

**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-I/II**

# **Ms. Shreeya Salunkhe**

# 

**A.Y. 2024 – 2025 (SEM-II)**

File Index page given in the stationary

| **Web Programming**  **SEMESTER – IV** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **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:** | | | | | |
| **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.  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.  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. | | | | | |
| **Course Learning Objectives:** This course will enable the students to:   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. | | | | | |
| **Course Outcome:** After taking this course, Students will be able to :   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** | **ICT Teaching / PowerPoint Presentation and Videos:**  **Use tools like Visual Studio Code (free).**  **Videos:**  [**https://www.coursera.org/learn/html-css-javascript-for-web-developers**](https://www.coursera.org/learn/html-css-javascript-for-web-developers) | | | | |
| **Self-study / Do it yourself /:**  **Practice creating basic HTML pages and enhancing them using CSS.** | | | | |
| **Experiential Learning Topics:**  **Design a simple webpage for coffee shop website** | | | | |
| **PBL - Project Based Learning:**  **Create a multi-page website (e.g., coffee shop website) using HTML and CSS.** | | | | |
|  | | | | | |
| **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** | **ICT Teaching / PowerPoint Presentation and Videos:**  **Use tools like Visual Studio Code (free).**  **Videos:**  [**https://www.coursera.org/learn/javascript-basics**](https://www.coursera.org/learn/javascript-basics) | | | | |
| **Self-study / Do it yourself /:**  **Solve exercises on JavaScript syntax, control structures, and functions** | | | | |
| **Experiential Learning Topics:**  **Build a web page with interactive elements (e.g., a simple calculator).** | | | | |
| **PBL - Project Based Learning:**  **Develop an interactive webpage that uses JavaScript to validate form inputs or perform basic calculations.** | | | | |
|  | | | | | |
| **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** | **ICT Teaching / PowerPoint Presentation and Videos:**  [**https://www.coursera.org/learn/building-interactive-web-pages-using-javascript**](https://www.coursera.org/learn/building-interactive-web-pages-using-javascript)  **Use tools like Visual Studio Code (free).** | | | | |
| **Self-study / Do it yourself /:**  **Practice exercises on DOM traversal and event handling.** | | | | |
| **Experiential Learning Topics:**  **Add dynamic behavior to a webpage using DOM and events (e.g., a to-do list app).** | | | | |
| **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.** | | | | |
|  | | | | | |
| **UNIT – IV** | **Server Side Scripting** | | | | **09 Hours** |
| Module 7 - Set up and configure a server environment using tools like Apache or TOMCAT, set up a PHP development environment.  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,  Module 9 - Form handling with HTML and PHP: Designing of Forms using HTML, Form Handling using GET and POST methods of Form | | | | | |
| **Pedagogy** | **ICT Teaching / PowerPoint Presentation and Videos:**  [**https://www.coursera.org/learn/web-applications-php**](https://www.coursera.org/learn/web-applications-php)  **Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management** | | | | |
| **Self-study / Do it yourself /:**  **Practice exercises on form handling and server-side scripting with PHP.** | | | | |
| **Experiential Learning Topics:**  **Create a basic form for data submission and handle it using PHP (e.g., feedback form).** | | | | |
| **PBL - Project Based Learning:**  **Develop a small server-side application (e.g., a contact form with email validation and submission).** | | | | |
|  | | | | | |
| **UNIT – V** | **Working with Databases and Web Application Development** | | | | **09 Hours** |
| 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.  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 | | | | | |
| **Pedagogy** | **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**  **Videos:**  [**https://www.coursera.org/learn/web-app**](https://www.coursera.org/learn/web-app) | | | | |
| **Self-study / Do it yourself /:**  **Exercises on creating and manipulating databases using PHP and MySQL.** | | | | |
| **Experiential Learning Topics:**  **Create a database and design a webpage to display its data dynamically.** | | | | |
| **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.** | | | | |

**Text Books:**

1. "HTML and CSS: Design and Build Websites" by Jon Duckett.
2. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins.
3. Achyut Godbole & Atul Kahate, ‖Web Technologies: TCP/IP to Internet Application Architectures‖, McGraw Hill Education publications, ISBN, 007047298X, 9780070472983.
4. Ralph Moseley & M. T. Savaliya, ―Developing Web Applications‖, Wiley publications, ISBN 13 : 9788126538676.

**Reference Books:**

1. Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke.
2. JavaScript: The Good Parts by Douglas Crockford.
3. CSS Secrets: Better Solutions to Everyday Web Design Problems by Lea Ver.
4. Web Technologies- Jeffery C. Jackson, ISBN 978-81-317-1715-8 Pearson 2015.
5. PHP Objects, Patterns, and Practice by Matt Zandstra
6. MySQL Cookbook by Paul DuBois.
7. Advanced PHP Programming - George Schlossnagle- ISBN 0-672-32561-6,2004.

**URLs (Optional) - List of Online Courses**

1. W3Schools HTML, CSS, JavaScript Tutorial: <https://www.w3schools.com/html/>
2. Mozilla Developer Network (MDN) Web Docs - HTML, CSS, JavaScript, DOM: <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML>
3. Project-Based Learning Resources:https://developer.mozilla.org/en-US/docs/Learn

**Contents beyond Syllabus:**

1. Web Essentials
2. Using JavaScript to handle form submission and login events (e.g., onsubmit, onclick)
3. JavaScript Form validations, General Input Validation, Password Validation
4. Storing user data (like a username) temporarily using localStorage or sessionStorage
5. Dynamically updating the content of the webpage, such as displaying a welcome message
6. Redirecting users using window.location

**List of Experiments:**

In this series of assignments, you will create a coffee shop / any other website step by step. Each assignment will focus on a different aspect of the website, covering various HTML elements, CSS, JavaScript, PHP and MySQL concepts.

**Laboratory/Project Assignment Guidelines:**

1. Project Selection:
   * Each student must select a unique project topic for their laboratory assignments.
   * The chosen project topic should align with the concepts covered in the course syllabus.
   * The chosen project topic should be approved by the course coordinator/ subject teacher.
   * Students have the freedom to choose their project topics based on their interests and career aspirations.
   * Project topics may include but are not limited to:
     + E-commerce website
     + Blogging platform
     + Online booking system
     + Content management system (CMS)
     + Discussion forum
     + Social networking platform
     + Task management application
     + Portfolio website
2. Laboratory Assignments:
   * Throughout the course, students will complete laboratory assignments related to their chosen project topic.
3. Evaluation Criteria:
   * The laboratory assignments and the final project will be evaluated based on criteria such as Structure and Semantics, Content Organization, Forms and Inputs, Links and Navigation, Styling and Layout, Design Consistency, Functionality, Code Quality and adherence to project requirements.
   * Students are expected to demonstrate creativity, and a comprehensive understanding of web development principles in their projects.
   * The laboratory assignments based on chosen project topics will be assessed based on several key criteria that reflect both technical proficiency and creative application in web development. These include:

* Structure & Semantics: Proper use of HTML to create a logical, accessible structure with meaningful and semantically correct elements.
* Content Organization: Clear and intuitive organization of content, ensuring ease of navigation and logical flow throughout the site.
* Forms & User Input: Effective implementation of forms and user input elements that are functional, validated, and accessible.
* Links & Navigation: Well-structured navigation and functional links that provide a seamless user experience.
* Styling & Layout: Visually appealing and responsive design, with a well-executed layout that adapts to various screen sizes.
* Design Consistency: Uniformity in design elements, including colors, typography, and spacing, to maintain a cohesive look and feel across the site.
* Functionality: Full functionality of all interactive elements, ensuring a bug-free, smooth experience for users.
* Code Quality & Best Practices: Clean, well-organized, and efficient code that adheres to modern web development best practices and is easy to maintain.

1. Submission and Presentation:
   * The project and project report/journal must be submitted within the specified deadline and should meet the specified requirements outlined by the course coordinator/ subject teacher.

Project Problem Statement-

Design and develop a basic website for a local coffee shop using HTML, CSS, JavaScript, PHP and MySQL. This website will serve as an online presence for the coffee shop, effectively communicating the brand identity, showcasing the menu, providing essential information, and allowing customers to easily get in touch or locate the shop.

The project directory is as follows:

coffee-shop-website/

├── css/

│ └── styles.css

├── html/

│ ├── index.html

│ ├── menu.html

│ ├── about.html

│ └── contact.html

├── js/

│ └── scripts.js

└── images/

| 1. | Create the basic structure of the coffee shop 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:     1. Home page     2. About page     3. Product/Service page     4. Testimonial/review page     5. Contact page     6. Starter blog posts 11. Create and collect design elements   These design elements define your brand personality and help customers feel what your brand represents through the use of:   * 1. Colors   2. Fonts and typography   3. Logos   4. Images and photos |
| --- | --- |
| 2. | HTML   1. Create a detailed home page for the coffee shop website. 2. Create a detailed menu/product page for the coffee shop website, listing all available items categorized appropriately. 3. Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout. 4. Create an about us page that provides detailed information about the coffee shop’s history, mission, and team. 5. Create a contact page that allows customers to easily get in touch with the coffee shop through a form. 6. Design and implement admin/user registration form for the coffee shop website. 7. Design and implement admin/user login form for the coffee shop website. |
| 3. | CSS   1. Enhance the layout of the coffee shop website using CSS Grid for the home page. 2. Use CSS Grid to layout the menu/product items in a structured and style the menu categories with appropriate headings, spacing, separators, images, descriptions, and prices. |
| 4. | CSS   1. 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. 2. Enhance and style the about us page with appropriate margins, paddings, and input field styles. 3. 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. 4. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles. 5. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles. |
| 5. | JavaScript   1. 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.   1. 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)** 2. Develop cart functionality to allow users to add items, update quantities, and remove items. |
| 6. | JavaScript   1. 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. 2. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus)** 3. 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)** 4. 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)** |
| 7. | PHP   1. 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). 2. Implement error handling to notify users of any issues during registration, such as validation errors. 3. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page. |
| 8. | PHP   1. 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). 2. Provide feedback to the user upon successful login, either through a confirmation message or a redirect to a welcome page. 3. Implement error handling to notify users of login failures due to incorrect credentials or other errors. 4. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page. |
| 9. | PHP and MySQL   1. 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. 2. 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. |
| 10. | PHP and MySQL   1. 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. 2. 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. |

# 

## **Experiment No.1**

## 

## Problem Statement:

## Create the basic structure of the Organic Skin Care Product 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:
11. Home page
12. About page
13. Product/Service page
14. Testimonial/review page
15. Contact page
16. Starter blog posts
17. Create and collect design elements
18. These design elements define your brand personality and help customers feel what your brand represents through the use of:
19. Colors
20. Fonts and typography
21. Logos
22. Images and photos

## 

## 

## 

## Objective:

* To implement a clean UI/UX focused on organic and health-conscious aesthetics.
* To enable user authentication through login and registration.
* To provide a shopping cart system using localStorage for item persistence.
* To allow users to view products, add them to cart, and check out.
* To build responsive design for compatibility across devices.

## 

## Theory:

### Project Design and Plan Document for Organic Skin Care Product Website

### 

### 1. Brief Information about the Project

The project involves designing a user-friendly and visually appealing e-commerce website for an organic skin care brand. The goal is to attract health-conscious consumers by showcasing natural and eco-friendly products, sharing customer testimonials, and providing a smooth shopping and contact experience. The website will include secure login and registration features to personalize the user experience and grant access to exclusive features such as saved carts and order history.

### 

### 2. Goals and Deliverables

#### Goals

* Develop an engaging and functional website for a **Skin Care Product Website**.
* Showcase the brand’s story, products, customer reviews, and contact details.
* Enable users to register, log in, and personalize their experience.
* Create a responsive website that works across all devices.

#### Deliverables

* Website Pages:
  + Home Page
  + About Page
  + Products/Services Page
  + Testimonials Page
  + Contact Page
  + Login Page
  + Registration Page
  + Starter blog posts or placeholder for future blogs (optional).
* Core Features:
  + Header and footer with consistent navigation.
  + Functional login and registration system.
  + Responsive design adaptable to mobile, tablet, and desktop.
  + Professional design with appropriate use of colors, fonts, and images.

### 3. Finalize the modules of the project

The Skin Care Product Website will have a modular structure that ensures easy navigation, usability, and maintenance. Each module corresponds to a distinct functionality or page, helping in modular development and integration. Below is a detailed description of the finalized modules:

### Website Modules

#### 1. Home Page Module

* Description:  
  The main page of the website welcomes users and highlights essential features. It sets the tone for the user experience.
* Features:
  + Hero section with the tagline and call-to-action buttons (e.g., "Order Now" or "Offers Available").
  + Overview of featured products or promotions.
  + Navigation menu linking to all website sections (e.g., About, Products, Testimonials, Contact, Login).
  + Footer with contact details, social links, and other information.

#### 2. About Page Module

* Description:  
  Offers visitors a glimpse of the organic skin care products brand’s story, mission, and values.
* Features:
  + Introduction to the organic skin care products history and team.
  + Showcase the brand's principles like quality, sustainability, and customer service.
  + Engaging visuals to reflect the nature’s vibe.

#### 3. Products/Services Page Module

* Description:  
  Displays the organic skin care products offerings in a user-friendly way.
* Features:
  + Categorized product (e.g.,Soaps, Facewash, Serums, Shampoo).
  + Images and details for each item, including price and description.
  + Option for filtering or searching products (future enhancement).

#### 4. Testimonials Page Module

* Description:  
  Shares positive customer reviews and builds trust with new visitors.
* Features:
  + Slider or grid layout showcasing testimonials.
  + Include a field or section for customers to submit their reviews (optional).

#### 5. Contact Page Module

* Description:  
  Enables visitors to get in touch with the organic skin care products team.
* Features:
  + A form for user inquiries (fields: Name, Email, Subject, Message).
  + Embedded map for the physical shop location.
  + Display contact details like phone number and working hours.

#### 6. Login Page Module

* Description:  
  Provides authentication functionality for returning users.
* Features:
  + Login form with fields for Email and Password.
  + "Forgot Password?" link.
  + Redirection to the registration page for new users.

#### 7. Registration Page Module

* Description:  
  Allows new users to sign up for an account.
* Features:
  + Registration form with fields for Name, Email, and Password creation.
  + Terms and conditions acceptance checkbox.
  + Submit button to create an account.

#### 8. Footer Module

* Description:  
  A common footer displayed across all pages.
* Features:
  + Links to Privacy Policy, Terms of Service, and social media pages.
  + Address and basic contact info.

### 

### 4. Define the audience

### Target Audience

a. Health-Conscious Customers Characteristics: Customers focused on organic ingredients, eco-friendly products, and transparent labeling.  
 Needs:

* Highlight the natural and organic ingredients used in the products.
* Include clear nutritional information or benefits for each product.
* Make sure vegan, cruelty-free, and eco-friendly labels are prominent.
* Offer product bundles focused on skincare routines for different skin types.

b. New Users (Unfamiliar Customers) Characteristics: People unfamiliar with the brand who are exploring organic skincare options for the first time.  
 Needs:

* A compelling "About Us" page detailing the brand’s mission, story, and values (e.g., sustainability, cruelty-free, etc.).
* Easy-to-navigate homepage with an introduction to best-sellers or featured items.
* Trust-building elements like reviews, testimonials, and certifications (e.g., organic certification).

c. Online Shoppers Characteristics: Customers who prefer buying skincare products online for convenience.  
 Needs:

* A smooth e-commerce experience with a user-friendly product page and checkout system.
* Integration with secure payment options, and a clear return/refund policy.
* Account creation options for quicker repeat purchases and tracking.

d. Professionals and Remote Workers Characteristics: Consumers who seek quick and effective skincare solutions amidst busy schedules.  
 Needs:

* Quick access to best-sellers or multi-use products (e.g., moisturizers with SPF).
* Options to purchase gift sets or subscription boxes.
* Delivery/pickup options for convenience.

e. Students Characteristics: Younger demographic looking for affordable, effective skincare options.  
 Needs:

* Special promotions or student discounts.
* A section for budget-friendly skincare routines and quick fixes for common skin issues (acne, oily skin, etc.).

f. Tourists and Travelers Characteristics: Visitors looking for local skincare products or souvenirs.  
 Needs:

* A "Travel Essentials" section with small, travel-friendly skincare kits.
* A clear map or location page for any physical store.
* Specialty items or limited-edition products inspired by local ingredients or traditions.

