MEDISEARCH

A PROJECT REPORT

Submitted by,

Kaousthub Reddy H	-	**20201CCS0072**
Aakash Adhikari	-	**20201CCS0085**
Tushar Patel	-	**20201CCS0114**
Mohammed Faisal Khan	-	**20201CCS0122**
Md Zohaer Anfaz M.A	-	**20201CCS0149**

Under the guidance of,

Dr. Nagaraja S R

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

CERTIFICATE

This is to certify that the Project report **"MediSearch"** being submitted by "Aakash Adhikari, Kaousthub Reddy H, Tushar Patel, Mohammed Faisal Khan, Md Zohaer Anfaz M.A" bearing roll number(s) "20201CCS0085, 20201CCS0072, 20201CCS0114, 20201CCS0122, 20201CCS0149" in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security) is a bonafide work carried out under my supervision.

Dr. Nagaraja S R
Associate Professor
School of CSE&IS
Presidency University

Dr. S.P. Anandaraj
Professor & HoD
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. SHAKKEERA L
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 **MediSearch** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering (Cyber Security),** is a record of our own investigations carried under the guidance of **Dr. Nagaraja S R , Associate 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.

Kaousthub Reddy H	**20201CCS0072**
Aakash Adhikari	**20201CCS0085**
Tushar Patel	**20201CCS0114**
Mohammed Faisal Khan	**20201CCS0122**
Md Zohaer Anfaz M.A	**20201CCS0149**

ABSTRACT

Hospital Finder is an innovative web application designed to tackle the daunting challenge of selecting the right hospital during medical emergencies. Leveraging Machine Learning, it simplifies the process of locating nearby hospitals based on specific medical needs, empowering individuals to make well-informed decisions. By integrating advanced technology into healthcare, Hospital Finder not only identifies the closest hospitals but also guides users to the most suitable options for their unique requirements. This revolutionary solution transforms healthcare access, offering precise, real-time information to potentially save lives, ease anxiety, and restore control to those facing emergencies. It marks a significant advancement in healthcare navigation, ensuring individuals access the right care when it's most critical. Hospital Finder's features and functionality illustrate its potential to revolutionize decision-making in seeking medical care.

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. S.P. Anandaraj**. 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. Nagaraja S R**, Associate Professor, 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 Co-ordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Co-ordinator **Ms. Manasa C M**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Kaousthub Reddy H - 20201CCS0072
Aakash Adhikari - 20201CCS0085
Tushar Patel - 20201CCS0114
Mohammed Faisal Khan - 20201CCS0122
Md Zohaer Anfaz M.A - 20201CCS0149

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

In the context of healthcare emergencies, timely access to real-time hospital data plays a crucial role in making informed decisions promptly. Despite its importance, challenges persist in obtaining up-to-date information during critical situations. This study introduces "Hospital Finder," an innovative application designed to tackle this issue by utilizing a centralized real-time database. Hospital Finder offers users instant and precise information on hospital facilities, specialist availability, and essential medical resources. The scarcity of real-time data has been a persistent obstacle in medical emergencies. Hospital Finder serves as a solution by leveraging a centralized real-time database to overcome challenges associated with obtaining timely information from healthcare facilities. The goal of this application is to empower individuals, healthcare providers, and emergency responders with the most relevant information during critical moments. As we delve into the details of Hospital Finder, this paper will explore its key features, architectural design (including a use case diagram), and recommendations for incorporating real-time data into healthcare systems. Recognizing potential challenges, alternative approaches to Hospital Finder will be discussed, providing a comprehensive overview of strategies to improve access to emergency healthcare. The section on evaluation and user testing will present insights into the application's performance. The conclusion will highlight key findings, emphasizing the transformative potential of Hospital Finder in reshaping healthcare access and decision-making during emergencies.

CHAPTER-2
LITERATURE SURVEY

2.1 Google Maps and Navigation Technologies

2.1.1 Historical Evolution of Mapping Technologies:

The progression of mapping technologies has been a pivotal element in the digital age, and Google Maps has become a noteworthy contributor to this domain. The article chronicles the historical advancement of Google Maps, underscoring its inception through the acquisition of 'Where 2 Technologies' by Google Inc. in 2004. This historical context aligns with the broader body of literature that recognizes the transformative impact of digital mapping applications in redefining navigation and location-based services.

2.1.2 Algorithms in Navigation:

The manuscript explores the algorithms utilized by Google Maps in navigation, particularly emphasizing Dijkstra's algorithm and the more efficient A* algorithm. This discourse on these algorithms corresponds with established literature on graph-based algorithms for optimizing routes. The examination of Dijkstra's algorithm and its drawbacks, attributed to heightened time and space complexity, echoes findings from previous studies, underscoring the necessity for more advanced algorithms in extensive applications such as Google Maps.

2.1.3 Street View and Image Processing:

The paper highlights the significance of introducing Google Maps' Street View feature. The incorporation of panoramic views into mapping applications is acknowledged in existing literature as a pioneering innovation. The manual capturing and processing of images, along with the utilization of Optical Flow to tackle issues such as parallax distortion, are consistent with prior research on image processing techniques in geospatial applications.

2.1.4 Geocoding and GPS Tracking:

The manuscript provides detailed insights into the function of GPS and geocoding for position identification on Google Maps. The application of Trilateration as a mathematical concept for GPS tracking corresponds with the wider body of literature on satellite-based tracking systems. The integration of WAAS (Wide Area Augmentation System) aligns with existing research emphasizing the significance of augmentation systems in improving the precision of location services based on GPS.

2.1.5 Estimated Time of Arrival (ETA) and Traffic Prediction:

The examination of estimating the time of arrival (ETA) on Google Maps is contextualized within the framework of the A* algorithm and real-time traffic information. The incorporation of multiple factors, including average speed, historical data, and officially recommended speeds, aligns with literature addressing the intricacies of forecasting travel times in dynamic settings. The focus on ongoing data collection and analysis corresponds with research underlining the significance of real-time data for precise traffic predictions.

2.1.6 Conclusion:

To sum up, the review of existing literature underscores the diverse facets of Google Maps, covering its historical development, algorithmic strategies, image processing methodologies, geolocation technologies, and real-time traffic forecasting. The amalgamation of these components characterizes Google Maps as a dynamic and advanced mapping tool, making substantial contributions to the broader conversation surrounding digital navigation technologies. Future studies might explore user perspectives, privacy considerations, and emerging trends in mapping applications to offer a more comprehensive insight into the field.

2.2 A smartphone based application to improve the health care system · December 2016

Healthcare is an essential requirement for every individual, and the global healthcare landscape presents both challenges and opportunities. Public healthcare systems, typically government-provided with minimal or no charges, face complexities in addressing the needs of a large patient population. To meet the increasing demand for quality healthcare, numerous private hospitals have been established. However, accessing relevant information about these facilities remains a challenge, particularly for individuals from rural areas seeking better services in urban locales.

To tackle these challenges and improve healthcare convenience for the masses, this paper suggests a web application with a comprehensive set of features. The choice of the Android platform is justified by its widespread use, affordability, and accessibility across various age groups and socioeconomic classes.

The proposed web application aims to offer the following features:

1. **Online Cabin Booking System:** Facilitating users in booking hospital cabins in advance.
2. **Intelligent Hospital Suggestions:** Providing recommendations based on both cost and quality metrics to assist users in selecting suitable hospitals.
3. **Hospital Information:** Offering details about the facilities and locations of hospitals.
4. **Doctor's Chamber Information:** Providing information about doctors in a city, allowing users to make appointments.
5. **Emergency Assistance:** Enabling users to make emergency calls for ambulances or healthcare services.
6. **Medicine Reminder System:** Implementing an alert system to remind users to take prescribed medicines at specific times.
7. **Body Mass Index (BMI) Calculator:** Helping users assess their BMI for health and fitness monitoring.

2.2.1 System Implementation:

The system architecture consists of two primary modules catering to administrators and general users. The administrator module facilitates the creation and modification of hospital and cabin details. On the other hand, the user module offers functionalities like online cabin booking, hospital recommendations, doctor appointments, emergency calls, medicine reminders, BMI calculation, and additional features. The development of the web application is executed on the Android OS due to its cost-effectiveness and extensive adoption.

2.2.2 Experimental Results:

Information for the application was gathered from ten well-known hospitals within a particular city. The objective of the application is to furnish users with precise and pertinent details, enabling them to make well-informed decisions when selecting a hospital. The assessment of quality includes evaluating factors such as the quantity of cabins, doctors, specialist physicians, and operating theatres. Feedback from a survey of 150 users indicated favourable opinions regarding the application's capability to tackle healthcare-related issues and its overall utility.

2.2.3 Conclusion:

The suggested web application acts as an all-encompassing tool to improve healthcare accessibility and convenience for users. User feedback suggests a significant perceived utility. Prospective enhancements could involve integrating artificial intelligence for symptom-based disease detection, further elevating the application's impact on daily life.

2.3 Advances in Hospital Real-Time Location Systems:

The healthcare environment is undergoing rapid changes, placing a growing emphasis on the significance of monitoring the whereabouts of individuals and items within healthcare facilities, particularly in emergency care situations. This review of literature delves into the progress made in Hospital Real-Time Location Systems (HRTLS) using innovative technologies, particularly within the healthcare landscape of Iran. The objective of the study is to offer insights into the applications, technologies, and effectiveness of HRTLS in medical environments.

2.3.1 Introduction:

The worldwide healthcare industry is undergoing a significant transformation, highlighting the crucial importance of real-time location tracking in mitigating challenges during emergencies and streamlining routine operations within healthcare institutions. With an increasing influx of patients requiring emergency care, hospitals encounter limitations such as resource scarcities and bed capacities. This review of literature investigates the development and importance of Hospital Real-Time Location Systems (HRTLS) in the healthcare sector, with a specific focus on the Iranian context.

2.3.2 Background:

The opening section underscores the growing significance of monitoring individuals and items within healthcare environments, particularly during crises, and acknowledges the challenges associated with this. It underscores the necessity for advanced information technologies to automate healthcare procedures, with a specific emphasis on the real-time tracking of entities within hospital premises.

2.3.3 Objectives:

The study's goals are delineated, centering on the proposition of Hospital Real-Time Location Systems (HRTLS) utilizing innovative technologies in the Iranian context. The narrative review methodology is presented, encompassing the collection and examination of articles and reports from pertinent sources spanning the period between 2006 and 2017.

2.3.4 Methods:

The methodology for the review entails an extensive exploration across databases including IDTechEx, IEEE, PubMed Central, Science Direct, EMBASE/Excerpta Medica, Scopus, Web of Science, Elsevier journals, WHO publications, and Google Scholar. The criteria for inclusion and exclusion, governing the selection of pertinent literature, are explicitly outlined.

2.3.5 Results:

The review discloses findings derived from the scrutiny of Hospital Real-Time Location Systems (HRTLS) applications in 15 specified countries, encompassing Australia, Belgium, Canada, Germany, Iran, Italy, Japan, Luxembourg, Mexico, Netherlands, Russia, South Korea, Turkey, the United Kingdom, and the United States. The applications are classified into eight distinct groups, which comprise tracking systems for elderly care, medical asset tracking, security for children and the elderly, monitoring blood transfusions, tracking medications, patient monitoring, real-time data collection, and healthcare environment monitoring.

2.3.6 Discussion:

In the discussion segment, an exploration of the ramifications of Hospital Real-Time Location Systems (HRTLS) in healthcare institutions is undertaken, highlighting its capacity to improve patient safety, streamline workflow, and enhance overall efficiency. The analysis considers challenges such as technological constraints, interoperability issues, privacy considerations, and the substantial implementation costs.

2.3.7 Implications for HRTLS:

The results bear significance for healthcare managers by offering guidance on the selection of Hospital Real-Time Location Systems (HRTLS) technologies that suit the particular needs of their organizations. The successful implementation of HRTLS is underscored by recognizing the economic implications and prioritizing policies related to patient and staff satisfaction.

2.3.8 Conclusion:

The concluding remarks underscore the importance of real-time location systems in attaining healthcare objectives, emphasizing the enhancement of efficiency, elevation of patient satisfaction, and reduction of time and costs. The research advocates for the incorporation of innovative technologies, such as the Internet of Things (IoT) and cloud computing, to develop and deploy comprehensive Hospital Real-Time Location Systems (HRTLS) within healthcare centers in Iran.

2.3.9 Ethical Considerations:

The conclusion recognizes ethical considerations, underscores the importance of maintaining the integrity of the research process, and emphasizes the significance of adhering to ethical standards in healthcare research. This review offers a thorough summary of the present status of Hospital Real-Time Location Systems (HRTLS), encompassing its applications, challenges, and implications in the healthcare sector, with a particular emphasis on the context of Iran.

2.4 Application of Model-Based Software Testing in Healthcare Domain

The incorporation of Internet of Things (IoT) technologies in the healthcare sector is a rapidly growing domain, characterized by the creation of inventive solutions aimed at improving patient care, optimizing processes, and enhancing overall healthcare results. This review explores key facets of IoT in healthcare, placing a focus on the importance of rigorous testing methodologies, specifically acceptance testing, to guarantee the dependability and security of these systems.

2.4.1 IoT in Healthcare:

The Internet of Things (IoT) is defined as a system of interconnected physical objects and devices that exchange data through central control servers based in the cloud. The utilization of IoT in healthcare offers significant possibilities, allowing for the remote monitoring and management of medical equipment and resources. Nevertheless, the swift evolution of IoT technology introduces complexities that require comprehensive testing.

2.4.2 Challenges in IoT Software Testing:

Although the Internet of Things (IoT) has been extensively embraced across diverse industries, there is a noticeable dearth of research concerning the testing of IoT software within the healthcare sector. The array of technologies utilized in the development of IoT systems presents challenges for efficient testing, and there is a limited presence of proposed solutions and approaches in this specific domain.

2.4.3 Healthcare Software Development and Testing:

The healthcare sector, known for its complex and unique demands, is undergoing substantial changes. Innovations like wearable technology and hospital indexing systems are reshaping the landscape of patient care. Due to the complexity of healthcare products, thorough testing is essential to guarantee quality, given the potential consequences on patient well-being, consumer costs, data privacy, and overall safety.

2.4.4 Integration of IoT and Deep Learning in Healthcare:

Suggestions such as incorporating Internet of Things (IoT) and deep learning methods highlight the potential for remote health monitoring and data analysis. The integration of IoT with a cloud network structure allows for the instantaneous collection and analysis of data, offering prompt assistance to individuals in critical conditions.

2.4.5 Security Testing in Healthcare Applications:

As healthcare increasingly depends on technology, the crucial need for security testing becomes evident. The security of patient data, especially health-related information, is paramount to prevent severe data breaches. Security testing is essential to guarantee that healthcare applications are not only secure but also resilient and free of errors across various scenarios.

