



MIT Art, Design and Technology University MIT School of Computing, Pune

Department of Information Technology

Lab Manual

Practical - Web Programming

Class - S.Y. (SEM-II), DA

Batch - DA-II

Pranita Potghan

Ms.

A.Y. 2024 – 2025 (SEM-II)

File Index page given in the stationary

<h2 style="text-align: center;">Web Programming</h2> <p style="text-align: center;">SEMESTER – IV</p>			
Course Code:	23IT2008	Course Credits:	02
Teaching Hours / Week (L:T:P):	0:0:4	CA Marks:	25
Total Number of Teaching Hours:		END-SEM Marks:	25
Course Pre-requisites:			
<p>Course Description: This course provides a comprehensive introduction to web technology, designed to help students develop a strong foundation in building and managing websites and web applications. The curriculum covers key topics such as HTML, CSS, and JavaScript, PHP, MySQL, which are essential for creating interactive, well-designed web pages. Students will also explore the principles of responsive design, ensuring that web applications are optimized for different devices and screen sizes.</p> <p>The course dives deeper into server-side technologies, including HTTP, web servers, and databases, allowing students to understand how websites function behind the scenes. Emphasis is placed on practical learning, and students will gain hands-on experience by working on projects that showcase their ability to design, develop, and deploy websites.</p> <p>By the end of the course, students will be proficient in using modern web technologies to create web applications. They will understand how to handle client-server interactions, manage user data, and implement various web technologies to enhance the functionality of their applications.</p>			
<p>Course Learning Objectives: This course will enable the students to:</p> <ol style="list-style-type: none"> 1. Understand fundamental concepts of front-end web development. 2. Enable students to create basic web pages incorporating essential elements such as images, hyperlinks, lists, tables, and forms. 3. Teach students how to use CSS to manage fonts, lists, colors, text alignment, and background images for a cohesive and aesthetically pleasing web design. 4. Develop an understanding of JavaScript scopes to manage the visibility and lifetime of variables and functions effectively. 5. Equip students with the skills to implement and handle JavaScript events, enabling enhanced user interactions through event-driven programming. 6. Apply comprehensive knowledge of HTML, CSS, and JavaScript to develop a complete front-end application. Utilize project-based learning to showcase problem-solving skills and creativity in web development projects. 7. Configure server environments with Apache/TOMCAT. 8. Set up a PHP development environment and write basic PHP scripts. 9. Master PHP programming constructs for web development tasks. 10. Create and process HTML forms, and manage MySQL database operations. 11. Develop comprehensive back-end applications using PHP and MySQL. 			
<p>Course Outcome: After taking this course, Students will be able to :</p> <ol style="list-style-type: none"> 1. Apply knowledge of HTML to create the structure of the webpage and CSS to style and layout the elements, making the application visually appealing. 2. Apply comprehensive knowledge of HTML, CSS, and JavaScript to develop a complete front-end application and utilize project-based learning to showcase problem-solving skills and creativity in web development projects. 3. Set up and configure a server environment using tools like Apache or TOMCAT and set up a PHP development environment. Write & execute simple PHP scripts, understanding PHP syntax and basic features, create HTML forms to collect user data and integrate with PHP for processing. 4. Design and develop a back-end application using PHP and MySQL, implementing CRUD 			

operations to manage data effectively.

UNIT - I	Introduction to HTML and Cascading Style Sheet	09 Hours
Module 1 - Markup Language (HTML): Introduction to HTML, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms		
Module 2 - CSS: Need for CSS, introduction to CSS, basic syntax and structure, Levels of style sheets, Style specification formats, BOX Model, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, Background images		
Pedagogy	<p>ICT Teaching / PowerPoint Presentation and Videos: Use tools like Visual Studio Code (free). Videos: <u>https://www.coursera.org/learn/html-css-javascript-for-web-developers</u></p> <p>Self-study / Do it yourself /: Practice creating basic HTML pages and enhancing them using CSS.</p> <p>Experiential Learning Topics: Design a simple webpage for coffee shop website</p> <p>PBL - Project Based Learning: Create a multi-page website (e.g., coffee shop website) using HTML and CSS.</p>	
UNIT - II	Front-End Development	09 Hours
Module 3 - Overview of JavaScript, including JS in an HTML (Embedded, External), Basic JS syntax, basic interaction with HTML		
Module 4 - Core features of JavaScript: Data types, Control Structures, Arrays, Functions and Scopes		
Pedagogy	<p>ICT Teaching / PowerPoint Presentation and Videos: Use tools like Visual Studio Code (free). Videos: <u>https://www.coursera.org/learn/javascript-basics</u></p> <p>Self-study / Do it yourself /: Solve exercises on JavaScript syntax, control structures, and functions</p> <p>Experiential Learning Topics: Build a web page with interactive elements (e.g., a simple calculator).</p> <p>PBL - Project Based Learning: Develop an interactive webpage that uses JavaScript to validate form inputs or perform basic calculations.</p>	
UNIT - III	Advanced Front-End Development	09 Hours
Module 5 - DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM		
Module 6 - JavaScript Events: JavaScript Events, Types of JavaScript Events, Objects in JS, Event Handling		
Pedagogy	<p>ICT Teaching / PowerPoint Presentation and Videos: <u>https://www.coursera.org/learn/building-interactive-web-pages-using-</u></p>	

	<p><u>javascript</u> Use tools like Visual Studio Code (free).</p> <p>Self-study / Do it yourself /: Practice exercises on DOM traversal and event handling.</p> <p>Experiential Learning Topics: Add dynamic behavior to a webpage using DOM and events (e.g., a to-do list app).</p> <p>PBL - Project Based Learning: Develop a web page with dynamic content (e.g., a task manager or interactive quiz) using DOM manipulation and event handling.</p>
--	---

UNIT - IV	Server Side Scripting	09 Hours
<p>Module 7 - Set up and configure a server environment using tools like Apache or TOMCAT, set up a PHP development environment.</p> <p>Module 8 -Introduction to PHP: : Introduction to PHP, Server side scripting Vs Client side scripting, Basic Development Concepts (Mixing PHP with HTML), Creating, Writing & Running First PHP Script, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions,</p> <p>Module 9 - Form handling with HTML and PHP: Designing of Forms using HTML, Form Handling using GET and POST methods of Form</p>		
<p>Pedagogy</p> <p>ICT Teaching / PowerPoint Presentation and Videos: <u>https://www.coursera.org/learn/web-applications-php</u> Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management</p> <p>Self-study / Do it yourself /: Practice exercises on form handling and server-side scripting with PHP.</p> <p>Experiential Learning Topics: Create a basic form for data submission and handle it using PHP (e.g., feedback form).</p> <p>PBL - Project Based Learning: Develop a small server-side application (e.g., a contact form with email validation and submission).</p>		

UNIT - V	Working with Databases and Web Application Development	09 Hours
<p>Module 10 - Working with databases using MySQL with PHP: MySQL database, create database, create table, primary key with AUTO_INCREMENT setting, Insert Data Into a Database Table, Select Data From a Database Table, Open or close a Connection to the MySQL Server.</p> <p>Module 11 - Web Application Development (Project): Develop the web application to handle client-server interactions, manage user data, and implement various web technologies to enhance the functionality of their applications. Example: Website for a Coffee Shop</p>		
<p>Pedagogy</p> <p>ICT Teaching / PowerPoint Presentation and Videos: Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management</p> <p>Videos: <u>https://www.coursera.org/learn/web-app</u></p> <p>Self-study / Do it yourself /:</p>		

	<p>Exercises on creating and manipulating databases using PHP and MySQL.</p>
	<p>Experiential Learning Topics: Create a database and design a webpage to display its data dynamically.</p>
	<p>PBL - Project Based Learning: Develop a fully functional web application (e.g., a Coffee Shop website or e-commerce platform) that integrates database functionality for data management.</p>

Experiment No.1**Problem Statement:**

1. Create the basic structure of the Crossword Book Store website, including the home page layout with a header, main content area, and footer.

Prepare a common project website design and plan document for all assignments. Consider following points:

1. Brief information about the project.
2. Set the goals & deliverables.
3. Finalize the modules of the project.
4. Define the audience.
5. Describe pain points & the ideal experience (On the basis of existing systems)
6. Set the visual direction
7. Map out the Project structure.
8. Plan the content for each page.
9. Add ideas for content, images & layout.
10. Determine your site structure or Create content for your core website pages:
 - a. Home page
 - b. About page
 - c. Product/Service page
 - d. Testimonial/review page
 - e. Support page
 - f. Starter blog posts
11. Create and collect design elements
12. These design elements define your brand personality and help customers feel what your brand represents through the use of:
 - a. Colors
 - b. Fonts and typography
 - c. Logos
 - d. Images and photos

Objective:

To design the basic structure of a Crossword Bookstore store website by planning its layout, content, and visual elements, ensuring it meets user needs and effectively represents the brand.

Theory:

Project Design and Plan Document for Console Store Website

1. Brief Information about the Project

The project aims to develop a modern, user-friendly, and visually engaging website for the Crossword Bookstore, catering not just to book lovers but also console and gaming enthusiasts. This platform is intended to bridge the gap between traditional bookstores and evolving digital consumer behavior by offering a comprehensive online experience. The website will include detailed product pages, customer reviews, a compelling "About Us" section, and interactive elements such as login/registration features, testimonials, and a contact platform. It will be responsive, mobile-friendly, and designed to provide a personalized experience based on user preferences and profiles.

2. Goals and Deliverables

Project Goals

- Develop a fully functional and aesthetically pleasing website for the Crossword Bookstore.
- Create an engaging experience that highlights the store's products and ethos.
- Integrate a user authentication system for secure login and registration.
- Provide seamless navigation across all devices with a responsive design.
- Build trust with users through testimonials and contact accessibility.
- Cater to console fans through specialized content and product recommendations.

Project Deliverables

Website Pages

- Home Page – Introduction to the store with promotional highlights.
- About Page – Background, mission, and story of the brand.
- Products/Services Page – Categorized product listings (books and consoles).
- Testimonials Page – Customer reviews and feedback.
- Contact Page – A form and details to get in touch with the store.
- Login Page – Secure user authentication page.
- Registration Page – New user sign-up page.
- Blog Section (Optional) – Placeholder for future content.

Core Features

- Uniform header and footer with global navigation links.
- Fully functional and secure login/registration system.
- Responsive layout for mobile, tablet, and desktop users.
- Professional design with optimized fonts, visuals, and colors.
- SEO and accessibility-ready structure (optional for future scale-up).

3. Finalized Modules of the Project

The website will be built using a modular approach to ensure ease of maintenance, scalability, and future enhancements. Each module represents a key component of the user journey or functionality.

1. Home Page Module

- Purpose: The central hub of user interaction upon landing.
- Key Features:
 - Hero image or banner with a strong call-to-action ("Explore Now", "Shop Now").

- Promotional highlights such as bestsellers or offers.
- Direct navigation to About, Products, Login, and more.
- Footer with quick links and store contact details.

2. About Page Module

- Purpose: Build trust and brand connection.
- Key Features:
 - Background on Crossword's journey, especially its console-related section.
 - Brand values such as quality, customer-first service, and sustainability.
 - Visual storytelling using images or animations.

3. Products/Services Page Module

- Purpose: Showcasing available items and encouraging purchases.
- Key Features:
 - Organized into categories (e.g., Fiction, Non-Fiction, Gaming Consoles).
 - Filters based on genre, price, and popularity.
 - Product cards with image, description, price, and "Add to Cart" button.

4. Testimonials Page Module

- Purpose: Build credibility and social proof.
- Key Features:
 - User reviews in carousel or grid format.
 - Option for users to submit new reviews.
 - Highlight high-rated or verified buyer comments.

5. Contact Page Module

- Purpose: Facilitate customer communication.
- Key Features:
 - Contact form (Name, Email, Subject, Message).
 - Embedded map showing store location.
 - Static contact information (Email, Phone, Address).

6. Login Page Module

- Purpose: Authentication for returning users.
- Key Features:
 - Fields for username/email and password.
 - "Forgot Password?" link.
 - Redirect to registration for new users.
 - Basic error handling for incorrect input.

7. Registration Page Module

- Purpose: Onboarding new users.
- Key Features:
 - Fields: Full Name, Email, Username, Password, Confirm Password.
 - Checkbox for newsletter subscription.
 - Secure password validation and confirmation.
 - Terms & Conditions agreement checkbox.

8. Footer Module

- Purpose: Persistent site-wide navigation and info access.
- Key Features:
 - Quick links (Privacy Policy, Terms of Use).
 - Social media icons (Facebook, Instagram, Twitter, etc.).
 - Store address, helpline, and copyright.

4. Define the Audience

Understanding the diverse user base of Crossword is essential for delivering relevant content, intuitive design, and value-added features. The website is designed with a user-centric approach to address varying needs:

a. Avid Readers & Book Collectors

- Who They Are: Passionate readers who frequently purchase books.
- What They Need:
 - Detailed book listings with author info and reviews.
 - Access to premium/collector editions.
 - Loyalty points, reward systems, and pre-order features.

b. Professionals & Remote Workers

- Who They Are: Working individuals seeking productivity books or using the bookstore café.
- What They Need:
 - Section for leadership, business, and productivity books.
 - Details of café amenities (Wi-Fi, seating).
 - In-store pickup for online orders.

c. Students

- Who They Are: Budget-conscious, tech-savvy learners.
- What They Need:
 - Discounts on academic materials and exam prep.
 - Fast, mobile-friendly navigation and checkout.
 - Bundles and special student offers.

d. Parents & Children

- Who They Are: Families looking for educational or recreational books.
- What They Need:
 - Kid-friendly interface with age-wise filters.
 - Calendar of storytelling sessions or kid events.
 - Activity kits and gift options for birthdays.

e. Tourists & Casual Visitors

- Who They Are: Occasional buyers or tourists exploring local culture.
- What They Need:
 - Map or store locator for nearby branches.
 - Regional literature and Indian author highlights.
 - Multilingual support or guidance.

f. New Visitors (First-Time Users)

- Who They Are: Users unfamiliar with the brand.
- What They Need:

- Simple, engaging onboarding and website guidance.
- Compelling brand story to build trust.
- Prominent customer testimonials.

g. Online Shoppers

- Who They Are: Users who prefer a fully digital experience.
- What They Need:
 - Secure, fast, and seamless shopping experience.
 - Filters by genre, price, language, and availability.
 - Wishlist, fast checkout, and delivery tracking.

Conclusion

The Crossword Bookstore Console Website will serve as a modern, responsive, and modular e-commerce platform designed to enhance the user experience, promote brand identity, and meet the evolving needs of diverse customer segments. With an intuitive UI/UX, secure login/registration system, and clear segmentation of features, the website will function as a powerful digital storefront that complements Crossword's offline presence while introducing innovative online conveniences.

Website Features Mapped to Audience Needs:

Audience Segment	Key Website Features Needed
Avid Readers & Collectors	Detailed product pages, wishlist, pre-order alerts, and curated lists
Professionals & Remote Workers	E-book/Audiobook sections, co-working info, fast online purchase + pickup
Students	Student discounts, academic book bundles, exam prep filters, mobile-optimized site
Parents & Children	Age-based filters, events section, interactive content for kids
Tourists & Casual Visitors	Location-based store finders, regional literature categories, local author highlights
New Visitors	Clean UI/UX, brand story, testimonials, FAQs
Online Shoppers	Secure login, robust filters, real-time inventory, order tracking, fast checkout

Why Understanding the Audience is Important

- Delivers a personalized and intuitive website experience that adapts to different user goals—whether it's discovering a new novel or ordering a textbook.
- Increases customer engagement and loyalty by addressing specific interests and preferences.
- Strengthens the Crossword brand through storytelling, trust-building elements, and responsive design.
- Enables targeted marketing campaigns, such as student sales, audiobook promotions, or family events, ensuring better ROI and reach.

1. Describe pain points & the ideal experience (On the basis of existing systems)

1. Identifying Pain Points of Existing Systems

a. Pain Point: Poor Navigation and Cluttered Interface

- Issue: The current website layout is often complex and cluttered, making it difficult for users to find specific books or categories.
- Impact: Users may get frustrated, leading to a higher bounce rate or abandonment of the site altogether.

b. Pain Point: Limited Online Purchasing Functionality

- Issue: Many existing systems lack user-friendly online shopping features, with complicated checkout processes and missing options like wishlist or easy cart management.
- Impact: Customers who prefer the convenience of online shopping may abandon their purchases, reducing conversion rates and overall sales.

c. Pain Point: Lack of Mobile Optimization

- Issue: The website isn't optimized for mobile devices, causing slow loading times and difficulty navigating or viewing content on smartphones or tablets.
- Impact: Mobile users struggle to explore books or complete transactions, resulting in a loss of potential customers who are increasingly shopping via mobile devices.

d. Pain Point: Insufficient Product Information

- Issue: Product pages often lack comprehensive details such as book synopses, author bios, customer reviews, and clear pricing information.
- Impact: Customers may leave the site due to incomplete information, as they need more context to make informed purchasing decisions.

e. Pain Point: Weak Engagement Strategies

- Issue: The website does not effectively leverage engagement strategies such as loyalty programs, discounts, or community-building content.
- Impact: Missed opportunities for customer retention, brand loyalty, and building an engaged community of readers and book lovers.

f. Pain Point: Inefficient Contact and Location Details

- Issue: Contact information and store location details may not be easily accessible, forcing users to dig through multiple pages to find this basic information.
- Impact: Customers may become frustrated or discouraged from visiting physical stores or making inquiries, potentially opting for competitors.

g. Pain Point: No Personalization Options

- Issue: The website lacks personalized features, such as book recommendations based on past purchases or preferences.

- Impact: Without personalization, customers may feel the website is impersonal, which can lower satisfaction and repeat business.

