A blue and white logo

Description automatically generated

**A**

**Project Report**

on

**SERVICELY**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

in

**COMPUTER SCIENCE**

**Submitted By**

Aaveg Tomar (2000290120003)

Arpit Puri (2000290120040)

Abhi Singhal (2000290120006)

**Under the supervision of**

Prof. Ajay Kumar Shrivastava

Professor

**Department of Computer Science**

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**May, 2024**

# DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: Arpit Puri

Roll number: 2000290120040

Signature:

Name: Aaveg Tomar

Roll number: 2000290120003

Signature:

Name: Abhi Singhal

Roll number: 2000290120006

Date:

# A blue and white logo Description automatically generated

# CERTIFICATE

This is to certify that Project Report entitled “Servicely” which is submitted by Aaveg Tomar, Arpit Puri, Abhi Singhal in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

**Date: Supervisor**

Dr. Ajay Kumar Shrivastava Professor

Department of Computer Science

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor **Dr. Ajay Kumar Shrivastava**, Head of the Department of Computer Science, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name: Arpit Puri

Roll number: 2000290120040

Signature:

Name: Aaveg Tomar

Roll number: 2000290120003

Signature:

Name: Abhi Singhal

Roll number: 2000290120006

|  |  |
| --- | --- |
| TABLE OF CONTENTS | Page No. |

# 

|  |  |
| --- | --- |
| DECLARATION |  |
| CERTIFICATE |  |
| ACKNOWLEDGEMENTS |  |
| ABSTRACT…………………………………………………………….. | i |
| LIST OF FIGURES …………………………..…………………….….. | ii |
| LIST OF TABLES……………………………………………….….…. | iii |
| LIST OF ABBREVIATIONS……………………………………..……. | iv |
|  |  |
| CHAPTER 1 INTRODUCTION  1.1 Introduction to Project……………………………………………  1.2 Project Category …………………………………………………  1.3 Objectives…………………………………………………………  1.4 Structure of Report……………………………………………….  CHAPTER 2 LITERATURE REVIEW  2.1 Literature Review……………………………………………….…  2.2 Research Gaps…………………………………………………..…  2.3 Problem Formulation.………………………………………………  CHAPTER 3 PROPOSED SYSTEM  3.1 Proposed System……………………………………………...……  3.2 Unique Features of the System…………………………….………  CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION  4.1 Feasibility Study (Technical, Economical, Operational) ……...……  4.2 Software Requirement Specification…………………………...……  4.2.1 Data Requirement…………………………………………...……  4.2.2 Functional Requirement…………………………………………  4.2.3 Performance Requirement………………………………………  4.2.4 Maintainability Requirement……………………………………  4.2.5 Security Requirement……………………………………………  4.3 SDLC Model Used……………………………………………...……  4.4 System Design………………………………………………….……  4.4.1 Data Flow Diagrams………………………………………….…  4.4.2 Use Case Diagrams………………………………………………  4.5 Database Design……………………………………………………...  CHAPTER 5 IMPLEMENTATION  5.1 Introduction Tools and Technologies Used…………………………  CHAPTER 6 TESTING, AND MAINTENANCE  6.1 Testing Techniques and Test Cases Used………………………………  CHAPTER 7 RESULTS AND DISCUSSIONS  7.1 Description of Modules with Snapshots………………………….…  7.2 Key findings of the project……………………………………….…  7.3 Brief Description of Database with Snapshots………………………  CHAPTER 8 CONCLUSION AND FUTURE SCOPE …………………  REFERENCES ………….………………………………………………  Research Paper Submission Proof……….………………………………… | 1-5  1  2  2  4  6-9  6  8  9  10-12  10  10  13-31  13  16  16  17  18  19  20  22  23  24  28  29  32-35  32  36-41  42-49  42  46  48  50-51  52  54 |

# ABSTRACT

The Home Based Service Provider platform represents a significant leap forward in the realm of home services, redefining how users connect with and engage service providers. Built on cutting-edge technologies like React JS, Tailwind CSS, Node.js, Express.js, MongoDB, and Socket.IO, this platform offers a comprehensive solution that combines advanced functionality with a seamless user experience. At its core, React JS and Tailwind CSS form the foundation of the platform's front-end. React JS, known for its component-based architecture, enables developers to create reusable UI components, resulting in a highly modular and scalable application. On the back-end, Node.js and Express.js drive the platform's server-side logic and API endpoints. Node.js, with its event-driven, non-blocking I/O model, delivers exceptional performance and scalability, crucial for handling a large number of concurrent requests. Express.js, a minimalist web framework for Node.js, simplifies the process of building robust and efficient web applications, providing features like routing, middleware support, and template engines. The integration of Socket.IO takes the platform's capabilities to the next level by enabling real-time, bidirectional communication between users and service providers. With Socket.IO, users can engage in instant messaging with service providers, discuss project details, share files, and receive timely updates on service status. The platform's user interface is designed to be both informative and engaging. Users can easily search for service providers based on location, services offered, ratings, and reviews. Users can make informed decisions by viewing service provider ratings and feedback from previous customers, fostering transparency and trust.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Figure No.** | **Description** | **Page No.** |
| 1 | 4.1 | DFD level 0 | 26 |
| 2 | 4.2 | DFD level 1 | 27 |
| 3 | 4.3 | Use Case Diagram | 28 |
| 4 | 4.4 | Workflow diagram | 31 |
| 5 | 6.1 | Test Case: 1 Customer Login | 39 |
| 6 | 6.2 | Test Case: 2 Provider Registration | 40 |
| 7 | 6.3 | Test Case: 3 Provider Login | 40 |
| 8 | 6.4 | Provider Dashboard | 41 |
| 9 | 6.5 | Chat Dashboard | 41 |
| 10 | 7.1 | User Home Page | 43 |
| 11 | 7.2 | Service Providers | 44 |
| 12 | 7.3 | User Sign in Page | 45 |
| 13 | 7.4 | Chat Console | 45 |
| 14 | 7.5 | Service Provider Registration Page | 46 |
| 15 | 7.6 | Database Collection 1 | 49 |
| 16 | 7.7 | Database Collection 2 | 49 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Table No** | **Description** | **Page No.** |
| 1 | 5.1 | Testing Professional Registration | 36 |
| 2 | 5.2 | Testing Professional Login | 37 |
| 3 | 5.3 | Testing Customer Registration | 37 |
| 4 | 5.4 | Testing chat dashboard | 38 |
| 5 | 5.5 | Testing Customer Login | 39 |

# LIST OF TABLES

# LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Abbreviations** | **Full Form** |
| 1 | MERN | Mongo Db, Express JS, React JS, Node JS |
| 2 | I/O | Input/output |
| 3 | NOSQL | No Structure Query Language Database |
| 4 | APIs | Application Program Interface |
| 5 | REST | Representational State Transfer |
| 6 | CSS | Cascading Style Sheet |

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

The Home Based Service Provider platform stands as a beacon of modernization in the realm of home services, intertwining an intricate web of cutting-edge technologies to orchestrate a seamless and intuitive user journey. Anchored by React JS and Tailwind CSS on the frontend, the platform's interface not only dazzles with its sleek aesthetics but also ensures unparalleled responsiveness and adaptability across diverse devices and screen sizes. Beneath this polished exterior lies a robust backend infrastructure powered by the dynamic duo of Node.js and Express.js, fortified by the versatility of MongoDB for efficient data management and storage. This formidable technological foundation not only guarantees the platform's scalability and performance but also lays the groundwork for swift iteration and evolution of features. A standout facet of the platform is its unwavering commitment to user empowerment and engagement, exemplified by meticulously curated provider profiles that furnish users with pertinent information such as experience, certifications, and customer ratings. Moreover, the platform's adept utilization of personalized advertisements ensures that users are seamlessly matched with the services they require, enhancing convenience and expediting the decision-making process. Central to the platform's ethos is its integrated chat functionality, powered by Socket.IO, which fosters real-time communication between users and service providers, nurturing trust and transparency at every juncture of the service journey. By cultivating a symbiotic feedback loop wherein service providers can view and respond to user ratings and reviews, the platform not only catalyzes continual improvement but also fortifies the bedrock of trust upon which lasting relationships are forged. In essence, the Home Based Service Provider platform stands as a testament to the transformative potential of technology when harnessed in service of enhancing user experience and facilitating seamless access to essential home services.

## 1.2 Project category

The Home Based Service Provider platform falls into the categories of "Internet Based" and "Application or System Development," as it leverages modern technologies to connect users with local service providers online. Through its robust backend and sleek frontend design, it embodies the essence of contemporary software development, optimizing user experience while facilitating efficient communication. By integrating React JS, Tailwind CSS, Node.js, Express.js, MongoDB, and Socket.IO, the platform represents a pinnacle of internet-based solutions, exemplifying the evolution of the web application development in the digital era.

## 1.3 Objectives

**1**. The project's overarching objective is to revolutionize the process of accessing local services by eliminating the need for intermediaries. By cutting out the middleman, the platform aims to streamline the connection between users and service providers, fostering direct interaction and enhancing efficiency in service delivery.

**2**. Central to the platform's ethos is the provision of an unrestricted space where users can seamlessly engage with service providers without encountering any barriers or impediments. This open communication channel not only facilitates swift and transparent transactions but also cultivates a sense of trust and accountability between both parties involved.

**3**. Moreover, the platform prioritizes flexibility in pricing structures, empowering users to negotiate directly with service providers to arrive at a fair and mutually beneficial agreement. This approach not only ensures transparency in pricing but also fosters a collaborative environment where both users and providers feel valued and respected.

**4**. To further enhance user experience, the platform offers robust search functionality, allowing users to effortlessly locate service providers based on their geographic proximity. This geographical targeting feature not only simplifies the process of finding suitable service providers but also optimizes efficiency by minimizing search time and effort.

**5**. At its core, the project aims to simplify and expedite the process of connecting individuals with the services they need, emphasizing ease and convenience as paramount considerations. By providing a user-friendly interface and seamless user experience, the platform endeavors to ensure that obtaining essential services is both effortless and expedient for all users.

**6**. Additionally, the project endeavors to bolster local economies by championing and promoting the services of homegrown businesses to a broader audience. By showcasing local service providers on the platform, the project aims to amplify their visibility and reach, thereby stimulating growth and sustainability within the local business ecosystem.

**7**. Furthermore, the project seeks to forge strategic partnerships with trusted service providers to diversify and expand the range of services offered on the platform. By collaborating with reputable businesses and professionals, the platform aims to enrich its service offerings and cater to a wider array of user needs, thereby enhancing its value proposition and utility.