2.4.6 Big Data Solutions in Healthcare:

The healthcare sector produces extensive datasets, encompassing detailed patient information. Big data solutions play a pivotal role in facilitating informed decisions regarding disease cures, research and development, and other healthcare-related aspects. Comprehensive testing of this data is imperative to guarantee its accurate implementation and achieve the desired outcomes.

2.4.7 Acceptance Testing in Healthcare Software:

Despite the pivotal role of software in healthcare, there is a notable dearth of research on acceptance testing within this field. Existing literature highlights challenges and progress in model-based testing and approaches to evaluating mobile usability. The suggested framework for acceptance testing in Internet of Healthcare Technology (IoHT) systems employs the user interface as the primary interaction medium, concentrating on black-box testing and verification using UML activity diagrams

2.4.8 UML Activity Diagrams in Test Generation:

UML activity diagrams prove to be a valuable resource for modeling interactive attributes and facilitating the development of test cases. Existing literature underscores the utilization of activity diagrams for constructing activity flow graphs, extracting pertinent details, and generating test cases according to activity path coverage criteria. An illustrative case study involving a diabetic telehealthcare scenario demonstrates the practical implementation of this methodology.

2.4.9 Conclusion:

The examined literature suggests a technique for conducting acceptance testing in IoT systems, specifically within the healthcare sector, using UML activity diagrams. The focus is on an efficient and cost-conscious method, and the application of this approach to a practical telehealthcare test case scenario emphasizes its potential usefulness. The collaborative efforts of multiple authors have played a role in shaping and refining this testing methodology, offering insights that could be valuable to practitioners dealing with similar challenges in ensuring the quality of IoT applications in healthcare.

In summary, the literature review underscores the crucial necessity for effective testing methodologies in the dynamic landscape of IoT in healthcare. The proposed acceptance testing approach, guided by UML activity diagrams, addresses existing research gaps and contributes to the establishment of dependable and secure IoT systems for healthcare applications.

2.5 User Satisfaction Evaluation for Mobile Medical Apps

The survey encompasses diverse facets, such as the introduction of mobile medicine, user contentment, indicator selection, establishment of an evaluation system, sentiment analysis, and the method for assessing satisfaction. The research puts forward an all-encompassing model that encompasses data collection and preprocessing, the creation of an evaluation index system, analysis of sentiment quantification, determination of the weight of evaluation indices, and a fuzzy comprehensive assessment of satisfaction.

2.5.1 Introduction:

Highlights the growing significance of mobile medicine in enhancing the allocation of medical resources. Tackles spatial disparities in medical services and the repercussions of the COVID-19 pandemic. Underlines the crucial role of mobile medical apps in enhancing the accessibility and efficiency of medical care.

2.5.2 User Satisfaction:

Defines user satisfaction and its relevance within the realm of mobile medical apps. Explores conventional approaches to gauging user satisfaction and introduces online reviews as valuable sources of data.

2.5.3 Selection of Indicators and Construction of Evaluation System:

Examines the constraints associated with relying solely on online ratings for user satisfaction assessment. Champions the utilization of online reviews, citing the benefits of extensive data volume and economical data collection. Introduces the latent Dirichlet allocation (LDA) topic model for extracting topics and notes its application in studies focused on satisfaction.

2.5.4 Sentiment Analysis:

Stresses the importance of considering users' opinions and attitudes in satisfaction assessment. Introduces sentiment analysis as a technique for discerning whether a user's perspective is positive or negative. Offers examples of how sentiment analysis has been employed in prior research studies.

2.5.5 Satisfaction Evaluation Method:

Spotlights the uncertainties inherent in individual perceptions and the intricacy of assessment metrics. Argues for the adoption of fuzzy comprehensive evaluation methods to address uncertainties. Introduces rough number theory as a method for quantifying expert cognition without prior knowledge.

2.5.6 Overall Architecture of the Evaluation Model:

Details the overarching structure of the proposed evaluation model. Enumerates five steps: data collection and preprocessing, construction of an evaluation index system, analysis of sentiment quantification, determination of weights, and fuzzy comprehensive evaluation.

2.5.7 Mobile Medical App User Satisfaction Evaluation Model:

Divides the model into distinct steps, encompassing data acquisition and pre-processing, identification of factors influencing satisfaction, and integration of evaluation information based on sentiment disposition analysis.

2.5.8 Data Acquisition and Pre-Processing:

Describes the collection of online review data using Python web crawler technology. Emphasizes the importance of pre-processing, including cleaning, Chinese word separation, and removal of irrelevant content.

2.5.9 Identification of Factors Influencing User Satisfaction:

Introduces the use of the LDA topic model for topic mining in online review texts. Highlights the advantages of LDA in extracting valuable potential topics and revealing relationships between topic words.

2.5.10 Example Analysis:

The review of existing literature underscores the practical implementation of the suggested approaches through the examination of online feedback on the Huawei App Store. The application of methods such as topic clustering, grounded theory, and a three-stage coding process is illustrative of the method's efficacy in pinpointing essential categories and establishing an assessment framework.

2.5.11 Conclusion and Contributions:

The study's conclusion underscores the contributions made, emphasizing its innovative approach to evaluating user satisfaction. Noteworthy aspects include the incorporation of expert opinions and attribute weights, along with the application of fuzzy mathematics. The model's significance in the dynamic field of mobile medical apps is highlighted, providing valuable insights for developers and managers seeking to improve user experience.

2.5.12 Limitations and Future Directions:

Recognizing its limitations, the study emphasizes the necessity for future research to adeptly filter out deceptive comments within user feedback. This points towards a potential area for further investigation to enhance the precision of sentiment analysis and user satisfaction assessment. In conclusion, this literature review offers a thorough examination of the methods utilized in assessing user satisfaction for mobile medical apps. It integrates sentiment analysis, expert opinions, and fuzzy mathematics, providing valuable insights for both researchers and practitioners aiming to comprehend and enhance user satisfaction within the ever-evolving realm of mobile health applications.

2.6 Smartphone Medical Apps:

The objective of this literature review is to furnish a comprehensive understanding of the developing research panorama related to smartphone medical apps. Concentrating on the timeframe spanning from 2010 to 2014, the review investigates how researchers have reacted to the transformative technology of smartphone medical apps. It provides insights into the motivations, challenges, and recommendations found in the current literature. The primary aim is to delineate the landscape of this nascent field, establishing a systematic classification and pinpointing fundamental features that delineate the intersection of healthcare and smartphone technology.

2.6.1 Introduction:

The integration of smartphones into healthcare represents a logical advancement, aligning with the progression of eHealth and mobile health (mHealth). Smartphones, characterized by their robust computing capabilities, connectivity features, and app installation capabilities, have emerged as potent instruments in the healthcare domain. The study charts the evolution of smartphones into personalized, portable computers, underscoring the importance of medical apps in delivering healthcare information, facilitating education, enabling intervention, enhancing adherence, and even contributing to diagnostic processes.

2.6.2 Methods:

The methodology section delineates a systematic strategy for collecting and categorizing data. A targeted exploration was conducted across prominent databases, such as MEDLINE, Web of Science, ScienceDirect, and IEEE Xplore, ensuring a thorough coverage of both medical and technical literature. The study's selection process, eligibility criteria, and data collection procedures are elaborated upon, emphasizing the stringent measures taken to screen and categorize articles with rigor.

2.6.3 Results:

The classification of research literature on smartphone medical apps is structured into four primary groups: reviews and surveys (68/133), studies on medical apps (43/133), reports on actual app development (17/133), and proposals of frameworks (5/133). The survey delineates the foundations of studies, encompassing actual downloaded apps, app descriptions, surveys, questionnaires, and app development. Furthermore, the study delves into the objectives and functionalities of medical apps within various medical specialties.

2.6.4 Discussion:

The discussion section thoroughly examines the importance of taxonomies in shaping future research trajectories, exposing gaps, and fostering a shared language within the research community. It investigates the driving forces behind the integration of smartphone apps in healthcare, hurdles impeding their optimal utilization, and suggestions put forth in existing literature. The taxonomy outlined in the paper illuminates the vibrant sectors in medical care that are adopting the mHealth trend and pinpoints less-explored areas, offering valuable insights for both researchers and practitioners.

2.6.5 Limitations:

Recognizing the inherent limitations of the survey, this section underscores constraints associated with source databases, the ever-changing nature of the field, and the balance between research activity representation and real-world impact. It accentuates the imperative for ongoing adaptation to emerging trends in mobile health and medical app technology.

2.6.6 Conclusion:

The concluding section encapsulates key findings, with a focal point on the dynamic nature of the smartphone medical apps landscape. The article enhances comprehension of available options and identifies research gaps, presenting itself as a valuable resource for newcomers to this field. It urges scholars to stay attuned to emerging trends, especially the integration of wearable gadgets linked to smartphones, orchestrated by apps, and empowered by built-in biosensors.

2.7. Fine-tuning a Large Language Model using Reinforcement Learning from Human Feedback for a Therapy Chatbot Application

2.7.1 Introduction:

In recent years, the swift expansion of machine learning and artificial intelligence (AI) has resulted in noteworthy advancements in Natural Language Processing (NLP). Particularly, Large Language Models (LLMs), like ChatGPT, have become essential tools. A burgeoning approach for refining LLMs is Reinforcement Learning from Human Feedback (RLHF). RLHF entails integrating human feedback to improve LLM performance, allowing for specialization in specific tasks such as customer service, language translation, and, as explored in this study, functioning as a virtual psychologist assistant.

2.7.2 Project Background:

This study constitutes a component of a larger initiative with the objective of constructing an AI chatbot specializing in Cognitive Behavioral Therapy. The broader project, overseen by Birger Moell, a licensed psychologist and PhD student at KTH, engages psychologists and peers at KTH. The primary aim is to investigate the potential of Reinforcement Learning from Human Feedback (RLHF) in customizing Large Language Models (LLMs) for particular domains. This endeavor seeks to foster innovation in machine learning for the benefit of both open-source and research communities.

2.7.3 Goal:

The principal objective is to assess the influence of Reinforcement Learning from Human Feedback (RLHF) on the refinement of Large Language Models (LLMs) within the realm of psychology. The research endeavors to determine whether RLHF can produce pertinent outcomes for mental health applications and, potentially, be extended to other domains such as healthcare and education. The focus lies on fostering ethical AI innovation within the field of psychology, with a commitment to inspiring responsible practices.

2.7.4 Ethical Aspect:

The ethical considerations in deploying Reinforcement Learning from Human Feedback (RLHF) for psychological chatbots are of utmost importance. The study delves into concerns such as the generation of toxic language stemming from biased training datasets and the accountability for the actions of the bot. The potential risks and consequences associated with utilizing AI in sensitive domains like mental health highlight the imperative for ethical guidelines and regulations.

2.7.5 Scientific Questions:

The study raises pivotal inquiries regarding the influence of Reinforcement Learning from Human Feedback (RLHF) on Large Language Model (LLM) responses in the context of mental health. Furthermore, it delves into the ethical implementation of RLHF in psychological practice and examines how regulatory frameworks can shape

AI innovation within the field of psychology.

2.7.6 Problem Definition:

To gauge the suitability of Large Language Model (LLM) outputs, the study assesses elements such as support, listening, validation, insight, guidance, impartiality, and respect. The development of evaluation criteria is guided by licensed psychologists. Psychology students from Karolinska Institutet and Stockholm University actively participate in surveys designed to measure their perceptions of the conversations generated by the LLM.

2.7.8 Expected Results:

The qualitative evaluation, conducted through surveys, seeks to showcase the potential enhancement in the Large Language Model (LLM) performance as a psychologist through Reinforcement Learning from Human Feedback (RLHF). The study involves a comparison between pre-trained LLMs and their RLHF-trained counterparts to gauge their effectiveness in meeting the established evaluation criteria.

2.7.9 Large Language Models:

Large Language Models (LLMs), especially transformers utilizing self-attention mechanisms, play a crucial role in contemporary language modelling. Transformers enable contextual comprehension, the processing of distant information, and the improvement of intelligent responses. These models come in various sizes, ranging from 10 billion to 280 billion parameters, and it has been reported that GPT-4 boasts a staggering one trillion parameters.

2.7.10 Fine-tuning:

Fine-tuning is the process of refining Large Language Models (LLMs) for specific outputs. Parameter Efficient Fine-Tuning techniques, such as Low-Rank Adaption, focus on minimizing the number of parameters that need fine-tuning. This approach aims to improve efficiency and reduce complexity in the fine-tuning process.

2.7.11 Reinforcement Learning from Human Feedback:

Reinforcement Learning from Human Feedback (RLHF) encompasses several steps, starting with pre-training Large Language Models (LLMs), collecting datasets to create a reward model, and fine-tuning through reinforcement learning. Human feedback plays a crucial role in guiding the reward model, mapping LLM outputs to scalar rewards. Proximal Policy Optimization (PPO), a policy-gradient algorithm, is employed in RLHF to fine-tune models.

2.7.12 Personal Data:

The integration of AI tools in psychology, which often involves dealing with sensitive personal data, necessitates careful consideration of privacy concerns. GDPR (General Data Protection Regulation) regulations play a crucial role in guaranteeing the secure processing of personal information and providing safeguards against potential data breaches. Compliance with these regulations is essential to ensure ethical and lawful use of personal data in the context of AI applications in psychology.

2.7.13 Conclusion:

The literature survey underscores the growing significance of Reinforcement Learning from Human Feedback (RLHF) in the fine-tuning of Large Language Models (LLMs), with a specific focus on applications in psychology. The inclusion of ethical considerations, scientific inquiries, and theoretical foundations establishes a comprehensive groundwork for the research. This emphasis on responsible AI development is particularly crucial in sensitive domains such as mental health.

2.8 Nearest Hospital Tracking and Disease Prediction

2.8.1 Healthcare Information Systems:

Examine the current body of literature pertaining to healthcare information systems, electronic health records (EHR), and mobile applications within the healthcare sector. Analyze the ways in which technology is employed to enhance patient care, optimize operational processes, and improve the overall delivery of healthcare services.

2.8.2 Hospital Information Systems:

Examine research that concentrates on hospital information systems and their influence on patient results. Explore the utilization of digital platforms in overseeing hospital resources, enhancing communication, and delivering improved healthcare services.

2.8.3 Real-time Healthcare Data:

Review studies pertaining to the real-time management of healthcare data. Comprehend the obstacles and advantages linked to real-time information in critical scenarios, such as determining ICU bed availability and monitoring blood bank status.

2.8.4 Mobile Health (mHealth) Applications:

Investigate the literature on mobile health applications, their functionalities, and their role in facilitating patient engagement and healthcare accessibility. Analyze how similar applications have addressed challenges in emergency healthcare situations.

2.8.5 Predictive Healthcare Systems:

Investigate research articles focusing on the incorporation of predictive models into healthcare systems. Gain insights into how predictive analytics can aid in diagnosing diseases based on symptoms, akin to the suggested functionality in the application.

