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

Name of the Student

Mr. YASH SUREJA

A.Y. 2024 – 2025 (SEM-II

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.

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 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 front-end application. Utilize project-based problem-solving 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, , List properties, 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**
	Self-study / Do it yourself /: **Practice creating basic HTML pages and enhancing them using CSS.**
	Experiential Learning Topics: **Design a simple webpage for Gym Management website**
	PBL - Project Based Learning: **Create a multi-page website (e.g., Gym Management 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**
	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 **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** **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 Gym Management

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**
	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 Gym Management 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. 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 Gym Management / 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.
    - 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
        - Online booking system
        - Content management system (CMS)
        - Discussion forum
        - Social networking platform
        - Task management application
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.
        - Code Quality & Best Practices: Clean, well-organized, and efficient code that adheres to modern web development best practices and is easy to maintain.
4. 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 Gym Management using HTML, CSS, JavaScript, PHP and MySQL. This website will serve as an online presence for the Gym Management, 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:

gym-management-website/

gymmer-website/

```
├── index.html
├── about.html
├── membership.html
├── trainers.html
├── contact.html
├── login.html
├── register.html
├── dashboard.php
├── .htaccess
├── css/ style.css
├── js/ script.js
├── php/
│   ├── login.php
│   ├── register.php
│   ├── logout.php
├── images/ logo.png
├── sql/ gymmer_db.sql
└── README.md
```

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: a. Home page b. About page c. Product/Service page d. Testimonial/review page e. Contact page f. 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: a. Colors b. Fonts and typography c. Logos d. Images and photos
2.	HTML A. Create a detailed home page for the coffee shop website. B. Create a detailed menu/product page for the coffee shop website, listing all available items categorized appropriately. C. Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout. D. Create an about us page that provides detailed information about the coffee shop's history, mission, and team. E. Create a contact page that allows customers to easily get in touch with the coffee shop through a form. F. Design and implement admin/user registration form for the coffee shop website. G. Design and implement admin/user login form for the coffee shop website.
3.	CSS

	A. Enhance the layout of the coffee shop website using CSS Grid for the home page. B. 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 A. Enhance the cart page to make it user-friendly and visually appealing. Style the cart items with appropriate margins, paddings, and input field styles to provide a seamless shopping experience. B. Enhance and style the about us page with appropriate margins, paddings, and input field styles. C. Enhance and style the contact page to make it user-friendly and visually appealing. Style the contact form with appropriate margins, paddings, and input field styles. D. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles. E. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles.
5.	JavaScript A. Implement user registration and login forms for the coffee shop website. These forms will allow users to create an account, log in, and access personalized features, such as saving favorite items or viewing order history. User Registration Form will allow new customers to sign up and create an account on the website. The form will capture basic user details, including the name, email address, and password (not limited to these fields). User Login Form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user. B. Provide validations for user registration and login forms to validate the input to ensure that all required fields are filled and that the email format is valid. (**Contents beyond Syllabus)** C. Develop cart functionality to allow users to add items, update quantities, and remove items.
6.	JavaScript A. The user login form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user. B. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus)** C. Use localStorage or sessionStorage to store authentication data, such as the user's email and login status. This ensures that once a user is logged in, they remain authenticated even after the page reloads or when they visit the site again. (**Contents beyond Syllabus)**

	D. Save the cart data to local storage when items are added, updated, or removed. Retrieve and load the cart data from local storage when the page loads. (**Contents beyond Syllabus)**
7.	PHP A. Develop a PHP script to handle user registration for the Coffee Shop website. The script should accept input from users for their name, email address, password, etc. (all required fields for registration). B. Implement error handling to notify users of any issues during registration, such as validation errors. C. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page.
8.	PHP A. Develop a PHP script to handle user login for the Coffee Shop website. The script should accept input from users for their login credentials. (all required fields for login). B. Provide feedback to the user upon successful login, either through a confirmation message or a redirect to a welcome page. C. Implement error handling to notify users of login failures due to incorrect credentials or other errors. D. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page.
9.	PHP and MySQL A. Develop a PHP script that allows users to manage their shopping cart for an e-commerce website (e.g., a Coffee Shop store). The script should allow users to add items to their cart, view their cart contents, and remove items if needed. B. Develop a PHP script to manage the shopping cart for an e-commerce website (e.g., a Coffee Shop store) using MySQL. This script should allow users to add items to their cart, view their cart contents, and remove items from the cart. The cart data should be stored in the MySQL database to allow persistence across sessions.
10.	PHP and MySQL 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.

Experiment No.1

Problem Statement:

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

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

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

Objective:

To design the foundational structure of a **Gym Management System** website by planning its layout, content, and visual elements. The system will streamline member interactions, class bookings, and administrative tasks while reinforcing the gym's brand identity and fostering user engagement

Theory:

Project Design and Plan Document for Gym Management Website

1. Brief Information about the Project

The project is to create a user-friendly and visually engaging website for a gym management system named **FitGym**. It aims to attract fitness enthusiasts, highlight membership plans, showcase trainer profiles, and offer features such as user registration and login. The system will support personalized dashboards, interactive workout schedules, and a contact platform. This web-based solution will enhance the gym's online presence and streamline user engagement by providing secure access to exclusive features and updates.

2. Goals and Deliverables

Goals

1. **Develop an engaging, functional website for a modern gym/fitness center.**
2. **Showcase the gym's facilities, membership plans, trainer profiles, and success stories.**
3. **Enable users to:**
    - **Register/log in to access personalized dashboards.**
    - **Book classes/personal training sessions.**
    - **Track progress (workouts, nutrition).**
4. **Ensure responsiveness across all devices (mobile, tablet, desktop).**

Deliverables

- Website Pages:

  1.Home Page
  2.About Page
  3.Membership Page
  4.Success Stories
  5.Contact Page
  6.Login/Registration Page
  7.Dashboard

- Core Features:

✅ **Header &Foote**– Consistent navigation across all pages.
✅ **User Authentication** – Secure login/registration for members, trainers, and admins.
✅ **Responsive Design** – Mobile, tablet, and desktop-friendly UI.
✅ **Professional Aesthetic** – High-energy visuals, modern fonts and motivational imagery.

3. Finalize the modules of the project

The Gym Management 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 primary entry point that captures attention and converts visitors into members.

Features:

- Hero Section
  - Motivational tagline (e.g., *"Unlock Your Potential"*)
  - CTA buttons: *"Join Now"*, *"Free Trial Session"*
- Key Highlights
  - Featured classes (e.g., "HIIT Blast: Mondays 7 AM")
  - Promotions (e.g., "20% Off Annual Memberships")
- Navigation Bar
  - Links to *Memberships, Classes, About, Testimonials, Contact, Login*
- Footer
  - Gym address, phone, social media, quick links

2. About Page Module
3. Description:
   Builds trust by showcasing the gym's ethos and team.
4. Features:

- Our Story
  - Founding journey, mission (*"Fitness for All"*)
- Meet the Trainers
  - Certifications, specialties (e.g., "Yoga Expert, 10+ Years")
  - Trainer photos + bios
- Facility Tour
  - Gallery of equipment, locker rooms, etc.

3. Services/Membership Page Module

Description:

Clear presentation of membership tiers and classes.

Features:

- Membership Plans
  - Basic/Premium/VIP tiers with pricing and perks (e.g., "24/7 Access")
  - *"Sign Up"* buttons
- Class Schedule
  - Filter by type (Cardio, Strength) or time
  - *"Book Now"* functionality
- Personal Training
  - Package options (1-on-1, group sessions)

4. Testimonials/Success Stories Module

Description:

Social proof to motivate sign-ups.

Features:

- Member Transformations
  - Before/after photos with progress metrics
- Video Testimonials
  - 30-sec clips of members sharing results
- Review Submission (Optional)
  - Form for members to share experiences

5. Contact Page Module

Description:

Ensures seamless communication.

Features:

- Inquiry Form
  - Fields: Name, Email, Subject (*"Membership Query"*, *"Personal Training"*)
- Google Maps Embed
  - Location pin with directions
- Live Chat (Optional)
  - Instant support during business hours

6. Login Page Module

Description:

Secure access for members and staff.

Features:

- Login Form
  - Email + Password fields
  - *"Forgot Password?"* link
- Role-Based Redirect
  - Members → Dashboard
  - Trainers/Admins → Management portal

7. Registration Page Module

Description:

Onboarding for new members.

