Expert Homecare

Mini Project Report

Submitted by

Mathew Peter

Reg. No. : AJC20MCA-I044

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2024-2025

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "Expert Homecare**"** is the bona fide work of **Mathew Peter (Regno: AJC20MCA-I044)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2024-25.

Ms. Nimmy Francis

Internal Guide

Mr. Binumon Joseph

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"Expert Homecare"** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (INMCA) from APJ Abdul Kalam Technological University, during the academic year 2024-2025.

Date:30/10/2024 **Mathew Peter**

KANJIRAPPALLY **Reg: AJC20MCA-I044**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph , Assistant Professor** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Nimmy Francis** , **Assistant Professor** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

Mathew Peter

ABSTRACT

This project introduces an advanced online service platform for home maintenance and repair services, offering a comprehensive range of solutions including plumbing, electrical work, and electronic gadget repairs. The platform is designed to streamline the entire process, allowing customers to easily book services tailored to their specific needs, preferred appointment times, and locations. With a focus on delivering service excellence in the districts of Ernakulam and Kottayam, users can browse through detailed profiles of service professionals, which include their qualifications, years of experience, and authentic customer reviews. This ensures customers can make informed decisions when choosing the right expert for their home maintenance tasks.

To ensure maximum convenience, the platform is accessible via both desktop and mobile devices, featuring an intuitive, easy-to-navigate interface. Once a service is completed, customers have the opportunity to provide feedback, sharing their experiences and rating the quality of work. This feedback is invaluable in helping the company maintain high standards and continuously improve service quality.

CONTENT

List of Abbreviation

UML	-	Unified Modeling Language
CSS	-	Cascading Style Sheet
HTML	-	Hyper Text Markup Language
SQL	-	Structured Query Language
NF	-	Normal Form

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Expert Homecare is an innovative online platform designed to provide seamless booking services for home maintenance and repair needs. The main objective of the platform is to offer customers a user-friendly website where they can easily book a wide range of services, such as plumbing, electrical repairs, and electronic gadget servicing. The platform also includes an admin interface for managing services, allowing service providers to add, update, or remove service listings and manage appointments efficiently.

In this system, customers can log in to the website, browse through detailed service categories, and select the appropriate service for their needs. Once a booking is made, a service professional is automatically allocated to the job, and the professional's details are displayed on the website for the customer to review.

The platform also includes a Cancel Booking feature, allowing customers to cancel their appointments if their plans change. In addition, Expert Homecare provides excellent customer support via a helpline and an online chat feature, enabling users to quickly resolve any queries or issues related to their bookings, including cancellation or service-related inquiries.

In summary, the Expert Homecare Service Booking System is designed to offer a hassle-free and convenient solution for home maintenance service bookings. Key features include online service bookings, booking management, service professional allocation, cancellation and refund options, an admin dashboard, customer support, and feedback-driven service improvements. The platform leverages modern web technologies and frameworks to ensure a smooth and efficient booking experience for customers, while providing powerful administrative tools for managing home maintenance services.

1.2 PROJECT SPECIFICATION

- User-friendly Website: The platform features a user-friendly design, making it easy for customers to navigate through various home maintenance services and select their preferred service providers. Accessible via computers and smartphones, the website offers convenience for customers to book services anytime, anywhere.

- Online Booking Service: Customers can search and select from a wide range of home maintenance services like plumbing, electrical repairs, and electronic gadget servicing. They can filter available service providers based on location, ratings, availability, and expertise. Detailed profiles of service providers are available, including qualifications,

customer reviews, and pricing, allowing customers to make informed decisions.

- Booking Management: Customers can manage their bookings directly from the website. They can view the details of their bookings, including the selected service provider and service date, and cancel bookings if necessary.

- Service Provider Selection: Customers have the flexibility to choose their preferred service provider from a list of professionals offering the service they need. The profile of each service provider includes their experience, expertise, customer ratings, and service charges, helping customers select the right professional based on their specific requirements.

- Cancellation and Refund Policy: The website outlines the cancellation policy, including any applicable fees or refund procedures. Customers can cancel their bookings, and if eligible, receive refunds as per the company's refund policy. This provides transparency and flexibility for customers in case their plans change.

- Admin Dashboard: The admin dashboard allows the platform administrators to manage services, service providers, and customer bookings. Admins can add, update, and remove services, monitor booking activities, and ensure the smooth functioning of the system. Admins can also monitor overall platform activity, ensuring smooth operations.

- Employee Dashboard: The Employee Dashboard allows employees to manage the platform by adding new service providers and updating their profiles with qualifications, experience, and availability. They can also introduce new services with descriptions, pricing, and booking availability. Additionally, employees can view and respond to customer feedback, manage queries for timely resolutions, and track system activity related to services and providers to maintain high operational standards.

- Service Provider Dashboard: The Service Provider Panel enables service professionals to update their profiles with qualifications, experience, and areas of expertise, view assigned bookings and customer details for upcoming jobs, and update job statuses (e.g., in progress, completed) while providing feedback on completed tasks. Additionally, they can track customer reviews and ratings to enhance their service quality.

- Customer Support: Expert Homecare offers robust customer support through a helpline, email, and online chat. Customers can get assistance with booking inquiries, cancellations, and other service-related issues.

- Feedback and Analytics: After every service, customers can leave feedback and ratings for the service provider, helping maintain high service standards. The platform also uses analytics tools to track user activity and system performance, ensuring continuous improvement and high customer satisfaction.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Our website is designed to provide you with a convenient and seamless experience for booking home maintenance and repair services. Whether you need plumbing, electrical repairs, or electronic gadget servicing, our platform offers a user-friendly interface that allows you to easily book, manage, and enjoy home maintenance services from the comfort of your home or smartphone.

Available 24/7, our website lets you browse through a wide selection of services, select the ones that meet your needs, and book them with just a few clicks. Our dedicated team manages all aspects of the service, including the allocation of qualified professionals, booking management, and customer support, ensuring a smooth and hassle-free experience for our valued customers.

We understand the importance of flexibility in your plans, which is why our website includes features such as cancellation options to accommodate any changes you may encounter. We also prioritize customer feedback as a vital tool for improvement; our team regularly reviews and analyzes this feedback to continuously enhance our services and website functionality.

In summary, our website is committed to delivering a reliable and user-friendly platform for all your home maintenance needs, ensuring you receive top-notch service and support every step of the way

2.2 LITERATURE REVIEW

The existing system for booking home maintenance and repair services primarily relies on traditional methods, where customers call service providers directly on their mobile phones. This approach often leads to limited accessibility and delays in securing appointments. Communication between customers and service providers can be inefficient, resulting in misunderstandings regarding services, availability, and pricing.

Additionally, customers frequently experience a lack of transparency regarding service provider qualifications, experience, and customer reviews, making it difficult to select the right professional for their needs. The existing system is also inflexible, offering little opportunity for customers to modify or cancel bookings easily, which can lead to frustration[3].

The existing system for home maintenance and repair services lacks several important features that would improve customer experience, sustainability, and transparency. While customer

feedback is collected, the current system does not have a structured mechanism specifically aimed at gathering insights on environmental sustainability or community impact. Accessibility is also limited, as the model relies primarily on phone calls for service requests, making it challenging for customers to access services outside of business hours. Moreover, the system does not provide transparent information on service provider qualifications or customer reviews, which can hinder customers' confidence in selecting the right service providers. Finally, real-time updates and notifications, such as appointment status and service provider arrivals, are absent, potentially leaving customers uncertain about the progress of their service requests. Implementing these features would not only enhance customer satisfaction but also promote environmental responsibility and an improved user experience across the platform[10]. The natural system studied in the context of home maintenance and repair services involves understanding the environmental and social dynamics surrounding service delivery while ensuring sustainability and minimizing negative impacts on communities. This is crucial for fostering responsible practices that enhance customer satisfaction and promote long-term service viability[12].

The study of the natural system in home maintenance includes various aspects, such as:

- Ecological Impact Assessment: Evaluating the potential impacts of maintenance and repair activities on local environments, including assessing how services may affect ecosystems, wildlife, and natural habitats. This may involve examining the potential for resource depletion, pollution from materials used in repairs, and the impact of service operations on community green spaces[8].
- Sustainable Resource Management: Managing resources involved in home maintenance and repair services, such as materials, water, and energy, to minimize environmental impacts. This includes promoting responsible material sourcing, waste reduction, recycling, and energy-efficient practices during service delivery, while considering the ecological footprint of various maintenance tasks[7].
- Feedback Mechanisms: Implementing structured feedback mechanisms that allow customers to share their experiences and suggestions regarding service delivery. This feedback is essential for continuously improving service quality and making informed decisions about how services can better align with environmental sustainability and community well-being[6].

The designed system studied in the context of a home maintenance and repair services platform encompasses a digital interface that facilitates online booking, management, and customer

support for users seeking maintenance solutions. This system is crafted with user-friendly features and functionalities to ensure a seamless and convenient booking experience for customers, while also providing efficient administrative capabilities for managing services.
Some key components of the designed system of Expert Homecare may include:

- User-Friendly Interface: The platform boasts an intuitive design that allows customers to easily navigate through various service categories, view detailed service descriptions, and make bookings with minimal effort[4].
- Service Provider Profiles: Each service provider has a dedicated profile showcasing their qualifications, experience, and customer reviews, empowering users to make informed decisions when selecting professionals for their maintenance needs[16].
- Customer Support Channels: Multiple support channels, such as helplines, emails, and online chat, are integrated into the system to assist users promptly with any inquiries or issues related to their bookings[19].
- Feedback and Improvement Mechanisms: Structured feedback tools are incorporated to collect customer reviews and suggestions, driving continuous improvements in service quality, and enhancing user experience[20].
- Data Security Measures: The platform employs robust security protocols to safeguard user data and transactions, ensuring a secure and trustworthy environment for all users.
- Book Services: The platform allows customers to easily book services by browsing various categories of home maintenance and repair options, selecting their preferred service, choosing an appointment date and time, creating or logging into an account, providing service details, and receiving a confirmation notification, all while offering a flexible cancellation policy to accommodate changes in plans[23].

The home maintenance and repair service platform integrates a range of advanced technologies to provide a streamlined, user-friendly, and secure experience for customers. Built using Django for the backend, along with JavaScript, jQuery, CSS, and HTML for the frontend, the platform ensures a responsive and interactive user interface that adapts seamlessly to different devices. MySQL serves as the database, providing robust data storage and management capabilities for efficient handling of user accounts, bookings, and service provider information[29].

Key features implemented include an online booking system that allows customers to easily browse services, select appointment times, and make secure payments through multiple options. A real-time tracking feature enables customers to monitor the status of their service requests, offering transparency and reducing uncertainty. Additionally, the system integrates a

comprehensive messaging module, which facilitates direct communication between customers and service providers, enhancing clarity and customer support. To protect user data and transactions, advanced security measures have been implemented, including data encryption and secure login protocols.

For added functionality, a digital image processing module is in place to ensure that only human photos are used for service provider profiles, improving professionalism and trustworthiness. The platform also includes structured feedback and rating mechanisms, allowing customers to review and rate their experiences, helping to continually improve service quality. These implemented technologies create a comprehensive digital solution that not only simplifies the booking process but also prioritizes security, communication, and an enhanced user experience[15].

2.3 PROPOSED SYSTEM

The proposed system aims to enhance the booking experience for home maintenance and repair services by offering a user-friendly interface through a website and mobile application that allows customers to effortlessly browse and book services. It will feature an online booking system where customers can view available services, complete with detailed descriptions, pricing, and customer reviews for informed decision-making. Customers will be able to select their preferred appointment date and time, with real-time availability displayed to avoid scheduling conflicts. User account management will enable customers to track past bookings and preferences, while service provider profiles will offer transparency regarding qualifications and ratings. Real-time notifications will keep customers updated about bookings and reminders, and flexible cancellation and rescheduling options will simplify changes to appointments. The platform will support multiple secure payment methods for convenient transactions and provide integrated customer support through live chat, email, and helplines for timely assistance. Additionally, a feedback and ratings system will allow customers to share their experiences, fostering continuous improvement in service quality. Overall, this comprehensive solution will enhance customer satisfaction and streamline operations in the home maintenance and repair service industry. Adding digital image processing to validate profile photos adds a layer of professionalism and authenticity, ensuring that service providers' profiles are represented accurately. These elements together create a seamless experience for users while maintaining quality and trust in the services provided.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- Convenient Booking Process: The platform allows customers to browse various service categories, select the service they need, specify appointment details, and receive instant confirmation, simplifying the entire booking experience

- Increased Accessibility: Customers can book services anytime and anywhere through the user-friendly website and mobile application, removing the constraints of traditional phone calls and business hours.

- Improved Transparency: Detailed service provider profiles, including qualifications, experience, and customer reviews, empower customers to make informed decisions when selecting a service.

- Flexibility in Modifications: Customers can easily cancel or reschedule appointments through their accounts, providing greater flexibility and accommodating changes in plans.

- Diverse Payment Options: The platform offers multiple secure payment methods, making transactions convenient and accessible for a wider range of customers

- Feedback and Reviews: The website would have a feedback and reviews feature where customers can provide feedback on their service experience, rate the service, and leave reviews. This can help improve customer satisfaction and provide valuable insights for business improvement

- Customer Support: The website would offer various customer support options, such as live chat, email, or phone support, to assist customers with any inquiries or issues related to their bookings or other services.

- Efficient Management Tools: Admin and employee dashboards provide robust management capabilities for overseeing bookings, service providers, and customer

feedback, leading to improved operational efficiency

- Mobile Responsiveness: The website would be mobile responsive, allowing customers to easily browse and book service from their smartphones or tablets, providing convenience and accessibility.

- Scalability: The system is designed to scale easily, accommodating growth in service offerings and user base without significant changes to the underlying infrastructure

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The results of the feasibility study should be a report that recommends whether it is worth carrying on with the requirements engineering and system development process.

The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule
- To determine whether the software can be integrated with other existing software.

3.1.1 Economical Feasibility

The economic feasibility analysis demonstrates that developing and deploying the online home maintenance and repair services platform is economically viable. Despite initial investments, the platform promises significant revenue generation through service fees, subscriptions, and transaction fees, alongside operational cost savings and efficiency gains. Projected Return on Investment (ROI) and a manageable payback period substantiate the financial viability, while effective risk mitigation strategies mitigate potential market and technology risks. Overall, the platform presents a sound business opportunity, poised to enhance market competitiveness, profitability, and long-term sustainability in the home services industry[6].

3.1.2 Technical Feasibility

The technical feasibility analysis confirms that using Django for backend development, along with JavaScript, jQuery, CSS, HTML for frontend development, and SQLite for database management, is feasible for building the online home maintenance and repair services

platform. The chosen technologies offer robust features, scalability options, and strong community support necessary for developing a scalable, secure, and user-friendly application. The next steps involve detailed planning, development, and testing phases to ensure successful implementation within the allocated time and budget[8].

3.1.3 Behavioral Feasibility

Based on the behavioral feasibility analysis, it is clear that implementing the online home maintenance and repair services platform is not only viable but also beneficial for the organization. The platform's user-friendly interface and comprehensive features are designed to meet the needs of both homeowners and service professionals, leading to high levels of acceptance and satisfaction among users[9].

User Acceptance: The intuitive design of the platform makes it accessible for users of all technical backgrounds, ensuring that homeowners can easily navigate the service booking process. The platform's simplicity fosters quick adoption, reducing any resistance to using online services.

Training and Support: Effective training programs for service professionals and administrative staff will be critical in ensuring they are comfortable with the platform. Ongoing support through FAQs, user manuals, and help desks will provide users with resources to address any concerns, thereby enhancing their confidence in utilizing the system.

Communication Channels: The integration of various communication channels, including live chat and dedicated helplines, allows for immediate assistance and query resolution. This accessibility enhances user satisfaction and fosters a sense of reliability, encouraging users to engage actively with the platform.

Feedback Mechanisms: Incorporating user feedback mechanisms enables customers and service providers to share their experiences and suggestions. By actively responding to this feedback, the organization demonstrates its commitment to user satisfaction, which can strengthen customer loyalty and encourage continued use of the platform.

3.1.4 Feasibility Study Questionnaire

- **How do you currently find and book service providers for home maintenance and repairs?**

  I usually rely on recommendations from friends or family, or search for service providers online and book them over the phone.

- **What are the payment options offered by your company?**

  currently we offer debit card, Net banking facility for payment.

- **What features would you find most useful in an online platform for booking home maintenance services?**

  Useful features would include a user-friendly interface, detailed service provider profiles, real-time tracking, and secure payment options.

- **How does your company handle customer inquiries and complaints?**

  The exact process for handling customer inquiries and complaints by generally, customers can contact the company through various channels such as email, phone, or a support ticket system.

- **It's your company send automated notifications to customers, such as booking confirmations and remainders?**

  Yes, the system can send automated notifications to customers, such as booking confirmations, reminders, and updates on maintenance and repair requests.

- **What concerns do you have regarding online payments for home maintenance services?**

  My main concern with online payments is security. I want to ensure that my payment information is protected and that the transaction is safe.

- **How likely are you to use an online platform that allows direct communication with service providers?**

  I would be very likely to use a platform that allows direct communication with service providers, as it would make coordination easier.

- **How would you rate the importance of user reviews and ratings when selecting a service provider?**

  I would rate user reviews and ratings as extremely important. They give me confidence in my choice of service provider.