2.8.6 User Experience in Healthcare Apps:

Examine relevant literature on the principles of user experience (UX) design within healthcare applications. Investigate how a user-friendly interface plays a pivotal role in enhancing patient engagement and satisfaction, as demonstrated by existing research.

2.8.7 Challenges in Emergency Healthcare Management:

Recognize the challenges and deficiencies present in current emergency healthcare management systems. Comprehend the limitations of prevailing approaches and explore how technological solutions, such as the envisioned application, can effectively address these issues.

2.8.8 Geospatial Healthcare Applications:

Investigate scholarly studies that centre on geospatial applications in healthcare, with a specific emphasis on those leveraging GPS technology for pinpointing healthcare facilities. Grasp how location-based services can enhance accessibility and response times in emergency situations, as indicated by relevant research.

2.8.9 Conclusion:

Through an extensive literature review, the suggested healthcare application can be situated within the current scholarly landscape, showcasing its innovativeness, potential contributions, and significance in tackling vital healthcare challenges.

2.9 NoSQL databases: Critical analysis and comparison

2.9.1 Introduction:

The introduction establishes the context by highlighting the issues arising from the non-uniform growth of data in distributed web applications and cloud computing. It underscores the demand for highly scalable and efficient non-relational databases, giving rise to the pivotal role of NoSQL databases as a significant alternative to conventional relational databases.

2.9.2 Classification of NoSQL Databases:

In this segment, NoSQL databases are classified into four primary types: key-value stores, graph databases, wide column stores, and document stores. The characteristics of each category are delineated, emphasizing their significance in meeting distinct data requirements.

2.9.3 Functional and Non-Functional Features Comparison:

A comprehensive analysis of both functional and non-functional aspects of NoSQL databases is undertaken. The survey encompasses factors including performance, scalability, flexibility, structure, complexity, de-normalization, joins, atomicity, aggregation, and keys. Additionally, the CAP theorem is introduced as a conceptual framework to grasp the limitations inherent in databases.

2.9.4 Selected Databases for Analysis:

The research strategically chooses particular databases from each category for a thorough examination. MongoDB (representing document stores), Cassandra (for wide column stores), Redis (exemplifying key-value stores), and Neo4j (representing graph databases) are selected for comparison, considering factors such as their data models, adherence to the CAP theorem, distributive properties, and other relevant criteria.

2.9.5 Importance of NoSQL Databases:

This section underscores the importance of NoSQL databases in effectively handling large, diverse, and dynamically evolving datasets. It emphasizes their capability to efficiently manage big data, deliver high velocity, and navigate through various complexities inherent in diverse datasets.

2.9.6 Comparison with SQL Databases:

A comprehensive comparison is presented between SQL (Relational Database Management System) and NoSQL databases. Variances in aspects such as data validity, query language, data type, storage, schema, flexibility, scalability, and ACID compliance are thoroughly examined and discussed.

2.9.7 Comparison Across NoSQL Categories:

The survey conducts a comparative analysis of databases within the four categories—wide column stores (Cassandra), document stores (MongoDB), key-value stores (Redis), and graph databases (Neo4j). A range of features, including database model, read/write/delete operations, language, licensing, data schema, predefined types, and more, are scrutinized to provide a comprehensive comparison.

2.9.8 Distributive Properties Analysis:

The research delves into the functionality of chosen NoSQL databases in a distributed environment. Features like sharding, partitioning, scaling, replication methods, and the handling of foreign keys are discussed to offer insights into the distributive properties of these databases.

2.9.9 Conclusion:

In conclusion, the summary encapsulates pivotal discoveries extracted from the literature survey. It accentuates both the strengths and limitations inherent in NoSQL databases, presenting valuable insights for researchers and practitioners engaged in the field of database management.

2.10 A Study of NoSQL Database

The emergence of Web Service 2.0 applications has led to an increased demand for the effective handling of big data. Traditional relational databases face challenges in scaling horizontally to meet the requirements of high performance, especially for applications dealing with extensive user data and concurrent access. In response to these challenges, a new category of databases known as NoSQL has gained popularity. The motivation behind the NoSQL movement stems from the growing demand for cloud computing and the evolving landscape of the Internet.

This paper aims to explore the features and data models of NoSQL databases in the context of a cloud computing environment. It also addresses the strengths and limitations of each NoSQL model and delves into the classification of NoSQL databases based on the CAP theorem (Consistency, Availability, Partition tolerance). The CAP theorem posits that in a distributed environment, designers can prioritize any two of the three properties—Consistency, Availability, and Partition tolerance.

2.10.1 Introduction:

Efficient storage and retrieval of data with scalability and availability are primary goals for NoSQL databases. NoSQL, which stands for "Not Only SQL," serves as an alternative to traditional relational databases. With the introduction of non-relational databases like MongoDB, Hbase, and Neo4j, the database industry has witnessed significant changes in recent years. Cloud vendors, driven by the demand for scalable and high-performance solutions, have embraced various types of databases based on their business requirements and strategies. Designers of pre-relational databases argue that NoSQL databases may not be efficient enough in handling data integrity. This paper is structured to discuss the importance of NoSQL databases, highlighting the ongoing debate between SQL and NoSQL, and addressing the challenges posed by traditional relational databases in the era of big data.

2.10.2 Background:

The SQL vs. NoSQL debate centers on the distinction between relational and non-relational databases. Traditional relational databases, characterized by normalized data models and strict ACID properties, require a predefined schema before storing data. This schema inflexibility becomes a bottleneck when dealing with the constant need to add new types of data in the era of Big Data. Large-scale web services like Google, Amazon, Yahoo, and Facebook encountered challenges in scaling relational databases to meet the demands of scalability and performance. These challenges led to the development of non-relational databases or NoSQL databases.

2.10.3 Features of NoSQL:

NoSQL databases exhibit three fundamental features: scale-out, flexible data structure, and replication. Scale-out refers to achieving high performance in a distributed environment by utilizing numerous general-purpose machines. NoSQL databases allow data distribution over a large number of machines, supporting automatic distribution when new machines are added to the cluster. Flexibility in terms of data structure means that NoSQL databases do not require a predefined schema, enabling users to store data of various structures in the same database table. Replication involves distributing copies of data to different systems to achieve redundancy and load distribution.

2.10.4 NoSQL Data Models:

NoSQL databases encompass various data models, each designed to address specific requirements.

2.10.5 Key-Value Data Stores:

Key-value data stores are designed to handle highly concurrent access to databases. Each data item consists of a unique key-value pair, with the key generated by the application and associated with the value. Key-value data stores use a hash function to locate data in the database, making them row-focused and efficient for retrieving data for complete entities.

2.10.6 Document-Oriented Data Stores:

Document-oriented databases are similar to key-value data stores but offer transparency in the stored data. Documents in this model are entities composed of named fields, allowing queries based not only on the key but also on attribute values within the document. Documents need to be self-describing and are stored in formats like XML, BSON, or JSON.

2.10.7 Column Family Data Stores:

Column family data stores enable the storage of data in a column-centric approach. Data is organized into rows and columns, with each column family consisting of columns. Rows are identified by a unique row key, and each column is a map of a key-value pair. Column family databases are efficient for handling large datasets that scale horizontally.

2.10.8 Graph Databases:

Graph databases specialize in handling highly linked data, representing entities as nodes and relationships between entities as edges. Nodes hold information about entities, and edges represent relationships. Graph databases are well-suited for scenarios with complex relationships between entities, such as social networking, data mining, and network management.

2.10.9 Transaction in NoSQL Databases:

The transaction models of SQL and NoSQL databases are at the core of the SQL vs. NoSQL debate. Traditional SQL databases adhere to strict ACID properties (Atomicity, Consistency, Isolation, Durability), ensuring a robust transactional framework. In contrast, NoSQL databases embrace the CAP theorem, which asserts that in a distributed environment, designers can prioritize any two of the three properties: Consistency, Availability, and Partition tolerance.

2.10.10 The CAP theorem has led to the classification of NoSQL databases into three categories based on their transaction models:

CA (Consistency, Availability): These databases prioritize data consistency and availability but may sacrifice partition tolerance. Examples include Vertica and Greenplum.

CP (Consistency, Partition tolerance): Databases in this category prioritize data consistency and partition tolerance, often at the expense of availability. Examples include Hypertable and BigTable.

AP (Availability, Partition tolerance): Databases in this category prioritize data availability and partition tolerance, with eventual consistency. Examples include Riak, CouchDB, and KAI.

The CAP theorem has been extended to PACELC (Partition, Availability, Consistency, Else, Latency, Consistency), highlighting the trade-off between availability and consistency based not only on partition tolerance but also on the existence of network partition and latency.

2.10.11 Comparison of NoSQL Databases:

Choosing the right NoSQL database for an enterprise involves considering factors such as business model, strategy, cost, and transaction model demands. The selection depends on the specific requirements of the applications and the characteristics of the data being handled.

Key-Value Stores: Suitable for applications focused on storing and retrieving opaque data items using a key as an identifier. However, it is limited in querying capabilities and updating individual fields.

Document Stores: Efficient for applications that require querying based on attribute values other than the key and support selective filtering of records.

Column-Family Stores: Ideal for applications dealing with large datasets, storing records with hundreds or thousands of fields, and retrieving subsets of those fields in most queries.

Graph Databases: Specialized for applications dealing with highly linked data and complex relationships between entities.

The choice of a NoSQL database depends on the specific needs of the enterprise, and there is no one-size-fits-all solution. The paper provides a comparison of various NoSQL databases, their data models, transaction models, and query languages.

2.10.12 Conclusion:

In conclusion, NoSQL databases represent a relatively new but rapidly evolving paradigm in the field of databases. Developed to address the challenges posed by big data, NoSQL databases come with their own set of limitations. There is currently no standardized query language for NoSQL databases, and each database system operates differently. Despite their immaturity and ongoing development, NoSQL databases offer unique features and advantages for handling large-scale data processing.

2.11 Application of Firebase in Android App Development

2.11.1 In-Depth Analysis of NoSQL Databases:

Walter Kriha [1] conducts a thorough examination of NoSQL databases, covering fundamental concepts, methodologies, and classifications. The article provides insights into various NoSQL categories, including key/value stores, document databases, and column-oriented databases. Kriha explores the merits and drawbacks of NoSQL databases, delivering a systematic perspective on their functionalities.

2.11.2 Comparative Exploration of SQL and NoSQL:

Supriya S. Pore and Swalaya B. Pawari [2] undertake a comparative study between SQL and NoSQL databases. The focus is on distinguishing between these two database paradigms and highlighting their respective foundational principles. The study underscores the impact of data consistency on the non-application of ACID properties in NoSQL databases, elucidating the distinctive features of each database genre.

2.11.3 NoSQL Databases Overview and Their Influences:

Vatika Sharma and Meenu Dave [3] present a panoramic view of NoSQL databases, underscoring their role in diminishing SQL dominance. The authors provide insights into the historical context and defining characteristics of NoSQL databases, discussing their transformative impact on the database landscape. This review contributes to comprehending the evolving landscape of database technologies and their ramifications.

2.11.4 Integrating Firebase with Android Apps:

Daniel Pan [4] imparts insights into the integration of Firebase with Android applications, offering foundational knowledge on structuring databases within Firebase. The article aids developers in seamlessly incorporating Firebase into their Android applications, ensuring a cohesive connection between the app and the Firebase platform.

2.11.5 Comparative Analysis of SQLite and Firebase:

Landon Cox [5] offers a comparative study between SQLite and Firebase, centering on the organization of data within a JSON tree for Firebase storage. The comparison provides valuable insights into the strengths and weaknesses of each database system, assisting developers in making informed decisions about their database technology selection.

2.11.6 Comparison of Various DBMS:

The paper presents comparative tables that elucidate the distinctions between Firebase and SQL, as well as Firebase and MS SQL Server. Key parameters such as data storage, schema flexibility, and developer support are meticulously compared, furnishing a comprehensive understanding of the capabilities and attributes of each database management system.

2.11.7 Steps for Firebase Integration with Android:

The paper provides a step-by-step guide for the seamless integration of Firebase into Android projects. The process involves creating a project in the Firebase console, configuring project settings, and adding Firebase dependencies to the Android app. The detailed instructions serve as a roadmap for developers, ensuring a successful integration.

2.11.8 Utilizing Firebase Features in Android Applications:

Comprehensive explanations are offered for integrating various Firebase features into Android applications. The sections cover authentication, real-time database usage, storage implementation, and data manipulation—providing a practical guide for developers seeking to harness Firebase functionalities in their Android projects.

2.11.9 Firebase and Android App Development:

The paper details the development of an Android app using Firebase, showcasing the user interface design and database structure. Screenshots and explanations demonstrate the key features of the app, highlighting the seamless integration of Firebase for efficient data management and user interaction.

2.11.10 Outcome:

The paper concludes by presenting screenshots illustrating the developed Android app leveraging Firebase. The home screen facilitates user uploads of images with captions, while the post screen enables users to view posts from others. This tangible outcome exemplifies the successful implementation of Firebase features in a real-world application scenario.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

3.1 Google Maps Literature Review (2.1.6 Conclusion):

Research Gap: The current research lacks an in-depth exploration of user experiences and privacy concerns associated with the utilization of Google Maps.

Future Research Opportunities: There is a need to delve into user perceptions, concerns, and experiences regarding privacy features and the usability of Google Maps.

3.2 Healthcare Real-Time Location Systems (2.3.8 Conclusion):

Research Gap: The current body of research does not adequately address the challenges and concerns linked to the adoption of IoT and cloud computing in implementing HRTLS in healthcare centres in Iran.

Future Research Opportunities: There is potential for future research to investigate obstacles and ethical issues related to integrating IoT and cloud computing in Iranian healthcare settings.

3.3. IoT System Acceptance Testing in Healthcare (2.4.9 Conclusion):

Research Gap: The current research has a limited focus on the scalability and generalizability of the proposed acceptance testing method for IoT applications in healthcare.

Future Research Opportunities: Future research should explore the adaptability of the proposed testing methodology to different healthcare scenarios and assess its scalability in larger-scale IoT implementations.

3.4 User Satisfaction Evaluation in Mobile Medical Apps (2.5.11 Conclusion and Contributions):

Research Gap: There is insufficient exploration of cultural and demographic factors that may influence user satisfaction with mobile medical apps.

Future Research Opportunities: Future research should examine the impact of cultural and demographic variables on user satisfaction to improve the applicability of the proposed model in diverse user populations.

3.5 Smartphone Medical Apps Landscape (2.6.6 Conclusion):

Research Gap: The current research has limited exploration of potential security and privacy concerns associated with the integration of wearable gadgets and biosensors in smartphone medical apps.

Future Research Opportunities: Future research should investigate the security implications and privacy considerations when integrating wearable technologies with smartphone medical apps.