Features:

- Sign-Up Form
  - Fields: Full Name, Email, Phone, Password
  - Membership type selection
- Waiver/T&C Checkbox
  - "I agree to the gym's liability terms"
- Payment Integration (Future)
  - Stripe/PayPal for instant membership activation

8. Footer Module

Description:

Consistent site-wide navigation and legal info.

Features:

- Quick Links
  - Privacy Policy, FAQs, Careers
- Social Media Icons
  - Instagram (progress pics), YouTube (workout demos)
- Emergency Contact
  - "Injury? Call (123) 456-7890"

4. Define the audience

Target Audience

The website serves diverse user groups with unique fitness goals and needs. Understanding these segments ensures the design, content, and features align with their expectations.

a. Fitness Enthusiasts

Characteristics:

- **Regular gym-goers focused on performance and progress.**
- **Interested in advanced equipment, personal training, and specialized classes.**

Needs:

- **Class schedules with difficulty levels (beginner/intermediate/advanced).**
- **Progress-tracking tools (workout logs, body metrics).**
- **Access to trainer bios and session booking.**

b. Busy Professionals

Characteristics:

- **Limited time, need efficient workouts.**
- **May prefer early-morning/late-night sessions.**

Needs:

- **24/7 access information.**
- **Express workout options (30-minute sessions).**
- **Mobile app integration for on-the-go booking.**

c. Beginners/New Members

Characteristics:

- **New to fitness or returning after a hiatus.**
- **Need guidance and low-pressure environments.**

Needs:

- **"Getting Started" guides.**
- **Beginner-friendly classes (e.g., "Intro to Strength Training").**
- **Free trial offers.**

d. Health & Wellness Seekers

Characteristics:

- **Focused on holistic health (yoga, nutrition, recovery).**
- **Prefer non-intimidating environments.**

Needs:

- **Mind-body class schedules (yoga, Pilates).**
- **Nutrition coaching options.**
- **Recovery amenities (sauna, physiotherapy).**

e. Athletes & Competitors

Characteristics:

- **Training for sports/competitions.**
- **Need performance analytics and coaching.**

Needs:

- **Advanced training programs.**
- **VO_2 max testing, body composition scans.**
- **Sports-specific conditioning classes.**

f. Seniors & Rehabilitation Clients

Characteristics:

- **Focus on mobility, joint health, or post-injury recovery.**

Needs:

- **Low-impact classes (aqua aerobics, chair yoga).**
- **Trainer certifications in senior fitness/rehab**

g. Online Shoppers

- Characteristics:
  - Prefer the convenience of ordering coffee products online, including beans and brewing equipment.
- Needs:
  - Product page with a seamless e-commerce experience.
  - Secure login and registration system for repeat shopping.

Website Features Mapped to Audience Needs:

Audience Segment	Key Features Needed
Fitness Enthusiasts	Class filters by intensity, performance dashboards, trainer booking.

Busy Professionals	Express workout slots, 24/7 access highlights, mobile booking.
Beginners	Onboarding guides, trial offers, beginner-class tags.
Health and Wellness seekers	Yoga/Pilates schedules, recovery amenities, nutritionist profiles.
Athletes	Advanced metrics tracking, sports conditioning programs..
Seniors	Low-impact class filters, rehab specialist bios.

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 Gym Management's story 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)

1. Identifying Pain Points of Existing Systems

a. Pain Point: Poor Navigation & Cluttered Interface

- Issue: Many gym websites have confusing layouts, making it hard to find class schedules, membership plans, or trainer details.
- Impact: Users leave due to frustration, reducing sign-up conversions.

b. Pain Point: Limited Online Booking & Payment

- Issue: Outdated systems require phone calls or in-person visits to book classes or memberships.
- Impact: Busy users abandon sign-ups due to inconvenience.

c. Pain Point: Lack of Mobile Optimization

- Issue: Non-responsive designs break on smartphones, hiding key features like schedules or checkout.
- Impact: 60%+ of users access sites via mobile—poor UX loses potential members.

d. Pain Point: Incomplete Class/Trainer Info

- Issue: Vague descriptions of classes (e.g., "HIIT Session") without intensity levels, equipment needed, or trainer credentials.
- Impact: Users book unsuitable sessions, leading to dissatisfaction.

e. Pain Point: Weak Member Engagement

- Issue: No progress tracking, challenges, or community features to keep users motivated.
- Impact: High dropout rates after initial sign-up.

f. Pain Point: Hidden Contact/Gym Details

- Issue: Address, hours, or contact forms buried in menus.
- Impact: Visitors can’t easily inquire or locate the gym, losing walk-ins.

g. Pain Point: No Personalization

- Issue: Generic workout plans; no adaptation for goals (weight loss, muscle gain) or skill levels.
- Impact: Users feel undervalued, reducing retention.

2. Crafting the Ideal Experience

To address these pain points, the website design and functionality should create a user-friendly, visually appealing, and highly interactive experience.

a. Intuitive Navigation and Clean Design

- Use a clear and consistent layout with a sticky navigation bar.
- Include links to all key pages (Home, About, Menu, Testimonials, Contact, Login/Sign Up).

b. Seamless Online Ordering

- Implement a robust e-commerce system allowing customers to browse products, add items to a cart, and complete purchases effortlessly.
- Provide features like "Order Now" buttons on the homepage and menu pages.

c. Mobile-Responsive Design

- Design with a mobile-first approach, ensuring compatibility across devices.
- Use flexible grids, touch-friendly elements, and optimized performance for fast loading times.

d. Comprehensive Product Information

- Include high-quality images, item descriptions, ingredients, prices, and allergy/dietary labels (e.g., gluten-free or vegan).
- Create filters for health-conscious customers, like "Low Calorie" or "Vegan Options."

e. Customer Engagement Features

- Introduce loyalty programs with a points system visible after login.
- Offer a blog with content like coffee brewing tips, health benefits, or shop news.
- Highlight customer reviews and testimonials on a dedicated page.

f. Easy Access to Contact and Location

- Include a contact page with a simple inquiry form, phone number, and email.
- Display an embedded map for the shop's physical location on the homepage or contact page.

g. Personalization

- Allow users to create accounts for saving their favorite items or past orders.
- Use a welcome message with the customer's name after login.
- Send personalized offers via email for registered users.

3. The Ideal User Journey

Step 1: Visiting the Website

- Users arrive at a welcoming homepage with clear navigation to different sections.

Step 2: Browsing the Menu

- Users navigate to the products/services page, view clear menus, and filter items based on preferences.

Step 3: Placing an Order

- Users can seamlessly add items to their cart and complete a purchase with minimal clicks.

Step 4: Finding Location or Contacting Support

- Users easily locate contact and location details for in-store visits or inquiries.

Step 5: Engaging with Content

- Users read blogs or testimonials for a deeper connection with the brand.

Step 6: Creating Loyalty

- Registered users receive personalized promotions or gain points through purchases.

6. Set the visual direction

1. Visual Design Goals

The design should energize, motivate, and build trust, aligning with the gym's brand identity.

Key principles:

- **High-Energy & Motivational – Inspire action with dynamic visuals.**
- **Modern & Professional – Clean, uncluttered layouts for easy navigation.**
- **Trust-Building – Showcase real members, trainers, and facilities.**
- **Gender-Inclusive – Avoid stereotypical "bodybuilder-only" imagery.**

2. Defining the Core Visual Elements

a. Color Palette

A bold yet balanced palette combining **powerful accents** with **neutral backgrounds** for readability.

Color	Hex Code	Usage
Power Red	#E63946	CTAs ("Join Now"), urgent alerts
Steel Grey	#2B2D42	Headers, footer, text (professional tone).
Pure White	#FFFFFF	Backgrounds, card spaces
Electric Blue	#457B9D	Interactive elements (buttons, links)
Active Orange	#F77F00	Highlights, discounts, energy cues

b. Typography

Fonts should **command attention** while remaining readable on all devices.

- Primary Font: *Bebas Neue (Bold, all-caps) – For impactful titles like "TRANSFORM YOUR LIFE TODAY."*
- Secondary Font: Open Sans (Light/Regular) – Clean, readable descriptions for class details, bios.
- Attributes: Use bold headings

c. Logos and Branding

A logo that communicates **strength and community**:

- Abstract dumbbell/heart fusion (fitness + passion).
- Silhouette of diverse athletes in motion.

d. Imagery and Icons

High-quality visuals can make the website feel alive and inviting.

