A PROJECT ON

"HOUSEHOLD SERVICE MANAGEMENT SYSTEM"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Sahil Apture
Biswaranjan Barik
Siddarth Shankar Singh
Amol Shrivastava
Mirza Abuzar Baig

UNDER THE GUIDENCE OF:
Mr.Snehal Jadhav
Faculty Member

Sunbeam Institute of Information Technology, PUNE.

ACKNOWLEDGEMENT

<u>CERTIFICATE</u>

This is to certify that the project work under the title '**Household Service Management System'** is done by Sahil Apture, Amol Shrivastava, Biswaranjan Barik, Mirza Abuzar Baig, Siddarth Shankar Singh in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Yogesh Kohle
Course Coordinator

Date : 3rd February, 2026

1. **INTRODUCTION TO PROJECT**

The web-based **Household Service Management System** is designed to provide a centralized and efficient platform that connects service consumers with verified household service providers. The system aims to simplify the process of discovering, booking, and managing household services such as plumbing, electrical work, cleaning, appliance repair, and other essential services through a secure and user-friendly application.

In traditional systems, users rely on manual searches, phone calls, or third-party references to find service providers, which often leads to delays, lack of transparency, and trust issues. The Household Service Management System overcomes these challenges by offering a digital solution where consumers can browse available services, compare vendors, and place service requests, while vendors can manage their profiles, services, and bookings efficiently.

The system provides role-based access and supports three primary roles: Consumer, Vendor, and Administrator. Consumers can register on the platform, browse services, raise service requests, track their bookings, and provide feedback. Vendors can register, manage their service listings, accept or reject service requests, update service status, and view their booking history. Administrators oversee the entire system by managing users, approving vendors, monitoring services, and ensuring smooth platform operation.

The application is implemented as a secure, scalable web-based system using a centralized backend architecture. The backend exposes RESTful APIs to handle business logic and data transactions, ensuring separation of concerns and maintainability. Security is enforced using authentication and authorization mechanisms to protect user data and restrict access based on roles.

Overall, the Household Service Management System enhances transparency, reliability, and efficiency in household service management by providing a single integrated platform for consumers, vendors, and administrators, thereby improving service accessibility and user experience.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS
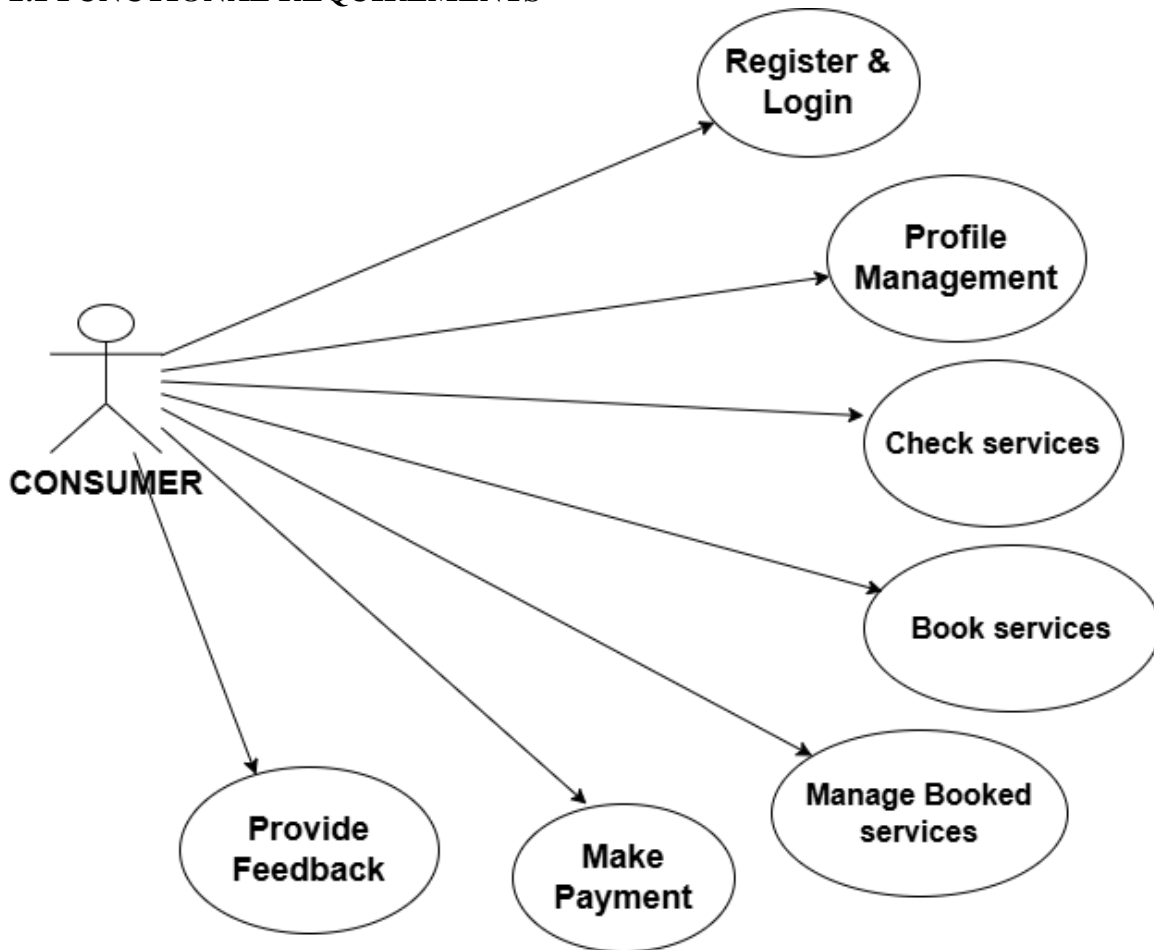
2.1   Consumer

The Consumer is an end user of the Household Service Management System who uses the platform to raise household service requests and track their progress. The system is designed such that the consumer is not involved in vendor selection. Vendor assignment is handled internally by the system and the administrator, ensuring fair distribution of work and simplified user interaction.

2.1.1   Consume Access and Account Management

A consumer may access the system either as a guest or as a registered user:

- A guest consumer can browse available service categories without logging into the system.

- A registered consumer can log in using valid credentials to access service request and tracking features.

Authentication is mandatory for performing any transactional operation. The system verifies consumer identity through secure login mechanisms before granting access.

2.1.2 Consumer Registration and Profile Creation

To raise a service request, the consumer must register with the system by providing required personal details such as name, email address, contact number, and address. Upon successful registration:

- A unique consumer profile is created in the system database.
- Login credentials are securely stored using password encoding mechanisms.
- The consumer can update personal profile information after logging in.

The system performs input validation to ensure correctness and prevent duplicate registrations.

2.1.3 Service Browsing

The system allows consumers to browse available household services without requiring vendor-specific information. The consumer can:

- View service categories (e.g., plumbing, electrical, cleaning, appliance repair).

- Read service descriptions.
- Select a service type for which assistance is required.

At this stage, vendor details are not displayed to the consumer.

2.1.4 Service Request Creation

After selecting a service category, a logged-in consumer can raise a service request by providing required details such as service description and address. The system performs the following actions:

1. Stores the service request in the system database.
2. Assigns an initial status of Pending to the request.
3. Restricts consumer access to vendor-related information.

2.1.5 Vendor Assignment and Visibility Control

Vendor assignment is handled by the system through administrative control or internal allocation logic. Once a vendor is assigned to a service request:

- The consumer is notified of the assignment.
- Limited vendor information (such as name and contact details) becomes visible to the consumer.
- Prior to assignment, all vendor information remains hidden from the consumer.

This approach ensures controlled disclosure of vendor details and prevents bias or misuse.

2.1.6 Service Status Tracking

Consumers can track the progress of their service requests through a dedicated dashboard. The system supports the following service states:

- Pending
- Assigned

- Accepted
- In Progress
- Completed
- Rejected

Status updates are reflected in real time based on vendor actions and administrative decisions.

2.1.7 Service Completion and History

After a service request is marked as Completed, the consumer can:

- View detailed service information.
- Access booking and service history.
- Track previous requests in a tabular format.