### 

| Audience Segment | Key Features Needed |
| --- | --- |
| Health-Conscious Customers | Categorized product pages highlighting ingredients (e.g., organic, vegan, cruelty-free), with nutritional facts or benefits, and eco-friendly certifications. |
| New Users | A user-friendly interface with intuitive navigation, an engaging "About Us" page, customer testimonials, and easy access to top-rated products. |
| Online Shoppers | Secure login and registration for repeat purchases, a well-organized product page with clear categories (e.g., skincare routines, face oils, moisturizers), and a smooth checkout process. |
| Students | Special promotions or student discounts, budget-friendly skincare bundles or packs, and clear navigation to quick skincare solutions for common concerns. |
| Professionals and Remote Workers | Multi-use skincare products (e.g., moisturizers with SPF), subscription options for skincare routines, and easy online ordering and delivery options. |
| Tourists and Travelers | Travel-sized products or kits for skincare on-the-go, location-based store finder, and regionally inspired, limited-edition products. |
|  |  |

### Why Understanding the Audience is Important

* Helps in creating engaging and relevant content tailored to users’ preferences.
* Enhances the user experience (UX) by addressing specific pain points and ensuring seamless navigation.
* Builds brand trust and attracts loyal customers who resonate with the organic skin care product and mission.
* Leads to targeted marketing campaigns, such as student promotions, subscription offers for enthusiasts, or health-focused messaging.

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

a. Pain Point: Poor Navigation and Cluttered Interface

* **Issue:** Many skincare websites have overly complex designs with too many options, making it difficult for users to find specific products or information.
* **Impact:** Users may abandon the site due to frustration with navigation or cluttered layouts.

b. Pain Point: Limited Product Information

* **Issue:** Many websites don’t provide enough detailed information about their products, such as ingredients, benefits, and suitability for different skin types.
* **Impact:** Customers may leave the site or feel hesitant to purchase due to insufficient product transparency.

c. Pain Point: Weak Engagement Strategies

* **Issue:** Lack of features like personalized recommendations, loyalty programs, or content like skincare tips and product usage guides.
* **Impact:** Missed opportunities to engage users and build customer loyalty.

d. Pain Point: Inefficient Checkout and Payment Process

* **Issue:** Complex checkout processes or limited payment options can lead to abandoned carts.
* **Impact:** Lost sales from customers who are deterred by lengthy or inconvenient purchase flows.

e. Pain Point: Lack of Mobile Optimization

* **Issue:** Non-responsive designs that make it difficult to navigate the site on mobile devices.
* **Impact:** Mobile users may struggle to view products, add items to their cart, or complete purchases.

f. Pain Point: Insufficient Customer Support Information

* **Issue:** Lack of clear contact or support options, such as chatbots or easy-to-find customer service details.
* **Impact:** Customers may feel unsupported, leading to frustration or loss of trust.

g. Pain Point: No Personalization or User Account Features

* **Issue:** Customers cannot save favorite products, track past orders, or receive tailored recommendations.
* **Impact:** Users may not return to the site as often, reducing the opportunity for repeat business.

### Crafting the Ideal Experience:

To address these pain points, the website design and functionality should provide an ideal, user-friendly experience:

a. Intuitive Navigation and Clean Design

* Use a clear layout with a sticky navigation bar.
* Ensure essential pages (Home, Products, About, Blog, Contact, Account) are easy to access.
* Simplify the menu and product categories for easy browsing.

b. Comprehensive Product Information

* Provide detailed product descriptions with high-quality images, ingredients, and benefits.
* Include dietary labels (e.g., vegan, gluten-free) and suitability for different skin types.
* Add customer reviews and ratings for each product.

c. Seamless Online Shopping Experience

* Allow customers to easily browse products and add them to their cart.
* Implement a simple, fast checkout process with multiple secure payment options.
* Offer features like "Subscribe & Save" for recurring orders or bundle offers for skincare routines.

d. Mobile-Responsive Design

* Design the site with a mobile-first approach, ensuring it works well on all devices.
* Optimize the layout for touch interactions, fast loading times, and easy navigation on smartphones and tablets.

e. Customer Engagement Features

* Introduce a loyalty program where customers earn points for purchases or product reviews.
* Create a blog or skincare tips section to keep customers engaged with the brand.
* Offer personalized product recommendations based on browsing or purchasing behavior.

f. Easy Access to Contact and Support

* Include a contact page with options for phone, email, and live chat.
* Provide an FAQ section for common customer queries (e.g., shipping, returns).
* Display physical store location(s) on the homepage or a dedicated location page.

g. Personalization and Account Features

* Allow customers to create accounts to save favorite products, track orders, and receive personalized recommendations.
* Display personalized offers or product suggestions based on browsing history or past purchases.
* Send follow-up emails with relevant promotions for registered users.

### The Ideal User Journey

**Step 1: Visiting the Website** Users land on a clean, welcoming homepage with clear calls-to-action (CTA), guiding them to product categories, blog content, or account features.

**Step 2: Browsing Products** Users can easily browse product categories (e.g., skincare routines, face creams, serums) and apply filters (e.g., vegan, cruelty-free, for oily skin) to narrow down their choices.

**Step 3: Adding Products to Cart** Users can effortlessly add products to their cart and view detailed product information, including reviews and usage tips, before making a purchase.

**Step 4: Checking Out** The checkout process is simple, secure, and offers multiple payment options. Users can review their cart, apply discounts, and complete the purchase with minimal steps.

**Step 5: Accessing Support and Contact Information** Users can easily find contact information or reach out for assistance via live chat, email, or the FAQ section if they have questions.

**Step 6: Engaging with Content** Users can explore blog posts, skincare tips, or product reviews to feel more connected to the brand and enhance their experience.

**Step 7: Building Loyalty** Registered users receive personalized product recommendations, and they can track their rewards points, encouraging them to return for future purchases.

### 6. Visual Design Goals

The design of your organic skincare website should communicate **purity, health, and trust**, while remaining **modern, clean, and user-friendly**.

* **Natural & Calming:** Reflect the freshness and gentle nature of organic skincare.
* **Clean & Minimalistic:** Focus on clarity and simplicity to allow products and content to shine.
* **Trustworthy & Professional:** Inspire confidence in product quality through consistent branding and layout.

### 

#### 1. Defining the Core Visual Elements

#### a. Color Palette

Use soothing, nature-inspired colors that reflect cleanliness and eco-consciousness.

| **Color** | **Hex Code** | **Usage** |
| --- | --- | --- |
| Botanical Green | #6B8E23 | Primary buttons, links, and highlights. |
| Soft White | #F8F8F2 | Background and clean sections. |
| Earth Clay | #A0522D | Accents and contrast areas. |
| Lavender Mist | #E6E6FA | Subtle highlights, especially for women-focused sections. |
| Charcoal Grey | #333333 | Text and UI elements for readability. |

#### b. Typography

* **Primary Font:** *Playfair Display* or *Poppins* – Elegant, modern serif/sans-serif for headings.
* **Secondary Font:** *Open Sans* or *Lato* – Clean and readable for body text.
* **Style Tips:** Use varied weights for emphasis, and soft line spacing for a breathable feel.

#### c. Logo and Branding

* Use elements like a **leaf**, **drop**, or **mortar & pestle** to symbolize natural skincare.
* Include the brand name with elegant typography to build a memorable identity.
* Create a **monochrome variant** for minimalistic headers or footer placement.

#### d. Imagery and Icons

Photography**:**

* High-quality images showing product textures (creams, oils).
* Pictures of nature-based ingredients (aloe vera, herbs, essential oils).
* Lifestyle shots of diverse customers using the products.

Icons:

* Use clean, line-style icons for navigation: skin types, eco-friendly, vegan, shop, contact.
* Use icons next to product features: "Paraben-free", "Cruelty-free", "Dermatologist-tested".

Hero Images:

* A rotating banner showcasing flagship products, offers, or natural ingredients.

#### 2. Applying Visual Design to Pages

#### a. Home Page

* Hero Banner**:** Full-width with overlays for promotions or best-sellers.
* Featured Sections: Quick access to categories like “By Skin Type,” “Routine Kits,” or “New Arrivals”.
* CTA Buttons: Use earthy tones with hover effects.

#### b. About Page

* Use real photos of founders, team, or the manufacturing process.
* Highlight brand values like sustainability, ingredient sourcing, and ethical testing.

#### c. Product Pages

* Product cards with clear photos, price, skin type suitability, and badges (e.g., vegan).
* Hover effects for details or quick “Add to Cart”.

#### d. Testimonials/Reviews Page

* Use real customer reviews in card or slider format.
* Highlight different skin concerns and how your products helped.

#### e. Contact Page

* Include a form, email, phone, and optionally live chat.
* Map embedded if you have a physical store.

#### f. Login/Registration Pages

* Soft, neutral backgrounds.
* Highlight buttons with green or clay tones.
* Clear form inputs with icons for a smooth experience.

#### 3. Layout and Design Hierarchy

* Sticky Navigation Bar: Minimal yet accessible for ease of browsing.
* Banner/Header: Emphasize promotions, seasonal offers, or values.
* Grid Layout: Use cards to organize product categories or testimonials.
* Visual Breaks: Use color or spacing to separate sections meaningfully.
* Prominent CTAs: “Shop Now,” “View Routines,” or “Sign Up & Save” in contrasting tones.

#### 4. Expected Impact of Visual Direction

* Enhanced Engagement: Calming and clean design encourages deeper browsing.
* Trust & Credibility: Transparent, ingredient-focused design builds customer trust.
* Stronger Brand Identity: Nature-focused palette and clean typography build consistency.
* Higher Conversion: Well-placed CTAs and intuitive UX lead to better purchase and signup rates.

### 7. Map out the Project structure

organic\_skincare\_website/

│

├── index.html # Home page

├── about.html # About Us page (brand story, mission, sustainability)

├── products.html # Product listing page (categorized by use/skin type)

├── testimonials.html # Customer testimonials and reviews

├── contact.html # Contact and support page

├── login.html # User login page

├── register.html # User registration page

├── blog.html # Blog page (skincare tips, ingredient highlights)

│

├── assets/

│ ├── css/

│ │ ├── style.css # Global stylesheet

│ │ ├── responsive.css # Responsive design for mobile/tablets

│ │

│ ├── js/

│ │ ├── main.js # Core interactivity (navigation, animations, etc.)

│ │ ├── formValidation.js # Scripts for login/registration form validation

│ │ ├── cart.js # Cart functionality (add/remove/update)

│ │ ├── userGreeting.js # Welcome message, personalized user experience

│ │

│ ├── images/

│ │ ├── logo.png # Brand logo

│ │ ├── hero\_banner.jpg # Hero image for home page

│ │ ├── products/ # Product images (face wash, cream, etc.)

│ │ ├── ingredients/ # Organic ingredients like aloe, lavender

│ │ ├── team/ # Brand team or manufacturing shots

│ │ ├── icons/ # UI icons (vegan, cruelty-free, cart, etc.)

│

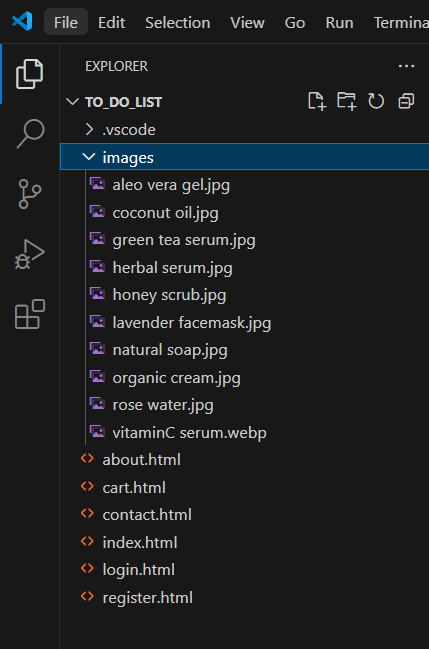
├── fonts/

│ ├── PlayfairDisplay/ # Heading font

│ ├── OpenSans/ # Body font

│

└── README.md # Project overview and setup instructions



### 8. Plan the content for each page

#### 1. Home Page

Purpose: Welcome visitors, introduce the brand, and direct them to key offerings.

Content Plan:

* Header:
  + Logo (left-aligned, nature-inspired).
  + Navigation Menu: *Home, About, Products, Testimonials, Contact*.
  + Login/Sign-Up (top-right corner).
* Hero Section:
  + High-quality banner (natural skincare products or model using products).
  + Tagline: *“Nature’s Touch for Your Skin”*.
  + CTA Button: *“Shop Now”* or *“Explore Our Products”*.
* Intro Section:
  + Short brand description (1-2 lines about purity, sustainability).
  + CTA: *“Learn More About Us”* linking to **About**.
* Featured Section:
  + Grid/carousel showcasing *Best Sellers*, *New Arrivals*, or *Skin Type Kits*.
  + Text: *“Discover What’s Trending in Clean Skincare”*.
* Footer:
  + Quick links, social media, contact info, and possibly newsletter signup.

#### 2. About Page

Purpose: Build emotional connection through your story, values, and team.

Content Plan:

* Header: (Same as home).
* Our Story:
  + Origin of the brand, mission, and what sets it apart.
  + Focus on *organic ingredients*, *eco-conscious packaging*, or *cruelty-free process*.
* Meet the Team:
  + Images and bios of the founders/skincare experts.
* Why Choose Us:
  + Highlight features like *dermatologist-tested*, *vegan-friendly*, *zero parabens*.
* Footer: (Same as home).

#### 3. Products Page

Purpose: Showcase your complete line of skincare products.

Content Plan:

* Header: (Same as home).
* Product Categories:
  + Tabs or filters by category: *Cleansers, Moisturizers, Serums, Masks, Kits*.
* Product Grid:
  + Cards showing product image, name, skin type, key benefits, and price.
* Highlight Section:
  + *Top Rated* or *Customer Favorites* slider.
* CTA:
  + *“Add to Cart”* or *“Log in to Order”*.
* Footer: (Same as home).

#### 4. Testimonials Page

Purpose: Build credibility with real user experiences.

Content Plan:

* Header: (Same as home).
* Customer Reviews:
  + Grid or slider with star ratings, quotes, and profile images.
  + Verified buyer tags for authenticity.
* Submit Your Experience:
  + Simple review submission form (Name, Email, Star Rating, Message).
* Footer: (Same as home).

#### 5. Contact Page

Purpose: Help customers reach out for product help, inquiries, or returns.

Content Plan:

* Header: (Same as home).
* Contact Form:
  + Fields: *Full Name, Email, Subject, Message*.
  + Validation on required fields.
* Store Info/Support:
  + Business hours, support email, and toll-free number.
* Google Map Embed:
  + If you have a physical location.
* Footer: (Same as home).

#### 6. Login Page

Purpose: Allow returning users to access accounts.

Content Plan:

* Form Fields:
  + Email and Password.
* Forgot Password:
  + Link to password recovery flow.
* CTA:
  + *“Don’t have an account? Sign Up Now!”*

#### 7. Registration Page

Purpose: Let new users create an account for personalized shopping.

Content Plan:

* Form Fields:
  + Full Name, Email, Password, Confirm Password.
* Validation:
  + Password strength, email format.
* Submit Button:
  + Registers user and redirects to homepage or profile.
* Footer: (Same as home).

### 9. Add ideas for content, images & layout

#### 1. Home Page

Layout Ideas:

* Header Section:
  + Fixed nav bar: Logo (left), menu links (center), *Login/Register* (right).
* Hero Section**:**
  + Full-width image of a glowing model applying a product or nature-inspired background (leaves, flowers).
  + Overlay tagline: *“Glow Naturally. Feel Beautiful.”*
  + CTA Button: *“Explore Our Products”*.
* Featured Products:
  + Cards showcasing 3-4 products: “Best Seller,” “New Arrival,” “Hydration Must-Have,” etc.
  + Include brief benefits and *“Shop Now”* buttons.

Content Ideas:

* Short intro message: *“Welcome to [Brand Name], your destination for pure, organic skincare.”*
* Promotions: *“Limited Time: 15% off on your first order!”*
* Link to product page.

Image Ideas:

* Banner: Radiant skin, nature-focused imagery.
* Product flat-lays with minimal backgrounds.
* Botanical close-ups (aloe, rose petals, herbs).

#### 2. About Page

Layout Ideas:

* Our Story Section:  
  + Timeline or column format showing brand origins, evolution, and commitment to sustainability.
* Team Section:
  + Founders and skin specialists shown in a clean grid with bios.
* Why Choose Us:
  + Icons + text for *Cruelty-Free, Eco-Friendly, Dermatologist Approved*.

Content Ideas:

* Brand mission: *“We believe skincare should be as pure as nature itself.”*
* Ethical sourcing, ingredient transparency.
* Impact and community initiatives (e.g., plastic-free packaging).

Image Ideas:

* Natural labs, founder portraits, ingredient farms.
* Collage of team and product-making process.

#### 3. Products/Services Page

Layout Ideas:

* Categories Section:
  + Tabs or dropdowns for *Cleansers, Serums, Moisturizers, Masks, Bundles*.
* Product Grid:
  + Image, name, price, description, key benefit icons.
* Highlight Slider:
  + *“Fan Favorites”* or *“Editor’s Picks”*.

Content Ideas:

* Ingredient focus: e.g., *“With soothing chamomile extract”*.
* Skin-type labels: *For oily skin*, *Sensitive-safe*.
* Seasonal kits (e.g., “Winter Glow Bundle”).

Image Ideas:

* Clean product shots with neutral/light backgrounds.
* Close-ups of textures (gel, cream, foam).
* Flat-lays with leaves, stones, eco props.