2. Crafting the Ideal Experience

To address the identified pain points, the website design should offer a seamless, intuitive, and engaging experience for all users. Here's how we can improve the current system:

a. Intuitive Navigation and Clean Design

- Solution: Use a clear and consistent layout with a sticky navigation bar for easy access to key sections (Home, About, Categories, Contact, Login/Sign Up).
- Benefit: Users can easily navigate through the website, find what they need, and enjoy a smooth experience from homepage to checkout.

b. Seamless Online Purchasing

- Solution: Implement a robust, easy-to-use e-commerce system with a user-friendly checkout process. Allow users to add items to their cart, save items to a wishlist, and complete purchases in a few clicks.
- Benefit: A streamlined purchasing experience leads to higher conversion rates, fewer abandoned carts, and a more satisfied customer base.

c. Mobile-Responsive Design

- Solution: Prioritize a mobile-first design, ensuring that the site works perfectly across all devices. Use responsive grids, touch-friendly navigation, and optimize for fast load times.
- Benefit: A smooth mobile experience encourages users to shop on-the-go, increasing sales from mobile users who are essential to the growing e-commerce market.

d. Comprehensive Product Information

- Solution: Include high-quality images, detailed product descriptions (e.g., book summaries, author info, reviews), price, and availability. For physical books, provide information on book editions, genres, and ISBN numbers.
- Benefit: Clear and informative product pages empower customers to make quick, confident purchasing decisions.

e. Customer Engagement Features

- Solution: Introduce features like a loyalty program that offers reward points with purchases, a blog with content on book recommendations, author interviews, and in-store events. Highlight customer reviews on product pages.
- Benefit: These features help build long-term customer relationships, encourage repeat business, and foster a loyal customer base.

f. Easy Access to Contact and Location

- Solution: Display a dedicated “Contact Us” page with inquiry forms, phone numbers, email addresses, and an interactive map showing store locations.

- Benefit: Users can quickly get in touch with customer support or locate physical stores, improving customer satisfaction and reducing frustration.

g. Personalization

- Solution: Allow users to create accounts to save favorite books, manage orders, and receive personalized book recommendations based on browsing history. Send tailored offers through email or push notifications.
- Benefit: Personalized experiences enhance customer engagement, making users feel valued and likely to return.

3. The Ideal User Journey

Step 1: Visiting the Website

- Experience: Users arrive at a visually appealing homepage with intuitive navigation and clearly displayed categories for books, events, and offers.

Step 2: Browsing the Catalog

- Experience: Users can easily navigate product categories (e.g., Fiction, Non-Fiction, Bestsellers) and use filters such as price, genre, or author to refine their search.

Step 3: Making a Purchase

- Experience: Users can add books to their cart with ease and complete the checkout process in just a few steps, including secure payment options and address details for delivery.

Step 4: Finding Location or Contacting Support

- Experience: If users prefer to visit in person, they can easily find store locations and hours via the website's interactive map and contact page.

Step 5: Engaging with Content

- Experience: Users read insightful blog posts or watch videos featuring author interviews, book reviews, or upcoming events. Customer reviews on products add an extra layer of engagement.

Step 6: Creating Loyalty

- Experience: Registered users receive personalized book recommendations, special offers, and points towards future purchases, encouraging loyalty and repeat visits.

This ideal experience ensures that the Crossword Bookstore's website is user-friendly, visually appealing, and highly interactive, addressing the pain points of existing systems while offering a seamless, enjoyable shopping journey for all visitors.

2. Set the Visual Direction for Crossword Bookstore Website

1. Visual Design Goals

The visual design of the Crossword Bookstore website should be inviting, engaging, and user-friendly while reflecting the unique character of the bookstore. The visual design should aim to:

- Welcoming and Comfortable: The design should mirror the cozy atmosphere of the physical store and offer a relaxed and comfortable experience for online shoppers, making them feel at ease when browsing the website.
- Modern and Minimalistic: A clean, uncluttered design with modern elements will ensure the website remains professional while also being user-centric and easy to navigate.
- Brand Representation: The design, color palette, typography, and imagery should reinforce the bookstore's values: a love for books, a community-driven environment, and a passion for curating and sharing knowledge. It should appeal to a diverse audience including book lovers, young readers, and students.

2. Defining the Core Visual Elements

a. Color Palette

A warm and inviting color palette will enhance the user experience, giving the website a cozy and approachable feel. Drawing inspiration from books and natural tones, the color scheme should evoke warmth and creativity.

Color	Hex Code	Usage
Bookish Brown	#6F4E37	Used in the header, footer, buttons, and key accents to establish warmth.
Soft Cream	#F5F5DC	Background color for contrast and a gentle atmosphere throughout the pages.
Deep Espresso	#3C2F2F	Text, headings, and other key accents for legibility and emphasis.
Warm Latte	#FAF3E0	Secondary background color to provide depth and subtle contrasts without overpowering the design.
Olive Green	#556B2F	Call-to-action buttons to stand out against the warm background, offering a natural, earthy feel.

b. Typography

The typography should be warm and welcoming while maintaining a clean and modern style that enhances readability.

- Primary Font: Poppins or Roboto (Sans-serif) – Used for headings and calls to action to create emphasis and clarity.
- Secondary Font: Open Sans or Lora – For body text and descriptions, offering excellent readability and a touch of sophistication.
- Font Attributes: Use bold for headings and lighter weights for body text to maintain a balanced, easy-to-read visual hierarchy.

c. Logos and Branding

- The logo should incorporate a stylized design inspired by books, reading, or an item related to a bookstore (e.g., a book icon or stack).
- The font should align with the modern yet cozy feel of the bookstore.
- Create a monochrome version of the logo for minimalistic or space-constrained areas like headers or footers.

d. Imagery and Icons

High-quality visuals will play a significant role in inviting customers to browse the website and make them feel connected to the brand.

- Photography:
 - Showcase vibrant images of books, cozy corners, and community spaces to reflect the atmosphere of the store.
 - Include close-ups of new releases, bestselling books, or customer favorites to engage visitors.
- Icons:
 - Simple, intuitive icons for categories like "Catalog," "Location," "Events," and "Contact."
- Hero Images:
 - Use a rotating banner or static hero image on the homepage that highlights featured books, sales, or special promotions.

3. Applying Visual Design to Pages

a. Home Page

- Banner Area: A large hero image with text overlay showcasing a seasonal promotion, new book arrivals, or exclusive deals.
- Color Scheme: The warm tones of bookish brown and soft cream will create a welcoming vibe, while action buttons will be highlighted in olive green.
- Typography: Bold, welcoming headlines such as "Welcome to Crossword Bookstore" in large font, with subheadings about the latest books or offers.

b. About Page

- Show authentic images of the bookstore, staff, and community events.
- Soft, inviting colors should dominate the page layout, aligning with the tone of the bookstore's story.
- Include sections about the bookstore's history, mission, and community initiatives.

c. Product/Service Page

- Product cards for books should prominently display images, prices, and brief descriptions.
- Each product page will have high-quality images, an option to view more details (e.g., summary, reviews), and the ability to add to cart easily.
- Hover effects should highlight interactive elements like "Add to Cart" or "More Info."

d. Testimonial Page

- Use real customer photos alongside their testimonials to build trust and a sense of community.
- A clean, simple card layout will help organize feedback and make it easy for users to scroll through testimonials.

e. Contact Page

- Embed a map to display the bookstore's physical location.

- Include contact forms for general inquiries and support, using the green call-to-action buttons to ensure they stand out.

f. Login and Registration Pages

- Keep the form layout minimal, easy to navigate, and aesthetically pleasing with soft background colors and clear input fields.
- Use consistent branding with call-to-action buttons in bookish brown or olive green.

4. Layout and Design Hierarchy

The design layout should create a seamless flow for the user to explore, browse, and purchase books:

1. Headers and Banners: Prominent and inviting, introducing key content, such as seasonal promotions or bestselling books.
2. Navigation Bar: Sticky and unobtrusive, ensuring easy navigation to various sections of the website (Home, Catalog, About, Contact, etc.).
3. Sections and Grids: Content should be structured neatly in grids or blocks, with clear breaks between sections to avoid overwhelming the user.
4. Call-to-Action: Buttons like "Add to Cart," "View Details," or "Sign Up" should be prominently displayed using contrasting colors (olive green or bookish brown) to catch the user's attention.

5. Expected Impact of Visual Direction

- Enhanced Engagement: The warm, welcoming design will encourage users to explore the website further and make them feel comfortable browsing.
- Stronger Branding: A cohesive color palette, typography, and imagery will strengthen the store's identity and reinforce its values.
- Better Retention: A well-organized, visually appealing layout will enhance the user experience, leading to repeat visits and customer loyalty.
- Higher Conversions: Effective use of call-to-action elements and well-placed product information will lead to increased sales and higher conversion rates.

7. Map out the Project structure

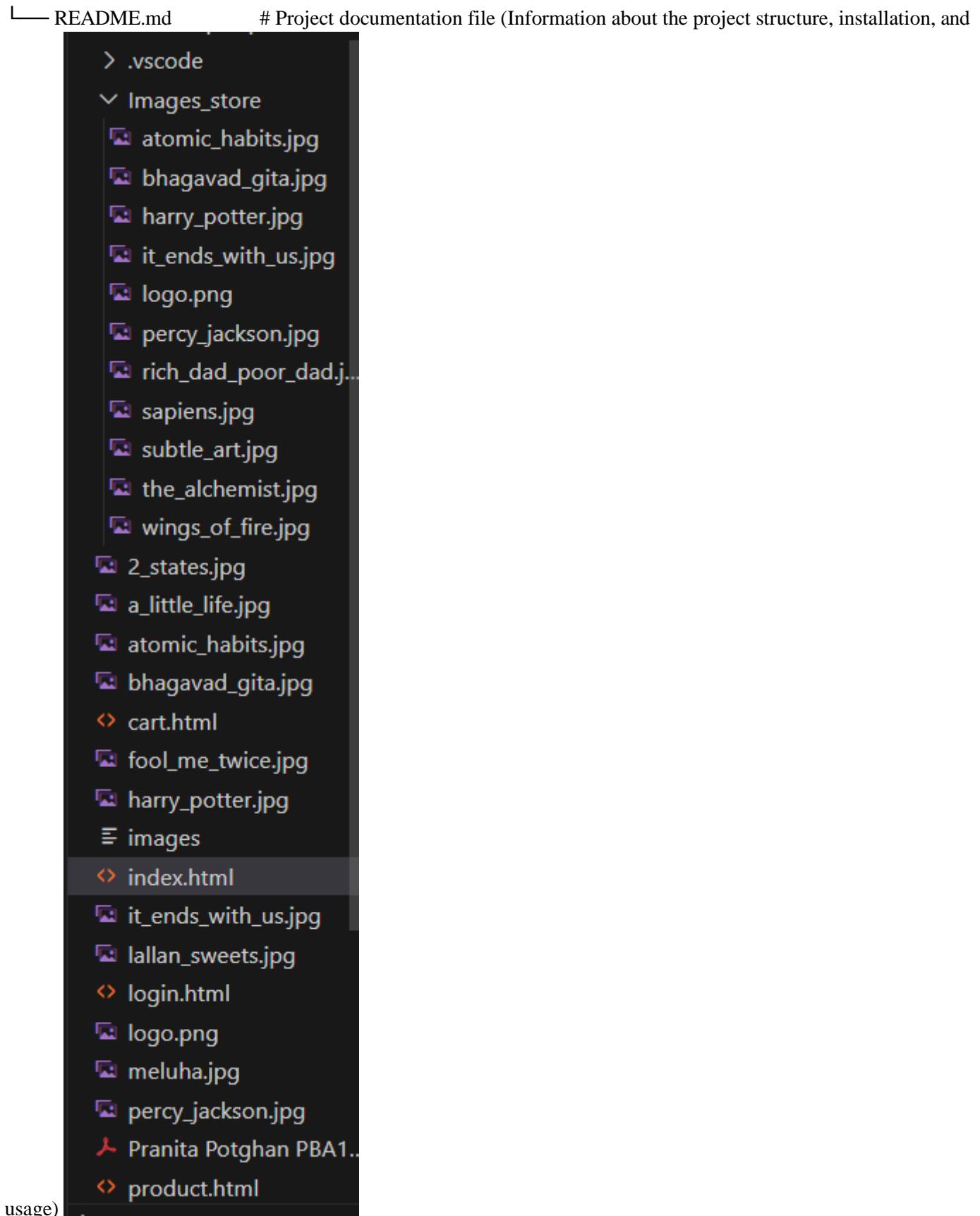
console_store_website/

```

|   |
|--- index.html      # Home page (Landing page with featured products, promotions, and navigation)
|--- about.html      # About page (Information about the bookstore, history, mission)
|--- books.html       # Products/Services page (Book catalog, new arrivals, categories)
|--- testimonials.html # Testimonials page (Customer feedback and reviews)
|--- contact.html     # Contact page (Contact form, Google Map, customer support details)
|--- login.html        # Login page (Login form for existing users)

```

```
|── register.html      # Registration page (Sign up form for new users)
|── newsletter.html   # Optional blog page (Blog for bookstore updates, book reviews, events)
|
|── assets/
|   |── css/
|   |   |── style.css    # Global CSS (Base styles for layout, fonts, etc.)
|   |   |── responsive.css # Media queries for mobile optimization (Adapt for different screen sizes)
|   |   |
|   |── js/
|   |   |── main.js       # Interactive scripts (Functions for interactions like adding to cart)
|   |   |── formValidation.js # Scripts for login, registration, and form validation
|   |   |
|   |── images/
|   |   |── logo.png      # Website logo (Branding element)
|   |   |── homepage_banner.jpg # Hero banner for the home page (Featured promotions or products)
|   |   |── console_catalog/  # Images for catalog items (Book covers, product images)
|   |   |── portable_consoles_photos/ # Images for about page (Staff, bookstore ambiance)
|   |   |── icons/          # Icons for UI elements (Cart, search, categories, etc.)
|
|── fonts/
|   |── Poppins/        # Primary font files (for headings and calls to action)
|   |── OpenSans/        # Secondary font files (for body text and descriptions)
```



8. Visual Design Ideas for All Pages:

Colors:

- Earth Tones: Use a warm color palette of brown, cream, and green for a cozy, welcoming atmosphere. These colors work well for conveying a sense of authenticity and nature, aligning with the idea of a bookstore with organic or sustainable products.
- Highlight Color: Gold or maroon can be used for key call-to-action (CTA) buttons, navigation items, and accents to bring a touch of elegance and attention to important actions like “Explore Menu” or “Purchase Now.”

Fonts & Typography:

- Headers: Use an inviting, elegant serif font like Playfair Display for headers. This creates a refined, literary feel, perfect for a bookstore.
- Body Text: Use a clean, readable sans-serif font like Lato or Roboto for the main body text. These fonts are modern and legible, ensuring a smooth reading experience for visitors.
- Button Text: Bold sans-serif fonts like Montserrat or Poppins for buttons ensure clarity and prominence in the UI.

Logos:

- Crossword Bookstore Logo: A stylized console device combined with elements representing books or gaming consoles, depending on whether the focus is more on gaming consoles or a wider bookstore experience. The logo should be sleek and modern but with a touch of playful elegance, hinting at both the seriousness of books and the excitement of consoles.

Images:

- Realistic, High-Quality Visuals: Use natural lighting to create warm, inviting images. Showcasing both the gaming consoles and cozy bookstore ambiance through visually appealing photography can enhance the overall user experience.
- Focus on Natural Elements: Given the warm, earth-tone color palette, include images that highlight natural elements such as wooden shelves, soft lighting, and organic materials. If the store has physical locations, images of its interiors, such as well-lit aisles, cozy corners, or customers browsing, would add an authentic touch.

9. Page-Specific Content, Images, and Layout Ideas:

1. Home Page

- Layout Ideas:
 - Header Section:
 - Use a fixed navigation bar to ensure easy access to all sections. Place the logo on the left, centralize the navigation links (Home, About, Products, Testimonials, Contact), and have the Login/Register button on the right.
 - Hero Section:
 - Full-width background image showcasing a high-quality visual of a console, product offerings, or the store's ambiance.
 - Overlay text with a catchy tagline like “Brewed with Love” for an inviting, friendly tone.

- CTA button: “Explore Menu” or “Visit Us” linking to the catalog or about page.
- Featured Section:
 - Use card-based layout or tiles to highlight featured products or offers like “Latte of the Day” or seasonal beverages (or in the case of a bookstore, “Book of the Week”).
- Footer:
 - Display quick links, social media icons, and basic contact info.
- Image Ideas:
 - A banner showcasing products (gaming consoles, cozy interiors, etc.).
 - Close-up shots of books, consoles, and cozy seating arrangements within the store.

2. About Page

- Layout Ideas:
 - Story Section:
 - A timeline layout or a two-column design to tell the history of the Crossword Bookstore, dividing major milestones with icons or small dividers.
 - Team Section:
 - Use a grid layout to showcase key team members with their images and bios, particularly focusing on the store founders, team members behind the products, etc.
 - Features Section:
 - Include icons and short text blocks emphasizing special features (e.g., Organic gaming consoles, Eco-friendly packaging, etc.).
- Image Ideas:
 - Photos of the founders, the team in action, or the console creation process.
 - Images showcasing the cozy ambiance of the store with customers browsing.

3. Products Page

- Layout Ideas:
 - Categories Section:
 - Divide the page into distinct product categories (e.g., Consoles, Accessories, Teas, Snacks, etc.). Each category can feature product images, descriptions, and prices in a clean grid format.
 - Highlight Section:
 - A carousel or slider showcasing top picks or bestsellers.

- CTA Section:
 - For each product, include “Purchase Now” or “Learn More” buttons to lead visitors to login/registration or detailed product pages.
- Image Ideas:
 - High-quality images of consoles, books, teas, and other items displayed in aesthetic setups, possibly with a minimalistic backdrop.

4. Testimonials Page

- Layout Ideas:
 - Feedback Section:
 - Use a carousel slider to rotate testimonials or feature them in a grid layout. Include star ratings for visual appeal.
 - Submit a Testimonial Section:
 - Add a simple form that lets visitors submit their feedback.
- Image Ideas:
 - Photos of happy customers enjoying products, either browsing the store or interacting with items.
 - Stylized screenshots or images of written reviews, possibly framed by aesthetic elements like shelves.

5. Contact Page

- Layout Ideas:
 - Contact Form Section:
 - Use a clean, minimalist form that allows visitors to input their name, email, subject, and message.
 - Map Section:
 - Embed a Google Map to show the exact location of the Crossword Bookstore.
 - Info Section:
 - Use icons for contact methods (phone, email) and display operating hours clearly.
- Image Ideas:
 - Photos of the storefront.
 - Visuals of contact icons (phone, email, map), possibly accompanied by an illustration of the store’s location or contact tools.

6. Login Page

- Layout Ideas:
 - Minimalist Interface:
 - Use a centered form with fields for email and password.
 - Include clear buttons for login, and a link for forgotten passwords.
 - For new users, link directly to the Registration Page.
 - Include a side banner or background image representing the store (e.g., an image of a console or cozy store corner).
- Image Ideas:
 - A background image showing a stylized console with a transparent overlay for the form area, ensuring a clean layout.