- Photography:
    - Action Shots: Members lifting, running, or in yoga poses (diverse ages/body types).
    - Facility Tours: Clean weight rooms, pools, locker rooms..
- Icons:

  Equipment categories

  Class booking

3. Applying Visual Design to Pages

a. Home Page

- **Hero Image: High-energy action shot (e.g., group class, trainer spotting a member).**
- **Text Overlay: Bold tagline like *"Unleash Your Potential"* in Bebas Neue (white, uppercase).**
- **CTA Buttons: "Join Today" (Power Red #E63946) + "Free Trial" (Electric Blue #457B9D).**

b. About Page

- **Authentic Photos: Trainers coaching, members achieving goals, facility tours.**
- **Diversity: Show all age groups/body types to emphasize inclusivity.**

c. Product/Service Page

- **Layout:** Grid with clear hierarchy (Class image → Title → Time/Price → "Book Now")

d. Testimonial Page

- Alternating cards with member photos + quotes.
- Arrow button in power red for

e. Contact Page

- Custom pins in Electric Blue with gym logo.

f. Login and Registration Pages

- "Sign Up" in Electric Blue
- "login" in Power Red

4. Layout and Design Hierarchy

The visual hierarchy ensures an easy and intuitive flow through the website:

1. Headers and Banners: Hero video of a high-energy workout session
2. Navigation Bar: Hero video of a high-energy workout session
3. Sections and Grids: **Card-based grids** for trainers
4. Call-to-Action: Fixed floating CTA on mobile

5. Expected Impact of Visual Direction

1. Enhanced Engagement: Dynamic class schedule with real-time availability.
2. Stronger Branding: Consistent use of Power Red for urgency (limited-time offers)
3. Better Retention: Personalized dashboards post-login (upcoming bookings, achievements)
4. Higher Conversions: Exit-intent popup: "Get 10% Off Your First Membership!"

7. Map out the Project structure

```
gym_management/
│
├── index.html           # Home page (with hero banner and CTAs)
├── about.html           # About Us (gym story, trainer bios)
├── memberships.html     # Membership plans & pricing
├── classes.html         # Class schedules and booking
├── trainers.html        # Trainer profiles and sessions
├── testimonials.html    # Success stories/reviews
├── contact.html         # Contact form + location map
├── login.html           # Member/login portal
├── register.html        # Registration/signup
├── dashboard/           # Post-login user area (optional)
│   ├── member.html      # Member profile/workout tracker
```

```
│   ├── admin.html        # Admin management panel
│
├── assets/
│   ├── css/
│   │   ├── styles.css       # Global styles (colors, typography)
│   │   ├── responsive.css    # Mobile/tablet breakpoints
│   │   ├── animations.css    # Micro-interactions (hover effects)
│   │
│   ├── js/
│   │   ├── main.js          # Core functionality
│   │   ├── class-booking.js  # Real-time schedule management
│   │   ├── form-validation.js # Login/signup validation
│   │   ├── progress-tracking.js # Workout analytics (for dashboard)
│   │
│   ├── images/
│   │   ├── branding/
│   │   │   ├── logo.png     # Gym logo (main + monochrome versions)
│   │   │   ├── favicon.ico
│   │   │
│   │   ├── hero/
│   │   │   ├── home-banner.jpg # High-energy group class
│   │   │   ├── about-banner.jpg # Facility tour
│   │   │
│   │   ├── classes/        # Yoga, HIIT, strength training
│   │   ├── trainers/       # Headshots + action shots
│   │   ├── testimonials/    # Before/after transformations
│   │   ├── icons/          # SVG icons (dumbbell, calendar, etc.)
│   │
│   ├── videos/            # Optional - workout teasers
│
├── fonts/
│   ├── BebasNeue/          # Bold headings (e.g., "JOIN TODAY")
│   ├── Montserrat/         # Body text (clean readability)
│
```

```
GYMMANAGER/
│
├── admin_panel/
├── assets/
│   ├── css/
│   ├── fonts/
│   ├── images/
│   ├── js/
│   └── plugins/
│
├── includes/
├── requests/
├── sections/
│
├── cart.php
├── checkout.php
├── index.php
├── offers_list.php
├── wishlist.php
```

8. Plan the content for each page

The website will include a minimum of 5 core pages, along with additional Login and Registration pages. This plan details the content for each page.

1.Home Page

Purpose:

Welcome visitors.

Promote gym services, trainers, and facilities.

Content Plan:

- Header:
  - Logo on the left.
  - Navigation Menu: Home, About, Services, Testimonials, Contact.
  - Login/Sign-Up button on the right.
- Hero Section:
  - High-quality banner image (gym interior, people training).
  - Tagline: *"Train Hard, Live Strong!"*
  - CTA Button: “Explore Memberships”

- Introduction Section:
  - Brief intro to the gym (when it opened, philosophy).
  - CTA: “Meet Our Trainers” (links to About or Trainers section).
- Highlight Section:
  - Grid or carousel of key offerings (personal training, group classes, nutrition plans).
  - Text: “Check Out Our Latest Training Programs!”
- Footer:
  - Quick links, social media, phone/email, gym address.

2. About Page

Purpose:

- Share the gym’s story, values, and community impact.

Content Plan:

- Header: (same as Home).
- About Us Section:
  - History of the gym: founding story, mission, and values (e.g., empowerment, inclusivity).
  - Emphasis on fitness philosophy (e.g., holistic health, strength through community).
- Meet the Team Section:
  - Profiles of personal trainers, nutritionists, and staff.
  - Short bios, specializations, and photos.
- Why Choose Us Section:
  - Unique Selling Points:
    - Certified trainers
    - Modern equipment
    - Flexible membership options
    - Nutrition guidance
- Footer: (same as Home).

3. Services Page

Purpose:

- Showcase gym services and membership options.

Content Plan:

- Header: (same as Home).
- Membership Options Section:
  - Categories: Basic, Premium, Personal Training, Online Programs.
  - Cards/tiles with details: duration, pricing, benefits.
- Programs & Classes:
  - Sections like: Strength Training, HIIT, Yoga, Zumba, CrossFit.
  - Description, trainer info, timing.
- Top Picks Section:
  - Popular packages or recommended combinations.
- CTA Section:

    - "Join Now" → leads to Login or Registration page.
- Footer: (same as Home).

4. Testimonials Page

Purpose:

- Build trust via real member experiences.

Content Plan:

- Header: (same as Home).
- Testimonials Section:
    - Quotes from members (transformation stories, satisfaction).
    - Optional: Embed Google or Instagram reviews.
    - Use of star ratings and photos for credibility.
- Submit Testimonial Section:
    - Simple form: Name, Membership Type, Review.
- Footer: (same as Home).

5. Contact Page

Purpose:

- Enable users to reach the gym for info or consultations.

Content Plan:

- Header: (same as Home).
- Contact Form:
    - Fields: Name, Email, Phone, Message.
- Location Section:
    - Embedded Google Map of gym location.
- Operating Hours:
    - Clearly listed weekly schedule.
- Call to Action:
    - Phone button or WhatsApp chat integration.
- Footer: (same as Home).

6.Login Page

Purpose:

- Allow members and admins to log in.

Content Plan:

- Login Form:
    - Fields: Email, Password.
    - Submit button.
- Password Recovery Link:
    - "Forgot your password?"
- CTA:
    - "Not a member yet? Sign Up Here!"

7. Registration Page

Purpose:

- Let new users sign up for gym services.

Content Plan:

- Form Fields:
  - Full Name
  - Email
  - Phone Number
  - Password / Confirm Password
  - Select Membership Plan
- Validation:
  - Minimum password strength
  - Mandatory fields
- Submit Button
- Footer: (same as Home)

9. Add ideas for content, images & layout

1. Home Page

Layout Ideas:

- Header:
  - Fixed navigation bar.
  - Login/Register on the right.
- Hero Section:
  - Overlay text: “Transform Your Life Today.”
  - CTA Button: “Join a Program” or “View Memberships”.

Content Ideas:

- Welcome message: *"Empowering every step of your fitness journey."*
- Promotion: *“Summer Fitness Challenge – Sign up now!”*

Image Ideas:

- Gym interior shots, action photos of clients lifting or stretching.
- Before/after transformation photos (with consent).
-

2. About Page

Layout Ideas:

- Our Story Section:
  - Timeline or vertical split layout (Mission, Origin, Vision).
- Meet the Trainers Section:
  - Grid layout with trainer photos, names, specialties.
- Why Choose Us:
  - Icons + text to showcase features: Certified Trainers, Flexible Hours, Diet Plans.