**1.4 Structure of Report**

In Chapter 1, we provide a succinct overview of our Home Service Provider Application, outlining its objectives and significance. We highlight the project's aim to develop an efficient personal Service Provider Application system and its potential impact on user convenience and productivity.

In Chapter 2, the literature review delves into existing research and discourse surrounding various Home Service Provider Application. We analyze the evolution of different platforms, discussing advancements, user experiences, and pertinent security and privacy concerns.

In Chapter 3, outlines our proposed voice assistant system, detailing its functionalities and the technologies utilized for its implementation. We elucidate the system's capabilities and how it addresses user requirements.

In Chapter 4, we conduct a feasibility study of our proposed system and provide a detailed software requirement specification. We discuss the chosen SDLC model and its relevance to the project's development.

In Chapter 5, In this section, we present an overview of the languages, tools, and technologies employed for implementing the voice assistant system. We delve into the libraries and algorithms utilized, offering insights into the implementation process and key modules.

In Chapter 6, This chapter focuses on testing techniques and methodologies utilized to ensure the functionality and reliability of our voice assistant system. We discuss unit testing, integration testing, functional testing, usability testing, and performance testing, along with details of the test environment.

In Chapter 7, Here, we provide a summary of the various modules comprising the Home Service Provider Application and discuss the outcomes of our project. We analyze the results, interpret their implications, and offer insights into the findings.

In Chapter 8, In the conclusion, we encapsulate the project's objectives, achievements, and contributions. We reflect on key findings, discuss potential future development directions, and underscore the significance of our work.

In Chapter 9, The references section includes citations of relevant research papers, articles, and sources used throughout the report, ensuring transparency and academic integrity.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Literature Review

**According to paper [1],[An online system for household services]** This Paper describes on the provision of domestic services at the doorstep just with one press. Many services supplied and the way the ordering and shipping of services takes place. This gadget may be utilized by any user who requires to look for family benefits through an electronic framework or a portable application. To give a confirmed and approved login module for the clients like assistance searchers, specialist organizations and the administrator, by giving fitting certifications at the hour of enrollment. A meticulously structured web environment tailored for clients offers unparalleled convenience in accessing a diverse range of services, elevating the overall user experience. The primary objective of web-based platforms for household services is to seamlessly deliver home assistance with a simple click. The emphasis lies in offering an array of services in a harmonious and well-organized manner. Verified users seeking assistance through the website can effortlessly utilize the online platform for their home service needs.

**Authored by Amruta Amol Bhawarthi, Kaustubh Muley, Kavya Amrutkar, Devendra Kawade, Anushka Kausadikar, Ayush Kawane, and Kaustubh Singh [2],** this paper introduces a website for home service providers. The platform addresses the escalating demand for efficient home maintenance solutions by offering a broad spectrum of services such as repairs, cleaning, and installations, all accessible through online booking with transparent pricing. Emphasizing user-friendly design and mobile compatibility, the system aims to streamline the process of accessing professional home services while enhancing customer satisfaction. Through an analysis of methodology, existing systems, and future prospects, the research underscores the potential of the proposed website to cater to evolving customer needs in the home services sector, providing a convenient and reliable solution for homeowners seeking quality assistance.

**Authored by Divya Chauhan and K. L. Bansa [3l,** this paper delves into the advantages of NOSQL databases, particularly focusing on MongoDB, in handling the challenges posed by the exponential growth and variety of data. With traditional relational databases struggling to cope with such demands, NOSQL databases offer scalability and efficiency in data storage and access. MongoDB, a leading document database, is explored through a case study, elucidating its representation format and querying processes, while comparing it to relational databases. The paper categorizes NOSQL databases into key-value, column-oriented, document, and graph stores, highlighting their respective features and tools. Additionally, it reviews related works and discusses MongoDB's features like flexibility, rich query language, sharding, ease of use, high performance, and availability. Through JSON representation examples and database query comparisons, the paper provides insights into MongoDB's practical implementation. Overall, it offers a comprehensive understanding of MongoDB's capabilities and its significance in addressing contemporary data management challenges.

**Shahrzad Shahriari, Mohammadreza Shahriari, Saeid gheiji [4]** tells about Socket.IO emerges as the ideal solution for real-time chat within our service provider-consumer web application. It establishes persistent two-way connections, enabling instant message delivery and a natural chat experience. By pushing data only when needed, Socket.IO reduces server load and scales effectively for a growing user base. Furthermore, its flexibility in message formats and seamless integration with various technologies enhance development ease. Ultimately, Socket.IO empowers real-time communication with minimal latency, leading to a positive user experience efficient interaction, faster issue resolution.

**In paper by Kunal Bhalgat, Sayali Desai, Rajeshri Mayanaikar, Aaditya Pardeshi[5]**, and Prof. Bhagyashree Dhakulkar, the "MAZDOOR" online application for household services is introduced. This innovative system connects users with various labor services such as electricians, plumbers, carpenters, and more. Through detailed labor profiles, users can easily find and contact service providers in their area. The system facilitates seamless transactions and feedback, enhancing the user experience. Focus on reducing unemployment and providing affordable services, "MAZDOOR" aims to be user-friendly, available in multiple languages, and adaptable for future service expansions.

## 2.2 Research Gaps

## Despite the comprehensive review of existing literature and the detailed description of the proposed Home Based Service Provider platform, several research gaps remain to be addressed:

## 1.Quality Assurance Mechanisms: Although the platform integrates user-driven feedback and rating systems, additional research could explore novel approaches to quality assurance, such as AI-driven sentiment analysis or proactive service monitoring, to uphold service standards and mitigate potential issues proactively.

## 2.Market Penetration Strategies: The report briefly mentions the platform's objective to bolster local economies by promoting homegrown businesses. However, deeper research into effective marketing strategies tailored to local service providers could elucidate avenues for increasing platform adoption and fostering sustainable growth within diverse communities.

## 3.Technological Advancements: While the project leverages state-of-the-art technologies such as React JS, Node.js, and MongoDB, ongoing technological advancements necessitate continuous research to ensure the platform remains at the forefront of innovation. Exploring emerging frameworks, databases, or communication protocols could further enhance scalability, performance, and security.

## 2.3 Problem Formulation

The problem formulation for the Home Based Service Provider platform revolves around the need for a user-friendly and efficient system to connect users with local service providers. Users often struggle to find reliable and trustworthy service providers for tasks such as plumbing, electrical work, car repairs. Additionally, service providers face challenges in reaching potential customers and managing their schedules effectively. This platform aims to bridge this gap by providing a centralized platform where users can easily search, view profiles, and directly communicate with providers.

# CHAPTER 3

# PROPOSED SYSTEM

## 3.1 Proposed System

The proposed Home Based Service Provider website endeavors to redefine the paradigm of how individuals interact with and procure services from local service providers, thereby orchestrating a transformative shift towards enhanced efficiency and unparalleled convenience. By leveraging the power of modern digital technologies, this platform seeks to transcend traditional barriers and complexities inherent in the process of sourcing and engaging local service professionals. Whether users require the expertise of electricians, plumbers, carpenters, or any other household service provider, this platform serves as a beacon of seamless connectivity, offering a streamlined and hassle-free experience from start to finish. Through intuitive interfaces, robust search functionalities, and real-time communication channels, the website empowers users to effortlessly identify, evaluate, and engage with service providers that align with their specific needs and preferences. By fostering direct interaction sans intermediaries, the platform nurtures a sense of transparency and trust, thereby enriching the overall user experience. In essence, the proposed Home Based Service Provider website epitomizes a paradigm shift in the realm of home services, where convenience, efficiency, and customer satisfaction reign supreme.

## 3.2 Unique Features of the System

**1. Enhanced User Experience:**

Through meticulous tracking of user behavior, the system ensures a tailored and personalized experience for each user.

By analyzing past searches and interactions, the system intelligently predicts user preferences, presenting relevant advertisements that align with their interests and needs.

**2. Streamlined Service Discovery:**

The platform's primary focus is to simplify the process of discovering and engaging with service providers through direct communication channels.

By removing intermediaries and enabling direct interaction between users and service providers, the system streamlines the entire service discovery process.

**3. Real-time Communication**:

Integration of real-time communication technologies, such as Socket.IO, facilitates instant and seamless interaction between users and service providers.

This direct communication channel enables swift responses to inquiries, rapid resolution of queries, and efficient booking of services.

**4. Service Provider Ratings:**

The system empowers users by providing access to comprehensive ratings and reviews of service providers.

Users can make informed decisions based on the feedback and experiences shared by previous customers, ensuring quality service delivery.

**5. Efficient Service Delivery:**

By leveraging cutting-edge technologies, the system optimizes service delivery processes to enhance speed, efficiency, and overall user satisfaction. Automation of routine tasks, streamlined communication channels, and personalized service offerings contribute to a seamless and efficient service delivery experience.

**6. Location-Based Search:**

Users can leverage the platform's location-based search functionality to easily locate service providers in their vicinity.

By narrowing down search results based on geographical proximity, users can find suitable service providers nearby, reducing travel time and increasing convenience.

# CHAPTER 4

# REQUIREMENT ANALYSIS AND SYSTEM

**SPECIFICATION**

## Feasibility Study

Conducting a feasibility study for a Home Based Service Provider platform covers three main aspects: Technical, Economical and Operational.

#### Technical Feasibility:

Technical feasibility evaluates the project's ability to be implemented from a technological standpoint. This involves assessing existing infrastructure, compatibility with required technology, and the availability of necessary resources.

The platform's technical feasibility is robust, utilizing modern technologies such as React JS, Tailwind CSS, Node.js, Express.js, MongoDB, and Socket.IO. These technologies are widely used and well-supported, ensuring stability, scalability, and security for the platform. The integration of Socket.IO for real-time communication and personalized ad display demonstrates the platform's advanced technical capabilities.

React JS and Tailwind CSS ensure a sleek and responsive user interface, enhancing user experience and engagement. Node.js and Express.js provide a reliable and scalable backend infrastructure, capable of handling a large number of users and service providers. Node.js and Express.js provide a reliable and scalable backend infrastructure, capable of handling a large number of users and service providers. Continuous integration and deployment practices ensure a stable and up-to-date platform, reducing downtime and improving reliability.

#### Economic Feasibility:

##### Promising Revenue Generation:

Cost The project holds significant promise in terms of economic feasibility, primarily due to its capacity to streamline the connection between users and service providers. By facilitating direct engagement between users and service professionals, the platform creates opportunities for increased revenue generation for both parties. Service providers stand to benefit from a wider customer base and increased service bookings facilitated by the platform, translating into enhanced revenue streams.