7. Registration Page

- Layout Ideas:
 - Clean Form Layout:
 - Focus on clear form fields like Full Name, Email, and Password with a password strength indicator.
 - Use checkboxes for opting into newsletters or special offers.
 - CTA Section:
 - Use a tagline like “Join us and enjoy exclusive rewards!”
- Image Ideas:
 - A welcoming banner image showing happy customers or product offerings.
 - Icons representing the form fields (e.g., a lock for password).

11. Crossword Bookstore Website Structure & Content Plan

1. Home Page

Header:

- Logo: Minimal, elegant logo with a stylized open book or quill pen.
- Navigation Links: Home | About Us | Books | Testimonials | Blog | Contact
- Call-to-Action: “Browse Books” or “Join Our Readers Club”

Hero Section:

- Background Image: Full-width cozy bookstore or reading nook photo.
- Text Overlay: “*Your Next Favorite Book is Just a Click Away.*”
- CTA Button: “Explore the Collection”

About Teaser Section:

- A short paragraph introducing Crossword Bookstore: *“A haven for readers since 2025. Discover bestsellers, rare finds, and curated reads—all in one literary hub.”*
- Link to “About Us”

Popular Books / Featured Picks Section:

- Grid view of 3–4 featured books (bestsellers, staff picks, new arrivals)
- Each with image, title, short teaser, and “Add to Cart” / “Read More”

Testimonials Preview:

- Carousel of 2–3 customer reviews with stars & short snippets
- Button: “Read All Reviews”

Footer:

- Quick Links | Social Media Icons | Address & Map | Newsletter Signup

2. About Page

Introduction:

- Brief story of the Crossword Bookstore: *“Founded in 2025 by book lovers, for book lovers, Crossword Bookstore brings stories to life—while supporting sustainability and community literacy.”*

Meet the Team:

- Photos and bios of team members: store manager, curators, book reviewers.

Our Mission & Values:

- Highlights like:
 - Promoting sustainable reading through second-hand books
 - Supporting local authors
 - Hosting community events & book clubs

Store Locations & Hours:

- Google Maps integration
- Weekly timings, special event dates

3. Books / Product Page

Book Categories:

- Fiction | Non-Fiction | Children's | Academic | Staff Picks | New Arrivals | On Sale

Each Book Listing Includes:

- Cover Image
- Title, Author, Genre, Price
- Condition (New / Pre-loved)
- Quick preview or synopsis
- “Add to Cart” or “View Details”

Special Collections Section:

- “Monthly Picks,” “Editor’s Choice,” “Rare Finds”

Online Purchasing & Delivery Info:

- Payment methods accepted
- Delivery timeline
- Pickup available at select stores

4. Testimonials Page

Customer Reviews:

- Grid or carousel format
- Each review with stars, name, and text

Featured Reviews:

- Pulled from Google, Instagram, etc.

Submit Your Review:

- Simple form for customers to add feedback

Review Policy:

- Short section about authenticity and transparency in displaying reviews

5. Contact / Support Page

Contact Form:

- Name, Email, Message
- Response within 24–48 hours

Support Info:

- Email: support@crosswordbookstore.com
- Phone: +91 XXXXXXXXXX
- FAQs Link

Location & Map:

- Address with Google Map integration

Social Media:

- Facebook | Instagram | Twitter | Goodreads

6. Starter Blog Posts

Blog Categories:

- Reading Lists: “Top 10 Must-Reads This Summer”
- Behind the Shelves: “How We Curate Books at Crossword”
- Literary Culture: “Why Second-Hand Books Hold Magic”
- Events: Book launches, guest author sessions, reading clubs

7. Login & Registration Pages

Login Page:

- Username/Email + Password
- “Forgot Password?”

Register Page:

- Name, Email, Password
- Checkbox: “Subscribe to Newsletter”

Account Dashboard:

- Order history, Wishlist, Profile info

8. Overall Website Structure

- Home

- About
- Books/Product Catalog
- Testimonials
- Blog
- Contact
- Login/Register

9. Visual Design

Color Palette:

- Primary: Classic Navy (#1E2D5C), Book Beige (#F4EDE4), Cream (#FFF8F0)
- Accent: Maroon (#7B3F00), Leaf Green (#6DBF3A)

Fonts:

- Headings: Lora or Merriweather (classic serif)
- Body: Open Sans or Roboto (clean sans-serif)

Imagery:

- Lifestyle photos of reading nooks, customers reading, book shelves
- Product shots of books with soft backgrounds
- Community & event photos for blog/about sections

Interactive Elements:

- Hover effects on CTAs
- Icons for categories (e.g.,  for Books,  for Locations)

Conclusion

This Crossword Bookstore website structure ensures an elegant and functional experience for visitors. From browsing books to reading blog posts and submitting reviews, the platform encourages exploration and builds trust with users. The thoughtful design, warm branding, and community feel create a digital space as welcoming as your physical store.

Experiment No.2

Problem Statement:

- Create a detailed home page for the Crossword Book Store website.
- Create a detailed menu/product page for the website, listing all available items categorized appropriately.
- Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout.
- Create an about us page that provides detailed information about the Crossword Book Store history, mission, and team.
- Create a contact page that allows customers to easily get in touch with the coffee shop through a form.
- Design and implement admin/user registration form for the website.
- Design and implement admin/user login form for the website.

Project Objective

To create a Crossword Bookstore Webpage using HTML, offering a user-friendly online platform for purchasing books, e-books, and related products. The website will provide a smooth, responsive experience with an emphasis on customer satisfaction, ease of navigation, and comprehensive book categories.

Introduction

In today's digital world, e-commerce platforms are essential for efficient buying and selling. This project focuses on developing a responsive and functional online bookstore where customers can browse a variety of books, from bestsellers to niche genres. The website aims to make book purchasing more accessible while promoting literacy and a love for reading.

The project integrates both front-end and back-end components to offer a seamless user experience, featuring product listings, cart management, and a user-friendly checkout system. The implementation will initially be based on HTML, CSS, and optionally extend into JavaScript or server-side scripting for future enhancements.

1. Home Page

Features

- Hero section featuring promotions or bestselling books
- Navigation bar for easy access to different sections (e.g., Categories, Contact Us)
- Call-to-action buttons such as “Shop Now” and “Browse Categories”
- Featured books or customer reviews

Importance

The homepage serves as the first impression for users, helping establish brand credibility and encouraging users to explore the website further. A clean, visually appealing layout increases engagement and improves user retention.

Technologies Used

- HTML for structure

- CSS for layout and styling
- Optional: CSS animations or JavaScript for interactive elements

2. Product / Menu Page

Features

- Categorized listings such as:
 - Fiction, Non-fiction, Educational, Children's Books, etc.
- Product details include:
 - Book cover image, title, and author
 - Price and availability
 - Description and rating
 - “Add to Cart” button

UX Enhancements

- Filter options (by genre, price range, or author) to improve product discoverability
- Sorting options for price, popularity, or ratings

Importance

A well-structured catalog allows users to easily browse and compare products. Clear presentation of book details improves user satisfaction and enhances conversion rates.

3. Cart Page

Features

- Display of all added products with details (quantity, subtotal, etc.)
- Options to update or remove items from the cart
- Checkout button leading to the final purchase stage

Optional Enhancements

- Cart persistence using localStorage (so the cart remains intact even after the page is refreshed)
- Real-time price updates when items are modified in the cart

Importance

The cart page plays a crucial role in the buying decision, allowing users to review their selections and finalize their purchase. An intuitive cart system reduces abandoned carts and improves the overall conversion rate.

4. About Us Page

Contents

- History of the bookstore

- Vision and mission statement
- Founder's message or store background
- Team introduction with photos and bios (optional)

Importance

Building trust is especially important in e-commerce. The "About Us" page humanizes the brand and assures customers that they are purchasing from a reliable source.

5. Contact Page

Form Includes

- Full Name
- Email Address
- Subject
- Message

Additional Elements

- Phone number and store address
- Embedded Google Map for location
- Social media links (Facebook, Instagram, etc.)

UX Focus

Easy access to customer service and contact information promotes user confidence and satisfaction. Clear communication helps resolve concerns and improves the overall customer experience.

6. User/Admin Registration Form

Fields

- Full Name
- Email Address or Phone Number
- Password & Confirm Password
- User Type (dropdown or radio buttons for User/Admin)

Importance

Enabling user and admin registration allows for personalized shopping experiences, order history tracking, and the potential for loyalty programs. It also lays the foundation for a secure backend for inventory and order management.

7. User/Admin Login Form

Fields

- Username or Email
- Password
- "Remember Me" checkbox
- "Forgot Password?" link

Security (for future phases)

- Basic input validation
- Password hashing and secure storage
- Two-factor authentication and login rate limiting in production

Access Levels

- Users: Can browse products, view order history, and purchase items
- Admins: Can manage inventory, process orders, and view customer analytics

Technological Stack Overview (for future versions)

Function	Tools (Planned)
Frontend	HTML, CSS, JavaScript
Backend	PHP / Node.js
Database	MySQL / MongoDB
Authentication	Session management, JWT
Storage	LocalStorage / Server DB

Sustainability Impact

The bookstore supports the promotion of literacy and knowledge sharing by providing access to books at affordable prices. By selling both physical and digital formats, the store helps reduce paper waste and environmental impact. Additionally, the platform aims to extend the lifespan of books by encouraging second-hand book sales and book exchanges, fostering a more sustainable reading culture.

Code:

A. Home page:

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

```
<title>Crossword Bookstore</title>
```

```
<style>
```

```
body {  
    font-family: Arial, sans-serif;  
    background: #f9f9f9;  
    margin: 0;  
}
```

```
.header {
```

```
    background-color: #f4c842;  
    padding: 15px 20px;  
    text-align: center;  
    font-size: 24px;  
    font-weight: bold;  
}
```

```
.navbar {
```

```
    background-color: #333;  
    overflow: hidden;  
    display: flex;  
    justify-content: space-between;  
    padding: 10px 20px;  
}
```

```
.navbar a {
```

```
    color: white;  
    text-decoration: none;  
    padding: 10px 15px;  
}
```

```
.navbar a:hover {
```

```
background-color: #f4c842;  
color: black;  
}
```

```
.cart {  
color: white;  
font-weight: bold;  
}
```

```
.categories {  
text-align: center;  
padding: 20px;  
}
```

```
.category-list {  
display: flex;  
justify-content: center;  
gap: 15px;  
flex-wrap: wrap;  
}
```

```
.category {  
background: #e0e0e0;  
padding: 10px 20px;  
border-radius: 5px;  
cursor: pointer;  
}
```

```
.category:hover {  
background-color: #f4c842;  
}
```

```
.book-container {
```

```
    display: flex;
```

```
    flex-wrap: wrap;
```

```
    justify-content: center;
```

```
    gap: 20px;
```

```
    padding: 20px;
```

```
}
```

```
.book {
```

```
    background: white;
```

```
    padding: 15px;
```

```
    width: 220px;
```

```
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
    text-align: center;
```

```
    border-radius: 8px;
```

```
}
```

```
.book img {
```

```
    width: 100%;
```

```
    height: 200px;
```

```
    object-fit: cover;
```

```
}
```

```
.book p.description {
```

```
    font-size: 14px;
```

```
    color: #555;
```

```
    margin: 10px 0;
```

```
}
```

```
.book button {
```

```
background: #f4c842;  
border: none;  
padding: 8px;  
cursor: pointer;  
margin-top: 10px;  
}
```

```
.book button:hover {  
background-color: #ffae00;  
}
```

```
.footer {  
text-align: center;  
background: #333;  
color: white;  
padding: 10px;  
margin-top: 20px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="header">Crossword Bookstore</div>
```

```
<div class="navbar">  
<div>  
<a href="index.html">Home</a>  
<a href="books.html">Books</a>  
<a href="about.html">About</a>  
<a href="contact.html">Contact</a>  
</div>
```

```
<div>
  <a href="login.html">Login</a>
  <a href="register.html">Register</a>
```

```
  <a href="cart.html" class="cart">Cart (<span id="cart-count">0</span>)</a>
</div>
```

```
</div>
```

```
<div class="categories">
```

```
  <h2>Book Categories</h2>
```

```
  <div class="category-list">
```

```
    <div class="category">Fiction</div>
```

```
    <div class="category">Non-Fiction</div>
```

```
    <div class="category">Science</div>
```

```
    <div class="category">Technology</div>
```

```
    <div class="category">Children</div>
```

```
</div>
```

```
</div>
```

```
<div class="book-container">
```

```
  <!-- Book 1 -->
```

```
  <div class="book">
```

```
    
```

```
    <h3>Atomic Habits</h3>
```

```
    <p>₹499</p>
```

```
    <p class="description">Transform your habits and life with James Clear's practical strategies for building better routines.</p>
```

```
    <button onclick="addToCart()">Add to Cart</button>
```

```
</div>
```

```
<!-- Book 2 -->
```

```
  <div class="book">
```

```

```

```
<h3>Bhagavad Gita</h3>
```

```
<p>₹299</p>
```

```
<p class="description">A timeless spiritual dialogue between Krishna and Arjuna on life, duty, and righteousness.</p>
```

```
<button onclick="addToCart()">Add to Cart</button>
```

```
</div>
```

```
<!-- Book 3 -->
```

```
<div class="book">
```

```

```

```
<h3>Harry Potter</h3>
```

```
<p>₹599</p>
```

```
<p class="description">J.K. Rowling's magical series that chronicles the life of the young wizard, Harry Potter.</p>
```

```
<button onclick="addToCart()">Add to Cart</button>
```

```
</div>
```

```
<!-- Book 4 -->
```

```
<div class="book">
```

```

```

```
<h3>It Ends With Us</h3>
```

```
<p>₹399</p>
```

```
<p class="description">Colleen Hoover's powerful novel about love, heartbreak, and breaking the cycle of abuse.</p>
```

```
<button onclick="addToCart()">Add to Cart</button>
```

```
</div>
```

```
<!-- Book 5 -->
```

```
<div class="book">
```

```

```

```
<h3>Wings of Fire</h3>
```

<p>₹350</p>

<p class="description">An inspiring autobiography of Dr. A.P.J. Abdul Kalam, India's missile man and former President.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 6 -->

<div class="book">

<h3>Percy Jackson</h3>

<p>₹499</p>

<p class="description">Rick Riordan's thrilling series that blends Greek mythology with modern adventure.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 7 -->

<div class="book">

<h3>Rich Dad Poor Dad</h3>

<p>₹399</p>

<p class="description">Robert Kiyosaki's bestselling book that challenges conventional ideas about money and wealth.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 8 -->

<div class="book">

<h3>Sapiens</h3>

<p>₹549</p>

<p class="description">Yuval Noah Harari's sweeping narrative of the history of humankind from evolution to the present.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 9 -->

<div class="book">

<h3>The Alchemist</h3>

<p>₹299</p>

<p class="description">Paulo Coelho's philosophical tale about a shepherd boy chasing his personal legend.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 10 -->

<div class="book">

<h3>Lallan Sweets</h3>

<p>₹399</p>

<p class="description">An emotional journey of a young entrepreneur who revives his family's sweet shop legacy.</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<!-- Book 11 -->

<div class="book">

<h3>The Stationary Shop of Tehran</h3>

<p>₹449</p>

<p class="description">A touching story of love and separation set in 1950s Iran by Marjan Kamali.</p>

<button onclick="addToCart()">Add to Cart</button>

```
</div>

<!-- Book 12 -->

<div class="book">
    
    <h3>Fool Me Twice</h3>
    <p>₹329</p>
    <p class="description">An intriguing tale of deception, mystery, and second chances.</p>
    <button onclick="addToCart()">Add to Cart</button>
</div>
</div>

<!-- Book 13 -->

<div class="book">
    
    <h3>2 States</h3>
    <p>₹299</p>
    <p class="description">Chetan Bhagat's witty and relatable love story about a couple from two different Indian states.</p>
    <button onclick="addToCart()">Add to Cart</button>
</div>

<!-- Book 14 -->

<div class="book">
    
    <h3>The Immortals of Meluha</h3>
    <p>₹350</p>
    <p class="description">Amish Tripathi's thrilling mythological fiction about Shiva's journey to becoming a god.</p>
    <button onclick="addToCart()">Add to Cart</button>
</div>
```

```

<!-- Book 15 -->
<div class="book">
    
    <h3>A Little Life</h3>
    <p>₹649</p>
    <p class="description">Hanya Yanagihara's powerful and emotional novel about friendship, trauma, and survival.</p>
    <button onclick="addToCart()">Add to Cart</button>
</div>

```

<div class="footer">© 2024 Crossword Bookstore. All Rights Reserved.</div>

```

<script>
let cartCount = 0;

function addToCart() {
    cartCount++;
    document.getElementById("cart-count").innerText = cartCount;
    alert("Book added to cart!");
}

</script>

```

</body>

</html>

Output:

A. Index/Home page output:

The screenshot shows a web browser window for 'Crossword Bookstore' at the URL 127.0.0.1:5500/index.html. The header includes a back arrow, forward arrow, refresh button, and a 'Finish update' button. The main navigation bar has links for Home, Books, About, Contact, Login, Register, and Cart (0). Below the navigation is a section titled 'Book Categories' with buttons for Fiction, Non-Fiction, Science, Technology, and Children. The main content area displays five book cards in a grid:

- Atomic Habits** by James Clear (₹499) - An Easy & Proven Way to Build Good Habits. **Add to Cart**
- Bhagavad Gita AS IT IS** (₹299) - A timeless spiritual dialogue between Krishna and Arjuna on life, duty, and righteousness. **Add to Cart**
- Harry Potter** (₹599) - J.K. Rowling's magical series that chronicles the life of the young wizard, Harry Potter. **Add to Cart**
- It Ends With Us** by Colleen Hoover (₹399) - Now a major motion picture! A powerful novel about love, heartbreak, and breaking the cycle of abuse. **Add to Cart**
- Wings of Fire** by A.P.J. Abdul Kalam (₹350) - An inspiring autobiography of Dr. A.P.J. Abdul Kalam, India's missile man and former President. **Add to Cart**

Code:

B. menu/product page:

code:

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Book Page - Crossword Bookstore</title>
```

```
<style>
```

```
  body {
    font-family: Arial, sans-serif;
    background: #f9f9f9;
    margin: 0;
  }
```

```
/* Header Section */
```

```
.header {
```

```
background-color: #f4c842;  
padding: 15px 20px;  
text-align: center;  
font-size: 24px;  
font-weight: bold;  
}
```