Content Ideas:

- Story of the gym's founding and growth.
- Core values: health, inclusivity, community.

Image Ideas:

- Trainers in action.
- Community events hosted at the gym.

3. Programs / Services Page

Layout Ideas:

- Category Tabs or Sections:
  - Examples: Strength, Cardio, HIIT, Yoga, Nutrition.
  - Each with image, description, price (or duration).
- CTA Under Each Card:
  - "Book Now" or "More Details".

Content Ideas:

- Short intros on what each program offers.
- Target outcomes: e.g., muscle gain, fat loss, flexibility.

Image Ideas:

- Trainers leading group classes.
- Members working out with intensity and positivity.

4. Testimonials Page

Layout Ideas:

- Slider or Masonry Grid:
  - Scrolling or staggered testimonials with ratings.
- Submit Form at the Bottom:
  - Simple layout for reviews (name, feedback, rating).

Content Ideas:

- Quotes from clients detailing their transformations.
- Star ratings, brief trainer shoutouts.

Image Ideas:

- Before/after progress photos.
- Client portraits with quotes.
- Wall of fame/achievement highlights.

5. Contact Page

Layout Ideas:

- Contact Form:
  - Name, Email, Phone, Message.
- Map Section:
  - Embedded Google Map of gym location.

- Details Section:
    - Icons for hours, phone, email, social media.

Content Ideas:

- Friendly message: *"Reach out to start your transformation today!"*
- Quick FAQ: gym rules, locker info, guest passes.

Image Ideas:

- Exterior photo of the gym.
- Contact icons in branded style (e.g., dumbbell-shaped phone icon).

6. Login Page

Layout Ideas:

- Minimalist two-panel layout.
    - Left: form (email, password).
    - Right: branded image or quote like *"Discipline starts here."*

Content Ideas:

- Email/password fields.
- "Forgot password?" link.
- "New here? Sign up!" below the form.

Image Ideas:

- Fitness-themed background (weightlifting silhouette, sunrise run).
- Clean UI with dark accents.

7. Registration Page

Layout Ideas:

- Option to choose plan/type of member (Basic, Pro, Elite).
- Password strength meter.

Content Ideas:

- Fields: Name, Email, Phone, Password, Membership Type.
- CTA: "Start Your Fitness Journey!"

Image Ideas:

- Motivational image/banner.
- Icons for fields (lock, envelope, phone).

8. Visual Design Ideas for All Pages

Color Scheme:

- Primary: Bold, energetic tones like navy blue, charcoal, and steel gray.
- Backgrounds: White or light gray with subtle textures.

Typography:

- Headers: Bold, modern font (e.g., *Oswald*, *Montserrat Extra Bold*).
- Body text: Clean sans-serif (e.g., *Roboto*, *Open Sans*).

Logos & Icons:

- Stylized dumbbell or bicep in the logo.
- Icons: muscular arm, heart rate, stopwatch, yoga pose.

Image Style:

- High-energy, dynamic photography.
- Use of lighting, grit, and action in imagery.
- Videos or animated GIFs for workouts or gym tours.

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

a. **Home page**
b. **About page**
c. **Product/Service page**
d. **Testimonial/review page**
e. **Contact page**
f. **Starter blog posts**

1.Home Page

Purpose: Convert visitors into members with high-energy visuals and clear CTAs. Content:

- Hero Section:
  - *Tagline:* "Transform Your Life – Start Today!"
  - *CTA Buttons:* "Join Now" (primary), "Free 3-Day Trial" (secondary).
  - *Background:* Video of a dynamic group workout session.
- Key Features Grid:
  - 24/7 Access | Certified Trainers | 100+ Weekly Classes.
- Class Highlights:
  - Carousel of popular classes (HIIT, Yoga, Strength) with brief descriptions and "Book Now" links.
- Membership Teaser:
  - "Starting at ₹3,999/month – No Lock-In Contracts!" Design:
- Color Scheme: Power Red (#E63946), Steel Gray (#2B2D42), Pure White.
- Typography: Bebas Neue (headings), Open Sans (body).

2.About Page

Purpose: Build trust through storytelling and transparency.
Content:

- Our Story:
  - "Since 2012, we've empowered 15,000+ members to achieve their fitness goals."
- Meet the Team:
  - Trainer bios with certifications (NASM, ACE) and specialties (e.g., "Post-Rehab Training").
  - Staff photos in action (coaching, sanitizing equipment).
- Facility Tour:
  - Gallery: Weight room, cardio zone, locker rooms, recovery area (sauna, physio). Design:
- Split-screen layout: Text on left, trainer video interviews on right.
- Milestone timeline: "2015: Voted Best Gym in Mumbai."

c. Services/Membership Page

Purpose: Drive sign-ups with clear pricing and class access.
Content:

- Membership Tiers:
  - Basic (Gym Access) – ₹3,999/month
  - Premium (+ Group Classes) – ₹5,999/month
  - VIP (+ Personal Training) – ₹9,999/month
- Class Schedule:
  - Interactive calendar with filters (time, intensity, trainer).
  - Hover effects show remaining slots.
- Personal Training:
  - Packages: 5 Sessions – ₹10,000 | 10 Sessions – ₹18,000. Design:
- Comparison table with "Most Popular" badge on Premium tier.
- Grid layout for classes with thumbnail previews.

d. Testimonial/Review Page

Purpose: Motivate sign-ups with real success stories.
Content:

- Member Transformations:
  - Before/after sliders with metrics (e.g., "Lost 25kg in 6 Months – Priya S.").
- Video Testimonials:
  - 30-60 sec clips of members sharing journeys.
- Social Proof:
  - "Rated 4.9/5 ★ by 2,000+ Members." Design:
- Masonry grid for testimonials.

- Auto-rotating video carousel.

e. Contact Page

Purpose: Simplify inquiries and location access.
Content:

- Form Fields:
  - Name, Email, Phone, "I'm interested in..." (Membership/Classes/PT).
- Location:
  - Google Maps embed with custom pin.
  - Address: "Near Bandra Station, Mumbai. Parking Available."
- FAQ Accordion:
  - "Do you offer corporate memberships?" | "Can I freeze my plan?" Design:
- Split layout: Form (left), Map + Address (right).
- CTA: " Call Us: +91 98765 43210."

f. Starter Blog Posts

Purpose: Drive SEO traffic and member engagement.
Sample Posts:

1. "5 Beginner-Friendly Workouts to Start Your Fitness Journey"
   - Target Keyword: "gym routines for beginners."
   - Content: Step-by-step guides with GIFs.
2. "How to Choose the Right Personal Trainer"
   - Target Keyword: "best personal trainer near me."
   - Content: Checklist + trainer interview quotes.
3. "Meal Prep 101: Nutrition Tips for Busy Professionals"
   - Target Keyword: "meal plans for weight loss."
   - Content: Infographic + downloadable PDF

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:

Purpose: Create a high-energy, motivational, and professional digital presence that reflects strength, inclusivity, and results-driven fitness.

1. Colors

****Primary Colors:**

-Power Red `#E63946`

- Use: Headers, urgent CTAs ("Join Now"), alerts.

- *Psychological Impact:* Evokes urgency, energy, and action.

- Steel Gray `#2B2D42`

- Use: Navigation bars, footer, text.

- Psychological Impact: Professionalism, stability, and sophistication.

- Pure White `#FFFFFF`

- Use: Backgrounds, card sections.

- Psychological Impact: Cleanliness, clarity, and focus.

Impact:

- Red and orange drive urgency for sign-ups.

- Gray and white ensure readability and professionalism.

- Blue balances energy with trust for holistic wellness.

2. Fonts and Typography

Heading Font:

- Bebas Neue (Bold, Uppercase)

Body Font:

- Open Sans (Light/Regular)

- Use:Paragraphs, class descriptions, FAQs.

- Why:Clean, highly readable, mobile-friendly.

3. Logo

Design:

- Icon: Abstract dumbbell fused with a heart rate line (symbolizing fitness and vitality).

- Text: Gym name in **Bebas Neue (uppercase), Electric Blue (`#457B9D`).**

- Variants:

- Full-color (icon + text).

- Monochrome (Steel Gray for headers).

4. Imagery and Photos

Types of Imagery:

1. **Action Shots:**

- High-intensity workouts (e.g., weightlifting, HIIT).

- Yoga/recovery sessions (calm, balanced vibe).

- *Example:* "Trainer spotting a member during a bench press."

2. Facility Photos:

- Clean, modern equipment.