This feature provides transparency and allows consumers to maintain records of all past service interactions.
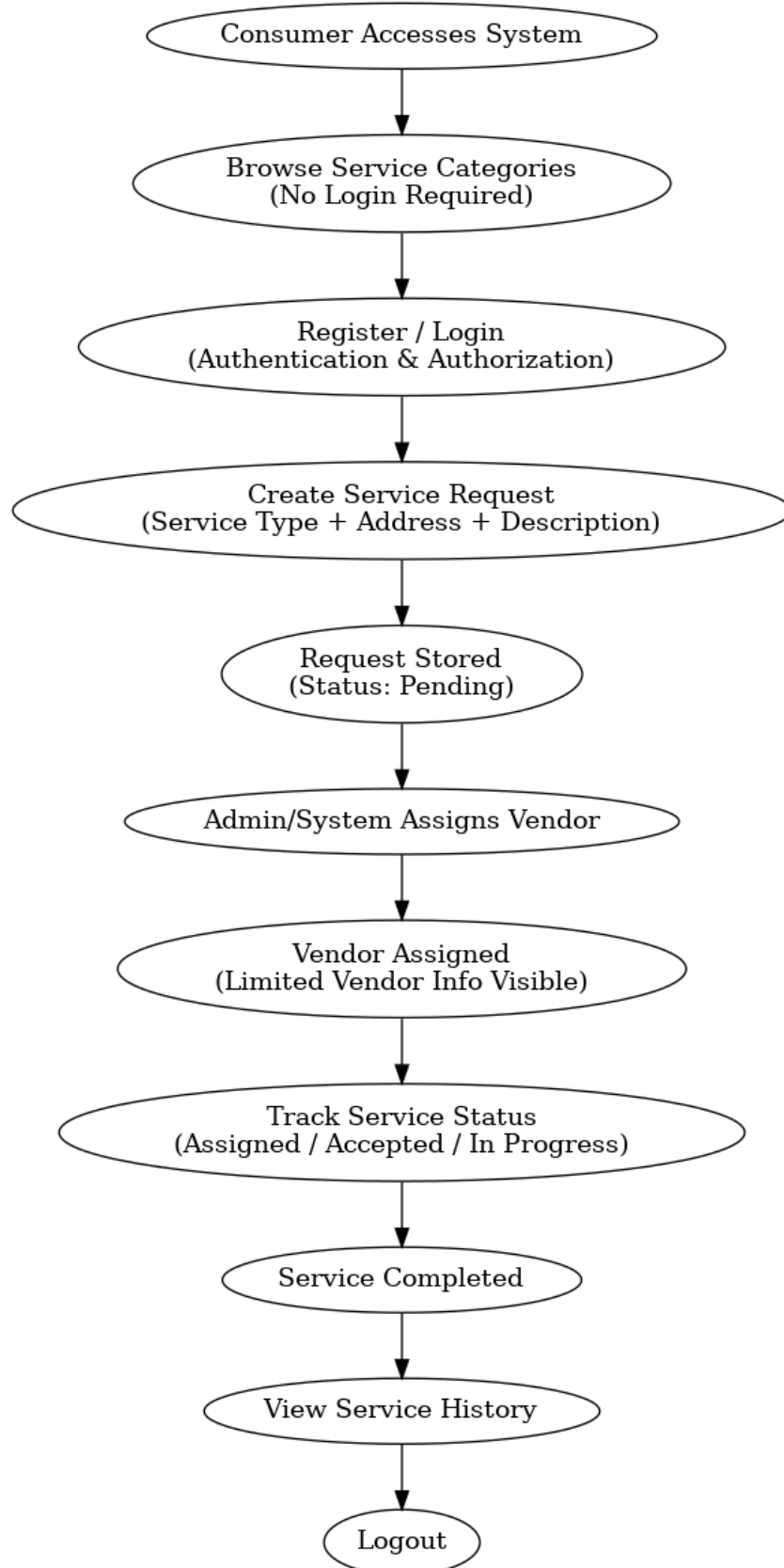
2.1.8 Security and Authorization

All consumer operations are protected using secure authentication and authorization mechanisms. The system ensures that:

- Only authenticated consumers can create or view service requests.
- Consumers can access only their own data.
- Unauthorized access is prevented using token-based authentication.

Session handling and role-based authorization ensure data integrity and privacy.

2.1.9 Consumer Flow Diagram

Consumer Flow Diagram – Household Service Management System

Consumer Accesses System

↓

Browse Service Categories
(No Login Required)

↓

Register / Login
(Authentication & Authorization)

↓

Create Service Request
(Service Type + Address + Description)

↓

Request Stored
(Status: Pending)

↓

Admin/System Assigns Vendor

↓

Vendor Assigned
(Limited Vendor Info Visible)

↓

Track Service Status
(Assigned / Accepted / In Progress)

↓

Service Completed

↓

View Service History

↓

Logout

The Consumer Flow Diagram illustrates the interaction sequence followed by a consumer while using the Household Service Management System. The flow begins with the consumer accessing the system and browsing available service categories without authentication. To raise a service request, the consumer must register or log in, after which a service request is created by providing service details and address information. The request is stored with a Pending status and forwarded for administrative or system-level vendor assignment. Vendor details remain hidden until assignment is completed. Once a vendor is assigned, the consumer can track service progress through various stages such as Assigned, Accepted, In Progress, and Completed. After service completion, the consumer can view service history and logout securely. This flow ensures controlled service allocation, data privacy, and transparency.

2.2 Vendor

The Vendor represents a service provider in the Household Service Management System who is responsible for delivering household services assigned through the platform. Vendors interact with the system to manage their profiles, view assigned service requests, update service status, and maintain service history. All vendor operations are governed by authentication and role-based authorization to ensure secure access.

To access the system, a vendor must register and create an account by providing required details such as business name, contact information, service categories offered, and operational location. Upon registration, vendor accounts are subject to verification and approval by the administrator before gaining full access to the system. This approval mechanism ensures that only authorized and verified vendors are allowed to operate on the platform.

Once approved, a vendor can log in to the system using valid credentials. Authentication is handled through a secure login mechanism, and unauthorized access to vendor-specific features is restricted. After successful login, the vendor is presented with a dashboard that displays service requests assigned to them by the system or administrator.

Vendors do not receive service requests directly from consumers. Instead, service requests are assigned after administrative review or system-level allocation. Once a service request is assigned, the vendor gains access to limited consumer information necessary to perform the service. Prior to assignment, the vendor cannot view consumer details.

The vendor has the ability to accept or reject assigned service requests. If a service request is accepted, the system updates the request status accordingly and allows the vendor to proceed with service execution. In case of rejection, the system records the action and reassigns the request as per administrative logic.

During service execution, the vendor can update the service status to reflect the current progress. Supported service states include Accepted, In Progress, and Completed. Status updates are stored in the system database and are visible to the consumer and administrator in real time.

Upon completion of a service, the vendor marks the request as Completed. This action finalizes the service lifecycle and allows the system to archive the request in the service history. Vendors can view a complete record of their past and ongoing service assignments through the system interface.

The system also allows vendors to manage their profile information, including contact details and service categories. Any modifications to critical information may require administrative approval to ensure data integrity and trustworthiness.

All vendor-related operations are protected using secure authentication and authorization mechanisms. Vendors can access only those service requests that are explicitly assigned to them, and they are restricted from viewing or modifying data belonging to other vendors. Token-based authentication ensures secure communication between the vendor interface and backend services.

