ActiPect

Home Service Solution

Project Documentations

June 2024

Contents

List of Figures

List of Equations

Abstract

This project is a comprehensive home service solution designed to streamline and optimize home maintenance and improvement tasks. It caters to admins, suppliers, service providers, and customers, providing each with a personalized dashboard accessible via an integrated web platform. Key features include account boosts for service providers, real-time calendar management, flexible pricing, secure payment processing, and appointment scheduling. Customers can submit service requests, manage appointments, place and cancel orders, and provide feedback through ratings and reviews. This platform aims to enhance convenience and efficiency in home service management, offering a seamless and user-friendly interface. The project was successfully completed within the time frame provided by the customer for 5000/- BDT.

1 Introduction

1.1 Project Overview

This project is a comprehensive home service solution designed to streamline and optimize home maintenance and improvement tasks. It caters to admins, suppliers, service providers, and customers, providing each with a personalized dashboard accessible via an integrated web platform. Key features include account boosts for service providers, real-time calendar management, flexible pricing, secure payment processing, and appointment scheduling. Customers can submit service requests, manage appointments, place and cancel orders, and provide feedback through ratings and reviews. This platform aims to enhance convenience and efficiency in home service management, offering a seamless and user-friendly interface.

1.2 Technologies Used

1.2.1 programming Language

1. **PHP**
   PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. It is embedded in HTML and executed on the server, generating dynamic web pages. Here, It is used to handle the backend of the website

2. **HTML**
   HTML (Hypertext Markup Language) is the standard markup language for creating web pages. It structures content using elements such as headings, paragraphs, and links.

3. **JavaScript**
   JavaScript is a versatile scripting language used for web development. It enables interactive and dynamic features on the client side, enhancing the user experience in web browsers.

1.2.2 Frameworks

- Bootstrap

1.2.3 Tools

- XAMPP

1.2.4 Database

- MySQL

1.2.5 Design Pattern

- MVC
  The Full form of MVC is Model View Controller. Here all the frontend or we can say the (HTML & PHP) is placed in the folder called View. Database (SQL) code in Model, backend or we can say (PHP) in Controller folder.

2 Overall Description

2.1 Product Perspective

“Home Service Solution” is the replacement of the manual hard copy sells process. The data have been stored in the hard file or papers, this website will store all of those in the website. Main goal of this project is to minimize the work and maximize the result of this result processing system.

2.1.1 User Classes and Characteristics

“Home Service Solution” has currently 2 Type of Users[4] called Customer and Service Provider among the 4 Users.

- Admin
- Customer
- Service Provider
- Supplier

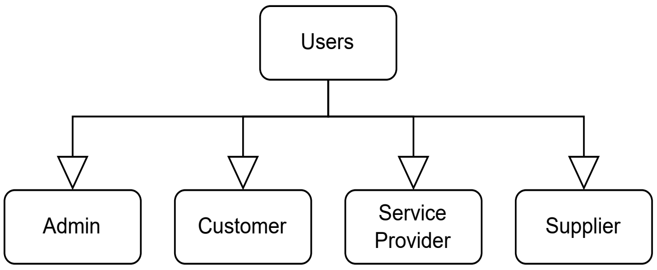


Figure 2.1: Type of Users

3

Architecture

3.1 Design Patterns

3.1.1 MVC

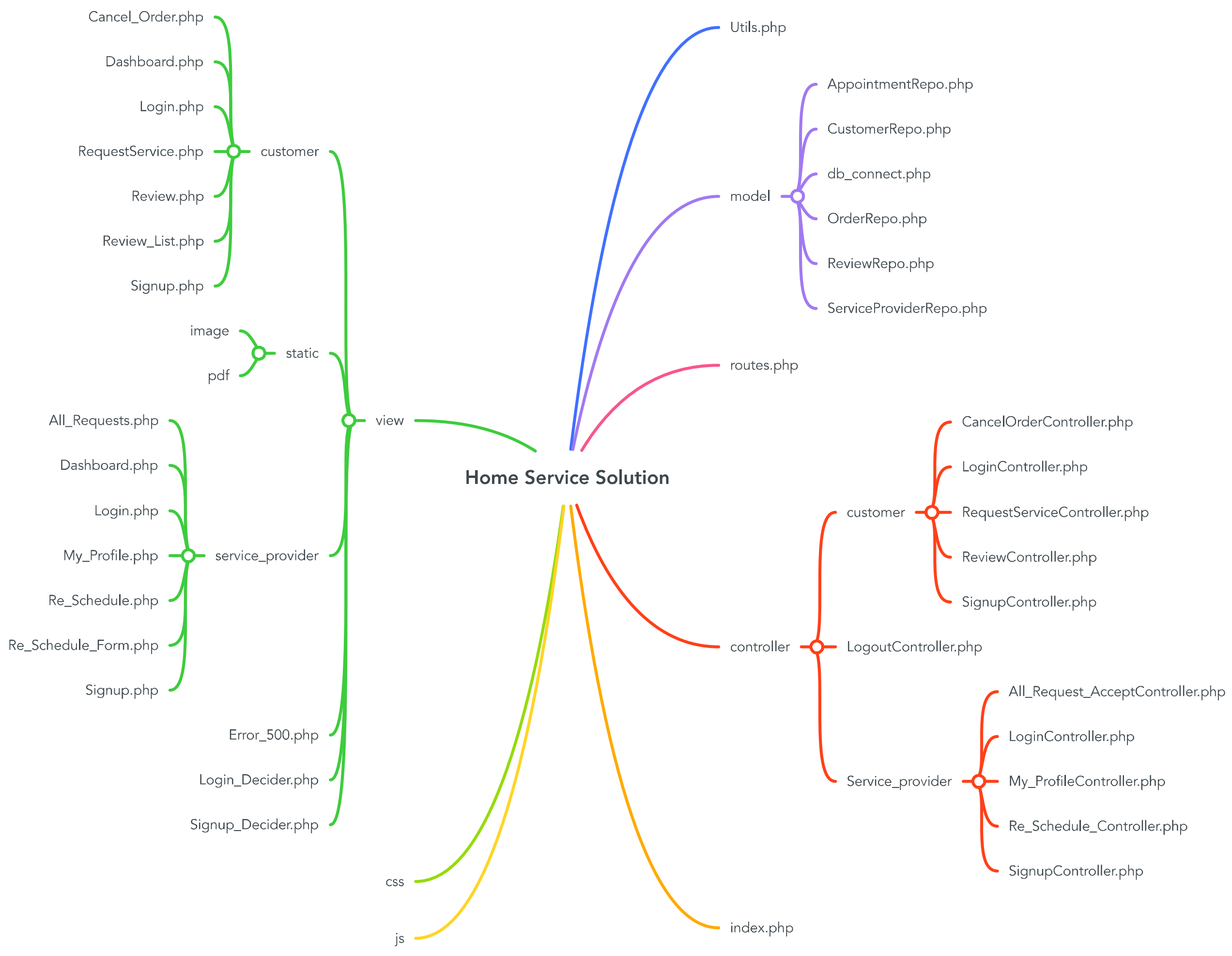


Figure 3.1: File Distributions in MVC[2] Structure

3.2 Data Flow

3.2.1 Level 0

Data flow of level-0[5] is the top level view. It actually encapsulate the whole system and shows only the main process. Here, as there are only one user(seller) currently, that is the reason there is only one main process in the system, which is to serve the seller.

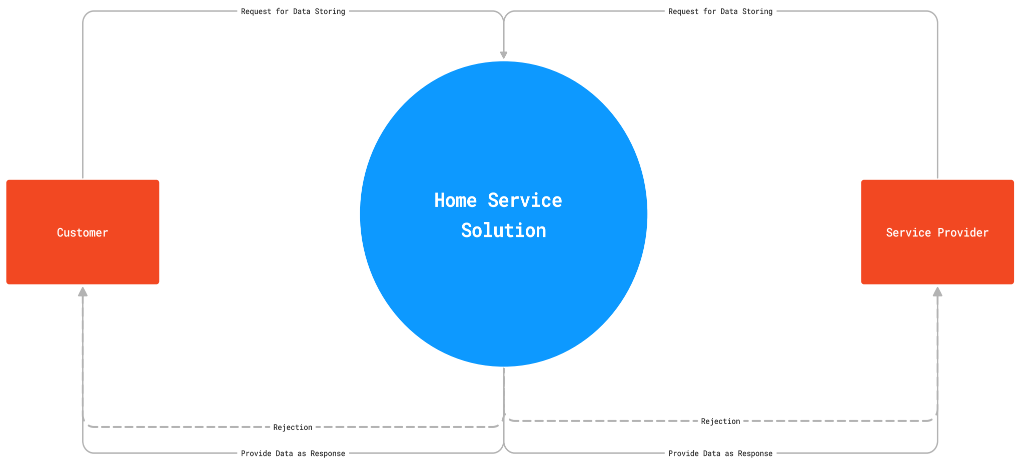


Figure 3.2: Data Flow Diagram Level - 0

3.2.2 Level 1

Data flow of level-1 [5] is the lower-middle level view. It shows a detailed view of the data flow through different functions for each user. For this project it is, seller.