- Locker rooms, showers, and recovery areas (sauna, physio).

3. Lifestyle Shots:

- Diverse members (ages, body types) achieving goals.

- Community group classes (smiling, high-energy).

4. Before/After Galleries:

- Member transformations with metrics (e.g., "Lost 20kg in 6 Months").

Style Guidelines:

- **Authentic:** Avoid overly staged photos.

- **Diversity:** Include all genders, ages, and fitness levels.

- **Color Grading:** High contrast with vibrant accents (red/orange highlights).

5. Interactive Elements and Buttons

CTAs:

- **Primary Buttons:**

- "Join Now" (Power Red `#E63946`).

- "Book a Free Trial" (Electric Blue `#457B9D`).

- *Hover Effect:* Slight shadow + scale-up (5%).

Icons:

- **Class Types:** 🏋️‍♂️ (Strength), 🧘‍♀️ (Yoga), 🚴‍♂️ (Cardio).

- Features: (Schedule), (Progress), (Location).

Conclusion

This design system creates a cohesive, action-driven experience that aligns with a gym's brand values:

- Energy: Bold reds and dynamic imagery inspire action.

- Trust: Steel Gray and Open Sans ensure professionalism.

- Inclusivity:Diverse photos and community-focused CTAs.

- Results: Before/after galleries and progress tracking.

By integrating these elements, the gym's website becomes a powerful tool for converting visitors into members while fostering loyalty and motivation.

Experiment No.2

Problem Statement:

2. HTML

A. Create a detailed home page for the gym management website.

B. Create a detailed menu/product page for the GYM management website, listing all available items categorized appropriately.

C. Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout.

D. Create an about us page that provides detailed information about the GYM management history, mission, and team.

E. Create a contact page that allows customers to easily get in touch with the GYM management through a form.

F. Design and implement admin/user registration form for the GYM management website.

G. Design and implement admin/user login form for the GYM management website.

Objective

To design and structure a set of core HTML pages for the Gym Management System that enable:

- A welcoming and informative **home page**.
- A detailed **product/menu page** for services or merchandise.
- A functional **cart page** for purchase review.
- Informative **About Us** and **Contact** pages.
- Interactive **user/admin registration and login forms**.

Theory

HTML (HyperText Markup Language) is the standard markup language for creating and structuring content on the web. It serves as the foundation for every webpage, enabling developers to define elements such as text, images, links, forms, and multimedia. In the context of a Gym Management System, HTML plays a crucial role in constructing an intuitive, functional, and user-friendly interface for both customers and administrators. It ensures that content is properly organized, visually appealing, and easy to navigate. Below is an overview of how HTML can be used to create various pages in a Gym Management System.

A. Home Page

The Home Page is often the first point of interaction for users, so it must be designed to be visually appealing and easy to navigate. HTML provides the building blocks for creating a clean layout that guides users to important sections of the website. Key elements on a gym's homepage typically include a header with the gym's logo and navigation menu, a section for featured services (such as memberships, training programs, and promotions), testimonials from satisfied customers, and call-to-action buttons (like "Join Now" or "Learn More").

HTML tags like <header>, <nav>, <section>, <article>, and <footer> are commonly used to structure the content. For example, the navigation bar can be created using <nav> and

(unordered list) elements to link to other pages like the Membership page, Contact page, or Product page. Sectioning content with <section> helps organize different parts of the page for better readability and search engine optimization (SEO).

B. Menu/Product Page

The Menu/Product Page in a Gym Management System lists various products and services offered, such as membership plans, training packages, supplements, and gear. HTML is essential for displaying these items in an organized, user-friendly manner. To create logical categories for the offerings, developers often use containers, headings like <h2> and <h3>, images with <img>, and tables or lists to present information in an easy-to-read format.

For example, the gym might offer different membership plans, categorized as "Monthly Plans," "Yearly Plans," and "Premium Packages." Each category would be placed in its own section, and a table could be used to compare the features and pricing of each plan. Using HTML elements such as <ul> for unordered lists or <table> for structured data makes it easy for users to browse through options. For enhanced accessibility, alt text should be added to images, and forms can be used for any interactive features, such as purchasing a membership or signing up for a package.

C. Cart Page

The Cart Page allows users to review the items they've selected before proceeding to checkout. This page requires HTML to display the cart's contents, including details such as the name, quantity, and price of each item. Developers typically use <table> or <ul> lists to structure the cart items, ensuring that the information is clearly displayed.

Each item in the cart might include an option to change the quantity or remove the item. HTML forms and buttons are used to handle these actions. For example, a <form> with <input> fields allows users to adjust the quantity of an item, while a <button> element can enable the user to proceed to checkout or remove an item. Although the logic for updating the cart and processing the checkout is usually handled by JavaScript or server-side languages like PHP, HTML provides the basic structure and ensures the page is interactive.

D. About Us Page

The About Us Page provides important information about the gym's mission, history, values, and team. HTML is used to create a narrative that is easy to read and understand. Heading elements like <h1> and <h2> can introduce different sections, such as the gym's history or team member profiles, while <p> (paragraph) elements structure the text. Images of the gym or team members can be inserted using the <img> tag, and videos can be included with the <video> tag.

For team profiles, the <figure> and <figcaption> tags can be used to display images alongside descriptions of each team member's role. By using <section> and <aside>, developers can organize content in a way that's easy to navigate, while also enhancing SEO.

E. Contact Page

The Contact Page is an essential part of any Gym Management System, as it allows users to get in touch with the gym for inquiries or support. HTML provides the framework for the contact form, which typically includes fields for the user's name, email address, message, and sometimes a phone number. Using the <form> element, developers can define the structure of

the form, with various <input> fields (such as <input type="text"> and <input type="email">) for collecting user information.

A <textarea> can be used for a longer message input, and a <button> with a type of "submit" will send the form data when clicked. To improve usability and accessibility, the form can include labels for each field using <label> tags, ensuring that screen readers and other assistive technologies can properly interpret the content.

F. Registration Form

The Registration Form is a key component for new users to create an account. HTML provides various input types, such as <input type="text"> for name, <input type="email"> for email, and <input type="password"> for passwords. Developers can also use <select> dropdown menus for selecting options like membership type or training preferences, and <input type="checkbox"> for agreement to terms and conditions.

G. The Login Form is typically simple, consisting of fields for the user's email address and password. HTML's <form> structure, combined with <input type="email">, <input type="password">, and <button type="submit">, allows users to securely log into the gym's management system. Using proper semantic HTML with accessible elements, such as labels for form inputs, enhances the experience for all users, including those using screen readers.

In conclusion, HTML is essential for building a structured and functional Gym Management System.

Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>FitLife Gym - Home</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <div class="logo">FitLife Gym</div>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="products.html">Memberships</a></li>
        <li><a href="cart.html">Cart</a></li>
        <li><a href="about.html">About Us</a></li>
        <li><a href="contact.html">Contact</a></li>
        <li><a href="login.html">Login</a></li>
        <li><a href="register.html">Register</a></li>
      </ul>
```

```
    </nav>
  </header>

  <section class="hero">
    <h1>Transform Your Body, Transform Your Life</h1>
    <p>Join FitLife Gym today and get 50% off your first month!</p>
    <button class="cta-button">Join Now</button>
  </section>

  <section class="features">
    <div class="feature">
      <h3>Personal Training</h3>
      <p>Get customized workout plans from certified trainers.</p>
    </div>
    <div class="feature">
      <h3>Group Classes</h3>
      <p>Yoga, HIIT, Zumba, and more!</p>
    </div>
    <div class="feature">
      <h3>24/7 Access</h3>
      <p>Work out anytime, day or night.</p>
    </div>
  </section>

  <section class="testimonials">
    <h2>What Our Members Say</h2>
    <div class="testimonial">
      <p>"Best gym in town! The trainers are amazing."</p>
      <p>- John D.</p>
    </div>
  </section>