#### 4. Testimonials/Reviews Page

Layout Ideas:

* Review Carousel:
  + Rotating reviews with customer names, photos, and star ratings.
* Form Section:
  + Submit testimonial: *Name, Skin Type, Product Used, Feedback*.

Content Ideas:

* Verified testimonials from long-time users.
* Pull quotes: *“Cleared my skin in weeks!”* or *“Feels light and smells divine.”*

Image Ideas:

* Customer selfies or soft-toned portraits.
* Screenshots of actual social media feedback.

#### 5. Contact Page

Layout Ideas:

* Form:
  + *Full Name, Email, Subject, Message* with validation.
* Map/Location:
  + If applicable (for physical pickup or store).
* Contact Info:
  + Icons for email, phone, WhatsApp, hours.

Content Ideas:

* Message: *“We’d love to assist with your skincare journey.”*
* Common FAQs (Shipping, Returns, Ingredients).

Image Ideas:

* Calming visual (lavender or water splash).
* Icons for contact modes.

#### 6. Login Page

Layout Ideas:

* Split layout: Left – login form; Right – skin model or product highlight.
* Clean, bordered input fields.

**C**ontent Ideas:

* Fields: *Email, Password* + *“Forgot Password?”* link.
* Message: *“Welcome back, natural beauty!”*
* CTA: *“New here? Register now”*.

Image Ideas:

* Soft neutral background.
* Skincare icons or minimalist textures.

#### 7. Registration Page

Layout Ideas:

* Centered form with gentle animations.
* Use icons for fields (e.g., envelope, lock).

Content Ideas:

* Fields: *Name, Email, Password, Confirm Password*.
* Checkbox: *“Subscribe to tips & offers.”*
* Tagline: *“Join our skincare family and get 10% off your first order!”*

Image Ideas:

* Background featuring a welcome pack or popular kit.
* Icons indicating benefits (discounts, faster checkout, order tracking).

#### 8. Global Visual Design Guidelines

* Colors: *Earthy Greens (#5A7D4F)*, Creamy Whites (#F9F6F1), Clay Browns (#A98467), and Rose Beige (#EED6C4).
* Fonts:
  + Primary: *Playfair Display* (elegant headings).
  + Secondary: *Roboto or Lato* (body text).
* Icons:
  + Minimal, line-style icons (leaf, droplet, sun, jar).
* Imagery:
  + High-resolution lifestyle shots, minimal props, focus on *natural beauty + clean skin*.

### 10. Determine your site structure or Create content for your core website pages:

1. **Home page**
2. **About page**
3. **Product/Service page**
4. **Testimonial/review page**
5. **Contact page**
6. **Starter blog posts**

#### 1. Home Page

Purpose:  
 Introduce the organic skincare brand and guide users to explore products, benefits, and information.

Sections and Content:

Header:

* Logo (with a leaf, flower, or herbal symbol)
* Navigation: Home, About, Products, Testimonials, Contact
* Login/Register buttons

Hero Section:

* Full-width image of natural ingredients or glowing skin
* Tagline: Pure Beauty. Powered by Nature.
* Button: Explore Products or Shop Now

About Teaser:

* Short intro like: “We believe in honest skincare—free from toxins, full of nature’s goodness.”
* Button: Learn More

Best Sellers or Featured Products:

* Grid of 3–4 items like Aloe Gel, Herbal Face Wash, Rose Water Mist
* Short name, image, price, and Add to Cart button

Benefits Section:

* Icons and brief texts: 100% Organic, Cruelty-Free, No Parabens, Eco-Friendly

Customer Love (Testimonial Preview):

* Snippets from reviews with stars
* Button: Read More

Footer:

* Quick links, email, social icons, location
* Newsletter sign-up box

#### 2. About Page

Purpose:  
 Tell the brand story, mission, and eco-friendly values.

Sections and Content:

Brand Story:

* Why and how the brand started
* Founder’s vision and passion for clean beauty

Our Values:

* Clean ingredients, transparent labels, no animal testing
* Eco-friendly packaging, community-driven sourcing

Behind the Ingredients:

* Where ingredients come from (e.g., Himalayan Rose, Neem from local farms)
* Photos of raw materials or preparation

Meet the Team (Optional):

* Founder and skincare experts with photos and short bios

Footer:

* Same as Home Page

#### 3. Products Page

Purpose:  
 Showcase all organic skincare items with categories and cart options.

Sections and Content:

Categories:

* Face Care, Body Care, Hair Care, Wellness

Product Grid:

* Each product with image, name, key benefits, price, and Add to Cart
* Option for filters (by skin type, ingredients, new arrivals

Popular Picks:

* Carousel or “Customer Favorites” section

CTA Section:

* Button: View Cart / Continue Shopping

Footer:

* Same as Home Page

#### 4. Testimonials Page

Purpose:  
 Highlight real customer experiences to build trust.

Sections and Content:

Customer Reviews:

* Grid or carousel of 5-star reviews
* Each includes user name, photo (optional), comment, and product use

Video/Testimonial Highlights:

* If available, video clips or before-after pictures

Submit Your Review:

* Simple form with name, product, comment, and star rating

Footer:

* Same as Home Page

#### 5. Contact Page

Purpose:  
 Let customers reach out with questions or feedback.

Sections and Content:

Contact Form:

* Fields: Name, Email, Subject, Message
* Submit Button

Customer Care Info:

* Email, phone number, response hours

Store Location / Map:

* If you have a physical store or warehouse

FAQs (Optional):

* Shipping, return policy, ingredient safety info

Footer:

* Same as Home Page

#### 6. Blog Page (Optional but Recommended)

Purpose:  
 Educate and engage with skin health and ingredient knowledge.

Suggested Categories:

* Skincare Tips
* Ingredient Spotlight
* DIY Beauty Recipes
* Brand Journey & Sustainability

Each Post:

* Title, intro paragraph, featured image
* Read More link
* Social sharing options

#### 7. Login Page

Purpose:  
 Allow returning users to access their account and orders.

Content:

* Email, password field.
* Forgot Password link
* Login button
* New user? Register now link

Visual:

* Background image with herbal tones or organic products

#### 8. Registration Page

Purpose:  
 Let new users create accounts for a personalized experience.

Content:

* Full Name, Email, Password, Confirm Password
* Checkbox: Subscribe to offers/newsletter
* Register button
* Already have an account? Login

### 11. Create and collect design elements

**These design elements define your brand personality and help customers feel what your brand represents through the use of:**

1. **Colors**
2. **Fonts and typography**
3. **Logos**
4. **Images and photos**

The design of an organic skincare website should reflect purity, natural beauty, sustainability, and trust. Your visitors should feel like they’ve stepped into a serene, botanical sanctuary—safe, honest, and rejuvenating. Here's how each design component helps create and strengthen that connection:

#### 1. Colors

Purpose: To visually communicate cleanliness, natural beauty, and sustainability.

Primary Colors:

* Leafy Green (#88B04B): Represents natural ingredients, growth, and healing. Ideal for CTAs, highlights, and background accents.
* Soft Ivory (#F9F7F3): Clean and elegant; works well for backgrounds and overall layout to convey purity and minimalism.
* Earthy Beige (#DCC9B6): Adds a soft, grounded tone, ideal for section backgrounds or subtle highlights.

Accent Colors:

* Botanical Teal (#4A7C59): A calm, earthy hue—great for headers or promotional banners.
* Floral Rose (#E7A1A1): A delicate, feminine color that can be used in product highlights or callouts (e.g., for rose-based products).

Psychological Impact:  
 This palette evokes calmness, cleanliness, and nature. It reinforces the organic, non-toxic promise of the brand and soothes the user as they explore.

#### 2. Fonts and Typography

Heading Font:

* Playfair Display or Cormorant Garamond (serif): Elegant, refined fonts that communicate beauty and care, suitable for headings like "Our Ingredients" or "Shop Now".

Body Font:

* Lato or Poppins (sans-serif): Clean and highly readable, they maintain a modern and minimal look, ideal for product descriptions and blog content.

Font Weights:

* Use bold for section titles, semi-bold for subheadings, and regular or light weight for paragraphs to keep the site airy and gentle.

Impact:  
 This combination bridges a luxury, boutique feel with clarity and approachability—matching the values of honest, accessible organic skincare.

#### 3. Logo

Logo Design:

* Incorporate a leaf, water droplet, flower petal, or a mortar and pestle symbolizing nature, skincare, and purity.
* Keep the logo clean and minimalist, ideally with line art style.

Color Palette for Logo:

* Use Leafy Green and Soft Ivory, with Botanical Teal for contrast.
* If the brand leans more toward floral or feminine, include Floral Rose accents.

Logo Usage:

* Place at the top left of every page and in the footer.
* Create a scalable version for product packaging, social media, and website favicon.

Impact:  
 The logo acts as your brand’s face—natural, trustworthy, and refined—helping customers immediately connect with your product philosophy.

#### 4. Imagery and Photos

Purpose: To build emotional trust and visually showcase product purity, results, and process.

Product Photography:

* Use high-resolution, natural lighting for all product shots.
* Show products with botanical ingredients beside them (e.g., aloe next to aloe gel).
* Include packaging close-ups to emphasize transparency and quality.

Lifestyle Imagery:

* Use photos of real people using the products in relaxed, natural environments—bathrooms, gardens, or morning routines.
* Highlight diverse skin tones and age ranges to show inclusivity and real beauty.

Behind the Scenes:

* Feature ingredient sourcing, mixing processes, or the team preparing products in small batches.
* Consider images of organic farms, local artisans, or sustainable packaging processes.

Impact:  
 Authentic visuals help visitors trust what they’re putting on their skin. They bring the values of the brand—natural, safe, and effective—into vivid, emotional focus.

#### 5. Interactive Elements and Buttons

Navigation Buttons:

* Use rounded-edge buttons in Leafy Green or Botanical Teal with white text.
* Button Text Examples: “Shop Now,” “Learn More,” “Get Glowing,” “See Ingredients.”

Hover Effects:

* Use soft glows, underline animation, or subtle background fade to indicate interactivity.
* Avoid harsh animations to maintain a gentle, soothing feel.

Icons:

* Use custom or flat-design icons that represent:  
  + A leaf (for natural ingredients)
  + A heart (for customer favorites)
  + A dropper (for serums and oils)
  + A shield (for skin safety)
  + A recycle symbol (for sustainability)

Tooltips and Popups:

* Use for ingredient definitions or quick tips (“What does hyaluronic acid do?”)
* Include newsletter or first-order discount popup—soft tones and a calming overlay background.

Impact:  
 These interactive elements encourage action without overwhelming the user. They guide the journey while reinforcing trust and clarity.

**Conclusion:**

The Organic Skincare Website project serves as a comprehensive exercise in applying core web development principles while promoting health-conscious and sustainable living. By thoughtfully designing and implementing essential pages—such as Home, About, Products/Services, Testimonials, Contact, and Login/Registration—the website delivers a seamless, user-friendly experience tailored to the needs of customers seeking natural skincare solutions.

The use of a harmonious color palette, calming typography, nature-inspired imagery, and consistent branding helps reinforce the identity of the organic skincare brand. This project emphasizes the importance of structured design—from setting clear goals and planning content architecture to creating visually appealing and responsive layouts. Through user-centric features and well-organized information, the site enhances online presence, builds customer trust, and effectively showcases the values of purity, sustainability, and wellness that define the brand.

## 

## 

## 

## 

## 

## **Experiment No.2**

## **Problem Statement:**

2. HTML

1. Create a detailed home page for the organic skin care products website.
2. Create a detailed menu/product page for the organic skin care products website, listing all available items categorized appropriately.
3. Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout.
4. Create an about us page that provides detailed information about the organic skin care products history, mission, and team.
5. Create a contact page that allows customers to easily get in touch with the organic skin care products shop through a form.
6. Design and implement admin/user registration form for the organic skin care products website.
7. Design and implement admin/user login form for the organic skin care products website.

## **Objective:**

To create a Coffee shop webpage using HTML.

## **Theory:**

**Purpose and Importance in E-commerce:** In a competitive e-commerce environment, especially in domains like organic skincare, a well-structured website builds trust and improves usability. With HTML, developers can create visually organized layouts that guide users through their shopping journey—from exploring products to completing a purchase or contacting support.

**Core Functional Pages and Their HTML Role:**

1. **Home Page:**
   * Acts as the landing page and first impression.
   * Displays featured products, promotions, and an overview of the brand.
   * Uses HTML elements like <header>, <nav>, <section>, <article>, <footer> to structure content.
2. **Menu/Product Page:**
   * Lists all available skincare products, categorized by type (e.g., cleansers, moisturizers, serums).
   * Uses <div>, <ul>, <li>, <img>, and <button> tags to display and structure items.
   * Semantic tags and IDs help with filtering and user navigation.
3. **Cart Page:**
   * Allows users to review, modify, or remove selected items before checkout.
   * Tables or lists are used to present items clearly, along with subtotal and total calculations.
4. **About Us Page:**
   * Shares the brand’s story, values, mission, and team background.
   * Builds customer trust and emotional connection through structured content and imagery.
5. **Contact Page:**
   * Contains a form with fields like name, email, subject, and message.
   * Uses <form>, <input>, <textarea>, and <button> elements to collect customer inquiries.
6. **Registration Form (Admin/User):**
   * Gathers user details like name, email, password, and user role.
   * Uses form controls for validation and account creation.
7. **Login Form (Admin/User):**
   * Allows users to securely log in using email/username and password fields.
   * Utilizes <form>, <input type="password">, and login button elements.

## **Code:**

A. Home page:

Code:  
<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Natural Skincare Shop - Home</title>

</head>

<body>

<header>

<h1>Natural Skincare Shop</h1>

</header>

<div class="navbar">

<a href="index.html">Home</a>

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

<a href="about.html">About Us</a>

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

</div>

<div class="top-right">

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

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

</div>

<div class="container">

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\natural soap.jpg" alt="Natural Soap">

<h3>Natural Soap</h3>

<p>$10</p>

<button onclick="addToCart('Natural Soap', 10)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\organic cream.jpg" alt="Organic Cream">

<h3>Organic Cream</h3>

<p>$15</p>

<button onclick="addToCart('Organic Cream', 15)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\herbal serum.jpg" alt="Herbal Serum">

<h3>Herbal Serum</h3>

<p>$20</p>

<button onclick="addToCart('Herbal Serum', 20)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\aleo vera gel.jpg" alt="Aloe Vera Gel">

<h3>Aloe Vera Gel</h3>

<p>$12</p>

<button onclick="addToCart('Aloe Vera Gel', 12)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\coconut oil.jpg" alt="Coconut Oil">

<h3>Coconut Oil</h3>

<p>$18</p>

<button onclick="addToCart('Coconut Oil', 18)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\rose water.jpg" alt="Rose Water Toner">

<h3>Rose Water Toner</h3>

<p>$8</p>

<button onclick="addToCart('Rose Water Toner', 8)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\lavender facemask.jpg" alt="Lavender Face Mask">

<h3>Lavender Face Mask</h3>

<p>$22</p>

<button onclick="addToCart('Lavender Face Mask', 22)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\honey scrub.jpg" alt="Honey Scrub">

<h3>Honey Scrub</h3>

<p>$14</p>

<button onclick="addToCart('Honey Scrub', 14)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\green tea serum.jpg" alt="Green Tea Serum">

<h3>Green Tea Serum</h3>

<p>$25</p>

<button onclick="addToCart('Green Tea Serum', 25)">Add to Cart</button>

</div>

<div class="product">

<img src="C:\Users\Sarika Kene\Desktop\College work\SEM IV\WP\to\_do\_list\images\vitaminC serum.webp" alt="Vitamin C Cream">

