A
PROJECT REPORT ON

Household Services Management System

SUBMITTED IN
PARTIAL FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

BY

Aman Gupta
Sohen Mondal
Pooja Bhandwalkar
Pamidimarri Vinay Kumar

UNDER THE GUIDENCE OF

Mrs. Pooja Jaiswal

AT

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE.

CERTIFICATE

This is to certify that the project

Household Services Management System

Has been submitted by

Aman Gupta
Sohen Mondal
Pooja Bhanwalkar
Pamidimarri Vinay Kumar

In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC AUG2024)** as prescribed by The **CDAC** ACTS, PUNE.

Place: Pune

Date: 11-FEB-2025

Mrs.Pooja Jaiswal
Project Guide

Mr.Yogesh Kolhe
Alumni Mentor

ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere gratitude to **Mr. Nitin Kudal(CEO)** and **Mr. Yogesh Kolhe(Course Coordinator)** for their invaluable guidance and support throughout this project. Their insights, encouragement, and constructive feedback at every stage have played a crucial role in shaping our work.

A special thanks to the entire faculty and staff of **Sunbeam Institute of Information Technology, Pune**, whose support and encouragement created a great learning environment for us.

This project would not have been possible without the help, patience, and motivation of everyone involved. We truly appreciate all the time and effort that went into making this a success.

Aman Gupta

Sohen Mondal

Pooja Bhanwalkar

Pamidimarri Vinay Kumar

PG-DAC AUG2024

SIIT Pune

ABSTRACT

The Household Services Platform is a web-based application that connects users with service providers for various household needs, such as plumbing, electrical work, cleaning, and more. Built with React for a smooth and responsive frontend, Java for a robust backend, and MySQL for secure data storage, the platform ensures a seamless experience. Users can easily book services, manage profiles, make secure payments, and leave reviews. Service providers can manage bookings, while admins oversee the system through a dedicated dashboard. This platform simplifies the process of finding and hiring trusted household service professionals.

INDEX

LIST OF TABLES

LIST OF FIGURES

1.Introduction

In today's fast-paced world, managing household tasks can be challenging. The Household Services Platform addresses this by offering a user-friendly web application that connects individuals with professional service providers for various home needs, such as plumbing, electrical work, and cleaning. Built with a responsive React frontend, a robust Java backend, and secure MySQL data storage, the platform ensures a seamless experience for users. Users can easily book services, manage profiles, make secure payments, and leave reviews, while service providers can efficiently handle bookings. Administrators maintain oversight through a dedicated dashboard. This integrated approach simplifies the process of finding and hiring trusted household service professionals, enhancing convenience and reliability for all parties involved.

2.Product Overview and Summary

2.1.Purpose

The Household Services Platform is designed to make it easy for users to connect with trusted professionals for various home tasks, such as plumbing, electrical repairs, and cleaning. By using React for the user interface, Spring Boot for the backend, and MySQL for data storage, the platform ensures a smooth and efficient experience.

Key Features:

- User-Friendly Booking: Users can quickly find and schedule services that fit their needs.
- Profile Management: Both users and service providers can manage their profiles and track service history.
- Secure Payments: The platform offers safe and straightforward payment options.
- Feedback System: After a service is completed, users can leave reviews, helping maintain quality and trust.

By integrating these technologies, the platform aims to provide a reliable and convenient solution for managing household services.

2.2.Scope

The project encompasses the development of the following components:

1. **User Interface (UI):** Designing an intuitive and responsive UI with React to facilitate easy navigation and service booking.
2. **Backend Development:** Implementing a robust backend using Spring Boot to handle user authentication, service listings, booking management, and communication between users and service providers.
3. **Database Management:** Utilizing MySQL for secure storage and retrieval of data, including user profiles, service provider details, service categories, bookings, and transaction records.
4. **Security Measures:** Implementing security protocols to protect user data, ensure secure payment processing, and maintain overall platform integrity.

By focusing on these areas, the Household Services Platform aims to provide a reliable and convenient solution for managing household services, enhancing the overall user experience.

2.3. User Class and Characteristics

User Class	Characteristics
Admin	Admin can see all the information of the customers and service providers. He can even delete a service providers profile if he finds it faulty.
Customer	Customer can register if he is a new user and can update his profile. They can schedule a service from the service provider and choose payment option after the service is done by the service provider. They can then give feedback only after payment is done.
Service Providers	Service provider is the main vendor that provides various service. He also has a dashboard that provides him with information about total customers requesting different services, his work rating, total amount to collect

2.4. Design and Implementation Constraints

Design Constraints:

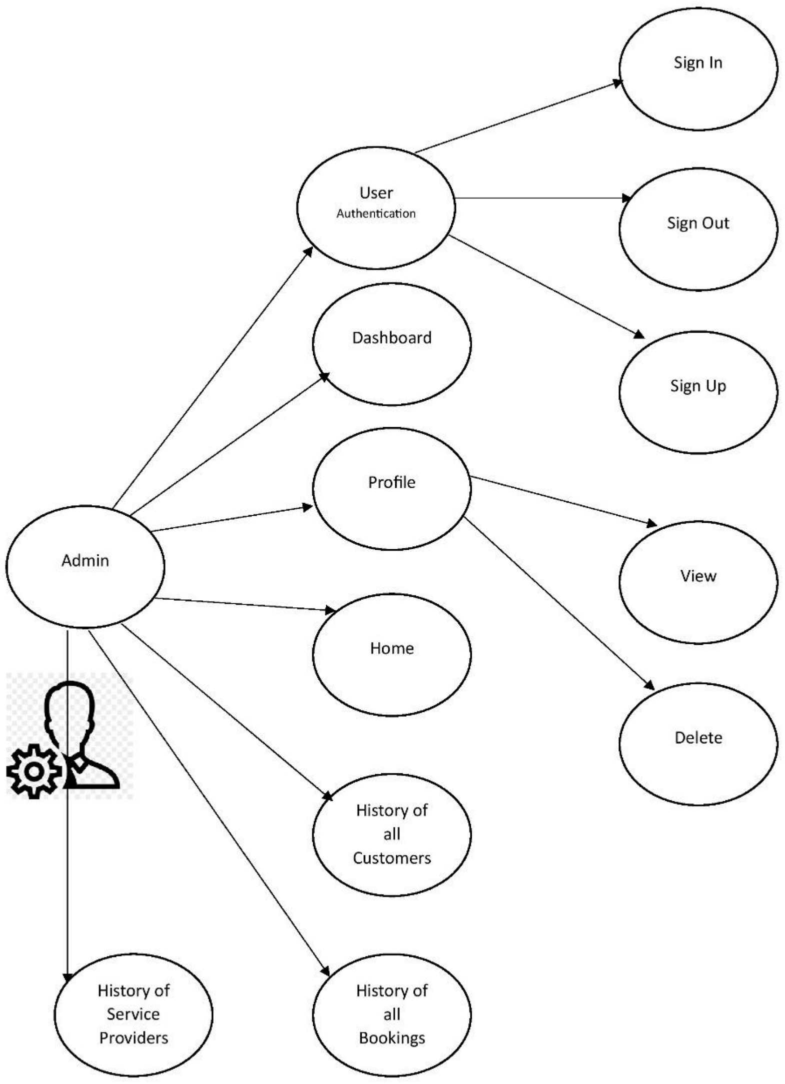
- **Technology Compatibility:** Ensure that React (frontend), Spring Boot (backend), and MySQL (database) integrate seamlessly to facilitate smooth data flow and communication.
- **User Experience (UX):** Design an intuitive and responsive interface with React to provide users with a seamless experience across various devices and screen sizes.
- **Security:** Implement robust security measures, including user authentication and authorization, to protect sensitive data and maintain user trust.

Implementation Constraints:

- **Database Design:** Structure the MySQL database efficiently to handle relationships between users, service providers, and services, ensuring data integrity and optimal performance.
- **API Development:** Develop RESTful APIs using Spring Boot to enable effective communication between the frontend and backend, ensuring they are well-documented and adhere to industry standards.
- **Performance Optimization:** Optimize both frontend and backend code to reduce latency and improve load times, enhancing the overall user experience.