##### Flexible Pricing Model:

A key aspect contributing to the economic viability of the project is its implementation of a flexible pricing model. This model allows for negotiation between users and service providers, fostering a competitive marketplace where prices are determined through transparent and fair interactions. The flexibility in pricing ensures that users can find services within their budget constraints, while service providers can adjust their rates based on market demand and competitive factors, thus maximizing revenue potential.

##### Additional Revenue Streams:

##### The personalized advertising system embedded within the platform serves as an additional revenue stream, further bolstering its economic feasibility. Service providers have the option to invest in targeted advertisements to reach potential customers effectively, thereby increasing their visibility and attracting more business opportunities. By offering targeted advertising opportunities, the platform not only benefits from additional revenue but also enhances the overall user experience by presenting users with relevant and timely advertisements aligned with their preferences and needs

##### Long-term Sustainability:

The economic feasibility of the project is also underscored by its potential for long-term sustainability and growth. As the platform gains traction and expands its user base, it is poised to attract more service providers and advertisers, further enhancing revenue streams. Moreover, ongoing innovation and adaptation to evolving market trends ensure that the platform remains competitive and relevant in the long run, thus securing its economic viability over time.

#### Operational Feasibility

**Simplified Service Procurement Process:** The operational feasibility of the platform is exceptionally high, primarily due to its ability to simplify the process of finding and hiring home service providers. By offering a centralized platform for accessing a diverse range of service professionals, the platform eliminates the need for users to navigate multiple channels or sources, thereby streamlining the entire service procurement process.

**Efficient Search and Discovery Mechanisms:** Users benefit from intuitive search functionalities that allow them to easily locate service providers based on their specific location and requirements. By leveraging location-based search features, users can swiftly identify nearby professionals, saving considerable time and effort in the process. Additionally, the platform's advanced filtering options enable users to refine their search results based on factors such as service type, ratings, and availability, further enhancing the efficiency of the service discovery process.

**Seamless Communication Channels:** The integrated chat functionality within the platform plays a pivotal role in enhancing operational efficiency and customer satisfaction. By facilitating seamless communication between users and service providers, the platform ensures that queries are addressed promptly and accurately, reducing response times and enhancing overall user experience. Through real-time messaging capabilities, users can convey their requirements, discuss service details, and schedule appointments with ease, thereby streamlining the operational workflow and fostering stronger relationships between users and providers.

**Accessibility and User-Friendliness:** Another aspect contributing to the operational feasibility of the platform is its accessibility and user-friendliness. The platform is designed with a user-centric approach, featuring an intuitive interface and straightforward navigation pathways that cater to users of all technical proficiencies. This accessibility ensures that users can easily navigate the platform, access relevant information, and engage with service providers without encountering any significant barriers, there by maximizing the operational efficiency and the user satisfaction.

## 4.2 Software Requirement Specification Document

#### 4.2.1 Data Requirement

##### User Data

The platform necessitates the collection of comprehensive user data to facilitate seamless interactions and personalized experiences. User profiles including name, email, contact information, and location. Authentication data for secure login and account management. User ratings and reviews for service providers.

**Service Provider Data**

Similarly, the platform requires comprehensive data from service providers to present users with accurate and relevant information. Provider profiles with details such as name, contact information, services offered, experience, and certifications. Pricing information and service packages offered by providers provide transparency and enable users to make informed decisions based on their budgetary constraints and service preferences. Provider availability schedule to match with user requests.

**Service Request Data**

Details of service requests including type of service, preferred time, location, and any additional instructions. Status of service requests (e.g., pending, in-progress, completed).

**Geolocation Data**

Geolocation data serves as a crucial component for matching users with nearby service providers and defining service areas and boundaries. By leveraging user and provider location data, the platform can optimize service recommendations and enhance the efficiency of service matching algorithms.

Geo-fencing data further refines service boundaries, ensuring that users are connected with providers operating within specified geographic regions, thereby enhancing service relevance and accuracy.

#### 4.2.2 Functional Requirement

**User Registration and Authentication:** Users should be able to register with the platform using email. Implement secure authentication mechanisms such as password hashing. Allow users to reset passwords if forgotten.

**Service Provider Registration:** Service providers should be able to create profiles with details like services offered, pricing, availability, and contact information. Implement secure authentication mechanisms such as password hashing.

**Search and Browse Services:** Users should be able to search for services based on categories, keywords, and location. View detailed service provider profiles including reviews, ratings, and service offerings**.**

**Real-time Communication:** Implement a chat or messaging system for users and service providers to communicate in real-time**.**

**Mobile Responsiveness:** Ensure the platform is responsive and accessible on various devices including smartphones and tablets. Optimize the user interface for mobile screens to provide a seamless user experience.

#### 4.2.3 Performance Requirement

**Response Time:** Ensuring fast response times is crucial for providing a seamless user experience on the platform. Pages should load quickly, and search results should be displayed promptly to minimize user frustration and enhance engagement. Aim for an average response time of under 5 seconds for most operations to meet user expectations regarding speed and efficiency.

**Concurrent Users:** The system should be able to handle concurrent users without significant performance degradation. Aim for a minimum of 500 concurrent users without noticeable slowdowns.

**Chat System Responsiveness:** Real-time communication through chat functionality is a key feature of the platform, facilitating direct interaction between users and service providers. It's imperative to minimize latency in sending and receiving messages to maintain smooth communication flow. Ensure that messages are delivered instantly to enhance user engagement and satisfaction, fostering efficient communication between all parties involved.

**Error Handling:** Error messages and notifications should be clear and informative to help users troubleshoot issues. The system should gracefully handle unexpected errors without crashing or disrupting user experience.

#### 4.2.4 Maintainability Requirement

Maintainability requirements focus on ensuring that the system can be easily maintained and updated over time. This includes documentation, modular design, and support for future updates and enhancements. We need to design the system in a way that facilitates ongoing maintenance and allows for seamless integration of new features and improvements.

**Modularity:** Implement a modular architecture that allows easy updates and modifications to different components of the platform. Use component-based design for the frontend, ensuring that each feature or section is encapsulated and reusable.

**Code Documentation:** Ensure thorough documentation of the codebase, including backend APIs, frontend components, and database schema. Provide clear and concise comments within the code to explain the logic and functionality.

**Version Control:** Utilizing Git for version control is essential for maintaining code integrity and facilitating collaboration within development teams. Git allows tracking changes, managing branches, and merging contributions effectively, enabling seamless collaboration on codebases. By regularly committing code changes and following a branching strategy for development and production releases, teams can maintain a structured and organized workflow, ensuring code stability and reliability throughout the development lifecycle.

**Dependency Management:** Effective dependency management is crucial for ensuring the stability, security, and performance of software projects. By managing dependencies carefully, teams can regularly update libraries and packages to incorporate security patches and performance improvements, mitigating the risk of vulnerabilities and optimizing application performance. Utilizing package managers like npm enables teams to handle dependencies and versions effectively, streamlining the process of integrating and updating external dependencies while maintaining consistency and reliability across the codebase.

**Coding Standards:** Adhere to a consistent coding style guide to maintain readability and consistency across the codebase. Conduct code reviews to ensure adherence to coding standards and best practices.

#### 4.2.5 Security Requirement:

**Secure Authentication:** Implement secure authentication mechanisms to ensure that user accounts are protected against unauthorized access. Utilize robust encryption protocols and password hashing techniques to safeguard user credentials during transmission and storage.

**Data Encryption:** Encrypt sensitive data both at rest and in transit to prevent unauthorized access and data breaches. Utilize strong encryption algorithms and secure communication protocols to protect data integrity and confidentiality**.**

**Session Management:** Implement secure session management techniques to prevent session hijacking and unauthorized access to user sessions. Use mechanisms such as session tokens, session expiration, and secure cookies to ensure that sessions are securely managed and authenticated.

**Secure Communication:** Use secure communication protocols such as HTTPS to encrypt data transmitted between clients and servers**.**

**Security Auditing and Logging:** Implement comprehensive logging and auditing mechanisms to track user activities, system events, and security incidents. Monitor and log security-related events to detect and respond to suspicious activities and security breaches effectively.

**Secure Configuration:** Ensure that the platform is securely configured according to industry best practices and security guidelines. Regularly update and patch system components, libraries, and dependencies to address known security vulnerabilities and minimize the risk of exploitation.

**Security Testing:** Conduct regular security assessments, penetration testing, and vulnerability scans to identify and remediate security weaknesses and vulnerabilities. Perform code reviews and security audits to ensure that security requirements are properly implemented and enforced.

**Incident Response:** Establish incident response procedures and protocols to effectively respond to security incidents and breaches. Define roles and responsibilities, establish communication channels, and develop incident response plans to mitigate the impact of security breaches and minimize downtime.

## 

## 4.3 SDLC Model Used

The Waterfall model, a traditional software development methodology, stands as the chosen framework for the development of the Home Based Service Provider platform. This decision aligns with the project's requirements and characteristics, ensuring a structured and sequential approach to software development.

**Requirements Analysis:** The initial phase of the Waterfall model focuses on gathering and analyzing requirements from stakeholders. In this stage, detailed requirements for the platform's features, functionalities, and user interactions will be documented. Stakeholder consultations and user feedback will be crucial in defining the scope and objectives of the project.

**System Design:** Once requirements are finalized, the system architecture will be designed using the MERN stack. Design phase includes defining the database schema in MongoDB, creating RESTful APIs with Express.js, designing the user interface with React.js, and planning the backend logic in Node.js.

**Implementation**: Development will begin based on the detailed design specifications. MongoDB will be used to store user and service provider data, Express.js will handle server-side logic and API development, React.js will create the frontend interface, and Node.js will serve as the runtime environment. Each module will be implemented sequentially, following the predefined MERN design.

**Testing**: Comprehensive testing will be conducted after the development of each module to ensure functionality, performance, and security. Testing includes unit testing for individual components, integration testing for combined modules, and system testing to evaluate the entire platform.

Deployment: Once all modules are developed, integrated, and tested, the platform will be deployed on servers. Deployment involves setting up the MERN stack on servers, configuring databases, and ensuring proper connectivity.

**Maintenance**: Post-deployment, ongoing maintenance and support will be provided to address any issues or updates. Regular monitoring, bug fixes, admin control and updates to the MERN stack components that will be getting performed to ensure that the platform will remains stable and secure.

