LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS5054NI Advanced Programming & Technologies

Assessment Type
50% Group Coursework

Semester
2024 Spring

Group Members

London Met ID	Student Name
22067560	Abhijay Dhoj Adhikari
22067545	Pratik Karanjit
22067528	Siddhanta Shrestha
22067526	Kristan Dharel
22067570	Krish Bhattarai

Project Title: Circuit Laptop Pasal

Assignment Due Date: Friday, May 10, 2024

Assignment Submission Date: Thursday, May 9, 2024

Submitted to: Mr. Prithivi Maharjan

Word Count: 4021

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

Table of Figures

Table of Tables

1. Introduction

The group coursework requires the development of an e-commerce website that sell electronics and gadgets. The e-commerce website aims to have user-friendly interfaces that allows the customers to navigate and have a smooth experience. The e-commerce website also incorporates admin login and register functionalities granting privileged access to administrators.

The Model-View-Controller (MVC) architectural pattern is required for the project which serves as the architectural background. In MVC pattern Model contains backend that contains all data logics, View contains all the frontend or GUI and Controller contains logics that controls how data is displayed. The website incorporates various essential features like the login feature, admin panel feature, and user profile page alongside other important features that are required in an e-commerce website.

1.1. Aim:

The aim of this coursework is to provide students with a practical experience in design and implementation of a dynamic ecommerce website system. The Model-View-Controller (MVC) design pattern is implemented to make the development process efficient. It also aims to create a database schema and implemented it into the system, allowing for a way to store data in an organized manner. This allows for efficient storage and easy retrieval of data.

1.2. Objectives

The objectives of this coursework are:

- Developing an e-commerce website which specializes in selling electronics and gadgets to the user.
- Creating a smooth and user-friendly interface for streamlined interaction which includes the selection of products and checkout process with safe and secure payment methods.
- Proper implementation and utilization of the database to handle product data, user information, and transactions.

- Creating servlets within controller package to handle user requests and respond to the user.

2. User Interface Design

2.1 Wireframe

Wireframe is an early blueprint of a website or a digital product that focuses on arranging different elements of the website such as buttons, functionalities, etc. It helps to pave path for completion of the project by working according to the wireframe. It gives the members of the team a visual representation of the system design. (Figma, 2024). Balsamic was used to create wireframes for the project.

The following are the concept designs created for the system:

2.1.1 Wireframe for Register Page

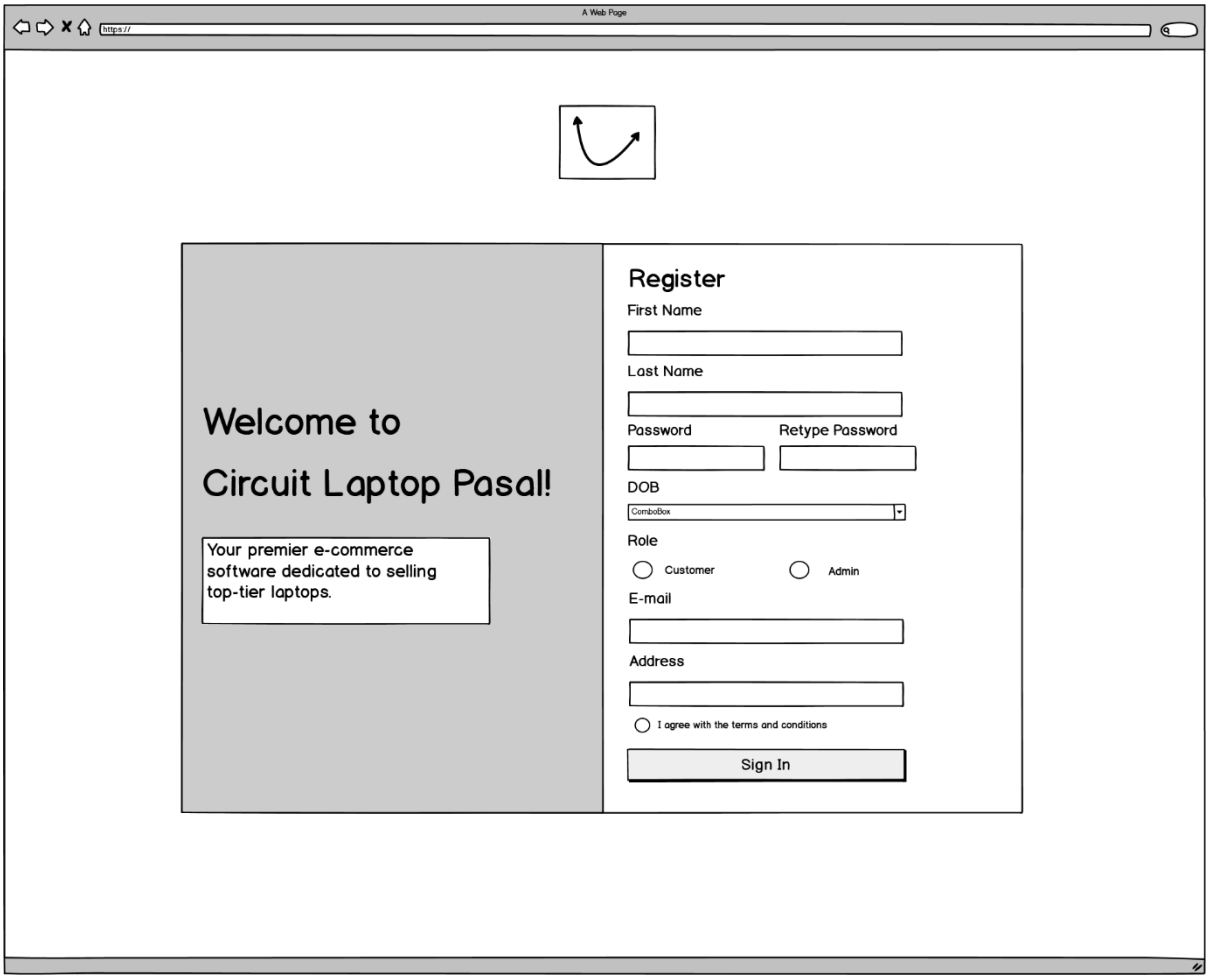


Figure 1: Wireframe for Register Page

2.1.2 Wireframe for Login Page

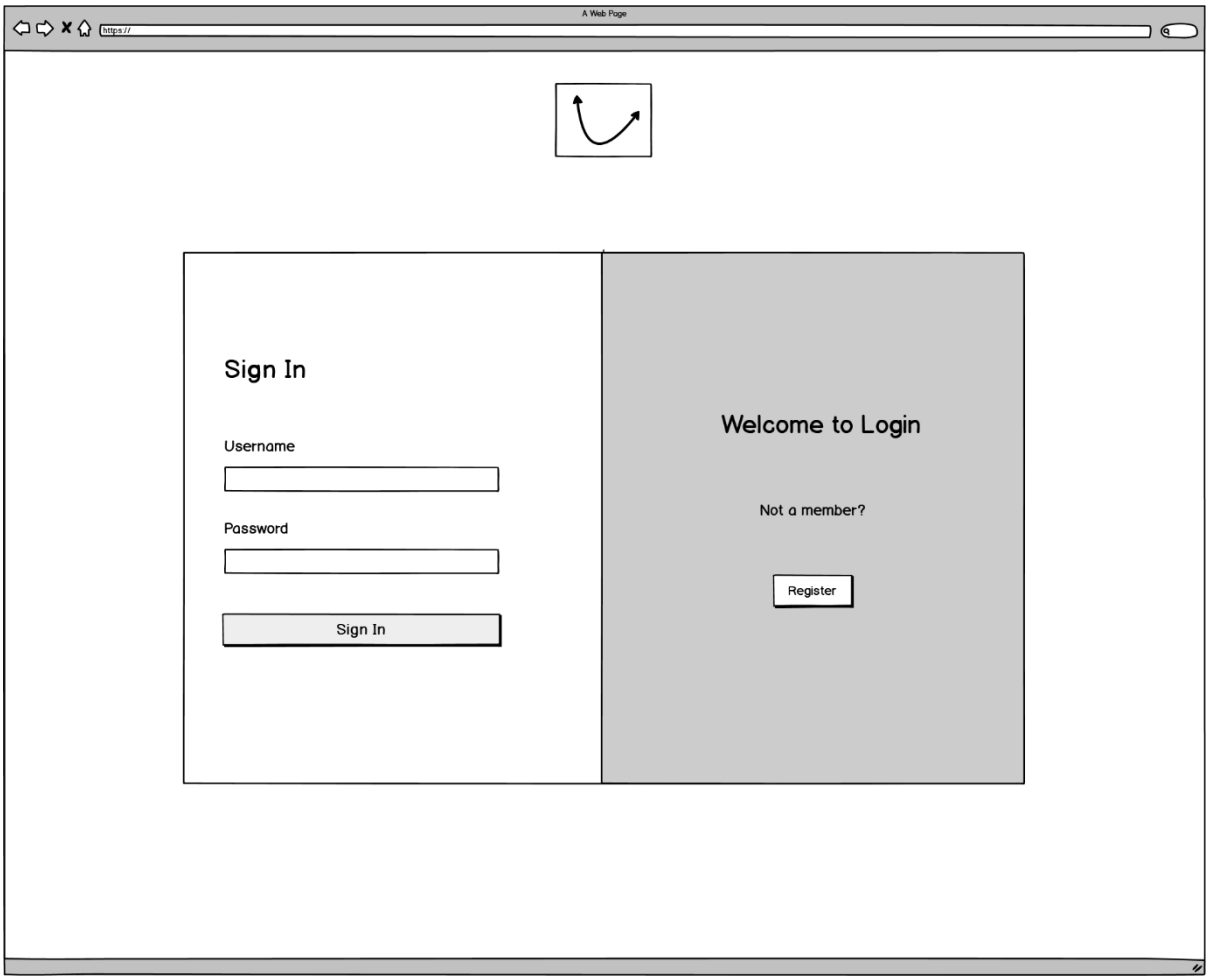


Figure 2: Wireframe for Login Page

2.1.3 Wireframe for User Profile Page

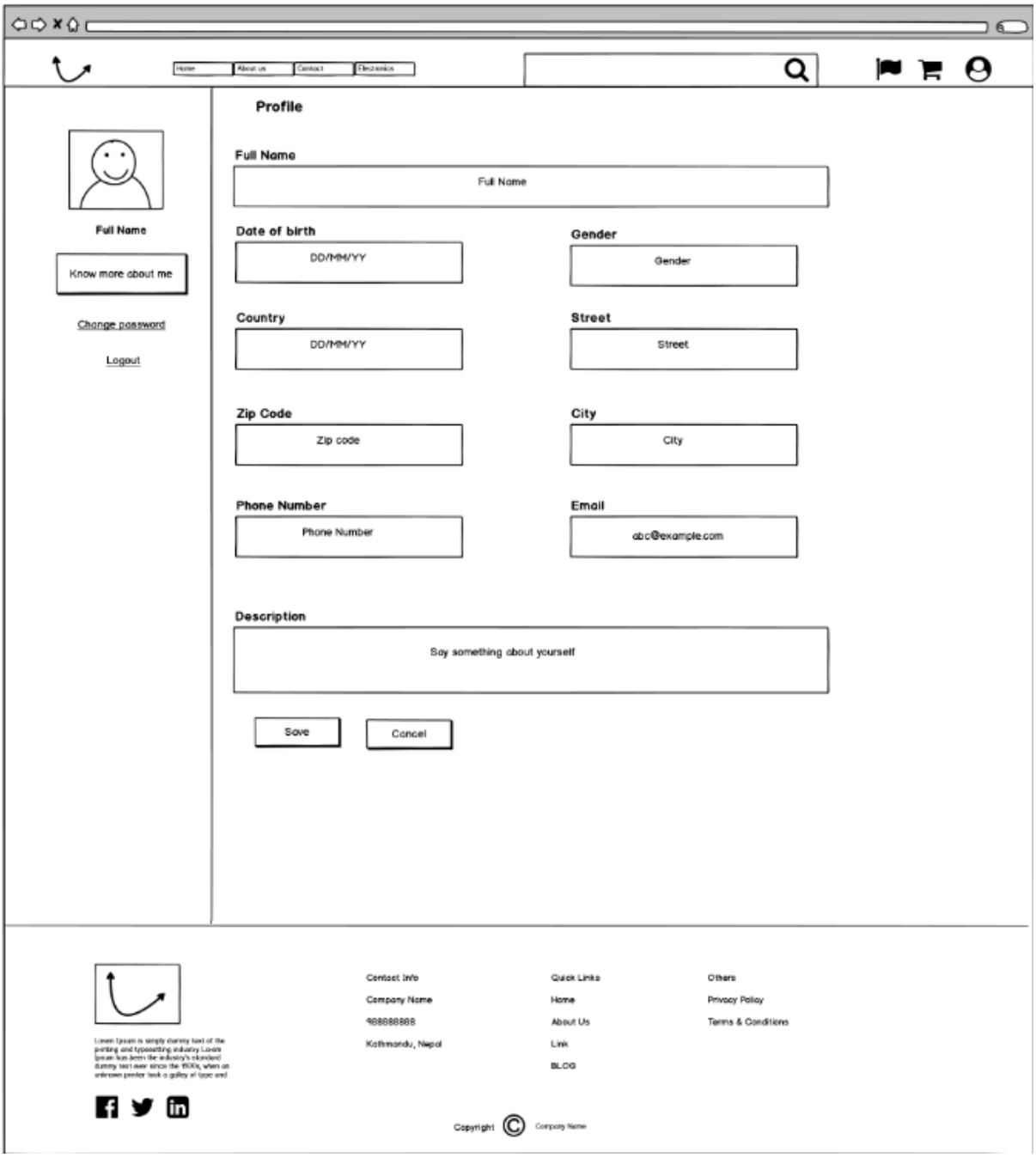


Figure 3: Wireframe for User Profile Page

2.1.4 Wireframe for Product Page

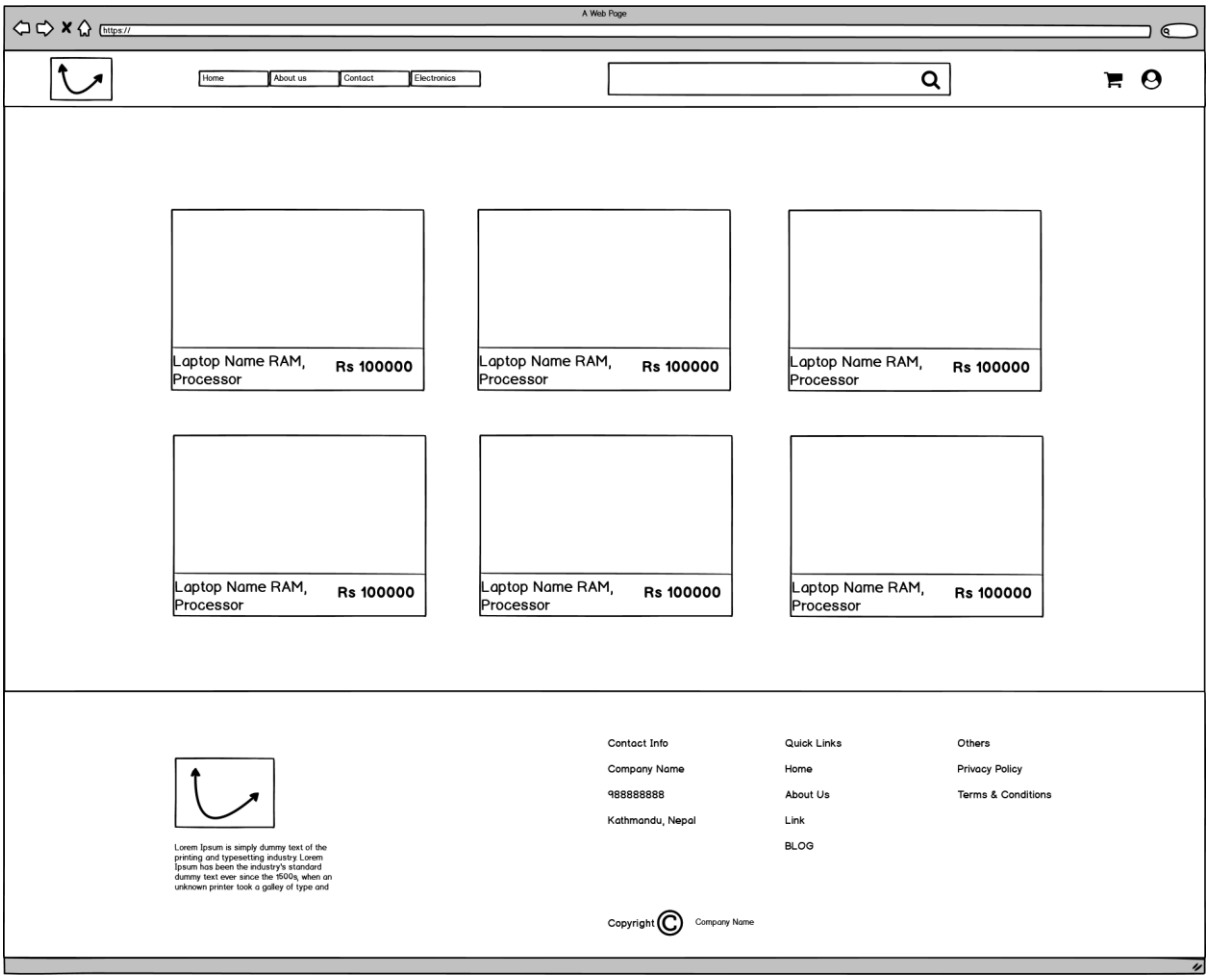


Figure 4: Wireframe for Product Page

2.1.5 Wireframe for Product Detail Page

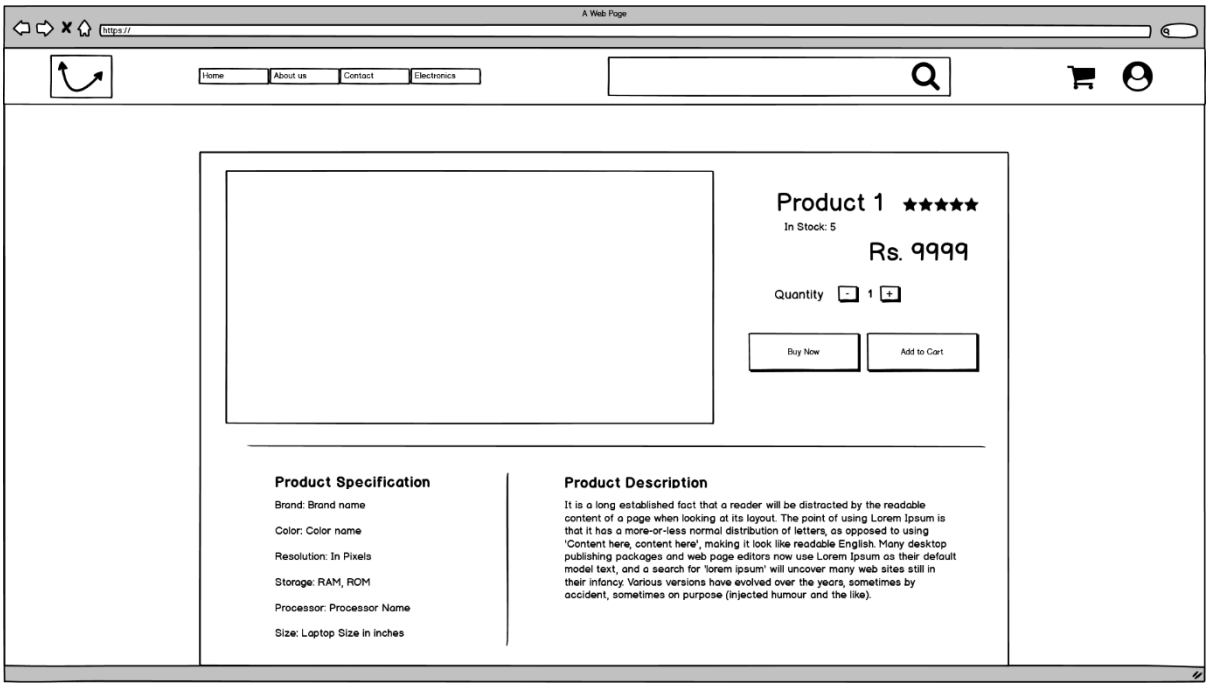


Figure 5: Wireframe for Product Detail Page

2.1.6 Wireframe for Cart Page

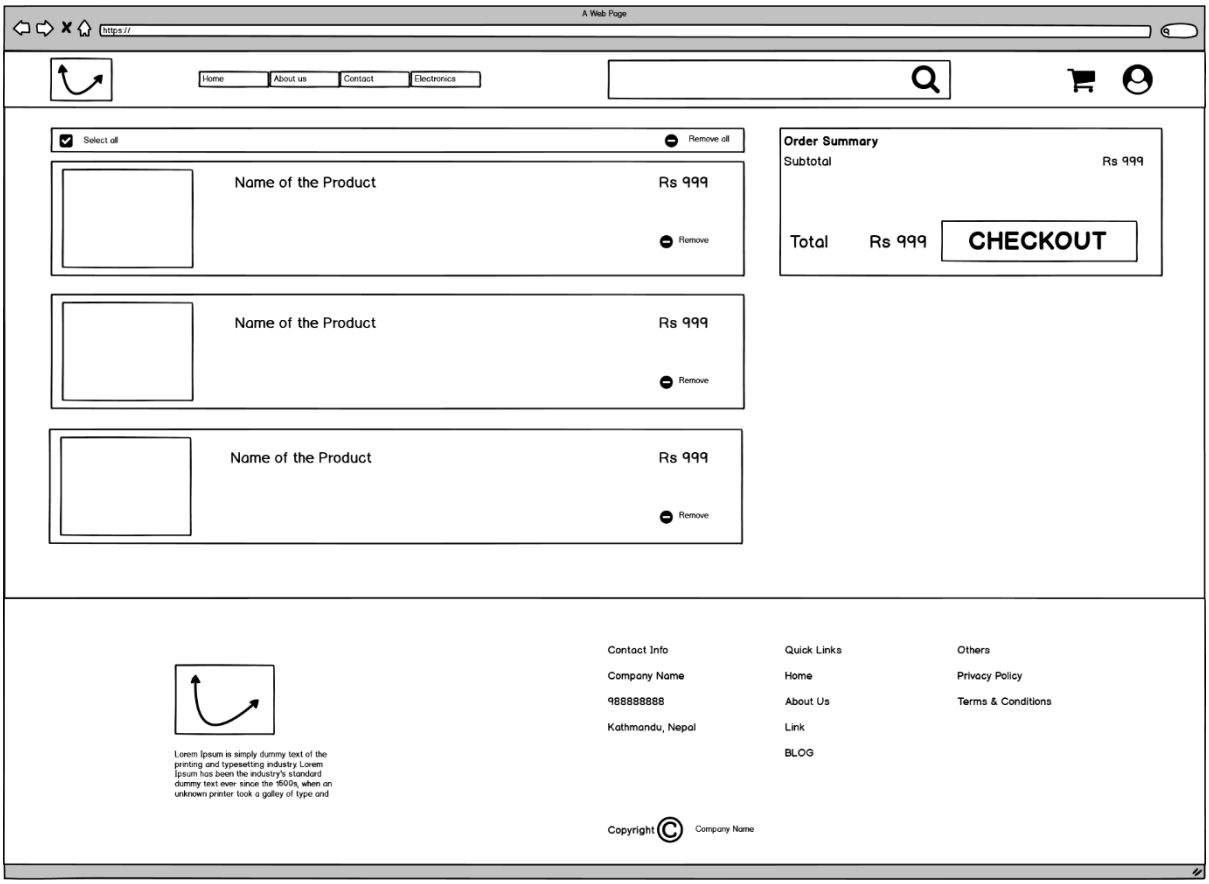


Figure 6: Wireframe for Cart Page

2.1.7 Wireframe for Admin Panel Page

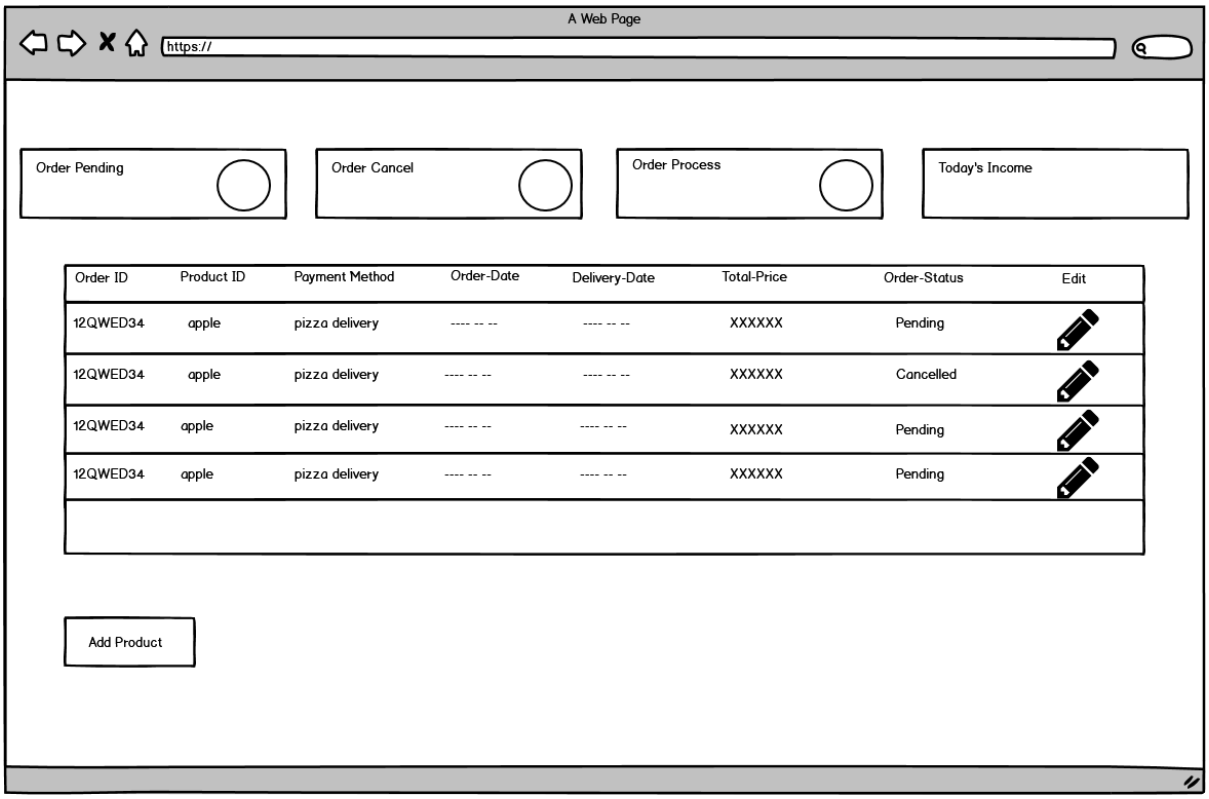


Figure 7: Wireframe for Admin Panel Page

2.1.8 Wireframe for About Us Page

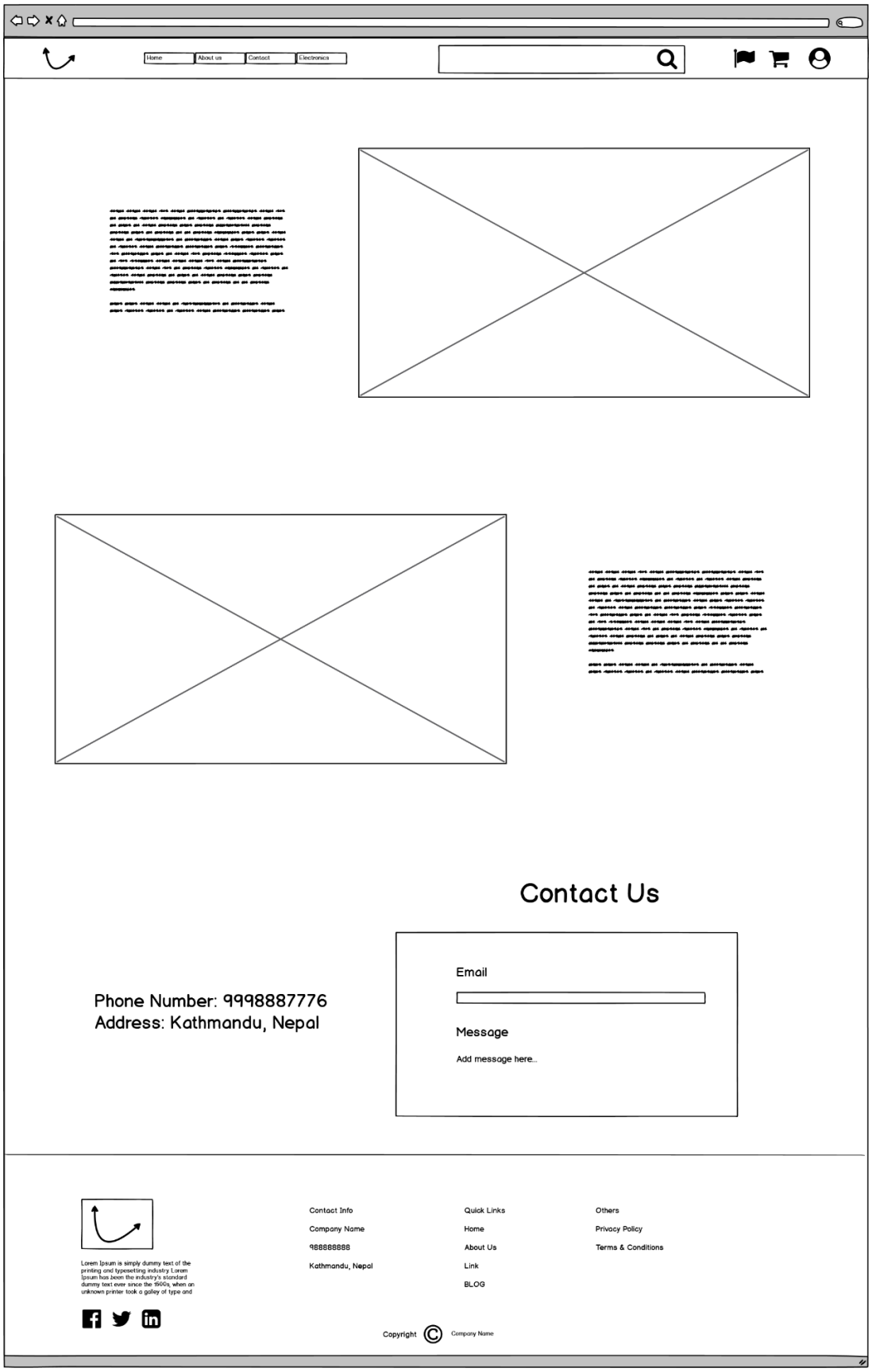


Figure 8: Wireframe for About Us Page

2.1.8 Wireframe for User History Page

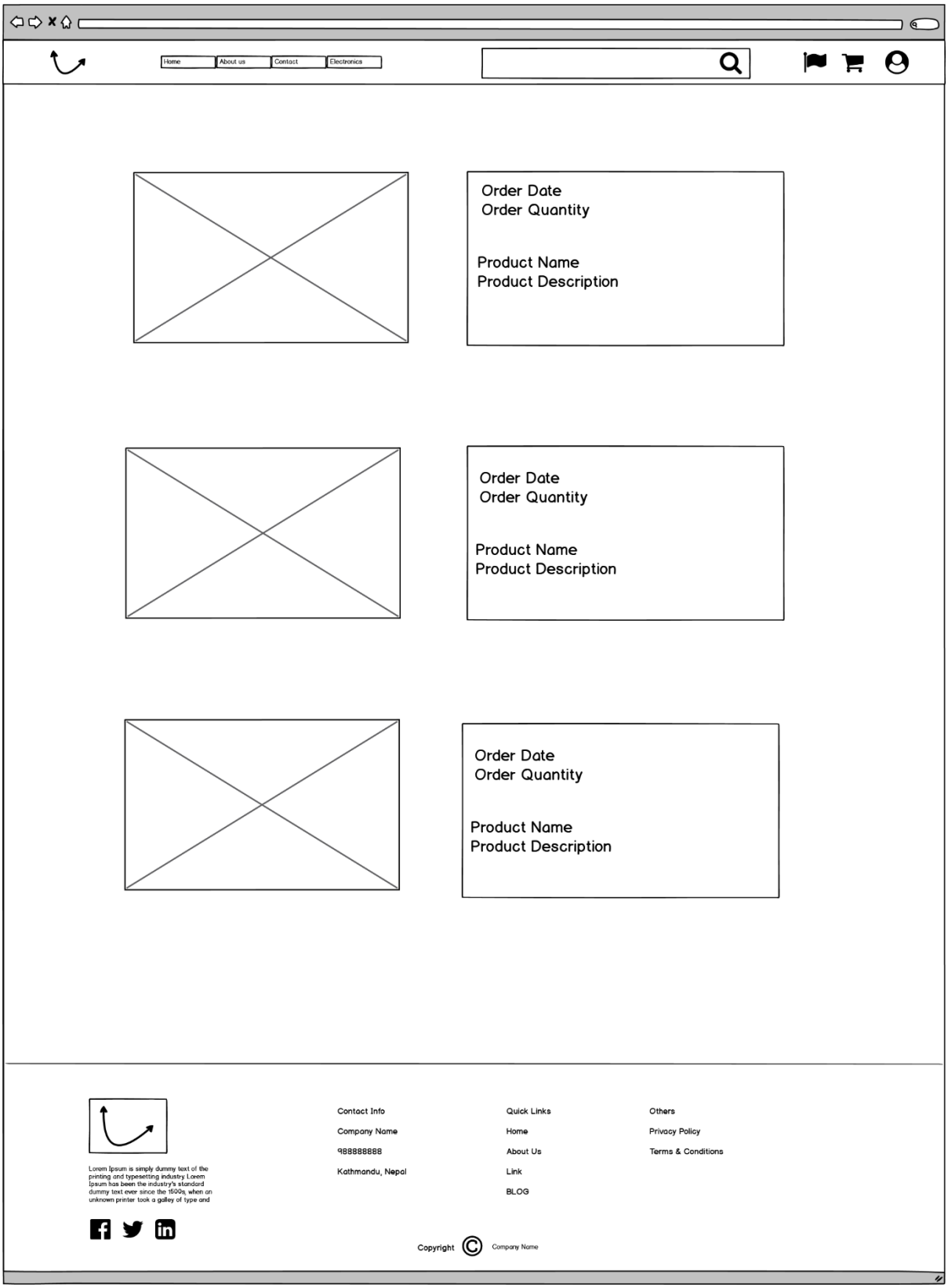


Figure 9: Wireframe for User History Page

2.2 Actual Design

These are the Actual designs that have been finalized for the system:

2.2.1 Register page

The users need to provide details such as their First Name, Last Name, DOB, email, and so on to be registered into the system. The users are also asked to set their username, their password and role during registration. Registration form has validation, the username and email cannot repeat with the existing username. If username and email repeat an error message is shown in the form.

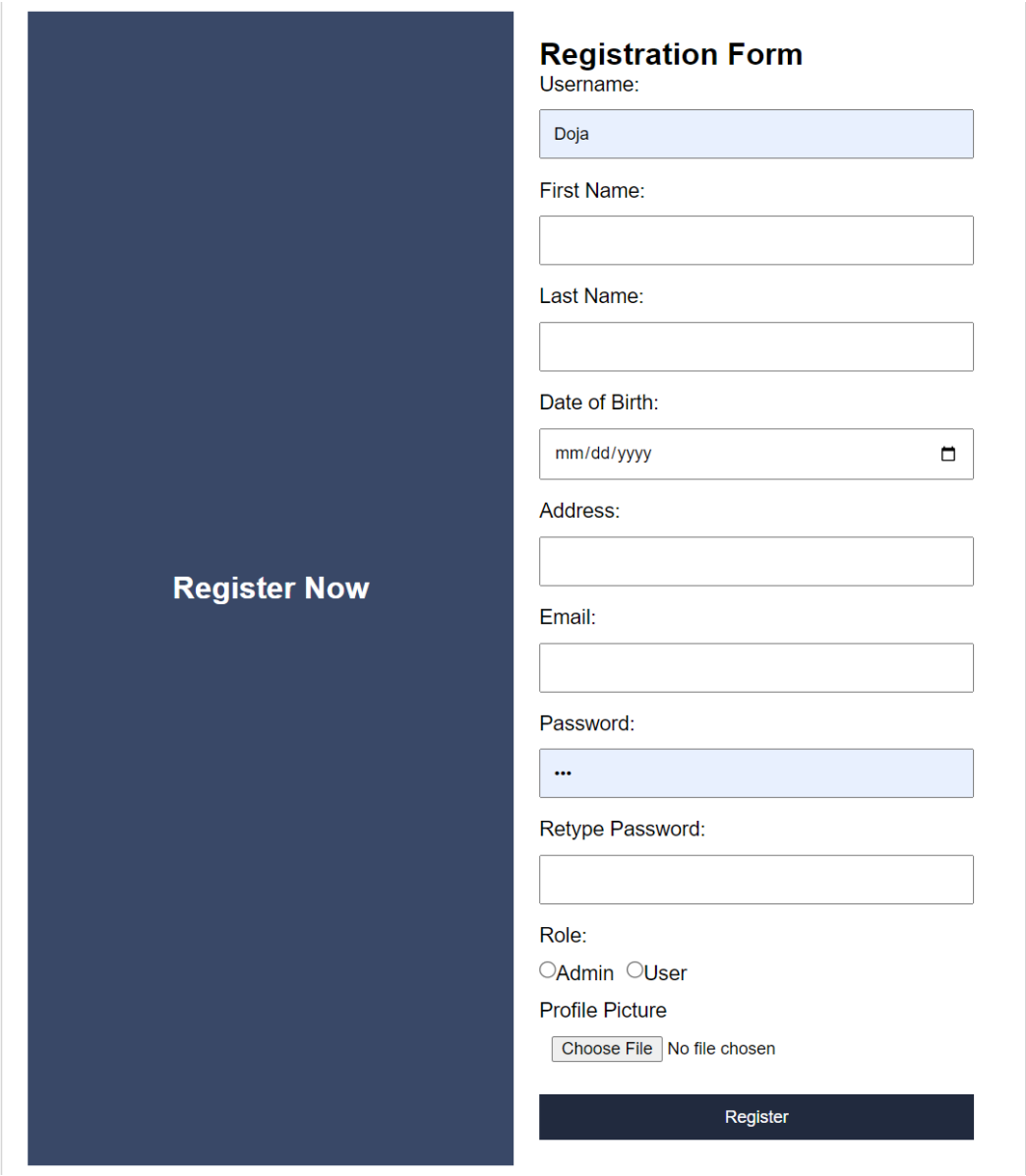


Figure 10: Actual Design for Register Page

2.2.2 Login page

Users must submit their username and password to login to the system. It also provides a link to the register page if in case a user isn't registered into the system. If a user provides invalid username or password, a suitable error message is displayed.

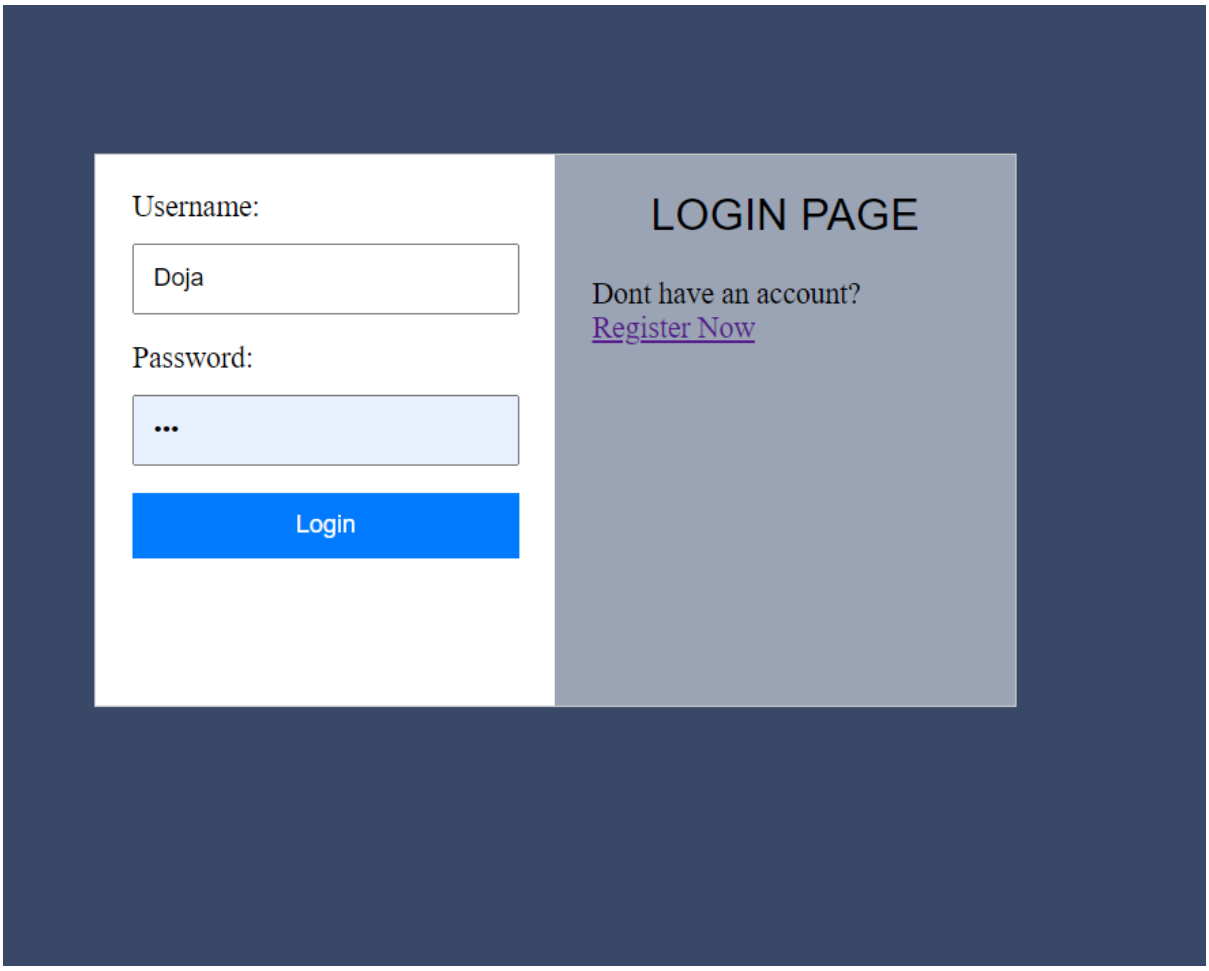
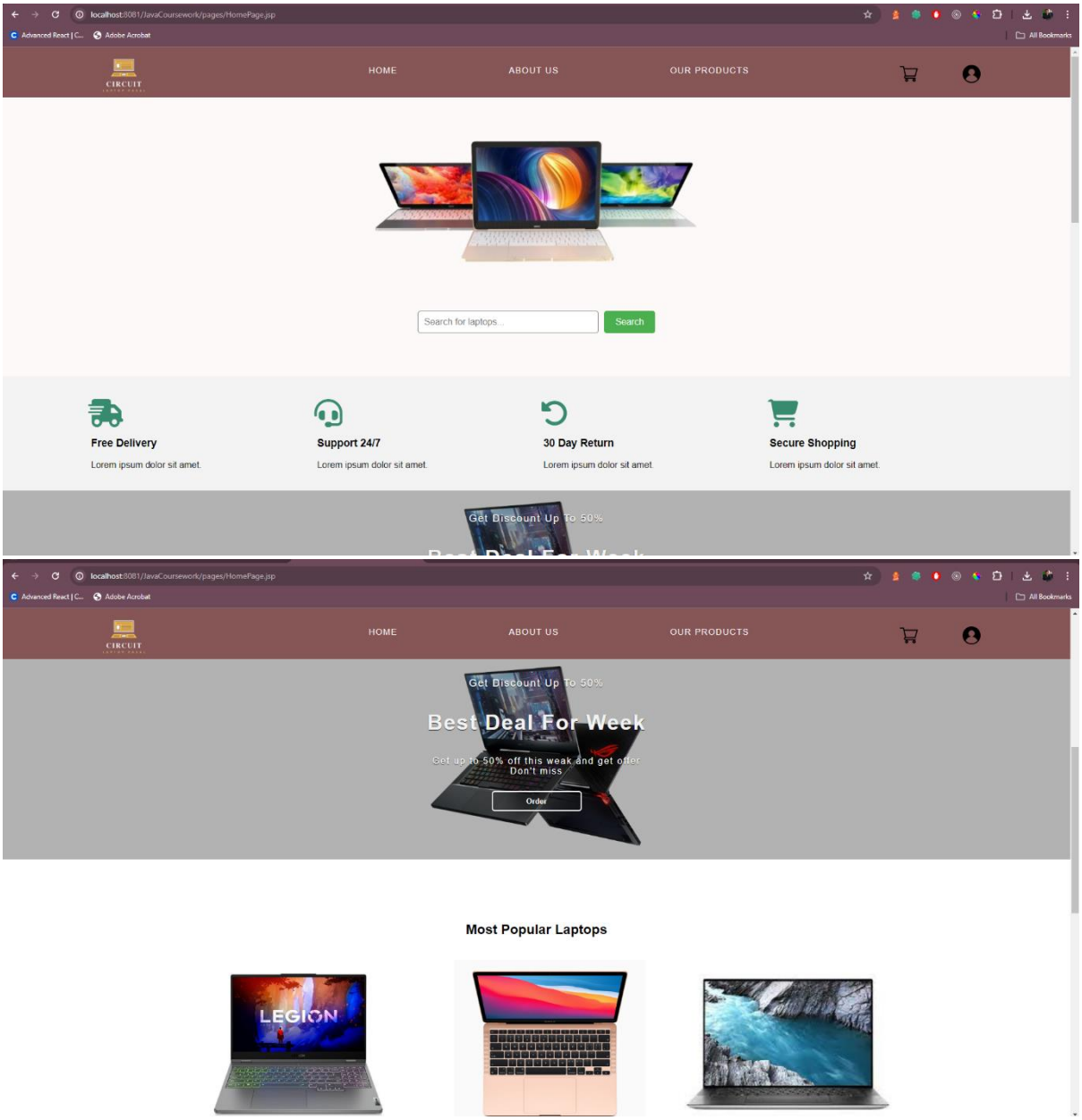


Figure 11: Actual Design for Login Page

2.2.3 Home Page

The home page is the first page that the users view when visiting a website. It implements a search functionality where the users get to search for a specific product. It also includes a navigation functionality where the users can navigate to different pages of the website such as the product page or the about us page. It also contains of a section where the best deals for the week and the most popular laptops are displayed.



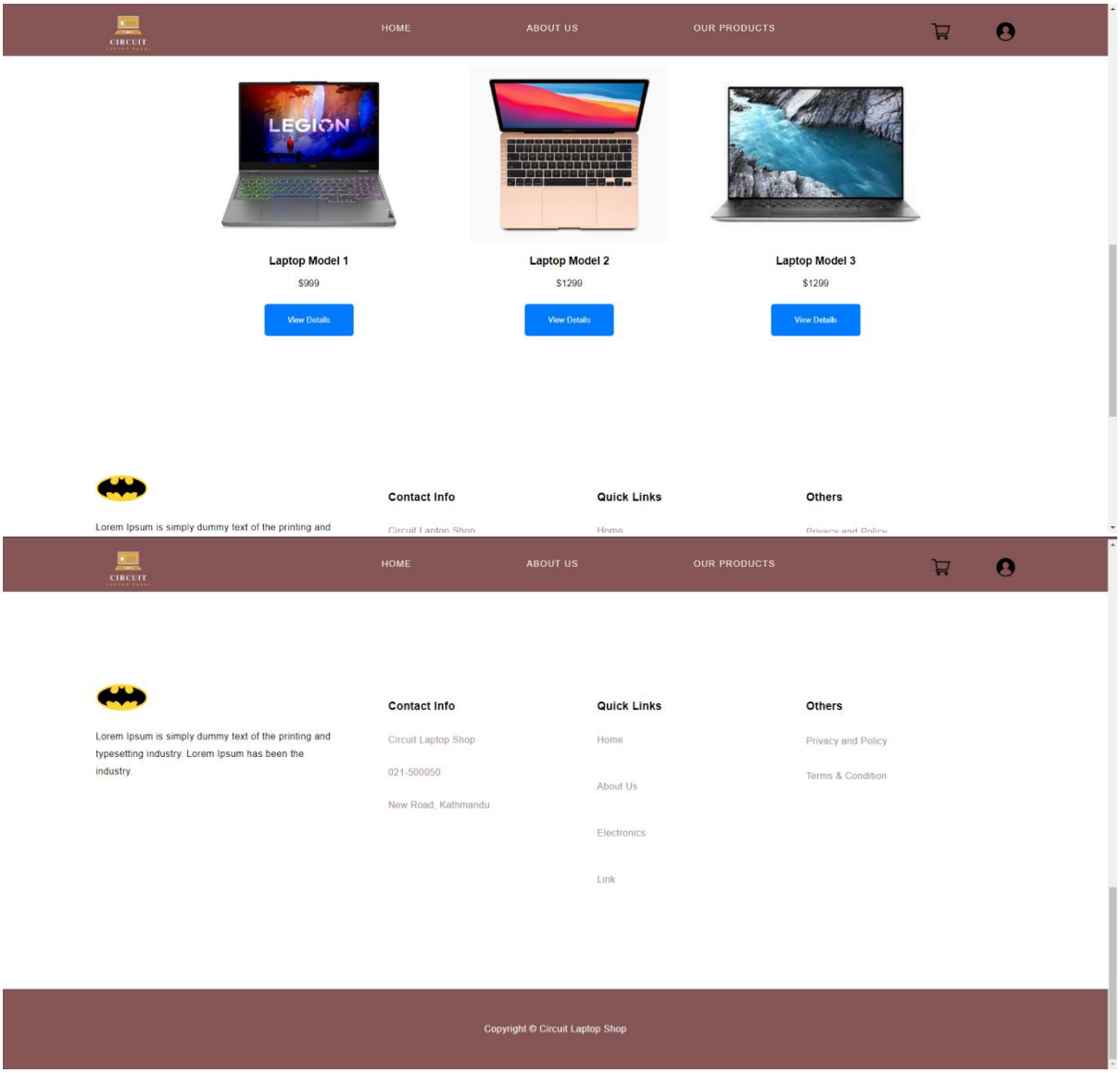
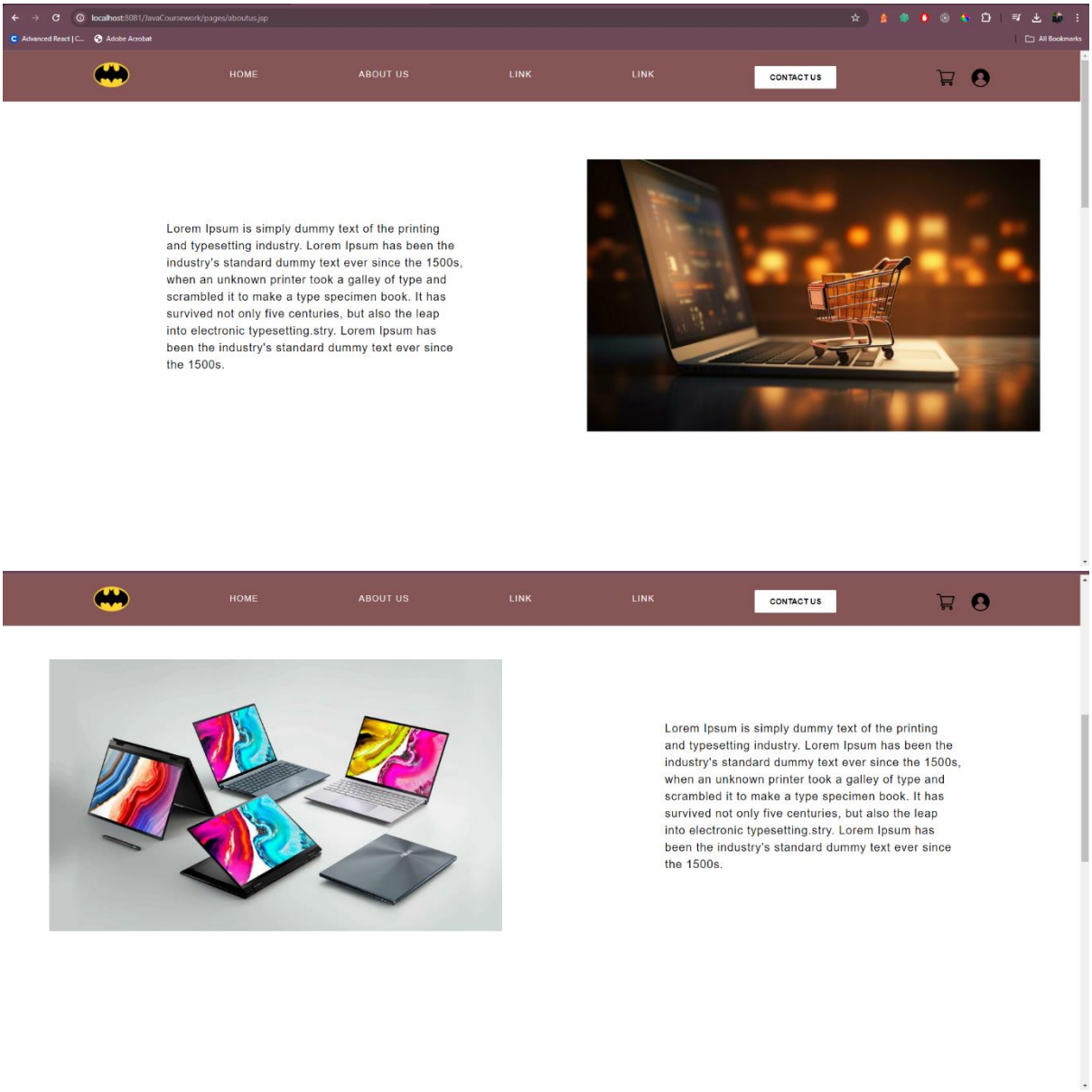


Figure 12: Actual Design for Home Page

2.2.4 About Us

The about up page contains information about the creators of the website. The users can also contact the support team by filling up the Contact Us form.



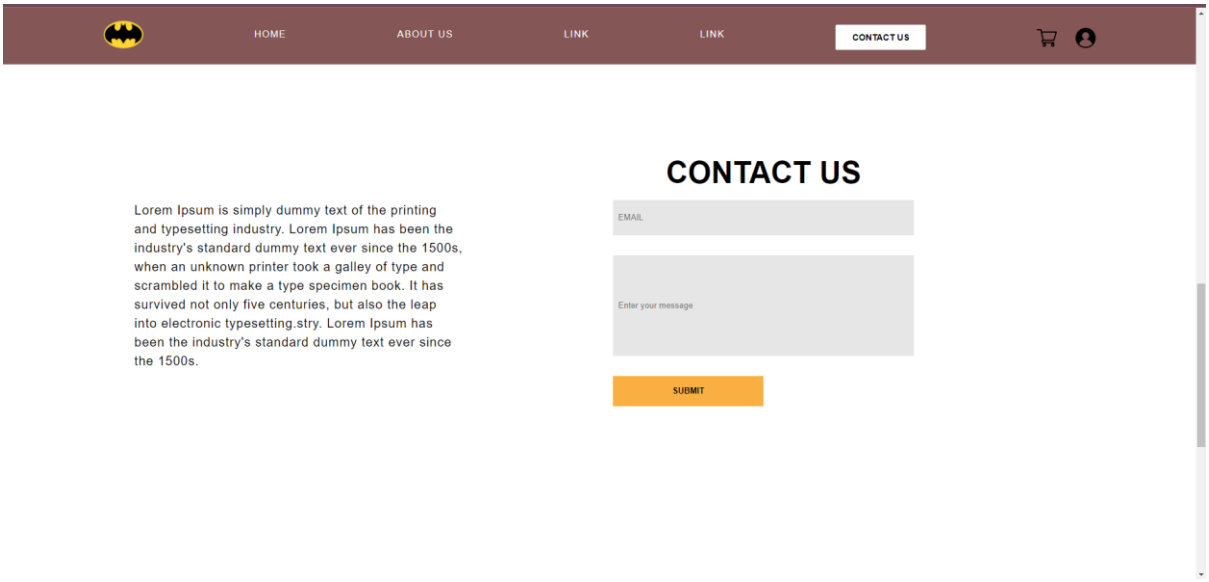


Figure 13: Actual Design for About Us page

2.2.5 User Profile Page

In the user profile page, users can view their information that the users had entered during the registration process. This includes details such as username, first name and DOB. Furthermore, the users also get the option to edit their user details if desired.

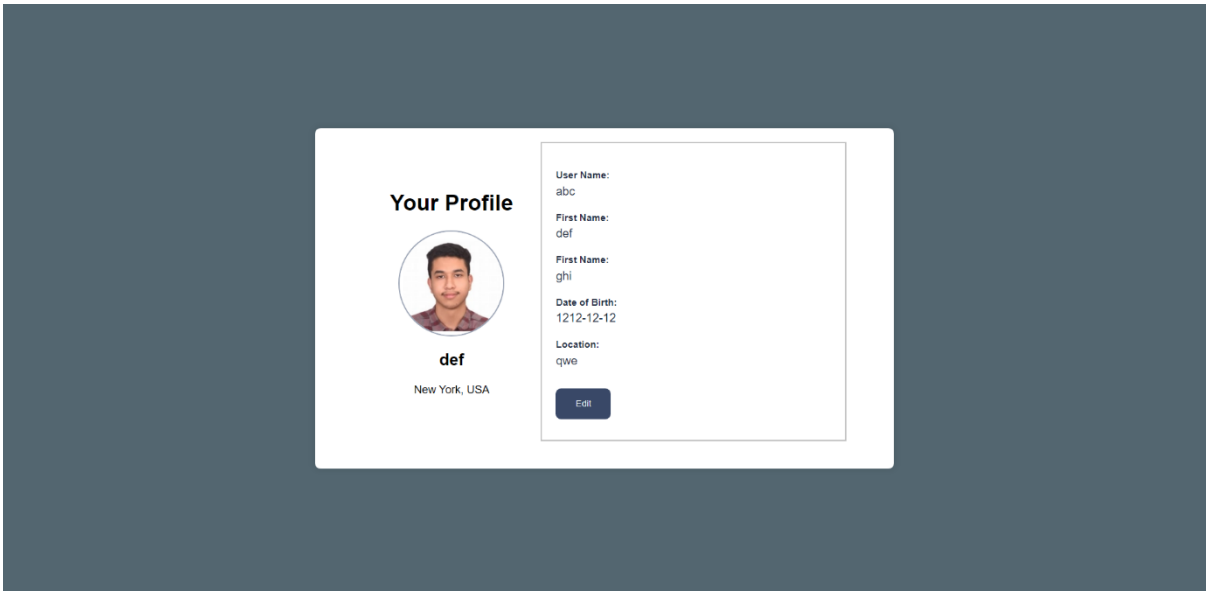


Figure 14: Actual Design for User Profile page

2.2.6 Profile Edit Page

The profile edit page is one of the main functionalities of the e-commerce website. The profile edit page contains the functionality for the users to edit their personal details that they set during the registration process.

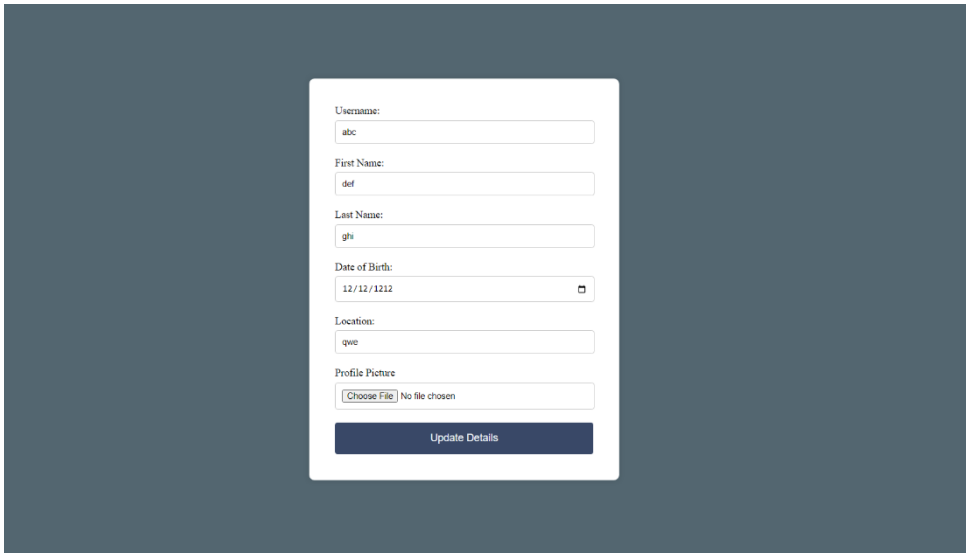


Figure 15: Actual Design for Profile Edit Page

2.2.7 Product page

The product page contains information about the products available to purchase. This includes information such as the Name and price of the product and an option to view more information about the products.

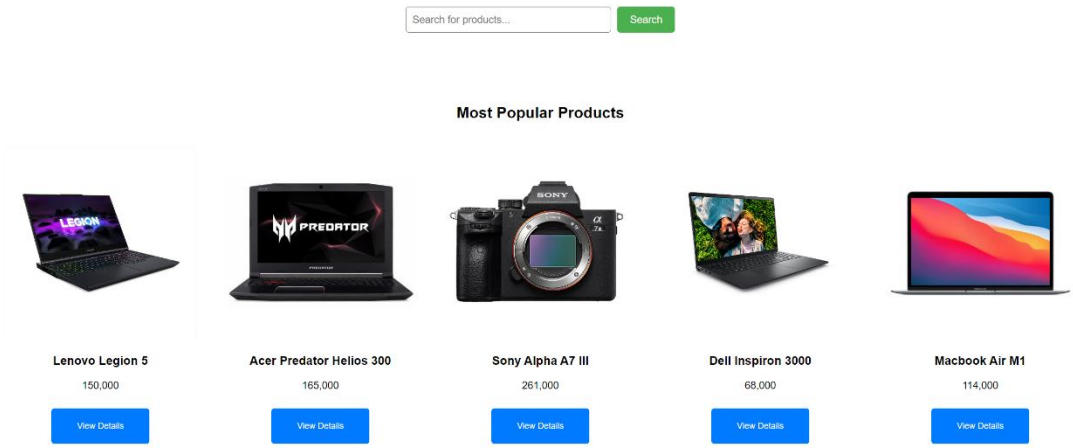


Figure 16: Actual Design for Product Page

2.2.8 Product Detail Page

The product Detail Page displays a detailed information about the product. This includes a high-quality image of the product, further details on its features and its availability. It also implements a call-to-action button such as “Add to Cart” button which allows the users to add the product to the cart page.

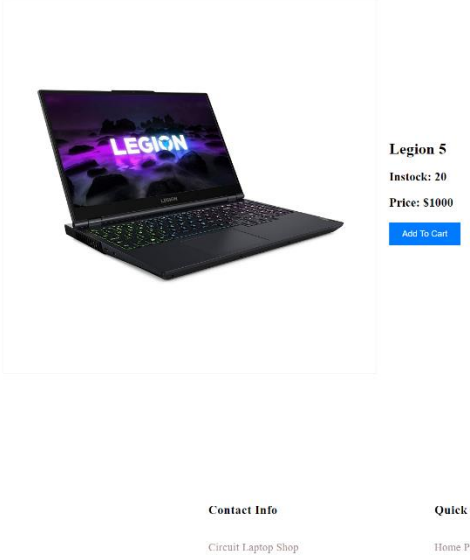


Figure 17: Actual Design for Product Detail page

2.2.9 Admin Panel Page

The Admin Panel Page allows the admins of the website to control and manage the contents of the website. This includes the ability to add or delete products into the system. The main objective of admin panel is to provide overall command of the site to the admin for management of contents of the site.

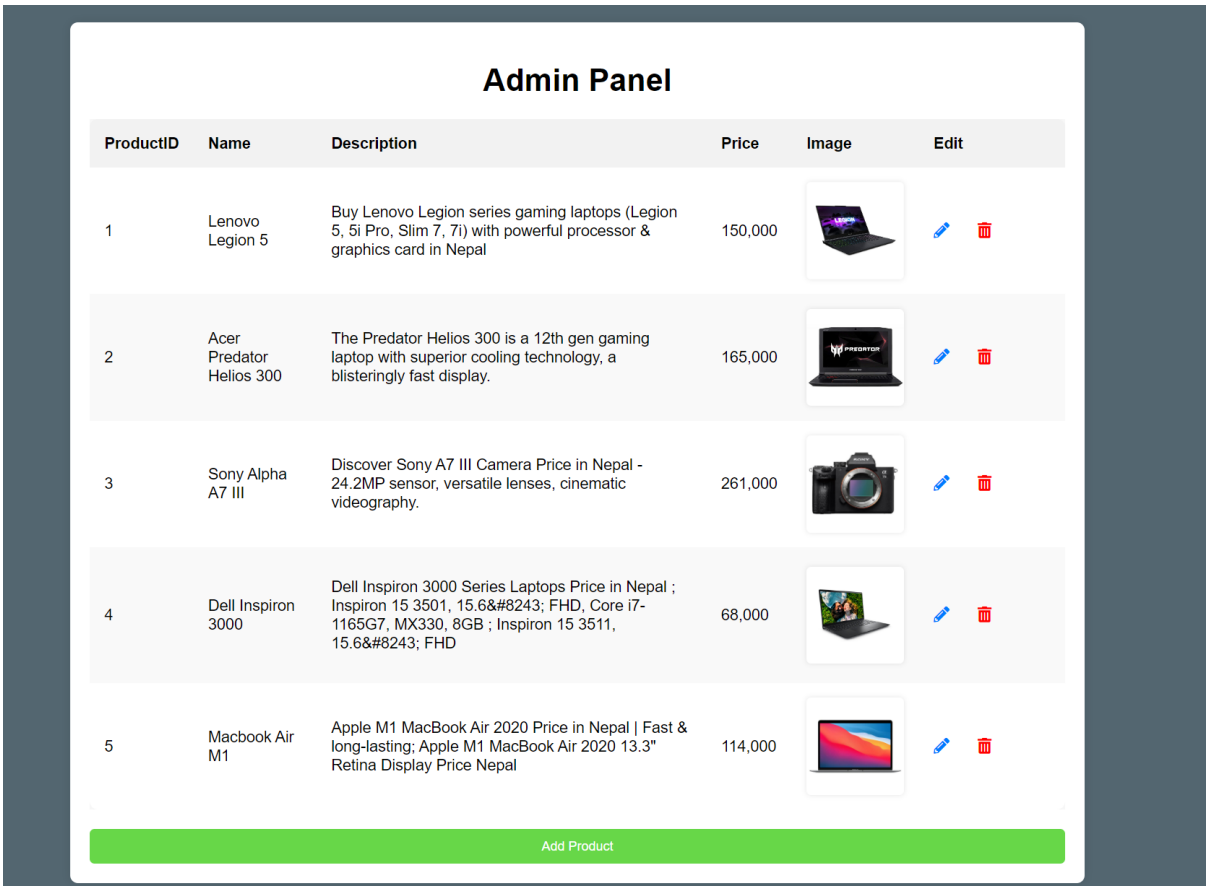


Figure 18: Actual Design Admin Panel page

2.2.10 Add Product Page

The Add Product Page allows the admins to add various products into the system. The product is displayed into the product page after a product has been added. Only admin have the authority to add products.

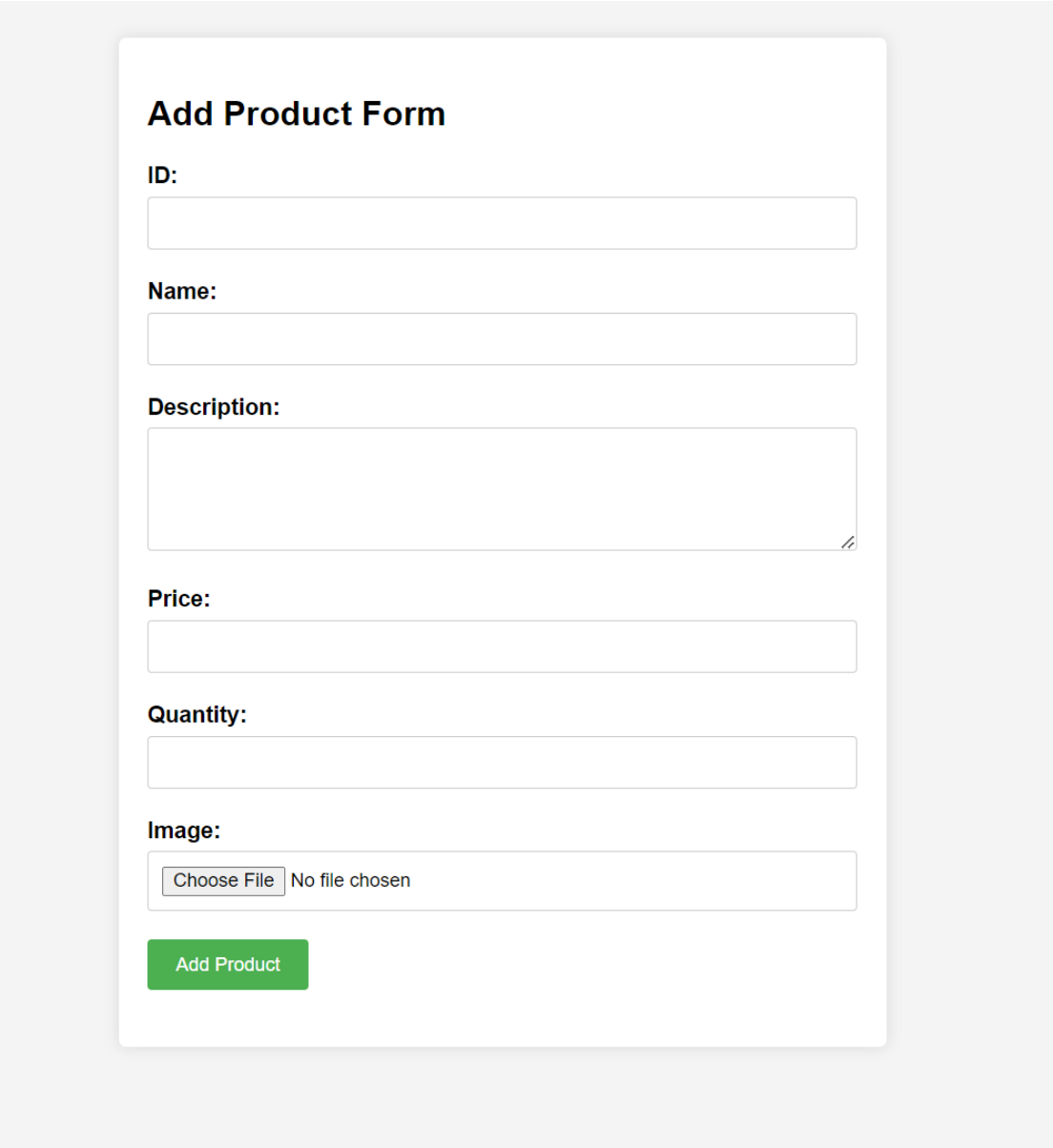


Figure 19: Actual Design for Add Product Page

2.2.11 Order History Page

The order history page contains information about all the products the user has purchased. This includes information such as Name, Price, and the Date of Purchase.

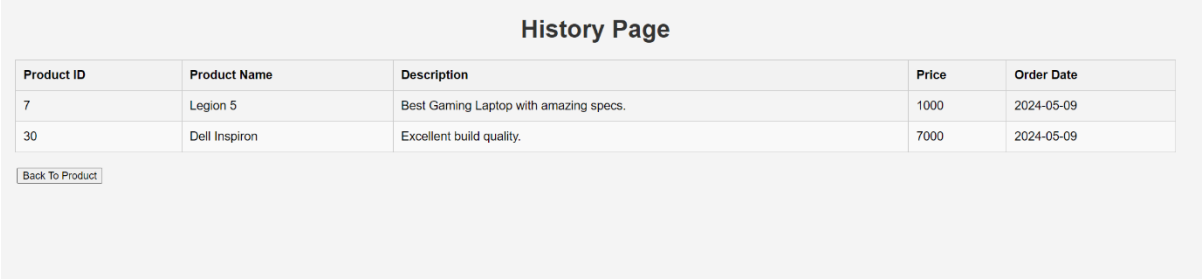


Figure 20: Actual Design for Order History page

3. Class diagram

A class diagram is a static structure diagram which describes the structure of the system. It demonstrates the classes, attributes and the operations of the system and helps visualize the relationship between the objects in the system. (Visual Paradigm, 2024)

3.1 Combined Class Diagram

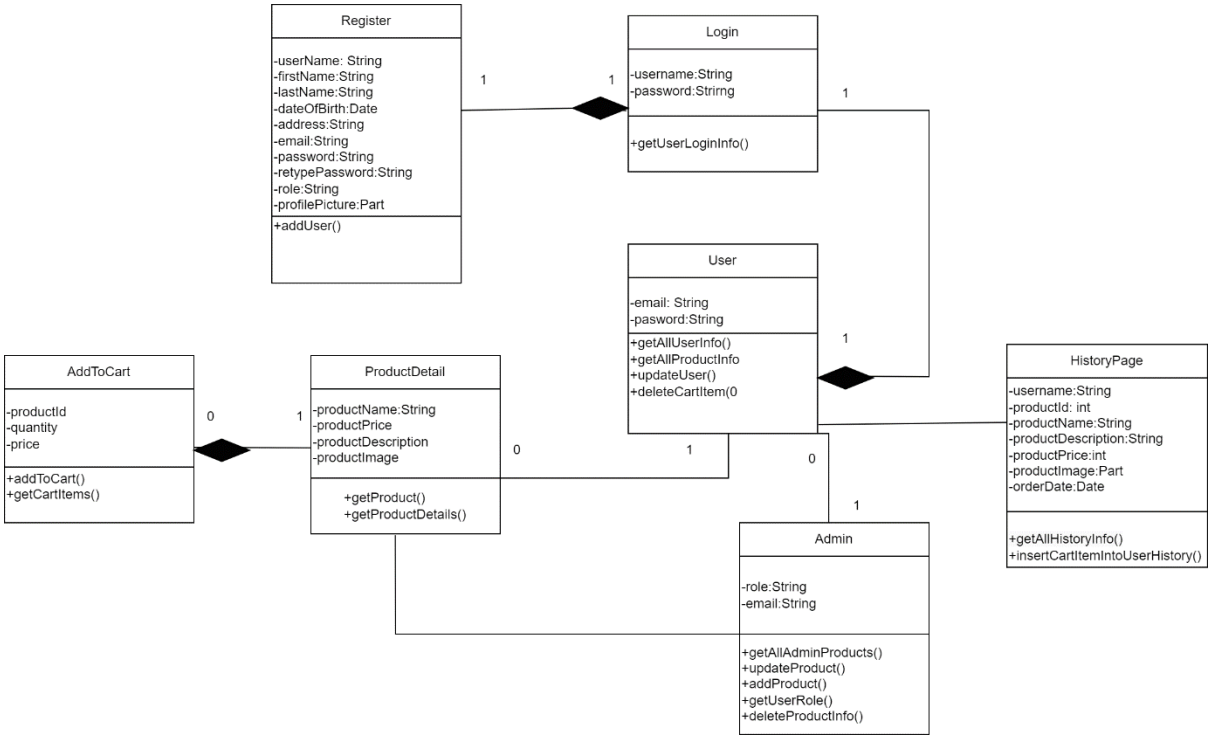


Figure 21: Combined Class Diagram for the system

3.2 Individual Class Diagram

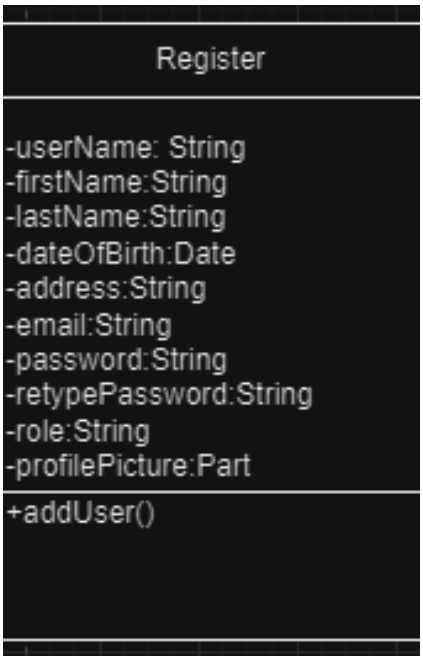


Figure 22: Individual Class Diagram for Register User

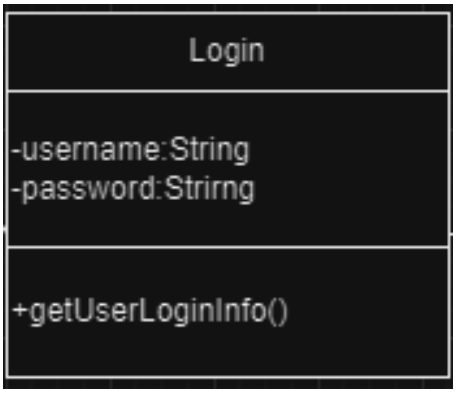


Figure 23: Individual Class Diagram for Login User.

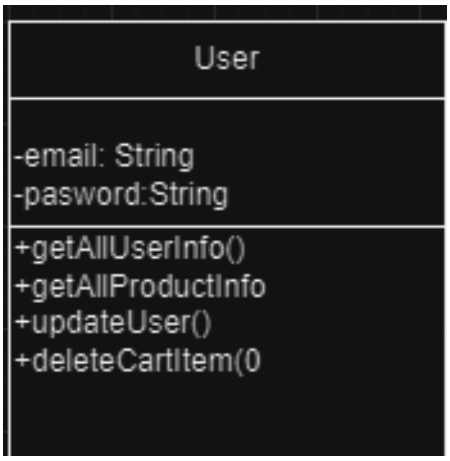


Figure 24: Individual Class Diagram for User.

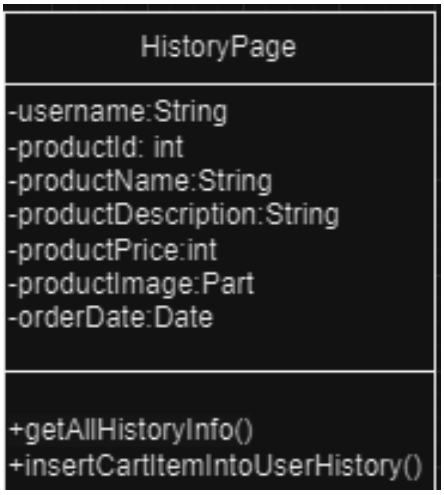


Figure 25: Individual Class Diagram for History Page.

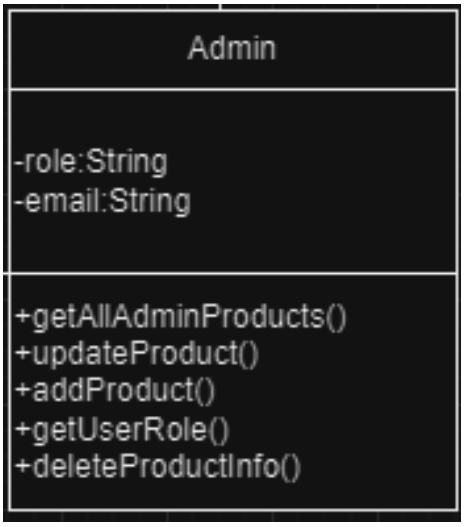


Figure 26: Individual Class Diagram for Admin.

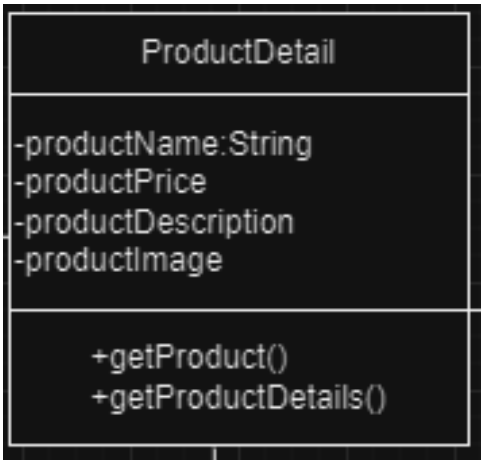


Figure 27: Individual Class Diagram for Product Detail

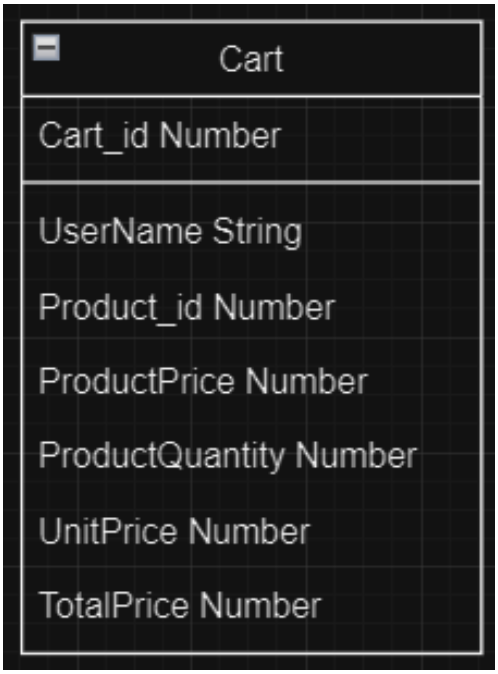


Figure 28: Individual Class Diagram for Cart.

4. Method Description

A method is a block of code that only executes when it is called. Parameters are pieces of data that you can pass into a method. The method name is first defined, followed by parentheses () (W3schools, 2024). A method description describes what the method does and how it functions.

Method	**Description**
doGet()	The doGet method manages HTTP GET requests by extracting username and password parameters from the request. It authenticates these parameters using 'Database Controller' and proceeds accordingly.
doPost()	The doGet method handles HTTP POST requests sent to the servlet. It calls the doGet method to transfer the request processing into it.
doPut()	The doPut method handles HTTP PUT requests which is used to update and update or replace an existing resource on a server.
doDelete()	The doDelete method handles HTTP DELETE requests which can be used to remove a resource from a server. When a file path is
getConnection()	The getConnection method establishes connection with the database returning a connection object.

addUser()	The addUser method establishes connection with the database allowing for adding new user to the database.
getProductListBasedOnSearch Criteria()	The getProductListBasedOnSearchCriteria method allows the retrieval of product from the database.
getUserLoginInfo()	The getUserLoginInfo methods retrives the login information of the user from the database during the login process.
getUserRole()	The getUserRole method handles role. It differentiate functionality of User and Admin.
getUser()	The getUser method gets data of individual registered user.
getProduct()	The getUser method gets data of the individual product added by the admin.
getAllUserInfo()	The getAllUserInfo method gets information of all the user present in the database.
getAllProductsInfo()	The getAllProductsInfo method gets all the products
getAllHistoryInfo()	The getAllHistoryInfo() method gives history of product of each user.
getAllAdminProducts()	The getAllAdminProducts method gives all the products added by the admin.
updateUser()	The updateUser method updates the changes made by the user in user profile.

updateProduct()	The updateProduct method updates the product which is accessed by admin.
deleteUserInfo()	The updateUserInfo method deletes users information from the database.
deleteCartItem()	The deleteCartItem method deletes products from cart by the user.
addProduct()	The addProduct method adds product to product page by admin.
getProductDetails()	The getProductDetail method gets all the detail of an individual product.
addToCart()	The addToCart method adds product to cart by user.
getCartItems()	The getCartItems method gets item from cart database.
insertCartItemsIntoUserHistory()	The insertCartItemsIntoUserHistory transfers the ordered product items into the purchase history page.

Table 1: Method Description Table

5. Test Cases

Testing involves searching for errors before the software is deployed fully. The main objective of testing is to check for possible errors that might have occurred while writing the syntax or when the correct statements were not mentioned (educative, 2023).

5.1 To check username or password mismatch.

TEST NO.	1
Objective:	To check username or password mismatch.
Action:	• While logging in, the username and password both should match that of the database. • If it does not match, an error message is displayed in the form itself.
Expected Result:	Error message should be displayed.
Actual Result:	Error message was displayed.
Conclusion:	The test ran successfully.

Table 2: Test 1 - To check username or password mismatch.

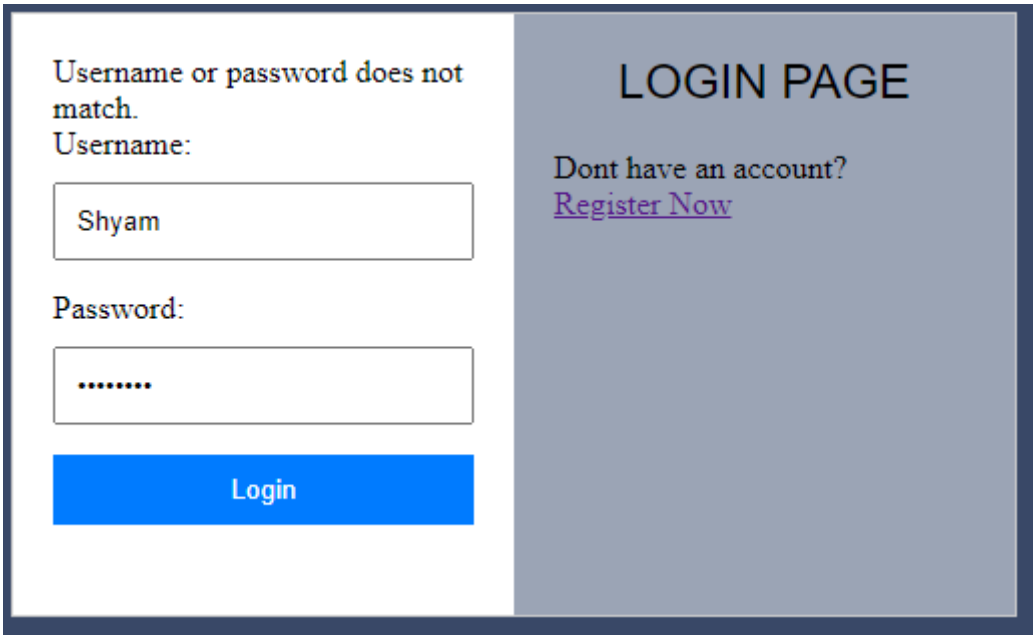
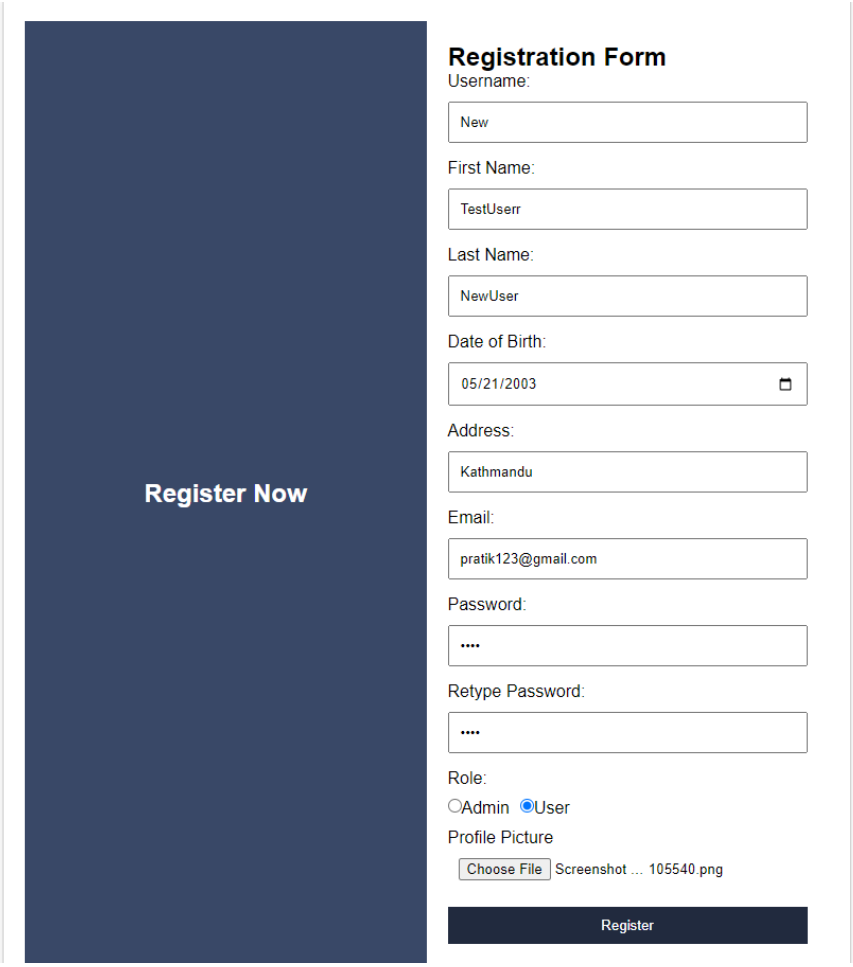


Figure 29: Test 1 - To check username or password mismatch.

5.2 To check uniqueness of username while registering.

TEST NO.	2
Objective:	To check uniqueness of username while registering.
Action:	• Data is populated in the Registration form. • This data is checked in the database. • If the username exists, an error message is displayed.
Expected Result:	Error message should be displayed.
Actual Result:	Error message was displayed.
Conclusion:	The test ran successfully.

Table 3: Test 2 - To check uniqueness of username while registering.



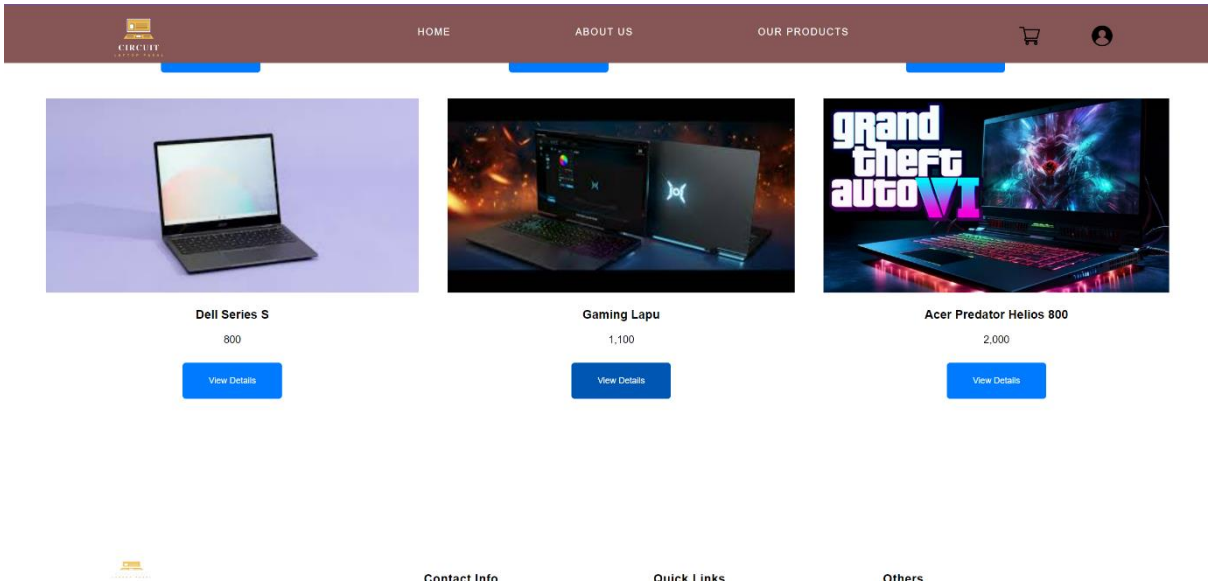

```
Result: -2
Case -2 entered.
```

Figure 30: Test 2 - To check uniqueness of username while registering.

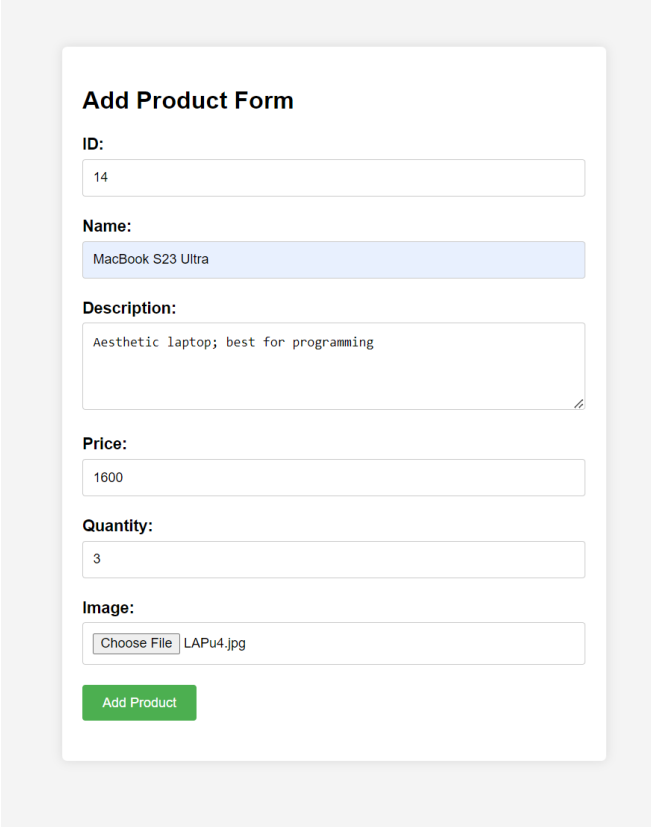
5.3 To check if data is being added to Products Page when Admin adds new products.

TEST NO.	3
Objective:	To check if data is being added to Products Page when Admin adds new products.
Action:	• Data is populated in the Add Product form. • This data is processed in the controller. • It finally gets added to the Products Page.
Expected Result:	Products should be added to product page.
Actual Result:	Products got added to product page.
Conclusion:	The test ran successfully.

Table 4: Test 3 - To check if data is being added to Products Page when Admin adds new products.


HOME
ABOUT US
OUR PRODUCTS
Dell Series S
800
View Details
Gaming Lapu
1,100
View Details
Acer Predator Helios 800
2,000
View Details

Contact Info
Quick Links
Others


Add Product Form
ID:
14
Name:
MacBook S23 Ultra
Description:
Aesthetic laptop; best for programming
Price:
1600
Quantity:
3
Image:
Choose File
LAPu4.jpg
Add Product

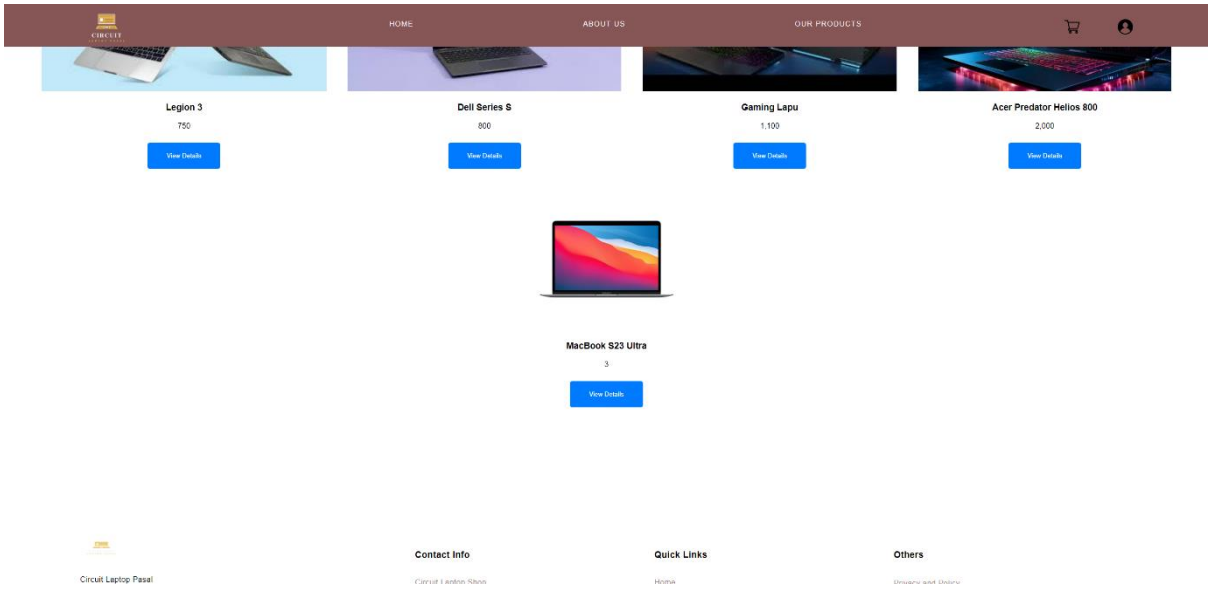
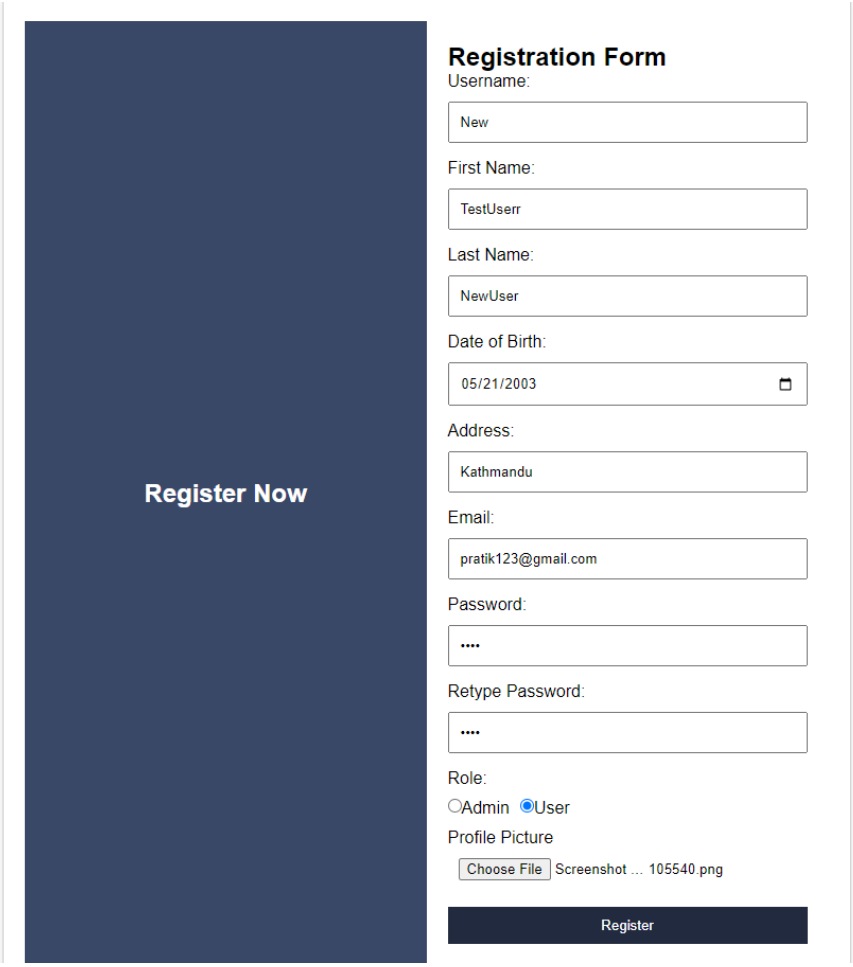


Figure 31: Test 3 - To check if data is being added to Products Page when Admin adds new products.

5.4. To check if registration data is being added to the database.

TEST NO.	4
Objective:	To check if registration data is being added to the database.
Action:	• Data is populated in the Registration form. • This data is processed in the controller. • It finally gets added to the database.
Expected Result:	Data should be added to the database.
Actual Result:	Data got added to the database.
Conclusion:	The test ran successfully.

Table 5: Test 4 - To check if registration data is being added to the database.



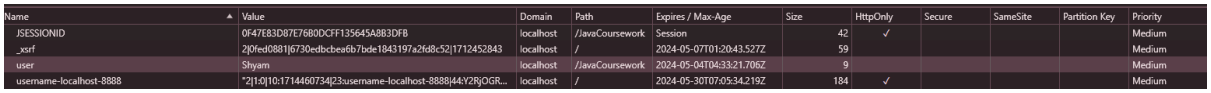
user_name	first_name	last_name	dob	address	email	password	role	image
Dharel	afsd	spf[	1212-12-31	sopw	ds@gmail.com	user	test123	
fdgh	wpier	wsdo	1212-12-12	hpt	g@g	user	test123	
cd	yyy	ttt	3333-03-31	iop	y@gmail.com	user	test	
abhijay	ert	ort	2232-12-31	pyrt	wer@gmail.com	test	user	
krish	uio	poi	2322-12-12	uyt	asdast@gmail.com	qwerty	user	
Sid	trtq	wer[	2392-12-12	yprt	sid@gmail.com	qwerty	user	
Kristan	ert	wostd	2232-12-31	pyrty	kristan@gmail.com	poi	user	
Pratik	siot	pstfdj	1223-12-08	dfgsk	sdkf@gmail.com	ttt	user	
asdas	sdfsf	asdff	1212-12-12	fyurty	rer@gmail.com	poi	user	
Lal	Mogan	yadav	1219-09-09	tep	ert@gmail.com	yyy	user	
Shyam	qwerty	parakr	1212-12-12	ktmm	pratikkk@gmail.com	poi123	admin	
Doja	Cat	Nepal	1212-12-12	pokhara	pkr@gmail.com	qwe	admin	
Sir	maam	pqwe	1212-12-12	psen	W@gmail.com	test	user	
RamK	Kumar	Paija	1212-12-12	ktmm	ptk@gmail.com	test	user	
oerasod	wow	wqpqw	1212-12-12	nepal	p@gmail.com	test	user	
abc	def	ghi	1212-12-12	qwe	o@gmail.com	test	user	pp_pic.jpeg
cdasda	trtsd	rtrt	1212-12-12	yhjkio	as@gmail.com	poi	user	pp_pic.jpeg
New	TestUserr	NewUser	2003-05-21	Kathmandu	pratik123@gmail.com	test	user	Screenshot 2024-02-29 105540.png

Figure 32: Test 4 - To check if registration data is being added to the database.

5.5. To create a session of 30 minutes for the user.

TEST NO.	5
Objective:	To create a session of 30 minutes for the user.
Action:	• A new session of 30 minutes is created and set to cookies. • Cookies store the data of the logged in user for the next 30 minutes.
Expected Result:	Session should be created and saved in cookies.
Actual Result:	Session was created and saved in cookies.
Conclusion:	The test ran successfully.

Table 6: Test 5 - To create a session of 30 minutes for the user.



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
JSESSIONID	0F47E83D87E7680DCFF135645A8B3DFB	localhost	/JavaCoursework	Session	42	✓				Medium
_xsrf	2|0fed0881|6730edbcbea6b7bde1843197a2fd8c52|1712452843	localhost	/	2024-05-07T01:20:43.527Z	59					Medium
user	Shyam	localhost	/JavaCoursework	2024-05-04T04:33:21.706Z	9					Medium
username-localhost-8888	"2|1:0|10:1714460734|23:username-localhost-8888|44:Y2RjOGR...	localhost	/	2024-05-30T07:05:34.219Z	184	✓				Medium

Figure 33: Test 5 - To create a session of 30 minutes for the user.

6. Tools and Libraries Used

Various tools and libraries have been used during the development.

6.1 Tools:

The tools used during the development of this project are:

6.1.1 Eclipse IDE

Eclipse IDE is an open-source software development platform which is popular among Java developers. It provides a workspace where developers can store source files, images, and other necessary files to work on. (tutorialspoint, 2024)

6.1.2 Balsamiq

Balsamiq is an easy-to-use tool used to create wireframes and UI designs. It allows the designers to communicate their vision to the developers without getting into specific details. (balsamiq, 2024)

6.1.3 XAMPP

XAMPP is a local host which allows developers to test websites in their local system before deploying them to a remote web server. It is a free-to-use software which allows for testing of PHP, MySQL projects. (EDUCBA, 2021)

6.1.4 Apache Tomcat 8.5

Apache Tomcat is an open-source web server and servlet that implements the Java Servlet and JSP specifications and is based on Java technology. (Badkar, 2023)

6.1.5 Ms Word

Ms Word is an application that allows to create both simple and complex documentation. It is a widely used commercial word processor that was designed by Microsoft. With the help of Ms Word, you can create, format, edit documents, sharing files and keep track of changes to review (Microsoft Support, 2024).

6.2 Libraries

Library in java is a special project in Integration Designer which is used when developing, managing versions, and organizing resources that have been shared between modules (IBM Business Process Manager, 2021). The libraries used during the development of this project are:

6.2.1 MySQL Connector/J

MySQL Connector/J offers connectivity for client applications created in Java Programming language. Several of the valuable extensions and JDBC API are implemented by Connector/J.

6.2.2 JSTL

JSP Standard Tag Library (JSTL) is a standard tag library which offers tags that controls the behaviour of JSP pages. It can be utilized for iteration and control statements, SQL, etc (Pankaj, 2022).

taglibs-standard-impl-1.2.5

The implementation classes for tag library are contained in this JAR file. It handles the logic when tags are utilized from the standard tag library in the JSP pages.

taglibs-standard-spec-1.2.5

The specification classes for tag library offered by Apache Taglibs are contained in in this JAR file. It follows the standard specifications in the project. It specifies the contracts and interfaces that any tag libraries follow to the standard.

7. Development Process

7.1 Project Planning and Requirement Gathering:

The primary goals for the system are to be defined. The details for the system are gathered. This includes details such as the features and functionalities of the system. The features and functionalities are then prioritized based on their importance towards the system. In this phase the read map of the project is created. All the working flow and the tools that are to be used while developing the site is discussed in this phase.

7.2 Database design:

A database is designed for storing and managing the data more effectively. For this a table is to be normalized which reduces the chance for duplication of data which improves data integrity. After normalization, the tables are created for storage of data. The tables created for the system are product table, user info and cart table. Having a database helps to secure all the data associated with the website and the user.

7.3 Development environment setup:

This process requires to install all the required software and necessary tools such as IDE, jar files etc for the development of the project. For this project, Eclipse was used as an IDE.

In this process, the environment in which the system is built is set up. Necessary tools such as an IDE and server are setup development of the system. For this project, the choice of IDE was the Eclipse IDE and Xampp was used as the local server environment. Jar files and Taglib files are also setup which will be beneficial for of the development process.

7.4 Create MVC Directory Structure:

The Model-View-Controller (MVC) architectural pattern is used for the development of the system. For this, the files are created and organized to their specified directories. Model contains the business logic of the system. It handles operations such as data retrieval and data manipulation of the

system. View is responsible for displaying the Interface of the system. Controller acts as the bridge between Model and View. It received the input from the user, processes the input and sends an appropriate response to the user. This pattern helps to break frontend and backend code into separate components. In MVC pattern Model contains backend that contains all data logics, View contains all the frontend or GUI and Controller contains logics that controls how data is displayed.

7.5 Development of controller layer:

Here all the mode classes like Product, User, and Order are created, and CRUD operation is implemented for interaction with the database.
Controller layer handles the requests the user and provides a proper response to the user. Various model classes such as Product and User are created here. CRUD operations are also implemented. It represents the functionalities for managing data in the database.

7.6 User Interface Design:

In this process, the user interface is designed for user-friendly interaction with the end user. In designing phase team came up with the prototype in which multiple wireframes were designed. It allows integration of java code into HTML pages. It enables use of custom tags or tag libraries to create various components. This encourages the reuse of code making the development and maintenance process smoother. Various JSP files are created to make it easier to generate Java based dynamic web applications. Here JSP file and CSS are applied for client-side functionality.

7.7 Integration of View with Controller:

In this process, the routing and handling of user requests is done. When a user interacts with the interface, an HTTP request is triggered in the system which sends the request to the server. The request is then taken by the controller. It then decides how to respond to the request based on the parameters or other information provided.

7.8 Shopping Cart implementation:

This is one of the main functionalities of the project, here all shopping carts are managed by the user and proceed to checkout. Shipping cart holds multiple products that the user would select to purchase before checking out. For completion of cart JSP and CSS used for its user interface, a servlet was made to request and respond to the user. All the navigation variables are defined in the string util file. The user is also capable of removing the items that are not necessary.

7.9 Testing:

Before the deployment of the system, all the functionalities are tested. This helps find flaws and errors on the website before fully deploying the system. It checks the quality, functionality, and the performance of the system. Testing also helps to show vulnerabilities of the website, it ensures the quality and integrity of the website.

7.10 Deployment:

After all the test cases are completed, the site is ready to be deployed the site to the production environment. Deploying website gives access to the user to use the services provided by website and stay update with the site.

To summarize, the development of dynamic e-commerce website was carried out using MVC pattern. The completion of the development process went smoothly by following certain steps. The process for developing dynamic e-commerce website were planning and resource gathering, database design, environment setup, creating directory structure according to MVC pattern, development of controller layer, shopping cart implementation, and testing.

8. Critical Analysis

Several challenges were faced during the development of an e-commerce website for electronics and gadgets. Some of them are discussed below:

i. Database management

Designing and managing the database schema efficiently in a group was a real challenge. Since we were using the MySQL localhost and not an online database, the process of sharing the database and having to import it repeatedly while making any change in it was a real hassle. One of the mail hassle is to work with Xampp as MySQL wasn't functioning properly due to various reasons, one of the main reason was clash between the port numbers.

ii. Implementation of the MVC pattern

The implementation of the MVC pattern and integrating it in the project was challenging. Dividing the project in the three packages required a proper understanding of the MVC design pattern.

MCV is one of the most popular patterns used for modern web applications as it allows the application to be scalable, maintainable and easy to expand. It involves separation of software into three main components Model, View, and Controller which makes code modular and maintainable. This pattern helps to break frontend and backend code into separate components. In MVC pattern Model contains backend that contains all data logics, View contains all the frontend or GUI and Controller contains logics that controls how data is displayed.

User Experience:

Maintaining good User Experience for the users was a big challenge. For better user experience the group worked on navigation which helps to redirect user to desired page. Making the website was another big challenge that the group faced. It is important for website to be responsive for smooth usage in all the devices.

Functionality:

Functionality is the main feature of the website. In this website there are multiple functionalities involved to make the website work and maintain the user experience. The main functionality of this project is login, registration, add to cart and check out process. It was a big challenge for the group to implement all the functionalities. The main functionalities where most of the problems faced were with add to cart, admin panel to add and delete products, update etc.

Testing:

Before deploying the website, it is necessary to run various test to ensure the quality of the website. Testing helps to detect bugs and errors of the website; it also helps identify smooth and proper functioning of the website from user's perspective. It was a grate challenge to cope up with all the tests. A lot of time was utilized on research to find solutions to the bugs and pass all the test cases.

9. Conclusion

To conclude, this project of developing an e-commerce website for a Laptop Shop has been a challenging yet fun learning experience. Implementation of various functionalities that are essential in real-life websites has provided invaluable insights into how features are integrated and their interdependencies.

Amongst the challenges, database management was one of the most challenging features. Storing user data, production information, and various other information through the help of the website and then pulling the data for the data was a challenging feature to learn. In addition, user authentication was also difficult as it needed appropriate validation and handling of user credentials.

In essence, the learning and implementing process of the e-commerce website has helped equip valuable skills that will be helpful in the future. Developing a full stack website using the MVC pattern brought many challenges which was rewarding. We learned about the tools used in the industry and working flow of the dynamic website that are used in industrial level.

References

Apache Friends, 2024. *XAMPP installers and Downloads for Apache Friends.* [Online] Available at: https://www.apachefriends.org/ [Accessed 4 May 2024].

Badkar, A., 2023. *What is Tomcat? Everything You Need to Know.* [Online] Available at: https://www.simplilearn.com/what-is-tomcat-article#:~:text=Tomcat%20is%20an%20open%2Dsource,JavaServer%20Pages%20(JSP)%20specifications. [Accessed 4 May 2024].

balsamiq, 2024. *The effortless wireframing tool built for your big ideas..* [Online] Available at: https://balsamiq.com/ [Accessed 4 May 2024].

EDUCBA, 2021. *What is XAMPP?.* [Online] Available at: https://www.educba.com/what-is-xampp/ [Accessed 9 May 2024].

Figma, 2024. *What is wireframing?.* [Online] Available at: https://www.figma.com/resource-library/what-is-wireframing/ [Accessed 7 May 2024].

freecodecamp, 2024. *The Model View Controller Pattern – MVC Architecture and Frameworks Explained.* [Online] Available at: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/ [Accessed 5 May 2024].

IBM Business Process Manager, 2021. *Libraries and JAR files overview.* [Online] Available at: https://www.ibm.com/docs/bg/bpm/8.5.7?topic=modules-libraries-jar-files-overview [Accessed 9 May 2024].

Microsoft Support, 2024. *What is Word?.* [Online] Available at: https://support.microsoft.com/en-us/office/what-is-word-aee9c7ff-f9c5-415f-80dc-103ad5e344d7 [Accessed 9 May 2024].

Pankaj, 2022. *JSTL Tutorial, JSTL Tags Example.* [Online] Available at: https://www.digitalocean.com/community/tutorials/jstl-tutorial-jstl-tags-example [Accessed 9 May 2023].

tutorialspoint, 2024. *Eclipse - Overview.* [Online] Available at: https://www.tutorialspoint.com/eclipse/eclipse_overview.htm [Accessed 4 May 2024].

Visual Paradigm, 2024. *What is Class Diagram?.* [Online] Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/ [Accessed 7 May 2024].

W3schools, 2024. *Java Methods.* [Online] Available at: https://www.w3schools.com/java/java_methods.asp [Accessed 9 May 2024].