<h3>Vitamin C Cream</h3>

<p>$30</p>

<button onclick="addToCart('Vitamin C Cream', 30)">Add to Cart</button>

</div>

</div>

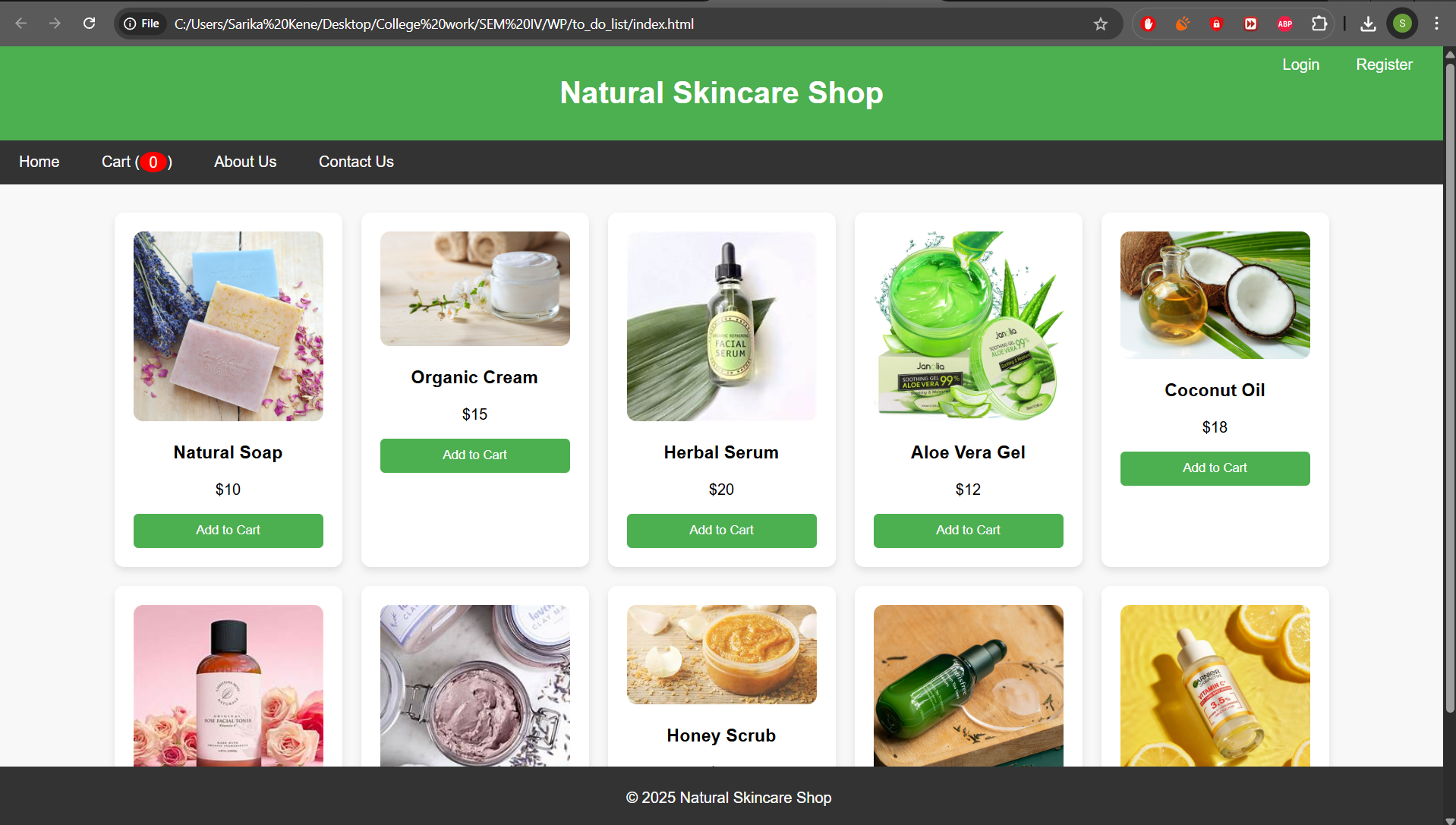
<footer>

<p>© 2025 Natural Skincare Shop</p>

</footer>

## **Output:**

A. Index/Home 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>Natural Skincare Shop - Cart</title>

</head>

<body>

<header>

<h1>Your Cart</h1>

</header>

<div class="navbar">

<a href="index.html">Home</a>

<a href="cart.html">Cart</a>

<a href="about.html">About Us</a>

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

</div>

<div class="cart-container">

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

<div class="cart-total">

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

</div>

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

</div>

<footer>

<p>© 2025 Natural Skincare Shop</p>

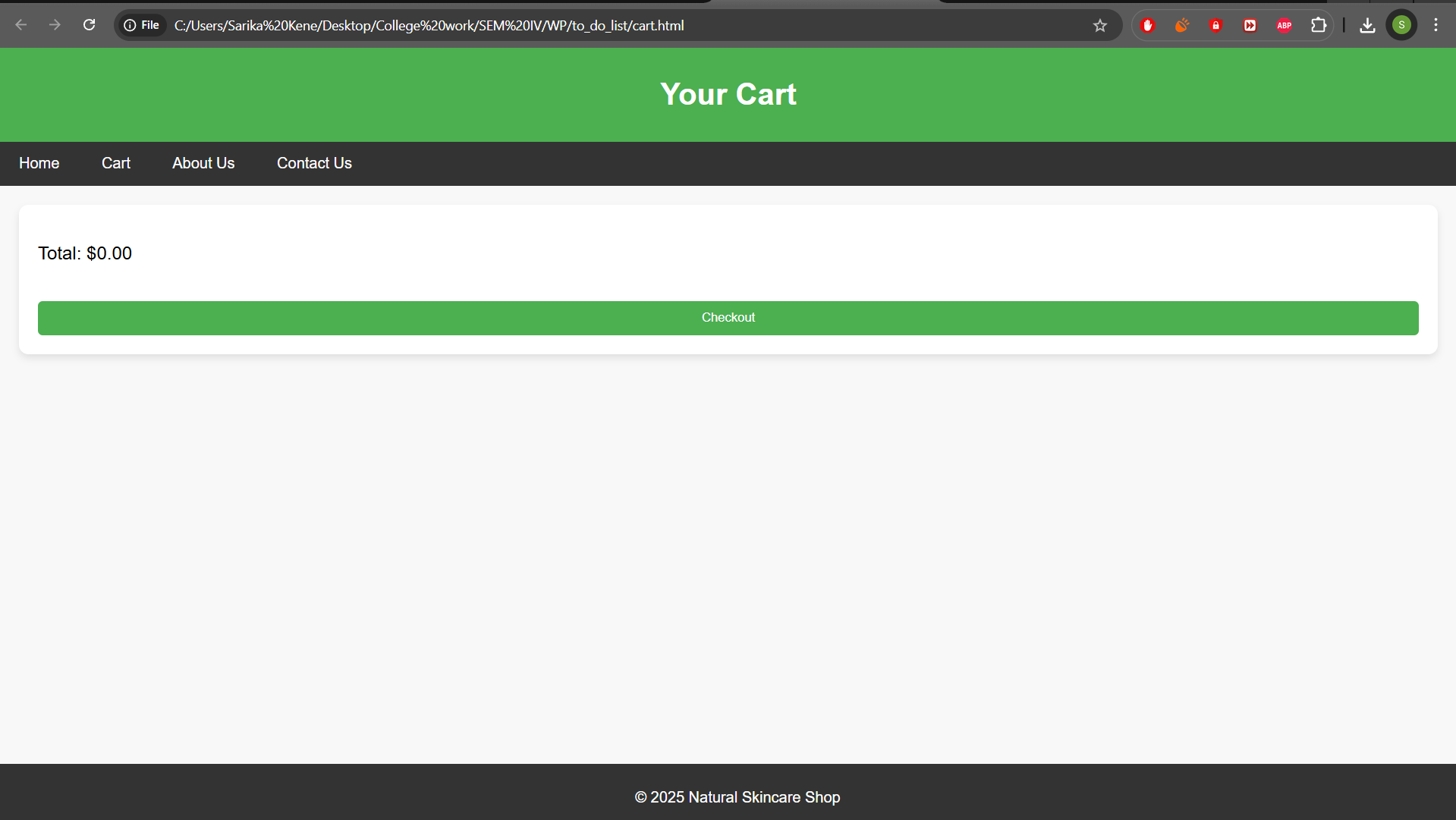
</footer>

</body>

</html>

## **Output:**

C. cart page output:



## **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>Natural Skincare Shop - About Us</title>

</head>

<body>

<header>

<h1>About Us</h1>

</header>

<div class="navbar">

<a href="index.html">Home</a>

<a href="cart.html">Cart</a>

<a href="about.html">About Us</a>

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

</div>

<div class="content">

<h2>Our Story</h2>

<p>We provide all-natural skincare products that are gentle on your skin and the environment.</p>

</div>

<footer>

<p>© 2025 Natural Skincare Shop</p>

</footer>

</body>

</html>

## **Output:**

D. about us page output:



## **Code:**

E. contact 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>Contact Us - Natural Skincare Shop</title>

</head>

<body>

<header>

<h1>Contact Us</h1>

</header>

<div class="form-container">

<h2>Complaint or Suggestion</h2>

<form id="contactForm">

<label for="name">Name:</label>

<input type="text" id="name" required />

<div id="name-error" class="error"></div>

<label for="email">Email:</label>

<input type="email" id="email" required />

<div id="email-error" class="error"></div>

<label for="subject">Subject:</label>

<input type="text" id="subject" required />

<div id="subject-error" class="error"></div>

<label for="message">Message:</label>

<textarea id="message" rows="5" placeholder="Write your complaint or suggestion here..." required></textarea>

<div id="message-error" class="error"></div>

<button type="submit">Submit</button>

</form>

</div>

<footer>

<p>© 2025 Natural Skincare Shop</p>

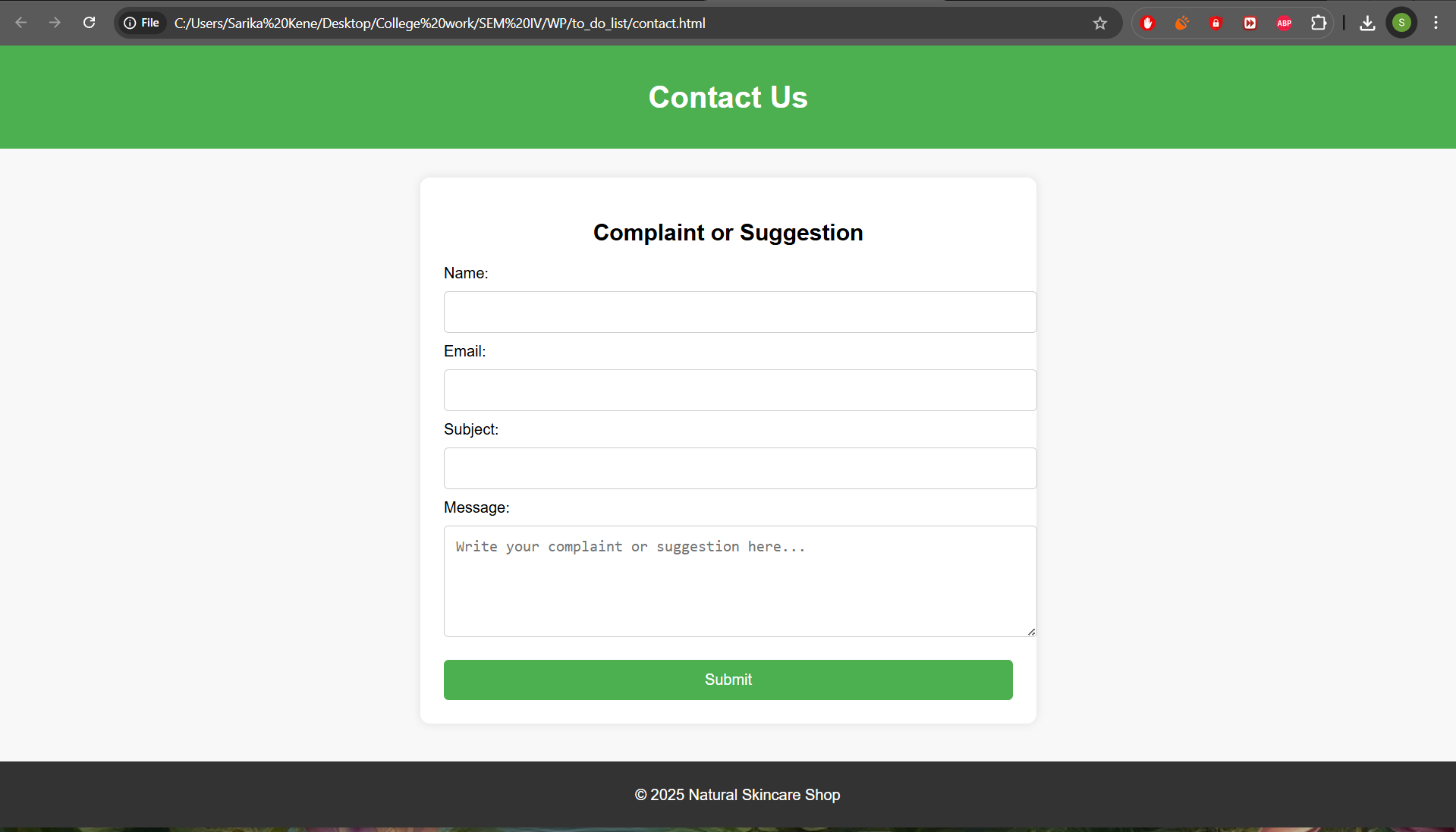
</footer>

</body>

</html>

## **Output:**

E. contact us page output:



## **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>Natural Skincare Shop - Register</title>

</head>

<body>

<header>

<h1>Register</h1>

</header>

<div class="form-container">

<form id="registerForm" action="index.html" method="POST" onsubmit="return validateForm()">

<label for="fullname">Full Name</label>

<input type="text" id="fullname" required />

<div id="name-error" class="error"></div>

<label for="email">Email</label>

<input type="email" id="email" required />

<div id="email-error" class="error"></div>

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

<input type="password" id="password" required />

<div class="show-pass">

<input type="checkbox" id="togglePass"> <label for="togglePass">Show Password</label>

</div>

<div id="password-error" class="error"></div>

<label for="repassword">Re-enter Password</label>

<input type="password" id="repassword" required />

<div class="show-pass">

<input type="checkbox" id="toggleRePass"> <label for="toggleRePass">Show Re-enter Password</label>

</div>

<div id="repassword-error" class="error"></div>

<button type="submit" id="submit-btn" disabled>Register</button>

</form>

</div>

<footer>

<p>© 2025 Natural Skincare Shop</p>

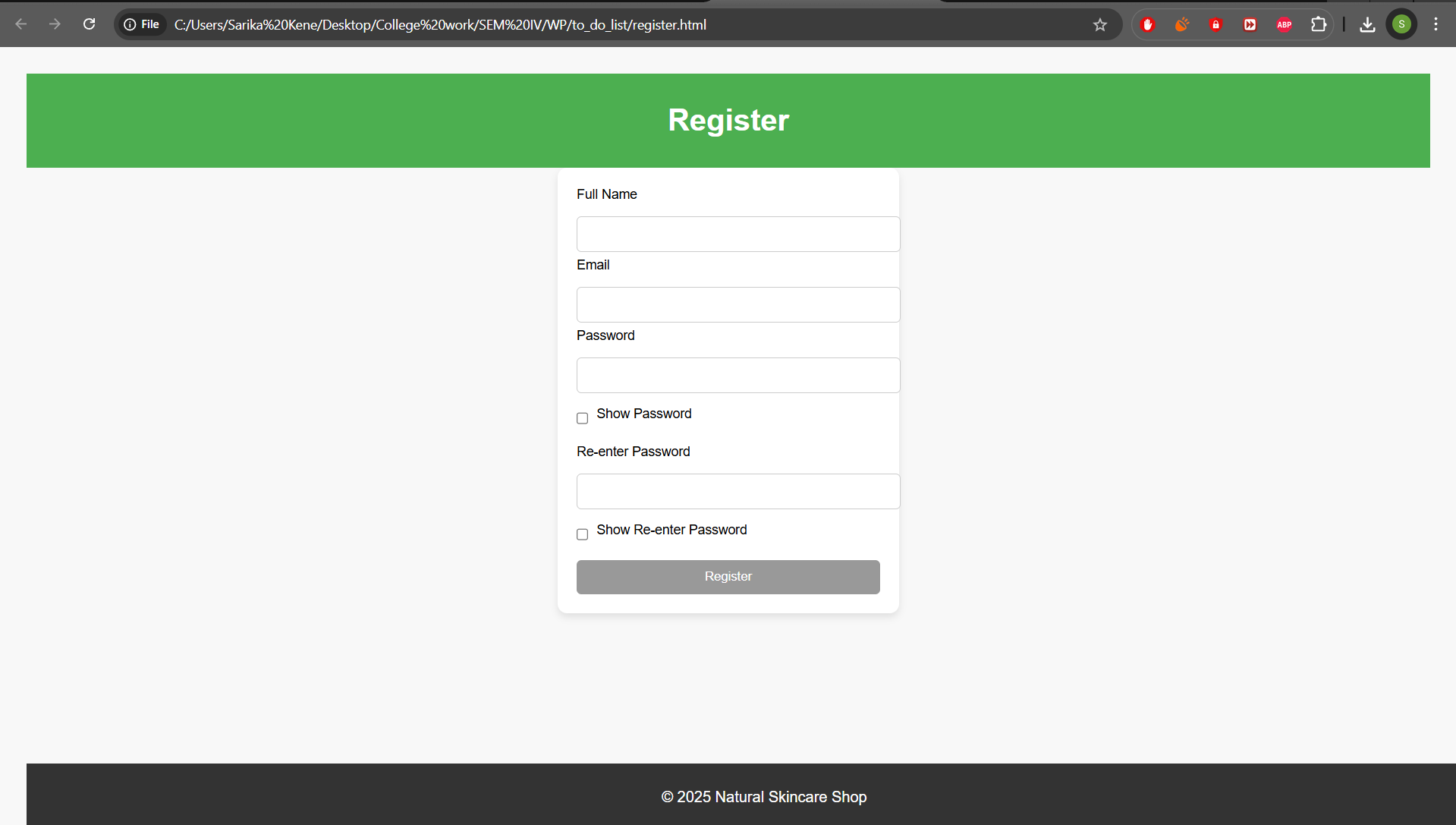
</footer>

</body>

</html>

## **Output:**

F. registration page output:



## **Code:**

G. login page:

Code:  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Login - Natural Skincare Shop</title>