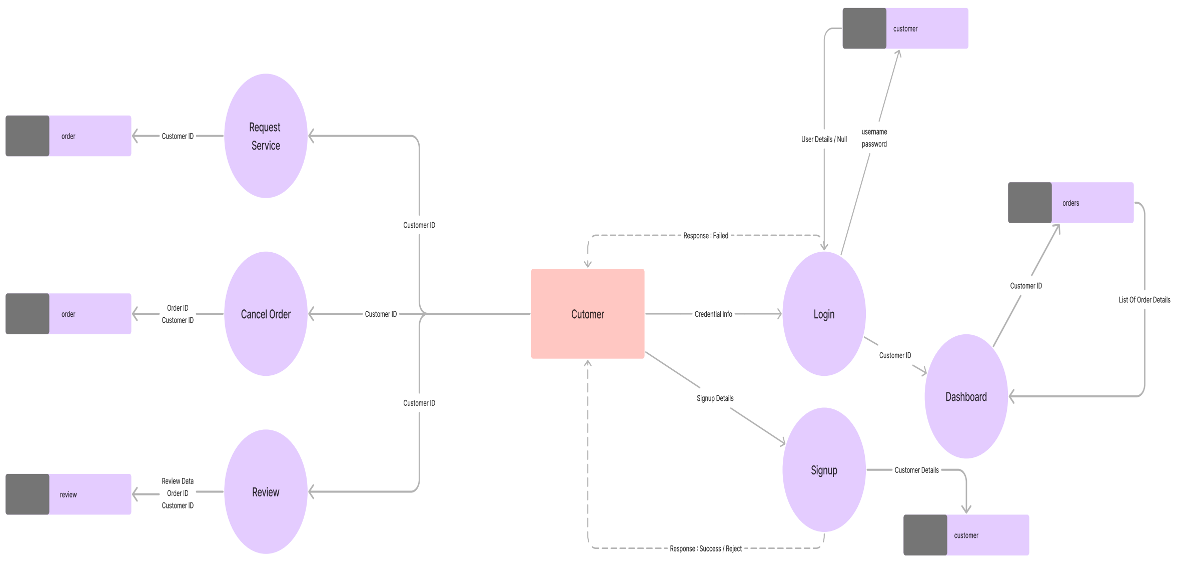


Figure 3.3: Data Flow Diagram Level - 1 Customer

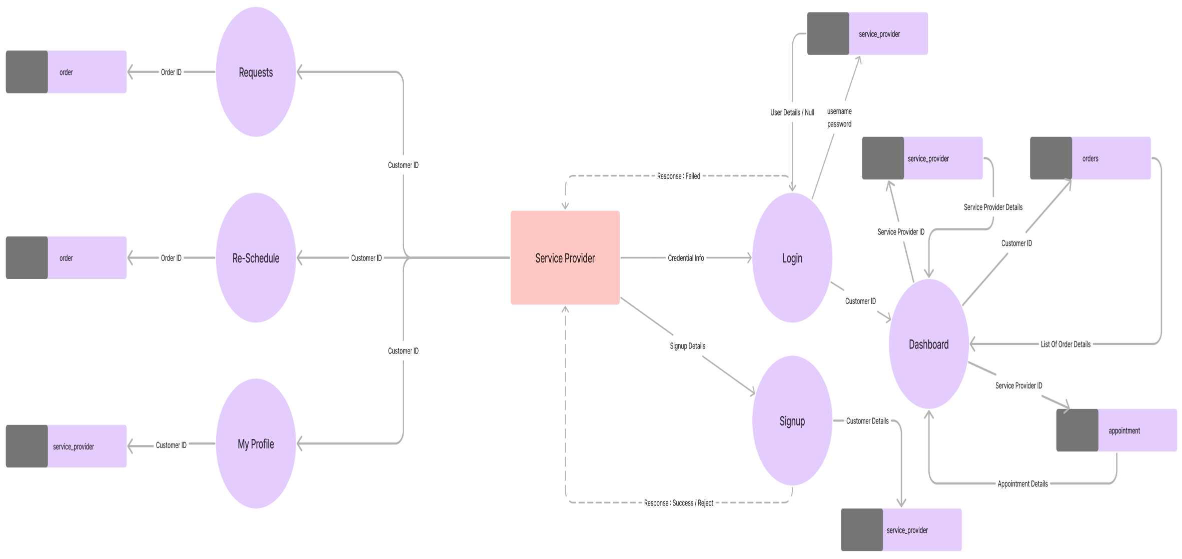


Figure 3.4: Data Flow Diagram Level - 1 Service Provider

3.3 Work Flow

Work flow diagram[5] is a simple step by step process a user can visit or do. It is used to give a detailed view how a user will use the product.