3.6 Reinforcement Learning in Healthcare for Psychological Applications (2.7.13 Conclusion):

Research Gap: There is a limited examination of the long-term ethical implications and potential biases introduced by RLHF in psychological applications.

Future Research Opportunities: Future research should explore the ethical considerations and potential biases associated with the long-term use of RLHF in psychological applications.

3.7 Healthcare Application Contextualization (2.8.9 Conclusion):

Research Gap: There is limited discussion on potential challenges and barriers to the practical implementation of the proposed healthcare application in real-world settings.

Future Research Opportunities: Future research should investigate the practical challenges, implementation barriers, and stakeholder perspectives related to deploying the proposed healthcare application.

3.8 NoSQL Databases in Database Management (2.9.9 Conclusion):

Research Gap: There is insufficient exploration of specific use cases or industries where NoSQL databases may outperform traditional relational databases.

Future Research Opportunities: Future research should examine industry-specific scenarios to identify contexts where NoSQL databases offer distinct advantages.

CHAPTER-4
PROPOSED METHODOLOGY

4.1 Project Initiation:

- Define project scope, objectives, and deliverables.
- Assemble a cross-functional team of developers, data scientists, and healthcare experts. Allocate resources and establish project timelines.

4.2 Requirements Analysis:

- Gather and document detailed functional and technical requirements based on the problem statement and objectives.
- Identify key features such as hospital search, recommendation algorithms, user registration, feedback mechanisms, and geospatial services.

4.3 Technology Stack Selection:

- Web App (Flutter for cross-platform development)
- Web App (Flutter for cross-platform development)
- Backend (Node.js for server-side development)
- Database (Firebase Realtime Database or Firestore for data storage)
- Machine Learning (Python with relevant libraries, such as scikit-learn or TensorFlow)

4.4 System Architecture Design:

- Design the system architecture, including the mobile and web app components, databases, and machine learning models.
- Create an API design for data communication between the web app and the backend.

4.5 Development: web App (Flutter):

- Implement the web app using Flutter, focusing on a user-friendly and intuitive interface. Integrate geospatial services for precise location data.
- Develop user registration and feedback mechanisms.

- Implement search functionality for nearby hospitals.
- Backend (Node.js).

4.6 Machine Learning:

- Develop and train Machine Learning models to analyze user preferences and hospital data.
- Fine-tune recommendation algorithms to offer personalized hospital suggestions based on medical requirements.
-

4.7 Data Integration:

- Integrate Firebase for data storage and authentication.
- Set up real-time data synchronization between the web app and Firebase to provide live updates.

4.8 Testing:

- Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
- Test geospatial accuracy, data reliability, and recommendation accuracy.
- Address any issues and refine the application based on user feedback.

CHAPTER-5
OBJECTIVES

5.1 Efficient Hospital Search:

Provide users with a user-friendly interface for searching and locating nearby hospitals quickly and efficiently.

5.2 Tailored Hospital Recommendations:

Employ Machine Learning algorithms to offer hospital recommendations based on the user's specific medical requirements, such as treatment, specialist doctors, and medicine/blood availability.

5.3 Real-time Information:

Ensure that the application provides up-to-date information on hospital facilities, services, and the availability of essential medical resources.

5.4 Emergency Preparedness:

Assist users in making informed decisions during medical emergencies, helping them find the most suitable hospital promptly.

5.5 Improved Health Outcomes:

Contribute to better health outcomes by connecting users to hospitals that offer the required expertise, treatment, and resources.

5.6 Reduce Search Time:

Minimize the time and effort individuals spend in searching for appropriate healthcare facilities, ultimately leading to faster access to care.

5.7 User-Friendly Interface:

Design a simple, intuitive, and easy-to-navigate interface to cater to a wide range of users, including those in distress.

5.8 Enhanced Transparency:

Promote transparency in the healthcare system by providing information about hospitals' services, specialities, and resources, facilitating informed decision-making.

5.9 Geospatial Accuracy:

Utilize geospatial technology to ensure precise location information, enabling users to find the nearest healthcare options.

5.10 Public Awareness:

Promote the application to reach a wider audience, ensuring that more individuals have access to its benefits during medical emergencies.

5.11 Feedback Mechanism:

Implement a feedback system for users to share their experiences, thereby improving the quality of recommendations and information provided.

5.12 Continual Improvement:

Regularly update and refine the application, incorporating user feedback and emerging technologies to enhance its performance and relevance.

5.13 Promote Healthcare Equity:

Ensure that the application is accessible to a wide range of users, promoting healthcare equity by aiding underserved communities in making informed healthcare decisions.

5.14 Community Engagement:

Foster a sense of community by encouraging users to share valuable insights and information that can benefit others in their vicinity.

5.15 Data Security:

Prioritize data security and privacy, safeguarding users' personal and medical information.

5.16 Collaboration with Healthcare Providers:

Establish partnerships with hospitals, clinics, and healthcare institutions to ensure the accuracy and completeness of information presented within the application.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 System Design:

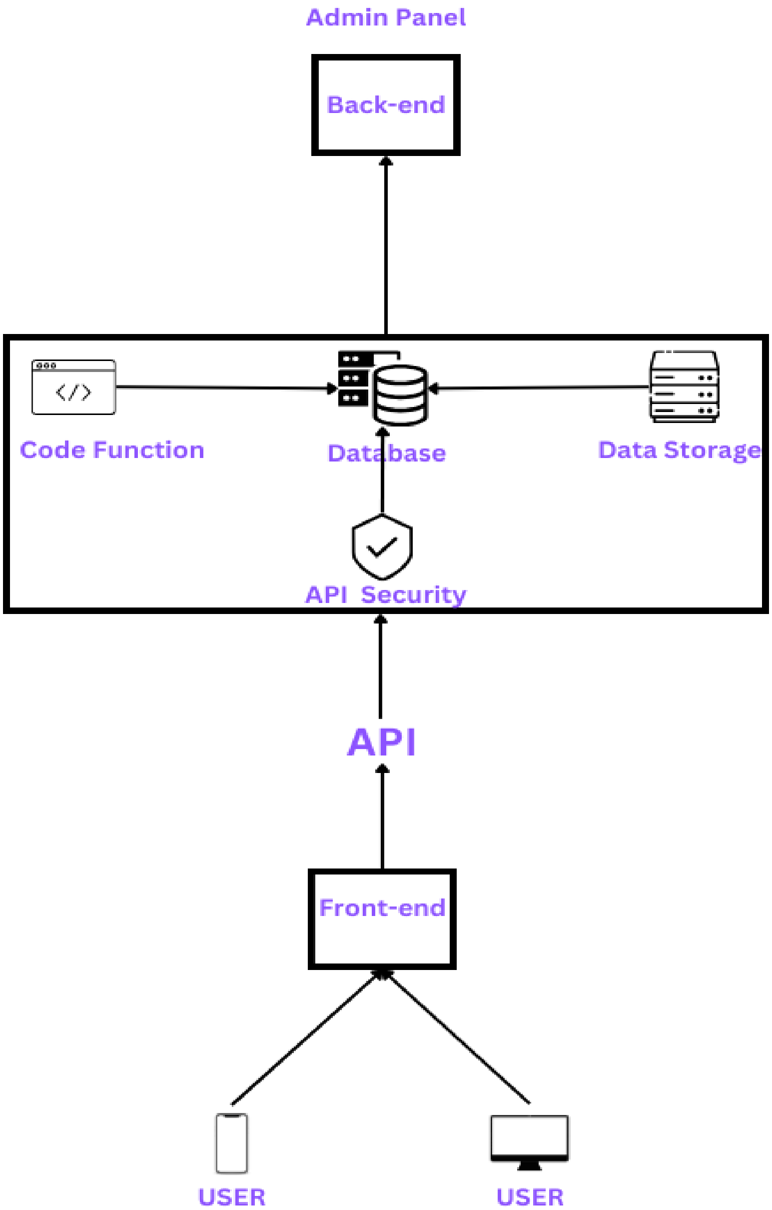


Figure 6.1

The intricately architected system in question represents a pinnacle of technological sophistication, embodying a multifaceted and nuanced approach to digital infrastructure. At its core, this system seamlessly amalgamates an administrative panel endowed with direct access to the backend, while concurrently harnessing the power of a secure and efficient API interface to bridge the frontend with the intricacies of the backend machinery. This architectural duality is not merely a testament to organizational prowess but serves as a testament to the deliberate pursuit of enhanced system manageability, scalability, and adaptability in the face of dynamic technological landscapes.

Delving into the administrative realm, the dedicated panel emerges as a central command nexus, empowering administrators with an unparalleled vantage point for overseeing and orchestrating intricate backend operations. The immediacy and granularity of control afforded by this direct backend access become instrumental in effecting real-time adjustments, optimizations, and troubleshooting, fostering an ecosystem where responsiveness to evolving demands is not just a virtue but a fundamental characteristic.

On the flip side of this technological tapestry lies the front end, intricately interwoven with the back end through a meticulously designed API layer. This strategic decoupling of the front end and back end not only exemplifies a modular approach but also facilitates the deployment of a diverse array of front-end applications without necessitating disruptive modifications to the foundational backend infrastructure. The API, serving as the linchpin of this symbiotic relationship, provides a standardized conduit for seamless communication, thereby augmenting interoperability and elevating the end-user experience to unprecedented heights.

Yet, the pièce de résistance of this technological symphony lies in the integration of Akamai's formidable API security apparatus. As a globally recognized authority in content delivery and cloud services, Akamai bestows upon the API layer an impenetrable shield against potential security threats and vulnerabilities. This fortified security paradigm encompasses an array of cutting-edge measures, encompassing robust encryption protocols, proactive threat detection mechanisms, and swift mitigation strategies. By leveraging Akamai's unparalleled expertise, the system attains a zenith of resilience against the omnipresent spectre of cyber threats, ensuring that the data traversing the intricate interplay between the front and backend remains sacrosanct in terms of confidentiality, integrity, and availability.

In summation, this meticulously sculpted system transcends the conventional boundaries of technological innovation. The dichotomy of administrative empowerment and secure API integration, bolstered by Akamai's prowess, not only optimizes the system's operational cadence but also erects an impregnable fortress against potential cyber adversaries. In doing so, it not only sets the bar for efficiency, scalability, and security but also heralds the advent of a new era in digital ecosystems era where technological fortitude converges seamlessly with administrative acumen to forge a paradigm that is as resilient as it is visionary.

6.2.1 Sequence Diagram Of User:

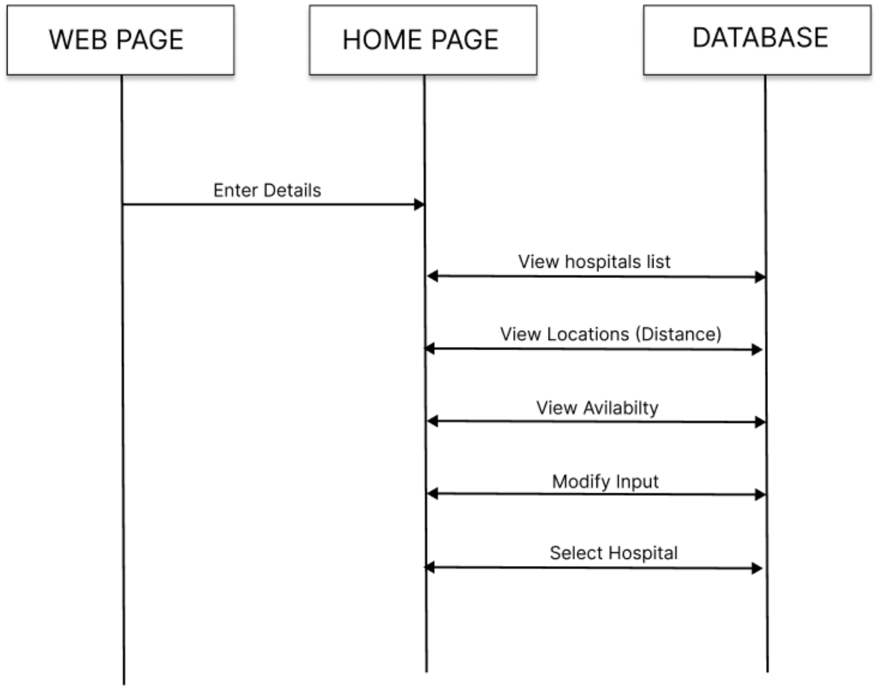


Figure 6.2.1

In the intricate and expansive narrative depicted by this meticulously crafted sequence diagram, the user's foray into the web page interface unfolds as a symphony of interactions, each note resonating with the complexity and depth inherent in digital orchestration. The overture commences with the user's poised engagement at the homepage, where they embark on a journey of information exchange by inputting a myriad of essential details, thereby initiating a sequence of events that traverse the intricate landscape of the system's architecture.

The web page, akin to a blank canvas, transforms into a dynamic tapestry as the user populates it with data, signaling the inception of a collaborative exchange of information.

As the user's keystrokes imbue the web page with life, the backend of the system assumes the role of a virtuoso conductor, orchestrating a harmonious ensemble of database calls. This pivotal juncture is demarcated by a graceful downward arrow, symbolizing the unidirectional flow of information from the user interface to the backend, where a complex dance with the database ensues. Armed with location-specific parameters, the backend intricately navigates the database, issuing tailored queries to extract a trove of data reflective of the user's specified criteria. This symbiotic exchange is a pivotal symphony, transforming raw data into a melodic composition of meaningful insights, laying the foundation for the subsequent movements in the grand symphony of user interaction.

The subsequent segment of this elaborate ballet unfolds with the backend processing the acquired details, transforming them into a curated menu of options presented to the user on the digital stage. This menu, a versatile palette of choices including "View Hospitals," "View Location," "View Availability," "Modify Input," and "Select Hospital," serves as a user-centric toolkit, bestowing upon the user an unparalleled agency in sculpting their digital experience. The arrows connecting these options delineate a dynamic dance, allowing the user to pirouette seamlessly between functionalities, emphasizing a design philosophy that places user empowerment at the forefront.

The climactic crescendo of this grand symphony is reached as the meticulously processed details are presented to the user, creating an interface that transcends mere information dissemination to embody a transformative and empowering experience. The bidirectional arrows in this final movement signify not only the fluidity of information but also the dynamic interplay between the user and the system, where each user choice becomes a resonating chord in the overarching symphony of the user experience.

In essence, this opulent and expansive sequence diagram encapsulates not merely a series of interactions but a saga of user engagement, from the initiation of input on the web page to the intricacies of backend processes. The resultant user interface emerges as a masterpiece, enriched with contextual and meaningful information, a testament not only to the technical prowess of the system but to the harmonious integration of technology and user-centric design, orchestrating a symphony that resonates with the personalized cadence of each user's hospital selection journey.

