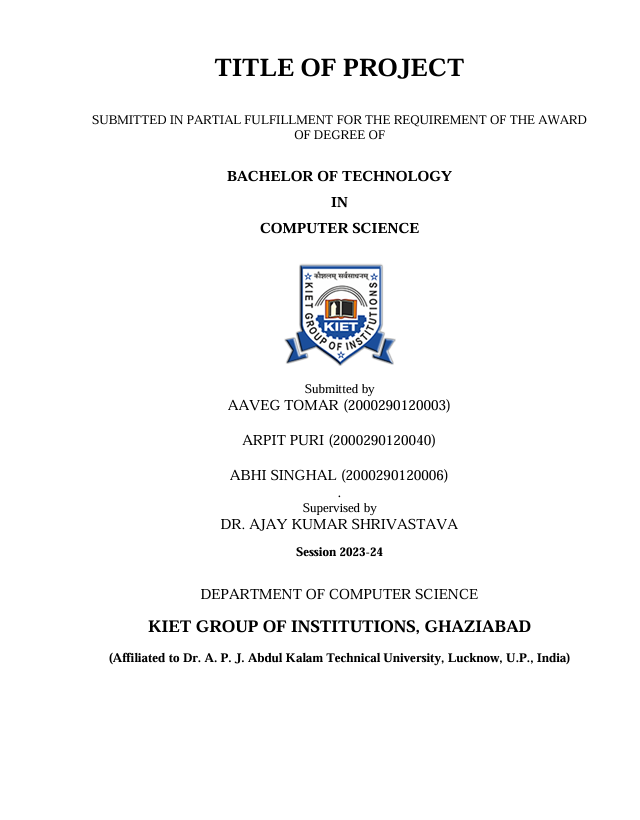
****

**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:- Aaveg Tomar , Abhi Singhal , Arpit Puri

Roll No.:- 2000290120003 , 2000290120006 , 2000290120040

Date:-

## CERTIFICATE

This is to certify that Project Report entitled “ Home Based Service Provider ” 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

(H.O.D.)

**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, Department of Computer Science, KIET, Ghaziabad, for his/her 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.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Shrivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

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

Date :

Signature: Signature: Signature:

Name: Aaveg Tomar Name : Arpit Puri Name : Abhi Singhal

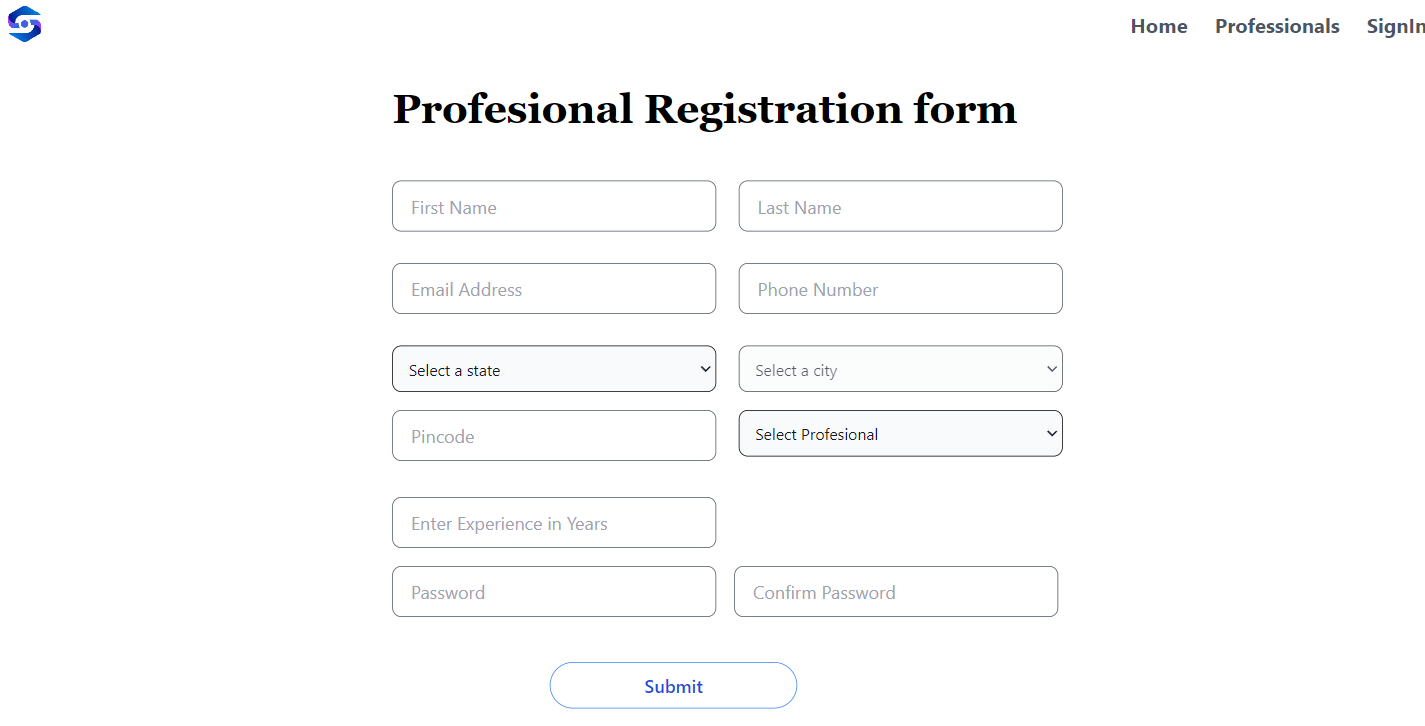
Roll No.: 2000290120003 Roll No.: 2000290120040 Roll No.: 2000290120003

