



# Car Shop Management System

Project Documentations

May 2024

# Contents

---

	Page
<i>List of Figures</i>	VI
<i>List of Tables</i>	VII
<i>List of Equations</i>	VII
<i>Abstract</i>	VIII
<b>1 Introduction</b>	<b>1</b>
1.1 Project Overview . . . . .	1
1.2 Technologies Used . . . . .	1
1.2.1 programming Language . . . . .	1
1.2.2 Frameworks . . . . .	1
1.2.3 Tools . . . . .	1
1.2.4 Database . . . . .	1
1.2.5 Design Pattern . . . . .	1
<b>2 Overall Description</b>	<b>2</b>
2.1 Product Perspective . . . . .	2
2.1.1 User Classes and Characteristics . . . . .	2
<b>3 Architecture</b>	<b>3</b>
3.1 Design Patterns . . . . .	3
3.1.1 MVC . . . . .	3
3.2 Data Flow . . . . .	4
3.2.1 Level 0 . . . . .	4
3.2.2 Level 1 . . . . .	5
3.3 Work Flow . . . . .	6
<b>4 Code Structure</b>	<b>7</b>
4.1 Directory Structure . . . . .	7
4.1.1 View . . . . .	7
4.1.2 Controller . . . . .	8
4.1.3 Model . . . . .	9
4.2 Modules and Components . . . . .	10
4.2.1 Session . . . . .	10
4.2.2 PHP Validation . . . . .	10
4.2.3 JavaScript Validation . . . . .	11
4.2.4 Bootstrap CSS . . . . .	11

<b>5</b>	<b>Database</b>	<b>12</b>
5.1	Tables . . . . .	12
5.1.1	user . . . . .	12
5.1.2	seller . . . . .	12
5.1.3	message . . . . .	12
5.1.4	car . . . . .	12
5.1.5	car_request . . . . .	12
5.2	Relationships . . . . .	13
5.3	Entity-Relationship (ER) <sup>[1]</sup> Diagram . . . . .	14
<b>6</b>	<b>Features and Functionalities</b>	<b>16</b>
6.1	Feature List . . . . .	16
6.2	Feature 1: Signup . . . . .	16
6.2.1	User Type . . . . .	16
6.2.2	Feature Description . . . . .	16
6.2.3	Functionalities . . . . .	16
6.2.4	Associated Files . . . . .	16
6.3	Feature 2: Login . . . . .	17
6.3.1	User Type . . . . .	17
6.3.2	Feature Description . . . . .	17
6.3.3	Functionalities . . . . .	17
6.3.4	Associated Files . . . . .	17
6.4	Feature 3: Logout . . . . .	17
6.4.1	User Type . . . . .	17
6.4.2	Feature Description . . . . .	17
6.4.3	Functionalities . . . . .	17
6.4.4	Associated Files . . . . .	17
6.5	Feature 4: Show New Cars . . . . .	18
6.5.1	User Type . . . . .	18
6.5.2	Feature Description . . . . .	18
6.5.3	Functionalities . . . . .	18
6.5.4	Associated Files . . . . .	18
6.6	Feature 5: Show All Cars . . . . .	18
6.6.1	User Type . . . . .	18
6.6.2	Feature Description . . . . .	18
6.6.3	Functionalities . . . . .	18
6.6.4	Associated Files . . . . .	18
6.7	Feature 6: Show Sold Cars . . . . .	19
6.7.1	User Type . . . . .	19
6.7.2	Feature Description . . . . .	19
6.7.3	Functionalities . . . . .	19
6.7.4	Associated Files . . . . .	19
6.8	Feature 7: Show Rented Cars . . . . .	19
6.8.1	User Type . . . . .	19
6.8.2	Feature Description . . . . .	19
6.8.3	Functionalities . . . . .	19
6.8.4	Associated Files . . . . .	19
6.9	Feature 8: Show Used Cars . . . . .	20
6.9.1	User Type . . . . .	20
6.9.2	Feature Description . . . . .	20
6.9.3	Functionalities . . . . .	20
6.9.4	Associated Files . . . . .	20

6.10	Feature 9: Show Car Requests	20
6.10.1	User Type	20
6.10.2	Feature Description	20
6.10.3	Functionalities	20
6.10.4	Associated Files	20
6.11	Feature 10: Show Comments	21
6.11.1	User Type	21
6.11.2	Feature Description	21
6.11.3	Functionalities	21
6.11.4	Frontend	21
6.12	Feature 11: Show Single Car Details	21
6.12.1	User Type	21
6.12.2	Feature Description	21
6.12.3	Functionalities	21
6.12.4	Frontend	21
6.13	Feature 12: Show Profit	22
6.13.1	User Type	22
6.13.2	Feature Description	22
6.13.3	Functionalities	22
6.13.4	Frontend	22
6.14	Feature 13: Revenue	22
6.14.1	User Type	22
6.14.2	Feature Description	22
6.14.3	Functionalities	22
6.14.4	Associated Files	22
6.15	Feature 14: Show Total Sold Car Amount	23
6.15.1	User Type	23
6.15.2	Feature Description	23
6.15.3	Functionalities	23
6.15.4	Associated Files	23
6.16	Feature 15: Show Total Available Car Number	23
6.16.1	User Type	23
6.16.2	Feature Description	23
6.16.3	Functionalities	23
6.16.4	Associated Files	23
6.17	Feature 16: Add New Car	24
6.17.1	User Type	24
6.17.2	Feature Description	24
6.17.3	Functionalities	24
6.17.4	Associated Files	24
6.18	Feature 17: Delete a Car	24
6.18.1	User Type	24
6.18.2	Feature Description	24
6.18.3	Functionalities	24
6.18.4	Associated Files	24
6.19	Feature 18: Edit Car Details	25
6.19.1	User Type	25
6.19.2	Feature Description	25
6.19.3	Functionalities	25
6.19.4	Associated Files	25
6.20	Feature 19: Show Car's Availability and Status	25
6.20.1	User Type	25

---

6.20.2	Feature Description	25
6.20.3	Functionalities	25
6.20.4	Associated Files	25
6.21	Feature 20: Make Brand New Car to Used	26
6.21.1	User Type	26
6.21.2	Feature Description	26
6.21.3	Functionalities	26
6.21.4	Associated Files	26
6.22	Feature 21: Make Rented Car to Available	26
6.22.1	User Type	26
6.22.2	Feature Description	26
6.22.3	Functionalities	26
6.22.4	Associated Files	26
6.23	Feature 22: Accept Customer Request	27
6.23.1	User Type	27
6.23.2	Feature Description	27
6.23.3	Functionalities	27
6.23.4	Associated Files	27

<b>7</b>	<b>Equations</b>	<b>28</b>
7.1	Total Revenue . . . . .	28
7.2	Total Profit . . . . .	28
7.3	Quantity of Sold Cars . . . . .	28
7.4	Quantity of Available Cars . . . . .	28
7.5	Current Month Income . . . . .	28
<b>8</b>	<b>Error Handling</b>	<b>29</b>
8.1	PHP Error Handling . . . . .	29
<b>9</b>	<b>Testing</b>	<b>30</b>
9.1	Test Case: Seller Signup . . . . .	30
9.2	Test Case: Login . . . . .	31
9.3	Test Case: Logout . . . . .	32
9.4	Test Case: Show New Cars . . . . .	33
9.5	Test Case: Show All Cars . . . . .	34
9.6	Test Case: Show Sold Cars . . . . .	35
9.7	Test Case: Show Rented Cars . . . . .	36
9.8	Test Case: Show Used Cars . . . . .	37
9.9	Test Case: Show Car Requests . . . . .	38
9.10	Test Case: Show Comments . . . . .	39
9.11	Test Case: Show Single Car Details . . . . .	40
9.12	Test Case: Show Yearly Sales, profits, revenue, total sold cars, total available cars . . . . .	41
9.13	Test Case: Add New Car . . . . .	42
9.14	Test Case: Delete a Car . . . . .	43
9.15	Test Case: Edit Car Details . . . . .	44
9.16	Test Case: Show car's availability and status . . . . .	45
9.17	Test Case: Make Brand New Car to Used . . . . .	46
9.18	Test Case: Make Rented Car to Available . . . . .	47
9.19	Test Case: Accept Customer Request . . . . .	48
9.20	Test Case: Reject Customer Request . . . . .	49
<b>10</b>	<b>Code Examples and Snippets</b>	<b>50</b>
10.1	Routes . . . . .	50
	<i>Definitions</i>	<i>51</i>
I.A	<i>Session</i> . . . . .	<i>51</i>
I.B	<i>Cookies</i> . . . . .	<i>51</i>
I.C	<i>Validation</i> . . . . .	<i>51</i>
I.D	<i>Difference between Bootstrap and Raw CSS</i> . . . . .	<i>51</i>
I.E	<i>URI</i> . . . . .	<i>51</i>
I.F	<i>HTTP Protocols</i> . . . . .	<i>52</i>
I.F.1	<i>GET</i> . . . . .	<i>52</i>
I.F.2	<i>POST</i> . . . . .	<i>52</i>
I.G	<i>XML VS XHTML</i> . . . . .	<i>52</i>
I.H	<i>DHTML</i> . . . . .	<i>52</i>
I.I	<i>DOM</i> . . . . .	<i>52</i>
	<i>Epilogue</i>	<i>53</i>
	<i>Assets</i>	<i>54</i>