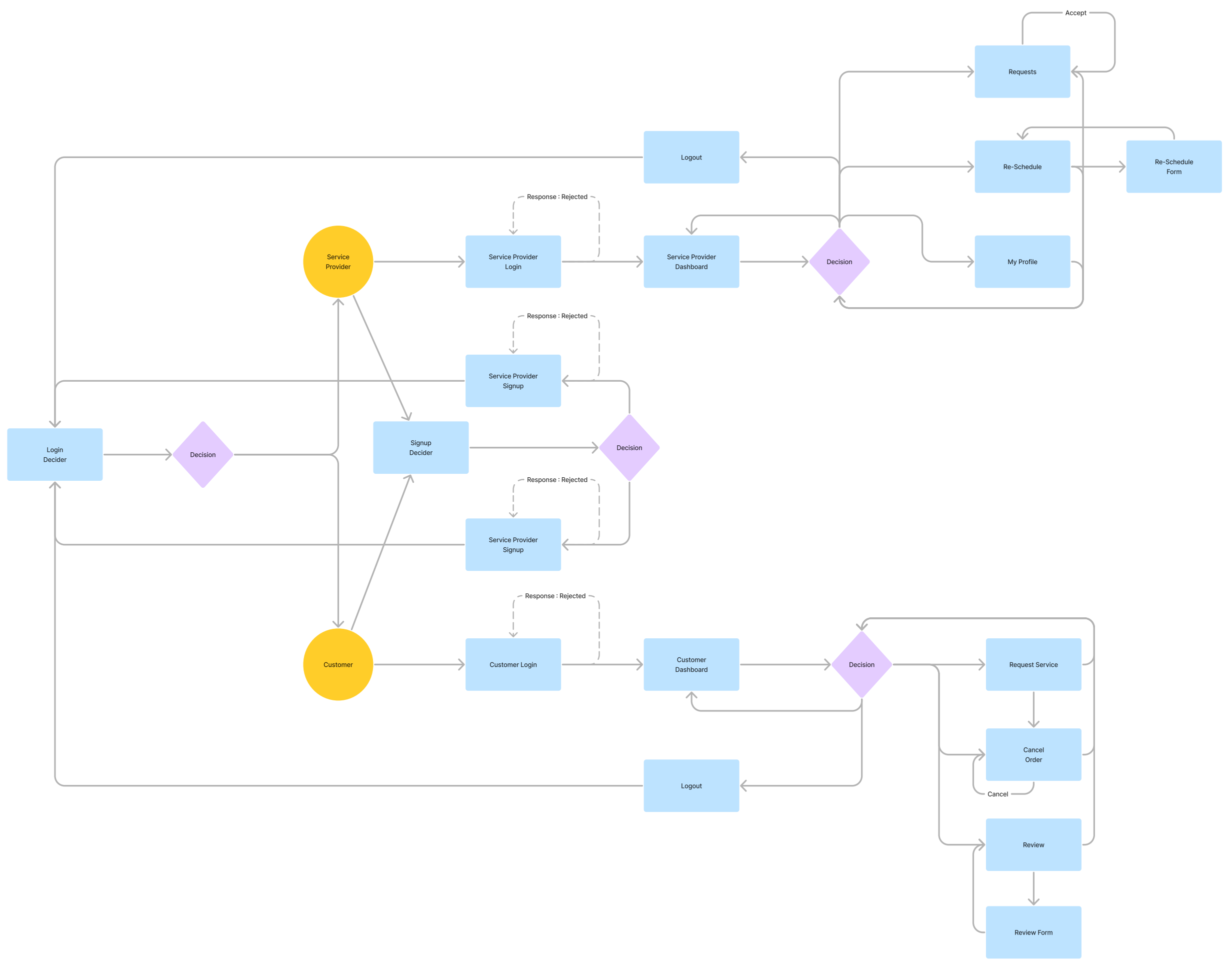
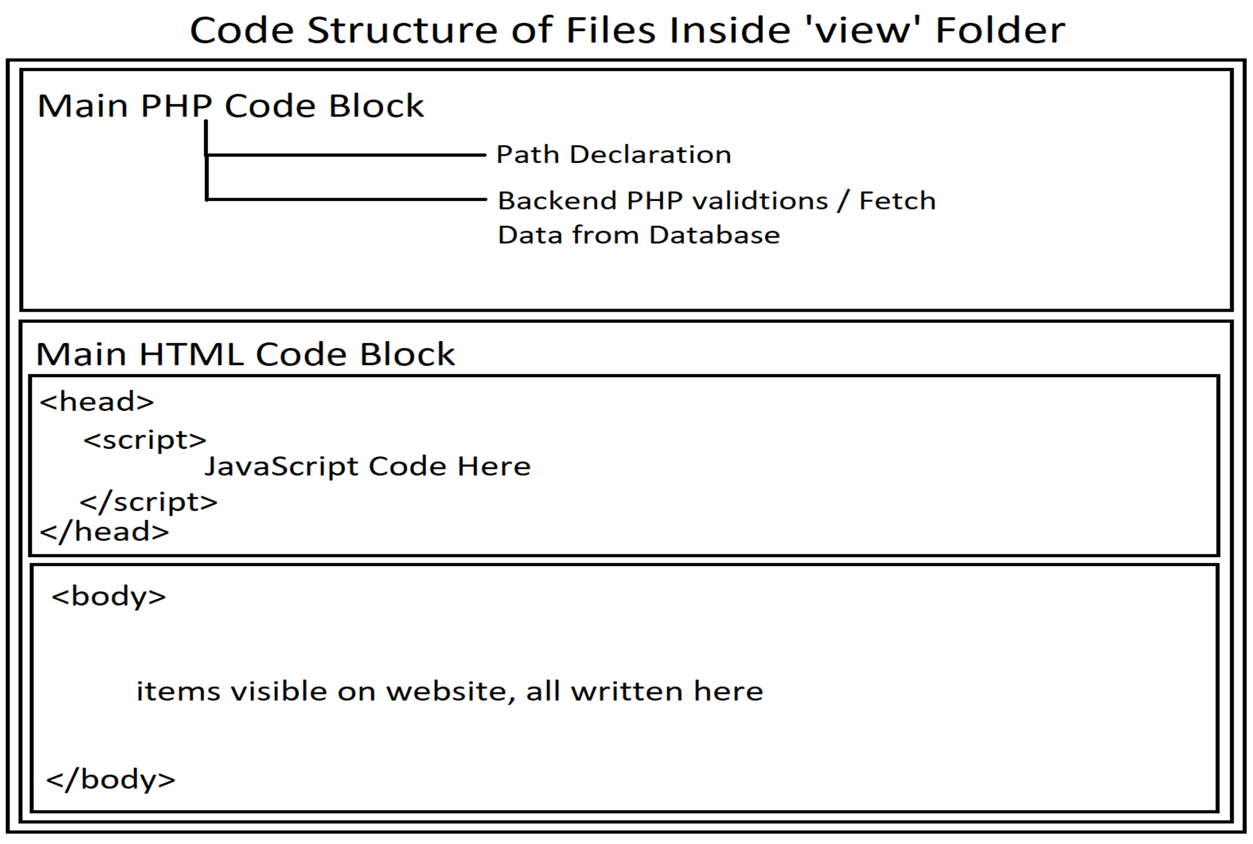


Figure 3.5: Work Flow Diagram

4 Code Structure

4.1 Directory Structure

4.1.1 View



Code Structure of Files Inside 'view' Folder
Main PHP Code Block
Path Declaration
Backend PHP validtions / Fetch Data from Database
Main HTML Code Block
<head>
<script>
JavaScript Code Here
</script>
</head>
<body>
items visible on website, all written here
</body>

Figure 4.1: View Folder Code Structure

4.1.2 Controller

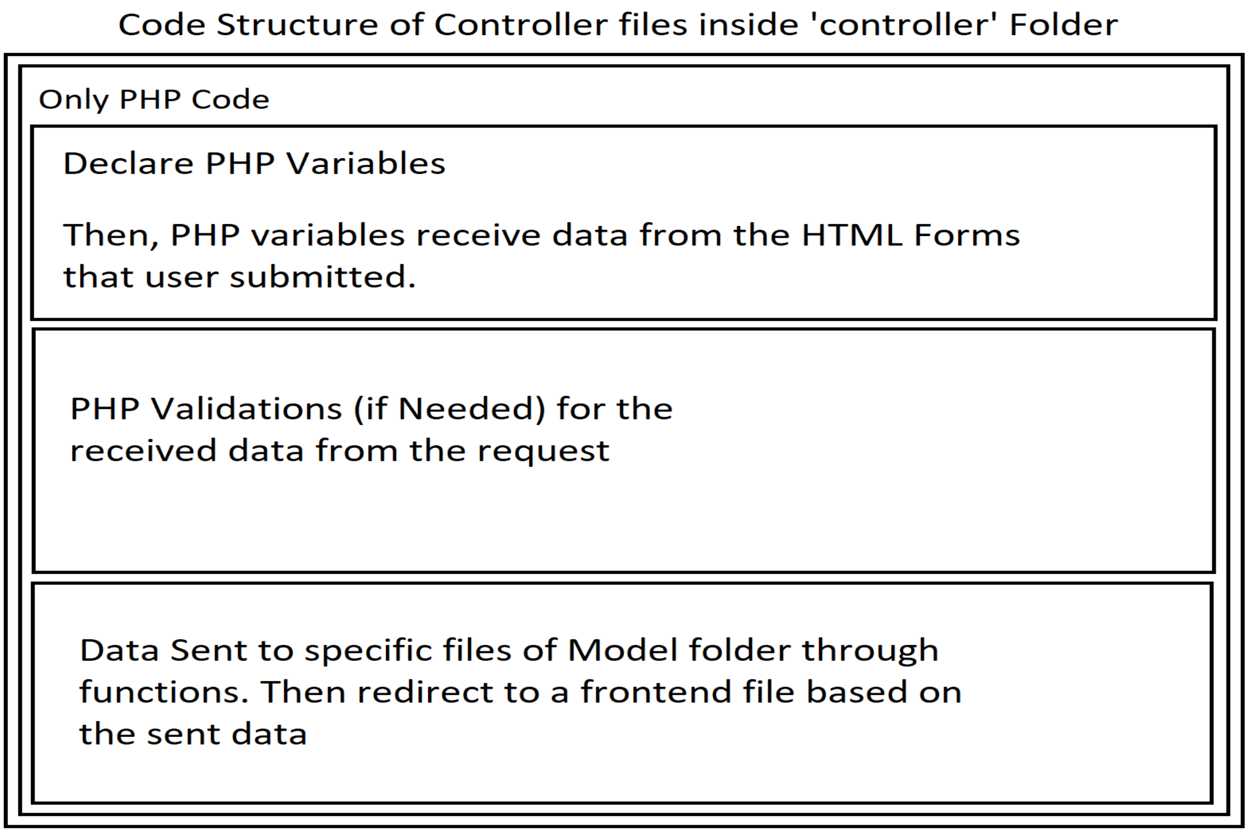

Code Structure of Controller files inside 'controller' Folder
Only PHP Code
Declare PHP Variables
Then, PHP variables receive data from the HTML Forms that user submitted.
PHP Validations (if Needed) for the received data from the request
Data Sent to specific files of Model folder through functions. Then redirect to a frontend file based on the sent data