2.2.1   Vendor Use Case Diagram

The Vendor Use Case Diagram illustrates the interaction between vendors and the system. It depicts major vendor activities such as login, viewing assigned requests, accepting or rejecting service requests, updating service status, and managing profile information.
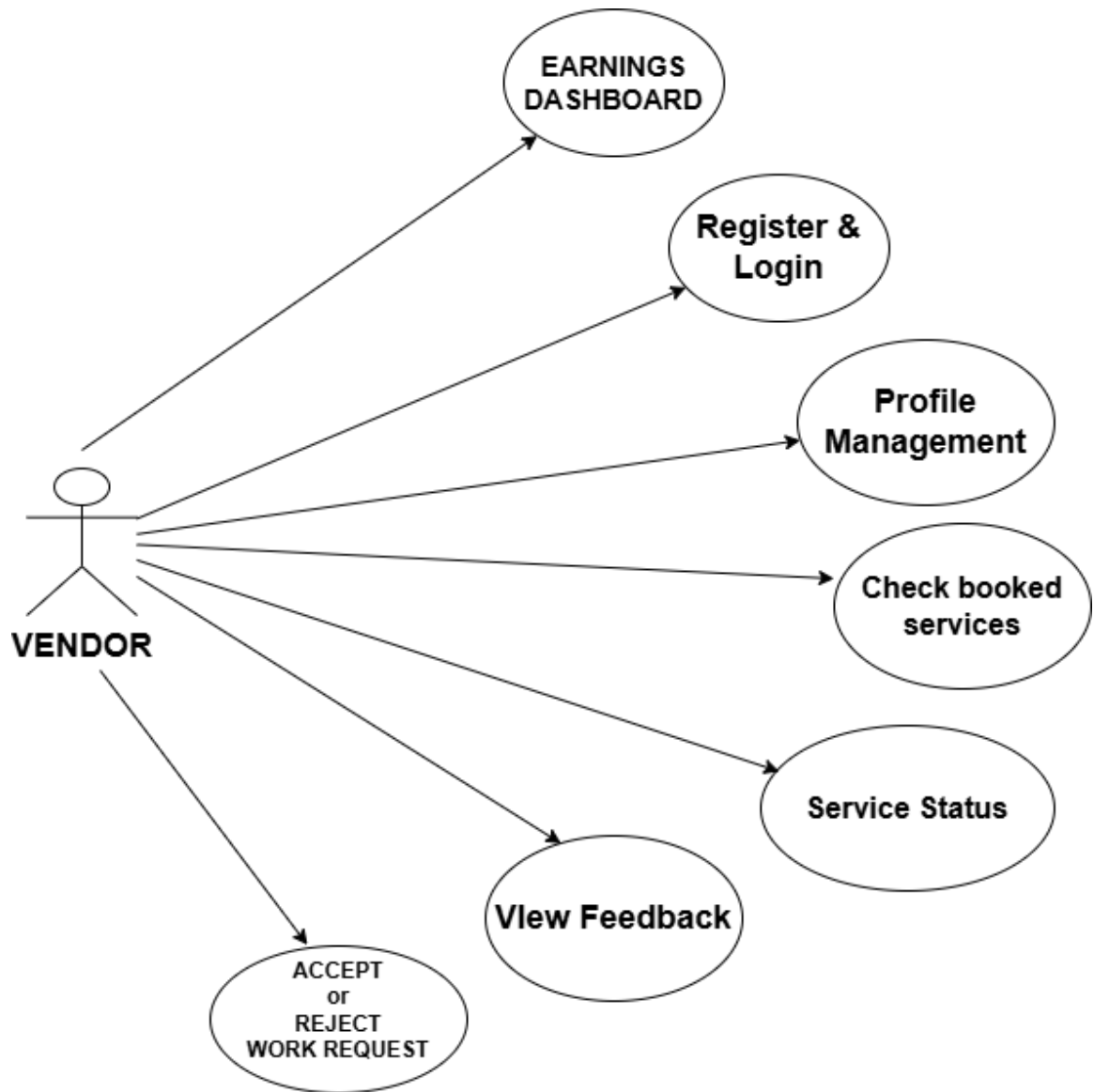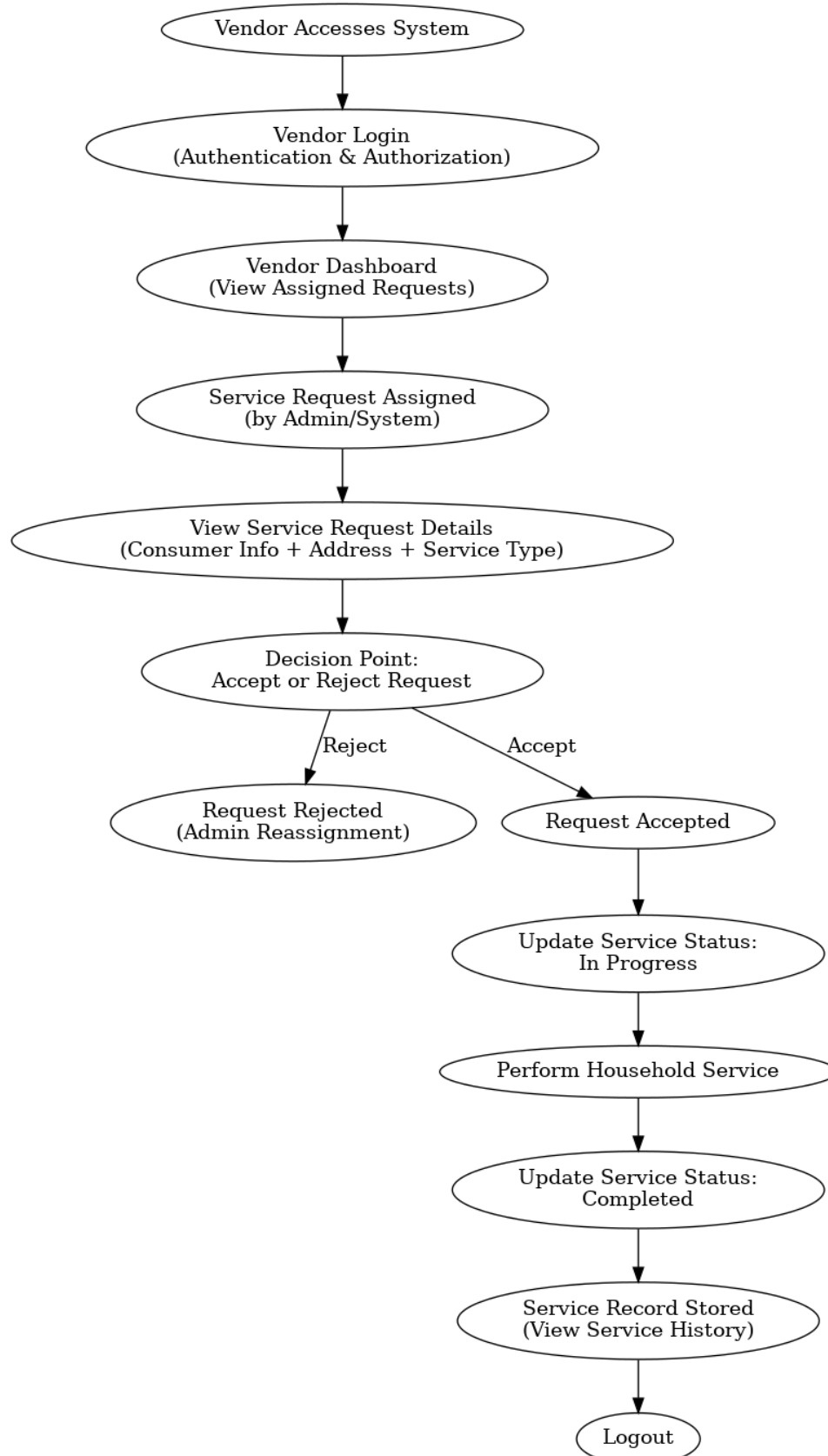
2.2.1 Vendor Flow Diagram

The Vendor Flow Diagram represents the complete operational workflow followed by a vendor within the Household Service Management System. The flow begins with vendor authentication and authorization, after which the vendor gains access to a dashboard displaying service requests assigned by the administrator or system. Upon receiving an assignment, the vendor can view relevant service and consumer details required for execution.

The vendor then decides whether to accept or reject the service request. Accepted requests progress through service execution stages, where the vendor updates the service status from In Progress to Completed. Rejected requests are returned to the system for reassignment. Upon completion, service records are archived and made available in the vendor's service history. This flow ensures controlled service allocation, accountability, and transparent status tracking.

Vendor Flow Diagram – Household Service Management System

```
        ┌─────────────────────────┐
        │  Vendor Accesses System │
        └─────────────────────────┘
                    │
                    ▼
        ┌─────────────────────────┐
        │      Vendor Login       │
        │ (Authentication &       │
        │   Authorization)        │
        └─────────────────────────┘
                    │
                    ▼
        ┌─────────────────────────┐
        │    Vendor Dashboard     │
        │ (View Assigned Requests)│
        └─────────────────────────┘
                    │
                    ▼
        ┌─────────────────────────┐
        │ Service Request Assigned│
        │    (by Admin/System)    │
        └─────────────────────────┘
                    │
                    ▼
        ┌─────────────────────────────────┐
        │   View Service Request Details  │
        │ (Consumer Info + Address +      │
        │        Service Type)            │
        └─────────────────────────────────┘
                    │
                    ▼
        ┌─────────────────────────┐
        │     Decision Point:     │
        │ Accept or Reject Request│
        └─────────────────────────┘
             Reject │      │ Accept
                    ▼      ▼
    ┌──────────────────┐  ┌──────────────────┐
    │ Request Rejected │  │ Request Accepted │
    │ (Admin           │  └──────────────────┘
    │  Reassignment)   │           │
    └──────────────────┘           ▼
                        ┌──────────────────────┐
                        │ Update Service Status:│
                        │      In Progress      │
                        └──────────────────────┘
                                   │
                                   ▼
                        ┌──────────────────────┐
                        │ Perform Household     │
                        │     Service           │
                        └──────────────────────┘
                                   │
                                   ▼
                        ┌──────────────────────┐
                        │ Update Service Status:│
                        │      Completed        │
                        └──────────────────────┘
                                   │
                                   ▼
                        ┌──────────────────────┐
                        │ Service Record Stored │
                        │ (View Service History)│
                        └──────────────────────┘
                                   │
                                   ▼
                        ┌──────────┐
                        │  Logout  │
                        └──────────┘
```

2.3   Admin

The Administrator is the central controlling authority in the Household Service Management System and is responsible for overseeing system operations, managing users, approving vendors, and ensuring proper assignment and execution of service requests. The administrator plays a critical role in maintaining system integrity, security, and smooth coordination between consumers and vendors.