</head>

<body>

<header>

<h1>Login</h1>

</header>

<div class="form-container">

<form id="loginForm">

<label for="username">Username</label>

<input type="text" id="username" required />

<div id="user-error" class="error"></div>

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

<input type="password" id="pass" required />

<button id="togglePassBtn" class="toggle-btn" type="button">Show Password</button>

<div id="pass-error" class="error"></div>

<button type="submit" id="submit-btn" disabled>Login</button>

</form>

</div>

<footer>

<p>© 2025 Natural Skincare Shop</p>

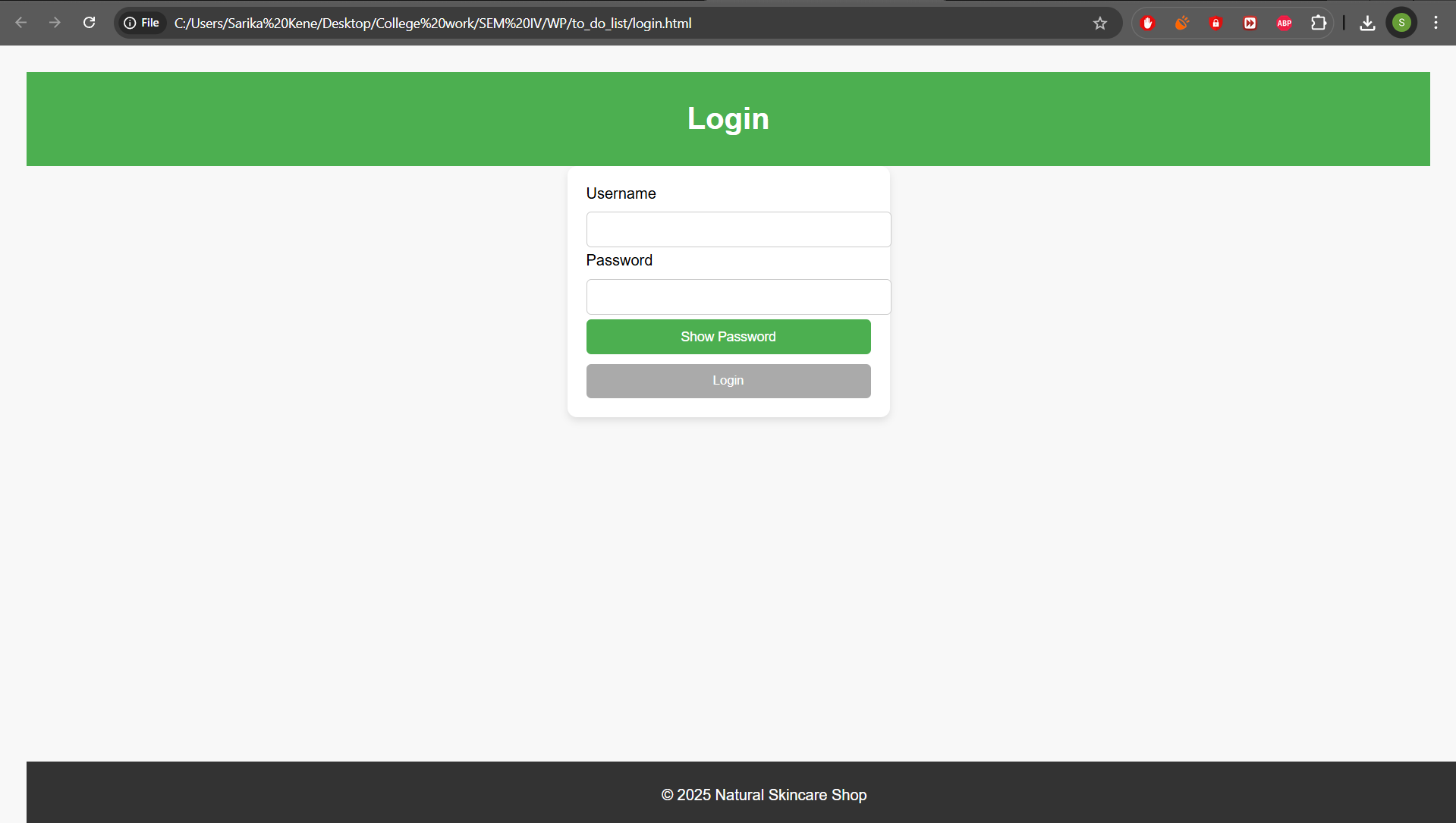
</footer>

</body>

</html>

## **Output:**

G. login page output:



## **Conclusion:**

The *Organic Skin Care Products* website project successfully demonstrates the application of modern web development techniques to build a functional and aesthetically pleasing e-commerce platform. Through the integration of HTML, CSS, and JavaScript, the site enables users to explore organic skincare items, manage their shopping cart, and securely log in or register. This project not only highlights technical skills in front-end and back-end development but also promotes sustainable and healthy living through an engaging digital experience.

## 

## **Experiment No.3**

### 

### Problem Statement:

CSS

1. Enhance the layout of the Organic Skin Care Products website using CSS Grid for the home page.
2. Use CSS Grid to layout the menu/product items in a structured and style the menu categories with appropriate headings, spacing, separators, images, descriptions, and prices.

### Theory:

CSS Theory for Enhancing the Layout of an Organic Skin Care Products Website using CSS Grid.

### Introduction to CSS Grid

* CSS Grid Layout is a two-dimensional system that allows developers to control both rows and columns, making it ideal for complex layouts such as e-commerce websites. It helps create clean, consistent, and adaptable page designs with ease, especially when displaying product grids and structured conte

### Why CSS Grid for this Website?

For an online organic skincare store, presentation and accessibility are key. CSS Grid allows:

* Product cards to be aligned neatly in a responsive grid.
* Flexible layout sections like "Best Sellers", "New Arrivals", and "Herbal Essentials".
* Proper alignment of product images, names, and prices.
* A seamless experience on both desktop and mobile.

#### 1. Home Page Layout with CSS Grid

**CSS Grid is used to define:**

* A full-width navigation header.
* A featured image or banner.
* A three-column layout for best-selling or new products.
* Customer testimonials in a row.
* Footer with contact details and links.

**Benefits:**

* Organizes sections visually and logically.
* Maintains structure on all screen sizes.
* Reduces dependency on media queries.

**2. Menu/Product Page Layout Using CSS Grid**

The home page showcases products using a responsive grid layout. Categories include:

* Soaps
* Creams
* Serums
* Oils
* Toners

Each product card contains:

* Image
* Title
* Description
* Price
* “Add to Cart” button

**Example CSS Grid Layout for Product Items:**

.products-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

grid-gap: 30px;

padding: 20px;

}

Each product-card inside this grid will have:

* A product image
* A title
* A short description
* Price (highlighted)
* "Add to Cart" button

**Additional Styling Concepts:**

* **Category Headings**: Styled with larger fonts, color backgrounds, or underlines to differentiate sections.
* **Separators**: Thin horizontal lines or borders can visually divide different product categories.
* **Hover Effects**: CSS transitions can enhance interactivity by highlighting cards or changing button styles on hover.
* **Responsive Design**: CSS Grid’s auto-fit and minmax() features allow the grid to adapt automatically to screen size, removing the need for complex media queries.

**Mobile Responsiveness with CSS Grid**

One of CSS Grid’s biggest strengths is its **responsive adaptability**. The grid-template-columns property with auto-fit ensures that items stack or spread out based on available screen space.

**Benefits for mobile users:**

* Grid automatically collapses to 1 or 2 columns
* Touch-friendly layout
* Ensures a smooth browsing experience

Code:

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f8f8f8;

}

header {

background-color: #4CAF50;

color: white;

padding: 10px 0;

text-align: center;

}

.navbar {

background-color: #333;

overflow: hidden;

}

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

display: inline-block;

}

.navbar a:hover {

background-color: #ddd;

color: black;

}

.top-right {

position: absolute;

top: 10px;

right: 20px;

}

.top-right a {

color: white;

padding: 8px 12px;

background-color: #4CAF50;

border-radius: 5px;

text-decoration: none;

margin-left: 10px;

}

.top-right a:hover {

background-color: #45a049;

}

.container {

display: flex;

flex-wrap: wrap;

justify-content: center;

margin-top: 20px;

}

.product {

background-color: white;

padding: 20px;

margin: 10px;

border-radius: 10px;

text-align: center;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

width: 200px;

}

.product img {

width: 100%;

border-radius: 10px;

}

.product button {

background-color: #4CAF50;

color: white;

padding: 10px;

border: none;

cursor: pointer;

width: 100%;

border-radius: 5px;

}

.product button:hover {

background-color: #45a049;

}

#cart-count {

background-color: red;

border-radius: 50%;

padding: 2px 10px;

color: white;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 8px;

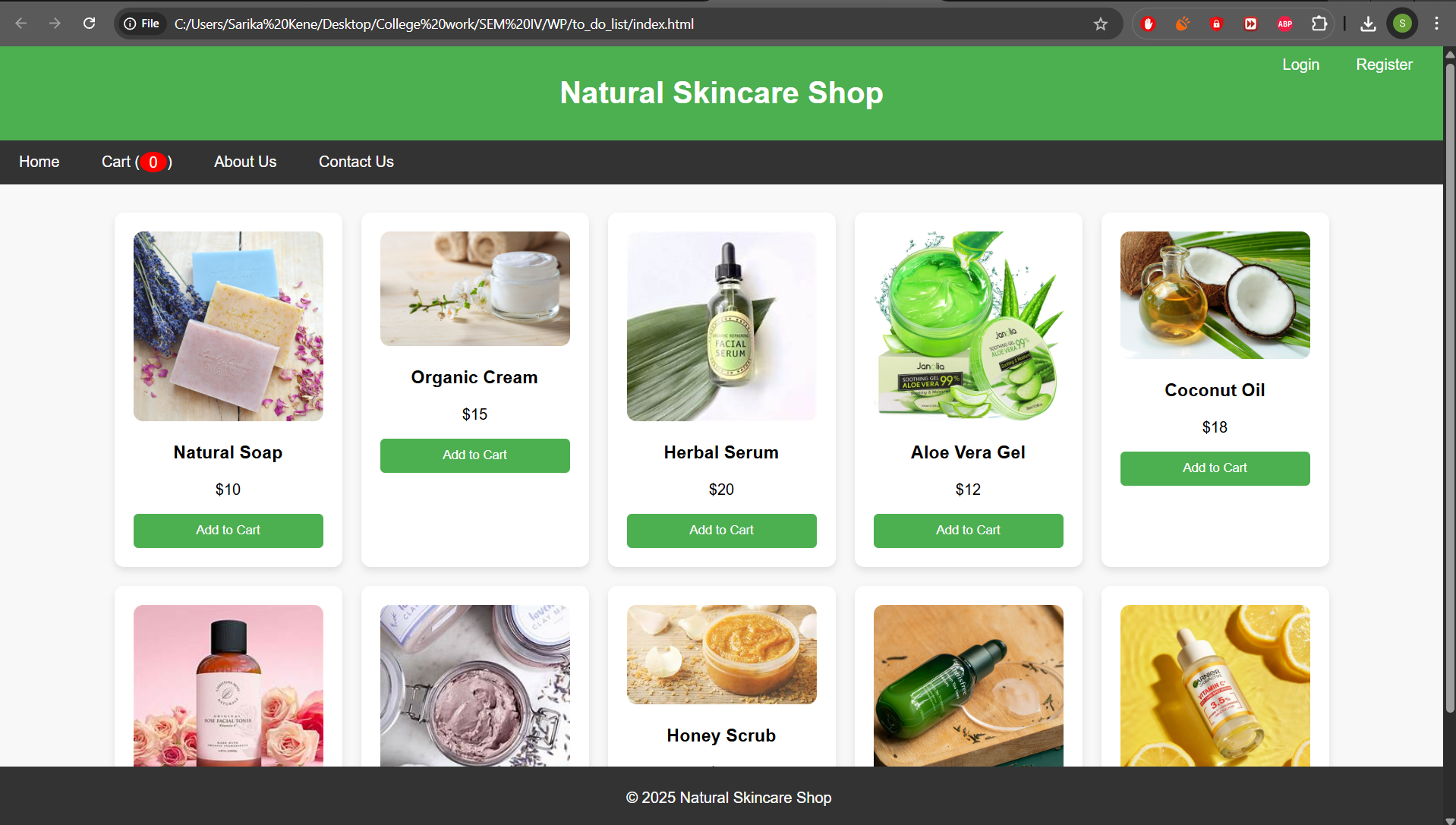
position: fixed;

width: 100%;

bottom: 0;

}

</style>



### Conclusion:

CSS Grid is a powerful layout system that enables the creation of modern, responsive, and visually structured websites. For the *Organic Skin Care Products* platform, CSS Grid has played a key role in enhancing the user experience by simplifying complex layouts and ensuring consistency across devices.

By implementing CSS Grid:

* The home page is cleanly divided into well-defined sections, improving readability and navigation.
* Product listings are neatly arranged, making it easy for customers to explore skincare items at a glance.
* The site adapts fluidly to various screen sizes without the need for excessive media queries.
* Consistent spacing and alignment across elements give the site a polished, professional look.

## **Experiment No. 4**

### Problem Statement:

| CSS   1. 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. 2. Enhance and style the about us page with appropriate margins, paddings, and input field styles. 3. 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. 4. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles. 5. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles. |
| --- |

### CSS Theory: Enhancing and Styling Key Pages in an Organic Skin Care Products Website

### 1. Why CSS Styling Matters in E-commerce Websites

The visual design of a website has a powerful impact on user engagement and perception—especially in a domain like skincare, where trust, cleanliness, and branding are essential. Clean and elegant styling improves readability, reinforces brand identity, and enhances the user experience (UX), especially on key pages such as the cart, contact, login, and registration pages.

Proper CSS styling using spacing, colors, alignment, and responsiveness:

* Makes content easier to read
* Enhances trust and credibility
* Encourages users to complete actions like registering or purchasing
* Provides a consistent and calming visual flow that aligns with wellness and skincare themes

#### Page-wise CSS Styling Theory

#### 1. Cart Page

This page helps users review their selected skincare products before purchase. It must be clear, visually organized, and action-friendly.

**Key Styling Techniques:**

* Use padding between items for visual separation
* Align product name, price, and action buttons cleanly
* Highlight total price with bold fonts and soft green backgrounds
* Consistent font sizes and smooth hover effects for checkout buttons
* Rounded corners and subtle shadows to enhance card-style containers

**Result:** A clean and structured cart page that promotes easy review and encourages checkout completion.

#### 2. About Us Page

This page builds your brand's story and emotional connection with visitors by presenting values like sustainability, purity, and care.

**Key Styling Techniques:**

* Line-height and text-justification for smooth reading
* Section spacing using padding and separators (e.g., "Our Story", "Mission", "Vision")
* Use soft natural color palettes (greens, creams)
* Circular or soft-bordered team images with spacing
* Highlight key quotes or values using callout boxes or styled text blocks

**Result:** A soothing and professional representation of the brand that builds credibility and emotional appeal.

#### 3. Contact Page

The contact page serves as a bridge between users and support. It should be user-friendly and visually inviting.

**Key Styling Techniques:**

* Center-align the form with balanced padding and spacing
* Equal width and rounded edges for input fields
* Clear visual feedback on focus (e.g., green border on active input)
* Use soft hover effects for the "Submit" button
* Show error messages in red and confirm messages in green for clarity

**Result:** An accessible and visually pleasing form that invites user interaction and builds support trust.

#### 4. User Registration Form

A good registration experience reflects the professionalism of your skincare brand. It should feel secure, clean, and easy to fill out.

**Key Styling Techniques:**

* Group inputs (Name, Email, Password) with consistent padding and alignment
* Use labels and placeholders for user guidance
* Highlight input focus and provide instant validation feedback
* Add shadows and soft backgrounds to the form card
* Buttons styled with gradient greens or soft accents, plus hover transitions

**Result:** An aesthetically pleasing and smooth registration process that builds user confidence and reduces errors.

#### 5. User Login Form

This form should be quick, focused, and styled to make users feel welcome and secure.