# List of Figures

---

	Page
2.1 Type of Users . . . . .	2
3.1 File Distributions in MVC <sup>[2]</sup> Structure . . . . .	3
3.2 Data Flow Diagram Level - 0 . . . . .	4
3.3 Data Flow Diagram Level - 1 . . . . .	5
3.4 Work Flow Diagram . . . . .	6
4.1 View Folder Code Structure . . . . .	7
4.2 Controller Folder Code Structure . . . . .	8
4.3 Model Folder Code Structure . . . . .	9
5.1 Relationship Simplification between tables . . . . .	13
5.2 ER <sup>[1]</sup> Diagram . . . . .	14
5.3 ER Diagram [Detailed <sup>[3]</sup> Version] . . . . .	15

# List of Equations

---

	Page
<i>Equation 7.1</i> . . . . .	28
<i>Equation 7.2</i> . . . . .	28
<i>Equation 7.3</i> . . . . .	28
<i>Equation 7.4</i> . . . . .	28
<i>Equation 7.5</i> . . . . .	28



# Abstract

---

This document presents the development and successful completion of a comprehensive car dealership management system. Designed specifically for sellers, this system enables efficient management of inventory through an integrated web platform. After logging in via the login page, which also includes a registration option, sellers are directed to their personalized homepage. Here, they can manage new, used, and rental cars, adding or removing vehicles as needed. Sellers can also update existing listings by editing car prices and quantities or halting the sale of any vehicle. Additionally, the system offers robust reporting tools, allowing sellers to generate detailed PDF reports on monthly or yearly sales, revenue, and profit for new, used, and rental car transactions. This feature enhances the ability to track performance and make informed business decisions. The project was completed on time, within a budget of 3000/- BDT, and over a period of 4 days. The customer has expressed satisfaction with the final product, noting its ease of use and comprehensive functionality. This car dealership management system promises to streamline the selling process, providing a seamless and user-friendly interface for car dealership management.

# Introduction

---

## 1.1 Project Overview

This project is a comprehensive car dealership management system designed for sellers, allowing them to efficiently manage and operate their inventory through an integrated web platform. Upon logging in from the login page, which includes a registration option, sellers are directed to their personalized homepage. Here, they can showcase new, used, and rental cars, with functionalities to add or remove vehicles from each category. Additionally, sellers have the capability to manage existing listings by editing car prices and quantities, as well as halting the sale of any vehicle currently available on the site. The system also provides powerful reporting tools, enabling sellers to generate detailed PDF reports on monthly or yearly sales, revenue, and profit across new, used, and rental car transactions. This robust solution aims to streamline the selling process, offering a seamless and user-friendly interface for car dealership management.

## 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.

# Overall Description

---

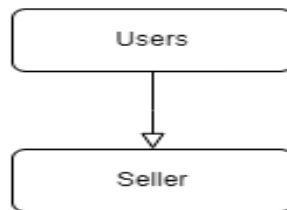
## 2.1 Product Perspective

“Car Shop Management System” 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

“Car Shop Management System” has currently 1 Type of User<sup>[4]</sup>.

- Seller



**Figure 2.1:** Type of Users

### 3.1 Design Patterns

#### 3.1.1 MVC

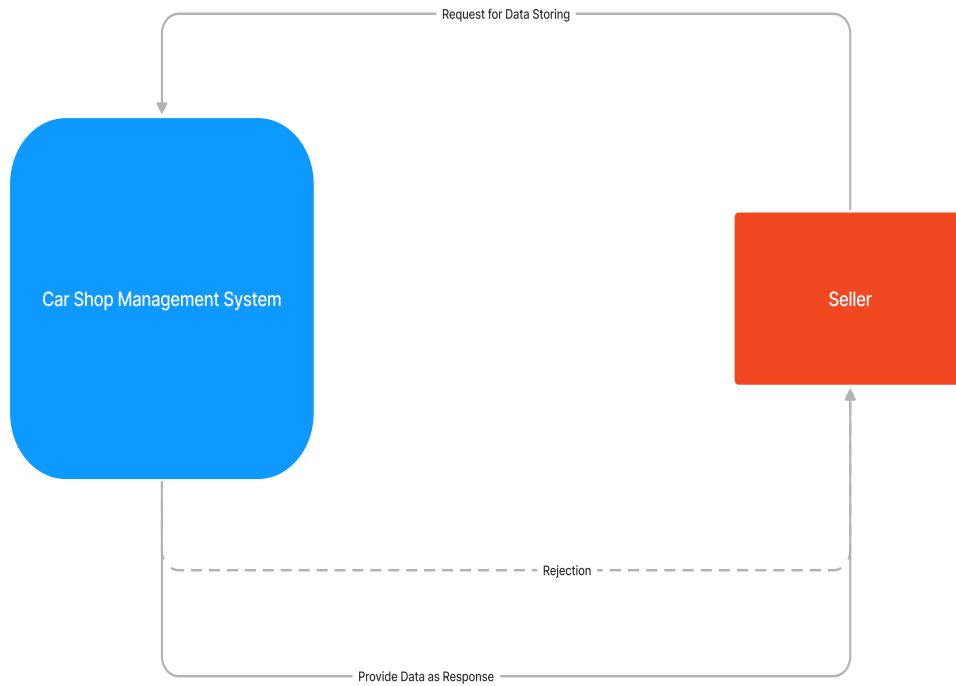


**Figure 3.1:** File Distributions in MVC<sup>[2]</sup> Structure

### 3.2 Data Flow

#### 3.2.1 Level 0

Data flow of level-0<sup>[5]</sup> 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<sup>[5]</sup> 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.

