A

MAJOR PROJECT

ON

E-SHOP

(A responsive E-commerce website)

Submitted

By

SAURABH KUMAR (10622560)

SHUBHAM KUMAR (10622564)

(DIPLOMA IN COMPUTER SCIENCE AND ENGINEERING)

6th SEM (Sec - B)

(2022 – 2025)

Under the Guidance of

Mr. Jatin Verma

PROFESSOR Department of computer science

Delhi Skill and Entrepreneurship University

UGC Recognized

Department of Computer Science

Faculty of Technology

Guru Nanak Dev College

Rohini, Delhi – 110089

Guru Nanak Dev College (DSEU)

Rohini, Delhi – 110089

Certificate

This is to certify that the project work entitled “E-SHOP” has been successfully carried out by “Saurabh kumar (10622560) and Shubham kumar (10622564)” for the subject Major Project during the academic year 2022-2025, semester-VI for partial fulfillment of Diploma in Computer Engineering. The work carried out during the semester is satisfactory.

Mr. Jatin Verma
Guide, Computer department

Acknowledgement

I express my heartfelt gratitude to everyone who contributed to the successful completion of this project, **"E-SHOP."**

First and foremost, I would like to thank my institution and the department for providing me with the opportunity to work on this project and for offering the necessary guidance and resources.

I am deeply indebted to my mentor/guide, **Mr. Jatin Verma**, whose continuous support, expertise, and valuable feedback were pivotal in shaping this project. Your guidance throughout every phase of this work has been invaluable.

I would also like to thank my friends and peers for their encouragement and support. Their constructive criticism and feedback helped me enhance the quality of this project.

This project has been a valuable learning experience, and I am truly thankful to everyone who helped me along the way.

Name: Saurabh Kumar (10622560)

Shubham Kumar (10622564)

Declaration

I hereby declare that the project titled "E-shop" submitted as part of my major project work is a result of my independent effort and research. The work presented here is original and has not been submitted in any form to any other institution or organization for any purpose.

I have developed this project entirely on my own, using resources and tools available in the public domain. Any references or external materials used in this project have been properly acknowledged.

I take full responsibility for the content of this project and ensure that it meets the ethical and academic standards required for its submission.

Saurabh kumar(10622560)

Shubham Kumar (10622564)

Abstract:

The **E-Shop** project is a responsive and user-friendly e-commerce website developed using HTML, CSS, JavaScript, and the Bootstrap framework. It enables users to browse a variety of products, view detailed information, manage a shopping cart, and complete a checkout process with order confirmation.

Bootstrap plays a key role in ensuring a consistent and mobile-friendly layout across all pages, enhancing the overall user experience. The site includes essential features such as product listings, search functionality, and informational pages like About and Contact. This front-end prototype lays the groundwork for a complete online shopping platform and offers scope for future expansion through backend integration, secure payment gateways, and user account systems.

Table of Contents:

Introduction

The **E-Shop** project is a front-end e-commerce website developed to replicate the core functions of an online shopping platform. It allows users to explore various products, view detailed information, add items to a cart, and proceed to checkout. Built using HTML, CSS, JavaScript, and Bootstrap, the site offers a clean interface and responsive design that works seamlessly across different screen sizes.

Bootstrap plays a key role in ensuring visual consistency and mobile compatibility throughout the website. Pages like Home, Products, Cart, Checkout, Contact, and Order Confirmation are designed to enhance user experience and mimic real-world e-commerce flows. While the current version focuses on front-end functionality, it lays the groundwork for integrating backend services, user accounts, and secure payment systems in future development phases.

Objectives

The **E-SHOP** project aims to achieve the following key objectives:

1. **Develop a responsive e-commerce website**: Using HTML, CSS, JavaScript, and Bootstrap to create a fully functional online shopping platform. The website will adapt to various screen sizes, ensuring accessibility across desktops, tablets, and mobile devices.

2. **Provide seamless user experience**: Allow users to browse through a wide range of products, view detailed information, add items to the cart, and complete purchases with an intuitive checkout process. The aim is to make the shopping experience smooth and efficient.

3. **Implement core e-commerce functionalities**: Simulate essential e-commerce features such as product listings, detailed product pages, cart management, and order confirmation. This will help users interact with the site as they would on a real online shopping platform.

4. **Create a user-friendly, adaptable layout**: Ensure the website's layout is consistent and easy to navigate, whether accessed on a small smartphone screen or a larger desktop monitor. The use of Bootstrap will ensure responsiveness across different devices.

5. **Enhance front-end development skills**: Provide an opportunity to apply web development techniques and tools in a real-world context, focusing on user

interface design, interactivity, and functionality.

6. **Establish a scalable prototype**: Develop a basic, scalable prototype that can later be extended with backend functionalities, such as user authentication, real-time inventory management, and secure payment gateways.
7. **Lay the groundwork for future enhancements**: This project serves as a foundational model for a full-fledged e-commerce platform. The front-end can be integrated with backend services, enabling more complex features like personalized user experiences and real-time product tracking.

Project Plan and Methodology

1. Planning (1 Week)

- **Goal**: Decide what the website needs.
- **Tasks**:
  - List features (e.g., product search, cart, checkout).
  - Draw simple sketches (how pages should look).
  - Choose tools (Bootstrap for design, JavaScript for functions).

2. Design (1 Week)

- **Goal**: Make the website look good and easy to use.
- **Tasks**:
  - Create color schemes and fonts.
  - Design buttons, menus, and product cards.
  - Test on mobile and desktop.

3. Development (4 Weeks)

- **Goal**: Build the website step by step.
- **Tasks (Weekly Breakdown)**:
  - **Week 1**: Show products with filters (price/category).
  - **Week 2**: Add a shopping cart (save items).
  - **Week 3**: Make checkout page (name, address, payment).
  - **Week 4**: Add search bar (find products fast).

