

SELF-EMPLOYMENT SERVICE BOOKING SYSTEM

*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

S HAMEED BASHA (21UECS0563) **(VTU19266)**
D HEMANTH KUMAR REDDY (21UECS0149) **(VTU19294)**
N VENKAT (21UECS0402) **(VTU19975)**

*Under the guidance of
Dr. R. ANANDH, M.E., Ph.D.,
ASSISTANT PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

SELF-EMPLOYMENT SERVICE BOOKING SYSTEM

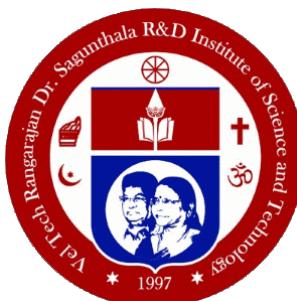
*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

S HAMEED BASHA (21UECS0563) **(VTU19266)**
D HEMANTH KUMAR REDDY (21UECS0149) **(VTU19294)**
N VENKAT (21UECS0402) **(VTU19975)**

*Under the guidance of
Dr. R. ANANDH, M.E., Ph.D.,
ASSISTANT PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "SELF-EMPLOYMENT SERVICE BOOKING SYSTEM" by " S HAMEED BASHA (21UECS0563), D HEMANTH KUMAR REEDY (21UECS0149), N VENKAT (21UECS0402)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

Dr. R. Anandh, M.E., Ph.D.,

Assistant Professor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

Signature of Professor In-charge

Dr. Peter Soosai Anandaraj A, M.E., Ph.D.,

Associate Professor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

S HAMEED BASHA

Date: / /

D HEMANTH KUMAR REDDY

Date: / /

N VENKAT

Date: / /

APPROVAL SHEET

This project report entitled " SELF-EMPLOYMENT SERVICE BOOKING SYSTEM " by S HAMEED BASHA (21UECS0563), D HEMANTH KUMAR REDDY (21UECS0149), N VENKAT (21UECS0402) is approved for the degree of B.Tech in Computer Science Engineering.

Examiners

Supervisor

Dr. R. ANANDH, M.E., Ph.D.,
ASSISTANT PROFESSOR

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN**, for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.**, for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.**, for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Dr. R. ANANDH, M.E., Ph.D.**, for his cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U. HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.**, for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

S HAMEED BASHA	(21UECS0563)
D HEMAANTH KUMAR REDDY	(21UECS0149)
N VENKAT	(21UECS0402)

ABSTRACT

Self-employment is the state of working for one self rather than employee. SelfEmployee Service Booking is an online market place that helps connect service providers with the service seekers. Being an online portal, users can effectively book everyday services on the portal. Customer just need to search through the website, filter their required services and then schedule a service as per their own convenience. It is an end to your everyday problems. Services are aimed at making several day-to-day tasks conveniently actionable and accessible to the Customers. Whether you want a beautician, a plumber, a carpenter, an electrician, a technician etc. is studded with self-employees hailing in all. The employee has a chance to choose his working hours. According to their working hours they will be paid. A single employee can register for multiple services and provide services for the customers. It aspires to make hiring a service employee as easy and straight forward as e-commerce companies have made buying products. For this project PHP,HTML,CSS,JS and MYSQL used to create this application. The system is designed to make it easy for customers to browse and select services, book appointments, and make payments securely. Customers can create profiles, view service provider ratings and reviews, and select their preferred service provider based on availability, price, and location. Service providers will have their own profiles, which will include their services, pricing, and availability. They will be able to manage their schedules, receive bookings and payments, and communicate with customers through the platform. Thus, the purpose of the site is to save the effort and time of both Customer and service provider.

Keywords: **Self-employment, Online marketplace, Customers, Service Providers, Service seekers, Convenience, Profiles, Location, Payments, E-commerce, Booking Appointments.**

LIST OF FIGURES

4.1	Architecture Diagram of Self-Employment Service Booking System	12
4.2	Use Case Diagram of Self-Employment Service Booking System	13
4.3	Class Diagram of Self-Employment Service Booking System	14
4.4	Sequence Diagram of Self-Employment Service Booking System	15
4.5	Activity Diagram of Self-Employment Service Booking System	16
4.6	Adimn module of self-Employment Service Booking System	17
4.7	User module of self-Employment Service Booking System	18
4.8	Employee moudle of self-Employment Service Booking System	19
5.1	Unit Testing	26
5.2	Integration Testing	31
5.3	System Testing	35
5.4	Test Result	35
6.1	Index Page	42
6.2	Employee Registration Page	42
6.3	Login Page	43
6.4	Admin Dashboard Page	43
6.5	Approved List	44
6.6	User Booking Page	44
8.1	Plagiarism Report	46
9.1	Poster Presentation	54

LIST OF ACRONYMS AND ABBREVIATIONS

CMS	Content Management System
CRM	Customer Relationship Management
CSS	Cascading Style Sheet
HTML	HyperText Markup Language
JS	JavaScript
MYSQL	MY Structured Query Language
OS	Operating System
PHP	Hypertext Preprocessor
MMS	Multimedia Message Service
SBM	Service Booking Management
SBS	Service Booking System
SES	Self-Employment Service
SESBS	Self-Employment Service Booking System
UI	User Interface
UX	User Experience
VSC	Visual Studio Code

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the project	2
1.3 Project Domain	2
1.4 Scope of the Project	3
2 LITERATURE REVIEW	4
3 PROJECT DESCRIPTION	8
3.1 Existing System	8
3.2 Proposed System	8
3.3 Feasibility Study	9
3.3.1 Economic Feasibility	9
3.3.2 Technical Feasibility	10
3.3.3 Social Feasibility	10
3.4 System Specification	10
3.4.1 Hardware Specification	10
3.4.2 Software Specification	11
3.4.3 Standards and Policies	11
4 METHODOLOGY	12
4.1 Self-Employment Service Booking System Architecture	12
4.2 Design Phase	13
4.2.1 Use Case	13
4.2.2 Class Diagram	14
4.2.3 Sequence Diagram	15

4.2.4	Activity Diagram	16
4.3	Module Description	17
4.3.1	Admin Module	17
4.3.2	User Module	18
4.3.3	Employee Module	19
4.4	Steps to execute/run/implement the project	20
4.4.1	Installation of software	20
4.4.2	Execution	20
4.4.3	Web Processing	21
5	IMPLEMENTATION AND TESTING	22
5.1	Input and Output	22
5.1.1	Input Design	22
5.1.2	Output Design	22
5.2	Testing	23
5.3	Types of Testing	23
5.3.1	Unit Testing	23
5.3.2	Integration Testing	26
5.3.3	System Testing	31
5.3.4	Test Result	35
6	RESULTS AND DISCUSSIONS	36
6.1	Efficiency of the Proposed System	36
6.2	Comparison of Existing and Proposed System	36
6.3	Sample Code	37
7	CONCLUSION AND FUTURE ENHANCEMENTS	45
7.1	Conclusion	45
7.2	Future Enhancements	45
8	PLAGIARISM REPORT	46
9	SOURCE CODE & POSTER PRESENTATION	47
9.1	Source Code	47
9.2	Poster Presentation	54
References		54

Chapter 1

INTRODUCTION

1.1 Introduction

Self-employment is not a regular or everyday occupation. Self-employment services are a plumber, an electrician, a carpenter, a technician, a beautician etc. These works depends on the one's necessity. Self-employment is the state of working for one-self rather than employer. They don't get any salary, they will be paid with wages according to the work. This application mainly useful for households, because they can get all the work done without going out. With our user-friendly interface, you can easily browse through our list of service providers, view their profiles, ratings, and reviews, and book appointments that suit your schedule.

A self-employed person is not often the same the same thing as being a business owner. The owner of a business, for instance, may hire employees and essentially become the boss—an employeeowner who operates and manages the business. Alternatively, a business owner has an ownership stake but may not be involved in the day-to-day operations of the company. In contrast, a person who is self-employed both owns the business and is also the primary or sole operator. The taxation rules that apply to those who are self-employed differ from the employee or a business owner.

The Self-Employment Service Booking System is built using PHP, HTML, CSS, JS, and MYSQL, which allows for a user-friendly interface for both service providers and customers. Service providers can create their profiles, manage their schedules, receive bookings and payments, and communicate with their customers through the platform. Customers, on the other hand, can create their profiles, view service provider ratings and reviews, select their preferred service provider based on availability, price, and location, and make secure payments.

1.2 Aim of the project

The main aim of the project is to create a web based application to the self employment service booking system. Self-Employee Service Booking is an online market place that helps connect service providers with the service seekers using HTML, CSS, JavaScript and Bootstrap as front-end and MySQL as back-end to help the present Society.

Self-Employment Service Booking System is to provide a practical solution for everyday problems by connecting customers with reliable service providers and supporting self employment. The "Self Employment Service Booking System" aims to provide an accessible and convenient platform that supports self employment, empowering individuals to work for themselves and earn a living. The platform's design allows service providers to set their working hours and provide services in multiple areas, creating opportunities for them to expand their client base and increase their income.

1.3 Project Domain

Web development is the process of creating and maintaining websites and web applications. It involves a wide range of activities, including web design, front-end development, back-end development, server-side scripting, database management, and website optimization. Web development typically begins with web design, which involves creating the visual layout, user interface, and overall aesthetics of a website or web application. This may include creating wire frames, mockups, and prototypes to conceptualize the design before it is implemented.

Front-end development involves writing code to create the user-facing components of a website or web application. This includes HTML, CSS, and JavaScript, which are the fundamental technologies used to build the structure, style, and interactivity of a website. Web development can go from developing a basic single static page of plain text to complex web applications. This domain is used by most of the organizations to build, create, and maintain their websites to attract their visitors who are visiting their website. PHP, HTML, CSS, JS and MYSQL are being used to develop this website.

1.4 Scope of the Project

The Main scope of the project is to create web pages for different modules(Admin login, Registration page, etc..) Integrate all the created web pages together. Test the website whether the data is correctly stored in the database and to make error free while navigating from one page to another.The system should allow service providers to register for an account, create profiles, and undergo verification checks to ensure their authenticity and credibility.

And make easy to utilize this web-based application with correct login credentials. The system should allow users to register for an account, create profiles, and authenticate their accounts.Test the website whether the data is correctly stored.The system should allow users and service providers to manage bookings, including scheduling, rescheduling, and cancellations. Users should be able to view their booking history, receive notifications about booking updates, and communicate with service providers.

Chapter 2

LITERATURE REVIEW

Ali, M. A., et al.,[1] provided a review of the current status, challenges, and opportunities of self-employment service booking platforms. It provided insights into the role of these platforms in the gig economy and highlighted the challenges and opportunities that arose in their operation. It began by defining self-employment service booking platforms and provided examples such as TaskRabbit, Uber, and Upwork. And then reviewed the literature on the topic and discussed the various challenges faced by self-employment service booking platforms. These challenges included trust and reputation, quality control, pricing, and regulatory issues.

Zhang, X., et al.,[2] conducted research on the influencing factors of user trust in self-employment service booking platforms. It identified the key factors that contributed to user trust in these platforms and provided insights into how platform providers could improve user trust. They reviewed the literature on trust in online platforms and identified several factors that influenced user trust, including platform reputation, perceived risk, and social influence. And then conducted a survey of users of self-employment service booking platforms to test the validity of these factors. The results of the study showed that platform reputation, perceived risk, and social influence were all significant factors that influenced user trust in self-employment service booking platforms. Specifically, users were more likely to trust platforms with a good reputation, low perceived risk, and high social influence.

S.S.S.H. Perera, et al.,[3] focused on the development and implementation of an online self-employment booking system with machine learning capability. It described the design, features, and functionality of the system, which is likely designed to facilitate the booking of services offered by self-employed individuals, such as freelancers, gig workers, or independent contractors. It

discussed how machine learning techniques are incorporated into the system to enhance its capabilities, such as personalized recommendations for service providers based on user preferences or past booking history, dynamic pricing based on demand and supply, fraud detection, or user behavior analysis.

Peng,S., et al.,[4] provided a comprehensive review of the application of artificial intelligence (AI) in the context of self-employment service booking. It discussed the current status and trends of using AI technologies in self-employment service booking platforms, such as online marketplaces or platforms that connect service seekers with self-employed individuals. The review covered various aspects of AI applications in self-employment service booking, including but not limited to, recommendation systems, intelligent pricing, chatbots or virtual assistants, fraud detection, user behavior analysis, and data analytics.

Wang,X., et al.,[5] focused on understanding the factors that influenced customers intention to continue using self-employment service booking platforms. It proposed and tested a dual process model that explained customers decision making processes and behaviors in the context of self-employment service booking platforms. The dual process model involved both cognitive and affective processes that influenced customers continuous use intention. The cognitive processes involved rational considerations, such as perceived usefulness, ease of use, perceived risk, and perceived value of the self-employment service booking platforms.

Liu, Y., et al.,[6] focused on the design of matching algorithms in the context of self-employment service booking platforms. Matching algorithms are critical components of such platforms as they facilitate the matching process between service seekers and service providers based on their preferences, requirements, and availability. It discussed the existing literature on matching algorithms in self-employment service booking platforms and identify the challenges and opportunities associated with algorithm design. A new approaches for improving the effectiveness and efficiency of matching algorithms, considering factors such as user preferences, location, availability, service quality, and other relevant constraints.