**Key Styling Techniques:**

* Center the form in the viewport with equal padding
* Minimalistic layout with space between fields
* Focused input fields with clear visual highlights
* Error messages styled in red; success or hints in green
* Use soft background colors with a clean white form container

**Result:** A modern, clean login interface that improves usability and builds trust with returning users.

**Code:**

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f8f8f8;

}

header {

background-color: #4CAF50;

color: white;

padding: 10px 0;

text-align: center;

}

.navbar {

background-color: #333;

overflow: hidden;

}

.navbar a {

color: white;

padding: 14px 20px;

text-decoration: none;

display: inline-block;

}

.navbar a:hover {

background-color: #ddd;

color: black;

}

.cart-container {

margin: 20px;

padding: 20px;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.cart-item {

display: flex;

justify-content: space-between;

padding: 10px;

border-bottom: 1px solid #ddd;

}

.cart-item:last-child {

border-bottom: none;

}

.cart-total {

font-size: 1.2em;

margin-top: 20px;

}

.checkout-button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

border: none;

cursor: pointer;

width: 100%;

border-radius: 5px;

margin-top: 20px;

}

.checkout-button:hover {

background-color: #45a049;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

position: fixed;

width: 100%;

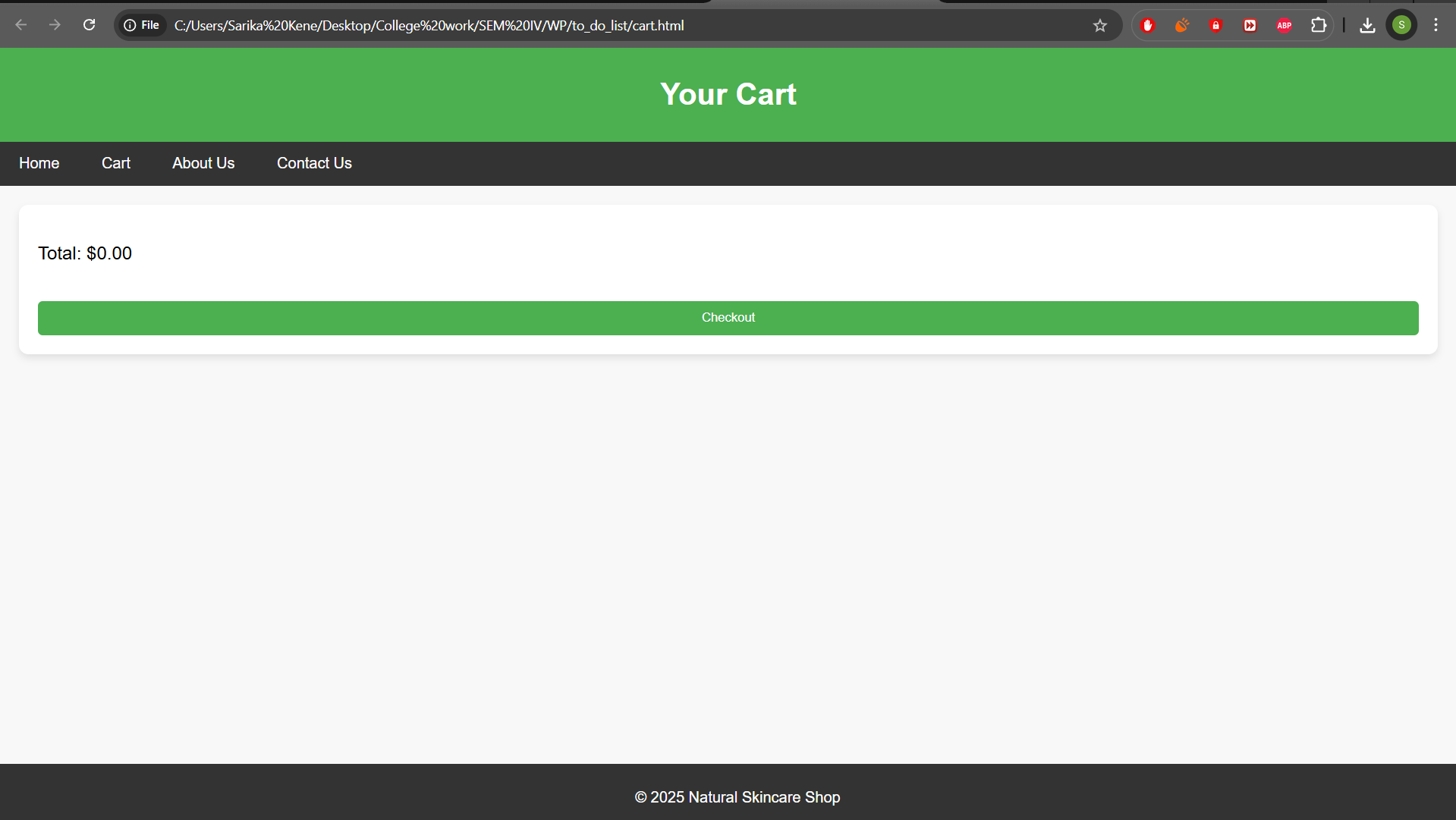
bottom: 0;

}

</style>

**Output:**

cart page  output:



**Code:**

registration page:

<style>

body {

font-family: Arial, sans-serif;

background-color: #f8f8f8;

padding: 20px;

}

header {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 10px;

}

.form-container {

width: 320px;

margin: auto;

padding: 20px;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.form-container input {

width: 100%;

padding: 10px;

margin: 10px 0 5px 0;

border-radius: 5px;

border: 1px solid #ccc;

}

.form-container label {

font-size: 14px;

display: block;

margin-bottom: 5px;

}

.form-container button {

width: 100%;

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.form-container button:disabled {

background-color: #999;

}

.form-container button:hover:not(:disabled) {

background-color: #45a049;

}

.error {

color: red;

font-size: 13px;

}

.success {

border-color: green !important;

}

.fail {

border-color: red !important;

}

.show-pass {

display: flex;

align-items: center;

margin: 5px 0 15px 0;

}

.show-pass input {

margin-right: 8px;

width: auto;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

position: fixed;

width: 100%;

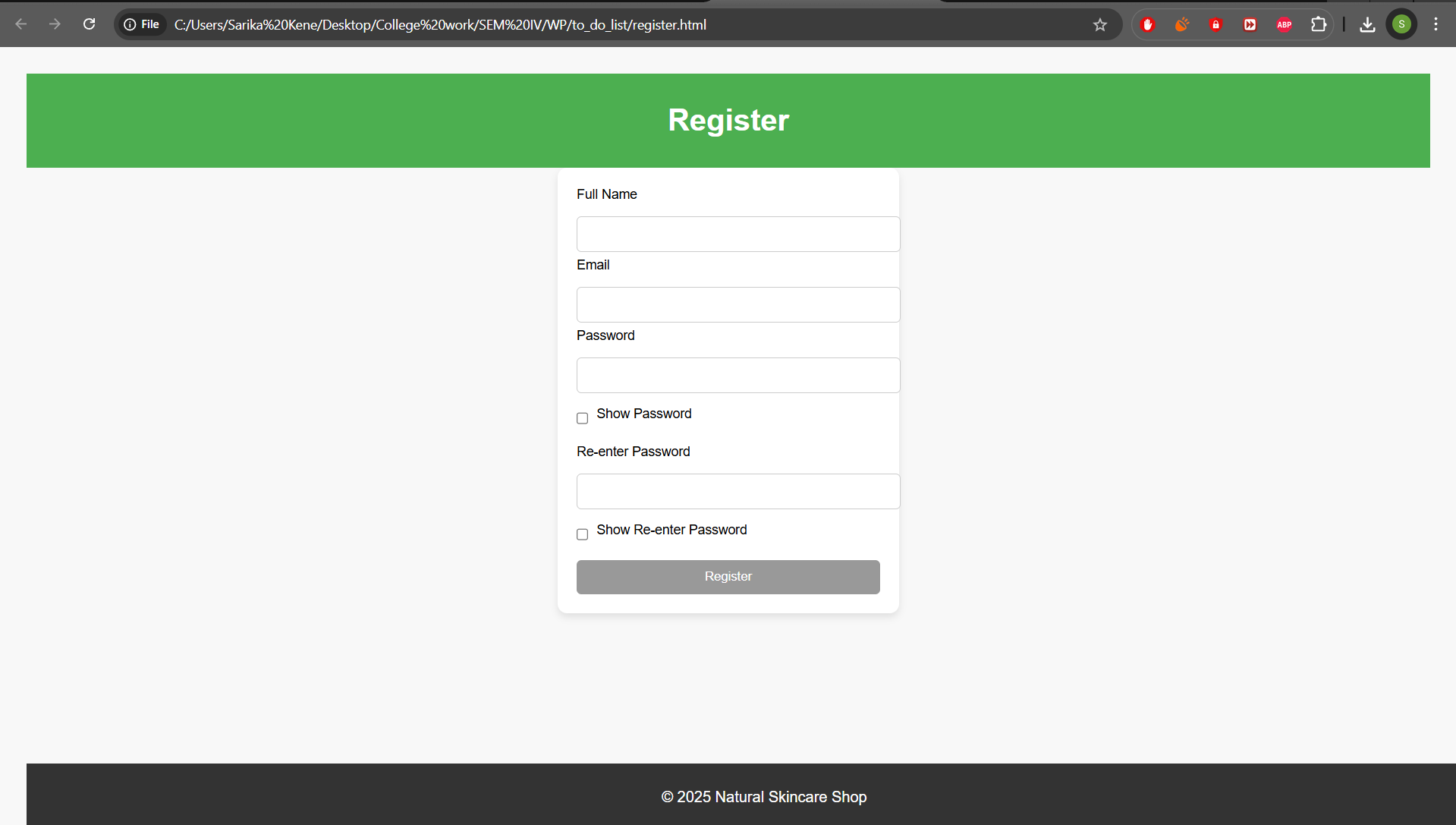
bottom: 0;

}

</style>

**Output:**

registration page  output:



**Code:**

login page:

<style>

body {

font-family: Arial, sans-serif;

background-color: #f8f8f8;

padding: 20px;

}

header {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 10px;

}

.form-container {

width: 300px;

margin: auto;

padding: 20px;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.form-container input {

width: 100%;

padding: 10px;

margin: 10px 0 5px;

border-radius: 5px;

border: 1px solid #ccc;

}

.form-container button {

width: 100%;

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.form-container button:disabled {

background-color: #aaa;

}

.form-container button:hover:not(:disabled) {

background-color: #45a049;

}

.error {

color: red;

font-size: 14px;

}

.success {

border-color: green !important;

}

.fail {

border-color: red !important;

}

.toggle-btn {

background-color: #ddd;

border: none;

padding: 5px 10px;

margin-bottom: 10px;

border-radius: 5px;

cursor: pointer;

font-size: 14px;

display: block;

margin-left: auto;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

position: fixed;

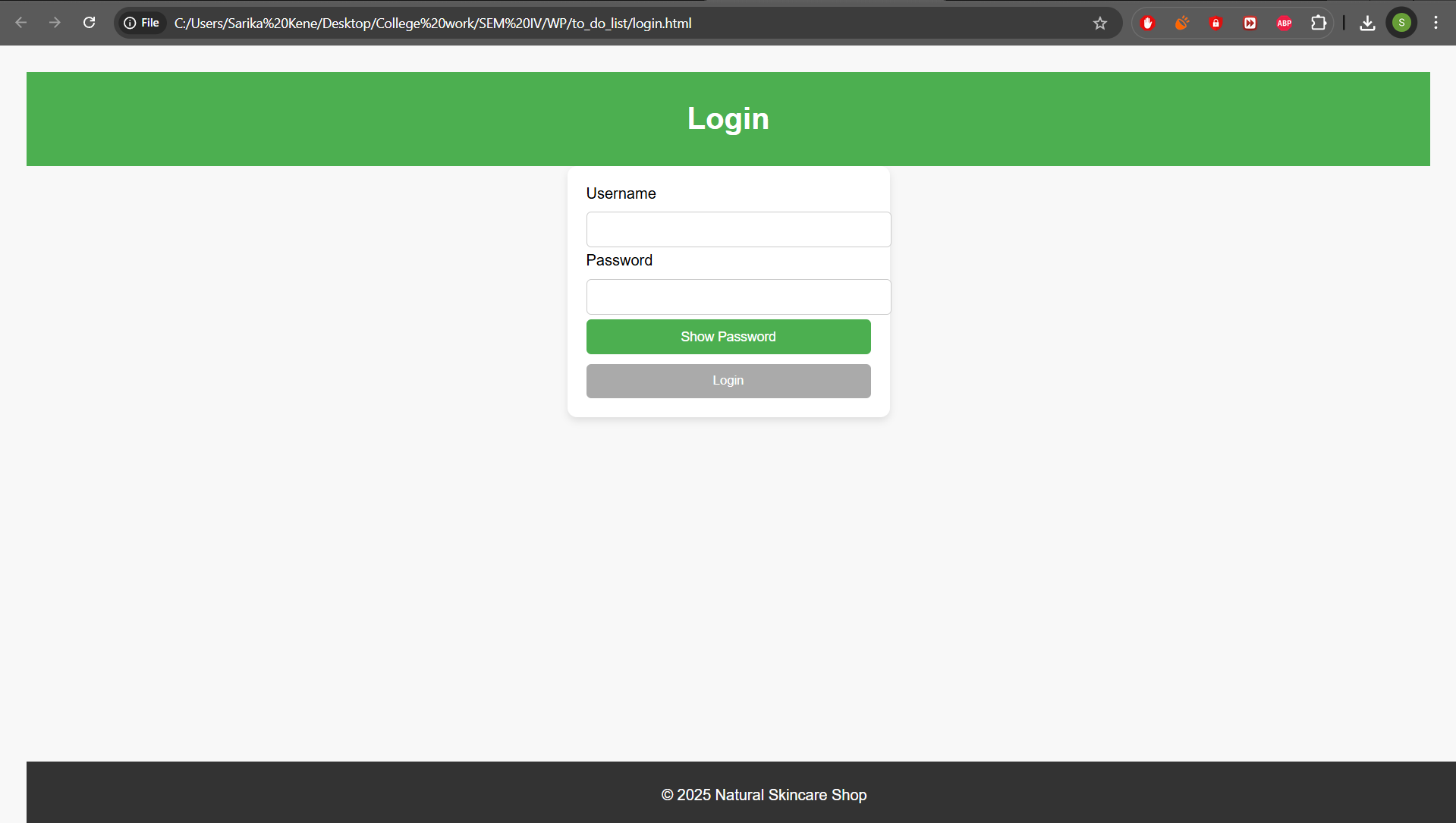
width: 100%;

bottom: 0;

}

</style>

**Output:**

login page  output:

### 

### Conclusion:

The visual and functional success of an e-commerce platform—especially one focused on organic skincare—relies heavily on thoughtful CSS styling. By applying appropriate margins, paddings, input field enhancements, and responsive layouts, the site achieves:

* A smoother and more enjoyable user experience (UX)
* Clearer readability and intuitive navigation
* A polished and trustworthy brand presentation
* Increased user engagement and higher conversion rates

Each page—whether it’s the cart, about us, contact, login, or registration—plays a vital role in the customer journey. Proper styling not only improves functionality but also conveys your brand’s values of care, purity, and professionalism.

In modern web design, CSS is not just about visual appeal—it’s a strategic tool for creating immersive and accessible user experiences.

## 

## 

## 

## 

## 

## 

## 

## 

## **Experiment No.5**

### Problem Statement:

| JavaScript   1. Implement user registration and login forms for the Organic Skin Care Products 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.   1. 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)** 2. Develop cart functionality to allow users to add items, update quantities, and remove items. |
| --- |

### JavaScript Theory:

User Registration, Login, Validation, and Cart Functionality

### Theory:

In modern web development, client-side scripting using **JavaScript** plays a vital role in delivering responsive, interactive, and seamless user experiences. For an e-commerce platform such as the **Organic Skin Care Products** website, JavaScript is essential for implementing key functionalities like user registration, login, form validation, and shopping cart management. These features ensure that users can engage with the platform easily, securely, and efficiently.

#### 1. User Registration and Login Forms

These forms serve as gateways to personalized user experiences by enabling account creation and secure access.

#### Registration Form

Users sign up by entering details such as full name, email, password, and confirming their password. JavaScript ensures the form is interactive and secure by:

* Preventing empty field submission
* Validating email format using regular expressions
* Checking password strength (length, characters, symbols)
* Ensuring password and confirm password fields match
* Providing real-time feedback to users for corrections

#### Login Form

Users access their account using registered credentials. JavaScript enhances the login process by:

* Checking for empty input fields
* Validating email format before authentication
* Matching credentials with stored user data (e.g., in localStorage)
* Redirecting to the home page or dashboard upon successful login
* Displaying error messages in case of incorrect login attempts

#### 2. JavaScript Form Validations

JavaScript is used to validate forms on the client-side before any data is sent to the server or stored. This improves UX by delivering immediate feedback and reducing server load.

**Common Validations Include:**

* Ensuring all required fields are completed
* Validating email format (/^[^ ]+@[^ ]+\.[a-z]{2,3}$/)
* Enforcing password rules (length, character types)
* Checking password match in registration
* Highlighting errors and confirming correct inputs visually

Validation also improves user trust and minimizes form submission errors, especially on the login and registration pages of the skincare site.

#### 3. Cart Functionality

The **shopping cart** is a core feature that enables users to select and review organic skincare products before making a purchase.

**JavaScript Cart Features:**

* Dynamically add products to the cart from the homepage
* Store cart data as an array of objects (e.g., { product: "Aloe Vera Gel", price: 12 })
* Update cart total in real-time as items are added or removed
* Persist cart state using localStorage, allowing the cart to retain items across sessions
* Modify quantity, calculate totals, and remove items using event-driven logic
* Reflect changes visually on the cart page using DOM manipulation

By utilizing JavaScript, the skincare website provides a fluid and engaging shopping experience that closely resembles modern commercial platforms.