6.2.2 Sequence Diagram Of Admin:

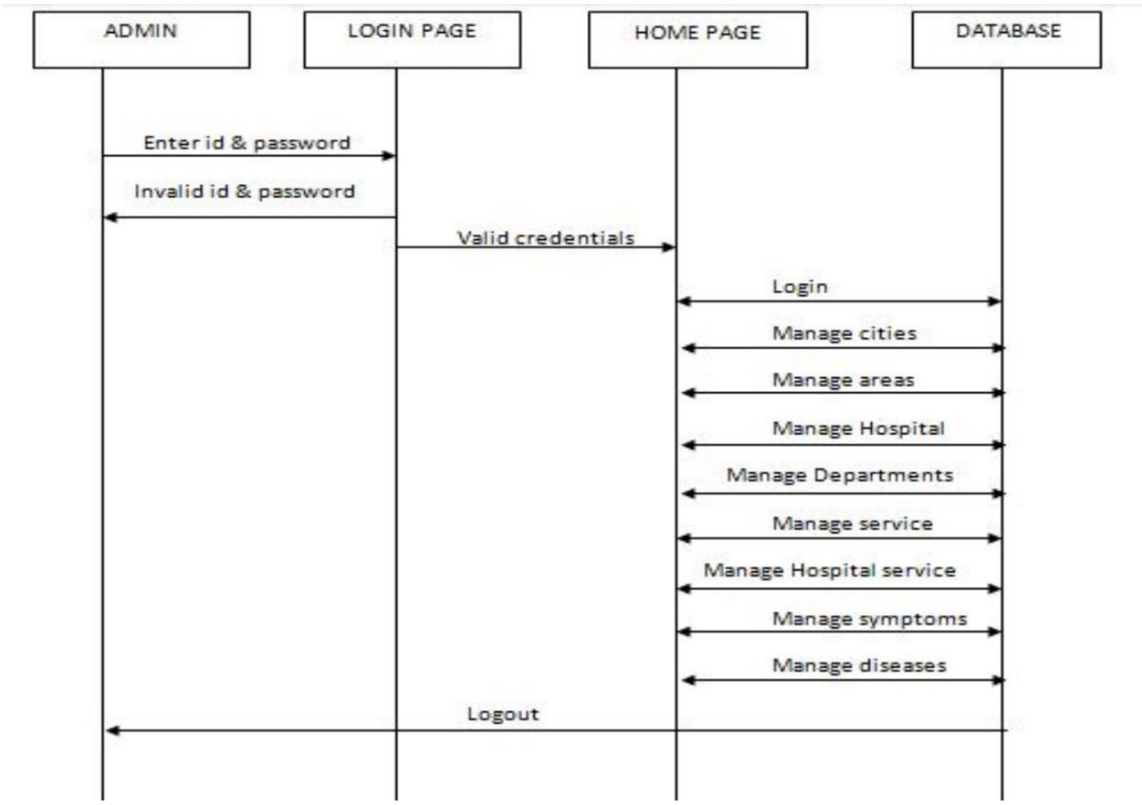


Figure 6.2.2

In the expansive canvas of the sequence diagram detailing the intricate orchestration of administrative processes, a panoramic tapestry unfolds, revealing the nuanced ballet of actions and interactions that characterize the administrative workflow within the system. The symphony begins with the initiation of the sequence by the administrative persona, who, from the helm of the admin panel interface, serves as the conductor of this digital symphony. This initial interaction sets in motion a dynamic series of events, illustrated by a cascade of arrows symbolizing the unidirectional flow of commands and responsive actions. The opening act encompasses the admin panel dispatching commands that act as catalysts for backend processes. These backend processes, akin to the unseen machinations backstage, encompass a multifaceted array of tasks such as user authentication, data retrieval, and system updates. The backend, in this orchestral analogy, assumes the role of a well-coordinated ensemble, responding to the maestro's commands by executing a symphony of operations with precision and synchrony.

As the administrative ballet progresses, the admin panel becomes the focal point for receiving feedback and updates from the backend, marked by arrows flowing back to the admin interface. This dynamic feedback loop exemplifies the real-time responsiveness of the system, affording administrators immediate insights into the outcomes of their commands and interventions. It is a digital dialogue between the administrator and the system, where every command initiates a responsive chord, resonating through the administrative landscape.The subsequent movements of this administrative symphony unfold as various functionalities take center stage, represented as distinct branches in the diagram. These functionalities encompass the intricacies of user management, system configuration, data analysis, and potentially even decision-making processes. Each branch, akin to a narrative subplot, contributes to the overarching storyline of administrative control and oversight, forming a rich and complex narrative tapestry.

The symmetrical and organized nature of the sequence diagram not only visually conveys the logical flow of actions within the administrative realm but also symbolizes the structured nature of administrative processes. Decision points and conditional branches, intricately woven into the diagram, underscore the flexibility and adaptability inherent in the administrative workflow, allowing administrators to navigate through diverse scenarios and contingencies with finesse.

In essence, this expansive sequence diagram paints a vivid and sprawling portrait of the administrative journey within the system, capturing the ebb and flow of commands, responses, and feedback. It transcends mere technical documentation to become a visual symphony that not only illuminates the intricacies of system administration but also underscores the pivotal role administrators play in orchestrating and maintaining the harmonious functionality of the digital ecosystem they steward.

6.3 Use Case Diagram:

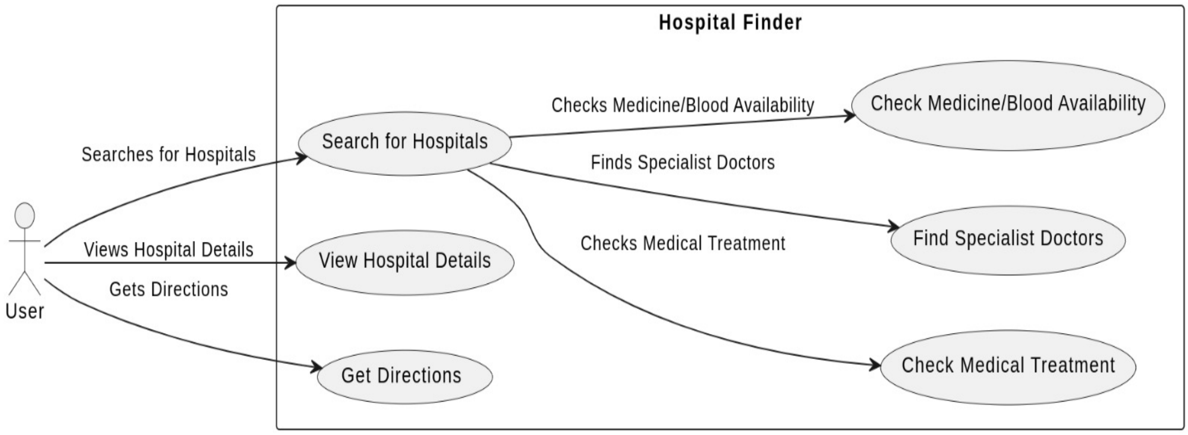


Figure 6.3

In the expansive landscape of the use case diagram depicting the intricacies of a Hospital Finder Application, a comprehensive narrative unfolds, delineating the multifaceted interactions and functionalities inherent in this digital healthcare ecosystem. At the heart of this diagram lies the central actor, the end user, representing individuals seeking medical care and services. The primary use cases radiate from these central figure-like spokes, embodying the diverse functionalities encapsulated within the application.

The user initiates the sequence by accessing the application, setting in motion a series of interconnected use cases that collectively orchestrate the user experience. These include "Search Hospitals," where users input specific criteria such as location, speciality, or services required, initiating a dynamic process of information retrieval and filtering. The "View Hospital Details" use case further allows users to delve into comprehensive information about a particular healthcare facility, including services offered, reviews, and contact details.

The application responds adeptly to the user's needs through the "Check Availability" use case, facilitating real-time information on the availability of medical services, appointment slots, and relevant logistical details. A critical use case, "User Registration/Login," ensures a secure and personalized experience, allowing users to create accounts, manage profiles, and access personalized features. The "Review Hospitals" use case enables users to contribute feedback, fostering a collaborative and transparent environment that aids others in their decision-making process.

The administrative facet of the Hospital Finder Application is elucidated through use cases such as "Manage Hospitals" and "Update Availability," empowering healthcare institutions to maintain accurate and up-to-date information on their profiles. The "Generate Reports" use case serves as a pivotal tool for administrators to extract insights and analyze user interactions, facilitating strategic decision-making and system optimization. Interwoven throughout the use case diagram are essential system functionalities, including "Notification Management" to keep users informed about appointments, updates, and relevant information.

The "Integration with Maps" use case signifies the seamless integration of geographical information, enhancing user navigation and facilitating location-based searches. In essence, this elaborate use case diagram encapsulates the nuanced dance of interactions within the Hospital Finder Application, painting a vivid picture of user engagement, administrative control, and system functionalities. It transcends the boundaries of mere functionality documentation, offering a holistic representation of the interconnected web of features that collectively contribute to an efficient, user-centric, and technologically advanced healthcare discovery platform. As the user becomes the focal point of this digital symphony, each use case resonates as a note in a harmonious melody, orchestrating an immersive and empowering healthcare experience for individuals and healthcare providers alike.

6.4 Activity Diagram:

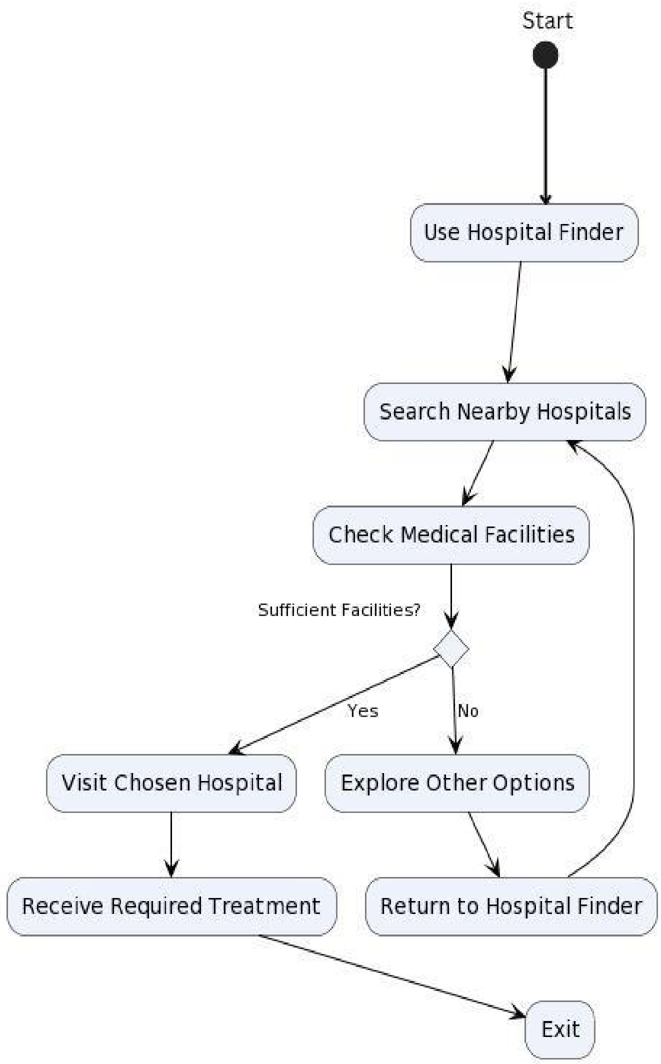


Figure 6.4

In the activity diagram the user will go to the website ‘Hospital Finder’ and click on the search bar to find the hospitals nearby by writing the disease, hospital or availability name. user will check whether the medical facility is sufficient or not:-

if yes:-

The user will choose the hospital, get the location details and map to navigate to the hospital using Google API and then exit.

if no:-

Users can search for other options or other filters like emergency medications, blood availability, tests, specialists etc. And again get the details and navigation to the hospitals.

6.5 Flowchart Diagram:

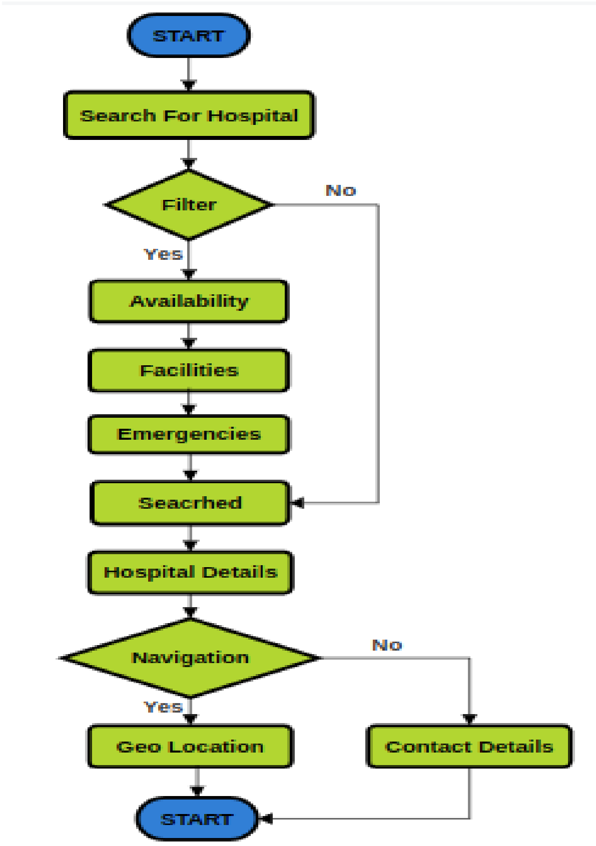


Figure 6.5

1. Start:-

- The process begins here.

2. Search for the hospital:-

2.1 Search Bar:-

- Search for the hospital name or disease name to get a hospital list.

2.2 Filter:-

If 'yes':-

- Searched based on Availability
- Searched based on Specialists
- Searched based on Medications

- Searched based on Test

If ‘no’:-

- Search hospital

3. Hospital Details:-

- Availability of Blood
- Availability of Emergency Medications
- Tests available
- Hospital address
- Hospital contact details
- Average hospital cost
- Distance of hospital from user
- Geolocation points of hospital

3.1 Navigation:-

if ‘yes’:-

- Hospital Geo Location and navigation using google API

If ‘no’:-

- Contact details of hospital with address and emergency number

4. End:-

- The process ends here

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

Figure 7.1

CHAPTER-8
OUTCOMES

8.1 Improved Healthcare Access:

Ensure that individuals can quickly and easily find appropriate healthcare facilities, reducing barriers to accessing essential medical care.

8.2 Enhanced Decision-Making:

Enable users to make well-informed decisions during medical emergencies, leading to better treatment outcomes.

8.3 Efficient Resource Utilization:

Promote the efficient utilization of hospital resources by matching patients with hospitals that can best address their needs, reducing overcrowding and wait times.

