**ONLINE WEBSITE FOR THE FASHIONER**

**A PROJECT REPORT**

**(Project Work I Phase I)**

**Submitted by**

**ABISHEK A**

**19CSR004**

**GOKULA KANNAN G**

**19CSR044**

**HAREESH RAJ R**

**19CSR049**

***in partial fulfillment of the requirements***

***for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING **

**KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE – 638 060**

**AUGUST 2022**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE – 638 060**

**AUGUST 2022**

**BONAFIDE CERTIFICATE**

This is to certify that the Project report entitled **THE FASHIONER** is the bonafide record of the project work done by **ABISHEK A (Register No: 19CSR004), GOKULAKANNAN G (Register No: 19CSR044), HAREESH RAJ R (Register No: 19CSR049),** in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Science and Engineering** of Anna University Chennai during the year 2022.

**SUPERVISOR HEAD OF THE DEPARTMENT**

**Date: (Signature with seal)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE – 638 060**

**AUGUST 2022**

**DECLARATION**

We affirm that the Project Report titled **THE FASHIONER** being submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**ABISHEK A**

**(Reg. No: 19CSR004)**

**GOKULAKANNAN G**

**(Reg. No: 19CSR044)**

**HAREESH RAJ R**

**(Reg. No: 19CSR049)**

**Date:**

I certify that the declaration made by the above candidates is true to the best of my knowledge.

**Date:**  **Name and Signature of the Supervisor with seal**

**ABSTRACT**

This is an application for a company named The Fashioner. It is a clothing store. This project is a solution for the company to sell their products online and enlarge their business and increase sales. It involves two types of users. One is a customer and the admin. In this application customers can only view and buy the products and the admin can add and update the stocks.

This project is used for ordering and maintaining the orders placed by the customers. It is also used to monitor the product stock and orders and customer details. Compared to the existing system of using pen and paper to maintain records this can be very helpful and can also reduce work by a huge margin.

This web application is intended to reduce the work of updating the customer about the new arrivals and offers now this website will help to frequently update the customers and also increase sales. Apart from the primary goal which is selling products online, it can also be used for advertisement by giving a homepage for the company. All these goals are satisfied by the project without compromising the security and data integrity of the application.

**ACKNOWLEDGEMENT**

Owing deeply to the supreme, we extend our thanks to the Almighty, who has blessed us to come out successfully with our project. We take pleasure to express our deep sense of gratitude to our beloved parents.

We express our sincere thanks and gratitude to our beloved Correspondent, **Thiru. P.SACHITHANANDAN** for allowing me to pursue this course. We are extremely thankful with no words of formal nature to the dynamic Principal, **Dr.V.BALUSAMY B.E (Hons), MTech, Ph.D.,** for providing the necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department, **Dr.N.SHANTHI B.E., M.E., Ph.D.,** for providing the necessary facilities.

We also express our sincere thanks to our project coordinator, **Mrs. S. MOHANASARANYA M.E.,** Assistant Professor for her constant encouragement of the development of our project.

We take immense pleasure to express our heartfelt thanks to our supervisor, **Ms.N.SASIPRIYAA M.E.,** Assistant Professor(SRG), Computer Science and Engineering, for her valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department for their support.

**CHAPTER 1**

**INTRODUCTION**

**Introduction**

In this project, you will have your own website that your customers can visit online so that customers don’t need to always physically visit the store when buying new clothes. This Website will maintain the stocks and display the products and users can place orders and purchase the products. So, this saves us time and it does not need any manual update of arrival to the customers via phone call, message, mail, etc. So, these problems will be overcome by building a web application.

**Existing System**

The company provides various solutions but it does not address the problem of managing service calls. A separate API is designed and another front-end web application for viewing it anywhere. This backend can later be connected with any kind of application. But currently, the hardware stores do not have any kind of system to do this. All this process is maintained in paper records. Besides maintaining service calls and engineer performances this website can also display information about the store itself.

**System study**

This web application has two types of users. One is the customer and then the admin. A customer has the privilege to view the products and place an order. Once the order is placed the customer's problem is that the admin can view the order and the admin will dispatch the goods to the customer at the specified address and can also provide rating and feedback. The admin has the privilege to see the pending orders that are assigned to them. Based on their performance feedback is given. The admin has control over the entire application. They have the privilege to update the price and stock of the goods and can also add new products to their website.

**Objective**

The main objective of this project is to solve the problem of maintaining and delivering the products to the customers. With this project, a customer would be able to place an order from anywhere. After ordering the admin will contact them. Apart from serving benefits to the customer, it also benefits the shop. They can manage the records of orders. The admin can view the orders and also manage the sales performance.

**Scope**

The future scope for this project is quite big. Since the project itself is designed as separate projects, the front end, and the back end. Both can be used as plug-and-play. Other front-end applications can use the data from the backend to display information. Similarly, any other backend application can be designed for the front end. This can also be updated with many features according to the needs.

**CHAPTER 2**

**GENERAL DESCRIPTION**

**Project perspective**

The main aim of the proposed system is to design a web application for The Fashioner to provide to their customers to buy products online. This web application can also be used to advertise their store. Admin has control over the entire application. The customers will also be able to get their services with less hassle.

**Limitations**

1. Payment portal is not included so every transaction must be done offline.

**Design and implementation constraints**

This application provides clear details about the products and orders booked by the customers. Although the admin must maintain the stock and orders placed by the customer which can be tedious.

**CHAPTER 3**

**REQUIREMENTS**

**Functional requirements**

A functional requirement describes a particular behavior or function of the system. Authorization levels, authentication, and customer contents are the requirements of the system. This website should have modules to display various tour packages category-wise with adequate pictures. Also, it should have a provision for getting customer feedback.

**Authorization level**

There is one level of authentication in the application. Admin login has the highest priority among all the users. Admin can upload the images, and include or modify various packages and its cost. He can also keep a record of the users signed in. The customers could only view the details that get displayed on the website.

**Authentication**

The admin can log in without any previous registration because the admin can directly store their details in the database. The customer can log in using their registered mobile number and password.

**Non-functional requirements**

These are the extra features that are given along with the functional requirements. They are:

**Usability**

This application can be used on any device with a browser

**Availability**

The application will be hosted on a server online providing 24/7 availability

**Scalability**

Java Servlet applications are scalable by nature and based on the number of users the company can also change the number of instances of the applications

**Performance**

It performs better than other applications because It creates a thread for each request, not a process, thus performing better than any other language application.

**Security**

The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the firm can gain access to the database that contains details about the customers. Generally, through customer login, they can view the products and their own cart and can place the order. No customer can view/modify the orders placed by other customers. Additionally, changes in images, order details, and cost details can only be viewed by the customer.

**User interface requirement**

1. Home page is just a website about the company. From the home page if the user is logged in then they can go to the dashboard otherwise they can log in

2. On the sign-up page the user can either log in or register. Users need to provide their name, mobile number, address, and password. All these fields are validated properly in both the front end and back end.

3. On the home page based on the user type the navigation is displayed

**Software requirements**

**HTML**

Hypertext Markup Language is the standard language for creating web and web With Sheets and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Browsers receive HTML documents from or from local storage and render them into multimedia web pages. HTML describes the structure of a web page including cues for the appearance of the document. elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as forms, may be embedded into the rendered page. It provides a means to create documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets HTML can embed programs 10 written in a language such as JavaScript which affects the behavior and content of web pages.

**CSS**

Cascading Style Sheets fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML. 12

**Internal style sheet**

An internal style sheet holds the CSS code for the web page in the head section of the file. This makes it easy to apply styles like classes or IDs to reuse the code. The downside of using an internal style sheet is that changes to the internal style sheet only affect the page the code is inserted into.

**External style sheet**

The External Style sheet is a . CSS file that you link your website to. This makes it so that whatever you change in the. CSS sheets will affect every page on your website. This prevents you from having to make many code changes on each page. This is for "***global***" site changes.

**Inline style sheet**

The Inline style is specific to the tag itself. The inline style uses the HTML"*style*" attribute to style a specific tag. This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it. The Inline style is good for one individual CSS change that you do not use repeatedly throughout the site.

**Javascript**

It is the language used for developing both the front end and the back end of the application. It is the native language of the web. This is used in the front end to provide interactivity. It is used in the back end for computations and it is also asynchronous

**SQL Server**

It is the database used in our application to store data. It provides a means to store data and query it in a structured form like tables. This helps in maintaining relationships and avoiding redundancies in data after normalization.