Code:

Registration page code:

<script>

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

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

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

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

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

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

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

let validName = false, validEmail = false, validPass = false, validRePass = false;

fullname.addEventListener('input', () => {

if (fullname.value.trim().length >= 3) {

validName = true;

fullname.classList.add('success');

fullname.classList.remove('fail');

document.getElementById('name-error').textContent = '';

} else {

validName = false;

fullname.classList.add('fail');

fullname.classList.remove('success');

document.getElementById('name-error').textContent = 'Full name must be at least 3 characters.';

}

enableSubmit();

});

email.addEventListener('input', () => {

const pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

if (pattern.test(email.value)) {

validEmail = true;

email.classList.add('success');

email.classList.remove('fail');

document.getElementById('email-error').textContent = '';

} else {

validEmail = false;

email.classList.add('fail');

email.classList.remove('success');

document.getElementById('email-error').textContent = 'Enter a valid email address.';

}

enableSubmit();

});

password.addEventListener('input', () => {

const passPattern = /^(?=.\*[0-9])(?=.\*[!@#$%^&\*])[a-zA-Z0-9!@#$%^&\*]{6,16}$/;

if (passPattern.test(password.value)) {

validPass = true;

password.classList.add('success');

password.classList.remove('fail');

document.getElementById('password-error').textContent = '';

} else {

validPass = false;

password.classList.add('fail');

password.classList.remove('success');

document.getElementById('password-error').textContent =

'Password must be 6–16 chars, include a number and special character.';

}

checkRePassword();

enableSubmit();

});

repassword.addEventListener('input', () => {

checkRePassword();

enableSubmit();

});

function checkRePassword() {

if (repassword.value === password.value && password.value !== '') {

validRePass = true;

repassword.classList.add('success');

repassword.classList.remove('fail');

document.getElementById('repassword-error').textContent = '';

} else {

validRePass = false;

repassword.classList.add('fail');

repassword.classList.remove('success');

document.getElementById('repassword-error').textContent = 'Passwords do not match.';

}

}

function enableSubmit() {

submitBtn.disabled = !(validName && validEmail && validPass && validRePass);

}

function validateForm() {

return validName && validEmail && validPass && validRePass;

}

togglePass.addEventListener('change', () => {

password.type = togglePass.checked ? 'text' : 'password';

});

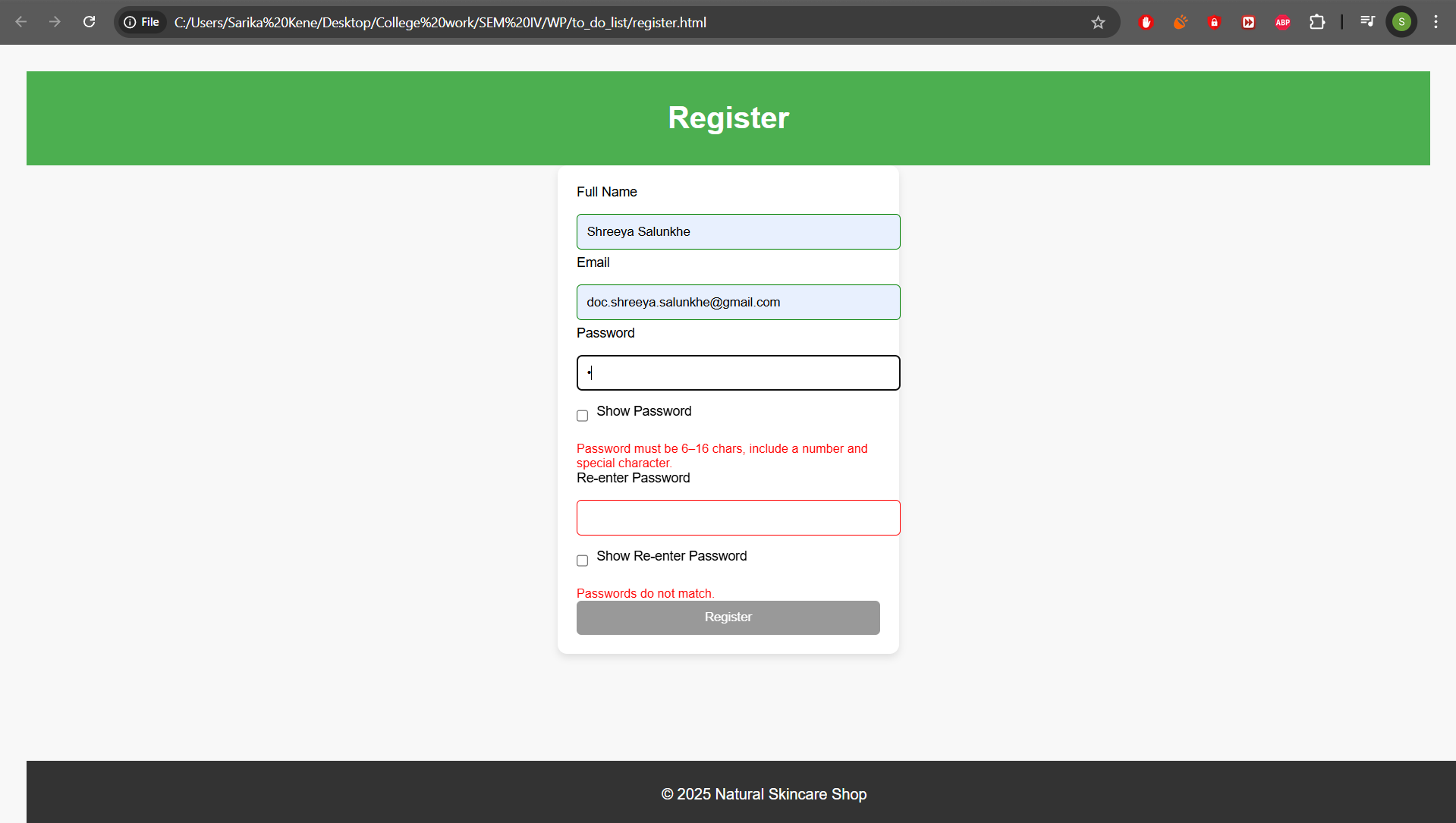
toggleRePass.addEventListener('change', () => {

repassword.type = toggleRePass.checked ? 'text' : 'password';

});

</script>

Output:



Login Page Code:

<script>

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

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

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

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

const userError = document.getElementById('user-error');

const passError = document.getElementById('pass-error');

let validUsername = false;

let validPass = false;

username.addEventListener('input', () => {

if (username.value.length >= 5) {

validUsername = true;

username.classList.add('success');

username.classList.remove('fail');

userError.textContent = '';

} else {

validUsername = false;

username.classList.add('fail');

username.classList.remove('success');

userError.textContent = 'Username must be at least 5 characters long.';

}

checkValidity();

});

pass.addEventListener('input', () => {

const pattern = /^(?=.\*[0-9])(?=.\*[!@#$%^&\*])[A-Za-z0-9!@#$%^&\*]{6,16}$/;

if (pattern.test(pass.value)) {

validPass = true;

pass.classList.add('success');

pass.classList.remove('fail');

passError.textContent = '';

} else {

validPass = false;

pass.classList.add('fail');

pass.classList.remove('success');

passError.textContent = 'Password must be 6–16 characters with at least 1 number and 1 special character.';

}

checkValidity();

});

function checkValidity() {

submitBtn.disabled = !(validUsername && validPass);

}

togglePassBtn.addEventListener('click', () => {

const type = pass.getAttribute('type') === 'password' ? 'text' : 'password';

pass.setAttribute('type', type);

togglePassBtn.textContent = type === 'text' ? 'Hide Password' : 'Show Password';

});

document.getElementById('loginForm').addEventListener('submit', function(e) {

e.preventDefault();

if (validUsername && validPass) {

alert('Login successful!');

window.location.href = 'index.html';

} else {

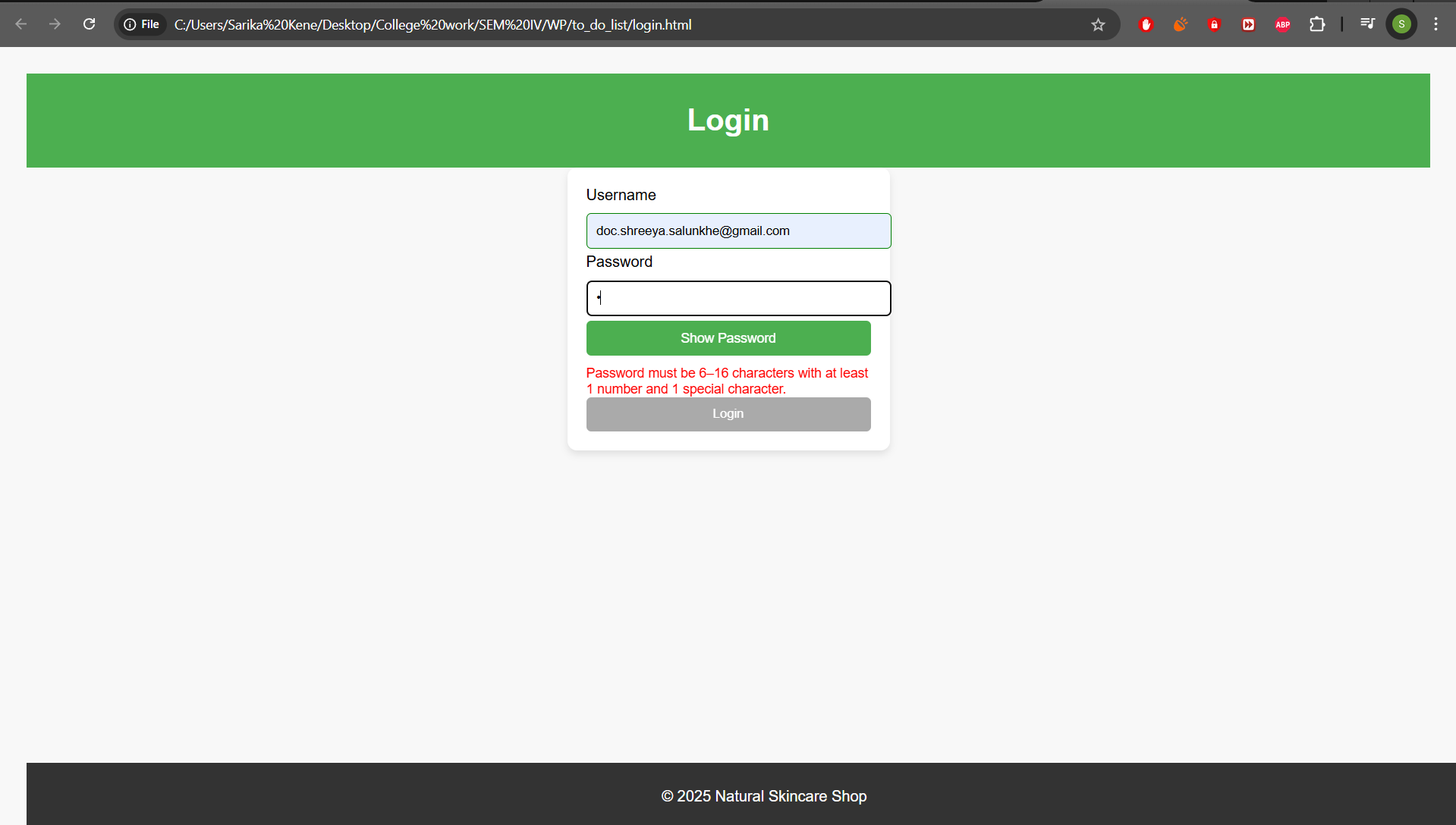
alert('Please check your credentials.');

}

});

</script>

Output:



### Conclusion

Implementing registration, login, validation, and cart functionality using JavaScript is essential for any user-focused e-commerce platform. These features significantly improve usability, ensure smoother interactions, and provide a seamless shopping experience.

For the *Organic Skin Care Products* website, JavaScript enables users to securely register and log in, receive instant form validation feedback, and manage their cart effortlessly. These functionalities establish a strong foundation for future enhancements such as personalized product recommendations, order history, wish lists, and secure checkout systems—making the site more dynamic and user-friendly.

## **Experiment No.6**

### Problem Statement:

| JavaScript   1. 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. 2. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus)** 3. 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)** 4. 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

Home page code:

<script>

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

const userGreeting = document.getElementById('user-greeting');

const loginLink = document.getElementById('login-link');

const logoutLink = document.getElementById('logout-link');

function updateCartCount() {

const cartCount = cart.length;

document.getElementById('cart-count').textContent = cartCount;

localStorage.setItem('cart', JSON.stringify(cart));

}

function checkUser () {

const username = localStorage.getItem('username');

if (username) {

userGreeting.textContent = `Hello, ${username}`;

loginLink.style.display = 'none';

logoutLink.style.display = 'inline';

} else {

userGreeting.textContent = '';

loginLink.style.display = 'inline';

logoutLink.style.display = 'none';

}

}

function logout() {

localStorage.removeItem('username');

checkUser ();

}

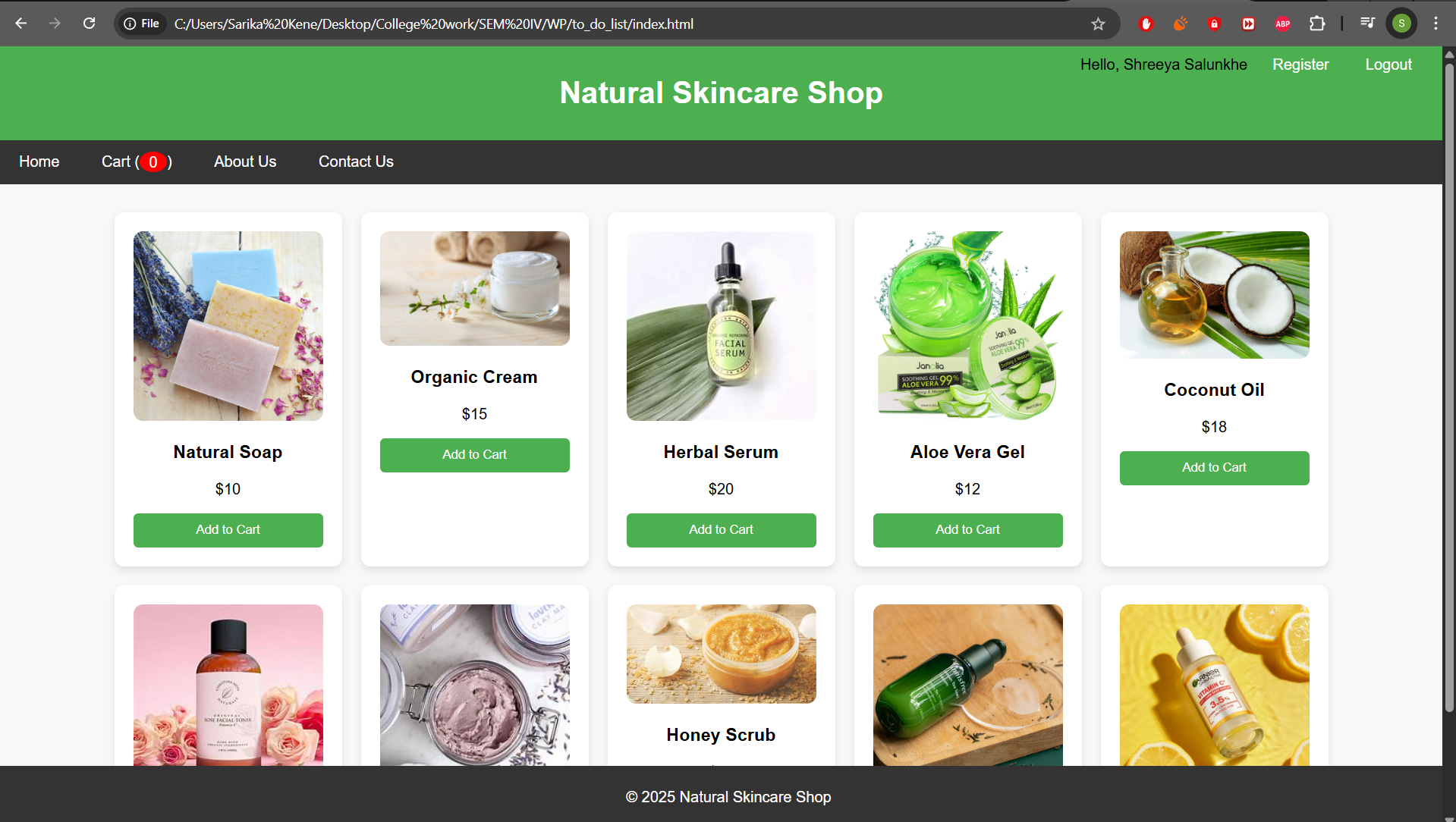
logoutLink.addEventListener('click', logout);

updateCartCount();

checkUser ();

</script>

Output:



### Conclusion

Using JavaScript in combination with the Web Storage API (localStorage / sessionStorage) greatly enhances the user experience and interactivity of web applications. For the *Organic Skin Care Products* website, implementing persistent login and cart functionality ensures users enjoy a smooth and uninterrupted shopping journey.