To access administrative functionalities, the administrator must log in using valid credentials. Authentication and authorization mechanisms ensure that only authorized administrators can access sensitive system-level operations. Upon successful login, the administrator is presented with an administrative dashboard that provides a consolidated view of system activities.

The administrator has full control over user management within the system. This includes viewing registered consumers and vendors, activating or deactivating user accounts when required, and monitoring overall user activity. This functionality helps in preventing misuse and maintaining platform reliability.

One of the key responsibilities of the administrator is vendor management. Vendors who register on the platform do not gain immediate access to service requests. The administrator verifies vendor details and either approves or rejects vendor registrations. Only approved vendors are allowed to receive service assignments, ensuring trust and service quality within the system.

The administrator is also responsible for service request management. All service requests raised by consumers are visible to the administrator. Based on service category, availability, and operational constraints, the administrator assigns an appropriate vendor to each service request. This controlled assignment mechanism ensures fair distribution of work and prevents consumers from directly selecting vendors.

Once a vendor is assigned, the administrator continuously monitors the service lifecycle. The system allows the administrator to track service status transitions such as Pending, Assigned, Accepted, In Progress, Completed, or Rejected. In cases where a vendor rejects a request or fails to act, the administrator can reassign the request to another vendor.

The administrator also performs system oversight and monitoring, ensuring that all operations follow defined workflows and business rules. This includes monitoring rejected requests, handling reassignment scenarios, and maintaining transparency between consumers and vendors.

All administrative actions are protected through role-based authorization. The administrator can access system-wide data, whereas consumers and vendors are restricted to their own data. Secure session handling and token-based authentication mechanisms ensure safe communication between the administrative interface and backend services.

2.3.1 Admin Use Case Diagram

The Admin Use Case Diagram represents the interaction between the administrator and the Household Service Management System. It highlights the administrative responsibilities such as managing users, approving vendor registrations, assigning vendors to service requests, monitoring service status, and handling reassignment in case of request rejection. This diagram emphasizes the administrator's role as the central controlling authority, ensuring secure access, controlled service allocation, and smooth coordination between consumers and vendors.
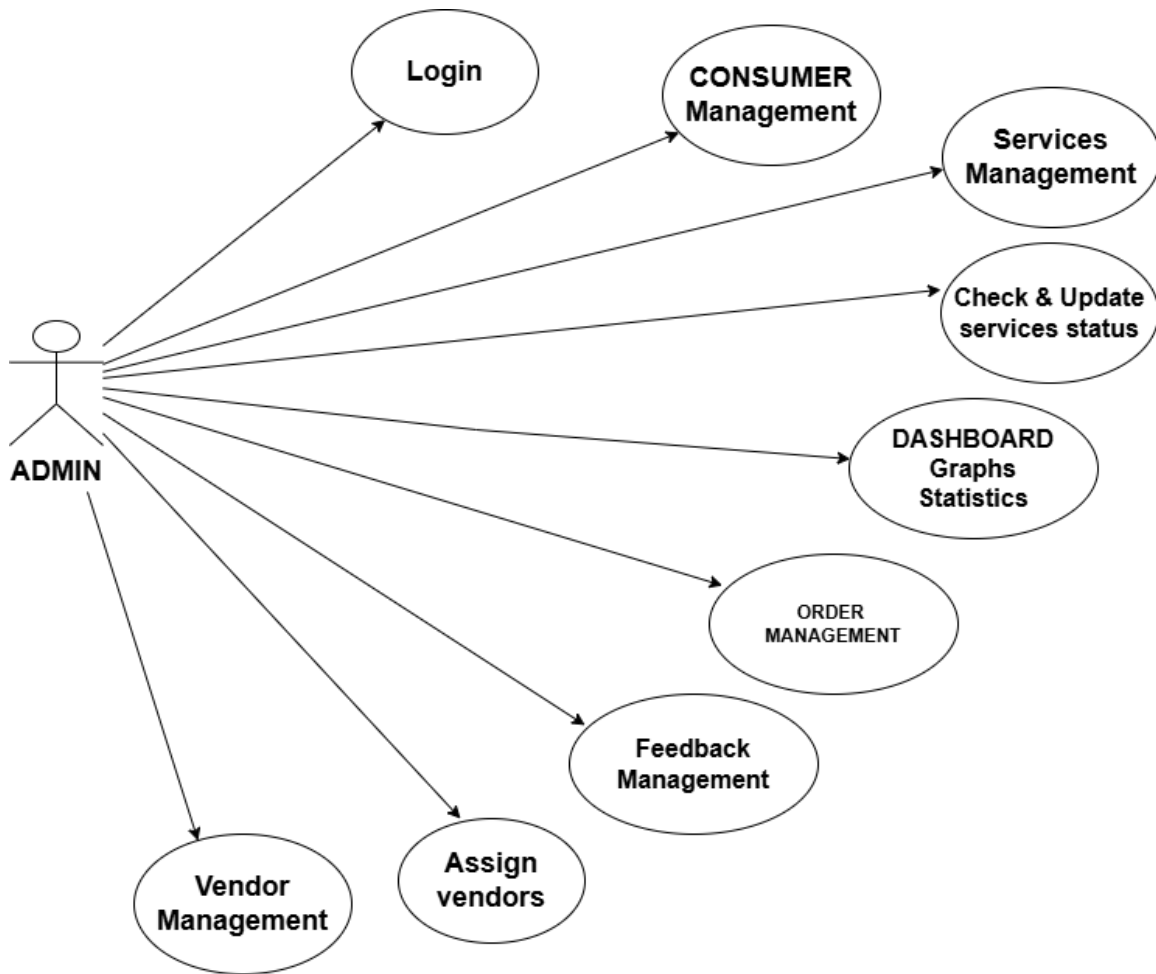
2.3.2 Admin Flow Diagram

The Admin Flow Diagram represents the operational workflow followed by the administrator within the Household Service Management System. The flow begins with administrator authentication and authorization, after which the administrator gains access to a centralized dashboard for system management.

From the dashboard, the administrator manages users and vendors, including approving or rejecting vendor registrations. The administrator monitors service requests raised by consumers and assigns appropriate vendors based on service requirements and availability. Throughout the service lifecycle, the administrator tracks service status and handles rejected requests by reassigning them when necessary. This flow highlights the

administrator's role as the central controlling authority, ensuring secure access, controlled service allocation, and smooth coordination between consumers and vendors.