```
.logo {  
width: 50px;  
vertical-align: middle;  
margin-right: 10px;  
}
```

```
/* Navigation Bar */  
.navbar {  
background-color: #333;  
overflow: hidden;  
display: flex;  
justify-content: space-between;  
padding: 10px 20px;  
}
```

```
.navbar a {  
color: white;  
text-decoration: none;  
padding: 10px 15px;  
}
```

```
.navbar a:hover {  
background-color: #f4c842;
```

```
color: black;  
}  
  
.cart {  
    color: white;  
    font-weight: bold;  
}  
  
/* Product Container */  
  
.product-container {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 20px;  
    padding: 20px;  
}  
  
.product {  
    background: white;  
    padding: 15px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    text-align: center;  
    border-radius: 8px;  
}  
  
.product img {  
    width: 100%;  
    height: 300px;  
    object-fit: cover;  
}
```

```
.product p {  
    font-size: 14px;  
    color: #555;  
    margin: 10px 0;  
}
```

```
.product button {  
    background: #f4c842;  
    border: none;  
    padding: 8px;  
    cursor: pointer;  
    margin-top: 10px;  
}
```

```
.product button:hover {  
    background-color: #ffae00;  
}
```

```
/* Footer */  
.footer {  
    text-align: center;  
    background: #333;  
    color: white;  
    padding: 10px;  
    margin-top: 20px;  
}
```

```
/* Copyright Nav Bar */  
.copyright-nav {  
    background-color: #f4c842;
```

```
padding: 10px;  
text-align: center;  
font-size: 14px;  
}  
  
.copyright-nav a {  
margin: 0 10px;  
color: #333;  
text-decoration: none;  
}  
  
.copyright-nav a:hover {  
text-decoration: underline;  
}  
</style>  
</head>  
  
<body>  
  
<!-- Header -->  
<div class="header">  
     Crossword Bookstore  
</div>  
  
<!-- Navigation Bar -->  
<div class="navbar">  
    <div>  
        <a href="index.html">Home</a>  
        <a href="books.html">Books</a>  
        <a href="about.html">About</a>  
    </div>  
</div>
```

```
<a href="contact.html">Contact</a>

</div>

<div>

    <a href="login.html">Login</a>

    <a href="register.html">Register</a>

    <a href="cart.html" class="cart">Cart (<span id="cart-count">0</span>)</a>

</div>

</div>

<!-- Product Grid --&gt;

&lt;div class="product-container" id="product-list"&gt;

    <!-- Book 1 --&gt;

    &lt;div class="product"&gt;

        &lt;img src="bhagavad_gita.jpg" alt="Bhagavad Gita"&gt;

        &lt;h3&gt;Bhagavad Gita&lt;/h3&gt;

        &lt;p&gt;A sacred scripture offering guidance on life, duty, and spirituality.&lt;/p&gt;

        &lt;p&gt;₹299&lt;/p&gt;

        &lt;button onclick="addToCart('Bhagavad Gita', 299)"&gt;Add to Cart&lt;/button&gt;

    &lt;/div&gt;

    <!-- Book 2 --&gt;

    &lt;div class="product"&gt;

        &lt;img src="percy_jackson.jpg" alt="Percy Jackson"&gt;

        &lt;h3&gt;Percy Jackson&lt;/h3&gt;

        &lt;p&gt;An adventurous series that follows Percy, a demigod, through mythical challenges.&lt;/p&gt;

        &lt;p&gt;₹499&lt;/p&gt;

        &lt;button onclick="addToCart('Percy Jackson', 499)"&gt;Add to Cart&lt;/button&gt;

    &lt;/div&gt;

    <!-- Book 3 --&gt;</pre>
```

```
<div class="product">  
      
    <h3>It Ends With Us</h3>  
    <p>A heart-wrenching tale of love, strength, and sacrifice.</p>  
    <p>₹399</p>  
    <button onclick="addToCart('It Ends With Us', 399)">Add to Cart</button>  
</div>
```

```
<!-- Book 4 -->  
  
<div class="product">  
      
    <h3>Harry Potter</h3>  
    <p>A magical journey that follows the life of a young wizard fighting evil.</p>  
    <p>₹599</p>  
    <button onclick="addToCart('Harry Potter', 599)">Add to Cart</button>  
</div>
```

```
<!-- Book 5 -->  
  
<div class="product">  
      
    <h3>Atomic Habits</h3>  
    <p>An insightful guide on building positive habits and breaking bad ones.</p>  
    <p>₹450</p>  
    <button onclick="addToCart('Atomic Habits', 450)">Add to Cart</button>  
</div>
```

```
<!-- Book 6 -->  
  
<div class="product">  
      
    <h3>Rich Dad Poor Dad</h3>
```

```
<p>A financial classic teaching the secrets of wealth creation.</p>
<p>₹350</p>
<button onclick="addToCart('Rich Dad Poor Dad', 350)">Add to Cart</button>
</div>

<!-- Book 7 -->
<div class="product">
  
  <h3>The Alchemist</h3>
  <p>A philosophical tale about pursuing one's dreams and destiny.</p>
  <p>₹299</p>
  <button onclick="addToCart('The Alchemist', 299)">Add to Cart</button>
</div>

<!-- Book 8 -->
<div class="product">
  
  <h3>Sapiens</h3>
  <p>A gripping journey through the history of humankind.</p>
  <p>₹599</p>
  <button onclick="addToCart('Sapiens', 599)">Add to Cart</button>
</div>

<!-- Book 9 -->
<div class="product">
  
  <h3>Wings of Fire</h3>
  <p>An inspiring autobiography of Dr. A.P.J. Abdul Kalam.</p>
  <p>₹320</p>
  <button onclick="addToCart('Wings of Fire', 320)">Add to Cart</button>
```

```
</div>

<!-- Book 10 (The Subtle Art of Not Giving a F***) -->

<div class="product">
    
    <h3>The Subtle Art of Not Giving a F***</h3>
    <p>A counterintuitive guide to living a good life by embracing honesty and resilience.</p>
    <p>₹399</p>
    <button onclick="addToCart('The Subtle Art of Not Giving a F***', 399)">Add to Cart</button>
</div>

</div>

<!-- Footer -->

<div class="footer">
    © 2024 Crossword Bookstore. All Rights Reserved.
</div>

<!-- Copyright Navigation -->

<div class="copyright-nav">
    <a href="privacy.html">Privacy Policy</a> |
    <a href="terms.html">Terms of Use</a> |
    <a href="faq.html">FAQ</a>
</div>

<script>
    // JavaScript for Cart Functionality

    let cartCount = 0;
    let cartItems = [];

    function addToCart(name, price) {
```

```

cartCount++;

document.getElementById("cart-count").innerText = cartCount;

// Add item to local storage

let item = { name: name, price: price, quantity: 1 };

let existingItems = JSON.parse(localStorage.getItem("cart")) || [];

let existingItem = existingItems.find(item => item.name === name);

if (existingItem) {

    existingItem.quantity += 1;

} else {

    existingItems.push(item);

}

localStorage.setItem("cart", JSON.stringify(existingItems));

alert(`${name} added to cart!`);

}

</script>

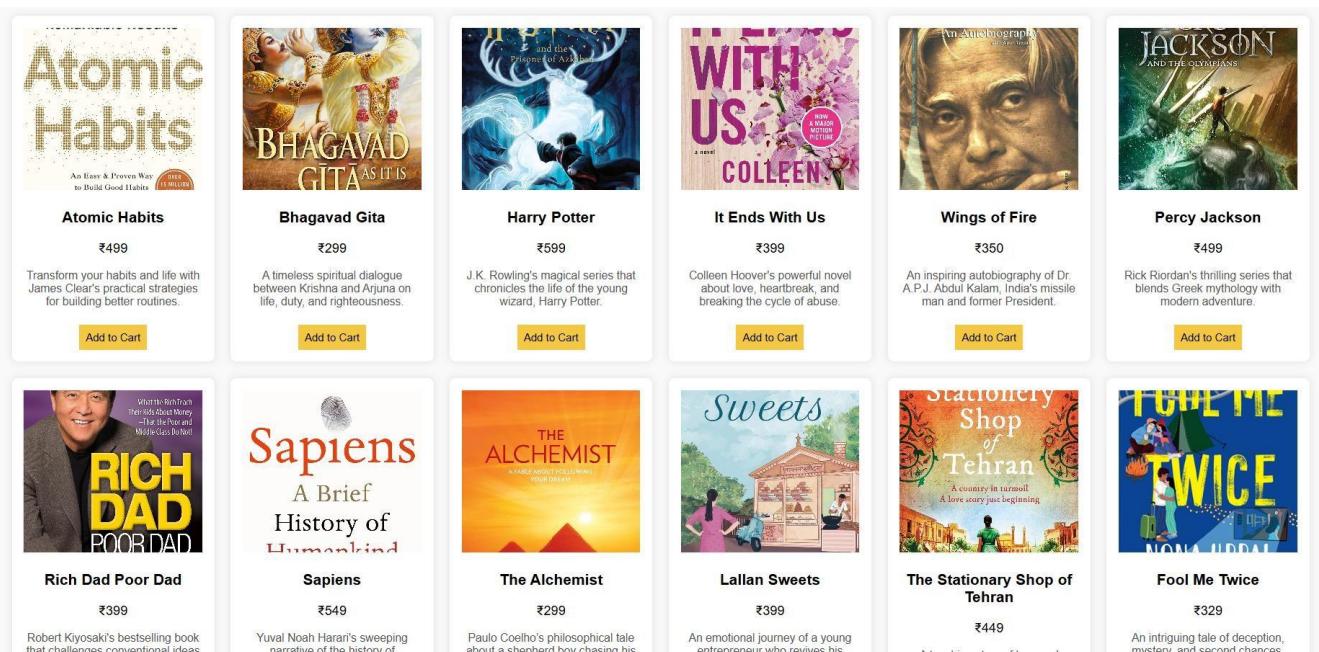
```

</body>

</html>

Output:

B. menu/product page output:

**Code:**

C. cart page:

code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Cart - Crossword Bookstore</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f9f9f9;

margin: 0;

}

/* Header Section */

.header {

background-color: #f4c842;

```
padding: 15px 20px;  
text-align: center;  
font-size: 24px;  
font-weight: bold;  
}
```

```
.navbar {  
background-color: #333;  
display: flex;  
justify-content: space-between;  
padding: 10px 20px;  
}
```

```
.navbar a {  
color: white;  
text-decoration: none;  
padding: 10px 15px;  
}
```

```
.navbar a:hover {  
background-color: #f4c842;  
color: black;  
}
```

```
.cart-container {  
width: 60%;  
margin: 20px auto;  
background: white;  
padding: 20px;  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
border-radius: 8px;  
}  
  




    width: 100%;  
    border-collapse: collapse;  
}  
  
th,  
td {  
    padding: 10px;  
    text-align: center;  
    border-bottom: 1px solid #ddd;  
}  
  
.footer {  
    text-align: center;  
    background: #333;  
    color: white;  
    padding: 10px;  
    margin-top: 20px;  
}  
  
.checkout-btn {  
    background: #f4c842;  
    padding: 10px 20px;  
    border: none;  
    cursor: pointer;  
    margin-top: 20px;  
}
```

```
.checkout-btn:hover {  
    background-color: #ffae00;  
}  
  
.cart-count {  
    background-color: red;  
    color: white;  
    border-radius: 50%;  
    padding: 5px 8px;  
    font-size: 12px;  
    margin-left: 5px;  
}  
</style>  
</head>  
  
<body>  
  
<!-- Header -->  
<div class="header">Crossword Bookstore</div>  
  
<!-- Navigation Bar -->  
<div class="navbar">  
    <a href="index.html">Home</a>  
    <a href="books.html">Books</a>  
    <a href="cart.html" class="cart"> Cart <span id="cart-count" class="cart-count">0</span></a>  
</div>  
  
<!-- Cart Section -->  
<div class="cart-container">
```

```

<h2>Your Cart</h2>

<table>

  <thead>
    <tr>
      <th>Book</th>
      <th>Price</th>
      <th>Quantity</th>
      <th>Total</th>
      <th>Action</th>
    </tr>
  </thead>

  <tbody id="cart-items"></tbody>

</table>

<button class="checkout-btn" onclick="checkout()">Checkout</button>

</div>

<!-- Footer -->

<div class="footer">&copy; 2024 Crossword Bookstore. All Rights Reserved.</div>

<script>

  // Load Cart Items from Local Storage

  function loadCartItems() {

    let cartItems = JSON.parse(localStorage.getItem("cart")) || [];

    let cartTable = document.getElementById("cart-items");

    cartTable.innerHTML = "";

    let totalAmount = 0;

    if (cartItems.length === 0) {
      cartTable.innerHTML = "<tr><td colspan='5'>Your cart is empty!</td></tr>";
      updateCartCount();
    }

    cartItems.forEach(item => {
      let rowHTML = "<tr><td>" + item.title + "</td><td>" + item.price + "</td><td>" + item.quantity + "</td><td>" + item.total + "</td><td>";  

      rowHTML += "<a href="#" data-item-id=" + item.id + ">Remove</a></td></tr>";  

      cartTable.innerHTML += rowHTML;
    });

    calculateTotal();
  }

  function calculateTotal() {
    let totalAmount = 0;
    let cartTable = document.getElementById("cart-items");
    let rows = cartTable.rows;
    for (let i = 1; i < rows.length; i++) {
      let cells = rows[i].cells;
      let totalCell = cells[4];
      let totalValue = totalCell.textContent;
      totalAmount += parseFloat(totalValue);
    }
    document.getElementById("total-amount").innerHTML = "Total Amount: " + totalAmount;
  }

  function updateCartCount() {
    let cartCount = document.getElementById("cart-count");
    let cartItems = JSON.parse(localStorage.getItem("cart"));
    if (cartItems) {
      cartCount.textContent = cartItems.length;
    } else {
      cartCount.textContent = 0;
    }
  }

  function checkout() {
    alert("Checkout successful!");
  }

</script>

```

```

    return;
}

cartItems.forEach((item, index) => {
    let total = item.price * item.quantity;
    totalAmount += total;

    cartTable.innerHTML += `
        <tr>
            <td>${item.name}</td>
            <td>₹${item.price}</td>
            <td>
                <button onclick="updateQuantity(${index}, -1)">-</button>
                ${item.quantity}
                <button onclick="updateQuantity(${index}, 1)">+</button>
            </td>
            <td>₹${total}</td>
            <td><button onclick="removeItem(${index})">Remove</button></td>
        </tr>
    `;
});

cartTable.innerHTML += `
    <tr>
        <td colspan="3"><strong>Total</strong></td>
        <td colspan="2"><strong>₹${totalAmount}</strong></td>
    </tr>
`;

updateCartCount(); // Update count after reload

```

```
}

// Update Cart Count

function updateCartCount() {

    let cartItems = JSON.parse(localStorage.getItem("cart")) || [];

    let itemCount = cartItems.reduce((sum, item) => sum + item.quantity, 0);

    document.getElementById("cart-count").innerText = itemCount;

}

// Remove Item from Cart

function removeItem(index) {

    let cartItems = JSON.parse(localStorage.getItem("cart")) || [];

    cartItems.splice(index, 1);

    localStorage.setItem("cart", JSON.stringify(cartItems));

    loadCartItems();

}

// Update Item Quantity

function updateQuantity(index, change) {

    let cartItems = JSON.parse(localStorage.getItem("cart")) || [];

    if (cartItems[index].quantity + change > 0) {

        cartItems[index].quantity += change;

    } else {

        cartItems.splice(index, 1);

    }

    localStorage.setItem("cart", JSON.stringify(cartItems));

    loadCartItems();

}

// Checkout and Clear Cart
```

```

function checkout() {
    if (confirm("Are you sure you want to proceed with the checkout?")) {
        alert("Thank you for your purchase! 🎉");
        localStorage.removeItem("cart");
        loadCartItems();
    }
}

// Load Cart Items on Page Load
window.onload = function () {
    loadCartItems();
};

</script>

</body>

</html>

```

Output:

C. cart page output:

Book	Price	Quantity	Total	Action
It Ends With Us	₹399	<input type="button" value="-"/> 1 <input type="button" value="+"/>	₹399	<input type="button" value="Remove"/>
Sapiens	₹599	<input type="button" value="-"/> 3 <input type="button" value="+"/>	₹1797	<input type="button" value="Remove"/>
Total			₹2196	

Checkout

Code:

D. about us page:

code:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>About Us - Crossword Bookstore</title>

<style>

/* Basic Styling */

body {

    font-family: Arial, sans-serif;

    background: #f9f9f9;

    margin: 0;

    padding: 0;

}

/* Header */

.header {

    background-color: #f4c842;

    padding: 15px;

    text-align: center;

    font-size: 24px;

    font-weight: bold;

}

/* Navigation Bar */

.navbar {

    background-color: #333;

    overflow: hidden;

    display: flex;

    justify-content: space-between;

}
```

```
padding: 10px 20px;  
}
```

```
.navbar a {  
    color: white;  
    text-decoration: none;  
    padding: 10px 15px;  
}
```

```
.navbar a:hover {  
    background-color: #f4c842;  
    color: black;  
}
```

```
/* About Section */
```

```
.about-container {  
    max-width: 900px;  
    margin: 20px auto;  
    background: white;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    text-align: center;  
}
```

```
.about-container h2 {  
    color: #333;  
}
```

```
.about-container p {
```

```
font-size: 16px;  
line-height: 1.6;  
color: #555;  
}  
  
/* Customer Reviews */  
  
.reviews {  
max-width: 800px;  
margin: 20px auto;  
background: white;  
padding: 20px;  
border-radius: 8px;  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
.review {  
padding: 10px;  
border-bottom: 1px solid #ddd;  
}  
  
.review:last-child {  
border-bottom: none;  
}  
  
.review strong {  
color: #333;  
}  
  
/* Footer */  
.footer {
```

```
text-align: center;  
background: #333;  
color: white;  
padding: 10px;  
margin-top: 20px;  
}  
</style>  
</head>  
<body>  
  
<!-- Header -->  
<div class="header">  
    About Us - Crossword Bookstore  
</div>  
  
<!-- Navigation Bar -->  
<div class="navbar">  
    <div>  
        <a href="index.html">Home</a>  
        <a href="books.html">Books</a>  
        <a href="about.html">About</a>  
        <a href="contact.html">Contact</a>  
    </div>  
    <div>  
        <a href="login.html">Login</a>  
        <a href="register.html">Register</a>  
        <a href="cart.html" class="cart">Cart (<span id="cart-count">0</span>)</a>  
    </div>  
</div>
```

<!-- About Section -->

<div class="about-container">

<h2>Welcome to Crossword Bookstore</h2>

<p>At Crossword, we believe in the magic of books and the joy of reading. Established in 1980, we have been a hub for book lovers, providing an extensive collection of fiction, non-fiction, academic, and children's books.</p>

<p>Our mission is to make books accessible and enjoyable for everyone, whether you prefer the latest bestsellers or timeless classics.</p>

<p>Visit our stores or shop online to explore an incredible world of books, curated just for you.</p>

</div>

<!-- Customer Reviews -->

<div class="reviews">

<h2>What Our Customers Say</h2>

<div class="review">

Rohan Sharma

<p>"A wonderful bookstore with a vast collection. The online shopping experience is seamless!"</p>

</div>

<div class="review">

Neha Verma

<p>"Crossword has always been my go-to place for finding rare books. Highly recommended!"</p>

</div>

<div class="review">

Amit Kapoor

<p>"Great customer service and a user-friendly website. The book delivery was super fast!"</p>

</div>

</div>

<!-- Footer -->

<div class="footer">

© 2024 Crossword Bookstore. All Rights Reserved.

