



**CS5054NI**

## **Advanced Programming and Technologies**

**50% Individual Coursework**

**Autumn 2024**

<b>Group Name: L2C15</b>			
<b>S.N.</b>	<b>Student Name</b>	<b>College ID</b>	<b>University ID</b>
1.	<b>Saman Gautam (L2C15)</b>	<b>NP01CP4A220312</b>	<b>22068007</b>
2.	<b>Anmol Adhikari (L2C15)</b>	<b>NP01CP4A220314</b>	<b>22068008</b>
3.	<b>Aagaman Mainali (L2C15)</b>	<b>NP01CP4A220472</b>	<b>22068131</b>
4.	<b>Rohit Raut (L2C15)</b>	<b>NP01CP4A220498</b>	<b>22068148</b>

**Assignment Due Date: Friday, May 10, 2024**

**Assignment Submission Data: Friday, May 10, 2024**

*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.*

## **Table of Contents:**

1.	Introduction: .....	1
1.1.	Objective: .....	2
1.2.	Aim:.....	2
2.	User Interface:.....	3
2.1.1.	User Interface Design for User Side Pages:.....	3
2.1.2.	User Interface Design for Admin Side Pages:.....	7
2.2.	Wireframe:.....	9
2.2.1.	Wireframe for the User Side Pages: .....	9
2.2.2.	Wireframe for Admin Pages:.....	16
2.3.	Actual Design:.....	20
2.3.1.	Registration Page: .....	20
2.3.2.	Login Page: .....	20
2.3.3.	Home Page:.....	21
2.3.4.	Product Page:.....	22
2.3.5.	Contact Us Page: .....	22
2.3.6.	User Profile:.....	23
2.3.7.	Admin Dashboard:.....	23
2.3.8.	Product Management: .....	24
2.3.9.	User Management: .....	24
3.	Class Diagram:.....	25
3.1.	Individual Class Diagram: .....	25
3.1.1.	Individual Class Diagram for model package:.....	25
3.1.2.	Individual class diagram of Util package:.....	28
3.1.3.	Individual class diagram of controller.servlet package:.....	29

3.2. Class Diagram for Overall System: .....	32
4. Method Description: .....	33
4.1. doGet():.....	33
4.2. doPost():.....	33
4.3. doDelete(): .....	33
4.4. processRequest():.....	34
4.5. addProduct(): .....	34
4.6. getAllProductInfo(): .....	34
4.7. deleteProduct():.....	34
4.8. updateProduct(): .....	35
4.9. getAllUserInfo(): .....	35
4.10. addData():.....	36
4.11. getLoginInfo():.....	36
4.12. checkEmailExists():.....	36
4.13. checkNumberIfExists():.....	37
4.14. checkUserNameIfExists(): .....	37
4.15. createSecureCookies(): .....	38
4.16. destroyCookies():.....	38
5. Test Cases: .....	40
5.1. Test Case 1:.....	40
5.2. Test Case 2:.....	42
5.3. Test Case 3:.....	44
5.4. Test Case 4:.....	45
5.5. Test Case 5:.....	46
5.6. Test Case 6:.....	47

5.7. Test Case 7:.....	49
5.8. Test Case 8:.....	51
5.9. Test Case 9:.....	52
5.10. Test Case 10: .....	54
5.11. Test Case 11: .....	56
5.12. Test Case 12: .....	57
5.13. Test Case 13: .....	61
5.14. Test Case 14: .....	63
5.15. Test Case 15: .....	65
6. Tools and Libraries Used: .....	67
6.1. Tools Used:.....	67
6.1.1. Eclipse:.....	67
6.1.2. XAMPP:.....	67
6.1.3. Figma:.....	68
6.1.4. Balsamiq:.....	68
6.1.5. MySQL-Connector-j-8.3.0:.....	68
6.1.6. MS Word:.....	69
6.2. Libraries Used:.....	69
6.2.1. Taglibs-standard-impl-1.2.5:.....	69
6.2.2. Taglibs-standard-spec-1.2.5:.....	70
6.2.3. Java.io.IOException:.....	70
6.2.4. HttpServletRequest: .....	70
6.2.5. HttpServletResponse:.....	71
6.2.6. ServletException:.....	71
7. Development:.....	72

7.1. Individual Contribution:.....	72
7.1.1. Admin Panel - Saman Gautam (22068007):.....	72
7.1.2. User Profile – Anmol Adhikari (22068008):.....	73
7.1.3. Aagaman Mainali (22068131):.....	73
7.1.4. Product Page – Rohit Raut (22068148):.....	74
7.2. Development Procedure: .....	76
8. Critical Analysis:.....	78
9. Conclusion: .....	80
10. References.....	81
11. Website References:.....	83

## Table of Figures:

Figure 1: Registration Design .....	3
Figure 2: Login page design .....	4
Figure 3: Home Page design .....	4
Figure 4: Product Page design .....	5
Figure 5: About Us Page .....	5
Figure 6: Contact Us Page .....	6
Figure 7: User Profile Design .....	6
Figure 8: Admin Dashboard design .....	7
Figure 9: Product Management design .....	7
Figure 10: Order Management Design .....	8
Figure 11: User Management Design .....	8
Figure 12: Wireframe of Register Page .....	9
Figure 13: Wireframe of Login Page .....	10
Figure 14: Wireframe of Home Page .....	11
Figure 15: Wireframe for Product Page .....	12
Figure 16: Wireframe for Contact Us Page .....	13
Figure 17: Wireframe for About-Us Page .....	14
Figure 18: Wireframe for Profile Edit Page .....	15
Figure 19: Wireframe for Admin Main Panel .....	16
Figure 20: Wireframe for Product Management Page .....	17
Figure 21: Wireframe for Order Management Page .....	18
Figure 22: Wireframe for User Management Page .....	19
Figure 23: Registration Final Page .....	20
Figure 24: Login Final Page .....	20
Figure 25: Home Page Final Page .....	21
Figure 26: Product Page .....	22
Figure 27: Contact Us Page .....	22
Figure 28: User Profile Page .....	23
Figure 29: Admin Dashboard Page .....	23
Figure 30: Product management Page .....	24

Figure 31: User Management Page .....	24
Figure 32: Class diagram for Login.java.....	25
Figure 33: Class diagram for model.java.....	26
Figure 34: Class Diagram for ProductAdd.java .....	27
Figure 35: Class Diagram for Authentication.java .....	28
Figure 36: Class Diagram for StringUtils.java.....	28
Figure 37: Class diagram for DeleteServlet.java .....	29
Figure 38: Class Diagram for LoginServlet.java .....	29
Figure 39: Class Diagram for LogoutServlet.java .....	29
Figure 40: Class Diagram for ProductAddServlet.java .....	30
Figure 41: Class Diagram for register.java .....	30
Figure 42: Class Diagram for UpdateProductServlet.java .....	30
Figure 43: Class Diagram for ProductAddServlet.java .....	31
Figure 44: Class Diagram for Overall System .....	32
Figure 45: Entering necessary credentials .....	41
Figure 46: Successfully registered and saved in database .....	41
Figure 47: Entering correct credential for user login.....	42
Figure 48: Redirected to Home page .....	43
Figure 49: Database user before changing user information.....	44
Figure 50: Database of user after changing user information.....	44
Figure 51: Cookie with username "XnezOptimus" was formed .....	45
Figure 52: Username "XnezOptimus" saved while logged into the website .....	46
Figure 53: Username cleared after pressing Logout .....	46
Figure 54: Correct credential for admin login. ....	47
Figure 55: Admin Sided Pages.....	48
Figure 56: Cookie created with username "Admin" .....	48
Figure 57: Database before adding another product. ....	49
Figure 58: Values that were to be added in database .....	50
Figure 59: Database after addition of product .....	50
Figure 60: Database "product" before deleting productID "67" .....	51
Figure 61: Pressing delete button for productID "67" .....	51

Figure 62: Database after deleting productID "67" .....	51
Figure 63: Giving wrong credential for login .....	52
Figure 64: Redirected to error.jsp page.....	53
Figure 65: Data of the product which already exist in the database .....	54
Figure 66: Changing information of the product by using updateproduct form .....	55
Figure 67: Information for the already existing product is changed.....	55
Figure 68: Data in "register" database.....	56
Figure 69: User list in user.jsp being displayed .....	56
Figure 70: Data that exist in the database.....	58
Figure 71: All the data being pulled from database .....	60
Figure 72: All the product details being pulled inside a product card in a managed way .....	62
Figure 73: Using same data to register in the website.....	64
Figure 74: Error message pop up after using same data to login .....	64
Figure 75: Clicking "Shop Now" button in home page .....	65
Figure 76: Redirected to Product page.....	66
Figure 77: Main Dashboard of Admin.....	72
Figure 78: User Profile Page .....	73
Figure 79: Error retrieving data from database in product card .....	75
Figure 80: Reference for Login Page (Facebook) .....	84
Figure 81: Reference for Home Page (Nikon.com) .....	85
Figure 82: Reference for Product Page (Itti Nepal) .....	86
Figure 83: Reference for Footer (Itti Nepal).....	86
Figure 84: Reference for navigation bar (canon.com) .....	86

## **Table of Tables:**

Table 1: To register in the system and store data onto the database and using the same data to login in the website.....	40
Table 2: To login to the website using user-based correct credential.....	42
Table 3: To test if user is able to edit their profile or not.....	44
Table 4: To check if the cookie and session is created or not after a login is successful .....	45
Table 5: To test if cookie is being destroyed or not after logging out or not. ....	46
Table 6: To test if admin-based login is working or not. ....	47
Table 7: To test if product is being added or not by using “Add Product” link.....	49
Table 8: To if a product is being deleted or not after clicking delete button in productmanagement.jsp.....	51
Table 9: To test if error is being thrown or not when using incorrect credential for login. ....	52
Table 10: To test if admin is able to update product data which already exists in database. ....	54
Table 11: To test if user detail is being displayed or not for user management.....	56
Table 12: To test if product details is being displayed or not in product management. .	57
Table 13: To test if product details is being displayed or not in product management. .	61
Table 14: To check if a user can register using same username and email.....	63
Table 15: To test if button “Shop Now” in home page is working or not. ....	65

## 1. Introduction:

In present time where internet has been developed a lot than before 5-10 years it is very much necessary to have an e-commerce website so that we can promote our business. E-commerce website help a business to grow faster and gain different new customers through internet. Because of these reason developers of a sophisticated and user-centric platform are increasing every year. So, to teach provide us knowledge about a good e-commerce platform this coursework was given whose main task was to create an e-commerce website for an electronic and gadget store by using MVC pattern which divides the application into three distinct packages. Each of different package's code would perform different work like database management, create connection with database and show webpages to the user. The tasks outlined in this coursework include the implementation of essential features such as user authentication, product management, order tracking, and user profile management.

The task that had to be completed in this coursework was divided into three different parts as task-1, task-2 and task-3. Task-1 focuses on critical features like Register, Login and Admin Panel Feature. The Login Feature requires database integration for secure authentication and session management, separating both users and administrators with distinct access privileges and permissions. Effective error handling mechanisms are implemented to guide users through the login process, while the Admin Panel Feature empowers administrators with comprehensive product management capabilities, including adding, updating, deleting products and also have an order tracking feature. One of the most core feature which is in an e-commerce website is completed in task-1. Task-2 dives deep into enhancing user experience in the website by implementing features like User Profile Page, Home Page, About Us Page and other pages which are used by user for the accessibility of different products and information. These features enrich user engagement by providing customized profile management, seamless communication channels, and intuitive product browsing experiences. With a focus on user satisfaction and system efficiency. This project aims to deliver an e-commerce platform that excels in functionality, security, and user experience and meeting the diverse needs of electronics and gadgets enthusiasts in the digital age.

This project is all about building a great e-commerce website for people who love electronics and gadgets. We're following the MVC pattern for organizing the website and make things neat and easy to manage. We have different tasks to do, like making sure users can log in safely, giving admins customized and special controls and creating pages where users can see their profiles, contact us, and shop comfortably. Our goal is to make shopping online for electronics and gadgets easy, and secure for everyone who visits our website.

### **1.1. Objective:**

The objective of this coursework is given below:

- To gain comprehensive knowledge understanding of MVC architectural pattern.
- To develop a proficiency in using and integrating database with web applications.
- To implement secure user authentication and data management functionalities.
- To learn implementing error handling and data validation processes more effectively.
- To develop a teamwork and collaboration skills by working together in same project.

### **1.2. Aim:**

The aim for this project is to successfully develop an innovative e-commerce website for electronics and gadgets enthusiasts by the usage of MVC architectural pattern.

## 2. User Interface:

A user interface (UI) on a software or a website is visual element of a system which includes interactive features that user interacts with to navigate, interact and search for different content on the site. It includes different elements such as button, menus, texts, form to images, text and videos that users see and interact with on their screens (Hook, 2021).

A good interface design makes sure that the user interaction in the website is enjoyable and efficient. It must be visually appealing, easy to understand and responsive across different devices and screen sizes. Many real-world businesses are dependent on web and mobile apps. This makes a company to focus and prioritize a good UI for their system. If a user interface is not good, for example; a text is being blend to the background then it will give user a hard time to read the paragraph (Jitnupong, 2017).

The design for user interface that our group created for the website is given below:

*Note: For the Figma design file please use the given link [\(Figma Design File\)](#).*

### 2.1.1. User Interface Design for User Side Pages:



Figure 1: Registration Design

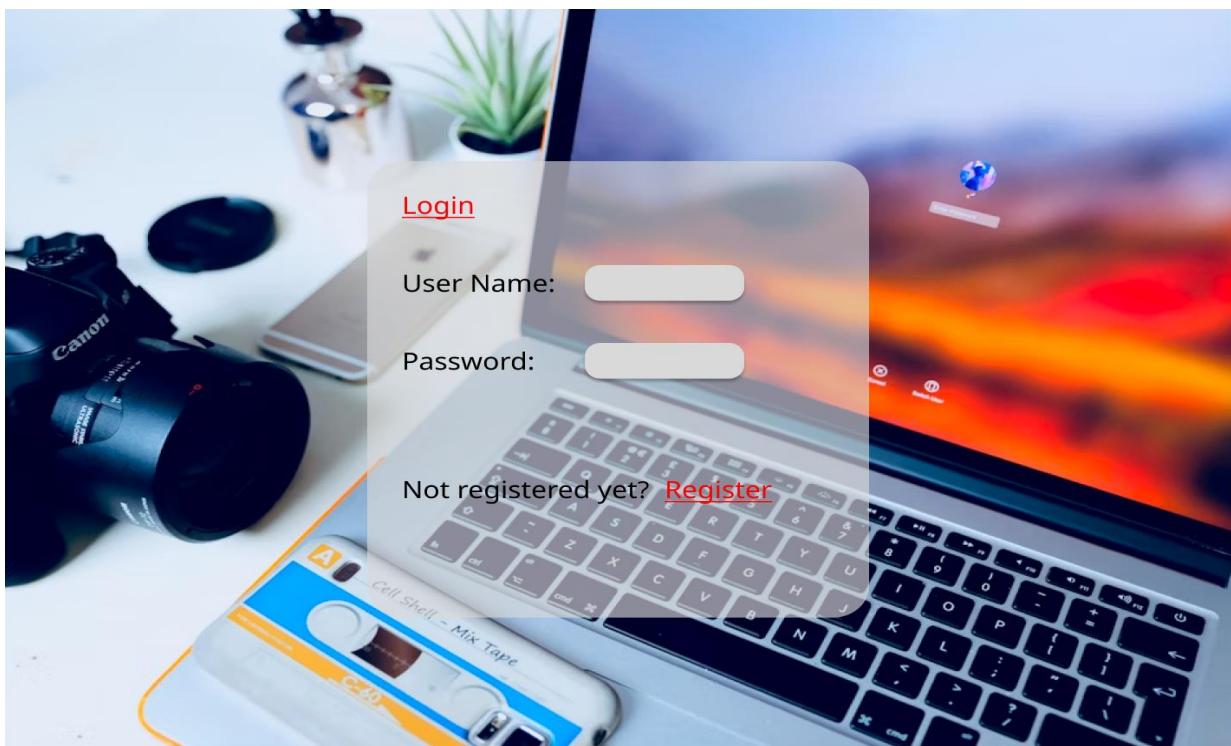


Figure 2: Login page design

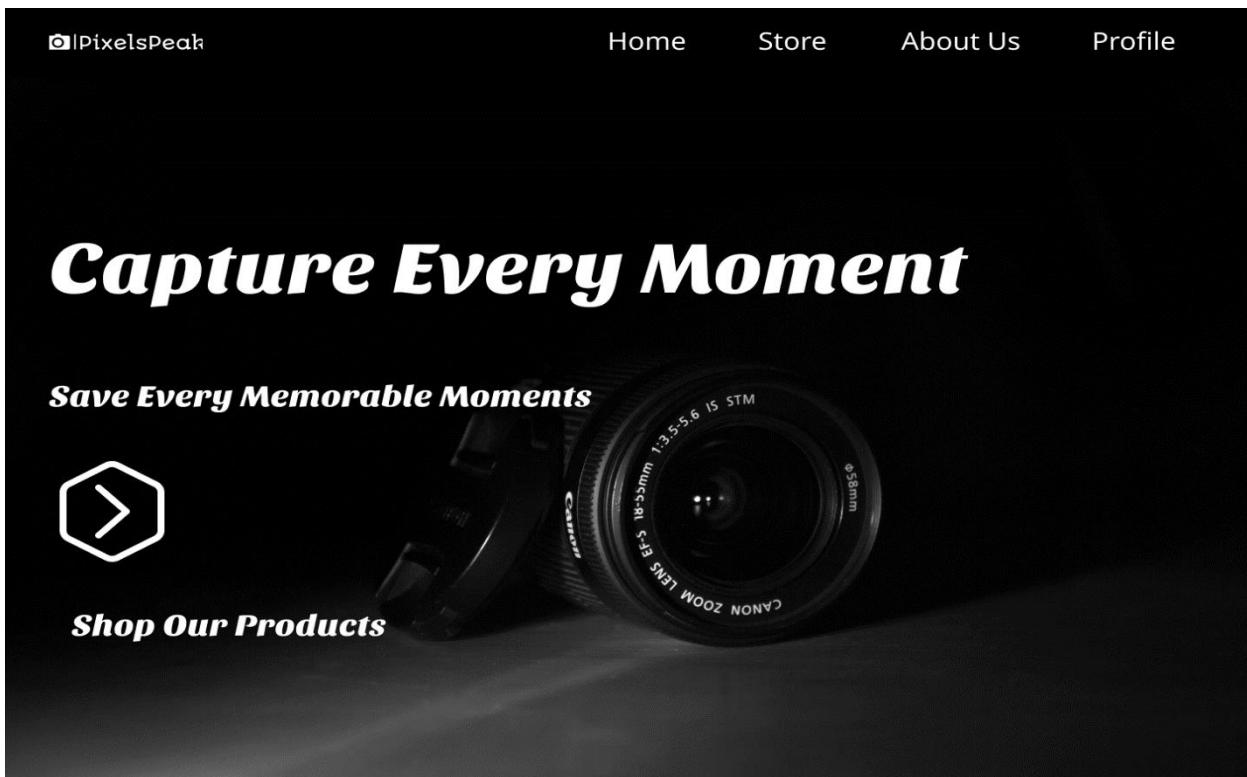


Figure 3: Home Page design

## Filters

## Price

\$1000 To \$14000

**Apply**

## Status

- In-Stock
- Out of Stock

## Company

- Sony
- Nikon
- Canon

**Check-Out**

## Camera/Accessories



Lorem ipsum dolor sit amet



\$ 1300

**Add To Cart**

Lorem ipsum dolor sit amet



Rs. 1300

**Add To Cart**

Lorem ipsum dolor sit amet



Lorem ipsum dolor sit amet



Figure 4: Product Page design

**Our Hardworking Team****Capturing Success as a Team**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vitae justo nec est venenatis dictum. Ut placerat urna ut lectus sagittis, vel venenatis velit laoreet. Sed lacinia urna nec ex faucibus, nec convallis est fermentum. Fusce auctor consectetur erat, id dictum nibh aliquet a. Curabitur vitae augue eu nunc aliquet lacinia. Donec consequat, turpis sit amet vehicula condimentum, ex eros blandit magna, at interdum neque metus nec felis. Integer sodales sem eu sapien dapibus, vel malesuada purus varius. Sed ullamcorper libero a elit mattis tincidunt. Sed congue, lorem sed efficitur convallis, neque nunc tristique turpis, nec varius ex mauris nec est. Sed in risl eget dolor ultrices malesuada ut sit amet urna. Vivamus tincidunt vitae orci vitae fermentum. Donec id consequat velit. Nunc ultricies consectetur elit, vitae rutrum magna gravida id. Maecenas vitae semper turpis.



Figure 5: About Us Page

## Contact Us

Say something to start a live chat

+977 9861841694

NP01CP4A220312@islingtoncollege.edu.np

Kamal Marg, Kamal Pokhari, Kathmandu, Nepal

Select Subject?

General Inquiry    General Inquiry    General Inquiry    General Inquiry

Message

Write your message..

Figure 6: Contact Us Page

## Username

Twitter

Facebook

Instagram

## Settings

Username	<u>Username</u>
Email	<u>Email</u>
Phone	<u>Phone</u>
Address	<u>Address</u>

Figure 7: User Profile Design

### 2.1.2. User Interface Design for Admin Side Pages:

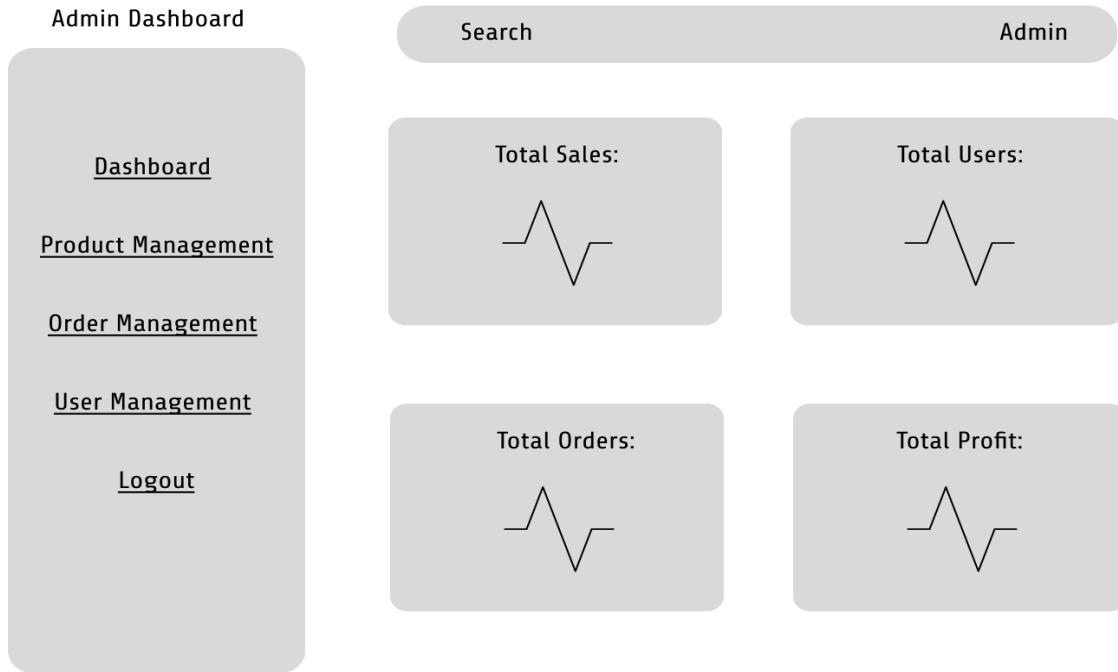


Figure 8: Admin Dashboard design

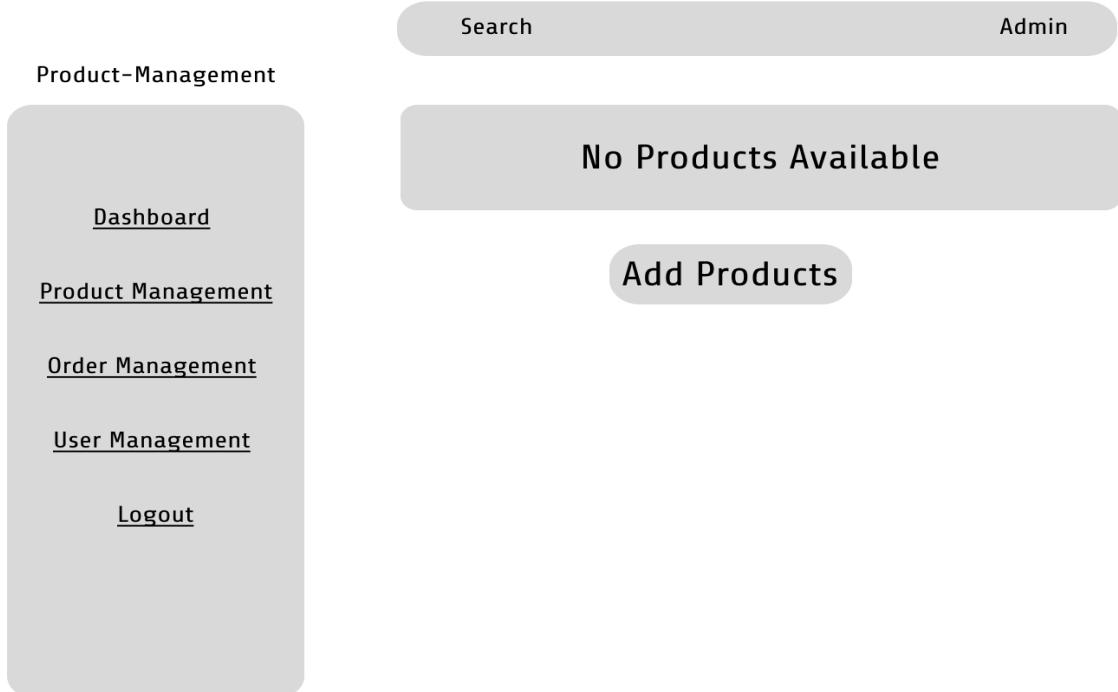
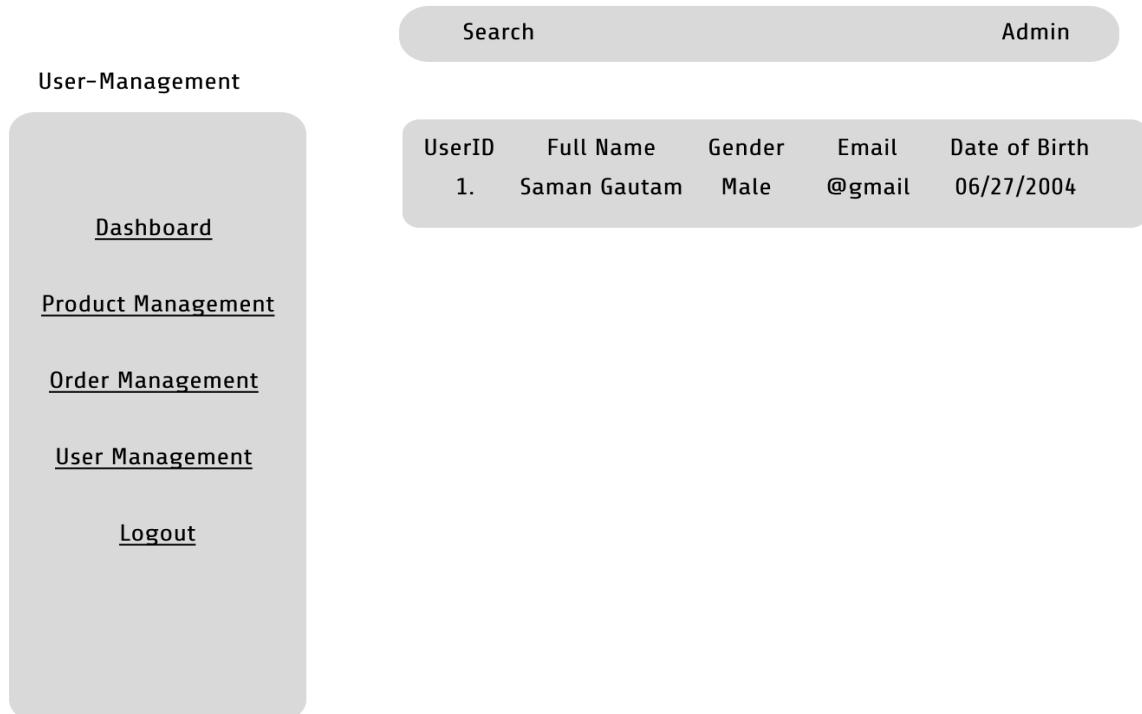


Figure 9: Product Management design

*Figure 10: Order Management Design**Figure 11: User Management Design*

## 2.2. Wireframe:

A wireframe is a visual representation of a website or application's layout, structure and functionality. It is like blueprint or a skeleton of a website that outlines the basic elements and placement of components like buttons, text, paragraph, images, etc. Wireframes mostly contains simple shapes, lines and other components to show various element's position such as navigation options, buttons, forms, etc. It mostly focuses on simplicity and position of components than an actual design like UI/UX design. Wireframes are used in early stages of development before the creation of UI prototype (Balsamiq, 2018).

### 2.2.1. Wireframe for the User Side Pages:

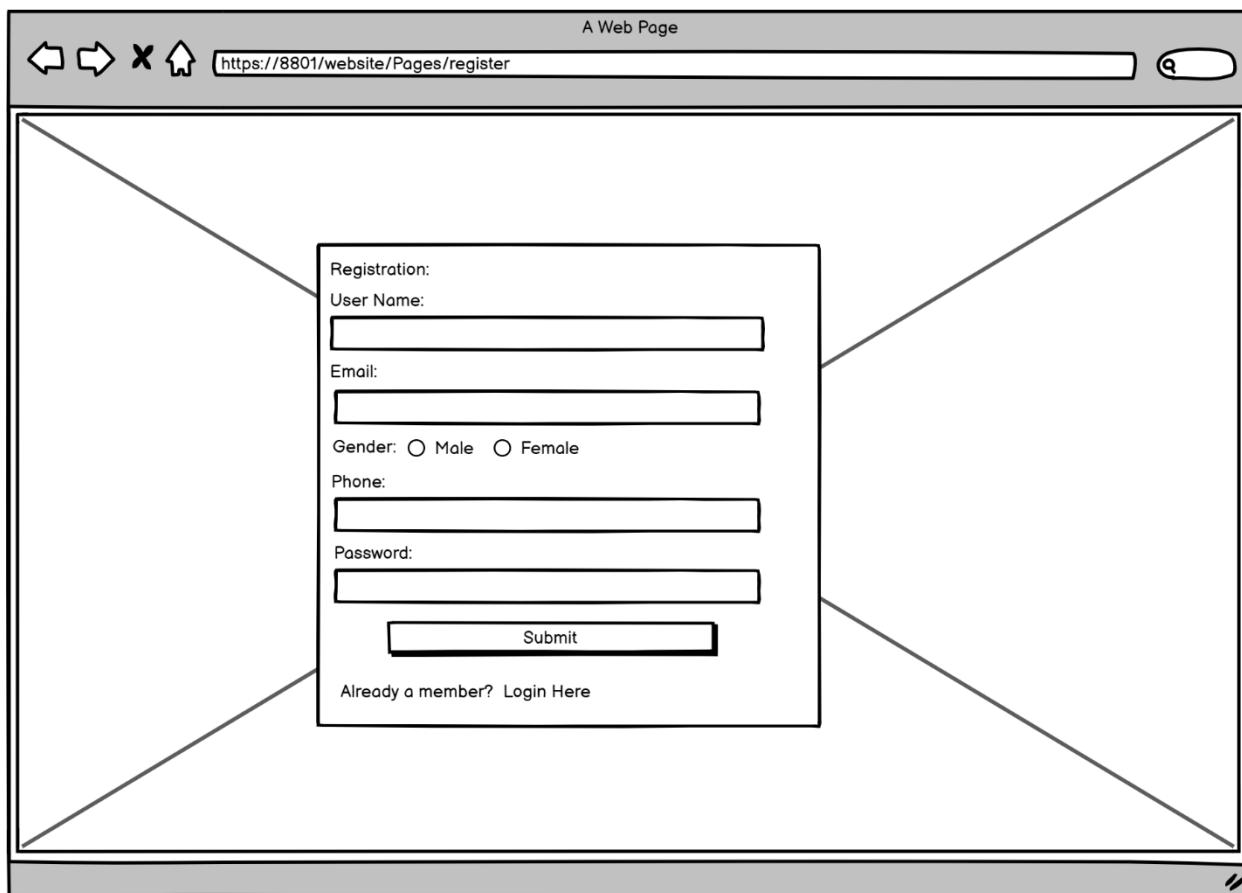


Figure 12: Wireframe of Register Page

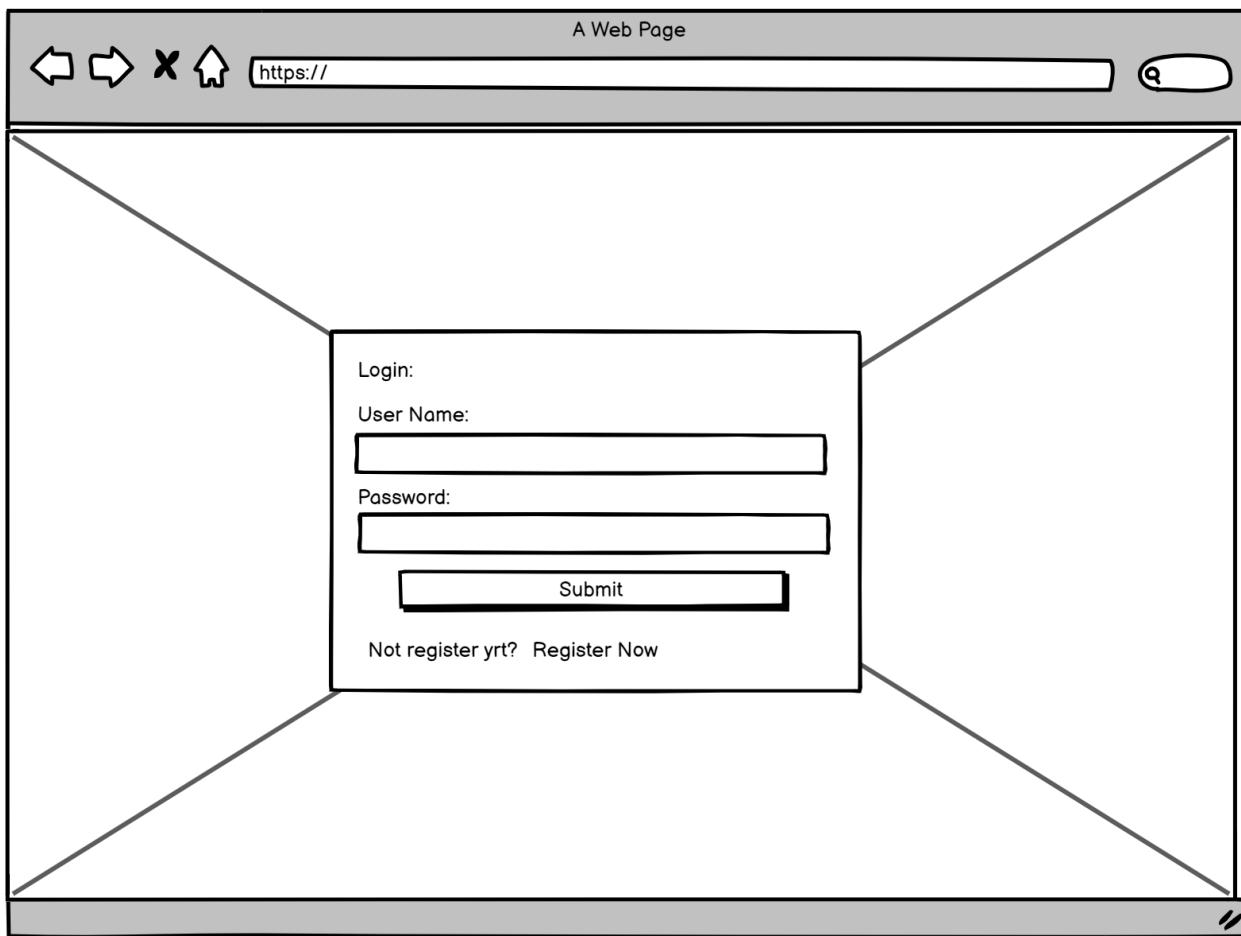


Figure 13: Wireframe of Login Page

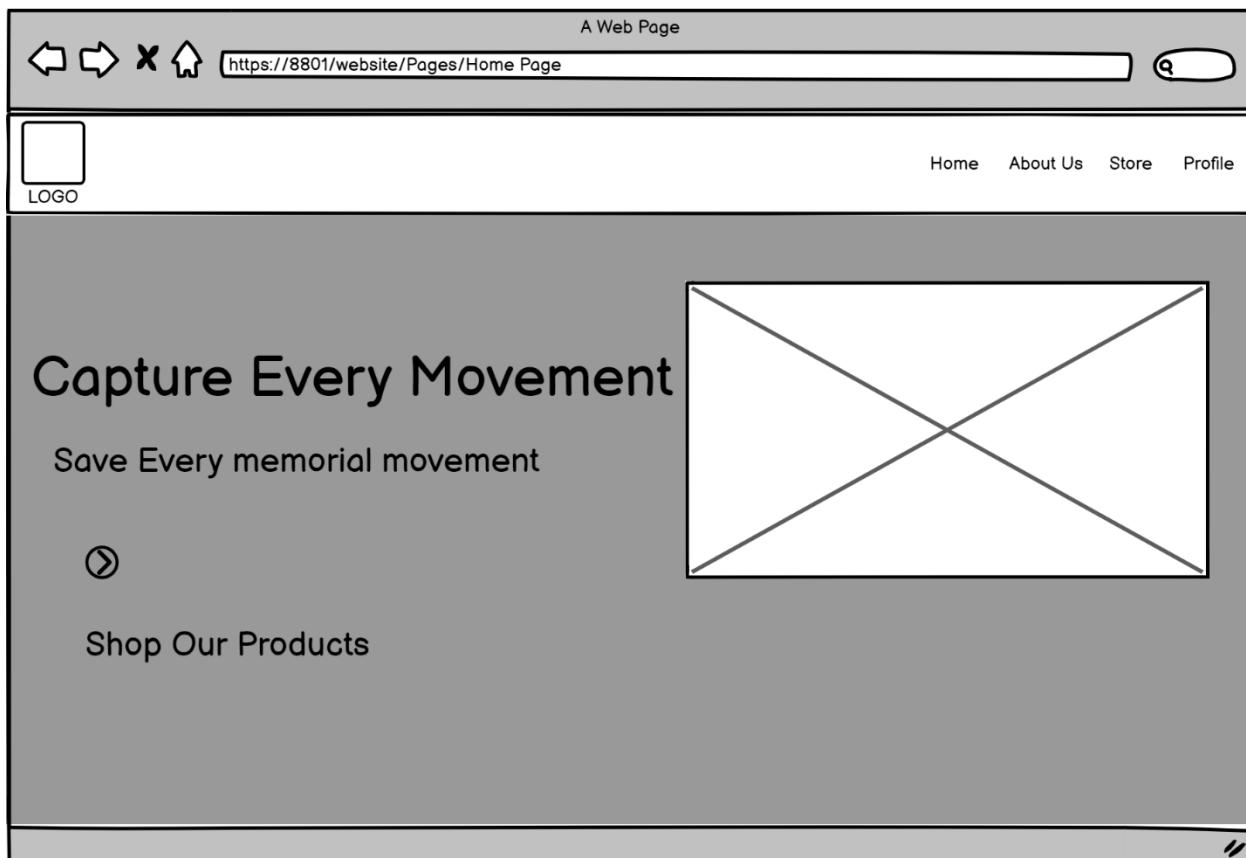


Figure 14: Wireframe of Home Page

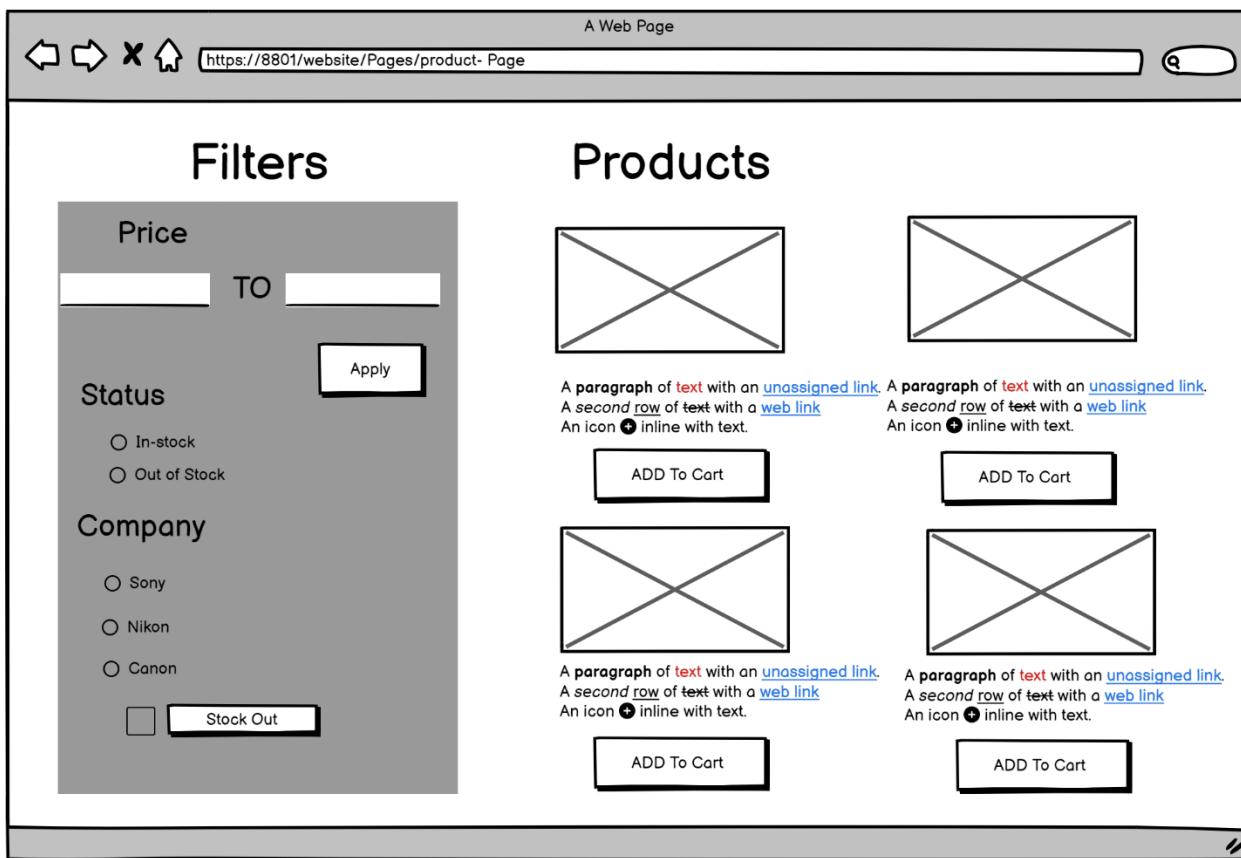


Figure 15: Wireframe for Product Page

A Web Page

<https://8801/website/Pages/Contact US>

## Contact US:

Say something to start a live chart.

phone 987867867878

phone anupraut1672@gmail.com

phone Kamla marga

First Name: \_\_\_\_\_

Last Name: \_\_\_\_\_

Email: \_\_\_\_\_

Phone Number: \_\_\_\_\_

Select Subject:

General Inquiry    General Inquiry    General Inquiry

Your Message:

\_\_\_\_\_

Figure 16: Wireframe for Contact Us Page

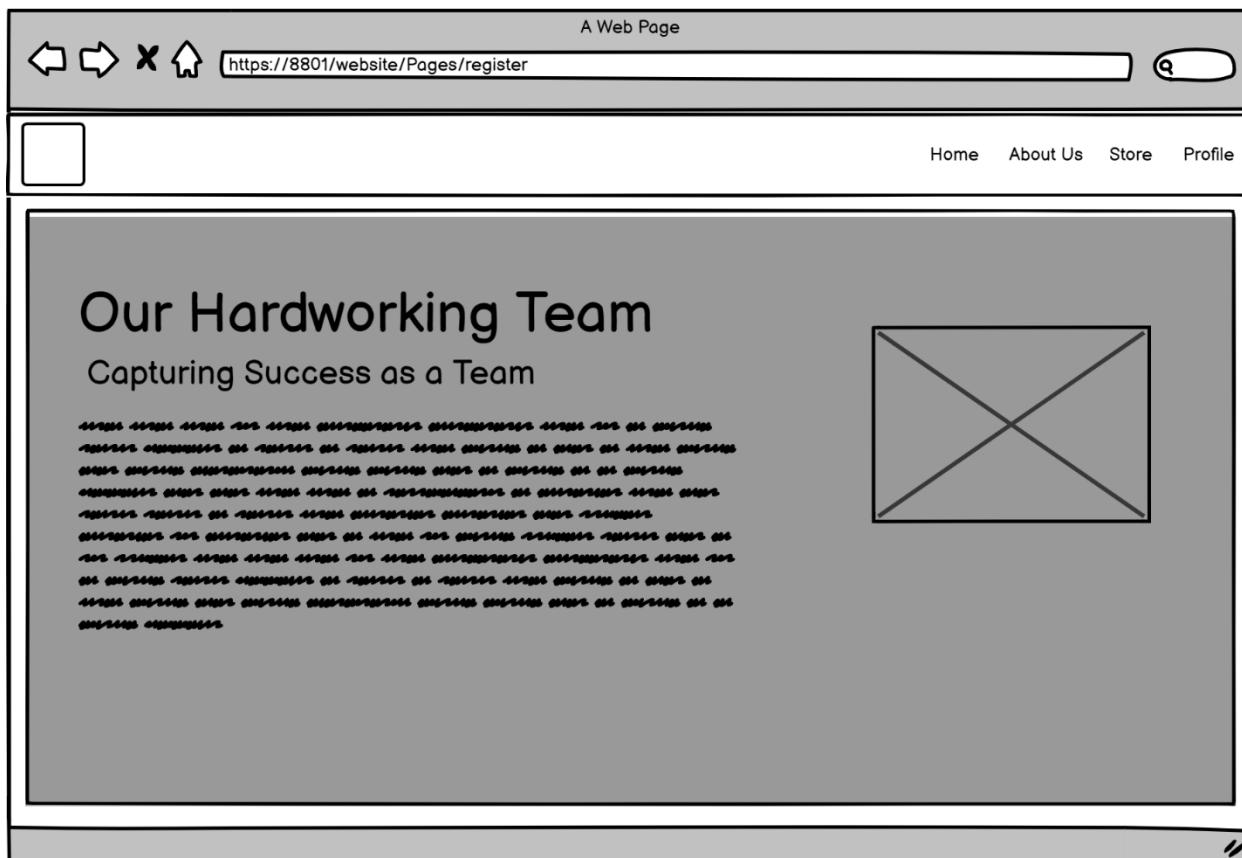


Figure 17: Wireframe for About-Us Page

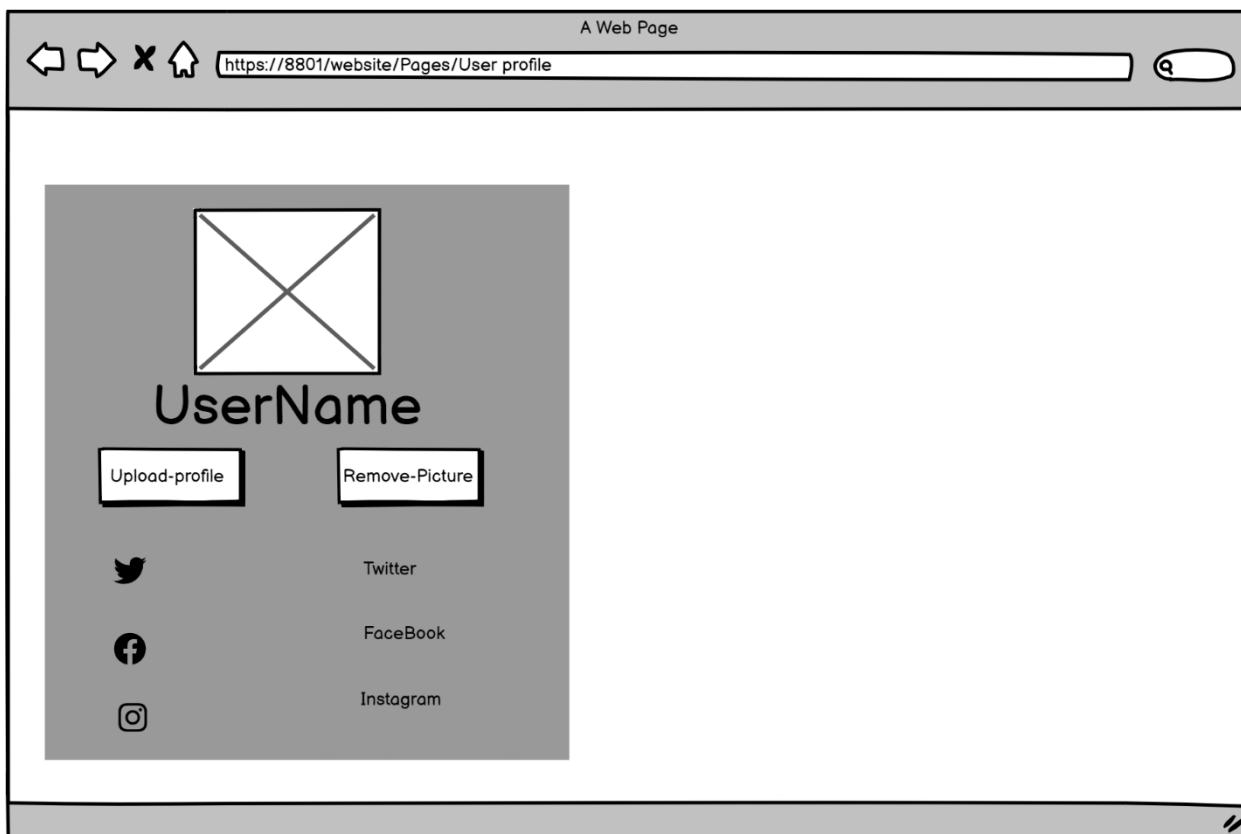


Figure 18: Wireframe for Profile Edit Page

## 2.2.2. Wireframe for Admin Pages:

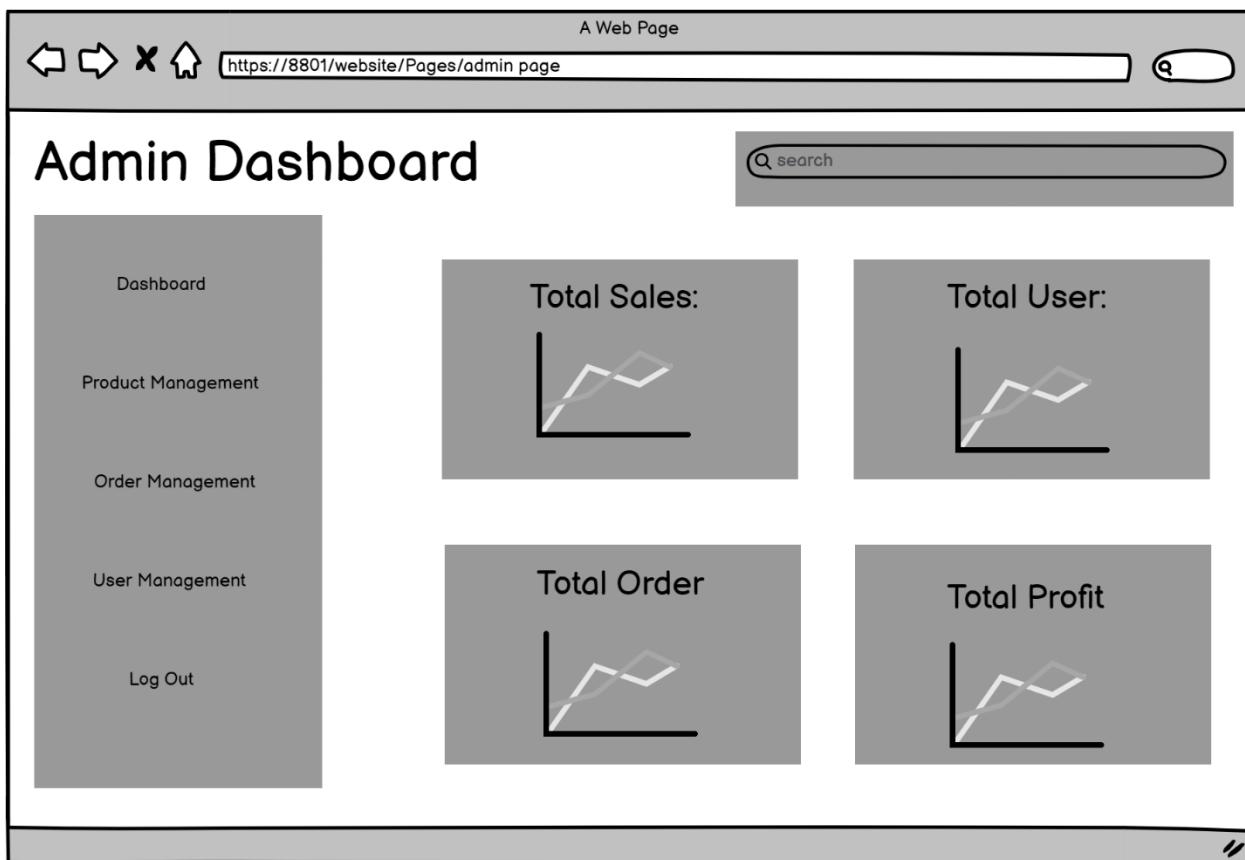


Figure 19: Wireframe for Admin Main Panel

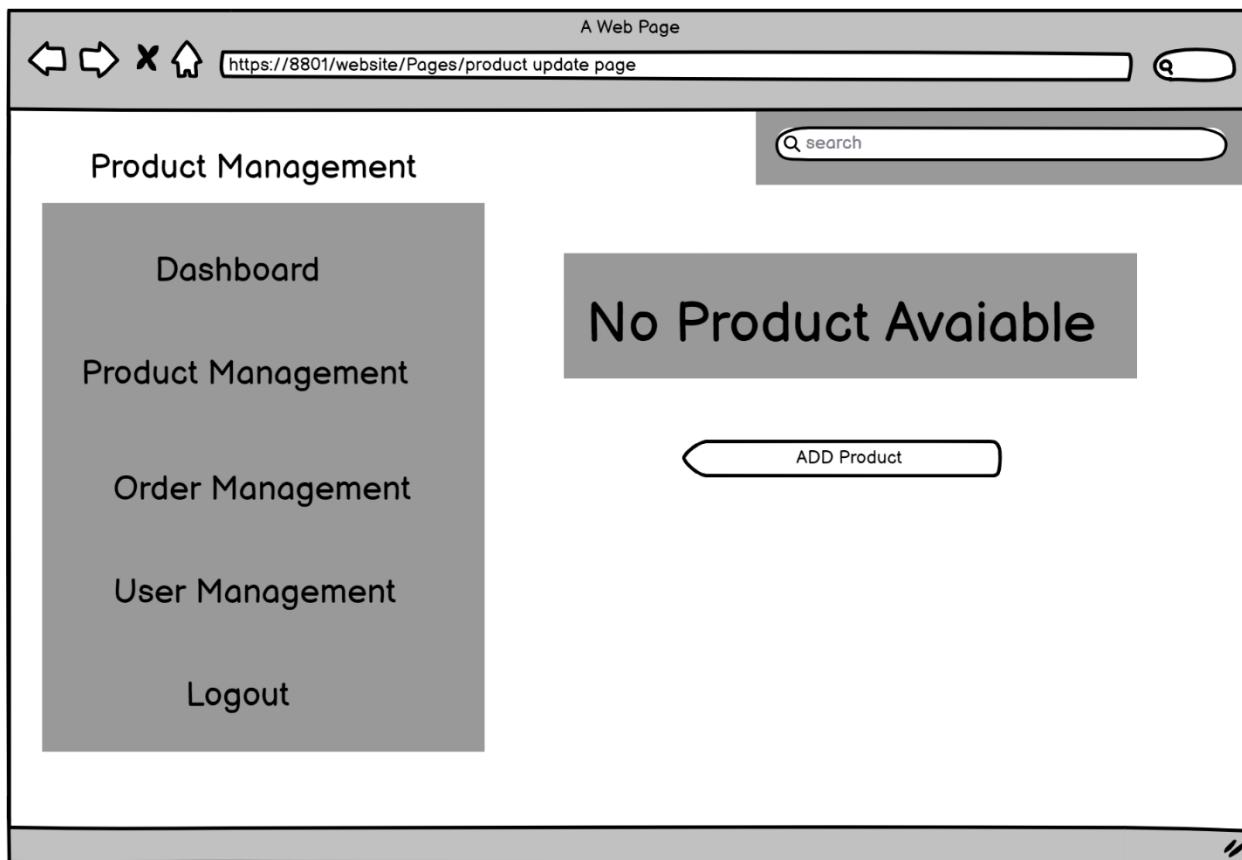


Figure 20: Wireframe for Product Management Page

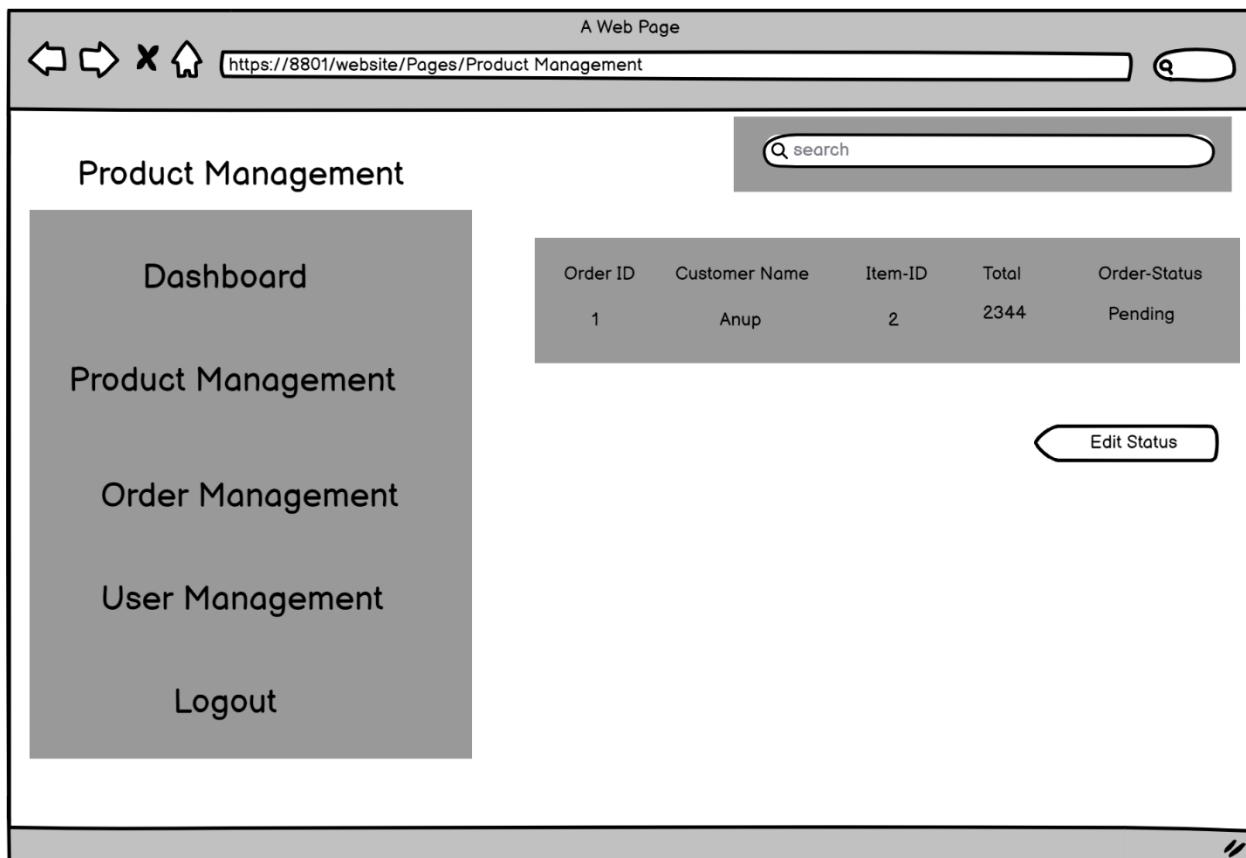


Figure 21: Wireframe for Order Management Page

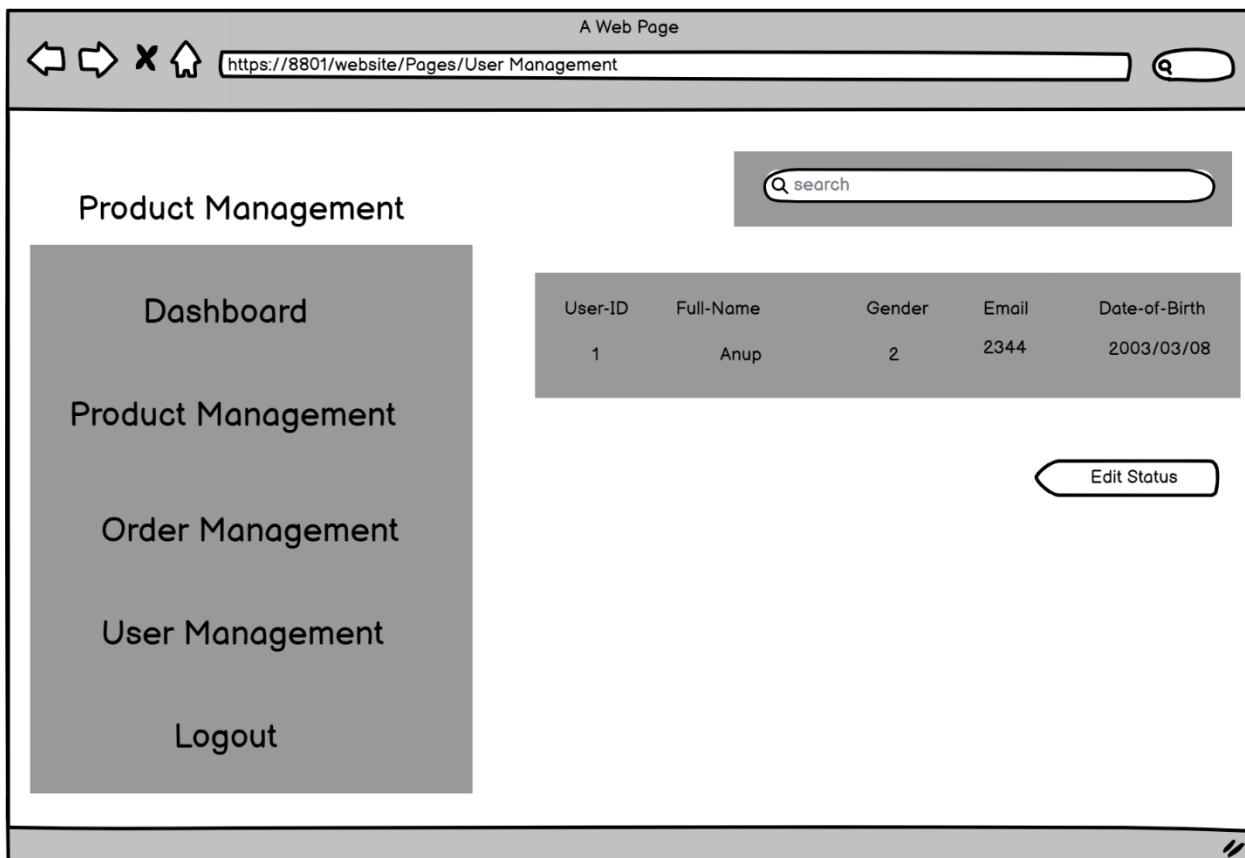


Figure 22: Wireframe for User Management Page

### 2.3. Actual Design:

The actual design of the website after completion is different than the Figma UI/UX design and wireframes.

#### 2.3.1. Registration Page:

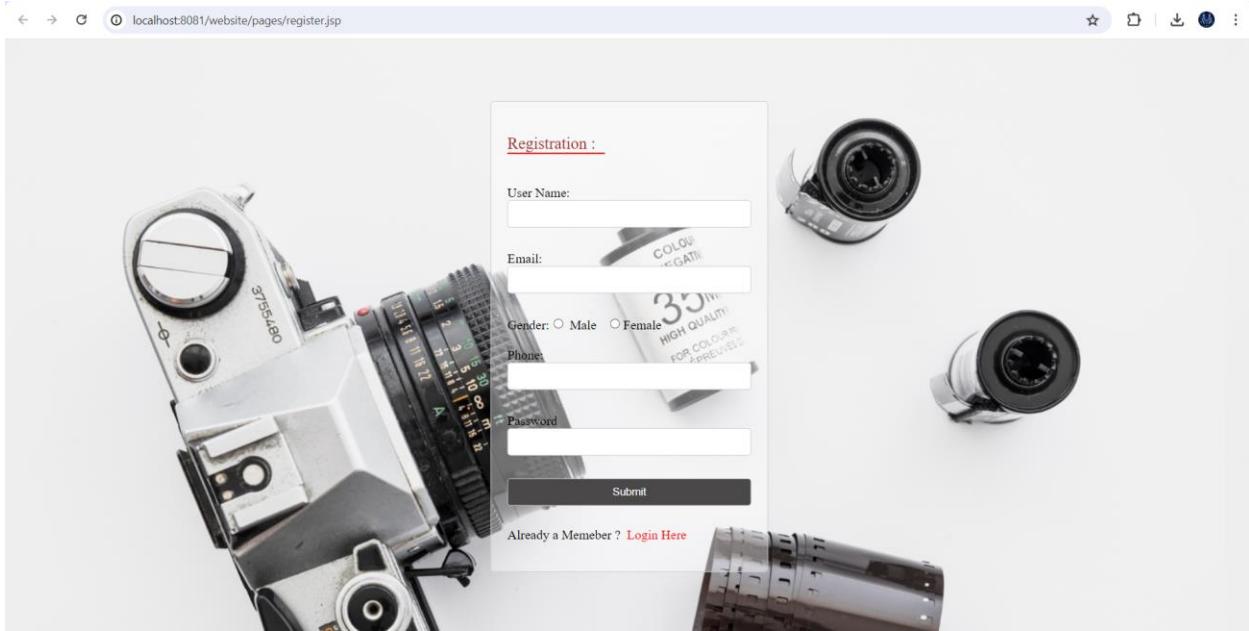


Figure 23: Registration Final Page

#### 2.3.2. Login Page:

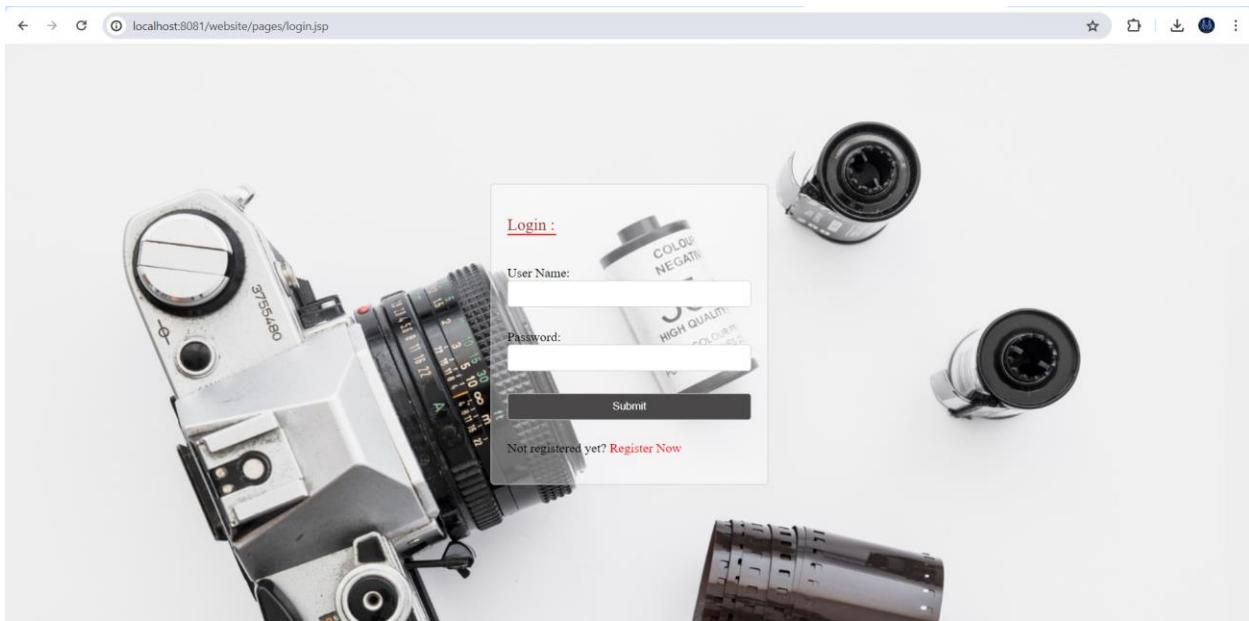
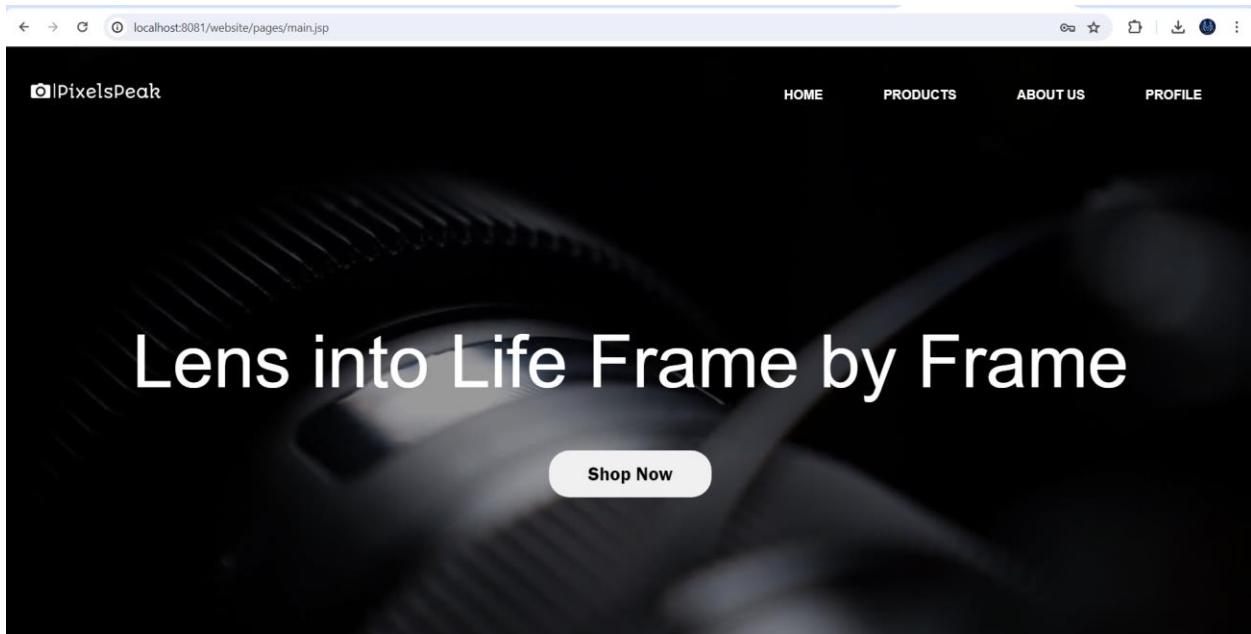


Figure 24: Login Final Page

### 2.3.3. Home Page:



#### Latest from Pixels



##### Never Break Your Flow

With intelligent features and diverse lenses, our systems seamlessly integrate with your creativity. Whether capturing speed or subtlety, autofocus ensures precision. Intuitive controls keep you focused, adjusting effortlessly. From city bustle to serene landscapes, every moment is clear. We guarantee your vision is never compromised because in the zone, flow is paramount.



##### Equip Your Vision Capture Your World

Our products are more than tools; they're gateways to creativity, enabling you to transform your vision into stunning reality. From high-resolution cameras to versatile lenses and cutting-edge accessories, we provide everything you need to capture every moment, every detail, and every emotion. With our equipment in hand, you're empowered to explore the world through your unique perspective, seizing every opportunity to tell your story with clarity and brilliance.

*Figure 25: Home Page Final Page*

### 2.3.4. Product Page:

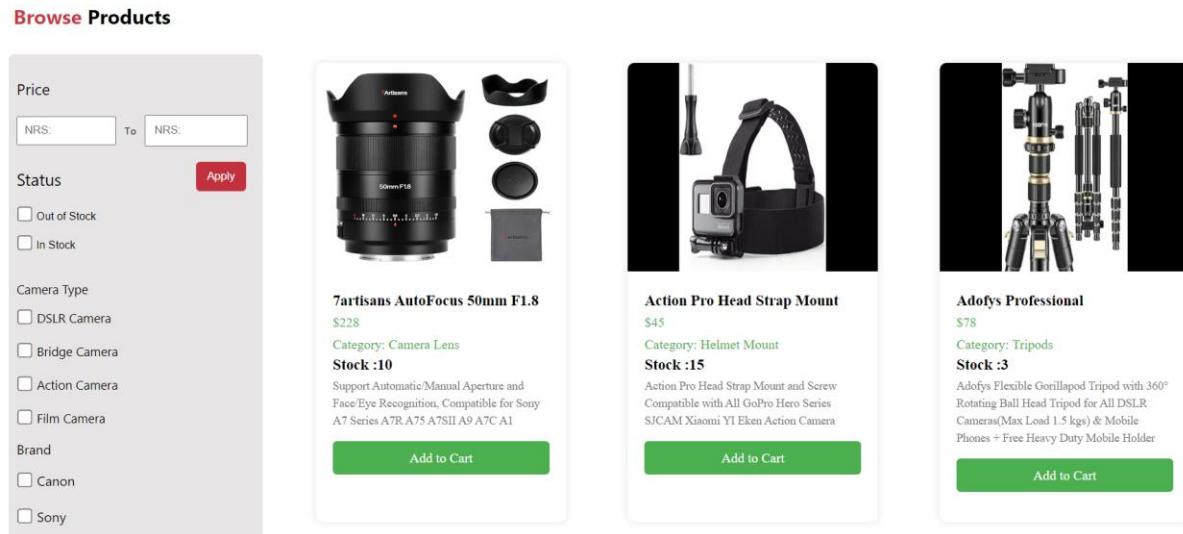


Figure 26: Product Page

### 2.3.5. Contact Us Page:

The screenshot shows the contact us page for PixelsPeak. It includes:

- PixelsPeak** logo and navigation menu with links to HOME, PRODUCTS, ABOUT US, and PROFILE.
- A large "Contact Us" form with fields for Name, Email, and Message, and a Submit button.
- PixelsPeak** footer with links to Home, Product, About Us, and Profile, and copyright information: Copyright © 2024 | Pixel Developers | All rights reserved.
- About Us** section with the text: "Pixel Peak, your go-to source for premium cameras. We cater to both photography enthusiasts and professionals, offering a carefully curated selection of top-notch equipment. With a commitment to innovation and customer satisfaction!"
- Contact information: KamalPokhari, Kathmandu; +977 9803184146; Pixels100@gmail.com

Figure 27: Contact Us Page

### 2.3.6. User Profile:

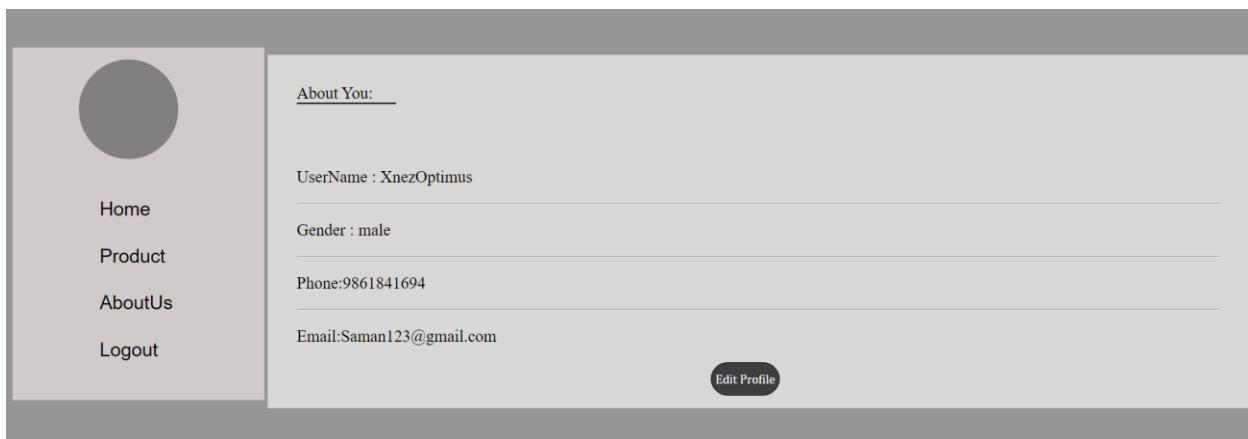


Figure 28: User Profile Page

### 2.3.7. Admin Dashboard:

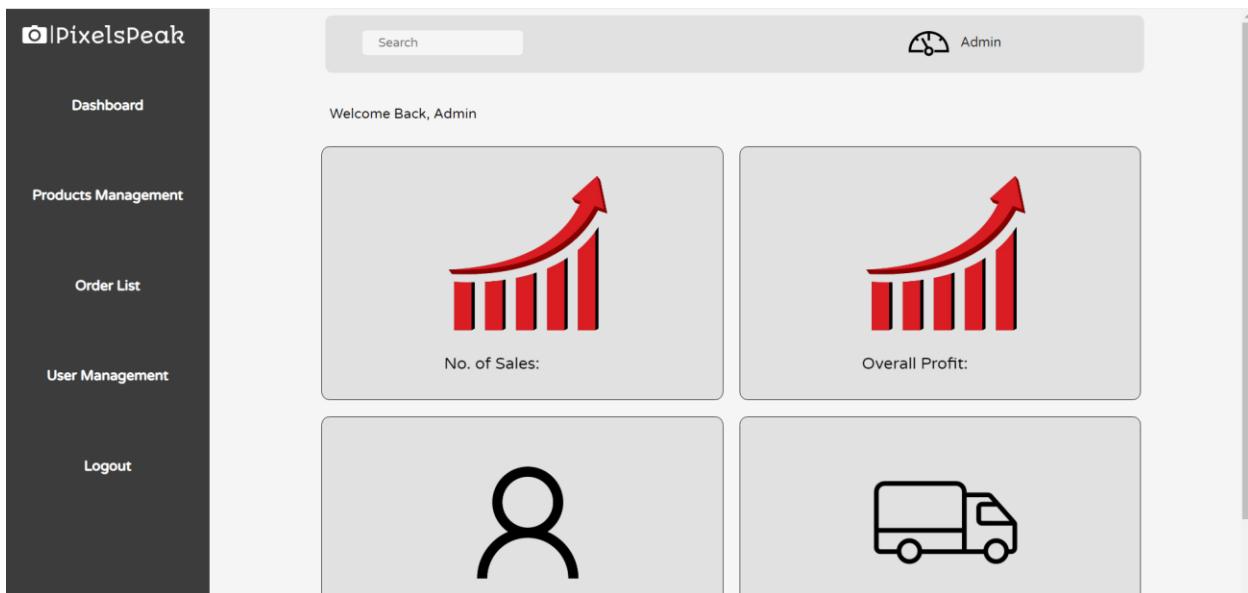


Figure 29: Admin Dashboard Page

### 2.3.8. Product Management:

The screenshot shows a web application interface for product management. On the left, a dark sidebar menu includes 'Dashboard', 'Products Management' (selected), 'Order List', 'User Management', and 'Logout'. The main content area has a header 'Add-New-Product' and a sub-header 'View Your Products'. A table lists six products with columns for SN, Product Name, Price, Stock Qty., Product Category, Product Description, Product Image, and Actions (Edit, Delete). The products are:

SN	Product Name	Price	Stock Qty.	Product Category	Product Description	Product Image	Actions
42	Tartisans AutoFocus 50mm F1.8	228	10	Camera Lens	Support Automatic/Manual Aperture and Face/Eye Recognition, Compatible for Sony A7 Series A7R A7SII A9 A7C A1		Edit Delete
43	Action Pro Head Strap Mount	45	15	Helmet Mount	Action Pro Head Strap Mount and Screw Compatible with All GoPro Hero Series SJCAM Xiaomi YI Eken Action Camera		Edit Delete
44	Adofys Professional	78	3	Tripods	Adofys Flexible Gorillapod Tripod with 360° Rotating Ball Head Tripod for All DSLR Cameras (Max Load 1.5 kgs) & Mobile Phones + Free Heavy Duty Mobile Holder		Edit Delete
45	Sony Tripod	57	27	Tripods	AmazonBasics WT3130T+WT3111H 50-Inch Lightweight Tripod with Bag, Load Capacity: 2 Kg		Edit Delete
46	Canon EF 75-	199	15	Camera Lens	Canon EF 75-300mm f/4-5.6 III Telephoto Zoom Lens for Canon SLR Cameras, 75-300 millimeter		Edit

Figure 30: Product management Page

### 2.3.9. User Management:

The screenshot shows a web application interface for user management. The sidebar menu is identical to Figure 30, with 'Products Management' selected. The main content area has a header 'View Your User' and a table listing two users with columns for SN, User Name, Email, Gender, and Phone. The users are:

SN	User Name	Email	Gender	Phone
4	Ram Prasad	test@gmail.com	Male	9876543210
7	saman	sam@gmail.com	male	0987654321

Figure 31: User Management Page

### 3. Class Diagram:

Class diagram can be considered as the building block of the Object-Oriented Programming, the class diagram which is created in the early developing phase of the project contains system's class, methods and attributed that class diagram in the later stage are converted into an actual software. The Class diagrams takes the form of a quick description of the operation of a system, the interactions between system components at various levels, and the implementation strategy for each component. We can represent a software using class diagrams at a high level of abstraction without having a look of the source code. It is a structure diagram in the Unified Modelling Language (UML) that's shows the system's classes, their attributes, relationship among different classes and constraints. It can also be considered as a blue print of a system which gives a layout for the software. There are different types of class diagram which is used for the representation of the system which makes it easier to make stakeholders and investors know about the project visually (IBM, 2024) (Felici, 2017).

There are different classes which are created according to their task and nature of work in this coursework. As we are following MVC architecture pattern the classes are also separated accordingly.

#### 3.1. Individual Class Diagram:

An individual class diagram represents the structure and relationships of individual classes with a system.

##### 3.1.1. Individual Class Diagram for model package:

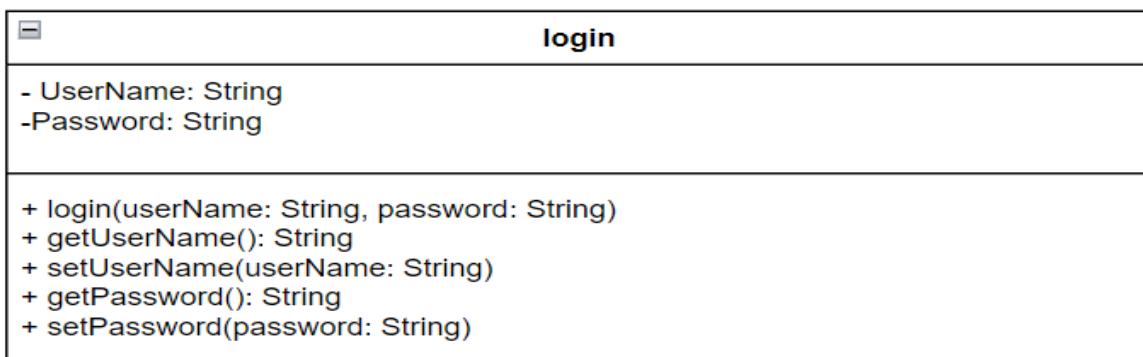


Figure 32: Class diagram for Login.java

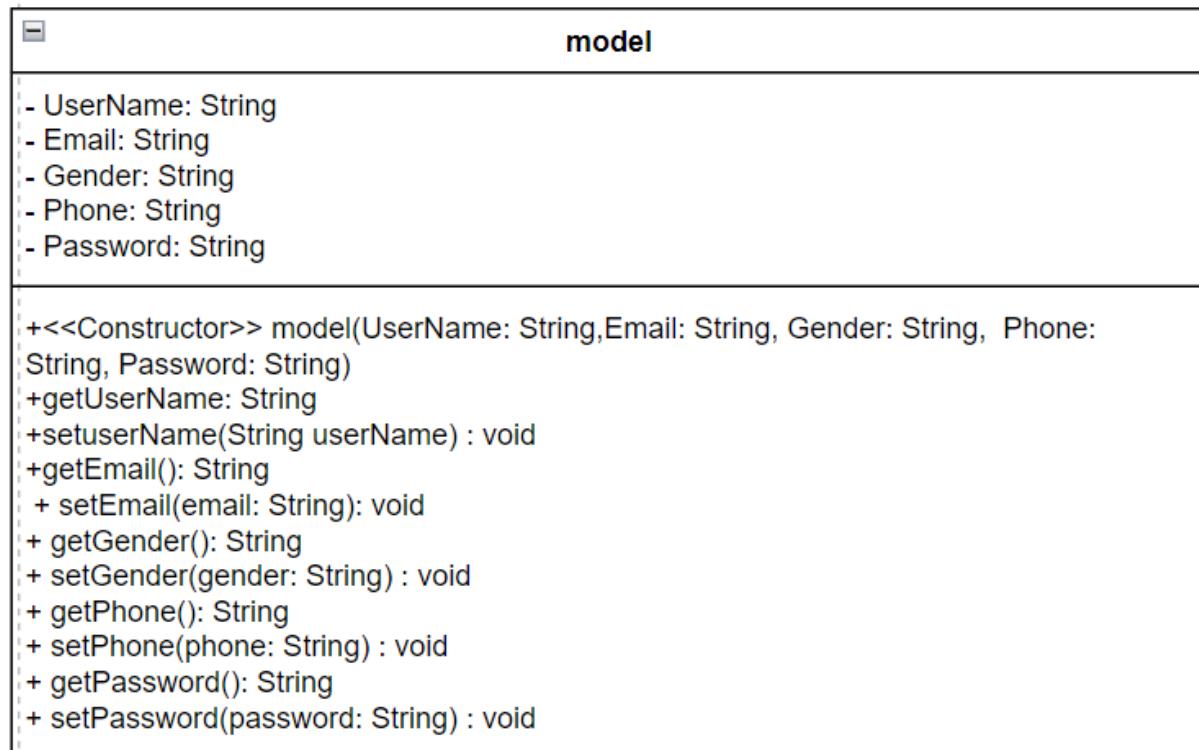


Figure 33: Class diagram for model.java

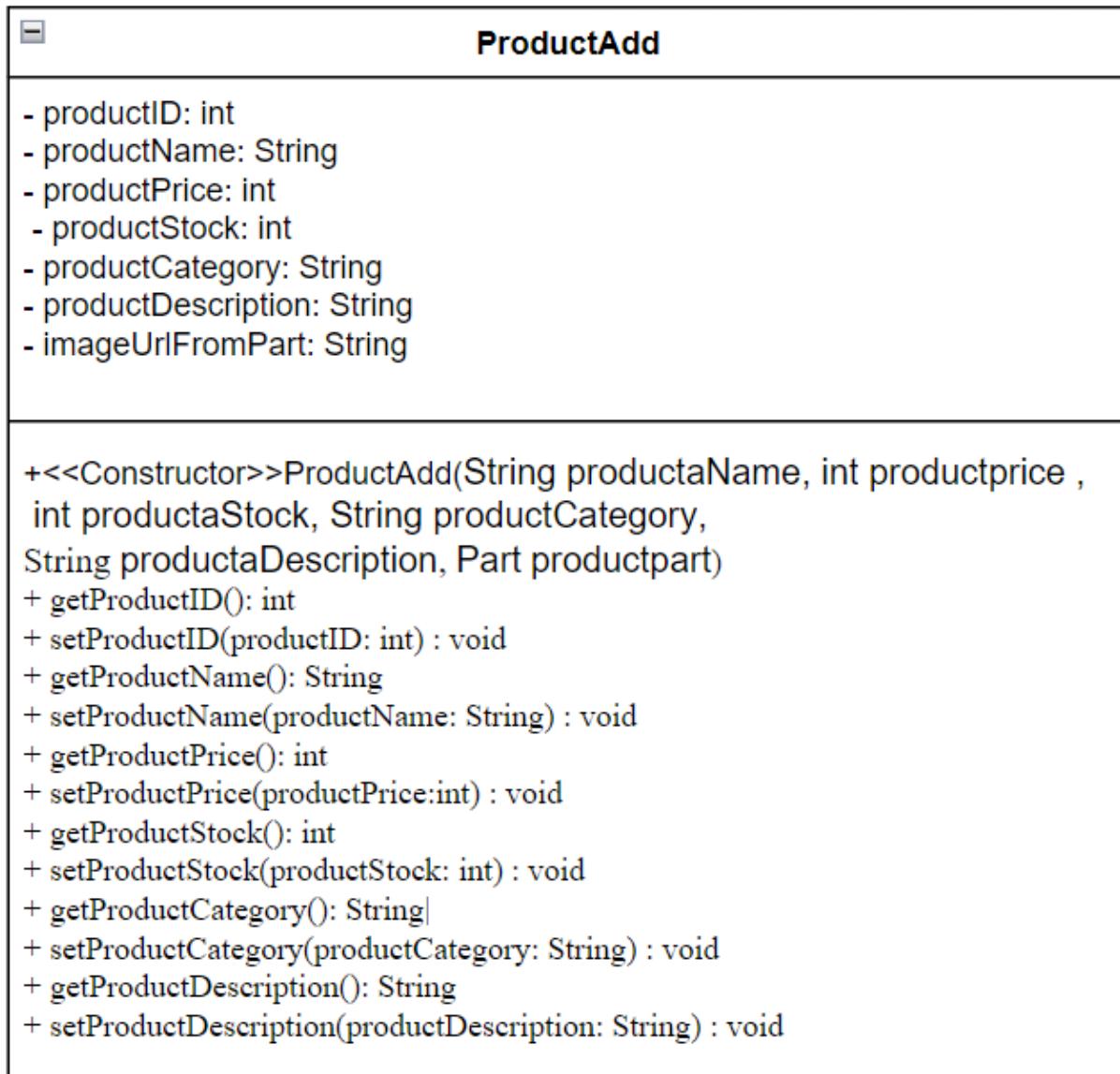


Figure 34: Class Diagram for ProductAdd.java

### 3.1.2. Individual class diagram of Util package:

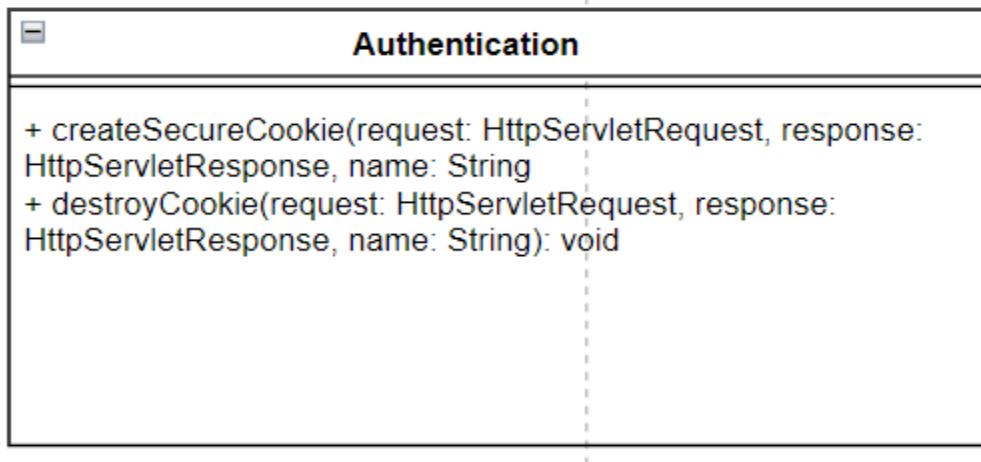


Figure 35: Class Diagram for *Authentication.java*

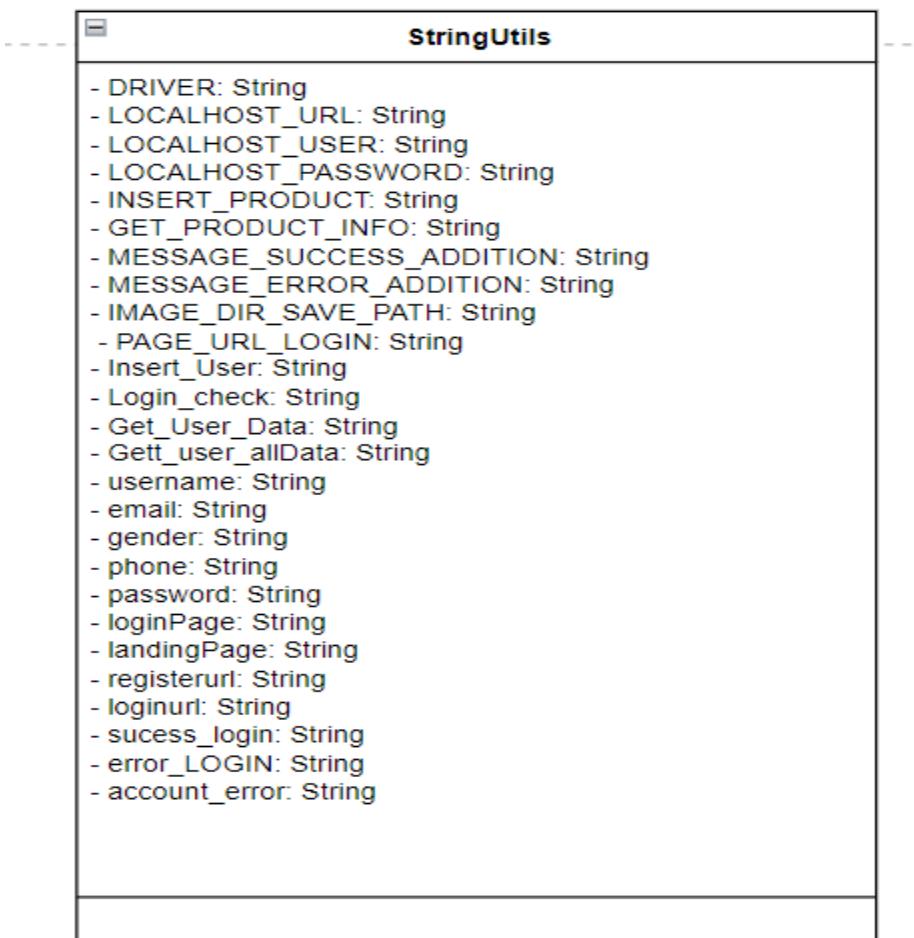


Figure 36: Class Diagram for *StringUtils.java*

### 3.1.3. Individual class diagram of controller.servlet package:

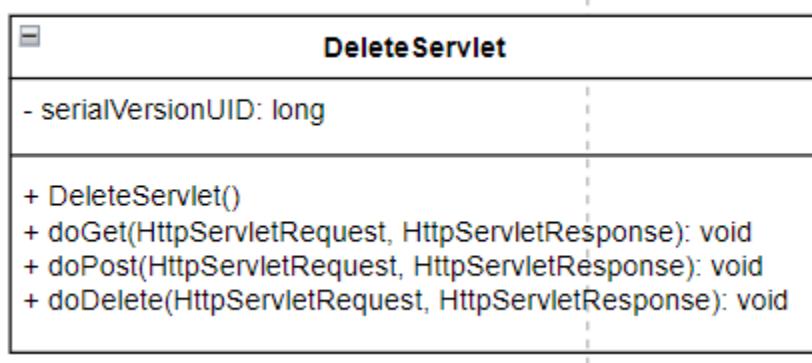


Figure 37: Class diagram for *DeleteServlet.java*

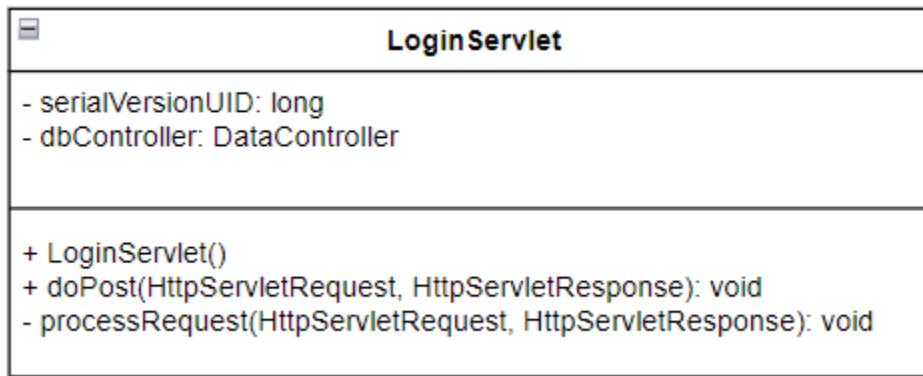


Figure 38: Class Diagram for *LoginServlet.java*

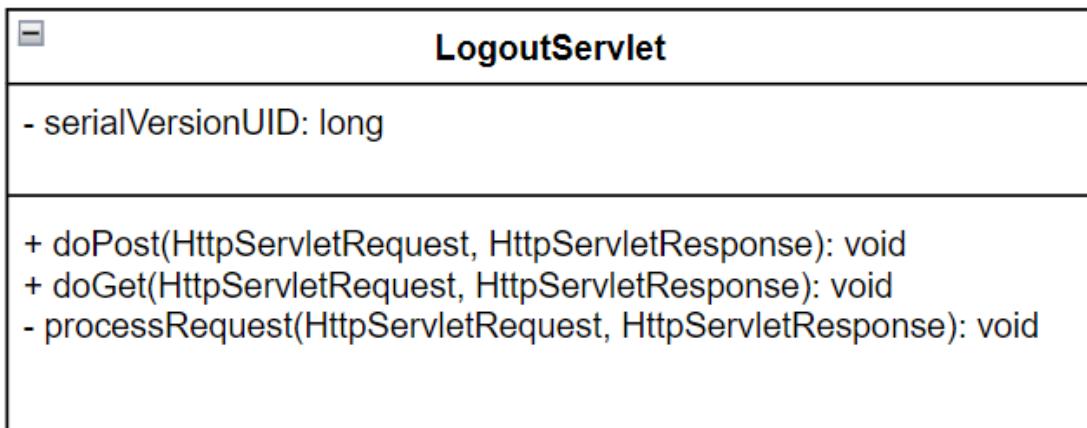


Figure 39: Class Diagram for *LogoutServlet.java*

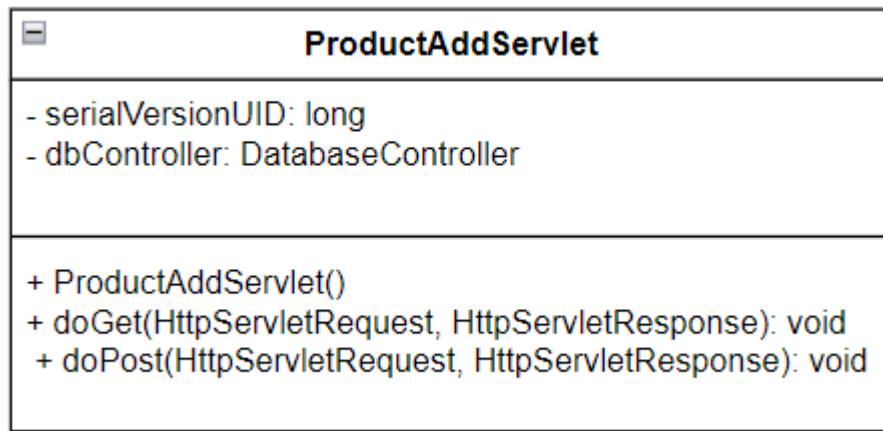


Figure 40: Class Diagram for `ProductAddServlet.java`

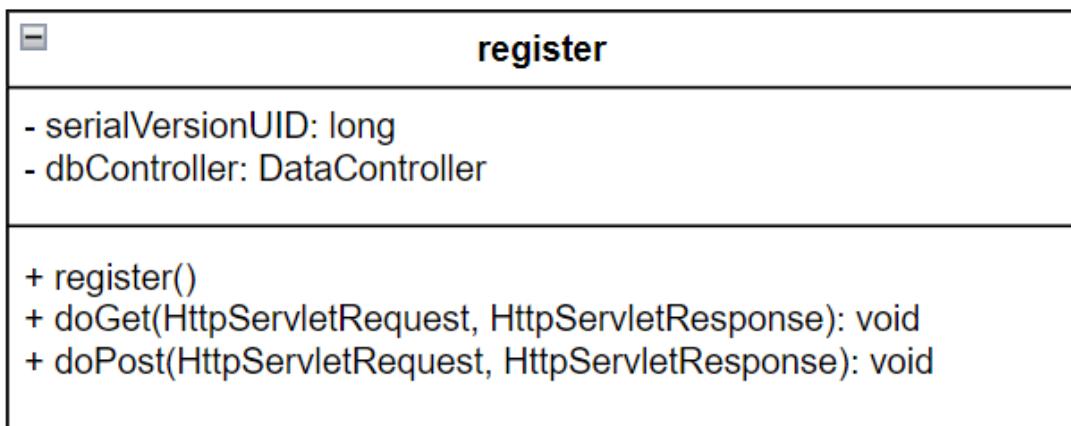


Figure 41: Class Diagram for `register.java`

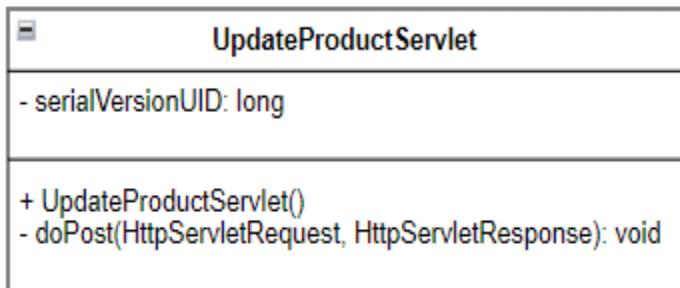


Figure 42: Class Diagram for `UpdateProductServlet.java`

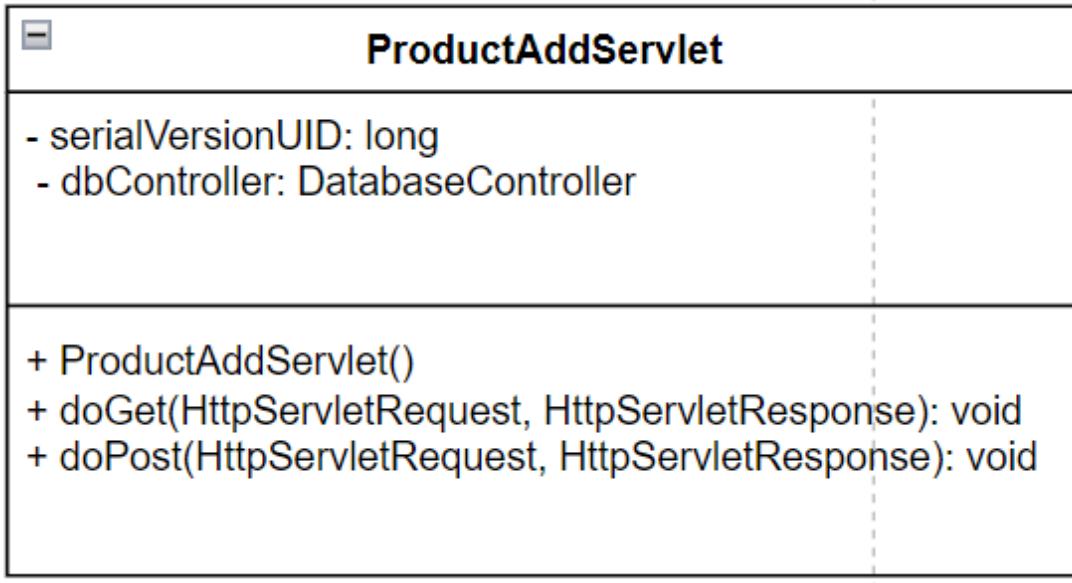


Figure 43: Class Diagram for *ProductAddServlet.java*

### 3.2. Class Diagram for Overall System:

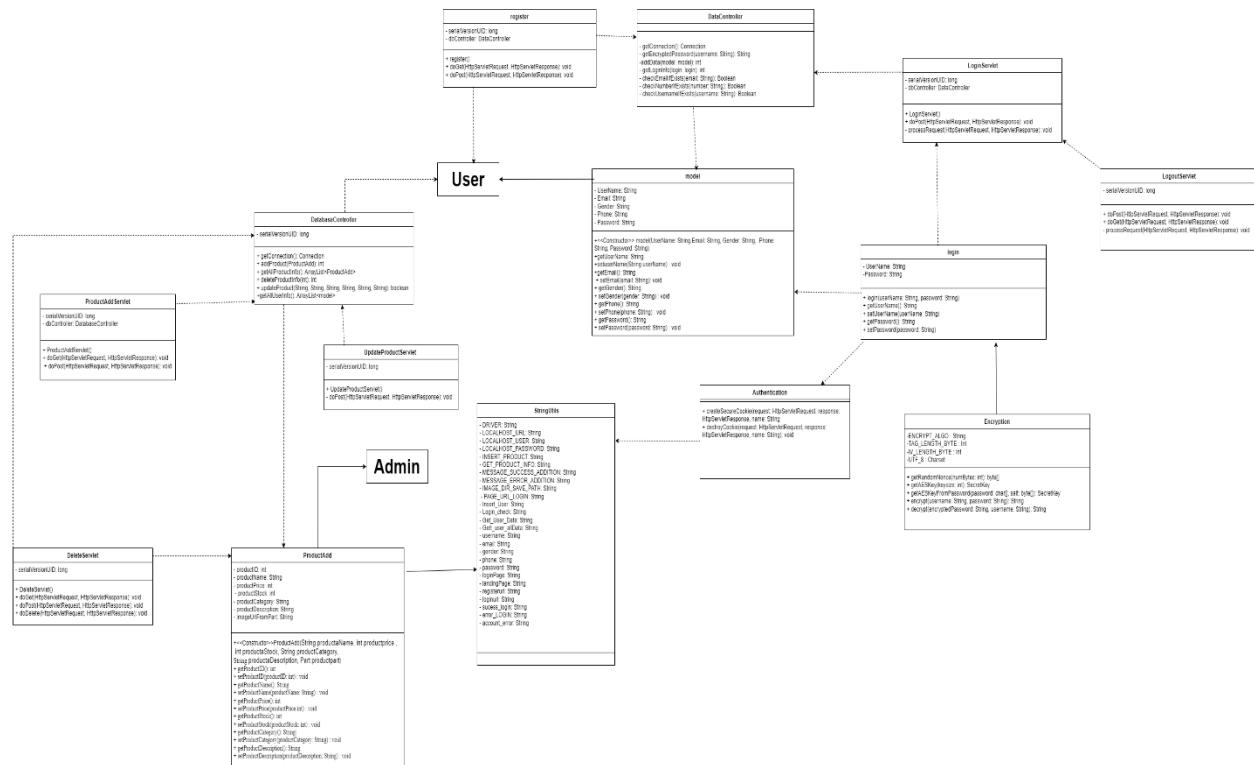


Figure 44: Class Diagram for Overall System

Note: As the image is blurry, please use the given link for better diagram quality ([Overall Class Diagram](#))

#### 4. Method Description:

The methods that are used to complete this website are described below:

##### 4.1. **doGet():**

This method, `doGet(HttpServletRequest request, HttpServletResponse response)`, is a protected method that overrides the `doGet()` method from the `HttpServlet` class. It does not return any value (void return type) and takes two parameters: `HttpServletRequest request` and `HttpServletResponse response`. Within this method, it delegates the request handling to the `doDelete()` method by invoking it with the same `HttpServletRequest` and `HttpServletResponse` objects passed to the `doGet()` method. This method serves as a convenient way to handle HTTP GET requests by reusing the logic implemented in the `doDelete()` method. It is typically used in servlets to handle GET requests by forwarding them to the appropriate handling method.

##### 4.2. **doPost():**

This method, `doPost(HttpServletRequest request, HttpServletResponse response)`, is a protected method that overrides the `doPost()` method from the `HttpServlet` class. It does not return any value (void return type) and takes two parameters: `HttpServletRequest request` and `HttpServletResponse response`. Within this method, it delegates the request handling to the `doGet()` method by invoking it with the same `HttpServletRequest` and `HttpServletResponse` objects passed to the `doPost()` method. This approach is commonly used to handle HTTP POST requests by reusing the logic implemented in the `doGet()` method. It ensures that both POST and GET requests are processed consistently within the servlet, promoting code reuse and maintainability.

##### 4.3. **doDelete():**

This method, `doDelete(HttpServletRequest request, HttpServletResponse response)`, is a protected method that overrides the `doDelete()` method from the `HttpServlet` class. It does not return any value (void return type) and takes two parameters: `HttpServletRequest request` and `HttpServletResponse response`.

#### **4.4. processRequest():**

The `processRequest(HttpServletRequest request, HttpServletResponse response)` method is a private method that typically encapsulates the main logic for handling HTTP requests within a servlet. It does not return any value (void return type) and takes two parameters: `HttpServletRequest request` and `HttpServletResponse response`.

#### **4.5. addProduct():**

This `addProduct(ProductAdd productAdd)` method appears to be a Java method within a class responsible for adding a new product to a database. The method is declared as public, indicating that it can be accessed from other classes. It returns an integer (int) value, which likely indicates the success or failure of the operation. The method takes a single parameter `ProductAdd productAdd`, which is an instance of the `ProductAdd` class containing information about the product to be added. The method initializes a String variable named `query` with a SQL query template stored in a constant (`StringUtils.INSERT_PRODUCT`). This query inserts a new product into a database table.

#### **4.6. getAllProductInfo():**

The `getAllProductInfo()` method is a public Java method that retrieves information about all products stored in a database and returns them as an `ArrayList` of `ProductAdd` objects. The method is declared as public, indicating that it can be accessed from other classes. It returns an `ArrayList<ProductAdd>`, which contains instances of the `ProductAdd` class representing product information. The method does not take any parameters. Inside a try block, the method prepares a SQL query for retrieving product details from the database. The query template is obtained from a constant (`StringUtils.GET_PRODUCT_DETAILS`).

#### **4.7. deleteProduct():**

The ``deleteProductInfo(int productID)`` method is a public Java method designed to delete product information from a database based on the provided product ID. The method is declared as `'public'`, indicating that it can be accessed from other classes. It returns an integer ('int'), which likely represents the number of rows affected by the deletion operation. The method takes an `'int'` parameter `'productID'`, which specifies the

ID of the product to be deleted. Inside a try-with-resources block, the method establishes a connection to the database using the `getConnection()` method (presumably defined elsewhere in the class). The connection is automatically closed after use due to the try-with-resources block. A `PreparedStatement` named `st` is created from the connection, and a SQL delete query template stored in a constant (`StringUtil.DELETE\_PRODUCT\_SQL`) is prepared. This query likely deletes a product from the database based on its ID.

#### **4.8. updateProduct():**

The `updateProduct` method is a static Java method responsible for updating product information in a database. The method is declared as `public static`, indicating that it can be accessed from other classes without requiring an instance of the class. It returns a boolean value indicating whether the update operation was successful. The method takes several string parameters (`productID`, `productName`, `productPrice`, `productStock`, `productCategory`, `productDescription`) representing the updated product information. Inside a try-with-resources block, the method establishes a connection to the database using the `DriverManager.getConnection()` method. The connection string, username, and password are obtained from constants defined in the `StringUtil` class (`StringUtil.LOCALHOST\_URL`, `StringUtil.LOCALHOST\_USER`, `StringUtil.LOCALHOST\_PASSWORD`). A `PreparedStatement` named `stmt` is created from the connection, and a SQL update query template stored in a constant (`StringUtil.UPDATE\_PRODUCT`) is prepared. This query updates a product's information in the database based on its ID.

#### **4.9. getAllUserInfo():**

The `getAllUserInfo()` method is a public Java method designed to retrieve information about all registered users from a database and return them as an `ArrayList` of model objects. The method is declared as public, indicating that it can be accessed from other classes. It returns an `ArrayList<model>`, which contains instances of the model class representing user information. The method does not take any parameters. Inside a try block, the method prepares a SQL query for retrieving user details from the database. The query template is obtained from a constant (StringUtil.GET\_USER).

#### **4.10. addData():**

The addData(model model) method is a public Java method responsible for adding user data to a database. The method is declared as public, indicating that it can be accessed from other classes. It returns an integer (int), which likely represents the success or failure of the operation. The method takes a parameter of type model, which represents the user data to be added. Inside a try-with-resources block, the method establishes a connection to the database using the getConnection() method (presumably defined elsewhere in the class). The connection is automatically closed after use due to the try-with-resources block. A PreparedStatement named st is created from the connection, and a SQL insert query template stored in a constant (StringUtils.Insert\_User) is prepared. This query inserts user data into the database.

#### **4.11. getLoginInfo():**

The getLoginInfo(login login) method is a Java method designed to check the login credentials of a user by querying a database. The method is declared as public, indicating that it can be accessed from other classes. It returns an integer (int), which represents the result of the login check. The method takes a parameter of type login, which likely contains the username and password entered by the user. Inside a try block, the method establishes a connection to the database using the getConnection() method (presumably defined elsewhere in the class). A PreparedStatement named st is created from the connection, and a SQL query template stored in a constant (StringUtils.Login\_check) is prepared. This query retrieves the username and password from the database based on the provided username.

#### **4.12. checkEmailExists():**

This method, checkEmailIfExists(String email), is designed to verify if a provided email address already exists within the database. It aims to facilitate user validation by querying the database for the presence of the specified email. Upon invocation, this method initiates a database query through a controller class (DBController), searching for matching records associated with the provided email address. If any corresponding records are found, the method returns true, indicating the existence of the email within the database. Conversely, if no matching records are found, it returns false.

Access Modifier: public

Return Type: Boolean

Parameters: email: A String representing the email address to be validated against the database.

#### **4.13. checkNumberIfExists():**

The checkNumberIfExists(String number) method is a Java utility intended to verify the presence of a provided phone number within a database. It offers functionality to ascertain whether a given phone number is already associated with an existing user record, aiding in user registration and validation processes. Upon invocation, this method executes a database query through a designated controller class (DBController or equivalent), searching for records that match the provided phone number. The method returns a Boolean value indicating the existence (true) or absence (false) of the specified phone number in the database.

Access Modifier: public

Return Type: Boolean

Parameters: number: A String representing the phone number to be verified against the database records.

#### **4.14. checkUserNamelfExists():**

The checkUsernamelfExists(String username) method is a Java utility designed to validate the existence of a provided username within a database. Its purpose is to determine whether the specified username is already associated with an existing user account, aiding in user registration and authentication processes. Upon invocation, this method executes a database query through a designated controller class (DBController or equivalent), searching for records that match the provided username. The method returns a Boolean value indicating whether the username exists (true) or not (false) in the database.

Access Modifier: public

Return Type: Boolean

Parameters:

username: A String representing the username to be verified against the database records.

#### **4.15. `createSecureCookies()`:**

The `createSecureCookie(HttpServletRequest request, HttpServletResponse response, String name, String value)` method is a Java utility function crafted to generate and set secure cookies within a web application context. Its primary objective is to establish cookies with enhanced security measures, ensuring data confidentiality and integrity during transmission over HTTPS connections. Upon invocation, this method constructs a new cookie object encapsulating the provided name-value pair, configures it with secure attributes, and sets an expiration time of 30 minutes. Additionally, the method orchestrates the creation or retrieval of an associated session, setting its maximum inactive interval to 30 minutes to synchronize with the cookie expiration time.

Access Modifier: public static

Return Type: void

Parameters:

request: A `HttpServletRequest` object representing the incoming HTTP request.

response: A `HttpServletResponse` object representing the outgoing HTTP response.

name: A String containing the name of the cookie.

value: A String containing the value of the cookie.

#### **4.16. `destroyCookies()`:**

The `destroyCookie(HttpServletRequest request, HttpServletResponse response, String name)` method is a Java utility function designed to invalidate and remove a specific cookie from the client's browser. Its primary objective is to revoke the existence of a

previously set cookie, thereby discontinuing its association with subsequent HTTP requests. Upon invocation, this method iterates through the existing cookies within the HTTP request, searching for a cookie matching the specified name. If a matching cookie is found, its expiration time is set to zero, effectively rendering it invalid. Subsequently, the method updates the cookie in the HTTP response to reflect the expiration change, ensuring its removal from the client's browser.

Access Modifier: public static

Return Type: void

Parameters:

request: A HttpServletRequest object representing the incoming HTTP request.

response: A HttpServletResponse object representing the outgoing HTTP response.

name: A String containing the name of the cookie to be destroyed.

## 5. Test Cases:

Testing is the process of checking for different error, performance or logic issues and bugs in a software. Testing process is very important before releasing a software or website because if a system has different types of bug and errors it will not perform good in the market and develops frustration among the user when using the application full of error. It involves execution of software components using manual or automated tools to evaluate one or more functionality of the respective software. The purpose of testing is to check for different errors and fix them if there are any errors. Java testing can be done at different levels such as unit testing, integration testing, system testing, and acceptance testing (IBM, 2024).

### 5.1. Test Case 1:

*Table 1: To register in the system and store data onto the database and using the same data to login in the website*

Test.no	1
Objective	To register in the system and store data onto the database and using the same data to login in the website
Arguments called	Attributes Called ❖ Username: Saman21 ❖ Email: saman21@gmail.com ❖ Gender: Male ❖ Phone:123456789 ❖ Password:1234
Expected Result	The entered Data should be stored in the database and upon login in the same data should be used to login

Actual Result	The data was successfully stored in database and the same data was used to login in the system
Testing Result	Successful

Registration :

User Name:  
Saman21

Email:  
saman21@gmail.com

Gender:  Male  Female

Phone:  
123456789

Password  
....

Submit

Already a Member ? [Login Here](#)

Figure 45: Entering necessary credentials

<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	Saman21	saman21@gmail.com	male	123456789	1234	NULL
--------------------------	-------------------------------------	-------------------------------------	---------------------------------------	----	---------	-------------------	------	-----------	------	------

Figure 46: Successfully registered and saved in database

## 5.2. Test Case 2:

Table 2: To login to the website using user-based correct credential.

Test.no	2
Objective	To login to the website using user-based correct credential.
Arguments called	Attributes Called ❖ Username: Saman21 ❖ Password:1234
Expected Result	After pressing login button, it will redirect the user to main.jsp page.
Actual Result	After pressing login button, it was redirected to the user to main.jsp page.
Testing Result	Successful

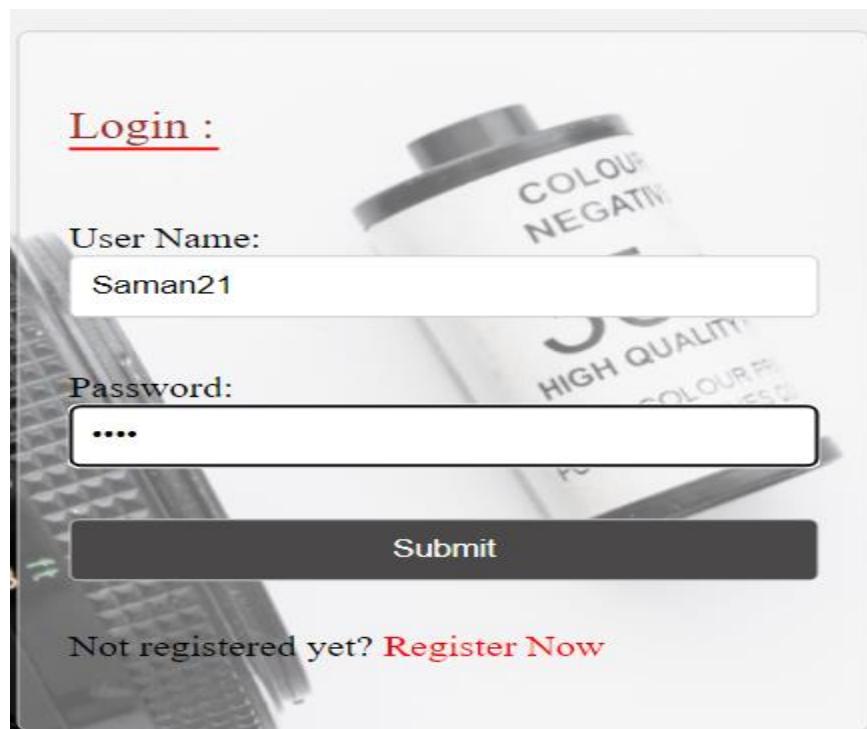
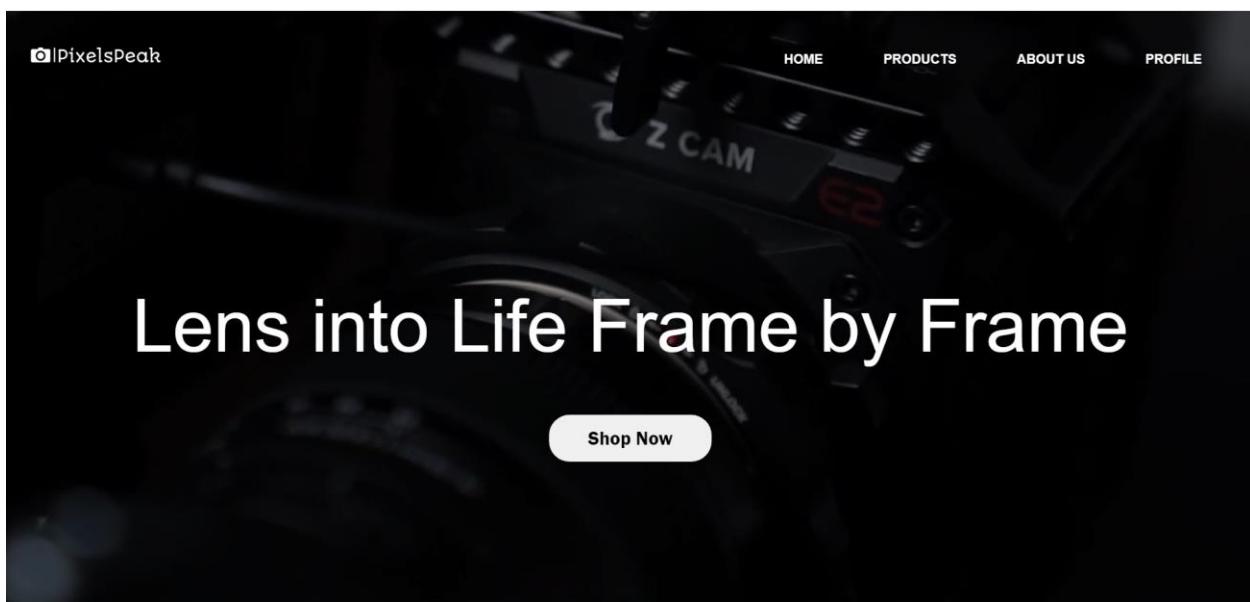


Figure 47: Entering correct credential for user login



*Figure 48: Redirected to Home page*

### 5.3. Test Case 3:

Table 3: To test if user is able to edit their profile or not.

<b>Test No:</b>	<b>3</b>
Objective:	To test if a user is able to edit their profile or not.
Action:	<p>Different credentials which were to be changed was given in the respective input boxes.</p> <ul style="list-style-type: none"> <li>➤ UserName: XnezOptimus</li> <li>➤ Email: Saman123@gmail.com</li> <li>➤ Phone: 9861841694</li> </ul> <p>After filling up all these information update buttons was clicked to add update all this information in database.</p>
Expected Result:	User information will be changed in database.
Actual Result:	User information was changed in database.

<b>Database User Table</b>								
	id	UserName	Email	Gender	Phone	Password	UserImage	
<input type="checkbox"/>	7	saman	sam@gmail.com	male	0987654321	saman123	NULL	<a href="#"> Edit</a> <a href="#"> Copy</a> <a href="#"> Delete</a>

Figure 49: Database user before changing user information

<b>Database User Table</b>								
	id	UserName	Email	Gender	Phone	Password	UserImage	
<input type="checkbox"/>	7	XnezOptimus	Saman123@gmail.com	male	9861841694	saman123	NULL	<a href="#"> Edit</a> <a href="#"> Copy</a> <a href="#"> Delete</a>

Figure 50: Database of user after changing user information

#### 5.4. Test Case 4:

Table 4: To check if the cookie and session is created or not after a login is successful

Test No:	4
Objective:	To check if the cookie and session is created or not after a login is successful.
Action:	Inspecting “Application” by using inspect option in the google chrome.
Expected Result:	Cookie information with username “XnezOptimus” would be formed.
Actual Result:	Cookie information with username “XnezOptimus” was formed.

Name	Value	Do...	P...	E...	Si...	H...	S...	S...	P...	Pr...
JSESSIONID	CE9FE3E934932...	loc...	/...	S...	42	✓				M...
xsrftoken	213629f8d512920...	loc...	/	2...	59					M...
username	XnezOptimus	loc...	/...	2...	19		✓			M...
username-localhost	"2 1:0 10:171465...	loc...	/	2...	184	✓				M...

Figure 51: Cookie with username "XnezOptimus" was formed

### 5.5. Test Case 5:

Table 5: To test if cookie is being destroyed or not after logging out or not.

Test No:	5
Objective:	To test if cookie is being destroyed or not after logging out or not.
Action:	Logout button was clicked in profile page to logout of the website.
Expected Result:	Cookie named "userName" would be cleared after pressing logout.
Actual Result:	Cookie named "userName" was cleared after pressing logout.

Name	Value	Do...	P...	E...	Si...	H...	S...	S...	P...	Pr...
JSESSIONID	CE9FE3E934932...	loc...	/...	S...	42	✓				M...
_xsrf	2 3629f8d5 2920...	loc...	/	2...	59					M...
username	XnezOptimus	loc...	/...	2...	19		✓			M...
username-localho...	"2 1:0 10:171465...	loc...	/	2...	184	✓				M...

Figure 52: Username "XnezOptimus" saved while logged into the website

Name	Value	Do...	P...	E...	Si...	H...	S...	S...	P...	Pr...
JSESSIONID	DD889D2AA131...	loc...	/...	S...	42	✓				M...
_xsrf	2 3629f8d5 2920...	loc...	/	2...	59					M...
username-localho...	"2 1:0 10:171465...	loc...	/	2...	184	✓				M...

Figure 53: Username cleared after pressing Logout

### 5.6. Test Case 6:

Table 6: To test if admin-based login is working or not.

Test No:	6
Objective:	To test if admin-based login is working or not.
Action:	Login credential for admin was given where following were the correct credentials: <ul style="list-style-type: none"> <li>• UserName: Admin</li> <li>• Password: Admin123</li> </ul>
Expected Result:	The admin would get redirected to admin side pages and will not be able to see user sided pages.
Actual Result:	The admin was redirected to admin side pages and was not be able to see user sided pages.

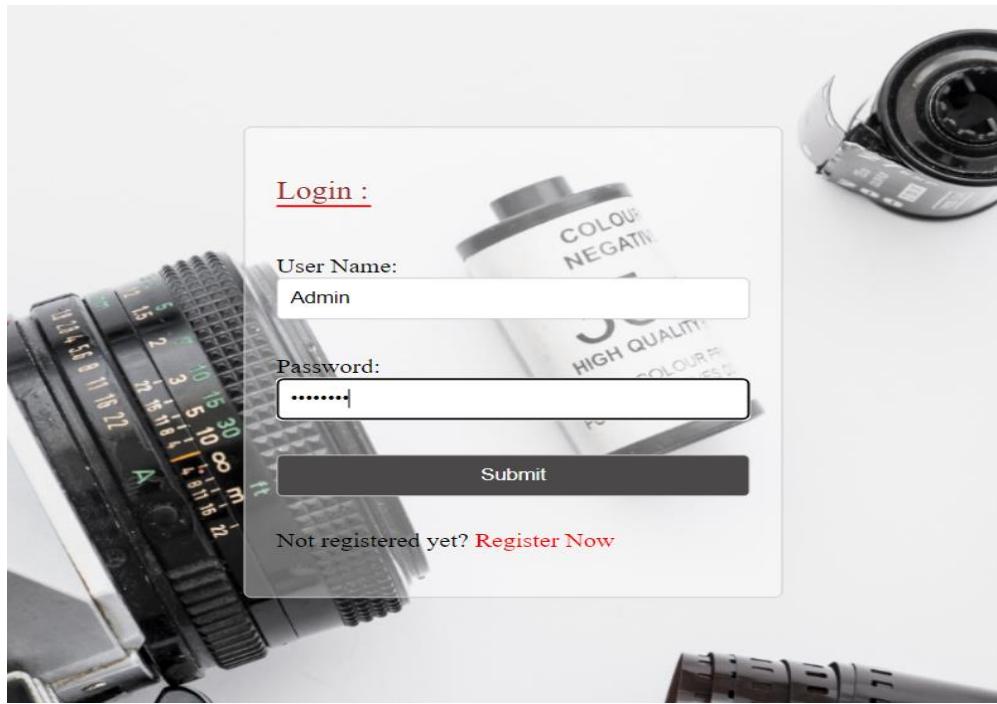


Figure 54: Correct credential for admin login.

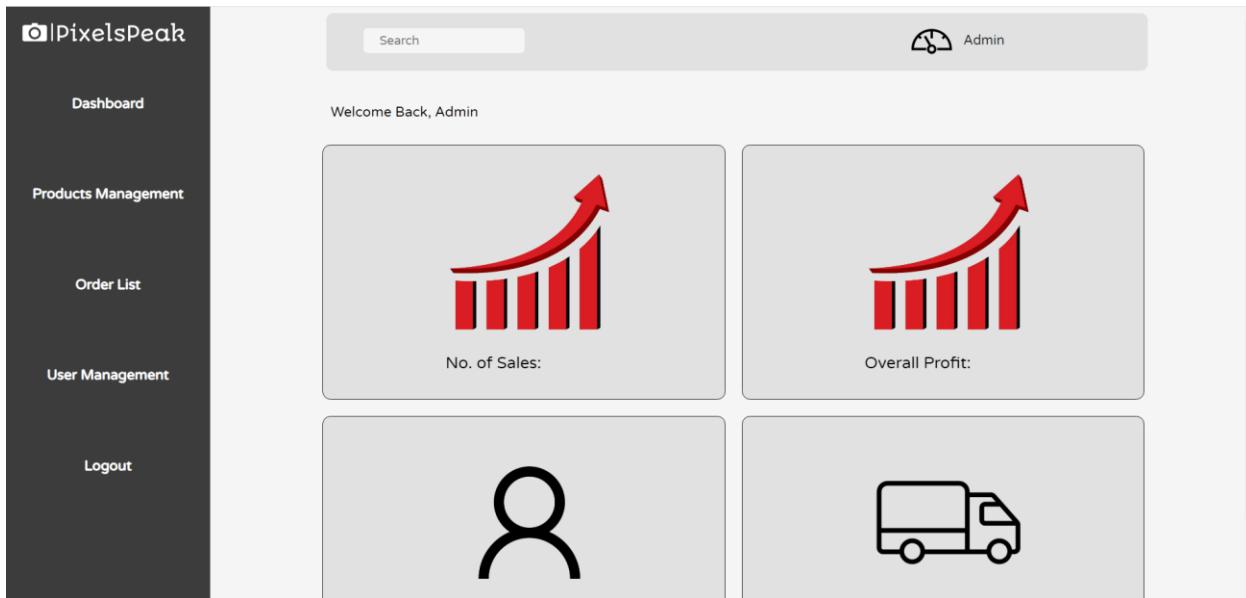


Figure 55: Admin Sided Pages

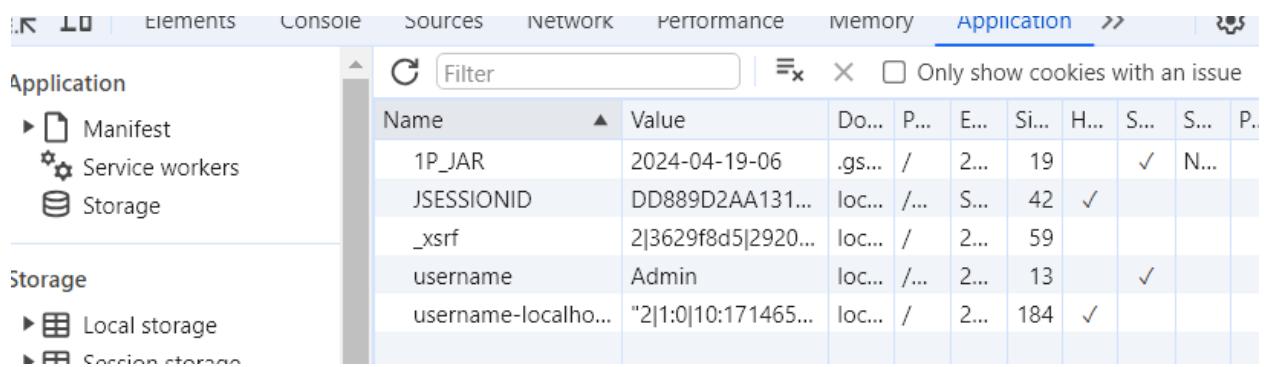


Figure 56: Cookie created with username "Admin"

### 5.7. Test Case 7:

Table 7: To test if product is being added or not by using “Add Product” link.

Test No:	7
Objective:	To test if product is being added or not by using “Add Product” link.
Action:	Following credentials were given which was to be added in the database. <ul style="list-style-type: none"> <li>• UserName: Admin</li> <li>• Password: Admin123</li> </ul>
Expected Result:	The admin would get redirected to admin side pages and will not be able to see user sided pages.
Actual Result:	The admin was redirected to admin side pages and was not be able to see user sided pages.

		24.2 MP	0.14	1.0 Camera	Digital S...	24.2 MP .png
<input type="checkbox"/>	Edit  Copy  Delete	65 Sony E Mount FE 50mm F1.8	299	2 Camera Lens	Large F1.8 maximum aperture enables beautiful defo...	Sony E Mount FE 50mm F1.8.jpeg
<input type="checkbox"/>	Edit  Copy  Delete	66 Sony FX6 Cinema Line	5998	9 Professional Camera	10.2MP 4K full-frame back-illuminated Exmor R CMOS...	Sony FX6 Cinema Line.png
<input type="checkbox"/> Check all  Edit  Copy  Delete  Export						
<input type="checkbox"/>	Show all	Number of rows: 25	Filter rows:	Search this table	Sort by key:	None

Figure 57: Database before adding another product.

**Product Name**

Test Case (Camera)

**Product Price**

1000

**Product Stock**

19

**Product Category**

Camera

**Product Description**

This is a test case addition of data in the database

Figure 58: Values that were to be added in database

<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	65 F1.8	299	2 Camera Lens	beautiful defo...	F1.8.jpeg
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	66 Sony FX6 Cinema Line	5998	9 Professional Camera	10.2MP 4K full-frame back-illuminated Exmor R CMOS...	Sony FX6 Cinema Line .png
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	67 Test Case (Camera)	1000	19 Camera	This is a test case addition of data in the database...	download.png

Check all  With selected:

Figure 59: Database after addition of product

## 5.8. Test Case 8:

Table 8: To if a product is being deleted or not after clicking delete button in productmanagement.jsp.

Test No:	8
Objective:	To if a product is being deleted or not after clicking delete button in productmanagement.jsp.
Action:	Clicking “Delete” button of the product that needs to be deleted from the database in the productmanagement.jsp.
Expected Result:	After pressing delete button the respective product will be deleted from database.
Actual Result:	After pressing delete button the respective product was deleted from database.



Figure 60: Database "product" before deleting productID "67"

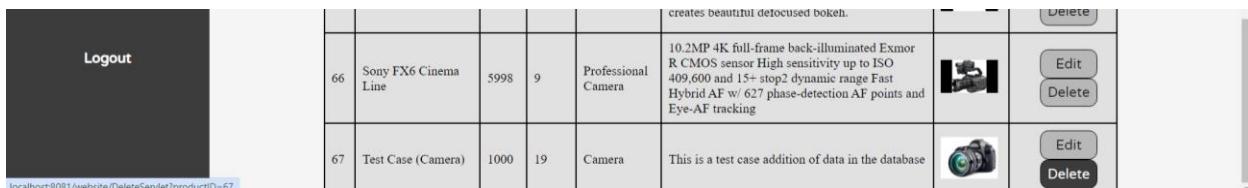


Figure 61: Pressing delete button for productID "67"



Figure 62: Database after deleting productID "67"

### 5.9. Test Case 9:

Table 9: To test if error is being thrown or not when using incorrect credential for login.

Test No:	9
Objective:	To test if error is being thrown or not when using incorrect credential for login.
Action:	Wrong credential for login which did not exist in database was given.
Expected Result:	After trying to login with wrong credential user will be redirected to error page.
Actual Result:	After trying to login with wrong credential user was redirected to error page.

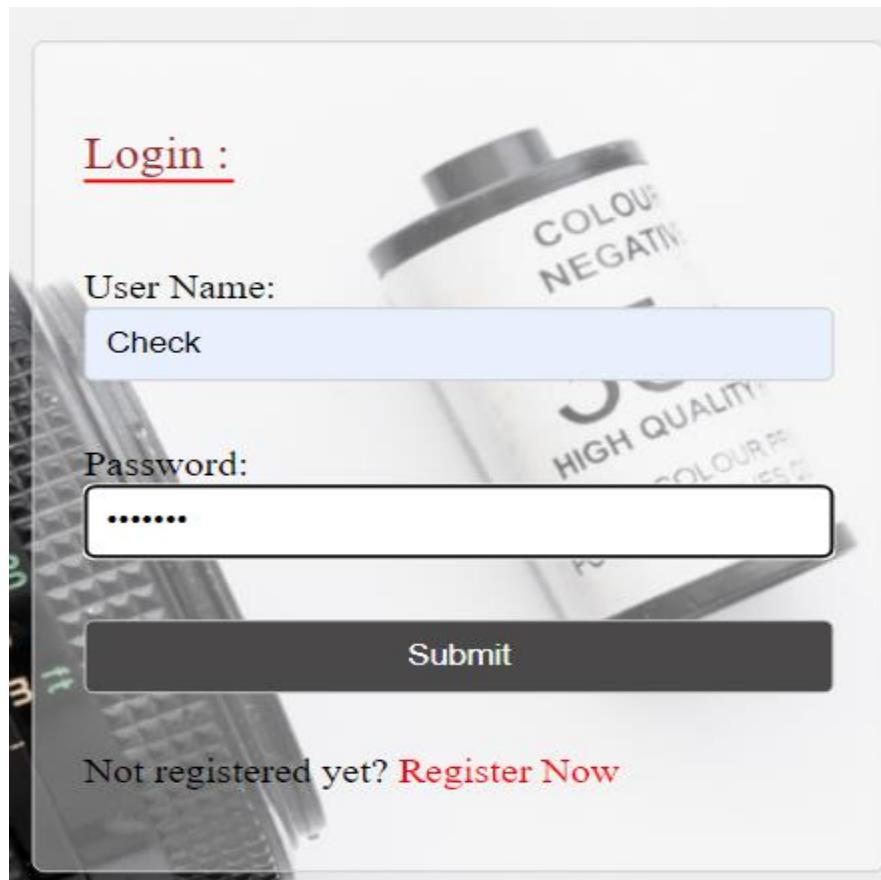


Figure 63: Giving wrong credential for login

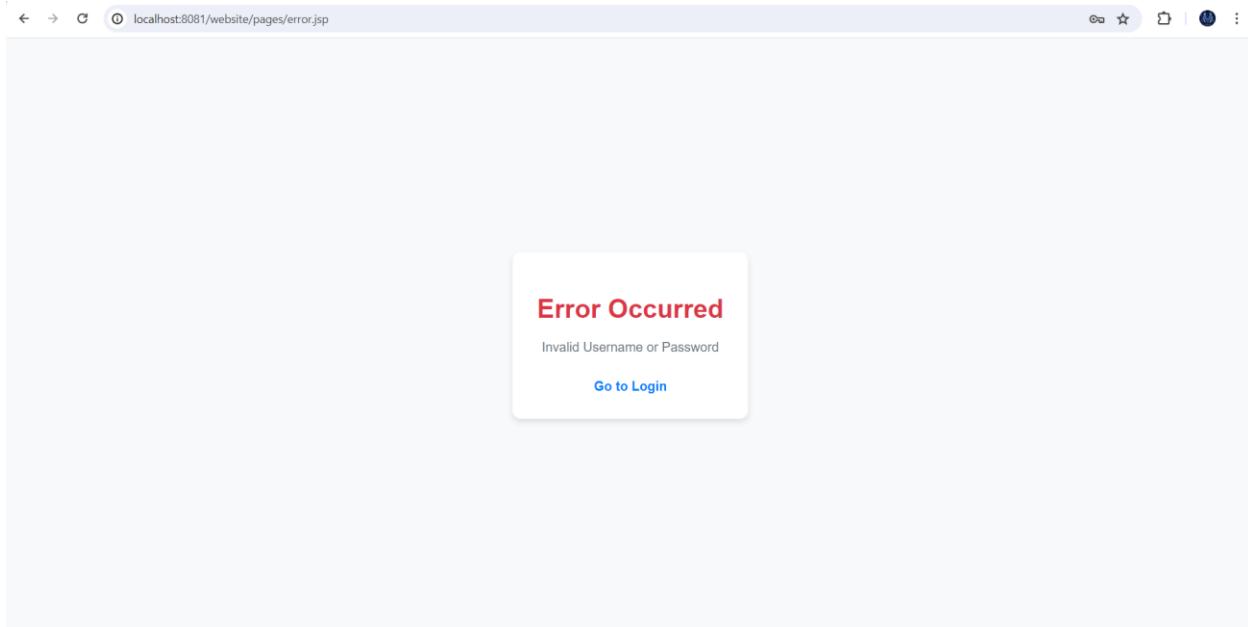
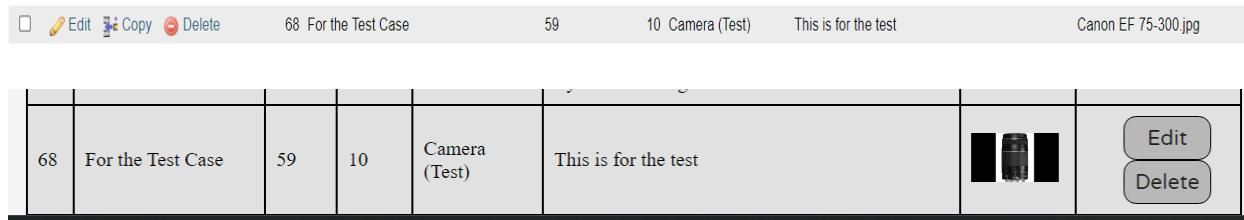


Figure 64: Redirected to error.jsp page

### 5.10. Test Case 10:

Table 10: To test if admin is able to update product data which already exists in database.

<b>Test No:</b>	<b>10</b>
Objective:	To test if admin is able to update product data which already exists in database.
Action:	Necessary data was given in updateproduct.jsp form to update the data of the respective product in the database.
Expected Result:	The data of the product which already existed in the database will be changed.
Actual Result:	The data of the product which already existed in the database was changed.



The screenshot shows a software interface with a toolbar at the top containing icons for Edit, Copy, and Delete. Below the toolbar is a table with six columns. The first column contains the ID '68'. The second column contains the product name 'For the Test Case'. The third column contains the quantity '59'. The fourth column contains the price '10'. The fifth column contains the category 'Camera (Test)'. The sixth column contains the description 'This is for the test'. To the right of the table is a preview image of a Canon EF 75-300.jpg file, which appears to be a camera lens. There are also 'Edit' and 'Delete' buttons next to the preview image.

68	For the Test Case	59	10	Camera (Test)	This is for the test	 <b>Edit</b> <b>Delete</b>

Figure 65: Data of the product which already exist in the database

**Product ID**  
68

**Product Name**

Changing for Test Case

**Product Price**

89

**Product Stock**

12

**Product Category**

Gimble (Test)

**Product Description**

This is being changed

Figure 66: Changing information of the product by using updateproduct form

68	Changing for Test Case	89	12	Gimble (Test)	This is being changed		<input type="button" value="Edit"/> <input type="button" value="Delete"/>
----	------------------------	----	----	---------------	-----------------------	---	---

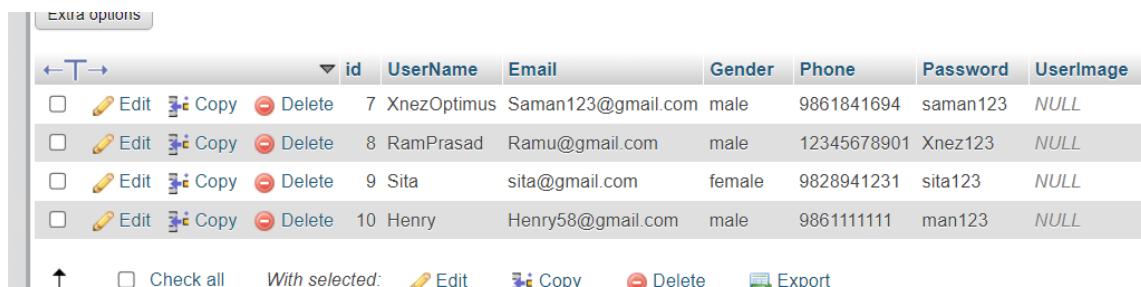


Figure 67: Information for the already existing product is changed.

### 5.11. Test Case 11:

Table 11: To test if user detail is being displayed or not for user management.

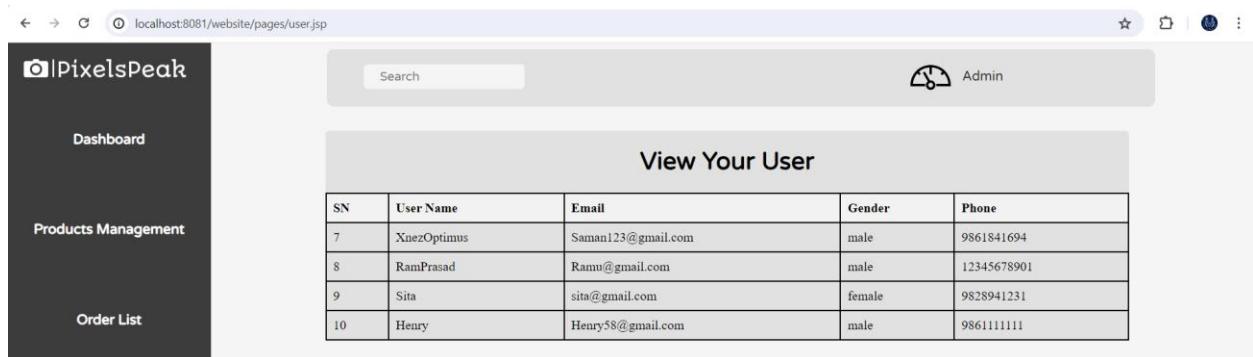
<b>Test No:</b>	11
Objective:	To test if user detail is being displayed or not for user management.
Action:	User.jsp was selected.
Expected Result:	All the values that exist in register database would be displayed in user.jsp page.
Actual Result:	All the values that existed in register database was displayed in user.jsp page.



<input type="checkbox"/>				7	XnezOptimus	Saman123@gmail.com	male	9861841694	saman123
<input type="checkbox"/>				8	RamPrasad	Ramu@gmail.com	male	12345678901	Xnez123
<input type="checkbox"/>				9	Sita	sita@gmail.com	female	9828941231	sita123
<input type="checkbox"/>				10	Henry	Henry58@gmail.com	male	9861111111	man123

↑     Check all    With selected:    Edit    Copy    Delete    Export

Figure 68: Data in "register" database



SN	User Name	Email	Gender	Phone
7	XnezOptimus	Saman123@gmail.com	male	9861841694
8	RamPrasad	Ramu@gmail.com	male	12345678901
9	Sita	sita@gmail.com	female	9828941231
10	Henry	Henry58@gmail.com	male	9861111111

Figure 69: User list in user.jsp being displayed

**5.12. Test Case 12:***Table 12: To test if product details is being displayed or not in product management.*

Test No:	12
Objective:	To test if product details is being displayed or not in product management
Action:	Productmanagement.jsp was selected.
Expected Result:	All the values that exist in product database would be displayed in productmanagement.jsp page.
Actual Result:	All the values that existed in product database was displayed in productmanagement.jsp page.

	productID	productName	productPrice	productStock	productCategory	productDescription	imagePart
<b>Click the drop-down arrow to toggle column's visibility.</b>							
<input type="checkbox"/>	42	Tartisans AutoFocus 50mm F1.8	228	10	Camera Lens	Support Automatic/Manual Aperture and Face/Eye Rec...	Tartisans AutoFocus 50mm F1.8.jpeg
<input type="checkbox"/>   	43	Action Pro Head Strap Mount	45	15	Helmet Mount	Action Pro Head Strap Mount and Screw Compatible w...	Action Pro Head Strap Mount.png
<input type="checkbox"/>   	44	Adofys Professional	78	3	Tripods	Adofys Flexible Gorillapod Tripod with 360° Rotati...	Adofys Professional.png
<input type="checkbox"/>   	45	Sony Tripod	57	27	Tripods	AmazonBasics WT3130T+WT3111H 50-Inch Lightweight T...	Amazon Basics 3110.png
<input type="checkbox"/>   	46	Canon EF 75-300mm	199	15	Camera Lens	Canon EF 75-300mm f/4-5.6 III Telephoto Zoom Lens ...	Canon EF 75-300.jpg
<input type="checkbox"/>   	47	Canon EOS 200D	623	7	Camera	Weighing only a little heavier than a bottle of wa...	Canon EOS 200D II.png
<input type="checkbox"/>   	48	Canon EOS R100	615	2	Camera	Capture memories effortlessly with the Canon EOS R...	Canon EOS R100.png
<input type="checkbox"/>   	49	Canon RF100mm F2.8L	999	2	Camera Lens	Maximum Macro Magnification of 1.4x Spherical Abe...	Canon RF100mm F2.8L.jpeg
<input type="checkbox"/>   	50	Canon XF705	8999	4	Professional Camera	The XF705 Professional Camcorder offers an outstan...	Canon XF705.png
<input type="checkbox"/>   	52	DTR 550 Foldable Tripod Stand	36	10	Tripods	DIGITEK (DTR 550 LW) 67 Inch Foldable Tripod Stand...	DIGITECK.png
<input type="checkbox"/>   	53	ELECTROPRIME 1pc	20	52	PCB	ELECTROPRIME 1PC TEC1-7106 TEC1 Thermolectric Hea...	ELECTROPRIME 1pc.png
<input type="checkbox"/>   	54	Fotopro X-Go HR Chameleon	95	12	Tripods	Fotopro X-Go HR Chameleon 6ft Aluminum Built in 2...	Fotopro X-Go HR Chameleon.png
<input type="checkbox"/>   	55	HUMBLE Head Strap Mount	99	9	Helmet Mount	MADE OF DURABLE NYLON MATERIAL WITH ANTI-SLIDE DES...	HUMBLE Head Strap Mount.png
<input type="checkbox"/>   	50	Canon XF705	8999	4	Professional Camera	The XF705 Professional Camcorder offers an outstan...	Canon XF705.png
<input type="checkbox"/>   	52	DTR 550 Foldable Tripod Stand	36	10	Tripods	DIGITEK (DTR 550 LW) 67 Inch Foldable Tripod Stand...	DIGITECK.png
<input type="checkbox"/>   	53	ELECTROPRIME 1pc	20	52	PCB	ELECTROPRIME 1PC TEC1-7106 TEC1 Thermolectric Hea...	ELECTROPRIME 1pc.png
<input type="checkbox"/>   	54	Fotopro X-Go HR Chameleon	95	12	Tripods	Fotopro X-Go HR Chameleon 6ft Aluminum Built in 2...	Fotopro X-Go HR Chameleon.png
<input type="checkbox"/>   	55	HUMBLE Head Strap Mount	99	9	Helmet Mount	MADE OF DURABLE NYLON MATERIAL WITH ANTI-SLIDE DES...	HUMBLE Head Strap Mount.png
<input type="checkbox"/>   	56	LEQTRONIQ 4K HD	227	7	Professional Camera	This video camera 4k has a new autofocus function...	LEQTRONIQ 4K HD.png
<input type="checkbox"/>   	57	Nikon D7500	899	2	Camera	Follow your passion wherever it leads with the sam...	Nikon D7500.png
<input type="checkbox"/>   	58	Osaka OS 550 Tripod 55	21	23	Tripods	Osaka OS 550 Tripod 55 Inches (140 cm) with Mobile...	Osaka OS 550 Tripod 55.png
<input type="checkbox"/>   	59	Panasonic AG-CX8ED	17955	1	Professional Camera	Industry's Smallest and Lightest 4K 60p Director's...	Panasonic AG-CX8ED.png
<input type="checkbox"/>   	60	Panasonic LUMIX G7	497	17	Camera	Panasonic LUMIX G7 4K Digital Camera, with LUMIX G...	Panasonic LUMIX G7.png
<input type="checkbox"/>   	61	SHOPEE Head Strap Mount	121	6	Head Strap	SHOPEE Head Strap Mount Compatible with GoPro GoPr...	SHOPEE Head Strap Mount.png
<input type="checkbox"/>   	62	Sigma 30 mm f1.4 DC DN	262	21	Camera Lens	Combining outstanding optical performance and com...	Sigma 30 mm f1.4 DC DN.jpg
<input type="checkbox"/>   	63	Sony Alpha ILCE-6100Y 24.2 MP	874	15	Camera	Sony Alpha ILCE-6100Y 24.2 MP Mirrorless Digital S...	Sony Alpha ILCE-6100Y 24.2 MP.png
<input type="checkbox"/>   	65	Sony E Mount FE 50mm F1.8	299	2	Camera Lens	Large F1.8 maximum aperture enables beautiful defo...	Sony E Mount FE 50mm F1.8.jpeg

Figure 70: Data that exist in the database

48	Canon EOS R100	615	2	Camera	Capture memories effortlessly with the Canon EOS R100 Mirrorless Camera. Perfect for beginners and casual snapshots, paired with the versatile RF-S18-45mm f/4.5-6.3 IS STM Lens.		<button>Edit</button> <button>Delete</button>
49	Canon RF100mm F2.8L	999	2	Camera Lens	Maximum Macro Magnification of 1.4x Spherical Aberration Control for Varying Bokeh Control Image Stabilisation of up to 8 stops		<button>Edit</button> <button>Delete</button>
50	Canon XF705	8999	4	Professional Camera	The XF705 Professional Camcorder offers an outstanding combination of 4K UHD image quality, features and functionality in a compact, portable design ideally suited for news gathering, outdoor events.		<button>Edit</button> <button>Delete</button>
52	DTR 550 Foldable Tripod Stand	36	10	Tripods	DIGITEK (DTR 550 LW) 67 Inch Foldable Tripod Stand with Phone Holder		<button>Edit</button> <button>Delete</button>
53	ELECTROPRIME 1pc	20	52	PCB	ELECTROPRIME 1PC TEC1-7106 TEC1 Thermoelectric Heatsink Cooler Peltier Plate Module. ELECTROPRIME 1pc Protection BMS PCB for 48/54.6V 13S 18650 Li-Ion Li-Polymer Battery 15-45A.		<button>Edit</button> <button>Delete</button>
54	Fotopro X-Go HR Chameleon	95	12	Tripods	Fotopro X-Go HR Chameleon 6ft Aluminum Built in 2-in-1 Tripod + Monopod Stand for DSLR Camera   FPH-52Q Ball Head   360°Horizontal Rotation   Upside Down Tripod   Payload 8 kg (Black)		<button>Edit</button> <button>Delete</button>
55	HUMBLE Head Strap Mount	99	9	Helmet Mount	MADE OF DURABLE NYLON MATERIAL WITH ANTI-SLIDE DESIGN: Light weight, portable action camera headband, easy to take around.		<button>Edit</button> <button>Delete</button>

SN	Product Name	Price	Stock Qty.	Product Category	Product Description	Product Image	Actions
42	7artisans AutoFocus 50mm F1.8	228	10	Camera Lens	Support Automatic/Manual Aperture and Face/Eye Recognition, Compatible for Sony A7 Series A7R A7S A7SII A9 A7C A1		<button>Edit</button> <button>Delete</button>
43	Action Pro Head Strap Mount	45	15	Helmet Mount	Action Pro Head Strap Mount and Screw Compatible with All GoPro Hero Series SJCAM Xiaomi YI Eken Action Camera		<button>Edit</button> <button>Delete</button>
44	Adofys Professional	78	3	Tripods	Adofys Flexible Gorillapod Tripod with 360° Rotating Ball Head Tripod for All DSLR Cameras (Max Load 1.5 kgs) & Mobile Phones + Free Heavy Duty Mobile Holder		<button>Edit</button> <button>Delete</button>
45	Sony Tripod	57	27	Tripods	AmazonBasics WT3130T+WT3111H 50-Inch Lightweight Tripod with Bag, Load Capacity: 2 Kg		<button>Edit</button> <button>Delete</button>
46	Canon EF 75-300mm	199	15	Camera Lens	Canon EF 75-300mm f/4-5.6 III Telephoto Zoom Lens for Canon SLR Cameras. 75-300 millimeter telephoto zoom lens with f 4-5.6 maximum aperture for Canon SLR cameras		<button>Edit</button> <button>Delete</button>
47	Canon EOS 200D	623	7	Camera	Weighing only a little heavier than a bottle of water*, the camera slides right into your bag for that everyday photography.		<button>Edit</button> <button>Delete</button>

57	Nikon D7500	899	2	Camera	Follow your passion wherever it leads with the same image quality, ISO range, image processing and energy efficiency of the award-winning D500 in an enthusiast-level DSLR.		<button>Edit</button> <button>Delete</button>
58	Osaka OS 550 Tripod 55	21	23	Tripods	Osaka OS 550 Tripod 55 Inches (140 cm) with Mobile Holder and Carry Case for Smartphone & DSLR Camera Portable Lightweight Aluminium Tripod		<button>Edit</button> <button>Delete</button>
59	Panasonic AG-CX8ED	17955	1	Professional Camera	Industry's Smallest and Lightest 4K 60p Director's Camcorder Wide 25mm* and Optical 24x Zoom Lens 4K High-Precision AF Hybrid O.I.S		<button>Edit</button> <button>Delete</button>
60	Panasonic LUMIX G7	497	17	Camera	Panasonic LUMIX G7 4K Digital Camera, with LUMIX G VARIO 14-42mm Mega O.I.S. Lens, 16 Megapixel Mirrorless Camera, 3-Inch LCD, DMC-G7KK (Black)		<button>Edit</button> <button>Delete</button>
61	SHOPEE Head Strap Mount	121	6	Head Strap	SHOPEE Head Strap Mount Compatible with GoPro GoPro Hero 10, 9, 8, Hero 7 Black, 7 Silver, 7 White, Hero 6, 5, 4, Session, 3+, 3, 2, 1, Hero, Fusion, Action Cameras		<button>Edit</button> <button>Delete</button>
62	Sigma 30 mm f1.4 DC DN	262	21	Camera Lens	Combining outstanding optical performance and compact design Introducing a new F1.4 standard lens for mirrorless cameras		<button>Edit</button> <button>Delete</button>
63	Sony Alpha ILCE-6100Y 24.2 MP	874	15	Camera	Sony Alpha ILCE-6100Y 24.2 MP Mirrorless Digital SLR Camera with 16-50 mm		<button>Edit</button> <button>Delete</button>

Figure 71: All the data being pulled from database

### 5.13. Test Case 13:

*Table 13: To test if product details is being displayed or not in product management.*

Test No:	13
Objective:	To test if product details is pulled and displayed in a product card in proper way in product page.
Action:	Product.jsp page was selected.
Expected Result:	All the values that exist in product database would be displayed in product page inside a product card one by one.
Actual Result:	All the values that existed in product database was displayed in product page inside a product card one by one.

**Sony Tripod**

\$57

Category: Tripods

**Stock :27**

AmazonBasics WT3130T+WT3111H 50-Inch Lightweight Tripod with Bag, Load Capacity: 2 Kg

[Add to Cart](#)
**Canon EF 75-300mm**

\$199

Category: Camera Lens

**Stock :15**

Canon EF 75-300mm f/4-5.6 III Telephoto Zoom Lens for Canon SLR Cameras. 75-300 millimeter telephoto zoom lens with f 4-5.6 maximum aperture for Canon SLR cameras

[Add to Cart](#)
**Canon EOS 200D**

\$623

Category: Camera

**Stock :7**

Weighing only a little heavier than a bottle of water\*, the camera slides right into your bag for that everyday photography.

[Add to Cart](#)
**Canon EOS R100**

\$615

Category: Camera

**Stock :2**

Capture memories effortlessly with the Canon EOS R100 Mirrorless Camera. Perfect for beginners and casual snapshots, paired with the versatile RF-S18-45mm f/4.5-6.3 IS STM Lens.

[Add to Cart](#)
**Tartisans AutoFocus 50mm F1.8**

\$228

Category: Camera Lens

**Stock :10**

Support Automatic/Manual Aperture and Face/Eye Recognition, Compatible for Sony A7 Series A7R A75 A7SII A9 A7C A1

[Add to Cart](#)
**Action Pro Head Strap Mount**

\$45

Category: Helmet Mount

**Stock :15**

Action Pro Head Strap Mount and Screw Compatible with All GoPro Hero Series SJCAM Xiaomi YI Eken Action Camera

[Add to Cart](#)
**Adofys Professional**

\$78

Category: Tripods

**Stock :3**

Adofys Flexible Gorillapod Tripod with 360° Rotating Ball Head Tripod for All DSLR Cameras (Max Load 1.5 kgs) & Mobile Phones + Free Heavy Duty Mobile Holder

[Add to Cart](#)
**Fotopro X-Go HR Chameleon**

\$95

Category: Tripods

**Stock :12**

Fotopro X-Go HR Chameleon 6ft Aluminum Built-in 2-in-1 Tripod + Monopod Stand for DSLR Camera | FPH-S2Q Ball Head | 360° Horizontal Rotation | Upside Down Tripod | Payload 8 kg (Black)

[Add to Cart](#)
**HUMBLE Head Strap Mount**

\$99

Category: Helmet Mount

**Stock :9**

MADE OF DURABLE NYLON MATERIAL WITH ANTI-SLIDE DESIGN: Light weight, portable action camera headband, easy to take around.

[Add to Cart](#)
**LEQTROLIQ 4K HD**

\$227

Category: Professional Camera

**Stock :7**

This video camera 4k has a new autofocus function, it can automatically focus and clear imaging, the shooting distance is 3.94inch ~ infinity.

[Add to Cart](#)
**Nikon D7500**

\$899

Category: Camera

**Stock :2**

Follow your passion wherever it leads with the same image quality, ISO range, image processing and energy efficiency of the award-winning D500 in an enthusiast-level DSLR.

[Add to Cart](#)

*Figure 72: All the product details being pulled inside a product card in a managed way*

### 5.14. Test Case 14:

*Table 14: To check if a user can register using same username and email.*

Test.no	14
Objective	To check if a user can register using same username and email.
Arguments called	Attributes Called ❖ Username: Saman21 ❖ Email: saman21@gmail.com ❖ Gender: Male ❖ Phone:123456789 ❖ Password:1234
Objective	To demonstrate the error handling done after the user enters the a user or email that already exist ion the database
Expected Result	A custom error message is thrown by the system
Actual Result	A message is successfully thrown indication error message or username
Testing Result	Successful

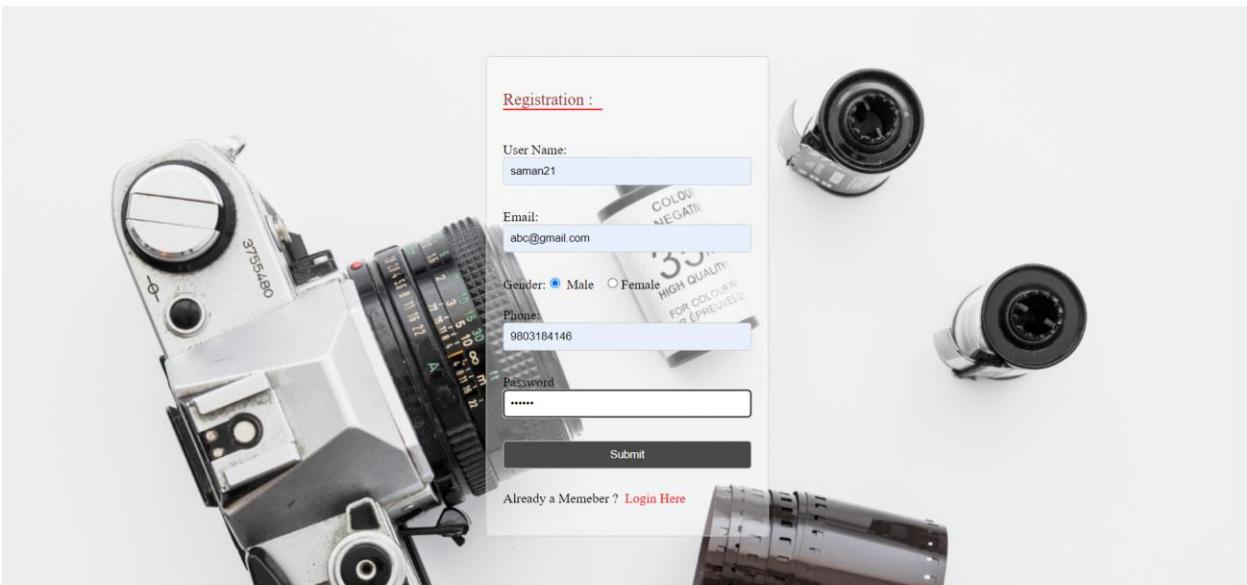


Figure 73: Using same data to register in the website

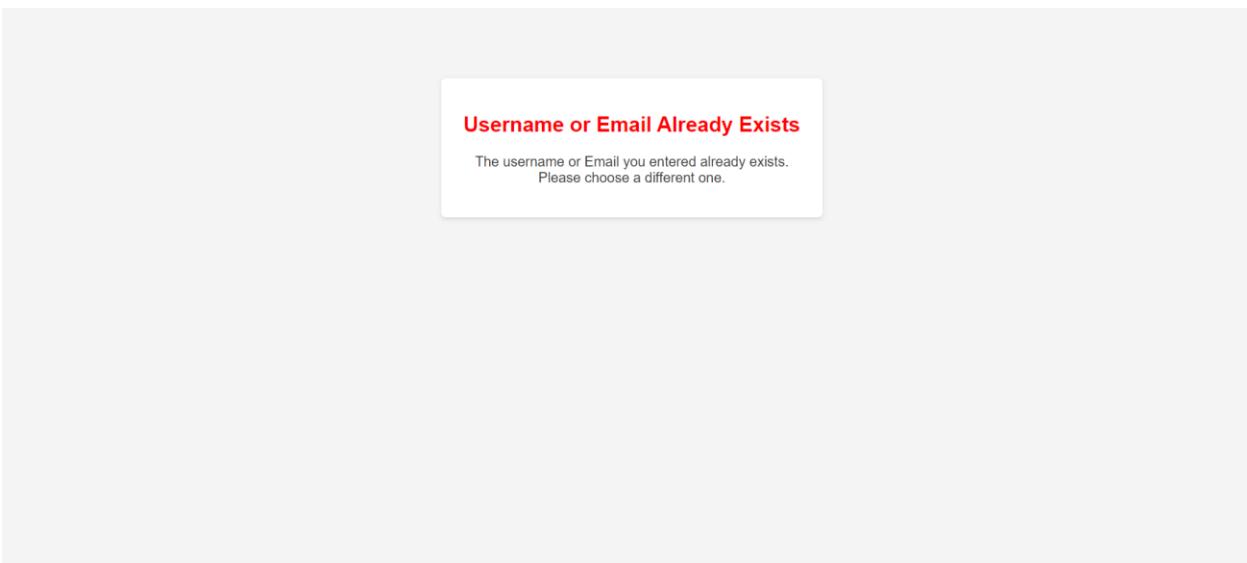


Figure 74: Error message pop up after using same data to login

### 5.15. Test Case 15:

Table 15: To test if button “Shop Now” in home page is working or not.

<b>Test No:</b>	15
Objective:	To test if button “Shop Now” in home page is working or not.
Action:	“Shop Now” button which is in Home page was clicked.
Expected Result:	The button will redirect the user to product page.
Actual Result:	The button redirected the user to product page.

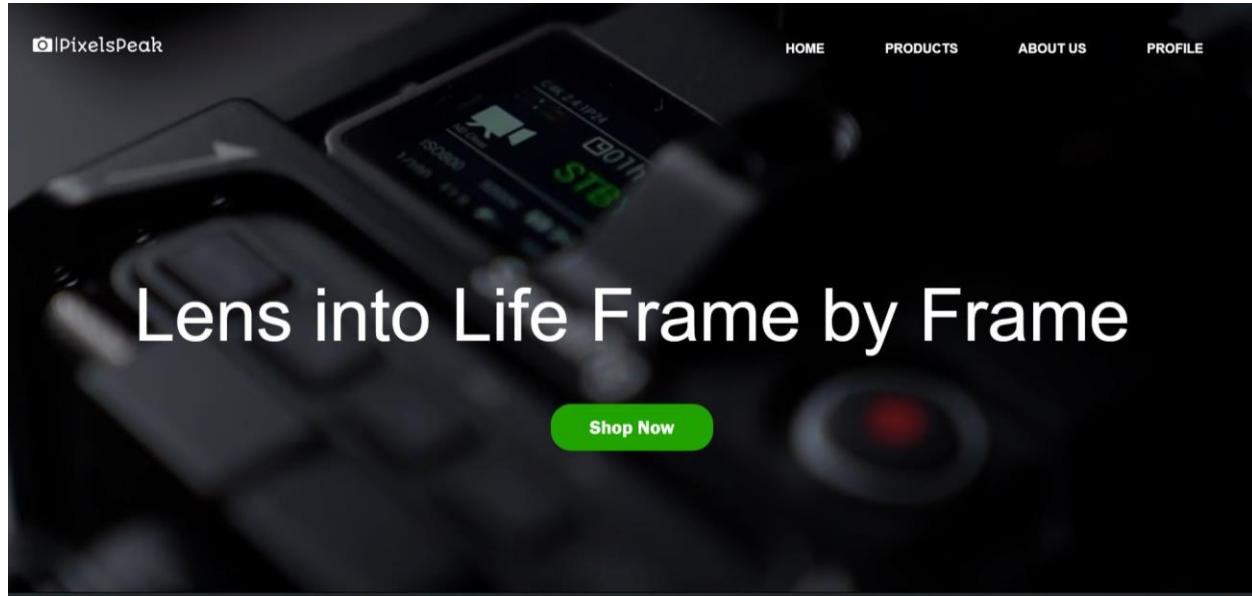


Figure 75: Clicking "Shop Now" button in home page

## Browse Products

**Price**

To

Apply



**7artisans AutoFocus 50mm F1.8**  
\$228  
Category: Camera Lens  
**Stock :10**  
Support Automatic/Manual Aperture and Face/Eye Recognition, Compatible for Sony A7 Series A7R A7S A7SII A9 A7C A1



**Action Pro Head Strap Mount**  
\$45  
Category: Helmet Mount  
**Stock :15**  
Action Pro Head Strap Mount and Screw Compatible with All GoPro Hero Series SJCAM Xiaomi Yi Eken Action Camera



**Adofys Professional**  
\$78  
Category: Tripods  
**Stock :3**  
Adofys Flexible Gorillapod Tripod with 360° Rotating Ball Head Tripod for All DSLR Cameras (Max Load 1.5 kgs) & Mobile Phones + Free Heavy Duty Mobile Holder

Figure 76: Redirected to Product page

## 6. Tools and Libraries Used:

There were different tasks which was needed to be completed. For different, task we used different tools and libraries. The tools and libraries which were used to complete the given task are given below:

### 6.1. Tools Used:

#### 6.1.1. Eclipse:

Eclipse IDE (Integrated Development Environment) is a widely used open-source development platform primarily for java. But it doesn't only support java and supports a variety of other programming languages via plugins. It provides numerous tools for software development which includes editing, debugging, version control and building applications. Eclipse is a highly customizable application which allows developers to add or remove plugins based on their needs. It is known for its extensibility which means developers can create their own plugins to enhance the functionality of the IDE. One of the key features of Eclipse is plugins, which can be installed anytime and tasks like web development, mobile app development can be done. It even supports languages like Python, C/C++, and more. Eclipse has a large community of users and developers. The developers are actively improving Eclipse. Eclipse is widely used in both academia and industry for developing various types of software applications. (Eclipse, 2001)

#### 6.1.2. XAMPP:

XAMPP is the most popular PHP development environment XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open-source package has been set up to be incredibly easy to install and to use. (Apache, 2002)

It stands for X: Stands for "cross-platform". XAMPP is designed to be compatible with different operating systems, including Windows, Linux, and macOS. A: Stands for "Apache". Apache is one of the most popular web servers in the world, and it's included in the XAMPP package as the primary server component. M: Stands for "MySQL". MySQL is a widely used relational database management system (RDBMS), and it's included in XAMPP to facilitate database-driven web applications. P: Stands for "PHP". PHP is a

popular server-side scripting language used for web development. XAMPP includes PHP to enable the execution of dynamic web content. P: Stands for "Perl" (EDX, 2024).

#### **6.1.3. Figma:**

Figma is a collaborative web application for interface design, with additional offline features enabled by desktop applications for macOS and Windows. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilizing a variety of vector graphics editor and prototyping tools. The Figma mobile app for Android and iOS allows viewing and interacting with Figma prototypes in real-time on mobile and tablet devices (Figma, 2024).

Figma is a web-based design tool renowned for its intuitive interface, powerful collaboration features, and robust prototyping capabilities. Offering a seamless experience for designers and teams, Figma facilitates real-time collaboration on design projects, enabling multiple users to work simultaneously on the same file from anywhere with an internet connection. Its vector editing tools, component-based design approach, and interactive prototyping features empower designers to create stunning user interfaces and prototypes with ease.

#### **6.1.4. Balsamiq:**

Balsamiq Wireframes is a user interface design tool for creating wireframes (sometimes called mock-ups or low-fidelity prototypes). You can use it to generate digital sketches of your idea or concept for an application or website, to facilitate discussion and understanding before any code is written. The completed wireframes can be used for user testing, clarifying your vision, getting feedback from stakeholders, or getting approval to start development (Balsamiq, 2009). Balsamiq offers a vast library of pre-built UI components, such as buttons, menus, and text boxes, that users can drag and drop onto the canvas to create wireframes quickly. These components can be easily customized to match the desired look and feel of the final product (Balsamiq, 2024).

#### **6.1.5. MySQL-Connector-j-8.3.0:**

MySQL Connector/J 8.3.0 is a JDBC driver provided by MySQL for Java applications to connect with MySQL databases. It implements the JDBC API, allowing developers to use standard Java interfaces and classes for database interaction. With

Connector/J, Java applications can establish connections to MySQL servers, execute SQL queries, handle result sets, manage transactions, and handle errors efficiently. It offers performance optimizations like connection pooling and statement caching to enhance database operations' efficiency. Overall, MySQL Connector/J 8.3.0 provides a reliable and efficient means for Java developers to interact with MySQL databases, facilitating seamless integration of database functionality into Java applications (MySQL, 2024).

#### **6.1.6. MS Word:**

It is a software used to make professional-quality documents, letters, reports, etc., MS Word is a word processor developed by Microsoft. It has advanced features which allow you to format and edit your files and documents in the best possible way (Byjus, 2011). Microsoft Word is a word processing software developed by Microsoft. It's part of the Microsoft Office suite of productivity software. Word allows users to create, edit, format, and save documents. It offers a wide range of features for text formatting, such as fonts, styles, spacing, and alignment. Users can also insert images, tables, charts, and other elements into their documents (MS Word, 2024).

### **6.2. Libraries Used:**

#### **6.2.1. Taglibs-standard-impl-1.2.5:**

Taglibs-standard-impl-1.2.5 is a Java library that provides an implementation of the Java Standard Tag Library (JSTL). JSTL is a collection of custom tags for JSP (Java Server Pages) pages. These custom tags simplify the development of JSP pages by providing a way to perform common tasks such as iteration, conditional logic, and formatting without requiring embedded Java code within the JSP.

Some of the key features provided by taglibs-standard-impl-1.2.5 include:

- Core tags: These tags provide functionality for iteration, conditionals, and variable manipulation within JSP pages.
- Formatting tags: These tags allow for formatting of data such as dates, numbers, and currencies.

- SQL tags: These tags provide support for executing SQL queries directly within JSP pages.
- XML tags: These tags allow for manipulation of XML data within JSP pages.

#### **6.2.2. Taglibs-standard-spec-1.2.5:**

The taglibs-standard-spec-1.2.5 library is part of the Java Standard Tag Library (JSTL). Unlike the implementation libraries like taglibs-standard-impl-1.2.5, which contain the actual code to execute JSTL functionality, the taglibs-standard-spec-1.2.5 library contains only the specification or the API for JSTL. This library defines the interfaces for all the JSTL tags, specifying their attributes, behaviours, and relationships. These interfaces are used by developers to create custom tags or by framework developers to implement the standard tags.

#### **6.2.3. `java.io.IOException`:**

`java.io.IOException` is an exception class in Java that indicates an error or unexpected condition occurred while performing input or output operations. It's part of the `java.io` package, which provides classes for handling input and output operations, such as reading from or writing to files, streams, and other sources.

This exception is a checked exception, meaning that code using methods that can throw `IOException` must either catch it using a try-catch block or declare that it throws `IOException` itself.

#### **6.2.4. `HttpServletRequest`:**

`HttpServletRequest` is an interface in the Java Servlet API, which provides a standard way for Java servlets to interact with client-side HTTP requests. Servlets are Java classes that extend the capabilities of servers to respond to requests.

`HttpServletRequest` represents an HTTP request from a client to a server and provides methods to access various parts of the request, such as parameters, headers, cookies, and the request body. Servlet containers, like Apache Tomcat or Jetty, provide implementations of this interface.

#### **6.2.5. HttpServletResponse:**

`HttpServletResponse` is an interface in the Java Servlet API, which provides a standard way for Java servlets to generate HTTP responses to client-side requests. Servlets are Java classes that extend the capabilities of servers to respond to requests.

`HttpServletResponse` represents an HTTP response that the server sends back to the client and provides methods to set response headers, status codes, and content. Servlet containers, like Apache Tomcat or Jetty, provide implementations of this interface.

#### **6.2.6. ServletException:**

`ServletException` is an exception class in the Java Servlet API that indicates a servlet-specific problem during servlet execution. It's a checked exception, meaning that servlet methods which can throw `ServletException` must either catch it using a try-catch block or declare that they throw it themselves.

This exception typically wraps other exceptions thrown by servlets or servlet-related classes. It's commonly used to handle errors that occur during the processing of client requests within servlets.

## 7. Development:

### 7.1. Individual Contribution:

#### 7.1.1. Admin Panel - Saman Gautam (22068007):

The website contained a feature of admin panel from which an admin could control core functions of the website like product addition, user management, product update and deletion and many different things which helps an e-commerce website to run smoothly. The admin can only be used by users who have admin role or the person who logs in using admin's login credential. The admin is not allowed to view user sided pages but the user who have admin role can view admin dashboard.

There were a lot of problem faced when creating the admin panel because it was the most important feature of the e-commerce website. The most difficult part of this feature was to add the product in database and update the existing product. These two features took my most of the time of the coursework. When I was adding products in the database it always used to throw error about "Error Inserting data" because of a null value being passed. But because of the team coordination and help from my respected tutors I was able to complete this task given to me by my team.

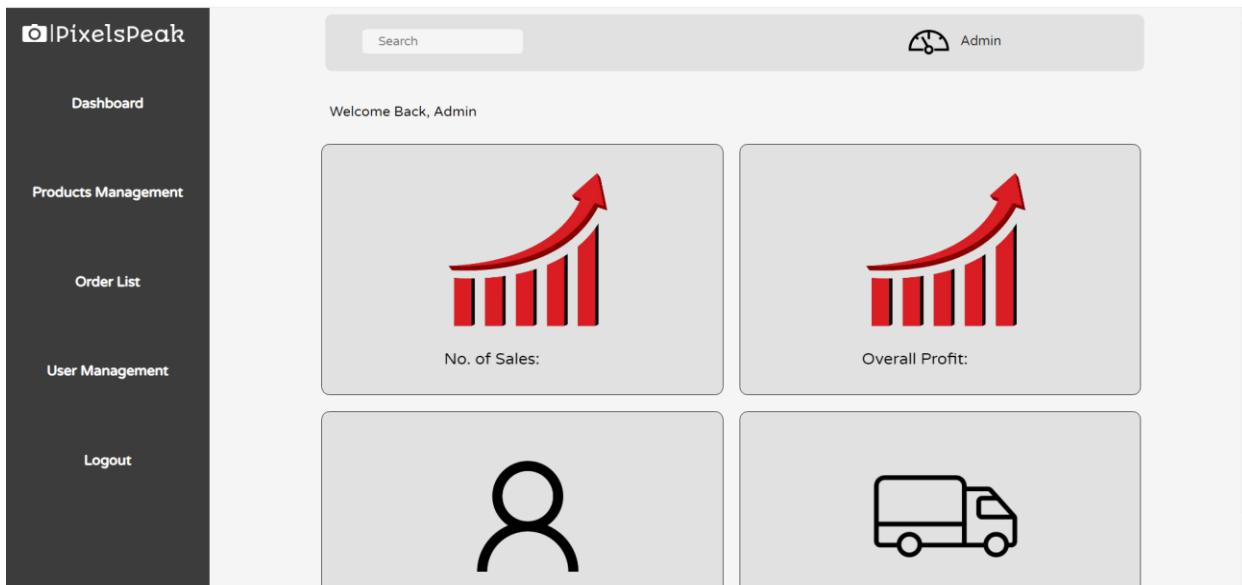
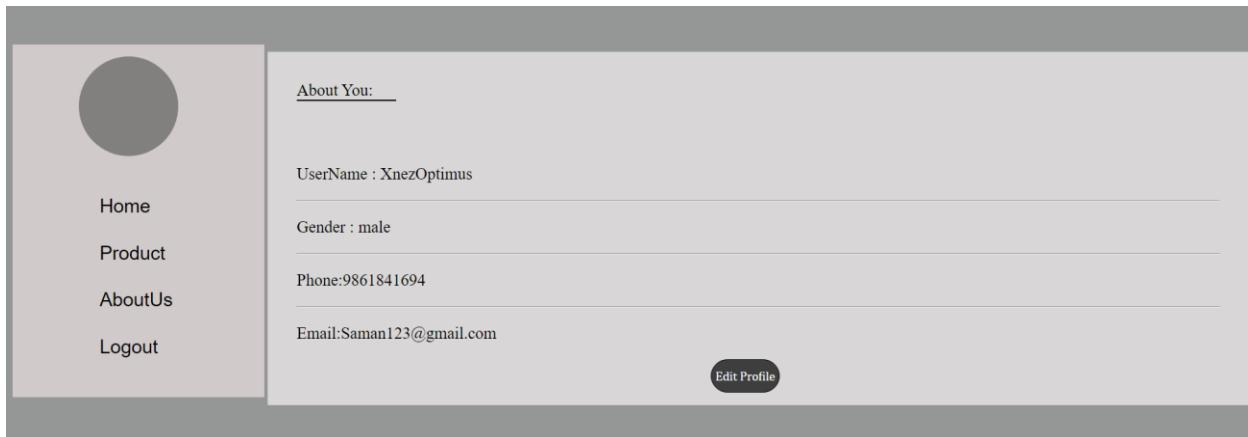


Figure 77: Main Dashboard of Admin

### 7.1.2. User Profile – Anmol Adhikari (22068008):

After logging in, the user is supposed to be able to see his details that he entered for login. Not only details he/she entered to login but one's he/she registered with. This page shows Username, Email, Phone, Gender of the user. This page also lets the user edit their details whenever they want, and it doesn't only edit the values on the webpage (User Profile) but also gets changed in the database. The ability to change their username, email etc. is very important for users as there is need to do so occasionally. Be it Username, Email, these are important for logging in and account recovery or login.

Even though there were a lot of errors when trying to make this page work, there was no error that couldn't be solved. Email and phone are the most dynamic credentials that may need constant update for security purposes. This makes editing them very important as they need to be changed occasionally.



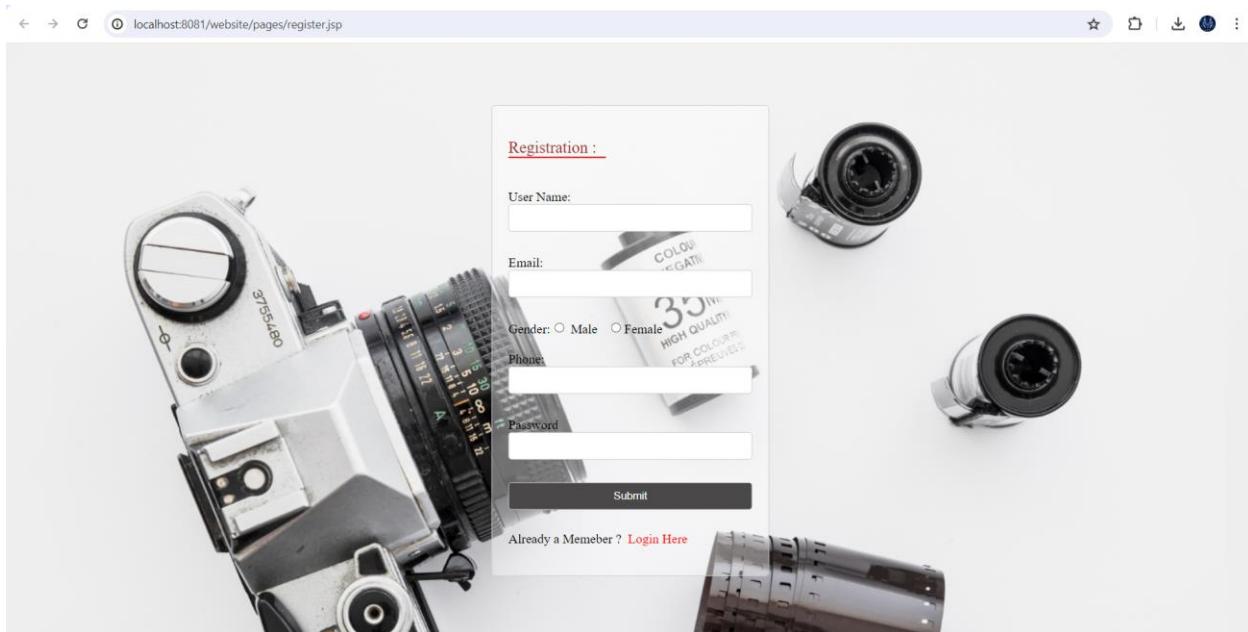
*Figure 78: User Profile Page*

### 7.1.3. Aagaman Mainali (22068131):

- Login & Register: Our website has a login portal and a registration portal through which user can be redirected to either home page or admin can be redirected to admin page based on their email but User can still view or browse the products even though they aren't registered in the system . In this segment of the website the user need to enter a user name as per their preference but the username must be unique so does the mail if the registered mail and email aren't unique system prompts error for username

or email exists and if user enters correct details they are redirected to login page where they need to enter their user name and password which they created at the registration page that has been stored in database

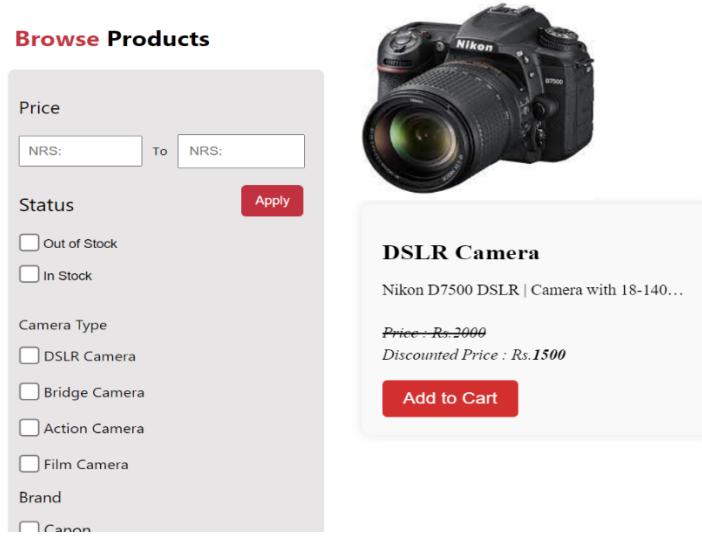
- Homepage UI: The home page or known as the landing page is the page where the user is first redirected to after a successful login in the login attempt. here the page contains what the website is about and glimpse of what it does and how it operates
- About US: About Us is one of the main pages of a website as it the same page through which the users of that website may interact with the administrators or any other person who looks after the system.



#### 7.1.4. Product Page – Rohit Raut (22068148):

Product page: Our website is considered as an e-commerce website in which various types of cameras can be shopped according to user desire. In this page user can see varieties of cameras in terms of brand, price and camera Type. In this page user can filter products in terms of price, brand, and camera type. With the help of the button "Add to cart" user can add products of their choice in their cart, in order to buy any kind of products.

Even though there were a lot of error during image extraction from database so, I decided to extract image from database using SQL query tag.



*Figure 79: Error retrieving data from database in product card*

But instead of extracting image from system database. I put images user “img src” tag in html and CSS. This was an appropriate approach so, after doing concern with our teachers I started doing code for extracting image from database according to MVC pattern.

```

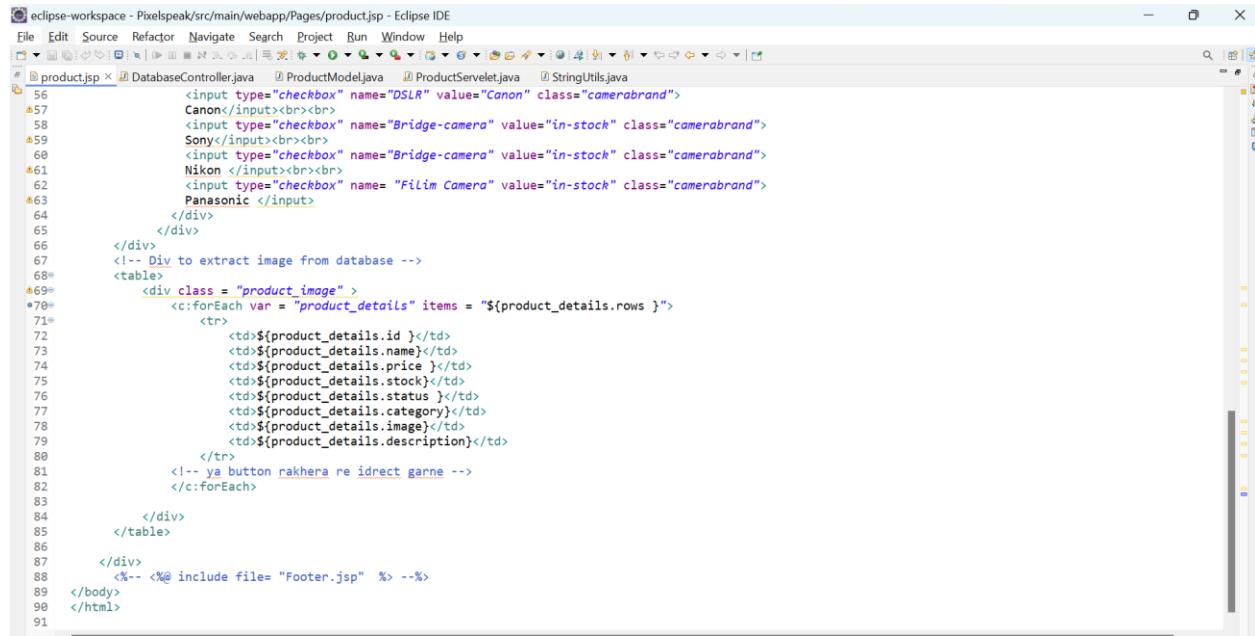
eclipse-workspace - Pixelspeak/src/main/java/model/ProductModel.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
Project Explorer product.jsp DatabaseController.java ProductModel.java ProductServlet.java StringUtil.java
src/main/java
    controller DatabaseController.java
    model ProductModel.java
    servlet ProductServlet.java
    util StringUtil.java
    Libraries
src/main/java
    controller DatabaseController.java
    model ProductModel.java
    servlet ProductServlet.java
    util StringUtil.java
src/main/java
    controller DatabaseController.java
    model ProductModel.java
    servlet ProductServlet.java
    util StringUtil.java
    waran

```

```

1 package model;
2
3 public class ProductModel {
4     //variable declaration
5     private String id;
6     private String name;
7     private String price;
8     private String stock;
9     private String status;
10    private String category;
11    private String image;
12    private String description;
13
14    //construction
15    public ProductModel(String id, String name, String price, String stock,
16                        String status, String category, String image,
17                        String description) {
18        super();
19        this.id = id;
20        this.name = name;
21        this.stock = stock;
22        this.price = price;
23        this.status = status;
24        this.category = category;
25        this.image = image;
26        this.description = description;
27    }
28    //getter method
29
30    public String getId() {
31        return id;
32    }

```



The screenshot shows the Eclipse IDE interface with the product.jsp file open in the editor. The code is JSP (Java Server Pages) with some Java code mixed in. It displays a form with checkboxes for camera brands and a table for product details. The code includes imports for java.util.List and javax.servlet.jsp.JspException, and various JSTL tags like <input>, <table>, <div>, and <c:forEach>. The page also includes a footer section with <%@ include file="Footer.jsp" %>.

```

eclipse-workspace - Pixelspeak/src/main/webapp/Pages/product.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
product.jsp DatabaseController.java ProductModel.java ProductServlet.java StringUtils.java
56     <input type="checkbox" name="DSLR" value="Canon" class="camerabrand">
57     Canon</input><br><br>
58     <input type="checkbox" name="Bridge-camera" value="in-stock" class="camerabrand">
59     Sony</input><br><br>
60     <input type="checkbox" name="Bridge-camera" value="in-stock" class="camerabrand">
61     Nikon </input><br><br>
62     <input type="checkbox" name="Filim Camera" value="in-stock" class="camerabrand">
63     Panasonic </input>
64   </div>
65 </div>
66 <!-- Div to extract image from database -->
67 <table>
68   <div class = "product_image" >
69     <c:forEach var = "product_details" items = "${product_details.rows }" >
70       <tr>
71         <td>${product_details.id }</td>
72         <td>${product_details.name}</td>
73         <td>${product_details.price }</td>
74         <td>${product_details.stock}</td>
75         <td>${product_details.status }</td>
76         <td>${product_details.category}</td>
77         <td>${product_details.image}</td>
78         <td>${product_details.description}</td>
79       </tr>
80     <!-- ya button rakhra re idrect garne -->
81   </c:forEach>
82 
83   </div>
84 </table>
85 </div>
86 <!-- <%@ include file="Footer.jsp" %> --%>
87 </body>
88 </html>
89
90
91

```

After implementing this it always flags an error so I decided to import image from database by implementing SQL Query Tag

## 7.2. Development Procedure:

Our development procedure for the e-commerce website project began with a proper planning phase where all of the team members discussed how the website would look like. We first decided to make wireframe from all the pages that we thought we would be making for this coursework. After the creation of wireframe our job was to create a simple UI design which would be base for the further development process, for this we used Figma to create a UI which was looking good and after everyone's approval it was the time for the coding process.

The project was to be made using JEE framework and MVC architecture pattern, it was our first time using MVC pattern so we had many difficulties when creating the website but after we did it for some days, we started getting used to it. All four team members divided different tasks among each other and started creating a dynamic website. We took the creation of database and front-end parallelly so that we won't have confusion when importing or exporting different data from or to the database. As we progressed the implementation of critical features such as the login system and admin panel demanded meticulous attention to detail. These two features were one of the most

difficult features to be implemented in the project. Database integration and session management were seamlessly integrated into the login feature, ensuring secure access and robust error handling. Simultaneously, the admin panel was made with different administration tool and after that our target was to polish the front-end user interface that we created.

Finally, after testing and error handling our project was ready to be integrated. We also faced different types of difficulties and error when integrating all the codes into one singular system which we dealt with ease. And at last, we documented our work explaining how our code was working and different types of tools and libraries used in it.

## 8. Critical Analysis:

During the development of our JSP-based web application, we encountered several challenges and problems that required careful attention and resolution. Below, we outlined some of the main challenges we faced along with the corresponding solutions:

### 1. IDE Errors and Code Errors:

**Problem:** During development, IDE (Integrated Development Environment) problems and code errors were frequent, resulting in inconsistent and dysfunctional code.

**Solution:** To identify and address problems early in the development process, we established routine code reviews and made use of IDE tools like syntax highlighting, code formatting, and error detection. In addition, we often restarted the IDE to maintain its usability.

### 2. Server Errors:

**Problem:** During deployment and testing, server issues such as HTTP 500 Internal Server Error and Tomcat server failures frequently happened.

**Solution:** In order to determine the actual causes of server problems, we carefully examined server logs. Debugging code logic, improving database queries, and verifying appropriate operation by sometimes restarting and altering its ports were common techniques.

### 3. XAMPP Errors:

**Problem:** XAMPP-related faults hampered local development and testing, such as failed MySQL database connections or misconfigured Apache servers.

**Solution:** We made sure that software versions were compatible, adjusted the XAMPP settings carefully, and sought help from internet forums and resources for troubleshooting. For future reference, we have included listed typical XAMPP issues and their fixes.

**4. Compatibility Issues:**

**Issue:** Compatibility issues came up when testing the web application on different browsers and devices, leading to inconsistent user experiences.

**Solution:** We employed use of Rem unit instead of pixel in code in some case where inconsistency in pixel units were found .we conducted cross-resolution testing to ensure compatibility with popular browsers (e.g., Chrome, Firefox, Safari) tablets, smartphones).

## 9. Conclusion:

In conclusion, there was opportunity as well as obstacles to overcome while creating this website within the JEE framework utilising JDBC, JSP, and the MVC design pattern. Our team faced a variety of challenges along the way, from user interface design to database connectivity, all of which required careful attention to detail code errors and creative problem-solving. Clearly, there were obstacles in the way of our task but every challenge, from figuring out difficult code to debugging them and managing the database, was a step closer to knowledge. But with dedication and cooperation, our team overcame these difficulties by depending on one another's skills and abilities to overcome problems and provide a solution.

We have the highest gratitude for the instructors, whose guidance and leadership lighted our way and provided invaluable assistance at every turn. Their constant encouragement and support inspired us to our goals and pushed us to completion of this project. We also want to express our sincere gratitude to our classmates, whose support and teamwork boosted our commitment and turned difficulties into just a minor hurdle. This project taught us many things about java development and critical thinking and problem-solving skills.

As we bring this project to a close, this project demonstrated how java are used in real life industry to create a website which is dynamic. The journey we embarked on was not just about building a website but also it was a test to our collective determination, resilience, and teamwork.

## 10. References

Apache, 2002. *Apachefriends*. [Online]

Available at: <https://www.apachefriends.org/>

[Accessed 03 05 2024].

Balsamiq, 2018. *Introduction to User Interface Design Through*, s.l.: Balsamiq.

Balsamiq, 2024. *Balsamiq*. [Online]

Available at: <https://balsamiq.com/>

[Accessed 07 05 2024].

Byjus, 2011. *Byjus*. [Online]

Available at: <https://byjus.com/govt-exams/microsoft-word/>

[Accessed 03 05 2024].

Eclipse, 2001. *Eclipse*. [Online]

Available at: <https://www.eclipse.org/home/whatis/>

[Accessed 03 05 2024].

EDX, 2024. *EDX*. [Online]

Available at:

[https://www.edx.org/learn/xampp#:~:text=XAMPP%2C%20which%20stands%20for%20Cross,ands%20Linux%20\(LAMP\)%20environments.](https://www.edx.org/learn/xampp#:~:text=XAMPP%2C%20which%20stands%20for%20Cross,ands%20Linux%20(LAMP)%20environments.)

[Accessed 07 05 2024].

Felici, M., 2017. *Class Diagrams*, Edinburgh: The University of Edinburgh.

Figma, 2024. *Figma*. [Online]

Available at: <https://www.figma.com/learn-more-about-figma/>

[Accessed 07 05 2024].

Hook, C., 2021. *IOE Notes*, Kathmandu: ClampHook Academy.

IBM, 2024. *IBM*. [Online]

Available at: <https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>

[Accessed 07 05 2024].

IBM, 2024. *IBM*. [Online]

Available at: <https://www.ibm.com/topics/software-testing>

[Accessed 08 05 2024].

Jitnupong, B., 2017. *Information System User Interface Design in Software Services Organization: A Small-Clan Case Study*, Bangkok, Thailand: National Institute of Development Administration.

MS Word, 2024. *MS Word*. [Online]

Available at: <https://www.microsoft.com/en/microsoft-365/word?market=af>

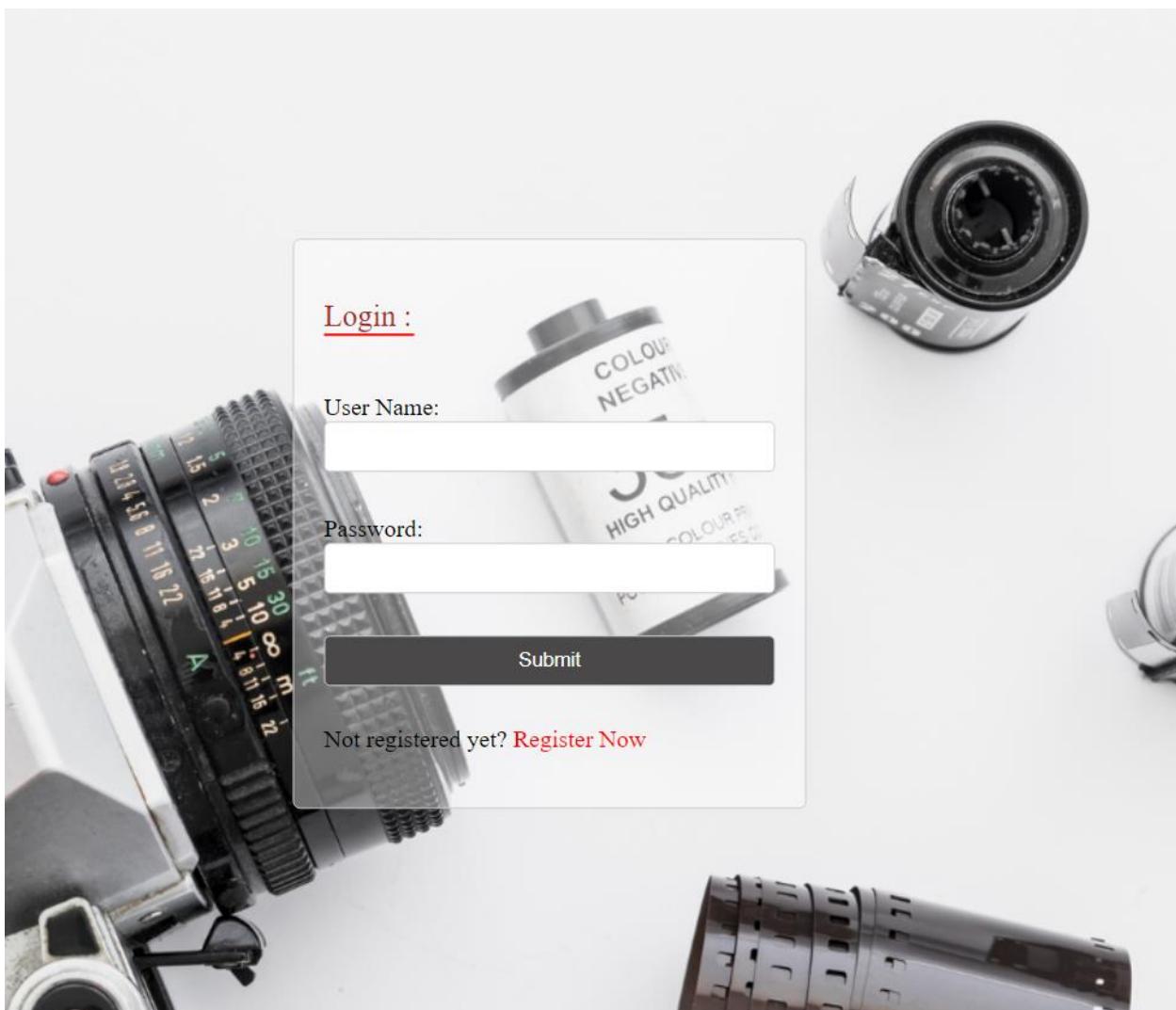
[Accessed 07 05 2024].

MySQL, 2024. *MySQL*. [Online]

Available at: <https://dev.mysql.com/doc/connector-odbc/en/connector-odbc-introduction.html#:~:text=MySQL%20Connector%2FODBC%20provides%20both,transactions%20and%20full%20Unicode%20compliance.>

[Accessed 07 05 2024].

## 11. Website References:



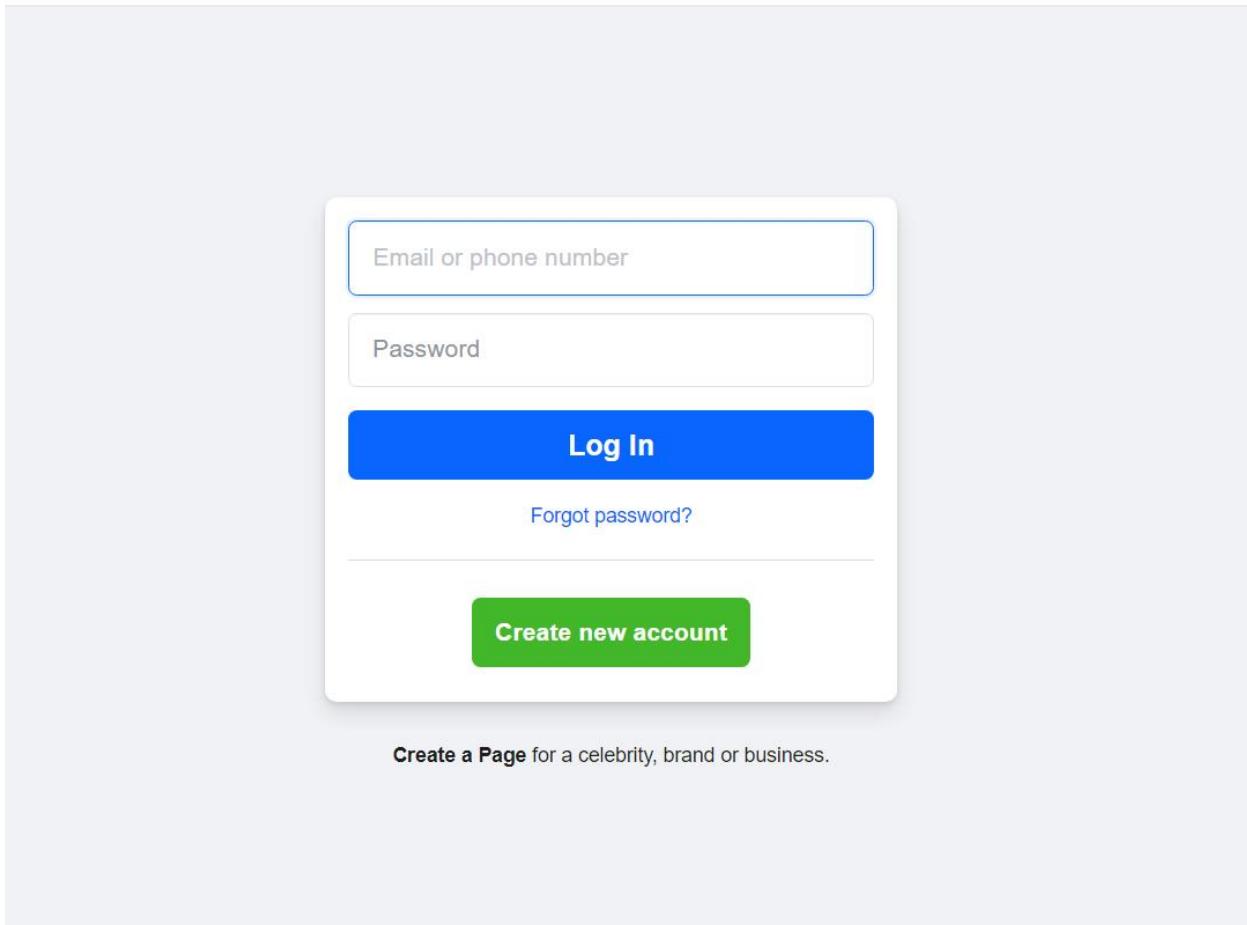


Figure 80: Reference for Login Page (Facebook)

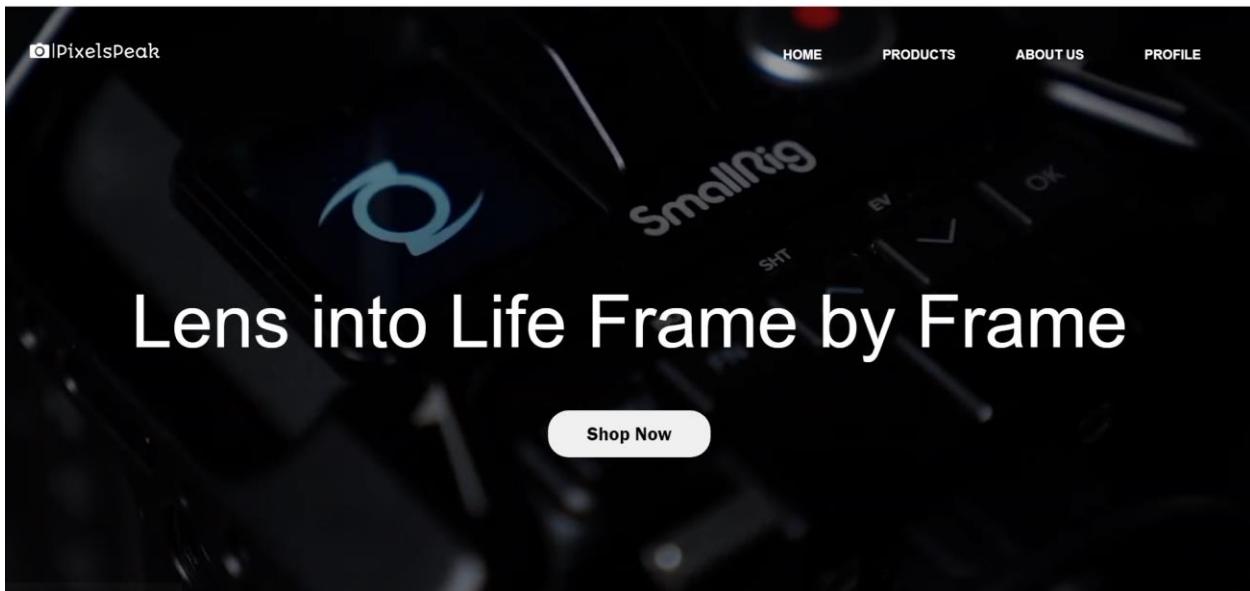




Figure 81: Reference for Home Page (Nikon.com)

Figure 82: Reference for Product Page (Itti Nepal)

Figure 83: Reference for Footer (Itti Nepal)

Figure 84: Reference for navigation bar (canon.com)