**JAVA Servlet**

A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

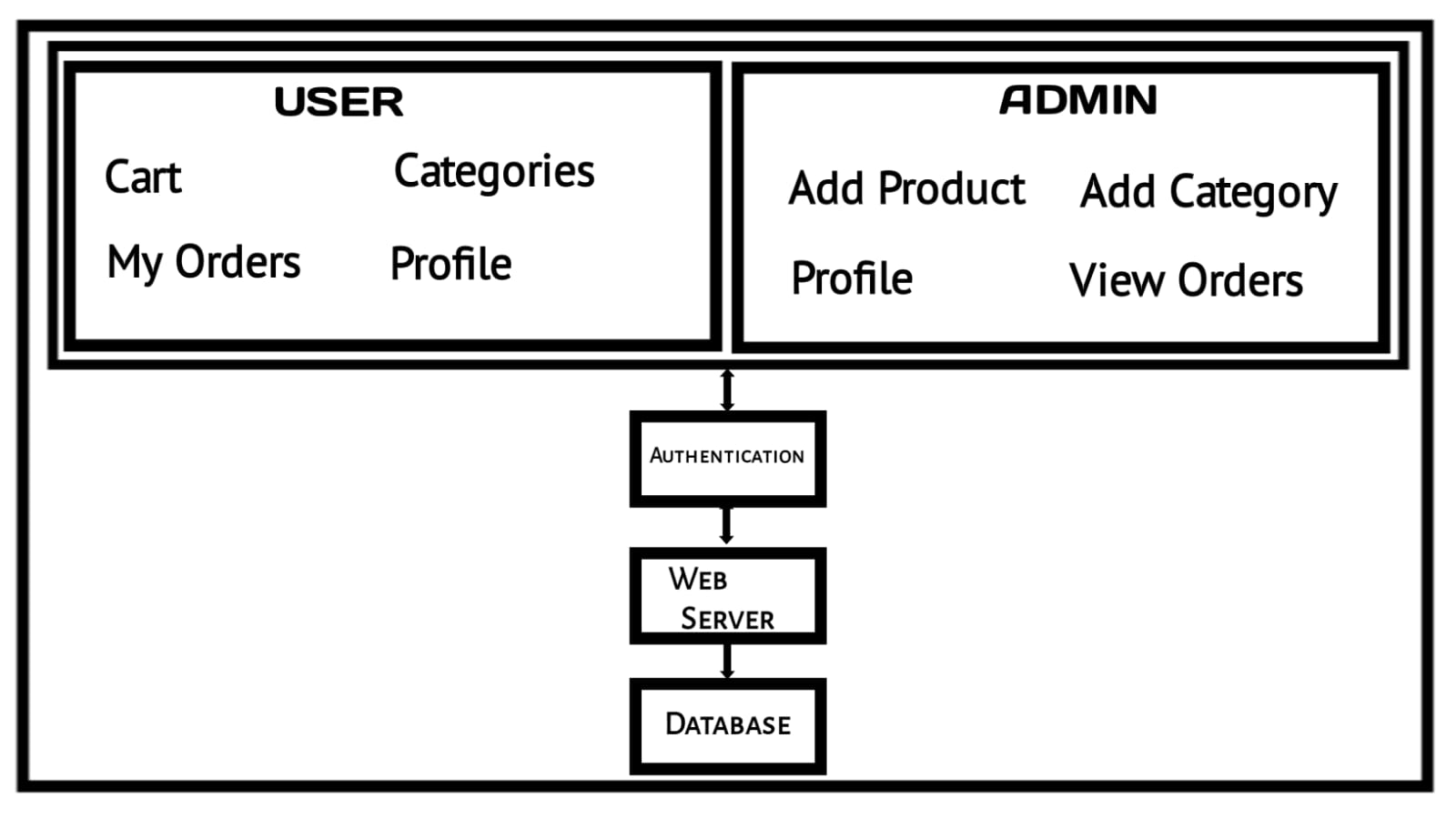
**AJAX**

Ajax is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page

**CHAPTER 4**

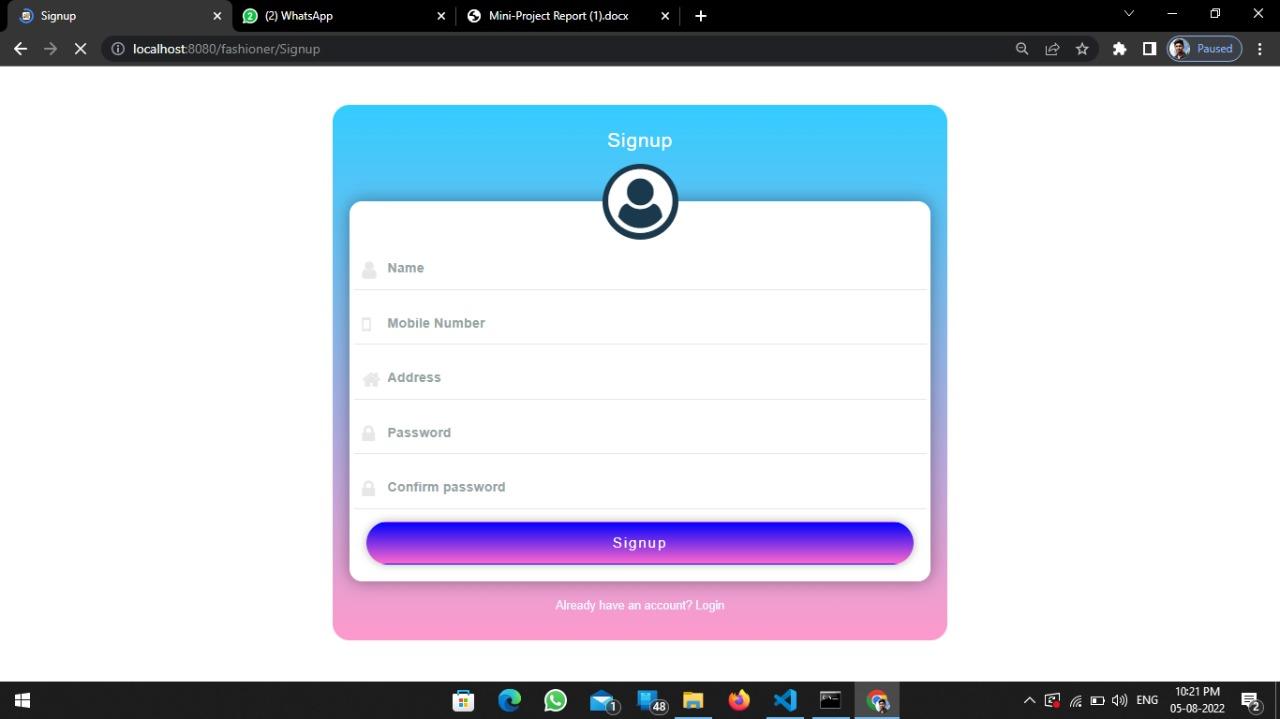
**DETAILED DESIGN**

**ARCHITECTURAL DESIGN**

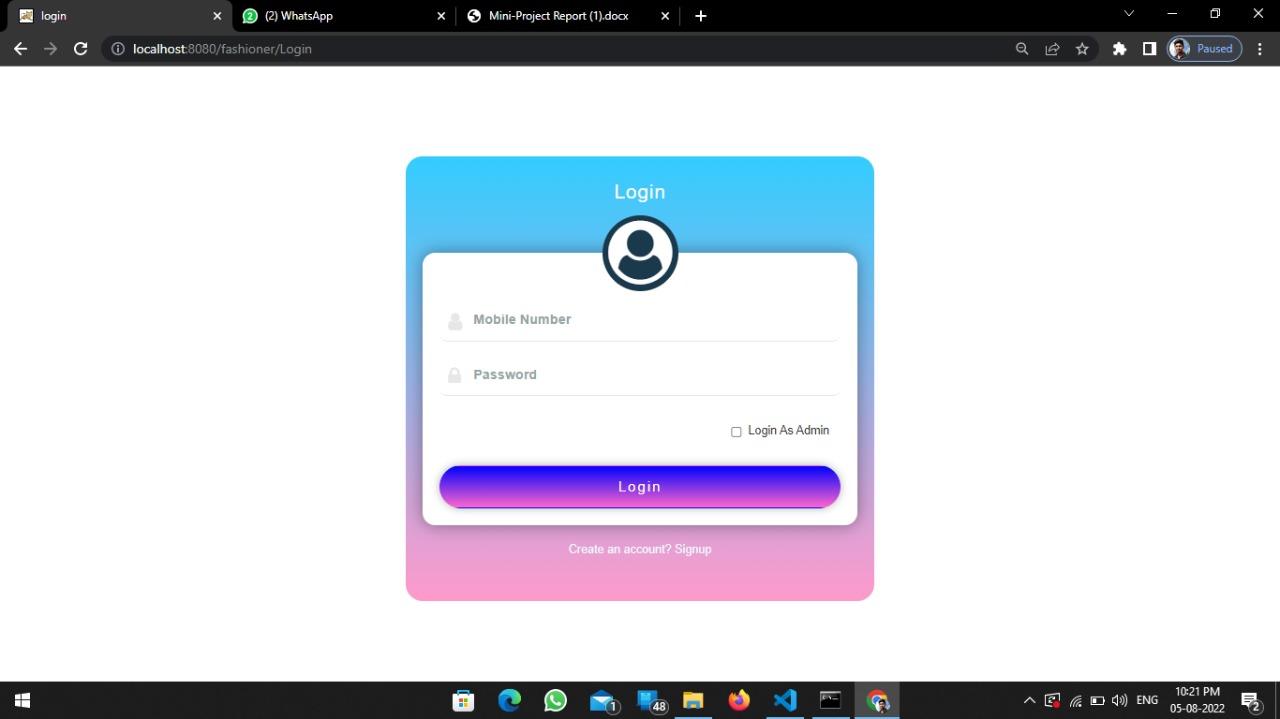