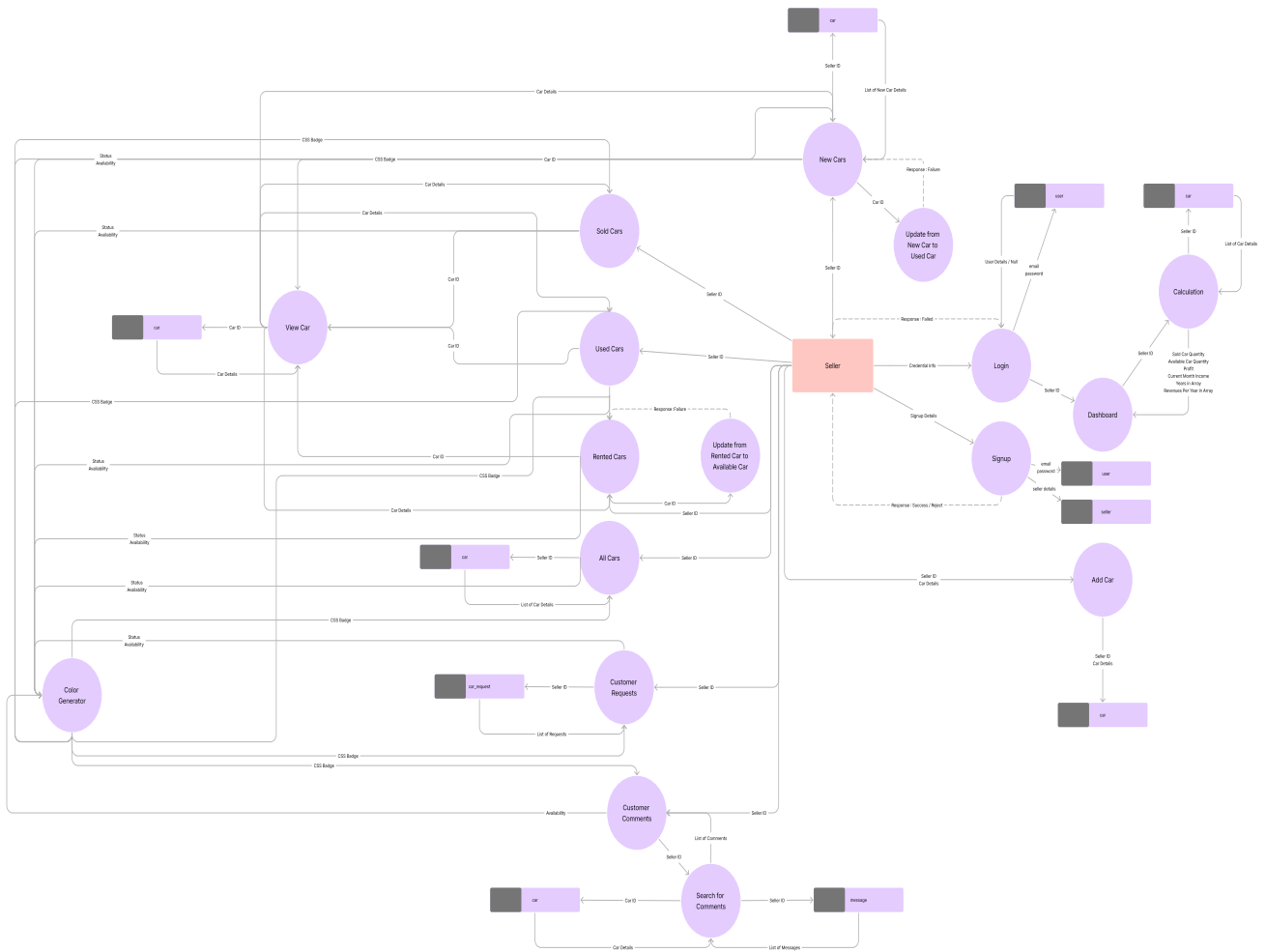
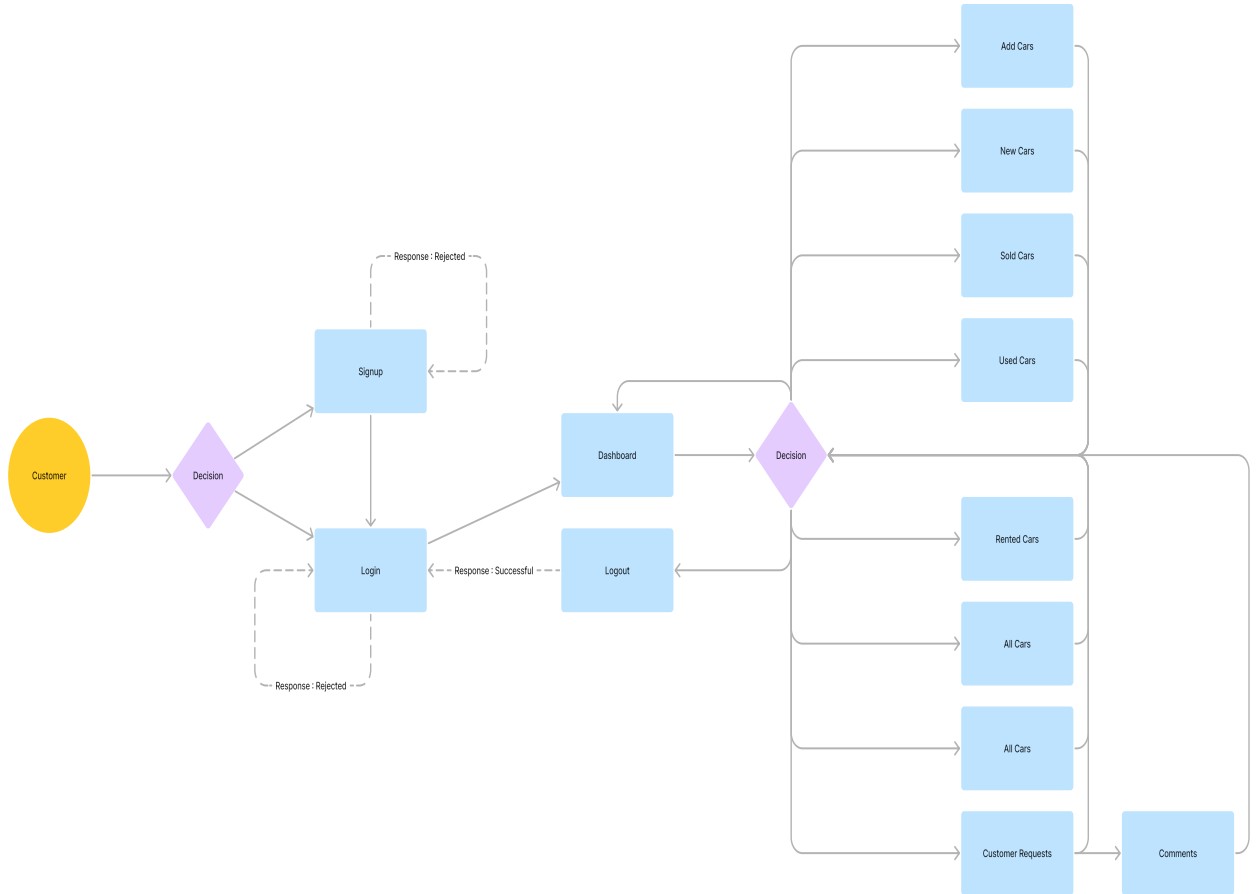


Figure 3.3: Data Flow Diagram Level - 1

### 3.3 Work Flow

Work flow diagram<sup>[5]</sup> 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 3.4:** Work Flow Diagram