Alam, M.R., et al.,[7] presented a mobile application designed for booking and managing self-employment services. The app utilized various technologies, including cloud computing and location-based services, to facilitate the booking process and ensure the timely and efficient delivery of services. It described the features of the app, which included service provider selection, real-time tracking of service providers, and payment processing. It discussed the design and implementation of the app and report on the results of a usability study conducted with a group of users.

Yoon, J., et al.,[8] proposed the user behavior on self-employment service platforms with different service delivery modes, specifically online-to-offline (O2O) and offline-to-online (O2O) platforms. It analyzed user behavior data from a self employment service platform and compare user characteristics, service usage patterns, and retention rates between the two service modes. The study found that users on O2O platforms tended to be more frequent and loyal users compared to those on O2O platforms. It also discussed the implications of these findings for platform operators and suggest strategies to increase user retention and engagement.

Wibowo, W., et al.,[9] described an intelligent self-employment service booking system that uses customer preferences to recommend service providers that best match their needs and preferences. It presented the design and implementation of the system, which includes a customer profile management module and a recommendation engine based on a collaborative filtering algorithm. It also report on the results of a user study conducted to evaluate the system's performance in terms of accuracy and usability.

Kim, M., et al.,[10] proposed the factors that influenced user retention in self employment service platforms and compared the determinants between online-to of fine (O2O) and offline-to-online (O2O) platforms. It found that trust, perceived usefulness, and social influence are important determinants of user retention in both O2O and O2O platforms. However, it also found that different factors, such as per ceived ease of use and perceived risk, have a greater impact on user retention in O2O platforms compared to O2O platforms. They

also provided valuable insights into the factors that contributed to user retention in self-employment service platforms and highlighted the importance of understanding the differences between O2O and O2O platforms when developing strategies to improve user retention.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

This is the existing system which is online based. Every operation was been performed manually. In another way, some intermediaries were involved. No particular web application was used. Self-employment service booking systems typically operate through online platforms that connect service seekers with service providers. These platforms can be web-based, allowing users to access and use the services from anywhere with an internet connection. Both service seekers and service providers create user profiles on the platform. Service seekers can create profiles with their preferences, requirements, while service providers can create profiles with their skills, availability.

Disadvantages:

- Consumes more time.
- Less employment.
- Under Management.
- Limited human interaction.

3.2 Proposed System

The proposed system consists of three major modules. The application has different registration pages for both employee and user. They also has their individual login credentials. According to the requirement of the user , they book the employee who has those skills. If the employee is available or interested to work they accepts the request from users.The proposed that is easy to navigate and use. Users should be able to quickly search for and book self-employment services, view service

providers profiles, and manage their bookings in less time.

The proposed system should allow service providers to create detailed profiles that showcase their skills, expertise, experience, and other relevant information. This can help users make informed decisions when booking services based on the service providers qualifications and reviews. It should allow users to book services based on their preferences, such as service type, location, date, and time. It should also provide scheduling functionality that allows service providers to manage their availability and accept or decline bookings based on their availability.

Advantages:

- Reduces the time.
- More opportunities.
- Increased business opportunities.
- There is less investment.
- Enhanced user experience.
- Cost-effective.
- Enhanced communication.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of a self-employment service booking system project can be assessed by considering various factors related to the potential costs and benefits associated with the system. The initial cost of developing the self-employment service booking system, including software development, database setup, hosting, and other associated expenses, should be evaluated. This may include hiring developers, purchasing software or tools, and any ongoing maintenance or updates required. The market demand for self-employment services and the potential customer base for the booking system should be analyzed.

3.3.2 Technical Feasibility

The technical feasibility of a self-employment service booking system project can be assessed by evaluating the technical aspects of its development and implementation. The technology requirements of the self-employment service booking system, including the software, hardware, and infrastructure needed to develop and run the system, should be evaluated. This may include programming languages, databases, frameworks, hosting solutions, and other technical components. The availability of the required infrastructure, resources, and technical support for the self-employment service booking system should be assessed. This may include evaluating the availability of servers, networks, storage, and other resources needed to host and operate the system effectively.

3.3.3 Social Feasibility

The social feasibility of a self-employment service booking system project can be assessed by evaluating the social, cultural, and ethical aspects of the system and its impact on various stakeholders. The willingness and acceptance of potential users to adopt and use the self-employment service booking system should be evaluated. This may involve conducting surveys, interviews, or focus groups to gather feedback from potential users about their attitudes, preferences, and concerns related to the system. Understanding the social dynamics and behaviors of the target user group is crucial in assessing the social feasibility of the project. The potential social impact of the self-employment service booking system on various stakeholders, such as service providers, customers, and the broader community, should be considered.

3.4 System Specification

3.4.1 Hardware Specification

- Processor: i3 or above.
- Speed: Computer/laptop with 1.6GHz.
- RAM: Minimum 4GB RAM.
- Storage: Minimum 1GB Disk Space.

3.4.2 Software Specification

- Operating system: Windows 7 and above.
- Database: MYSQL.
- Scripting Language: PHP.
- Front–End: HTML, CSS, JavaScript and Bootstrap
- Back–End: MySQL.
- Web Server: XAMPP Server v7.4 and above.
- Code Editor: VS Code

3.4.3 Standards and Policies

- **Data privacy and security policies:** Self-employment service booking systems often collect personal and sensitive data from both service providers and users. Therefore, it is important to have policies and regulations in place to protect the privacy and security of this data.
- **Payment policies:** Self-employment service booking systems should have clear policies regarding payment processing, including pricing transparency, payment methods, and dispute resolution mechanisms.
- **Accessibility policies:** Self-employment service booking systems should ensure that their platforms are accessible to all users, including those with disabilities. This may include providing alternative interfaces or support for assistive technologies.
- **ISO/IEC 27001:** This is a widely recognized international standard for information security management systems. It provides a framework for managing and protecting sensitive data, which is particularly relevant for self-employment service booking systems that collect and store personal and financial information.
- **ISO 9001:** This is another international standard that provides a framework for quality management. It can help self-employment service booking systems to establish procedures for service delivery, customer support, and continuous improvement.

Chapter 4

METHODOLOGY

4.1 Self-Employment Service Booking System Architecture

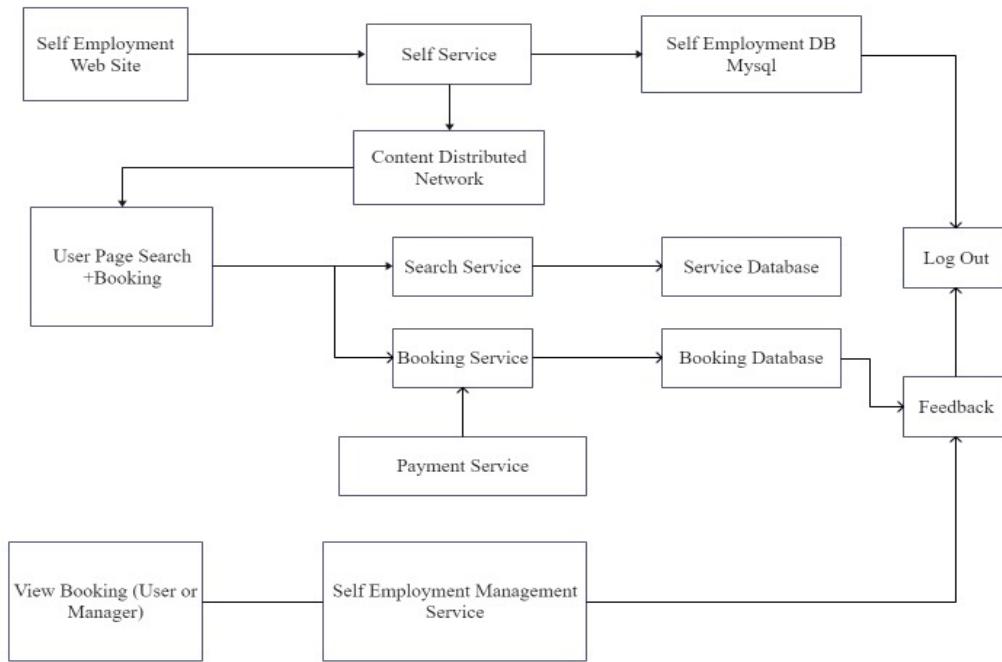


Figure 4.1: Architecture Diagram of Self-Employment Service Booking System

Figure 4.1 shows the architecture diagram of the self-employment service booking system. The architecture of a self-employment service booking system project typically involves several components that work together to facilitate the booking, scheduling, and management of self-employed service providers and their services. The specific architecture and technologies used may vary depending on the requirements, technologies, and development frameworks chosen for the self-employment service booking system project.

4.2 Design Phase

4.2.1 Use Case

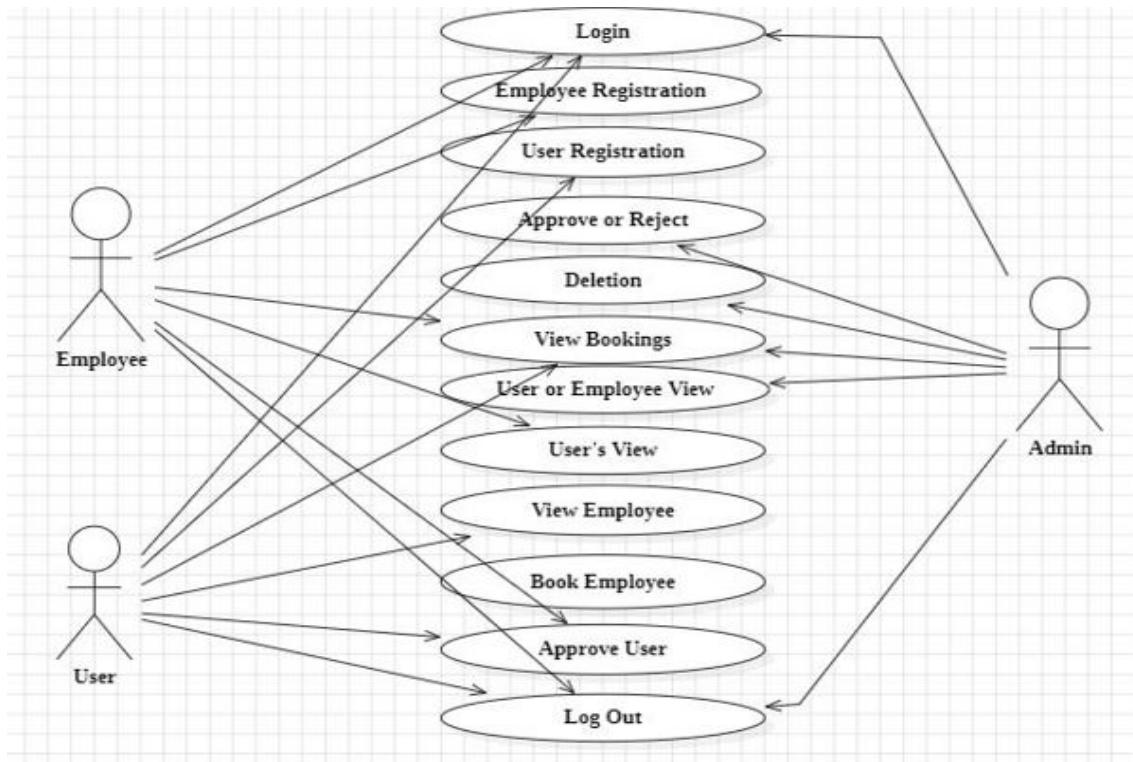


Figure 4.2: Use Case Diagram of Self-Employment Service Booking System

Figure 4.2 shows the use case diagram of the self-employment service booking system. The use case diagram for a self-employment service booking system involves two main actors: Employee and Customers/Users. Employee register on the platform and manage their services and bookings, while Customers search for services, make bookings, and provide feedback. The diagram includes use cases such as Register as a Service Provider, Search for Services, Book Service, Manage Bookings, and Rate and Review. Overall, the use case diagram shows how they facilitates the interaction between Employee and Customers to book and manage services.