8.4 User Satisfaction:

Strive for high user satisfaction by providing a user-friendly, reliable, and accurate healthcare resource search experience.

8.5 Timely Assistance:

Ensure that individuals receive timely medical assistance during emergencies, potentially saving lives and reducing the severity of health conditions.

8.6 Reduced Stress and Anxiety:

Minimize the stress and anxiety experienced by individuals and their families when searching for healthcare options during critical situations.

8.7 Promotion of Health Equity:

Contribute to reducing healthcare disparities by offering equitable access to healthcare information and resources for all users.

8.8 Community Building:

Foster a sense of community by encouraging users to share information and experiences, creating a supportive network of individuals seeking medical care.

8.9 Data Accuracy:

Continuously strive for the highest level of data accuracy and reliability to maintain trust among users. Continuous Improvement: Regularly update and improve the application based on user feedback, technological advancements, and changes in healthcare services.

CHAPTER-9
RESULTS AND DISCUSSIONS

9.1 Results

9.1.1 Hospital Finder's Impact on Healthcare Access:

Hospital Finder, with its centralized real-time database, has demonstrated a transformative impact on emergency healthcare access. The application ensures the delivery of accurate, up-to-the-minute information on hospital facilities, specialist availability, and medical resources. This immediate accessibility to critical data has redefined decision-making during medical emergencies.

9.1.2 Advantages of Real-Time Data Implementation:

The real-time database approach in Hospital Finder enhances accuracy, responsiveness, and user experience. It has shown significant promise in reducing delays, improving response times, and empowering individuals to make well-informed decisions during critical medical situations.

9.1.3 User Empowerment and Potential Impact:

Hospital Finder not only provides nearby hospital options but also offers personalized recommendations, revolutionizing healthcare access. Its potential to save lives, reduce delays, and restore control to individuals during emergencies marks a significant leap in healthcare navigation.

9.2 Discussion

9.2.1 Significance of Real-Time Data in Healthcare:

The research underscores the pivotal role of real-time data in reshaping healthcare decision-making. Hospital Finder exemplifies how timely and accurate information is indispensable during emergencies, emphasizing the need for broader adoption of similar systems in healthcare.

9.2.2 Application's Contributions and Future Directions:

Hospital Finder's contributions extend beyond its current functionality. Integrating ambulance services, real-time bed availability tracking, and incorporating user feedback mechanisms are proposed future directions, showcasing the continuous evolution of the application for enhanced emergency healthcare.

9.2.3 Challenges and Alternative Approaches:

The paper acknowledges challenges in centralized real-time databases and explores alternative approaches like decentralized networks, blockchain, federated learning, health information exchanges, cloud-based solutions, and differential privacy techniques. Each approach presents unique advantages and challenges, enriching the discourse on emergency healthcare access strategies.

CHAPTER-10
CONCLUSION

10.1 Conclusion

Our effort has produced a system that is intended to provide a wealth of information on different hospitals. This cutting-edge tool makes it simple for users to get hospital-related data by delivering it online and through SMS in a seamless manner. Patients who integrate this healthcare application are provided with a carefully selected list of hospitals in their area, along with information about the services that each one offers. The software acts as a clearinghouse for relevant hospital data, including speciality offerings, service categories, and more. Its functionality includes the ability for users to send SMS requests for information, which has shown to be both functional and consistent with our expectations, as evidenced by its current 8.2 rating.

10.2 Future Enhancements

Understanding that there is always room for improvement, the application can progress even further with improvements focused on the needs of the user. One approach is to utilize surveys to get user feedback so that the app can be improved according to what the majority of users need most. A tool that allows users to evaluate and compare other hospitals and their offerings might be a crucial addition, enabling people to make well-informed selections. Creating a forum where people can talk about their experiences as patients could be a great way to help newcomers have a better understanding of what to expect from different medical facilities. These improvements are in line with our mission to continuously improve the usability and utility of the application, empowering and enlightening the community of people looking for healthcare.

REFERENCES

[1] Mehta, Heeket & Kanani, Pratik & Lande, Priya. (2019). Google Maps. International Journal of Computer Applications. 178. 41-46. 10.5120/ijca2019918791.

[2] Imteaj, Ahmed & Hossain, Md. (2016). A smartphone based application to improve the health care system of Bangladesh. 1-6. 10.1109/MEDITEC.2016.7835358.

[3] Iran J Public Health, Advances in Hospital Real-Time Location Systems, Vol. 48, No.4, Apr 2019, pp.593-602

[4] Jha, Pragya & Sahu, Madhusmita & Bisoy, Sukant & Sain, Mangal. (2022). Application of Model-Based Software Testing in the HealthCare Domain. Electronics. 11. 2062. 10.3390/electronics11132062.

[5] Int J Environ Res Public Health. 2022 Jun; 19(12): 7466. Published online 2022 Jun 17. doi: 10.3390/ijerph19127466

[6] Hussain, Muzammil & Al-Haiqi, Ahmed & Zaidan, A. & Bahaa, Bilal & Mat Kiah, Miss Laiha & Anuar, Nor & Abdulnabi, Mohamed. (2015). The Landscape of Research on Smartphone Medical Apps: Coherent Taxonomy, Motivations, Open Challenges and Recommendations. Computer methods and programs in biomedicine. 122. 10.1016/j.cmpb.2015.08.015.

[7] Fine-tuning a Large Language Model using Reinforcement Learning from Human Feedback for a Therapy Chatbot Application - Desirée Bill, Theodor Eriksson

[8] Selvam, P. P. (2017). Nearest Hospital Tracking and Disease Prediction.

[9] Gholamhosseini, Leila & Sadoughi, Farahnaz & SAFAEI, Aliasghar. (2019). Hospital Real-Time Location System (A Practical Approach in Healthcare): A Narrative Review

Article. Iranian Journal of Public Health. 48. 593-602. 10.18502/ijph.v48i4.980.

[10] Biswajeet Sethi, Samaresh Mishra, Prasant ku. Patnaik. "A Study of NoSQL Databases." School of Computer Engineering, KIIT University, Bhubaneswar, India.

[11] Khawas, Chunnu & Shah, Pritam. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications. 179. 49-53. 10.5120/ijca2018917200.

APPENDIX-A
PSEUDOCODE

1. CalculateRoadDistance Function:-

1. Initialize Headers:

- Create a map called requestHeaders.
- Add key-value pairs for required headers: "Access-Control-Allow-Headers" and "Access-Control-Allow-Origin".

2. Construct URL:

- Create a URL string using Google Maps API.
- Include parameters for origins (lat1, lon1) and destinations (lat2, lon2).
- Specify travel mode as driving.
- Utilize an API key for authentication.

3. Make API Request:

- Use the constructed URL and headers to send an HTTP GET request.
- Await the response.

4. Handle Response:

- Check if the response status code is 200 (OK).
- Decode the JSON response body.
- Extract necessary data:
- Access the 'rows' array.
- Retrieve 'elements' from the first row.
- Get the distance value in meters.
- Print the distance.
- Convert the distance from meters to kilometers by dividing by 1000.
- Return the distance in kilometers.

4. Error Handling:

- If the response status code is not 200, throw an exception indicating the error and including the status code.

2. SearchLocaldata Function:-

1. Search Function:

- Take a list of maps, data and a searchValue string as inputs.

2. Initialize Results List:

- Create an empty list of results to store matched data.

3. Filter Data:

- For each map (doc) in 'allData':
- Check if the 'Emergency-Medication' list contains 'searchValue'.
- Check if the 'Blood-Availability' list contains 'searchValue'.
- Check if the 'Test-Map' keys or 'Specialists' keys contain a part or whole of 'searchValue'. Check if the 'Name' field contains 'searchValue'.
- Include the map if any of these conditions match.

4. Calculate Average Cost:

- Iterate through each result in 'search-results'.
- If 'Test-Map' exists in the result:
- Calculate the average cost from numeric values within the map values.
- Update the result with the calculated average cost ('Avg-Cost' field).
- Increment a counter and accumulate values for averaging. 5. Update UI (Assuming Flutter's 'setState'):
- Use Flutter's 'setState' method to update the UI with the newly calculated average cost for each matching result.

3. Medisearch Pseudo Code:-

1. Start:

2. Initialize the user's location using GPS or manual input.

3. Ask users for required specialization, blood availability, or facilities needed.

4. Store user inputs: specialization, blood availability, facilities.

5. Retrieve hospital databases with details: name, location, specializations, blood availability, and facilities.

6. Calculate the distance between the user's location and each hospital in the database.

7. Filter hospitals based on user inputs:

a. Create an empty list to store matching hospitals.

b. For each hospital in the database:

- Check if the hospital offers specified specialization or required facilities.
- If blood availability is specified, verify the hospital has a required blood type.
- If conditions match, add a hospital to the list of matching hospitals.

8. Sort list of matching hospitals based on proximity:

a. Use distance calculation to arrange hospitals closest to the user's location.

9. Display sorted hospitals' information:

a. Show names, locations, and essential details of each hospital.

10. Prompt user interaction:

a. Ask the user to select a hospital or click for more details.

11. User selects a hospital:

a. Retrieve detailed information from the database for the selected hospital.

b. Display details: address, contact, services offered.

c. Provide options for navigation or directions to the selected hospital using GPS or maps.

12. End.

APPENDIX-B
SCREENSHOTS

Homepage :

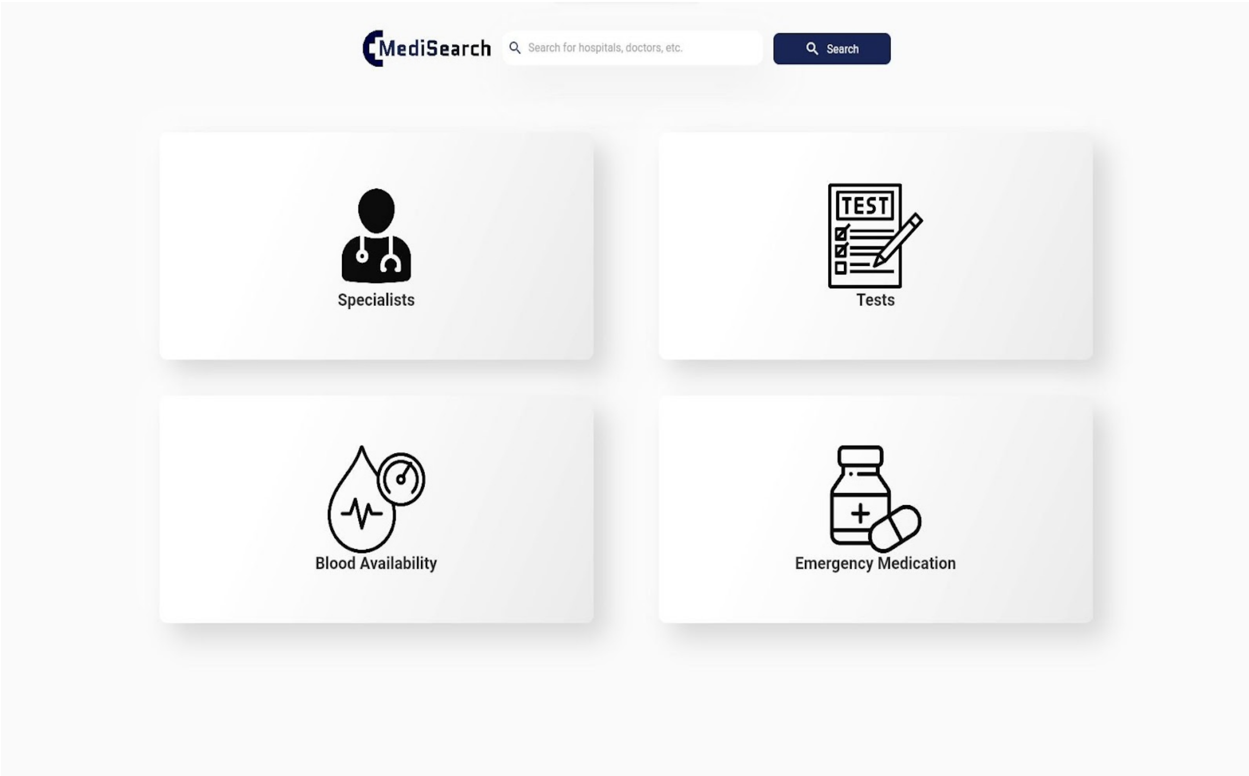


Figure 11.1

This is the Medisearch app's home page, where users can search for hospitals, check the availability of blood, receive emergency prescriptions, find specialists, schedule tests, and more.

Steps:-

1. Home Page > Blood Availability
2. Select Hospital > Shows the details of the hospital
3. Click on Distance > Shows location address and Contact Details Location in Map (using Google API)

Search Page:

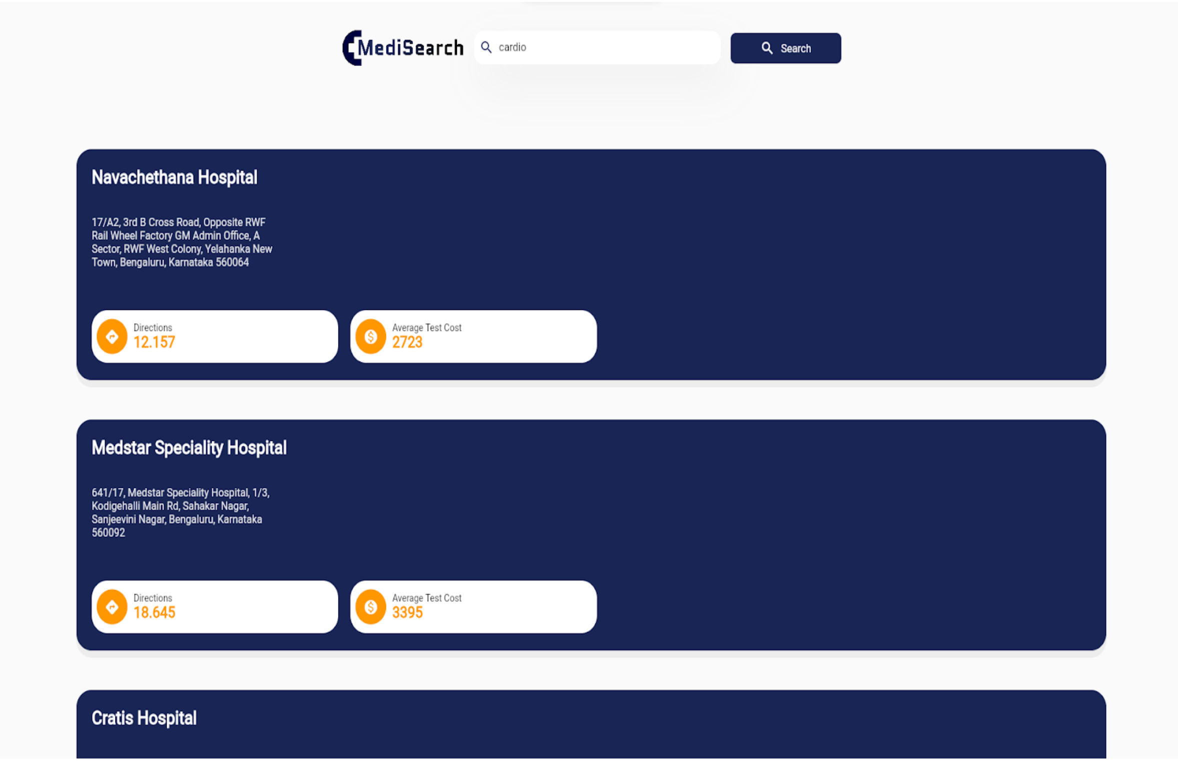


Figure 11.2

The user will find a list of nearby hospitals on this page, along with information on their average cost and distance from their current location to the hospital.