Admin Flow Diagram – Household Service Management System
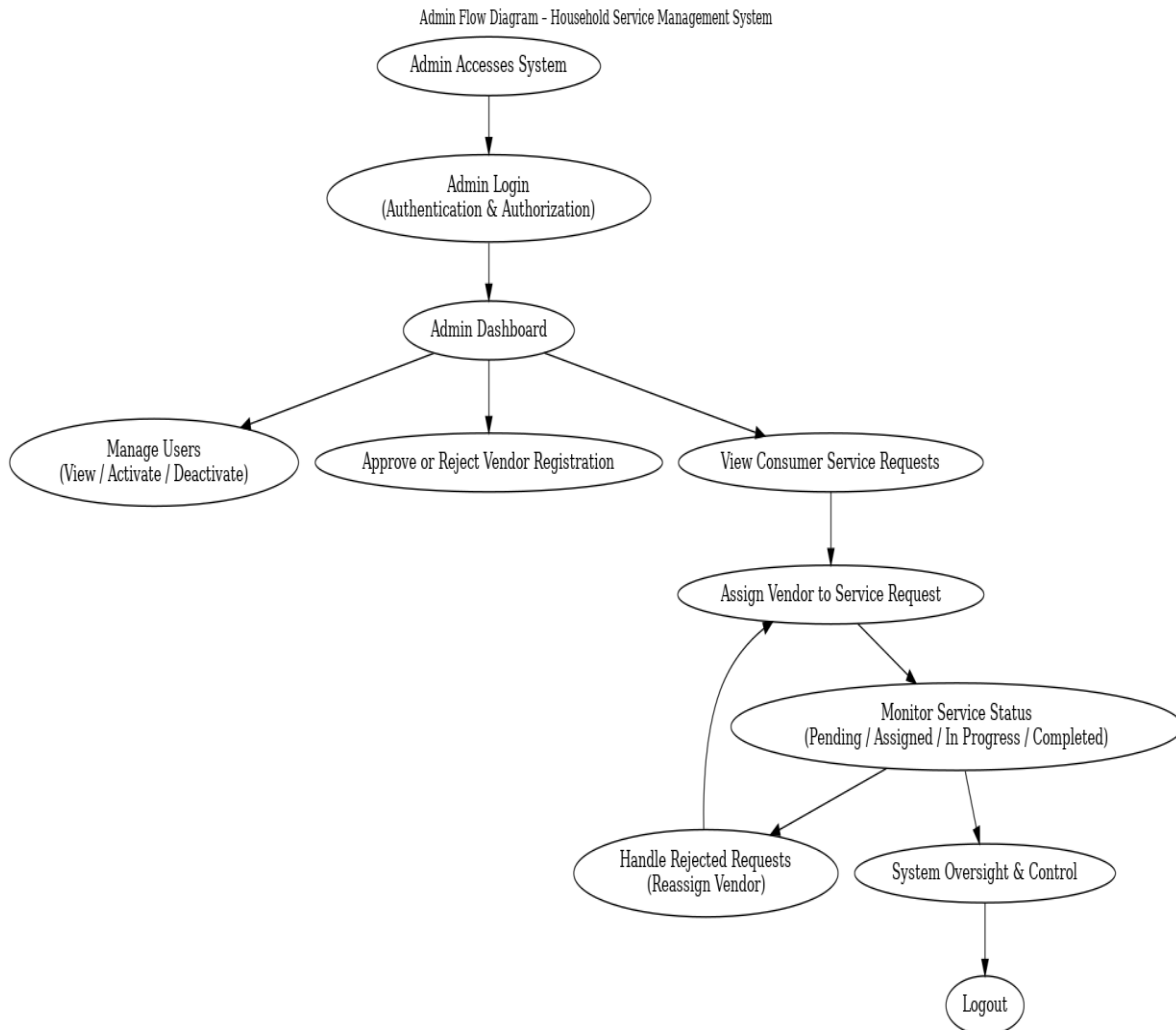
2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface

The Household Service Management System provides a web-based user interface that allows consumers, vendors, and administrators to interact with the system efficiently. The interface is designed to be simple, consistent, and easy to understand for all users. Separate interfaces are provided based on user roles to ensure clarity and controlled access.
Go to Appendix B for user interfaces

2.2.2 Performance

Number of Concurrent Users:

    The Household Service Management System shall be able to handle a large number of concurrent users. The system is designed to support multiple consumers submitting service requests, vendors updating service status, and administrators managing assignments simultaneously without significant performance degradation.

Service Request Processing:

    The system is capable of handling temporary server or network failures. Due to the use of a centralized backend architecture and reliable data persistence mechanisms, operations such as service request creation, vendor assignment, and service status updates remain consistent even if the server disconnects temporarily.

2.2.3 Constraint

    The Household Service Management System shall be able to handle a high number of service-related transactions and inquiries concurrently. System performance is dependent on server capacity, network availability, and database response time, and the system operates within these constraints efficiently.

2.2.4 Other Requirements:

Hardware Interfaces:

    The Household Service Management System is expected to function on standard computer hardware with sufficient processing power, memory, and storage to support web application execution.

Software Interfaces:

    The system shall work on commonly used operating systems and web browsers. It is configured to operate with a relational database system and runs on a web server environment. Users can access the system through standard web browsers without the need for additional software installation.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: User_Info

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	user_id	BIGINT	—	0
0	dob	DATE	—	1
UNI	email	VARCHAR	255	0
0	first_name	VARCHAR	100	1
0	last_name	VARCHAR	100	1
0	password	VARCHAR	255	1
UNI	phone	VARCHAR	10	0
0	role	ENUM	—	1
0	user_status	ENUM	—	1

Table2: ADDRESS

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	address_id	BIGINT	—	0
0	city	VARCHAR	100	1
0	home_no	VARCHAR	100	1
0	pincode	VARCHAR	10	1
0	state	VARCHAR	100	1
0	town	VARCHAR	100	1
MUL	user_id	BIGINT	—	0

Table: CONSUMER

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	consumer_id	BIGINT	—	0
0	reward_points	INT	—	0
UNI	user_id	BIGINT	—	1

Table: CONSUMER_REVIEW

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	consumer_review	BIGINT	—	0
0	description	TINYTEXT	—	1
0	rating	INT	—	1
UNI	order_id	BIGINT	—	0

Table: CONTACT_SUPPORT

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	contact_id	BIGINT	—	0
0	category	VARCHAR	255	1
0	description	TEXT	—	1
0	email	VARCHAR	255	0
0	name	VARCHAR	255	0

Table: LANGUAGES

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	lang_id	BIGINT	—	0
UNI	lang_name	VARCHAR	255	0

Table: ORDERS

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	order_id	BIGINT	—	0
0	order_date_time	DATETIME(6)	—	1
0	order_price	DOUBLE	—	1
0	priority	ENUM	—	1
0	status	ENUM	—	1

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
0	time_slot	DATETIME(6)	—	1
MUL	consumer_id	BIGINT	—	0
UNI	consumer_review_id	BIGINT	—	1
UNI	consumer_transaction_id	BIGINT	—	1
MUL	service_id	BIGINT	—	0
MUL	vendor_id	BIGINT	—	1
UNI	vendor_review_id	BIGINT	—	1
MUL	address_id	BIGINT	—	0

Table: SERVICE_IMAGES

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	service_image_id	BIGINT	—	0
0	image	LONGBLOB	—	1
MUL	service_id	BIGINT	—	0

Table: SERVICES

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	service_id	BIGINT	—	0
0	long_desc	TEXT	—	0
0	active	BIT	1	0
0	category	ENUM	—	0
0	price	DOUBLE	—	0
UNI	service_name	VARCHAR	255	0
0	short_desc	VARCHAR	255	0

Table: USER_IMAGES

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	img_id	BIGINT	—	0
0	image	TINYBLOB	—	1
UNI	user_id	BIGINT	—	0

Table: USER_LANGUAGE

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	user_id	BIGINT	—	0
PRI	lang_id	BIGINT	—	0

Table: VENDOR

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	vendor_id	BIGINT	—	0
UNI	aadhar_no	VARCHAR	15	0
0	experience	INT	—	0
UNI	pan_no	VARCHAR	15	0
UNI	user_id	BIGINT	—	0

Table: VENDOR_BANKING

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	banking_id	BIGINT	—	0
UNI	account_no	VARCHAR	255	0
0	bank_name	VARCHAR	255	0
0	branch_name	VARCHAR	255	1
0	holder_name	VARCHAR	255	0
0	ifsc_code	VARCHAR	255	1
UNI	vendor_id	BIGINT	—	1

Table: VENDOR_REVIEW

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	vendor_review	BIGINT	—	0
0	description	TINYTEXT	—	1
0	rating	INT	—	1
UNI	order_id	BIGINT	—	0

Table: VENDOR_SERVICE

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	vendor_id	BIGINT	—	0
PRI	service_id	BIGINT	—	0

Table: VENDOR_TRANSACTION

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	vendor_transaction_id	BIGINT	—	0
0	amount	DOUBLE	—	0
0	status	ENUM	—	0
0	timestamp	DATETIME(6)	—	1
UNI	order_id	BIGINT	—	0

Table: VENDOR_WALLET

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	wallet_id	BIGINT	—	0
0	balance	DOUBLE	—	0
0	created_at	DATETIME(6)	—	1
0	status	ENUM	—	1
0	updated_at	DATETIME(6)	—	1
UNI	vendor_id	BIGINT	—	0

E-R Diagram,Dataflow diagram and Class Diagram:

Go to Appendix A

4. 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, Services, Vendor, Languages	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	getVendorById	Methods should use **verbs** or verb phrases.
Parameter	Camel	userName, ifscCode	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	Disposable	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_foreColor, _backColor	Use underscore camel casing for the private member variables
Exception Class	Pascal with "Exception" suffix	WebException,	

Table: CONSUMER_TRANSACTION

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes,0=No)
PRI	consumer_transaction_id	BIGINT	—	0
0	amount	DOUBLE	—	0
0	payment_id	VARCHAR	255	0
0	status	ENUM	—	0
0	timestamp	DATETIME(6)	—	1
UNI	order_id	BIGINT	—	0

Comments

- Comment each type, each non-public type member, and each region declaration.

- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.

- Separate comments from comment delimiters (apostrophe) or // with one space.

- Begin the comment text with an uppercase letter.

- End the comment with a period.

- Explain the code; do not repeat it.

5. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.
GENERAL TESTING
Table: General Test Cases

SR NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Register Page	Redirected to login page	OK	According to validation
2	Login Page	Redirect to home page	OK	Please enter username and password again
3	Reset Login	Only user password will be reset	OK	Nothing
4	Quick Search Services	Displays respective services	OK	Nothing
5	Book Services	All required fields must be filled for booking	OK	Nothing
6	Login Verification	Checks whether user is logged in or not	OK	Nothing
7	Add Address Details for Services	Address added successfully	OK	Nothing
8	Go to Orders Page	Displays information about orders	OK	Nothing
9	Add Services in Service Table	Service data saved successfully in service table	OK	Nothing
10	Transaction	On back, user is redirected to previous	OK	Nothing

SR NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
		page		
11	View Transaction History	Displays all previous transactions	OK	Nothing
12	Logout	User is logged out from the system	OK	Nothing

STATIC TESTING

Table: Static Testing Details

SR NO	DEVIATION	PROGRAM
1	Commenting not followed	All Web Application

6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	SLC PHASE	ADDITIONAL NOTES
01 DEC 2025	Project allotment and user requirements gathering	Feasibility Study	Group formed and discussion about project
06 DEC 2025	Discussion about components and start of creation	Development	Component creation started
07 DEC 2025	Component creation	Development	Frontend completed
08 DEC 2025	Project evaluation	First Phase	Well completed
20 DEC 2025	Designing use cases, class diagram, collaboration diagram, ER diagram and user interfaces	Requirement Analysis and Design Phase	Database design completed
01 JAN 2026	Business logic component	Design Phase	----------

	design started		
05 JAN 2026	*Coding phase started*	*Coding Phase*	*70% of class library implemented*
20 JAN 2026	*Implementation of web application started*	*Coding Phase*	*Class library development ongoing*
22 JAN 2026	*Off*	*Off*	*Off*
25 JAN 2026	*Implementation of web application continued*	*Coding Phase and Unit Testing*	*Class library modified as per requirement*
27 JAN 2026	*Project evaluation*	*Phase II*	*Well completed*
01 FEB 2026	*Project deployment*	*Deployment*	*Deployed on AWS*
02 FEB 2026	*Frontend and backend integration*	*Coding Phase and Unit Testing*	*Module integration completed*
03 FEB 2026	*Final project evaluation*	*Final Phase*	*Well completed*

Appendix A

Entity Relationship Diagram
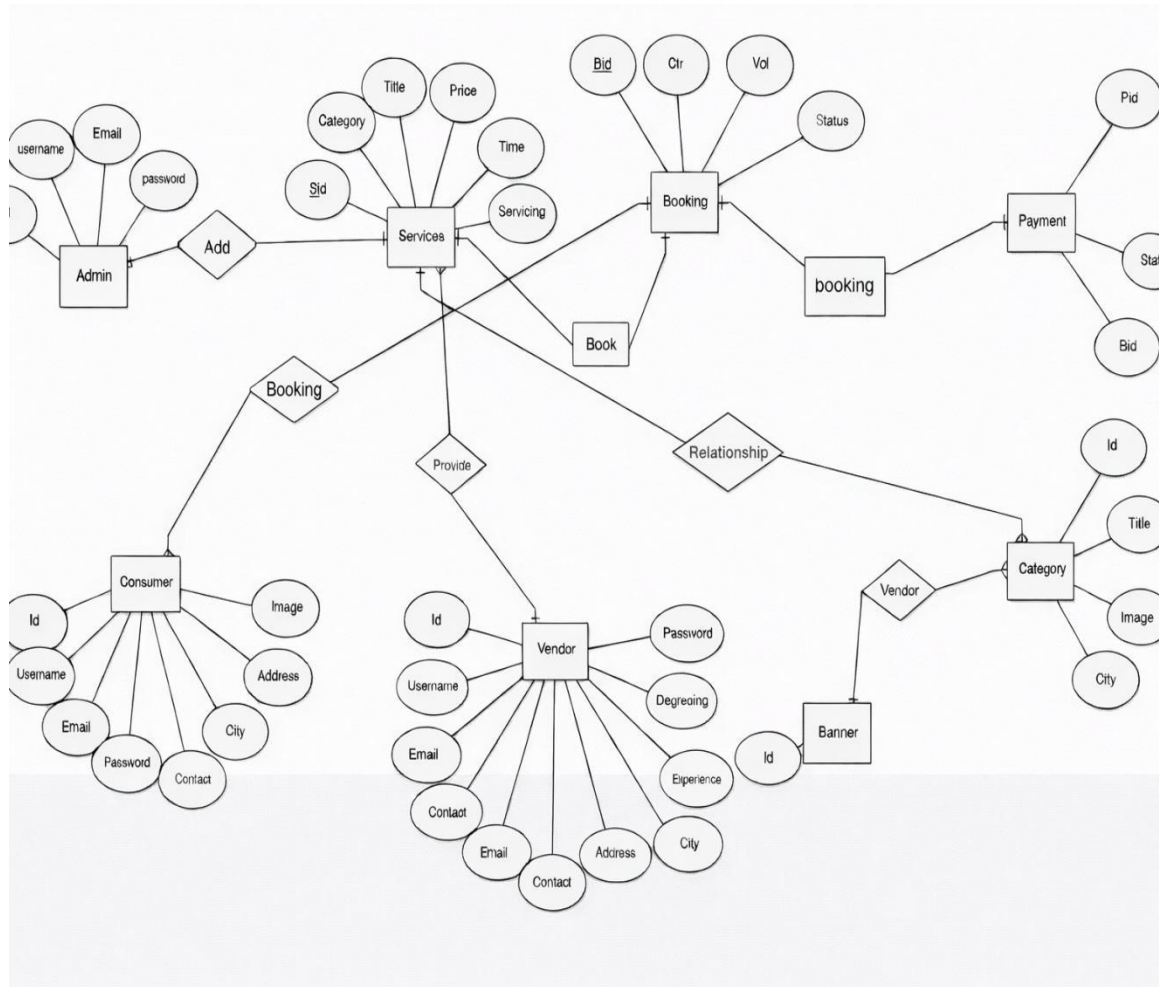
Data Flow Diagram
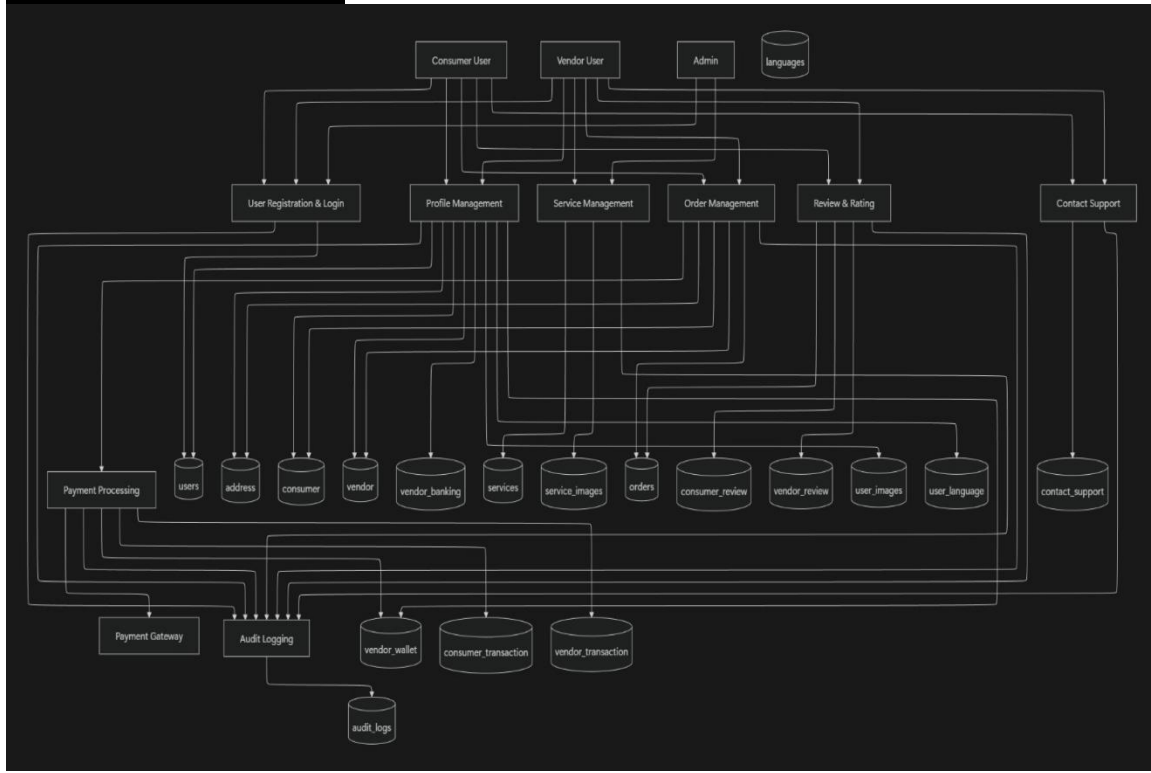
Class Diagram
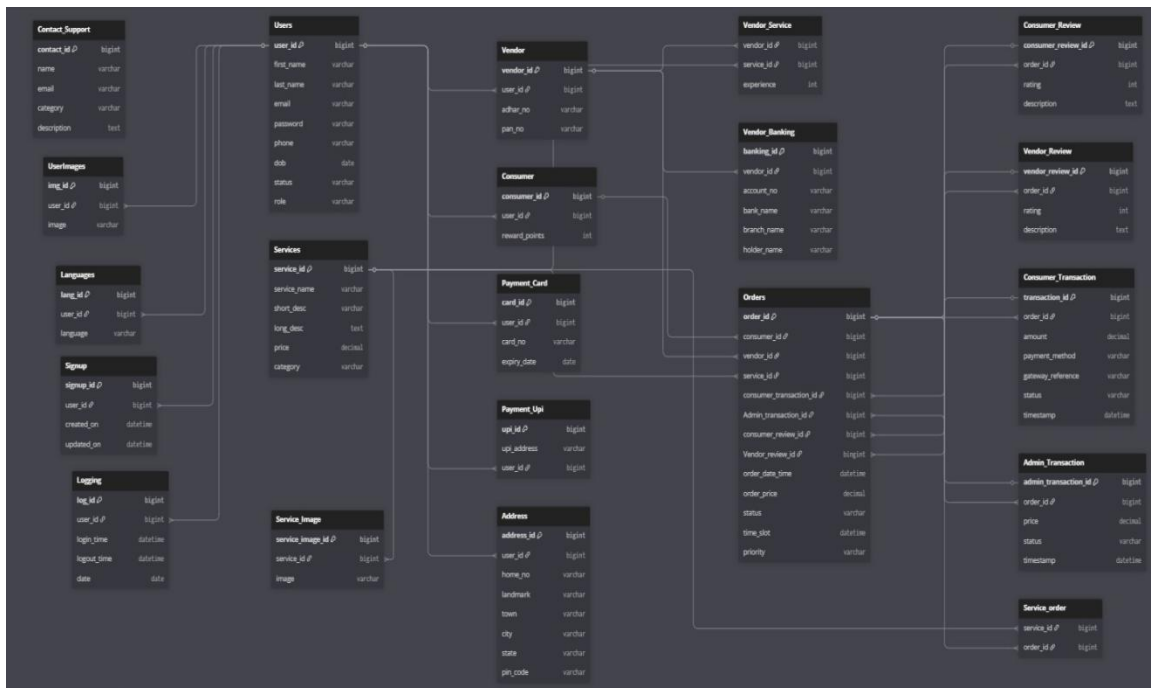
Appendix B