4.2.2 Class Diagram

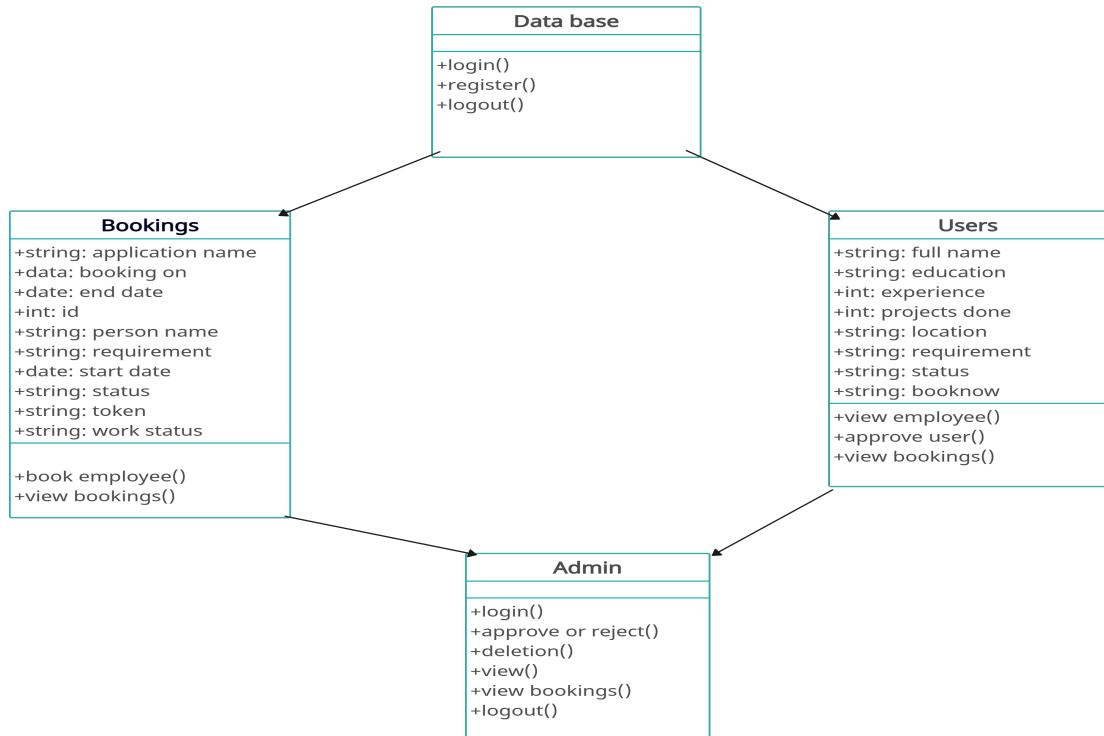


Figure 4.3: Class Diagram of Self-Employment Service Booking System

Figure 4.3 shows the class diagram of the self-employment service booking system. A class diagram for a self-employment service booking system shows the various classes, their attributes, and the relationships between them. The main classes include Employee, Customer/User, Booking, Service, Payment, and Review. Employee and Customer/User classes have attributes such as name, contact information, and ID, while Booking and Payment classes have attributes such as date/time and amount. The Service class has attributes such as service name and price, while the Review class has attributes such as rating and comments. The class diagram provides a visual representation of the entities and their relationships, enabling developers to design and implement the self-employment service booking system.

4.2.3 Sequence Diagram

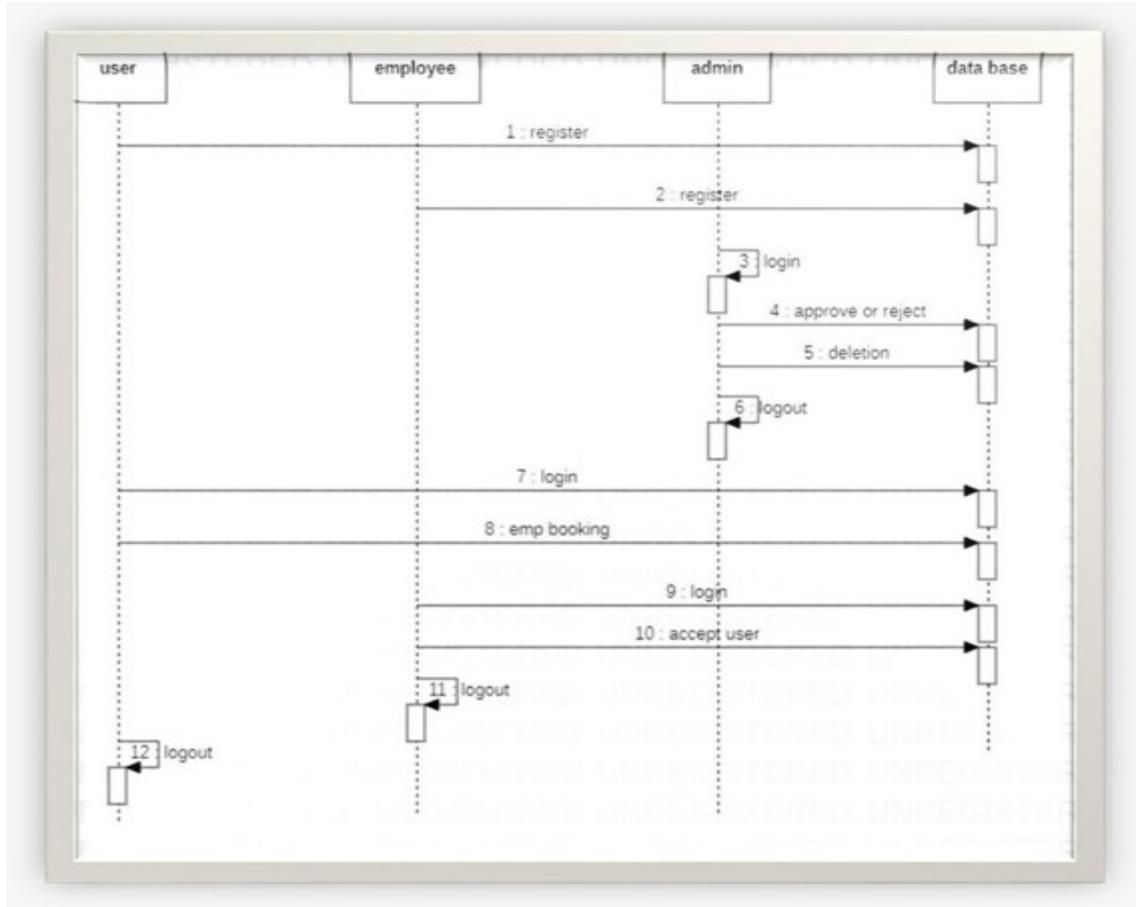


Figure 4.4: Sequence Diagram of Self-Employment Service Booking System

Figure 4.4 shows the sequence diagram of the self-employment service booking system. A sequence diagram for a self-employment service booking system illustrates the sequence of interactions between the actors and the system. The diagram shows the steps involved in making a booking, starting with the Customer/User searching for a service and ending with the Employee confirming the booking. The diagram includes actors such as the Customer/User, the Service Provider, and the system. The Employee receives the booking request, reviews the details, and either accepts or rejects the booking. The sequence diagram provides a visual representation of the flow of actions and interactions between the actors and the system, enabling developers to identify potential issues and improve the system design.

4.2.4 Activity Diagram

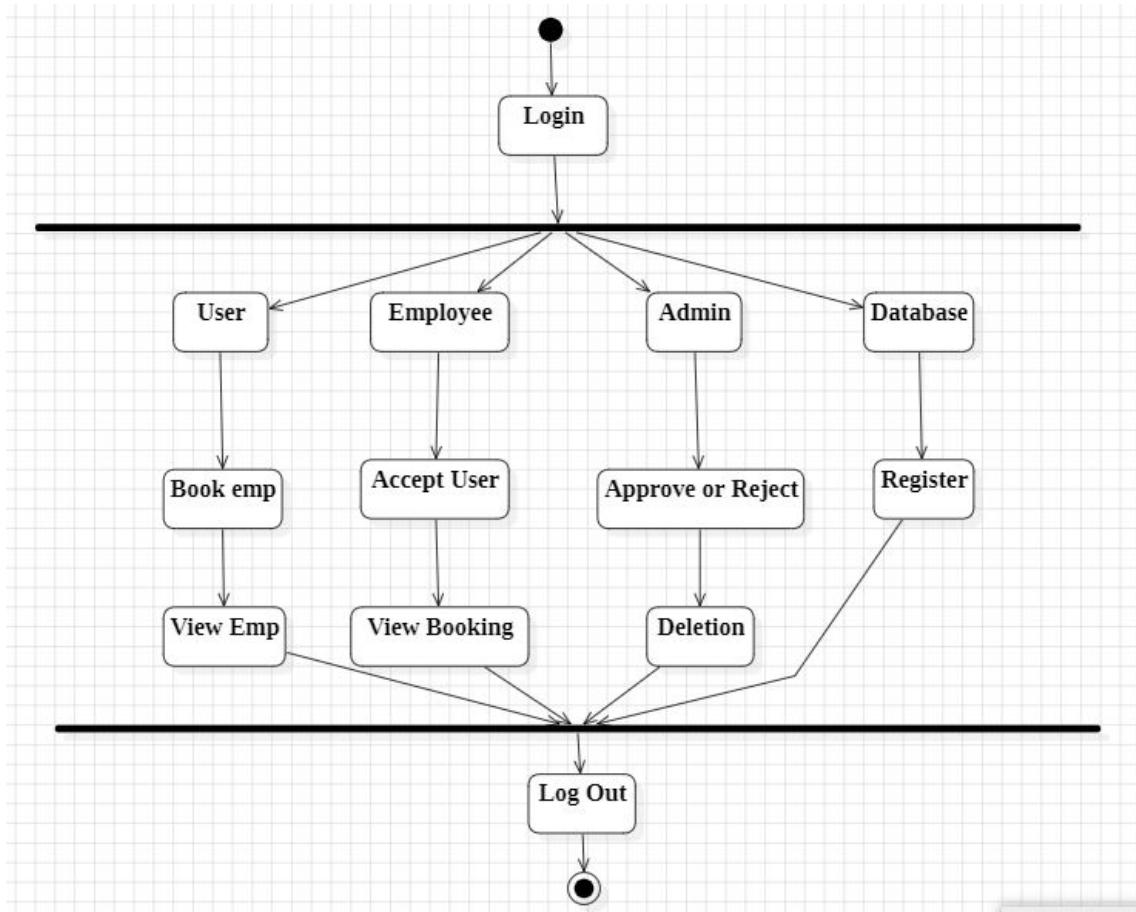


Figure 4.5: Activity Diagram of Self-Employment Service Booking System

Figure 4.5 shows the activity diagram of the self-employment service booking system. An activity diagram for a self-employment service booking system shows the flow of activities involved in the process of booking a service. The diagram illustrates the steps taken by the Customer/User and the Employee, as well as the system's responses at each step. The activities involved include searching for a service, selecting a service, providing personal information, making a payment, and confirming the booking. The Employee's activities include receiving and reviewing the booking request, confirming the availability of the service, and either accepting or rejecting the booking. It provides a visual representation of the process flow and helps developers to identify potential bottlenecks and improve the system design.

4.3 Module Description

4.3.1 Admin Module

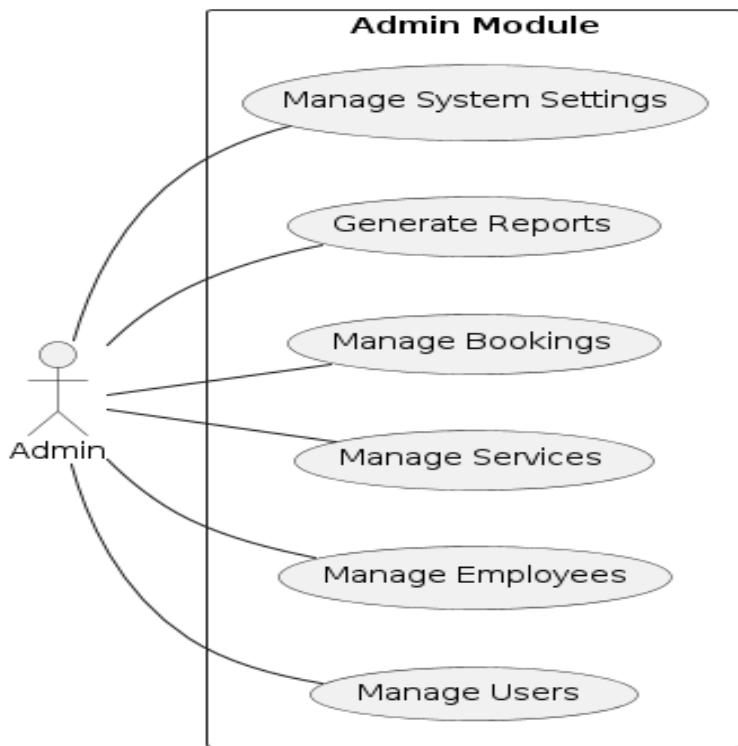


Figure 4.6: Admin module of self-Employment Service Booking System

Figure 4.6 Admin module enables the platform owner to manage user accounts, service providers, bookings, payments, reviews, and other aspects of the system, ensuring smooth and efficient operations. Approves the users and employees by knowing their details. Dashboard is the main interface of the admin module, providing an overview of key metrics such as the number of registered users, service providers, bookings, and revenue generated. The self-employed service booking system. It illustrates how customers and service providers interact with the web portal to access services, while the admin manages the system through the admin dashboard. The database stores data related to service providers, customers, services, and bookings. The services as the backbone of the self-employed service booking system, facilitating seamless management of user accounts, service providers, bookings, payments, and reviews. Through the comprehensive dashboard, the platform owner gains valuable insights into key metrics such as user registrations, service provider listings, booking volumes, and revenue generated. With the ability to approve users and employees based on detailed information, the admin ensures the integrity and

security of the platform. This centralized control allows for efficient oversight of all system operations, guaranteeing a smooth and reliable user experience. The database serves as the repository for crucial data, housing information on service providers, customers, services offered, and booking details. By leveraging the admin module, the platform owner can effectively monitor and optimize the system to meet the evolving needs of both customers and service providers, fostering growth and success in the self-employed service industry.