</div>

```
</body>
```

```
</html>
```

Output:

D. about us page output:

About Us - Crossword Bookstore

Welcome to Crossword Bookstore

At Crossword, we believe in the magic of books and the joy of reading. Established in 1980, we have been a hub for book lovers, providing an extensive collection of fiction, non-fiction, academic, and children's books.

Our mission is to make books accessible and enjoyable for everyone, whether you prefer the latest bestsellers or timeless classics.

Visit our stores or shop online to explore an incredible world of books, curated just for you.

What Our Customers Say

Rohan Sharma
"A wonderful bookstore with a vast collection. The online shopping experience is seamless!"

Neha Verma
"Crossword has always been my go-to place for finding rare books. Highly recommended!"

Amit Kapoor
"Great customer service and a user-friendly website. The book delivery was super fast!"

© 2024 Crossword Bookstore. All Rights Reserved.

Code:

E. contact us page:

code:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Second hand consoles</title>
```

```
<style>
```

```
* {
```

```
    margin: 0;
```

```
    padding: 0;
```

```
    box-sizing: border-box;
```

```
}
```

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
}
```

```
nav {  
    background-color: #333;  
    padding: 1rem;  
}
```

```
nav ul {  
    display: flex;  
    justify-content: center;  
    list-style: none;  
}
```

```
nav ul li {  
    margin: 0 15px;  
}
```

```
nav ul li a {  
    color: white;  
    text-decoration: none;  
    font-weight: bold;  
}
```

```
.content {  
    background-color: white;
```

```

padding: 20px;
text-align: center;
border-radius: 8px;
}

</style>

</head>

<body>

<nav>

<ul>

<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\homepage.html">Home</a></li>
<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\login.html">Login</a></li>
<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\registration.html">Registration</a></li>
<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\contact_us.html">Contact Us</a></li>
<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\about_us.html">About Us</a></li>
<li><a href="C:\\Users\\ojas\\OneDrive\\Desktop\\WPL\\WPL TA-1\\product.html">Products</a></li>

</ul>

</nav>

<div class="content">
<h2>Contact Us</h2>

</div>

</body>

</html>

```

Output:

E. contact us page output:

Contact Us

Your Full Name

Your Email Address

Subject  pranitapp1587@gmail.com
Pranita Potghan

Write your message...  Manage addresses...

Send Message

Code:

F. registration page:

code:

Crossword Bookstore

[Home](#) [Books](#) [About](#) [Contact](#)

[Login](#) [Register](#) [Cart](#)

Create an Account

Full Name:

Email:

Password:

Confirm Password:

Register

© 2024 Crossword Bookstore. All Rights Reserved.

Output:

F. registration page output:

Create an Account

Full Name:

Username:

Email:

Phone Number:

Date of Birth: dd-mm-yyyy

Gender:

Address:

Preferred Genre:

Password: *

Confirm Password: *

Subscribe to newsletter

Register

Code:

G. login page:

code:

Crossword Bookstore

[Home](#) [Books](#) [About](#) [Contact](#)

[Login](#) [Register](#) [Cart](#)

Login to Crossword

Email:

Password:

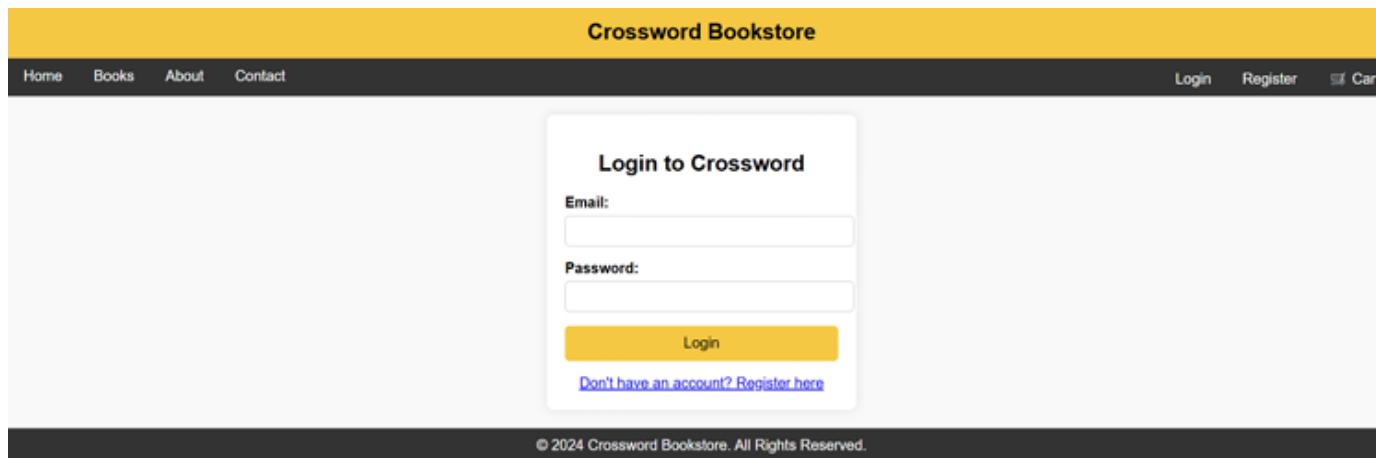
Login

[Don't have an account? Register here](#)

© 2024 Crossword Bookstore. All Rights Reserved.

Output:

G. login page output:



Conclusion

The Crossword Bookstore website combines practical e-commerce features with a sustainability-driven mission. The use of structured web design, user-friendly forms, and clear product categorization ensures a seamless experience for both shoppers and admins. With further backend integration, it can evolve into a fully operational online business supporting green technology use and community engagement.

Experiment No. 3

Problem Statement:

Enhance the layout of the Crossword Bookstore website's home page using CSS Grid. Use CSS Grid to organize the featured book sections and display book categories (Fiction, Non-fiction, Children's Books, Academic, etc.) in a structured, visually appealing format with appropriate headings, spacing, separators, images, descriptions, and prices.

Theory:

CSS Theory for Enhancing the Layout of a Bookstore Website Using CSS Grid

Introduction to CSS Grid

CSS Grid Layout is a powerful 2-dimensional layout system that allows developers to create complex, responsive web page layouts. Unlike Flexbox, which works in one direction (either row or column), CSS Grid can manage both rows and columns simultaneously, making it ideal for a homepage and book listing layout.

For an e-commerce bookstore website like Crossword, CSS Grid makes it easier to:

- Showcase a variety of books cleanly
- Align text, images, and buttons uniformly
- Build mobile-friendly layouts without writing complex media queries

Why Use CSS Grid for the Crossword Bookstore Website?

In an online bookstore, users expect a neat, easy-to-browse interface for different categories and featured selections. CSS Grid helps in:

- Structuring book cards in a neat grid (like 3x3, 4x4)
- Creating visually distinct sections such as “Best Sellers,” “New Arrivals,” or “Editor’s Picks”
- Aligning book images, descriptions, and prices for a uniform look
- Ensuring responsive behavior across all device sizes

1. Home Page Layout with CSS Grid

The homepage can be divided into structured sections using CSS Grid:

- Navigation Bar: Full-width header with logo and links
- Hero Section: A large banner image or featured deal
- Book Highlights: A 3-column grid of bestsellers or recommended books
- Category Grid: Grid blocks for categories like Fiction, Non-fiction, etc.
- Footer: Contact information, store locator, newsletter signup

Benefits:

- Easy to define page sections
- Precise control over alignment and spacing
- Reduces need for external libraries or media queries

2. Book/Product Listing Page Layout Using CSS Grid

Book items are displayed in categories such as:

- Fiction
- Non-fiction
- Children’s Books
- Academic & Competitive
- Stationery & Gifts

Each book is presented as a card arranged in a responsive grid.

CSS Grid Code Example:

css

CopyEdit

```
.books-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 30px;
    padding: 20px;
}
```

Each book card includes:

- Book Cover Image
- Title
- Author
- Description or tagline
- Price (highlighted)
- “Add to Cart” / “Buy Now” Button

Additional Styling Concepts:

- Category Headings: Use bold fonts, underlines, or background highlights to separate sections.
- Separators: Use horizontal lines to divide different categories.
- Hover Effects: Highlight the card, elevate it slightly, or change button color on hover.
- Responsive Design: Using auto-fit and minmax(), grid auto-adjusts for mobile, tablet, and desktop without writing breakpoints.

Mobile Responsiveness with CSS Grid

CSS Grid automatically adapts to smaller screens:

- Books stack into one or two columns
- Buttons and text remain legible
- Improves user experience without manual adjustment

Summary:

CSS Grid is a powerful layout tool ideal for structuring an online bookstore like Crossword. It ensures consistency, enhances aesthetics, and improves user interaction—especially for displaying books across categories and devices.

Code:

```
<!DOCTYPE html>  
<html>  
<head>
```

```
    <title>Second Hand Consoles</title>
```

```
<style>
```

```
    .image-container {  
        display: flex;  
        justify-content: center;  
        gap: 20px;  
        padding: 20px;  
    }
```

```
.image-container div {  
    text-align: center;  
    width: 40%;  
}
```

```
.image-container img {  
    width: 100%;  
    height: auto;  
    aspect-ratio: 16 / 9;  
    border-radius: 8px;  
    border: 3px solid #333;  
}
```

```
* {  
    margin: 0;  
    padding: 0;
```

```
  box-sizing: border-box;  
}  
  
  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f4f4f4;  
}  
  
  
nav {  
  background-color: #333;  
  padding: 1rem;  
}  
  
  
nav ul {  
  display: flex;  
  justify-content: space-between;  
  list-style: none;  
  align-items: center;  
}  
  
  
.nav-left, .nav-right {  
  display: flex;  
  gap: 15px;  
}  
  
  
nav ul li a {  
  color: white;  
  text-decoration: none;  
  font-weight: bold;  
}
```

```
.content {  
    background-color: white;  
    padding: 20px;  
    text-align: center;  
    border-radius: 8px;  
    margin: 20px;  
}  
  
footer {  
    background-color: #333;  
    color: white;  
    text-align: center;  
    padding: 10px;  
}  
  
.logout-btn {  
    background: none;  
    border: none;  
    color: white;  
    font-weight: bold;  
    cursor: pointer;  
}  
</style>  
</head>  
<body>  
<nav>  
<ul>  
<div class="nav-left">  
<li><a href="homepage.html">Home</a></li>
```

```
<li><a href="contact_us.html">Contact Us</a></li>
<li><a href="about_us.html">About Us</a></li>
<li><a href="product.html">Products</a></li>
</div>

<div class="nav-right" id="authLinks">
    <!-- This will be filled by JavaScript -->
</div>

</ul>

</nav>

<div class="content">
    <h2>Welcome to Our Website!</h2>
    <h3>We sell premium grade second-hand gaming consoles!</h3>
</div>

<div class="content">
    <h3>Products</h3>
    <p>Check out our latest products!</p>
    <div class="image-container">
        <div>
            
            <p>Price: $350</p>
        </div>
        <div>
            
            <p>Price: $500</p>
        </div>
    </div>
</div>
</div>
```

```
<footer>

    <p>&copy; 2025 Tsunami. All Rights Reserved.</p>

</footer>

<script>

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

const userDetails = JSON.parse(localStorage.getItem('userDetails'));

if (userDetails && userDetails.username) {

    authLinks.innerHTML = `

        <li style="color: white;">Welcome, ${userDetails.username}</li>
        <li><button class="logout-btn" onclick="logout()">Logout</button></li>
    `;

} else {

    authLinks.innerHTML = `

        <li><a href="login.html">Login</a></li>
        <li><a href="registration.html">Registration</a></li>
    `;

}

function logout() {

    localStorage.removeItem('userDetails');

    window.location.reload();

}

</script>

</body>

</html>
```

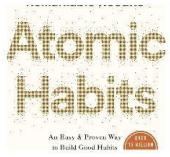
Output

Crossword Bookstore

Login Register Cart (0)

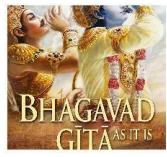
Book Categories

Fiction Non-Fiction Science Technology Children



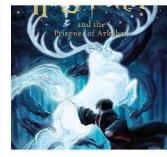
Atomic Habits
₹499
Transform your habits and life with James Clear's practical strategies for building better routines.

[Add to Cart](#)



Bhagavad Gita
₹299
A timeless spiritual dialogue between Krishna and Arjuna on life, duty, and righteousness.

[Add to Cart](#)



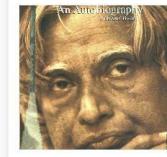
Harry Potter
₹599
J.K. Rowling's magical series that chronicles the life of the young wizard, Harry Potter.

[Add to Cart](#)



It Ends With Us
₹399
Colleen Hoover's powerful novel about love, heartbreak, and breaking the cycle of abuse.

[Add to Cart](#)



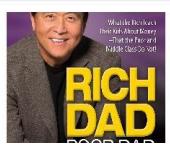
Wings of Fire
₹350
An inspiring autobiography of Dr. A.P.J. Abdul Kalam, India's missile man and former President.

[Add to Cart](#)

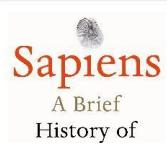


Percy Jackson
₹499
Rick Riordan's thrilling series that blends Greek mythology with modern adventure.

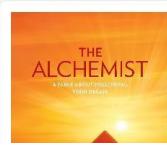
[Add to Cart](#)



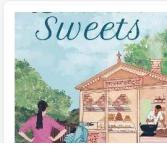
RICH DAD POOR DAD



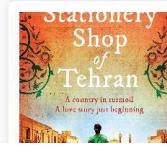
Sapiens
A Brief History of



THE ALCHEMIST



Sweets



Stationery Shop of Tehran



I Want You Twice

Conclusion

CSS Grid is a powerful tool for building modern, responsive, and structured websites. In the case of a Crossword Bookstore, **CSS Grid simplifies complex layout structures**, enhances visual clarity, and provides a clean, user-friendly interface.

By using CSS Grid:

- The **home page** becomes visually appealing and sectioned clearly for better navigation.
- The **product menu** is organized and readable, allowing users to quickly explore items.
- The site adapts beautifully across devices without writing dozens of media queries.
- Layout and spacing between elements remain consistent, ensuring a **professional and polished appearance**.

In summary, CSS Grid plays a crucial role in improving the **aesthetic appeal, usability, and responsiveness** of your e-commerce platform—making it both functional and engaging for your users.

Experiment No.4

Problem Statement

CSS

- A. Enhance the cart page to make it user-friendly and visually appealing. Style the cart items with appropriate margins, paddings, and input field styles to provide a seamless shopping experience.
- B. Enhance and style the about us page with appropriate margins, paddings, and input field styles.
- C. Enhance and style the contact page to make it user-friendly and visually appealing. Style the contact form with appropriate margins, paddings, and input field styles.
- D. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles.
- E. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles.

CSS Theory: Enhancing and Styling Key Pages in a Second-Hand Gaming Console Website

1. Why CSS Styling Matters in E-commerce Websites

When users land on your site, the first thing they notice is how it looks and feels. Clean, well-structured, and visually appealing interfaces significantly improve user trust, navigation, and engagement.

Whether it's a cart, contact form, or registration page, proper styling with CSS margins, paddings, spacing, input design, and color schemes:

- Makes the content easier to read
- Provides a sense of structure and flow
- Enhances accessibility and user experience (UX)
- Encourages actions like completing a purchase, registering, or submitting a form

Page-wise CSS Styling Theory

1. Cart Page

The cart page is where users review their selected products before checking out, so it needs to be clear, clean, and action-oriented.

Key Styling Techniques:

- Add padding around each cart item for separation
- Use margins to space out product name, quantity input, price, and "remove" button

- Style input fields (quantity, update buttons) with soft borders and enough clickable area
- Highlight the total amount with a bold font and distinct background
- Use consistent font sizes and spacing for price breakdown and tax summaries

Result: A structured layout that minimizes confusion and maximizes conversion.

2. About Us Page

This page tells your brand's story, builds credibility, and helps users connect emotionally with your mission.

Key Styling Techniques:

- Use line height, padding, and justified alignment for readability
- Add white space between sections like “Our Story,” “Our Mission,” and “Our Team”
- Use subtle background colors or separator lines for each section
- Style images (e.g., founders/team) with rounded borders and spacing
- Highlight values or mission using boxes, grids, or quote-styling

Result: A professional and inviting presentation that builds trust.

3. Contact Page

Your contact page should make it effortless for users to reach out for support, queries, or feedback.

Key Styling Techniques:

- Style input fields with equal width, padding, and soft border-radius
- Use margin-bottom to separate form fields
- Provide visual feedback on focus (e.g., border color change)
- Add submit button styling for emphasis (hover effects, background color)
- Layout the form centrally with balanced padding on all sides

Result: A visually appealing and accessible form that encourages engagement.

4. Admin/User Registration Form

This form is critical for onboarding new users/admins, and should feel secure and easy to use.

Key Styling Techniques:

- Organize input fields in logical groups (e.g., personal info, password)
- Add labels and placeholders for clarity
- Use consistent input sizes, padding, and spacing
- Style the form card with shadows, rounded borders, and a light background

- Include hover effects for buttons and inline validation messages

Result: An intuitive form that encourages complete and accurate registration.