4. Testing (1 Week)

- **Goal**: Fix bugs and make sure everything works.
- **Tasks**:
  - Test buttons, forms, and cart.
  - Check on different browsers (Chrome, Firefox).
  - Ask friends to try it and give feedback.

5. Launch & Updates (Ongoing)

- **Goal**: Put the website online and keep improving.
- **Tasks**:
  - Upload to a free host (GitHub Pages).
  - Watch how users behave (e.g., what they buy).
  - Fix small errors and add new features later.

Methodology:

The project followed a **mix of Agile and Prototyping** approaches to build the website quickly while staying flexible for changes.

- **Agile**:
  - Worked in **small steps** (like building the cart first, then checkout).
  - Tested often and fixed issues as they came up.
- **Prototyping**:
  - Used **Bootstrap** to design fast without starting from scratch.
  - Used **localStorage** as a temporary "database" before adding real backend.

Key Steps :-

1. **Plan & Sketch**
   - Listed what the website needed (e.g., product pages, cart).
   - Drew simple layouts (no fancy designs yet).
2. **Build Step-by-Step**
   - **First**: Made product listings with filters. Then Added a cart (saved items in browser memory).
   - **Later**: Built checkout and order confirmation.
3. **Test & Fix**
   - Clicked through every button/form to catch mistakes.
   - Checked if it worked on phones and computers.
4. **Launch & Improve**
   - Put the site online (e.g., GitHub Pages).
   - Added small upgrades later (like better search).

Tools and Technologies

HTML: Builds the basic structure of the E-Shop website, organizing content like headings, images, and links for easy navigation and SEO.
It forms the backbone of every page, ensuring content is well-structured and accessible.

CSS: Styles the website—handles layout, colors, fonts, and makes the design responsive for all screen sizes.
It enhances user experience by making the site visually appealing and easy to navigate.

JavaScript: Adds interactivity like product filtering, cart updates, and real-time user input handling during checkout.
It makes the website dynamic, allowing features to work without reloading the page.

Bootstrap: Ensures a mobile-friendly, responsive design using pre-built components like grids, buttons, and navbars.
It speeds up development and maintains a consistent look across all pages.

VS Code: The main code editor used, with features like syntax highlighting, IntelliSense, debugging, and Git integration.
It improves productivity by offering extensions and tools tailored for web development.

Git & GitHub: Git manages version control; GitHub stores the code online for easy collaboration and progress tracking.
They help keep your project organized and make teamwork smoother.

Features

1. **Product Browsing**: The E-Shop website offers a user-friendly product browsing experience, where users can explore a wide range of products categorized by type, price, or popularity. Each product listing includes essential information such as product name, price, description, and image. Users can filter and sort products based on various criteria, making it easy to find specific items.

2. **Product Detail Pages**: Every product has a dedicated product detail page that provides in-depth information, including larger images, detailed descriptions, specifications, and customer reviews. This helps users make informed decisions before adding products to their shopping cart. The product page also includes related items or recommendations to encourage further exploration.

3. **Shopping Cart**: The website features an interactive shopping cart that allows users to add, remove, or modify the quantity of items. The cart dynamically updates as users interact with it, showing a summary of the selected products, their prices, and the total cost. This feature ensures a smooth shopping experience and helps users keep track of their purchases.

4. **Checkout Process**: The checkout process is designed to be straightforward and user-friendly, guiding users through a series of simple steps to complete their purchase. Users can review their cart, enter shipping and payment details, and confirm their order. The process minimizes friction and ensures users can easily finalize their purchase without confusion.

5. **Order Confirmation**: After completing the checkout process, users are presented with an order confirmation page that displays the details of their purchase, including the order number, items purchased, total cost, and estimated delivery date. This page provides reassurance to users and confirms that their transaction was successfully processed.

6. **Search Functionality**: The E-Shop website includes a search bar that allows users to quickly find products by entering keywords or product names. The search results are displayed in real time, showing matching products, and allowing users to narrow down their choices with additional filters. This feature enhances navigation and makes it easier to find specific products.

7. **Responsive Design**: The website is fully responsive, meaning it automatically adjusts its layout and design to fit different screen sizes and devices. Whether users are browsing on a desktop, tablet, or smartphone, they will have a consistent and optimized experience. This is made possible through the use of Bootstrap's responsive grid system and mobile-first design principles.

8. **About and Contact Pages**: The E-Shop website includes informational pages such as "About Us" and "Contact Us." The "About Us" page provides background information on the business, its mission, and values, while the "Contact Us" page includes a contact form and other ways for customers to reach out for support, inquiries, or feedback. These pages help build trust and engagement with users.

Who is the User?

The user of this e-commerce website is primarily an **online customer** who wants to purchase computer-related products such as laptops, gadgets, accessories, and smart home devices. This platform is designed for a wide range of users, including:

1. **Students** – who are looking for affordable yet powerful devices for studies, online classes, or coding assignments.
2. **Working Professionals** – who need reliable computers, peripherals, and accessories for work-from-home setups or office use.
3. **Gamers and Tech Enthusiasts** – who regularly buy high-performance accessories like gaming keyboards, headphones, or graphic-heavy laptops.
4. **General Home Users** – who want to upgrade their homes with smart devices such as smart TVs, tablets, or speakers.
5. **Occasional Buyers** – who may not be tech experts but want a smooth and trustworthy online shopping experience for tech products.

These users expect:

- A **clean and responsive interface** for browsing products.
- The ability to **search and filter** products by name, category, or price.
- A **secure and fast checkout process**.
- **Reliable product information** and **images**.
- **Customer support** and **order confirmation** for trust and transparency.

The site is built keeping **user convenience** in mind — allowing them to browse, add items to the cart, and place an order without needing technical knowledge.

Who is the Admin?

- The admin is someone who logs in with special credentials (like admin@site.com) and gets access to a **dashboard** or **admin panel** to manage everything.