3.Requirements

3.1 Functional Requirements



Admin Flow

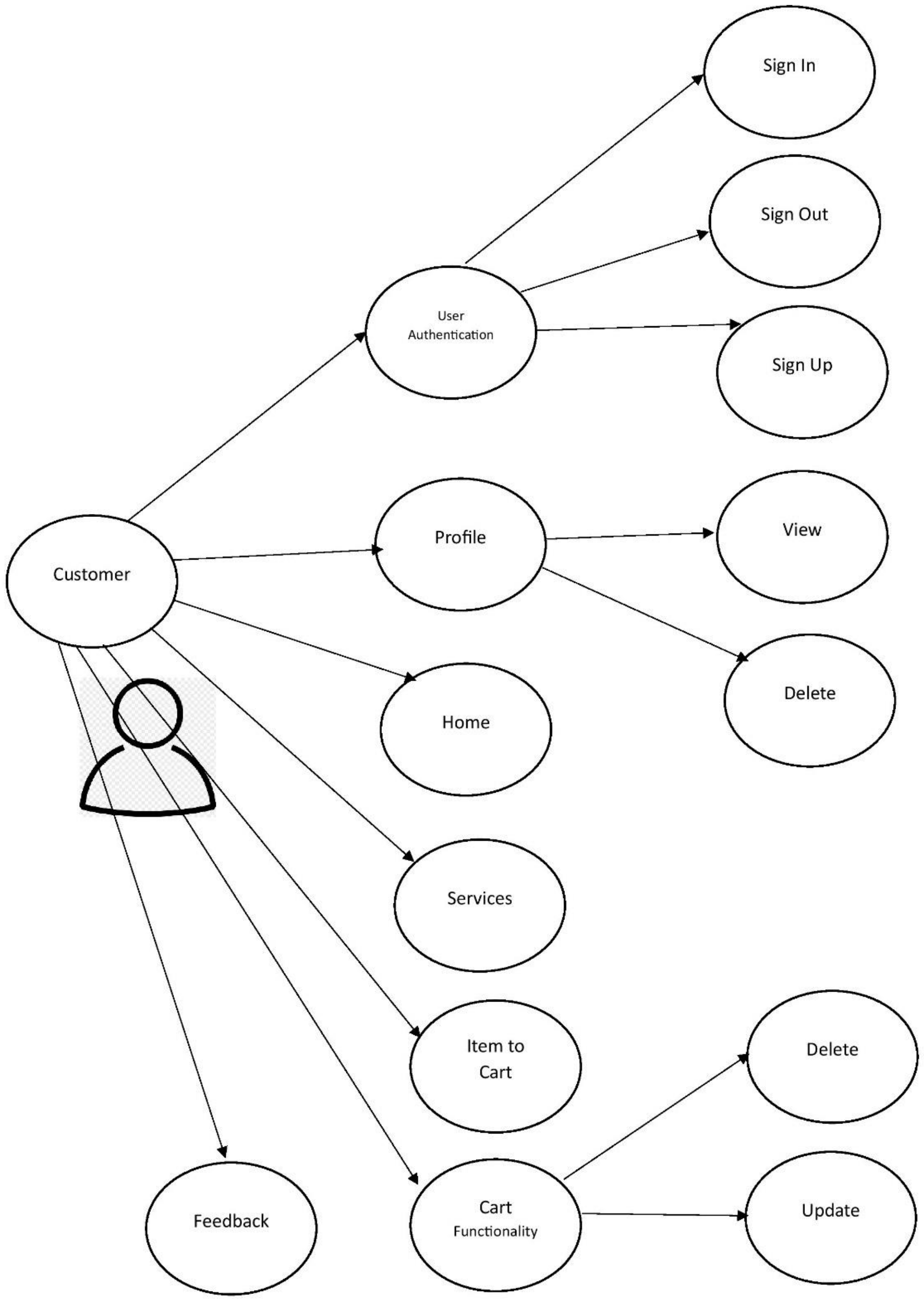
3.1.1 User Case for Administrator

3.1.1.1 Home Page

- **Objective**: Admin access to system functionalities
- **Features**: Sign in to access the functionalities.

3.1.1.2 Admin Home Page

- **Objective:** Oversee the management of customers and service providers
- **Features:**
    - View all Types of Customers
    - View all Types of Service Providers.
    - View all Types of Services requested by the customer.
    - Delete a Service Provider.



Customer Flow

3.1.2 User Case for Customer

3.1.2.1 Customer Login Page

- **Objective**: To authorize right customer with his login credentials
- **Features**: Sign in to access the functionalities.

3.1.2.2 Customer Home Page

- **Objective**: Access to all the services provided.
- **Features**: Customer can select a service which would redirect him to the cart page.

3.1.2.3 Add to Cart Page

- **Objective**: Access to all the services selected by the customer.
- **Features**: Customer can view all the services selected and can schedule the time by clicking on the service box.

3.1.2.4 Schedule Time Drop down Menu

- **Objective**: Access to all time slots that a customer can select.
- **Features**: Show all the time a service provider schedules his service.

3.1.2.5 Cart Booking

- **Objective**: Redirects to all the service selected by customer.
- **Features**:
  - Delete a service

    Customer can delete a selected service.
  - Confirm all orders

    Customer can confirm their orders.

3.1.2.6 Confirm Booking Page

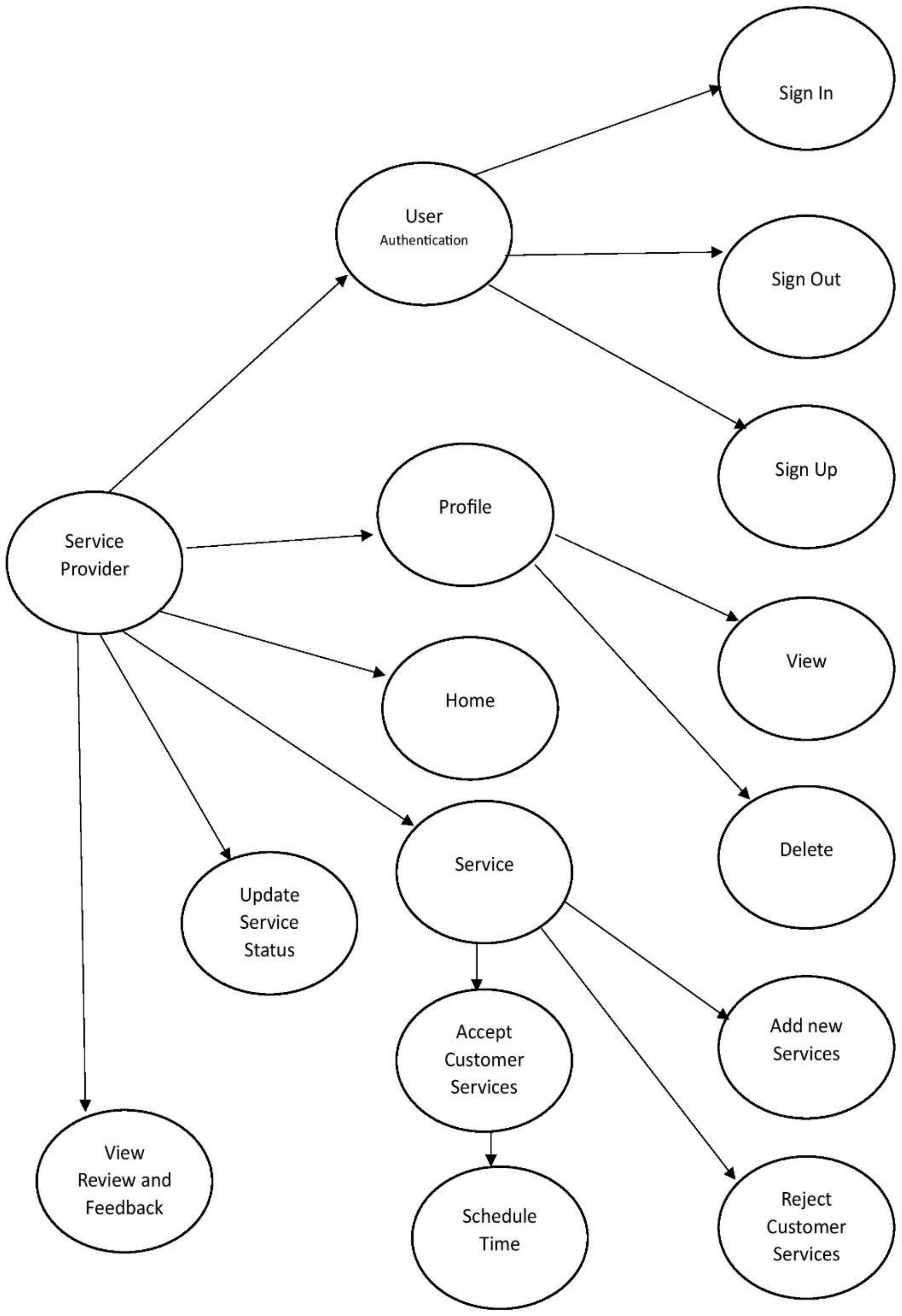
- **Objective**: Shows all the services selected by the customer.
- **Features**: Redirects him/here to the payment page after the service is done.