5. Admin/User Login Form

Login forms should be quick to use, visually balanced, and provide immediate clarity for mistakes.

Key Styling Techniques:

- Center the login form on the page
- Add sufficient padding inside the form container
- Style input fields with enough spacing and highlight on focus
- Use subtle background colors or semi-transparent overlays
- Style error messages in red and success in green
- Provide clear visual hierarchy (larger font for "Login", smaller for "Forgot Password?")

Result: A clean and efficient login interface that builds user confidence.

Code:

cart page:

code:

```
<!DOCTYPE html>

<html>
<head>
<title>Shopping Cart</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
}
.cart-container {
    background-color: white;
    padding: 20px;
    margin: 50px auto;
    width: 50%;
}
```

```
border-radius: 8px;  
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
}  
  
.cart-items {  
list-style: none;  
padding: 0;  
}  
  
.cart-items li {  
display: flex;  
justify-content: space-between;  
padding: 10px;  
border-bottom: 1px solid #ddd;  
}  
  
.remove-item {  
background-color: red;  
color: white;  
border: none;  
padding: 5px 10px;  
cursor: pointer;  
border-radius: 5px;  
}  
  
.remove-item:hover {  
background-color: darkred;  
}  
  
.checkout-btn, .home-btn {  
display: block;  
margin: 10px auto;  
padding: 10px 15px;  
border: none;  
color: white;  
cursor: pointer;
```

```

border-radius: 5px;
width: 200px;
}

.checkout-btn {
background-color: #007bff;
}

.checkout-btn:hover {
background-color: #0056b3;
}

.home-btn {
background-color: #28a745;
}

.home-btn:hover {
background-color: #218838;
}


```

</style>

</head>

<body>

<div class="cart-container">

<h2>Your Shopping Cart</h2>

<ul id="cart-items" class="cart-items">

<p id="total-price">Total Price: \$0</p>

<button class="checkout-btn" onclick="checkout()">Proceed to Checkout</button>

<button class="home-btn" onclick="goHome()">Return to Home</button>

</div>

<script>

```

let cartItems = JSON.parse(localStorage.getItem('cart')) || [];
let totalPrice = cartItems.reduce((sum, item) => sum + item.price, 0);

function renderCart() {

```

```

let cartList = document.getElementById('cart-items');

let totalPriceElement = document.getElementById('total-price');

cartList.innerHTML = "";

cartItems.forEach((item, index) => {
    let li = document.createElement('li');
    li.innerHTML = `${item.name} - ${item.price} <button class="remove-item" onclick="removeItem(${index})">Remove</button>';
    cartList.appendChild(li);
});

totalPriceElement.innerText = `Total Price: ${totalPrice}`;

}

function removeItem(index) {
    totalPrice -= cartItems[index].price;
    cartItems.splice(index, 1);
    localStorage.setItem('cart', JSON.stringify(cartItems));
    renderCart();
}

function checkout() {
    alert("Proceeding to checkout...");
}

function goHome() {
    window.location.href = "homepage.html"; // Redirect to the home page
}

renderCart();

</script>

</body>

</html>

```

Output:

cart page output:

Your Cart

| Book | Price | Quantity | Total | Action |
|---|-------|---|-------|---------------------------------------|
| It Ends With Us | ₹399 | <input type="button" value="-"/> 1 <input type="button" value="+"/> | ₹399 | <input type="button" value="Remove"/> |
| Sapiens | ₹599 | <input type="button" value="-"/> 1 <input type="button" value="+"/> | ₹599 | <input type="button" value="Remove"/> |
| Total | | | | ₹998 |
| <input type="button" value="Checkout"/> | | | | |

© 2024 Crossword Bookstore. All Rights Reserved.

Code:

registration page:

code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Register</title>

    <style>

        * {

            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
        }

        body {

            background-color: #f4f4f4;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        .container {

            background-color: white;
            padding: 25px;
            border-radius: 10px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
        }
    </style>
</head>

<body>

    <div class="container">

        <h1>Welcome to Registration Page</h1>
        <p>Please fill the form below to register</p>
        <form>
            <label>Name:</label>
            <input type="text" name="name" required>
            <label>Email:</label>
            <input type="email" name="email" required>
            <label>Password:</label>
            <input type="password" name="password" required>
            <label>Confirm Password:</label>
            <input type="password" name="confirm_password" required>
            <button type="submit">Register</button>
        </form>
    </div>
</body>
```

```
width: 350px;  
text-align: center;  
}
```

```
h2 {  
margin-bottom: 20px;  
}
```

```
.form-group {  
position: relative;  
margin-bottom: 15px;  
text-align: left;  
}
```

```
label {  
font-weight: bold;  
}
```

```
input {  
width: 100%;  
padding: 10px;  
margin-top: 5px;  
border: 1px solid #ccc;  
border-radius: 6px;  
font-size: 14px;  
outline: none;  
transition: border 0.3s ease-in-out;  
}
```

```
input:focus {
```

```
border-color: #007bff;  
}
```

```
.password-container {  
    position: relative;  
}  
  
}
```

```
.toggle-password {  
    position: absolute;  
    right: 10px;  
    top: 50%;  
    transform: translateY(-50%);  
    cursor: pointer;  
    font-size: 16px;  
    color: #777;  
}  
  
}
```

```
.valid {  
    border: 2px solid green !important;  
}  
  
}
```

```
.invalid {  
    border: 2px solid red !important;  
}  
  
}
```

```
.error {  
    color: red;  
    font-size: 12px;  
    margin-top: 3px;  
    height: 14px;
```

```
}
```

```
button {  
    background-color: #007bff;  
    color: white;  
    padding: 12px;  
    border: none;  
    border-radius: 6px;  
    cursor: pointer;  
    width: 100%;  
    margin-top: 10px;  
    font-size: 16px;  
    opacity: 0.5;  
    transition: background-color 0.3s ease-in-out, opacity 0.3s ease-in-out;  
}  
  
button:hover {  
    background-color: #0056b3;  
}
```

```
button:enabled {  
    opacity: 1;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">  
    <h2>Register</h2>  
    <form id="registrationForm">
```

```

<div class="form-group">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <p class="error" id="usernameError"></p>
</div>

<div class="form-group password-container">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <span class="toggle-password" onclick="togglePassword('password', this)">≡|</span>
    <p class="error" id="passwordError"></p>
</div>

<div class="form-group password-container">
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword" required>
    <span class="toggle-password" onclick="togglePassword('confirmPassword', this)">≡|</span>
    <p class="error" id="confirmPasswordError"></p>
</div>

<button type="submit" id="registerButton" disabled>Register</button>
</form>

<p id="successMessage" style="color: green; display: none;">Registration successful! Redirecting...</p>
</div>

<script>
    const form = document.getElementById("registrationForm");
    const usernameField = document.getElementById("username");
    const passwordField = document.getElementById("password");
    const confirmPasswordField = document.getElementById("confirmPassword");
    const registerButton = document.getElementById("registerButton");

    const validationRules = {

```

```
username: {  
    regex: /^[a-zA-Z0-9]{5,}$/,  
    errorMsg: "Username must be at least 5 characters and contain only letters and numbers."  
},  
  
password: {  
    regex: /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{6,16}$/,  
    errorMsg: "Password must be 6-16 characters, with uppercase, lowercase, number, and special character."  
}  
};  
  
function validateField(field, rule) {  
    const value = field.value.trim();  
    const errorElement = document.getElementById(field.id + "Error");  
  
    if (rule.regex.test(value)) {  
        field.classList.add("valid");  
        field.classList.remove("invalid");  
        errorElement.textContent = "";  
        return true;  
    } else {  
        field.classList.add("invalid");  
        field.classList.remove("valid");  
        errorElement.textContent = rule.errorMsg;  
        return false;  
    }  
}  
  
function validateConfirmPassword() {  
    const password = passwordField.value;  
    const confirmPassword = confirmPasswordField.value;
```

```
const errorElement = document.getElementById("confirmPasswordError");

if (confirmPassword === password && confirmPassword !== "") {
    confirmPasswordField.classList.add("valid");
    confirmPasswordField.classList.remove("invalid");
    errorElement.textContent = "";
    return true;
} else {
    confirmPasswordField.classList.add("invalid");
    confirmPasswordField.classList.remove("valid");
    errorElement.textContent = "Passwords do not match.";
    return false;
}

}

function validateForm() {
    const isUsernameValid = validateField(usernameField, validationRules.username);
    const isPasswordValid = validateField(passwordField, validationRules.password);
    const isConfirmPasswordValid = validateConfirmPassword();

    registerButton.disabled = !(isUsernameValid && isPasswordValid && isConfirmPasswordValid);
}

function togglePassword(fieldId, icon) {
    const field = document.getElementById(fieldId);
    if (field.type === "password") {
        field.type = "text";
        icon.textContent = 'visibility' // Hide Password Icon
    } else {
        field.type = "password";
    }
}
```

```
icon.textContent = "eye" // Show Password Icon
}

}

usernameField.addEventListener("input", validateForm);

passwordField.addEventListener("input", validateForm);

confirmPasswordField.addEventListener("input", validateForm);

form.addEventListener("submit", function (event) {
    event.preventDefault();
    document.getElementById("successMessage").style.display = "block";
    setTimeout(() => window.location.href = "homepage.html", 2000);
});

</script>

</body>
</html>
```

Output:

registration page output:

The screenshot shows a 'Create an Account' form on a website. The form fields include: Full Name (text input), Username (text input), Email (text input), Phone Number (text input), Date of Birth (date input with placeholder 'dd-mm-yyyy'), Gender (dropdown menu with 'Select' option), Address (text input), Preferred Genre (dropdown menu with 'Select' option), and Password (text input). The form is centered on a light gray background.

Code:

login page:

code:

```
<!DOCTYPE html>

<html>
<head>
<title>Login</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

```

```
form {
```

```
background-color: white;
```

```
padding: 25px;  
border-radius: 8px;  
box-shadow: 0 0 10px rgba(0,0,0,0.1);  
width: 300px;  
}
```

```
h2 {  
margin-bottom: 15px;  
text-align: center;  
}
```

```
label {  
display: block;  
margin-top: 10px;  
font-weight: bold;  
}
```

```
input.box {  
width: 100%;  
padding: 8px;  
margin-top: 5px;  
border: 1px solid #ccc;  
border-radius: 4px;  
}
```

```
#show-pass {  
margin-top: 10px;  
background: none;  
border: none;  
color: blue;
```

```
    cursor: pointer;  
    text-decoration: underline;  
}  
  
#submit-btn {
```

```
    margin-top: 15px;  
    width: 100%;  
    padding: 10px;  
    background-color: #333;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    opacity: 0.5;
```

```
}
```

```
#submit-btn:enabled {  
    opacity: 1;  
}
```

```
.msg {  
    margin-top: 10px;  
    font-weight: bold;  
    text-align: center;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
    <h2>Login Form</h2>
```

```
<label for="username">User Name</label>

<input type="text" class="box" placeholder="Enter User name" id="username" name="username">

<label for="pass">Password</label>

<input type="password" class="box" placeholder="Enter Password" id="pass" name="pass">

<button id="show-pass">Show Password</button>

<input type="submit" id="submit-btn" value="Login" disabled>

<div class="msg"></div>

</form>

<script>

const submit = document.getElementById('submit-btn');

const msgElement = document.querySelector('.msg');

const showPassBtn = document.getElementById('show-pass');

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

const passwordInput = document.getElementById('pass');

const validUser = "OjasUmate";

const validPass = "Ojas@123";

// Enable login button when both fields are filled

usernameInput.addEventListener('input', validateForm);

passwordInput.addEventListener('input', validateForm);

function validateForm() {

if (usernameInput.value.trim() && passwordInput.value.trim()) {

submit.disabled = false;
```

```
    } else {
        submit.disabled = true;
    }
}

// Toggle password visibility

showPassBtn.addEventListener('click', function (e) {
    e.preventDefault();
    passwordInput.type = passwordInput.type === "password" ? "text" : "password";
    showPassBtn.textContent = passwordInput.type === "password" ? "Show Password" : "Hide Password";
});

// Handle form submission

submit.addEventListener('click', function (e) {
    e.preventDefault();

    let enteredUser = usernameInput.value.trim();
    let enteredPass = passwordInput.value;

    if (enteredUser === validUser && enteredPass === validPass) {
        msgElement.style.color = 'green';
        msgElement.textContent = 'Successfully logged in';

        localStorage.setItem('userDetails', JSON.stringify({ username: enteredUser }));
    }

    setTimeout(() => {
        window.location.href = "homepage.html";
    }, 2000);
} else {
    msgElement.style.color = 'red';
}
```

```

        msgElement.textContent = 'Invalid Username or Password';

    }

});
```

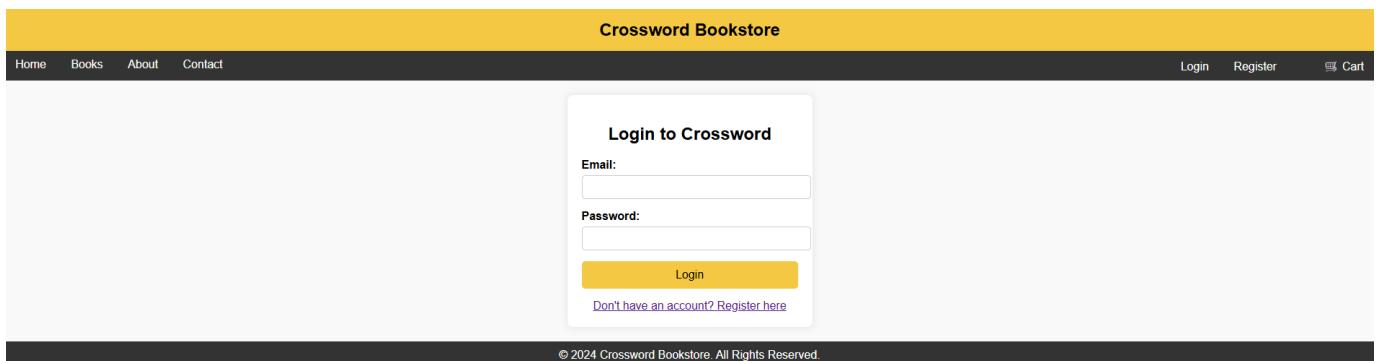
</script>

</body>

</html>

Output:

login page output:



Conclusion

The visual and functional success of any e-commerce platform, especially one like your second-hand gaming console website, relies heavily on how well the pages are styled using CSS. Applying appropriate margins, paddings, spacing, and input field enhancements ensures:

- Better user experience (UX)
- Improved readability and accessibility
- A more polished, professional appearance
- Higher engagement, conversion, and trust

Each page — whether it's the cart, about, contact, or form — serves a critical role in the user's journey. Styling them properly not only improves usability but also communicates quality, attention to detail, and brand identity.

In modern web design, CSS is not just about making things look pretty — it's about guiding users through a seamless experience, one pixel at a time.

Experiment No.5

Problem Statement:

JavaScript

- A. Implement user registration and login forms for the coffee shop website. These forms will allow users to create an account, log in, and access personalized features, such as saving favorite items or viewing order history. User Registration Form will allow new customers to sign up and create an account on the website. The form will capture basic user details, including the name, email address, and password (not limited to these fields). User Login Form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.
- B. Provide validations for user registration and login forms to validate the input to ensure that all required fields are filled and that the email format is valid. (Contents beyond Syllabus)
- B. Develop cart functionality to allow users to add items, update quantities, and remove items.

JavaScript Theory: User Registration, Login, Validation, and Cart Functionality

Introduction

In modern web development, client-side scripting using JavaScript is essential for creating interactive, responsive, and user-friendly applications. For an e-commerce website, particularly one focusing on second-hand gaming consoles, implementing registration, login, form validation, and shopping cart functionality is a core requirement to facilitate smooth user engagement and personalized services.

1. User Registration and Login Forms

These forms are critical for establishing user identity and enabling personalized user experiences. JavaScript is used to enhance the responsiveness and usability of these forms before the data is submitted to the server or stored locally in a prototype.

Registration Form

The registration form allows new users to create an account by entering their personal details. This form typically includes fields like full name, email address, password, confirm password, and optionally phone number or address.

Key responsibilities of JavaScript in registration:

- Ensuring that no field is left empty
- Verifying the validity of the email using regular expressions

- Checking that the password meets certain criteria (e.g., minimum length, use of special characters)
- Validating that both password and confirm password fields match
- Providing real-time feedback to the user in case of errors

Login Form

The login form allows returning users to access their accounts using their email and password.

Key responsibilities of JavaScript in login:

- Ensuring that the email and password fields are not empty
- Validating the format of the email address
- Matching the input credentials with previously registered data (locally or via backend)
- Redirecting the user to a dashboard or main page upon successful authentication

2. JavaScript Form Validations

Form validation ensures the accuracy and completeness of user input. It is crucial for data integrity and a better user experience.

Typical validation tasks include:

- Ensuring all mandatory fields are filled
- Validating email address formats using regular expressions
- Verifying password strength (length, characters, etc.)
- Checking that passwords match
- Displaying inline error messages when incorrect input is detected

Client-side validation is often complemented by server-side validation for enhanced security, but using JavaScript provides immediate feedback and reduces unnecessary server requests.

3. Cart Functionality

The shopping cart is an essential component of any e-commerce website. It allows users to review their selections, modify quantities, and proceed to checkout.

Key JavaScript implementations for the cart include:

- Adding selected products to the cart dynamically
- Updating the quantity of items and recalculating totals
- Removing items from the cart
- Storing the cart state in local storage or session storage for persistence
- Rendering cart items in real-time using dynamic DOM manipulation

By maintaining the cart structure as an array of objects in JavaScript, developers can efficiently manage item details, prices, and totals.

Code:

F. registration page:

code:

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Register</title>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
  }
  body {
    background-color: #f4f4f4;
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
  }
  .container {
    background-color: white;
    padding: 25px;
  }
</style>
```

```
border-radius: 10px;  
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
width: 350px;  
text-align: center;  
}
```

```
h2 {  
margin-bottom: 20px;  
}
```

```
.form-group {  
position: relative;  
margin-bottom: 15px;  
text-align: left;  
}
```

```
label {  
font-weight: bold;  
}
```

```
input {  
width: 100%;  
padding: 10px;  
margin-top: 5px;  
border: 1px solid #ccc;  
border-radius: 6px;  
font-size: 14px;  
outline: none;  
transition: border 0.3s ease-in-out;  
}
```

```
input:focus {  
    border-color: #007bff;  
}  
  
.password-container {  
    position: relative;  
}  
  
.toggle-password {  
    position: absolute;  
    right: 10px;  
    top: 50%;  
    transform: translateY(-50%);  
    cursor: pointer;  
    font-size: 16px;  
    color: #777;  
}  
  
.valid {  
    border: 2px solid green !important;  
}  
  
.invalid {  
    border: 2px solid red !important;  
}  
  
.error {  
    color: red;  
    font-size: 12px;
```

```
margin-top: 3px;  
height: 14px;  
}  
  
  
button {  
background-color: #007bff;  
color: white;  
padding: 12px;  
border: none;  
border-radius: 6px;  
cursor: pointer;  
width: 100%;  
margin-top: 10px;  
font-size: 16px;  
opacity: 0.5;  
transition: background-color 0.3s ease-in-out, opacity 0.3s ease-in-out;  
}  
  
  
button:hover {  
background-color: #0056b3;  
}  
  
  
button:enabled {  
opacity: 1;  
}  
  
</style>  
</head>  
<body>  
  
  
<div class="container">
```

```

<h2>Register</h2>

<form id="registrationForm">

  <div class="form-group">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <p class="error" id="usernameError"></p>
  </div>

  <div class="form-group password-container">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <span class="toggle-password" onclick="togglePassword('password', this)">≡ | </span>
    <p class="error" id="passwordError"></p>
  </div>

  <div class="form-group password-container">
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword" required>
    <span class="toggle-password" onclick="togglePassword('confirmPassword', this)">▣ | </span>
    <p class="error" id="confirmPasswordError"></p>
  </div>

  <button type="submit" id="registerButton" disabled>Register</button>
</form>

<p id="successMessage" style="color: green; display: none;">Registration successful! Redirecting...</p>
</div>

<script>

  const form = document.getElementById("registrationForm");
  const usernameField = document.getElementById("username");
  const passwordField = document.getElementById("password");
  const confirmPasswordField = document.getElementById("confirmPassword");
  const registerButton = document.getElementById("registerButton");

  ...
</script>