- **What types of services do you most commonly seek (e.g., plumbing, electrical, gadget repair)?**

  Most commonly seek plumbing and electrical services, with occasional gadget repair.

- **What are the most important factors when choosing a service provider (e.g., price, reliability, reviews)?**

  The most important factors for me are reliability, positive customer reviews, and reasonable pricing.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i7
RAM	-	8 G B
Hard disk	-	S S D

3.2.2 Software Specification

Front End	-	HTML, CSS, JavaScript
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, Django, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Django

Django is a high-level Python web framework that facilitates rapid development and clean, pragmatic design, following the Model-Template-View (MTV) architectural pattern to separate application logic from the presentation layer. It includes a powerful built-in admin interface for managing application data, an Object-Relational Mapping (ORM) system for database interaction, and a flexible URL routing mechanism that creates clean, SEO-friendly URLs. Django simplifies form handling, provides robust security features against common vulnerabilities, and is designed for scalability to handle high traffic loads. With a large and active community, Django offers a rich ecosystem of reusable packages and plugins, along with support for building RESTful APIs through the Django REST Framework. Additionally, it supports internationalization and localization, making it adaptable for different languages and regions. Overall, Django is a versatile framework suitable for developing a wide range of applications, from content management systems and e-commerce websites to social media applications and data-driven platforms.

3.3.2 SQLite

SQLite, the most widely used self-contained, serverless, and zero-configuration SQL database engine, is developed, maintained, and supported by the SQLite Development Team. The official SQLite website provides the latest information about SQLite software, including documentation, downloads, and tools.

SQLite is a database management system.

A database is a structured collection of data that can range from a small personal list to large sets of information in applications. SQLite manages this data, allowing users to add, access, and process data in a structured format. Unlike MySQL, SQLite is serverless, meaning it does not require a separate server process to operate. The entire database is stored in a single file on the local disk, making it lightweight and highly portable. This simplicity makes SQLite a central part of many applications, especially in mobile and embedded systems, where managing smaller amounts of data efficiently is crucial.

SQLite is commonly used in web browsers, operating systems, and applications, making it a versatile tool in modern computing environments.

SQLite databases are relational.

A relational database stores data in separate tables rather than putting everything in one large structure. SQLite follows this model, organizing data into tables with rows and columns. Unlike MySQL, SQLite does not require setting up a database server; instead, it stores everything in a single file, which makes it lightweight and easy to use. The tables, indexes, views, and triggers in SQLite offer a flexible way to structure and query data using SQL (Structured Query Language).

In SQLite, you can define relationships between different tables using primary keys, foreign keys, and constraints to enforce rules such as one-to-one, one-to-many, unique, or required relationships. This helps ensure that your data remains consistent, valid, and free from duplication or orphans.

SQLite also supports the SQL standard, including many SQL92, SQL:1999, and SQL:2003 features. Although it omits some advanced SQL features found in larger systems like MySQL, SQLite's implementation of SQL is still powerful enough for most applications. SQLite is

particularly well-suited for embedded applications, mobile apps, and small to medium-sized projects that require easy data management without the overhead of a full-fledged database server.

SQLite software is Public Domain.

Unlike MySQL, which uses an Open-Source license, SQLite is in the public domain, meaning it is free for anyone to use, modify, and distribute without any licensing restrictions. You can download and use SQLite without cost, and you are free to include it in any application, whether open-source or proprietary, without needing to pay royalties or comply with any specific license.

Since SQLite is public domain, there are no restrictions on how you can use or modify it, and you are not required to make your changes publicly available. This makes SQLite particularly attractive for developers who want a lightweight database engine for both open-source and commercial projects without the legal complexities of other database systems. If you are interested in more detailed licensing information, SQLite offers optional warranty and support agreements for commercial applications.

The SQLite Database Engine is lightweight, reliable, and simple to use.

SQLite is a fast, serverless database engine that requires minimal setup and is designed to operate with low overhead, making it ideal for desktop, laptop, and mobile devices. It stores all data in a single file and runs directly within the application, eliminating the need for a separate database server. This makes SQLite highly portable and easy to integrate into software projects. While not intended for handling heavy, concurrent transactional workloads like MySQL, SQLite is excellent for small to medium-sized projects where simplicity, efficiency, and quick deployment are key priorities. It operates efficiently alongside other applications, using minimal system resources.

SQLite works as an embedded database system.

The SQLite database software is a serverless, self-contained database engine designed to be embedded directly within applications. Unlike MySQL, which follows a client/server model, SQLite operates entirely within the application's process, making it extremely lightweight and

easy to manage. SQLite does not require separate client programs, administrative tools, or backends—it is a single library that provides a complete database system. This makes it an ideal choice for embedded systems, mobile devices, and applications that require a small, fast, and standalone product without the overhead of a client/server setup. SQLite also offers a range of application programming interfaces (APIs) for easy integration with different programming environments.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance, and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- ❖ Class diagram
- ❖ Object diagram
- ❖ Use case diagram
- ❖ Sequence diagram
- ❖ Activity diagram
- ❖ State chart diagram
- ❖ Deployment diagram
- ❖ Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing, and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

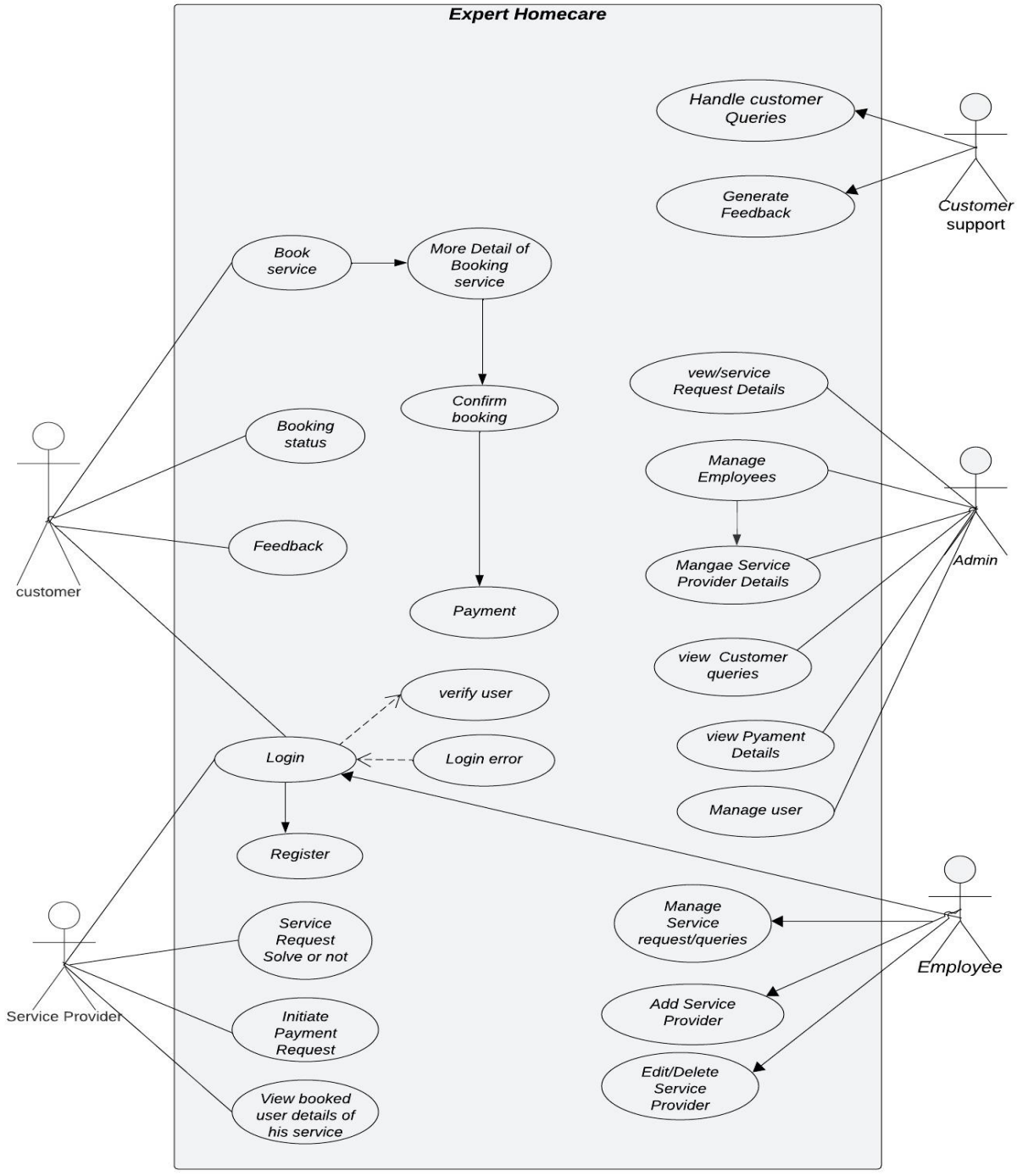
- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to them roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.

- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.

Use notes whenever required to clarify some important points.



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the in which these interactions take place. We can also use the terms event diagrams order or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

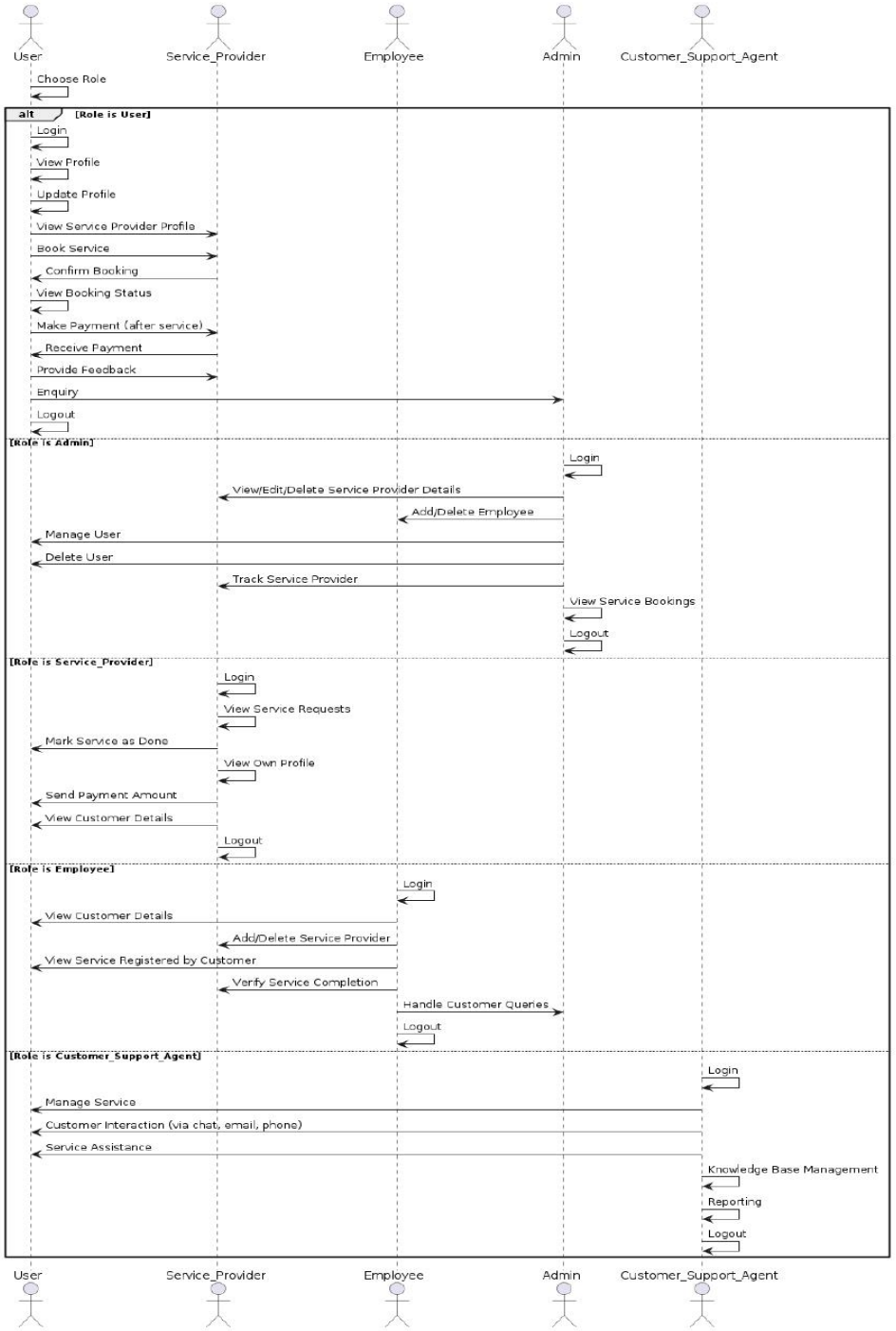
i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

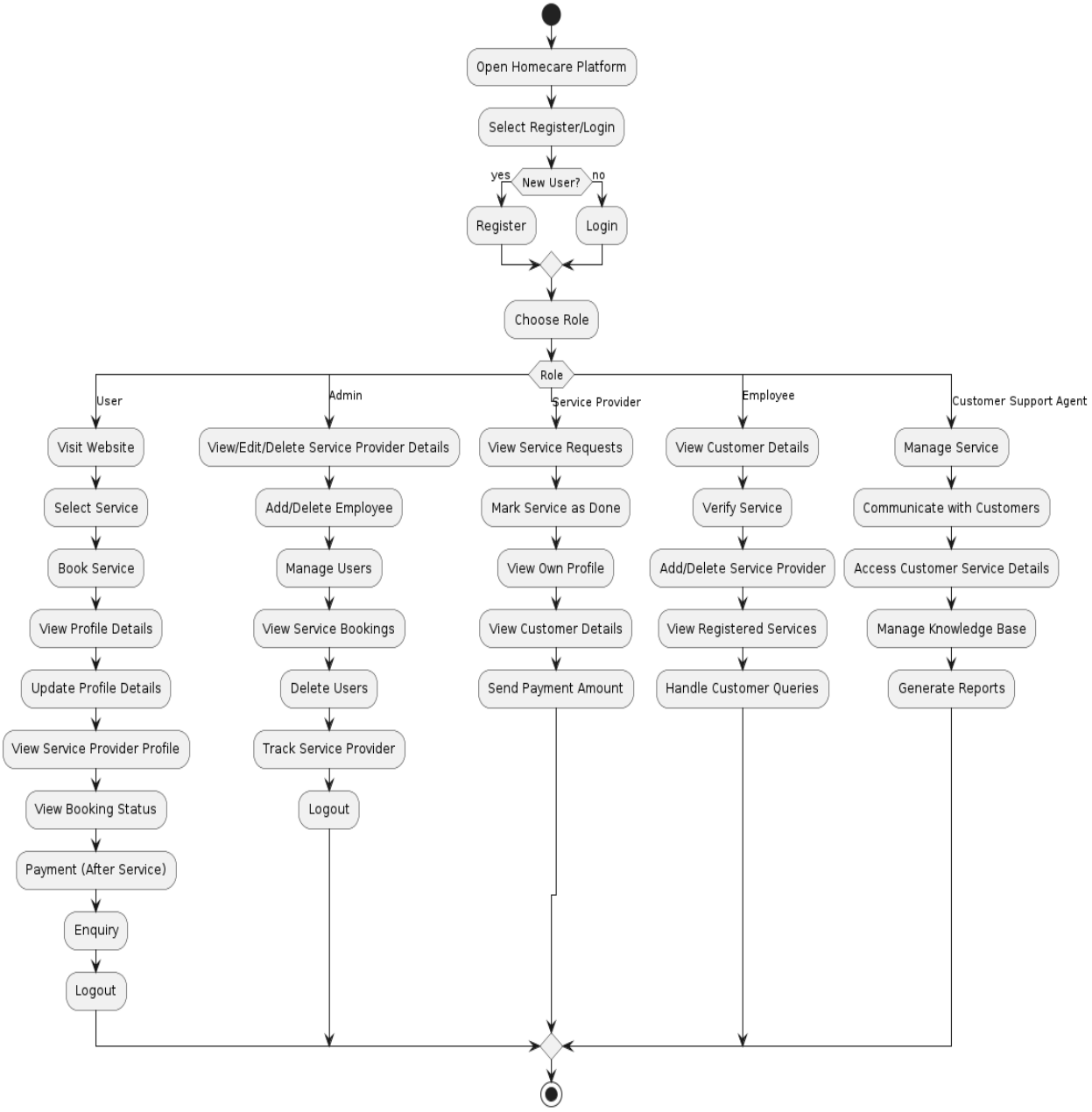

User
Service_Provider
Employee
Admin
Customer_Support_Agent
Choose Role
alt
[Role is User]
Login
View Profile
Update Profile
View Service Provider Profile
Book Service
Confirm Booking
View Booking Status
Make Payment (after service)
Receive Payment
Provide Feedback
Enquiry
Logout
[Role is Admin]
Login
View/Edit/Delete Service Provider Details
Add/Delete Employee
Manage User
Delete User
Track Service Provider
View Service Bookings
Logout
[Role is Service_Provider]
Login
View Service Requests
Mark Service as Done
View Own Profile
Send Payment Amount
View Customer Details
Logout
[Role is Employee]
Login
View Customer Details
Add/Delete Service Provider
View Service Registered by Customer
Verify Service Completion
Handle Customer Queries
Logout
[Role is Customer_Support_Agent]
Login
Manage Service
Customer Interaction (via chat, email, phone)
Service Assistance
Knowledge Base Management
Reporting
Logout
User
Service_Provider
Employee
Admin
Customer_Support_Agent

4.2.3 State Chart Diagram

A particular form of diagram used in computer science and related subjects to explain how systems behave is called a state diagram. State diagrams call for the system being represented to consist of a finite number of states; occasionally, this is the case, and other times, it's only an acceptable abstraction. State diagrams come in a variety of shapes and sizes, each with a distinct meaning. State diagrams are there to provide a system's behavior an abstract explanation. In order to evaluate and demonstrate this behavior, a sequence of events that could occur in one or more hypothetical states is employed. "Each diagram typically depicts objects of a single class and monitor the different states of its objects across the system," according to this.