3.1.2.7 Payment Page

- **Objective**: Shows all Payment options.
- **Features**: Customer can make payment in any of the above options.

3.1.2.6 Feedback Page

- **Objective**: Shows Feedback Page
- **Features**: He can leave their feedback for Service Provider. Redirects to Customer Home Page.



Service Provider Flow

3.1.3 User Case for Service Provider

3.1.3.1 Service Provider Login

- **Objective**: To authorize right service provider with his/her login credentials.
- **Features**: Sign in to access the home page.

3.1.3.2 Service Provider Home page

- **Objective**: To show all the service requested by the customer.
- **Features**: Select a request to be redirected to the dashboard.

3.1.3.3 Dashboard

- **Objective**: Show total number of job done, money earned, rating , reviews.
- **Features**: Just hove rover the images to see the same.

3.1.3.4 Service Provider Profile Page

- **Objective**: Help Service provider to update his information like area of service, phone number.
- **Features**: Update Service Provider information.

3.2 Non-Functional Requirements

3.2.1 Interface

User interfaces must be intuitive and user-friendly.

3.2.2 Constraints and Performance

- Number of Concurrent Users: The system should handle at least 1000 transactions/inquiries per second.
- System Resilience: The application should be resilient to temporary server failure.

3.2.3 Hardware and Software Requirements

Hardware- Intel Core i5 or higher (or AMD equivalent), 8 GB RAM, 512 GB SSD or larger.

Software-

Operating Systems: MS Windows 13, Ubuntu 22.04.

Database: MySQL.

Server: Embedded Tomcat.

Browsers: Compatible with modern web browsers.

3.2.4 System Design

Front-End: Developed using React.js.

Back-End: Built with Spring Boot for server-side logic.

Database: MySQL for storing user data, orders, and other system information.

Server: Embedded Tomcat for hosting the application.

4.Project Design

4.1 Data Model

4.1.1 Database Design

Table 1: User

Field	Type	Null	Key and auto-increment	Default
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	dateTime(6)	Yes		NULL
address	varchar(255)	Yes		NULL
email	varchar(255)	Yes		NULL
name	varchar(255)	Yes		NULL
password	varchar(255)	Yes		NULL
phone_no	varchar(255)	Yes		NULL
role	Enum	Yes		NULL

Table 2: Booking

Field	Type	Null	Key and auto-increment	Default
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
booking_slot	int	No		NULL
date	date	Yes		NULL
status	varchar(6)	Yes		NULL

customer_id	bigint	Yes	MUL	NULL
payment_id	bigint	Yes	MUL	NULL
serviceprovider_id	bigint	Yes	MUL	NULL
Services_id	bigint	Yes	UNI	NULL

Table 3: Feedback

Field	**Type**	**Null**	**Key and auto-increment**	**Default**
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
content	Varchar(225)	Yes		NULL
rating	int	No		NULL
booking_id	bigint	Yes	UNI	NULL
customer_id	bigint	Yes	MUL	NULL
serviceprovider_id	bigint	Yes	MUL	NULL

Table 4: Payments

Field	**Type**	**Null**	**Key and auto-increment**	**Default**
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
day	datetime(6)	Yes		NULL
payment_mode	tinyint	Yes		NULL

Status	varchar(255)	Yes		NULL
total_amt	double	No		NULL
customer_id	bigint	Yes	MUL	NULL
serviceprovider_id	bigint	Yes	MUL	NULL

Table 5: Service_provider

Field	**Type**	**Null**	**Key and auto-increment**	**Default**
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
earning	double	No		NULL
exp	int	No		NULL
customer_id	bigint	Yes	MUL	NULL
sp_id	bigint	No	UNI	NULL

Table 6: Services

Field	**Type**	**Null**	**Key and auto-increment**	**Default**
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
name	varchar(255)	Yes		NULL
price	double	No		NULL
status	varchar(255)	Yes		NULL