**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 templating 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. The chat dashboard, powered by Socket.IO, provides a seamless channel for direct communication. Users can initiate conversations, ask questions, negotiate terms, and schedule appointments in real-time. Service providers receive instant notifications for new messages, ensuring prompt responses and efficient service delivery. This direct line of communication streamlines the entire process, eliminating the need for intermediary platforms or third-party services.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION……………………………………………………………………. | ii |
| CERTIFICATE……………………………………………………………………… | iii |
| ACKNOWLEDGEMENTS…………………………………………………………. | iv |
| ABSTRACT………………………………………………………………………..... | v |
| LIST OF FIGURES…………………………………………………………………. | ix |
| LIST OF TABLES…………………………………………………………………… | xi |
| LIST OF ABBREVIATIONS………………………………………………………. | xii |
|  |  |
| CHAPTER 1 INTRODUCTION PAGE NO.  1.1 INTRODUCTION TO PROJECT  1.2 PROJECT CATEGORY  1.3 OBJECTIVES  1.4 PROBLEM FORMULATION  1.5 PROPOSED SYSTEM  1.6 UNIQUE FEATURES OF THE SYSTEM  CHAPTER 2. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION  2.1 FEASIBILITY STUDY  2.2 SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT  2.3 SDLC MODEL TO BE USED  CHAPTER 3. SYSTEM DESIGN  3.1 DETAIL DESIGN  3.2 SYSTEM DESIGN USING DFD LEVEL 0 AND LEVEL 1  3.3 USE CASE DIAGRAM  3.4 DATABASE DESIGN  3.4.1 ER DIAGRAMS    CHAPTER 4. IMPLEMENTATION, TESTING, AND MAINTENANCE  4.1 INTRODUCTION TO LANGUAGES,TOOLS AND TECHNOLOGIES USED FOR IMPLEMENTATION  4.2 TESTING TECHNIQUES AND TEST CASES USED  CHAPTER 5. RESULTS AND DISCUSSIONS  5.1 USER INTERFACE REPRESENTATION  5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM  5.2 SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH  5.3 BACK ENDS REPRESENTATION  5.3.1 SNAPSHOTS OF DATABASE TABLES WITH BRIEF DESCRIPTION  CHAPTER 6. CONCLUSION AND FUTURE SCOPE  REFERENCES |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **1 INTRODUCTION**   * 1. **INTRODUCTION**   The Home Based Service Provider platform introduces a modern approach to accessing home services, leveraging cutting-edge technologies like React JS, Tailwind CSS, Node.js, Express.js, MongoDB, and Socket.IO. This platform empowers users to effortlessly connect with local service professionals such as electricians and plumbers. The frontend, built with React JS and Tailwind CSS, ensures a sleek and intuitive user interface. Provider profiles showcase relevant details such as experience and customer ratings, aiding users in their decision-making process. Enables service providers to view and respond to user ratings and reviews. Users benefit from timely and personalized advertisements within their account, increasing the likelihood of finding the services they need quickly and conveniently. A unique feature of the platform is its integrated chat functionality with Socket.IO, facilitating real-time communication between users and service providers. This direct interaction enhances trust and transparency, elevating the overall user experience. The platform streamlines home service tasks, offering convenience and satisfaction to users seeking reliable assistance for their household needs.  **1.2 PROJECT CATEGORY**  The Home Based Service Provider platform falls under the categories of  "Internet Based" and "Application or System Development," as it is an  online platform developed to connect users with local service providers,  using modern technologies for efficient communication.  1    **1.3 OBJECTIVES**   * + The project aims to provide easy access to local service providers without a middleman involved.   + A platform will be offered where users can directly interact with service providers, without any barriers in between.   + The platform will provide flexibility in pricing, allowing users to negotiate directly with providers for a fair and reasonable price.   + Users will be able to search for providers based on their location, making the process of finding suitable service providers easy and efficient.   + The goal of the project is to make it as easy and convenient as possible for individuals to connect with service providers and obtain the services they require in a timely and efficient manner.   + **Aim to support and promote local businesses by showcasing their services to a wider audience.**   + **Establish partnerships with trusted service providers to expand the range of services offered on the platform.**   **1.4 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 service providers. The  goal is to enhance user experience , improve service quality , and streamline  the process of finding and hiring local service professionals.  2  **1.5 UNIQUE FEATURES OF THE SYSTEM**   * **Enhanced User Experience:** By tracking user behavior and displaying relevant advertisements based on their frequent searches, the system provides a more personalized and efficient service discovery process. * **Timely and Personalized Ads:** Users benefit from timely and personalized advertisements within their account, increasing the likelihood of finding the services they need quickly and conveniently. * **Streamlined Service Discovery:** The platform's focus on direct communication between users and service providers, enabled by modern technologies, streamlines the process of finding and booking services. * **Real-time Communication:** Integration of SocketIO allows for real-time communication between users and service providers, facilitating quick responses and efficient service booking. * **Service Provider Ratings:** Enables service providers to view and respond to user ratings and reviews. * **Efficient Service Delivery:** Overall, the system aims to enhance the quality of service delivery by leveraging these technologies to improve user experience, speed, and efficiency. * **Location-Based Search: Users can search for providers based on their location, making it easy and efficient to find suitable service providers nearby.