Figure 4.2: Controller Folder Code Structure

4.1.3 Model

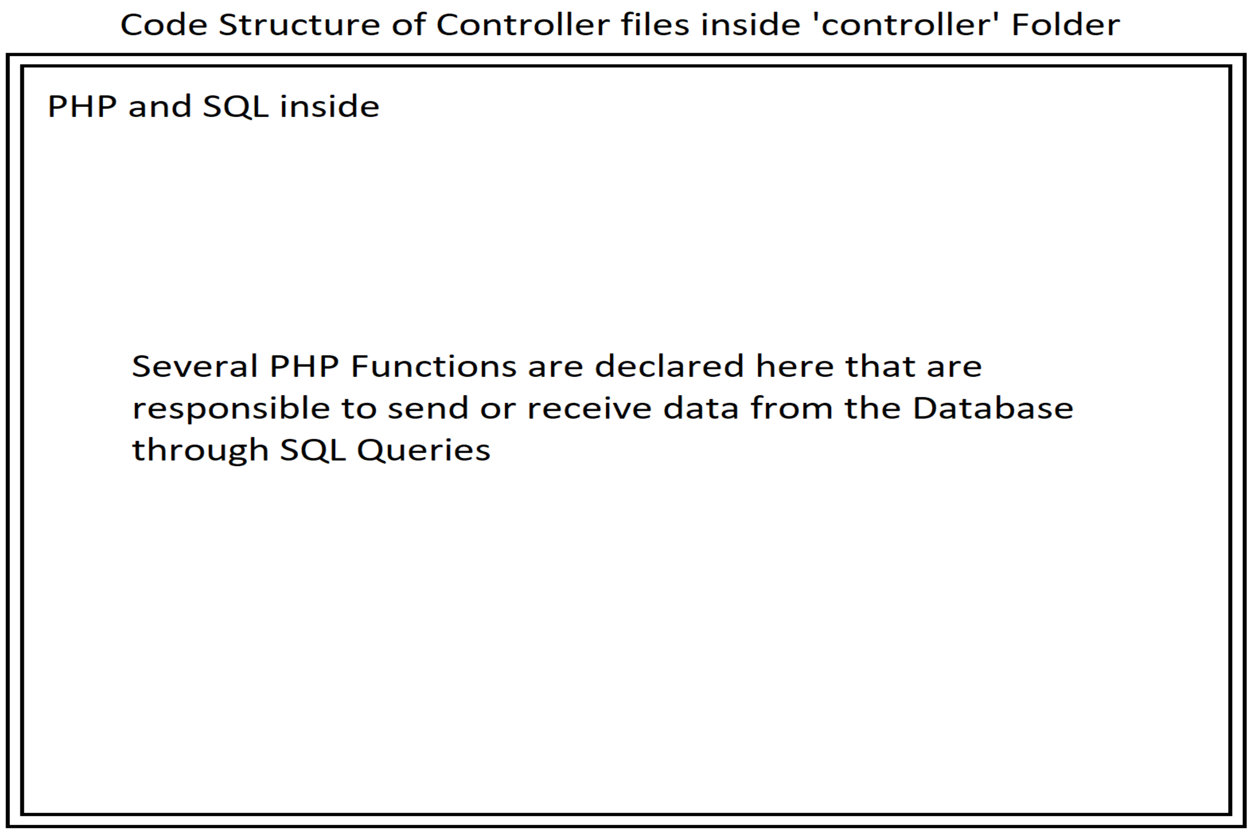

Code Structure of Controller files inside 'controller' Folder
PHP and SQL inside
Several PHP Functions are declared here that are responsible to send or receive data from the Database through SQL Queries

Figure 4.3: Model Folder Code Structure

4.2 Modules and Components

- Session
- PHP Validation
- JavaScript Validation
- Bootstrap CSS

4.2.1 Session

```php
<?php
session_start();
$_SESSION['variable_name'];
?>
```

Listing 4.1: Session Code

To use Session or Session Variables (also known as global variables), we have start the Session first by using session_start();. And to use session variables, we use $_SESSION['variable_name'];

4.2.2 PHP Validation

```php
<?php
$username = $_POST['username'];
if (empty($username)) {
    $everythingOK = FALSE;
    $everythingOKCounter += 1;
    echo '<br>username Error 1<br>';
} elseif (!preg_match('/^[a-zA-Z0-9_]+$/', $username)) {
    $everythingOK = FALSE;
    $everythingOKCounter += 1;
    echo '<br>username Error 2<br>';
} else {
    $everythingOK = TRUE;
}
?>
```

Listing 4.2: PHP Validation Code Example

PHP validations has been used in almost all Controller files where the requests got received.

4.2.3 JavaScript Validation

```
function showModal(message) {
    document.getElementById("validationMessage").innerHTML = message;
    document.getElementById("validationModal").style.display = "block";
}

close_modal = () => {
    document.getElementById("validationModal").style.display = "none";
    // location.reload(true);
}

function validateForm() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    // var emailErrorLabel = document.getElementById("loginEmailError");

    var alphanumericRegex = /^[a-zA-Z0-9]+$/;
    // Check if the username is empty or null
    if (username === "" || username === null) {
        showModal("Username is Required");
        return false;
    }else if (!alphanumericRegex.test(username)) {
        showModal("Username should contain only alphanumeric
            characters.");
        return false;
    }

    if (password === "" || password === null) {
        showModal("Password is Required");
        return false;
    }else if (password.length < 8) {
        showModal("Password must be at least 8 characters long.");
        return false;
    } else if (!password.match(/^(?=.*[a-z])(?=.*[A-Z])
        (?=.*\d)(?=.*[^a-zA-Z0-9]).{8,}$/)) {
        showModal("Password must include at least 1 special
            character, 1 number, 1 small letter, and 1 capital letter.");
        return false;
    }

    return true;
}
```

Listing 4.3: JavaScript Validation Code

We can find javaScript validations only in the files of the view folder. More Specifically inside the <script></script> tag of the HTML code.

4.2.4 Bootstrap CSS

We can find all the CSS code along with bootstrap CSS classes declarations inside view > css folder. And these codes are used in every single files inside the view folder, HTML codes.

5 Database

5.1 Tables

5.1.1 customer

- id (PK)
- name
- address
- username
- password

5.1.2 service_provider

- id (PK)
- username
- password
- status
- profile_mode

5.1.3 appointment

- id (PK)
- date
- service_provider_id (FK)
- order_id (FK)

5.1.4 orders

- id (PK)
- type
- address
- customer_id (FK)
- service_provider_id (FK)
- status

5.1.5 payment

- id (PK)
- status
- order_id (FK)
- customer_id (FK)
- service_provider_id (FK)

5.1.6 review

- id (PK)
- rating
- comment
- customer_id (FK)
- order_id (FK)