## 4.4 System design

The System Design module encompasses the architectural framework and technical infrastructure of the Home Based Service Provider platform. It involves the structuring of components such as databases, servers, APIs, and communication protocols to ensure scalability, reliability, and security. The design incorporates principles of modularity, abstraction, and encapsulation to facilitate ease of maintenance and future expansion. Additionally, it addresses aspects like load balancing, caching, and fault tolerance to optimize performance and availability. Overall, the System Design module lays the foundation for a robust and efficient platform capable of meeting user demands and accommodating future growth. In system design, careful consideration is given to scalability, ensuring the platform can accommodate growth without sacrificing performance. Security measures are implemented at every level to safeguard user data and maintain user trust. Additionally, fault tolerance mechanisms are integrated to minimize downtime and ensure uninterrupted service availability. Regular performance monitoring and optimization efforts are conducted to maintain system efficiency and responsiveness.

## 4.4.1 Data Flow Diagram

## The Data Flow Diagram (DFD) illustrates the flow of data within the Home Based Service Provider platform. It showcases how data moves through different processes and entities, depicting interactions between users, service providers, and the system. The diagram typically includes entities like users, service providers, databases, and external systems, with arrows indicating the direction of data flow. Each process is labeled with its function, clarifying how data is manipulated or transformed at each stage.

## DFD Level 0

## In the level 0 Data Flow Diagram (DFD) for the Home Based Service Provider platform, it provides an overview of the system's main processes and their interactions. At this level, it typically includes entities such as users, service providers, and the system itself. The primary processes are depicted as bubbles, representing high-level functions like user registration, service booking, and provider management. Arrows indicate the flow of data between these processes and entities.

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system, illustrating how data is processed and transmitted between different components. The DFD is composed of various levels, with Level 0 being the most fundamental and high-level view. A Level 0 DFD, also known as a Context Diagram, provides an overview of the entire system, encapsulating its primary functions and interactions with external entities. Here is an in-depth explanation of DFD Level 0:

**Components of a DFD Level 0**

Processes:

Represented by a single process node (usually a circle or rounded rectangle), the process at Level 0 encapsulates the entire system's functionality. This node is labeled with the name of the system, indicating that it encompasses all the data processing activities that occur within the system.

**External Entities:**

External entities are sources or destinations of data that interact with the system. These are represented by squares or rectangles and are typically labeled with nouns such as "Customer," "Supplier," "User," or "Administrator." External entities provide input data to the system and receive output data from the system, but they are not involved in the internal processing.

**Data Flows:**

Data flows are depicted by arrows that show the direction of data movement between processes, external entities, and data stores. These arrows are labeled with the names of the data being transmitted, providing a clear understanding of what information is being exchanged.

**Data Stores:**

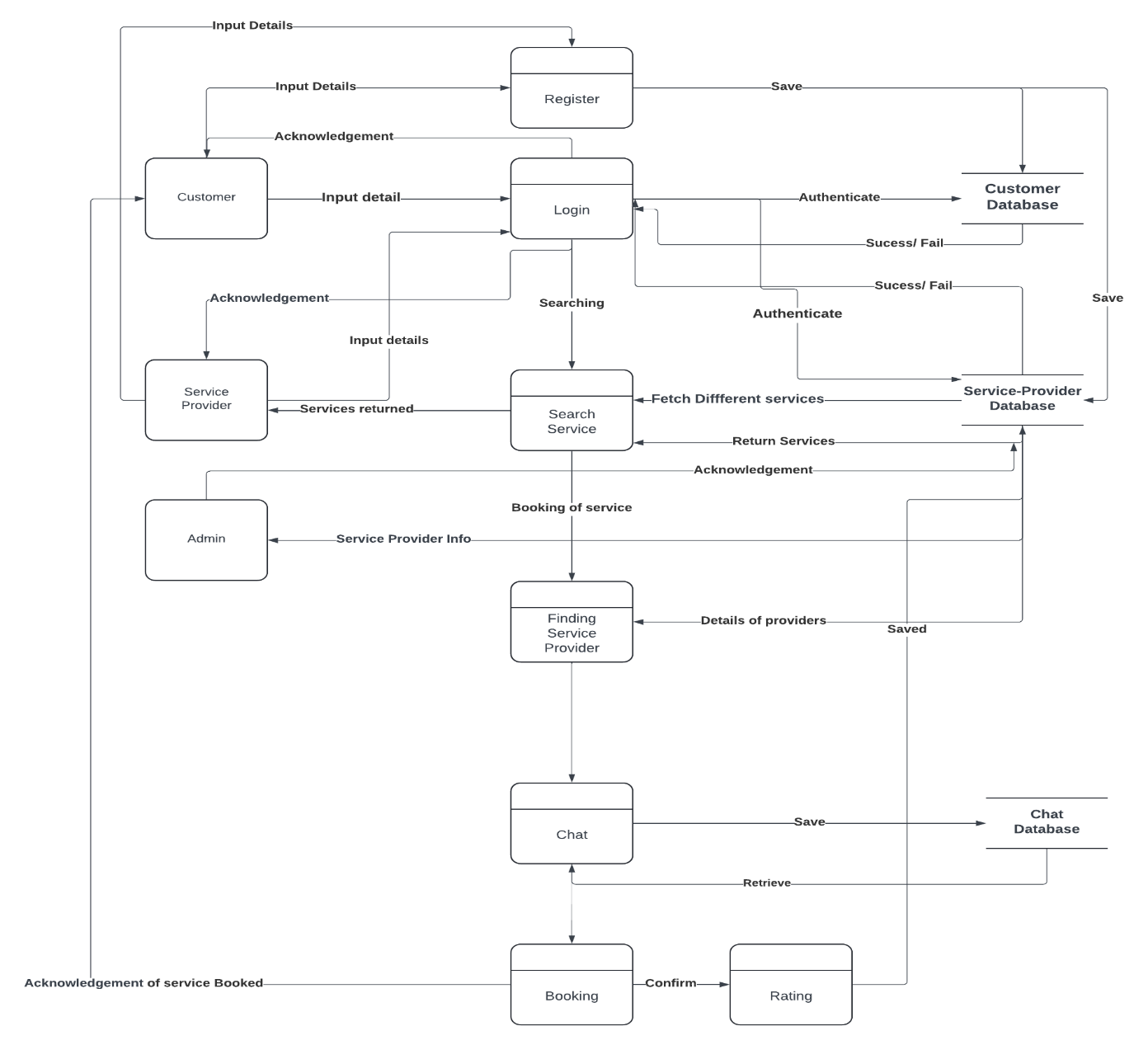
Although not always included in Level 0 DFDs, data stores represent places where data is held within the system. They are depicted as open-ended rectangles (or parallel lines) and are labeled with nouns describing the data they contain, such as "Customer Data" or "Order Information."

## 

**Figure 4.1 DFD Level 0**

**DFD level 1**

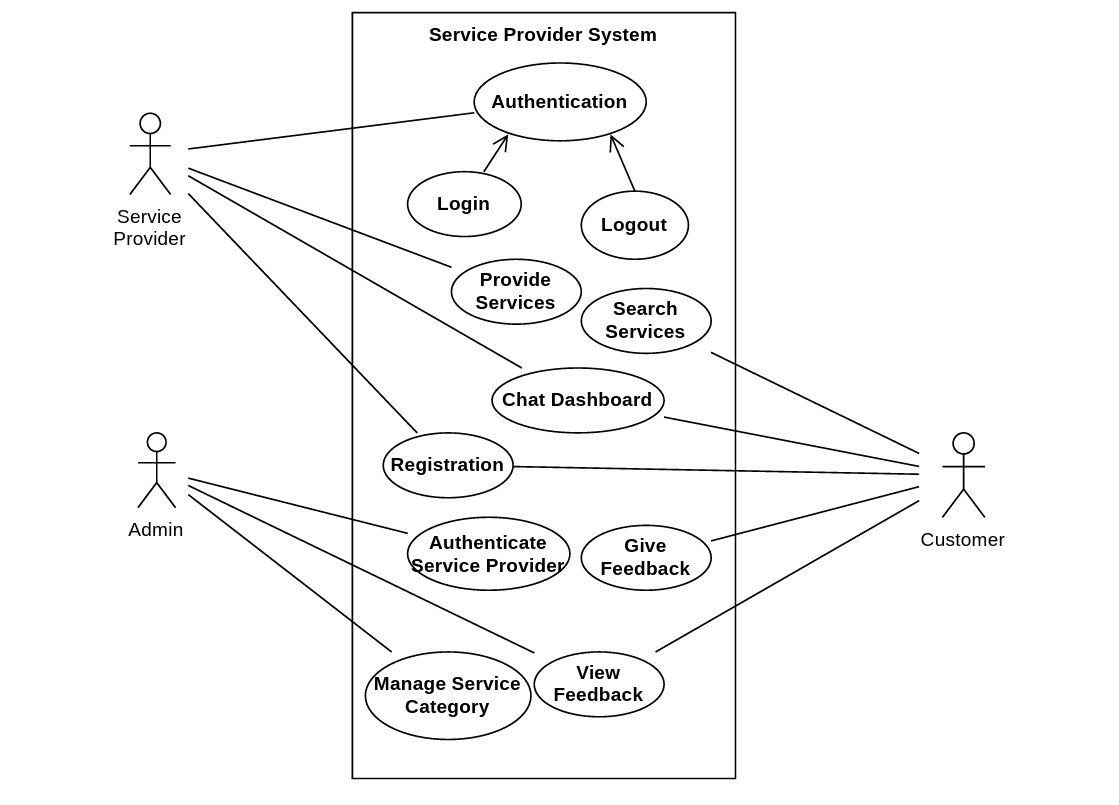
1-Level provides a more detailed view of the system by breaking down the major processes identified in the level 0 Data Flow Diagram (DFD) into sub-processes. Each sub-process is depicted as a separate process on the level 1 Data Flow Diagram (DFD). In 1-level Data Flow Diagram (DFD), the context diagram is decomposed into multiple bubbles. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level Data Flow Diagram (DFD) into subprocesses.



**Figure 4.2 DFD Level**

### **4.4.2 Use Case Diagram**

### The Use Case Diagram for the Home Based Service Provider platform illustrates various user interactions and system functionalities. It identifies actors such as users and service providers and outlines their roles and actions within the system. Use cases represent specific scenarios or tasks users can perform, guiding system development and ensuring alignment with user requirements.

****

**Figure 4.3 Use Case Diagram**

### **4.5 Database design**

#### In the system design of the Home Service Provider website, MongoDB plays a critical role as the database management system, offering a flexible and scalable solution to handle vast amounts of data efficiently. MongoDB's document-oriented model is particularly well-suited for this application, as it allows for the storage of complex and varied data structures in JSON-like documents. Within MongoDB, three main collections are utilized: one for storing service provider data, another for customer data, and a third for managing chat conversations.