Roles of an Admin :-

1. **Product Management**
    - Add new products with images, prices, and details.
    - Edit or update product info.
    - Remove outdated or sold-out products.
2. **Order Management**
    - View all customer orders.
    - Change order status: *Pending* → *Shipped* → *Delivered*.
    - Cancel fraudulent or duplicate orders.
3. **Customer Support**
    - Respond to contact form messages or customer complaints.
    - View user reviews/feedback.
4. **Inventory Tracking**
    - Check stock levels.
    - Restock or mark products as out-of-stock.
5. **User Management (optional)**
    - View registered users.
    - Ban or block fake accounts.
6. **Analytics (optional future feature)**
    - Track sales, most viewed products, and profits.
    - Generate reports.

Steps to Buy a Product

1. **Search or Browse Products**
   - Go to the homepage or use the **search bar** (search.html) to find the product you want.
   - Apply filters like **category** or **price range** to narrow down the options.
2. **View Product Details**
   - Click on any product card to open the **product details page** (product-view.html).
   - Review product info like **name, description, price, specs**, and **images**.
3. **Add to Cart**
   - Click the **"Add to Cart"** button.
   - The product will be saved in your **shopping cart** (usually in localStorage).
4. **Go to Cart**
   - Navigate to the **Cart page** (cart.html) from the navbar or a cart icon.
   - Check product quantity, remove items, or continue shopping.
5. **Proceed to Checkout**
   - Once satisfied with your cart, click on **"Proceed to Checkout"**.
6. **Enter Shipping Details**
   - On the **checkout page** (checkout.html), fill in your **name, address, contact info**, etc.
7. **Choose Payment Option**
   - Select a **payment method** (Cash on Delivery, Card, UPI, etc. — based on your project design).
8. **Place the Order**
   - Click on the **"Place Order"** or **"Confirm Order"** button.
   - Show an order confirmation message (can redirect to a success.html if implemented).
9. **Order Confirmation**
   - You may display an order ID, estimated delivery time, and contact support info.

Steps to Add to Cart

1. **Search or Browse Products**
    - Go to the **homepage** or use the **search bar** on search.html.
    - Browse through products or use filters (category, price) to find what you want.
2. **Open Product Details**
    - Click on a product card to go to the **product-view.html** page.
    - This page shows product info like name, image, description, price, etc.
3. **Click on "Add to Cart"**
    - On the product-view page, click the **"Add to Cart"** button.
4. **JavaScript Action Triggers**
    - When the button is clicked:
        - Product data (like ID, name, price, quantity) is saved in the browser's **localStorage**.
        - A confirmation message (e.g. "Added to cart!") might pop up.
5. **Cart Updates Automatically**
    - The **cart icon** (if present in navbar) can show updated item count using JavaScript.
    - You can now visit cart.html to view, modify, or delete products in the cart.

Steps in the Payment Procedure

1. **Go to Cart Page**
    - Visit **cart.html**.
    - Review products you've added: name, price, quantity, and total.
    - Click on the **"Proceed to Checkout"** button.
2. **Checkout Page (checkout.html)**
    - You'll be redirected to the **checkout form** where you need to enter:
        - Name
        - Email
        - Address
        - City
        - State
        - Zip code
        - Payment method (Card, UPI, COD, etc.)
3. **Form Validation**
    - JavaScript checks if all fields are filled.
    - If something's missing, it shows an alert and blocks submission.
4. **Submit the Form**
    - Once all inputs are valid, clicking the **"Place Order"** button triggers JavaScript:
        - It clears the cart from **localStorage**.
        - Shows an alert: **"Order placed successfully!"**
        - Redirects you to the homepage or a thank-you page (if added).

System Architecture

The architecture of my e-commerce website follows a **three-layered client-side model**:

1. Presentation Layer (Front-End)

- **Technologies Used:** HTML, CSS, Bootstrap, JavaScript
- **Function:** This is the layer the user interacts with directly. It handles:
  - Display of products
  - Navigation between pages (Home, Cart, Checkout, Contact)
  - Form inputs (like the contact form or checkout form)
  - Visual responsiveness for all devices (mobile, tablet, desktop)

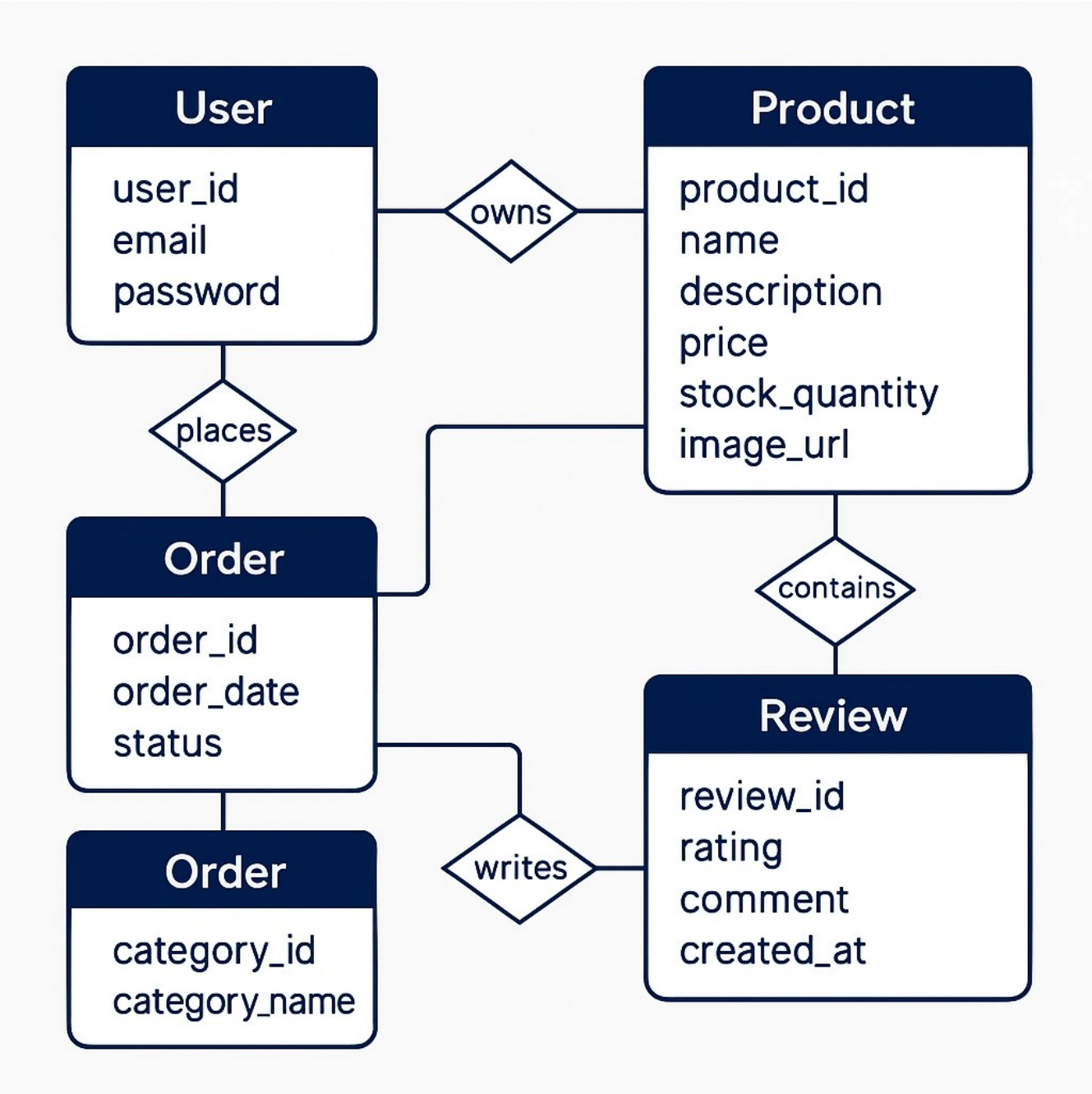
2. Application Logic Layer (Client-Side Scripting)