4.3.2 User Module

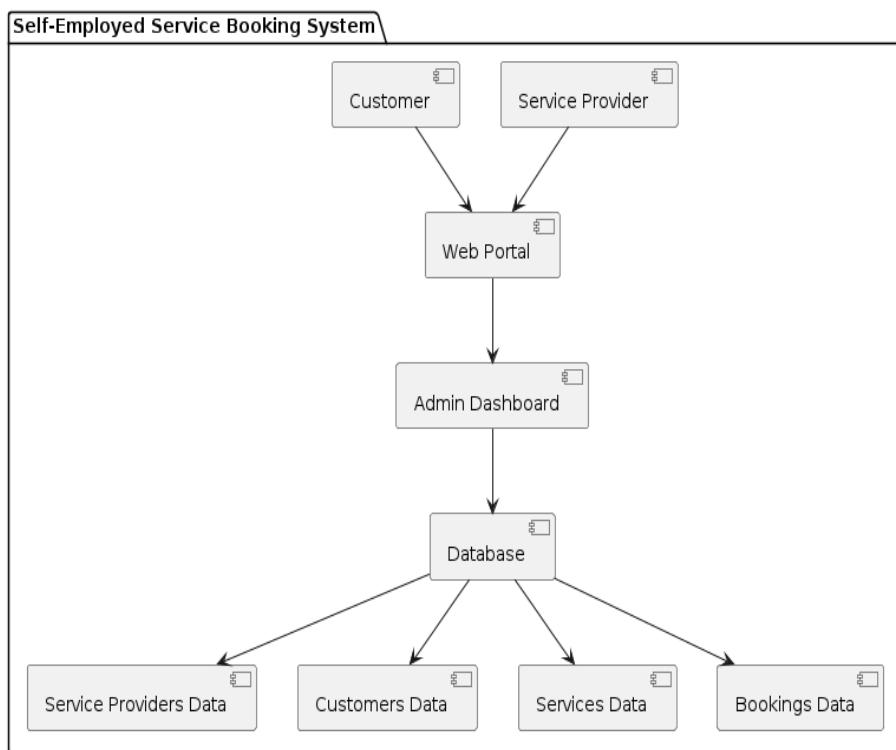


Figure 4.7: User module of self-Employment Service Booking System

Figure 4.7 User first registers and creates his login credentials. Later with those login credentials they enters into there page and performs the operations like viewing details of employee and booking the services. Users can book a service by selecting the provider, date, and time. They can also provide additional details about their requirements and request a quote. Users can contact customer support if they need help with their booking or have any other questions or issues. At the core lies the User Dashboard, serving as the central interface where users initiate their interactions. From here, users can effortlessly access and engage with a multitude of features. They can Browse Services to explore the diverse array of offerings available, or em-

ploy the Search Services function to pinpoint specific services based on their requirements. Once a desired service is identified, users can proceed to Book Appointment, streamlining the process of scheduling appointments with service providers at their convenience. Additionally, users can leverage View Service Provider Profiles to gain comprehensive insights into service providers, including their offered services, pricing details, availability, and user ratings and reviews, enabling informed decision-making. The View Ratings and Reviews feature further enhances user confidence by providing access to feedback from previous customers. Users have the autonomy to manage their profiles through Manage Profile, facilitating updates and ensuring accuracy of their information. Moreover, the platform facilitates secure and convenient transactions with Make Payments, enabling users to complete payments for booked services seamlessly. Lastly, Communication with Service Providers serves as a vital channel for direct interaction, allowing users to engage in discussions, seek clarifications, and address queries or concerns with service providers. Together, these integrated functionalities within the "User Module" collectively enrich the user experience, empowering users to efficiently browse, book, manage, and communicate with service providers on the Self-Employee Service Booking platform.

4.3.3 Employee Module

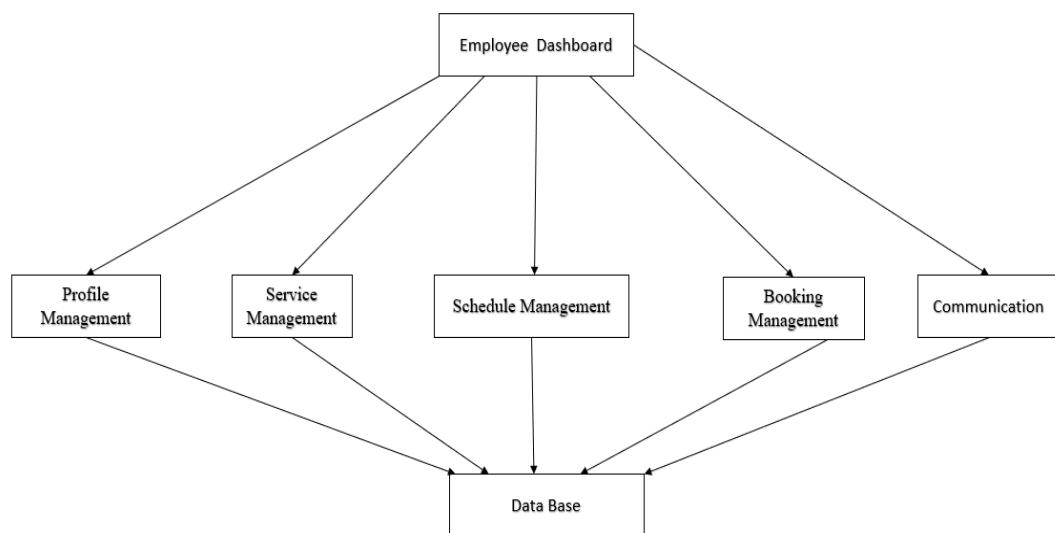


Figure 4.8: Employee moudle of self-Employment Service Booking System

Figure 4.8 Employee also performs registration and creates login credentials. Later they login into the page and can view users. They may accept/reject or delete the user's request. Employees can create and manage their services by specifying the service category, description, pricing, and availability. They can also view and respond to customer requests and messages. Employees can view their upcoming and past bookings, accept or reject booking requests, and communicate with customers to confirm the booking details and requirements. The Employee Module within the Self-Employee Service Booking system, encapsulating the core functionalities essential for service providers. At the heart of the module lies the "Employee Dashboard," serving as a centralized interface for employees to manage their operations effectively. All these functionalities interact with the underlying "Database" component, which serves as the repository for crucial data related to employee profiles, services, schedules, bookings, and communication logs. This integration ensures data integrity, accessibility, and reliability, facilitating smooth operations within the Employee Module.

4.4 Steps to execute/run/implement the project

Below are the steps followed to implement the project:

4.4.1 Installation of software

To implement this project, below are the softwares required:

- XAMPP Server to run the application.
- VScode for the code editor.

4.4.2 Execution

- All the module code is arranged and placed in a separate folder called "Self Employment".
- Copy paste the "Self Employment" folder in this path "c:/xampp/htdocs/".
- Now configure the database in phpmyadmin by creating a database named "employment service". To create the data tables import the employment service.sql file into the database named "employment service".

- Now in a browser paste the link "http://localhost/Self Employment/" to open the application.
- Now login to the system using there credentials. After logging in, user can perform different operations according to their privileges.

4.4.3 Web Processing

- Once open the index page after the above process then create a user registration, employee registration.
- Then from the admin panel there is option to accept or reject, once accept then the user and employee can get access to the web site.
- After entering the username, password then press login button.
- Then for user and employee they observe the dashboard with the inbuilt previous details.
- After this users can book required service by the booking option. Need to wait for the Employee approval for his required service.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The user should be able to create an account with their personal information, such as name, email, phone number, and address. First, you need to design a user interface that is easy to use and allows users to quickly find the services they need. Next, you should provide a list of available services, categorized by type. Users should also be able to see the availability of service providers and book appointments easily. The booking process should be streamlined and allow users to pay for the services securely and easily. Reviews and ratings should be available to help users make informed decisions, and notifications should be sent to remind users of upcoming appointments. Service providers should have access to a dashboard where they can manage their availability, view bookings and earnings, and update their profile. Finally, customer support options should be provided to address any issues or concerns users may have. By incorporating these inputs, you can create a self-employment service booking system that is user-friendly, efficient, and effective in connecting users with service providers.

5.1.2 Output Design

The system should display the user's availability for bookings, such as the days and times they are willing to work, and indicate which timeslots are already booked. The system should display the user's service description, including any pricing information, minimum and maximum order size, and any other relevant details. The system should display the user's location where they provide their services. The system should send a confirmation email or message to the user and the client when a booking is made, indicating the date and time of the booking, and any other relevant details. The system should display reviews and ratings left by clients on the user's service page, indicating the client's name, rating, and review text.

5.2 Testing

Testing is an essential part of the development process in a self-employment service booking system project. It is crucial to ensure that the system is reliable, meets user requirements, and works as expected. There are several types of testing that are typically used in such projects, including unit testing, integration testing, system testing, white box testing, and black box testing.

5.3 Types of Testing

5.3.1 Unit Testing

Unit testing is a type of testing that involves testing individual modules or components of the system. It is typically performed by developers to ensure that each unit of code works correctly and meets the functional requirements. In the context of a self-employment service booking system, unit testing may involve testing the functionality of the user interface, database, or other components of the system.

Input

```
1
2 <?php include 'includes/connection.php';?>
3 <?php include 'includes/navbar.php';?>
4
5 <?php
6 // session_start();
7 if (isset($_POST['login'])) {
8     $username = $_POST['username'];
9     $password = $_POST['password'];
10    mysqli_real_escape_string($conn, $username);
11    mysqli_real_escape_string($conn, $password);
12
13    $query = "SELECT * FROM users WHERE username = '$username'";
14    $result = mysqli_query($conn, $query) or die (mysqli_error($conn));
15    if (mysqli_num_rows($result) > 0) {
16        while ($row = mysqli_fetch_array($result)) {
17            $id = $row['id'];
18            $username = $row['username'];
19            $pass = $row['password'];
20            $fullname = $row['fullname'];
21            $email = $row['email'];
22            $role = $row['role'];
23            $education = $row['education'];
```

```

24     $skills= $row[ 'skills '];
25     $address = $row[ 'address '];
26     $status=$row[ 'status '];
27
28 //echo "$education";
29 if ($password == $pass) { // password_verify($password ,
30     $password )
31     $_SESSION[ 'id ']= $id;
32     $_SESSION[ 'username ']= $username;
33     $_SESSION[ 'fullname ']= $fullname;
34     $_SESSION[ 'email ']= $email;
35     $_SESSION[ 'role ']= $role;
36     $_SESSION[ 'education ']= $education;
37     $_SESSION[ 'skills ']= $skills;
38     $_SESSION[ 'address ']= $address;
39     $_SESSION[ 'status ']= $status;
40     if($_SESSION[ 'status ']== 'approved '){
41
42         if($_SESSION[ 'role ']== 'skilledperson '){
43             header('location:employee/users.php');
44         }
45         else{
46             header('location: user/dashboard.php');
47         }
48         else{
49             echo "<script>alert('You need to login after admin approval..')";
50             window.location.href= 'login.php';</script>";
51         }
52 // echo "<script>alert('Welcome to Admin login ');
53 //window.location.href= 'admindisplay.php';</script >";
54
55     }
56     else {
57         echo "<script>alert('invalid username/password ');
58         window.location.href= 'login.php';</script>";
59
60     }
61 }
62 }
63 else {
64     echo "<script>alert('invalid username/password ');
65     window.location.href= 'login.php';</script>";
66
67 }
68 }
69 ?>
70 \begin{figure}
71     \centering

```

```

73 \includegraphics[width=0.5\linewidth]{test_result.jpg}
74 \caption{Enter Caption}
75 \label{fig:enter-label}
76 \end{figure}
77 <body style="background-color: #2053df"><br><br><br><br><br><br><br><br><br><br>
78 <div>
79   <h1 style="font-size: 30px; font-weight: bold; text-shadow: 2px 2px orange; text-align: center;">
80     Log-in </h1><br>
81   <div style="width: 400px; background-color: white; border: double; border-radius: 10px; margin:
82     auto;">
83     <br><br>
84     <table style="height: 200px; margin: auto; width: 300px">
85       <tr><td> <input type="text" name="username" class="form-control" placeholder="Enter User
86         name" required=""></td></tr>
87       <tr><td> <input type="password" name="password" class="form-control" placeholder="Enter
88         Password" required=""></td></tr>
89       <tr><td><input type="submit" name="login" class="form-control btn-primary" value="Log in"
90           ><br><div class="login-help">
91             <a href="#">Register </a>      <a href="#">Forgot Password </a>
92           </div></td></tr>
93       </form>
94
95     </table>
96
97   </div>
98   <br><br>
99
100 <?php include 'includes/footer.php';?>
```

Test result

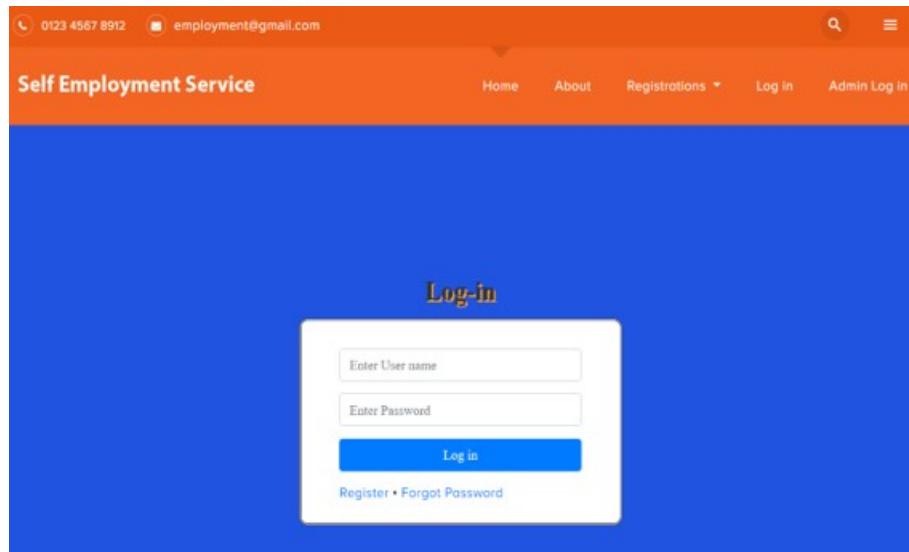


Figure 5.1: Unit Testing

As shown in figure 5.1 Login page was designed to enter your email and password to access account. Manage your bookings and view your profile after login.