#### The Service Provider collection stores essential information about service providers registered on the platform. Each document in this collection represents a service provider and includes details such as their profile information, services offered, availability, contact information, ratings, and reviews. Leveraging MongoDB's schema-less architecture, this collection can accommodate a diverse range of data types and structures, allowing for flexibility in how service provider information is stored and accessed.

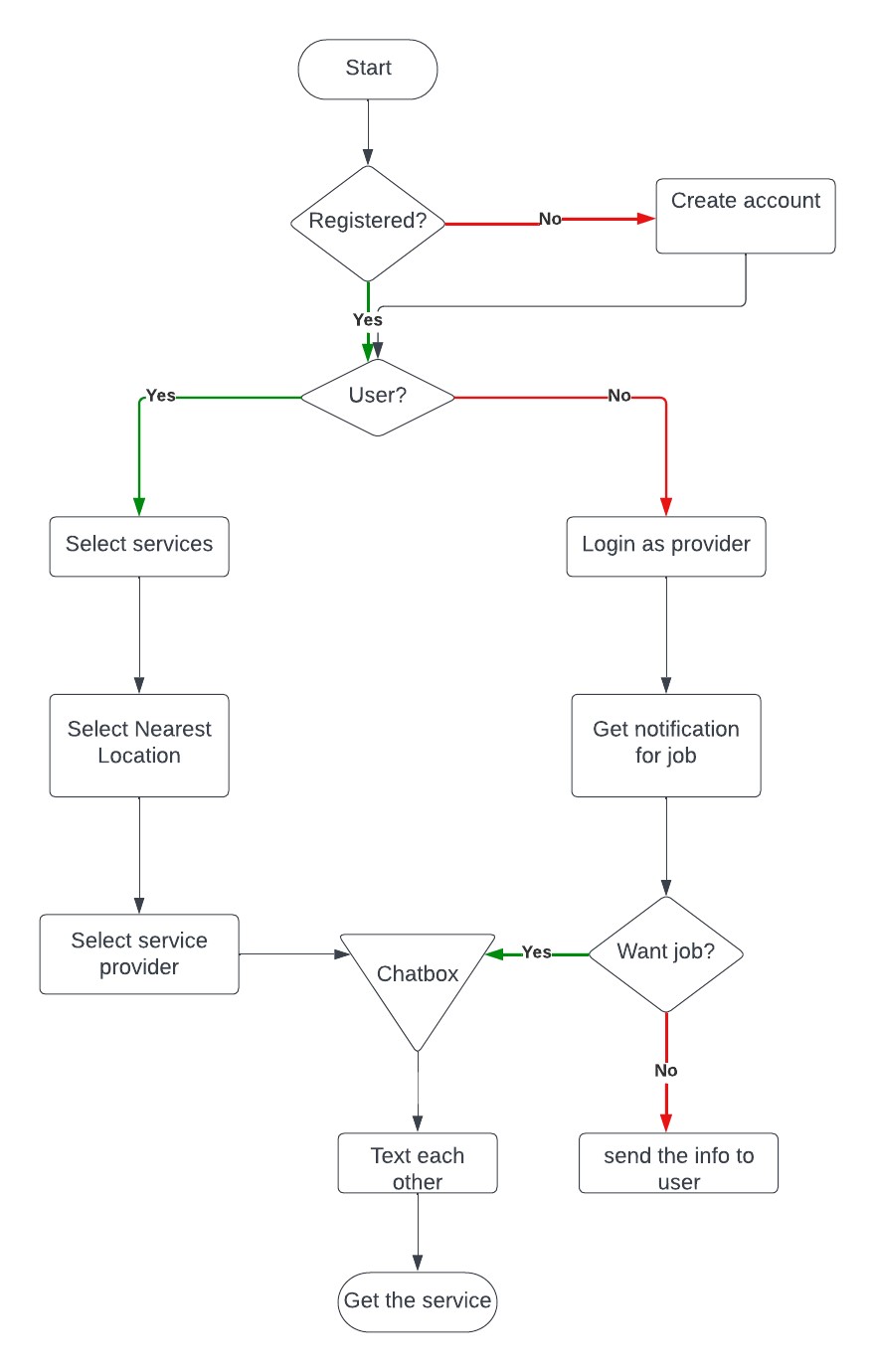
#### The Customer collection stores detailed information about registered customers on the platform. Each document represents a customer, encompassing their profile details, booking history, preferences, and interactions with service providers. MongoDB's capability to manage nested data structures is particularly advantageous here, as it allows for the storage of intricate customer profiles with multiple attributes and embedded objects.

#### The third collection is dedicated to managing real-time chat conversations between customers and service providers. The Chat collection stores chat messages exchanged between users, along with metadata such as timestamps, sender IDs, and conversation IDs. By utilizing MongoDB for chat data storage, the platform ensures seamless communication between users, with messages being reliably stored and retrieved. MongoDB's indexing and querying capabilities enable efficient retrieval of chat history, allowing users to access past conversations quickly and easily.

#### Regarding scalability, MongoDB's distributed architecture and sharding support facilitate horizontal scaling, enabling the platform to handle increasing data volumes and traffic as the user base expands. Sharding partitions data across multiple servers, distributing the workload and enhancing overall performance and scalability. Additionally, MongoDB's replication features ensure high availability and fault tolerance by maintaining multiple data copies across different nodes, minimizing the risk of data loss and downtime. This robust infrastructure supports the platform's growth and ensures a reliable user experience. The Service Provider collection holds vital information about service providers registered on the platform. Each document in this collection represents a service provider and encompasses details such as their profile information, services offered, availability, contact information, ratings, and reviews. MongoDB is crucial as the database management system, providing a flexible and scalable solution to efficiently handle large volumes of data. MongoDB's document-oriented model is ideal for this application, enabling the storage of complex and diverse data structures in JSON-like documents. The platform utilizes three primary collections within MongoDB: one for storing service provider data, another for customer data, and a third for managing chat conversations.

**Workflow diagram**





**Figure 4.4 Workflow Diagram**

# CHAPTER 5

**IMPLEMENTATION**

##### 5.1 Introduction Tools and Technologies Used

* + 1. **Languages Used:**

**Frontend**

**React JS:** A popular JavaScript library for building interactive user interfaces. React provides a component-based architecture, enabling the creation of reusable UI components.

**Tailwind CSS:** A utility-first CSS framework that allows for rapid styling and customization. Tailwind's utility classes streamline the design process and ensure consistent styling across the platform.

**Backend**

**Node.js:** A runtime environment for executing JavaScript code server-side. Node.js provides a scalable and efficient backend solution, allowing for non-blocking, event-driven I/O operations.

**Express.js:** A lightweight and flexible Node.js framework for building web applications and APIs. Express simplifies routing, middleware handling, and request/response processing.

**Database**

**MongoDB:** A NoSQL database that offers flexibility and scalability. MongoDB's document-based structure is well-suited for storing user profiles, service provider information, and service requests. It allows for efficient retrieval and storage of JSON-like documents**.**

**Real-Time Communication**

**Socket.IO:** A JavaScript library for enabling real-time, bidirectional communication between clients and servers. Socket.IO facilitates instant messaging and notifications, enhancing user engagement and interaction.

##### Tools:

**Github**: Version Control: GitHub serves as a robust version control system, enabling developers to track changes, collaborate on code, and manage revisions effectively.

Code Repository: The project's codebase is securely hosted on GitHub, providing a centralized repository for storing, sharing, and managing all code files, ensuring accessibility and collaboration among team members.

**Git**: Version control system for managing code changes and collaboration among developers.

**Postman**: API testing tool for validating backend APIs and endpoints.system for tracking changes to the project codebase and collaborating with team members.

**5.1.3 Implementation of Modules**

**1. Implementation of Real-Time Communication**

**Access Context and State:** Access context state and dispatch functions using useContext(userContext).

**Implement Send Message Function:** Write the sendMessage() function to handle sending messages. Ensure it sends HTTP requests to appropriate endpoints and emits a 'send' event through the socket.

**Implement Receive Message Function:** Set up an effect hook to listen for incoming messages. Define a function to handle received messages and register it with the socket.

**Implement Update User Data Function:** Write functions (setProData() and setUserData()) to update provider and user data. These functions should send HTTP requests and update local state with received data.

**2. Finding the Nearest Service Provider**

**Manage Location State:** Use useState to manage the location state. Initialize it with an object containing loaded and coordinates properties.

**Define Success and Failure Handlers:** Implement onSuccess and onFailure functions to handle successful and failed geolocation requests, respectively.

**Handle Button Click:** Implement handleButtonClick function to fetch reverse geocode data using the provided API. Dispatch the location data to the context upon successful retrieval.

**Initialize Geolocation:** In the useEffect hook with an empty dependency array ([]), check if geolocation is supported. If supported, for changes in the user's position using navigator.geolocation.watchPosition.

**Trigger Reverse Geocoding:** Use another useEffect hook with the location.loaded dependency to trigger the reverse geocoding process when the location is loaded.

**3. Search Services**

**Import Data:** Import the data you want to search from. In this example, it's imported from a JSON file named MOCK\_DATA.json.

**Define Component Function:** Define the SearchBar component function. Manage State: Use the useState hook to manage the state of the search input value.

**Define Change and Search Handlers:** Implement onChange to update the search input value and onSearch to handle search actions. This function can be extended to fetch search results from an API.

**Render Search Input:** Render the search input field along with a search button. Use the Link component from react-router-dom to navigate to the search results page when the button is clicked.

**Display Search Suggestions:** Render search suggestions based on the input value. Filter the data based on the search term and display matching results.

**4. Provide Rating to the Service Provider**

**Star Icons:** Renders a row of star icons representing the rating scale. FaStar icons are rendered using the Font Awesome library. The number of stars is determined by the array length (5 in this case). Each star is conditionally styled based on whether its index is less than the current rating (rated).

**Rating State:** Updates the rating state (rated) upon clicking a star icon. Clicking a star updates the state to the index of the clicked star plus one, effectively setting the rating value.

**Displayed Rating:** Shows the current rating value with one decimal point. The current rating (rated) is displayed alongside the star icons.