- **Technology Used:** JavaScript
- **Function:** This layer handles:
  - Dynamic product display and interactivity
  - Shopping cart functionality (adding/removing items)
  - Cart calculations (total price, item count)
  - Form validation (e.g., checking if required fields are filled)
  - Data passing between pages using localStorage or session storage

3. Data Layer (Temporary Client Storage)

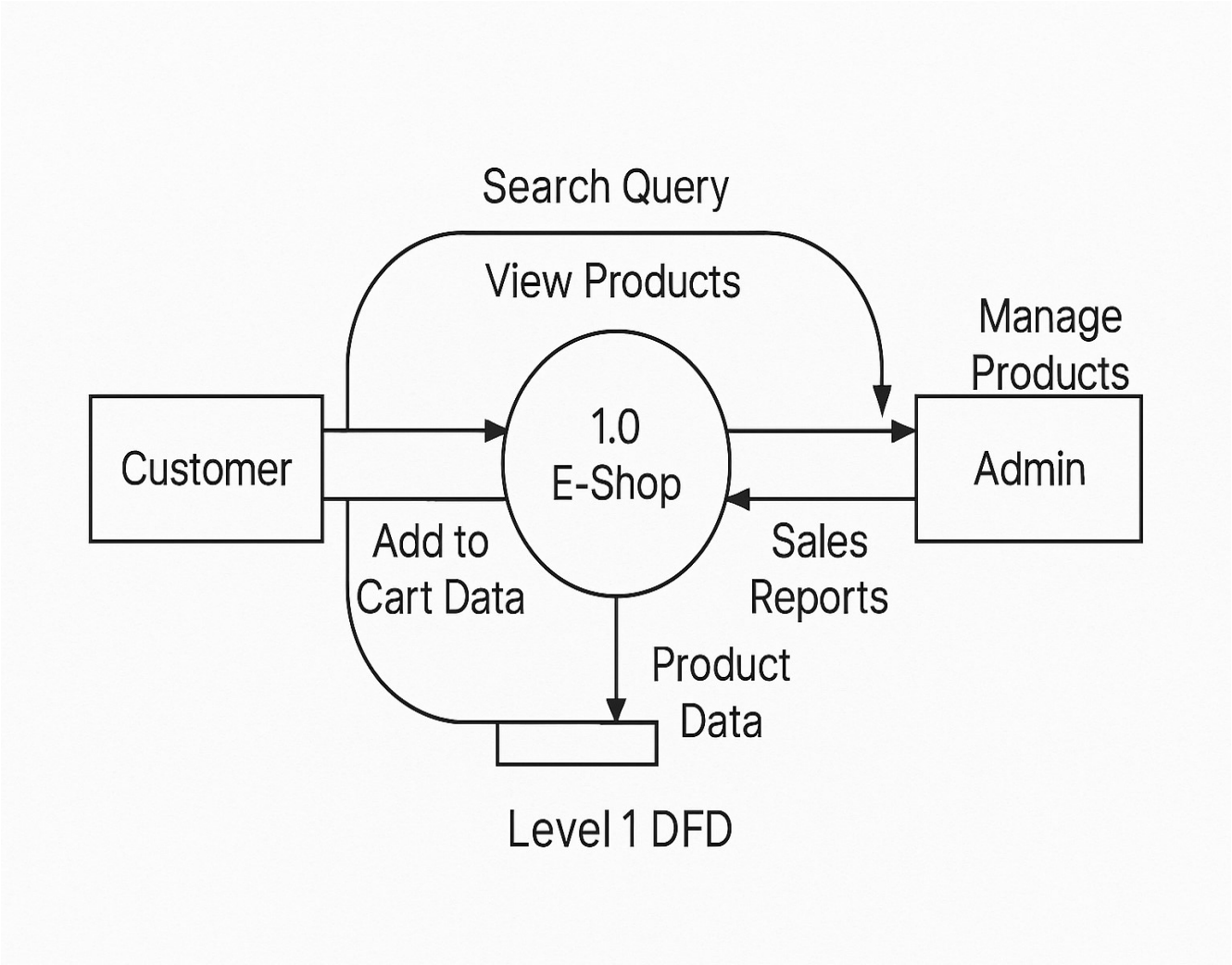
- **Technology Used:** localStorage or sessionStorage
- **Function:**
  - Stores cart data during the session
  - Helps pass data between cart, checkout, and confirmation pages
  - No back-end/database is involved; all data is stored temporarily in the browser