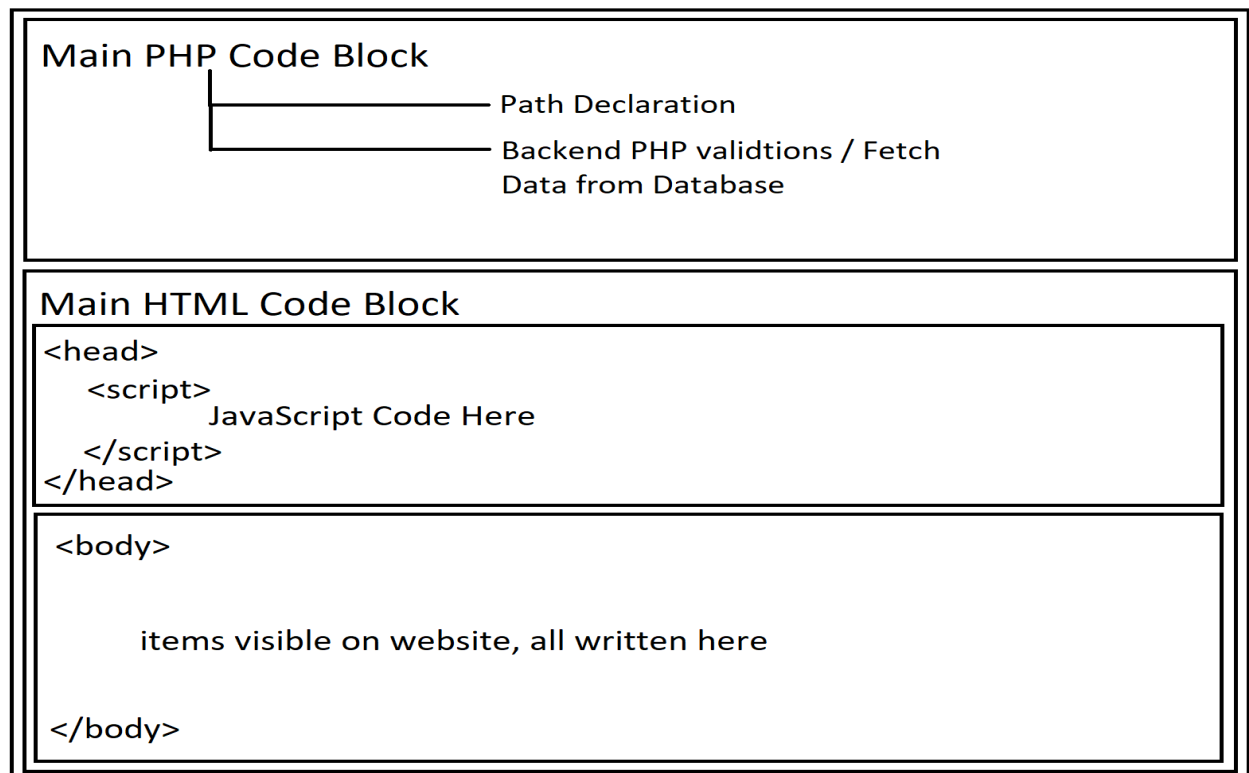
# Code Structure

---

## 4.1 Directory Structure

### 4.1.1 View

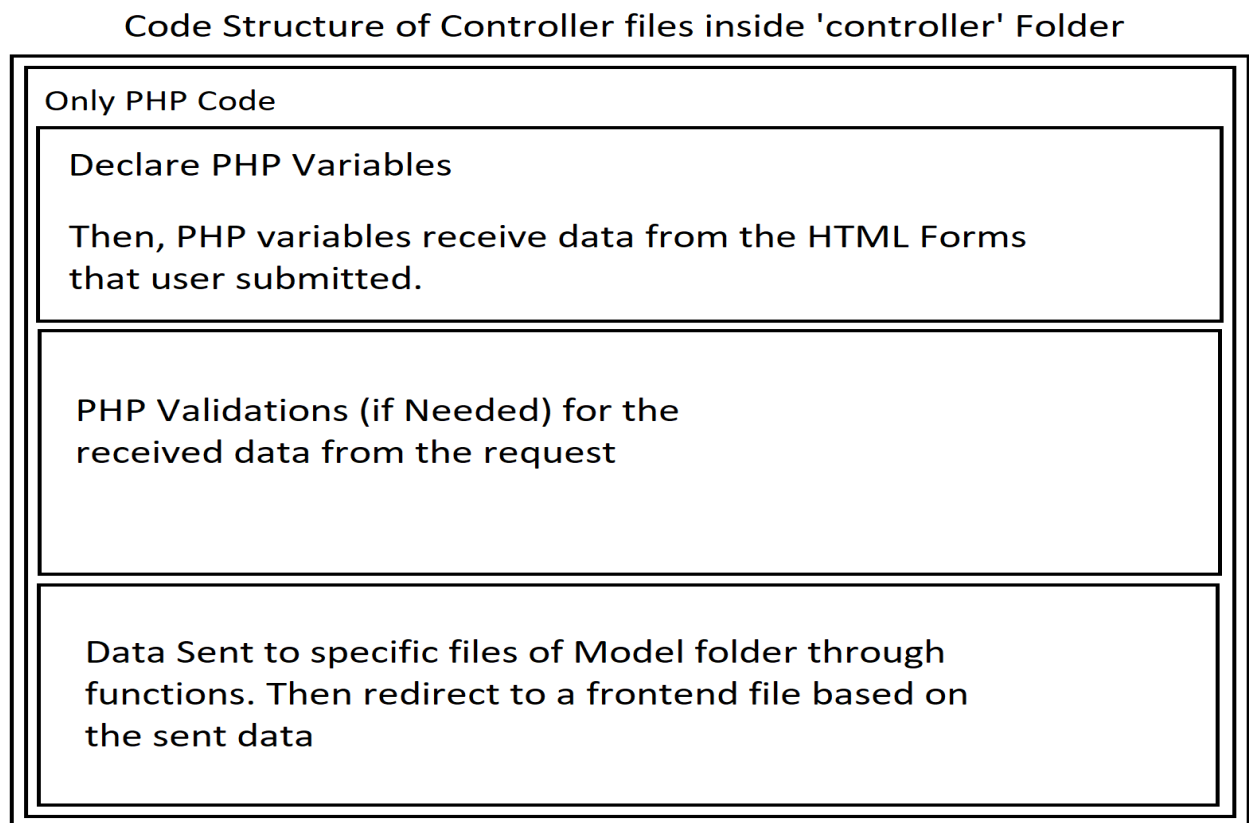
#### Code Structure of Files Inside 'view' Folder



**Figure 4.1:** View Folder Code Structure



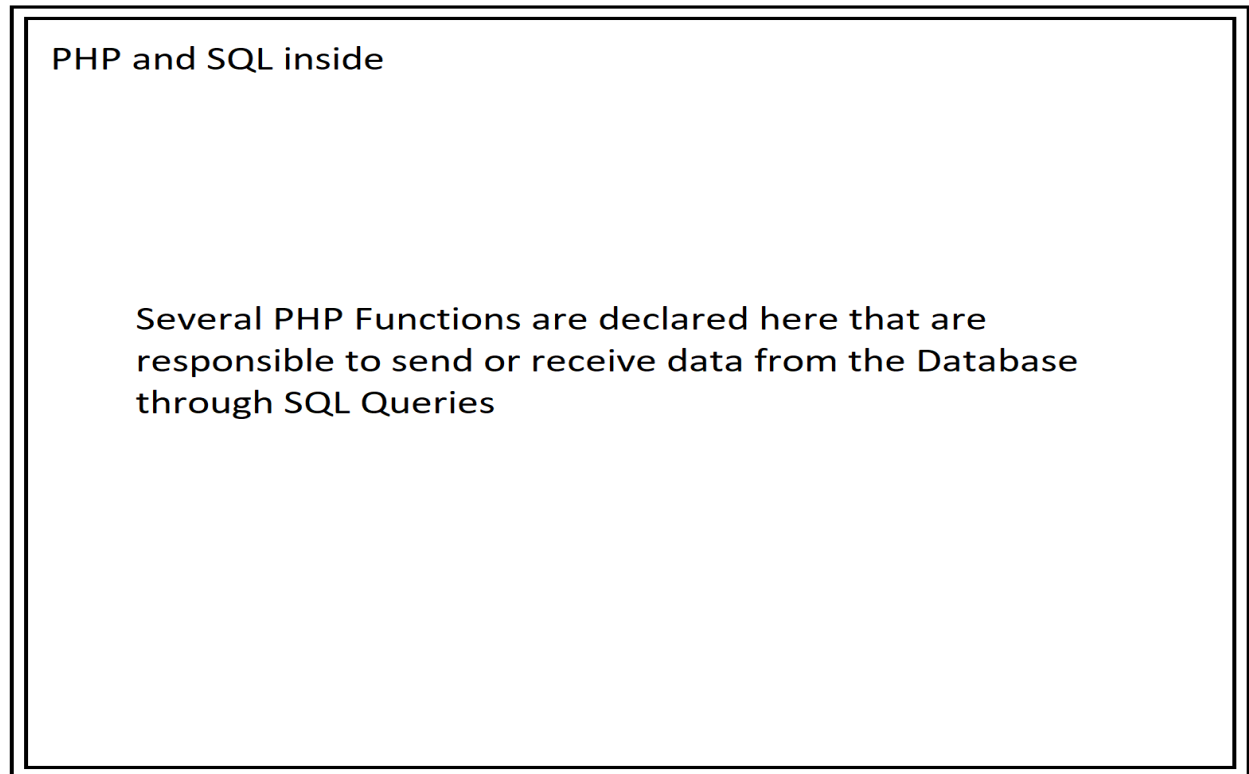
#### 4.1.2 Controller



**Figure 4.2:** Controller Folder Code Structure

### 4.1.3 Model

#### Code Structure of Controller files inside 'controller' Folder



**Figure 4.3:** Model Folder Code Structure

## 4.2 Modules and Components

- Session
- PHP Validation
- JavaScript Validation
- Bootstrap CSS

### 4.2.1 Session

```
1 <?php
2 session_start();
3 $_SESSION['variable_name'];
4 ?>
```

**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

```
1 <?php
2 /* Email Validation
3 $email = $_POST['email'];
4 if (empty($email)) {
5     $everythingOK = FALSE;
6     $everythingOKCounter += 1;
7     echo '<br>Mail Error 1<br>';
8 } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
9     $everythingOK = FALSE;
10    $everythingOKCounter += 1;
11    echo '<br>Mail Error 2<br>';
12 } else {
13    $everythingOK = TRUE;
14 }
15 ?>
```

**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

```
1 function validateEmail(email) {
2     var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
3     return emailRegex.test(email);
4 }
5
6 // Function to show modal with validation message
7 function showModal(message) {
8     document.getElementById("validationMessage").innerHTML = message;
9     document.getElementById("validationModal").style.display = "block";
10 }
11 close_modal = () => {
12     document.getElementById("validationModal").style.display = "none";
13     // location.reload(true);
14 }
15 function validateForm() {
16     var email = document.getElementById("login_email").value;
17     var password = document.getElementById("login_password").value;
18     document.getElementById("loginEmailError");
19     if (email === "" || email === null) {
20         // emailErrorLabel.innerHTML = "Email is required.";
21         showModal("Email is Required");
22         return false;
23     }
24     if (password === "" || password === null) {
25         // Display error message
26         showModal("Password is Required");
27         return false;
28     }
29     return true; // Return true if all validations pass
30 }
```