5.3.2 Integration Testing

Integration testing involves testing how different modules or components of the system work together. It is typically performed to ensure that the individual units of code can communicate with each other and work as expected. In the context of a self-employment service booking system, integration testing may involve testing how the booking module interacts with the payment module or how the user interface interacts with the database.

Input

```
1 <?php include ('includes/connection.php'); ?>
2 <?php include('includes/adminheader.php'); ?>
3
4
5 <div id="wrapper">
6
7     <?php include 'includes/adminnav.php';?>
8     <div id="page-wrapper">
9
10        <div class="container-fluid">
```

```

12
13     <div class="row">
14         <div class="col-lg-12">
15             <h1 class="page-header">
16                 Welcome
17                 <small><?php echo $_SESSION[ 'name' ]; ?></small>
18             </h1>
19         <?php if($_SESSION[ 'role' ] == 'admin') {
20             ?>
21         <h3 class="page-header">
22             <center> <marquee width = 70% ><font color="green" > Notes uploaded by
23             College </font ></marquee></center>
24         </h3>
25     <div class="row">
26         <div class="col-lg-12">
27             <div class="table-responsive">
28
29         <form action="" method="post">
30             <table class="table table-bordered table-striped table-hover">
31
32             <thead>
33                 <tr>
34                     <th>Name</th>
35                     <th>User Name</th>
36                     <th>Role </th>
37                     <th>Email </th>
38                     <th>Gender </th>
39                     <th>Status </th>
40                     <th>Delete </th>
41
42
43                 </tr>
44             </thead>
45             <tbody>
46
47                 <?php
48
49 $query = "SELECT * FROM users WHERE token='no'";
50 $run_query = mysqli_query($conn, $query) or die(mysqli_error($conn));
51 if (mysqli_num_rows($run_query) > 0) {
52     while ($row = mysqli_fetch_array($run_query)) {
53         $name = $row[ 'name' ];
54         $username= $row[ 'username' ];
55         $role = $row[ 'role' ];
56         $email = $row[ 'email' ];
57         $gender = $row[ 'gender' ];
58
59         echo "<tr>";
60

```

```

61 echo "<td>$name</td>";
62 echo "<td>$username </td>";
63 echo "<td> $role </td>";
64 echo "<td>$email </td>";
65 echo "<td>$gender </td>";
66
67
68 echo "<td><a onClick=\"$javascript: return confirm('Are you sure you want to approve this user?')\"
69 \>\" href='?approve=$username'><i class='fa fa-times' style='color: red;'></i>Approve</a></td>";
70 ;
71
72 echo "<td><a onClick=\"$javascript: return confirm('Are you sure you want to delete this post?')\"
73 \>\" href='?del=$username'><i class='fa fa-times' style='color: red;'></i>delete </a></td>";
74
75 echo "</tr>";
76
77
78
79         </tbody>
80     </table>
81 </form>
82 </div>
83 </div>
84 </div>
85 <?php
86
87 if (isset($_GET['del'])) {
88     $note_del = mysqli_real_escape_string($conn, $_GET['del']);
89     $file_uploader = $_SESSION['username'];
90     $del_query = "DELETE FROM users WHERE username='$username'";
91     $run_del_query = mysqli_query($conn, $del_query) or die (mysqli_error($conn));
92     if (mysqli_affected_rows($conn) > 0) {
93         echo "<script>alert('user deleted successfully');
94             window.location.href='index.php';</script>";
95     }
96     else {
97         echo "<script>alert('error occured.try again!');</script>";
98     }
99 }
100
101 if (isset($_GET['approve'])) {
102     $note_approve = mysqli_real_escape_string($conn, $_GET['approve']);
103     $approve_query = "UPDATE users SET token='yes' WHERE username='$username'";
104     $run_approve_query = mysqli_query($conn, $approve_query) or die (mysqli_error($conn));
105     if (mysqli_affected_rows($conn) > 0) {
106         echo "<script>alert('user approved successfully');
107             window.location.href='index.php';</script>";

```

```

108     }
109     else {
110         echo "<script>alert(' error occured.try again!');</script>";
111     }
112 }
113
114 ?>
115 <?php
116 }
117 }
118 else {
119 ?>
120
121
122 <h3 class="page-header">
123             <center> <marquee width = 70% ><font color="green" ><?php echo $_SESSION
124                 [ 'course' ]; ?> Engineering </font><font color="brown">User Approvals
125                 </font></marquee></center>
126             </h3>
127
128         </div>
129     </div>
130 <div class="row">
131 <div class="col-lg-12">
132     <div class="table-responsive">
133
134 <form action="" method="post">
135     <table class="table table-bordered table-striped table-hover">
136
137         <thead>
138             <tr>
139                 <th>Name</th>
140                 <th>Description </th>
141                 <th>Type </th>
142                 <th>Uploaded by</th>
143                 <th>Uploaded on</th>
144                 <th>Download</th>
145                 <th>Status </th>
146
147             </tr>
148         </thead>
149         <tbody>
150
151             <?php
152                 $currentusercourse = $_SESSION[ 'course' ];
153
154                 $query = "SELECT * FROM uploads WHERE file_uploaded_to = '$currentusercourse' AND status = 'approved
155                     ' ORDER BY file_uploaded_on DESC";
156
157                 $run_query = mysqli_query($conn, $query) or die(mysqli_error($conn));

```

```

155 if (mysqli_num_rows($run_query) > 0) {
156     $row = mysqli_fetch_array($run_query);
157     $file_id = $row['file_id'];
158     $file_name = $row['file_name'];
159     $file_description = $row['file_description'];
160     $file_type = $row['file_type'];
161     $file_date = $row['file_uploaded_on'];
162     $file = $row['file'];
163     $file_uploader = $row['file_uploader'];
164
165     echo "<tr>";
166     echo "<td>$file_name </td>";
167     echo "<td>$file_description </td>";
168     echo "<td>$file_type </td>";
169     echo "<td><a href='viewprofile.php?name=$file_uploader' target='_blank'> $file_uploader </a></td>
170             >";
171     echo "<td>$file_date </td>";
172     echo "<td><a href='allfiles/$file' target='_blank' style='color:green'>Download </a></td>";
173     echo "<td>approved by Admin </td>";
174     echo "</tr>";
175
176 }
177 }
178 ?>
179 </tbody>
180         </table>
181 </form>
182 </div>
183 </div>
184 </div>
185 <?php }>
186
187 ?>
188
189
190
191
192 <script src="js/jquery.js"></script>
193
194
195     <script src="js/bootstrap.min.js"></script>
196 </body>
197 </html>

```

Test result

The screenshot shows a web application titled 'Self Employment Service -- Admin'. The left sidebar has 'Dashboard', 'Users', and 'Bookings' options. The main content area is titled 'Approved Users list given below.' and contains a table with three rows of user data:

ID	Full Name	Role	Phone No	Education	Address	Requirement	Status	Delete
11	Harshitha	skilledperson	2147483647	B Tech	Proddatur	Bookings	approved	<button>Delete</button>
1	Ayyappa	skilledperson	1231231234	PG	hyd hyd TS india	software development	approved	<button>Delete</button>
2	Kamalesh Reddy	Admin	2147483647	Masters	USA USA USA	software development		

Figure 5.2: Integration Testing

As shown in figure 5.2 The approved users list page was designed to display a list of registered users who have been approved by the admin. This page allows admins to manage user access to the system.

5.3.3 System Testing

System testing involves testing the system as a whole to ensure that it meets the functional and non-functional requirements. It typically involves testing the user interface, database, and other components of the system. In the context of a selfemployment service booking system, system testing may involve testing how the booking process works from start to finish, including user registration, service selection, and payment processing.

Input

```
1 <?php
2 session_start();
3 // error_reporting(0);
4 include('includes/config.php');
5 if (strlen($_SESSION['id'])==0) {
6     header('location:logout.php');
7 } else{ ?>
8 <!DOCTYPE html>
9 <html lang="en">
```

```

10<head>
11    <meta charset="UTF-8" />
12    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
13        scalable=no" />
14    <title>Dashboard</title>
15    <link href="vendors/vectormap/jquery-jvectormap-2.0.3.css" rel="stylesheet" type="text/css" />
16    <link href="vendors/jquery-toggles/css/toggles.css" rel="stylesheet" type="text/css">
17    <link href="vendors/jquery-toggles/css/themes/toggles-light.css" rel="stylesheet" type="text/css
18        ">
19    <link href="vendors/jquery-toast-plugin/dist/jquery.toast.min.css" rel="stylesheet" type="text/
20        css">
21    <link href="dist/css/style.css" rel="stylesheet" type="text/css">
22</head>
23
24<body>
25
26    <!-- HK Wrapper -->
27    <div class="hk-wrapper hk-vertical-nav">
28
29        <?php include_once('includes/navbar.php');
30        include_once('includes/sidebar.php');
31        ?>
32            <div id="hk_nav_backdrop" class="hk-nav-backdrop"></div>
33            <!-- / Vertical Nav -->
34            <!-- Main Content -->
35            <div class="hk-pg-wrapper">
36                <!-- Container -->
37                <div class="container-fluid mt-xl-50 mt-sm-30 mt-15">
38                    <!-- Row -->
39                    <div class="row">
40                        <div class="col-xl-12">
41                            <div class="hk-row">
42
43        <?php
44        $quer=mysqli_query($con,"select requirment from users");
45        $listedcat=mysqli_num_rows($quer);
46        if (! $listedcat) {
47            die(mysqli_error($con));
48        }
49
50        <div class="col-lg-3 col-md-6">
51            <div class="card card-sm">
52                <div class="card-body">
53                    <div class="d-flex justify-content-between mb-5">
54                        <div>
55                            <span class="d-block font-15 text-dark font-weight-500">Categories </span>
56                        </div>

```

```

57 <div>
58 </div>
59 </div>
60 <div class="text-center">
61 <span class="d-block display-4 text-dark mb-5"><span class="counter-anim"><?php echo $listedcat;?></
62     span></span>
63 <small class="d-block">Listed Categories </small>
64 </div>
65 </div>
66 </div>
67
68
69 <?php
70 $role="skilledperson";
71 $query=mysqli_query($con,"select role from users where role='$role' ");
72
73 $listedcomp=mysqli_num_rows($query);
74 ?>
75 <div class="col-lg-3 col-md-6">
76 <div class="card card-sm">
77 <div class="card-body">
78 <div class="d-flex justify-content-between mb-5">
79 <div>
80 <span class="d-block font-15 text-dark font-weight-500">Skilled Persons </span>
81 </div>
82 <div>
83 </div>
84 </div>
85
86 <div class="text-center">
87 <span class="d-block display-4 text-dark mb-5"><span class="counter-anim"><?php echo $listedcomp
88 ;?></span></span>
89 <small class="d-block">Skilled Persons </small>
90 </div>
91 </div>
92 </div>
93
94
95
96 </div>
97
98     </div>
99     <!-- / Container -->
100
101     <!-- Footer -->
102 <?php include_once('includes/footer.php');?>
103         <!-- / Footer -->
104     </div>

```

```

105      <!-- /Main Content -->
106
107    </div>
108    <!-- /HK Wrapper -->
109
110    <!-- jQuery -->
111    <script src="vendors/jquery/dist/jquery.min.js"></script>
112    <script src="vendors/popper.js/dist/umd/popper.min.js"></script>
113    <script src="vendors/bootstrap/dist/js/bootstrap.min.js"></script>
114    <script src="dist/js/jquery.slimscroll.js"></script>
115    <script src="dist/js/dropdown-bootstrap-extended.js"></script>
116    <script src="dist/js/toggle.min.js"></script>
117    <script src="vendors/jquery-toggles/toggles.min.js"></script>
118    <script src="dist/js/toggle-data.js"></script>
119    <script src="vendors/waypoints/lib/jquery.waypoints.min.js"></script>
120    <script src="vendors/jquery.counterup/jquery.counterup.min.js"></script>
121      <script src="vendors/jquery.sparkline/dist/jquery.sparkline.min.js"></script>
122      <script src="vendors/vectormap/jquery-jvectormap-2.0.3.min.js"></script>
123      <script src="vendors/vectormap/jquery-jvectormap-world-mill-en.js"></script>
124    <script src="dist/js/vectormap-data.js"></script>
125      <script src="vendors/owl.carousel/dist/owl.carousel.min.js"></script>
126      <script src="vendors/jquery-toast-plugin/dist/jquery.toast.min.js"></script>
127      <script src="vendors/apexcharts/dist/apexcharts.min.js"></script>
128    <script src="dist/js/irregular-data-series.js"></script>
129    <script src="dist/js/init.js"></script>
130
131  </body>
132
133  </html>
134  <?php } ?>

```

Test Result

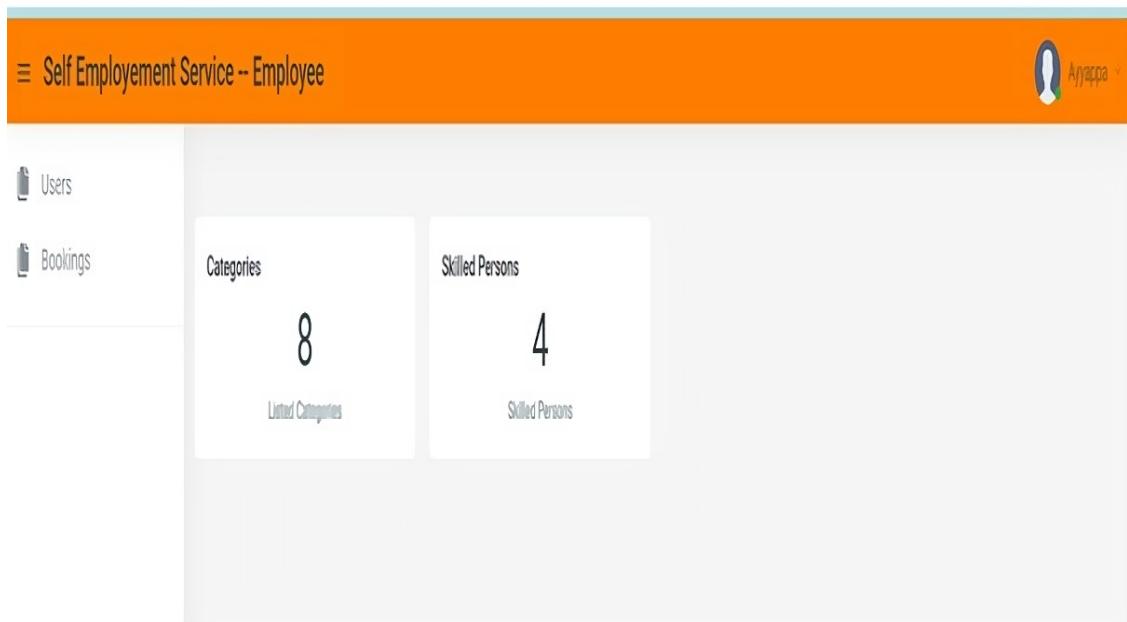


Figure 5.3: System Testing

As shown in figure 5.3, Admin dashboard page was designed to provide a summary of bookings, user data, system updates, count of skilled persons and categories.