ER diagram

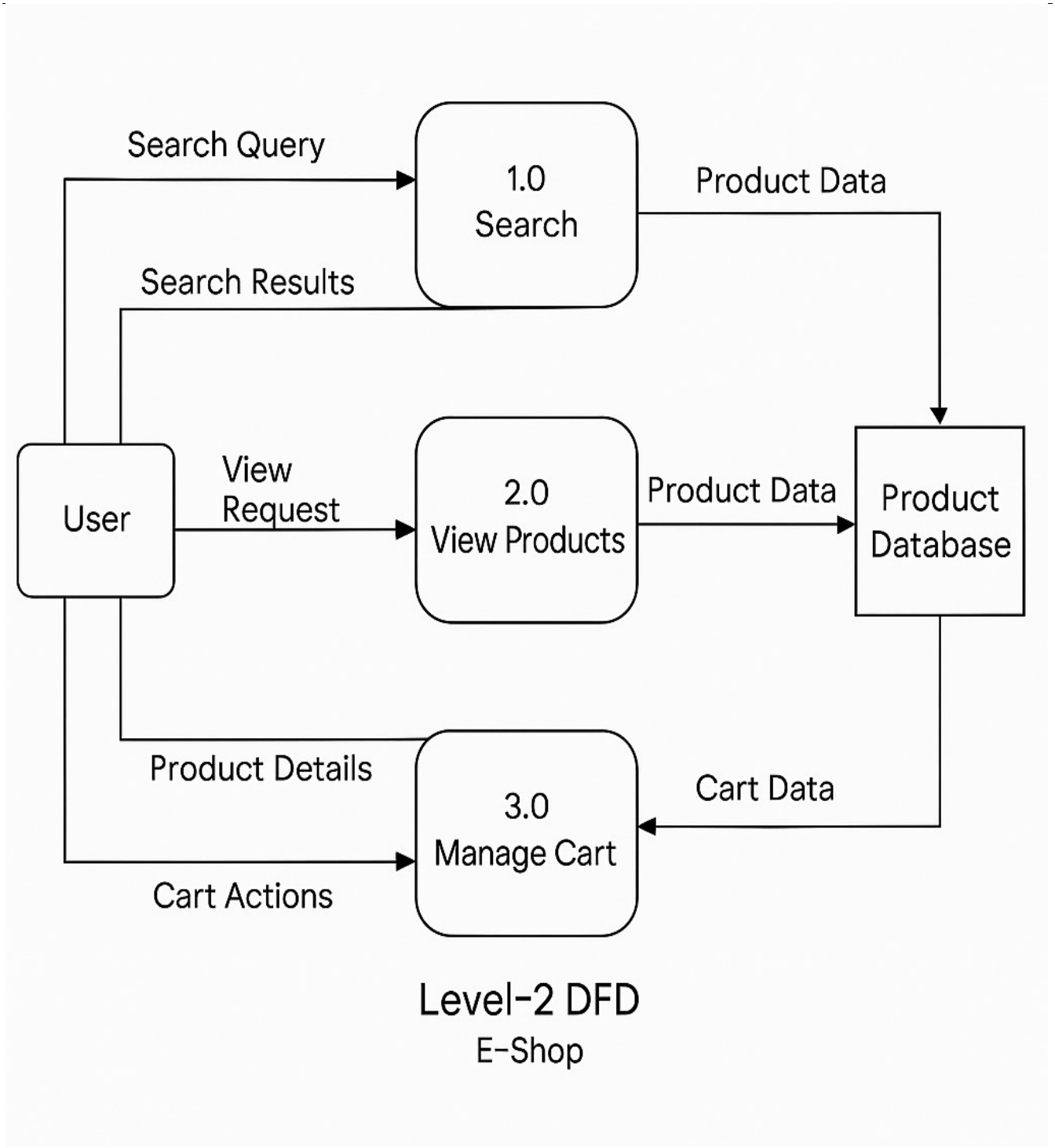

User
user_id
email
password
owns
Product
product_id
name
description
price
stock_quantity
image_url
places
Order
order_id
order_date
status
contains
Review
review_id
rating
comment
created_at
writes
Order
category_id
category_name

Data flow diagram

Level 1 DFD



Level 2 DFD


Search Query
1.0
Search
Product Data
Search Results
User
View
Request
2.0
View Products
Product Data
Product
Database
Product Details
Cart Data
3.0
Manage Cart
Cart Actions
Level-2 DFD
E-Shop

Working Flow of the Application

1. **Homepage**: Users land on the homepage, where they can view featured products and navigate to other pages like Products, Cart, and Contact.
2. **Product Browsing**: On the Products page, users browse and filter products, or search for specific items.
3. **Product Details**: Clicking on a product opens its detailed page, where users can view more info and add it to their cart.
4. **Adding to Cart**: Users add products to the cart, which updates the cart icon and the total price dynamically.
5. **Cart Review**: Users review the cart, modify quantities, or remove items before proceeding to checkout.
6. **Checkout**: Users enter their shipping and payment details on the checkout page.
7. **Order Confirmation**: After placing the order, users are shown a confirmation page with order details and a unique order ID.
8. **Post-Purchase**: Users can browse more products or exit the site, with future enhancements planned for user accounts and backend integration.

Conclusion

The E-Shop project successfully demonstrates the fundamental components of an e-commerce website, utilizing **HTML**, **CSS**, **JavaScript**, and **Bootstrap** to create a responsive and user-friendly interface. The platform allows users to browse products, manage their cart, and complete the checkout process efficiently, offering a seamless shopping experience. Rigorous testing ensured the website functions smoothly across various devices and browsers.