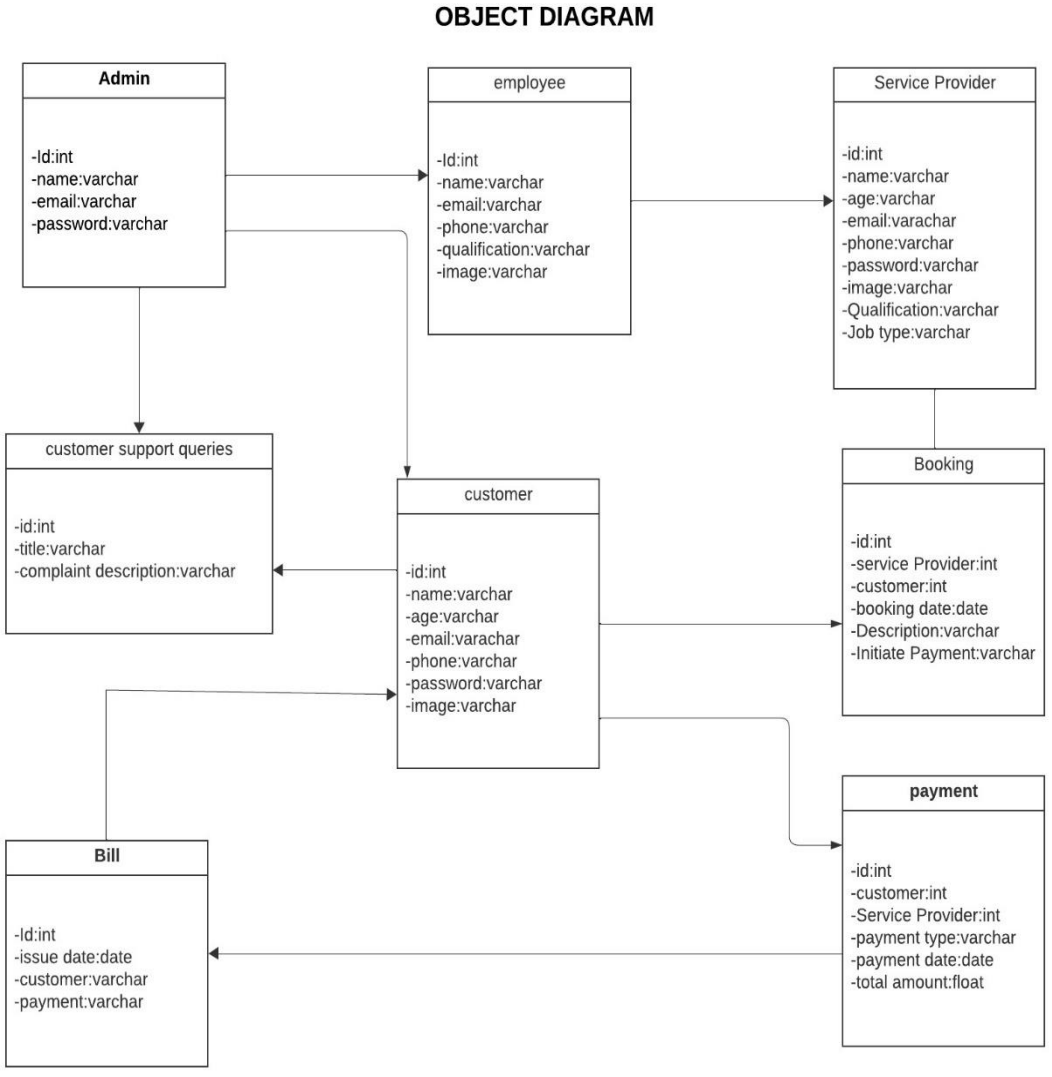
4.2.4 Activity Diagram

Activity diagrams depict how different levels of abstraction of activities are linked to provide a service. Typically, an event should be completed by some activities, particularly when the activity is intended to do multiple separate goals that need coordination. Another typical requirement is how the events in a single use case interact with one another, particularly in use cases where operations may overlap and require coordination. It may also be used to show how a collection of interrelated use cases interacts to reflect business operations.


Open Homecare Platform
Select Register/Login
New User?
yes
no
Register
Login
Choose Role
Role
User
Admin
Service Provider
Employee
Customer Support Agent
Visit Website
Select Service
Book Service
View Profile Details
Update Profile Details
View Service Provider Profile
View Booking Status
Payment (After Service)
Enquiry
Logout
View/Edit/Delete Service Provider Details
Add/Delete Employee
Manage Users
View Service Bookings
Delete Users
Track Service Provider
Logout
View Service Requests
Mark Service as Done
View Own Profile
View Customer Details
Send Payment Amount
View Customer Details
Verify Service
Add/Delete Service Provider
View Registered Services
Handle Customer Queries
Manage Service
Communicate with Customers
Access Customer Service Details
Manage Knowledge Base
Generate Reports

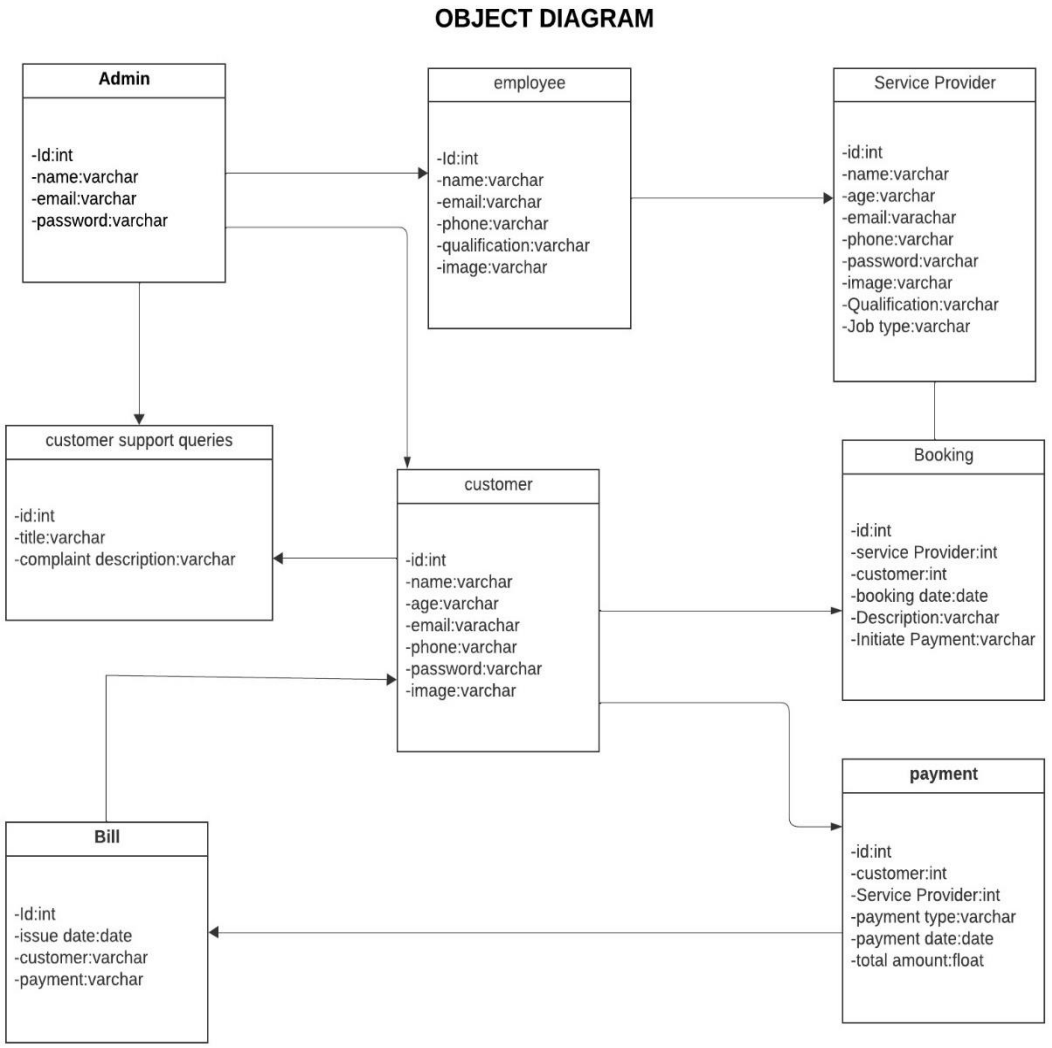
4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of the application. Class diagrams are useful for visualizing, describing, and documenting various system components as well as for writing executable code for software applications. A class diagram describes the constraints imposed on the system together with the properties and operations of a class. The only UML diagrams that can be directly converted into object-oriented languages are class diagrams, which are extensively utilized in the designing of object-oriented systems. An assortment of classes, interfaces, affiliations, partnerships, and limitations are displayed in a class diagram. It also goes by the name "structural diagram."



4.2.6 Object Diagram

Class diagrams are necessary before object diagrams can be created since they are the ancestor of object diagrams. An object diagram represents a particular instance of a class diagram. The underlying concepts used in class and object diagrams are the same. Object diagrams may also describe the static view of a system, although this static view only depicts a current state of the system. Object diagrams are used to show links between a set of things



4.2.7 Component Diagram

Component diagrams have different behaviors and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams may also be used to construct systems that can be run

4.2.8 Deployment Diagram

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical components of a system, deployment diagrams assist describe the hardware structure of a system

4.2.9 Collaboration Diagram

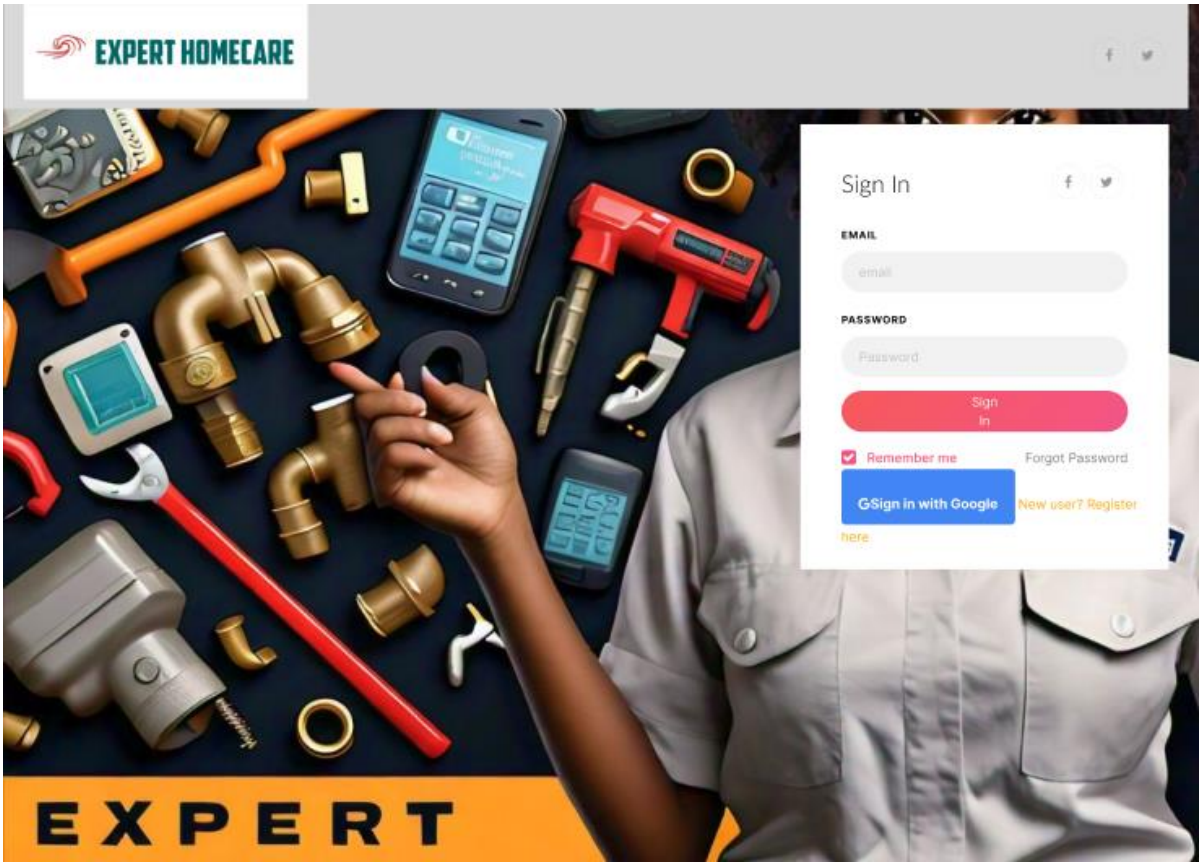
A collaboration diagram is a type of diagram that shows the interactions between objects in a system. It focuses on the roles and responsibilities of each object and how they collaborate to achieve a common goal. A collaboration diagram is also known as a communication diagram or an interaction diagram.

A collaboration diagram consists of objects, links, and messages. Objects are represented by rectangles with the object name and optionally the class name. Links are represented by solid lines that connect objects. They indicate the relationships or associations between objects. Messages are represented by arrows that show the direction and sequence of communication between objects. They can have labels that indicate the name, parameters, and return value of the message.

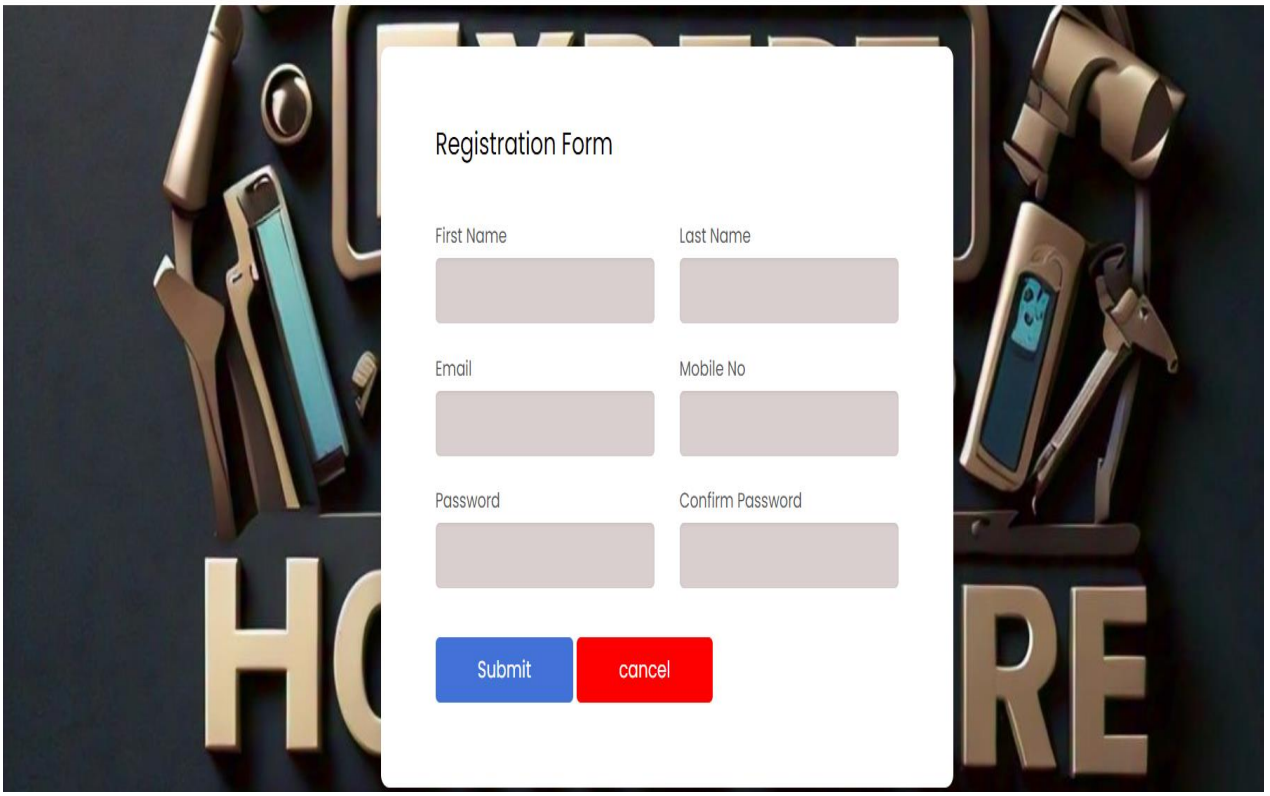
A collaboration diagram can be used to model different scenarios or use cases in a system. It can help to visualize the dynamic behavior and interactions of objects, as well as their roles and responsibilities. A collaboration diagram can also show the concurrency and synchronization of messages, as well as conditional and iterative logic.