```

```
const validationRules = {

    username: {

        regex: /^[a-zA-Z0-9]{5,}$/,
        errorMsg: "Username must be at least 5 characters and contain only letters and numbers."
    },

    password: {

        regex: /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{6,16}$/,
        errorMsg: "Password must be 6-16 characters, with uppercase, lowercase, number, and special character."
    }
};

function validateField(field, rule) {

    const value = field.value.trim();
    const errorElement = document.getElementById(field.id + "Error");

    if (rule.regex.test(value)) {

        field.classList.add("valid");
        field.classList.remove("invalid");
        errorElement.textContent = "";
        return true;
    } else {

        field.classList.add("invalid");
        field.classList.remove("valid");
        errorElement.textContent = rule.errorMsg;
        return false;
    }
}

function validateConfirmPassword() {
```

```

const password = passwordField.value;

const confirmPassword = confirmPasswordField.value;

const errorElement = document.getElementById("confirmPasswordError");

if (confirmPassword === password && confirmPassword !== "") {

    confirmPasswordField.classList.add("valid");

    confirmPasswordField.classList.remove("invalid");

    errorElement.textContent = "";

    return true;

} else {

    confirmPasswordField.classList.add("invalid");

    confirmPasswordField.classList.remove("valid");

    errorElement.textContent = "Passwords do not match./";

    return false;

}

}

function validateForm() {

    const isUsernameValid = validateField(usernameField, validationRules.username);

    const isPasswordValid = validateField(passwordField, validationRules.password);

    const isConfirmPasswordValid = validateConfirmPassword();

    registerButton.disabled = !(isUsernameValid && isPasswordValid && isConfirmPasswordValid);

}

function togglePassword(fieldId, icon) {

    const field = document.getElementById(fieldId);

    if (field.type === "password") {

        field.type = "text";

        icon.textContent = "visibility_off" // Hide Password Icon
    }
}

```

```
        } else {
            field.type = "password";
            icon.textContent = "eye-off" // Show Password Icon
        }
    }

usernameField.addEventListener("input", validateForm);
passwordField.addEventListener("input", validateForm);
confirmPasswordField.addEventListener("input", validateForm);

form.addEventListener("submit", function (event) {
    event.preventDefault();
    document.getElementById("successMessage").style.display = "block";
    setTimeout(() => window.location.href = "homepage.html", 2000);
});

</script>

</body>
</html>
```

Output:

F. registration page output:

The screenshot shows a 'Create an Account' form on a website. The form fields and their current values are:

- Full Name:** Pra
- Username:** kl
Error message: 4-12 alphanumeric characters.
- Email:** pranitapp1587@gm.
Error message: Invalid email format.
- Phone Number:** 09850181587
Error message: Enter a valid 10-digit number.
- Date of Birth:** dd - mm - yyyy
- Gender:** Select

Code:

G. login page:

code:

```
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

```

```
form {
    background-color: white;
    padding: 25px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

```
width: 300px;  
}  
  
h2 {  
    margin-bottom: 15px;  
    text-align: center;  
}  
  
label {  
    display: block;  
    margin-top: 10px;  
    font-weight: bold;  
}  
  
input.box {  
    width: 100%;  
    padding: 8px;  
    margin-top: 5px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}  
  
#show-pass {  
    margin-top: 10px;  
    background: none;  
    border: none;  
    color: blue;  
    cursor: pointer;  
    text-decoration: underline;  
}
```

```
#submit-btn {  
    margin-top: 15px;  
    width: 100%;  
    padding: 10px;  
    background-color: #333;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    opacity: 0.5;  
}  
  
#submit-btn:enabled {  
    opacity: 1;  
}  
  
.msg {  
    margin-top: 10px;  
    font-weight: bold;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
<form>  
<h2>Login Form</h2>  
<label for="username">User Name</label>  
<input type="text" class="box" placeholder="Enter User name" id="username" name="username">
```

```
<label for="pass">Password</label>

<input type="password" class="box" placeholder="Enter Password" id="pass" name="pass">

<button id="show-pass">Show Password</button>

<input type="submit" id="submit-btn" value="Login" disabled>

<div class="msg"></div>

</form>

<script>

const submit = document.getElementById('submit-btn');

const msgElement = document.querySelector('.msg');

const showPassBtn = document.getElementById('show-pass');

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

const passwordInput = document.getElementById('pass');

const validUser = "OjasUmate";

const validPass = "Ojas@123";

// Enable login button when both fields are filled

usernameInput.addEventListener('input', validateForm);

passwordInput.addEventListener('input', validateForm);

function validateForm() {

    if (usernameInput.value.trim() && passwordInput.value.trim()) {

        submit.disabled = false;

    } else {

        submit.disabled = true;

    }

}
```

```
}

// Toggle password visibility

showPassBtn.addEventListener('click', function (e) {

    e.preventDefault();

    passwordInput.type = passwordInput.type === "password" ? "text" : "password";

    showPassBtn.textContent = passwordInput.type === "password" ? "Show Password" : "Hide Password";

});

// Handle form submission

submit.addEventListener('click', function (e) {

    e.preventDefault();

    let enteredUser = usernameInput.value.trim();

    let enteredPass = passwordInput.value;

    if (enteredUser === validUser && enteredPass === validPass) {

        msgElement.style.color = 'green';

        msgElement.textContent = 'Successfully logged in';

        localStorage.setItem('userDetails', JSON.stringify({ username: enteredUser }));

        setTimeout(() => {

            window.location.href = "homepage.html";

        }, 2000);

    } else {

        msgElement.style.color = 'red';

        msgElement.textContent = 'Invalid Username or Password';

    }

});
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

G. login page output:

The screenshot shows the login interface for the Crossword Bookstore. At the top, there's a yellow header bar with the text "Crossword Bookstore". Below it is a dark grey navigation bar. The main content area is white and features a centered login form. The form has a title "Login to Crossword". It contains two input fields: one for "Email" with the value "praitapp158" and one for "Password" with the value ".....". To the left of the password field is a small orange exclamation mark icon. Below the inputs, a red-bordered box displays a validation message: "Please include an '@' in the email address. 'praitapp158' is missing an '@'". At the bottom of the form is a yellow "Login" button and a blue "Don't have an account? Register here" link.

Conclusion

Implementing registration, login, validation, and cart features using JavaScript is fundamental for any user-centered e-commerce platform. These features not only enhance usability but also ensure smooth interactions, better data handling, and a seamless user journey.

For a second-hand gaming console website, these JavaScript functionalities provide the foundation for user management and interactive shopping experiences. Users can securely register and log in, receive immediate validation feedback, and manage their cart efficiently. This setup lays the groundwork for integrating advanced features like order history, wish lists, and secure checkout in future iterations.

JavaScript, therefore, plays a critical role in transforming a static product listing site into a dynamic and functional e-commerce platform.

Experiment No.6

Problem Statement:

JavaScript

- A. The user login form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.
- B. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus**)
- C. Use localStorage or sessionStorage to store authentication data, such as the user's email and login status. This ensures that once a user is logged in, they remain authenticated even after the page reloads or when they visit the site again. (**Contents beyond Syllabus**)
- D. Save the cart data to local storage when items are added, updated, or removed. Retrieve and load the cart data from local storage when the page loads. (**Contents beyond Syllabus**)

JavaScript Theory: Persistent Login and Cart Functionality using Web Storage API

Introduction

In modern web applications, offering a seamless user experience requires maintaining user session states and data across different pages or after a page refresh. JavaScript's Web Storage API—comprising localStorage and sessionStorage—is a lightweight solution to store data on the client side. For a second-hand gaming console website, using these features can significantly improve usability by allowing persistent login sessions and retaining cart data even after page reloads or temporary site exits.

1. Persistent Login using localStorage/sessionStorage

The login system allows users to securely enter their credentials (email and password) to gain access to their accounts. Once validated, their login status and user identifier (like email or user ID) are stored in the browser using either:

- localStorage: Stores data with no expiration time, persisting even after the browser is closed and reopened.
- sessionStorage: Stores data only for the duration of the page session (i.e., until the tab or browser is closed).

Implementation Features:

- After successful login, JavaScript stores:

- userEmail: to identify the current user
- isLoggedIn: a boolean flag to indicate the login status
- On subsequent visits or page reloads:
 - JavaScript checks for these flags and either redirects to the dashboard or shows the login screen
- Logout functionality clears the stored values, ending the session

Benefits:

- Eliminates the need to re-login on every visit
- Enhances user convenience and session continuity
- Reduces server load for small-scale or prototype apps

2. Cart Data Management using localStorage

Shopping carts are central to any e-commerce website. Users expect that the items they add remain intact even if they leave or refresh the page. localStorage enables this by preserving the state of the cart.

Implementation Features:

- Every time a user adds, removes, or updates a product in the cart:
 - JavaScript serializes the cart array/object into JSON
 - This data is saved to localStorage
- On page load:
 - JavaScript checks if cart data exists in localStorage
 - If it does, it parses and loads it into the cart view
- The cart remains persistent until explicitly cleared

Benefits:

- Prevents loss of user data on reload or accidental tab closure
- Creates a more seamless and intuitive shopping experience
- Ensures continuity across visits without requiring account creation

Use Cases Beyond the Syllabus (Advanced Learning):

These implementations represent concepts often covered beyond standard academic curricula:

- Managing state with client-side storage
- Working with JSON and JavaScript objects dynamically
- Handling user sessions in single-page or multi-page applications without a backend
- Creating realistic e-commerce simulations or prototypes for portfolio projects

Code:

A. Home page:

code:

```
<!DOCTYPE html>

<html>
<head>
<title>Login</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
```

```
form {
    background-color: white;
    padding: 25px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    width: 300px;
}
```

```
h2 {
    margin-bottom: 15px;
    text-align: center;
}
```

```
label {  
    display: block;  
    margin-top: 10px;  
    font-weight: bold;  
}
```

```
input.box {  
    width: 100%;  
    padding: 8px;  
    margin-top: 5px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}
```

```
#show-pass {  
    margin-top: 10px;  
    background: none;  
    border: none;  
    color: blue;  
    cursor: pointer;  
    text-decoration: underline;  
}
```

```
#submit-btn {  
    margin-top: 15px;  
    width: 100%;  
    padding: 10px;  
    background-color: #333;  
    color: white;
```

```
border: none;  
border-radius: 4px;  
cursor: pointer;  
opacity: 0.5;  
}  
  
  
#submit-btn:enabled {  
    opacity: 1;  
}  
  
  
.msg {  
    margin-top: 10px;  
    font-weight: bold;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
    <form>  
        <h2>Login Form</h2>  
        <label for="username">User Name</label>  
        <input type="text" class="box" placeholder="Enter User name" id="username" name="username">  
        <br/>  
        <label for="pass">Password</label>  
        <input type="password" class="box" placeholder="Enter Password" id="pass" name="pass">  
        <br/>  
        <button id="show-pass">Show Password</button>  
        <br/>  
        <input type="submit" id="submit-btn" value="Login" disabled>
```

```
<div class="msg"></div>

</form>

<script>

const submit = document.getElementById('submit-btn');

const msgElement = document.querySelector('.msg');

const showPassBtn = document.getElementById('show-pass');

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

const passwordInput = document.getElementById('pass');

const validUser = "OjasUmate";

const validPass = "Ojas@123";

// Enable login button when both fields are filled

usernameInput.addEventListener('input', validateForm);

passwordInput.addEventListener('input', validateForm);

function validateForm() {

    if (usernameInput.value.trim() && passwordInput.value.trim()) {

        submit.disabled = false;

    } else {

        submit.disabled = true;

    }

}

// Toggle password visibility

showPassBtn.addEventListener('click', function (e) {

    e.preventDefault();

    passwordInput.type = passwordInput.type === "password" ? "text" : "password";

    showPassBtn.textContent = passwordInput.type === "password" ? "Show Password" : "Hide Password";
})
```

```
});

// Handle form submission

submit.addEventListener('click', function (e) {
    e.preventDefault();

    let enteredUser = usernameInput.value.trim();
    let enteredPass = passwordInput.value;

    if (enteredUser === validUser && enteredPass === validPass) {
        msgElement.style.color = 'green';
        msgElement.textContent = 'Successfully logged in';

        localStorage.setItem('userDetails', JSON.stringify({ username: enteredUser }));

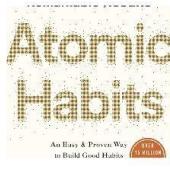
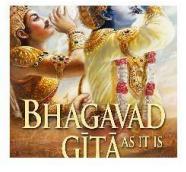
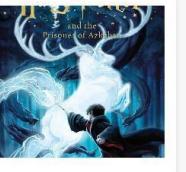
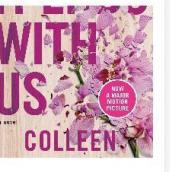
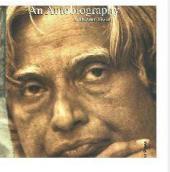
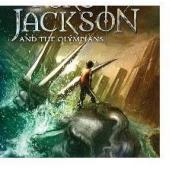
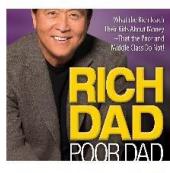
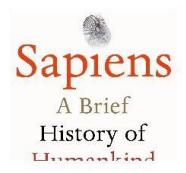
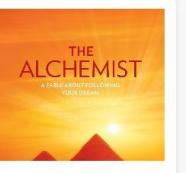
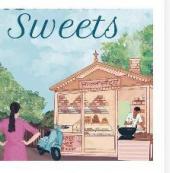
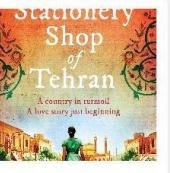
        setTimeout(() => {
            window.location.href = "homepage.html";
        }, 2000);
    } else {
        msgElement.style.color = 'red';
        msgElement.textContent = 'Invalid Username or Password';
    }
});

</script>

</body>
</html>
```

Output:

- A. Index/Home page output:

 Atomic Habits ₹499 Transform your habits and life with James Clear's practical strategies for building better routines. Add to Cart	 Bhagavad Gita ₹299 A timeless spiritual dialogue between Krishna and Arjuna on life, duty, and righteousness. Add to Cart	 Harry Potter ₹599 J.K. Rowling's magical series that chronicles the life of the young wizard, Harry Potter. Add to Cart	 It Ends With Us ₹399 Colleen Hoover's powerful novel about love, heartbreak, and breaking the cycle of abuse. Add to Cart	 Wings of Fire ₹350 An inspiring autobiography of Dr. A.P.J. Abdul Kalam, India's missile man and former President. Add to Cart	 Jackson and the Olympians ₹499 Rick Riordan's thrilling series that blends Greek mythology with modern adventure. Add to Cart
 Rich Dad Poor Dad ₹399 Robert Kiyosaki's bestselling book that challenges conventional ideas about money and wealth. Add to Cart	 Sapiens ₹549 Yuval Noah Harari's sweeping narrative of the history of humankind from evolution to the present. Add to Cart	 The Alchemist ₹299 Paulo Coelho's philosophical tale about a shepherd boy chasing his personal legend. Add to Cart	 Sweets ₹399 An emotional journey of a young entrepreneur who revives his family's sweet shop legacy. Add to Cart	 The Stationery Shop of Tehran ₹449 A touching story of love and separation set in 1950s Iran by Marjan Kamali. Add to Cart	 Fool Me Twice ₹329 An intriguing tale of deception, mystery, and second chances. Add to Cart

Conclusion

Using JavaScript in combination with the Web Storage API (localStorage/sessionStorage) significantly enhances user experience and functionality in web development. For a second-hand gaming console website, implementing persistent login and cart functionality ensures that users have a smooth, uninterrupted interaction with the site.

By storing authentication states and cart data locally:

- Users remain logged in across sessions
- Cart items persist across visits
- The website feels more responsive and user-centric

These techniques mimic real-world behavior found in professional e-commerce platforms, making them excellent additions to projects meant for academic distinction or professional portfolios. Ultimately, mastering such features prepares developers to build more dynamic, reliable, and user-friendly web applications.

Experiment no.7

Problem Statement:

PHP

- A. Develop a PHP script to handle user registration for the Coffee Shop website. The script should accept input from users for their name, email address, password, etc. (all required fields for registration).
- B. Implement error handling to notify users of any issues during registration, such as validation errors.
- C. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page.

A. Develop a PHP script to handle user registration for the Coffee Shop website. The script should accept input from users for their name, email address, password, etc. (all required fields for registration).