Homepage:
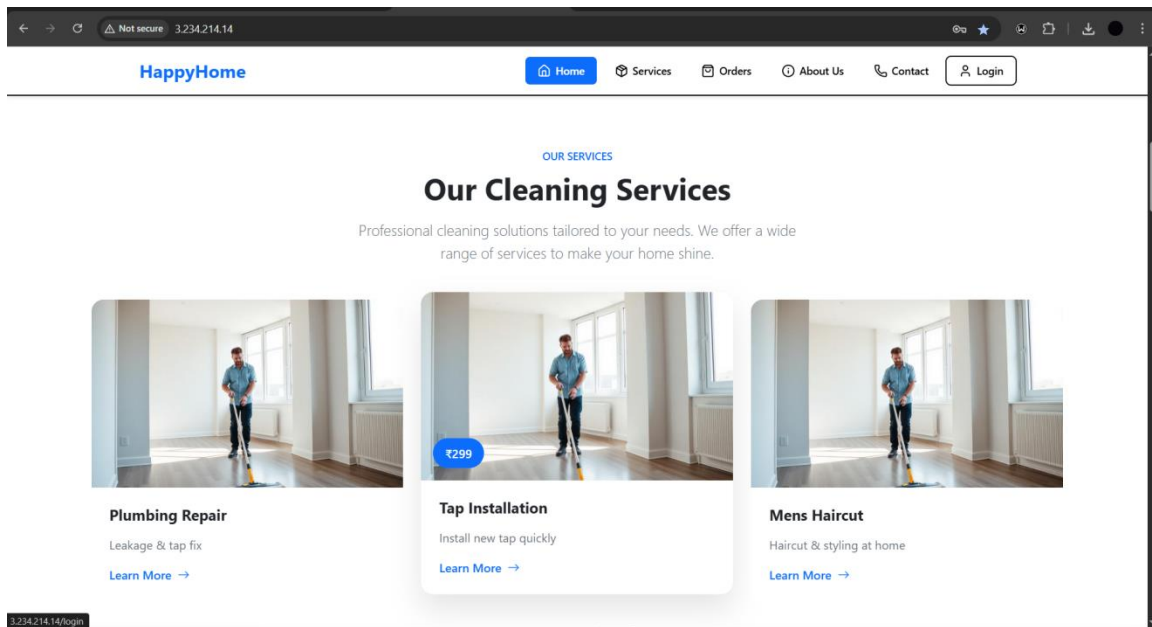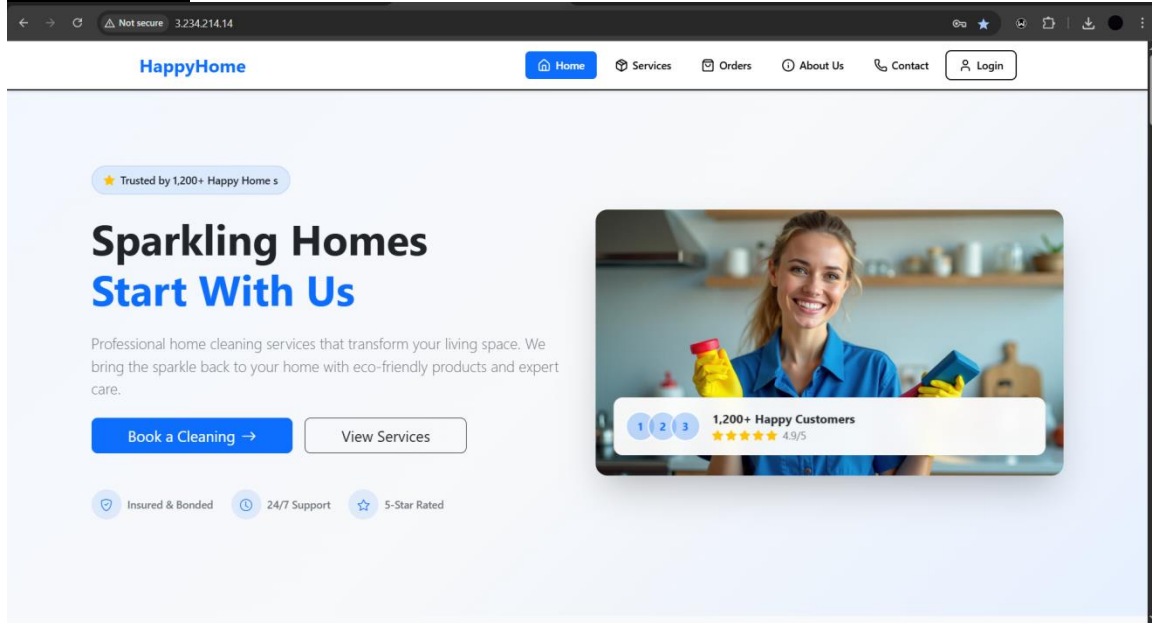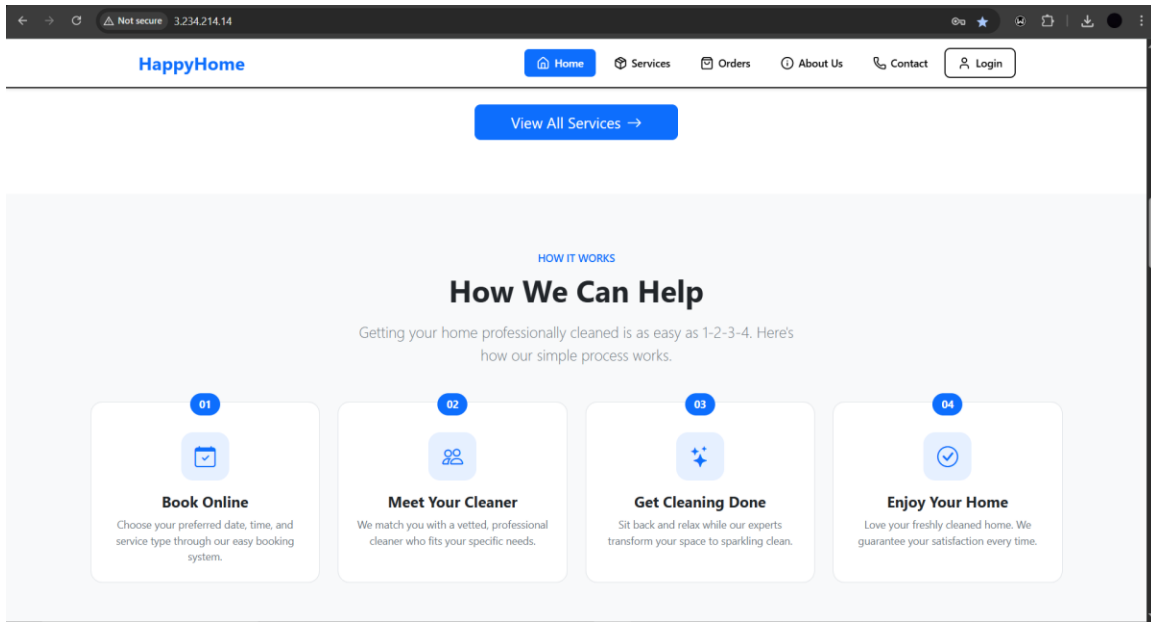
Login:
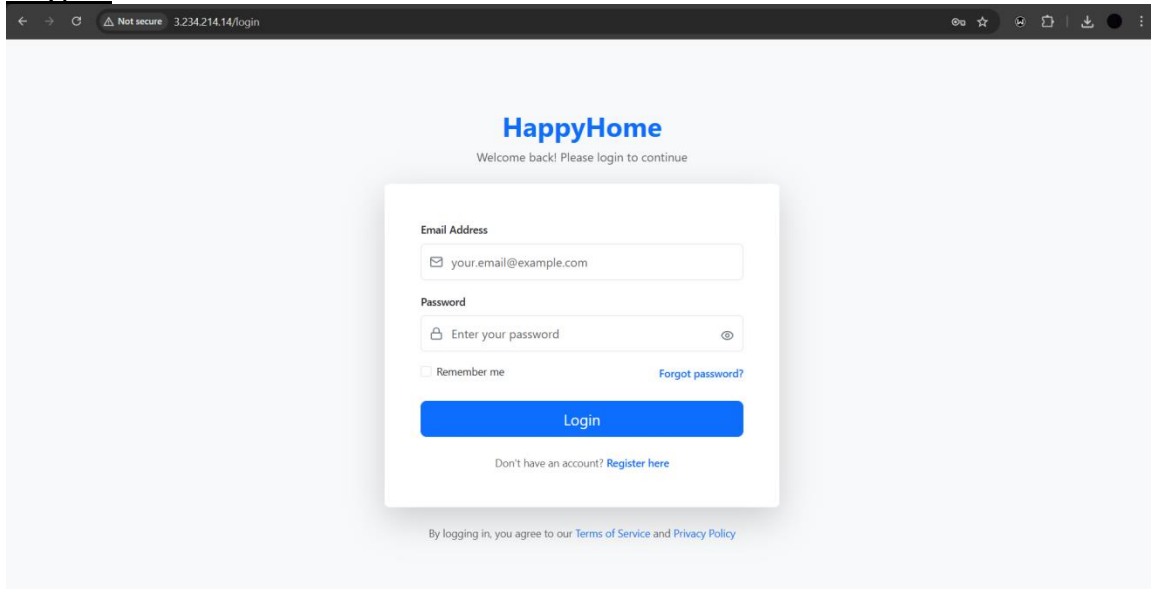
Consumer/Vendor Registration