5.3.4 Test Result

APPROACH	EXPECTED OUTPUT	ORIGINAL OUTPUT	RESULT
Unit Testing	Display of login page with logo, input field to enter mail and password, submit button without any errors.	Display of login page with logo, input field to enter mail and password, submit button without any errors.	SUCCESS
Integration Testing	Display of account details and tips of administrator together at a time.	Display of account details and tips of administrator together at a time.	SUCCESS
System Testing	Successful display of employee dashboard.	Successful display of employee dashboard.	SUCCESS

Figure 5.4: Test Result

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system for self employment service booking system should have a user-friendly interface that is easy to navigate and use. Users should be able to quickly understand how to book services, input necessary information, select service options, and confirm bookings without encountering any difficulties or confusion. A well-designed user interface can enhance the efficiency of the system by allowing users to complete bookings quickly and easily. The system should provide realtime updates on service availability, allowing users to see up-to-date schedules and available time slots. This helps prevent double bookings or conflicts and ensures that users can book services at their preferred times without delays or cancellations. Real-time availability updates can improve the efficiency of the system by providing accurate and timely information to users.

6.2 Comparison of Existing and Proposed System

Existing System:

- **User Interface:** The existing system has a complex and confusing user interface, making it difficult for users to navigate and book services efficiently.
- **Booking Process:** The existing system has a lengthy and cumbersome booking process with multiple steps and manual inputs, leading to delays and potential errors.
- **Availability Updates:** The existing system lacks real-time availability updates, resulting in potential conflicts and double bookings.
- **Notifications and Reminders:** The existing system has limited automated notifications and reminders, leading to missed appointments and miscommunications.

Proposed system:

- **User Interface:** The proposed system has a user-friendly interface that is easy to navigate and use, improving the efficiency of booking services.
- **Booking Process:** The proposed system has a streamlined and efficient booking process with minimal steps and clear instructions, reducing delays and potential errors.
- **Availability Updates:** The proposed system provides real-time availability updates, preventing conflicts and double bookings.
- **Notifications and Reminders:** The proposed system has automated notifications and reminders, ensuring users stay informed about their bookings and reducing missed appointments.

6.3 Sample Code

Login Page Code:

```
1 <?php include 'includes/connection.php';?>
2 <?php include 'includes/navbar.php';?>
3
4
5 <?php
6 // session_start();
7 if (isset($_POST['adminLogin'])) {
8     $username = $_POST['username'];
9     $password = $_POST['password'];
10    mysqli_real_escape_string($conn, $username);
11    mysqli_real_escape_string($conn, $password);
12
13    $query = "SELECT * FROM users WHERE username = '$username'";
14    $result = mysqli_query($conn, $query) or die (mysqli_error($conn));
15    if (mysqli_num_rows($result) > 0) {
16        while ($row = mysqli_fetch_array($result)) {
17            $id = $row['id'];
18            $username = $row['username'];
19            $pass = $row['password'];
20            $fullname = $row['fullname'];
21            $email = $row['email'];
22            $role = $row['role'];
23            $education = $row['education'];
24            //echo "$education";
25        }
26    }
27}
```

```

25 if ($password == $pass) { // password_verify($password ,
26     $password )
27     $_SESSION['id'] = $id;
28     $_SESSION['username'] = $username;
29     $_SESSION['fullname'] = $fullname;
30     $_SESSION['email'] = $email;
31     $_SESSION['role'] = $role;
32     $_SESSION['education'] = $education;
33     if($_SESSION['role']=='Admin'){
34         header('location: admin/dashboard.php');
35     } else{
36         echo "<script>alert('Admin only login here you need to click on login option only');"
37         window.location.href= 'adminlogin.php';</script>";
38     }
39     //echo "<script>alert('Welcome to Admin login');"
40     //window.location.href= 'admindisplay.php';</script >";
41 }
42 }
43 else {
44     echo "<script>alert('invalid username/password');"
45     window.location.href= 'adminlogin.php';</script>";
46 }
47 }
48 }
49 }
50 else {
51     echo "<script>alert('invalid username/password');"
52     window.location.href= 'adminlogin.php';</script>";
53 }
54 }
55 ?>
56
57
58 <body style="background-color: #2053df"><br><br><br><br><br><br><br><br><br><br><br><br><br>
59 <div >
60     <h1 style="font-size: 30px; font-weight: bold; text-shadow: 2px 2px orange; text-align: center;">
61         Admin Log-in </h1><br>
62         <div style="width: 400px; background-color: white; border: double; border-radius: 10px; margin:
63             auto;">
64             <br><br>
65             <table style="height: 200px; margin: auto; width: 300px">
66                 <form method="POST">
67
68                     <tr><td> <input type="text" name="username" class="form-control" placeholder="Enter User
69                         name" required=""></td></tr>
<tr><td> <input type="password" name="password" class="form-control" placeholder="Enter
                         Password" required=""></td></tr>

```

```

70     <tr><td><input type="submit" name="adminLogin" class="form-control btn-primary" value="
71         Login"></td></tr></form>
72     </table>
73 
74     </div>
75     <br><br>
76 
77 <?php include 'includes/footer.php';?>
78

```

User Register Code:

```

1 <?php include 'includes/connection.php';?>
2 <?php include 'includes/navbar.php';?>
3
4
5
6 <?php
7 if (isset($_POST['signup'])) {
8 session_start();
9
10    $fullname = mysqli_real_escape_string($conn, $_POST['fullname']);
11    $fathername=mysqli_real_escape_string($conn,$_POST['fathername']);
12    $dateofbirth=mysqli_real_escape_string($conn,$_POST['dateofbirth']);
13    $registredon = date("Y-m-d");
14    $role='user';
15    $email = mysqli_real_escape_string($conn, $_POST['email']);
16    $phoneno=$_POST['phoneno'];
17    $username = mysqli_real_escape_string($conn, $_POST['username']);
18    $password = mysqli_real_escape_string($conn, $_POST['password']);
19    $password2 = mysqli_real_escape_string($conn, $_POST['password2']);
20    $gender=$_POST['gender'];
21
22    $address=mysqli_real_escape_string($conn, $_POST['address']);
23    $requirment=mysqli_real_escape_string($conn, $_POST['requirment']);
24
25 if ($_POST['password'] !== $_POST['repassword2'])
26 {
27 echo "<center><font color='red'>Passwords do not match </font></center>";
28 }
29
30    $username = $_POST['username'];
31    $checkusername = "SELECT * FROM users WHERE username = '$username'";
32    $run_check = mysqli_query($conn, $checkusername) or die(mysqli_error($conn));
33    $countusername = mysqli_num_rows($run_check);
34    if ($countusername > 0 ) {
35        echo "<center><font color='red'>Username is already taken! try a different
one</font></center>";

```

```

36         }
37
38     // $email = $_POST['email'];
39     $checkemail = "SELECT * FROM users WHERE email = '$email'";
40     $run_check = mysqli_query($conn , $checkemail) or die(mysqli_error($conn));
41     $countemail = mysqli_num_rows($run_check);
42
43     if ($countemail > 0 ) {
44         echo "<center><font color='red'>Email is already taken! try a different one
45             </font></center>";
46     }
47
48     else{
49
50         $query = "INSERT INTO users(fullname,fathername,dateofbirth,registredon,role,email,phoneno,
51             username,password,gender,education,skills,experience,projects,address,reuirment,token)
52             VALUES ('$fullname' , '$fathername' , '$dateofbirth' , '$registredon' , '$role' , '$email' ,
53                 '$phoneno' , '$username' , '$password' , '$gender' , '' , '' , '' , '$address' ,
54                 '$reuirment' , '' )";
55         $result = mysqli_query($conn , $query) or die(mysqli_error($conn));
56         if (mysqli_affected_rows($conn) > 0) {
57             echo "<script>alert('SUCCESSFULLY REGISTERED');
58             window.location.href='login.php';</script>";
59         }
60     }
61
62
63
64 ?>
65
66 <body style="background-color: #2053df"><br><br><br><br><br><br><br><br><br><br><br>
67 <div >
68     <h1 style="font-size: 30px; font-weight: bold;text-shadow: 2px 2px orange; text-align: center;">
69         Registration for User</h1><br>
70     <div style="width: 500px; background-color: white; border: double; border-radius: 10px; margin:
71         auto;">
72         <br><br>
73         <table style="height: 700px; margin: auto; width: 400px">
74             <form method="post">
75                 <tr><td><input type="text" name="fullname" class="form-control" placeholder="Enter Full Name
76                     required=""></td></tr>
77                 <tr><td><input type="text" name="fathername" class="form-control" placeholder="Enter Father
78                     Name" required=""></td></tr>
79                 <tr><td>Date of birth: <input type="date" name="dateofbirth" class="form-control" required
80                     =""></td></tr>
81                 <tr><td> <input type="email" name="email" class="form-control" placeholder="Enter email id"
82                     required=""></td></tr>

```

```

75 <tr><td> <input type="text" name="phoneno" class="form-control" placeholder="Enter phone
    number" required=""></td></tr>
76 <tr><td> <input type="text" name="username" class="form-control" placeholder="Enter User
    name" required=""></td></tr>
77 <tr><td> <input type="password" name="password" class="form-control" placeholder="Enter
    Password" required=""></td></tr>
78 <tr><td> <input type="password" name="password2" class="form-control" placeholder="Enter
    Confirm Password" required=""></td></tr>
79 <tr><td> <select class="form-control" name="gender" required="">
80     <option value="">--Select Gender--</option>
81     <option>Male</option>
82     <option>Female</option>
83     <option>Transgender</option>
84 </select></td></tr>
85
86
87 <tr><td><textarea name="address" placeholder="Enter Address" class="form-control" rows="5"
    cols="50" required=""></textarea></td></tr>
88 <tr><td><input type="text" name="requirment" placeholder="Enter your requirment" class="form-
    control" required=""></td></tr>
89 <tr><td>I agree..<input type="checkbox" required=""></td></tr>
90 <tr><td><input type="submit" name="signup" id="submit" class="form-control btn-primary"
    value="Submit"></td></tr></form>
91 </table>
92 </div>
93
94 </div>
95 <br><br>
96
97 <?php include 'includes/footer.php';?>

```

Output

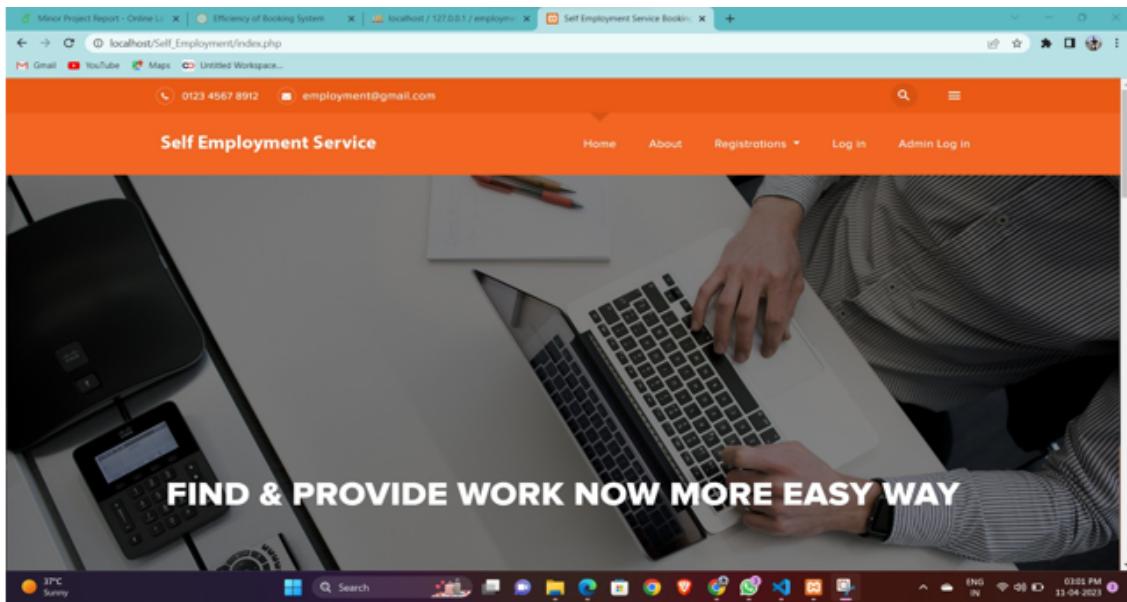


Figure 6.1: Index Page

As shown in figure 6.1, index page was designed to make it easy for users to access the services, admin able to login through admin login.