4.3 USER INTERFACE DESIGN USING FIGMA

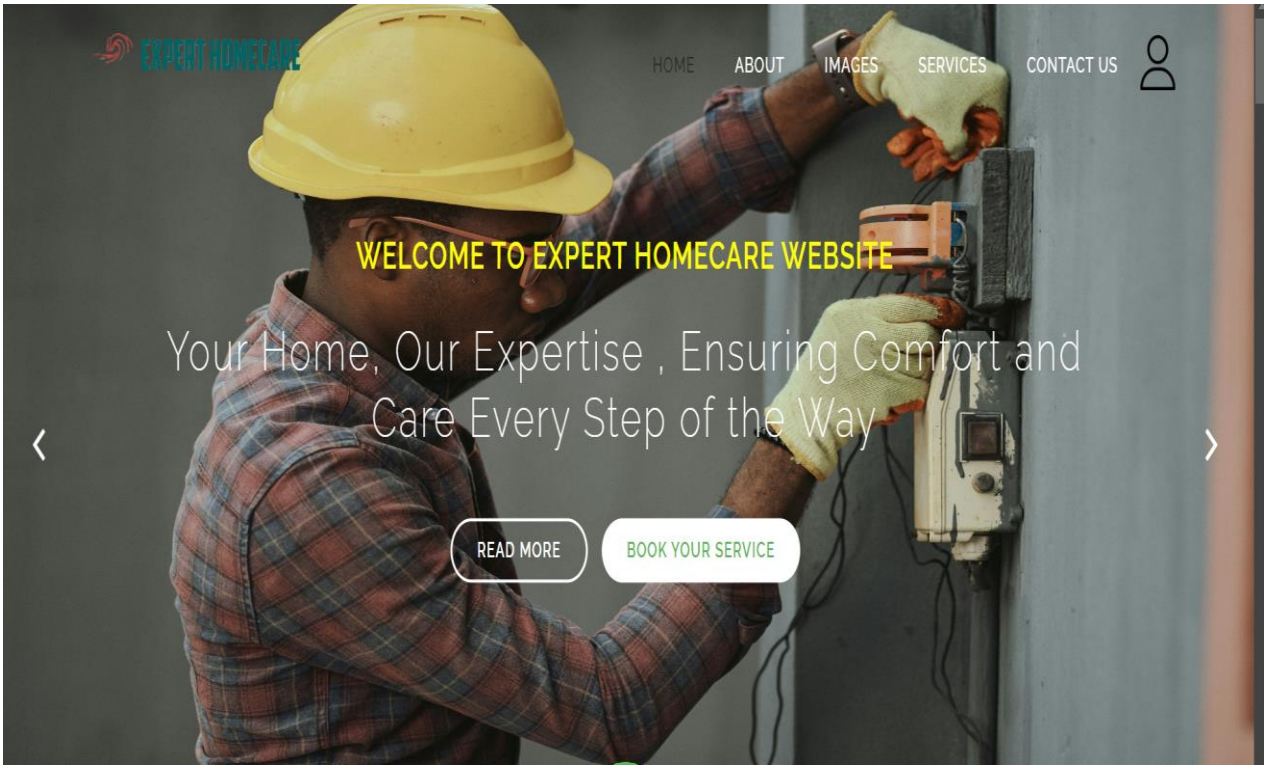
Form Name: Login



Form Name: Register



Form Name: Home Page



4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys, and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute.

Fourth Normal Form

The fourth normal form (4NF) is a database normalization rule that further refines data modeling by addressing multi-valued dependencies. When a table contains multiple

independent sets of repeating data, we can break it down into smaller tables, with each table containing only one set of related data.

This reduces data redundancy and improves data consistency by ensuring that each table represents a single, well-defined concept or entity. To achieve 4NF, we need to ensure that all multi-valued dependencies are removed from the table, and that each table contains only attributes that are functionally dependent on the primary key.

Fifth Normal Form

5NF is indeed the highest level of normalization in relational database design, and it deals with complex data models that involve multiple overlapping multi-valued dependencies. In 5NF, tables are decomposed into smaller tables in order to eliminate any possible redundancy caused by overlapping dependencies, while ensuring that there is no loss of data.

The goal of 5NF is to ensure that each table represents a single entity or relationship, and that the data is organized in a way that minimizes redundancy, eliminates anomalies, and improves data integrity. By breaking down tables into smaller, more specialized tables, 5NF helps to eliminate potential issues with update anomalies, insertion anomalies, and deletion anomalies, which can occur when data is not properly normalized.

4.4.3 Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize(filter) the input string while processing a SQL query based on user input. For instance, the user and password input is a typical scenario. In that scenario, the server response would provide access to the 'target user' account without requiring a password check.

4.4.4 Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order.

4.5 TABLE DESIGN

1.Login_Table

Primary key: Login_Id

No:	Fieldname	Data Type	Size	Key Constraints	Description
1	Login_Id	Int	10	Primary Key	Primary Key of Login Table
2	Login_username	Varchar	10	Not Null	Username
3	Login_Password	Varchar	10	Not Null	Password
4	Login_Status	Varchar	10	Not Null	Active or Inactive

2.Users_Table

Primary key: User_Id

Foreign key: Login_Id references to Login Table

No:	Fieldname	Data Type	Size	Key Constraints	Description
1	User_Id	Int	20	Primary Key	Primary Key of User Details Table
2	Login_Id	Int	10	Foreign Key	Key References to Login Table
3	Users_First Name	Varchar`	20	Not Null	User First Name
4	Users_Last Name	Varchar	20	Not Null	User Last Name
5	Users_Address	Varchar	20	Not Null	User Address
6	Users_Phone	Varchar	20	Not Null	User Phone
7	Users_Email	Varchar	20	Not Null	User Email
8	Users_Photo	Varchar	20	Not Null	User Photo

3. Service_Provider_Table

Primary key: Provider_Id

No:	Fieldname	Data Type	Size	Key Constraints	Description
1	Provider_Id	Int	20	Primary Key	Primary Key Of Service Provider Table
2	Provider_Name	Varchar	20	Not Null	Service Provider Name
3	Provider_Address	Varchar	20	Not Null	Service Provider Address
4	Provider_Phone	Varchar	20	Unique	Service Provider Phone
5	Provider_Email	Varchar	20	Not Null	Service Provider Email
6	Provider_Occupation	Varchar	10	Not Null	Service Provider Occupation
7	Provider_Location	Varchar	80	Not Null	Service Provider Location
8	Provider_Experience	Varchar	2	Not Null	Service Provider Experience
9	Provider_Education_ Details	Varchar	10	Not Null	Service Provider Education Details
10	Provider_Status	Varchar	10	Not Null	Active or Inactive
11	Provider_Joindate	Varchar	20	Not Null	Service Provider Join Date

4.Booking_Table

Primary key: Book_Id

Foreign key: Provider_Id references table Service Provider Table

Foreign key: User_Id references table Users Table

No:	Fieldname	Data type	Size	Key Constraints	Description
1	Book_Id	Int	20	Primary Key	Primary Key of Booking Table
2	provider_Id	Int	20	Foreign Key	Key References to Service Provider Table
3	User_Id	Int	20	Foreign Key	Key References to User Details Table
4	Booking_ Date	Int	100	Foreign Key	Key References to Booking_Dates Table
5	Booking_Status	Varchar	20	Not Null	Booking Status
6	Booking_Address	Varchar	50	Not Null	Booking Address of Service Needed

7	Booking_Notes	Varchar	20	Not Null	Small Description About the Service that we need
8	Booking_Amount	Varchar	5	Not Null	Amount Should be Paid After the Service Done
9	Booking_Status	Varchar	20	Not Null	Booking Service Has Been Completed or not

5.Employees_Table

Primary key: User_Id

Foreign key: Login_Id references table Users_Table

No:	Fieldname	Data type	size	Key Constraints	Description
1	Employee_Id	Int	20	Primary Key	Primary Key of Employee Table
2	Employee_Name	Varchar	20	Not Null	Employee Name
3	Employee_Address	Varchar	20	Not Null	Address
4	Employee_Email	Varchar	20	Not Null	Email
5	Employee_Phone	Varchar	20	Not Null	Phone
6	Employee_Location	Varchar	20	Not Null	Employee Location
7	Employee_year of experience	int	2	Not Null	Year of Experience
8	Employee_joindate	Varchar	20	Not Null	Employee join date
9	Employee_experience certificate	Varchar	20	Not Null	Experience
10	Employee_Driving Licence	Varchar	20	Not Null	Driving Licence
11	Employee_Status	Varchar	20	Not Null	Active or Inactive

6.Payments_Table

Primary key: Payment_Id

Foreign key: Book_Id references table BookingTable

No:	Fieldname	Data type	size	Key Constraints	Description
1	Payment_Id	Int	20	Primary Key	Primary Key of Payment Table
2	Book_Id	Int	20	Foreign Key	Key References to Book Details Table
3	Payment_Date	Date	20	Not Null	Payment Date
4	Payment_Amount	Varchar	20	Not Null	Amount wanted to pay on trip
5	Payment_Status	Varchar	20	Not Null	success or failed

7.Feedbacks_Table

Primary key: Feedback_Id

Foreign key: User_Id references table Users Table

Foreign key: Service_Provider_Id references table Service Provider Table

No:	Fieldname	Data type	size	Key Constraints	Description
1	Feedback_Id	Int	20	Primary Key	Primary Key of Feedback Table
2	User_Id	Varchar	20	Foreign Key	Key References to User Details Table
3	Service_Provider_Id	Int	20	Foreign Key	Key References to Service Provider Table
4	Service_Id	Int	20	Foreign Key	Key References to Service Table
5	Feedback_About Service	Varchar	500	Not Null	About Service
6	Feedback_Rating	Int	2	Not Null	Give Rating For Our Service Done
7	Feedback_Suggestion	Varchar	100	Not Null	Small Description What Need to Improve for our service

8.Service_Table

Primary key: Service_Id

No:	Fieldname	Data type	size	Key Constraints	Description
1	Service_Id	Int	20	Primary Key	Primary Key of ServiceTable
2	Service_name	varchar	30	Not Null	Service Name
3	Service_Description	varchar	70	Not Null	Service_Description
4	Status_Status	Varchar	10	Not Null	Service Status

9.Booking_Dates_Table

Primary key: Booking_Dates_Id

Foreign key: Booking_Id references table Booking Table

No:	Fieldname	Data type	size	Key Constraints	Description
1	Booking_Dates_id	Int	20	Primary Key	Primary Key of Booking Dates
2	Booking_Id	Int	10	Foreign Key	Key References to Booking Table
3	Time_slot	varchar	10	Not Null	Time Slot
4	Booking_Dates	Varchar	10	Not Null	Booking_Dates

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user wanted[22].

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met. There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation, and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is Whitebox oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested[20].

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested with all forms, code, modules, and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs.
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

A test case suite is executed using specialized automated testing software tools as part of the software testing technique known as automation testing. The test stages are meticulously carried out by a human performing manual testing while seated in front of a computer. Additionally, the automation testing software may generate thorough test reports, compare expected and actual findings, and enter test data into the System Under Test. Software test automation necessitates significant financial and material inputs. Repeated execution of the same test suite will be necessary during subsequent development cycles. This test suite can be recorded and replayed as needed using a test automation tool. No further human involvement is needed once the test suite has been automated.

5.2.6 Selenium Testing

Selenium is a free and open-source tool for testing web applications across multiple browsers and operating systems. Selenium Test Scripts can be written in different programming languages, including Java, C#, JavaScript, Python, etc. Automation performed using the Selenium framework is referred to as Selenium Automation testing.

Example:

Test Case 1

Code

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.edge.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.microsoft import EdgeChromiumDriverManager
from selenium.webdriver.edge.options import Options
import unittest
import time
class TestUserRegistration(unittest.TestCase):
    @classmethod
    def setUpClass(cls):
        # Setup Edge WebDriver
        edge_service = Service(EdgeChromiumDriverManager().install())
        edge_options = Options()
        cls.driver = webdriver.Edge(service=edge_service, options=edge_options)
        cls.driver.maximize_window()
    def test_register_user(self):
        driver = self.driver
        driver.get("http://127.0.0.1:8000/register")
        first_name_input = driver.find_element(By.ID, "txt1")
        first_name_input.send_keys("Mathew")
        last_name_input = driver.find_element(By.ID, "txt2")
        last_name_input.send_keys("Peter")
        phone_input = driver.find_element(By.ID, "phone1")
        phone_input.send_keys("9744594282")
        email_input = driver.find_element(By.ID, "email1")
        email_input.send_keys("arya@gmail.com")
        password_input = driver.find_element(By.ID, "password")
        password_input.send_keys("Mathew@2002")
        confirm_password_input = driver.find_element(By.ID, "cpassword")
        confirm_password_input.send_keys("Mathew@2002")
        submit_button = driver.find_element(By.ID, "register")
        submit_button.click()
        time.sleep(5)
    @classmethod
    def tearDownClass(cls):
        cls.driver.quit()
if __name__ == "__main__":
    unittest.main()
```

Screenshot

```
============== RESTART: C:\Users\Mail2\Downloads\seleniumtest2.py ==============
**************************************************
***************REGISTRATION TEST**********************
**************************************************
Registration successful!
.
----------------------------------------------------------------------
Ran 1 test in 37.371s

OK
```

Test Report

Test Case 1

Project Name: Expert Homecare					
Registration Test Case					
Test Case ID: Test_1			**Test Designed By: Mathew Peter**		
Test Priority(Low/Medium/High):			**Test Designed Date: 18/10/2024**		
Module Name: Registration Page			**Test Executed By: Nimmy Francis**		
Test Title: Registration of user account			**Test Execution Date: 20/10/2024**		
Description: Registration user account					
Pre-Condition: User should have unique email					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigate to the Registration page	http://127.0.0.1:8000/register/	Registration page should be displayed	Registration page was loaded	Pass
2	Enter first name	“Mathew”	The first name field should be filled with "Mathew"	First name entered	Pass
3	Enter last name	“Peter”	The last name field should be filled with "Peter"	Last name entered	Pass
4	Enter email	“mathew@gmail.com	The email field should be filled with “arya@gmail.com”	Email entered	Pass
5	Enter phone number	“9697969696”	The phone number field should be filled with “9744594282"	Phone number entered	pass
6	Enter password	“Mahtew#2002”	The password field should be filled with	Password inserted	Pass

			“Mahtew#2002”		
7	Enter confirm password	“Mahtew#2002”	The confirm password field should be filled with “Mahtew#2002”	Confirm password inserted	Pass
8	Click on the register button	N/A	If registration is successful, user should be redirected to login page	User was redirected to login page	Pass
Post-Condition: First name, last name, email, phone, photo, password, confirm password can be registered					

Test Case 2:

Code

```
from selenium import webdriver
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.edge.service import Service
from webdriver_manager.microsoft import EdgeChromiumDriverManager
# Clear cached EdgeDriver
import os
import shutil
cache_dir = os.path.expanduser("~/.cache/selenium")
if os.path.exists(cache_dir):
    shutil.rmtree(cache_dir)
# Set up the driver using WebDriverManager and Service object for Edge
service = Service(EdgeChromiumDriverManager().install())
driver = webdriver.Edge(service=service)
# Navigate to the login page
driver.get("http://127.0.0.1:8000/login1/")
# Enter the username
username_input = driver.find_element(By.ID, "email")
username_input.send_keys("mathew@gmail.com")
# Enter the password
password_input = driver.find_element(By.ID, "password")
password_input.send_keys("Mathew@2002")

# Submit the form
submit_button = driver.find_element(By.ID, "submit")
submit_button.click()

# Wait for 10 seconds
time.sleep(10)
# Close the browser window
driver.quit()
```

Screenshot

```
**************************************************
***************Login TEST***********************
**************************************************
Login successful!
.
----------------------------------------------------------------------
Ran 1 test in 23.763s

OK
```

Test report

Test Case 2

Project Name: Expert Homecare					
Login Test Case					
Test Case ID: Test_2			**Test Designed By: Mathew Peter**		
Test Priority (Low/Medium/High):			**Test Designed Date: 20/10/2024**		
Module Name: Login Page			**Test Executed By: Nimmy Francis**		
Test Title: Login to user account			**Test Execution Date: 20/10/2024**		
Description: Login user account					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigate to the login page	http://127.0.0.1:8000/login1/	Login page should be displayed	Login page was loaded	Pass
2	Enter email	“mathew@gmail.com”	The email field should be filled with “mathew@gmail.com”	email entered	Pass
3	Enter Password	“Mathew$2002”	The password field should be filled with “Mathew$2002”	Password entered	Pass
4	Click on	N/A	If login is successful,	User was redirected to	Pass

	the submit button		user should be redirected to user login page	user login page	
Post-Condition: Email and password is checked with database values for successful login.					

Test Case 3:

Code

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
import unittest
class TestService(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()  # Ensure you have ChromeDriver set up properly
        self.driver.get("http://127.0.0.1:8000/login1/")
        sleep(2)
    def test_add_place(self):
        driver = self.driver
        username_input = driver.find_element(By.ID, "username")
        username_input.send_keys("mathew@gmail.com")
        sleep(2)
        password_input = driver.find_element(By.ID, "password")
        password_input.send_keys("admin")
        sleep(2)
        submit_button = driver.find_element(By.ID, "submit")
        submit_button.click()
        sleep(2)
        driver.get("http://127.0.0.1:8000/admin1/addservice1")
        sleep(4)
        service_name_input = driver.find_element(By.ID, "txt1")
        service_name_input.send_keys("Plumbing Service")
        sleep(2)
        description_name_input = driver.find_element(By.ID, "txt2")
        description_name_input.send_keys("At Expert Homecare, our expert plumbers offer fast, reliable solutions for all plumbing issues. ")
        sleep(2)
        upload_input = driver.find_element(By.ID, "inputfileupload")
        upload_input.send_keys("D:/mathew/miniproject-s9/uploads/1.jpg")
        sleep(2)
        submit_button = driver.find_element(By.ID, "register")
        submit_button.click()
        sleep(3)
        if driver.current_url == "http://127.0.0.1:8000/admin1/displayserviceadmin/":
            print("Place Added successful!")
        else:
            print("Place Added Failed!")
    def tearDown(self):
        self.driver.quit()
if __name__ == '__main__':
    unittest.main()
```