Consumer Booking a Service
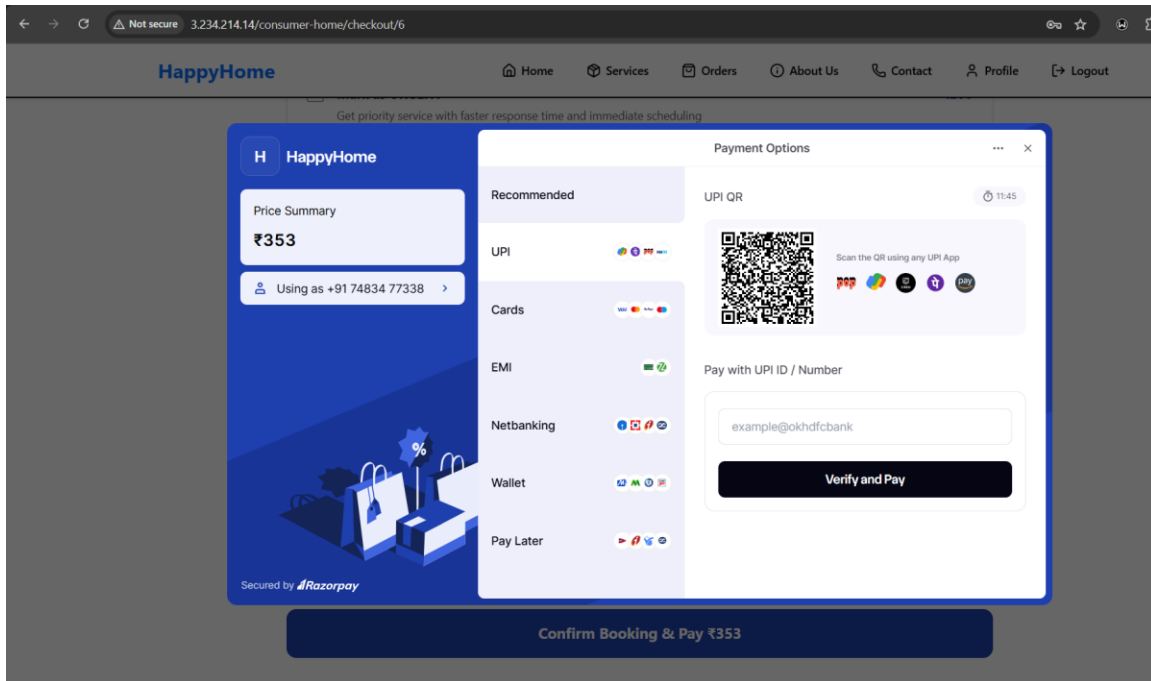
VendorDashboard

Vendor Order History

VendorProfile:

AdminDashboard:

Admin Manage Service:

Admin Add New Service:

Admin

7.REFERENCES:

http://www.google.com

https://www.urbancompany.com/

http://www.webdevelopersjournal.com/

http://www.w3.org

http://www.wikipedia.org

https://www.researchgate.net/

https://docs.oracle.com/javase/8/docs/api/

https://mvnrepository.com/

https://docs.spring.io/spring-boot/

https://www.mysql.com/

https://developer.mozilla.org/en-US/

https://docs.aws.amazon.com/

https://getbootstrap.com/