**GitHub Link --** https://github.com/AavegTomar/ServiceProvider/tree/master

# CHAPTER 6

**TESTING AND MAINTAINANCE**

##### Testing techniques and Test Cases Used

##### 6.1.1 Manual testing – Decision Table Testing

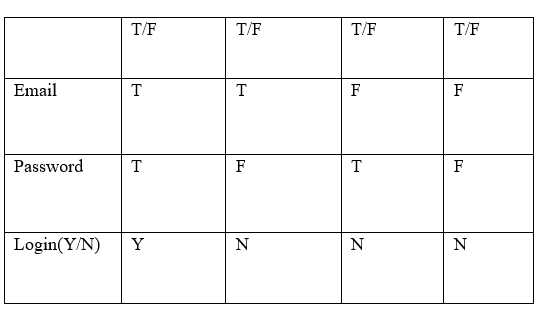
##### Decision table testing, a black-box testing technique, involves creating a table that represents possible inputs, conditions, and corresponding actions or outputs. Each combination of inputs and conditions leads to a specific action or output, allowing testers to verify that the system behaves as expected under different scenarios.

**Test Case 1 : Professional Registration**

##### Table 5.1 Professional Registration

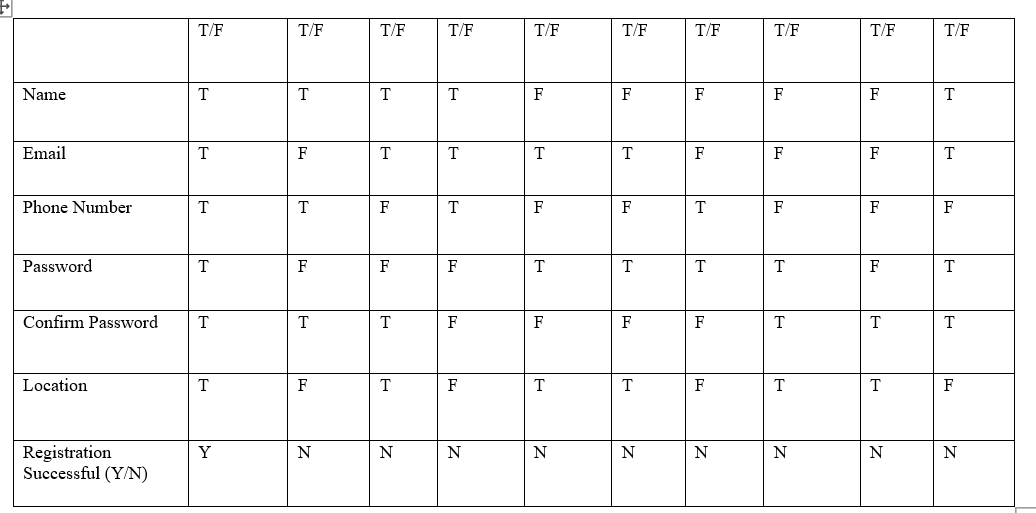
**Test Case 2 : Professional Login**

**Table 5.2 Professional Login**

****

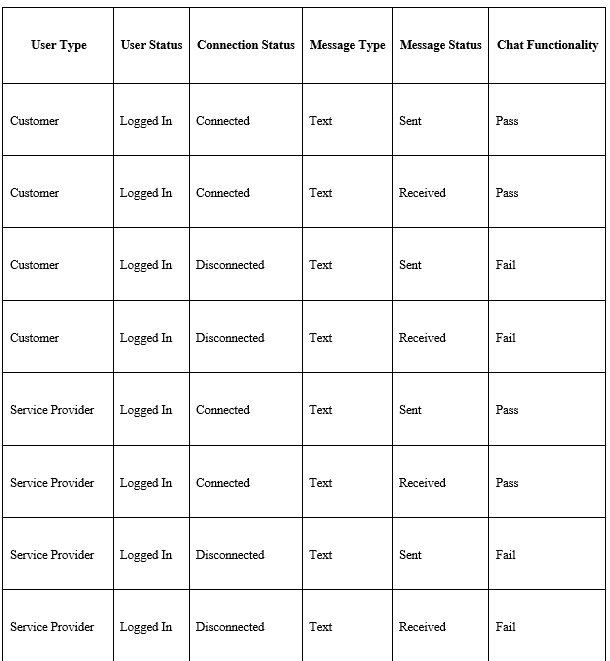
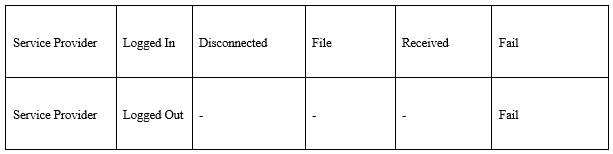
**Test Case 3 : Customer Registration**

**Table 5.3 Customer Registration**



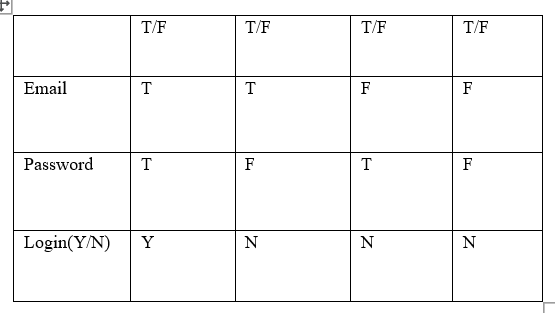
**Test Case 4 : Chat Dashboard**

**Table 5.4 Chat Dashboard**

****

**Test Case 4 : Customer Login**

**Table 5.5 Customer Login**

****

**6.1.2 Automation testing – Selinium**

|  |
| --- |
| **Test Case 1 : Customer Login**    **Figure 6.1 Customer Login**  **Test Case 2. Provider Registration**    **Figure 6.2 Provider Registration**  **Test Case 3. Provider Login**    **Figure 6.3 Provider Login**    **Figure 6.4 Provider Dashboard**  **Test Case 4. Chat Dashboard**    **Figure 6.5 Chat Dashboard** |

# CHAPTER 7

# RESULTS AND DISCUSSIONS

# 7.1 Description of Modules with Snapshots

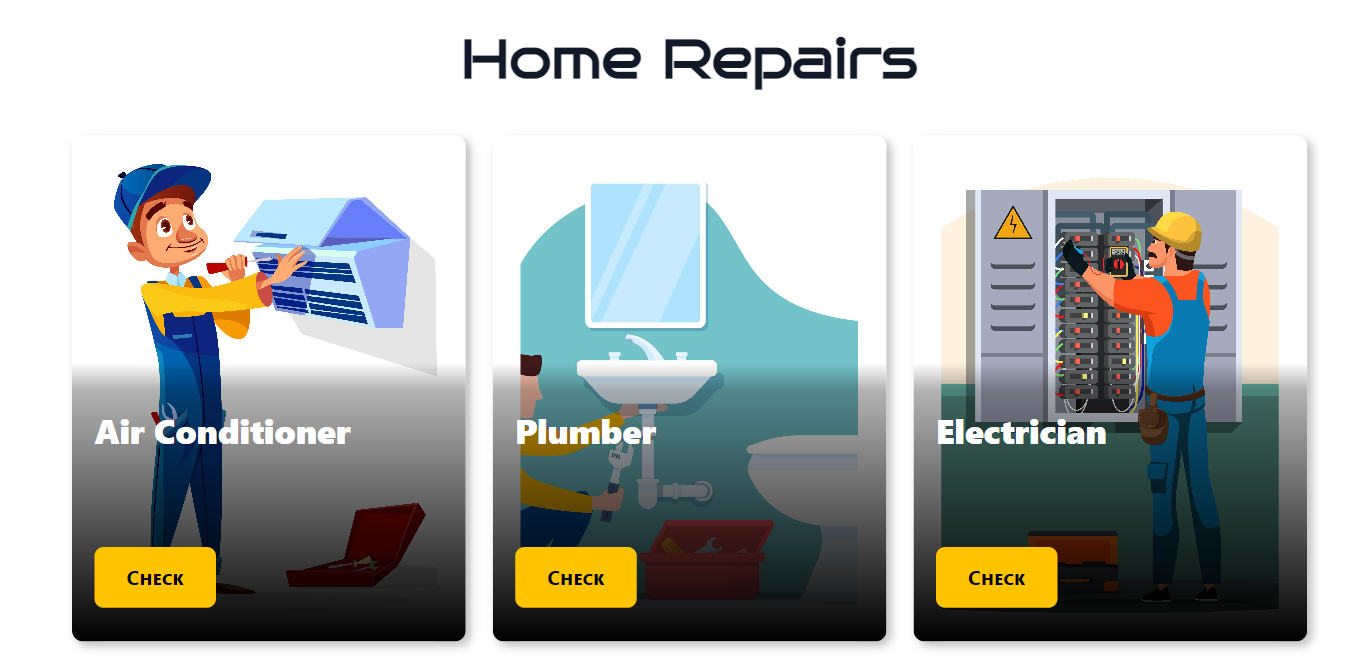
# The Home Based Service Provider platform comprises several essential modules designed to streamline user interaction and service delivery. Firstly, the User Authentication module ensures secure registration, login, and authentication processes for users. Service Provider Profile management allows detailed provider profiles, including experience and ratings, enhancing user decision-making. Users can efficiently search and filter service providers based on location and service type using the Search and Filter module. Real-time Chat functionality fosters seamless communication between users and providers, facilitating instant interaction. Service Booking enables direct service booking, managing scheduling and notifications. Integration with a Payment Gateway ensures secure payment processing for transactions. The Review and Rating module enables users to provide feedback, enhancing trust and accountability. Lastly, the Admin Dashboard provides administrators with effective tools to manage users, providers, and platform content. Together, these modules ensure a smooth and efficient user experience, empowering users and providers alike.

## 7.1.1 User Interface Representation

## The User Interface (UI) Representation module serves as the visual gateway for users interacting with the Home Based Service Provider platform. It encompasses the design elements and layout that users encounter while navigating the platform. The UI design prioritizes sleek aesthetics, intuitive navigation, and responsiveness across various devices and screen sizes. Through the adept utilization of modern design principles, such as clear typography, consistent branding, and intuitive user flows, the UI Representation module ensures a visually appealing and user-friendly experience. It incorporates features like interactive elements, visually engaging graphics, and well-organized layouts to enhance usability and engagement. Moreover, the UI Representation module aligns closely with the platform's branding and ethos, conveying a sense of trustworthiness, professionalism, and reliability to users. Overall, this module plays a pivotal role in shaping users' perceptions and experiences, ultimately contributing to the platform's success and user satisfaction.