****

**USER DESIGN**

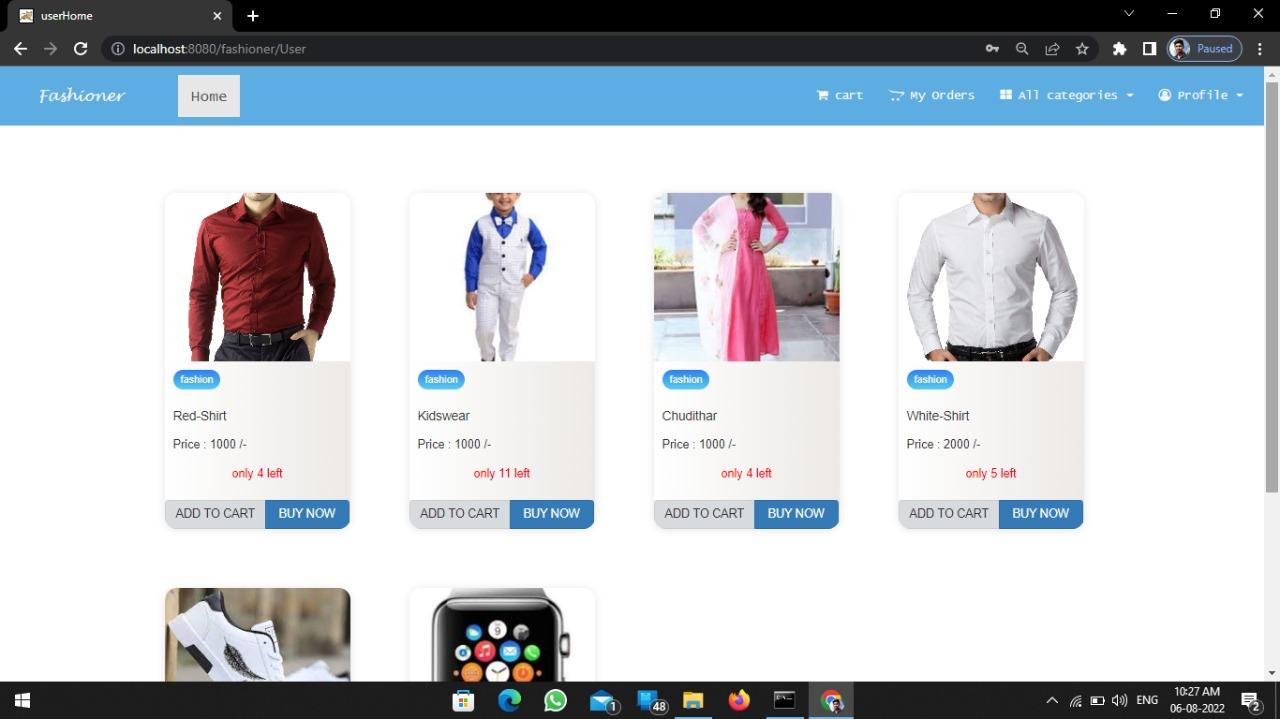
**Sign-up Page**

****

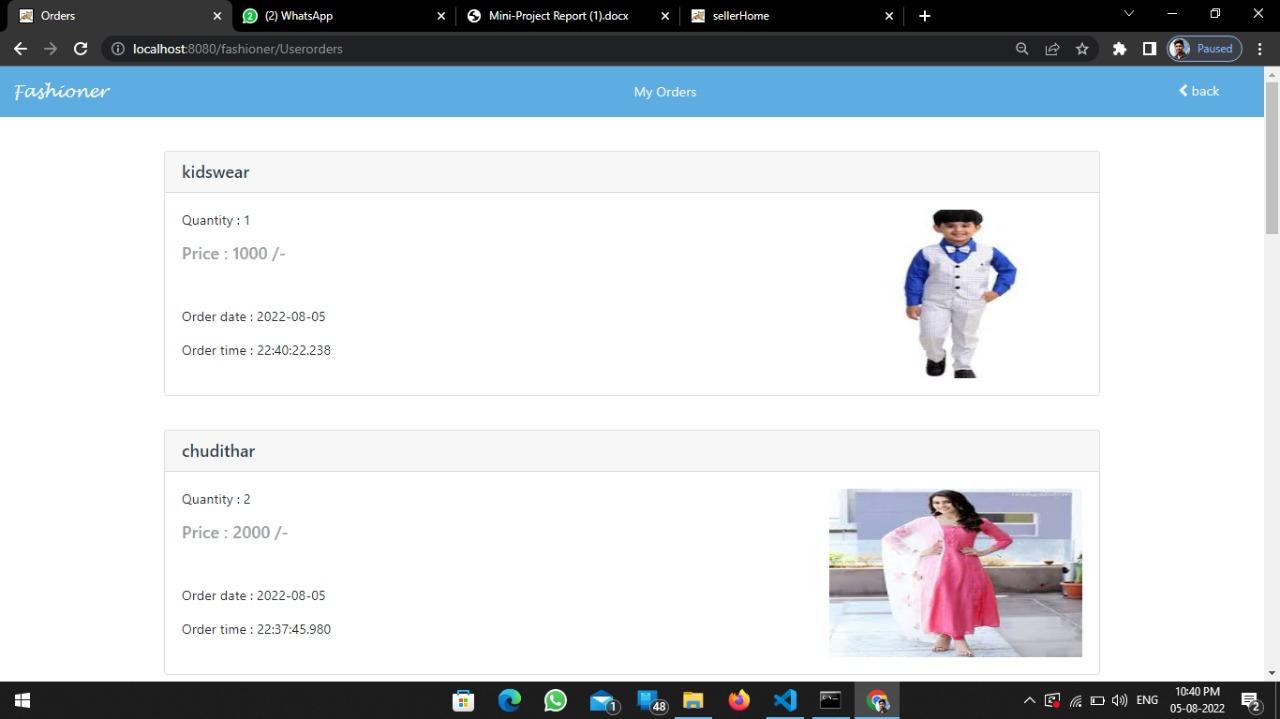
**Login Page**

****

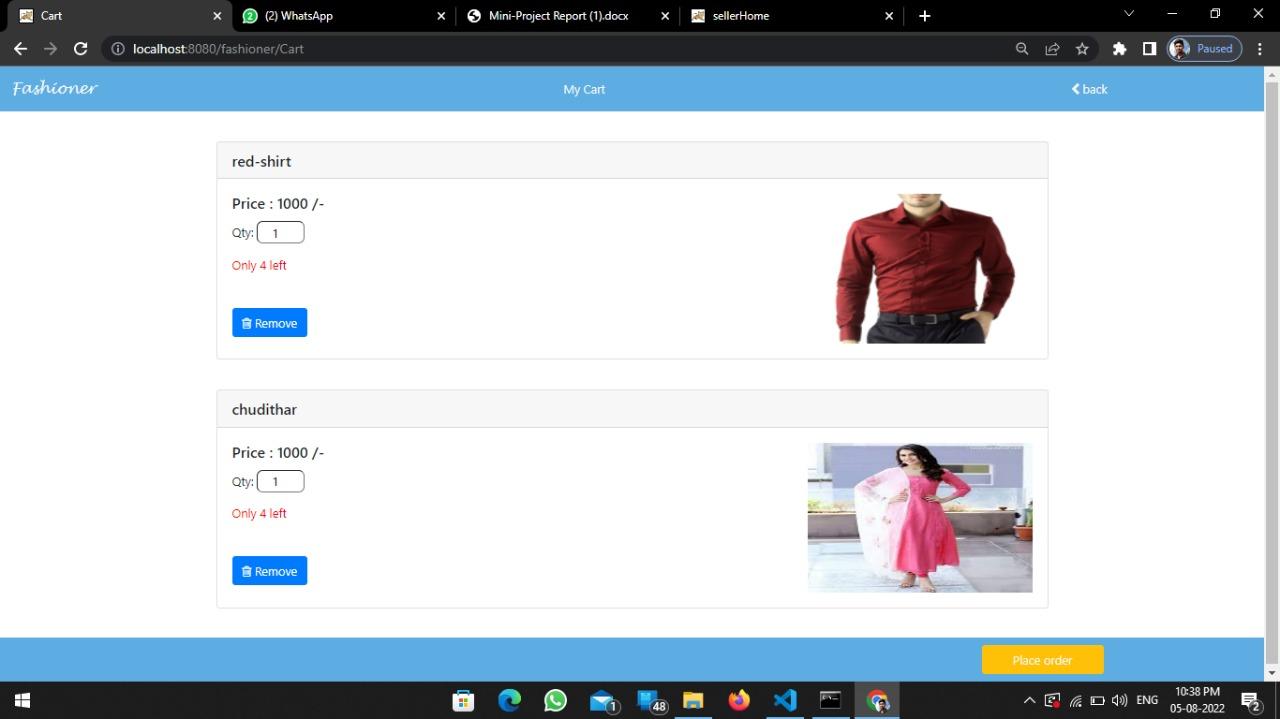
**Home Page**

****

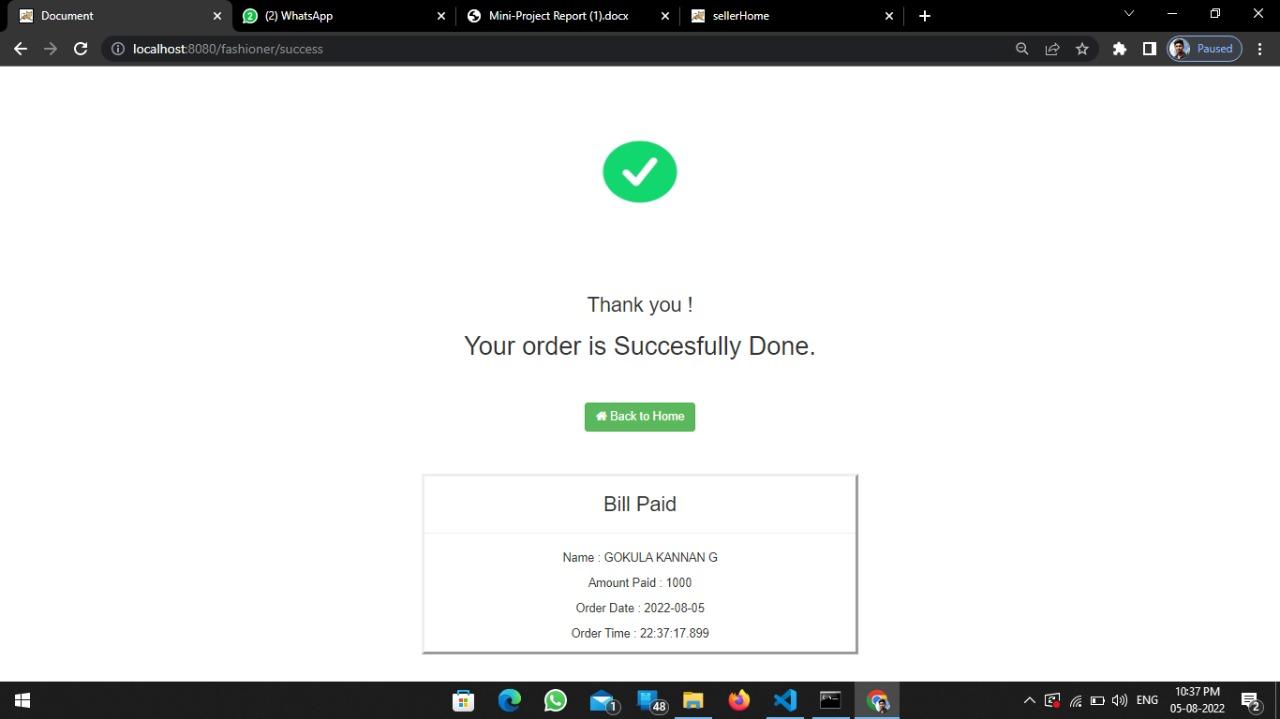
**MY ORDERS**

****

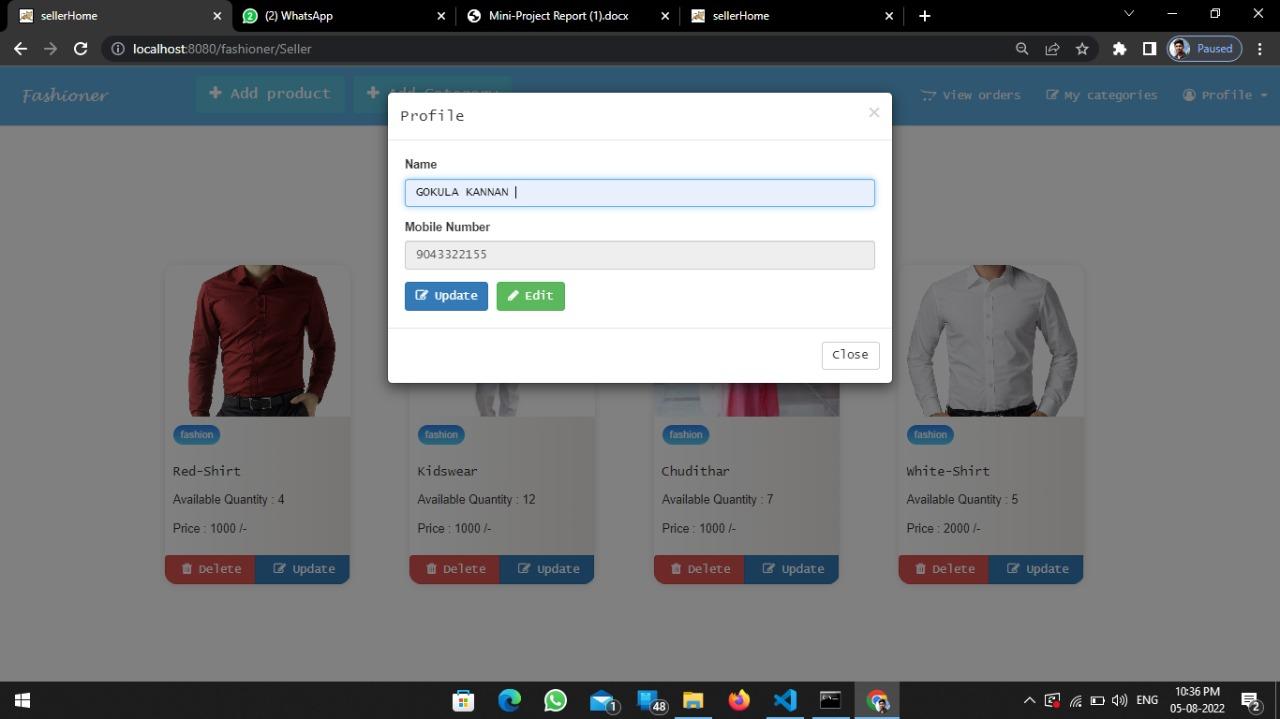
**MY CART**

****

**ORDER SUMMARY**

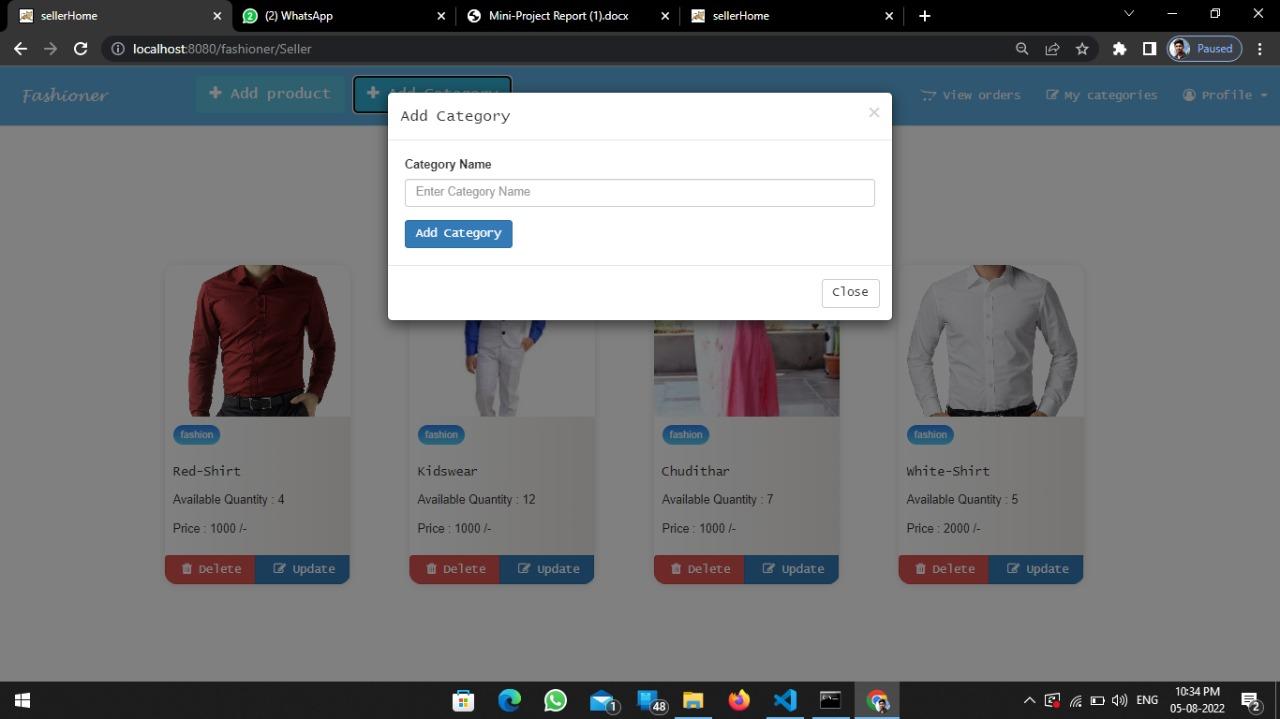