This project serves as a foundational prototype for future developments, such as backend integration, secure payment systems, and user account management. While the current version focuses on front-end features, it paves the way for a fully functional e-commerce platform with the potential for scalability and future enhancements.

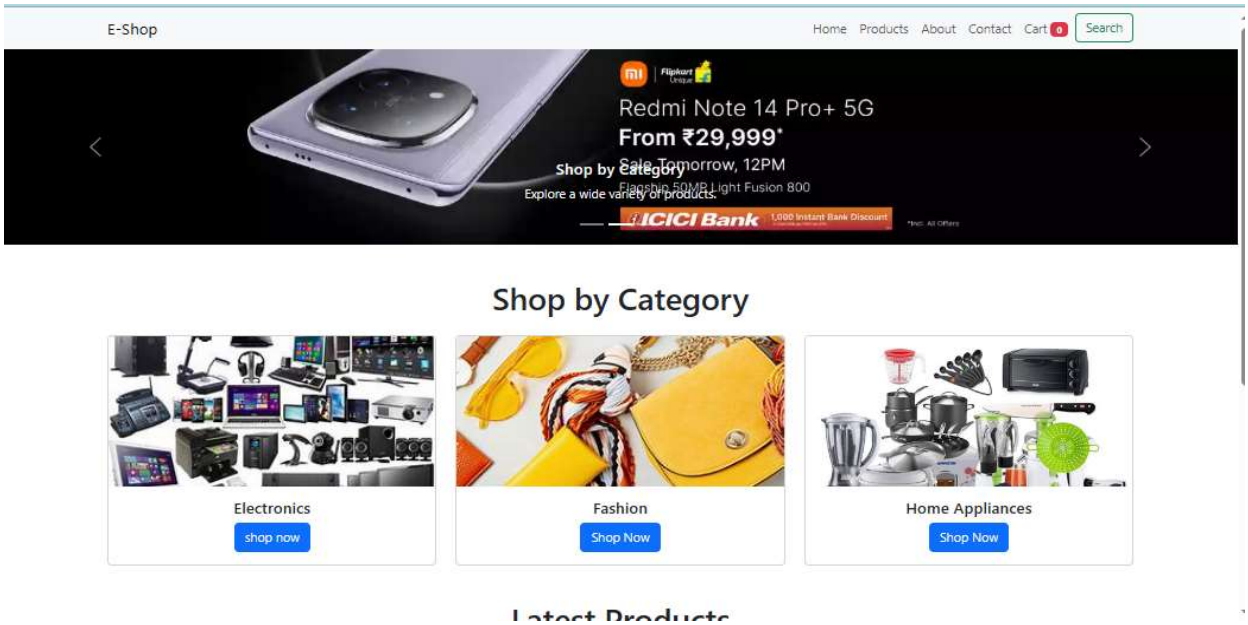
Future Scope

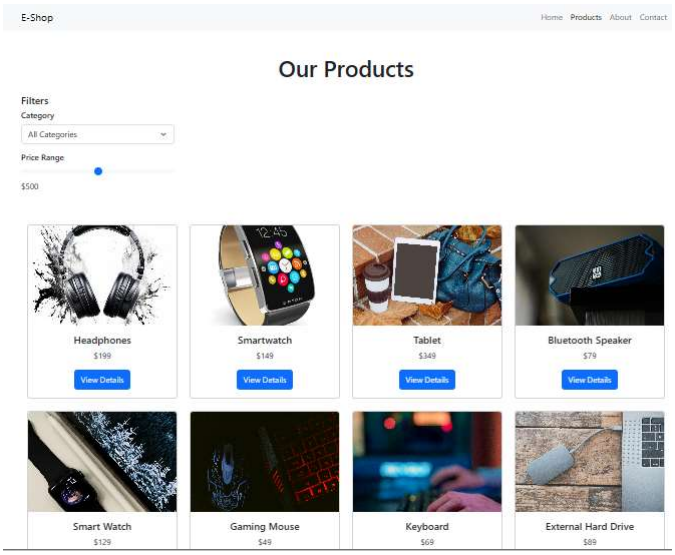
The E-Shop project serves as a solid foundation for developing a fully functional and feature-rich e-commerce platform. In the future, several enhancements can be implemented to expand its capabilities and improve the user experience:

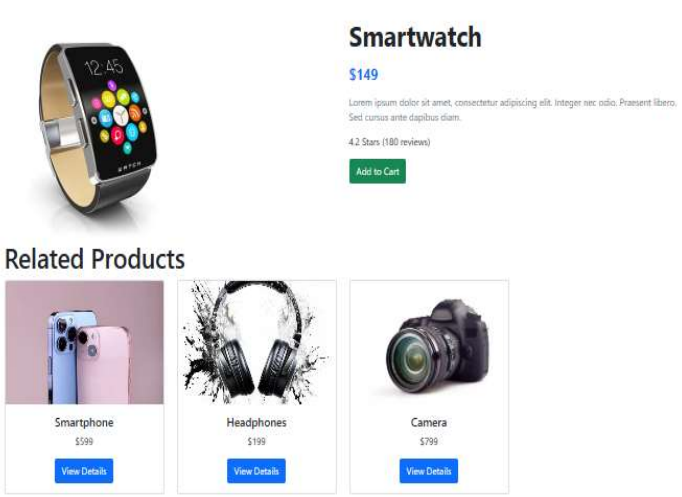
1. **Backend Integration**: Implementing a backend system using technologies like **Node.js**, **PHP**, or **Python (Django/Flask)** will allow for dynamic content, user authentication, and real-time updates for product inventory and user accounts.
2. **Payment Gateway Integration**: Adding secure payment gateways (e.g., **Stripe**, **PayPal**, **Razorpay**) will enable users to make transactions, providing a complete online shopping experience.
3. **User Accounts and Authentication**: Developing a user account system with features like order history, wishlists, and personalized recommendations will improve customer engagement and retention.
4. **Real-Time Inventory Management**: Integrating inventory management systems to track stock levels and automate restocking notifications can streamline operations for both customers and store administrators.
5. **Advanced Search and Filtering**: Enhancing the product search and filtering options using machine learning algorithms will allow users to find products more efficiently based on preferences and past browsing behavior.
6. **Mobile App Development**: Developing a mobile application for iOS and Android would extend the platform's reach and provide users with a dedicated, optimized shopping experience.
   Performance Optimization: Further optimizing page load speeds, image compression, and server-side caching will ensure faster browsing and a smoother experience, especially during high traffic periods.