B. Implement error handling to notify users of any issues during registration, such as validation errors.

C. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page.

User registration is a fundamental component of web applications, particularly in e-commerce platforms like your second-hand gaming console website. PHP is widely used on the server side to handle form submissions, validate user inputs, interact with databases (like MySQL), and ensure secure data processing.

In this system, the registration form captures user details (name, email, password, etc.). Once submitted, the PHP script validates the inputs and then stores them securely into a database. To maintain security, user passwords are hashed before storage.

Core Elements of the PHP Registration Script:

1. Form Handling: Grabs data using `$_POST`.
2. Validation: Ensures fields are not empty and email is valid.
3. Password Hashing: Uses `password_hash()` to securely hash passwords.
4. Database Interaction: Uses MySQLi or PDO to store user data.
5. Error Handling: Displays messages for missing fields or registration failures.
6. User Feedback: Provides confirmation or redirection upon success.

CODE:-

```
<?php
// db_connect.php (include this file wherever needed)
$host = 'localhost';
$user = 'root';
```

```
$password = "";
$dbname = 'gaming_store';

$conn = new mysqli($host, $user, $password, $dbname);
```

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

?>
Registration:-
```

```
<?php
include 'db_connect.php';

$name = $email = $password = "";
$errors = [];

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get input values and sanitize
    $name = trim($_POST["name"]);
    $email = trim($_POST["email"]);
    $password = trim($_POST["password"]);

    // Basic validation
    if (empty($name)) $errors[] = "Name is required.";
    if (empty($email) || !filter_var($email, FILTER_VALIDATE_EMAIL)) $errors[] = "Valid email is required.";
    if (empty($password) || strlen($password) < 6) $errors[] = "Password must be at least 6 characters.";

    // If no errors, proceed to store user
    if (empty($errors)) {
        $hashedPassword = password_hash($password, PASSWORD_BCRYPT);
```

```

$stmt = $conn->prepare("INSERT INTO users (name, email, password) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $name, $email, $hashedPassword);

if ($stmt->execute()) {
    echo "<p>Registration successful. <a href='login.html'>Click here to login</a>.</p>";
} else {
    echo "<p>Error: " . $stmt->error . "</p>";
}

$stmt->close();

} else {
    foreach ($errors as $error) {
        echo "<p style='color:red;'>$error</p>";
    }
}

$conn->close();
}

?>

```

Conclusion

Implementing user registration with PHP provides the backbone of user management in your website. By securely collecting, validating, and storing user data, you enable personalized experiences and functionalities such as login, saving favorites, or viewing past orders.

This system:

- Promotes user trust by securing sensitive data like passwords.
- Ensures data integrity through server-side validation.
- Enhances the user experience with real-time feedback and clear error handling
-

Experiment 8

- A. Develop a PHP script to handle user login for the Coffee Shop website. The script should accept input from users for their login credentials. (all required fields for login).
- B. Provide feedback to the user upon successful login, either through a confirmation message or a redirect to a welcome page.
- C. Implement error handling to notify users of login failures due to incorrect credentials or other errors.
- D. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page.

Theory: PHP Login System

A user login system is a fundamental component of most websites, especially e-commerce platforms. It enables secure access to personalized features like managing carts, tracking orders, or viewing saved products. In PHP, login functionality typically involves:

- Capturing login credentials via a form (email and password).
- Validating inputs.
- Comparing credentials against stored data in a database.
- Starting a session upon successful login.
- Redirecting or displaying a welcome message.
- Showing errors for invalid credentials.

Security Aspects:

- **Password Hashing & Verification:** Passwords are stored as hashes using `password_hash()` during registration. PHP's `password_verify()` is used to compare hashes during login.
- **Session Handling:** PHP sessions are used to maintain the user's login state across pages.

CODE:-

```
<?php
session_start();
include 'db_connect.php';

$email = $password = "";
$errors = [];

if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```

$email = trim($_POST["email"]);

$password = trim($_POST["password"]);

// Basic validation

if (empty($email) || !filter_var($email, FILTER_VALIDATE_EMAIL)) {

    $errors[] = "Please enter a valid email address./";

}

if (empty($password)) {

    $errors[] = "Please enter your password./";

}

// Proceed only if no validation errors

if (empty($errors)) {

    $stmt = $conn->prepare("SELECT id, name, email, password FROM users WHERE email = ?");

    $stmt->bind_param("s", $email);

    $stmt->execute();

    $result = $stmt->get_result();

    if ($result && $result->num_rows === 1) {

        $user = $result->fetch_assoc();

        if (password_verify($password, $user['password'])) {

            // Correct login

            $_SESSION["user_id"] = $user['id'];

            $_SESSION["user_name"] = $user['name'];

            $_SESSION["user_email"] = $user['email'];

            echo "<p>Welcome, <strong>" . htmlspecialchars($user['name']) . "</strong>! Redirecting to home...</p>";

            header("refresh:2;url=home.php"); // redirect after 2 seconds

            exit();

        } else {

    }

}

```

```

$errors[] = "Incorrect password.";

}

} else {

$errors[] = "No account found with that email.';

}

$stmt->close();

}

$conn->close();

}

```

// Display errors if any

```

foreach ($errors as $error) {

echo "<p style='color:red;'>$error</p>";

}

?>

```

Login form:-

```

<form action="login.php" method="POST">

<h2>Login</h2>

<label>Email:</label><br>

<input type="email" name="email" required><br><br>

<label>Password:</label><br>

<input type="password" name="password" required><br><br>

<input type="submit" value="Login">

</form>

```

Dashboard:-

```
<?php
```

```
session_start();

if (!isset($_SESSION["user_id"])) {
    echo "Access denied. Please <a href='login.html'>login</a>.";
    exit();
}

echo "<h2>Welcome back, " . htmlspecialchars($_SESSION["user_name"]) . "</h2>";
echo "<p>You are logged in with email: " . htmlspecialchars($_SESSION["user_email"]) . "</p>";
echo "<a href='logout.php'>Logout</a>";
?>

<?php
session_start();
session_destroy();
header("Location: login.html");
exit();
?>
```

Conclusion

Implementing a login system with PHP ensures a secure and user-friendly experience for your second-hand gaming console website. By validating input, securely verifying credentials, and using PHP sessions:

- You allow users to access personalized features.
- You prevent unauthorized access.
- You improve user engagement and trust.

This login system:

- Supports **secure authentication** using hashed passwords.
- Provides **real-time feedback** for incorrect credentials.
- Ensures **session persistence** and protects pages using login checks.

Crossword Bookstore

Home Books About Contact Login Register

Create an Account

Full Name:

Username:

Email:

Phone Number:

Date of Birth:

Gender:

Address:

Preferred Genre:

```
mysql> SELECT * FROM registration;
+-----+-----+-----+-----+-----+-----+-----+-----+
| full_name | username | email | phone_number | date_of_birth | gender | address |
| preferred_genre | password | confirm_password | newsletter_subscription |
+-----+-----+-----+-----+-----+-----+-----+
| Pranita Potghan | pnand | pranitapp1587@gmail.com | 9850181587 | 2025-04-18 | Female | B1/404, Accolade Society |
| Kharadi, Pune | Fiction | ?????????? | ?????????? | 1 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Experiment 9

Problem Statement

PHP and MySQL

- A. Develop a PHP script that allows users to manage their shopping cart for an e-commerce website (e.g., a Coffee Shop store). The script should allow users to add items to their cart, view their cart contents, and remove items if needed.
- B. Develop a PHP script to manage the shopping cart for an e-commerce website (e.g., a Coffee Shop store) using MySQL. This script should allow users to add items to their cart, view their cart contents, and remove items from the cart. The cart data should be stored in the MySQL database to allow persistence across sessions.

- A. Develop a PHP script that allows users to manage their shopping cart for an second hand gaming consoles website. The script should allow users to add items to their cart, view their cart contents, and remove items if needed.
- B. Develop a PHP script to manage the shopping cart for an second hand gaming consoles website using MySQL. This script should allow users to add items to their cart, view their cart contents, and remove items from the cart. The cart data should be stored in the MySQL database to allow persistence across sessions

Theory: PHP Shopping Cart System

A shopping cart is a core component of any e-commerce platform. It serves as a temporary storage space where users can collect and manage the items they wish to purchase. In the case of a second-hand gaming consoles website, where products can be unique and availability may be limited to single units, the shopping cart system plays an even more critical role.

Two Types of Cart Management Systems in PHP:

A. Session-Based Shopping Cart (Without MySQL)

This approach uses PHP sessions to temporarily store cart data in memory while the user is browsing. It is useful for fast prototyping and requires no database interaction.

Key Characteristics:

- Cart data is stored in `$_SESSION`.
- Data persists during the browsing session.
- No need to log in to use the cart.
- Items are lost if the session expires or the browser is closed.

Operations Supported:

- **Add to Cart:** Add items by storing product ID, name, quantity, and price in session.

- **View Cart:** Display the contents stored in session.
- **Remove from Cart:** Unset item by ID or index from the session.

Advantages:

- Simple to implement.
- No database overhead.

Limitations:

- Not persistent after session end.
- Not scalable for logged-in user experiences.

B. Database-Based Shopping Cart (With MySQL)

This is the professional and scalable approach where cart data is stored in a **MySQL database**. It allows cart contents to persist across user sessions, devices, and logins.

Key Characteristics:

- Each user has a unique cart identified by user ID.
- Cart contents are stored in a cart table, and optionally a cart_items table for item details.
- Requires user login or session management.

Operations Supported:

- **Add to Cart:** Insert or update records in the cart_items table.
- **View Cart:** Query database for all cart items belonging to a specific user.
- **Remove from Cart:** Delete an item from the database by item ID or cart ID.

Advantages:

- Cart is persistent and user-specific.
- Works across sessions and devices.
- Enables cart analytics and user behavior tracking.

Limitations:

- Requires more setup and error handling.
- Needs secure login system to link cart with user.

CODE:-

```
CREATE TABLE cart_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
```

```

product_id INT NOT NULL,
product_name VARCHAR(255),
quantity INT DEFAULT 1,
price DECIMAL(10, 2),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Conclusion

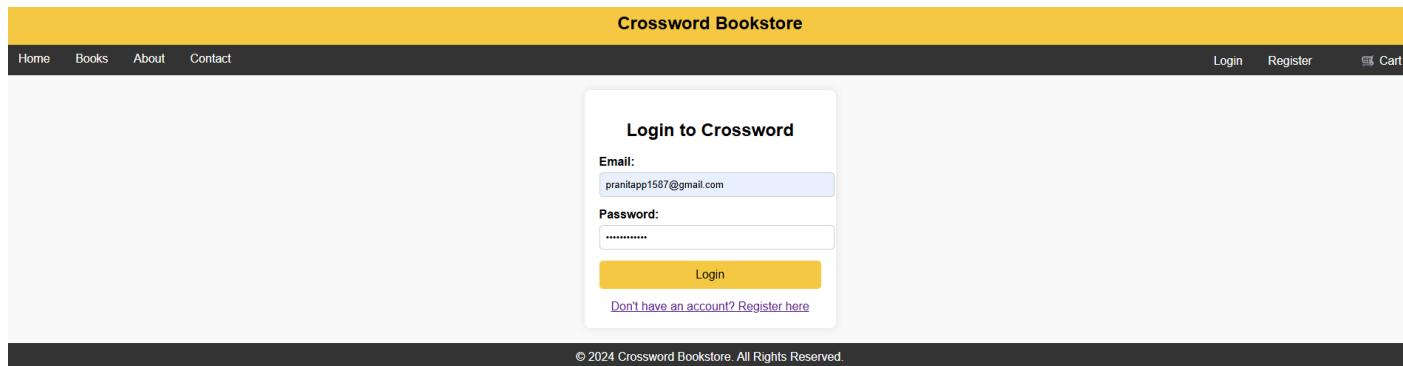
A shopping cart system, whether session-based or database-driven, is essential for enhancing the user experience and improving sales on your second-hand gaming console website.

When using PHP:

- **Session-based carts** offer fast and simple cart functionality, ideal for guest users.
- **MySQL-backed carts** provide reliable, persistent storage across sessions and devices—ideal for logged-in users and production-level systems.

For a fully functional and scalable website, the **MySQL-based cart** is highly recommended, as it:

- Improves user experience with persistent carts.
- Enables personalization and user analytics.
- Supports consistent item tracking (especially when each console unit is unique).



```

mysql> SELECT * FROM login;
+-----+-----+
| name | password |
+-----+-----+
| Pranita Potghan | ****      |
+-----+-----+
1 row in set (0.00 sec)

```

Experiment 10

A. Develop a PHP script to handle the checkout process for users who are ready to complete their purchase. The script should process the cart data and provide feedback to the user upon successful or failed checkout.

B. Develop a PHP script that processes the checkout process for users who are ready to complete their purchase, integrating the MySQL database for handling user and order information. The script should validate user input, process the cart data, and provide feedback upon successful or failed checkout.

Theory: PHP Checkout Process

The **checkout process** is the final and most crucial step in any e-commerce platform. It translates the user's cart into an official order, capturing necessary details such as billing, shipping, and payment, then recording it into the database for processing and fulfillment.

On a second-hand gaming console website, where products may be unique or limited, a **robust and accurate checkout system** ensures that stock integrity is maintained and customer satisfaction is upheld.

Two Approaches to Checkout

A. Session-Based Checkout (Without Database Order Management)

In this basic approach:

- All data is stored in the session (`$_SESSION['cart']`).
- On checkout, a confirmation message is shown.
- Useful for simple or demo applications. **Workflow:**
 1. Retrieve cart from `$_SESSION`.
 2. Validate input fields (name, email, address).
 3. Show success or error message.
 4. Clear cart after checkout.

Advantages:

- Quick to implement.
- Minimal setup required.

Limitations:

- Data not persistent.
- Not scalable or production-ready.
- No order history.

B. MySQL-Based Checkout System

This advanced and scalable approach:

- Stores order details in a MySQL database.
- Supports persistence, analytics, and back-end processing.
- Links orders to logged-in users.

Workflow:

1. Validate user session or login status.
2. Retrieve cart items from session or database.
3. Validate checkout fields (shipping info, contact).
4. Insert data into orders and order_items tables.
5. Display success/failure message.
6. Clear session cart.

Code:-

MYSQL Code

CREATE TABLE orders (

```
id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT,
customer_name VARCHAR(255),
customer_email VARCHAR(255),
customer_address TEXT,
total DECIMAL(10, 2),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

CREATE TABLE order_items (

```
id INT AUTO_INCREMENT PRIMARY KEY,
order_id INT,
product_id INT,
product_name VARCHAR(255),
quantity INT,
price DECIMAL(10, 2)
);
```

Checkout session:-

```
<?php  
session_start();  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    if (!isset($_SESSION['cart']) || empty($_SESSION['cart'])) {  
        echo "Your cart is empty!";  
        exit;  
    }  
  
    $name = $_POST['name'] ?? " ";  
    $email = $_POST['email'] ?? " ";  
    $address = $_POST['address'] ?? " ";  
  
    if (empty($name) || empty($email) || empty($address)) {  
        echo "Please fill in all required fields.";  
        exit;  
    }  
  
    echo "<h2>Order Summary</h2>";  
    $total = 0;  
    foreach ($_SESSION['cart'] as $item) {  
        echo "{$item['name']} - Qty: {$item['quantity']} - ₹{$item['price']} <br>";  
        $total += $item['quantity'] * $item['price'];  
    }  
  
    echo "<p>Total: ₹$total</p>";  
    echo "<p>Thank you, $name! Your order has been placed.</p>";
```

```

// Clear the cart

unset($_SESSION['cart']);

} else {

echo "Invalid request method.";

}

?>

```

MySQL-Based PHP Checkout Script:-

```

<?php

session_start();

$conn = new mysqli('localhost', 'root', '', 'gaming_store');

if ($conn->connect_error) {

die("Database connection failed: " . $conn->connect_error);

}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

if (!isset($_SESSION['cart']) || empty($_SESSION['cart'])) {

echo "Your cart is empty.';

exit;

}

$name = $_POST['name'] ?? '';

$email = $_POST['email'] ?? '';

$address = $_POST['address'] ?? '';

$user_id = $_SESSION['user_id'] ?? 0;

if (empty($name) || empty($email) || empty($address)) {

echo "All fields are required.';

```

```

exit;

}

$total = 0;

foreach ($_SESSION['cart'] as $item) {

    $total += $item['quantity'] * $item['price'];

}

$stmt = $conn->prepare("INSERT INTO orders (user_id, customer_name, customer_email, customer_address,
total) VALUES (?, ?, ?, ?, ?)");

$stmt->bind_param("isssd", $user_id, $name, $email, $address, $total);

if ($stmt->execute()) {

    $order_id = $stmt->insert_id;

    $itemStmt = $conn->prepare("INSERT INTO order_items (order_id, product_id, product_name, quantity, price)
VALUES (?, ?, ?, ?, ?)");

    foreach ($_SESSION['cart'] as $item) {

        $itemStmt->bind_param("iisid", $order_id, $item['id'], $item['name'], $item['quantity'], $item['price']);

        $itemStmt->execute();

    }

    echo "<h2>Checkout Successful</h2>";

    echo "Thank you, <strong>$name</strong>. Your order ID is <strong>$order_id</strong>. <br>Total: ₹$total";

    unset($_SESSION['cart']);

} else {

    echo "Checkout failed. Please try again. ";

}

$stmt->close();

$conn->close();

```

```

} else {
    echo "Invalid request.";
}

?>

```

Conclusion

The checkout process is the most vital component of an e-commerce platform—it turns intent into action. For your second-hand gaming consoles website:

Use Case Importance:

- **Unique item inventory** means precise, real-time cart tracking is essential.
- **Persistence** through MySQL helps avoid loss of user choices and enables full order management.
- **Session-based approach** is useful in early development or guest checkout situations.

Session-Based Checkout Summary:

- Simple and fast.
- Best suited for demos or early-stage projects.
- Not ideal for multi-session or long-term tracking.

MySQL-Based Checkout Summary:

- Scalable and professional.
- Captures order history.
- Supports user-specific orders, data analytics, and future features like order cancellation or tracking.

Your Cart					
Book	Price	Quantity	Total	Action	
It Ends With Us	₹399	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	₹399	<input type="button" value="Remove"/>	
Sapiens	₹599	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	₹599	<input type="button" value="Remove"/>	
Total					₹998
<input type="button" value="Checkout"/>					