Screenshot

```
=============== RESTART: C:\Users\Mathew\Downloads\addemployee.py ==
************************************************
*****************Place TEST*********************
************************************************
Place Added successful!
```

Test report

Test Case 3

Project Name: Expert Homecare					
Add Tourist Place Test Case					
Test Case ID: Test_4			**Test Designed By: Mathew Peter**		
Test Priority (Low/Medium/High):			**Test Designed Date: 20/10/2024**		
Module Name: Service			**Test Executed By: Nimmy Francis**		
Test Title: Add Service			**Test Execution Date: 20/10/2024**		
Description: Add Service					
Pre-Condition: All field should be filled					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigate to the login page	http://127.0.0.1:8000/login1	Login page should be displayed	Login page was loaded	Pass
2	Enter email	“mathew1@gmail.com”	The email field should be filled with “mathew1@gmail.com”	email entered	Pass
3	Enter password	“Mathew1$2002”	The password field should be filled with “Mathew1$2002”	Password entered	Pass
4	Click on the submit button	N/A	If login is successful, admin should be redirected to admin	Admin was redirected to admin dashboard page	Pass

			dashboard page		
5	Navigate to the Add Service Page	http://127.0.0.1:8000/admin1/addservice1	Add tourist page should be displayed	Add Service page was loaded	Pass
6	Enter Service	Plumbing Service	The service field should be filled with “Plumbing Service”	Service entered	Pass
7	Enter Description	“At Expert Homecare, our expert plumbers offer fast, reliable solutions for all plumbing issues.”	The Description field should be filled with a description about the plumbing service	Description Entered	Pass
8	Enter image	“2.jpg”	The image field should be filled with “2.jpg”	Image inserted	Pass
9	Click on the Add button	N/A	If service is added successful, then admin should be redirected to admin dashboard page	Admin was redirected to admin dashboard page	Pass
Post-Condition: Place and image is successfully added					

Test Case 4:

Code

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from time import sleep
int()
driver = webdriver.Chrome()
driver.get("http://127.0.0.1:8000/login1/")
username_input = driver.find_element (By.ID, "email")
username_input.send_keys("mail2mathewpeter2001@gmail.com")
sleep(2)
password_input = driver.find_element(By.ID, "password")
password_input.send_keys("Mathew@2001")
submit_button = driver.find_element (By.ID, "submit")
submit_button.click()
sleep(2)
driver.get("http://127.0.0.1:8000/updatepassword")
sleep(4)
username_input = driver.find_element(By.ID, "password")
username_input.send_keys("Mathew")
sleep(2)
username_input = driver.find_element(By.ID, "cpassword")
username_input.send_keys("Mathew")
sleep(2)
submit_button = driver.find_element (By.ID, "register")
submit_button.click()
sleep(3)
element = driver.find_element(By.ID, "txt1")
value = element.get_attribute("Mathew")
print(value)
sleep(3)
print("*****************************************************")
print("*************PROFILE TEST*****************************")
print("*****************************************************")
if value == "Mathew" :
    print("Tested successfully")
else:
  print("Testing failed")
```

Screenshot

Test report

Test Case 4

Project Name: Expert Homecare					
Profile Test Case					
Test Case ID: Test_4			**Test Designed By: Mathew Peter**		
Test Priority (Low/Medium/High):			**Test Designed Date: 20/10/2024**		
Module Name: Update Profile Password			**Test Executed By: Nimmy Francis**		
Test Title: Update Profile Password			**Test Execution Date: 20/10/2024**		
Description: Update Profile Password					
Pre-Condition: All field should be filled					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigate to the login page	http://127.0.0.1:8000/login1/	Login page should be displayed	Login page was loaded	Pass
2	Enter email	“mathew1@gmail.com”	The email field should be filled with “mathew1@gmail.com”	email entered	Pass
3	Enter password	“Mathew1$2002”	The password field should be filled with “Mathew1$2002”	Password entered	Pass

4	Click on the submit button	N/A	If login is successful, user should be redirected to admin user login page	User was redirected to user login page	Pass
5	Navigate to the user profile page	http://127.0.0.1:8000/updatepassword	User profile page should be displayed	User profile page was loaded	Pass
6	Enter password	“mathew233”	The Password field should be filled with “mathew233”	Password entered	Pass
7	Enter confirm password	“mathew233”	The confirm Password field should be filled with “mathew233”	Confirm password entered	Pass
8	Click on the update button	N/A	If user profile page is updated successful, then user should be redirected to user login page	User was redirected to user profile page	Pass
Post-Condition: Password has been successfully updated					

CHAPTER 6

IMPLEMENTATION

6.1INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system. Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process will not take place

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database, and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

System maintenance is an effective component such that System maintenance refers to the ongoing activities required to ensure that a system or application operates effectively and efficiently after it has been implemented. It involves regular updates, bug fixes, and performance optimizations to keep the system running smoothly and securely.

System maintenance is essential to ensure that a system remains operational and effective after implementation. By establishing maintenance procedures and following them consistently, project teams can ensure that the system operates smoothly, remains secure, and continues to meet the needs of the end-users.

6.2.4 Hosting

Hosting is the service that makes your website or app accessible on the internet by storing its data on a server, which is essentially a powerful computer managed by a hosting provider. When you choose a hosting plan, you're renting space on a server so users can reach your content at any time by simply typing in your site's address. Hosting providers offer various types of plans to suit different needs. In shared hosting, for instance, multiple websites share one server, making it affordable but possibly slower during peak times. VPS (Virtual Private Server) hosting offers a middle-ground, where a single server is divided into virtual environments, each with dedicated resources for better performance. For sites with high traffic, dedicated hosting allocates an entire server to one client, offering maximum control and resources.

Eg. Render

Render hosting is ideal for teams who want to focus on building and deploying their applications rather than managing backend infrastructure. With Render, deployments are automated and can be easily connected to Git repositories, enabling a seamless workflow where code changes can be instantly reflected in production. It offers serverless scalability, so applications automatically adjust to handle increases in traffic, making it highly adaptable for businesses experiencing growth or fluctuations in user demand. Render supports a wide variety of tech stacks, including languages and frameworks like Node.js, Python, Go, and Ruby, making it versatile for diverse projects. Additionally, Render provides built-in monitoring and logging tools, allowing users to track app performance and troubleshoot issues quickly. For those needing database solutions, Render also offers fully managed PostgreSQL databases, eliminating the need for manual database setup and maintenance.

Procedure for hosting a website on Render

1. Sign Up on Render

- Go to Render.com and create an account. You can sign up with your GitHub or GitLab account, or create a new account with an email and password.

2. Create a New Web Service

- Once you're logged in, go to the Render dashboard.
- Click on "New +" in the top right corner and select "Web Service".

3. Connect Your Git Repository

- Choose the GitHub or GitLab repository that contains your website's code. Render will ask for permissions to access your repositories.
- Select the repository for your project and grant access.

4. Configure Deployment Settings

- Specify the branch you want to deploy (e.g., main or master).
- Choose your runtime environment (Node.js, Python, Ruby, etc.) if Render doesn't auto-detect it.
- Set the build and start commands if necessary. For example:
  - Build command: npm install for Node.js projects.
  - Start command: npm start for Node.js projects.

5. Select the Region and Plan

- Choose your deployment region (e.g., US or Europe) based on where most of your users are located.
- Select a pricing plan. Render offers a free tier for smaller projects and paid options for more resources.

6. Deploy Your Website

- Click "Create Web Service" to start the initial deployment.
- Render will automatically build and deploy your website. You can monitor the deployment status on the dashboard.

7. Configure Domain and SSL (Optional)

- After deployment, Render assigns a temporary URL to your website (e.g., your-app.onrender.com).
- To add a custom domain, go to your web service settings and add your domain name.
- Render provides a free SSL certificate that auto-renews, ensuring secure access over HTTPS.

8. Monitor and Manage Your Website

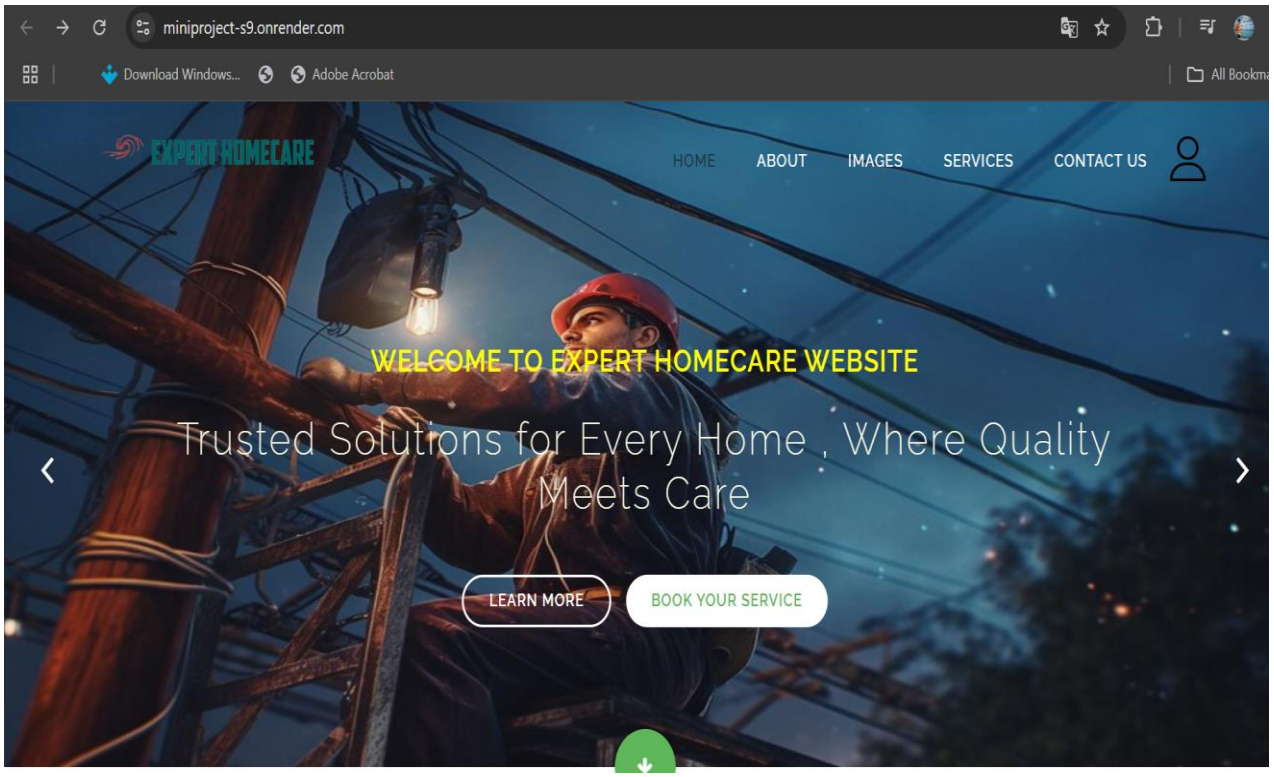
- ❖ Use Render's built-in logging and monitoring tools to track your website's performance.
- ❖ If you make changes to your code, just push them to the specified branch, and Render will automatically redeploy the updated version.

Hosted Website: Render

Hosted Link: https://miniproject-s9.onrender.com/

Hosted Link QR Code

Screenshot



CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Expert Homecare offers a reliable solution for booking essential home maintenance services, such as plumbing, electrical work, and gadget repairs, tailored specifically for the districts of Ernakulam and Kottayam. With a user-friendly platform, customers can easily connect with qualified professionals, view detailed profiles, schedule appointments, and track service progress in real-time. The platform also includes secure payment options, transparent invoicing, and a built-in messaging system. Additionally, user feedback is collected to ensure the smooth functioning of our website.

7.2 FUTURE SCOPE

The future scope of Expert Homecare is with plans to integrate an AI chatbot that will provide 24/7 support to handle common customer quires, guide users through booking processes, and recommend services based on user needs, making the platform highly accessible and responsive. Adding a voice typing feature will allow users to interact more naturally by speaking , improving accessibility for users with varied needs and enhancing convenience. A redesigned, visually appealing user interface will ensure smoother navigation and a more intuitive experience, enabling users to locate services, view professional profiles, and make appointments with minimal effort.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

[1] Beginning Software Engineering-Rod Stephens

[2] Reference Book 6: Pressman, R.S., "Software Engineering: A Practitioner's Approach", McGraw Hill SE ,7th Edition (2010)

[3] Woods, R. E., and González, R. C. (2002). processing digital images. Prentice Hall.

[4] Shen, H., and Zhang, Y. (2018). An overview of algorithms for image segmentation. 33(3), 477-495, Journal of Computer Science and Technology.

[5] In 1994, Lindeberg, T. A fundamental technique for examining image structures is scale-space theory. 213-241 in Journal of Applied Statistics, 21(1).Mallat, S. (1989). A theory for the multiresolution signal decomposition: The wavelet representation in the IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674-693.

[6] E. T. Jaynes (2003). Probability Theory: The Scientific Logic. Cambridge University Press.

[7] D. G. Lowe (2004). unique visual characteristics derived from scale-invariant keypoints. 60(2), 91-110, International Journal of Computer Vision.

[8] Tuytelaars, T., Van Gool, L., & Bay, H. (2006). Surf: Enhanced robustness and speed. Conference on Computer Vision in Europe, 404–417.

[9] Calonder, M., Fua, P., Lepetit, V., and Strecha, C. (2010). Brief: Robust independent elementary features that are binary. Conference on Computer Vision in Europe, 778-792.Rublee, E., Rabaud, V., Avidan, S., Blanc, M., & Mori, G. (2011). Orb: An efficient alternative to sift or surf. IEEE International Conference on Computer Vision, 2564-2571.

[10] Zhang, K., Ping Lu, C., Kermany, D. S., & Goldbaum, M. (2018). fundus photos and labelled optical coherence tomography (OCT) for ophthalmology deep learning. 1722–1734 in Ophthalmology, 125(12).

[11] Zhang, an L., Yang, M.-H., and Yang, J. (2015). An overview of deep learning for face detection and recognition. Image Understanding and Computer Vision, 137, 60–76.

[12] Zhang, Y., Li, X., and Wang, X. (2020). Digital image processing and machine learning for industrial flaw identification. Manufacturing Processes Journal, 56, 435–450.

[13] Warde-Farley, D., Xu, B., Goodfellow, I., Pouget-Abadie, J., Mirza, M., Ozair, S.,... & Bengio, Y. (2014). adversarial netts that are generative. Systems for Neural Information Processing, 2672–2680.

[14] Li, Z., Zhang, Y., and Zhang, Y. (2019). Problems and solutions in real-time image processing. 1059–1075 in Journal of Real-Time Image Processing, 16(4).LeCun, Y.,

Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[15] In 2019, Arute, F., Babbush, R., Bacon, D. J., Bardin, J. C., Barends, R., Babbush, Arya, K., & Zalcman, A. Using a programmable superconducting processor to achieve quantum supremacy. 505–510 in Nature, 574(7779).

[16] Horiuchi, T., and Indiveri, G. (2011). Frontier computing: An approach inspired by the brain. Neuroscience Frontiers, 5, 108.

[17] Liu, Z., Chen, Y., and Liu, S. (2017). Deep convolutional neural networks are used for denoising. IEEE Image Processing Transactions, 26(6), 2778-2790.

[18] Li, W., and Wang, X. (2018). learned how to use deep convolutional neural networks for image reduction. Image Processing Transactions, IEEE, 27(7), 3407-3418.

[19] Liu, Y., Huang, J., and Wang, L. (2019). edge detection with sophisticated techniques in noisy situations. Letters on Pattern Recognition, 120, 44–53.Zhou, Z., Liang, a L., & Xu, Q. (2020). Real-time object detection for autonomous vehicles using image processing. IEEE Transactions on Intelligent Transportation Systems, 21(2), 547-559.

[20] Gao, X., Chen, H., & Xu, Y. (2021). generative models for high-resolution images. IEEE Transactions on Machine Intelligence and Pattern Analysis, 43(8), 2800-2812.

[21] Paindavoine, M., Sun, S.a., and Pang, S. (2006). "Face recognition for the Gabor wavelets and adaptive boosting." 2360–2364 in Neurocomputing, 69(16–18).

[22] Brox, T., Fischer, P., and Ronneberger, O. (2015). "U-Net: Convolutional networks for biomedical image segmentation." Computer-Assisted Intervention and Medical Image Computing (MICCAI), 234–241.

[23] Girshick, R., Redmon, J., Divvala, S., & Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." Conference on Pattern Recognition and Computer Vision (CVPR), IEEE, 779-788.

[24] Viola, P., & Jones, M. (2001). Rapid object detection in a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1, I-511-I-518. doi:10.1109/CVPR.2001.99051

[25] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503. doi:10.1109/LSP.2016.2603342.

[26] Sun, J., Girshick, R., He, K., and Ren, S. (2015). Faster R-CNN: Using region proposal networks to detect objects in real time. Neural Information Processing Systems: Advances, 28, 91-99.

[27] Girshick, R., Redmon, J., Divvala, S., & Farhadi, A. (2016). You just glance once:

Real-time, unified object detection. doi:10.1109/CVPR.2016.91 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

[28] Jia, X., Wang, H., and Wang, a Y. (2020). Region ideas for adaptive multi-face detection in real-time applications. 135, 238-245. doi:10.1016/j.patrec.2020.05.014. Pattern Recognition Letters.

[29] Wu, X., Wang, X., and Zhang, Y. (2021). Multiple face detection in cluttered scenarios using a hybrid deep learning model. doi:10.1016/j.jvcir.2021.103251 Journal of Visual Communication and Image Representation, 80, 103251.

WEBSITES:

- https://www.nicesnippets.com/blog/how-to-forgot-password-send-mail-django
- https://www.gooezy.com/plumber
- https://www.taskmario.com/kochi/electrician-164

CHAPTER 9
APPENDIX

9.1 Sample Code

Login Page