5.2 Relationships

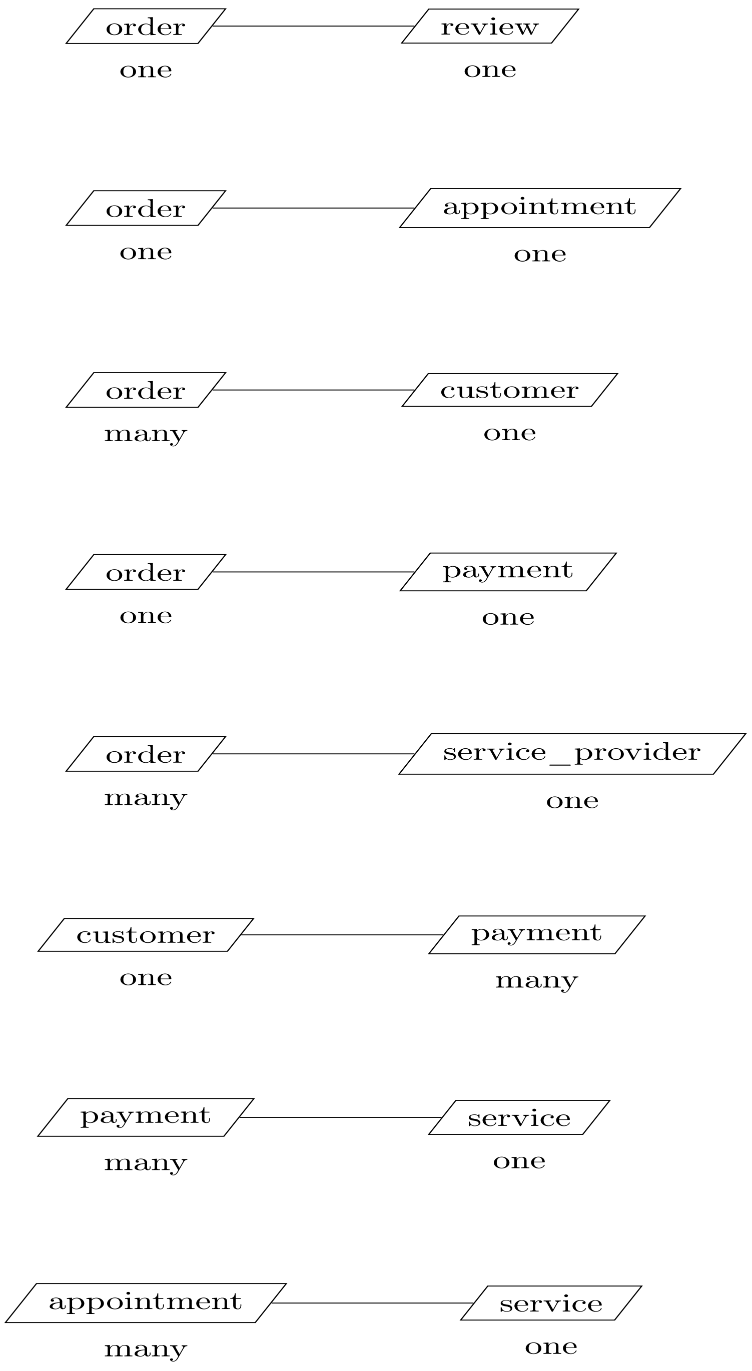


Figure 5.1: Relationship Simplification between tables

5.3 Entity-Relationship (ER) [1] Diagram

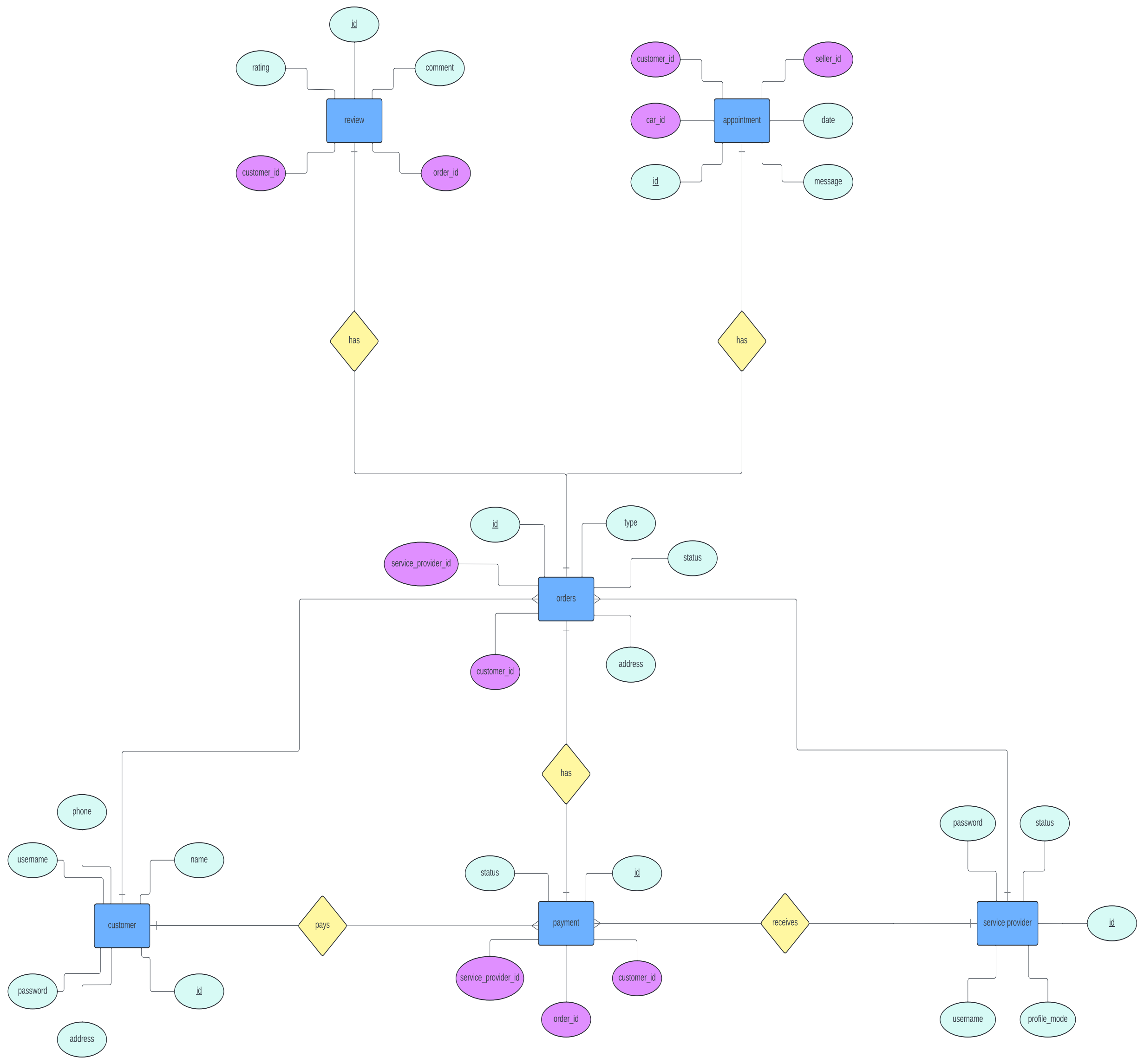


Figure 5.2: ER [1] Diagram

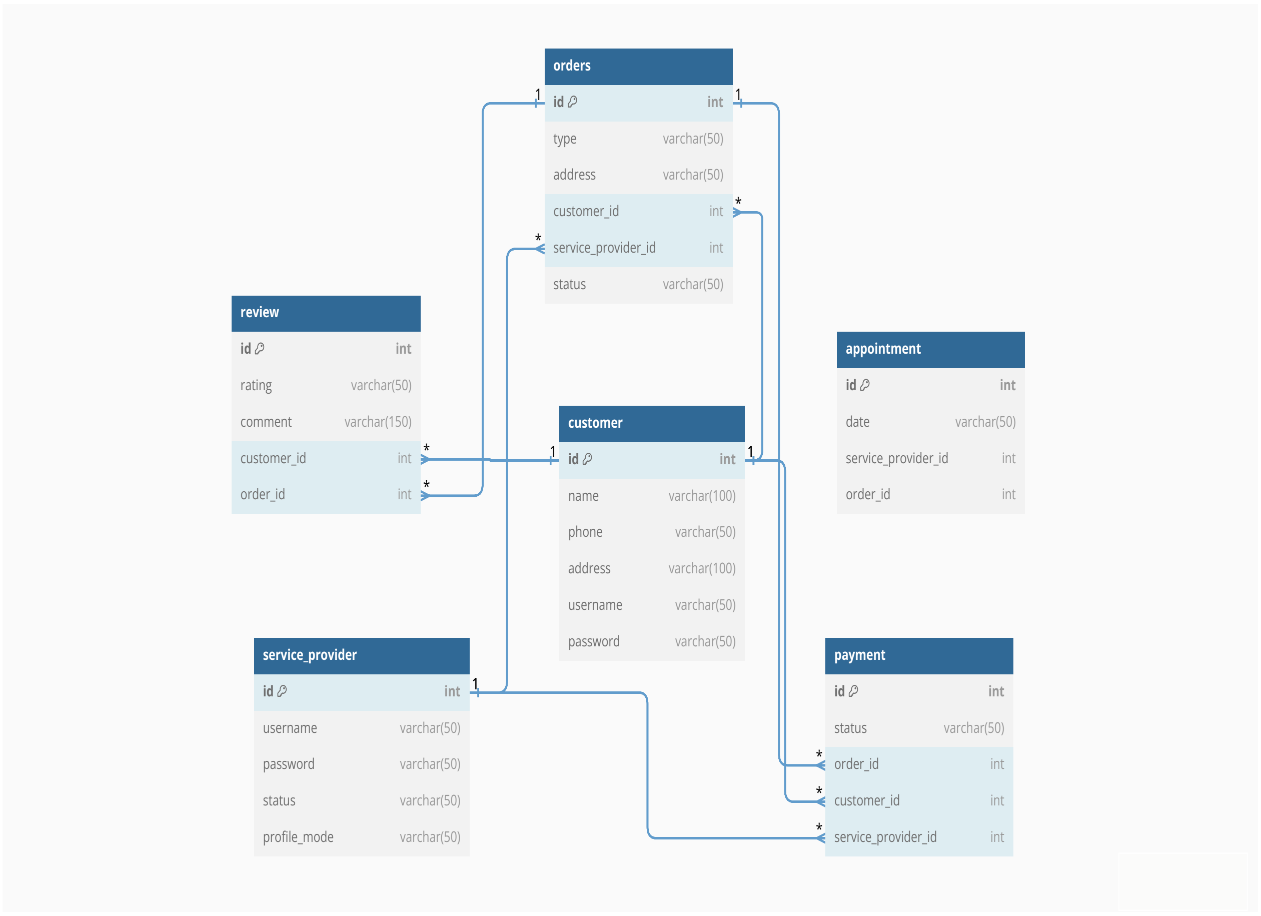


Figure 5.3: ER Diagram [Detailed[3] Version]

6 Features and Functionalities

6.1 Feature List

Customer

- Customer Signup
- Customer Login
- Customer Logout
- Send Service Orders
- Cancel Orders
- Provide Ratings & Review

Service Provider

- Service Provider Signup
- Service Provider Login
- Service Provider Logout
- Boosting account
- View All Requests in a Request Pool
- Accept a Request
- Re-Schedule an Accepted Request
- Show Own Profile
- Show Total Request Completed, Total Income, Upcoming Next Appointment

6.2 Feature 1: Customer Signup

6.2.1 User Type

Customer

6.2.2 Feature Description

Allows a new customer to create an account by providing necessary information and agreeing to the platform's terms and conditions.

6.2.3 Functionalities

CRUD Functionalities

- Create

Other Functionalities

- JavaScript Validation
- PHP Validation

6.2.4 Associated Files

- Associated View (Front-end) : Signup.php
- Associated Controller : SignupController.php
- Associated Model : CustomerRepo.php

6.3 Feature 2: Customer Login

6.3.1 User Type

Customer

6.3.2 Feature Description

Enables a customer to access their account by entering their username and password.

6.3.3 Functionalities

CRUD Functionalities

- Read

Other Functionalities

- JavaScript Validation
- PHP Validation
- Session

6.3.4 Associated Files

- Associated View (Front-end) : Login.php
- Associated Controller : LoginController.php
- Associated Model : CustomerRepo.php

6.4 Feature 3: Customer Logout

6.4.1 User Type

Customer

6.4.2 Feature Description

All type of users need to logout to exit the website.

6.4.3 Functionalities

CRUD Functionalities

➤ NONE

Other Functionalities

- Delete Session

6.4.4 Associated Files

- Associated View (Front-end) :
  Review_List.php, Review.php, Cancel_Order.php, Dashboard.php, RequestService.php
- Associated Controller : LogoutController.php
- Associated Model : NONE

6.5 Feature 4: Send Service Orders

6.5.1 User Type

Customer

6.5.2 Feature Description

Send Service Orders allows customers to submit service requests seamlessly through the system

6.5.3 Functionalities

CRUD Functionalities

- Create

Other Functionalities

- JavaScript Validation
- PHP Validation
- Session

6.5.4 Associated Files

- Associated View (Front-end) : RequestService.php
- Associated Controller : RequestServiceController.php
- Associated Model : OrderRepo.php

6.6 Feature 5: Cancel Orders

6.6.1 User Type

Customer

6.6.2 Feature Description

Cancel Orders enables sellers to easily retract or cancel pending orders within the system."

6.6.3 Functionalities

CRUD Functionalities

- Update

Other Functionalities

- Session

6.6.4 Associated Files

- Associated View (Front-end) : Cancel_Order.php
- Associated Controller : CancelOrderController.php
- Associated Model : OrderRepo.php

6.7 Feature 6: Provide Ratings & Review

6.7.1 User Type

Customer

6.7.2 Feature Description

Provide Ratings & Review empowers sellers to rate and review their interactions with customers, enhancing transparency and trust within the system.

6.7.3 Functionalities

CRUD Functionalities

- Read
- Create

Other Functionalities

- JavaScript Validation
- PHP Validation
- Session

6.7.4 Associated Files

- Associated View (Front-end) : Review_List.php, Review.php
- Associated Controller : ReviewController.php
- Associated Model : ReviewRepo.php

6.8 Feature 7: Service Provider Signup

6.8.1 User Type