#### 7.1.1.1 User Home Page

The User Home Page module is the central hub for users on the Home Based Service Provider platform, offering an intuitive and personalized experience. It features a comprehensive dashboard displaying relevant content tailored to each user's needs and preferences. This includes recommended service providers based on user preferences and past bookings, an overview of recent bookings with details such as service provider names and statuses, and real-time notifications related to bookings, messages, promotional offers, and platform announcements. The module also provides easy navigation to various sections, including a powerful search tool for finding service providers based on criteria like location and ratings, a detailed booking history log, and account settings for updating profile information and managing payment methods. Additional features include the ability to mark and manage favorite service providers, an integrated review and rating system, and displays of current promotions and offers. Designed for seamless navigation with a responsive design for desktops, tablets, and smartphones.

****

**Figure 7.1 User Home Page**

# 7.1.1.2 Service Providers

|  |
| --- |
|  |
| **Figure 7.2 Service Providers** |

#### 7.1.1.3 User Sign in Page

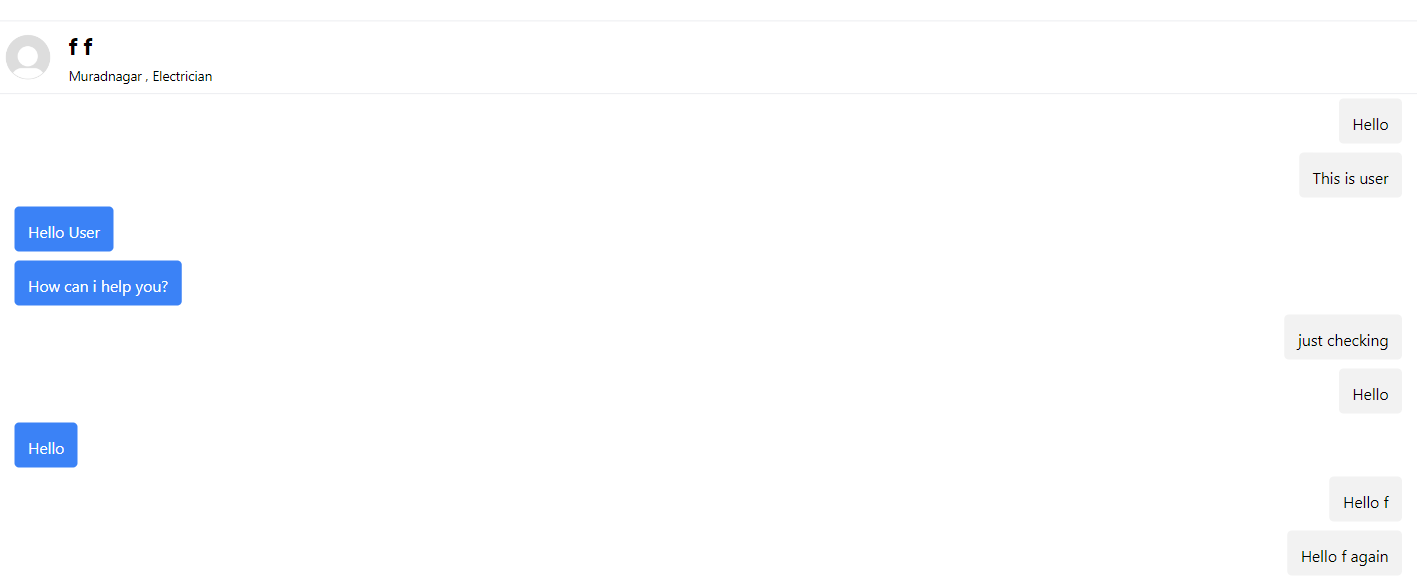
# User can do sign in if there has already have the account otherwise it login as guest login or signup and you can create account and then login.

# 

|  |
| --- |
|  |
| **Figure 7.3 User Sign In Page** |

**7.1.1.4 Chat Console**

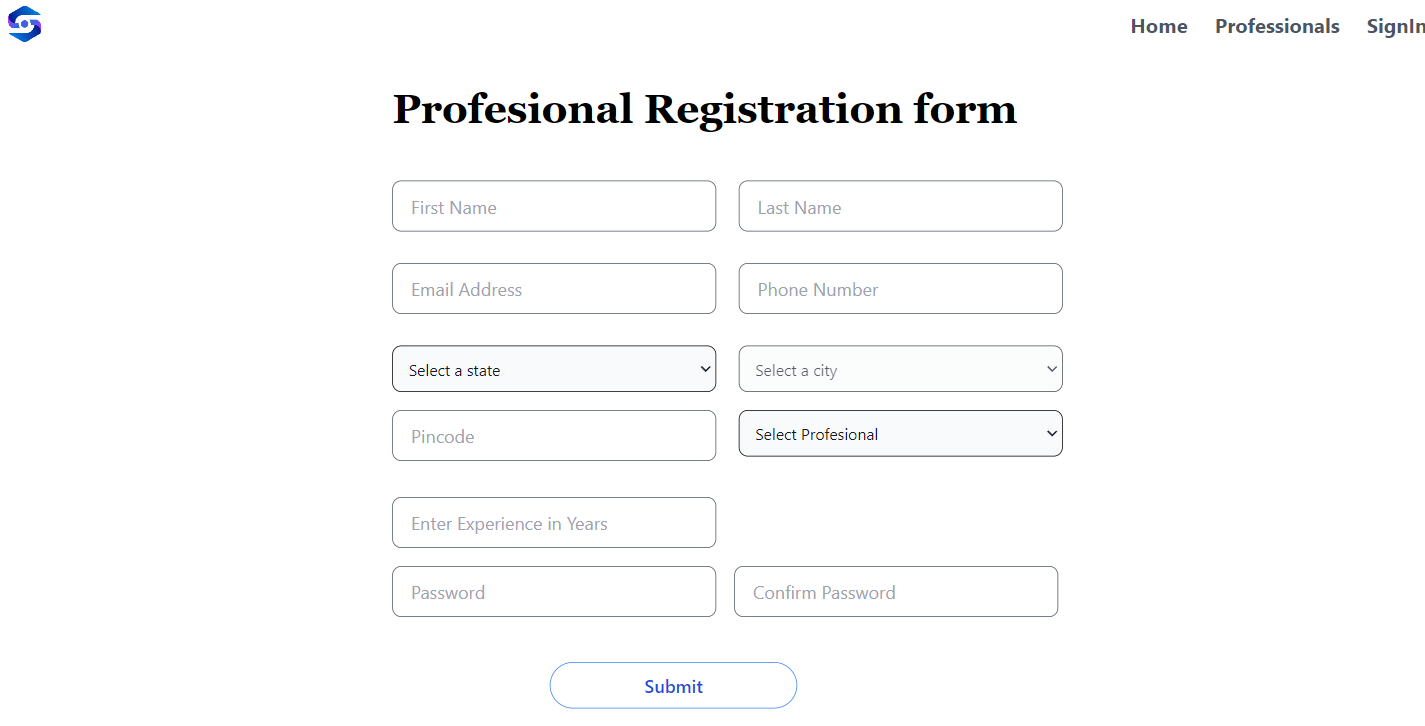
User and provider can chat and do negotiation on the price and also can fix the schedule, price according to its relevance time**.** The use of socket IO make the chatting feature faster than traditional way of extracting the chats from database and then showing the chat to the users as it take time to again and again interact with database which is resolved through the use of socket io.

****

**Figure 7.4 Chat Console**

**7.1.1.5 Service Provider Registration**

Service Providercan do registration if they do not have already existing account otherwise they can directly login as guest login or signup and you can create account and then login.

****

**Figure 7.5 Service Provider Registration**

**7.2 Key Findings of Modules**

**User Sign Up and Sign In:** Users can easily register for an account on the platform, providing necessary details such as username, email, and password. The sign-in process allows users to securely access their accounts, providing a seamless entry point to the platform's services. Key findings include the usability and effectiveness of the sign-up and sign-in flows, ensuring a smooth onboarding experience for users.

**Service Provider Registration:** Service providers can register on the platform by providing necessary information such as business details, services offered, and contact information. The registration process may include verification steps to ensure the authenticity of service providers and the quality of services offered. Key findings include the effectiveness of the registration process in onboarding service providers and the ease of providing required information.

**Service Provider Login:** Authentication mechanisms ensure secure access to service provider accounts, protecting sensitive business information. Key findings focus on the usability and security of the login process for service providers, facilitating efficient management of their service offerings.

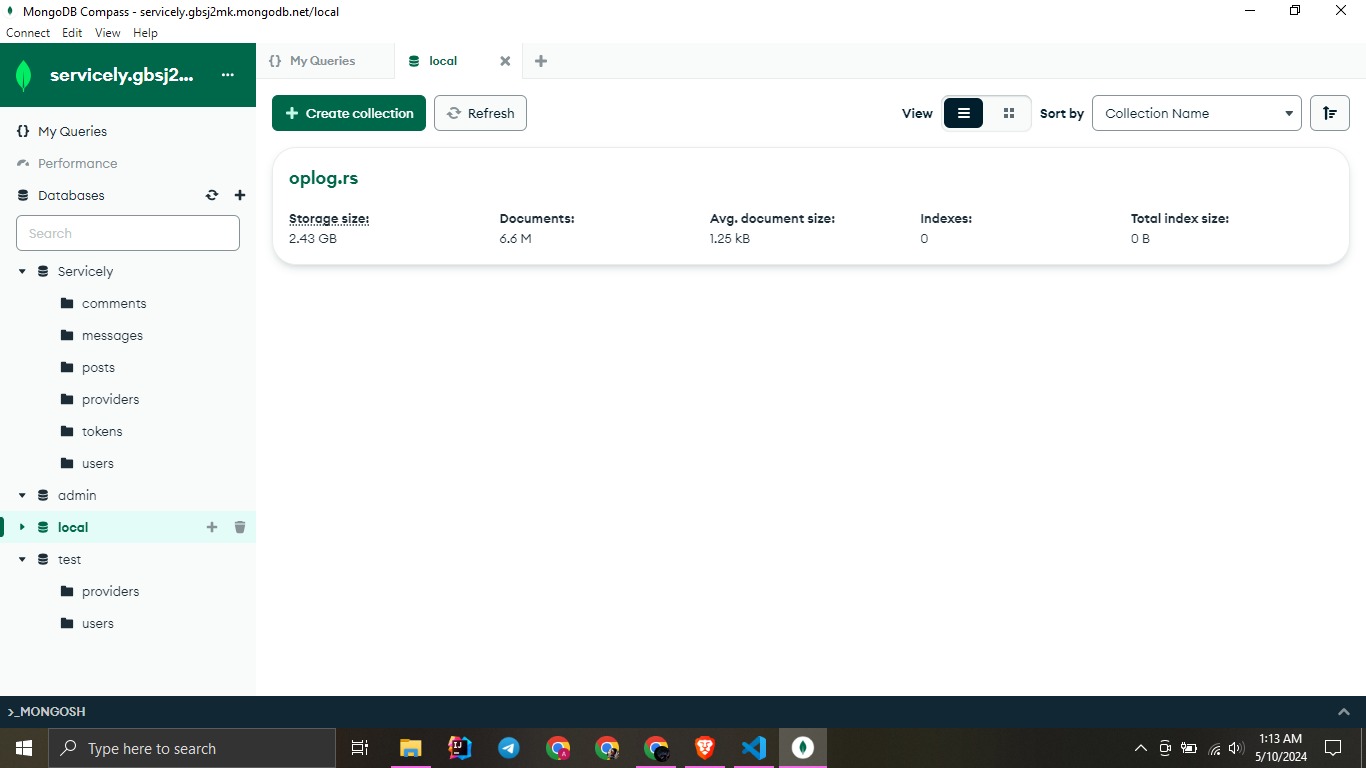
**Chat Interface:** The chat interface enables direct communication between users and service providers, facilitating real-time interaction for service inquiries, scheduling, and updates. Features such as message delivery notifications and typing indicators enhance the user experience and promote seamless communication. Key findings include the responsiveness and reliability of the chat interface, as well as user satisfaction with the quality of communication facilitated by the platform. These key findings provide insights into the effectiveness and usability of each module within your Home Based Service Provider website, informing future iterations and improvements to enhance overall user satisfaction and platform functionality.