```
<!DOCTYPE html>
{% load static %}

<html lang="en">
<head>
   <title>Login 05</title>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">


   <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap" rel="stylesheet">
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
   <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Expires" content="0">
   <link rel="stylesheet" href="{% static 'css/style1.css' %}">
</head>
<script>
   document.addEventListener("DOMContentLoaded", function() {
      // Function to prevent back navigation
      function preventBack() {
         history.pushState(null, null, location.href);
         window.onpopstate = function () {
            history.go(1);
         };
      }

      // Check if the loggedOut flag is set
      if (document.cookie.includes("loggedOut=true")) {
         preventBack(); // Prevent back navigation
         // Optionally, remove the cookie to prevent future issues
         document.cookie = "loggedOut=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";
      } else {
```

```
            preventBack(); // Ensure the user can't go back to this page
        }
    });
</script>
<body>
    <section class="ftco-section">
        <div style="margin-top:-100px">
            <div class="container">
                <div class="row justify-content-center">
                    <div class="col-md-6 text-center mb-5">
                    </div>
                </div>
                <div class="login-wrap p-4 p-lg-5">
                    <div class="d-flex">
                        <div class="w-100">
                            <h3 class="mb-4">Sign In</h3>
                        </div>

                        <div class="w-100">
                            <p class="social-media d-flex justify-content-end">

                                <a href="#" class="social-icon d-flex align-items-center
justify-content-center"><span class="fa fa-facebook"></span></a>
                                <a href="#" class="social-icon d-flex align-items-center
justify-content-center"><span class="fa fa-twitter"></span></a>

                                <!-- Replace 'social:begin' with the correct URL name for your
Google OAuth2 provider -->
                            </p>
                        </div>
                    </div>
                    {% if messages %}
    {% for message in messages %}
    <div class="alert alert-success" role="alert" id="delete-success-message" >
            {{ message }}
        </div>
    {% endfor %}
{% endif %}
 <form action="{% url 'logincustomer' %}" method="post" class="signin-form">
```

```
    {% csrf_token %}
 <div class="form-group mb-3">
   <label class="label" for="name1">Email</label>
<input type="email" class="form-control" placeholder="email" name="email"
required>
</div>
                <div class="form-group mb-3">
                   <label class="label" for="password">Password</label>
                   <input type="password" class="form-control"
placeholder="Password" name="password" required>
                </div>
                <div class="form-group">
                   <button type="submit" id="submit" class="form-control btn btn-
primary submit px-3">Sign In</button>
                </div>
                <div class="form-group d-md-flex">
                   <div class="w-50 text-left">
                      <label class="checkbox-wrap checkbox-primary mb-0">Remember
me
                         <input type="checkbox" checked>
                         <span class="checkmark"></span>
                      </label>
                   </div>
                   <div class="w-50 text-md-right">
                      <a href="{% url 'forgot' %}">Forgot Password</a>
                   </div>
                </div>


             </form>
             <div class="google-signin-box">
                <a href="{% url 'social:begin' 'google-oauth2' %}" class="social-icon
d-flex align-items-center justify-content-center">
                   <span class="fa fa-google"></span> Sign in with Google
                </a>
             </div>


             <a href="{% url 'register' %}"><font color="orange">New user? Register
here</font></a>
          </div>
       </div>
```

```
            </div>
        </section>
        <script>
             document.addEventListener("DOMContentLoaded", function() {
            var messageElement = document.getElementById("delete-success-message");
            if (messageElement) {
                // alert(messageElement.innerText);  // This will show a browser alert with the
message