  <footer>
    <p>&copy; 2024 FitLife Gym. All rights reserved.</p>
  </footer>

  <script src="script.js"></script>
</body>
</html>
```

Experiment No.3

Problem Statement:

CSS

A. Enhance the layout of the Gym Management website using CSS Grid for the home page.

Objective

To enhance the layout and visual appeal of the website using **CSS Grid**, enabling a structured, responsive design for the homepage and product/menu listings with organized categories, headings, images, and pricing.

Theory

CSS (Cascading Style Sheets) is a cornerstone technology used to enhance the visual presentation of web pages. It allows web developers to control the layout, color, typography, and positioning of elements on a webpage. One of the most powerful layout systems in CSS is CSS Grid, which enables the creation of complex and responsive grid-based designs. It allows developers to easily define rows and columns within a container and control how elements are placed within these grid areas. When it comes to designing a Gym Management website, CSS Grid can be an invaluable tool for creating an organized, visually appealing layout that improves the user experience. Here's how CSS Grid can be applied to enhance the layout of the Home Page and Menu/Product Pages of a Gym Management website.

1. CSS Grid for the Home Page Layout

The Home Page of a Gym Management website serves as the first point of contact with customers. It must immediately communicate the brand's identity and showcase key offerings like the menu, special promotions, and store ambiance. A well-structured homepage provides easy access to different sections, ensuring that users can quickly find what they are looking for.

With CSS Grid, developers can create a clean, structured, and responsive layout for the homepage. The page can be divided into rows and columns that represent distinct content sections such as:

- Header: This section typically contains the Gym Management's logo, navigation menu, and possibly a call-to-action (CTA) button (e.g., "Order Now").
- Main Banner or Hero Section: A large image or video background that represents the ambiance or theme of the Gym Management.
- Featured Products or Promotions: Showcasing a selection of the Gym Management's signature drinks or seasonal offerings.
- Footer: Including essential information like location, contact details, social media links, and hours of operation.

Using CSS Grid, the layout can be divided into a two-dimensional grid of rows and columns. For instance, the homepage grid could have three columns for the content sections (header, banner, and footer) and multiple rows to organize the content. Here's a simple example of how the grid layout might be defined:

With this structure, the main content of the page is neatly organized into distinct areas, making it both visually appealing and functional. The use of grid-template-columns and grid-template-rows allows developers to control the proportions and flow of content, creating a cohesive design that adapts well across devices.

2. Menu/Product Page Layout with CSS Grid

The Menu/Product Page is critical for a Gym Management website. It should clearly display the Gym Management's offerings, such as different coffee blends, pastries, seasonal items, and other beverages. This page needs to be highly structured, with clear visual separation between categories, product names, descriptions, and prices. CSS Grid provides an ideal way to layout these items in a systematic, easy-to-read format.

A typical product menu could be organized into various categories, such as:

- Coffee Types (e.g., Espresso, Latte, Cappuccino)
- Pastries and Snacks
- Seasonal Specials
- Beverages (e.g., Juices, Smoothies)

Each category might contain a list of items with corresponding descriptions, images, and prices. CSS Grid can be employed to divide the page into structured areas that make it easy for customers to scan and select their desired products.

Defining a Menu Layout with CSS Grid:

A basic approach for creating a grid layout might involve defining columns for each product's image, name, description, and price. Here's an example of how to structure the grid for the menu items:

Breakdown of the Layout:

1. Grid Container: The .menu-container is defined as a grid with four equal-width columns using grid-template-columns: repeat(4, 1fr);.
2. Individual Menu Item: Each menu item is also structured as a grid using grid-template-columns: 1fr 2fr;.
3. Spacing and Styling: The gap property defines the spacing between both rows and columns, ensuring that menu items are visually separated
4. Images and Responsiveness: The images are set to fill their grid cell with width: 100% and height: auto to ensure they scale properly across different screen sizes.

3. Responsive Design Considerations

One of the key benefits of using CSS Grid is its responsiveness. By adjusting the grid's structure based on the screen size, developers can create layouts that look great on both large desktop screens and smaller mobile devices. For example, on smaller screens (like smartphones), the grid can be redefined to display fewer columns.

This ensures that the layout adapts seamlessly to different devices, enhancing the user experience.

Conclusion

CSS Grid is an incredibly powerful tool for creating structured, responsive, and visually appealing layouts for websites. By using CSS Grid to organize the home page and menu/product page of a Gym Management website, developers can easily manage content and create a clean, user-friendly experience. CSS Grid provides a flexible and maintainable approach to building layouts that scale across devices, ensuring that the website looks good no matter where it's viewed. With its ability to handle complex designs with simple code, CSS Grid is an essential technique for modern web design.

```
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    margin: 0;
    padding: 0;
    line-height: 1.6;
    background-color: #f4f6f8;
    color: #2c3e50;
}
header {
    background: linear-gradient(90deg, #1f1c2c, #928dab);
    color: #fff;
    padding: 1rem 2rem;
    display: flex;
    justify-content: space-between;
    align-items: center;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.2);
}
.logo {
    font-size: 2rem;
    font-weight: bold;
    background: linear-gradient(to right, #ff6a00, #ee0979);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}
nav ul {
    list-style: none;
```