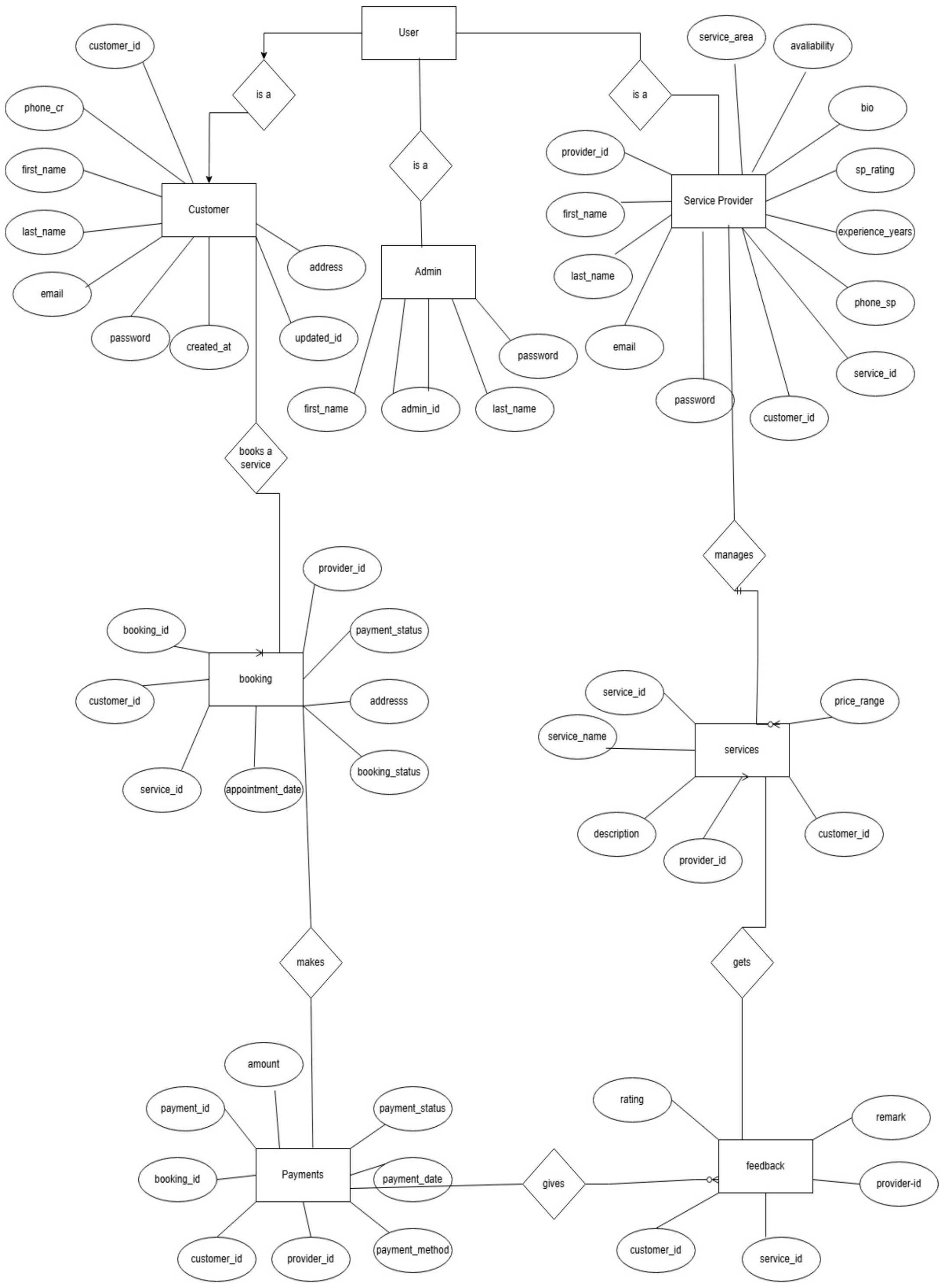
service_provider_id	bigint	yes	MUL	NULL

Table 7: Address

Field	**Type**	**Null**	**Key and auto-increment**	**Default**
id	bigint	No	Prim / auto-increment	NULL
created_on	date	Yes		NULL
updated_on	datetime(6)	Yes		NULL
adr_line1	varchar(100)	Yes		NULL
adr_line2	varchar(100)	Yes		NULL
city	varchar(100)	Yes		NULL
country	varchar(100)	Yes		NULL
state	varchar(100)	Yes		NULL
zip_code	varchar(100)	Yes		NULL

4.2 Process Model

4.2.1 ER Diagram



CODING STANDARDS IMPLEMENTED

Naming and Capitalization:

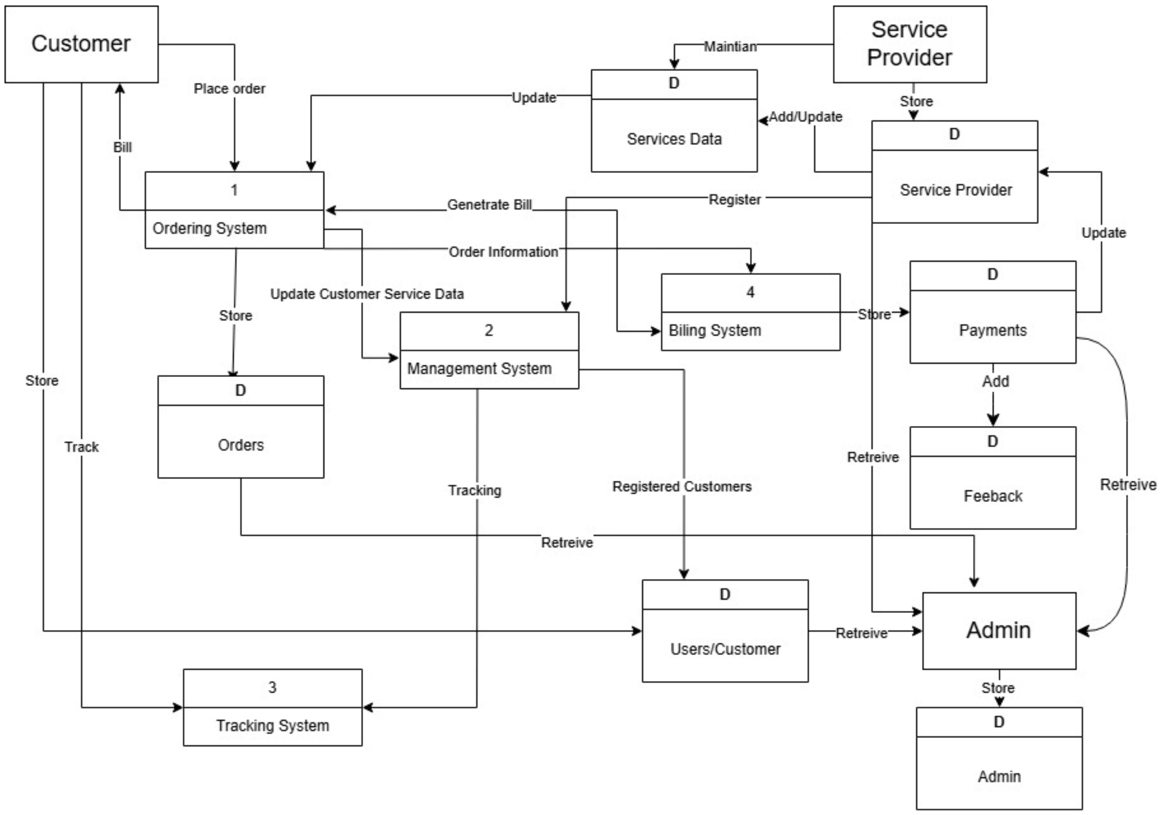
Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	User, Service_provider, Services,Bookings	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	findByName, existsByNameAndServiceprovider	Method should use verbs or verb phrases.
Parameter	Camel	name, provider,id	Use descriptive parameter names. Parameter names should descriptive enough that the name of the parameter and its type can be used to determine its meaning most scenarios.
Annotation	Pascal	SpringBootApplication	Use @ at start of annotation
DTOs	Camel	BaseDTO, ApiResponseDTO,BookingDTO	Use to transfer data between the processes

Interface	Pascall with "I" prefix	ServiceProviderDao	Do not use the **'_'** sign
Exception Class	Pascal with "Exception suffix"	ResourceNotFoundException	