Service Provider

6.8.2 Feature Description

Allows a new service provider to create an account by providing necessary information and agreeing to the platform's terms and conditions.

6.8.3 Functionalities

CRUD Functionalities

- Create

Other Functionalities

- JavaScript Validation
- PHP Validation

6.8.4 Associated Files

- Associated View (Front-end) : Signup.php
- Associated Controller : SignupController.php
- Associated Model : ServiceProviderRepo.php

6.9 Feature 8: Service Provider Login

6.9.1 User Type

Service Provider

6.9.2 Feature Description

Enables a service provider to access their account by entering their username and password.

6.9.3 Functionalities

CRUD Functionalities

- Read

Other Functionalities

- JavaScript Validation
- PHP Validation
- Session

6.9.4 Associated Files

- Associated View (Front-end) : Login.php
- Associated Controller : LoginController.php
- Associated Model : CustomerRepo.php

6.10 Feature 9: Boosting account

6.10.1 User Type

Service Provider

6.10.2 Feature Description

Boosting Account allows service providers to enhance their account visibility and reach more customers through promotional tools.

6.10.3 Functionalities

CRUD Functionalities

➤ NONE

Other Functionalities

- Delete Session

6.10.4 Associated Files

- Associated View (Front-end) : Dashboard.php, All_Requests.php, My_Profile.php, Re_Schedule.php, Re_Schedule_Form.php,
- Associated Controller : LogoutController.php
- Associated Model : NONE

6.11 Feature 10: View All Requests in a Request Pool

6.11.1 User Type

Service Provider

6.11.2 Feature Description

View All Requests in a Request Pool enables service providers to browse and select from a comprehensive list of service requests.

6.11.3 Functionalities

CRUD Functionalities

- Read

Other Functionalities

- Session

6.11.4 Frontend

- Associated View (Front-end) : All_Requests.php
- Associated Controller : NONE
- Associated Model : NONE

6.12 Feature 11: Accept a Request

6.12.1 User Type

Service Provider

6.12.2 Feature Description

Accept a Request allows service providers to review and accept service requests from customers efficiently.

6.12.3 Functionalities

CRUD Functionalities

- Update

Other Functionalities

- Session

6.12.4 Frontend

- Associated View (Front-end) : All_Requests.php
- Associated Controller : All_Request_AcceptController.php
- Associated Model : OrderRepo.php

6.13 Feature 12: Re-Schedule an Accepted Request

6.13.1 User Type

Service Provider

6.13.2 Feature Description

Re-Schedule an Accepted Request enables service providers to modify the timing of previously accepted service requests for better flexibility

6.13.3 Functionalities

CRUD Functionalities

- Update

Other Functionalities

- Session

6.13.4 Frontend

- Associated View (Front-end) : Re_Schedule.php, Re_Schedule_Form.php
- Associated Controller : Re_Schedule_Controller.php
- Associated Model : AppointmentRepo.php

6.14 Feature 13: Show Own Profile

6.14.1 User Type

Service Provider

6.14.2 Feature Description

Show Own Profile allows service providers to view and manage their personal and professional information within the system.

6.14.3 Functionalities

CRUD Functionalities

- Read

Other Functionalities

- Session

6.14.4 Associated Files

- Associated View (Front-end) : My_Profile.php
- Associated Controller : My_Profile.php
- Associated Model : ServiceProviderRepo.php

6.15 Feature 14: Show Total Request Completed, Total Income, Upcoming Next Appointment

6.15.1 User Type

Service Provider

6.15.2 Feature Description

This feature allows service providers to view their total completed requests, total income earned, and their upcoming appointment.