**Search Services:** The "Search Services" functionality is a critical feature in any service-oriented application, enabling users to find and select the services they need quickly and efficiently. Allows users to easily navigate through a variety of services, saving time and effort. Helps in tailoring the search results based on user preferences and previous interactions.

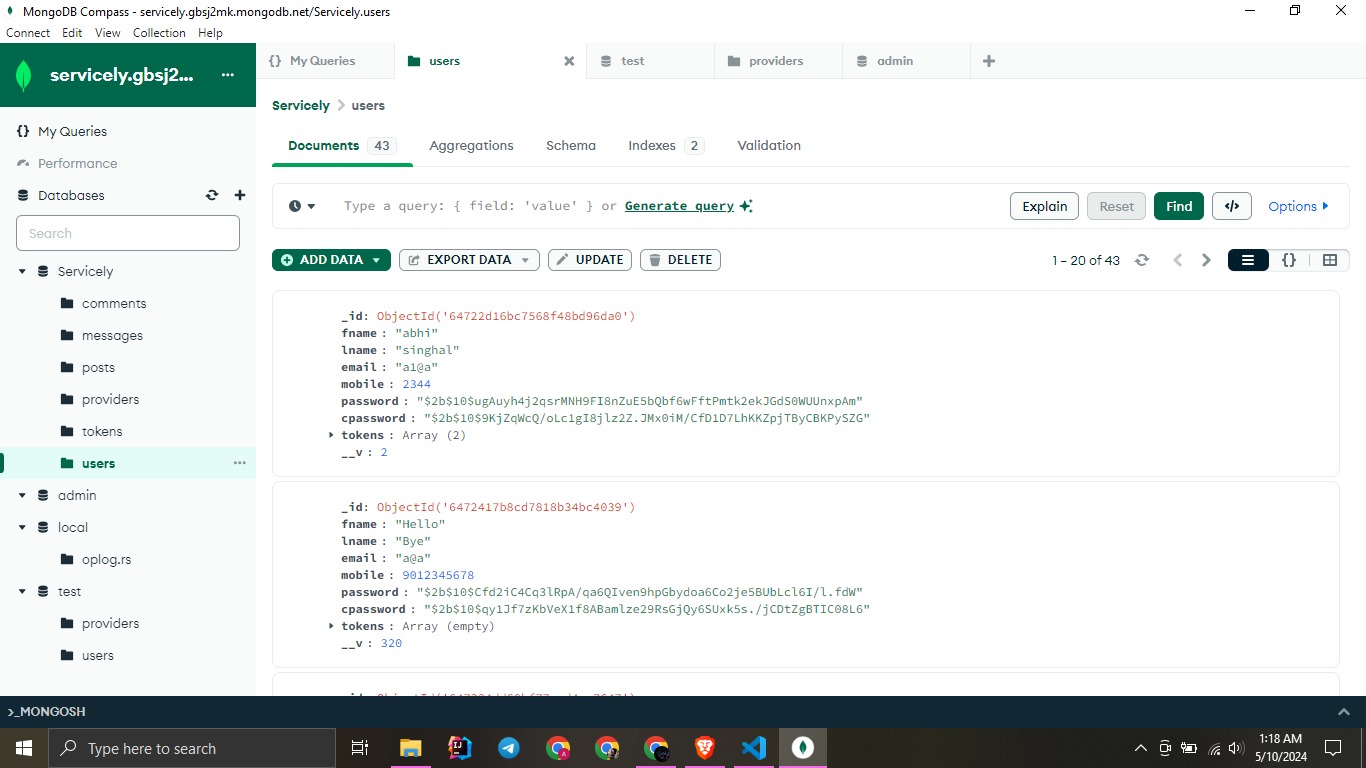
**Nearest Service Provider on the Basis of Location:** The platform offers a robust location-based search functionality that enables users to find service providers in their vicinity with ease. Users have the option to manually input their location or enable geolocation services to automatically detect their current position. Once the location is determined, the platform displays a comprehensive list of service providers sorted by proximity to the user's location, making it exceptionally convenient to find nearby options. Key findings include the impressive accuracy of the location-based search feature, the efficiency of retrieving and displaying relevant service providers, and high user satisfaction with the ability to easily access nearby services. Additionally, users appreciate the intuitive interface, the speed of the search results, and the overall reliability of the platform in helping them find local services quickly and efficiently.

**7.3 Brief Description of Database with Snapshots**

The Home Based Service Provider platform employs MongoDB, a NoSQL database, hosted on MongoDB Atlas. MongoDB's document-oriented structure suits the platform's complex data needs, while Atlas ensures high availability, scalability, and security. With automated backups and fine-grained access control, MongoDB Atlas provides robust data management. It offers seamless scalability to handle growing user traffic and data volumes. Additionally, built-in encryption safeguards sensitive user information, aligning with the platform's security standards. Overall, MongoDB Atlas serves as a reliable and efficient database solution for the platform's operations, supporting its functionality and ensuring data integrity.



**Figure 7.6 Database Collection 1**



**Figure 7.7 Database Collection 2**

**CHAPTER 8**

##### CONCLUSION AND FUTURE SCOPE

##### 8.1 Conclusion and Future Scope

In this final chapter, we provide a summary of the project's key findings and achievements, along with insights into potential future directions and areas for further exploration.

##### 8.1.1 Conclusion

The Home Based Service Provider platform introduces a modern solution to connect users with local service providers, leveraging advanced technologies like React JS, Tailwind CSS, Node.js, Express.js, MongoDB, and Socket.IO. By offering a seamless user experience, personalized advertisements, real-time communication, and location-based search, the platform aims to streamline the process of finding and booking home services. With its focus on enhancing user satisfaction and convenience, the platform addresses the challenges faced by both users and service providers in accessing and managing home services.

**8.1.2 Future Scope**

**Expansion of Services:** The platform can expand its services to include a wider range of home service categories, catering to diverse user needs.

**Integration with Payment Gateways:** Integrating payment gateways will allow users to make secure online payments for services, enhancing convenience and trust.

**AI-based Recommendations:** Implementing AI algorithms to analyze user behavior and provide personalized service recommendations can improve user satisfaction and retention.

**Geographic Expansion:** Scaling the platform to cover more regions and cities will increase its reach and impact, benefiting both users and service providers.

**Enhanced Analytics:** Incorporating advanced analytics tools will provide valuable insights into user preferences, service demand, and market trends, enabling informed decision-making and strategic planning.

**Social Integration:** Integrating social media platforms for user authentication and sharing reviews and recommendations can enhance user engagement and promote viral growth.

**Partnerships and Collaborations:** Forming partnerships with local businesses, service associations, and government agencies can help expand the platform's network and service offerings, fostering growth and sustainability.

**Continuous Improvement:** Regular updates, feature enhancements, and bug fixes based on user feedback and market trends will ensure the platform remains competitive and relevant in the evolving home services industry

**REFERENCES**

1. N. M. Indravasan, Adarsh G, Shruthi C, Shanthi K, Dadapeer ,“An Online System for Household Services”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume 6, Issue 13, 2018.
2. Amruta Amol Bhawarthi, Kaustubh Muley, Kavya Amrutkar, Devendra Kawade, Anushka Kausadikar, Ayush Kawane, Kaustubh Singh June 2023| IJIRT | Volume 10 Issue 1 | ISSN: 2349-6002 Website for Home Service Provider
3. Using the Advantages of NOSQL: A Case Study on MongoDB Divya Chauhan K. L. Bansal International Journal on Recent and Innovation Trends in Computing and Communication Volume: 5 Issue: 2 ISSN: 2321-8169 90 – 93
4. Shahrzad Shahriari, Mohammadreza Shahriari, Saeid gheiji. “E-Commerce And Its Impacts on Global Trend And Market” .International Journal of Research – Granthaalayah. Vol.3 (Iss.4): April 2015.
5. Kunal Bhalgat, Sayali Desai, Rajeshri Mayanaikar, Aaditya Pardeshi, Prof. Bhagyashree Dhakulkar “MAZDOOR”- Online Application for Household Services Issue 3, May-June-2021.
6. Dr. Krishna Kant Agrawal, Tanya Goel, Tarun Gariya, Vibhu Saxena, “At DoorStep: An Innovative Online Application for Household Services”, Journal of Xi'an University of Architecture & Technology, ISSN No : 1006-7930, Volume XII, Issue IV, 2020.
7. Ms. Prachi S. Tambe, Nikam Poonam, Gunjal Trupti, Jadhav Priti, Parakhe Sonali, “An Online System for Home Services”, International Journal of Scientific Development and Research (IJSDR), ISSN: 2455 2631, Volume 5, Issue 9, September 2020.
8. Hegde Sharaj Bhaskar Shyamala, Krishnamoorthy Rao, Padmanabha Bhandarkar, Prateek Prakash Vetekar, Geetha Laxmi,” An Android Application for Home Services”, International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0056, VOLUME: 07 ISSUE: 05, MAY 2020.

GitHub Link https://github.com/AavegTomar/ServiceProvider/tree/master

**Research Paper Submission Proof**