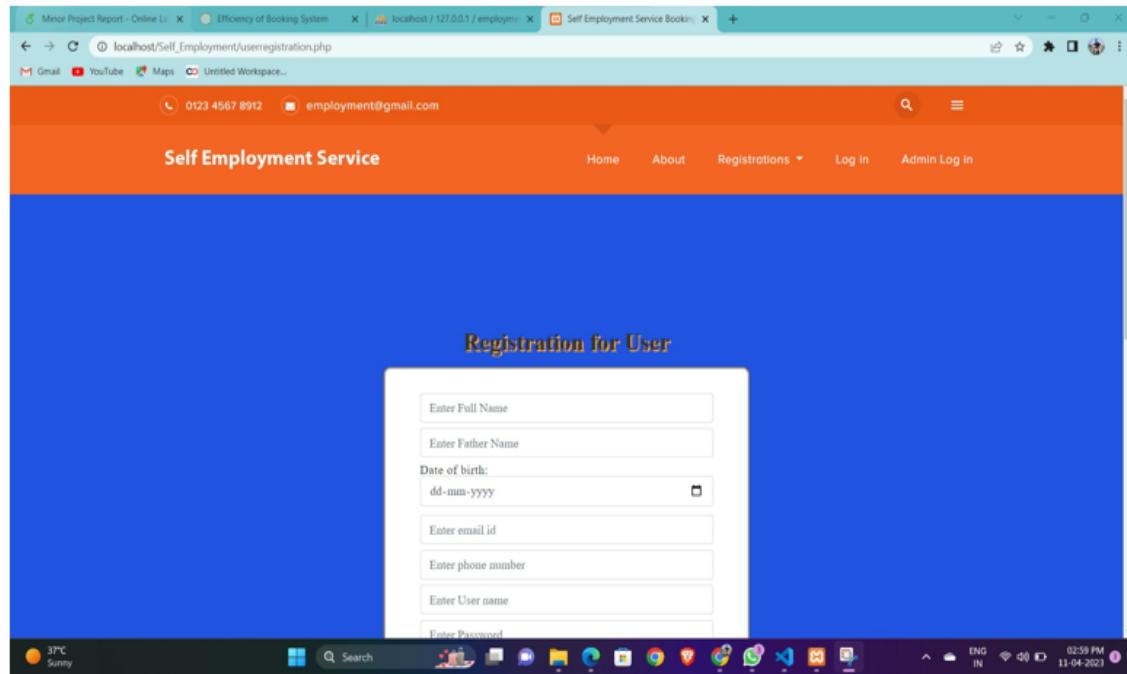


Figure 6.2: Employee Registration Page

As shown in figure 6.2, Employee registration page was deigned to allows new employees to create an account by providing personal and professional information, as well as setting up a password for the account.

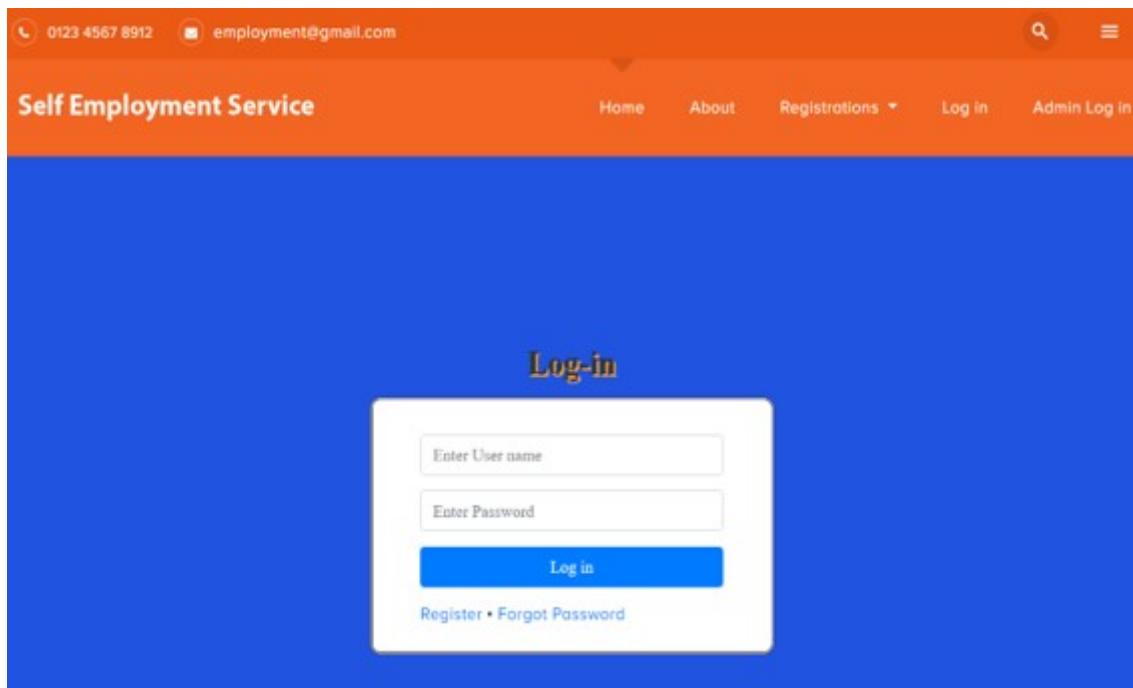


Figure 6.3: **Login Page**

As shown in figure 6.3, User registration page was designed to allows new users to create an account by providing personal and business information, as well as setting up a password for the account.

A screenshot of the admin dashboard. The top navigation bar is orange with the text "Self Employment Service - Employee" and a logo for "Ayyappa". On the left, there are two menu items: "Users" and "Bookings". The main content area shows two cards: one for "Categories" with the number 8 and another for "Skilled Persons" with the number 4. Both cards have links below them: "Listed Categories" and "Skilled Persons".

Figure 6.4: **Admin Dashboard Page**

As shown in figure 6.4, Admin dashboard page was designed to provide a summary of bookings, user data, system updates, count of skilled persons and categories.

Self Employment Service -- Admin								
		Users > Users						
		Approved Users list given below .						
<hr/>								
ID	Full Name	Role	Phone No	Education	Address	Requirement	Status	Delete
11	Harshitha	skilledperson	2147483647	B Tech	Proddatur	Bookings	approved	<button>Delete</button>
1	Ayyappa	skilledperson	1231231234	PG	hyd hyd TS india	software development	approved	<button>Delete</button>
2	Kamalesh Reddy	Admin	2147483647	Masters	USA USA USA	software development		

Figure 6.5: Approved List

As shown in figure 6.5, The approved users list page was designed to display a list of registered users who have been approved by the admin. This page allows admins to manage user access to the system.

Self Employment Service -- User								
Skilled Persons > Skilled Persons								
Approved skilledperson list given below You need to select for your business..								
<hr/>								
10	Items							
ID	Full Name	Education	Experience	Projects Done	Location	Requirement	Status	Book
1	Ayyappa	PG	7 years	7	hyd hyd TS india	software development	approved	<button>Book Now</button>
4	Sushma	MBA	2 years	2	HYD HYD TS	Marketing	approved	<button>Book Now</button>
5	Ramadevi	Degree	2 years	2	HYD HYD TS	Training	approved	<button>Book Now</button>
11	Harshitha	B Tech	5 years	2	Proddatur	Bookings	approved	<button>Book Now</button>

Figure 6.6: User Booking Page

As shown in figure 6.6, The User booking page was designed to allow registered users to browse available services, select a service, and book an appointments.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

This web application is mainly useful for those who are skilled but they are unemployed. Admin is the main person who decides to work on this web application. The main intention behind this web application is to provide employment who are skilled and looking for an employment. This application is helpful for providing interaction between user and employee without any intermediaries. The self-employment service booking system project is a valuable tool for individuals who offer their services as independent contractors. The system provides an easy and convenient way for service providers to manage their schedules, accept and schedule bookings, and communicate with their clients.

7.2 Future Enhancements

In recent times we don't have any application for booking services. So now we have developed an application for user and employee. In future we can add fixed costs for particular work within the web application. Developing a mobile app would enable users to access the system from their smartphones or tablets, making it more convenient and accessible. The system could send automated reminders to users about upcoming appointments, reducing the risk of missed appointments and cancellations. Integrating with popular payment systems like PayPal, Stripe, or Square would allow for seamless and secure payments, making the booking and payment process more efficient. The system could offer customized packages that allow users to book multiple services at once, saving them time and money. These are the features that will be enhanced in the future.

Chapter 8

PLAGIARISM REPORT

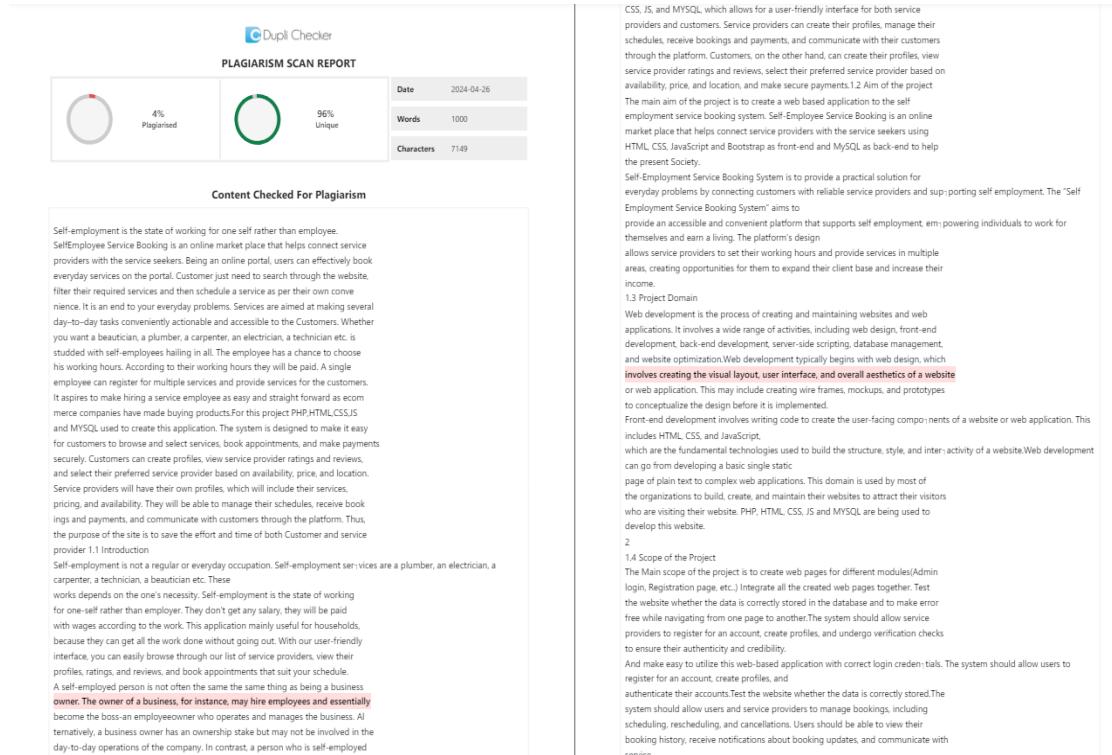


Figure 8.1: Plagiarism Report

Chapter 9

SOURCE CODE & POSTER

PRESENTATION

9.1 Source Code

```
1
2<?php include 'includes/connection.php';?>
3<?php include 'includes/navbar.php';?>
4
5<body>
6
7<section id="intro">
8<div class="carousel-item active">
9<div class="carousel-background"></div>
10<div class="carousel-container">
11<div class="carousel-content">
12<h2 class="font-color-white">Find & Provide Work Now more Easy Way</h2>
13
14</div>
15</div>
16</div>
17</section>
18
19
20<br><br>
21
22
23<section id="Job-Category">
24<div class="container">
25<h3 class="text-center">Choose Work Category </h3>
26<div class="vertical-space-30"></div>
27
28<div class="vertical-space-60"> </div>
29<div class="row">
30<div class="col-lg-3 col-md-6 max-width-50">
31<div class="box background-color-white-light">
32<div class="circle">
33
34</div>
35<h6>Education & Training </h6>
```

```

36
37 </div>
38 </div>
39 <div class="col-lg-3 col-md-6 max-width-50">
40 <div class="box background-color-white-light">
41 <div class="circle">
42 
43 </div>
44 <h6>Sales and Marketing </h6>
45
46 </div>
47 </div>
48 <div class="col-lg-3 col-md-6 max-width-50">
49 <div class="box background-color-white-light">
50 <div class="circle">
51 
52 </div>
53 <h6>Carpentry Work </h6>
54
55 </div>
56 </div>
57 <div class="col-lg-3 col-md-6 max-width-50">
58 <div class="box background-color-white-light">
59 <div class="circle">
60 
61 </div>
62 <h6>Construction Work </h6>
63
64 </div>
65 </div>
66 <div class="col-lg-3 col-md-6 max-width-50">
67 <div class="box background-color-white-light">
68 <div class="circle">
69 
70 </div>
71 <h6>Art & Design Work </h6>
72
73 </div>
74 </div>
75 <div class="col-lg-3 col-md-6 max-width-50">
76 <div class="box background-color-white-light">
77 <div class="circle">
78 
79 </div>
80 <h6>Web Development </h6>
81
82 </div>
83 </div>
84 <div class="col-lg-3 col-md-6 max-width-50 margin-left-18">
85 <div class="box background-color-white-light">