6.15.3 Functionalities

CRUD Functionalities

- Read

Other Functionalities

- Session

6.15.4 Associated Files

- Associated View (Front-end) : Dashboard.php
- Associated Controller : Dashboard.php
- Associated Model : CalculationRepo.php

7 Equations

7.1 Accepted Orders

The number of accepted orders is the count of orders with the status ‘Accepted’.

$$\text{Accepted\_Orders} = \sum_{\text{orders}} (\text{status} == \text{‘Accepted’}) \tag{7.1}$$

7.2 Total Income

Total income is calculated based on the service type provided.

$$\text{Total\_Income} = \sum_{\text{orders}} (\text{income\_for\_type}(\text{type})) \tag{7.2}$$

Where the income for each type is defined as:

$$\text{income\_for\_type}(\text{type}) = \begin{cases} 2000 & \text{if type} = \text{‘AC Servicing’} \\ 1500 & \text{if type} = \text{‘Home Cleaning’} \\ 3000 & \text{if type} = \text{‘Plumbing and Sanitation’} \\ 5000 & \text{if type} = \text{‘House Shifting’} \\ 0 & \text{otherwise} \end{cases}$$

7.3 Upcoming Appointment

The upcoming appointment is the next appointment date that is after the current date, formatted as ’jS F Y’.

$$\text{Upcoming\_Appointment} = \min_{\text{orders}} (\text{date} > \text{current\_date}) \tag{7.3}$$

8 Error Handling

8.1 PHP Error Handling

```php
<?php
header("Location: {$another_page}")
?>
```

Listing 8.1: PHP Error Handling Code Example

If any error occurs or mis-validation occurs in the project, then

- The PHP code of the Controller files redirect to previous view file. That means it will take user to the page from where the request has been received in the Controller file.
- The PHP code of the View files redirect to Login Page.
- The PHP code of the db_connect.php file, which is also the Database connection file, uses 'die()' function to handle the error.

9 Testing

9.1 Test Case: Customer Signup

Description

Create an Account in the website

Test Steps

1. Click the Customer card on the Login Decider
2. Click the Signup link on the Login.php
3. Click the Customer card on the Signup Decider
4. Fill-up all the information
5. Click Signup Button

Test Data

- Username : test123
- Password : testPass1@
- Name : Test Name
- Phone : 019231414986
- Address : Test Address

Expected Output

Login Decider Page can be seen

Actual Output

Login Decider Page is visible

Result

PASS

Notes

To Signup properly, every data must be filled up and also check the terms and conditions checkbox. If all data is not filled up and user clicks register button, a message pops up.

9.2 Test Case: Customer Login

Description

Login to the System

Test Steps

1. Click the Customer card on the Login Decider
2. Fill-up all the information
3. Click Login Button

Test Data

- Username : test123
- Password : testPass1@

Expected Output

Customer Dashboard Page can be seen

Actual Output

Customer Dashboard Page is visible

Result

PASS

Notes

If all data is not filled up and user clicks login button, a message pops up.

9.3 Test Case: Customer Logout

Description

Logout from the account

Test Steps

1. Click ‘Logout’ from the Navigation Panel on the left side

Test Data

Expected Output

Login Decider Page can be seen

Actual Output

Login Decider Page is visible

Result

PASS

Notes

If there is any problem with the login page path, only then, any error can be occurred

9.4 Test Case: Send Service Orders

Description

Send a Request for a service as an order.

Test Steps

1. Click on 'Request Service' button from the left side drawer
2. Provide the necessary data
3. Click 'Place Order' button.

Expected Output

Cancel Order Page can be seen

Actual Output

Cancel Order Page is visible

Result

PASS

Notes

If the the necessary data is not being provided properly, then there should be a error box on the top right corner with the error details

9.5 Test Case: Cancel Orders

Description

If customer wants, he/she can cancel the orders before completing it.

Test Steps

1. Click on 'Cancel Order' button from the left side drawer
2. Click on the 'Cancel' button on the order that needs to be cancelled

Expected Output

Cancel Order Page can be seen and the cancelled order should not be seen

Actual Output

Cancel Order Page can be seen and the cancelled order is not visible

Result

PASS

Notes

Cancel Order page can be empty and provides potential errors if the user tries to show the page without login or by any other means.

9.6 Test Case: Provide Ratings and Reviews

Description

Customer can only provide reviews those which order has been Completed and customer can provide reviews only once for an order.

Test Steps

1. Click on ‘Review’ button from the left side drawer
2. Click on ‘Give Review’ button on the order which review that the customer wants to provide
3. Fill up the form
4. Click on ‘Send Review’ button

Expected Output

Review Page can be seen and the order which review has been provided, should not be seen

Actual Output

Review Page is visible and the order which review has been provided, is not visible

Result

PASS

Notes

There should be an error if the necessary data is not being provided

9.7 Test Case: Service Provider Signup

Description

Create an Account in the website

Test Steps

1. Click the Service Provider card on the Login Decider
2. Click the Signup link on the Login.php
3. Click the Service Provider card on the Signup Decider
4. Fill-up all the information
5. Click Signup Button

Test Data

- Username : test456
- Password : testPass1@

Expected Output

Login Decider Page can be seen

Actual Output

Login Decider Page is visible

Result

PASS

Notes

To Signup properly, every data must be filled up and also check the terms and conditions checkbox. If all data is not filled up and user clicks register button, a message pops up.

9.8 Test Case: Service Provider Login

Description

Login to the System

Test Steps

1. Click the Customer card on the Login Decider
2. Fill-up all the information
3. Click Login Button

Test Data

- Username : test123
- Password : testPass1@

Expected Output

Service Provider Dashboard should can be seen

Actual Output

Service Provider Dashboard Page is visible

Result

PASS

Notes

If all data is not filled up and user clicks login button, a message pops up.

9.9 Test Case: Service Provider Logout

Description

Logout from the account

Test Steps

1. Click 'Logout' from the Navigation Panel on the left side

Test Data

Expected Output

Login Decider Page can be seen

Actual Output

Login Decider Page is visible

Result

PASS

Notes

If there is any problem with the login page path, only then, any error can be occurred

9.10 Test Case: Boosting account

Description

Service Provider sends requests for a pro account.

Test Steps

1. Click on ‘My Profile’ button from the left side drawer
2. Click on ‘Apply for Pro Mode’ button.

Expected Output

My Profile Page can be seen

Actual Output

My Profile Page is visible

Result

PASS

Notes

There should be an error if any user tries to show the page without login.

9.11 Test Case: View All Requests in a Request Pool

Description

Each Service Provider can see all the requests in a list so that he/she can accept any request that he/she likes.

Test Steps

1. Click on 'Requests' button from the left side drawer

Expected Output

All the pending Orders and the addresses should be seen.

Actual Output

All the pending Orders and the addresses is visible

Result

PASS

Notes

The Order's list can not be seen if there is any database issue or nobody has send any order requests till visiting that page

9.12 Test Case: Accept a Request

Description

Service Provider can accept any request given by any customer

Test Steps

1. Login to the account or click on 'Requests' button from the left side drawer
2. Click the 'Accept' button of the order which needs to be accepted

Expected Output

Request Page can be seen and the accepted order should not be seen.

Actual Output

Request Page is visible and the accepted order is not visible

Result

PASS

Notes

There should be an error if the order id is not found in the backend which the user have accepted

9.13 Test Case: Re-Schedule an Accepted Request

Description

Service provider has the ability to change the date to deliver the service

Test Steps

1. Click on ‘Re-Schedule’ button from the left side drawer
2. Click ‘Re-Schedule’ Button
3. Fillup the form
4. Click ‘Re-Schedule IT’ button

Test Data

- Date : 07/15/2024

Expected Output

‘Re-Schedule’ page can be seen and the Updated date can be seen

Actual Output

‘Re-Schedule’ page is visible and the Updated date is visible

Result

PASS

Notes

An error can be visible if the id of the order is not found in the backend which the service provider wants to re-schedule

9.14 Test Case: Show Own Profile

Description

A Service Provider can see own profile details.

Test Steps

1. Click on ‘My Profile’ button from the left side drawer

Expected Output

‘My Profile’ page can be seen along with the profile details

Actual Output

‘My Profile’ page is visible along with the profile details

Result

PASS

Notes

There should be an error if the user somehow manages to show the page before login into the account.

