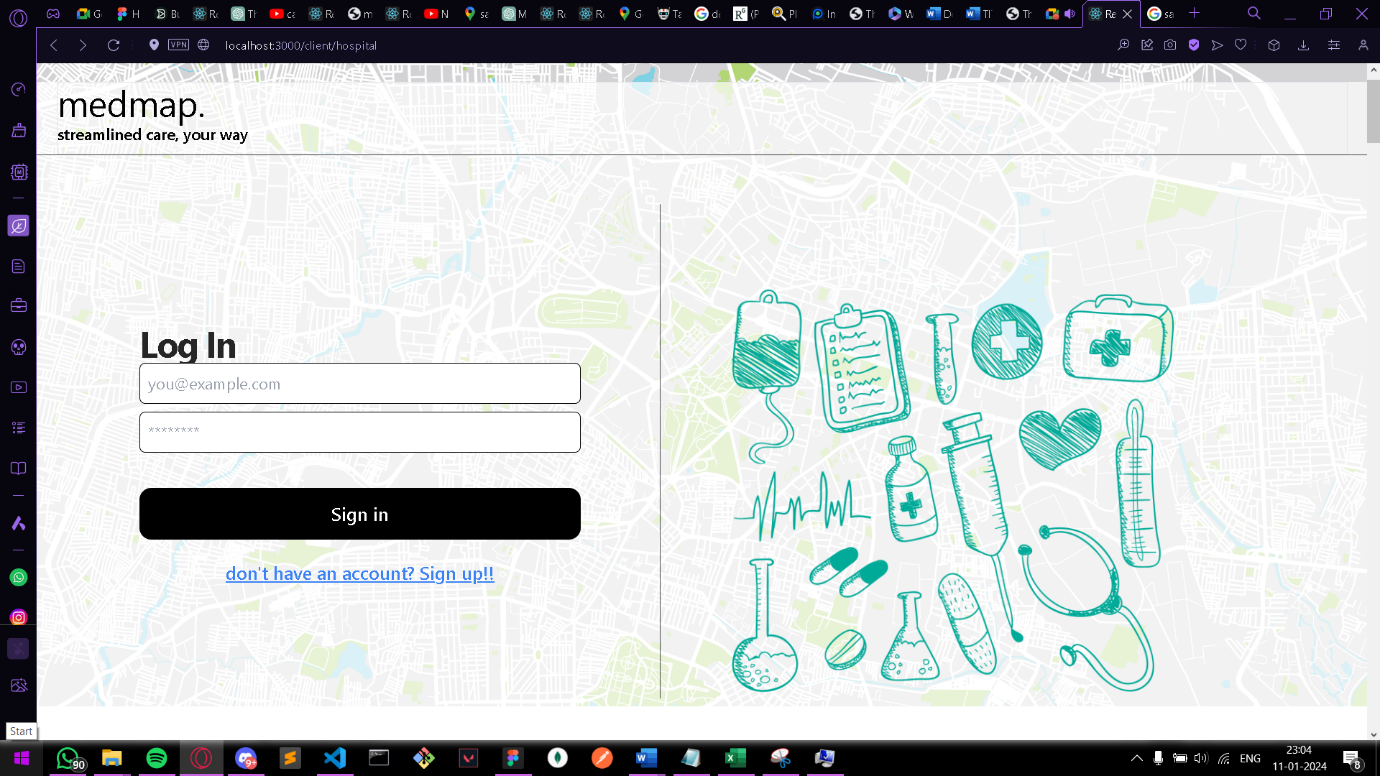


Fig c.0.5



Fig c.0.6