```
    display: flex;
    gap: 1.5rem;
    margin: 0;
    padding: 0;
}
nav ul li a {
    color: #fff;
    text-decoration: none;
    font-weight: 500;
    transition: color 0.3s;
}
nav ul li a:hover {
    color: #ffd166;
}
.hero {
    background: linear-gradient(rgba(0,0,0,0.5), rgba(0,0,0,0.5)),
                url('https://via.placeholder.com/1200x400') center/cover no-repeat;
    color: #fff;
    text-align: center;
    padding: 6rem 2rem;
    position: relative;
}
.hero h1 {
    font-size: 3.5rem;
    margin-bottom: 1rem;
    color: #ffd166;
```

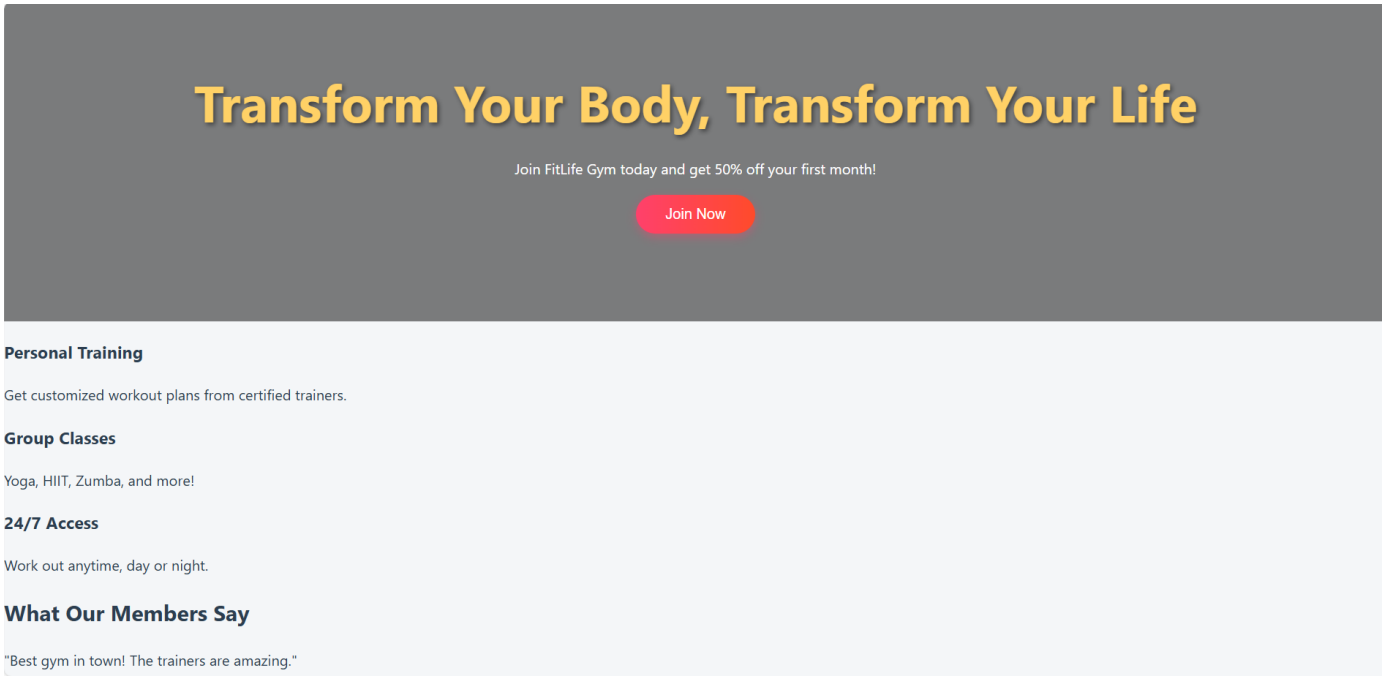
```
    text-shadow: 2px 2px 5px rgba(0,0,0,0.5);
}
.cta-button {
    background: linear-gradient(to right, #ff416c, #ff4b2b);
    color: #fff;
    border: none;
    padding: 0.75rem 2rem;
    font-size: 1rem;
    cursor: pointer;
    border-radius: 25px;
    transition: background 0.3s, transform 0.2s;
    box-shadow: 0 4px 10px rgba(255, 65, 108, 0.4);
}
.cta-button:hover {
    transform: scale(1.05);
    background: linear-gradient(to right, #ff4b2b, #ff416c);
}
.product-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(260px, 1fr));
    gap: 2rem;
    padding: 2rem;
}
.product-card {
    background: #fff;
    border: none;
```

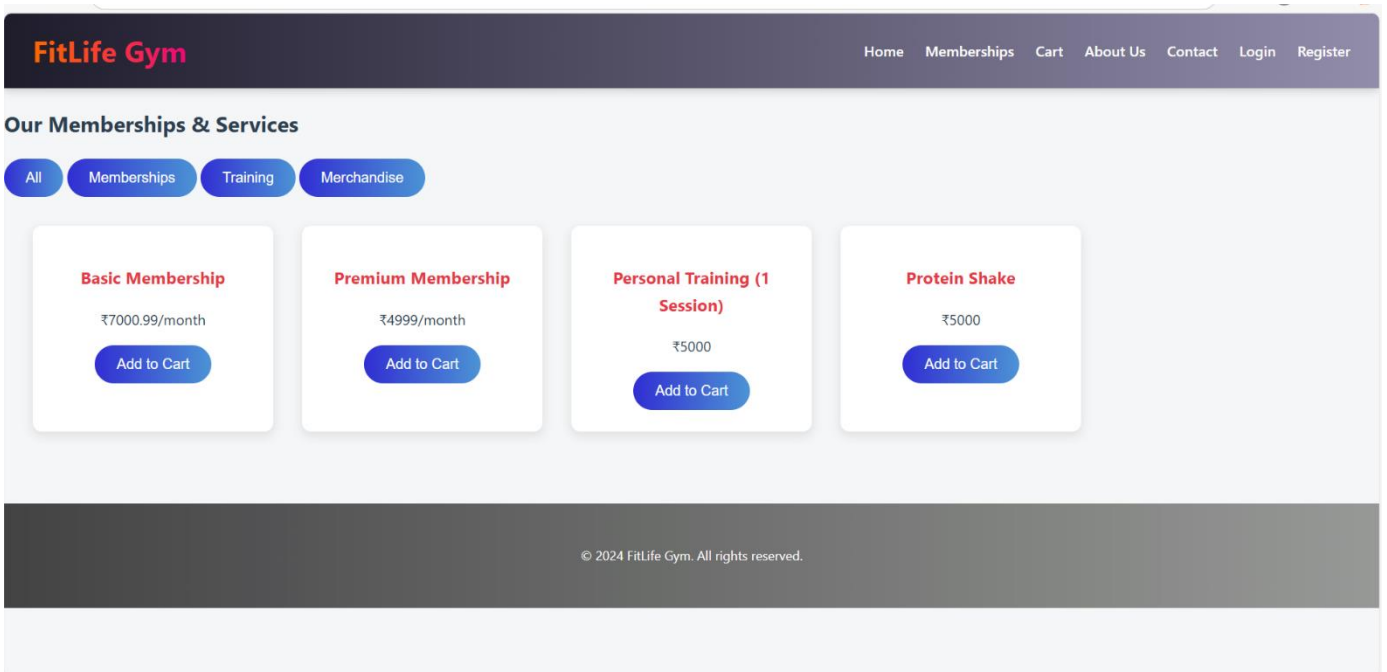
```
    padding: 1.5rem;
    border-radius: 10px;
    text-align: center;
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);
    transition: transform 0.3s, box-shadow 0.3s;
}
.product-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 6px 18px rgba(0,0,0,0.15);
}
.product-card h3 {
    color: #e63946;
}
.product-card img {
    max-width: 100%;
    border-radius: 8px;
    margin-bottom: 1rem;
}
/* Cart Page */
.cart-items {
    margin: 2rem;
}

.cart-summary {
    border-top: 2px solid #ccc;
    padding: 1rem 2rem;
```

```
    text-align: right;
    background: #fff;
    border-radius: 8px;
    margin: 1rem 2rem;
}
/* Forms */
.auth-form,
.contact-form {
    max-width: 500px;
    margin: 2rem auto;
    padding: 2rem;
    background: #ffffff;
    border-radius: 12px;
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.05);
}
input, textarea, select {
    width: 100%;
    padding: 0.75rem;
    margin-bottom: 1.25rem;
    border: 1px solid #ccc;
    border-radius: 6px;
    font-size: 1rem;
    transition: border-color 0.3s, box-shadow 0.3s;
}
input:focus, textarea:focus, select:focus {
    border-color: #ff4b2b;
```