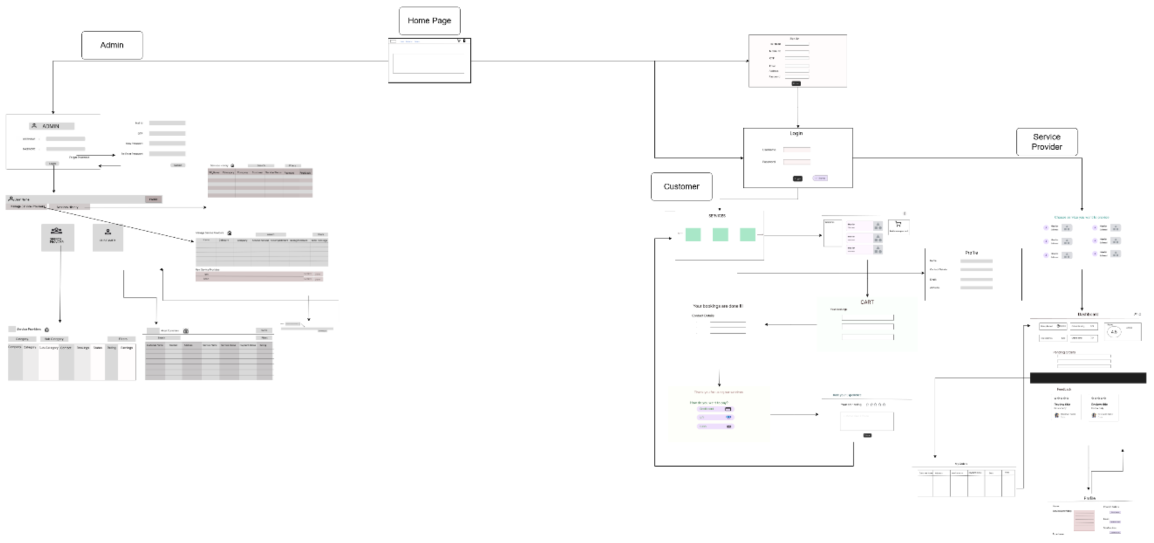
4.2.2 Data Flow Diagram and Web Diagram

Data Flow Diagram

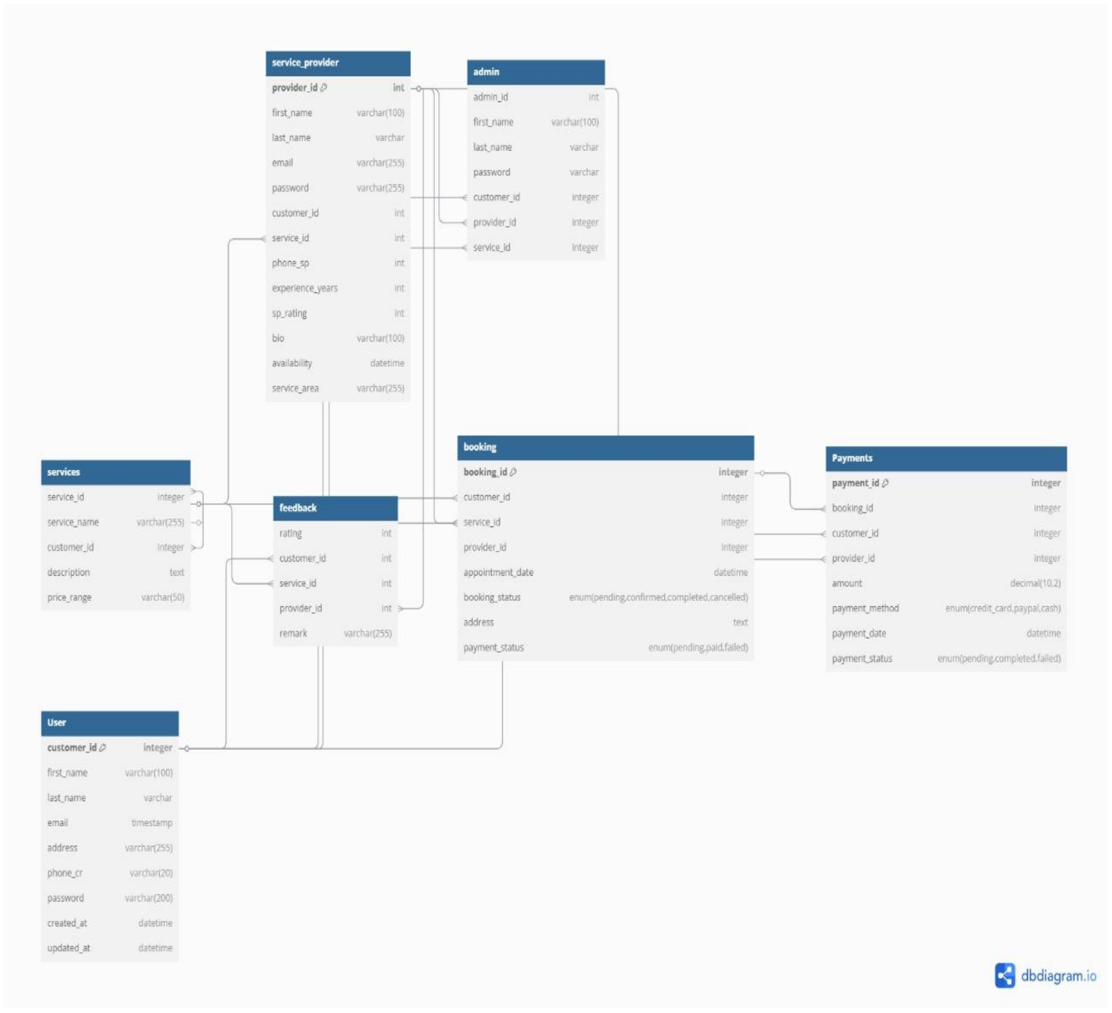


Data Flow Diagram

Web Diagram


Home Page
Admin
Customer
Service Provider
ADMIN
Login
CART
Profile
Dashboard

<u>Class Diagram</u>



5.Test Report

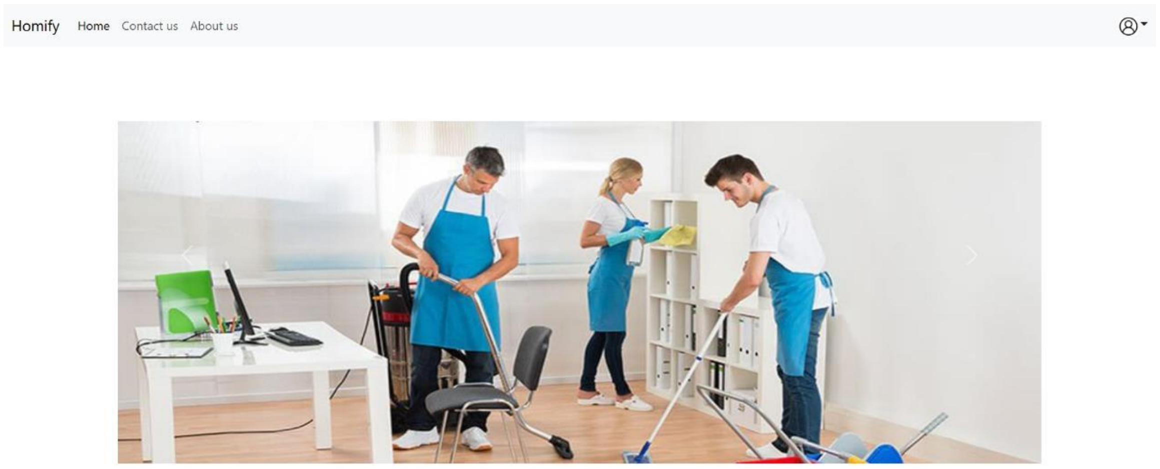
SR-No	Test Case	Expected Result	Actual Result	Error Message
1	Login Page	Login successfully	OK	Nothing
2	Sign Up Page	Redirect to login page	OK	Nothing
3	Admin-Home Page	Fetch customers and service providers details	OK	
4	Admin-Service Provider Page	Fetch service providers	OK	Failed to fetch service provider list
5	Admin-Booking Page	Fetch all booking details	OK	Failed to fetch booking list
6	Customer-Home page	Fetch all customer details	OK	Failed to fetch customer details
7	Customer-AddToCart	Fetch all customer booking	OK	Nothing
8	Customer-Cart	Fetch all customer to edit the order	OK	Deletion not performed

9	Customer-Confirmation	Confirm Customer orders	OK	Nothing
10	Customer-Feedback	Feedback of service	OK	Nothing
11	Customer-Payment	Choosing Payment option	OK	Payment not successful
12	Service Provider-Home Page	Fetch all service of customers	OK	Nothing
13	Service Provider-Dashboard Page	Fetch information about total money earned, customer reviews	OK	Nothing
14	Service Provider-Profile Page	Update User Profile	OK	Profile not Updated

6.Project Screenshots

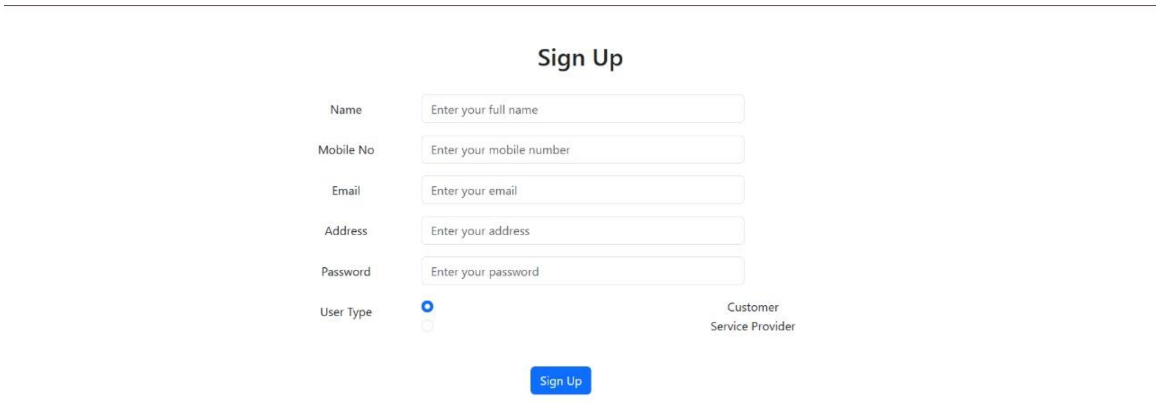
Home Page

Url - http://localhost:3000/



Sign in Page

Url - http://localhost:3000/signup



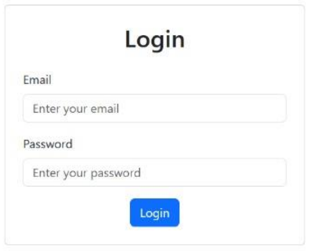
Sign Up

Name	Enter your full name
Mobile No	Enter your mobile number
Email	Enter your email
Address	Enter your address
Password	Enter your password
User Type	[✓ Customer] Service Provider