Steps:-

1. Home Page > Click on Search bar
2. Enter Disease or Hospital name > Shows the list of hospitals
3. Click on any Hospital > Show location address and Contact Details
4. Location in Map (using Google API)

Hospital Details:

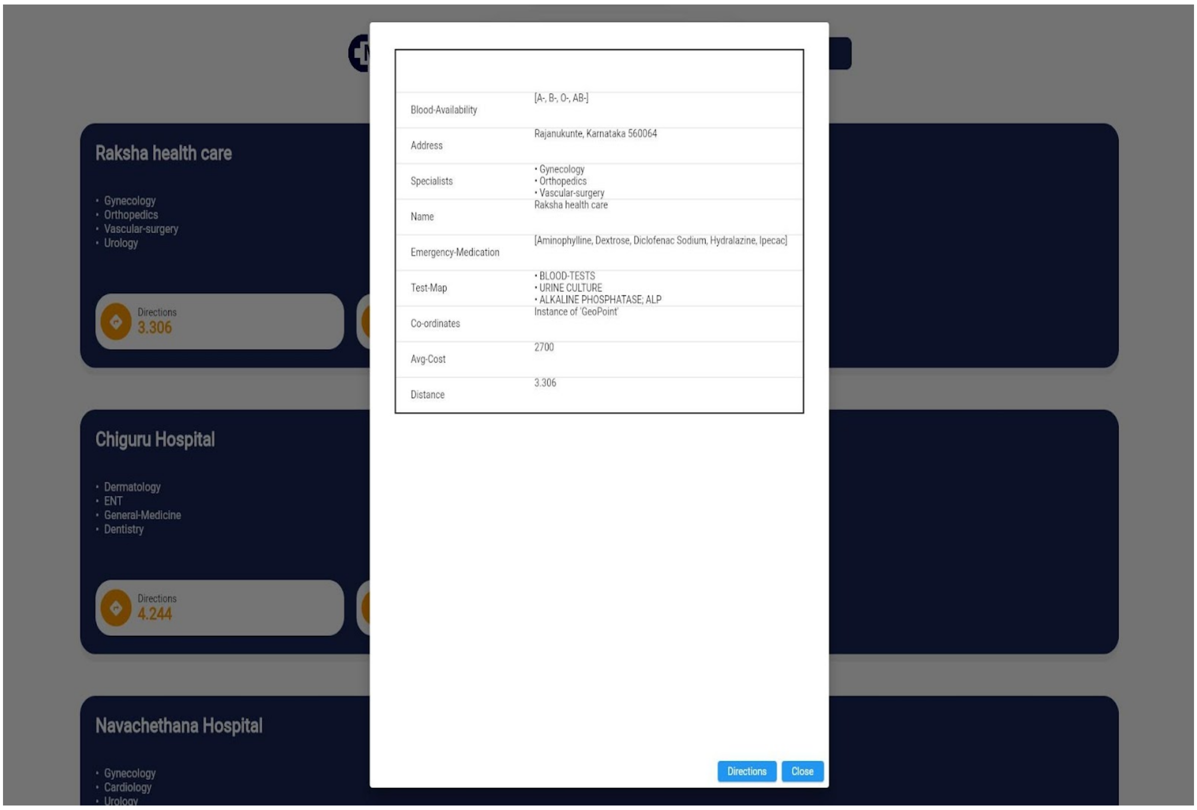


Figure 11.3

The user will find information about surrounding hospitals on this page, including the facilities' coordinates, addresses, tests, bed availability, average cost, distance, and specialists.

Steps:-

1. Home Page > Blood Availability
2. Select Hospital > Shows the details of the hospital
3. Click on Distance > Shows location address and Contact Details Location in Map (using Google API)

Specialists Page:

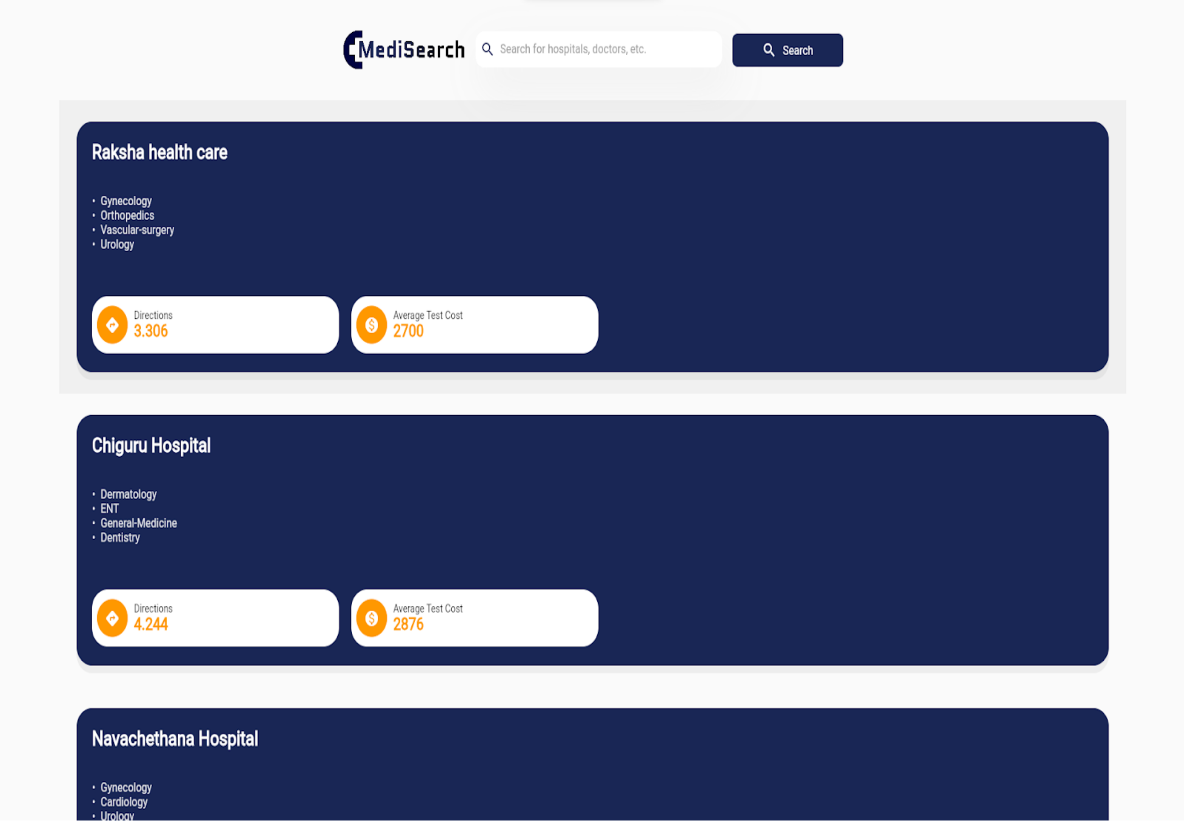


Figure 11.4

In order to save people from having to travel from place to place in search of specialization availability during an emergency, this page provides a list of all the surrounding hospitals together with a list of all the specialists that are available in those institutions, such as cardiologists and orthopedic specialists.

Steps:-

1. Home Page > Specialists
2. Select Hospital > Shows the details of specialists present in the hospital
3. Click on Distance > Shows location address and Contact Details
4. Location in Map (using Google API)

Tests Page:

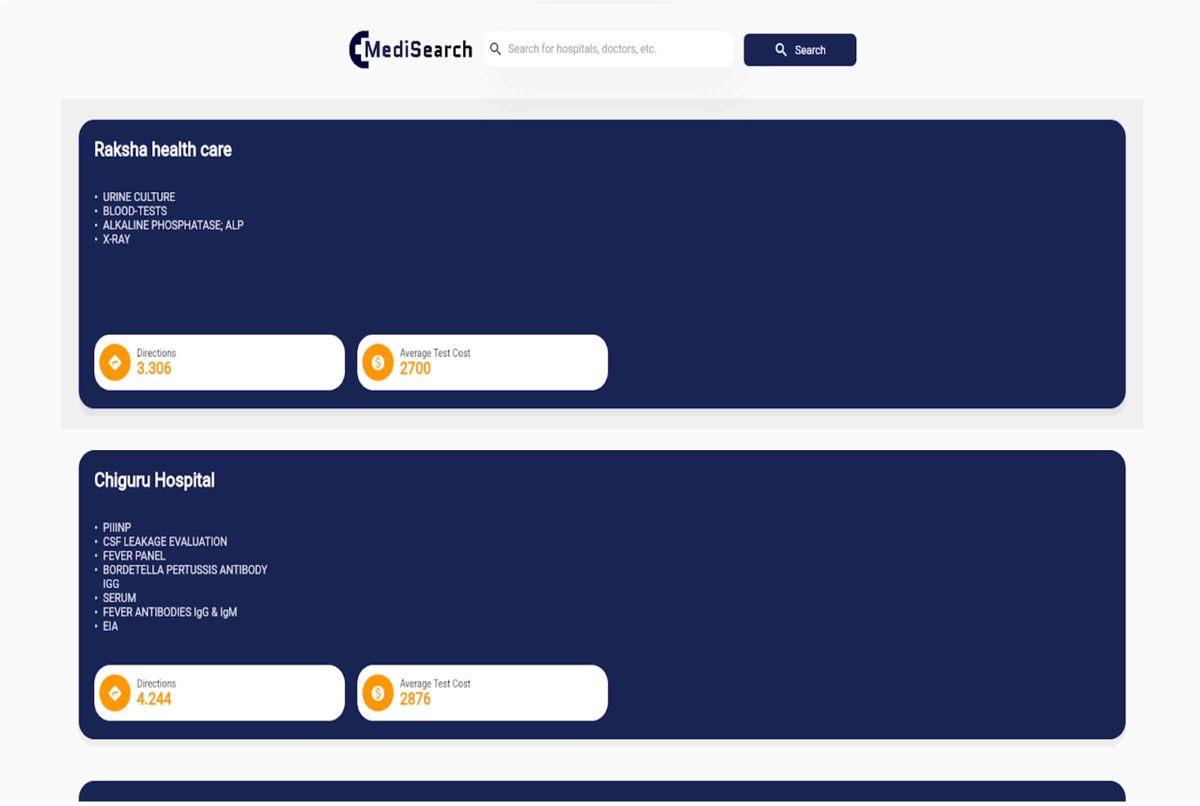


Figure 11.5

The user will find a list of local hospitals on this page, along with information on all the blood types that are available, their usual costs, and the distance between their current location and the hospital.

Steps:-

1. Home Page > Test
2. Select Hospital > Shows the details of Tests done in hospitals with their costs
3. Click on Distance > Shows location address and Contact Details Location in Map (using Google API)

Blood Availability Page:

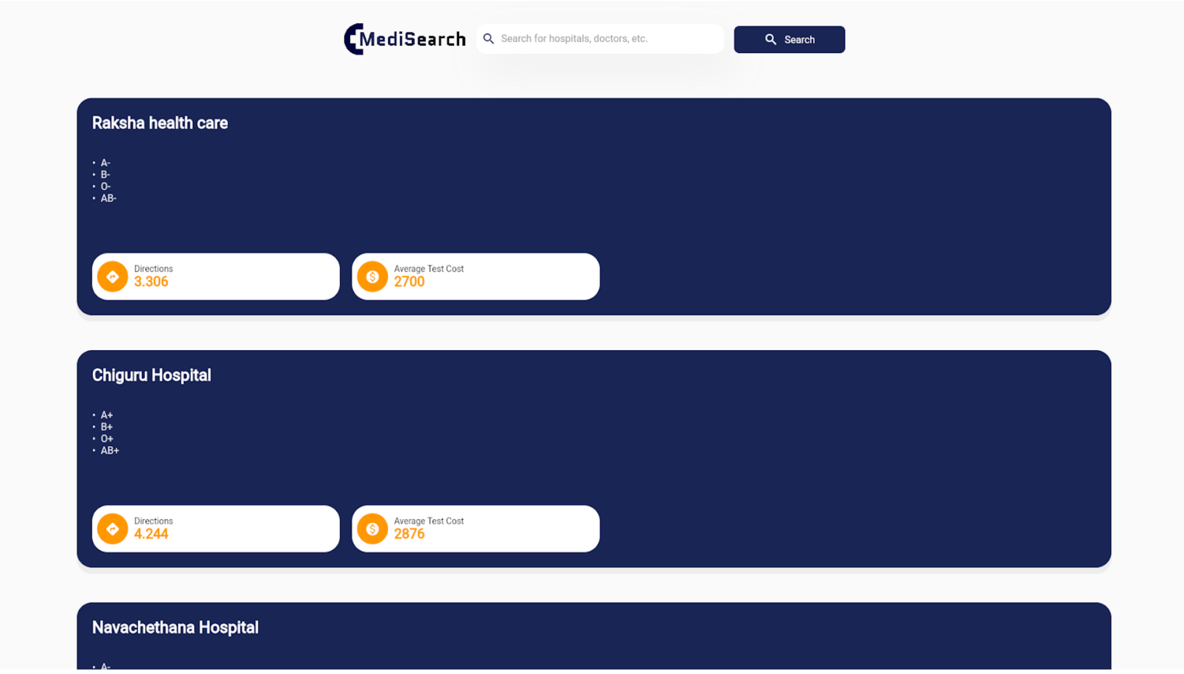


Figure 11.6

The user will find a list of local hospitals on this page, along with information on all the blood types that are available, their usual costs, and the distance between their current location and the hospital.

