



islington college
(इस्लिंग्टन कॉलेज)

Module Code & Module Title
CS5054NI Advanced Programming & Technologies

Assessment Type
50% Group Coursework

Semester
2024 Spring

Group Members

London Met ID	Student Name
22068751	JENISH KATUWAL
22067761	AESMARIKA ARYAL
22067807	RIJAN PAUDEL
22067773	SUJAL DEV PANT
22068022	KUSHAL GURUNG

Project Title: DIGITAL DURBAR

Assignment Due Date: Friday, May 10, 2024

Assignment Submission Date: Friday, May 10, 2024

Submitted to: Mr. Prithivi Maharjan

Word Count: 3166

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Contents

1.	Introduction.....	1
1.1.	Aims:.....	1
1.2.	Objectives:	1
2.	User Interface Design	2
2.1.	Wireframe.....	2
	7
2.2.	Actual Design.....	9
3.	Class Diagram.....	19
4.	Method Description	20
4.1.	DBConnection Class:.....	20
4.1.1.	DatabaseConnection	20
4.2.	DAO Class	21
4.2.1.	CustomerDao	21
4.3.	Servlet.....	22
4.3.1.	LoginServlet.....	22
4.3.2.	SignupServlet	22
4.3.3.	LogoutServlet	23
4.3.4.	UpdateCustomerProfileServlet	23
5.	Test Cases:.....	24
5.1	To test whether the user details get stored in the database after registration.....	24
5.2	To test whether the password authentication is valid.....	26
5.3	To test whether the order details are stored in the database after placing an order.	28
5.4	To test whether the admin can add new products.	30
5.5	To test whether the search filter works or not.	32
5.6	To test whether the user can view their history of ordered items.	33
5.7	To test whether the user can log in after registering.	35
5.8	To test whether the admin can delete an existing product.	37
5.9	To test whether the user can access the cart page while they are not logged in.	38
6.	Tools and Libraries Used:	40
6.1.	Tools used:	40
6.2.	Techniques Used:.....	42

6.3. Libraries	47
7. Developmental Process:	48
7.1. Planning the system:.....	48
7.2. Design and Fronted Development.....	48
7.3. Development.....	55
7.4. Testing	57
7.5. Maintenances:.....	57
8. Critical Analysis.....	58
9. Conclusion	62
10. References	63

Table of Figures

Figure 1: Wireframe of Register	2
Figure 2: Wireframe of Login.....	3
Figure 3: Wireframe of Home	4
Figure 4: Wireframe of Cart.....	5
Figure 5: Wireframe of About.	6
Figure 6: Wireframe Admin Panel	7
Figure 7: Wireframe User Profile.....	8
Figure 8: Register.....	9
Figure 9 login.	10
Figure 10 product.	11
Figure 11 products	12
Figure 12: About Us.	13
Figure 13: About Us	13
Figure 14: About Us	14
Figure 15: Cart.	15
Figure 16: Profile.	16
Figure 17 Order Detail.....	16
Figure 18: Admin Pane Add section.....	17
Figure 19: Admin Midsection.....	17
Figure 20: Admin Order and Ship.....	18
Figure 21: Class Diagram.....	19
Figure 22: User details	25
Figure 23: User registration data.....	25
Figure 24: Entering wrong password in login.	26
Figure 25: inserting the wrong password.....	27
Figure 26: Add to cart.....	29
Figure 27: placing order	29
Figure 28: Order details in database.	29
Figure 29: Admin login.	30
Figure 30: Adding new product.	31
Figure 31: product added.	31
Figure 32: Search function.	32
Figure 33: Display product.	33
Figure 34: add cart.	34
Figure 35: order place.	34
Figure 36: Order history in profile.	35
Figure 37: enter correct data.	36
Figure 38: home page.	36
Figure 39: Admin panel.	37
Figure 40: Deleting product.	38
Figure 41: Home page without Login.....	39

Figure 42: Please Login.....	39
Figure 43: Eclipse.....	40
Figure 44: Tomcat	40
Figure 45: balsamiq.....	41
Figure 46: Java Servlet Class	43
Figure 47:JSP	44
Figure 48: DAO CLASS.....	45
Figure 49: Creation of DIGITAL DURBAR.....	48
Figure 50: Mudita Store.....	50
Figure 51: Design reference for home.....	51
Figure 52: Bigbye IT WORLD.....	51
Figure 53: Design reference for product alignment	51
Figure 54: Mudita Footer design.	52
Figure 55: Our footer design.	52
Figure 56: UI discussion and Code sharing.....	53
Figure 57: Design Discussion.....	54
Figure 58: Database design code.	55
Figure 59: Database schema Shared.....	56
Figure 60: UI modifications.....	56
Figure 61: Challege 1	58
Figure 62: Challenge 2	59
Figure 63: Challege 1 fixed	59
Figure 64: Challenge 2	60
Figure 65: Challenge 2 fixed	61

Table of Tables

Table 1: Method Description for DbConnection.....	20
Table 2: Method Description for CustomerDao	21
Table 3: Method Description for LoginServlet	22
Table 4: Method Description for SignupServlet	22
Table 5: Method Description for LogoutServlet	23
Table 6: Method Description for UpdateCustomerProfileServlet	23
Table 7 Test 1	24
Table 8 Test 2	26
Table 9 Test 3	28
Table 10 Test 4	30
Table 11 Test 5	32
Table 12 Test 6	33
Table 13 Test 7	35
Table 14: Test 8	37
Table 15: Test 9	38

1. Introduction.

An e-commerce website for storing and selling laptops was built using the Model-View-Controller (MVC) design pattern. The Controller handles session management and updates, the View manages the user interface, and the Model controls data encryption. MySQL, Java Server Pages (JSP), Java Servlets, and the Eclipse IDE were utilized. Eclipse was used for coding and testing, MySQL for storing HTML and JSP, and Apache Tomcat as the server. The system allows customers to browse, search, add products to cart, view orders, and user authentication.

1.1. Aims:

- To develop a comprehensive e-commerce website for electronics and gadgets, following the MVC pattern.
- To demonstrate proficiency in implementing web-based applications using Java, JSP, Servlets, and integrating them with a database.
- To apply software engineering principles, including proper documentation, design patterns, and coding standards.

1.2. Objectives:

- Implement a secure user authentication system with separate roles for administrators and regular users.
- Develop an administrative panel that enables product management functionalities.
- Create a user-friendly interface for regular users, allowing them to search and browse products, add items to a shopping cart, place orders, and manage their profiles.
- Incorporate validation and exception handling mechanisms to ensure data integrity and provide appropriate error messages for better user experience.

2. User Interface Design

2.1. Wireframe

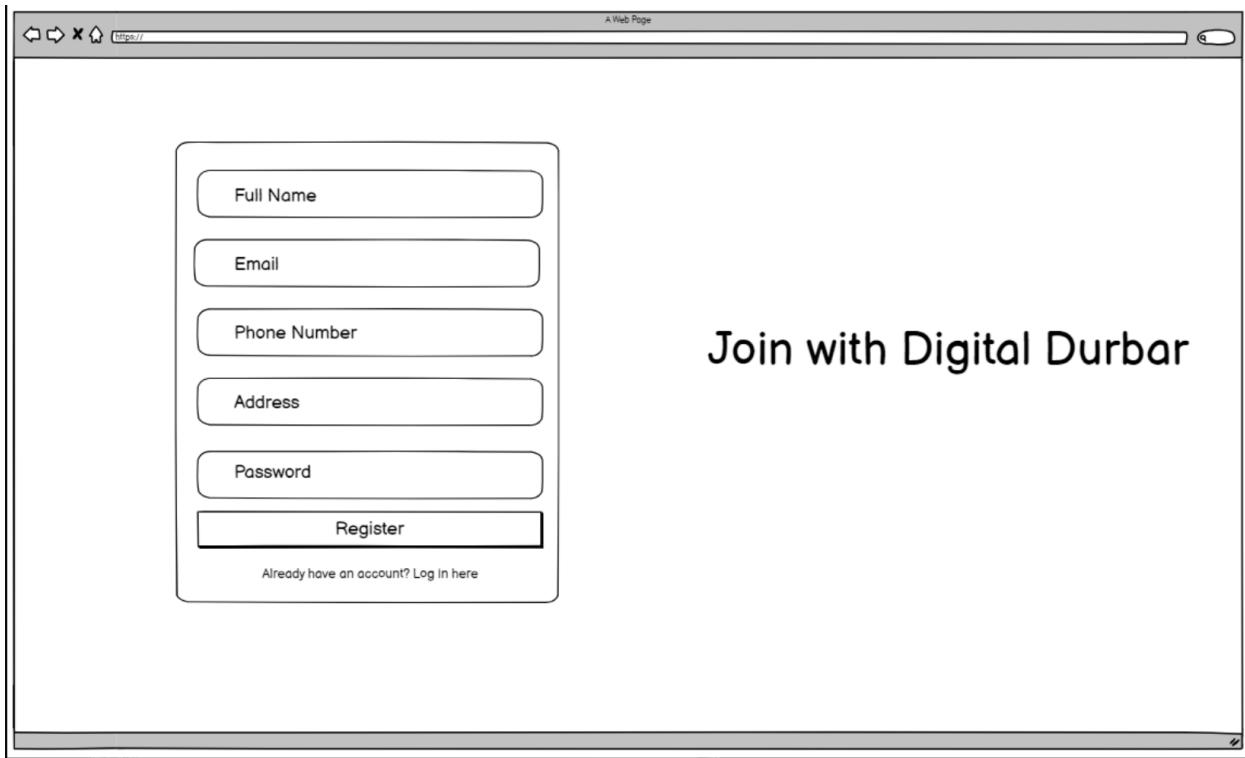


Figure 1: Wireframe of Register

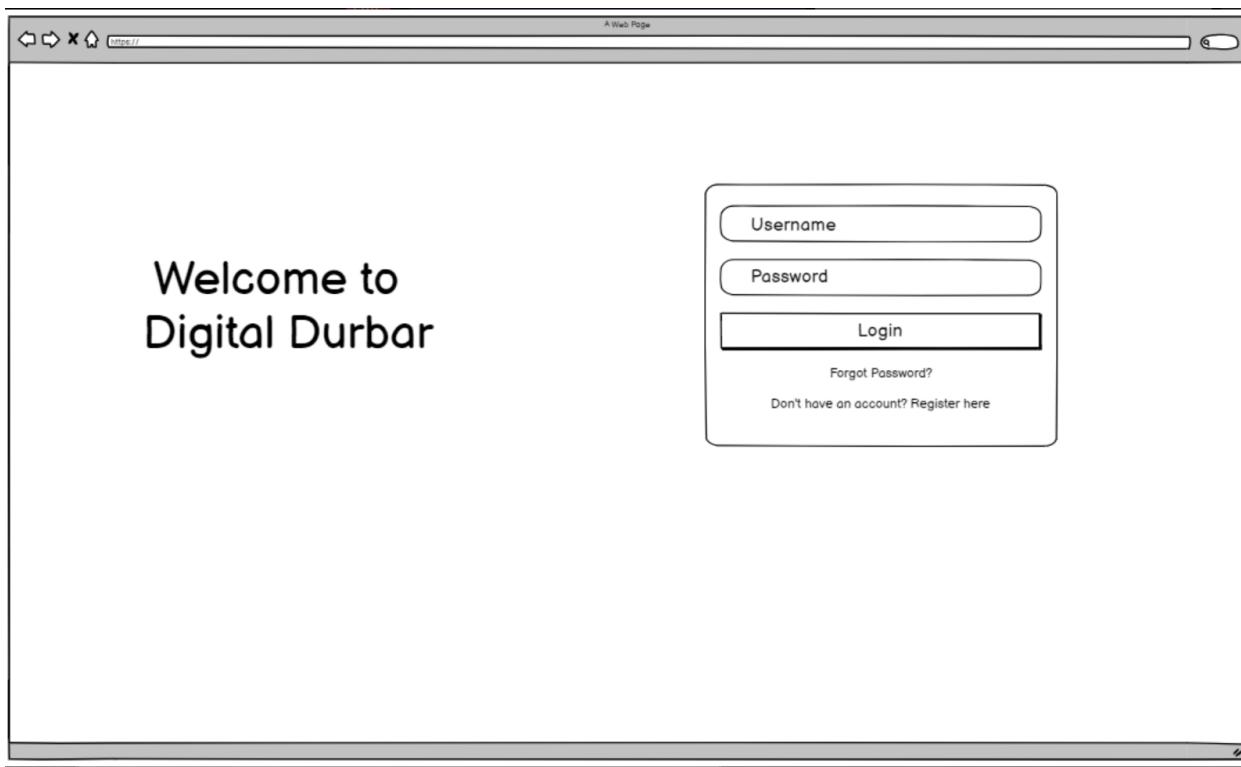


Figure 2: Wireframe of Login



Figure 3: Wireframe of Home



Figure 4: Wireframe of Cart.

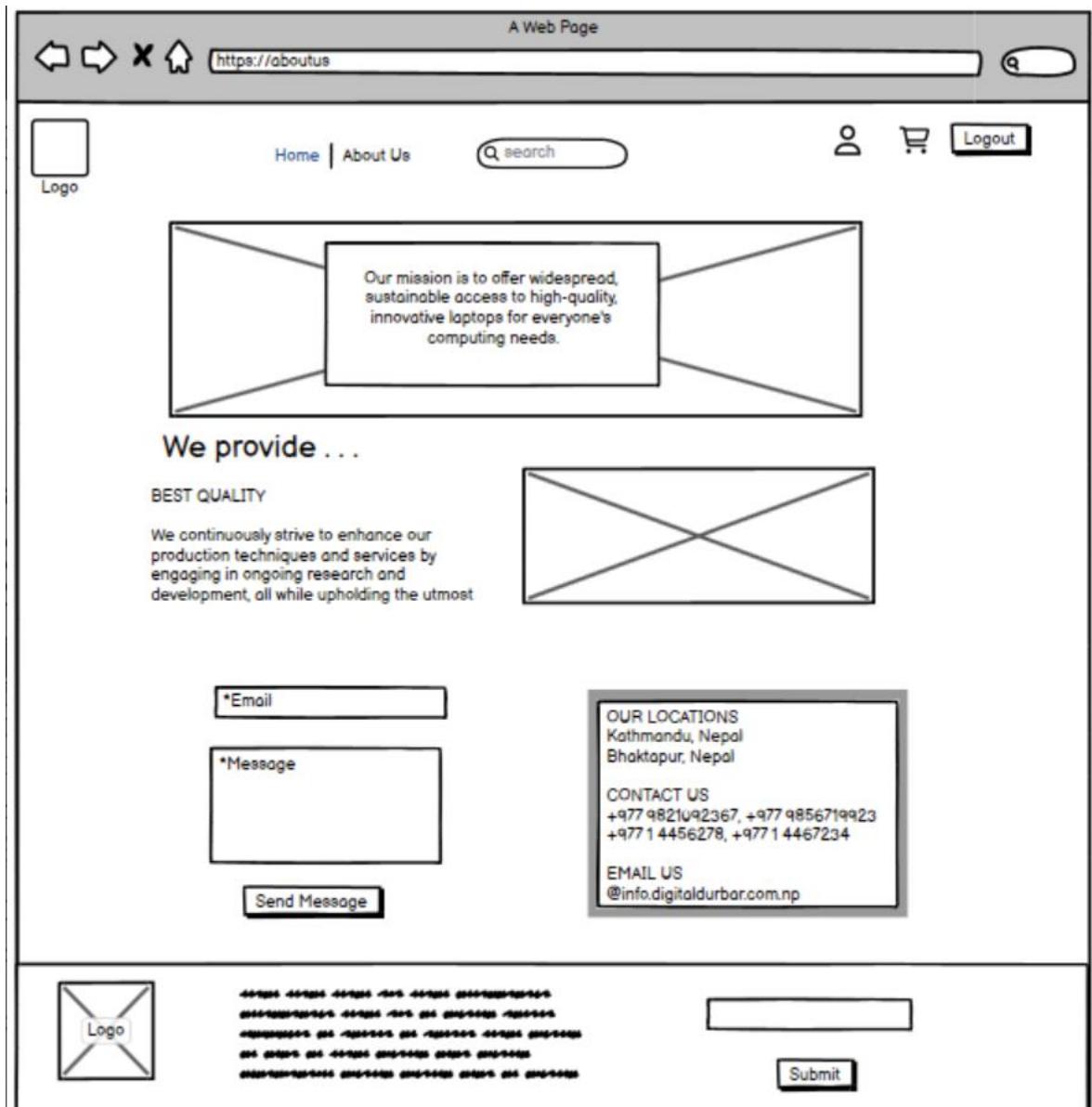


Figure 5: Wireframe of About.

The wireframe admin panel is organized into several sections:

- Top Bar:** Includes "ADMIN PANEL", "Dashboard", and "Logout".
- Add Product Section:** Contains fields for "Product Name", "Product Price", "Product Image" (with a file input placeholder "No file chosen"), and a "Add Product" button.
- Pending Order Details Section:** Displays a table with columns: ID, Product Name, Customer ID, Order Date, Total Price, Order Status, Qty, and Edit. An example row shows ID 1, Product Name Laptop, Customer ID Rs 1000, and Order Status Pending.
- Ship Order Details Section:** Displays an empty table with the same columns as the pending orders table.
- Decline Order Details Section:** Displays an empty table with the same columns as the pending orders table.

Figure 6: Wireframe Admin Panel

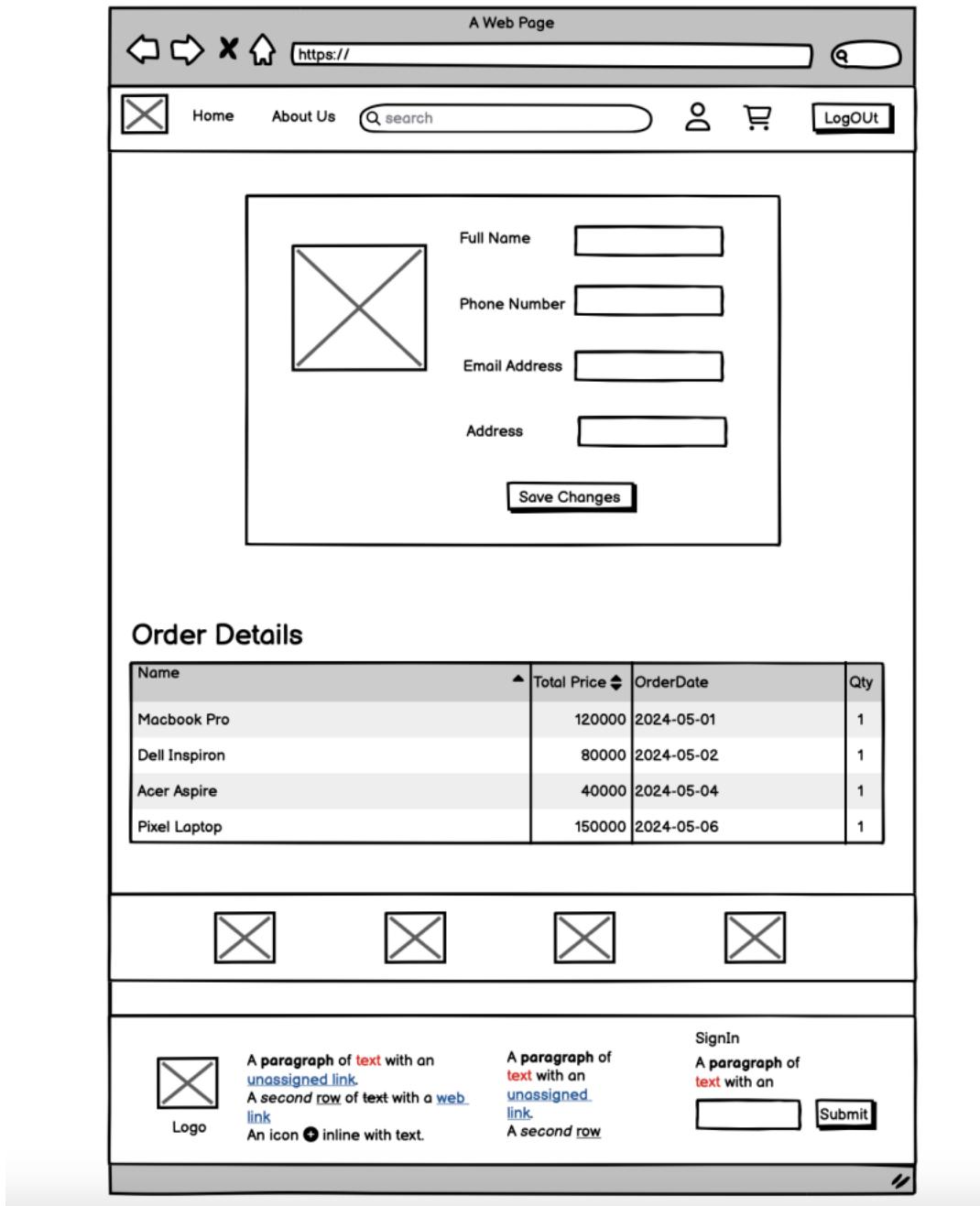


Figure 7: Wireframe User Profile

2.2. Actual Design

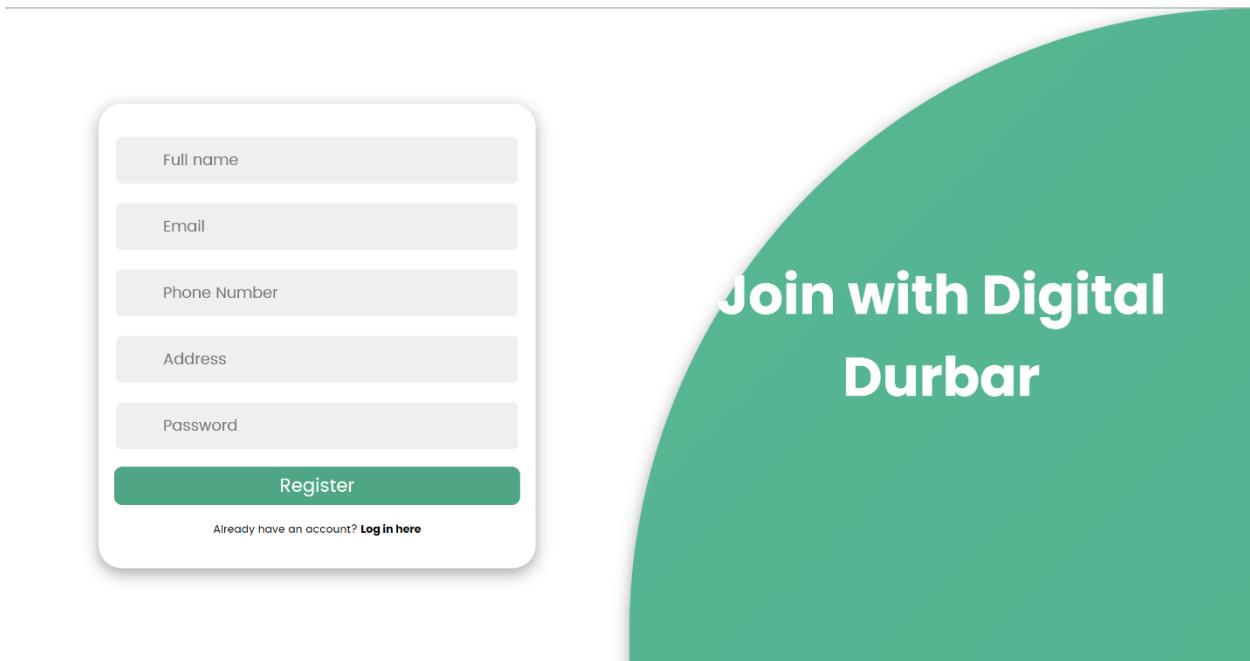


Figure 8: Register.

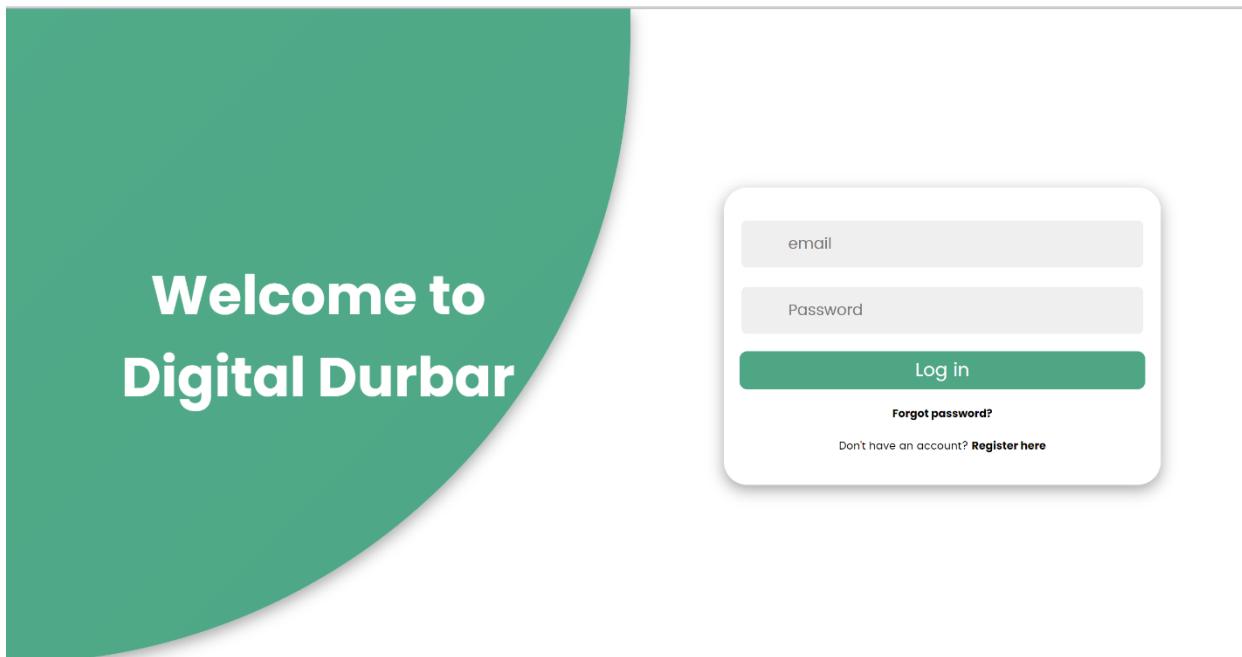
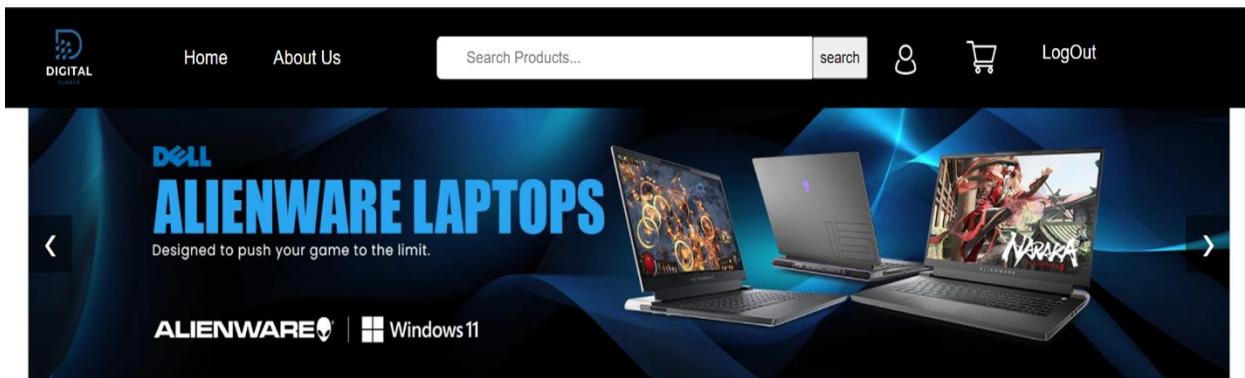


Figure 9 login.



Our Products

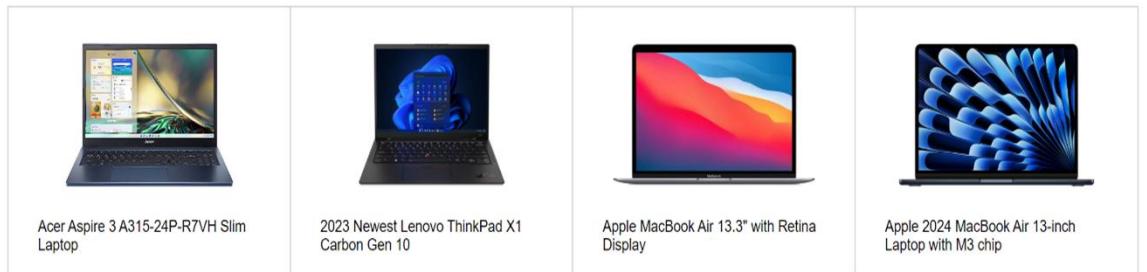


Figure 10 product.

Rs 100000	Rs 190000	Rs 87000	Rs 140000
Add to Cart			

Free Shipping
Free Shipping All Over Nepal

100% Genuine
100% Genuine Products

Easy Return Policy
3 Days Easy Return Policy

Secure Payment
Shop Online Without Hesitation

Digital Durbar is your one-stop shop for all your laptop needs. We provide high-quality laptops at affordable prices.

Contact us:
info@digitaldurbar.com

Quick Links

- [Gaming Laptops](#)
- [Business Laptops](#)
- [Student Laptops](#)
- [Workstation Laptops](#)
- [About](#)
- [Blogs](#)
- [More](#)

About

We are committed to providing the best customer experience and top-notch products. Learn more about our company and our mission.

Sign In

Please sign in to access your account or create one to start shopping.

Figure 11 products .

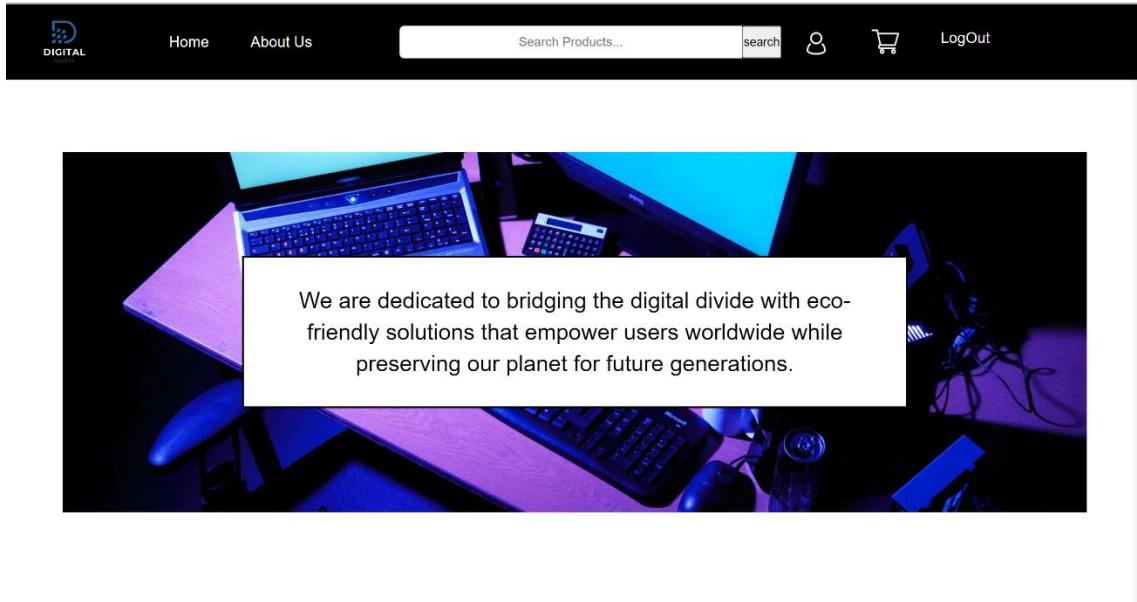


Figure 12: About Us.

Our Services

Best Quality

We continuously strive to enhance our production techniques and services by engaging in ongoing research and development, all while upholding the utmost quality standards. Our aim is to deliver unparalleled service to our clients, ensuring that every task is executed with precision and efficiency.



Figure 13: About Us

Contact Form

Email Address:

Message:

Submit

Contact Information

Our Locations

Kathmandu, Nepal
Bhaktapur, Nepal

Contact Us

+977 9821092367, +977 9856719923
+977 1 4456278, +977 1 4467234

Email Us

@info.digitaldurbar.com.np

 **Free Shipping**
Free Shipping All Over Nepal

 **100% Genuine**
100% Genuine Products

 **Easy Return Policy**
3 Days Easy Return Policy

 **Secure Payment**
Shop Online Without Hesitation

Digital Durbar is your

Quick Links

Coming Soon

About

We are committed to providing

Sign In

Please sign in to access your

Figure 14: About Us

The screenshot shows a web-based shopping cart interface. At the top left is the Digital India logo. The top navigation bar includes links for Home and About Us, and a search bar labeled "Search Products...". A large banner for "DELL ALIENWARE LAPTOPS" is displayed, featuring the text "Designed to push your game to the limit.", the Alienware logo, and the Windows 11 logo.

Cart Items

		Qty:	Rs	Action	Action
	Qty: 1	70000	Delete	PlaceOrder	
	Qty: 1	120000	Delete	PlaceOrder	
	Qty: 1	130000	Delete	PlaceOrder	
	Qty: 1	210000	Delete	PlaceOrder	

Our Products

	Acer Aspire 3 A315-24P-R7VH Slim Laptop
	2023 Newest Lenovo ThinkPad X1 Carbon Gen 10
	Apple iMac Display

Figure 15: Cart.

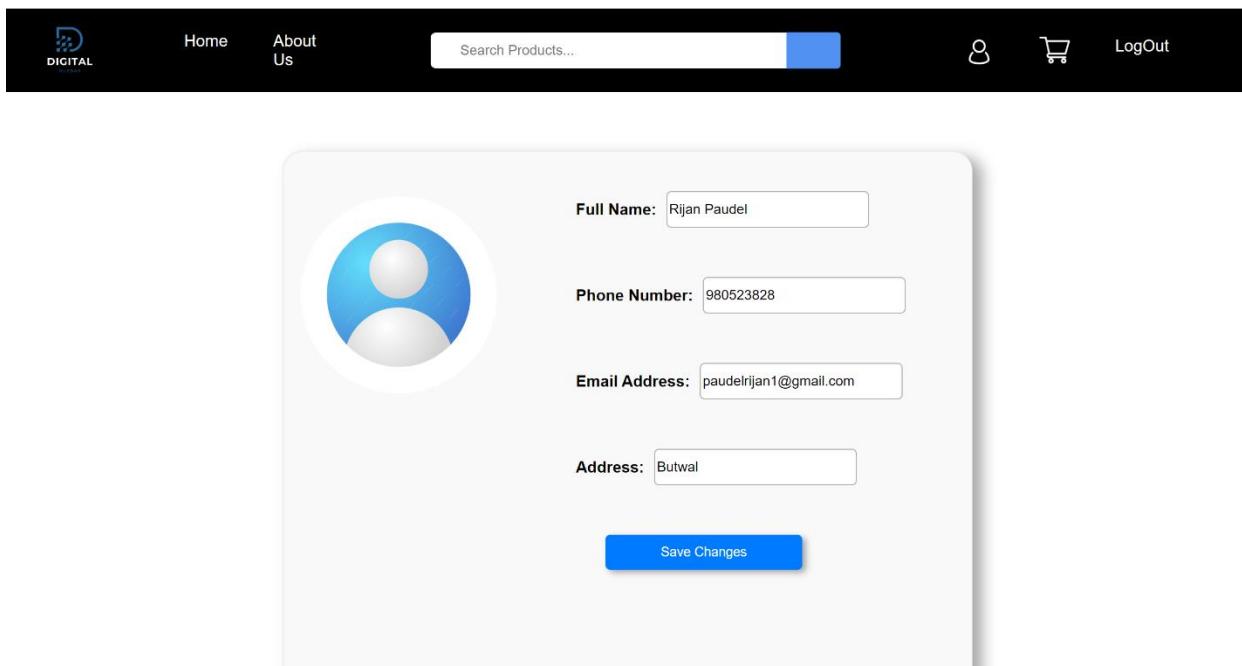


Figure 16: Profile.

Order Details

ID	Product Name	Order Date	Total Price	Order Status	QTY
5	MacBook Pro 14	2024-05-03	Rs 120000	SHIP	1
6	Acer Aspire	2024-05-03	Rs 1500	SHIP	1
7	Pixel Laptop	2024-05-03	Rs 421	SHIP	1
8	MacBook Pro 14	2024-05-03	Rs 120000	DECLINE	1
9	MacBook Pro 14	2024-05-03	Rs 120000	DECLINE	1
10	Acer Aspire	2024-05-03	Rs 4210	SHIP	1
11	Pixel Laptop	2024-05-09	Rs 421	SHIP	1
12	Pixel Laptop	2024-05-09	Rs 421	DECLINE	1
13	Pixel Laptop	2024-05-09	Rs 421	DECLINE	1
14	Pixel Laptop	2024-05-09	Rs 421	PENDING	1
15	Pixel Laptop	2024-05-09	Rs 421	PENDING	1

Figure 17 Order Detail.

ADMIN PANEL

Dashboard  LogOut

Add Product

Product Name:

Product Price:

Product Image:

Add Product

Specification of Product

ID	Product Name	Product Price	Product Image	Edit
				

Figure 18: Admin Pane Add section.

ID	Product Name	Product Price	Product Image	Edit
20	Acer Aspire 3 A315-24P-R7	Rs 70000		Edit Delete
21	2023 Newest Lenovo Think	Rs 120000		Edit Delete
22	Apple MacBook Air 13.3	Rs 130000		Edit Delete
				

Figure 19: Admin Midsection.

ID	Product Name	Customer ID	Order Date	Total Price	Qrder Status	QTY	EDIT
14	Pixel Laptop	10	2024-05-09	Rs 421	PENDING	1	<button>Ship</button> <button>Decline</button>
15	Pixel Laptop	10	2024-05-09	Rs 421	PENDING	1	<button>Ship</button> <button>Decline</button>

ID	Product Name	Customer ID	Order Date	Total Price	Qrder Status	QTY
1	Mac	1	2024-04-30	Rs 120000	SHIP	1
3	Acer Aspire	12	2024-05-01	Rs 1500	SHIP	1
4	Acer Aspire	13	2024-05-01	Rs 1500	SHIP	1
5	MacBook Pro 14	10	2024-05-03	Rs 120000	SHIP	1
6	Acer Aspire	10	2024-05-03	Rs 1500	SHIP	1
7	Pixel Laptop	10	2024-05-03	Rs 421	SHIP	1
10	Acer Aspire	10	2024-05-03	Rs 4210	SHIP	1
11	Pixel Laptop	10	2024-05-09	Rs 421	SHIP	1

Figure 20: Admin Order and Ship

3. Class Diagram

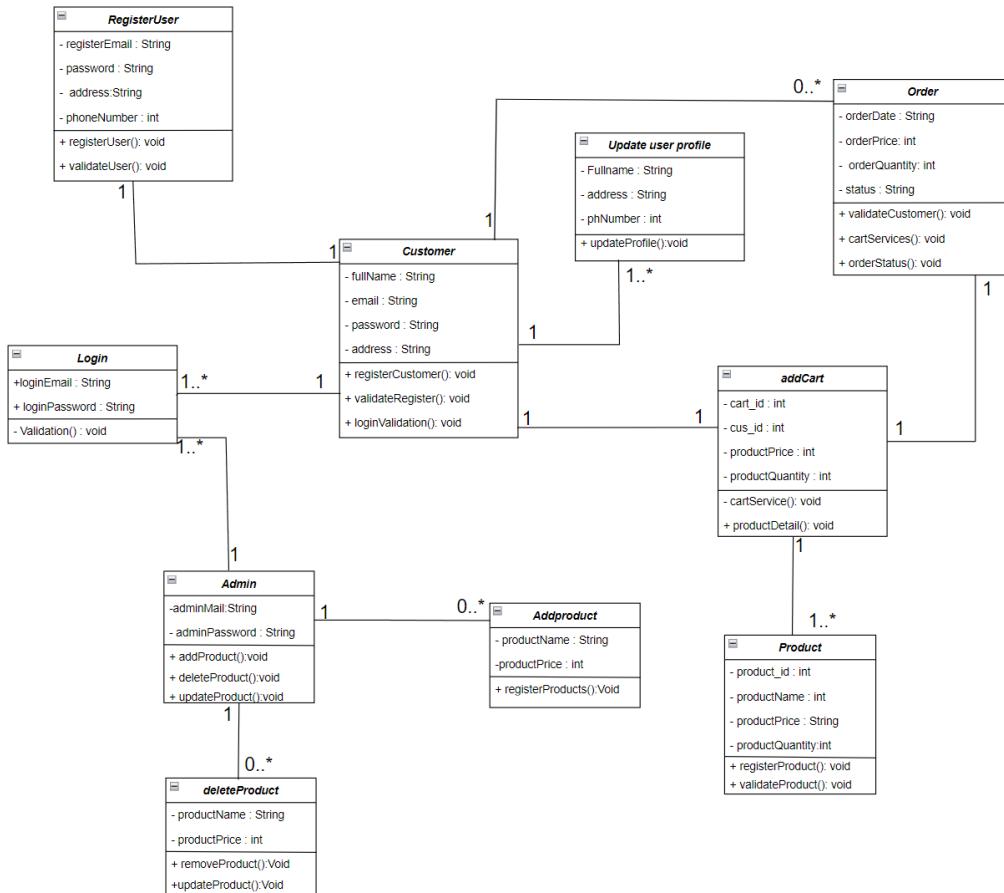


Figure 21: Class Diagram

4. Method Description

4.1. DBConnection Class:

4.1.1. DatabaseConnection

S.N.	Method Name	Method Description
1.	getConnection()	This method establishes the database connection by checking for an existing connection, creating a new connection with the provided database URL, password, and username by loading the JDBC driver.

Table 1: Method Description for DbConnection

4.2. DAO Class

4.2.1. CustomerDao

S.N.	Method Name	Method Description
1.	registerCustomer(Customer customer)	This method registers a new user with required credentials in the Customer Model class by preparing an SQL update query statement and returns true if the operation is success and false if it fails.
2.	Customer login(String email, String password)	This method checks the credentials of the user like email & password while logging into the system by preparing an SQL select query statement of the specified email provided by user.
3.	updateCustomer(Customer customer)	This method updates the user details when the user desires. The SQL update query statement is prepared, and each detail of the Customer Model class is updated.

Table 2: Method Description for CustomerDao

4.3. Servlet

4.3.1. LoginServlet

S.N.	Method Name	Method Description
1.	doPost(HttpServletRequest request, HttpServletResponse response)	The method handles the HTTP POST request for user login. It compares the provided email and password with the Admin and Customer records in the database. Based on the user role, it redirects admins to the admin panel and regular users to the home page.

Table 3: Method Description for LoginServlet

4.3.2. SignupServlet

S.N.	Method Name	Method Description
1.	doPost(HttpServletRequest request, HttpServletResponse response)	This method handles the HTTP request for user registration. It retrieves user credentials (fullname, email, phone number, address, password) from the request, encrypts the password using AES, creates a new Customer instance via CustomerDao, and stores the user data in the database

Table 4: Method Description for SignupServlet

4.3.3. LogoutServlet

S.N.	Method Name	Method Description
1.	doGet(HttpServletRequest request, HttpServletResponse response)	This method handles the HTTP GET request of when the user navigates to Logout option, by removing & killing session attributes and redirecting them to Registration/ Login page.

Table 5: Method Description for LogoutServlet

4.3.4. UpdateCustomerProfileServlet

S.N.	Method Name	Method Description
1.	doPost(HttpServletRequest request, HttpServletResponse response)	This method handles the HTTP POST method of updating the user profile in the database by retrieving the current session user's profile details, calling the CustomerDao's updateCustomer().

Table 6: Method Description for UpdateCustomerProfileServlet

5. Test Cases:

5.1 To test whether the user details get stored in the database after registration.

Test	1
Objective	To test whether the user details get stored in the database after registration
Action	The page register.jsp is called and the user details are entered in the registration form. After clicking on the Submit button, dbConnection, customer java, and customer dao are initiated which registers the user and stores user data in the database.
Expected result	The user registration details will be stored in the database.
Actual result	The user registration details are stored in the database.
Conclusion	The test was successful.

Table 7 Test 1

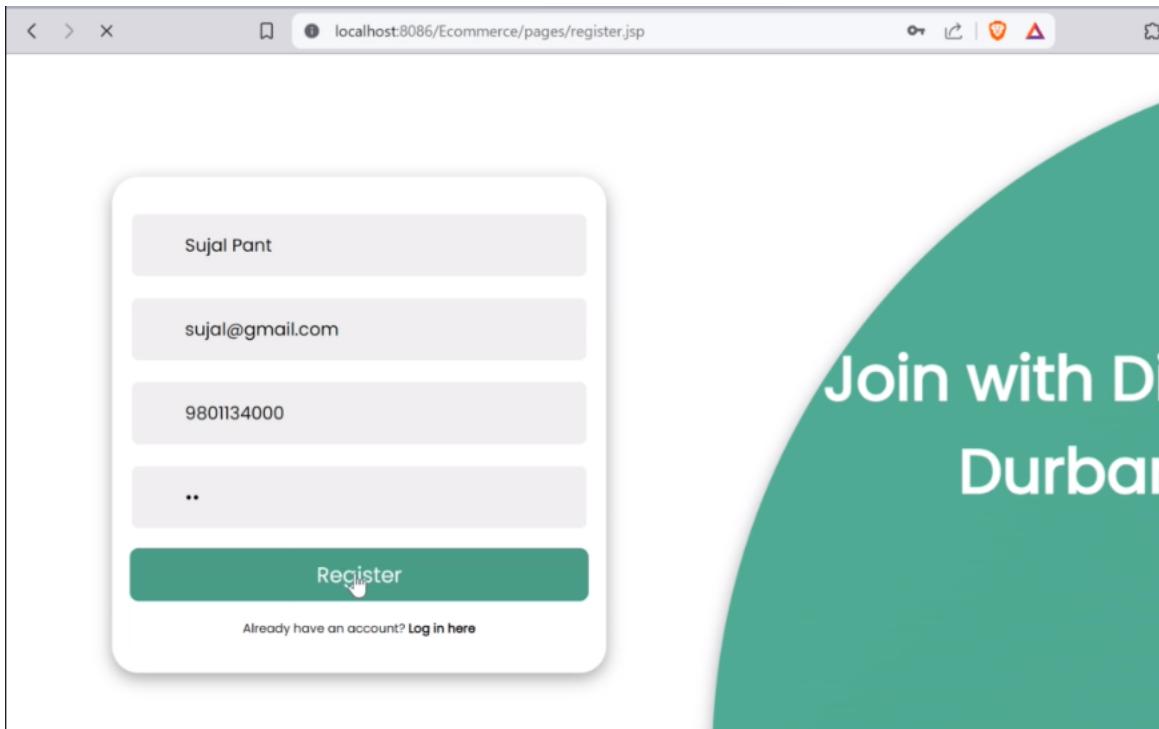


Figure 22: User details .

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	10	Rijan Paudel	paudelrijan1@gmail.com	9805238286	OYAR+buny5xD5w+Y7i0hzflH8K8Dq47e1PnrPumBdZRGf
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	11	Rijan Paudel	paudelrijan12@gmail.com	9805238280	8Y+Y6oJ9xRiX+mEGEiavHFka1J0u9aNM5riVUyZo/TDz'
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	12	Kushal Gurung	kushal@gmail.com	9805238299	RpMZEaUupclHdQhMC2St02yUmdFWzQsq2fQvpMdmN
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	13	Sujal Pant	sujal@gmail.com	9801134000	mfulU3ZQAFmFqST+0IH6wj9VAJu58vTO37TvaENMUCj

Figure 23: User registration data.

5.2 To test whether the password authentication is valid.

Test	2
Objective	To test whether the password authentication is valid.
Action	In the user login page, the password is entered wrong intentionally to check if the error is shown when the password is incorrect or not.
Expected result	The login form will display “Invalid Credentials”.
Actual result	The login form displays “Invalid Credentials”.
Conclusion	The test was successful.

Table 8 Test 2

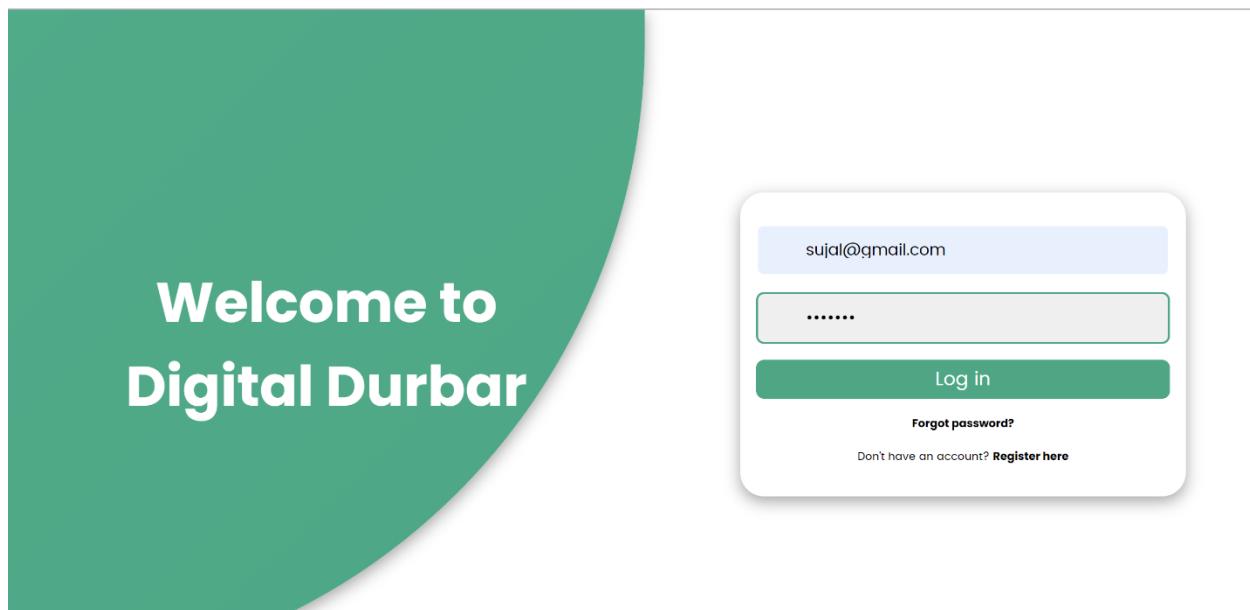


Figure 24: Entering wrong password in login.

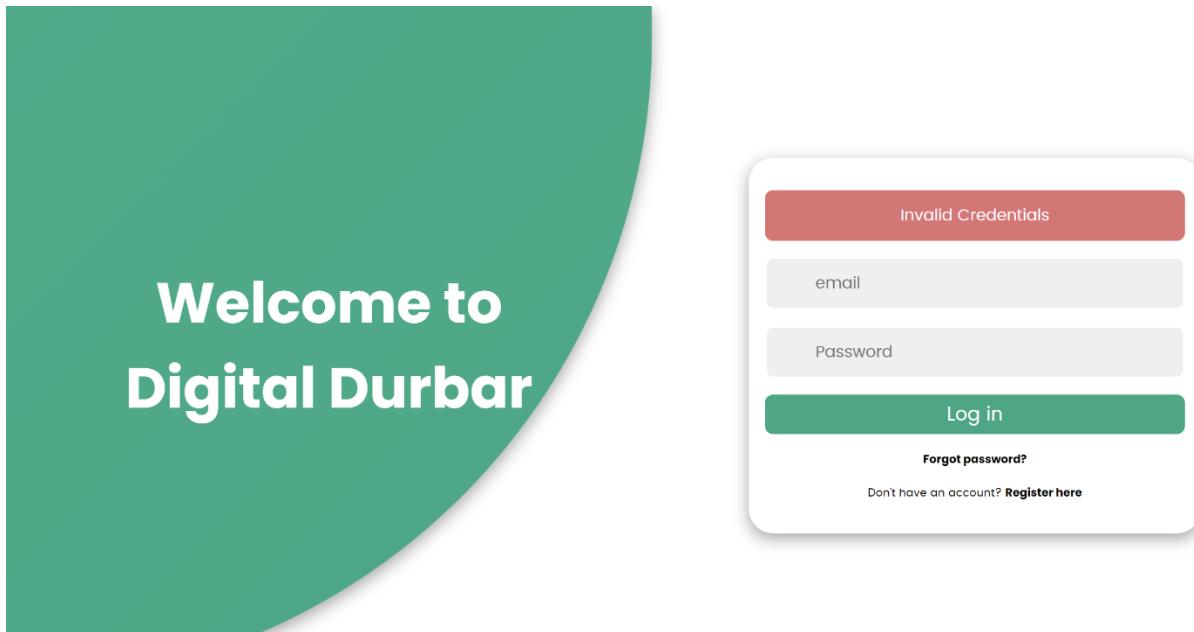
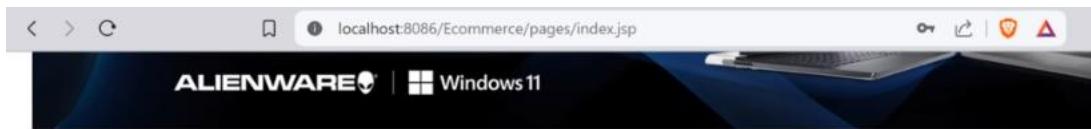


Figure 25: inserting the wrong password.

5.3 To test whether the order details are stored in the database after placing an order.

Test	3
Objective	To test whether the order details are stored in the database after placing an order.
Action	After the product is added to the cart, the “Place Order” button is clicked dbConnection, placeOrder java, and order dao are initiated which places the order, and the details are then stored in the database.
Expected result	The order details will be stored in the database.
Actual result	The order details are stored in the database.
Conclusion	The test was successful.

Table 9 Test 3



Our Products

MacBook Pro 14	Pixel Laptop	Acer Aspire
Rs 120000	Rs 421	Rs 1500
Add to Cart	Add to Cart	Add to Cart

Figure 26: Add to cart.

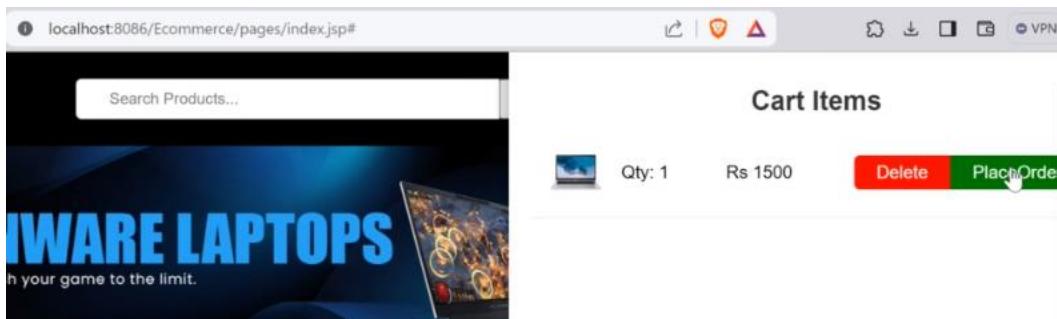


Figure 27: placing order

ID	productName	productId	customerId	orderDate	totalPrice	orderStatus	quantity
1	Mac	1	1	2024-04-30	120000	SHIP	1
4	Acer Aspire	7	13	2024-05-01	1500	SHIP	1

Figure 28: Order details in database.

5.4 To test whether the admin can add new products.

Test Number	4
Objective	To test whether the admin can add new products.
Action	Admin credentials are input in the login form which takes us to the admin panel, the admin enters the product details in the add product section which then can be viewed on the products page.
Expected result	New products with product details will be added.
Actual result	New products with product details are added.
Conclusion	The test was successful.

Table 10 Test 4

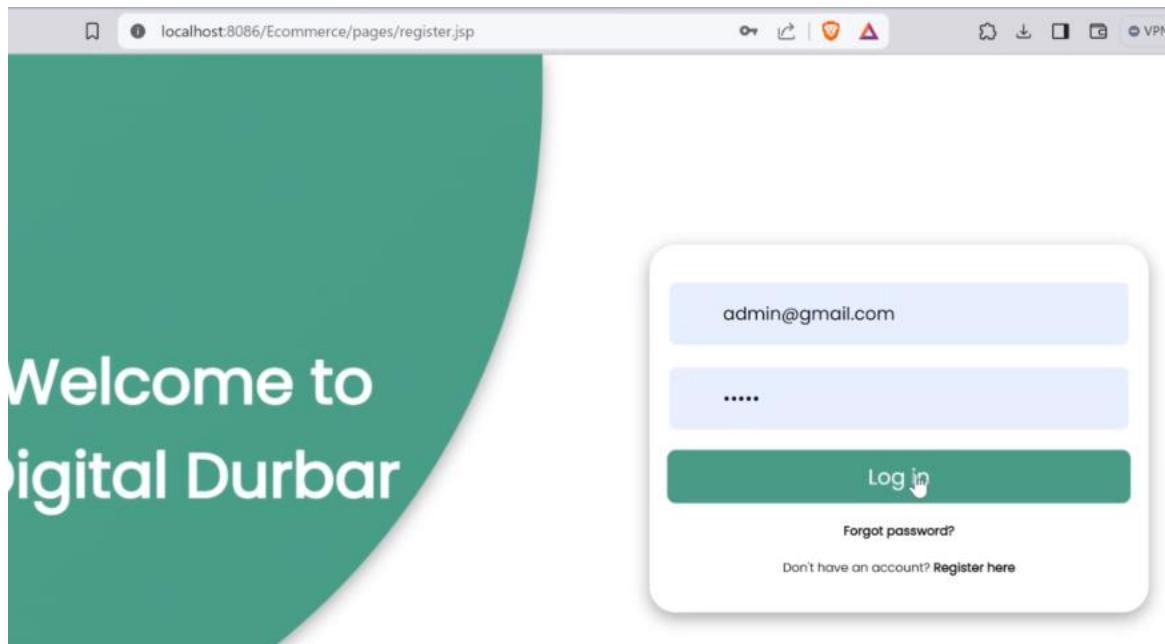


Figure 29: Admin login.

ADMIN PANEL

Dashboard

Add Product

Product Name:
Legion

Product Price:
1200

Product Image:
 Apple 2023 MacBook Pro Laptop M3 Pro chip.jpg

Add Product

Figure 30: Adding new product.

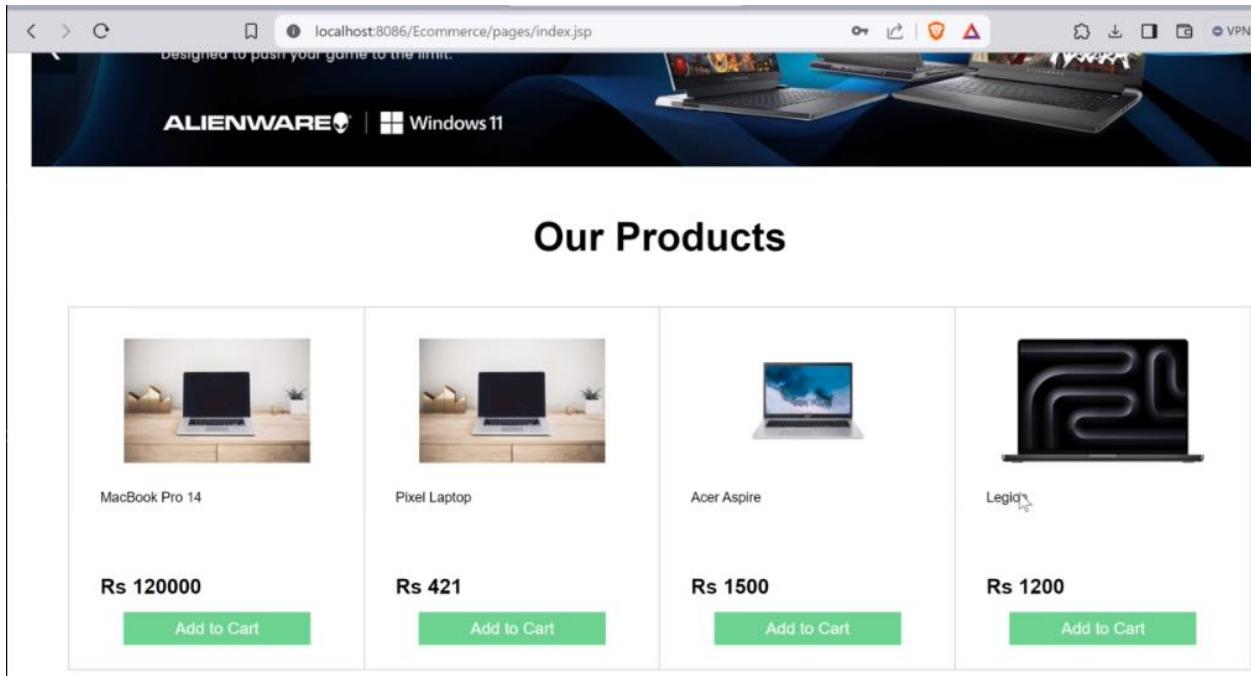


Figure 31: product added.

5.5 To test whether the search filter works or not.

Test	5
Objective	To test whether the search filter works or not.
Action	The product name is entered in the search bar, search button is clicked which then displays the product that we searched for.
Expected result	The product that we searched for will be filtered out and displayed.
Actual result	The product that we searched for is filtered out and displayed.
Conclusion	The test was successful.

Table 11 Test 5



Our Products



Figure 32: Search function.

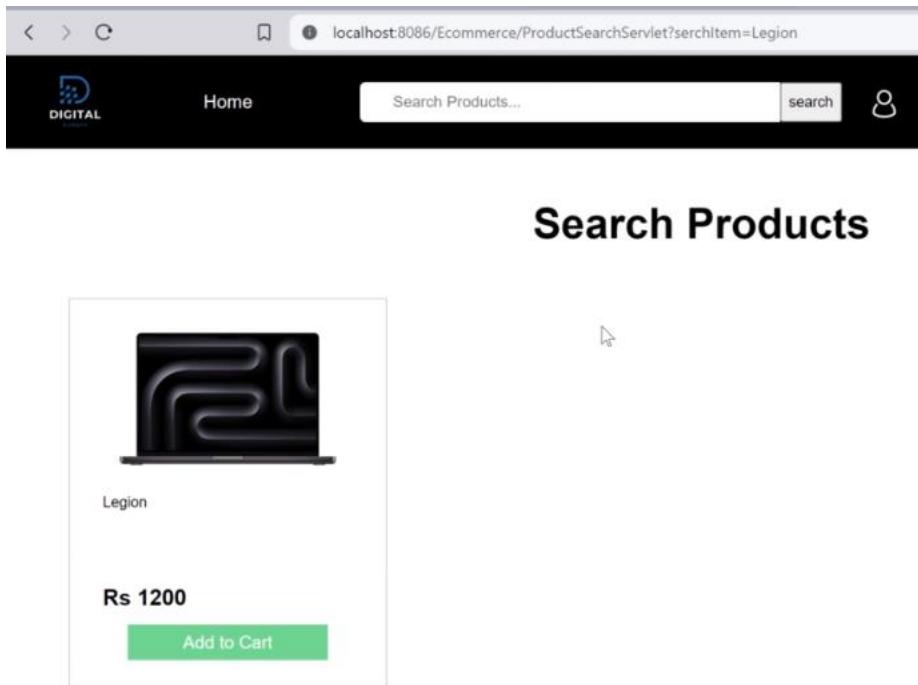


Figure 33: Display product.

5.6 To test whether the user can view their history of ordered items.

Test	6
Objective	To test whether the user can view their history of ordered items or not.
Action	After placing an order, the ordered item details are stored in the database and the user profile section.
Expected result	The user will be able to view their history of ordered items.
Actual result	The user can view their history of ordered items.
Conclusion	The test was successful.

Table 12 Test 6

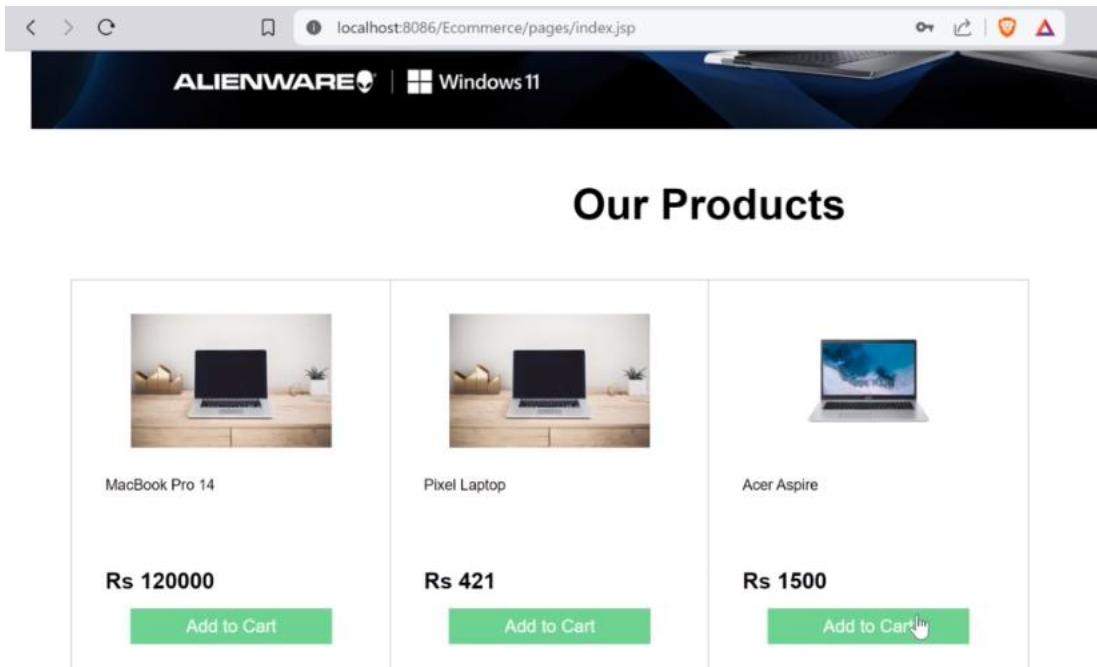


Figure 34: add cart.

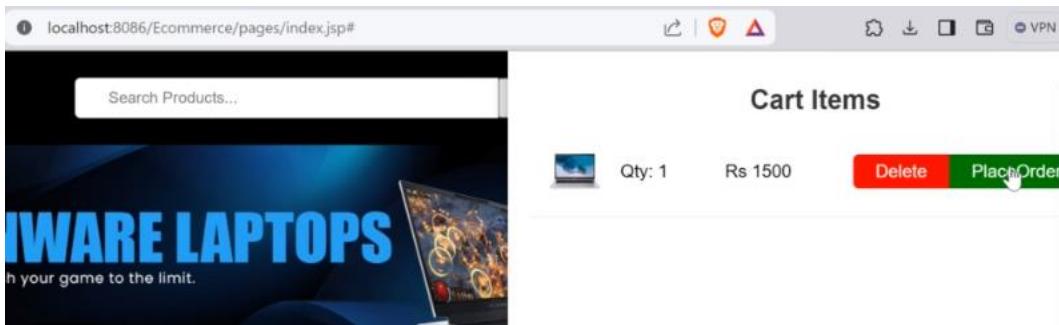


Figure 35: order place.

Order Details

ID	Product Name	Order Date	Total Price	Order Status	QTY
4	Acer Aspire	2024-05-01	Rs 1500	PENDING	1

Figure 36: Order history in profile.

5.7 To test whether the user can log in after registering.

Test	7
Objective	To test whether the user can log in after registering or not
Action	After registering in the system, the user logs in the system by entering their login credentials in the login form.
Expected result	The user will be able to log in and be redirected to the index page.
Actual result	The user can log in and get redirected to the index page
Conclusion	The test was successful.

Table 13 Test 7

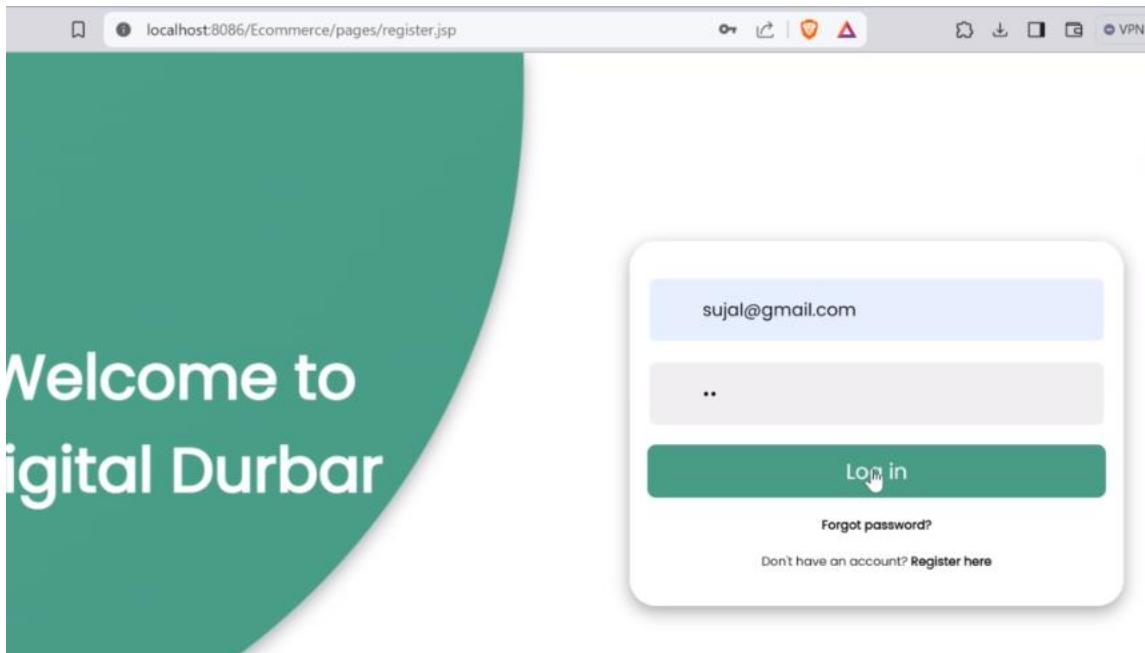


Figure 37: enter correct data.



Figure 38: home page.

5.8 To test whether the admin can delete an existing product.

Test	8
Objective	To test whether the admin can delete an existing product.
Action	After logging in as the admin, the admin panel is displayed where the admin clicks on the delete product button which deletes the product from the system.
Expected result	The product will be deleted when we view the product index page.
Actual result	The product is deleted when we view the product index page.
Conclusion	The test was successful.

Table 14: Test 8

Pending Order Details

ID	Product Name	Customer ID	Order Date	Total Price	Qrder Status	QTY	EDIT
----	--------------	-------------	------------	-------------	--------------	-----	------

Ship Order Details

Figure 39: Admin panel.



Our Products

	
MacBook Pro 14	Pixel Laptop
Rs 120000	Rs 421
Add to Cart	Add to Cart

Figure 40: Deleting product.

5.9 To test whether the user can access the cart page while they are not logged in.

Test	9
Objective	To test whether the user can access the cart page while they are not logged in
Action	The user tries to click on the cart icon while not logged in. A “Please Login” message is displayed.
Expected result	A message telling the user to log in will be displayed.
Actual result	A message telling the user to log in is displayed.
Conclusion	The test was successful.

Table 15: Test 9



Figure 41: Home page without Login

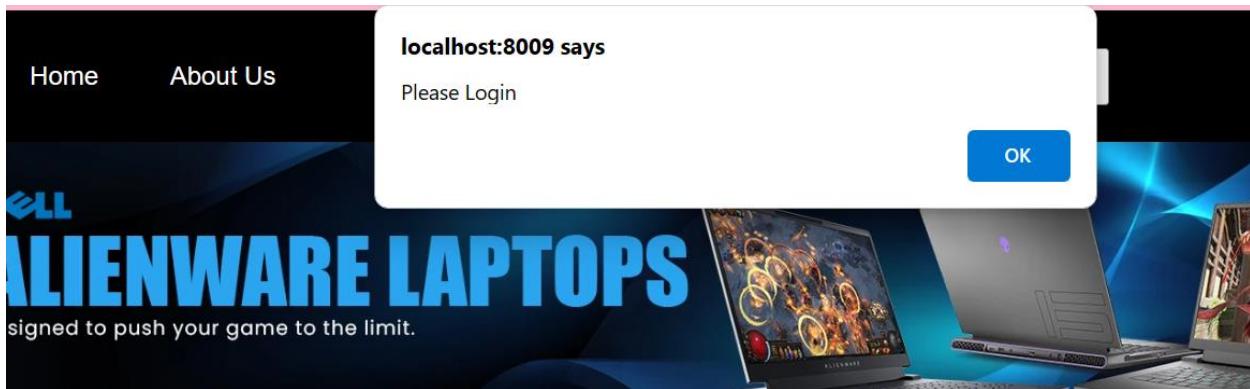


Figure 42: Please Login.

6. Tools and Libraries Used:

6.1. Tools used:

1. Eclipse IDE:

Eclipse is a free Java-based development platform that is well-known for its plugin ecosystem, which enables programmers to create and test code in additional languages. (Hanna, 2023)



Figure 43: Eclipse.

2. Apache Tomcat:

The Apache Software Foundation is the developer of Apache Tomcat, an open-source web server and servlet container. To run Java Servlets, Java Server Pages (JSP), and other Java web components, it offers a Java-based environment. (tomcat.apache, 2024)



Figure 44: Tomcat

3. MySQL:

The relational database management system MySQL is well-known for being flexible and open source. It provides speed, scalability, dependability, and user-friendliness for applications of all sizes. (w3school, 2024)

4. Balsamiq:

A tool for user interface design that may be used to create wireframes is Balsamiq Wireframes (also termed mock-ups or low-fidelity prototypes). (balsamiq, 2024)



Figure 45: balsamiq..

6.2. Techniques Used:

The techniques that are used in the completion of building the system are:

1. Java Servlet:

A web application using servlet technology was developed. Servlets, which are Java classes deployed on web servers, enable building dynamic web pages and interactive applications by extending the server's ability to handle different HTTP requests. Servlets offer an API with classes and interfaces in the javax.servlet package, allowing for server-side processing and generation of dynamic web content. (javatpoint, 2024).

In the coursework, Various Java Servlets are used to make control and launch the post and get method.

```

1 package com.servlet;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 import com.model.Customer;
14
15 import Dao.CustomerDao;
16
17 /**
18 * Servlet implementation class UpdateCustomerProfileServlet
19 */
20 @WebServlet(asyncSupported = true, urlPatterns = { "/UpdateCustomerProfileServlet" })
21 public class UpdateCustomerProfileServlet extends HttpServlet {
22     private static final long serialVersionUID = 1L;
23
24     /**
25      * @see HttpServlet#HttpServlet()
26      */
27     public UpdateCustomerProfileServlet() {
28         super();
29         // TODO Auto-generated constructor stub
30     }
31
32     /**
33      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
34      */
35     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
36         // TODO Auto-generated method stub
37         response.setContentType("text/html");
38         PrintWriter out = response.getWriter();
39
40         String fullName = request.getParameter("fullName");
41         String phoneNumber = request.getParameter("phoneNumber");
42         String address = request.getParameter("address");
43
44         System.out.println(address);
45
46         // get user from session
47         HttpSession session = request.getSession();
48         Customer customer = (Customer) session.getAttribute("currentCustomer");
49
50         //customer.setAddress(address);
51
52         CustomerDao customerDao = new CustomerDao();
53         customer.setFullName(fullName);
54         customer.setPhoneNumber(phoneNumber);
55         customer.setAddress(address);
56         if (customerDao.updateCustomer(customer)) {
57             out.print("Customer profile updated successfully!");
58         } else {
59             out.print("Failed to update customer profile.");
60         }
61     }
62 }

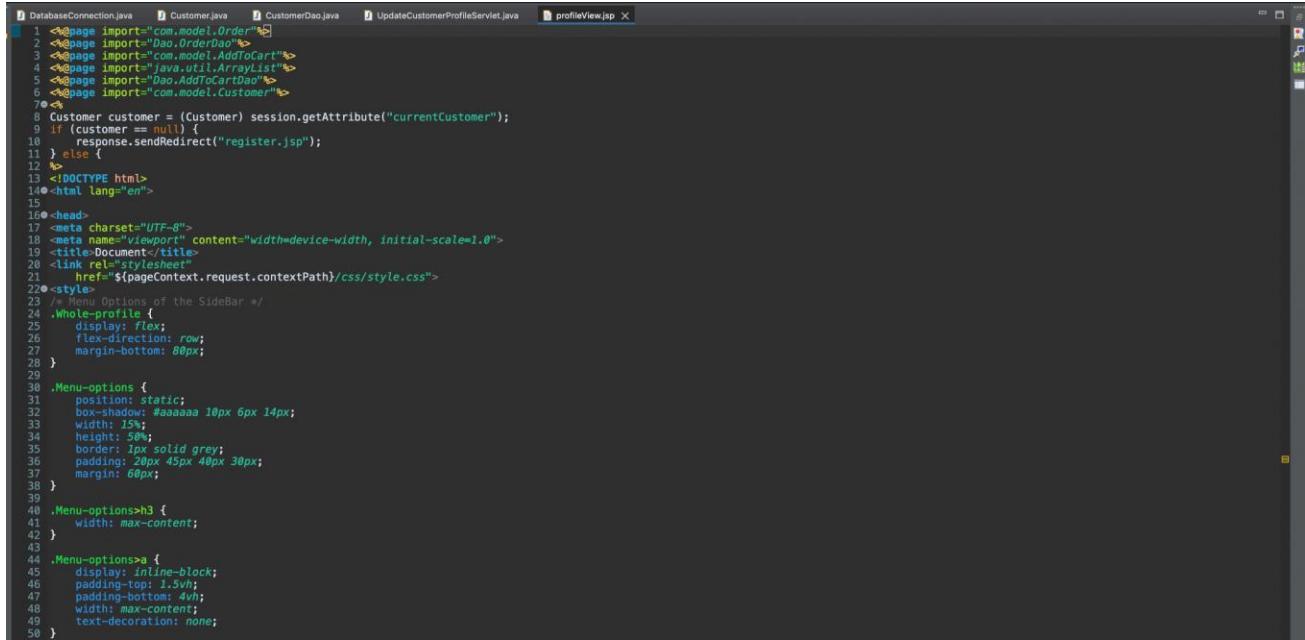
```

Figure 46: Java Servlet Class

2. Java Server Page

In Java, JSP stands for Java Server Pages. It's a server-side technology used in web application development. Web content that is dynamic is produced using it. HTML and JSP tags are both included in JSP. This involves inserting Java code into HTML pages using JSP elements.

In our coursework Jsp files are used to construct the user interface as well as certain functionality that improves and comforts the user experience.



```

1 <%@page import="com.model.Order" %>
2 <%@page import="Dao.OrderDao" %>
3 <%@page import="com.model.AddToCart" %>
4 <%@page import="java.util.ArrayList" %>
5 <%@page import="Dao.AddToCartDao" %>
6 <%@page import="com.model.Customer" %>
7 <%
8 Customer customer = (Customer) session.getAttribute("currentCustomer");
9 if (customer == null) {
10     response.sendRedirect("register.jsp");
11 } else {
12 }
13 <!DOCTYPE html>
14 <html lang="en">
15
16 <head>
17 <meta charset="UTF-8">
18 <meta name="viewport" content="width=device-width, initial-scale=1.0">
19 <title>Document</title>
20 <link rel="stylesheet" href="${pageContext.request.contextPath}/css/style.css">
21 </style>
22 /* Menu Options of the SideBar */
23 .Whole-profile {
24     display: flex;
25     flex-direction: row;
26     margin-bottom: 80px;
27 }
28
29 .Menu-options {
30     position: static;
31     box-shadow: #aaaaaa 10px 6px 14px;
32     width: 15%;
33     height: 50%;
34     border: 1px solid grey;
35     padding: 20px 45px 40px 30px;
36     margin: 60px;
37 }
38
39 .Menu-options>h3 {
40     width: max-content;
41 }
42
43 .Menu-options>a {
44     display: inline-block;
45     padding-top: 1.5vh;
46     padding-bottom: 4vh;
47     width: max-content;
48     text-decoration: none;
49 }
50 }

```

Figure 47:JSP

3. DAO Classes

Data Access Object is referred to as DAO. It's a design pattern used to isolate the application's remaining components from the data source's access, which is usually a database. Model-View-Controller (MVC) architecture places DAOs in the Model layer.

In our coursework we have constructed various DAO classes to manage the interaction with the database.

```
1 DatabaseConnection.java 2 Customer.java 3 CustomerDao.java X 4 UpdateCustomerProfileServlet.java
5 package Dao;
6
7 import java.sql.*;
8
9 public class CustomerDao {
10     private static Connection con = DatabaseConnection.getConnection();
11
12     public boolean registerCustomer(Customer customer) {
13         boolean flag = false;
14         try {
15             String query = "INSERT INTO Customer(fullName, email, phoneNumber, password, address) VALUES(?,?,?,?,?)";
16             PreparedStatement st = CustomerDao.con.prepareStatement(query);
17
18             st.setString(1, customer.getFullName());
19             st.setString(2, customer.getEmail());
20             st.setString(3, customer.getPhoneNumber());
21             st.setString(4, customer.getPassword());
22             st.setString(5, customer.getAddress());
23
24             st.executeUpdate();
25             flag = true;
26         } catch (Exception e) {
27             e.printStackTrace();
28         }
29         return flag;
30     }
31
32     public Customer login(String email, String password) {
33         Customer customer = null;
34         try {
35             String query = "SELECT * FROM customer WHERE email = ?";
36
37             PreparedStatement pst = CustomerDao.con.prepareStatement(query);
38             pst.setString(1, email);
39
40             ResultSet set = pst.executeQuery();
41
42             if (set.next()) {
43                 customer = new Customer(0, query, query, query, query);
44
45                 customer.setId(set.getInt("ID"));
46                 customer.setFullName(set.getString("fullName"));
47                 customer.setEmail(set.getString("email"));
48                 customer.setPhoneNumber(set.getString("phoneNumber"));
49                 customer.setAddress(set.getString("address"));
50
51                 String passwordFromDb = set.getString("password");
52                 String decryptPassword = PasswordEncryptionWithAES.decrypt(passwordFromDb, email);
53
54                 if (decryptPassword != null & set.getString("Email").equals(email))
55                     && decryptPassword.equals(password)) {
56                     return customer;
57                 }
58             }
59         } catch (Exception e) {
60             e.printStackTrace();
61         }
62     }
63 }
```

Figure 48: DAO CLASS

4. Model classes.

In web development, model classes especially those used in the Model-View-Controller (MVC) architecture represent an application's data and business logic.

```
1 DatabaseConnection.java 2 Customer.java X CustomerDao.java 3 UpdateCustomerProfileServlet.java
4 package com.model;
5
6 public class Customer {
7     private int ID;
8     private String fullName;
9     private String email;
10    private String phoneNumber;
11    private String password;
12    private String address;
13
14    // Constructor without customer ID
15    public Customer(String fullName, String email, String phoneNumber, String password, String address) {
16        super();
17        this.fullName = fullName;
18        this.email = email;
19        this.phoneNumber = phoneNumber;
20        this.password = password;
21        this.address = address;
22    }
23
24    // Constructor with customer ID
25    public Customer(int ID, String fullName, String email, String phoneNumber, String password) {
26        super();
27        this.ID = ID;
28        this.fullName = fullName;
29        this.email = email;
30        this.phoneNumber = phoneNumber;
31        this.password = password;
32        this.address = address;
33    }
34
35    public String getAddress() {
36        return address;
37    }
38
39    public void setAddress(String address) {
40        this.address = address;
41    }
42
43    // Getter Setter
44    public int getID() {
45        return ID;
46    }
47
48    public void setID(int ID) {
49        this.ID = ID;
50    }
51
52    public String getFullName() {
53        return fullName;
54    }
55
56    public void setFullName(String fullName) {
57        this.fullName = fullName;
58    }
59}
```

5. MVC:

MVC is an architectural paradigm for software that is frequently used in online application design and development. The brief overview of each component:

1. Model: The data and business logic of the application are represented by the model.
2. View: The View oversees managing user interactions and presenting the data from the Model to the user. It shows the graphical elements, forms, and HTML pages that make up the user interface.
3. Controller: The Controller serves as a go-between for the View and the Model. Through the View, it takes in user input, processes it, and updates the View with the results. Controllers manage user requests, plan the data transfer between the Model and the View, and regulate the operation of the program.

(tutorialspoint, 2024)

In our course work we separate the files as:

1. Model Class: Represents the data and business logic of the application.
2. Servlet Class: Acts as the controller, handling user requests, interacting with the model and deciding which view to render.
3. DAO Class: Handles data access operations, typically interacts with the database. The DAO typically falls under the Model component in the MVC architecture.
4. JSP: Represent the view, responsible for rendering the user interface and presenting data to the client.

6.3. Libraries

1. JDBC

Database management systems can be accessed by Java programs thanks to the API known as the Java Database Connection specification API that enables Java applications to interact with databases, allowing them to perform various operations in database management systems . (javatpoint, 2024)

7. Developmental Process:

7.1. Planning the system:

The first part of the development was to select the kind of e-commerce website, setting up the necessary software to create the dynamic webpage, and looking up reference websites that the design can be based on are the steps in the project planning process.

Our team selected the e-commerce based for laptops and named the project DIGITAL DURBAR.

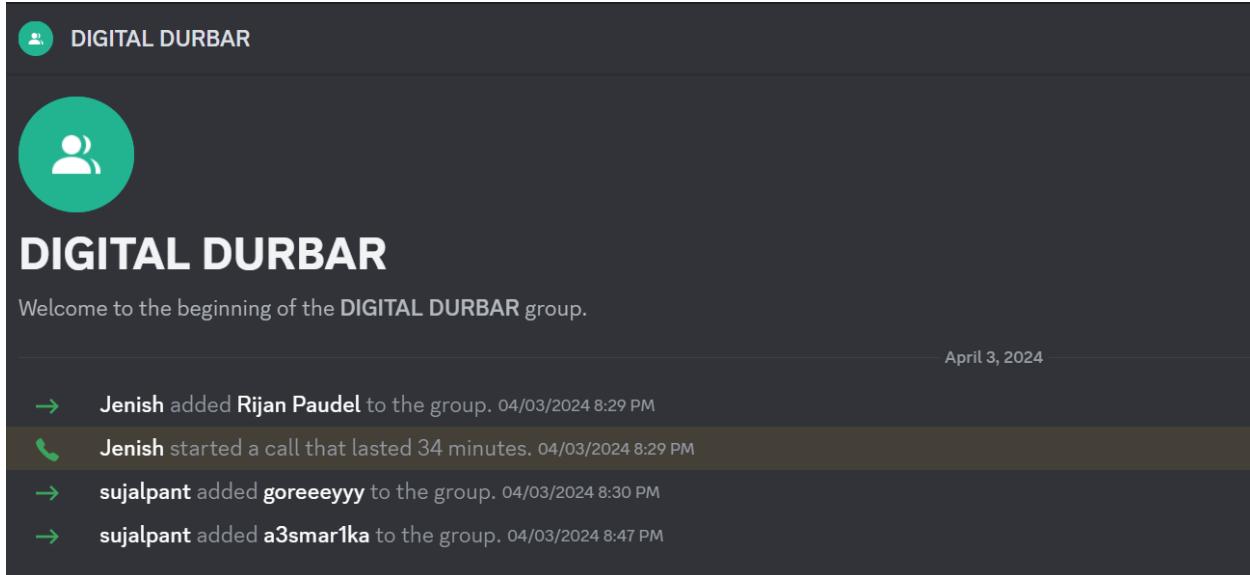


Figure 49: Creation of DIGITAL DURBAR

Here, discord chat was created for easy communication between the teammates and further planning was done via discord channel via calls.

7.2. Design and Frontend Development

Design process involved creating wireframes and prototypes to visualize the website's look and functionality. Extensive research was done on Nepalese laptop

e-commerce sites, and two sites were selected as design references after discussion.

Mudita store was selected for making the top portion of the home page as:

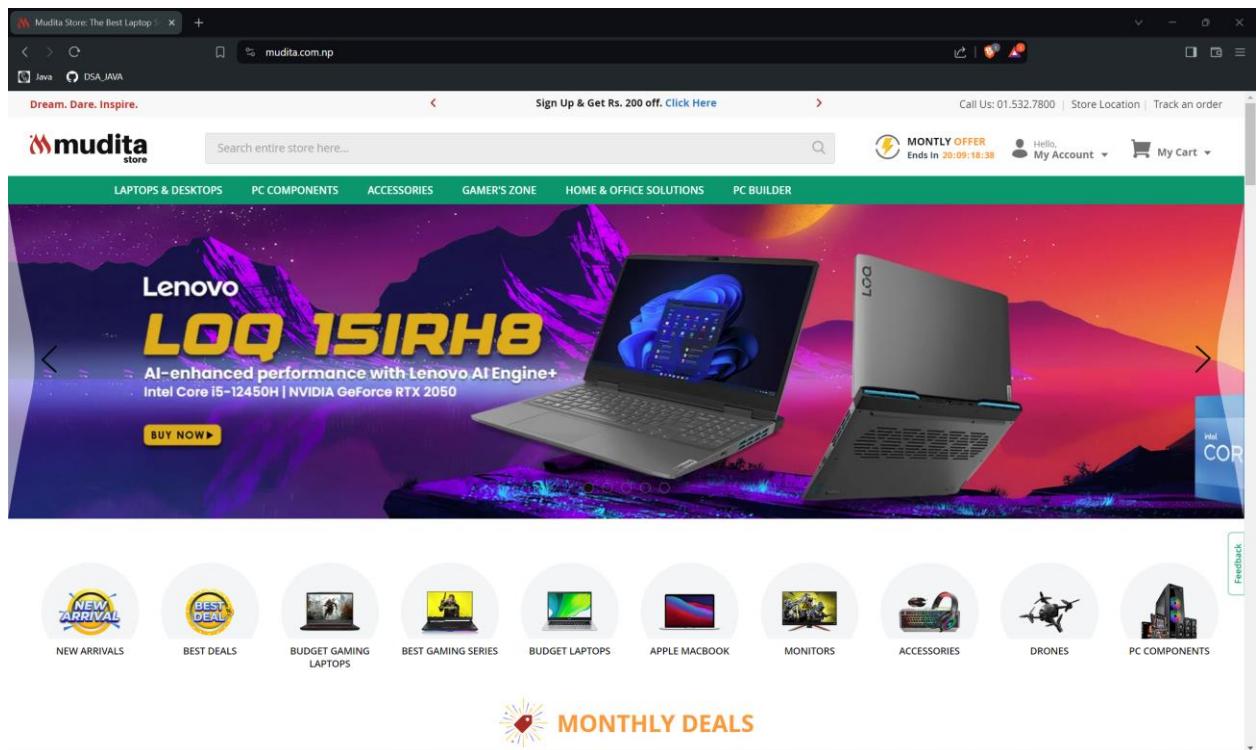


Figure 50: Mudita Store



Our Products

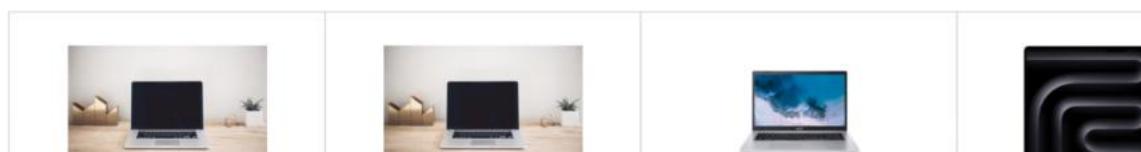


Figure 51: Design reference for home.

Product Alignment and product display idea was taken from BIGBYTE IT WORD PVT LMTD as:

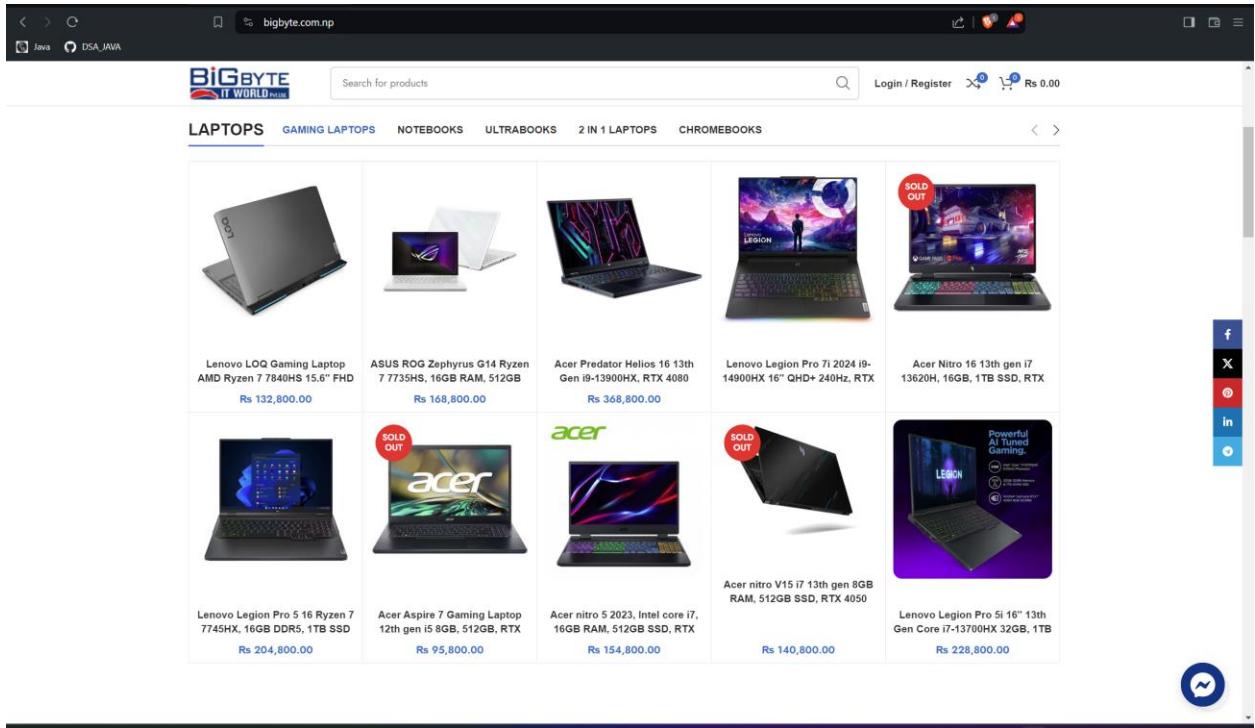


Figure 52: Bigbyte IT WORLD

Laptop	Carbon Gen 10	Display	Laptop with M3 chip
Rs 70000 Add to Cart	Rs 120000 Add to Cart	Rs 130000 Add to Cart	Rs 210000 Add to Cart
HP Notebook Laptop, 15.6" HD Touchscreen	Acer Nitro 17 Gaming Laptop AMD Ryzen 7	Lenovo IdeaPad 1 Student Laptop, Intel Dual Core Processor	Acer Aspire 1 A115-32-C96U Slim Laptop
Rs 100000 Add to Cart	Rs 190000 Add to Cart	Rs 87000 Add to Cart	Rs 140000 Add to Cart

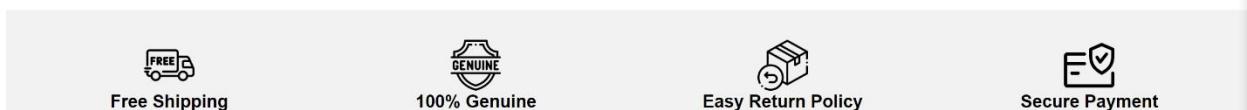


Figure 53: Design reference for product alignment.

The footer section was taken from the Mudita store as:

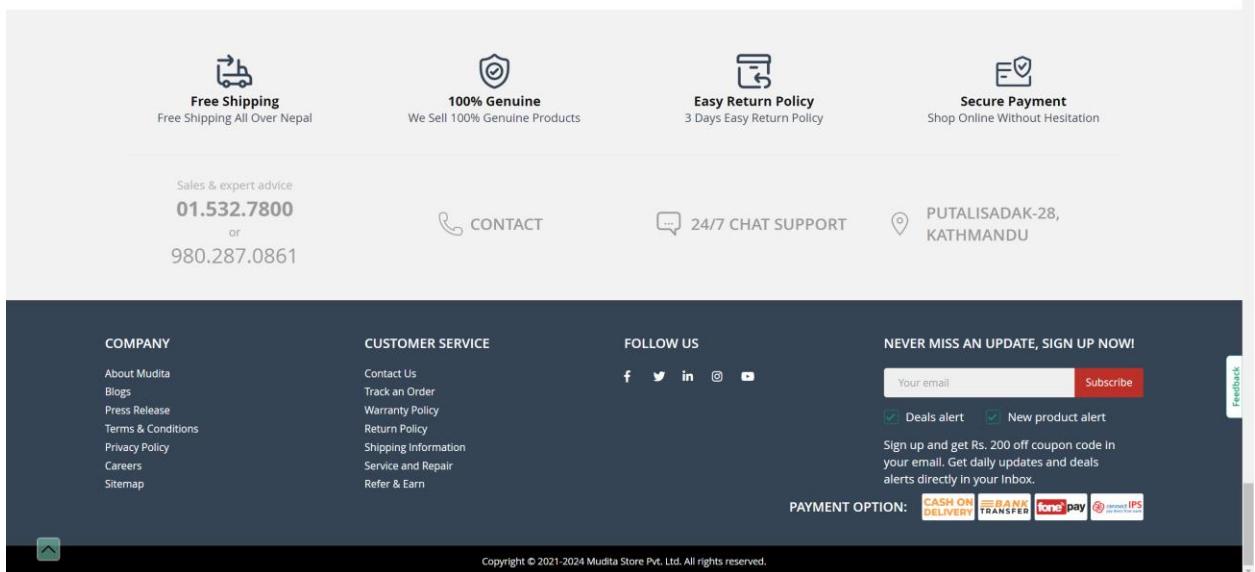


Figure 54: Mudita Footer design.

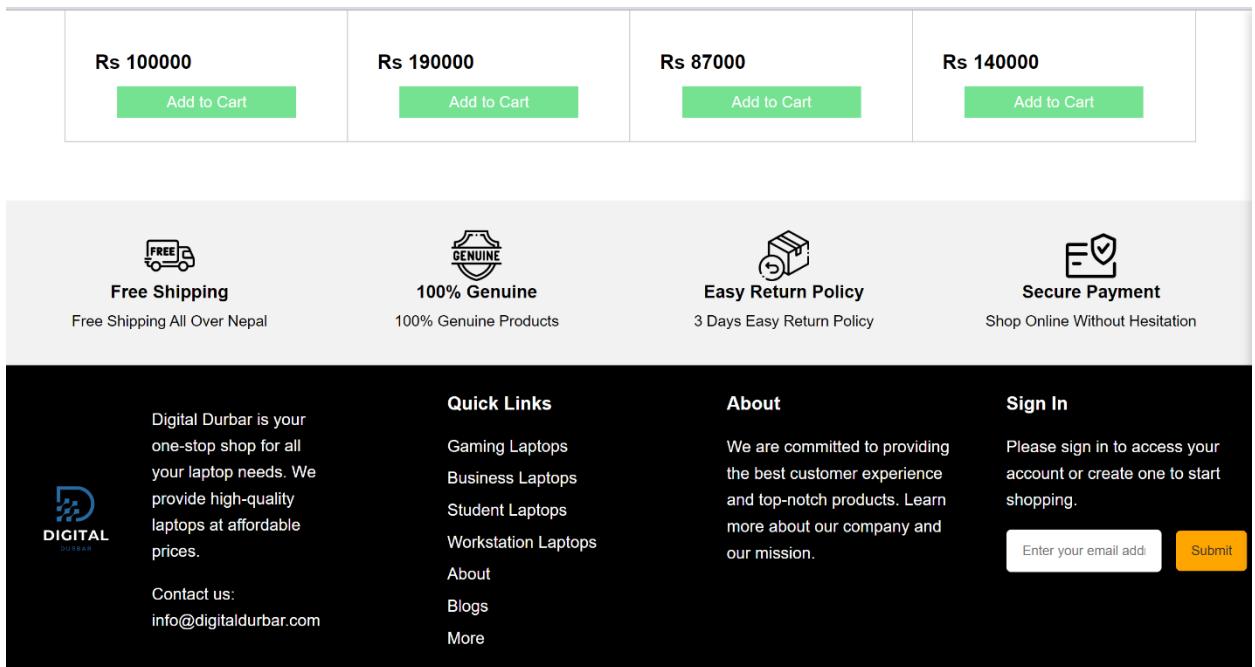


Figure 55: Our footer design.

Wireframe ideas and real development via HTML, CSS, JS languages were on progress and files and ideas was shared on discord as:

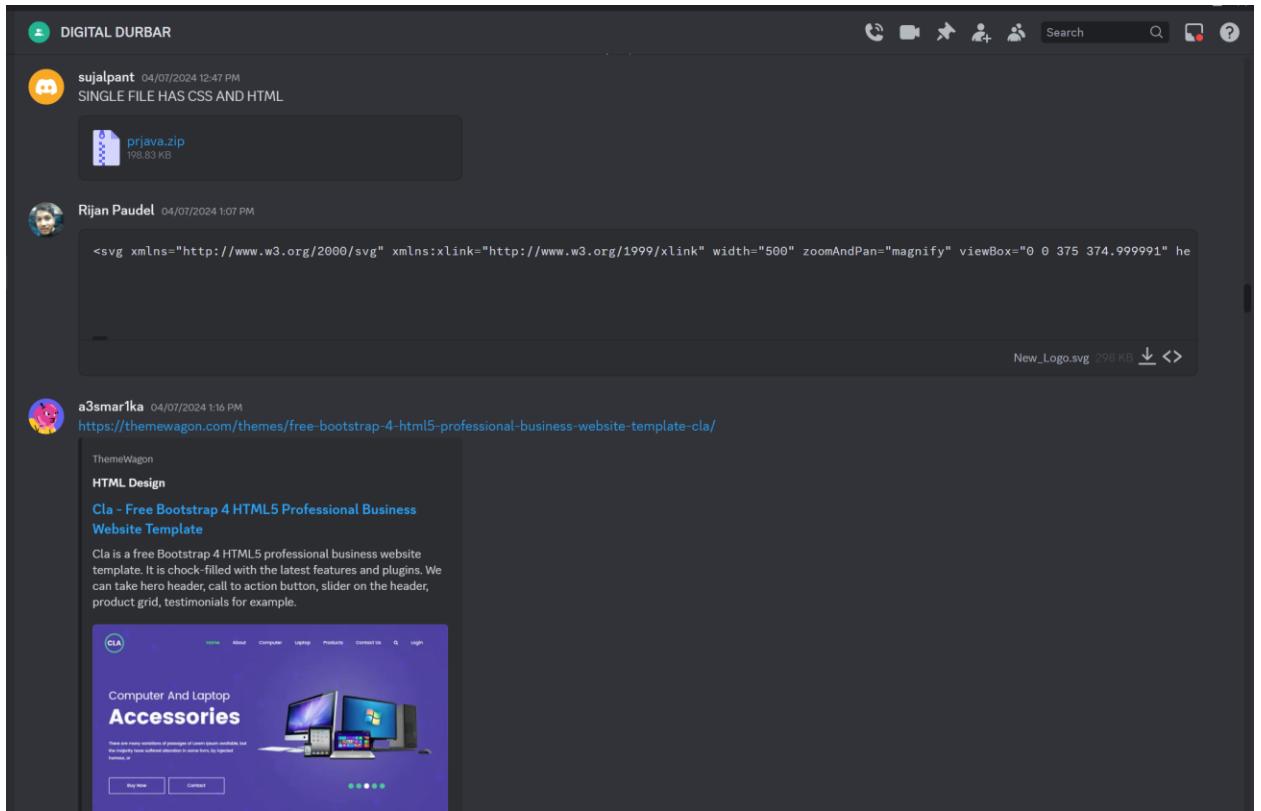


Figure 56: UI discussion and Code sharing.

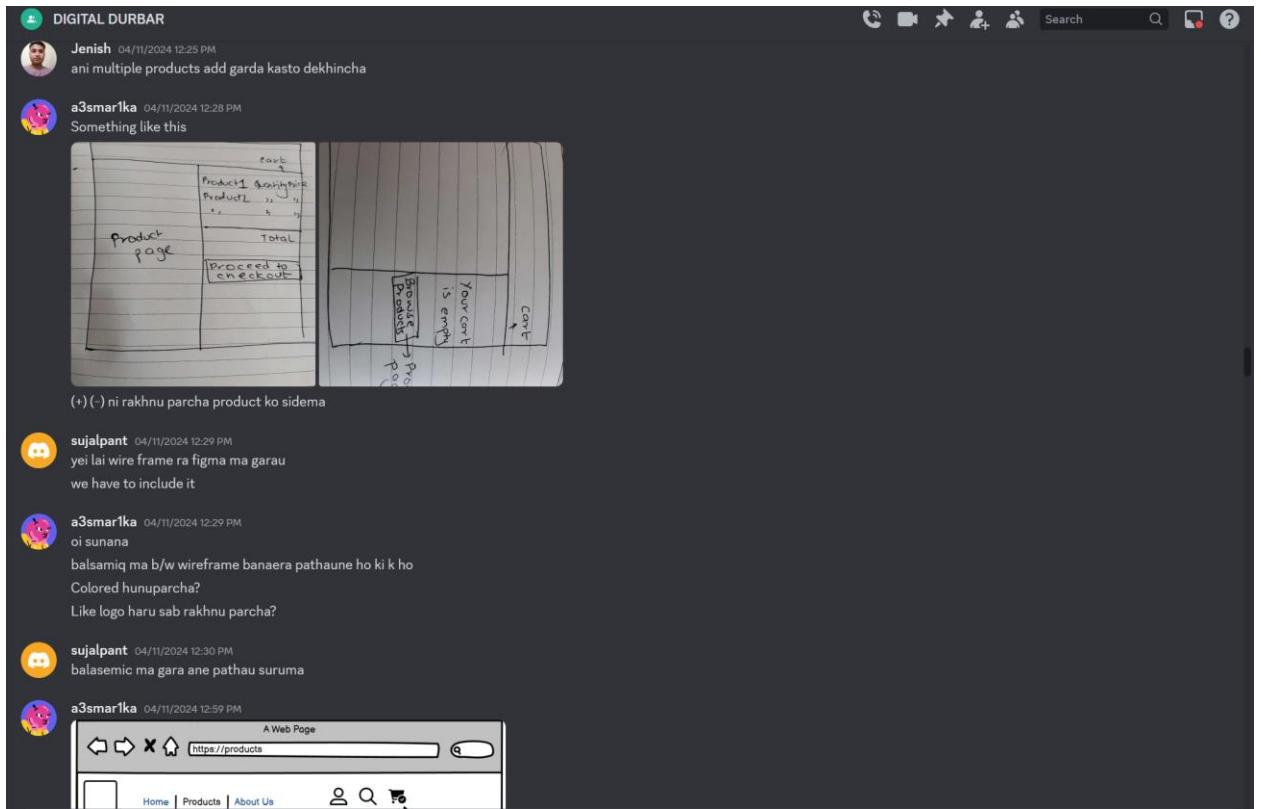
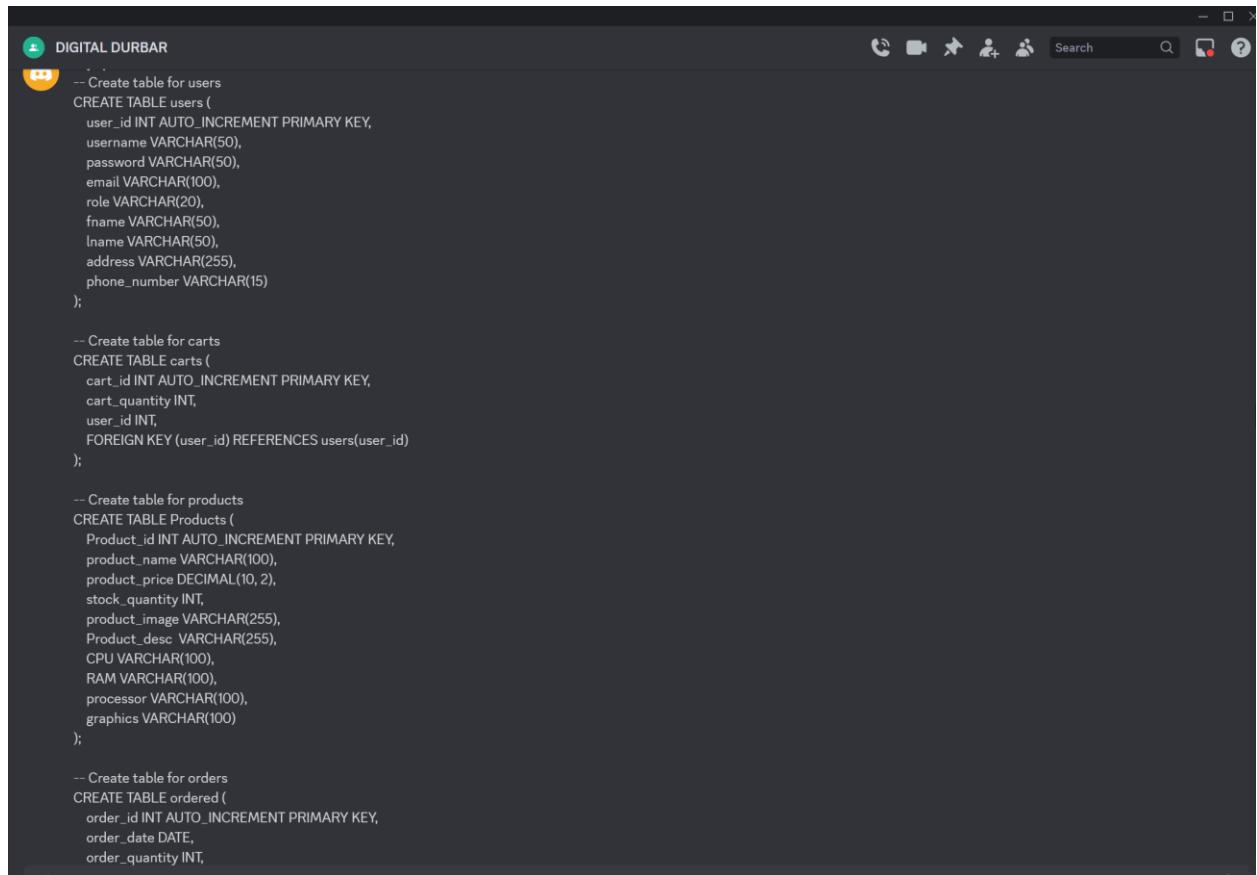


Figure 57: Design Discussion

Other pages such as Sign In , Register , Admin panel and User Profile was fully developed by us.

7.3. Development

The first stage of development was to set up database schema to store user credentials. Various database shema was shared and analysed via discord as:



A screenshot of a Discord message window titled "DIGITAL DURBAR". The message contains four blocks of SQL code for creating tables: users, carts, Products, and ordered. The users table has columns for user_id (primary key, auto-increment), username, password, email, role, fname, lname, address, and phone_number. The carts table has columns for cart_id (primary key, auto-increment), cart_quantity, user_id (foreign key referencing users.user_id), and product_id. The Products table has columns for Product_id (primary key, auto-increment), product_name, product_price (decimal), stock_quantity, product_image, product_desc, CPU, RAM, processor, and graphics. The ordered table has columns for order_id (primary key, auto-increment), order_date, and order_quantity.

```
-- Create table for users
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50),
    password VARCHAR(50),
    email VARCHAR(100),
    role VARCHAR(20),
    fname VARCHAR(50),
    lname VARCHAR(50),
    address VARCHAR(255),
    phone_number VARCHAR(15)
);

-- Create table for carts
CREATE TABLE carts (
    cart_id INT AUTO_INCREMENT PRIMARY KEY,
    cart_quantity INT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Create table for products
CREATE TABLE Products (
    Product_id INT AUTO_INCREMENT PRIMARY KEY,
    product_name VARCHAR(100),
    product_price DECIMAL(10, 2),
    stock_quantity INT,
    product_image VARCHAR(255),
    product_desc VARCHAR(255),
    CPU VARCHAR(100),
    RAM VARCHAR(100),
    processor VARCHAR(100),
    graphics VARCHAR(100)
);

-- Create table for orders
CREATE TABLE ordered (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    order_date DATE,
    order_quantity INT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

Figure 58: Database design code.

The database schema file was shared after verifying the context of the website as:

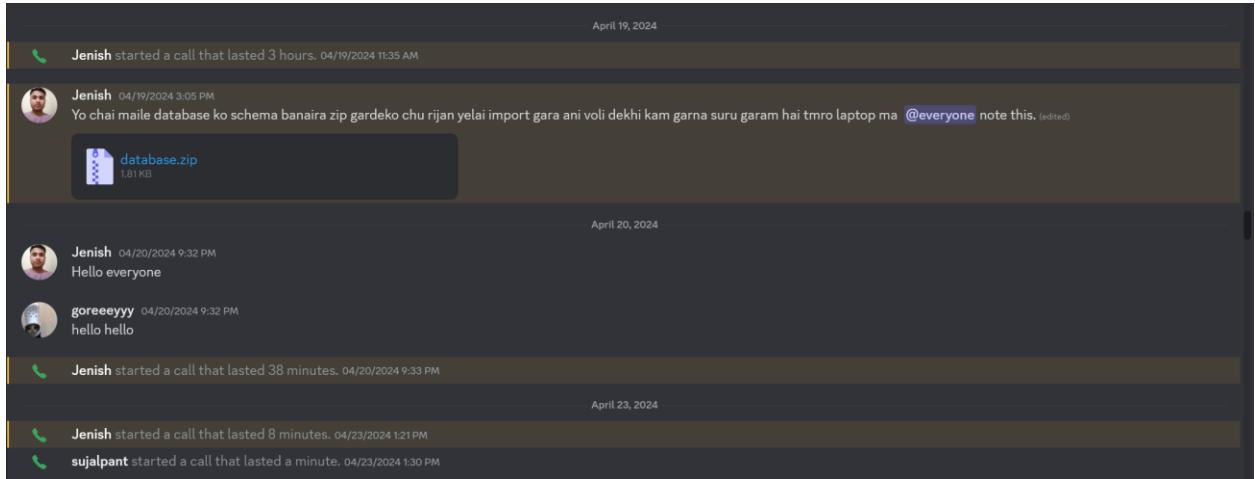


Figure 59: Database schema Shared.

Various Changes were also made in middle for making the UI look better and codes were shared on Discord:

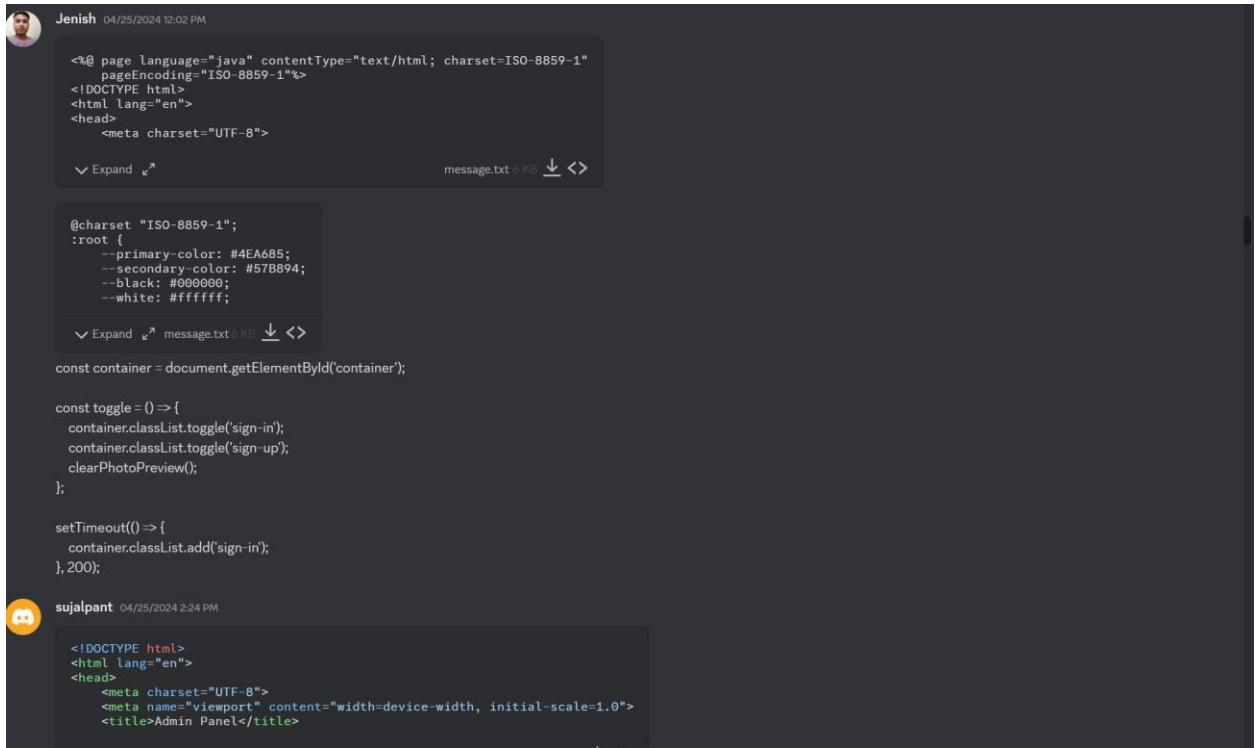


Figure 60: UI modifications.

The project includes servlets for handling user authentication (LoginServlet and SignupServlet) and customer profile updates (UpdateCustomerProfileServlet). It also includes servlets for managing products (AddProductServlet, DeleteProductServlet, ProductSearchServlet, and EditProductServlet) and orders (PlaceOrder, ChangeOrderStatusServlet, EditProductServlet, and DeleteProductServlet).

The CustomerDao class handles user authentication and profile updates, while the OrderDao, AddToCartDao, and ProductDao classes interact with the database for product and order operations.

Additionally, the PasswordEncryptionWithAes class provides methods for encrypting and decrypting passwords using the AES algorithm with PBKDF2 key generation.

7.4. Testing

After completion, functionality testing was done. We tested redirecting to the login form when adding items to cart without logging in, registering/logging in takes user to home page to search, add to cart and sort items. Admin functionalities like updating, deleting, and adding products were tested. Incorrect username/password authentication and database updates for registered details were also verified.

7.5. Maintenances:

After the testing phase, the bugs which occurred while testing phase were fixed in this phase to have a proper and stable website where the user can have a comforting experience exploring the website.

8. Critical Analysis

1. A major challenge faced during this project was integrating code from different development environments due to varying configurations and dependencies across team members' laptops. To resolve this, a single laptop was used for the entire project development, ensuring a consistent setup. However, this experience highlighted the importance of proper version control practices and collaborative coding tools like Git/GitHub for future projects to effectively manage code changes and maintain a clear version history while allowing individual development environments.

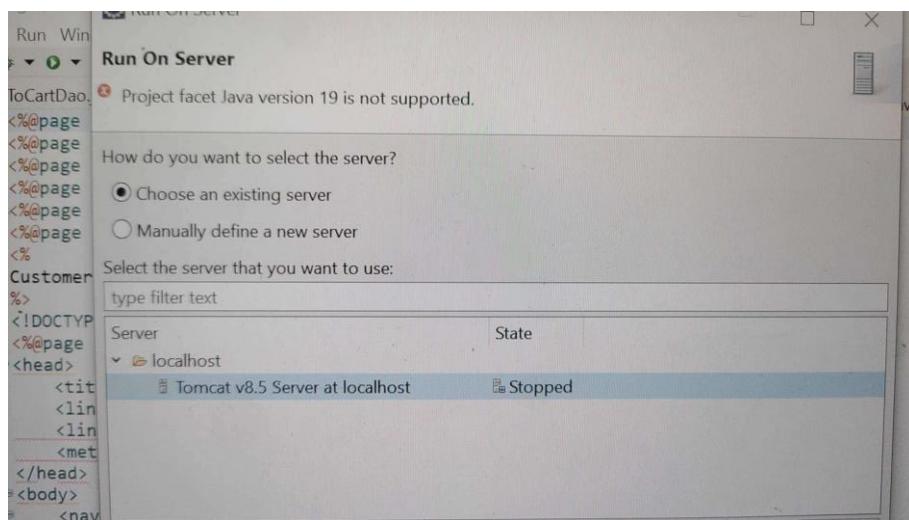


Figure 61: Challenge 1

From the error message "Project facet Java version 19 is not supported," it appears that the code or project was developed on a specific version of the server or runtime environment, but when attempting to integrate or run it on another system, the version mismatch caused compatibility issues and prevented the code from running properly.

2. We encountered an issue where MySQL in XAMPP wouldn't start, showing the error "MySQL shutdown unexpectedly." After consulting a YouTube tutorial, we repaired the MySQL database due to possible corruption by copying the data folder, deleting certain files and folders, replacing them with backups, and restarting the MySQL service. This experience highlighted the importance of utilizing alternative resources for troubleshooting technical problems.

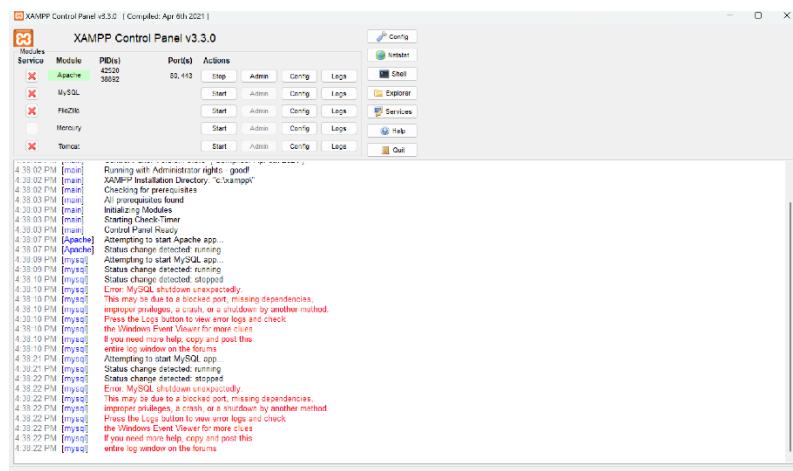


Figure 62: Challenge 2

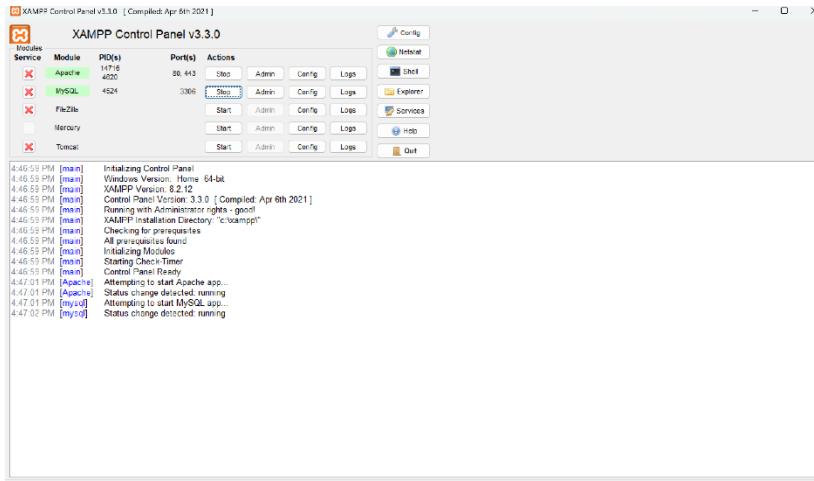


Figure 63: Challege 1 fixed

Multiple error was face during our development process one of them is explain below:

3. The problem we faced was that after placing an order from the cart, the order didn't get saved in the database. This issue occurred when we placed the order from cart.

```
55
56     if (true) {
57         AddToCartDao addToCart = new AddToCartDao();
58         out.println("product added");
59         if (addToCart.deleteCartItem(cartId)) {
60             response.sendRedirect(referer);
61         } else {
62             out.println("product added and cart not deleted");
63         }
64     } else {
65         out.println("product not");
66     }
67 }
```

Figure 64: Challenge 2

We attempted to delete the cart item inside the if (true) block, assuming that the checkout process is successful. However, there was no indication that the order is being saved in the database.

```
56     if (orderDao.checkOut(productId, customerId, totalPrice, qty)) {
57         AddToCartDao addToCart = new AddToCartDao();
58         out.println("product added");
59         if (addToCart.deleteCartItem(cartId)) {
60             response.sendRedirect(referer);
61         } else {
62             // error
63             out.println("product added and cart not deleted");
64         }
65     } else {
66         // error
67         out.println("product not");
68     }
69
70     out.print(cartId);
71     out.print(productId);
72 }
73 }
```

Figure 65: Challenge 2 fixed

We are first checking out the order using `orderDao.checkOut()` method, which likely involves saving the order details to the database. If the checkout process is successful, then we proceed to delete the cart item.

This ensures that the order is saved before attempting to delete the cart item, thus fixing the issue where the order wasn't getting saved in the database. By ensuring that the order is properly saved before deleting the cart item, we prevented any loss of order information.

9. Conclusion

This project gave us insights into implementing the MVC architecture in the Spring Framework of Java. We comprehended how separating the business logic, control logic, and user presentation logic through Models, Views, and Servlets facilitates modular and maintainable code. Additionally, we learned about the DAO pattern, which separates the data access logic between the Model and Controller classes.

10. References

balsamiq, 2024. *balsamiq*. [Online]

Available at: <https://balsamiq.com/wireframes/desktop/docs/intro/>
[Accessed 2 May 2024].

figma, 2024. *figma*. [Online]

Available at: <https://www.figma.com/about/>
[Accessed 2 May 2024].

Hanna, K. T., 2023. *techttarget*. [Online]

Available at: [https://www.techttarget.com/searchapparchitecture/definition/Eclipse-
Eclipse-](https://www.techttarget.com/searchapparchitecture/definition/Eclipse-Eclipse-)

Foundation#:~:text=Eclipse%20started%20in%202001%20when,complementing%20Apache's%20open%20source%20community.

[Accessed 2 May 2024].

javatpoint, 2024. *javatpoint*. [Online]

Available at: <https://www.javatpoint.com/servlet-tutorial>
[Accessed 3 May 2024].

javatpoint, 2024. *javatpoint*. [Online]

Available at: <https://www.javatpoint.com/java-jdbc>
[Accessed 2 May 2024].

tomcat.apache, 2024. *tomcat.apache*. [Online]

Available at:

<https://tomcat.apache.org/#:~:text=The%20Apache%20Tomcat%C2%AE%20software,of%20the%20Jakarta%20EE%20platform.>

[Accessed 2 May 2024].

tutorialspoint, 2024. *tutorialspoint*. [Online]

Available at:

https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
[Accessed 2 May 2024].

w3school, 2024. *w3school*. [Online]

Available at: <https://www.w3schools.com/MySQL/default.asp>
[Accessed 2 May 2024].