****

**UPDATE PROFILE**

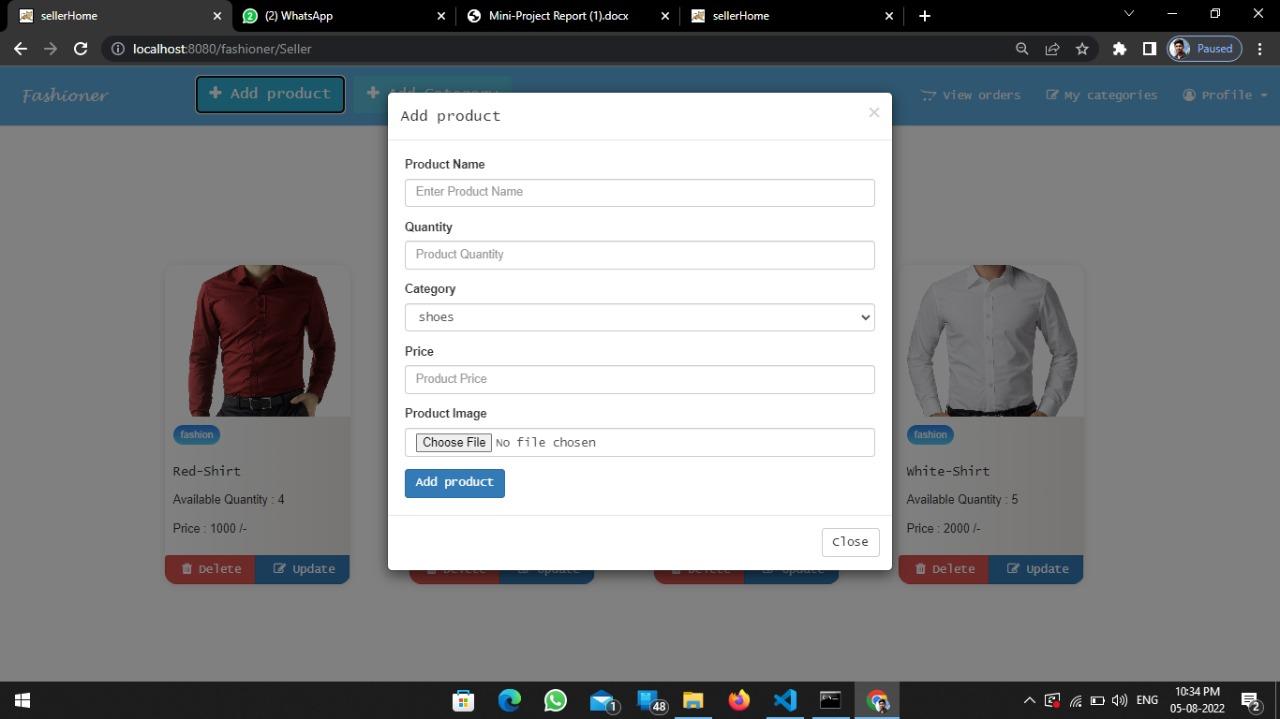
****

**ADMIN - INTERFACE**

**ADD CATEGORY**

****

**ADD PRODUCT**

****

**CHAPTER 5**

**MODULE DESCRIPTION**

**Modules description**

**User interface Module**

The user interface has different packages, login/register modules, and order management modules.

**Profile**

1. This module is for customers to view their details.

2. The user is allowed to update their details whenever it is necessary.

**Cart**

1. In this user can add or delete products to the cart.

2. From the cart the user can place the order.

**Orders**

1. In this the user can view the orders.

2. The admin will process the pending orders.

**Categories**

1. It shows the categories.

2. It is exclusively for admin.

**Admin interface Module**

The admin interface has different packages, login/register modules, and call management modules.

**Add Product**

1. It is exclusively for admin.

2. The admin will add products to the database that will be displayed

on the user's home page.

**Add Category**

1. It displays different reports like calls and feedback.

2. It is exclusively for admins.

**Profile**

1. It is for admin and engineers.

2. The admin will assign the pending calls to the engineer.

3. The service engineer will service the pending calls.

**View Orders**

1. It shows a detailed view of the history of customer services.

2. It is exclusively for admin.

**Authentication**

1. All users who are associated with services need to log in or register.

2. This helps to manage the history of each user.

3. For a customer, their call history can be stored.

4. For an engineer, their performance history can be stored.

**Server**

1. These scripts are used to interact with the database and provide information.

2. They also help in authentication with hashing algorithms, providing tokens, etc.

**Database**

1. This is the module used to persist data.

2. It stores all data in a manner such that it is easy to retrieve.

**CHAPTER 6**

**TESTING**

**Validation testing**

Testing is the only way to know if the system works properly before deployment. It helps improve the quality of the product. Different kinds of testing are done here.

Test 1:

While registering none of the fields must be empty.

Mobile Number must be valid.

This is checked in both the front end and back end.

Test 2:

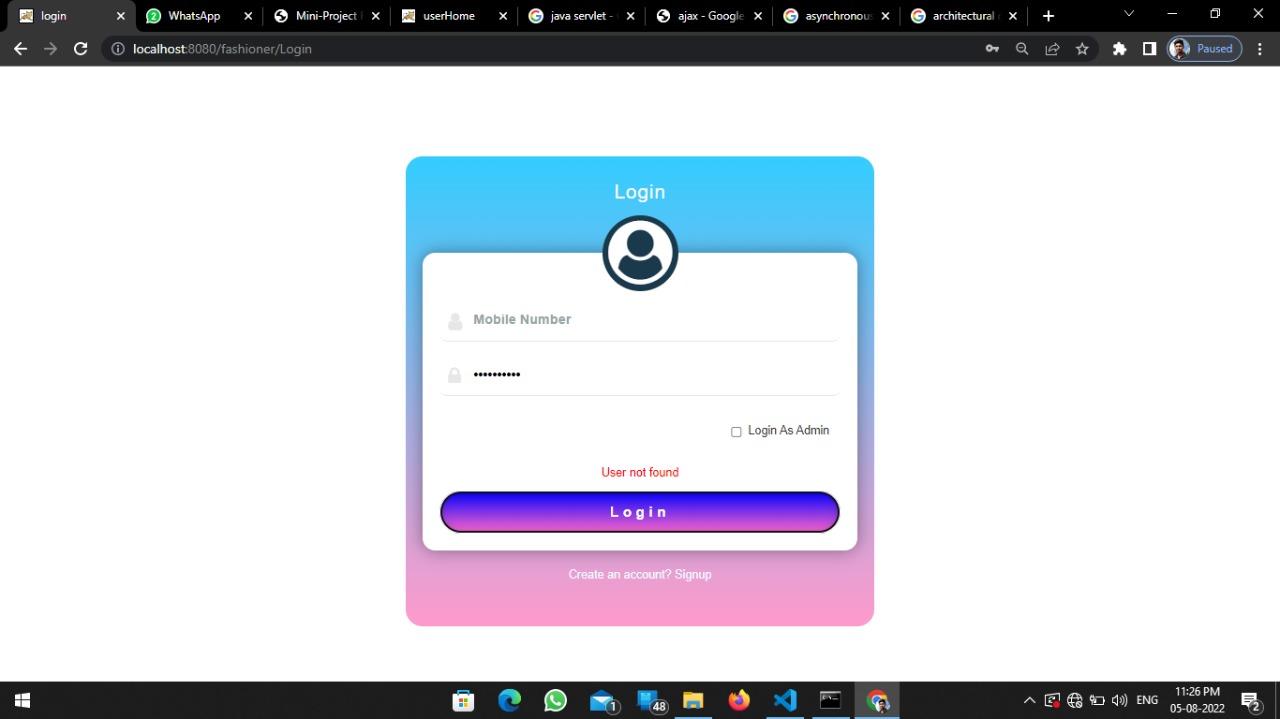
In login, the mobile number and the password of the user must match

This is checked only on the back end.

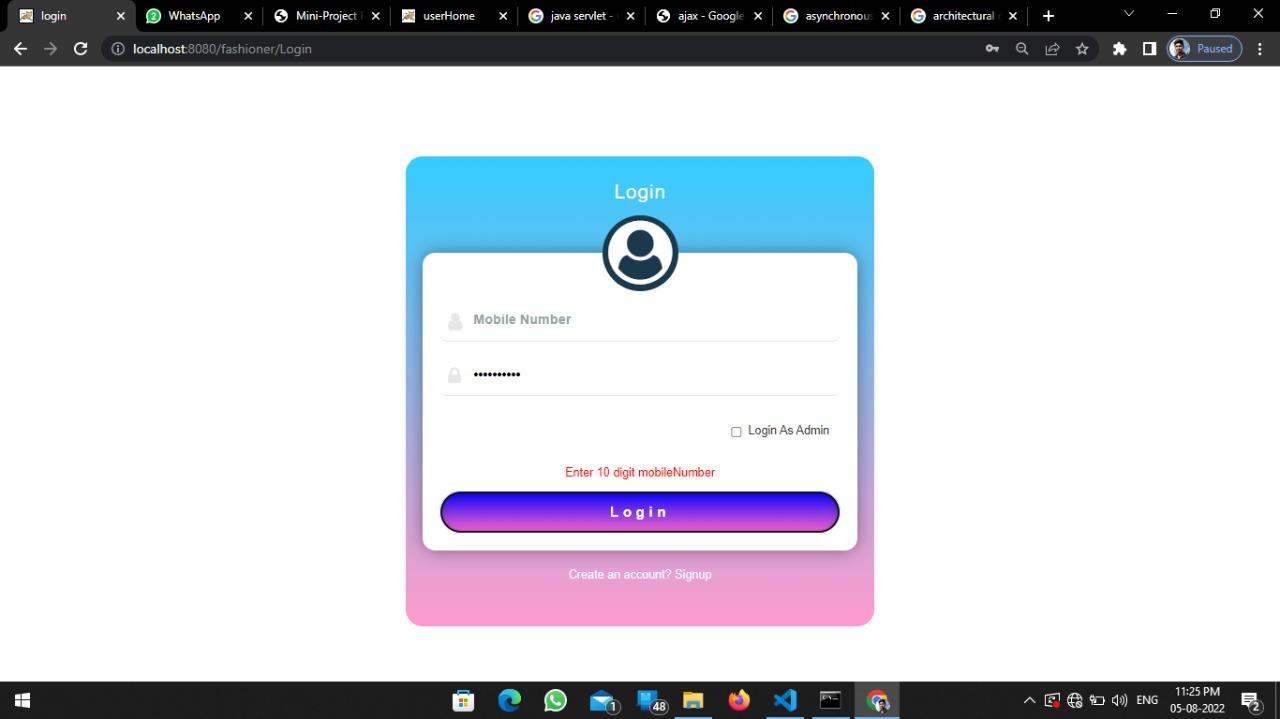
The results are shown in the front end.

**Snapshots Of Validation Testing**

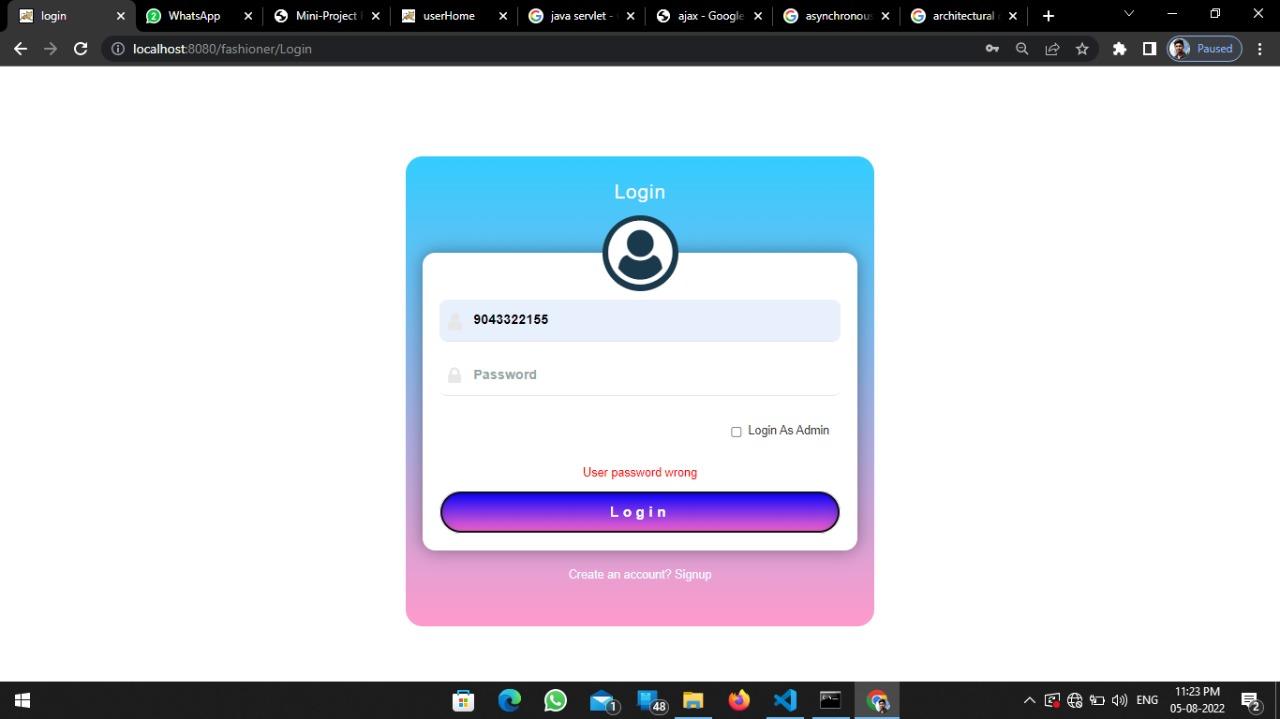
**User not found**

****

**Invalid Mobile Number**

****

**Invalid Password**

****

**Unit testing**

Integration Testing is a part of software testing where different individual units are combined and tested as a group. The purpose of integration testing is to find out the faults in the interaction between integrated units.

In our project, this testing was done continuously when implementing a functionality. This was tested manually.

**Verification Testing**

The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase.

**Test case 1** The UI must be completely responsive and must satisfy the client's requirement

**Test case 2**  Data stored must not be redundant The project properly manages the company details and information is stored properly.

**Test analysis**

The system has successfully passed the mentioned cases and works well.

**CHAPTER 7**

**CONCLUSION AND FUTURE SCOPE**

**Conclusion and future work**

Developing a web application “THE FASHIONER '' is surely a profitable deal as this application helps to carry out tasks with ease and thereby reduces time and money on manpower and materials. The proposed system helps users to get an idea about the company and its services. Using this website, customers can view various arrivals that are offered by the fashioner and they can provide feedback on their services. In the future, the project can be enhanced by providing online payment.

**CHAPTER 8**

**APPENDICES**

**CODE**

**UserHome.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>userHome</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="resources/js/checkUser.js"></script>

<script src="resources/js/getUserProduct.js"></script>

<script src="resources/js/getUserCategory.js"></script>

<script src="resources/js/getUserProfile.js"></script>

<link rel="stylesheet" href="resources/css/userHome.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<nav class="navbar navbar-default navbar-fixed-top" style="padding: 10px;background-color: #5DADE2 ; ">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#"

style="font-family:Lucida Handwriting;color: #F4F6F7;margin-left: 5px;">Fashioner</a>

</div>

<ul class="nav navbar-nav" style="margin-left: 50px">

<li class="active"><a onclick="onloadhome()" style=" font-family: Lucida Console;font-size: large;">Home</a>

</li>

<li><a style="color: #F4F6F7;font-family:Lucida Console;margin-left: 400px;" id="filterproduct"> </a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a style="color: #F4F6F7;font-family:Lucida Console;" onclick="loadcartpage()"> <i

class="fa fa-shopping-cart" aria-hidden="true"></i> cart</a></li>

<li><a style="color: #F4F6F7;font-family:Lucida Console;" onclick="loadorderpage()"> <i class="fa fa-opencart"

aria-hidden="true"></i> My Orders</a></li>

<li class="dropdown">

<a class="dropdown-toggle" data-toggle="dropdown"

style="font-family:Lucida Console;color: #F4F6F7 ;background-color: #5DADE2; "><i class="fa fa-th-large"

aria-hidden="true"></i> All categories

<span class="caret"></span></a>

<ul class="dropdown-menu" id="getUserCategory">

</ul>

</li>

<li class="dropdown">

<a class="dropdown-toggle" data-toggle="dropdown"

style="font-family:Lucida Console;color: #F4F6F7 ;background-color: #5DADE2; "><span

class="fa fa-user-circle-o"></span> Profile

<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#" data-toggle="modal" data-target="#profileModal" style="font-family: Lucida Console;"><span

class="fa fa-user"></span> My Profile</a></li>

<li><a href="#" data-toggle="modal" data-target="#passwordModal" style="font-family: Lucida Console;"><i

class="fa fa-unlock-alt" aria-hidden="true"></i> Change password</a></li>

<hr>

<li><a href="#" style="font-family: Lucida Console;" onclick="logout()"><span

class="glyphicon glyphicon-log-in"></span> Logout</a></li>

</ul>

</li>

</ul>

</div>

</nav>

<!-- -->

<div class="container-fluid" style="margin-left: 180px;">

<div class="cont" id="productcard" style="margin-top: 150px;">

<!-- display card div -->

</div>

</div>

<!-- buy now modal -->

<!-- Modal -->

<div class="modal fade" id="buynowModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Order Summary</h4>

</div>

<div style="padding:20px ;">

<form id="buyNow" name="buyNow">

<div class="form-group">

<input type="hidden" class="form-control" id="buynowProductid" name="buynowProductid" disabled>

</div>

<div class="form-group">

<input type="hidden" class="form-control" id="totalqty" name="totalqty" disabled>

</div>

<div class="form-group">

<label for="email">Product Name</label>

<input type="text" class="form-control" id="buynowProductname" name="buynowProductname" disabled>

</div>

<div class="form-group">

<label for="email">price</label>

<input type="text" class="form-control" id="buynowprice" name="buynowprice" disabled>

</div>

<div class="form-group">

<label for="">Quantity</label>

<input type="number" min="0" max="" class="form-control" id="buynowquantity" name="buynowquantity"

onchange="totalpriceupdate()">

</div>

<div class="form-group">

<label for="email">Total price</label>

<input type="text" class="form-control" id="buynowtotal" name="buynowtotal" disabled>

</div>

<div class="form-group">

<label for="email">Address</label>

<input type="text" class="form-control" id="buynowaddress" name="buynowaddress" disabled>

</div>

<p id="" style="color:red ;text-align: center;"></p>

<button type="submit" class="btn btn-primary"> Place Order </button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<!-- view profile modal -->

<!-- Modal -->

<div class="modal fade" id="profileModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Profile</h4>

</div>

<div style="padding:20px ;">

<form>

<div class="form-group">

<label for="name">Name</label>

<input type="text" class="form-control" id="username" name="username" disabled>

</div>

<div class="form-group">

<label for="mobile">Mobile Number</label>

<input type="text" class="form-control" id="usernumber" name="usernumber" disabled>

</div>

<div class="form-group">

<label for="address">Address</label>

<input type="text" class="form-control" id="useraddress" name="useraddress" disabled>

</div>

<p id="" style="color:red ;text-align: center;"></p>

<button type="button" class="btn btn-primary" onclick="updatename()"><i class="fa fa-pencil-square-o"

aria-hidden="true"></i> Update <button type="button" class="btn btn-success" style="margin-left: 10px;"

onclick="editname()"><i class="fa fa-pencil"></i> Edit</button> </button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<!-- change password modal -->

<!-- Modal -->

<div class="modal fade" id="passwordModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Change password</h4>

</div>

<div style="padding:20px ;">

<form id="changepassword">

<div class="form-group">

<label for="email">Old password</label>

<input type="text" class="form-control" id="oldpassword" name="oldpassword">

</div>

<div class="form-group">

<label for="email">New password</label>

<input type="text" class="form-control" id="newpassword" name="newpassword">

</div>

<p id="passworderror" style="color:red ;text-align: center;"></p>

<button type="submit" class="btn btn-primary">Change password</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<script src="resources/js/userHome.js"></script>

</body>

</html>

**SellerHome.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>sellerHome</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="resources/js/checkUser.js"></script>

<script src="resources/js/getProduct.js"></script>

<script src="resources/js/getProfile.js"></script>

<script src="resources/js/getCategory.js"></script>

<script src="resources/js/editCategory.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="resources/css/sellerHome.css">

</head>

<body>

<nav class="navbar navbar-default navbar-fixed-top" style="padding: 10px;background-color: #5DADE2 ; ">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#" style="font-family:Lucida Handwriting;color: #F4F6F7;" >Fashioner</a>

</div>

<ul class="nav navbar-nav" style="margin-left: 90px;">

<!-- <li class="active"><a href="#" style=" font-family: Lucida Console;font-size: large;">Home</a></li> -->

<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#productModal" style="font-family: Lucida Console;"><i class="fa fa-plus"></i> Add product</button>

<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#categoryModal" style="font-family: Lucida Console;"><i class="fa fa-plus"></i> Add Category</button>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="#" style="font-family: Lucida Console;color: #F4F6F7;" onclick="vieworders()"><i class="fa fa-opencart" aria-hidden="true"></i> View orders</a></li>

<li><a href="#" data-toggle="modal" data-target="#editcategoryModal" style="font-family: Lucida Console;color: aliceblue;" ><i class="fa fa-pencil-square-o" aria-hidden="true"></i> My categories</a></li>

<li class="dropdown">

<a class="dropdown-toggle" data-toggle="dropdown" style="font-family:Lucida Console;color: #F4F6F7 ;background-color: #5DADE2; "><span class="fa fa-user-circle-o" ></span> Profile

<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#" data-toggle="modal" data-target="#profileModal" style="font-family: Lucida Console;" ><span class="fa fa-user"></span> My Profile</a></li>

<li><a href="#" data-toggle="modal" data-target="#passwordModal" style="font-family: Lucida Console;" ><i class="fa fa-unlock-alt" aria-hidden="true"></i> Change password</a></li>

<hr><li><a href="#" style="font-family: Lucida Console;" onclick="logout()"><span class="glyphicon glyphicon-log-in"></span> Logout</a></li>

</ul>

</li>

</ul>

</div>

</nav>

<!-- product card -->

<div class="container-fluid" style="margin-left: 180px;">

<div class="cont" id="productcard" style="margin-top: 140px;">

<!-- display card div -->

</div>

</div>

<!-- Edit category modal -->

<!-- Modal -->

<div class="modal fade" id="editcategoryModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Edit category</h4>

</div>

<div style="padding:20px ;">

<form id="editcategory">

<!-- your category -->

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<!-- view profile modal -->

<!-- Modal -->

<div class="modal fade" id="profileModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Profile</h4>

</div>

<div style="padding:20px ;">

<form >

<div class="form-group">

<label for="email">Name</label>

<input type="text" class="form-control" id="sellername" name="sellername" disabled>

</div>

<div class="form-group">

<label for="email">Mobile Number</label>

<input type="text" class="form-control" id="sellernumber" name="sellernumber" disabled>

</div>

<p id="" style="color:red ;text-align: center;"></p>

<button type="button" class="btn btn-primary" onclick="updatename()"><i class="fa fa-pencil-square-o" aria-hidden="true"></i> Update <button type="button" class="btn btn-success" style="margin-left: 10px;" onclick="editname()"><i class="fa fa-pencil"></i> Edit</button> </button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<!-- change password modal -->

<!-- Modal -->

<div class="modal fade" id="passwordModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Change password</h4>

</div>

<div style="padding:20px ;">

<form id="changepassword">

<div class="form-group">

<label for="email">Old password</label>

<input type="text" class="form-control" id="oldpassword" name="oldpassword" >

</div>

<div class="form-group">

<label for="email">New password</label>

<input type="text" class="form-control" id="newpassword" name="newpassword" >

</div>

<p id="passworderror" style="color:red ;text-align: center;"></p>

<button type="submit" class="btn btn-primary">Change password</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<!-- Modal -->

<!-- Trigger the modal with a button -->

<!-- Add product Modal -->

<div class="modal fade" id="productModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Add product</h4>

</div>

<div style="padding:20px ;">

<form name="product" id="product" method="post" enctype="multipart/form-data">

<div class="form-group">

<label for="email">Product Name</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Enter Product Name" required>

</div>

<div class="form-group">

<label for="pwd">Quantity</label>

<input type="text" class="form-control" id="quantity" name="quantity" placeholder="Product Quantity" required>

</div>

<div class="form-group">

<label for="pwd">Category</label>

<select class="form-control" id="category" name="category">

<!-- dynamic Category -->

</select>

</div>

<div class="form-group">

<label for="pwd">Price</label>

<input type="text" class="form-control" id="price" name="price" placeholder="Product Price" required>

</div>

<div class="form-group">

<label for="pwd">Product Image</label>

<input type="file" class="form-control" id="image\_upload" name="file" required>

</div>

<p id="producterror" style="color:red ;text-align: center;"></p>

<button type="submit" class="btn btn-primary">Add product</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<!-- Add Category modal -->

<!-- Modal -->

<div class="modal fade" id="categoryModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Add Category</h4>

</div>

<div style="padding:20px ;">

<form name="productCategory" id="productCategory">

<div class="form-group">

<label for="email">Category Name</label>

<input type="text" class="form-control" id="categoryname" name="categoryname" placeholder="Enter Category Name">

</div>

<p id="categoryerror" style="color:red ;text-align: center;"></p>

<button type="submit" class="btn btn-primary">Add Category</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<script src="resources/js/sellerHome.js"></script>

</body>

</html>

**userHome.js**

//buynow

$("#buyNow").submit(function (e) {

var productid = $("#buynowProductid").val();

var quantity = $("#buynowquantity").val();

var price = $("#buynowtotal").val();

var totalquantity = $("#totalqty").val();

// alert(productid+" "+price+" "+quantity);

if (confirm("Are you sure to place order?")) {

var data = {

"buynowProductid": productid,

"buynowquantity": quantity,

"buynowtotal": price,

"totalquantity": totalquantity

};

$.ajax({

url: "buynow",

type: "POST",

data: data,

success: function (res) {

console.log(res);

window.location.replace("success");

},

error: function (res) {

alert("Something went wrong");

}

});

}

e.preventDefault();

});

//totalpriceupdate

function totalpriceupdate() {

var quantity = $("#buynowquantity").val();

var price = $("#buynowprice").val();

$("#buynowtotal").val(quantity \* price);

}

//set order summary

function setordersummary(name, price, quantity, productid, address) {

var qty = $("#qty").val();

// alert(name+" "+price+" "+qty);

$("#buynowProductname").val(name);

$("#buynowprice").val(price);

$("#buynowaddress").val(address);

$("#buynowquantity").val(1);

$("#buynowquantity").attr("max", quantity);

$("#buynowtotal").val(price);

$("#buynowProductid").val(productid);

$("#totalqty").val(quantity);

// <p>Qty: <input type="number" size="2" min="0" max=${res[i].quantity} id="qty" style="background-color:#EAEDED;border-radius:8px;text-align:center "></p>

}

//add to cart

function addToCart(productId) {

// alert("product id "+productId);

data = {

"productid": productId

};

$.ajax({

url: "addtocart",

type: "POST",

data: data,

success: function (res) {

console.log(res);

alert("Product added to cart");

},

error: function (res) {

console.log(res);

if (res.responseText === "product already exist") {

alert("product already exist in cart");

}

}

})

}

//load cart page

function loadcartpage() {

window.location.replace("Cart");

}

// load order page

function loadorderpage() {

window.location.replace("Userorders");

}

//load home page

function onloadhome() {

window.location.replace("User");

}

//get category products

function getcategoryproduct(categoryid) {

// alert("category id"+categoryid);

var data = {

id: categoryid

};

$.ajax({

url: "getcategoryproduct",

type: "GET",

data: data,

success: function (res) {

console.log(res);

document.querySelector("#filterproduct").innerHTML = res[0].categoryname;

document.querySelector("#productcard").innerHTML = "";

var dynamic = document.querySelector("#productcard");

for (var i = 0; i < res.length - 1; i++) {

var fetch = document.querySelector("#productcard").innerHTML;

dynamic.innerHTML = ` <div class="card">

<div class="card\_\_header">

<img src="uploads/${res[i].image}" alt="card\_\_image" class="card\_\_image" width="600">

</div>

<div class="card\_\_body">

<p style=" font-size:12px" class="tag tag-blue">${res[i].categoryname}</p>

<h4>${res[i].productname}</h4>

<p>Price : ${res[i].price} /-</p>

<p style="color:red;text-align:center" >only ${res[i].quantity} left</p>

</div>

<div class="btn-group">

<button type="button" class="btn btn-default" id="cartbtn" onclick="addToCart(${res[i].productId})">ADD TO CART</button>

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#buynowModal" onclick="setordersummary('${res[i].productname}',${res[i].price},${res[i].quantity},${res[i].productId},'${res[res.length - 1].address}')" >BUY NOW</button>

</div>

</div>`+ fetch;

}

}

})

}

//change password

$("#changepassword").submit(function (e) {

if (confirm("Are you sure to change password?")) {

var form = $(this);

$.ajax({

url: "changeuserpassword",

type: "POST",

data: form.serialize(),

success: function (res) {

console.log(res);

alert("succesfully password changed");

$.ajax({

url: "logout",

type: "GET",

});

window.location.replace("Login");

},

error: function (res) {

if (res.responseText === "Password mismatch") {

$("#oldpassword").val("");

}

document.getElementById("passworderror").innerHTML = res.responseText;

}

});

}

e.preventDefault();

});

// update name

function updatename() {

if (confirm("Are you sure to update profile?")) {

var name = $("#username").val();

var address = $("#useraddress").val();

var data = {

"name": name,

"address": address

};

$.ajax({

url: "updateusername",

type: "POST",

data: data,

success: function (res) {

window.location.replace("User");

}

});

}

}

//set edit name

function editname() {

document.getElementById("username").disabled = false;

document.getElementById("useraddress").disabled = false;

}

**Login.js**

$("#login").submit(function (e) {

document.getElementById("error").innerHTML = "";

var mobilenumber = $("#mobilenumber").val();

var password = $("#password").val();

var isChecked = $("#check").is(':checked');

var role;

if (isChecked == true) {

role = "Seller";

} else {

role = "User";

}

// alert("number "+mobilenumber+" password "+password+" role "+role);

var data = {

"mobilenumber": mobilenumber,

"password": password,

"role": role

};

$.ajax({

url: "login",

type: "POST",

data: data,

success: function (res) {

console.log("success " + res);

alert("Succesfully loggedin as " + res.role);

if (res.role == "User") {

window.location.replace("User");

} else if (res.role == "Seller") {

window.location.replace("Seller");

}

},

error: function (res) {

console.log(res);

if (res.responseText === "Enter 10 digit mobileNumber") {

$("#mobilenumber").val("");

}

else if (res.responseText === "User not found" || res.responseText === "Seller not found") {

$("#mobilenumber").val("");

}

else if (res.responseText === "User password wrong" || res.responseText === "Seller password wrong") {

$("#password").val("");

}

document.getElementById("error").innerHTML = res.responseText;

}

});

e.preventDefault();

});

**SignUp.js**

$("#signup").submit(function(e)

{

document.getElementById("error").innerHTML="";

var form = $(this);

$.ajax({

url:"signup",

type:"POST",

data:form.serialize(),

success:function(res){

console.log("success "+res);

alert("Succesfully signuped");

window.location.replace("Login");

},

error:function(res){

console.log(res);

if(res.responseText === "Mobile number already exist")

{

$("#mobilenumber").val("");

}

else if(res.responseText === "Enter 10 digit mobileNumber")

{

$("#mobilenumber").val("");

}

else if(res.responseText === "password mismatch")

{

$("#password").val("");

$("#confirmpassword").val("");

}

document.getElementById("error").innerHTML=res.responseText;

}

});

e.preventDefault();

});

**Back-End**

**getProfile.java**

import org.json.JSONArray;

import org.json.JSONObject;

@WebServlet("/getprofile")

public class getProfile extends HttpServlet {

protected void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

HttpSession session = req.getSession();

int sellerid = (int) session.getAttribute("id");

try {

String dbDriver = "com.mysql.jdbc.Driver";

String dbURL = "jdbc:mysql:// localhost:3306/";

String dbName = "fashioner";

String dbUsername = "root";

String dbPassword = "Gokul@2002";

Class.forName(dbDriver);

Connection con = DriverManager.getConnection(dbURL + dbName,dbUsername,dbPassword);

PreparedStatement st = con.prepareStatement("select \* from seller where sellerId=?");

st.setInt(1, sellerid);

ResultSet rs=st.executeQuery();

ResultSetMetaData rsmd = rs.getMetaData();

JSONArray json = new JSONArray();

while(rs.next()) {

int numColumns = rsmd.getColumnCount();

JSONObject obj = new JSONObject();

for (int i=1; i<=numColumns; i++) {

String column\_name = rsmd.getColumnName(i);

obj.put(column\_name, rs.getObject(column\_name));

}

json.put(obj);

}

res.setContentType("application/json");

res.setCharacterEncoding("UTF-8");

res.getWriter().print(json.toString());

res.getWriter().flush();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Updateusername.java:**

@WebServlet("/updateusername")

public class updateUserName extends HttpServlet {

protected void doPost(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

HttpSession session = req.getSession();

int userid = (int) session.getAttribute("id");

String name=req.getParameter("name");

String address=req.getParameter("address");

try {

String dbDriver = "com.mysql.jdbc.Driver";

String dbURL = "jdbc:mysql:// localhost:3306/";

String dbName = "fashioner";

String dbUsername = "root";

String dbPassword = "Gokul@2002";

Class.forName(dbDriver);

Connection con = DriverManager.getConnection(dbURL + dbName,dbUsername,dbPassword);

PreparedStatement st = con.prepareStatement("UPDATE users SET name=?,address=? where userId=?");

st.setString(1, name);

st.setString(2, address);

st.setInt(3, userid);

st.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Login.java:**

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.json.JSONObject;

@WebServlet("/login")

public class login extends HttpServlet {

public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException, ServletException {

String originalPassword = "";

int id = 0;

String mobileNumber = req.getParameter("mobilenumber");

String password = req.getParameter("password");

String role = req.getParameter("role");

try {

String dbDriver = "com.mysql.jdbc.Driver";

String dbURL = "jdbc:mysql:// localhost:3306/";

String dbName = "fashioner";

String dbUsername = "root";

String dbPassword = "Gokul@2002";

Class.forName(dbDriver);

Connection con = DriverManager.getConnection(dbURL + dbName, dbUsername, dbPassword);

if (mobileNumber.length() != 10) {

res.setStatus(406);

res.getWriter().print("Enter 10 digit mobileNumber");

return;

}

if (role.equals("User")) {

PreparedStatement st = con.prepareStatement("select \* from users where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet rs = st.executeQuery();

if (!rs.next()) {

res.setStatus(406);

res.getWriter().print("User not found");

return;

}

st = con.prepareStatement("select \* from users where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet r1 = st.executeQuery();

while (r1.next()) {

id = r1.getInt("userId");

originalPassword = r1.getString("password");

if (originalPassword.equals(password) == false) {

res.setStatus(406);

res.getWriter().print("User password wrong");

return;

}

}

} else if (role.equals("Seller")) {

PreparedStatement st = con.prepareStatement("select \* from seller where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet rs = st.executeQuery();

if (!rs.next()) {

res.setStatus(406);

res.getWriter().print("Seller not found");

return;

}

st = con.prepareStatement("select \* from seller where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet r1 = st.executeQuery();

while (r1.next()) {

id = r1.getInt("sellerId");

originalPassword = r1.getString("password");

if (originalPassword.equals(password) == false) {

res.setStatus(406);

res.getWriter().print("Seller password wrong");

return;

}

}

}

HttpSession session = req.getSession();

session.setAttribute("mobilenumber", mobileNumber);

session.setAttribute("role", role);

session.setAttribute("id", id);

session.setMaxInactiveInterval(-1);

JSONObject user = new JSONObject();

user.put("mobilenumber", mobileNumber);

user.put("password", password);

user.put("orgpassword", originalPassword);

user.put("role", role);

user.put("status", "success");

res.setContentType("application/json");

res.setCharacterEncoding("UTF-8");

res.getWriter().print(user.toString());

res.getWriter().flush();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**SignUp.java:**

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.json.JSONObject;

@WebServlet("/signup")

public class signup extends HttpServlet {

public String getDate() {

String date = java.time.LocalDate.now().toString();

return date;

}

public String getTime() {

String time = java.time.LocalTime.now().toString();

return time;

}

public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException, ServletException {

// PrintWriter out=res.getWriter();

String name = req.getParameter("name");

String mobileNumber = req.getParameter("mobilenumber");

String password = req.getParameter("password");

String confirmPassword = req.getParameter("confirmpassword");

String address = req.getParameter("address");

String role = req.getParameter("optradio");

try {

String dbDriver = "com.mysql.jdbc.Driver";

String dbURL = "jdbc:mysql:// localhost:3306/";

String dbName = "fashioner";

String dbUsername = "root";

String dbPassword = "Gokul@2002";

Class.forName(dbDriver);

Connection con = DriverManager.getConnection(dbURL + dbName, dbUsername, dbPassword);

if (role.equals("user")) {

PreparedStatement st = con.prepareStatement("select \* from users where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet r1 = st.executeQuery();

if (r1.next()) {

res.setStatus(406);

res.getWriter().print("Mobile number already exist");

return;

}

if (mobileNumber.length() != 10) {

res.setStatus(406);

res.getWriter().print("Enter 10 digit mobileNumber");

return;

}

if (password.equals(confirmPassword) == false) {

res.setStatus(406);

res.getWriter().print("password mismatch");

return;

}

st = con.prepareStatement(

"insert into users(name,mobilenumber,password,date,time,address)values(?,?,?,?,?,?)");

st.setString(1, name);

st.setString(2, mobileNumber);

st.setString(3, password);

st.setString(4, getDate());

st.setString(5, getTime());

st.setString(6, address);

st.executeUpdate();

st.close();

}

if (role.equals("seller")) {

PreparedStatement st = con.prepareStatement("select \* from seller where mobilenumber=?");

st.setString(1, mobileNumber);

ResultSet r1 = st.executeQuery();

if (r1.next()) {

res.setStatus(406);

res.getWriter().print("Mobile number already exist");

return;

}

if (mobileNumber.length() != 10) {

res.setStatus(406);

res.getWriter().print("Enter 10 digit mobileNumber");

return;

}

if (password.equals(confirmPassword) == false) {

res.setStatus(406);

res.getWriter().print("password mismatch");

return;

}

st = con.prepareStatement("insert into seller(name,mobilenumber,password,date,time) values(?,?,?,?,?)");

st.setString(1, name);

st.setString(2, mobileNumber);

st.setString(3, password);

st.setString(4, getDate());

st.setString(5, getTime());

st.executeUpdate();

st.close();

}

JSONObject user = new JSONObject();

user.put("name", name);

user.put("mobilenumber", mobileNumber);

user.put("password", password);

user.put("role", role);

user.put("Date", getDate());

user.put("time", getTime());

user.put("status", "success");

res.setContentType("application/json");

res.setCharacterEncoding("UTF-8");

res.getWriter().print(user.toString());

res.getWriter().flush();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**REFERENCES**

Javatpoint  **-** [**https://www.javatpoint.com/servlet-tutorial**](https://www.javatpoint.com/servlet-tutorial)

Tutorialspoint  **-** [**https://www.tutorialspoint.com/servlets/index.htm**](https://www.tutorialspoint.com/servlets/index.htm)

Ajax  **-** [**https://www.w3schools.com/js/js\_ajax\_intro.asp**](https://www.w3schools.com/js/js_ajax_intro.asp)

W3schools - [**https://www.w3schools.com/php/php\_mysql\_intro.asp**](https://www.w3schools.com/php/php_mysql_intro.asp)