Steps:-

1. Home Page > Blood Availability
2. Select Hospital > Shows the details of the hospital
3. Click on Distance > Shows location address and Contact Details
4. Location in Map (using Google API)

Emergency Medications Page :

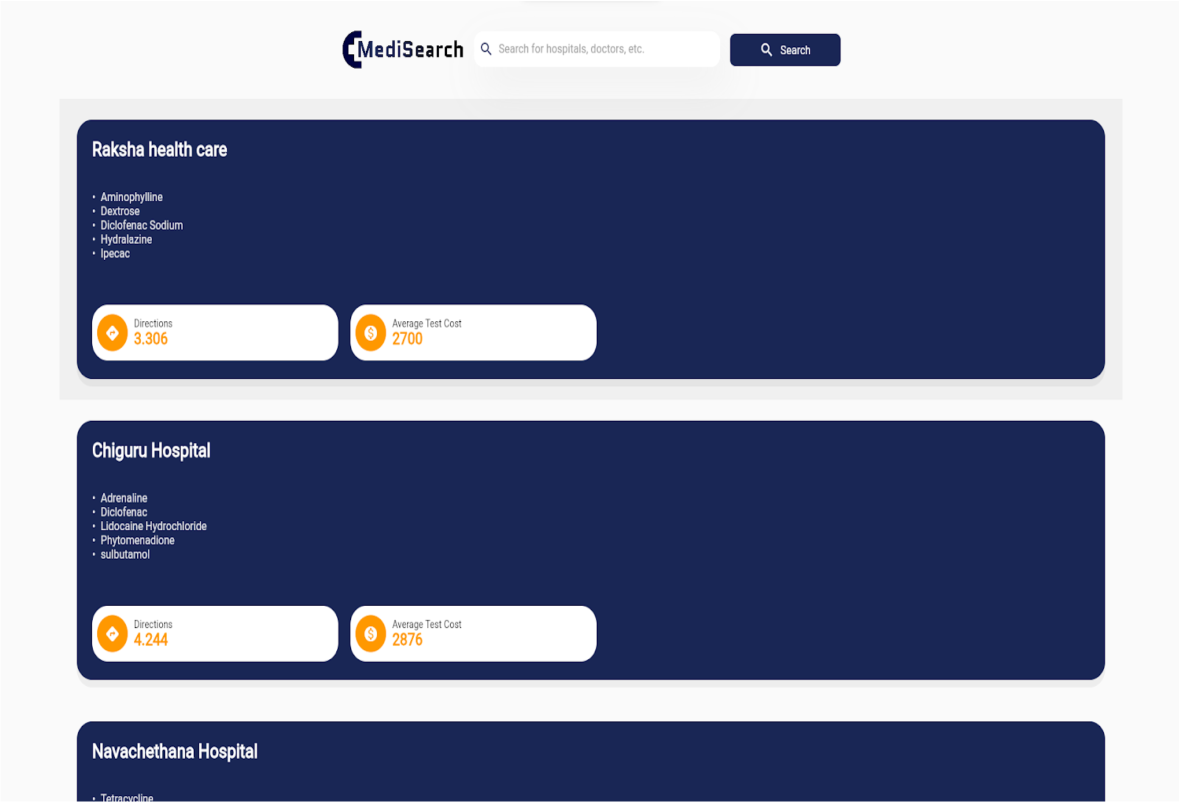


Figure 11.7

The user will find a list of local hospitals on this page, along with information about all of the emergency medications that are available, their typical cost, and the distance between their current location and the hospital.

Steps:-

1. Home Page > Emergency Medications
2. Search for medication or select hospital > Shows the details of emergency medication available in the hospital at the time of emergency
3. Click on Distance > Shows location address and Contact Details Location in Map (using Google API)

Backend(DataBase) :

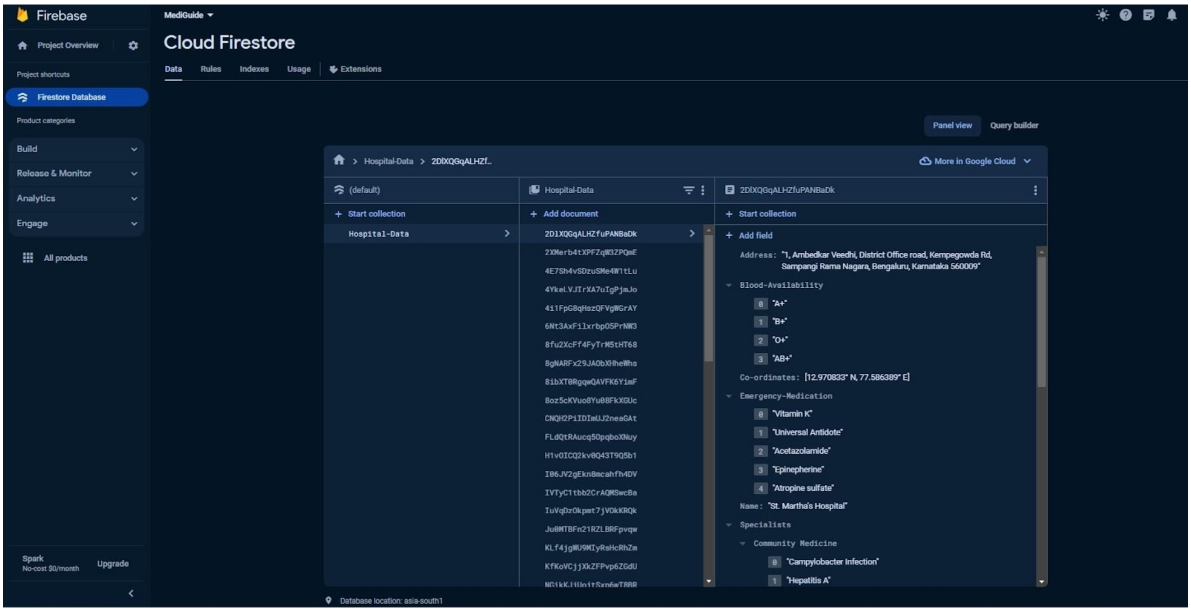


Figure 11.8

The Firebase Query Panel refers to a part of Firebase services where you can interactively execute queries against your Firebase Realtime Database or Cloud Firestore.

1. **Firebase Realtime Database:** If you're using the Realtime Database, the Query Panel allows you to test and execute queries against your database. You can filter, order, and limit the data retrieved from your database using various query parameters like 'orderByChild', 'equalTo', 'limitToFirst', etc. This helps in testing queries before implementing them in your application.
2. **Cloud Firestore:** For Firestore, the Query Panel allows you to create and test queries against your Firestore collections. Firestore provides more complex querying options compared to the Realtime Database, such as compound queries, filtering, sorting, and limiting data retrieval based on certain conditions.

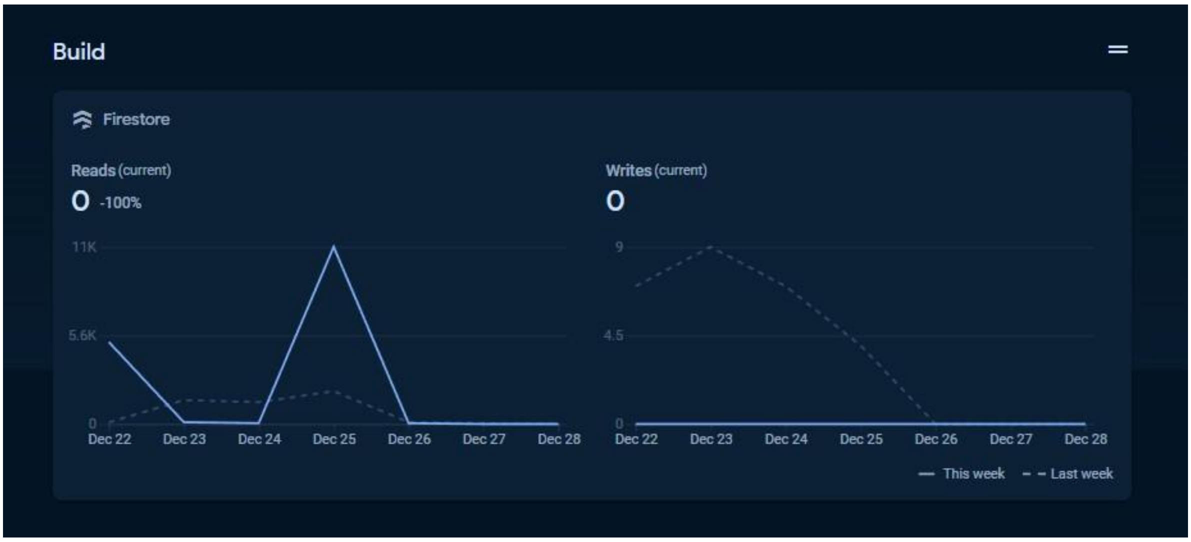


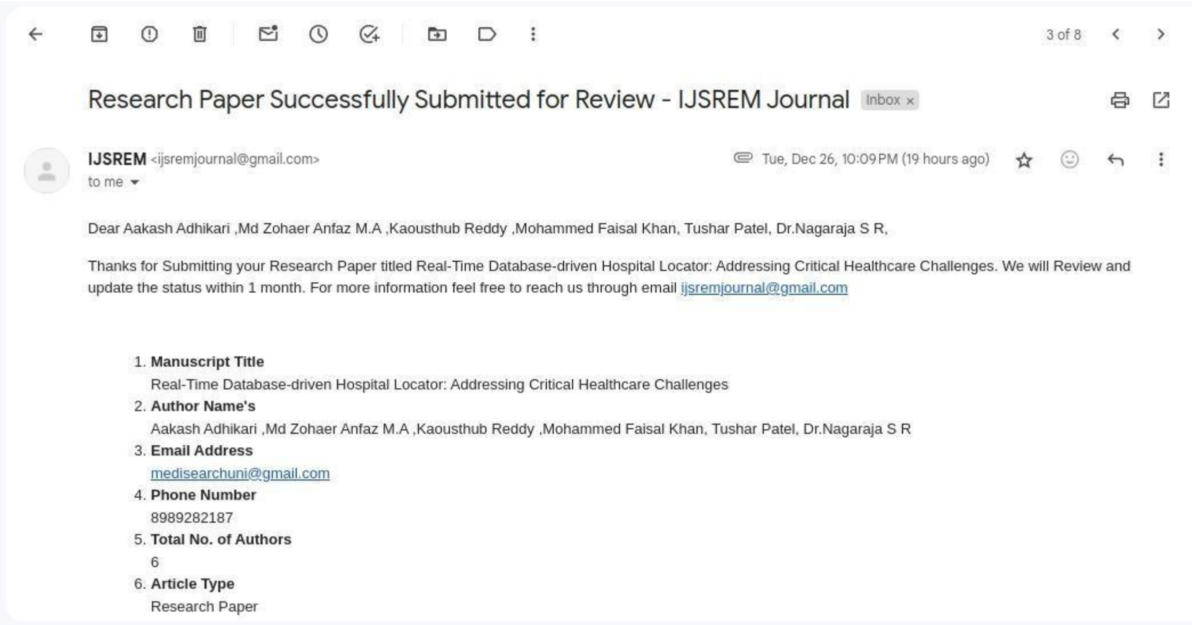
Figure 11.9

The Firebase Usage Panel, also known as the Firebase Console, is a web interface provided by Google Firebase that allows developers and administrators to manage and monitor their Firebase projects. It serves as a control center for various Firebase services and provides insights into usage, analytics, performance, and more. Here are some key features and functionalities of the Firebase Usage Panel:

1. **Project Overview:** It gives you a summary of your Firebase project, including the services enabled, recent activity, and important metrics like active users, storage usage, and database operations.
2. **Authentication:** You can manage user authentication settings, such as enabling different sign-in methods like email/password, Google, Facebook, etc. It allows you to view user details, manage user accounts, and set up authentication rules.
3. **Realtime Database / Firestore:** These are Firebase's NoSQL databases. The panel lets you manage and view data, set security rules, and monitor usage and performance metrics.
4. **Cloud Storage:** Firebase offers cloud storage for user-generated content like images, videos, and files. The usage panel allows you to upload files, manage storage buckets, set access rules, and monitor usage.

APPENDIX-C
ENCLOSURES

1. Conference Paper Presented Certificates of all students.



3 of 8

Research Paper Successfully Submitted for Review - IJSREM Journal Inbox

IJSREM <ijsremjournal@gmail.com>
to me

Tue, Dec 26, 10:09 PM (19 hours ago)

Dear Aakash Adhikari ,Md Zohaer Anfaz M.A ,Kaousthub Reddy ,Mohammed Faisal Khan, Tushar Patel, Dr.Nagaraja S R,

Thanks for Submitting your Research Paper titled Real-Time Database-driven Hospital Locator: Addressing Critical Healthcare Challenges. We will Review and update the status within 1 month. For more information feel free to reach us through email ijsremjournal@gmail.com

1. **Manuscript Title**
   Real-Time Database-driven Hospital Locator: Addressing Critical Healthcare Challenges
2. **Author Name's**
   Aakash Adhikari ,Md Zohaer Anfaz M.A ,Kaousthub Reddy ,Mohammed Faisal Khan, Tushar Patel, Dr.Nagaraja S R
3. **Email Address**
   medisearchuni@gmail.com
4. **Phone Number**
   8989282187
5. **Total No. of Authors**
   6
6. **Article Type**
   Research Paper

Research Paper ID : IJSREM27772

Manuscript Title

Real-Time Database-driven Hospital Locator: Addressing Critical Healthcare Challenges

Author Name's Aakash Adhikari ,Md Zohaer Anfaz M.A ,Kaousthub Reddy ,Mohammed Faisal Khan, Tushar Patel, Dr.Nagaraja S R

Email Address

medisearchuni@gmail.com

2. Include certificate(s) of any Achievement/Award won in any project related event.


ISSN: 2582-3930

ACCEPTANCE CERTIFICATE

Impact Factor: 8.176
DOI Prefix: 10.55041

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

An Open Access Scholarly Journal || Index in major Databases & Metadata

We are pleased to inform you that your Research Paper titled

Real-Time Database-driven Hospital Locator: Addressing Critical Healthcare Challenges

has been ACCEPTED for publication in

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

VOLUME 07 ISSUE 12 DECEMBER - 2023

We are delighted to see your commitment & hardwork to share your research is being recognized. We look foreward to helping you with all of your publication needs. Thank you for choosing IJSREM!!

Crossref CiteFactor

Editor-in-chief
IJSREM

www.ijsrem.com ijsremjournal@gmail.com

Research Paper ID: IJSREM27772

The Research Paper issued by the "Internal Journal of Scientific Research In Engineering & Management (IJSREM)" comes under UGC care.

Publishing Authors Are:-

Aakash Adhikari, Md Zohaer Anfaz M.A, Kaousthub Reddy, Mohammed Faisal Khan, Tushar Patel, Dr.Nagaraja S R

3. Similarity Index / Plagiarism Check report clearly showing the Percentage (%). No need for a page-wise explanation.

Research Paper Plagiarism Check :

Real-Time Database-driven Hospital Locator: Addressing Critical Healthcare Challenges

ORIGINALITY REPORT

5%	3%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

Project Report Plagiarism Check :

MEDISEARCH

ORIGINALITY REPORT

7%	7%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS