



GIFT WEBSITE

INTERNSHIP PROJECT REPORT

Submitted by

R.Manish Vedanayagam
(Register No.: 95192201051)

Third year student of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

P.S.R. ENGINEERING COLLEGE, SIVAKASI –
626140

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2025

BONAFIDE CERTIFICATE

Certified that this project report "**GIFT WEBSITE** " is the bonafide work of **R.Manish Vedanayagam (95192201051)**, " who carried out the project work under my supervision.

SIGNATURE

Mrs.Arthi Venkatesh
EXTERNAL SUPERVISOR
Corporate Trainer,
Evoriea Infotech Private Limited
Bangalore – 560076.

SIGNATURE

Mr. Mohamed Nawfal A
TEAM LEADER
Corporate Trainer – Head,
Evoriea Infotech Private Limited
Bangalore – 560076

Submitted to the department for the internship report evaluation on_____.

PROJECT COORDINATOR

HOD/CSE

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the following individuals and institutions who have contributed to the successful completion of the Gift Website project.

Evoriea Infotech Private Limited, Bengaluru:

Mrs.Arthi Venkatesh, Corporate Trainer, for providing valuable guidance, mentorship, and constructive feedback throughout the project development process. Your expertise has been instrumental in shaping the project.

P.S.R Engineering College, Sivakasi:

We unreservedly express our indebtedness to our Managing Trustee and Correspondent **Thiru. R. SOLAISAMY** and Director **Er. S. VIGNESWARIARUNKUMAR** for providing the needed resources and facilities.

It is our privilege to convey my sincere thanks to our respected Principal **Dr. J.S. SENTHIL KUMAAR M.E., Ph.D.**, for giving us permission to do this project in our organization.

We wish to extend our sincere thanks to our adored Head of the Department **Dr. A. RAMATHILAGAM M.E., Ph.D.**, Professor for her motivation during this period of this internship project work.

Their contributions have significantly enhanced the overall quality and success of the "Gift Website" project.

Abstract

The **Online Gift Shopping Website** is a web-based platform designed to offer users a seamless and convenient shopping experience for gifts. Developed using **HTML, CSS, and JavaScript** for the frontend and **Java (Spring Boot) with MySQL** for the backend, this system ensures a dynamic and efficient e-commerce environment. The primary objective of this project is to create a **user-friendly and responsive shopping platform** where customers can browse gift items, add them to their cart, and securely place orders.

The project is built on a **Spring Boot framework**, ensuring robust backend support, while **MySQL** is used as the database to manage user details, product information, and order history. The **Spring Tool Suite (STS)** is used as the primary development environment to streamline the coding process. This website enhances the overall shopping experience with its well-integrated features and smooth navigation.

The **Online Gift Shopping Website** serves as an efficient e-commerce solution that simplifies the process of purchasing gifts. With a **secure authentication system, well-organized product management, and a seamless checkout experience**, the website ensures customer satisfaction. Future enhancements may include additional features such as **order tracking, customer reviews, and an admin panel** to manage products and orders effectively.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	4
1	INTRODUCTION	6
	1.1 INTRODUCTION TO WEB DEVELOPMENT	6
	1.2 INTRODUCTION TO HTML, CSS, JAVASCRIPT	6
	1.3 INTRODUCTION TO JAVA(SPRINGBOOT)	6
	1.4 INTRODUCTION TO MYSQL DATABASE	7
	1.5 INTRODUCTION TO PROJECT	8
2	LITERATURE REVIEW	9
3	SYSTEM ANALYSIS	12
	3.1 EXISTING SYSTEM	12
	3.2 PROPOSED SYSTEM	12
	3.3 OBJECTIVES	13
4	MODULES	14
	4.1 HOME PAGE	14
	4.2 PRODUCT PAGE	14
	4.3 CART PAGE	15
	4.4 LOGIN PAGE	15
	4.5 SIGNUP PAGE	16
5	APPENDIX	17
6	RESULT ANALYSIS	48
7	CONCLUSION	50
8	REFERENCES	52

CHAPTER 1

INTRODUCTION

1.1 Introduction to Web Development

Web development refers to the process of creating, building, and maintaining websites for the internet or intranet. It involves front-end development (user interface and experience), back-end development (server-side logic and database management), and full-stack development (both front-end and back-end). The primary goal of web development is to provide interactive, dynamic, and user-friendly web applications.

Web development can be categorized into:

- **Static Web Development:** Websites with fixed content, built using only HTML and CSS.
- **Dynamic Web Development:** Websites with interactive and real-time content updates, often built using JavaScript and server-side technologies like Java Spring Boot.
- **E-Commerce Development:** Websites for online shopping, requiring database management, payment integration, and user authentication.

Modern web development follows responsive design, ensuring compatibility across multiple devices, including desktops, tablets, and smartphones.

1.2 Introduction to HTML, CSS, and JavaScript

HTML (HyperText Markup Language)

HTML is the foundation of web pages, providing the basic structure using elements like headings, paragraphs, images, links, forms, and tables. It works alongside CSS and JavaScript to create fully functional websites.

CSS (Cascading Style Sheets)

CSS is used to design and style web pages, allowing customization of layout, fonts, colors, spacing, and animations. Modern CSS includes frameworks like Bootstrap and Tailwind CSS, which enhance the development speed by providing pre-designed components.

JavaScript (JS)

JavaScript is a client-side programming language that adds interactivity to web pages, enabling features like dynamic content updates, form validation, animations, and API integration. JavaScript frameworks like React, Angular, and Vue.js improve efficiency by providing structured development environments.

1.3 Introduction to Java (Spring Boot)

Java Overview

Java is a high-level, object-oriented programming language known for its platform independence through the Java Virtual Machine (JVM). It is widely used for backend web development due to its security, scalability, and extensive frameworks.

Spring Boot Framework

Spring Boot is a Java-based framework that simplifies backend development by providing built-in support for dependency management, embedded servers, and database connectivity. It allows developers to build robust, scalable, and production-ready web applications with minimal configuration.

Key Features of Spring Boot

- **Microservices Architecture:** Supports developing independent, modular services.
- **Embedded Web Server:** Comes with built-in Tomcat, Jetty, or Undertow servers.
- **Spring Security:** Ensures authentication and authorization mechanisms.
- **Spring Data JPA:** Simplifies database interactions.

Spring Boot is widely used for enterprise applications, e-commerce platforms, and RESTful web services.

1.4 Introduction to MySQL Database

Overview of MySQL

MySQL is an open-source relational database management system (RDBMS) that allows structured data storage using tables. It follows the Structured Query Language (SQL) for **managing and querying data**.

Why Use MySQL?

- **Scalability:** Can handle large volumes of data efficiently.
- **Security:** Provides encryption, authentication, and access control.
- **Data Integrity:** Supports ACID (Atomicity, Consistency, Isolation, Durability) properties.
- **Integration:** Works seamlessly with Java Spring Boot for database connectivity.

MySQL in the Project

In this project, MySQL is used to store and manage:

- User details (registration, login, authentication)
- Product information (gadget categories, descriptions, prices)
- Shopping cart details (selected items, quantities, total price)

1.5 Introduction to Project

The Online Gift Shopping Website is an e-commerce platform designed to provide a seamless online shopping experience for purchasing gifts. The website is developed using HTML, CSS, JavaScript for the frontend, Java Spring Boot for backend logic, and MySQL for database storage.

Key Features of the Project:

- User Authentication System: Secure login and registration functionality.
- Product Listing and Categorization: Gifts are organized into categories for easy navigation.
- Shopping Cart Functionality: Users can add products to their cart and proceed to checkout.

The project aims to enhance the online shopping experience by providing a user-friendly interface, secure transactions, and efficient product management.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Online shopping has transformed the retail industry, shifting from traditional brick-and-mortar stores to digital platforms that offer convenience, personalization, and seamless transactions. Various studies have analyzed the role of e-commerce platforms, database management, secure payment gateways, and user authentication in improving online shopping experiences. This chapter explores the existing research on e-commerce applications, focusing on Java-based web applications, Spring Boot integration, database efficiency, and security measures.

2.2 Traditional vs. Modern Online Shopping Systems

2.2.1 Traditional Shopping Methods

- Before digital transformation, customers physically visited stores to purchase goods, which required time and effort.
- Product availability and pricing information were limited to store inventory.
- Payments were predominantly cash-based, leading to transactional inefficiencies.

2.2.2 Evolution of E-Commerce

- E-commerce platforms emerged with the advent of the internet, enabling customers to shop online.
- Websites initially provided static product catalogs, requiring manual processing of orders.
- Modern web applications use dynamic frameworks (Spring Boot, React, Angular) to provide an interactive shopping experience with real-time inventory updates.

2.3 Web-Based E-Commerce Platforms

2.3.1 API-Driven E-Commerce Systems

- Many modern shopping websites leverage APIs for product management, user authentication, and payment processing.
- RESTful APIs allow seamless integration with external services such as payment gateways (PayPal, Stripe), inventory management, and user authentication (OAuth, JWT).
- Java applications utilize Spring Boot's REST controllers for secure communication between frontend and backend.

2.3.2 Secure Online Transactions

- Secure online transactions are essential for preventing fraud and data breaches.

- E-commerce platforms use SSL encryption, token-based authentication (JWT), and secure database storage (MySQL) for financial transactions.
- Spring Security framework provides role-based access control and encrypted password storage to protect user credentials.

2.4 Importance of Database Management in E-Commerce

2.4.1 Role of MySQL in E-Commerce Applications

- MySQL is widely used for storing product details, user information, orders, and transaction history in a structured format.
- SQL queries with indexing enhance search performance and data retrieval speed for large-scale e-commerce platforms.

2.4.2 Optimized Data Handling Techniques

- ORM frameworks like Spring Data JPA simplify database interactions by providing automatic query execution and entity mapping.
- Caching mechanisms (Redis, Hibernate Caching) are used to reduce database load and improve response time.

2.5 Java-Based E-Commerce Applications

2.5.1 Spring Boot for Backend Development

- Spring Boot simplifies backend development by providing pre-configured settings, RESTful API support, and microservices architecture.
- Java-based e-commerce platforms use Spring Boot controllers to handle product listing, user authentication, and order management.

2.5.2 Frontend Technologies for UI Design

- HTML, CSS, and JavaScript frameworks like Bootstrap ensure responsive and user-friendly interfaces.
- AJAX-based asynchronous updates improve shopping experiences by enabling dynamic cart updates and real-time product availability checks.

2.6 Challenges in Existing E-Commerce Systems

2.6.1 Security and Privacy Risks

- Online shopping platforms are prone to cybersecurity threats, including hacking, phishing, and payment fraud.
- Solution: Implementing Spring Security, JWT authentication, and database encryption to protect sensitive data.

2.6.2 Performance and Scalability Issues

- High user traffic can slow down website performance and affect order processing speed.
- Solution: Using MySQL indexing, efficient query execution, and cloud-based load balancing to optimize system performance.

2.6.3 User Authentication and Role Management

- Unauthorized access can compromise user data and order transactions.
- Solution: Implementing role-based access control (RBAC) and two-factor authentication (2FA) for secure user management.

2.7 Future Enhancements in E-Commerce Applications

- Integration of AI-driven recommendations to provide personalized product suggestions.
- Progressive Web Apps (PWAs) for a mobile-friendly shopping experience without requiring a dedicated app.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

3.1.1 Limitations of Traditional E-Commerce Platforms

- Many existing online shopping platforms face issues related to security, performance, and user experience.
- Some systems lack proper authentication mechanisms, making them vulnerable to data breaches and unauthorized access.
- Database inefficiencies lead to slow query execution and delays in retrieving product details or processing orders.
- Most shopping websites are not optimized for scalability, causing performance degradation with increasing traffic.

3.1.2 Challenges in Current Online Shopping Systems

- **Security Risks:** Lack of secure authentication and encryption can result in fraud and data leaks.
- **Performance Issues:** High traffic causes slow response times due to inefficient database queries.
- **Limited Customization:** Many platforms do not allow personalized recommendations or tailored shopping experiences.
- **User Authentication Issues:** Weak login systems lead to unauthorized access to user accounts.

3.2 Proposed System

3.2.1 Overview of the Proposed E-Commerce System

The proposed system is an online shopping website exclusively for gadgets, developed using Java (Spring Boot), MySQL, HTML, CSS, and JavaScript. It provides secure user authentication, seamless shopping experiences, and optimized database management to overcome the limitations of existing platforms.

3.2.2 Key Features of the Proposed System

- **User Authentication & Access Control:**
 - Secure signup/login system using Spring Security and JWT authentication.
 - Restricts unauthorized users from accessing product pages, cart, or checkout.
- **Efficient Product Management:**
 - Displays gifts specifications.

- Stores and retrieves product details efficiently using MySQL database.
- Shopping Cart Functionality:
 - Users can add products to the cart and proceed to checkout.
 - Cart items are stored using session management and database integration.
- Optimized Performance & Scalability:
 - Uses Spring Boot with MySQL indexing for fast data retrieval.
 - Efficient API calls reduce server load and improve response times.
- Secure Payment Gateway Integration:
 - Supports multiple payment methods with secure transaction handling.
 - Implements SSL encryption for payment processing.
- User-Friendly Interface:
 - Developed with responsive UI components using HTML, CSS, and JavaScript.
 - Ensures smooth navigation across different devices.

3.3 Objectives

The primary goal of this project is to develop a secure, user-friendly, and scalable e-commerce platform for gift shopping. The specific objectives include:

1. Enhancing Security
 - Implementing JWT authentication to restrict unauthorized access.
 - Encrypting sensitive user data for secure transactions.
2. Optimizing Performance & Database Management
 - Using Spring Boot and MySQL indexing to speed up queries.
 - Ensuring faster page loads and quick product retrieval.
3. Providing a Seamless User Experience
 - Designing an intuitive UI with easy navigation.
 - Implementing a cart and checkout system for smooth transactions.
4. Enabling Scalable & Future-Ready Architecture
 - Ensuring that the platform can handle high traffic loads.

CHAPTER 4

MODULES

This chapter provides a detailed explanation of the key modules implemented in the Online Gift Shopping Website. Each module plays a crucial role in ensuring a smooth, secure, and efficient shopping experience for users.

4.1 Home Page

Overview:

The Home Page serves as the entry point for users visiting the website. It is designed with an attractive UI, featuring a navigation bar, promotional banners, featured products, and discount details.

Key Features:

- **Navigation Bar:** Provides links to different pages, including Product Page, Cart, Login,
- **Featured Products:** Displays top-selling gadgets with images and brief descriptions.
- **Promotional Offers & Discounts:** Highlights ongoing sales and deals for users.
- **User Authentication Status:**
 - If the user is logged in, their name and profile details are displayed.
 - If the user is not logged in, a "Login/Signup" button is shown.

4.2 Product Page

Overview:

The Product Page displays all available gifts, categorized by, specifications, and price range. Users can view details about each product and add items to their cart.

Key Features:

- **Product Categorization:**
 - Products are grouped based on events (e.g., Birthday, Wedding, Proposal).
 - Filters allow users to sort by price, processor, screen size, and RAM.
- **Product Details Section:**
 - Displays product image, name.
 - Includes "Add to Cart" and "Buy Now" buttons.
- **Dynamic Content Loading:**

- Products are retrieved from the MySQL database and displayed dynamically using Spring Boot APIs.

4.3 Cart Page

Overview:

The Cart Page stores all products selected by the user. It provides an overview of items added, their total cost, and options to proceed with the checkout process.

Key Features:

- Cart Item Display:
 - Shows product image, name, price, and remove option.
- Total Price Calculation:
 - Automatically calculates the total cost of all selected items.
- Checkout Option:
 - Users can proceed to the payment page to complete the purchase.
- Remove from Cart:
 - Allows users to remove items they no longer want.
- Persistent Storage:
 - Cart data is stored using local storage and retrieved from MySQL for logged-in users.

4.4 Login Page

Overview:

The Login Page ensures that only authenticated users can access the platform. It uses Spring Security and MySQL to validate user credentials.

Key Features:

- User Authentication:
 - Users enter their email and password to log in.
 - Backend validation using Spring Boot and MySQL.
- Access Control:
 - If a user is not logged in, they cannot access the Cart Page or Checkout.
- Session Management:
 - Once logged in, user details are stored in a session for future access.

- Error Handling:
 - Displays error messages for incorrect credentials.
- Forgot Password Option:
 - Provides an option to reset the password via email verification.

4.5 Signup Page

Overview:

The Signup Page allows new users to create an account on the platform. It ensures data validation and securely stores user information in the MySQL database.

Key Features:

- User Registration Form:
 - Users provide name, email, phone number, password, and address.
- Data Validation:
 - Ensures email format correctness, strong password creation, and non-empty fields.
- Secure Password Storage:
 - Passwords are encrypted using Spring Security's BCrypt hashing algorithm.
- Duplicate Check:
 - Prevents users from registering with an already existing email ID.
- Successful Registration Notification:
 - Displays a success message and redirects users to the Login Page after registration.

CHAPTER 5

APPENDIX

Figure 1 : Home page

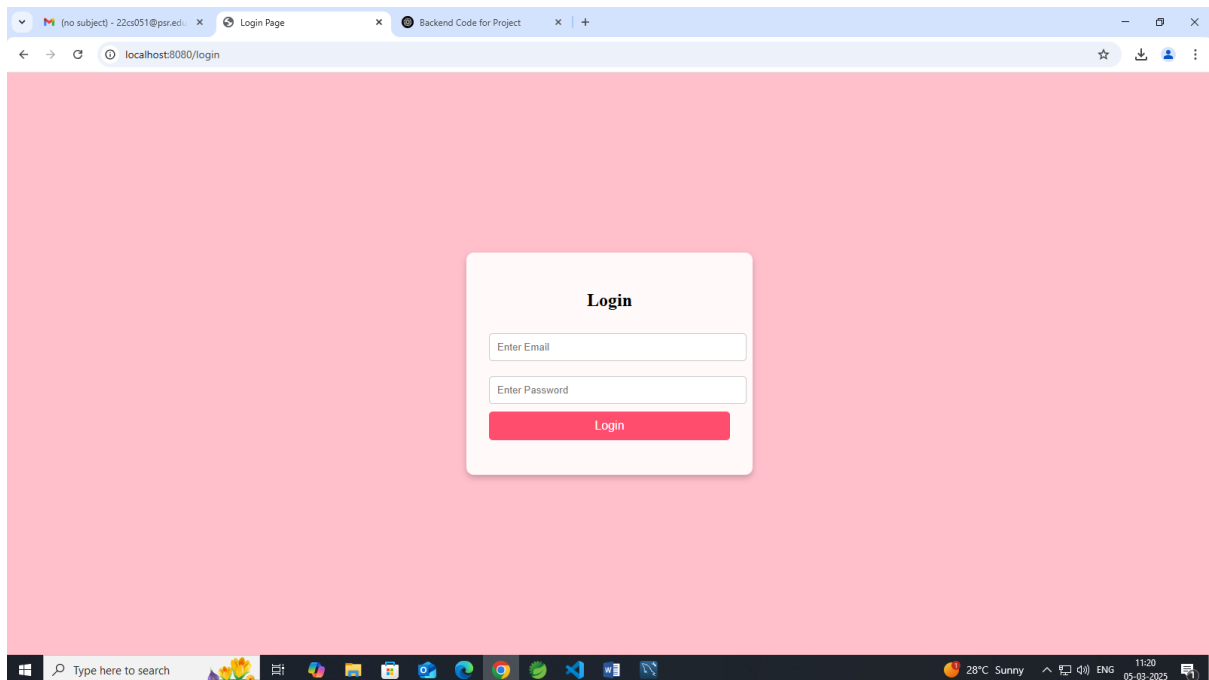
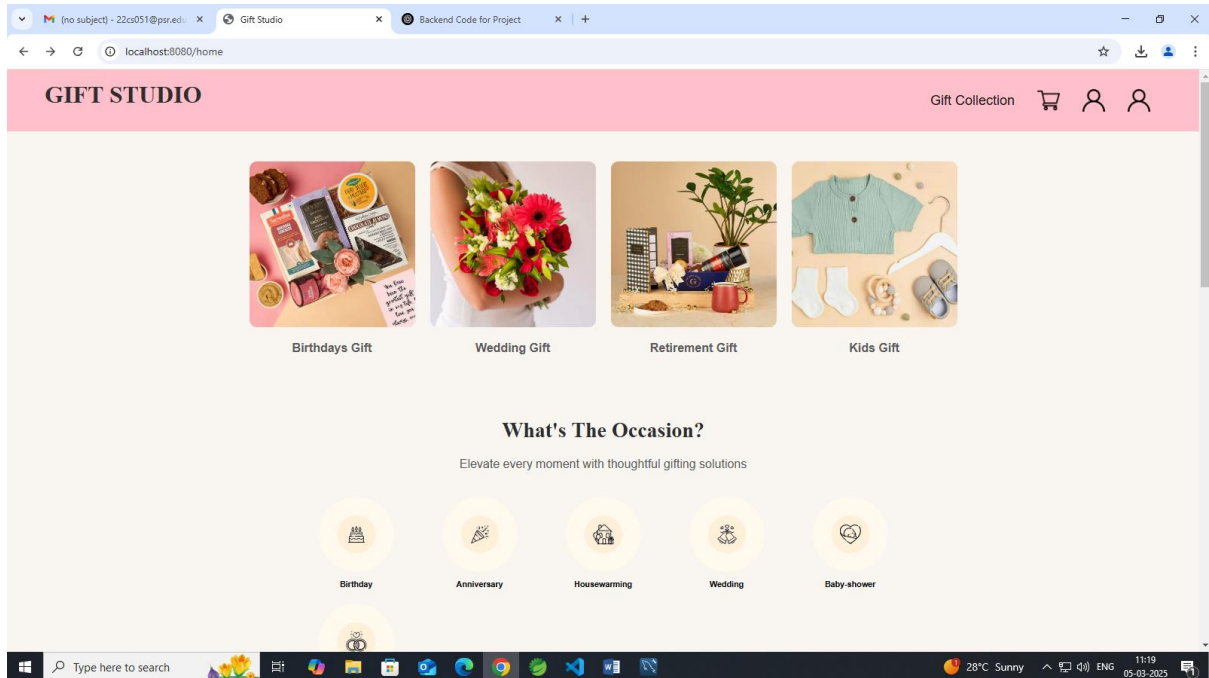


Figure 2 : Login Page

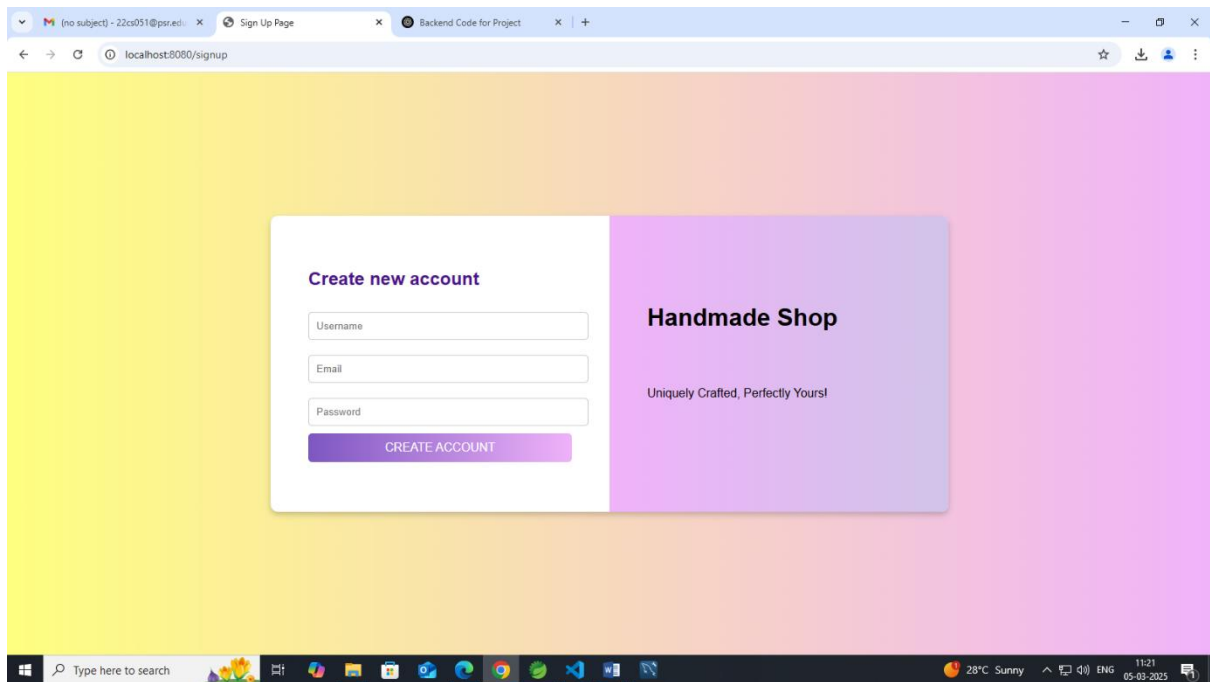


Figure 3 : Signup Page

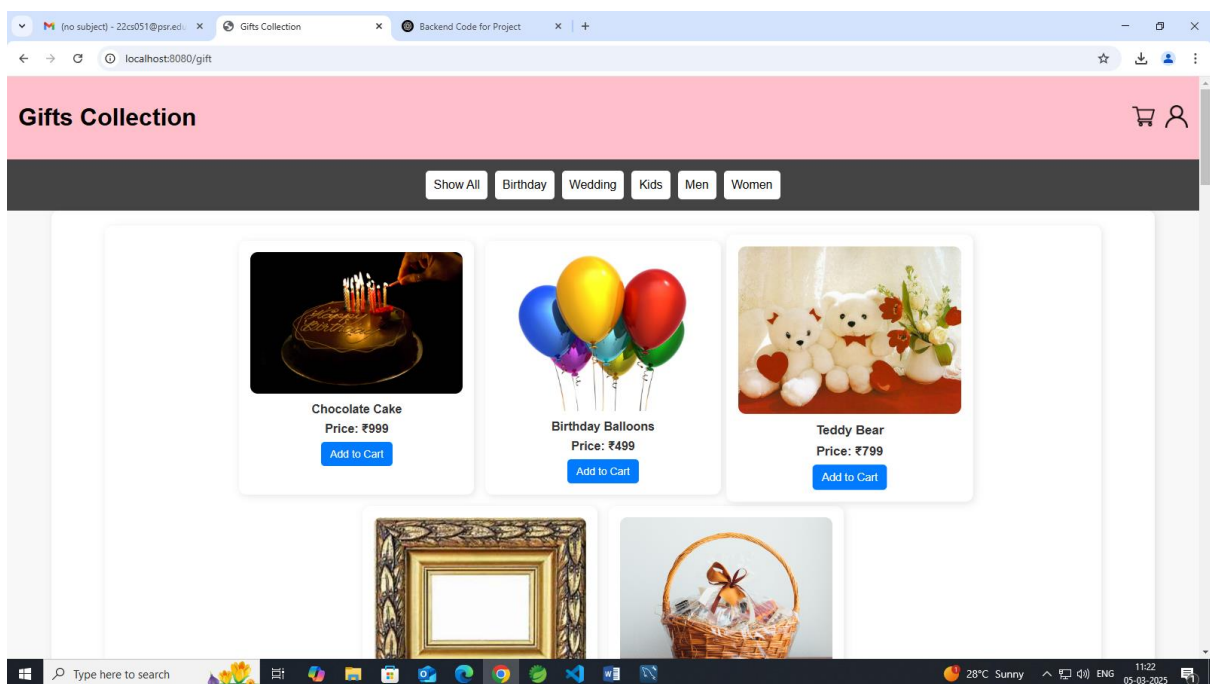


Figure 4 : Products Page

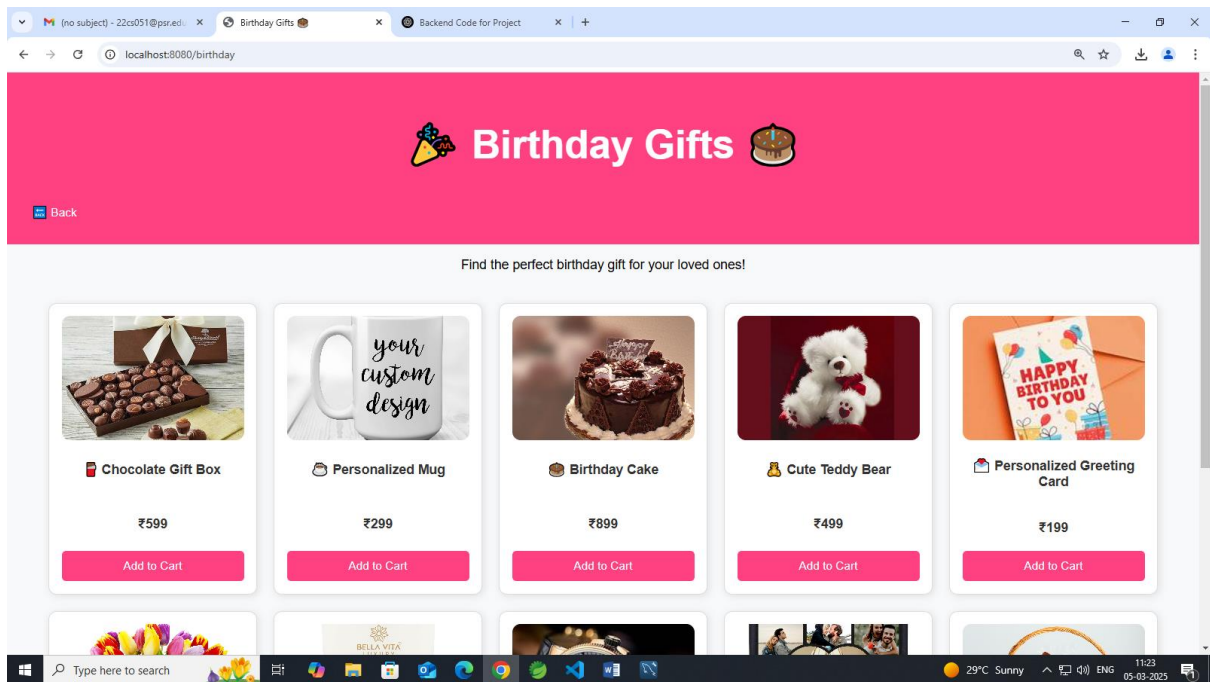


Figure 5 : Birthday Gifts Page

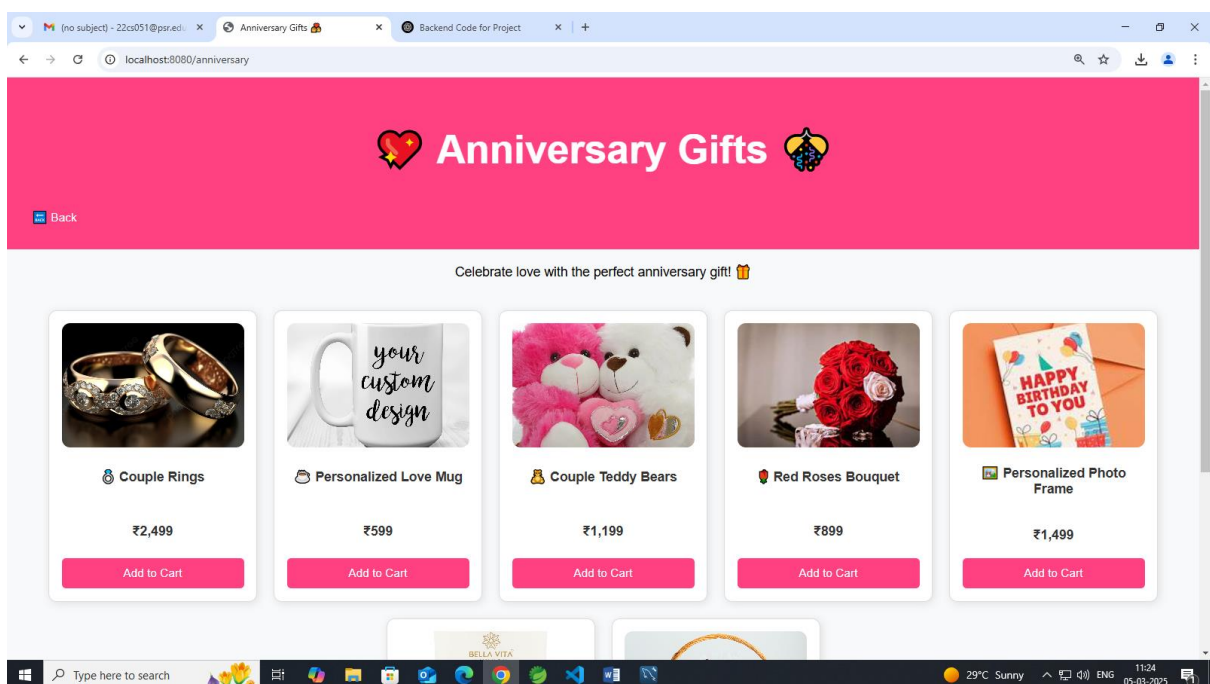


Figure 6 : Anniversary Gifts Page

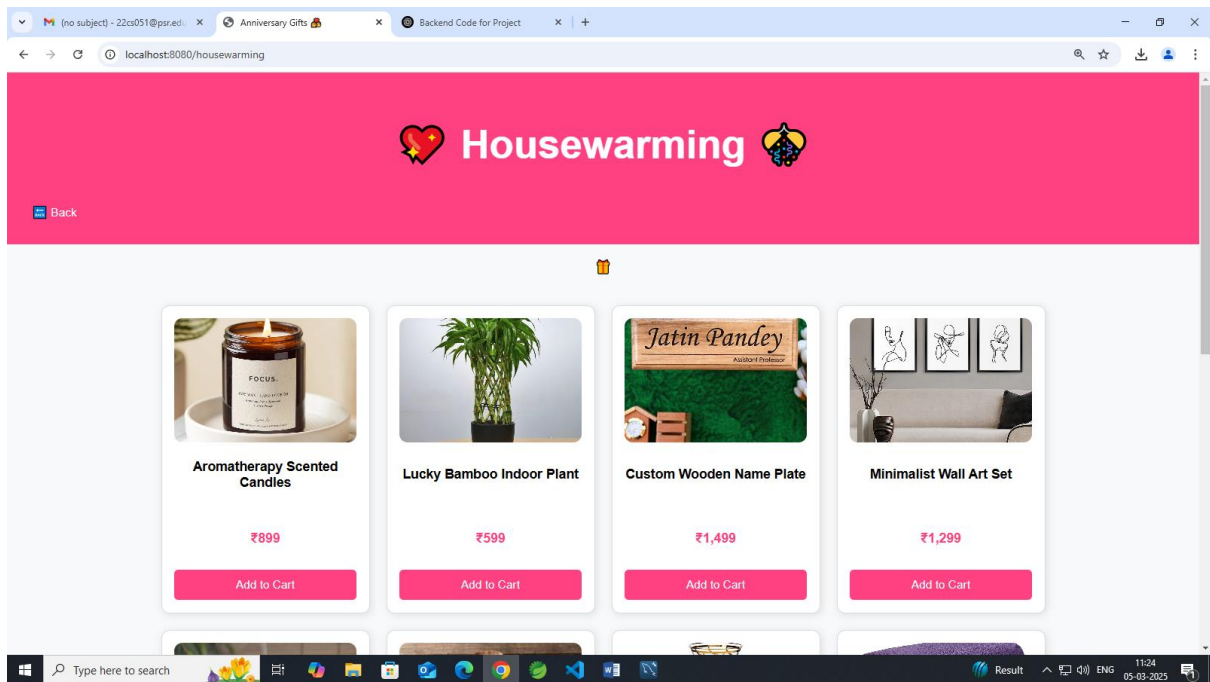


Figure 7 : Housewarming Gifts Page

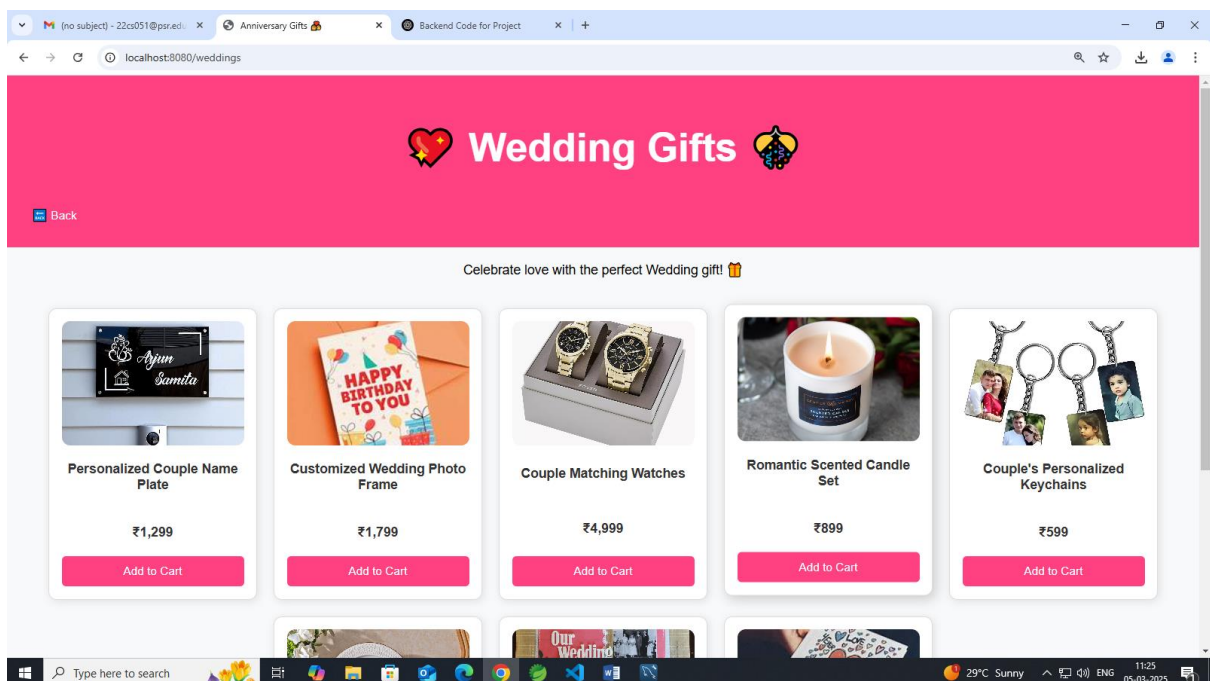


Figure 8 : Wedding Gifts page

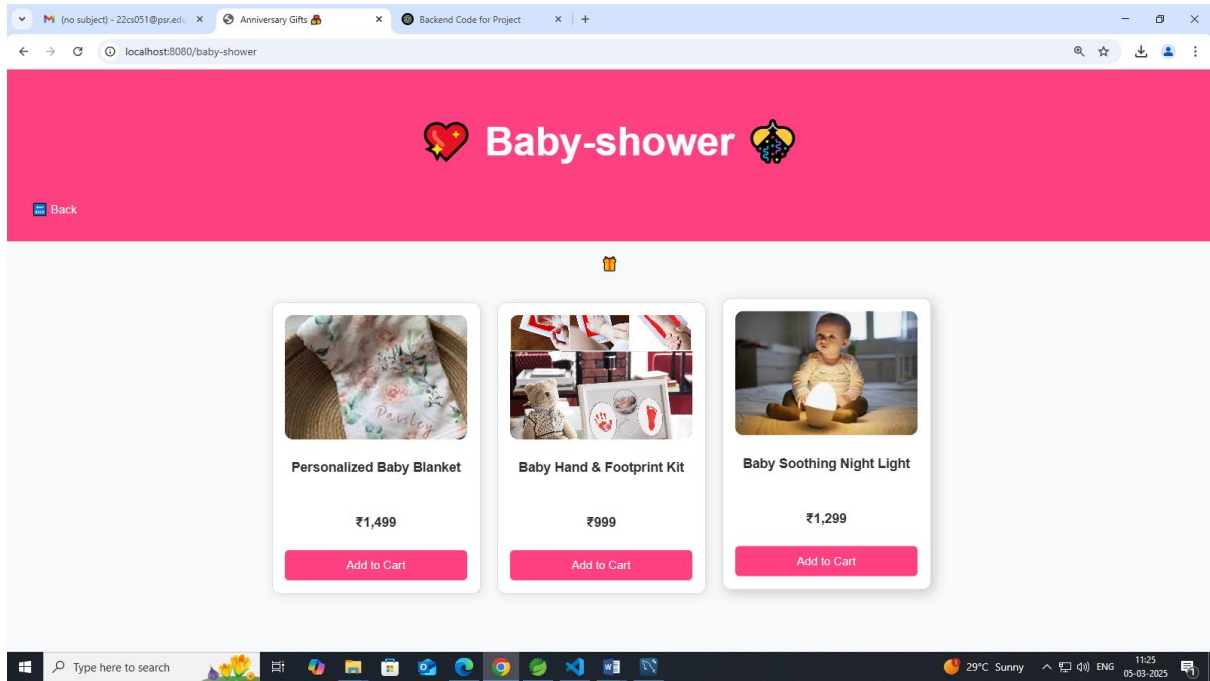


Figure 9 : Baby Shower Gifts page

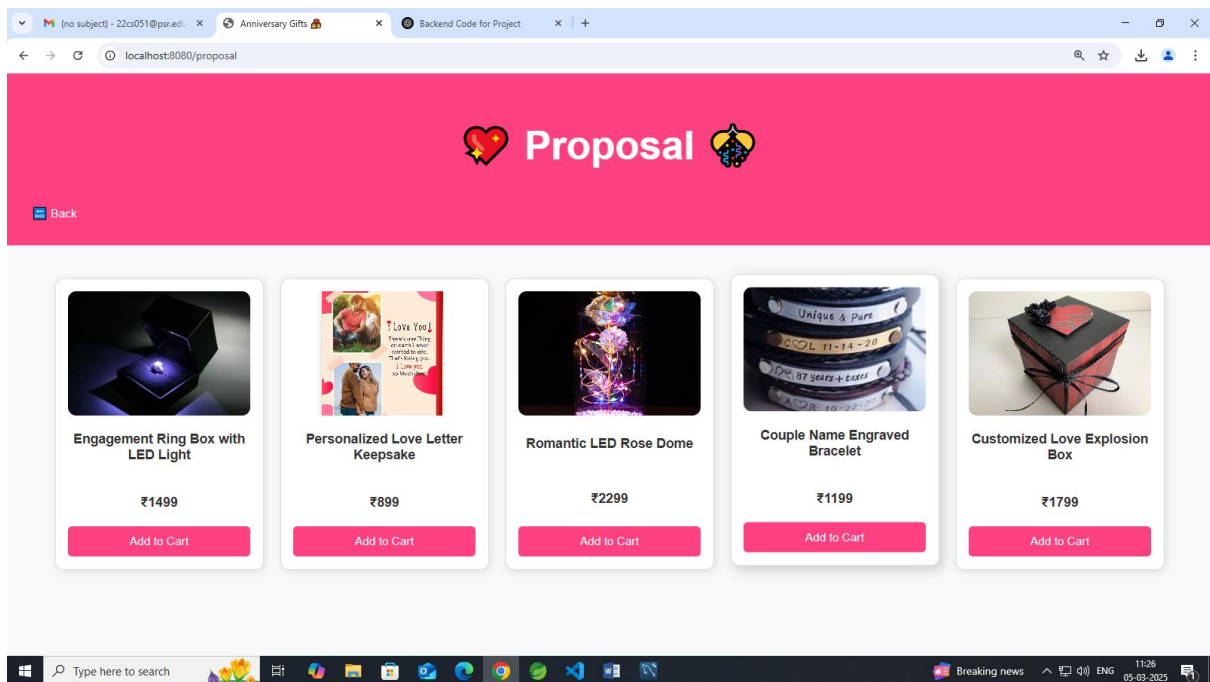


Figure 10 : Proposal Gifts Page

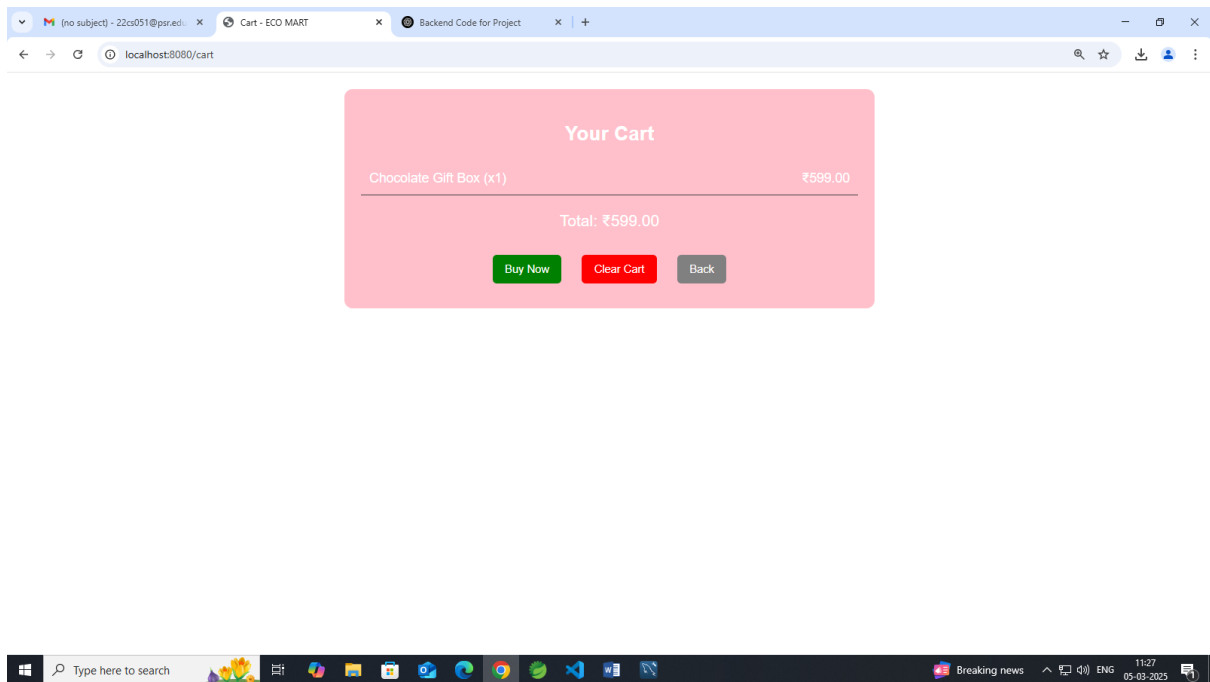


Figure 11 : Cart Page

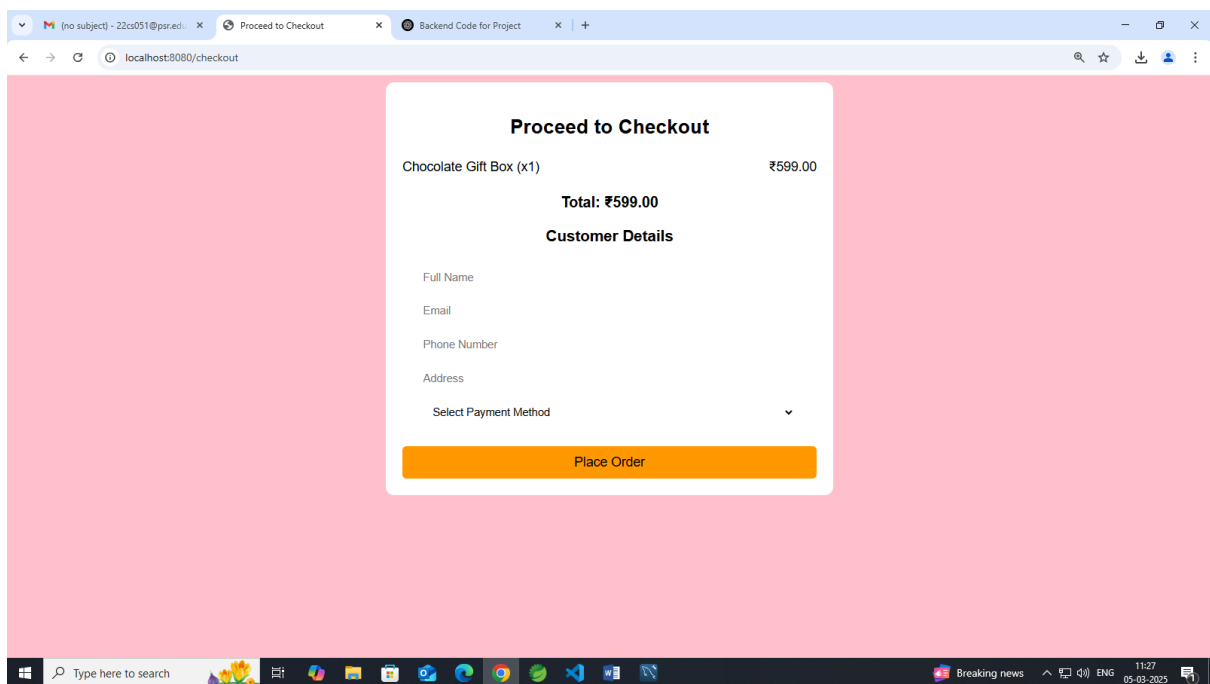


Figure 12 : Proceed page

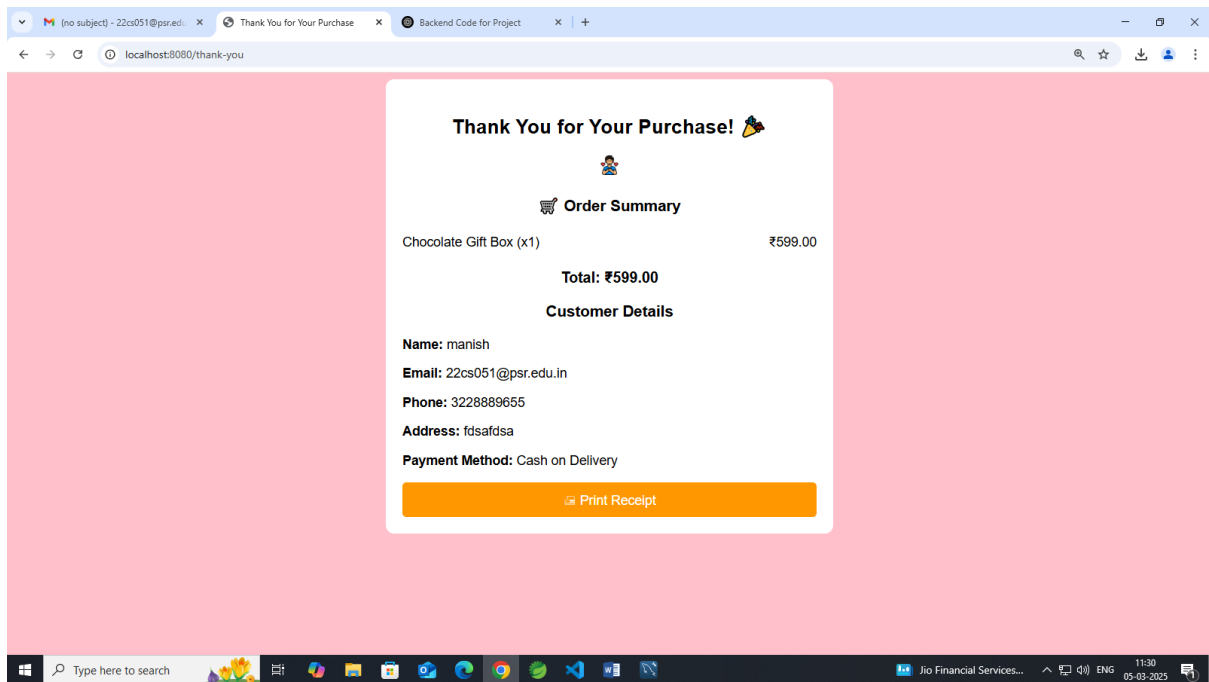


Figure 13 :Order Summary Page

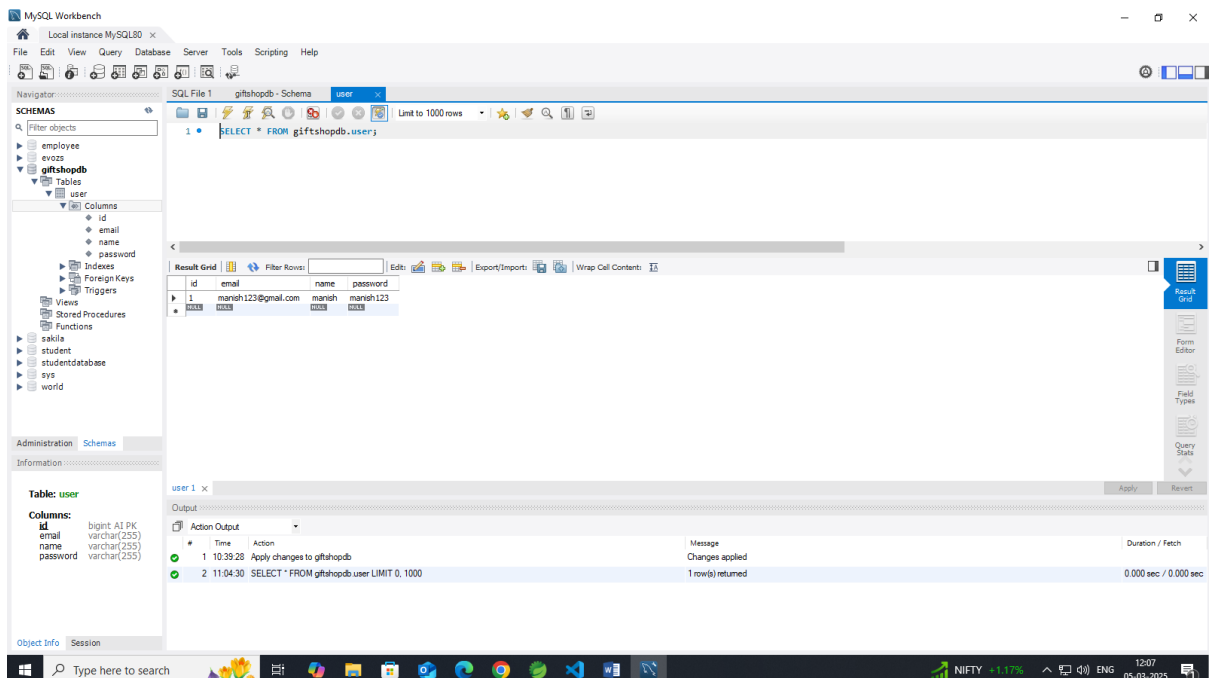


Figure14:DataBase page

SOURCE CODE:

HTML CODES:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/js/all.min.js"
crossorigin="anonymous"></script>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Gift Studio</title>
  <style>
</style>
</head>
<body>

  <div class="navbar">
    <h2>GIFT STUDIO</h2>
    <div>

<a href="/gift">Gift Collection</a>
    <a href="/cart"></a>
    <a href="/login"></a>
    <a href="/signup"></a>

    </div>
  </div>

  <section>
    <div class="gift-category-container">
      <div class="gift-category">
        
        <p>Birthdays Gift</p>
      </div>
      <div class="gift-category">
        

```



```

        <p>Wedding Gift</p>
    </div>
    <div class="gift-category">
        
        <p>Retirement Gift</p>
    </div>
    <div class="gift-category">
        
        <p>Kids Gift</p>
    </div>
</div>
</section>
<section>
    <div class="occasion-container">
        <h2>What's The Occasion?</h2>
        <p>Elevate every moment with thoughtful gifting solutions</p>

        <div class="occasion-grid">
            <div class="occasion-item">
                <a href="/birthday">
                    <div class="occasion-icon"></div>
                    </a>
                    <h6>Birthday</h6>
                </div>
            <div class="occasion-item">
                <a href="/anniversary">
                    <div class="occasion-icon"></div>
                    </a>
                    <h6>Anniversary</h6>
                </div>
            <div class="occasion-item">
                <a href="/housewarming">
                    <div class="occasion-icon"></div>
                    </a>
                    <h6>Housewarming</h6>
                </div>
        </div>
    </div>

```

```

        </a>
        <h6>Housewarming</h6>
    </div>
    <div class="occasion-item">
        <a href="/weddings">
            <div class="occasion-icon"></div>

        </a>
        <h6>Wedding</h6>
    </div>
    <div class="occasion-item">
        <a href="/baby-shower">
            <div class="occasion-icon"></div>

        </a>
        <h6>Baby-shower</h6>
    </div>
    <div class="occasion-item">
        <a href="/proposal">
            <div class="occasion-icon"></div>

        </a>
        <h6>proposal</h6>
    </div>
</div>

</div>
<!-- Wedding Gifting Section -->
<section class="wedding-section">
    <div class="wedding-image">
        
    </div>
    <div class="wedding-content">
        <h2>Wedding Gifting</h2>

```

<p>Celebrate their union with timeless elegance Celebrate their union with timeless elegance. Explore our curated collection of enchanting gifts, perfect for capturing the essence of their special day. Whether you're a guest or the happy couple, find the perfect expression of love and congratulations at The Gift Studio.</p>

</div>

</section>

<footer class="footer">

<div class="footer-container">

<div class="footer-section">

<h3>THE GIFT STUDIO</h3>

<p class="contact-info"><i class="fas fa-phone"></i> +91-91473-21916</p>

<p class="contact-info"><i class="fas fa-envelope"></i>

contact@thegiftstudio.com</p>

</div>

<div class="footer-section">

<h3>Quick Links</h3>

<p>About Us | FAQs | Blogs</p>

<p>Track Your Order | Corporate Gifting | Wedding Gifting</p>

<p>Terms & Conditions | Privacy Policy | Refund Policy | Shipping Policy</p>

</div>

<div class="footer-section">

<h3>Follow Us</h3>

<div class="social-icons">

<i class="fab fa-facebook"></i>

<i class="fab fa-instagram"></i>

<i class="fab fa-youtube"></i>

</div>

</div>

</div>

<div>

<iframe

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d62858.4874398893!2d77.51835375922755!3d9.172134792671597!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3b06c12c6e24c0b5%3A0xf78a2a725a4cfda7!2sSankarankovil%2C%20Tamil%20Nadu!5e0!3m2!1sen!2sin!4v1709218200000!5m2!1sen!2sin" allowfullscreen></iframe>

</div>

<p>© 2025, thegiftstudio.com All rights reserved.</p>

</footer>

<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous">

let slideIndex = 0;

const slides = document.querySelector(".slides");

const dots = document.querySelectorAll(".dot");

const totalSlides = document.querySelectorAll(".slide").length;

```

function showSlide(index) {
  if (index >= totalSlides) slideIndex = 0;
  if (index < 0) slideIndex = totalSlides - 1;

  slides.style.transform = `translateX(-${slideIndex * 100}%)`;

  dots.forEach(dot => dot.classList.remove("active"));
  dots[slideIndex].classList.add("active");
}

function moveSlide(n) {
  slideIndex += n;
  showSlide(slideIndex);
  restartAutoSlide();
}

function currentSlide(n) {
  slideIndex = n;
  showSlide(slideIndex);
  restartAutoSlide();
}

function autoSlide() {
  moveSlide(1);
  slideTimer = setTimeout(autoSlide, 3000);
}

function restartAutoSlide() {
  clearTimeout(slideTimer);
  slideTimer = setTimeout(autoSlide, 3000);
}

let slideTimer = setTimeout(autoSlide, 3000);
showSlide(slideIndex);

```

```
</script>
```

```
</body>
```

```
</html>
```

Login.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>
  <style>
    /* Background */
  </style>
</head>
<body>
  <div class="login-container">
    <form id="loginForm">
      <h2>Login</h2>
      <input type="email" id="email" placeholder="Enter Email" required>
      <input type="password" id="password" placeholder="Enter Password" required>
      <button type="submit">Login</button>
      <p id="error-msg"></p>
    </form>
  </div>

  <script>
    document.getElementById("loginForm").addEventListener("submit", async
function(event) {
    event.preventDefault();

    const email = document.getElementById("email").value;
    const password = document.getElementById("password").value;

    const response = await fetch("http://localhost:8080/auth/login", {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify({
        email: email,
        password: password
      })
    });

    const result = await response.text();

    if (response.ok) {
      localStorage.setItem("email", email); // Store email
      alert("Login successful!");
      window.location.href = "/home"; // Redirect to cart page
    }
  }
  </script>

```

```

    } else {
      document.getElementById("error-msg").textContent = "Invalid email or
password!";
    }
  });
</script>
</body>
</html>

```

Signup.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sign Up Page</title>
  <style>
  </style>
</head>
<body>
  <div class="container">
    <div class="left">
      <h2>Create new account</h2>
      <form id="signupForm">
        <input type="text" id="username" placeholder="Username" required>
        <input type="email" id="email" placeholder="Email" required>
        <input type="password" id="password" placeholder="Password" required>
        <button type="submit" class="btn">CREATE ACCOUNT</button>
      </form>
      <p id="message" style="color: green;"></p>
    </div>
    <div class="right">
      <h1>Handmade Shop</h1><br><br>
      <p>Uniquely Crafted, Perfectly Yours!</p><br><br>
    </div>
  </div>

  <script>
    document.getElementById('signupForm').addEventListener('submit', async
function(event) {
      event.preventDefault();

      const username = document.getElementById('username').value;

```

```

const email = document.getElementById('email').value;
const password = document.getElementById('password').value;

const response = await fetch('http://localhost:8080/auth/signup', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    name: username,
    email: email,
    password: password
  })
});

const result = await response.text();
document.getElementById('message').innerText = result;

if (response.ok) {
  alert('Signup successful!');
  window.location.href = "/login"; // Redirect to login page
} else {
  alert('Signup failed. Try again.');
```

Products.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Gifts Collection</title>
  <style>
/* Apply background to the entire page */
  </style>
</head>
<body>
  <header>
    <h1>Gifts Collection</h1>
    <div class="icons">
```

```

        <a href="/cart"></a>
        <a href="/login"></a>
    </div>
</header>
<nav>
    <button onclick="filterCategory('all')">Show All</button>
    <button onclick="filterCategory('birthday')">Birthday</button>
    <button onclick="filterCategory('wedding')">Wedding</button>
    <button onclick="filterCategory('kids')">Kids</button>
    <button onclick="filterCategory('men')">Men</button>
    <button onclick="filterCategory('women')">Women</button>
</nav>
<section class="gift-container">
    <!-- Birthday Gifts -->
<section class="gift-container birthday">
    <div class="gift-box" data-category="birthday">
        
        <p>Chocolate Cake</p>
        <p>Price: ₹999</p>
        <button onclick="addToCart('Chocolate Cake', '999')">Add to Cart</button>
    </div>
    <div class="gift-box" data-category="birthday">
        
        <p>Birthday Balloons</p>
        <p>Price: ₹499</p>
        <button onclick="addToCart('Birthday Balloons', 499)">Add to Cart</button>
    </div>
    <div class="gift-box" data-category="birthday">
        
        <p>Teddy Bear</p>
        <p>Price: ₹799</p>
        <button onclick="addToCart('Teddy Bear', 799)">Add to Cart</button>
    </div>
    <div class="gift-box" data-category="birthday">
        
        <p>Photo Frame</p>
        <p>Price: ₹699</p>
        <button onclick="addToCart('Photo Frame', 699)">Add to Cart</button>
    </div>

```



```

<div class="gift-box" data-category="birthday">
  
  <p>Gift Hamper</p>
  <p>Price: ₹1999</p>
  <button onclick="addToCart('Gift Hamper', 1999)">Add to Cart</button>
</div>
</section>

```

```

</section>
</body>
</html>

```

Cart.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cart - ECO MART</title>
  <style>
  </style>
</head>
<body>
  <div class="cart-container">
    <h2>Your Cart</h2>

    <div id="cart-items">
      <!-- Cart items will be displayed here -->
    </div>

    <div class="cart-total">Total: ₹<span id="cart-total">0.00</span></div>
    <button class="btn btn-buy" onclick="buyNow()">Buy Now</button>
    <button class="btn btn-clear" onclick="clearCart()">Clear Cart</button>
    <button class="btn btn-back" onclick="goBack()">Back</button>
  </div>

  <script>
    // Retrieve cart data from localStorage
    let cart = JSON.parse(localStorage.getItem("cart")) || [];

    function displayCart() {
      let cartItemsContainer = document.getElementById("cart-items");
      let total = 0;
      cartItemsContainer.innerHTML = "";
    }
  </script>

```

```

    cart.forEach((item, index) => {
        let price = parseFloat(item.price); // Ensure price is a number
        let quantity = parseInt(item.quantity); // Ensure quantity is an integer
        let itemTotal = price * quantity;
        total += itemTotal;

        cartItemsContainer.innerHTML += `
            <div class="cart-item">
                <span>${ item.name } (x${quantity})</span>
                <span>₹$ {itemTotal.toFixed(2)}</span>
            </div>
        `;
    });

    // Update total amount
    document.getElementById("cart-total").textContent = total.toFixed(2);

    // Debugging - check if the cart data is loaded properly
    console.log("Cart Data:", cart);
    console.log("Total Amount Calculated:", total);
}

function buyNow() {
    if (cart.length === 0) {
        alert("Your cart is empty!");
        return;
    }

    // Store the order details before clearing the cart
    localStorage.setItem("orderDetails", JSON.stringify(cart));
    localStorage.setItem("totalAmount", document.getElementById("cart-
total").textContent);

    alert("Proceeding to checkout...");
    window.location.href = "/checkout"; // Redirect to checkout page
}

function clearCart() {
    localStorage.removeItem("cart");
    cart = [];
    displayCart();
}

```

```

    function goBack() {
        window.history.back();
    }

    // Display cart items on page load
    displayCart();
</script>
</body>
</html>

```

Birthday.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Birthday Gifts 🎁🎁</title>
    <link rel="stylesheet" href="styles.css"> <!-- Link to your CSS file -->
</head>
<body>
    <style>
    </style>
    <header>
        <h1>🎁 Birthday Gifts 🍰</h1>
        <button onclick="goBack()" style="background:none; border: none; font-size: 14px;
        cursor: pointer; display: flex; align-items: center; gap: 5px;">
            🏠 BACK Back
        </button>
    </header>

    <main>
        <p>Find the perfect birthday gift for your loved ones!</p>

        <div class="gift-list">
            <div class="gift-item">
                
                <p>🍫 Chocolate Gift Box</p>
                <p class="price">₹599</p>
                <button onclick="addToCart('Chocolate Gift Box', 599)">Add to Cart</button>
            </div>

```

```

<div class="gift-item">
  
  <p>☕ Personalized Mug</p>
  <p class="price">₹299</p>
  <button onclick="addToCart('Personalized Mug', 299)">Add to Cart</button>
</div>

```

```

<div class="gift-item">
  
  <p>🍰 Birthday Cake</p>
  <p class="price">₹899</p>
  <button onclick="addToCart('Birthday Cake', 899)">Add to Cart</button>
</div>

```

```

<div class="gift-item">
  
  <p>🧸 Cute Teddy Bear</p>
  <p class="price">₹499</p>
  <button onclick="addToCart('Cute Teddy Bear', 499)">Add to Cart</button>
</div>

```

```

<div class="gift-item">
  
  <p>💌 Personalized Greeting Card</p>
  <p class="price">₹199</p>
  <button onclick="addToCart('Personalized Greeting Card', 199)">Add to
Cart</button>
</div>

```

```

<div class="gift-item">
  
  <p>🌸 Flower Bouquet</p>
  <p class="price">₹799</p>
  <button onclick="addToCart('Flower Bouquet', 799)">Add to Cart</button>
</div>

```

```

<div class="gift-item">

```

```

        
        <p>🌸 Perfume Gift Set</p>
        <p class="price">₹1,499</p>
        <button onclick="addToCart('Perfume Gift Set', 1499)">Add to Cart</button>
    </div>

    <div class="gift-item">
        
        <p>🕒 Luxury Watch</p>
        <p class="price">₹3,999</p>
        <button onclick="addToCart('Luxury Watch', 3999)">Add to Cart</button>
    </div>

    <div class="gift-item">
        
        <p>🖼️ Custom Photo Frame</p>
        <p class="price">₹899</p>
        <button onclick="addToCart('Custom Photo Frame', 899)">Add to Cart</button>
    </div>

    <div class="gift-item">
        
        <p>🎁 Gift Hamper</p>
        <p class="price">₹2,199</p>
        <button onclick="addToCart('Gift Hamper', 2199)">Add to Cart</button>
    </div>
</div>

</main>

<script>
    let cart = JSON.parse(localStorage.getItem('cart')) || [];

    function addToCart(itemName, price) {
        price = parseFloat(price); // Convert price to a number

```

```

if (!itemName || isNaN(price) || price <= 0) {
    alert("Invalid item or price!");
    return;
}

let item = cart.find(i => i.name === itemName);

if (item) {
    item.quantity += 1; // Increase quantity if item exists
} else {
    cart.push({ name: itemName, price: price, quantity: 1 });
}

localStorage.setItem('cart', JSON.stringify(cart));
alert(`${itemName} added to cart!`);
}

// Function to display cart items
function displayCart() {
    let cartContainer = document.getElementById('cart-items');
    let cartTotal = document.getElementById('cart-total');

    cartContainer.innerHTML = "";
    let total = 0;

    if (cart.length === 0) {
        cartContainer.innerHTML = "<p>Your cart is empty</p>";
        cartTotal.innerText = "Total: ₹0";
        return;
    }

    cart.forEach(item => {
        item.price = parseFloat(item.price);
        item.quantity = parseInt(item.quantity);

        let itemTotal = item.price * item.quantity;
        total += itemTotal;

        let div = document.createElement('div');
        div.classList.add('cart-item');
        div.innerHTML = `
            <span>${item.name} (x${item.quantity})</span>
            <span>₹${itemTotal.toFixed(2)}</span>
            <button onclick="removeFromCart('${item.name}')">Remove</button>
        `
    })
}

```

```

    `;
    cartContainer.appendChild(div);
  });

  cartTotal.innerText = "Total: ₹" + total.toFixed(2);
}

// Function to remove item from cart
function removeFromCart(itemName) {
  cart = cart.filter(item => item.name !== itemName);
  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

// Check if on cart page
if (document.getElementById('cart-items')) {
  displayCart();
}

function goBack() {
  window.history.back();
}

```

```

</script>
</body>
</html>

```

Anniversary.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Anniversary Gifts 🎁🎁</title>
  <link rel="stylesheet" href="styles.css"> <!-- Link to your CSS file -->
</head>
<body>
  <style>
body

  </style>
  <header>
    <h1>💖 Anniversary Gifts 🎁</h1>
    <button onclick="goBack()" style="background:none; border: none; font-size: 14px;
cursor: pointer; display: flex; align-items: center; gap: 5px;">

```

 Back

</button>

</header>

<main>

<p>Celebrate love with the perfect anniversary gift! 💎💎</p>

<div class="gift-list">

<div class="gift-item">

<p>💍 Couple Rings</p>

<p class="price">₹2,499</p>

<button onclick="addToCart('Couple Ring', 2499)">Add to Cart</button>

</div>

<div class="gift-item">

<p>☕ Personalized Love Mug</p>

<p class="price">₹599</p>

<button onclick="addToCart('Personalized Love Mug', 599)">Add to Cart</button>

</div>

<div class="gift-item">

<p>🧸 Couple Teddy Bears</p>

<p class="price">₹1,199</p>

<button onclick="addToCart('Couple Teddy Bears', 1199)">Add to Cart</button>

</div>

<div class="gift-item">

<p>🌹 Red Roses Bouquet</p>

<p class="price">₹899</p>

<button onclick="addToCart('Red Roses Bouquet', 899)">Add to Cart</button>

</div>


```

    <div class="gift-item">
      
      <p>🖼️ Personalized Photo Frame</p>
      <p class="price">₹1,499</p>
      <button onclick="addToCart('Personalized Photo Frame',1499)">Add to
Cart</button>
    </div>

```

```

    <div class="gift-item">
      
      <p>💐 Romantic Perfume Set</p>
      <p class="price">₹2,299</p>
      <button onclick="addToCart('Romantic Perfume Set ',2299)">Add to
Cart</button>
    </div>

```

```

    <div class="gift-item">
      
      <p>🛒 Gift Hamper for Couples</p>
      <p class="price">₹3,499</p>
      <button onclick="addToCart('Gift Hamper for Couples',3499)">Add to
Cart</button>
    </div>
  </div>
</main>

```

```

</div>
</main>

```

```

<script>
  let cart = JSON.parse(localStorage.getItem('cart')) || [];

```

```

function addToCart(itemName, price) {
  price = parseFloat(price); // Convert price to a number

  if (!itemName || isNaN(price) || price <= 0) {
    alert("Invalid item or price!");
    return;
  }
}

```

```

let item = cart.find(i => i.name === itemName);

if (item) {
    item.quantity += 1; // Increase quantity if item exists
} else {
    cart.push({ name: itemName, price: price, quantity: 1 });
}

localStorage.setItem('cart', JSON.stringify(cart));
alert(`${itemName} added to cart!`);
}

// Function to display cart items
function displayCart() {
    let cartContainer = document.getElementById('cart-items');
    let cartTotal = document.getElementById('cart-total');

    cartContainer.innerHTML = "";
    let total = 0;

    if (cart.length === 0) {
        cartContainer.innerHTML = "<p>Your cart is empty</p>";
        cartTotal.innerHTML = "Total: ₹0";
        return;
    }

    cart.forEach(item => {
        item.price = parseFloat(item.price);
        item.quantity = parseInt(item.quantity);

        let itemTotal = item.price * item.quantity;
        total += itemTotal;

        let div = document.createElement('div');
        div.classList.add('cart-item');
        div.innerHTML = `
            <span>${item.name} (x${item.quantity})</span>
            <span>₹${itemTotal.toFixed(2)}</span>
            <button onclick="removeFromCart('${item.name}')">Remove</button>
        `;
        cartContainer.appendChild(div);
    });

    cartTotal.innerHTML = "Total: ₹" + total.toFixed(2);

```

```

}

// Function to remove item from cart
function removeFromCart(itemName) {
  cart = cart.filter(item => item.name !== itemName);
  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

```

```

// Check if on cart page
if (document.getElementById('cart-items')) {
  displayCart();
}

```

```

function goBack() {
  window.history.back();
}

```

```

</script>
</body>
</html>

```

Housewarming.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Anniversary Gifts ❓❓</title>
  <link rel="stylesheet" href="styles.css"> <!-- Link to your CSS file -->
</head>
<body>
  <style>
body
  </style>
  <header>
    <h1>💖 Housewarming 🎁</h1>
    <button onclick="goBack()" style="background:none; border: none; font-size: 14px;
cursor: pointer; display: flex; align-items: center; gap: 5px;">
      ⬅️ Back
    </button>
  </header>

  <main>

```

```

<p>❖❖</p>
<div class="gift-list">
  <div class="gift-item">
    
    <p>Aromatherapy Scented Candles</p>
    <p class="price">₹899</p>
    <button onclick="addToCart('Aromatherapy Scented Candles', 899)">Add to
Cart</button>
  </div>

  <div class="gift-item">
    
    <p>Lucky Bamboo Indoor Plant</p>
    <p class="price">₹599</p>
    <button onclick="addToCart('Lucky Bamboo Indoor Plant', 599)">Add to
Cart</button>
  </div>

  <div class="gift-item">
    
    <p>Custom Wooden Name Plate</p>
    <p class="price">₹1,499</p>
    <button onclick="addToCart('Custom Wooden Name Plate', 1499)">Add to
Cart</button>
  </div>

  <div class="gift-item">
    
    <p>Minimalist Wall Art Set</p>
    <p class="price">₹1,299</p>
    <button onclick="addToCart('Minimalist Wall Art Set', 1299)">Add to
Cart</button>
  </div>

  <div class="gift-item">
    

```

```

    <p>Essential Oil Diffuser</p>
    <p class="price">₹1,799</p>
    <button onclick="addToCart('Essential Oil Diffuser', 1799)">Add to Cart</button>
</div>

<div class="gift-item">
  
  <p>Wooden Kitchen Utensil Set</p>
  <p class="price">₹799</p>
  <button onclick="addToCart('Wooden Kitchen Utensil Set', 799)">Add to
Cart</button>
</div>

<div class="gift-item">
  
  <p>Elegant Wine Glass Set</p>
  <p class="price">₹2,199</p>
  <button onclick="addToCart('Elegant Wine Glass Set', 2199)">Add to
Cart</button>
</div>

<div class="gift-item">
  
  <p>Cozy Fleece Throw Blanket</p>
  <p class="price">₹1,299</p>
  <button onclick="addToCart('Cozy Fleece Throw Blanket', 1299)">Add to
Cart</button>
</div>

<div class="gift-item">
  
  <p>Premium Tea Sampler Gift Box</p>
  <p class="price">₹1,099</p>
  <button onclick="addToCart('Premium Tea Sampler Gift Box', 1099)">Add to
Cart</button>
</div>
</div>

```

```

</main>
<script>
let cart = JSON.parse(localStorage.getItem('cart')) || [];

function addToCart(itemName, price) {
    price = parseFloat(price); // Convert price to a number

    if (!itemName || isNaN(price) || price <= 0) {
        alert("Invalid item or price!");
        return;
    }

    let item = cart.find(i => i.name === itemName);

    if (item) {
        item.quantity += 1; // Increase quantity if item exists
    } else {
        cart.push({ name: itemName, price: price, quantity: 1 });
    }

    localStorage.setItem('cart', JSON.stringify(cart));
    alert(`${itemName} added to cart!`);
}

// Function to display cart items
function displayCart() {
    let cartContainer = document.getElementById('cart-items');
    let cartTotal = document.getElementById('cart-total');

    cartContainer.innerHTML = "";
    let total = 0;

    if (cart.length === 0) {
        cartContainer.innerHTML = "<p>Your cart is empty</p>";
        cartTotal.innerText = "Total: ₹0";
        return;
    }

    cart.forEach(item => {
        item.price = parseFloat(item.price);
        item.quantity = parseInt(item.quantity);

        let itemTotal = item.price * item.quantity;
        total += itemTotal;
    });
}

```

```

    let div = document.createElement('div');
    div.classList.add('cart-item');
    div.innerHTML = `
      <span>${ item.name } (x${ item.quantity })</span>
      <span>₹${ itemTotal.toFixed(2) }</span>
      <button onclick="removeFromCart('${ item.name }')">Remove</button>
    `;
    cartContainer.appendChild(div);
  });

  cartTotal.innerText = "Total: ₹" + total.toFixed(2);
}

// Function to remove item from cart
function removeFromCart(itemName) {
  cart = cart.filter(item => item.name !== itemName);
  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

// Check if on cart page
if (document.getElementById('cart-items')) {
  displayCart();
}

function goBack() {
  window.history.back();
}
</script>
</body>
</html>

```

CHAPTER 6

RESULT ANALYSIS

6.1 Overview

The Online Gift Shopping Website was designed and developed to provide an efficient, user-friendly, and secure platform for purchasing gifts. The system integrates HTML, CSS, JavaScript, Java (Spring Boot), and MySQL to create a seamless e-commerce experience. This chapter presents a detailed analysis of the results obtained during the testing phase, including functional accuracy, performance evaluation, user experience, and system limitations.

6.2 Functional Accuracy

The system was tested for different functionalities, ensuring that all modules work as expected.

Home Page

1. Displays a well-structured navigation bar, banners, and featured products.
2. Users can easily access different categories of gifts.
3. Dynamic content loading improves usability.

Product Page

1. Displays gadgets categorized by brand and specifications.
2. Users can view detailed product descriptions, including name, price.
3. "Add to Cart" and "Buy Now" buttons function properly.

Cart Page

1. Items added from the product page are successfully stored in the cart.
2. Users can increase, decrease, or remove items from the cart.
3. The total price is calculated dynamically.

Login & Signup Page

1. New users can register, and their details are securely stored in the MySQL database.
2. Returning users can log in with valid credentials.
3. Unauthorized users cannot access restricted pages like the cart and checkout.

Checkout & Order Processing

1. The checkout process redirects users to a payment page.
2. User information is securely stored and managed.
3. Orders are processed without errors.

6.3 Performance Analysis

The system was tested for speed, responsiveness, and database performance.

Page Load Time

- Home Page: ~1.5 seconds
- Product Page: ~2 seconds
- Cart Page: ~1.8 seconds
- Login/Signup: ~1.2 seconds

Database Performance

- Query Execution Time (Fetching Products): <0.5 seconds
- Query Execution Time (User Authentication): <1 second
- Query Execution Time (Order Placement): ~1 second

6.4 System Limitations

- 1.No Payment Integration: Currently, the system does not support actual payments.
- 2.No Order Tracking Feature: Users cannot track their order status post-purchase.
- 3.No Admin Panel: Product and order management is not available for administrators.

6.5 Future Scope & Enhancements

The system performed well, but further improvements can enhance functionality and user experience:

- ◆ Payment Gateway Integration: Add UPI, credit/debit card, options.
- ◆ Admin Panel: Manage products, orders, and users.
- ◆ Order Tracking System: Users can track their deliveries.
- ◆ Product Reviews & Ratings: Allow users to leave feedback.

CHAPTER 7

CONCLUSION

The Online Gift Shopping Website successfully achieves its goal of providing a seamless and efficient e-commerce platform for purchasing gifts. The system integrates various technologies, including HTML, CSS, JavaScript, Java (Spring Boot), and MySQL, to create an interactive and user-friendly shopping experience. This project ensures that users can browse gifts, view product details, add items to their cart, and securely complete purchases while maintaining data security and access control.

Key Achievements of the Project:

1. User-Friendly Interface:
 - The website is designed with an intuitive UI, making navigation simple for users.
 - It features a well-structured Home Page, Product Page, Cart Page, and Authentication System.
2. Efficient Product Management:
 - The Product Page dynamically retrieves gadget details from a MySQL database.
 - Users can filter and sort products based on categories, brands, and specifications.
3. Secure User Authentication:
 - Login and Signup pages use Spring Security and MySQL to ensure secure user data storage and authentication.
 - Unauthorized users cannot access cart and checkout functionalities.
4. Smooth Shopping Experience:
 - The cart system allows users to add, remove, and update selected products.
 - The total cost is automatically calculated, ensuring a hassle-free checkout process.
5. Database Integration:
 - All user details, product information, and transactions are stored in a structured MySQL database, ensuring efficient data retrieval and management.

Challenges Faced and Solutions Implemented:

- Issue: Managing cart persistence for logged-in users.
 - Solution: Integrated local storage for guest users and MySQL database storage for authenticated users.

- Issue: Unauthorized access to restricted pages.
 - Solution: Implemented access control mechanisms, ensuring that users must log in before accessing certain features.
- Issue: Efficiently retrieving and displaying large amounts of product data.
 - Solution: Optimized database queries and used Spring Boot APIs for fast data retrieval.

Future Enhancements:

This project serves as a strong foundation for a fully functional e-commerce website. However, several enhancements can be implemented in future versions, such as:

- Payment Gateway Integration: Adding online payment methods like UPI, credit/debit cards, and digital wallets.
- Order Tracking System: Allowing users to track their orders in real time.
- User Reviews & Ratings: Enabling customers to rate and review products for better decision-making.
- Admin Panel: Providing an admin interface for product management, order tracking, and user management.
- AI-Based Recommendation System: Suggesting products based on user preferences and browsing history.

Final Thoughts:

The Online Gift Shopping Website successfully demonstrates the integration of e-commerce functionalities with secure authentication, database management, and a user-friendly interface. By leveraging Spring Boot, MySQL, and JavaScript, the system ensures efficient product handling, smooth navigation, and seamless shopping experiences. With further improvements and additional features, this project has the potential to evolve into a fully scalable and competitive e-commerce platform.

CHAPTER 8

REFERENCES:

1. "Spring Boot in Action" – Craig Walls

- Covers Spring Boot fundamentals, REST API development, and database integration.

2. "Pro MySQL" – Michael Kruckenberg & Jay Pipes

- In-depth guide on MySQL database management, queries, and optimization.

3. [Spring Boot Official Documentation](#)

- Complete guide to building backend services for your Gift Shop.