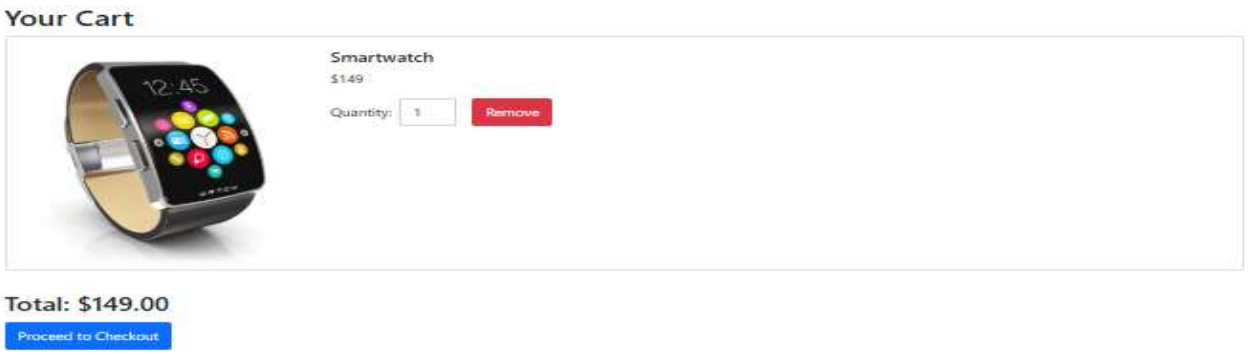
With these enhancements, the E-Shop project has the potential to evolve into a fully-fledged, scalable e-commerce solution, ready to handle larger traffic volumes, offer personalized experiences, and integrate with various external systems.

Images of Website









Coding

→ index.html

```
 <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>E-commerce Website</title>
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="css/styles.css">
<!-- Font Awesome CDN -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css" rel="stylesheet">


</head>
<body>
  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container">
      <a class="navbar-brand" href="#">E-Shop</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link" href="index.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="products.html">Products</a>
          </li>

          <li class="nav-item">
```

```
            <a class="nav-link" href="about.html">About</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="contact.html">Contact</a>
          </li>
          <li class="nav-item">

              <a class="nav-link" href="cart.html">
                Cart
                <span id="cart-count" class="badge bg-danger text-
white">0</span>
              </a>


          </li>

          <li class="nav-item">
            <a href="search.html" class="btn btn-outline-success">Search</a>
          </li>


        </ul>
      </div>
    </div>
  </nav>

  <!-- Carousel -->
  <div id="homeCarousel" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-indicators">
      <button type="button" data-bs-target="#homeCarousel" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
      <button type="button" data-bs-target="#homeCarousel" data-bs-slide-to="1"
aria-label="Slide 2"></button>
      <button type="button" data-bs-target="#homeCarousel" data-bs-slide-to="2"
aria-label="Slide 3"></button>
    </div>
    <div class="carousel-inner">
      <div class="carousel-item active">
        <img src="images/slider-1.webp" class="d-block w-100" alt="...">
        <div class="carousel-caption d-none d-md-block">
          <h5>Welcome to E-Shop</h5>
          <p>Find the best deals here!</p>
```

```
        </div>
      </div>
      <div class="carousel-item">
        <img src="images/slider-2.webp" class="d-block w-100" alt="...">
        <div class="carousel-caption d-none d-md-block">
          <h5>Shop by Category</h5>
          <p>Explore a wide variety of products.</p>
        </div>
      </div>
      <div class="carousel-item">
        <img src="images/slider-3.webp" class="d-block w-100" alt="...">
        <div class="carousel-caption d-none d-md-block">
          <h5>Latest Products</h5>
          <p>Check out the newest arrivals!</p>
        </div>
      </div>
    </div>
    <button class="carousel-control-prev" type="button" data-bs-
target="#homeCarousel" data-bs-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-
target="#homeCarousel" data-bs-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Next</span>
    </button>
  </div>
```

→ styles.css

```
#scrollToTopBtn {
```

```
  position: fixed;
  bottom: 20px;
  right: 20px;
  z-index: 1000;
}
#scrollToTopBtn:hover {
  background-color: #495057; /* Slightly lighter on hover */
}
/* Custom button styles */
#scrollToTopBtn
  {
  background-color: hsla(350, 100%, 57%, 0.875); /* Example color */
  border: 2px solid black;
}

/* Ensure links are clickable */
a {
  pointer-events: auto !important;
}


  /* Make images responsive */
img {
    max-width: 100%;
    height: auto;
  }

  /* Adjust font sizes for small screens */
  h1, h2, h3, h4 {
    font-size: calc(1.5rem + 1vw);
  }
/* Add subtle hover effect for products */
.card:hover {
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    transform: translateY(-5px);
    transition: transform 0.3s ease-in-out, box-shadow 0.3s ease-in-out;
  }
```