9.15 Test Case: Show Total Request Completed, Total Income, Upcoming Next Appointment

Description

Edit any car details for the seller's car shop that is logged in currently

Test Steps

1. Login to the account and or click on 'dashboard' button from the left side drawer

Test Data

Expected Output

'Dashboard' page can be seen

Actual Output

'Dashboard' page is visible

Result

PASS

Notes

Dashboard information's can not be seen if any user somehow manages to view the page without login or if the user_id has not been saved properly

10 Code Examples and Snippets

10.1 Routes

```php

<?php

// Define routes
$routes = [
    'INDEX' => '/Home_Service_Solution/index.php',
    '500_error' => '/Home_Service_Solution/view/Error_500.php',

    'login_decider' => '/Home_Service_Solution/view/Login_Decider.php',
    'signup_decider' => '/Home_Service_Solution/view/Signup_Decider.php',

    'customer_login' => '/Home_Service_Solution/view/customer/Login.php',
    'service_provider_login' => '/Home_Service_Solution/view/service_provider/
    Login.php',

    'customer_dashboard' => '/Home_Service_Solution/view/customer/Dashboard.php',
    'service_provider_dashboard' => '/Home_Service_Solution/view/service_provider/
    Dashboard.php',

    'customer_signup' => '/Home_Service_Solution/view/customer/Signup.php',
    'service_provider_signup' => '/Home_Service_Solution/view/service_provider/
    Signup.php',

//  Customer

    'customer_request' => '/Home_Service_Solution/view/customer/
    RequestService.php',
    'customer_cancel_order' => '/Home_Service_Solution/view/customer/
    Cancel_Order.php',
    'customer_review_form' => '/Home_Service_Solution/view/customer/Review.php',
    'customer_review_list' => '/Home_Service_Solution/view/customer/
    Review_List.php',

//    Service-Provider

    'service_provider_all_requests' => '/Home_Service_Solution/view/
    service_provider/All_Requests.php',
    'service_provider_re_schedule' => '/Home_Service_Solution/view/
    service_provider/Re_Schedule.php',
    'service_provider_re_schedule_form' => '/Home_Service_Solution/view/
    service_provider/Re_Schedule_Form.php',
    'service_provider_my_profile' => '/Home_Service_Solution/view/
    service_provider/My_Profile.php',

];

?>
```

Listing 10.1: Project File Routes

```

<?php

$backend_routes = [
    'customer_login_controller' => '/Home_Service_Solution/controller/customer/
    LoginController.php',
    'service_provider_login_controller' => '/Home_Service_Solution/controller/
    Service_provider/LoginController.php',

    'customer_signup_controller' => '/Home_Service_Solution/controller/customer/
    SignupController.php',
    'service_provider_signup_controller' => '/Home_Service_Solution/controller/
    Service_provider/SignupController.php',

    'logout_controller' => '/Home_Service_Solution/controller/
    LogoutController.php',

    'request_service_controller' => '/Home_Service_Solution/controller/customer/
    RequestServiceController.php',

//     Customer

    'customer_cancel_order_controller' => '/Home_Service_Solution/controller/
    customer/
    CancelOrderController.php',
    'review_controller' => '/Home_Service_Solution/controller/customer
    /ReviewController.php',

//     Service-Provider

    'service_provider_all_request_accept_controller' => '/Home_Service_Solution/
    controller/Service_provider/All_Request_AcceptController.php',
    'my_profile_controller' => '/Home_Service_Solution/controller/
    Service_provider/My_ProfileController.php',
    're_schedule_controller' => '/Home_Service_Solution/controller/
    Service_provider/Re_Schedule_Controller.php',



];


$image_routes = [
    'user_icon' => '/Home_Service_Solution/view/static/image/user.png',
    'customer_icon' => '/Home_Service_Solution/view/static/image/customer.png',
    'service_provider_icon' => '/Home_Service_Solution/view/static/image/
    service_provider.png',

    'logo_icon' => '/Home_Service_Solution/view/static/image/logo.png',
];


?>
```

Listing 10.2: Project File Routes

Definitions

I.A Session

A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the users computer. Session variables hold information about one single user, and are available to all pages in one application. A session is started with the session_start() function. Session variables are set with the PHP global variable: $_SESSION. The session_start() function must be the very first thing in your document. Before any HTML tags.

I.B Cookies

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values. setcookie(name, value, expire, path, domain, secure, httponly);

I.C Validation

Validation in JavaScript and PHP refers to the process of ensuring that data entered by users meets specified criteria, such as format, type, or range, before it is processed or submitted. In JavaScript, this typically involves using functions or regular expressions to check input fields on the client side, providing immediate feedback to users. In PHP, validation often occurs on the server side, where data is checked after it's submitted, ensuring security and consistency in data handling. Both JavaScript and PHP validations aim to enhance the user experience and protect against errors and malicious inputs.

I.D Difference between Bootstrap and Raw CSS

Bootstrap and raw CSS represent two different approaches to styling and structuring web content. Bootstrap is a popular front-end framework that offers pre-designed CSS and JavaScript components, providing a ready-made toolkit for building responsive and visually appealing websites with ease. It simplifies the process by offering predefined classes and components, reducing the need for extensive custom styling. On the other hand, raw CSS involves writing custom Cascading Style Sheets from scratch, allowing for greater control and flexibility in design but requiring more time and expertise to implement. While Bootstrap offers convenience and consistency, raw CSS offers unlimited customization options tailored to specific design needs.

I.E URI

Full form is 'Unified Resource Identifier'. This is a string of characters used to identify a resource on the internet. It serves as a unique address that allows you to locate and access resources such as web pages, files, images, or services. A URI typically consists of a scheme (such as "http" or "https"), followed by a colon and two slashes, then the specific location or identifier of the resource. URIs can also include additional components like query parameters or fragments to provide more detailed information about the resource or how to interact with it. Overall, URIs play a crucial role in enabling communication and navigation across the vast landscape of the World Wide Web.

I.F HTTP Protocols

I.F.1 GET

This is used to request data from the database.

I.F.2 POST

This is used to send data to the database.'

I.G XML VS XHTML

XML (Extensible Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used for structuring and presenting data. XML is a flexible, general-purpose language that allows users to create their own tags and define their document structure, primarily used for data storage and transport. XHTML, on the other hand, is a stricter and more XML-compliant version of HTML, designed to improve web standards by ensuring that web pages are properly formatted and well-structured. While XML is used for a wide range of applications beyond web pages, XHTML is specifically tailored for creating web content that can be reliably displayed across different browsers and devices.

I.H DHTML

DHTML stands for Dynamic HTML, it is totally different from HTML. The DHTML is based on the properties of the HTML, JavaScript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods.

I.I DOM

a standard for accessing documents like XML and HTML, The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. The DOM (Document Object Mapper) is separated into 3 different parts / levels:

- Core DOM - standard model for any structured document
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents
- The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

Conclusion

The comprehensive home service solution project has successfully achieved its objective of streamlining and optimizing home maintenance and improvement tasks. By catering to the diverse needs of admins, suppliers, service providers, and customers, the platform offers a robust and personalized experience for all users. The integrated web platform, with its key features such as account boosts for service providers, real-time calendar management, flexible pricing, secure payment processing, and appointment scheduling, has proven to be both efficient and user-friendly.

Customers benefit greatly from the platform's ability to handle service requests, manage appointments, place and cancel orders, and provide feedback through ratings and reviews. This has enhanced convenience and improved the overall user experience in managing home services. Service providers and suppliers also gain from the tailored dashboards, enabling them to manage their services more effectively and reach a broader audience.

The project, completed within the stipulated time frame and budget of 5000/- BDT, demonstrates the feasibility and effectiveness of implementing a comprehensive home service solution. It stands as a testament to the potential of technology in improving everyday tasks and fostering better service management. Overall, this platform significantly enhances the efficiency and convenience of home service management, making it a valuable tool for all stakeholders involved.

Assets

[1] Lucidchart. Lucidchart - online diagram and flowchart software. https://lucid.app/. Accessed: 2024-05-29.

[2] MindMeister. Mindmeister - online mind mapping. https://www.mindmeister.com/. Accessed: 2024-05-29.

[3] Dbdiagram.io. Dbdiagram.io - database relationship diagrams design tool. https://dbdiagram.io/home. Accessed: 2024-05-29.

[4] diagrams.net. Diagrams.net (formerly draw.io). https://app.diagrams.net/. Accessed: 2024-05-29.

[5] Figma. Figma - the collaborative interface design tool. https://www.figma.com/. Accessed: 2024-05-29.