                // Optional: Hide the message after 5 seconds
                setTimeout(function() {
                    messageElement.style.display = 'none';
                }, 3000);
            }
        });
         </script>
        </script>
        <script src="{% static 'js/jquery.min.js' %}"></script>
        <script src="{% static 'js/popper.js' %}"></script>
        <script src="{% static 'js/bootstrap.min.js' %}"></script>
        <script src="{% static 'js/main.js' %}"></script>
        <nav1>
            <div style="margin-left:60px;margin-top:5px"><i><font size="5"
color="orange"><img src ="{% static 'images/1.png' %}"
width="250"></font></i></div>
            <div class="w-100">
                <div style="margin-top:-60px;margin-left:-60px;">
                    <p class="social-media d-flex justify-content-end">
                        <a href="#" class="social-icon d-flex align-items-center justify-content-
center"><span class="fa fa-instagram"></span></a>
                        <a href="#" class="social-icon d-flex align-items-center justify-content-
center"><span class="fa fa-facebook"></span></a>
                        <a href="#" class="social-icon d-flex align-items-center justify-content-
center"><span class="fa fa-twitter"></span></a>
                    </p>
                </div>
            </div>
        </nav1>
    </body>
</html>
```

9.2 Screen Shots

Registration Form

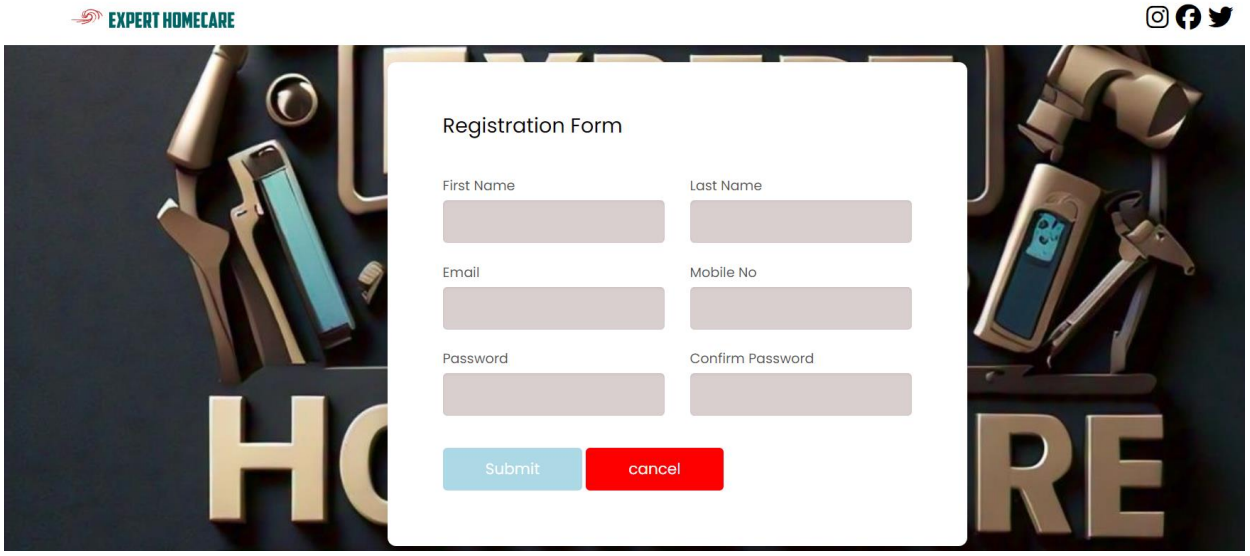


Figure 1:Registraion Form

Login Page

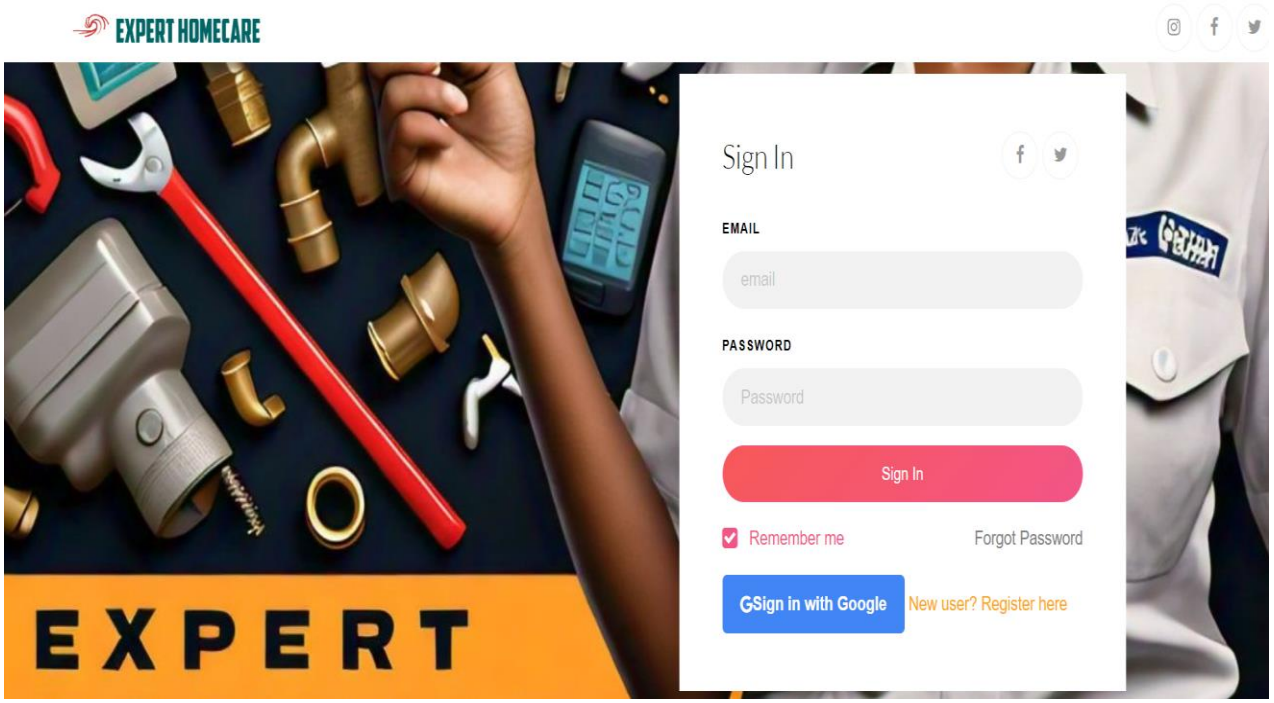
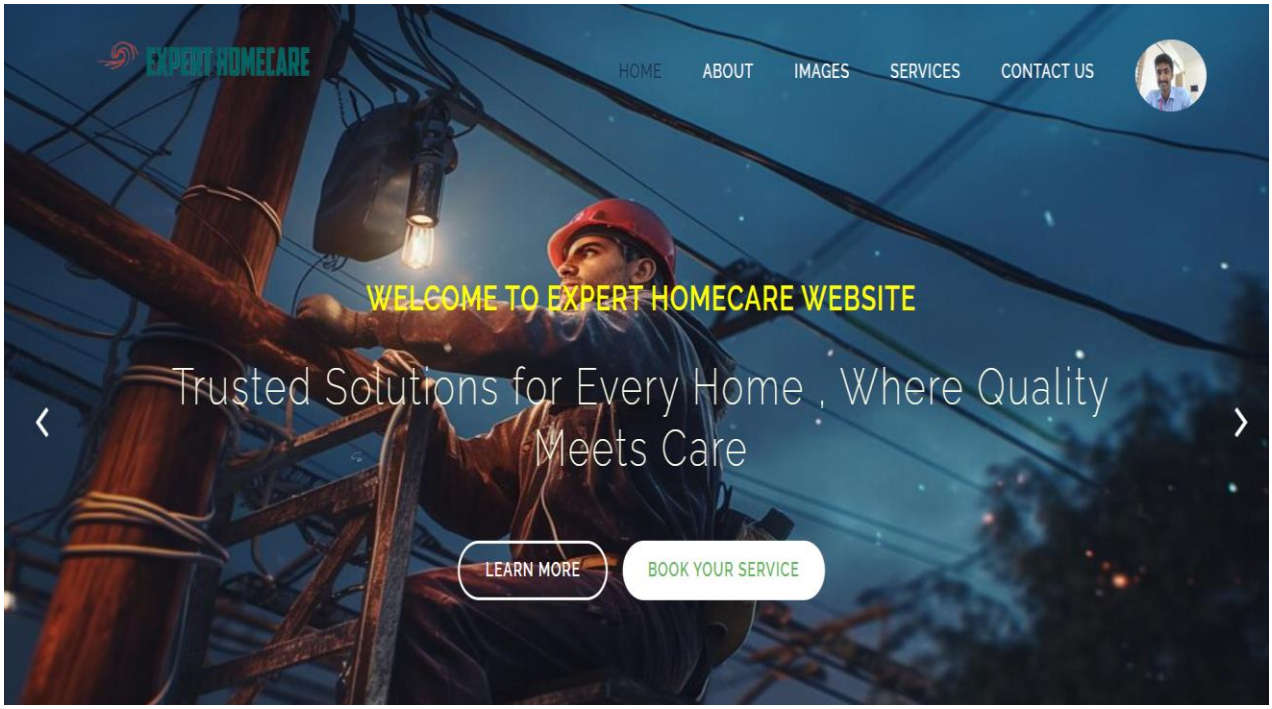


Figure 2:Login Page

Customer Home page



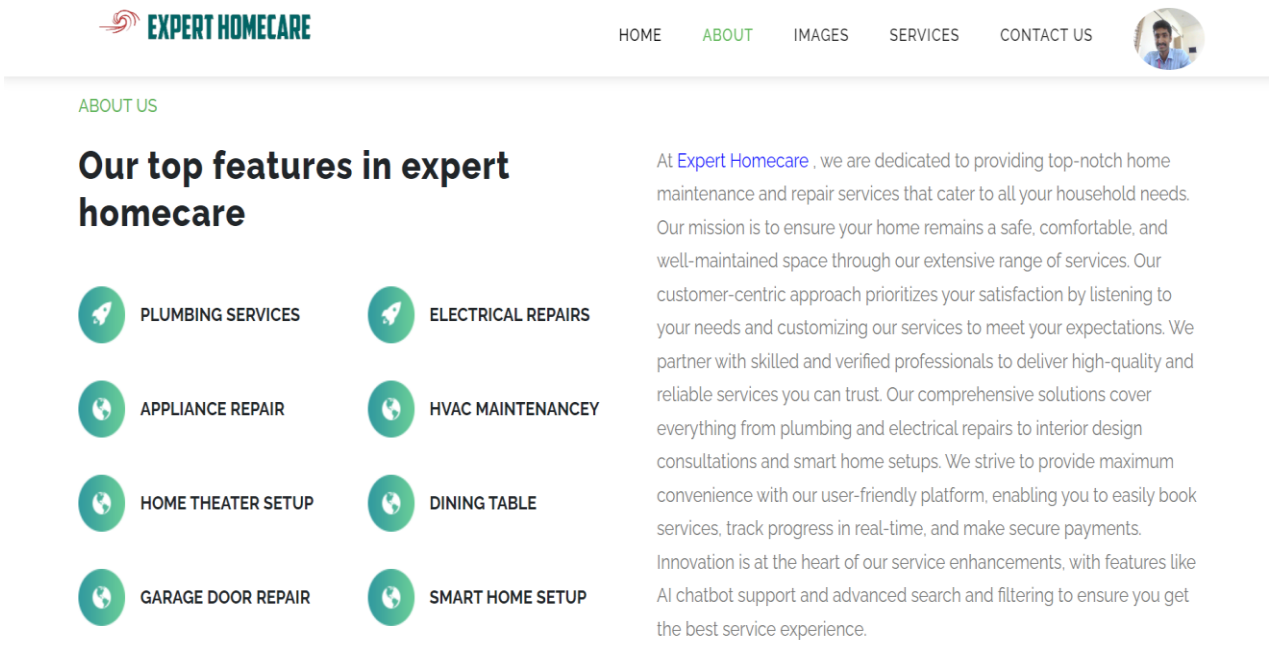


Figure 3: Home page

Service Page

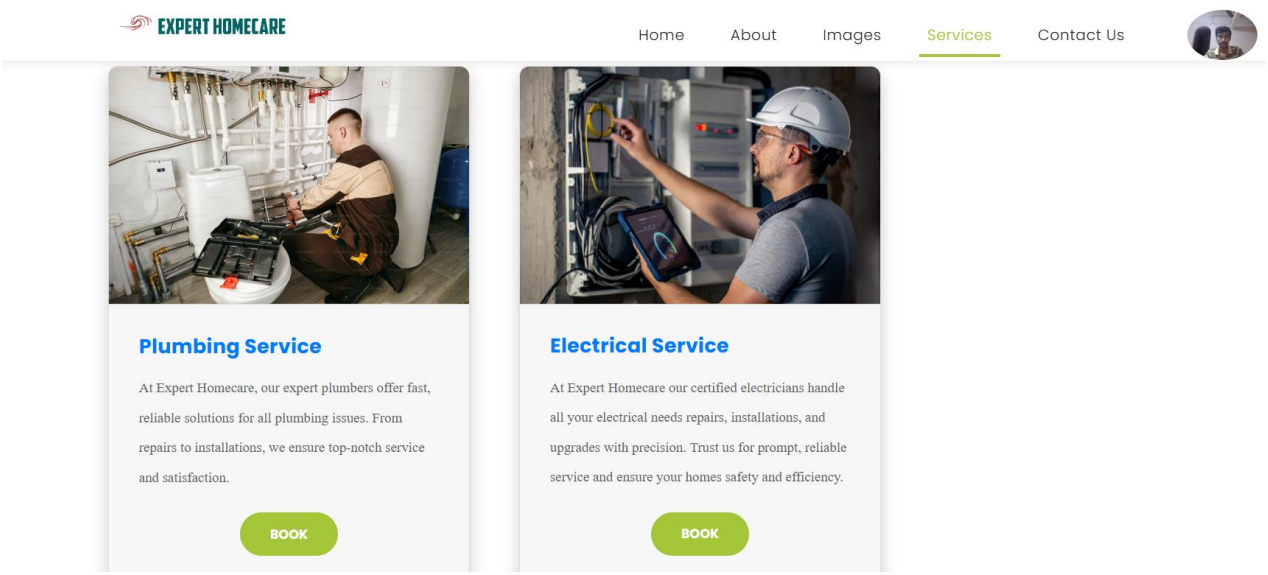


Figure 4:Service Page

Service Provider Table

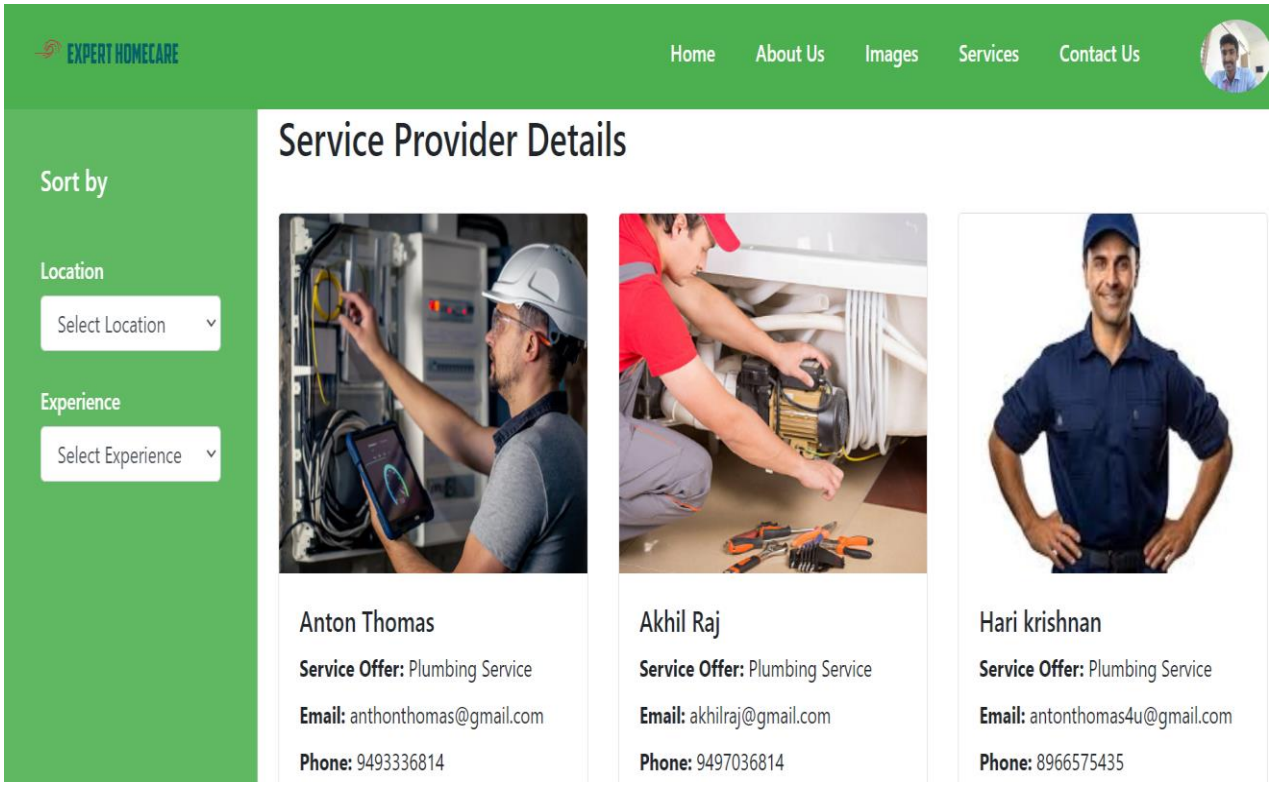


Figure 5: Service Provider Table

Admin Dashboard

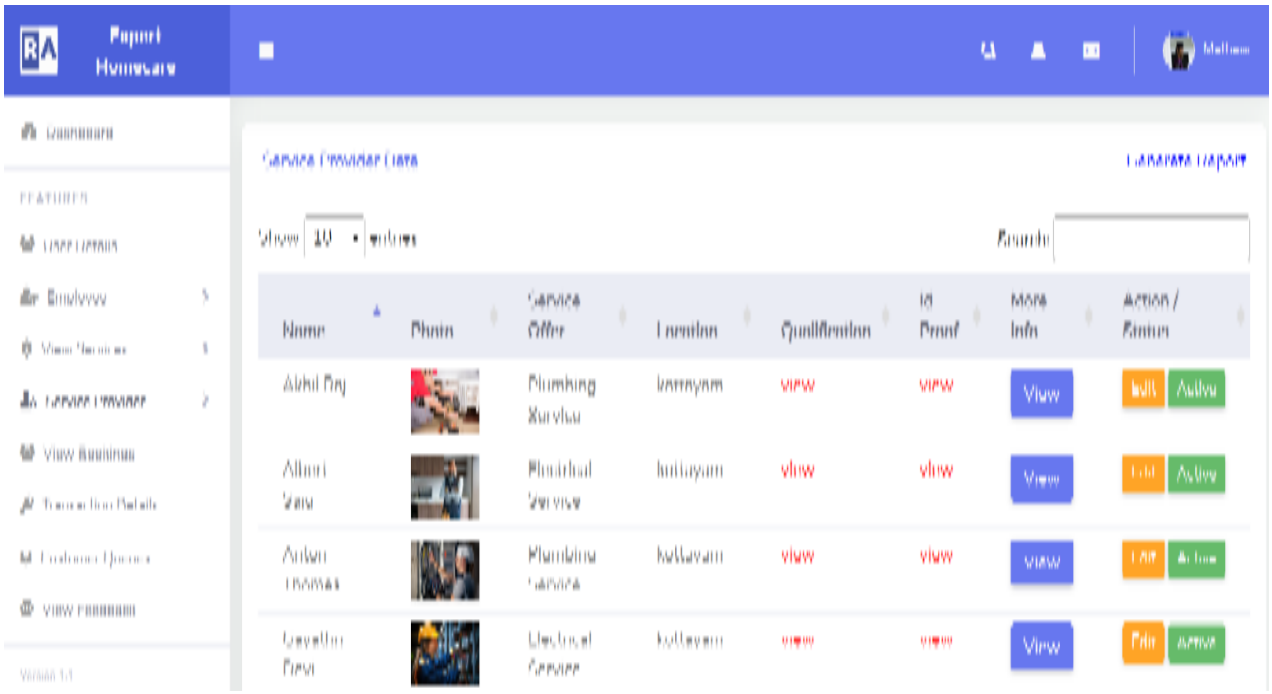


Figure 6: Admin Dashboard

Employee Dashboard

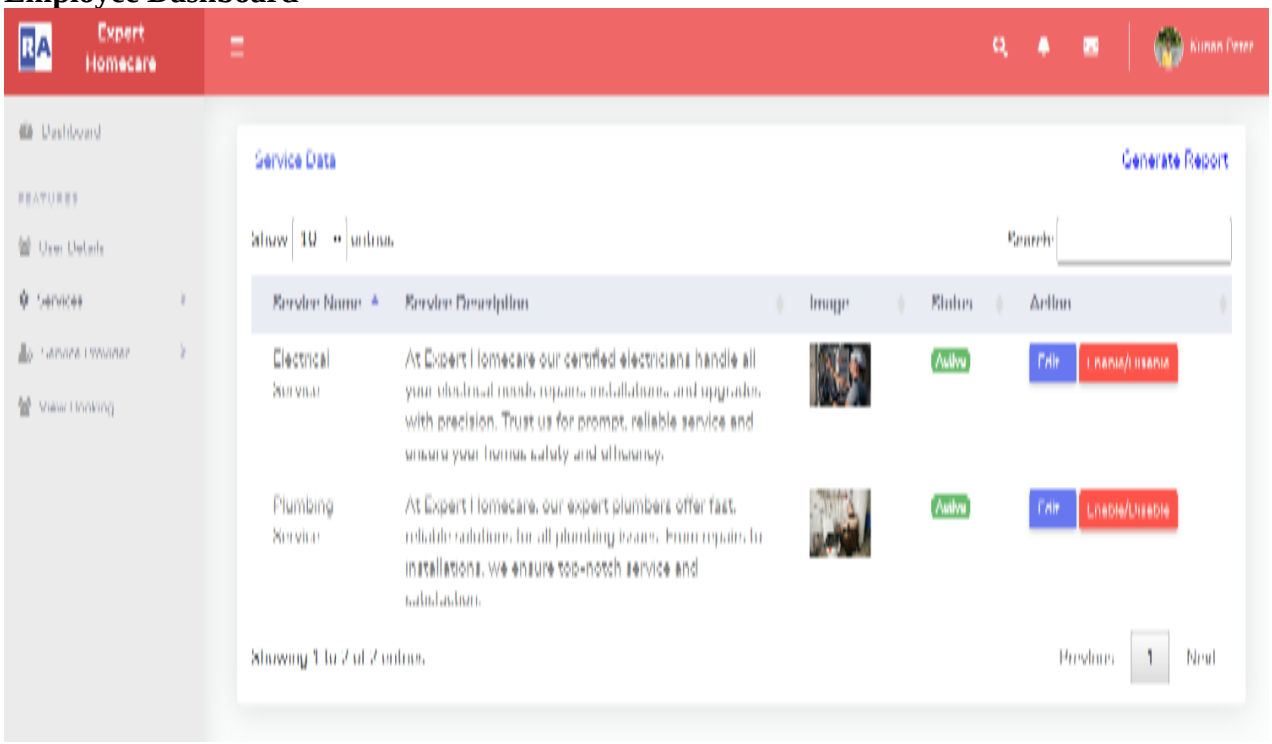


Figure 7:Employee Dashboard

Service Provide Dashboard

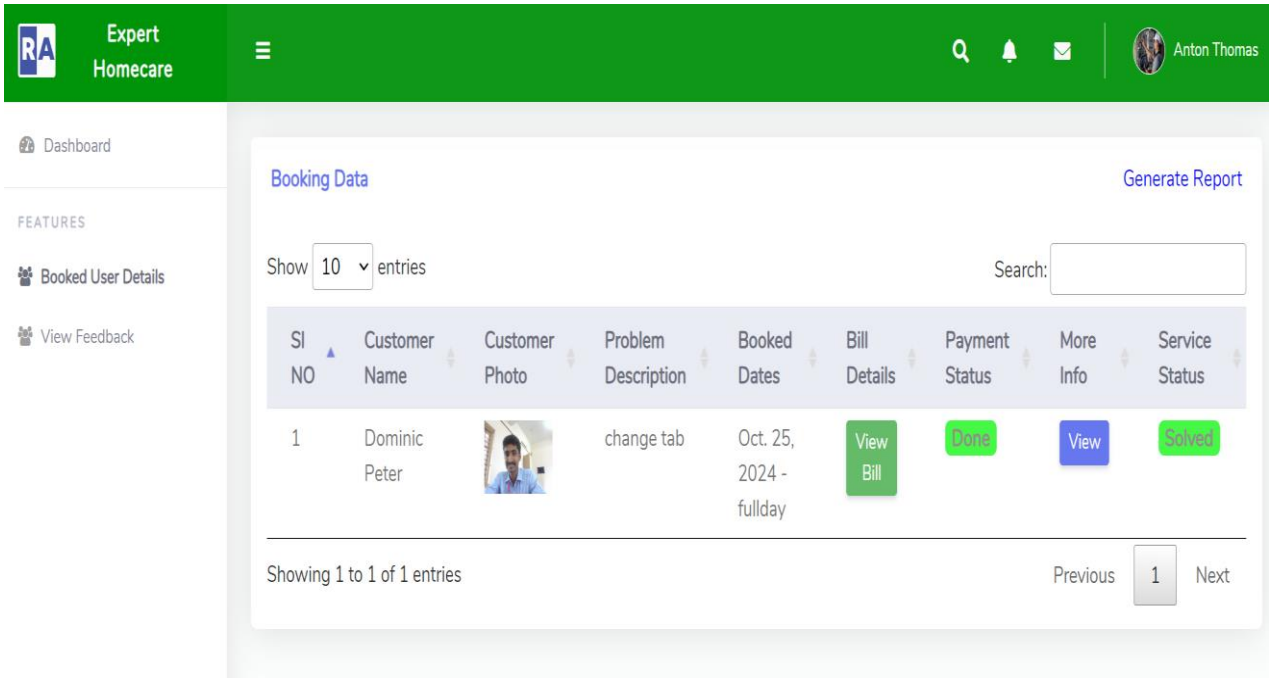


Figure 8: Service Provider Dashboard

Chat Person

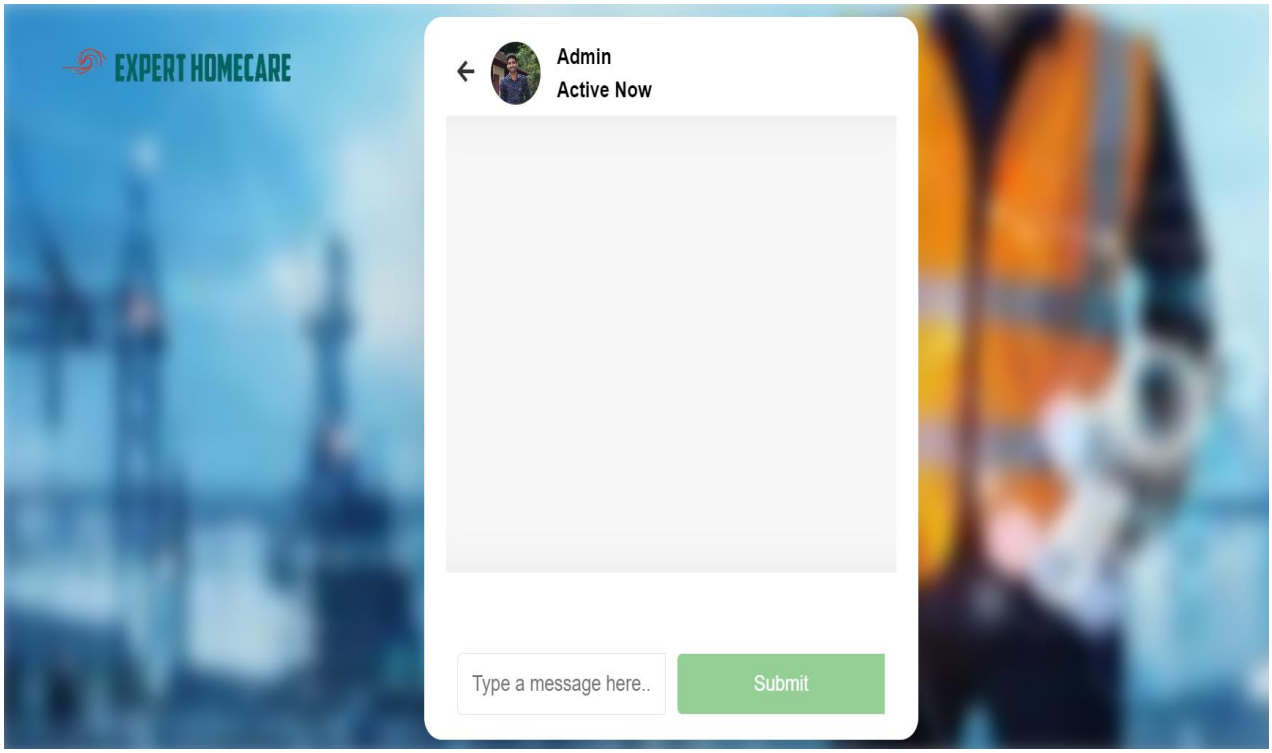
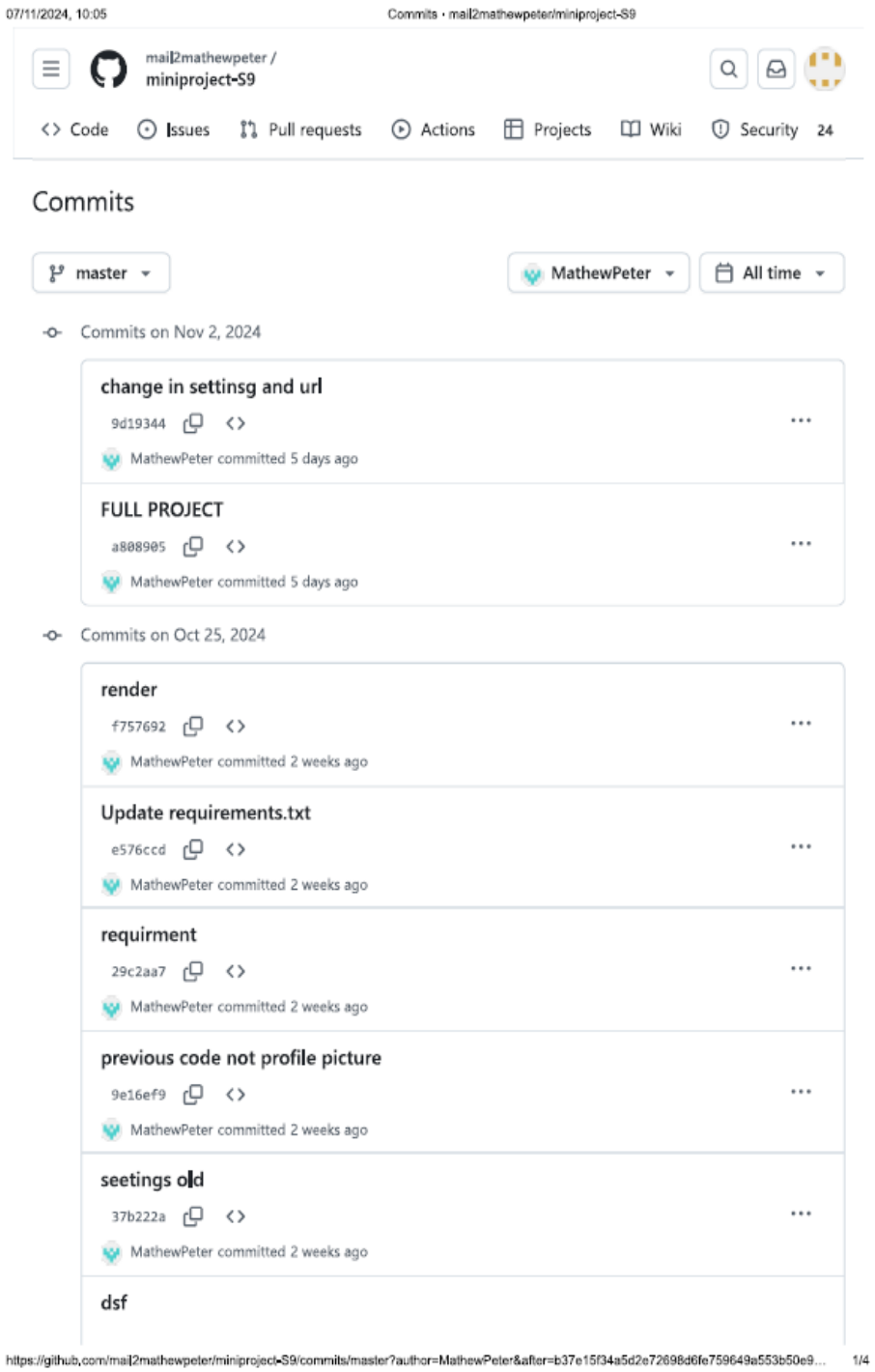
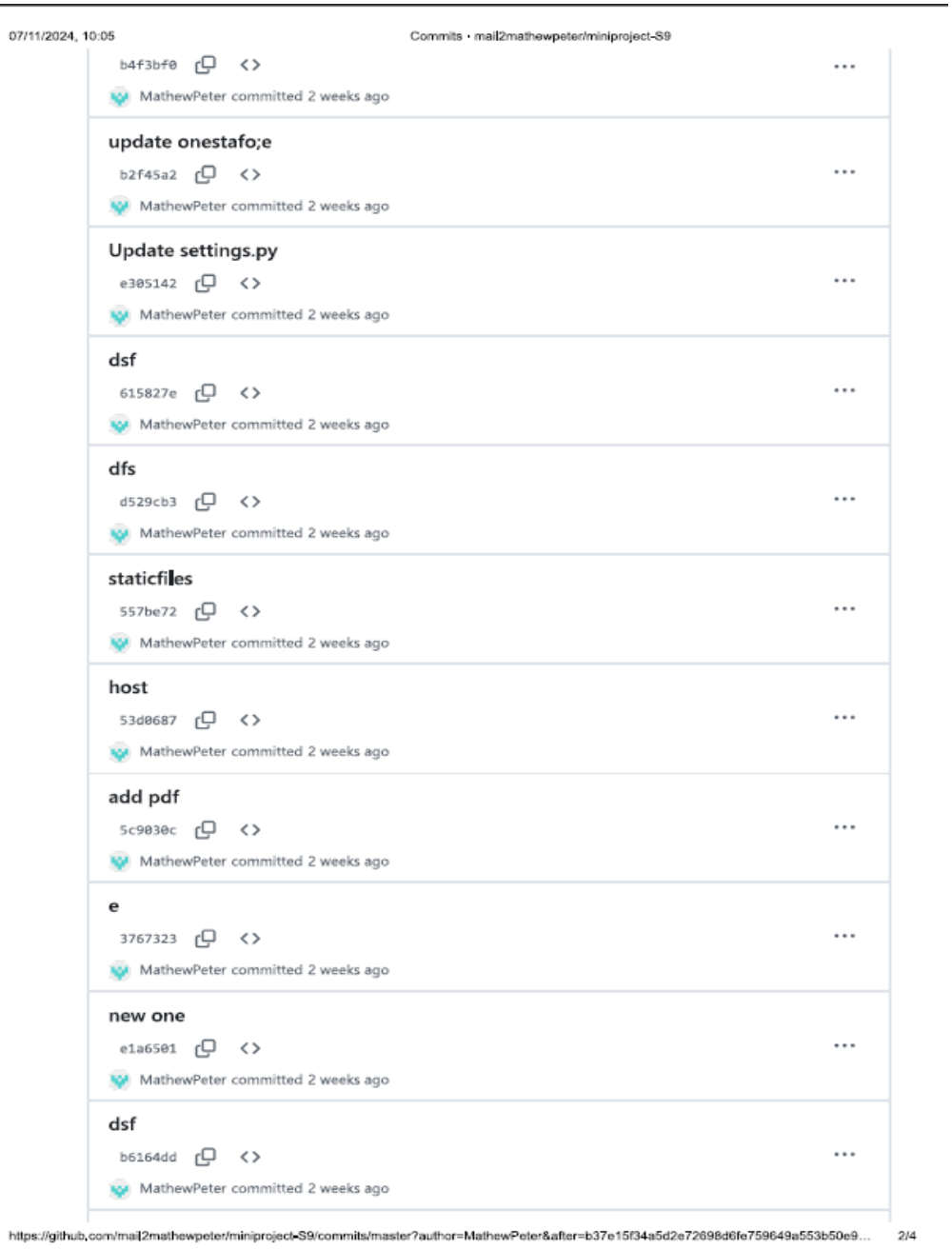