**   **3**  **2. REQUIREMENT ANALYSIS AND SYSTEM**  **SPECIFICATION**  **2.1 FEASIBILITY STUDY**  **The feasibility study for the Home Based Service Provider platform**  **covers three main aspects: technical, economical, and operational.**  **Technical Feasibility**   * **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.**   **Economical Feasibility**   * **The project's economic feasibility is promising, as it aims to streamline the connection between users and service providers, potentially increasing revenue for both parties.** * **The platform's flexible pricing model allows for negotiation between users and providers, fostering a competitive and fair marketplace.** * **The personalized advertising system provides an additional revenue stream, as service providers can pay for targeted ads to reach interested users.**   **Operational Feasibility**   * **The operational feasibility of the platform is high, as it simplifies the process of finding and hiring home service providers.** * **Users can easily search for providers based on their location and specific needs, reducing the time and effort required to connect with suitable professionals.** * **The integrated chat functionality facilitates seamless communication between users and providers, enhancing operational efficiency and customer satisfaction.**   **2.2 SOFTWARE REQUIREMENT SPECIFICATION**  **2.2.1 PRODUCT SCOPE**  The scope of our project is to designing a complete environment .for online service booking. The main aim of the project is to provider an easy to use application for services provided for customer. We often get frustrated while taking the appointment of service provider because there the many problems are occur,  like the service provider is busy art somewhere else or  his not receiving our call or his cost is very high according to problem. So in this project we will remove this headache.  **2.2.2 DATA REQUIREMENT**  **User Data**   * **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**   * **Provider profiles with details such as name, contact information, services offered, experience, and certifications.** * **Pricing information and service packages offered by providers.** * **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**   * User and provider location data for matching nearby service providers with user requests. * Geofencing data to define service areas and boundaries for providers.   **2.2.3 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.   **2.2.4 PERFORMANCE REQUIREMENT**  **Response Time**   * The platform should have fast response times for loading pages and displaying search results. * Aim for an average response time of under 5 seconds for most operations.   **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 chat functionality should have minimal latency for sending and receiving messages. * Ensure that messages are delivered instantly to maintain smooth communication between users and service providers.   **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.   **2.2.5 MAINTAINABILITY REQUIREMENT**  **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**   * Utilize Git for version control to track changes, manage branches, and collaborate with team members effectively. * Regularly commit code changes and follow a branching strategy for development and production releases.   **Dependency Management**   * Manage dependencies carefully, regularly updating libraries and packages to ensure security patches and performance improvements. * Use package managers like npm to handle dependencies and versions effectively.   **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.   **2.3 SDLC MODEL**  Waterfall model will be adapted for the development of the Home Based  Service Provider platform .  **Requirements Analysis**   * Detailed requirements will be gathered from stakeholders, focusing on user needs and system specifications for the home service platform.   **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, and updates to the MERN stack components will be performed to ensure the platform remains stable and secure.   **3. SYSTEM DESIGN**  **3.1 USE CASE DIAGRAM**      **3.2 WORK FLOW DIAGRAM** |  |
| **3.3 SYSTEM DESIGN**  **3.3.1 DFD LEVEL 0** |  |
|  |  |
|  |  |
| **3.3.2 USING DFD LEVEL 1 FOR ADMIN** |  |
| **3.3.3 DFD LEVEL 1 FOR FOR SERVICE PROVIDER** |  |
|  |  |
|  |  |
| **3.3.4 DFD LAVEL 1 FOR THE CUSTOMER** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **3.4 ER DIAGRAM** |  |
|  |  |
| **4. IMPLEMENTATION, TESTING, AND**  **MAINTENANCE**  **4.1 INTRODUCTION TO LANGUAGES AND**  **TECHNOLOGIES** |  |
| **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.   **4.2 INRODUCTION TO 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**.**   **VS Code**   * Integrated development environment (IDE) for coding, debugging, and project management**.**   **POSTMAN**   * API testing tool for validating backend APIs and endpoints.   **4.3 TESTING TECHNIQUES AND TEST CASES**  **4.3.1 MANUAL TESTING - DECISION TABLE**  **Test Case 1 : Professional Registration** |  |
| **Test Case 2 : Professional Login** |  |
| **Test Case 3 : Customer Registration** |  |
| **Test Case 4 : Customer Login** |  |
| **Test Case 5 : Chat Dashboard** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **4.3.2 AUTOMATION TESTING – SELINIUM**  **Test Case 1 : Customer Login**    **Test Case 2. Provider Registration**    **Test Case 3 . Provider Login**      **Test Case 4 . Chat Dashboard** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **5. RESULTS AND DISCUSSIONS**  **5.1 USER INTERFACE REPRESENTATION**  **5.1.1 USER HOME PAGE**   * User can access the various services from the home page after the login . |  |
|  |  |
| **5.1.2 Service Providers**   * When user click on connect it can see the information and rating of the service provider and communicate with them**.** |  |
|  |  |
|  |  |
| **5.1.3 USER SIGNIN PAGE**   * User do if have already have the account other wise it login as guest login or signup and can create account and then login . |  |
|  |  |
|  |  |
| **5.1.4 Chat Console**   * User and provider can chat and do negotiation on the price and also can fix the schedule according to its relevance time **.** |  |