**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.

## 5.1 Tables

### 5.1.1 user

- id (PK)
- email
- password
- type

### 5.1.2 seller

- id (PK)
- name
- phone
- user\_id (FK)

### 5.1.3 message

- id (PK)
- message
- date
- seller\_id (FK)
- customer\_id (FK)
- car\_id (FK)

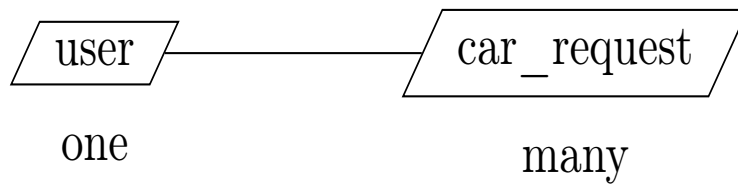
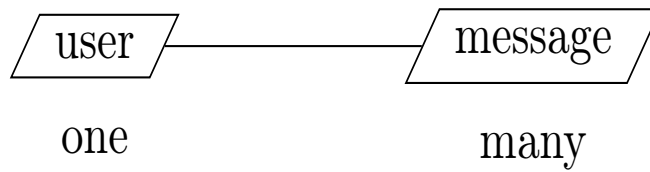
### 5.1.4 car

- id (PK)
- model
- brand
- status
- original\_price
- user\_id (FK)
- availability
- date

### 5.1.5 car\_request

- id (PK)
- car\_id (FK)
- seller\_id (FK)
- customer\_id (FK)
- date
- status
- request\_type

## 5.2 Relationships



**Figure 5.1:** Relationship Simplification between tables

### 5.3 Entity-Relationship (ER)<sup>[1]</sup> Diagram

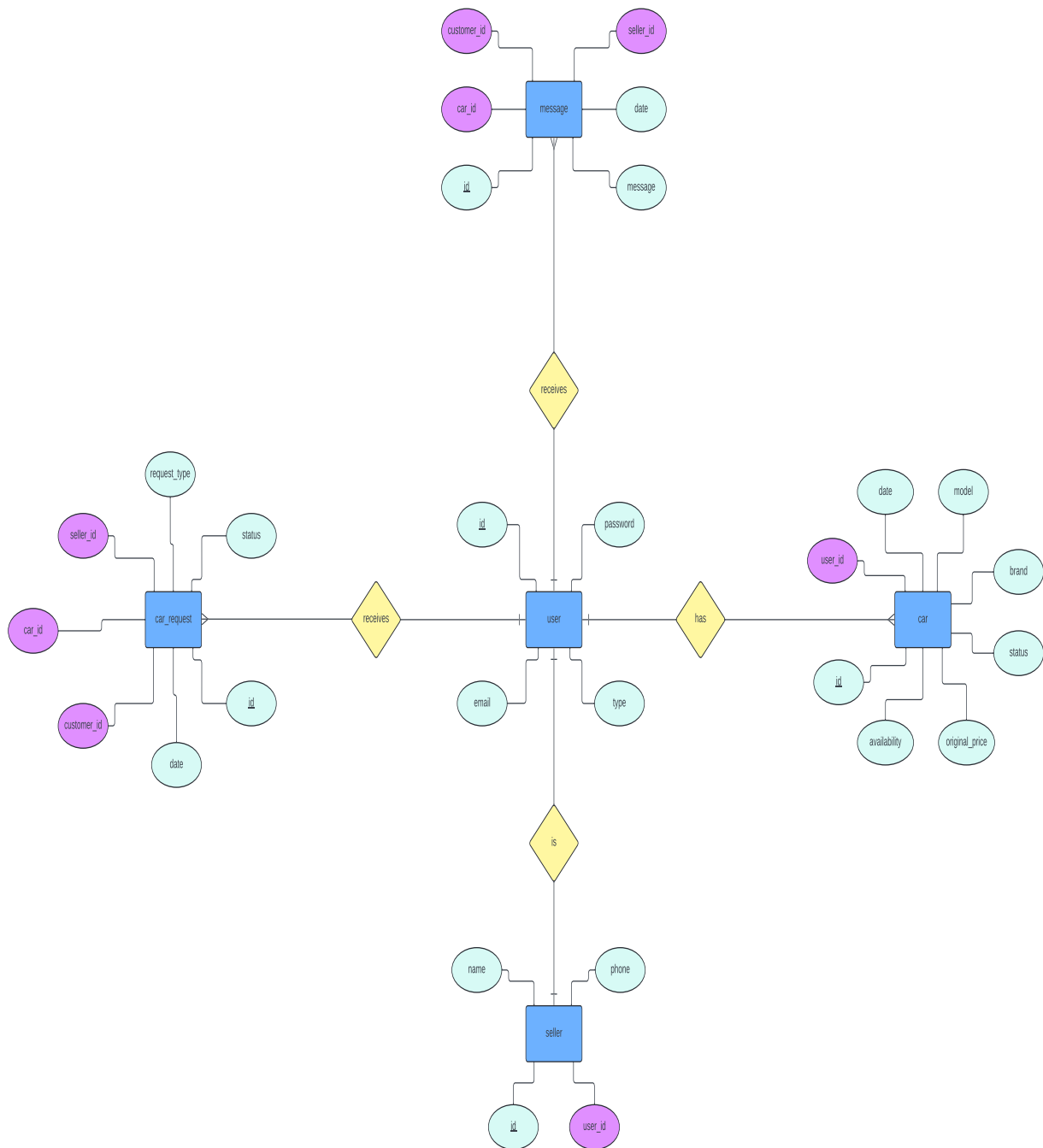
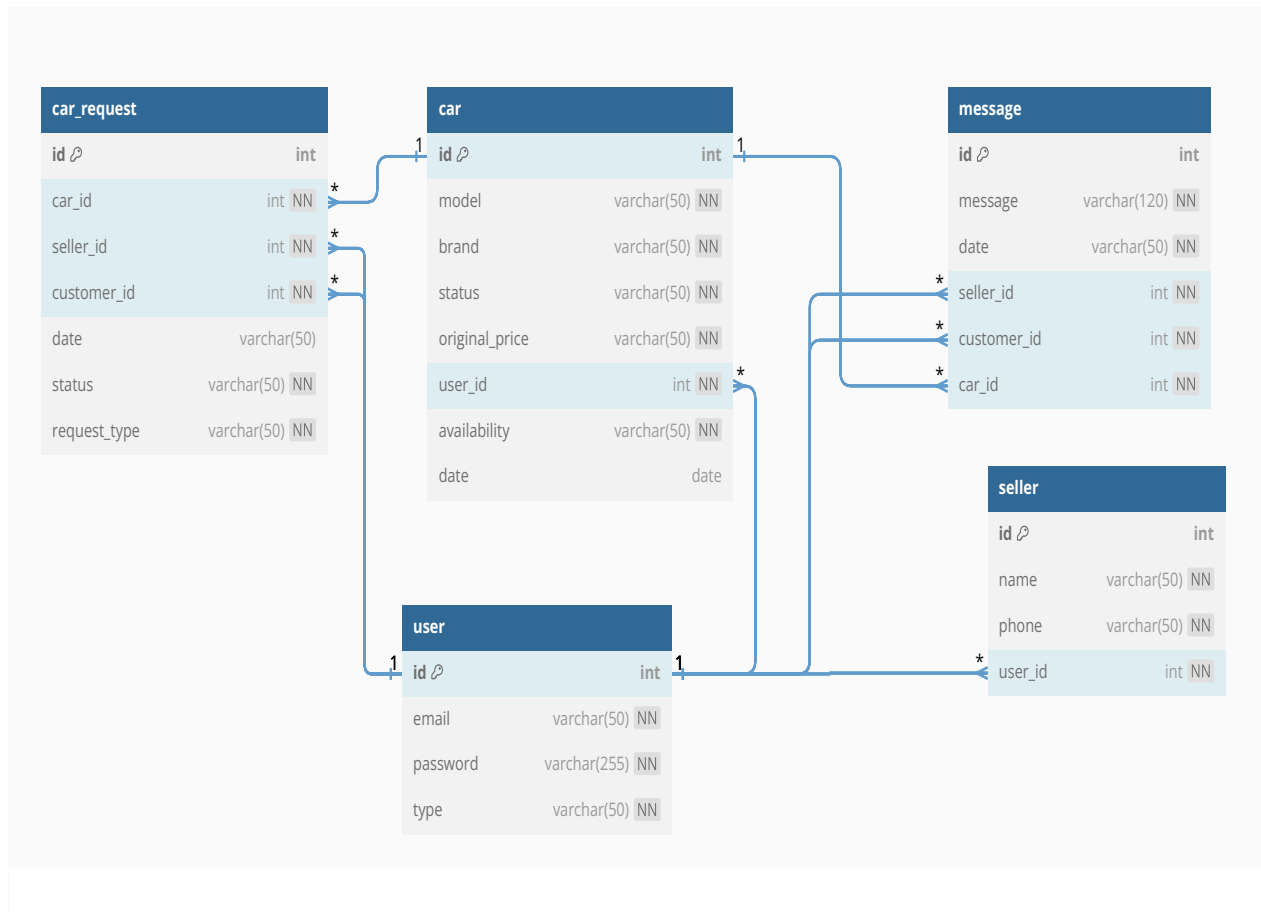