By storing login states and cart data locally:

* Users remain logged in across sessions without re-authentication
* Cart items are preserved even after refreshing or revisiting the site
* The website feels faster, smarter, and more user-centric

These features emulate real-world functionality found in professional e-commerce platforms, making them valuable additions to academic projects and personal portfolios. Ultimately, mastering these techniques empowers developers to create responsive, dynamic, and customer-friendly web applications.

## **Experiment no.7**

### Problem Statement:

| PHP   1. Develop a PHP script to handle user registration for the Organic Skin Care Products website. The script should accept input from users for their name, email address, password, etc. (all required fields for registration). 2. Implement error handling to notify users of any issues during registration, such as validation errors. 3. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page. |
| --- |

### Theory:

### PHP User Registration Script – Organic Skin Care Products Website

#### A. Develop a PHP Script to Handle User Registration

User registration is a fundamental component of any web-based system, especially for an e-commerce platform like the **Organic Skin Care Products** website. PHP is commonly used for backend operations such as processing form data, validating input, and interacting with databases like MySQL.

In this system, the **registration form** captures user details such as full name, email address, and password. Upon submission, the PHP script validates the input, securely hashes the password, and stores the data in a database.

## 

#### Core PHP Elements Used:

1. **Form Handling:** PHP uses $\_POST to collect form data after submission.
2. **Validation:** Checks are implemented to ensure:  
   * No field is empty
   * Email format is valid
   * Password meets minimum security requirements
3. **Password Hashing:** Uses password\_hash() to encrypt the password before storing it, ensuring security against data breaches.
4. **Database Interaction:** MySQLi or PDO is used to connect and execute SQL queries that insert new user records.

#### B. Implement Error Handling for User Input

To ensure data quality and usability, the system checks:

* That all form fields are filled
* Email format is valid (filter\_var($email, FILTER\_VALIDATE\_EMAIL))
* Password and confirm password match (if applicable)

**Error Handling Includes:**

* Displaying appropriate messages for each validation failure
* Preventing submission if any required field is invalid
* Using server-side checks to supplement front-end JavaScript validation

#### C. Provide User Feedback After Successful Registration

Once registration is successful:

* The user is notified with a **success message**
* Optionally, the user can be **redirected to the login page** for authentication
* A confirmation message such as: "Registration successful! Please log in to continue." can be displayed

### Benefits of Using PHP for Registration in Skincare E-commerce:

* Handles data securely on the server-side
* Allows for custom validation and feedback messages
* Can be easily integrated with login, order history, and user dashboards
* Supports secure practices like password hashing and SQL injection protection

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") {

// Sanitize and validate inputs

$name = trim($\_POST["name"]);

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

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

// Validation checks

if (empty($name)) $errors[] = "Name is required.";

if (empty($email) || !filter\_var($email, FILTER\_VALIDATE\_EMAIL)) $errors[] = "A valid email address is required.";

if (empty($password) || strlen($password) < 6) $errors[] = "Password must be at least 6 characters.";

// If no errors, store the 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 style='color:green;'>Registration successful. <a href='login.html'>Click here to login</a>.</p>";

} else {

echo "<p style='color:red;'>Error: " . $stmt->error . "</p>";

}

$stmt->close();

} else {

foreach ($errors as $error) {

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

}

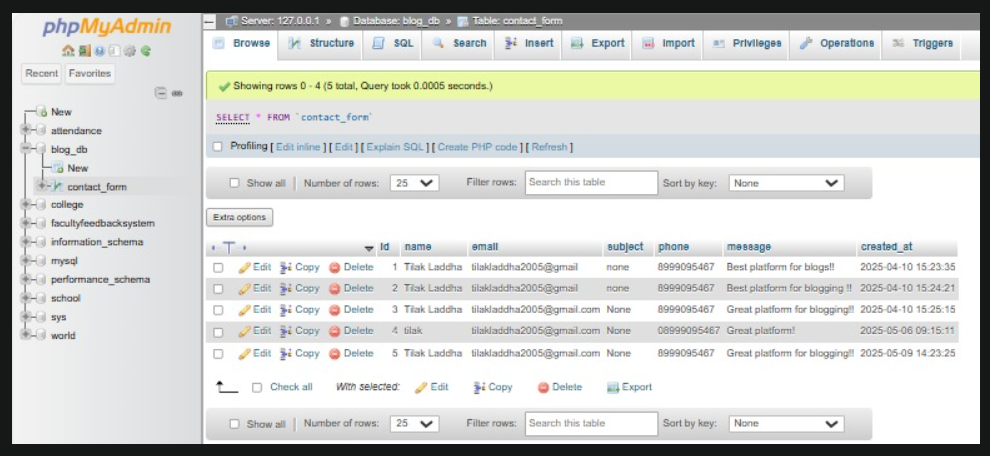
}

$conn->close();

}

?>

Dashboard:



### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### Conclusion

Implementing user registration with PHP forms the foundation of user management in the *Organic Skin Care Products* website. By securely collecting, validating, and storing user information, the system enables essential personalized features such as account login, saved preferences, and order tracking.

This implementation:

* Promotes **user trust** through secure handling of sensitive data like passwords using hashing.
* Ensures **data integrity** via robust server-side validation.
* Enhances the **user experience** with real-time feedback and clear, actionable error messages.

Overall, this PHP-based registration system plays a critical role in building a secure, user-friendly, and functional e-commerce platform.

## 

## 

## 

## 

## **Experiment 8**

### Problem Statement:

| PHP   1. Develop a PHP script to handle user login for the Organic Skin Care Products website. The script should accept input from users for their login credentials. (all required fields for login). 2. Provide feedback to the user upon successful login, either through a confirmation message or a redirect to a welcome page. 3. Implement error handling to notify users of login failures due to incorrect credentials or other errors. 4. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page. |
| --- |

### 

### A. Develop a PHP Script to Handle User Login

The **login system** is a vital component of the *Organic Skin Care Products* website, allowing registered users to securely access their accounts. PHP is used on the server side to validate credentials, manage sessions, and provide access control.

The login form typically includes:

* **Email address**
* **Password**

When the form is submitted, PHP:

* Validates the fields
* Fetches user details from the database
* Verifies the password using password\_verify()
* Starts a session or stores login data via localStorage/cookies

### B. Provide Feedback on Successful Login

After successful authentication:

* A **welcome message** is displayed (e.g., "Welcome, [username]")
* OR the user is **redirected to the home page** (e.g., index.html)
* Optionally, localStorage can store the username to display personalized greetings across pages

### C. Implement Error Handling for Login Failures

Common login errors include:

* Empty fields
* Invalid email format
* Incorrect email or password combination

**Your PHP script should:**

* Check that both fields are filled
* Validate the email format
* Show a clear error if the login fails (e.g., "Incorrect email or password")
* Avoid revealing which field was incorrect for security purposes

### D. Final Feedback or Redirection on Success

Once logged in:

* Start a PHP session or use client-side JavaScript to reflect login state
* Show a welcome message (Hello, [user])
* Redirect to a **home page** (e.g., index.html) or **user dashboard**

This enhances user experience and prepares the platform for additional personalized features like order history, wishlists, and profile settings.

**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=dashboard.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:

<?php

session\_start();

if (!isset($\_SESSION["user\_id"])) {

echo "Access denied. Please <a href='login.php'>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>";

?>

Logout.php:

<?php

session\_start();

session\_destroy();

header("Location: login.php");

exit();

?>

Db\_connect .php:

<?php

$servername = "localhost"; // Your server name

$username = "your\_username"; // Your database username

$password = "your\_password"; // Your database password

$dbname = "your\_database"; // Your database name

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

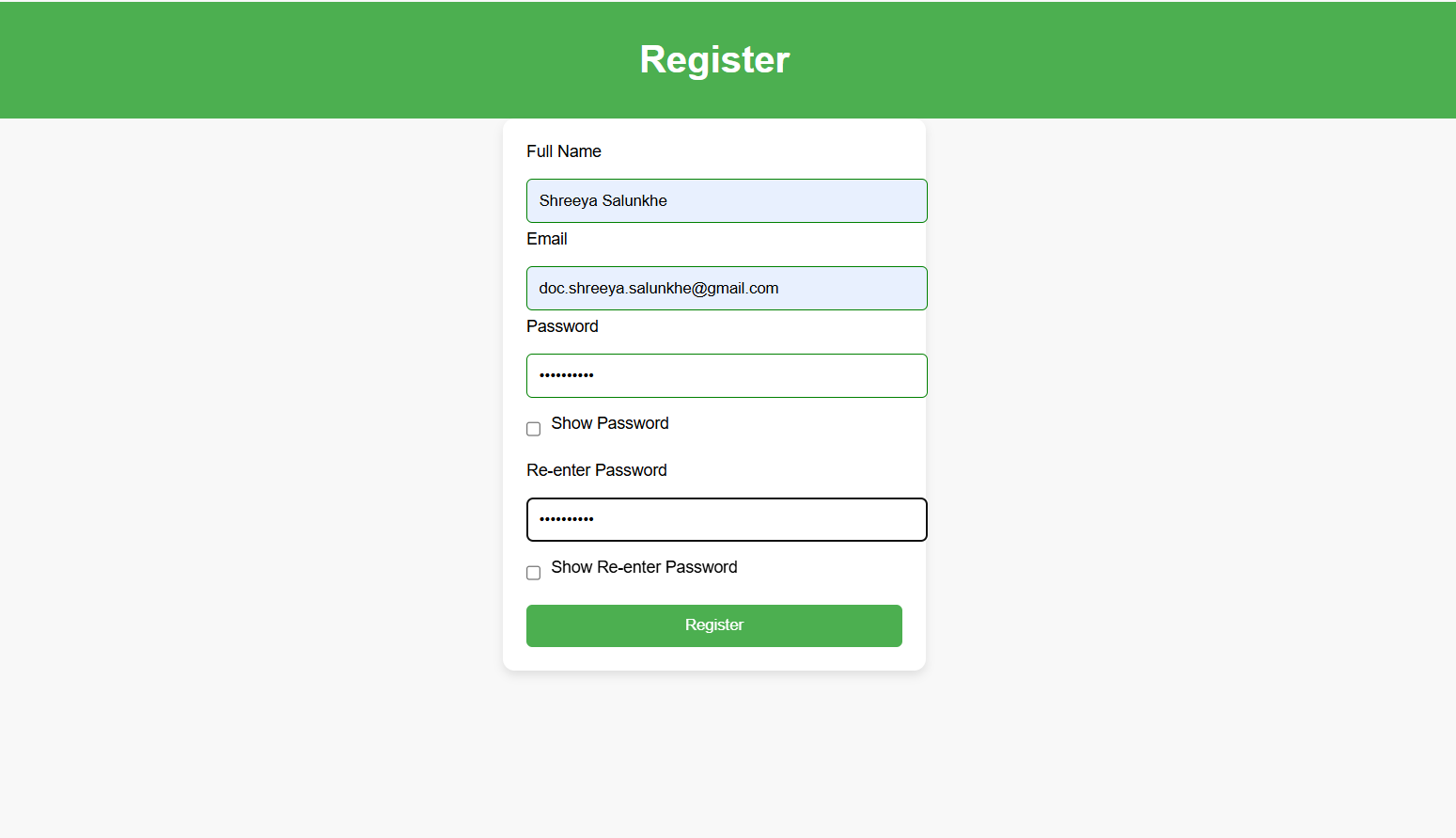
// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>



**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.

## 

## 

## 

## **Experiment 9**

**Problem Statement:**

A. Develop a PHP script that allows users to manage their shopping cart for an Organic Skin Care Product 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 Organic Skin Care Product 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 for Organic Skin Care Products Website

A shopping cart is a core component of any e-commerce platform. It acts as a temporary or persistent space where users can collect and manage organic skincare products they intend to purchase. For an **Organic Skin Care Products** website, where items might be handcrafted, seasonal, or available in limited quantities, the shopping cart system becomes even more crucial for ensuring smooth and personalized user experience.

#### Two Types of Cart Management Systems in PHP

#### A. Session-Based Shopping Cart (Without MySQL)

This method uses PHP sessions to temporarily store cart data in the server's memory while a user is browsing. It is suitable for simpler implementations or fast prototyping, especially when persistence isn't required beyond a single session.

##### **Key Characteristics:**

* Cart data is stored in the $\_SESSION superglobal.
* Cart information exists only during the current session.
* No login is required to use the cart.
* Cart is cleared once the browser is closed or the session expires.

##### **Operations Supported:**

* **Add to Cart**: Store product ID, name, quantity, and price in the session.
* **View Cart**: Retrieve and display items stored in the session.
* **Remove from Cart**: Delete a specific item from the session.

##### **Advantages:**

* Easy to implement.
* No database configuration needed.
* Lightweight and fast for short-term use.

##### **Limitations:**

* Cart data is lost after the session ends.
* Not ideal for logged-in users or long-term shopping experiences.

#### B. Database-Based Shopping Cart (With MySQL)

This approach is more professional and scalable, making it ideal for production websites where users log in and expect their carts to persist across devices and sessions. It uses a MySQL database to store cart data permanently (until the user checks out or manually clears the cart).

##### **Key Characteristics:**

* Each cart is tied to a specific logged-in user via a unique user\_id.
* Products are stored in a cart table and detailed in a cart\_items table.
* Requires authentication or user session to track individual carts.

##### **Operations Supported:**

* **Add to Cart**: Insert or update items in the cart\_items table.
* **View Cart**: Fetch all items from the database linked to a specific user.
* **Remove from Cart**: Delete items by item\_id or cart\_id.

##### **Advantages:**

* Cart persists across logins and devices.
* Enables user-specific cart experiences.
* Supports analytics and product preference tracking.

##### **Limitations:**

* Requires login system integration.
* More complex to set up and maintain.
* Needs proper validation and error handling.
* .

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) NOT NULL,

quantity INT DEFAULT 1,

price DECIMAL(10, 2) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(id) ON DELETE CASCADE

;



#### 

#### Conclusion

A robust shopping cart system is vital for any e-commerce website, especially for an Organic Skin Care Products platform where product availability and personalization matter. While session-based carts are simple and suitable for temporary use, a MySQL-backed cart offers the persistence, scalability, and enhanced user experience needed for a professional and user-friendly website.

## **Experiment 10**

**Problem Statement:**

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 for Organic Skin Care Products Website

The checkout process is the final and most crucial step in any e-commerce platform. It converts the user's shopping cart into a confirmed order by capturing essential details such as billing, shipping, and payment information. On an **Organic Skin Care Products** website, where products may be handcrafted, seasonal, or limited in stock, a robust and accurate checkout system helps maintain inventory integrity while ensuring a smooth customer experience.

### Two Approaches to Checkout

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

This basic approach is suitable for simpler websites or demonstration purposes.

##### **Workflow:**

* Cart data is retrieved from $\_SESSION['cart'].
* User input (name, email, address) is validated.
* A success or error message is displayed.
* The cart is cleared after confirmation.

##### **Advantages:**

* Easy and fast to implement.
* No database setup required.

##### **Limitations:**

* Data is not stored permanently.
* No order history or tracking.
* Not suitable for live or production environments.

#### B. MySQL-Based Checkout System

This is a more advanced and professional approach designed for scalable, real-world applications.

##### **Workflow:**

* Validate if the user is logged in.
* Retrieve cart items from the session or MySQL database.
* Validate checkout fields (e.g., shipping address, contact info).
* Insert order data into orders and order\_items tables.
* Show confirmation message.
* Clear the session or database-based cart.

##### **Advantages:**

* Persistent and reliable order storage.
* Enables order tracking and history for users.
* Ideal for production-level websites.

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,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

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),

FOREIGN KEY (order\_id) REFERENCES orders(id) ON DELETE CASCADE

);



##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### **Checkout Session (checkout.php)**

<?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 (process\_checkout.php)

<?php

session\_start();

$conn = new mysqli('localhost', 'root', '', 'your\_database\_name'); // Update with your database name

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; // Assuming user\_id is stored in session after login

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'];

}

// Insert order into the orders table

$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;

// Insert order items into the order\_items table

$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']); // Clear the cart after successful checkout

} else {

echo "Checkout failed. Please try again.";

}

$stmt->close();

$itemStmt->close();

$conn->close();

} else {

echo "Invalid request.";

}

?>

Conclusion

The checkout process is the most vital component of an e-commerce platform—it transforms customer intent into actual sales. For an **Organic Skin Care Products** website:

#### Use Case Importance:

* Products may be handcrafted or available in limited quantities, so accurate, real-time cart and inventory tracking is essential.
* Using MySQL ensures persistence of user selections and enables full order management for both customers and administrators.
* A session-based approach can be useful during early development or for simple guest checkouts.