```

```

86 <div class="circle">
87 
88 </div>
89 <h6>Doctor Work</h6>
90
91 </div>
92 </div>
93 <div class="col-lg-3 col-md-6 max-width-50">
94 <div class="box background-color-white-light">
95 <div class="circle">
96 
97 </div>
98 <h6>Engineer / Architects </h6>
99
100 </div>
101 </div>
102 </div>
103 <div class="vertical-space-40"></div>
104 <a href="#" class="Brows-All-Category">Brows All Category </a>
105 </div>
106 <div class="vertical-space-85"></div>
107 </section>
108
109
110
111
112
113 <section id="Featuread-Company">
114 <div class="vertical-space-85"></div>
115 <div class="container text-center">
116 <h3 class="text-center">Featuread Works</h3>
117 <div class="vertical-space-30"></div>
118 <p class="max-width">Self-employed persons may be involved in a variety of occupations but generally  
are highly skilled at a particular kind of work.
119 </p>
120 <div class="row">
121 <div class="col-xs-12 col-sm-12 col-md-12">
122 <div class="owl-carousel Featuread-Company-carousel">
123 <a href="#" class="Featuread-Company-item">
124 <div class="media text-align-center medial">
125 
126 <div class="media-body text-left text-align-center">
127 <h6>Mshape Themes Limited.</h6>
128 <i class="material-icons">account_balance</i>
129 <span class="text">Mshape inc.</span>
130 <br />
131 <i class="material-icons">place</i>
132 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
133 <br />
134 <i class="material-icons">person</i>

```

```

135 <span class="text font-size font-color-orange">3 Open Position </span>
136 </div>
137 </div>
138 </a>
139 <a href="#" class="Featuread-Company-item">
140 <div class="media text-align-center medial">
141 
142 <div class="media-body text-left text-align-center">
143 <h6>Scalegoss Limited.</h6>
144 <i class="material-icons">account_balance </i>
145 <span class="text">Scalegoss inc.</span>
146 <br />
147 <i class=" material-icons">place </i>
148 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
149 <br />
150 <i class=" material-icons">person </i>
151 <span class="text font-size font-color-orange">3 Open Position </span>
152 </div>
153 </div>
154 </a>
155 <a href="#" class="Featuread-Company-item">
156 <div class="media text-align-center medial">
157 
158 <div class="media-body text-left text-align-center">
159 <h6>Jopitar Themes Limited.</h6>
160 <i class="material-icons">account_balance </i>
161 <span class="text">Jopitar inc.</span>
162 <br />
163 <i class=" material-icons">place </i>
164 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
165 <br />
166 <i class=" material-icons">person </i>
167 <span class="text font-size font-color-orange">3 Open Position </span>
168 </div>
169 </div>
170 </a>
171 <a href="#" class="Featuread-Company-item">
172 <div class="media text-align-center medial">
173 
174 <div class="media-body text-left text-align-center">
175 <h6>Supwin Themes Limited.</h6>
176 <i class="material-icons">account_balance </i>
177 <span class="text">Supwin inc.</span>
178 <br />
179 <i class=" material-icons">place </i>
180 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
181 <br />
182 <i class=" material-icons">person </i>
183 <span class="text font-size font-color-orange">3 Open Position </span>
184 </div>

```

```

185 </div>
186 </a>
187 <a href="#" class="Featuread-Company-item">
188 <div class="media text-align-center medial">
189 
190 <div class="media-body text-left text-align-center">
191 <h6>Mshape Themes Limited.</h6>
192 <i class="material-icons">account_balance</i>
193 <span class="text">Mshape inc.</span>
194 <br />
195 <i class=" material-icons">place</i>
196 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
197 <br />
198 <i class=" material-icons">person</i>
199 <span class="text font-size font-color-orange">3 Open Position </span>
200 </div>
201 </div>
202 </a>
203 <a href="#" class="Featuread-Company-item">
204 <div class="media text-align-center medial">
205 
206 <div class="media-body text-left text-align-center">
207 <h6>Scalegoss Themes Limited.</h6>
208 <i class="material-icons">account_balance</i>
209 <span class="text">Scalegoss inc.</span>
210 <br />
211 <i class=" material-icons">place</i>
212 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
213 <br />
214 <i class=" material-icons">person</i>
215 <span class="text font-size font-color-orange">3 Open Position </span>
216 </div>
217 </div>
218 </a>
219 <a href="#" class="Featuread-Company-item">
220 <div class="media text-align-center medial">
221 
222 <div class="media-body text-left text-align-center">
223 <h6>Mshape Themes Limited.</h6>
224 <i class="material-icons">account_balance</i>
225 <span class="text">Mshape inc.</span>
226 <br />
227 <i class=" material-icons">place</i>
228 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
229 <br />
230 <i class=" material-icons">person</i>
231 <span class="text font-size font-color-orange">3 Open Position </span>
232 </div>
233 </div>
234 </a>
```

```

235 <a href="#" class="Featuread-Company-item">
236 <div class="media text-align-center medial">
237 
238 <div class="media-body text-left text-align-center">
239 <h6>Scalegoss Themes Limited.</h6>
240 <i class="material-icons">account_balance</i>
241 <span class="text">Scalegoss inc.</span>
242 <br />
243 <i class=" material-icons">place</i>
244 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
245 <br />
246 <i class=" material-icons">person</i>
247 <span class="text font-size font-color-orange">3 Open Position </span>
248 </div>
249 </div>
250 </a>
251 <a href="#" class="Featuread-Company-item">
252 <div class="media text-align-center medial">
253 
254 <div class="media-body text-left text-align-center">
255 <h6>Jopitar Themes Limited.</h6>
256 <i class="material-icons">account_balance</i>
257 <span class="text">Jopitar inc.</span>
258 <br />
259 <i class=" material-icons">place</i>
260 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
261 <br />
262 <i class=" material-icons">person</i>
263 <span class="text font-size font-color-orange">3 Open Position </span>
264 </div>
265 </div>
266 </a>
267 <a href="#" class="Featuread-Company-item">
268 <div class="media text-align-center medial">
269 
270 <div class="media-body text-left text-align-center">
271 <h6>Supwin Themes Limited.</h6>
272 <i class="material-icons">account_balance</i>
273 <span class="text">Supwin inc.</span>
274 <br />
275 <i class=" material-icons">place</i>
276 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
277 <br />
278 <i class=" material-icons">person</i>
279 <span class="text font-size font-color-orange">3 Open Position </span>
280 </div>
281 </div>
282 </a>
283 <a href="#" class="Featuread-Company-item">
284 <div class="media text-align-center medial">

```

```

285 
286 <div class="media-body text-left text-align-center">
287 <h6>Mshape Themes Limited.</h6>
288 <i class="material-icons">account_balance </i>
289 <span class="text">Mshape inc.</span>
290 <br />
291 <i class=" material-icons">place </i>
292 <span class="text font-size">11907 Doyle Cape Cydneyview</span>
293 <br />
294 <i class=" material-icons">person </i>
295 <span class="text font-size font-color-orange">3 Open Position </span>
296 </div>
297 </div>
298 </a>
299 </div>
300 </div>
301 </div>
302 <div class="vertical-space-85"></div>
303 </div>
304 </section>
305
306
307 <?php include 'includes/footer.php';?>
```

9.2 Poster Presentation



SELF-EMPLOYMENT SERVICE BOOKING SYSTEM

Department of Computer Science & Engineering
School of Computing
10214CS602 – MINOR PROJECT - II
WINTER SEMESTER 2023-2024

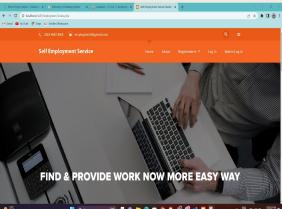
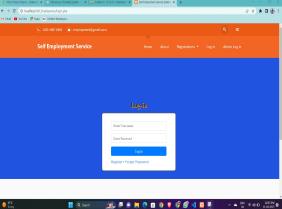
ABSTRACT <p>Self-employment is the state of working for oneself rather than employee. Self-Employee Service booking is an online marketplace that helps connect service providers with the service seekers. Being an online platform, it is very effective to book everyday services from the comfort of your home. Customer just need to search through the website, filter their required services, and then schedule a service as per their convenience. It is an end to your everyday problems. Services are aimed at making several day-to-day tasks conveniently actionable and accessible to the users. Whether you want a beautician, a plumber, a carpenter, an electrician, a technician etc. is studied with self-employed having in all. The employee has a chance to choose his working hours. According to their working hours they will be paid. A single employee can register for multiple services and provide services for the customer. It aspires to make hiring a service employee as easy and straightforward as e-commerce companies have made buying products.</p>	INTRODUCTION <p>Self-employment is not a regular or everyday occupation. Self-employment Services are a plumber, an electrician, a carpenter, a technician, a beautician etc. These services depend on the one's necessity. Self-employment is the state of working for one-self rather than employer. They don't get any salary they will be paid with wages according to the work. This application mainly useful for households, because they can get all the work done without going out.</p>	RESULTS <p>Thus, This application is helpful for providing interaction between user and employee without any intermediaries. The self-employment service booking system project is a valuable tool for individuals who offer their services as independent contractors. The system provides an easy and convenient way for service providers to manage their schedules, accept and schedule bookings, and communicate with their clients.</p>	STANDARDS AND POLICIES <p>Service level agreements (SLAs): The system should have well-defined SLAs that outline the expected level of service, availability, and response times, as well as escalation procedures in case of system issues or downtime.</p> <p>Testing and quality assurance policies: The system should have a comprehensive testing and quality assurance policy in place to ensure that it meets functional, performance, and security requirements.</p> <p>Accessibility standards: The system should follow accessibility standards, such as the Web Content Accessibility Guidelines (WCAG), to ensure that it can be used by people with disabilities.</p>
Output:  			
CONCLUSIONS <p>This web application is mainly useful for those who are skilled but they are unemployed. Admin is the main person who decides to work on this web application. The main intention behind this web application is to provide employment who are skilled and looking for an employment. This application is helpful for providing interaction between user and employee without any intermediaries. The self-employment service booking system project is a valuable tool for individuals who offer their services as independent contractors. The system provides an easy and convenient way for service providers to manage their schedules, accept and schedule bookings, and communicate with their clients.</p>			
ACKNOWLEDGEMENT <ol style="list-style-type: none"> 1. Project Supervisor: Dr. Anandhi. M, E, Ph. D. 2. Project supervisor Contact No: 9842344492 3. Project supervisor Mail ID: dranandhi@veltech.edu.in 			

Figure 9.1: Poster Presentation

References

- [1] Ali, M. A., Mohamed, A. R. (2021). A review of self-employment service booking platforms: Current status, challenges and opportunities. *Computers in Human Behavior*, 117, 106671.
- [2] Ge, L., Zhang, X. (2021). Research on the influencing factors of user trust in self-employment service booking platforms. *Journal of Service Theory and Practice*, 31(2), 198-215.
- [3] S. S. S. H. Perera, T. I. Fernando, and H. M. N. Dilum Bandara (2021). An Online Self-Employment Booking System with Machine Learning Capability published in the Proceedings of the 2021 6th International Conference on Computer Science and Information Technology.
- [4] Peng, S., Zhu, L., Wang, Y. (2021). Artificial intelligence and self-employment service booking: A review and future research agenda. *International Journal of Information Management*, 57, 102325.
- [5] Wang, X., Zhu, Z., Zhang, W. (2021). Understanding customers continuous use intention of self-employment service booking platforms: A dual-process model. *International Journal of Hospitality Management*, 97, 102898.
- [6] Liu, Y., Xiong, Z., Chen, X. (2022). Matching algorithm design in self-employment service booking platforms. *Journal of Systems Science and Information*, 10(1), 1-13.
- [7] Alam, M. R., Reza, S. M. S., Hossain, M. S. (2022). A smart mobile application for self-employment service booking and management. *Journal of Ambient Intelligence and Humanized Computing*, 13(1), 331-348.
- [8] Yoon, J., Kim, M., Jeon, H. (2021). A comparison of user behavior between online-to-offline and offline-to-online service platforms: Evidence from a self-employment service platform. *Journal of Business Research*, 134, 491-503.
- [9] Kim, M., Lee, Y., Jeon, H. (2021). Understanding the determinants of user retention in self-employment service platforms: A comparison of online-tooffline and offline-to-online service platforms. *Journal of Business Research*, 129, 783-793.

- [10] Wibowo, W., Japarianto, E., Pratama, Y. (2021). An intelligent selfemployment service booking system based on customer preferences. International Journal of Advanced Science and Technology, 30(7s), 1479-1487.