**5.1.5 Service Provider Registration**

****

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 1.1 |  | 2 |
| 2.1 |  | 9 |
| 3.1 |  | 12 |

**LIST OF ABBREVIATIONS**

MERN MongoDb , ExpressJS , ReactJS , NodeJS

I/O Input/Output

NOSQL No Structure Query Language Database

APIs Application Program Interface

REST Representational State Transfer

CSS Cascading Style Sheet

**REFERENCES**

1. 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.
2. Yash Veer Singh; Harikesh Singh; Jitendra Kumar Chauhan
3. Ionic: Advanced HTML5 Hybrid Mobile App Framework. (2016). Retrieved August 15, 2016, from http://ionicframework.com/ [17] Firebase | App Success Made Simple. (2016). Retrieved on August 15, 2016, <https://firebase.google.com/>
4. Kunal Bhalgat1 , Sayali Desai1 , Rajeshri Mayanaikar1 , Aaditya Pardeshi1 , Prof. Bhagyashree Dhakulkar “MAZDOOR”- Online Application for Household Services Issue 3, May-June-2021.
5. 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.
6. Nikam Poonam R, Gunjal Trupti T, Jadhav Priti V, Parakhe Sonali K, Ms. Prachi S. Tambe , “Survey on Home Service Provider”, 2019 International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0056 , Volume: 06, Issue:12, Dec 2019.
7. Dr. Krishna Kant Agrawal, Tanya Goel, Tarun Gariya, Vibhu Saxena ,“AtDoorStep: An Innovative Online Application for Household Services”, Journal of Xi'an University of Architecture & Technology , ISSN No : 1006-7930, Volume XII, Issue IV, 2020.