Figure 5.2: ER<sup>[1]</sup> Diagram



**Figure 5.3:** ER Diagram [Detailed <sup>[3]</sup> Version]



# Features and Functionalities

---

## 6.1 Feature List

- Seller Signup
- Seller Login
- Seller Logout
- Show New Cars
- Show All Cars
- Show Sold Cars
- Show Rented Cars
- Show Used Cars
- Show Car Requests
- Show Comments
- Show Single Car Details
- Show Yearly Sales
- Show Profits
- Show Revenue
- Show Total Sold Car Amount
- Show Total Available Cars Number
- Add New Car
- Delete a Car
- Edit Car Details
- Show Car's Availability and Status
- Make Brand New Car to Used
- Make Rented Car to Available
- Accept Customer Request
- Reject Customer Request

## 6.2 Feature 1: Signup

### 6.2.1 User Type

Seller

### 6.2.2 Feature Description

Allows a new seller 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) : Seller\_Signup.php
- Associated Controller : Seller\_SignupController.php
- Associated Model : UserRepo.php

### 6.3 Feature 2: Login

#### 6.3.1 User Type

Seller

#### 6.3.2 Feature Description

Enables a seller 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 : UserRepo.php

### 6.4 Feature 3: Logout

#### 6.4.1 User Type

Seller

#### 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) :  
Add\_Car.php, All\_Cars.php, All\_Comments.php, Customer\_Requests.php, Edit\_Car.php,  
New\_Car.php, Rented\_Car.php, Seller\_Dashboard.php, Sold\_Car.php, Used\_Car.php,  
View\_Car.php
- Associated Controller : LogoutController.php
- Associated Model : NONE

## **6.5 Feature 4: Show New Cars**

### **6.5.1 User Type**

Seller

### **6.5.2 Feature Description**

Displays a list of newly added cars available for sale or rent.

### **6.5.3 Functionalities**

#### **CRUD Functionalities**

- Read

#### **Other Functionalities**

- Session

### **6.5.4 Associated Files**

- Associated View (Front-end) : New\_Car.php
- Associated Controller : New\_Car.php
- Associated Model : CarRepo.php

## **6.6 Feature 5: Show All Cars**

### **6.6.1 User Type**

Seller

### **6.6.2 Feature Description**

Displays a comprehensive list of all cars available in the system for viewing.

### **6.6.3 Functionalities**

#### **CRUD Functionalities**

- Read

#### **Other Functionalities**

- Session

### **6.6.4 Associated Files**

- Associated View (Front-end) : All\_Cars.php
- Associated Controller : All\_Cars.php
- Associated Model : CarRepo.php

**6.7 Feature 6: Show Sold Cars****6.7.1 User Type**

Seller

**6.7.2 Feature Description**

Provides a list of cars that have been successfully sold by the seller.

**6.7.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.7.4 Associated Files**

- Associated View (Front-end) : Sold\_Car.php
- Associated Controller : Sold\_Car.php
- Associated Model : CarRepo.php

**6.8 Feature 7: Show Rented Cars****6.8.1 User Type**

Seller

**6.8.2 Feature Description**

Displays a list of cars that are currently being rented out by the seller.

**6.8.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.8.4 Associated Files**

- Associated View (Front-end) : Rented\_Car.php
- Associated Controller : Rented\_Car.php
- Associated Model : CarRepo.php

## **6.9 Feature 8: Show Used Cars**

### **6.9.1 User Type**

Seller

### **6.9.2 Feature Description**

Presents a collection of pre-owned cars available for sale by the seller.

### **6.9.3 Functionalities**

#### **CRUD Functionalities**

- Read

#### **Other Functionalities**

- Session

### **6.9.4 Associated Files**

- Associated View (Front-end) : Used\_Car.php
- Associated Controller : Used\_Car.php
- Associated Model : CarRepo.php

## **6.10 Feature 9: Show Car Requests**

### **6.10.1 User Type**

Seller

### **6.10.2 Feature Description**

Displays a list of requests from potential buyers or renters interested in purchasing or renting cars from the seller.

### **6.10.3 Functionalities**

#### **CRUD Functionalities**

- Read

#### **Other Functionalities**

- Session

### **6.10.4 Associated Files**

- Associated View (Front-end) : Customer\_Requests.php
- Associated Controller : Customer\_Requests.php
- Associated Model : CarRepo.php, Car\_RequestRepo.php

**6.11 Feature 10: Show Comments****6.11.1 User Type**

Seller

**6.11.2 Feature Description**

Displays a list of requests from potential buyers or renters interested in purchasing or renting cars from the seller.

**6.11.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.11.4 Frontend**

- Associated View (Front-end) : All\_Comments.php
- Associated Controller : All\_Comments.php
- Associated Model : CarRepo.php, MessageRepo.php

**6.12 Feature 11: Show Single Car Details****6.12.1 User Type**

Seller

**6.12.2 Feature Description**

Displays detailed information about a specific car, including its status and availability.

**6.12.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.12.4 Frontend**

- Associated View (Front-end) : View\_Car.php
- Associated Controller : View\_Car.php
- Associated Model : carRepo.php

**6.13 Feature 12: Show Profit****6.13.1 User Type**

Seller

**6.13.2 Feature Description**

Displays the seller's total profit earned from car sales and rentals.

**6.13.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.13.4 Frontend**

- Associated View (Front-end) : Seller\_Dashboard.php
- Associated Controller : Seller\_Dashboard.php
- Associated Model : CalculationRepo.php

**6.14 Feature 13: Revenue****6.14.1 User Type**

Seller

**6.14.2 Feature Description**

Shows the total revenue generated by the seller from all car sales and rentals.

**6.14.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.14.4 Associated Files**

- Associated View (Front-end) : Seller\_Dashboard.php
- Associated Controller : Seller\_Dashboard.php
- Associated Model : CalculationRepo.php

**6.15 Feature 14: Show Total Sold Car Amount****6.15.1 User Type**

Seller

**6.15.2 Feature Description**

Displays the total monetary value of all cars sold by the seller.

**6.15.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.15.4 Associated Files**

- Associated View (Front-end) : Seller\_Dashboard.php
- Associated Controller : Seller\_Dashboard.php
- Associated Model : CalculationRepo.php

**6.16 Feature 15: Show Total Available Car Number****6.16.1 User Type**

Seller

**6.16.2 Feature Description**

Displays the total number of cars currently available for sale or rent by the seller.

**6.16.3 Functionalities****CRUD Functionalities**

- Read

**Other Functionalities**

- Session

**6.16.4 Associated Files**

- Associated View (Front-end) : Seller\_Dashboard.php
- Associated Controller : Seller\_Dashboard.php
- Associated Model : CalculationRepo.php



**6.17 Feature 16: Add New Car****6.17.1 User Type**

Seller

**6.17.2 Feature Description**

Allows the seller to add a new car to the inventory by entering its details.

**6.17.3 Functionalities****CRUD Functionalities**

- Create / Insert

**Other Functionalities**

- JavaScript Validation
- PHP Validation
- Session

**6.17.4 Associated Files**

- Associated View (Front-end) : Add\_Car.php
- Associated Controller : Add\_CarController.php
- Associated Model : CarRepo.php

**6.18 Feature 17: Delete a Car****6.18.1 User Type**

Seller

**6.18.2 Feature Description**

Enables the seller to remove a car from the inventory.

**6.18.3 Functionalities****CRUD Functionalities**

- Delete

**Other Functionalities**

- Session

**6.18.4 Associated Files**

- Associated View (Front-end) : All\_Cars.php
- Associated Controller : All\_Cars\_DeleteController.php
- Associated Model : CarRepo.php

**6.19 Feature 18: Edit Car Details****6.19.1 User Type**

Seller

**6.19.2 Feature Description**

Displays a list of requests from potential buyers or renters interested in purchasing or renting cars from the seller.