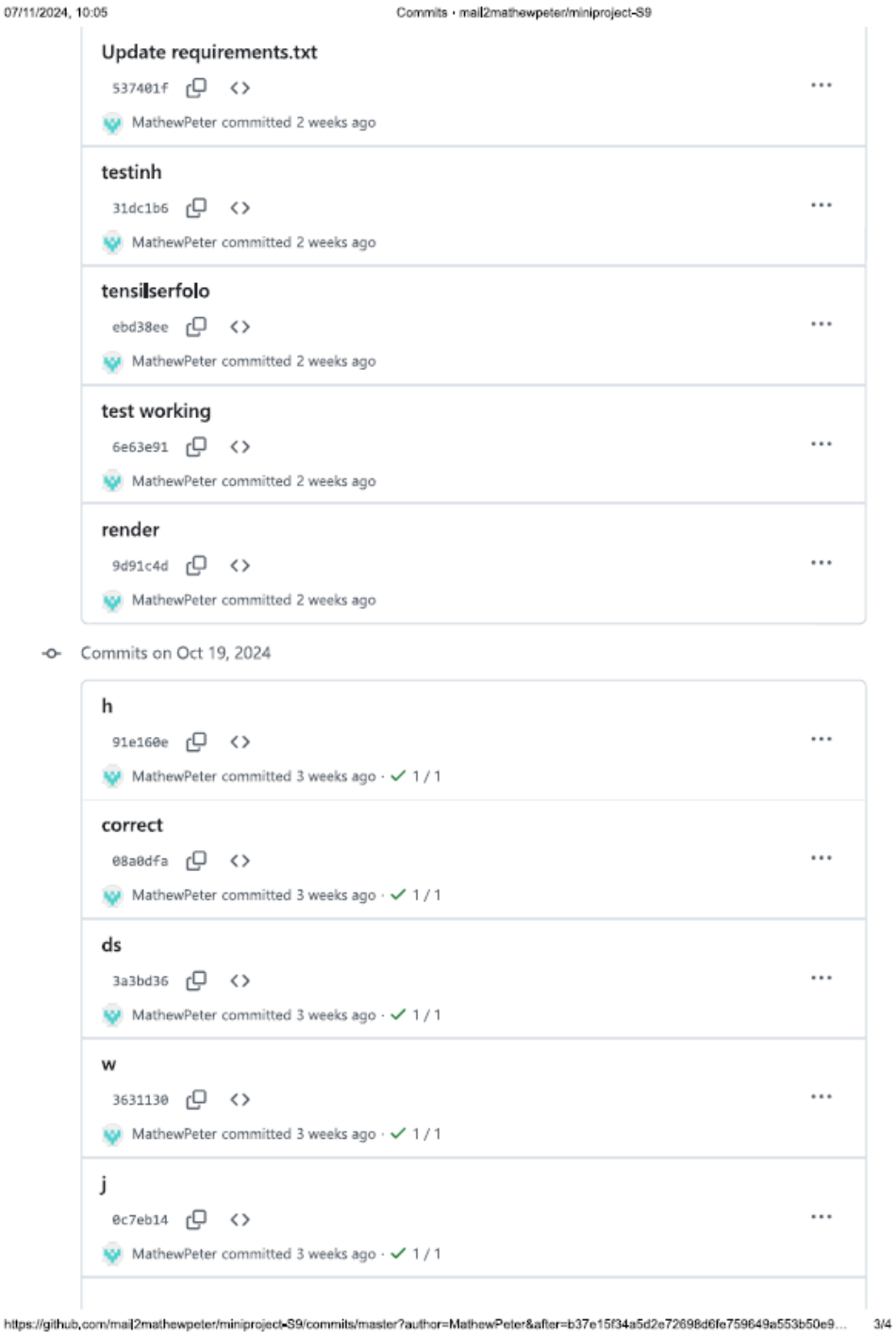
Figure 9: Chat Person

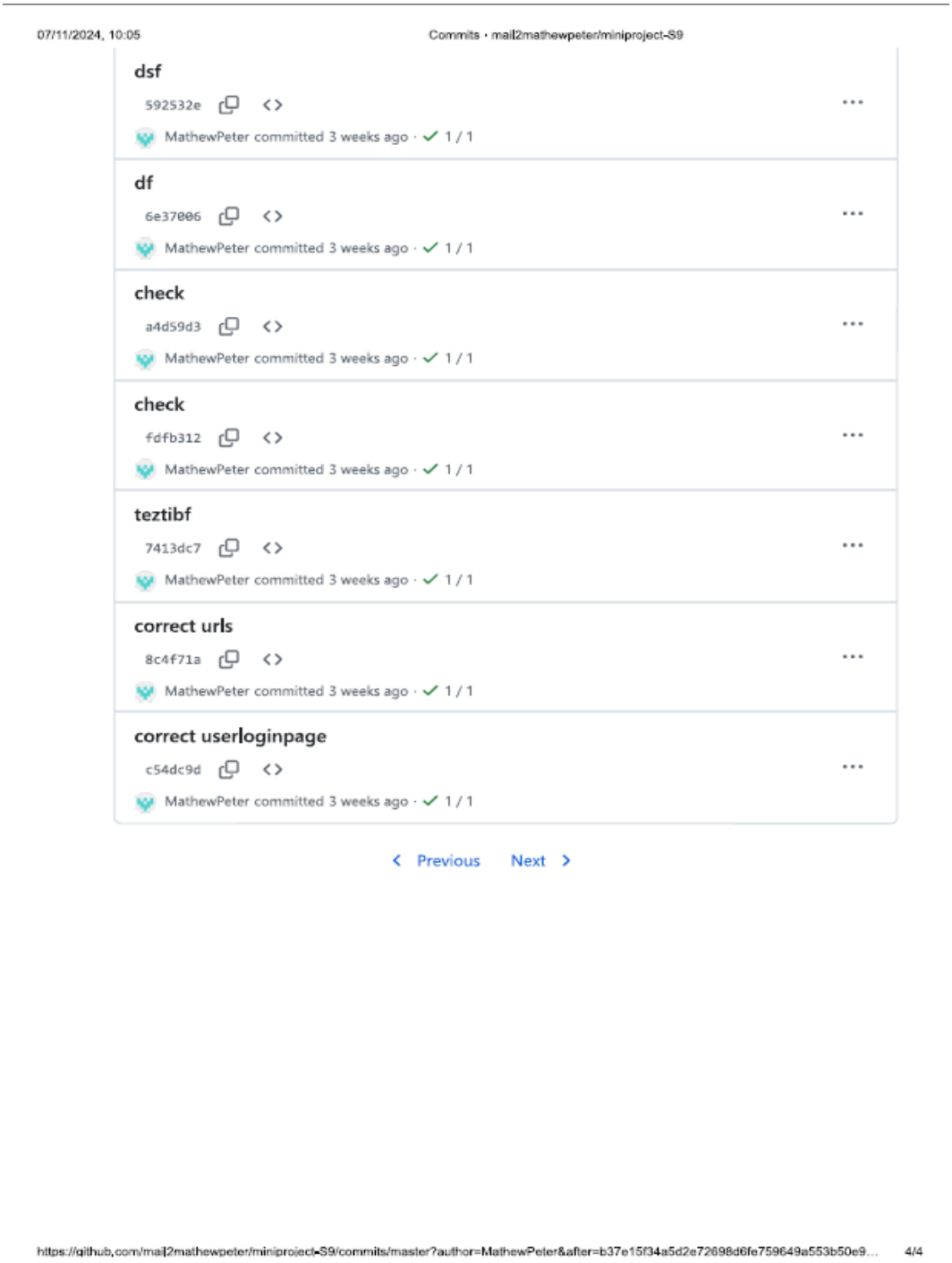
9.3 GIT LOG

Git log is enclosed herewith for verification of the project's software development lifecycle:




07/11/2024, 10:05
Commits · mail2mathewpeter/miniproject-S9
b4f3bf0
MathewPeter committed 2 weeks ago
update onestafo;e
b2f45a2
MathewPeter committed 2 weeks ago
Update settings.py
e305142
MathewPeter committed 2 weeks ago
dsf
615827e
MathewPeter committed 2 weeks ago
dfs
d529cb3
MathewPeter committed 2 weeks ago
staticfiles
557be72
MathewPeter committed 2 weeks ago
host
53d0687
MathewPeter committed 2 weeks ago
add pdf
5c9030c
MathewPeter committed 2 weeks ago
e
3767323
MathewPeter committed 2 weeks ago
new one
e1a6501
MathewPeter committed 2 weeks ago
dsf
b6164dd
MathewPeter committed 2 weeks ago
https://github.com/mail2mathewpeter/miniproject-S9/commits/master?author=MathewPeter&after=b37e15f34a5d2e72698d6fe759649a553b50e9...
2/4


07/11/2024, 10:05
Commits · mail2mathewpeter/miniproject-S9
Update requirements.txt
537401f
MathewPeter committed 2 weeks ago
testinh
31dc1b6
MathewPeter committed 2 weeks ago
tensilserfolo
ebd38ee
MathewPeter committed 2 weeks ago
test working
6e63e91
MathewPeter committed 2 weeks ago
render
9d91c4d
MathewPeter committed 2 weeks ago
Commits on Oct 19, 2024
h
91e160e
MathewPeter committed 3 weeks ago · 1 / 1
correct
08a0dfa
MathewPeter committed 3 weeks ago · 1 / 1
ds
3a3bd36
MathewPeter committed 3 weeks ago · 1 / 1
w
3631130
MathewPeter committed 3 weeks ago · 1 / 1
j
0c7eb14
MathewPeter committed 3 weeks ago · 1 / 1
https://github.com/mail2mathewpeter/miniproject-S9/commits/master?author=MathewPeter&after=b37e15f34a5d2e72698d6fe759649a553b50e9...
3/4


07/11/2024, 10:05
Commits · mail2mathewpeter/miniproject-S9
dsf
592532e
MathewPeter committed 3 weeks ago · 1 / 1
df
6e37006
MathewPeter committed 3 weeks ago · 1 / 1
check
a4d59d3
MathewPeter committed 3 weeks ago · 1 / 1
check
fdfb312
MathewPeter committed 3 weeks ago · 1 / 1
teztibf
7413dc7
MathewPeter committed 3 weeks ago · 1 / 1
correct urls
8c4f71a
MathewPeter committed 3 weeks ago · 1 / 1
correct userloginpage
c54dc9d
MathewPeter committed 3 weeks ago · 1 / 1
Previous
Next
https://github.com/mail2mathewpeter/miniproject-S9/commits/master?author=MathewPeter&after=b37e15f34a5d2e72698d6fe759649a553b50e9...
4/4