-> **Script.js**

```
let debounceTimer;
document.getElementById('searchQuery').addEventListener('input', function() {
  clearTimeout(debounceTimer);
```

```
  debounceTimer = setTimeout(function() {
    performSearch(); // Call the search function after a delay
  }, 300); // 300ms delay
});

// Get the category filter and price range slider
const categoryFilter = document.getElementById('categoryFilter');
const priceRange = document.getElementById('priceRange');
const priceValue = document.getElementById('priceValue');

// Update the priceValue span when the range slider changes
priceRange.addEventListener('input', function () {
  priceValue.textContent = priceRange.value;
  filterProducts(); // Call filter function on price range change
});

// Get all product cards
const productCards = document.querySelectorAll('.card');
// Update products based on category and price filter
categoryFilter.addEventListener('change', function () {
  filterProducts(); // Call filter function when category is changed
});
// Function to filter products
function filterProducts() {
  const selectedCategory = categoryFilter.value;
  const selectedPrice = parseInt(priceRange.value);

  productCards.forEach(card => {
    const cardCategory = card.getAttribute('data-category');
    const cardPrice = parseInt(card.getAttribute('data-price'));

    // Check if the product matches the category and price criteria
    if ((selectedCategory === 'All Categories' || selectedCategory ===
cardCategory) &&
        cardPrice <= selectedPrice) {
      card.style.display = 'block'; // Show card
    } else {
      card.style.display = 'none'; // Hide card
    }
  });
}
// Set initial price value on page load
```

```
priceValue.textContent = priceRange.value;

// Initial call to filter products when page loads
filterProducts();
```

→ cart.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Your Cart</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="index.html">E-Com</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link" href="index.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="products.html">Products</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" href="cart.html">Cart</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

```
  <!-- Cart Section -->
  <div class="container mt-5">
    <h2>Your Cart</h2>
    <div id="cart-items" class="row">
      <!-- Cart items will be dynamically inserted here -->
    </div>
    <div id="cart-summary" class="mt-3">
      <h3>Total: $<span id="total-price">0.00</span></h3>
      <button id="checkout-btn" class="btn btn-primary">Proceed to
Checkout</button>
    </div>
  </div>
```

-> checkout.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Checkout - E-Shop</title>

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

    <div class="container-fluid">

      <a class="navbar-brand" href="index.html">E-Shop</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
```

```
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link" href="index.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="products.html">Products</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="cart.html">Cart</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" href="checkout.html">Checkout</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <!-- Checkout Section -->
  <div class="container mt-5">
    <h1 class="text-center mb-4">Checkout</h1>
```

➔ **contact.html**

```
<!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
```

```
    <div class="container">

      <a class="navbar-brand" href="#">E-Shop</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarNav">

        <ul class="navbar-nav ms-auto">

          <li class="nav-item">

            <a class="nav-link" href="index.html">Home</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="products.html">Products</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="cart.html">Cart</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="about.html">About</a>

          </li>

          <li class="nav-item">

            <a class="nav-link active" href="contact.html">Contact</a>

          </li>

        </ul>

      </div>

    </div>
```

```
  </nav>

  <!-- Contact Us Section -->

  <div class="container mt-5">

    <h1 class="text-center mb-4">Contact Us</h1>

    <!-- Contact Info -->

    <div class="row">

      <div class="col-md-6">

        <h4>Our Contact Information</h4>

        <p><strong>Address:</strong> 123 E-Shop Street, New Delhi, India</p>

        <p><strong>Phone:</strong> +915053622585</p>

        <p><strong>Email:</strong> sksksk8053@gmail.com</p>

      </div>
```

➔ **scroll.html**

```
<!-- Scroll to Top Button -->

  <button id="scrollToTopBtn" class="btn btn-primary position-fixed bottom-3
end-3 d-none" title="Back to Top">

    &#8679;

  </button>

  <!-- JavaScript -->

  <script defer>

    const scrollToTopBtn = document.getElementById('scrollToTopBtn');

    const productList = document.getElementById('productList');



    // Show button when scrolling down

    window.addEventListener('scroll', () => {

      if (window.scrollY > 200) {
```

```
        scrollToTopBtn.classList.remove('d-none');
      } else {
        scrollToTopBtn.classList.add('d-none');
      }
    });
    // Smooth scroll to top
    scrollToTopBtn.addEventListener('click', () => {
      window.scrollTo({
        top: 0,
        behavior: 'smooth',
      });
    });
    document.addEventListener('DOMContentLoaded', () => {
    // Perform search when the search button is clicked
    const searchButton =
document.querySelector("button[onclick='performSearch()']");
    if (searchButton) {
      searchButton.addEventListener('click', performSearch);
    }
    function performSearch() {
      const query = document.getElementById('searchQuery').value.toLowerCase();
      const products = document.querySelectorAll('.card');  // Make sure we're
targeting the correct elements
      let resultsFound = false;
      console.log('Searching for:', query); // Log the search query
      products.forEach(product => {
        const productName = product.querySelector('.card-
title').textContent.toLowerCase();
```

```
        const productCategory = product.getAttribute('data-
category').toLowerCase();

        // Log each product's name and category being checked

        console.log('Checking product:', productName, 'Category:',
productCategory);

        // Check if the query matches the product name or category

        if (productName.includes(query) || productCategory.includes(query)) {

          product.style.display = 'block'; // Show matching product

          resultsFound = true;

          console.log('Match found:', productName);

        } else {

          product.style.display = 'none'; // Hide non-matching product

        }

      });

      // Show the correct results header

      const resultsHeader = document.getElementById('resultsHeader');

      resultsHeader.textContent = resultsFound ? 'Search Results' : 'No Results
Found';

    }

  });

</script>

  <!-- Bootstrap JS -->

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script> <script
src="js/scripts.js"></script>
```

GitHub Deployment

The E-SHOP project has been deployed on GitHub Pages to make it accessible to a broader audience. GitHub Pages is a convenient way to host static websites and projects directly from a repository. By deploying the project, users can access the application using a publicly available link without needing to download or run the code locally.

The deployed link can be shared with others for testing or demonstration purposes. Additionally, this deployment ensures the project's version control is maintained on GitHub, allowing for future updates and improvements to be reflected instantly.

Here is the deployment link: https://saurabhhkumarrr.github.io/E-SHOP/