**6.19.3 Functionalities**

CRUD Functionalities

- Update

**Other Functionalities**

- JavaScript Validation
- PHP Validation
- Session

**6.19.4 Associated Files**

- Associated View (Front-end) : All\_Cars.php
- Associated Controller : All\_Cars\_EditController.php
- Associated Model : CarRepo.php

**6.20 Feature 19: Show Car's Availability and Status****6.20.1 User Type**

Seller

**6.20.2 Feature Description**

Displays the current availability and status of each car in the seller's inventory.

**6.20.3 Functionalities**

CRUD Functionalities

- Read

**Other Functionalities**

- Session

**6.20.4 Associated Files**

- Associated View (Front-end) : All\_Cars.php, All\_Comments.php, Customer\_Requests.php, Edit\_Car.php, New\_Car.php, Rented\_Car.php, Sold\_Car.php, Used\_Car.php, View\_Car.php
- Associated Controller : All\_Cars.php, All\_Comments.php, Customer\_Requests.php, Edit\_Car.php, New\_Car.php, Rented\_Car.php, Sold\_Car.php, Used\_Car.php, View\_Car.php
- Associated Model : ColorGenerator.php

**6.21 Feature 20: Make Brand New Car to Used****6.21.1 User Type**

Seller

**6.21.2 Feature Description**

Allows the seller to change the status of a brand new car to a used car.

**6.21.3 Functionalities****CRUD Functionalities**

- Update

**Other Functionalities**

- Session

**6.21.4 Associated Files**

- Associated View (Front-end) : New\_Car.php
- Associated Controller : New\_Car\_UsedController.php
- Associated Model : CarRepo.php

**6.22 Feature 21: Make Rented Car to Available****6.22.1 User Type**

Seller

**6.22.2 Feature Description**

Allows the seller to update a rented car's status to available for sale or rent again.

**6.22.3 Functionalities****CRUD Functionalities**

- Update

**Other Functionalities**

- Session

**6.22.4 Associated Files**

- Associated View (Front-end) : Rented\_Car.php
- Associated Controller : Rented\_Car\_AvailableController.php
- Associated Model : CarRepo.php

**6.23 Feature 22: Accept Customer Request****6.23.1 User Type**

Seller

**6.23.2 Feature Description****6.23.3 Functionalities****CRUD Functionalities**

- Update

**Other Functionalities**

- Session

**6.23.4 Associated Files**

- Associated View (Front-end) : Customer\_Requests.php
- Associated Controller : Customer\_Requests\_AcceptController.php
- Associated Model : Car\_RequestRepo.php

# Equations

---

## 7.1 Total Revenue

Total revenue is calculated as the sum of the prices of sold cars and the revenue generated from rented cars.

$$\text{Revenue} = \sum_{\text{cars}} (\text{price} \times \text{availability\_factor}) \quad (7.1)$$

## 7.2 Total Profit

Total profit is calculated as the difference between the price of sold cars and the cost price (80% of the original price) for unsold rented cars.

$$\text{Profit} = \sum_{\text{cars}} (\text{price} - \text{cost\_price}) \quad (7.2)$$

## 7.3 Quantity of Sold Cars

The quantity of sold cars is the count of cars with availability status 'Sold'.

$$\text{Sold\_Quantity} = \sum_{\text{cars}} (\text{availability} == \text{'Sold'}) \quad (7.3)$$

## 7.4 Quantity of Available Cars

The quantity of available cars is the count of cars with availability status 'Available'.

$$\text{Available\_Quantity} = \sum_{\text{cars}} (\text{availability} == \text{'Available'}) \quad (7.4)$$

## 7.5 Current Month Income

Current month income is the sum of prices of cars sold or rented in the current month.

$$\text{Current\_Month\_Income} = \sum_{\text{cars}} (\text{price} \times \text{availability\_factor}) \quad (7.5)$$

# Error Handling

---

## 8.1 PHP Error Handling

```
1 <?php
2 header("Location: {$another_page}")
3 ?>
```

**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: Seller Signup

#### Description

Create an Account in the website

#### Test Steps

1. Click the Signup link from Login.php
2. Choose User Type
3. Fill-up all the information
4. Check the Terms and Conditions check-box
5. Click Register Button

#### Test Data

- Name : Test
- Email : test3@gmail.com
- Password : testPass1@
- Phone : 019231414986

#### Expected Output

Login Page can be seen

#### Actual Output

Login 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: Login****Description**

Login to the System

**Test Steps**

1. Fill-up email and password
2. Click Login Button

**Test Data**

- Email : test3@gmail.com
- Password : testPass1@

**Expected Output**

Dashboard Page can be seen

**Actual Output**

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: Logout****Description**

Logout from the account

**Test Steps**

1. Click the user icon on the top right corner
2. Click 'Logout' from the dropdown

**Test Data****Expected Output**

Login Page can be seen

**Actual Output**

Login 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: Show New Cars****Description**

Show all the cars which have the status named 'Brand New'

**Test Steps**

1. Click on 'New Cars' button from the left side drawer

**Expected Output**

New Car List Page can be seen

**Actual Output**

New Car List Page is visible

**Result**PASS**Notes**

New car list 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.

**9.5 Test Case: Show All Cars****Description**

Show all the cars List

**Test Steps**

1. Click on 'All Cars' button from the left side drawer

**Expected Output**

All Car List Page can be seen

**Actual Output**

All Car List Page is visible

**Result**PASS**Notes**

All car list 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.

**9.6 Test Case: Show Sold Cars****Description**

Show all the cars which have the availability named 'Sold'

**Test Steps**

1. Click on 'Sold Cars' button from the left side drawer

**Expected Output**

Sold Car List Page can be seen

**Actual Output**

Sold Car List Page is visible

**Result**PASS**Notes**

Sold car list 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.

**9.7 Test Case: Show Rented Cars****Description**

Show all the cars which have the availability named 'Rented'

**Test Steps**

1. Click on 'Rented Cars' button from the left side drawer

**Expected Output**

Rented Car List Page can be seen

**Actual Output**

Rented Car List Page is visible

**Result**PASS**Notes**

Rented car list 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.

**9.8 Test Case: Show Used Cars****Description**

Show all the cars which have the status named 'Used'

**Test Steps**

1. Click on 'Used Cars' button from the left side drawer

**Expected Output**

Used Car List Page can be seen

**Actual Output**

Used Car List Page is visible

**Result**PASS**Notes**

Used car list 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.

**9.9 Test Case: Show Car Requests****Description**

Show all the customer requests that has been sent to him.

**Test Steps**

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

**Expected Output**

Customer Requests List Page can be seen

**Actual Output**

Customer Requests List Page is visible

**Result**PASS**Notes**

Customer Requests list 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.

**9.10 Test Case: Show Comments****Description**

Show all the comments from the customer that has been sent to him for the sold cars.

**Test Steps**

1. Click on 'Customer Requests' button from the left side drawer
2. Click on 'Comments' tab from the breadcrumbs that is visible at the top-center of the screen

**Expected Output**

Comments List Page can be seen

**Actual Output**

Comments List Page is visible

**Result**PASS**Notes**

Comments list 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.



**9.11 Test Case: Show Single Car Details****Description**

Show a single car information's by selecting any car

**Test Steps**

1. Click on 'Sold Cars' button or 'Used Cars' or 'Rented Cars' from the left side drawer
2. Click on 'View' button

**Expected Output**

A detailed information of that car can be seen with a brand logo

**Actual Output**

A detailed information of that car with a brand logo is visible

**Result**

PASS

**Notes**

Car details 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 or somehow manages to view the page without selecting any car to show the details.

**9.12 Test Case: Show Yearly Sales, profits, revenue, total sold cars, total available cars****Description**

Show the sales per year from the beginning of the account till now through a graph

**Test Steps**

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

**Expected Output**

A graph with yearly sales, a card with total-revenue, total-profit, car-sold, car-available, income can be seen

**Actual Output**

A graph with yearly sales, a card with total-revenue, total-profit, car-sold, car-available, income 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

**9.13 Test Case: Add New Car****Description**

Add a new Car for the seller's car shop that is logged in currently

**Test Steps**

1. Click on 'Add Car' button from the left side drawer
2. Fill-up all the information
3. Click 'Add Car' Button

**Test Data**

- Brand : DB12
- Model : Aston Martin
- Status : Brand New
- Price : 50000

**Expected Output**

'All Cars' page can be seen along with the new car

**Actual Output**

'All Cars' page along with the new car is visible

**Result**

PASS

**Notes**

To add new car properly, every data must be filled up. If all data is not filled up and user clicks register button, a message pops up.