Sign Up

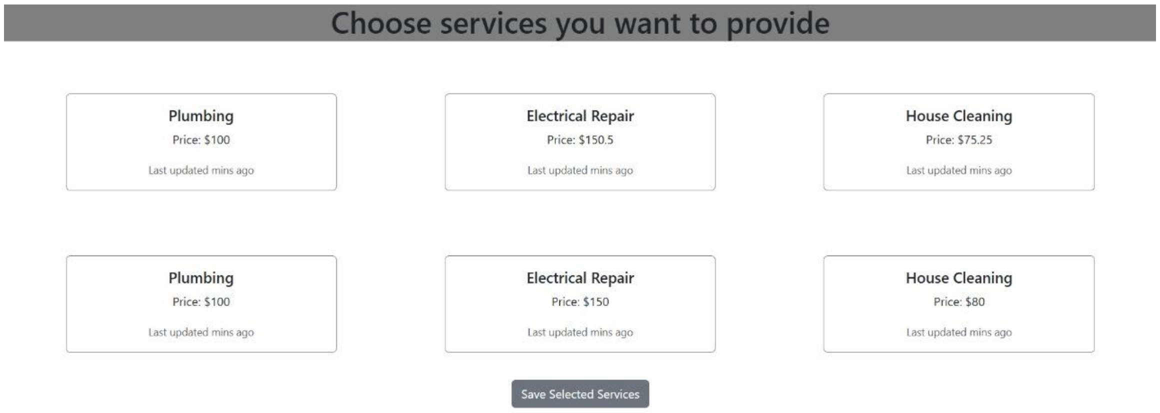
Common Login Page

Url- http://localhost:3000/login?



Service Provider- Service Page

Url- http://localhost:3000/servicelist?


Choose services you want to provide

Plumbing
Price: $100
Last updated mins ago

Electrical Repair
Price: $150.5
Last updated mins ago

House Cleaning
Price: $75.25
Last updated mins ago

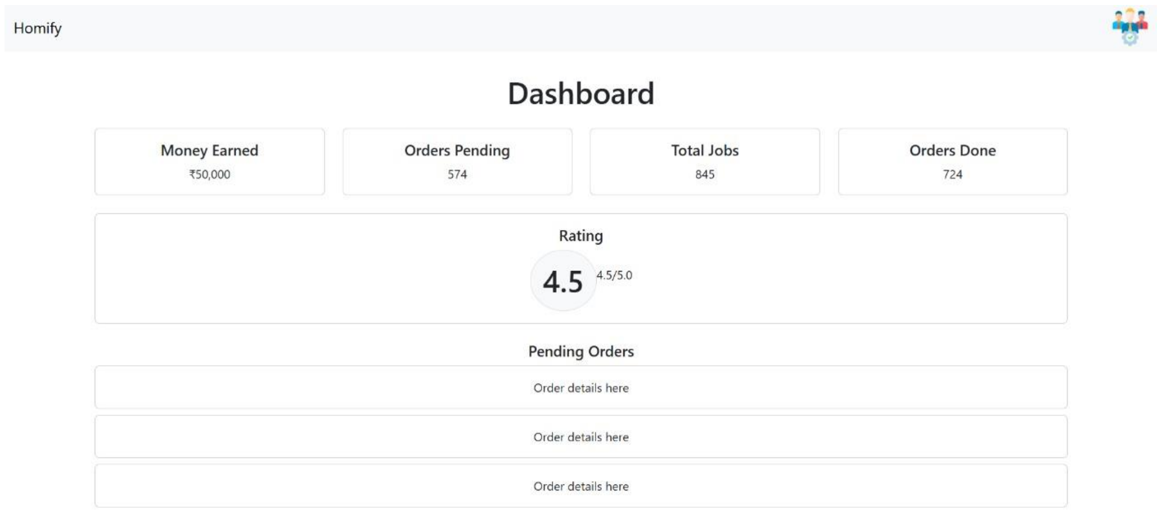
Plumbing
Price: $100
Last updated mins ago

Electrical Repair
Price: $150
Last updated mins ago

House Cleaning
Price: $80
Last updated mins ago

Service Provider-Dashboard

Url- http://localhost:3000/dashboard


Homify

Dashboard

Money Earned	Orders Pending	Total Jobs	Orders Done
₹50,000	574	845	724

Rating

4.5 4.5/5.0

Pending Orders

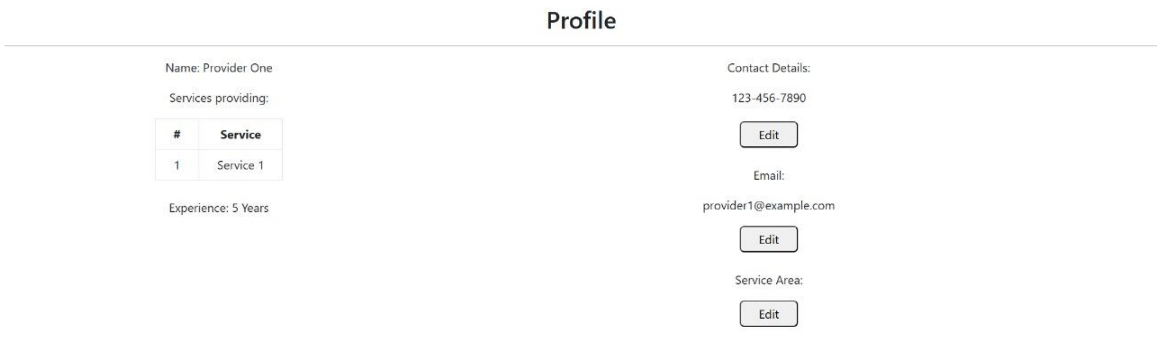
Order details here

Order details here

Order details here

Service Provider -Profile Page

Url - http://localhost:3000/spprofile


Profile

Name: Provider One

Services providing:

#	Service
1	Service 1

Experience: 5 Years

Contact Details:

123-456-7890

Edit

Email:

provider1@example.com

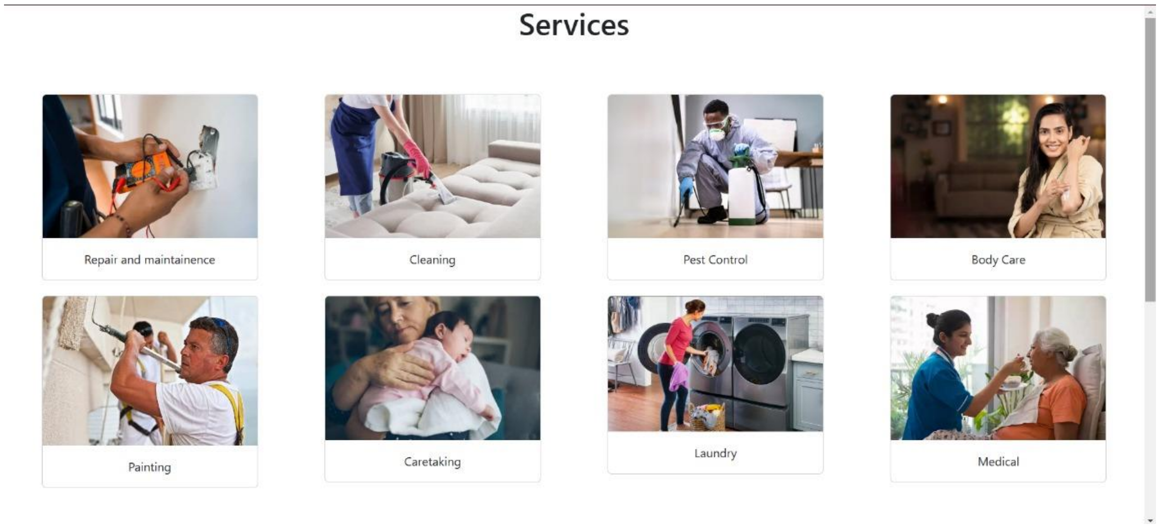
Edit

Service Area:

Edit

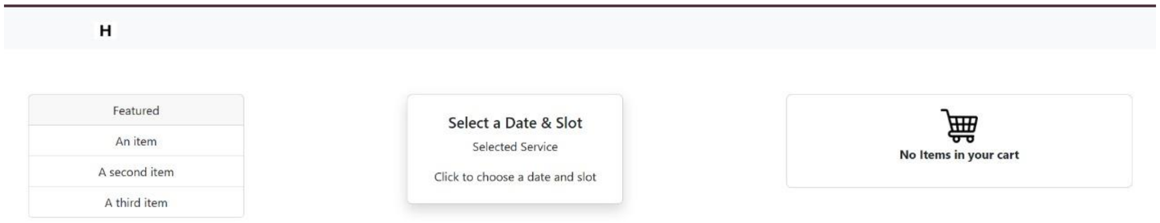
Customer- Services Page

Url - http://localhost:3000/services?



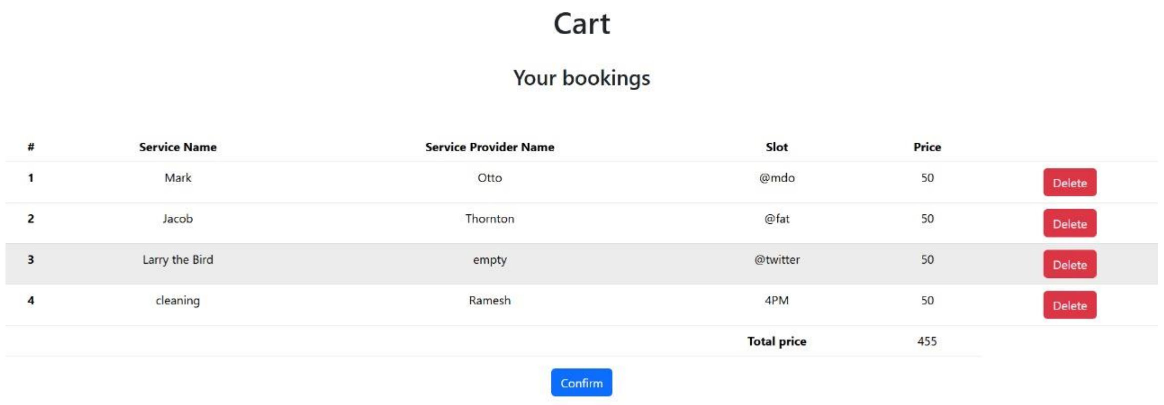
Customer – Cart Page

Url - http://localhost:3000/addtocart



Customer – Customer Bookings Page

Url - http://localhost:3000/cart



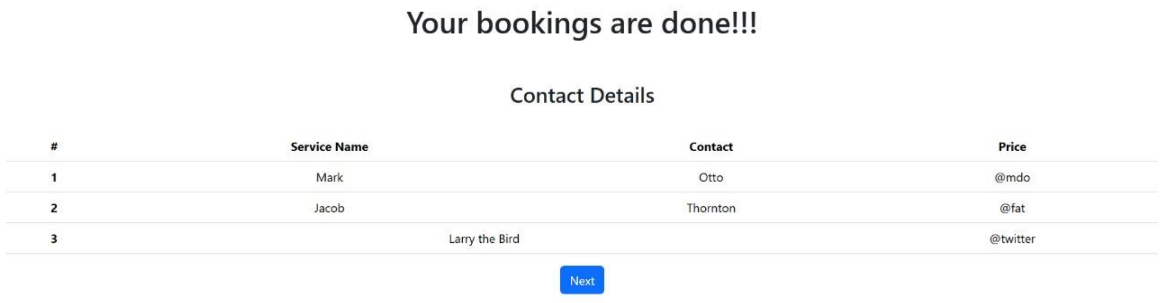
Cart

Your bookings

#	Service Name	Service Provider Name	Slot	Price	
1	Mark	Otto	@mdo	50	Delete
2	Jacob	Thornton	@fat	50	Delete
3	Larry the Bird	empty	@twitter	50	Delete
4	cleaning	Ramesh	4PM	50	Delete
			Total price	455	

Customer- Confirm Booking Page

Url - http://localhost:3000/confirmation



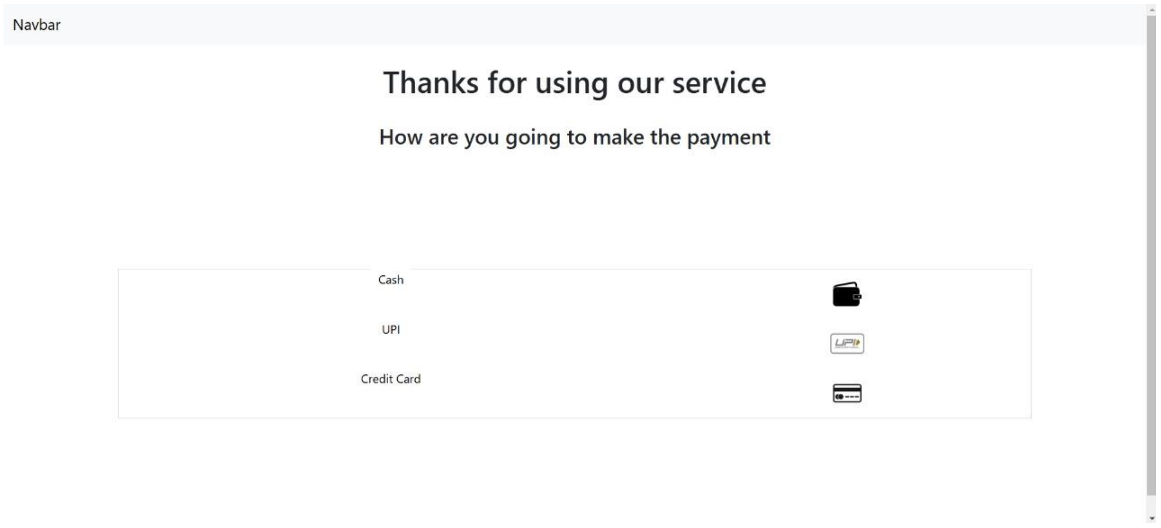
Your bookings are done!!!

Contact Details

#	Service Name	Contact	Price
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Customer – Payment

Url - http://localhost:3000/payment


Navbar

Thanks for using our service

How are you going to make the payment

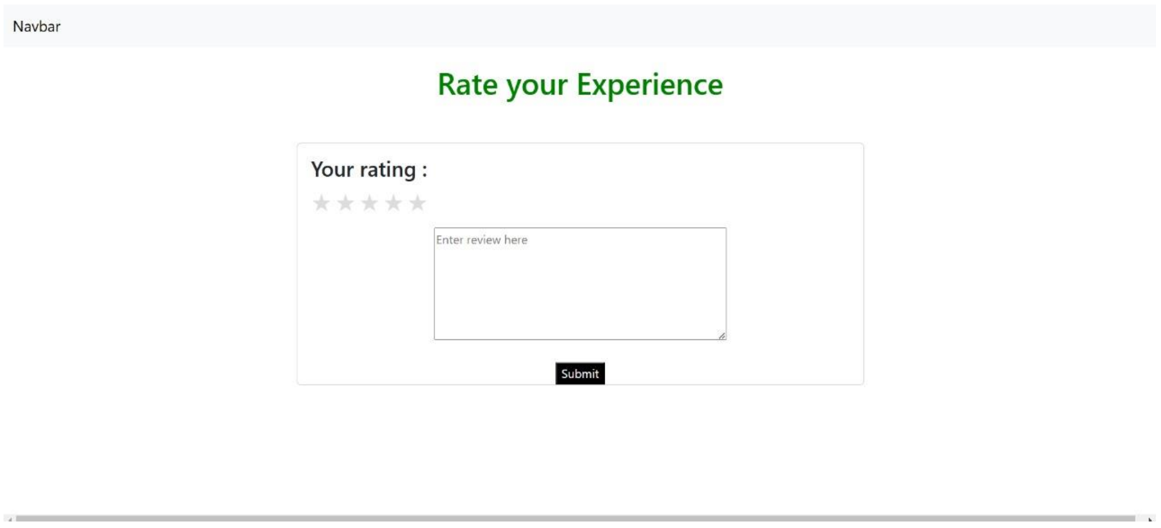
Cash

UPI

Credit Card

Customer - Feedback

Url - http://localhost:3000/feedback


Navbar

Rate your Experience

Your rating :