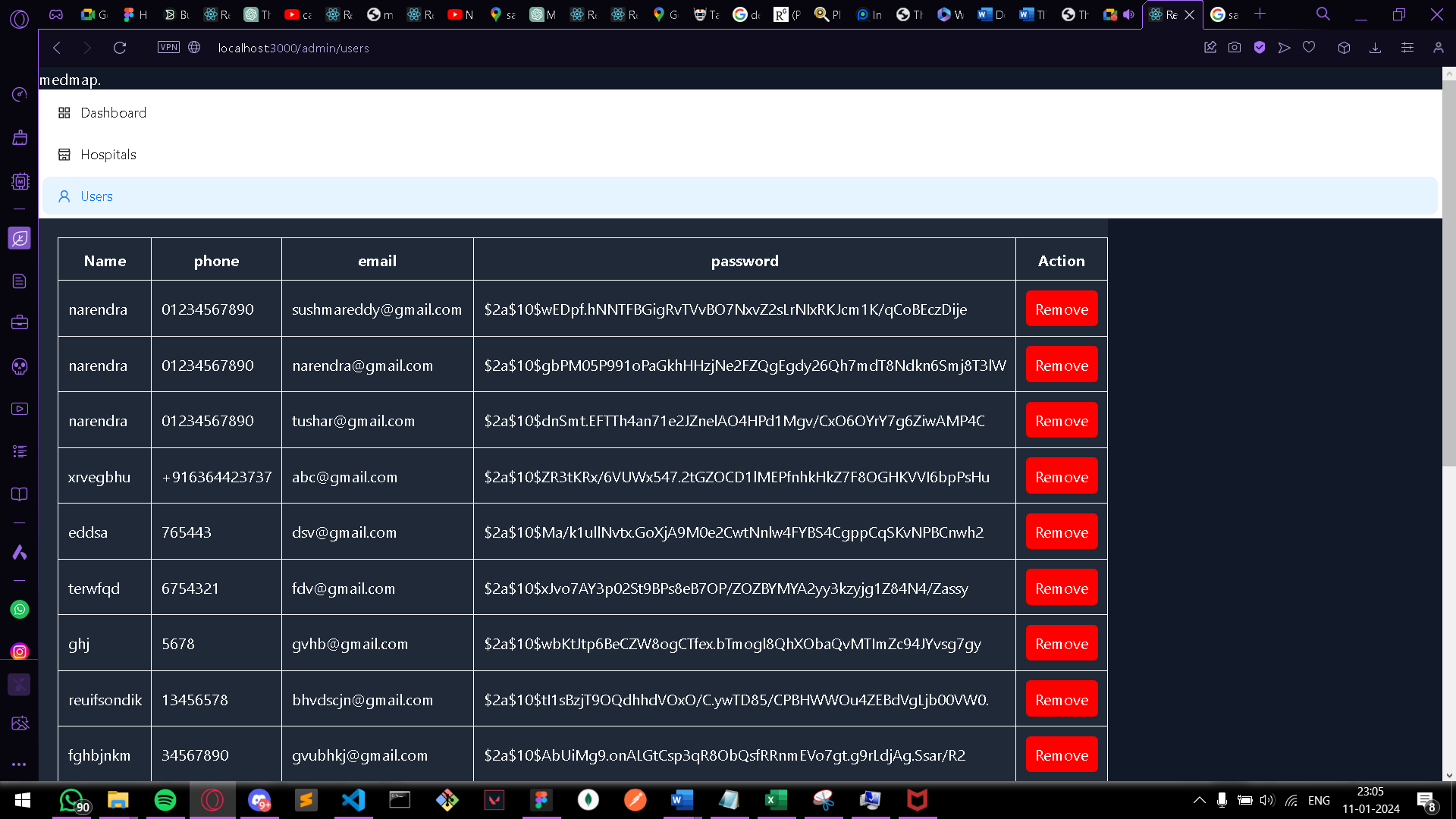


Fig c.0.7

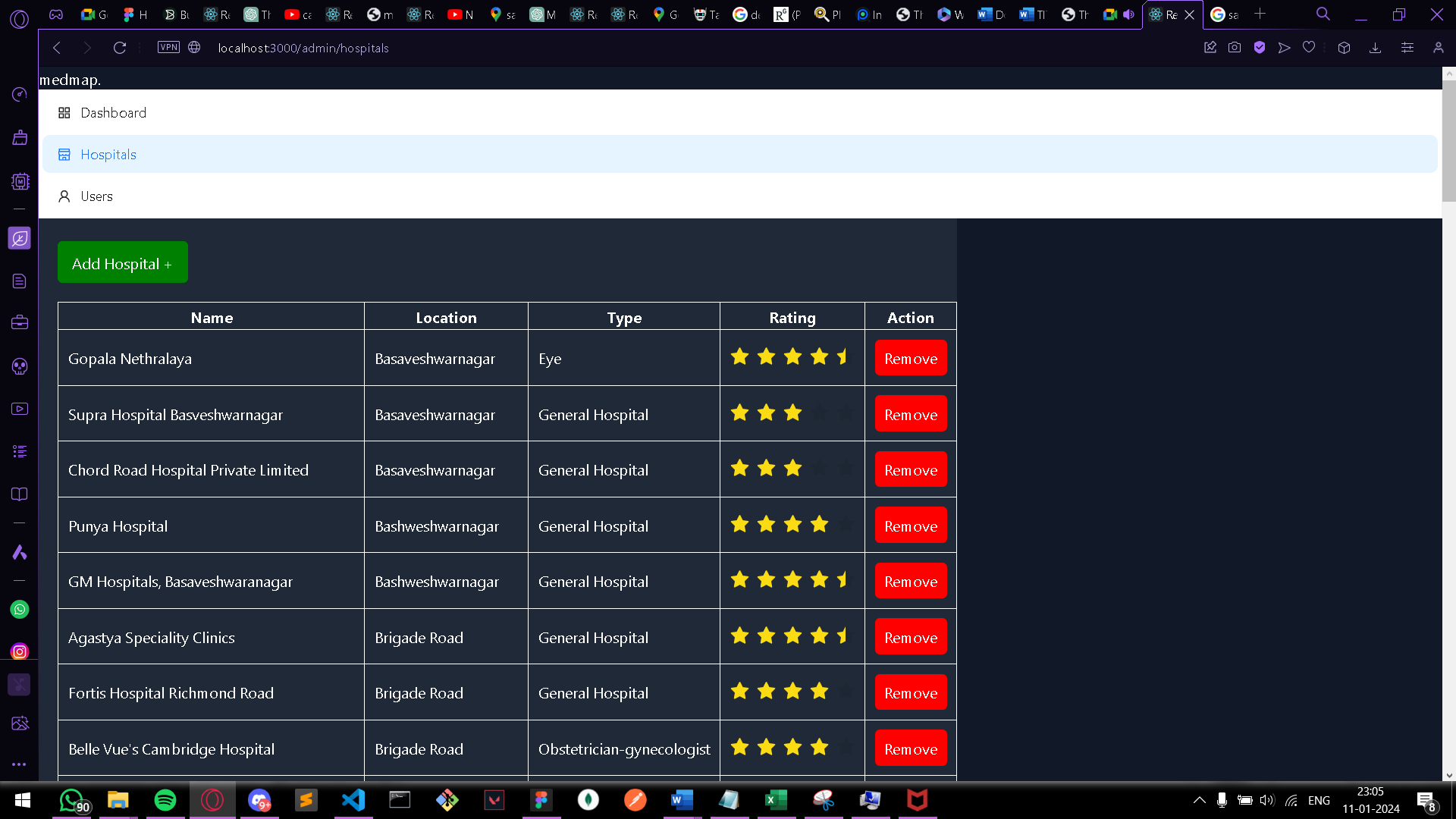


Fig c.0.8