**9.14 Test Case: Delete a Car****Description**

Delete a new Car for the seller's car shop that is logged in currently

**Test Steps**

1. Click on 'All Cars' button from the left side drawer
2. Click on 'Delete' button of the car that should get deleted

**Expected Output**

'All Cars' page can be seen and the deleted car should not be seen

**Actual Output**

'All Cars' page is visible and the deleted car is not visible

**Result**PASS**Notes**

To add new properly, every data must be filled up. If all data is not filled up and user clicks register button, a message pops up.

**9.15 Test Case: Edit Car Details****Description**

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

**Test Steps**

1. Click on 'All Cars' button from the left side drawer
2. Click on 'Edit' button of the car that should get deleted
3. Fill-up all the information
4. Click 'Add Car' Button

**Test Data**

- Brand : DB12
- Model : Aston Martin
- Status : Brand New
- Price : 50002

**Expected Output**

'All Cars' page can be seen with the updated car data

**Actual Output**

'All Cars' page with the updated car data is visible

**Result**

PASS

**Notes**

To update new car properly, every data must be filled up. If all data is not filled up and user clicks register button, a message pops up.

**9.16 Test Case: Show car's availability and status****Description**

Show each car's availability and status as if it is available or rented or sold. On the other hand it is brand new or used. Those things can be understood by the availability and the status

**Test Steps**

1. Login to the account or click on 'New Cars' or 'Sold Cars' 'Used Cars' or 'Rented Cars' or 'All Cars' or 'Customer Requests' button from the left side drawer or click 'Comments' breadcrumbs on the top-middle of the screen

**Expected Output**

Car information's along with their availability and status can be seen

**Actual Output**

Car information's along with their availability and status is visible

**Result**PASS**Notes**

Car's availability or status can not be seen if there is no car that the seller have.

**9.17 Test Case: Make Brand New Car to Used****Description**

Seller can turn or update a car's status from brand new to used, as if the car got used in any case before selling it

**Test Steps**

1. Click on 'New Cars' button from the left side drawer
2. Click on 'Used' button of the car which status needs to be updated

**Expected Output**

New Car List Page can be seen and the updated car should not be seen there. But that very car should be seen in the 'Sold Cars' page

**Actual Output**

New Car List Page is visible and the updated car is not there. And that very car is visible in the 'Sold Cars' page

**Result**PASS**Notes**

New car list 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.

**9.18 Test Case: Make Rented Car to Available****Description**

Seller can turn or update a car's availability from rented to available, as if the car has been returned by the person who rent it and furthermore, it can be given as rent for new customers now.

**Test Steps**

1. Click on 'Rented Cars' button from the left side drawer
2. Click on 'Available' button of the car which availability needs to be updated

**Expected Output**

Rented Car List Page can be seen and the updated car should not be seen there

**Actual Output**

Rented Car List Page is visible and the updated car is not there

**Result**PASS**Notes**

New car list 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.



**9.19 Test Case: Accept Customer Request****Description**

Seller can accept the purchase or rent request from the customers

**Test Steps**

1. Click on 'Customer Requests' button from the left side drawer
2. Click on 'Accept' button of the request that needs to be accepted among those requests that are highlighted or not faded.

**Expected Output**

Customer Requests List Page can be seen and the accepted request's status should be as 'Accepted' and that request should be appeared as faded.

**Actual Output**

Customer Requests List Page is visible and the accepted request's status is visible as 'Accepted' and that request has appeared as faded.

**Result**PASS**Notes**

New car list 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.

**9.20 Test Case: Reject Customer Request****Description**

Seller can reject the purchase or rent request from the customers

**Test Steps**

1. Click on 'Customer Requests' button from the left side drawer
2. Click on 'Reject' button of the request that needs to be accepted among those requests that are highlighted or not faded.

**Expected Output**

Customer Requests List Page can be seen and the accepted request's status should be as 'Rejected' and that request should be appeared as faded.

**Actual Output**

Customer Requests List Page is visible and the accepted request's status is visible as 'Rejected' and that request has appeared as faded.

**Result**PASS**Notes**

New car list 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.

# Code Examples and Snippets

## 10.1 Routes

```

1 <?php
2
3 // Define routes
4 $routes = [
5     'INDEX' => '/index.php',
6     '500_error' => '/500.php',
7     'login' => '/view/Login.php',
8     'signup_decider' => '/view/Signup_Decider.php',
9     'seller_signup' => '/view/seller/Seller_Signup.php',
10    'seller_dashboard' => '/view/seller/Seller_Dashboard.php',
11    'seller_add_cars' => '/view/seller/Add_Car.php',
12    'seller_new_cars' => '/view/seller/New_Car.php',
13    'seller_used_cars' => '/view/seller/Used_Car.php',
14    'seller_rented_cars' => '/view/seller/Rented_Car.php',
15    'seller_all_cars' => '/view/seller/All_Cars.php',
16    'seller_car_requests' => '/view/seller/Customer_Requests.php',
17    'seller_view_car' => '/view/seller/View_Car.php',
18    'seller_edit_car' => '/view/seller/Edit_Car.php',
19    'seller_sold_car' => '/view/seller/Sold_Car.php',
20    'seller_all_comments' => '/view/seller/All_Comments.php',
21 ];
22
23 $backend_routes = [
24     'login_controller' => '/controller/LoginController.php',
25     'logout_controller' => '/controller/LogoutController.php',
26     'seller_signup_controller' => '/controller/seller/
27         Seller_SignupController.php',
28     'seller_add_car_controller' => '/controller/seller/Add_CarController.php',
29     'seller_new_cars_used_controller' => '/controller/seller/New_CarController/
30         New_Car_UsedController.php',
31     'seller_rented_cars_available_controller' => '/controller/seller/
32         Rented_CarController/Rented_Car_AvailableController.php',
33     'seller_all_cars_delete_controller' => '/controller/seller/All_CarsController/
34         All_Cars_DeleteController.php',
35     'seller_edit_car_controller' => '/controller/seller/All_CarsController/
36         All_Cars_EditController.php',
37     'seller_car_request_accept_controller' => '/controller/seller/
38         Customer_RequestsController/Customer_Requests_AcceptController.php',
39     'seller_car_request_reject_controller' => '/controller/seller/
40         Customer_RequestsController/Customer_Requests_RejectController.php',
41 ];
42
43
44 $image_routes = [
45     'contact' => '/view/static/image/contact.jpg',
46     'seller_image' => '/view/static/image/seller.jpg',
47     'user_icon' => '/view/static/image/user.png',
48 ];
49
50 ?>

```

**Listing 10.1:** PHP Validation Code Example

# 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 development and successful implementation of the comprehensive car dealership management system marks a significant achievement in streamlining dealership operations for sellers. This robust system provides an integrated web platform that enhances inventory management, from the addition and removal of vehicles to the updating of listings with ease. The inclusion of detailed reporting tools empowers sellers with the ability to generate comprehensive PDF reports on sales, revenue, and profits, facilitating informed business decisions and performance tracking.

Throughout the development process, the project adhered strictly to the planned timeline and budget, culminating in its completion within 4 days and a budget of 3000/- BDT. The system's user-friendly interface and comprehensive functionalities have garnered positive feedback from the customer, who has expressed satisfaction with its ease of use and effectiveness.

This project demonstrates the potential for technological solutions to significantly improve operational efficiency in car dealership management. By automating and simplifying various aspects of inventory and sales management, this system not only saves time but also enhances accuracy and productivity for sellers. The success of this project underscores our commitment to delivering high-quality, practical solutions that meet the specific needs of our clients.

# Assets

---

- <sup>[1]</sup> Lucidchart. Lucidchart - online diagram and flowchart software. <https://lucid.app/>. Accessed: 2024-05-29.
- <sup>[2]</sup> MindMeister. Mindmeister - online mind mapping. <https://www.mindmeister.com/>. Accessed: 2024-05-29.
- <sup>[3]</sup> Dbdiagram.io. Dbdiagram.io - database relationship diagrams design tool. <https://dbdiagram.io/home>. Accessed: 2024-05-29.
- <sup>[4]</sup> diagrams.net. Diagrams.net (formerly draw.io). <https://app.diagrams.net/>. Accessed: 2024-05-29.
- <sup>[5]</sup> Figma. Figma - the collaborative interface design tool. <https://www.figma.com/>. Accessed: 2024-05-29.