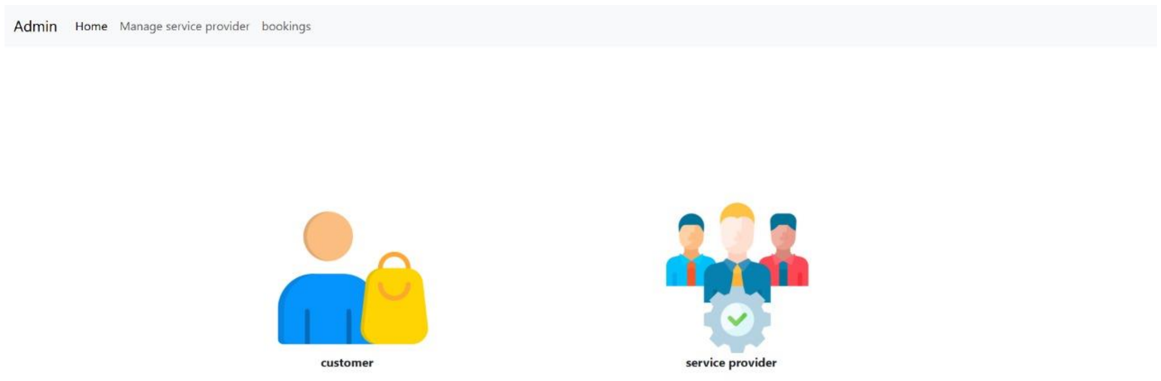
★★★★★

Enter review here

Submit

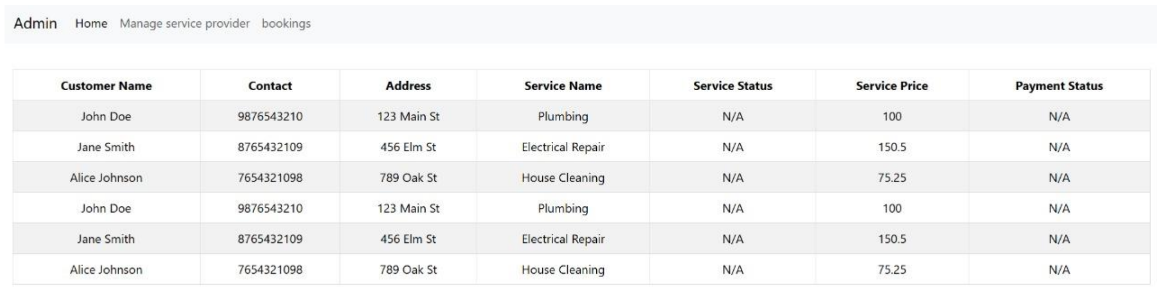
Admin- Admin Home Page

Url - http://localhost:3000/adminhome



Admin- Customer Tab Page

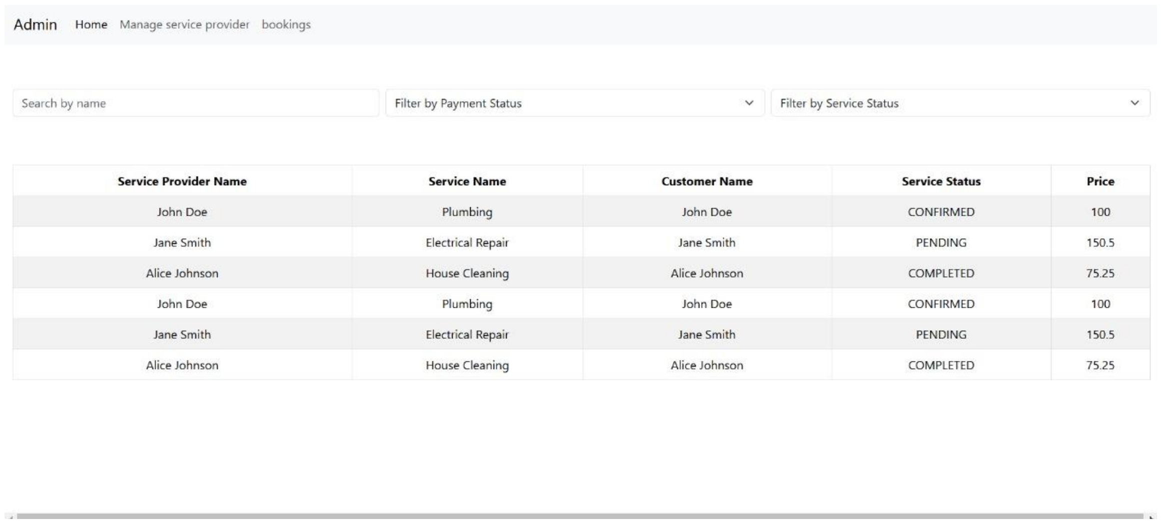
Url - http://localhost:3000/customer



Customer Name	Contact	Address	Service Name	Service Status	Service Price	Payment Status
John Doe	9876543210	123 Main St	Plumbing	N/A	100	N/A
Jane Smith	8765432109	456 Elm St	Electrical Repair	N/A	150.5	N/A
Alice Johnson	7654321098	789 Oak St	House Cleaning	N/A	75.25	N/A
John Doe	9876543210	123 Main St	Plumbing	N/A	100	N/A
Jane Smith	8765432109	456 Elm St	Electrical Repair	N/A	150.5	N/A
Alice Johnson	7654321098	789 Oak St	House Cleaning	N/A	75.25	N/A

Admin- Manage Service Provider Page

Url - http://localhost:3000/serviceprovider



Service Provider Name	Service Name	Customer Name	Service Status	Price
John Doe	Plumbing	John Doe	CONFIRMED	100
Jane Smith	Electrical Repair	Jane Smith	PENDING	150.5
Alice Johnson	House Cleaning	Alice Johnson	COMPLETED	75.25
John Doe	Plumbing	John Doe	CONFIRMED	100
Jane Smith	Electrical Repair	Jane Smith	PENDING	150.5
Alice Johnson	House Cleaning	Alice Johnson	COMPLETED	75.25

Conclusion

In wrapping up, the **Household Services Platform** has been crafted to make life easier by connecting people with reliable professionals for tasks like plumbing, electrical repairs, and cleaning. By using **React** for the user-friendly interface, **Spring Boot** for the sturdy backend, and **MySQL** for secure data storage, we've ensured the platform is both efficient and easy to use. This setup simplifies the process of finding and hiring trusted service providers, making home maintenance more straightforward and less stressful for everyone involved.

REFERENCES

1. **Spring Boot Documentation**

URL: https://spring.io/projects/spring-boot

2. **React.js Documentation**

URL: https://reactjs.org/docs/getting-started.html

3. **Java Programming Language**

URL: https://www.oracle.com/java/

4. **MySQL Workbench Documentation**

URL: https://dev.mysql.com/doc/workbench/en/

5. **Spring Boot with React**

URL: https://www.baeldung.com/spring-boot-react-and-redux

6. **Java Persistence API (JPA) Documentation**

URL: https://www.eclipse.org/eclipselink/documentation/2.7/

7. **MDN Web Docs**

URL: https://developer.mozilla.org/