```
    box-shadow: 0 0 5px rgba(255, 75, 43, 0.5);
    outline: none;
}
button {
    background: linear-gradient(to right, #322fd3, #4d94d6);
    color: #fff;
    border: none;
    padding: 0.75rem 1.5rem;
    font-size: 1rem;
    cursor: pointer;
    border-radius: 30px;
    transition: background 0.3s, transform 0.2s;
}
button:hover {
    background: linear-gradient(to right, #2e45d9, #2740bb);
    transform: translateY(-2px);
}
/* Footer */
footer {
    background: linear-gradient(to right, #434343, #696b67aa);
    color: #fff;
    text-align: center;
    padding: 2rem 1rem;
    margin-top: 3rem;
    font-size: 0.9rem;
}
```

OUTPUT


Transform Your Body, Transform Your Life
Join FitLife Gym today and get 50% off your first month!
Join Now
Personal Training
Get customized workout plans from certified trainers.
Group Classes
Yoga, HIIT, Zumba, and more!
24/7 Access
Work out anytime, day or night.
What Our Members Say
"Best gym in town! The trainers are amazing."


FitLife Gym
Home
Memberships
Cart
About Us
Contact
Login
Register
Our Memberships & Services
All
Memberships
Training
Merchandise
Basic Membership
₹7000.99/month
Add to Cart
Premium Membership
₹4999/month
Add to Cart
Personal Training (1 Session)
₹5000
Add to Cart
Protein Shake
₹5000
Add to Cart
© 2024 FitLife Gym. All rights reserved.

Experiment No.4

CSS

A. Enhance the cart page to make it user-friendly and visually appealing. Style the cart items with appropriate margins, paddings, and input field styles to provide a seamless shopping experience.

B. Enhance and style the about us page with appropriate margins, paddings, and input field styles.

C. Enhance and style the contact page to make it user-friendly and visually appealing. Style the contact form with appropriate margins, paddings, and input field styles.

D. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles.

E. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles

Objective

To improve the usability, consistency, and visual appeal of key web pages (Cart, About Us, Contact, Registration, and Login) by applying CSS properties such as margin, padding, border radius, and input styling.

Theory

CSS (Cascading Style Sheets) is the key technology used to control the appearance and layout of web pages. For a seamless and visually appealing user experience, CSS enables developers to fine-tune elements such as margins, paddings, fonts, colors, and input field styles. When designing pages like the **Cart Page**, **About Us Page**, **Contact Page**, and **Registration/Login Forms** on a website, CSS can be employed to make the design both functional and aesthetically pleasing. This enhances the user experience by ensuring ease of navigation, clarity of information, and a modern, cohesive look.

1. Enhancing the Cart Page

The **Cart Page** is one of the most critical parts of an e-commerce website, as it allows users to review their selected items before proceeding to checkout. A well-styled cart page should offer clear information and easy-to-use features.

Key Styles for Cart Page:

- **Margins and Padding:** Proper spacing between cart items, headers, and buttons is crucial for a neat layout. For instance, each item in the cart should have sufficient padding between it and other items to avoid a cramped look. A margin around the entire cart area ensures that the content doesn't feel crowded on the page.
- **Input Fields and Buttons:** Cart pages typically include input fields for adjusting quantities or removing items. These fields should be styled clearly, with appropriate padding to ensure they are easy to interact with. Buttons like "Proceed to Checkout" should be prominent with clear colors, padding, and hover effects for better engagement.

2. Enhancing the About Us Page

The **About Us Page** tells the story of the business, its mission, values, and team members. This page should be inviting, visually balanced, and easy to read. Proper use of margins, padding, and typography enhances readability and ensures users can easily scan through the content.

Key Styles for About Us Page:

- **Margins and Padding:** Adequate spacing between headings, images, and paragraphs is essential to avoid a cluttered layout. A good rule of thumb is to use generous padding inside sections and margin between sections to provide breathing space.

3. Enhancing the Contact Page

The **Contact Page** allows users to get in touch with the business. A well-styled contact page improves user experience and makes it easy for visitors to reach out. The contact form should be clearly structured, with appropriately spaced input fields and buttons.

Key Styles for Contact Page:

- **Margins and Padding:** Spacing around the form, as well as between the form fields, makes the page less overwhelming and improves accessibility. Fields should have adequate padding to make them easy to interact with, and the submit button should stand out.

By styling the form inputs, buttons, and overall layout, the contact page becomes more visually appealing and functional.

4. Enhancing the Admin/User Registration Form

The **Registration Form** is essential for capturing user details during account creation. The form should be easy to fill out, with clearly styled fields for each input type.

Key Styles for Registration Form:

- **Margins and Padding:** Form fields should have appropriate padding to ensure the text is easy to read and select. The submit button should be large and clearly labeled.

5. Enhancing the Admin/User Login Form

The **Login Form** should be simple but stylish. It must allow users to input their credentials quickly, so input fields and buttons should be optimized for easy interaction.

Key Styles for Login Form:

- **Margins and Padding:** Similar to the registration form, input fields should have sufficient space around them, and the button should be large enough to attract attention.

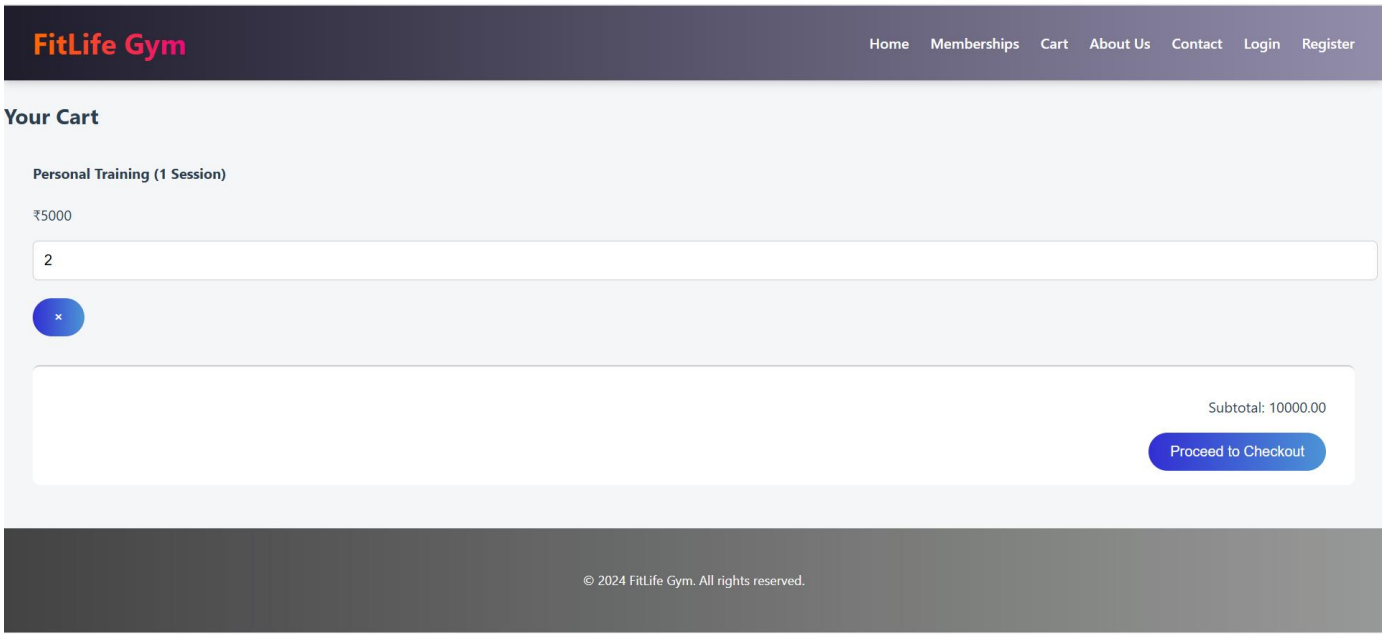
Conclusion

CSS plays an essential role in enhancing the visual appeal and usability of web pages. For pages like the **Cart**, **About Us**, **Contact**, and **Registration/Login Forms**, careful attention to margins, padding, and input field styles improves the overall user experience. These changes create a visually organized layout, making it easy for users to navigate, interact with forms, and engage with the website content. With well-designed forms and content areas, the website becomes both functional and aesthetically pleasing, leading to a better overall experience for the user.

CODE

A. /* Cart Page */

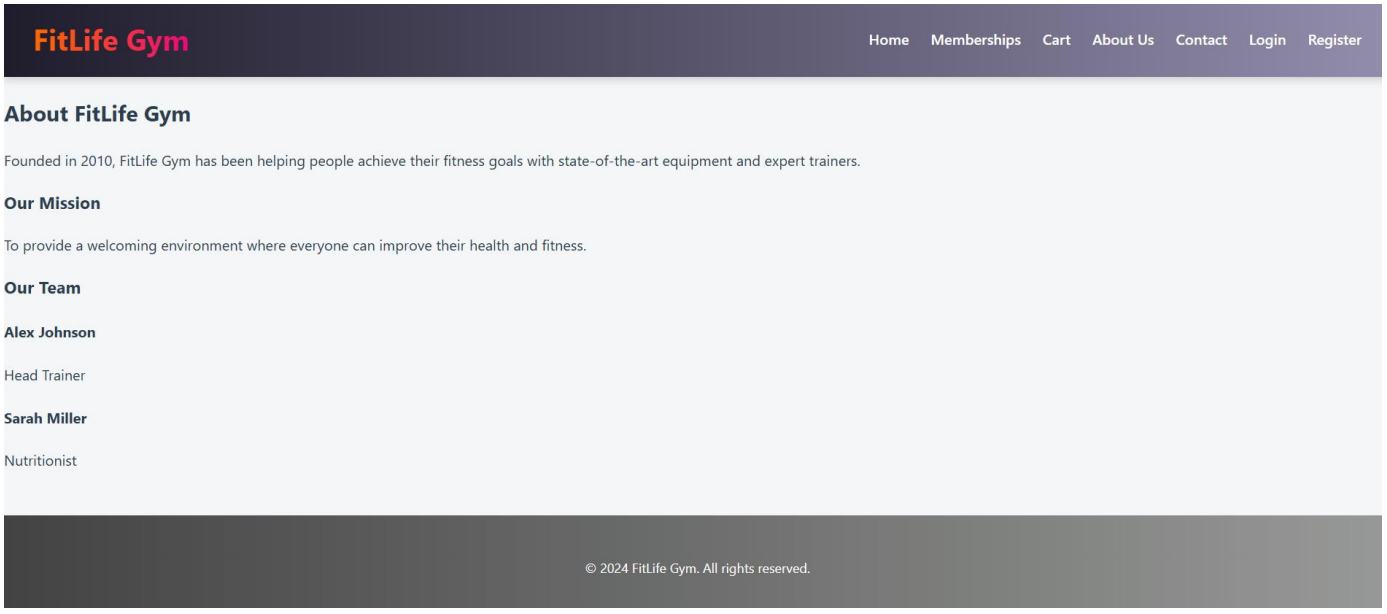
```
.cart-items {
   margin: 2rem;
}
.cart-summary {
   border-top: 2px solid #ccc;
   padding: 1rem 2rem;
   text-align: right;
   background: #fff;
   border-radius: 8px;
   margin: 1rem 2rem;
}
```



B. About Us Page

```
/* Forms */
.auth-form,
.contact-form {
   max-width: 500px;
   margin: 2rem auto;
```

```
  padding: 2rem;
  background: #ffffff;
  border-radius: 12px;
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.05);
}

input, textarea, select {
  width: 100%;
  padding: 0.75rem;
  margin-bottom: 1.25rem;
  border: 1px solid #ccc;
  border-radius: 6px;
  font-size: 1rem;
  transition: border-color 0.3s, box-shadow 0.3s;
}

input:focus, textarea:focus, select:focus {
  border-color: #ff4b2b;
  box-shadow: 0 0 5px rgba(255, 75, 43, 0.5);
  outline: none;
}
```

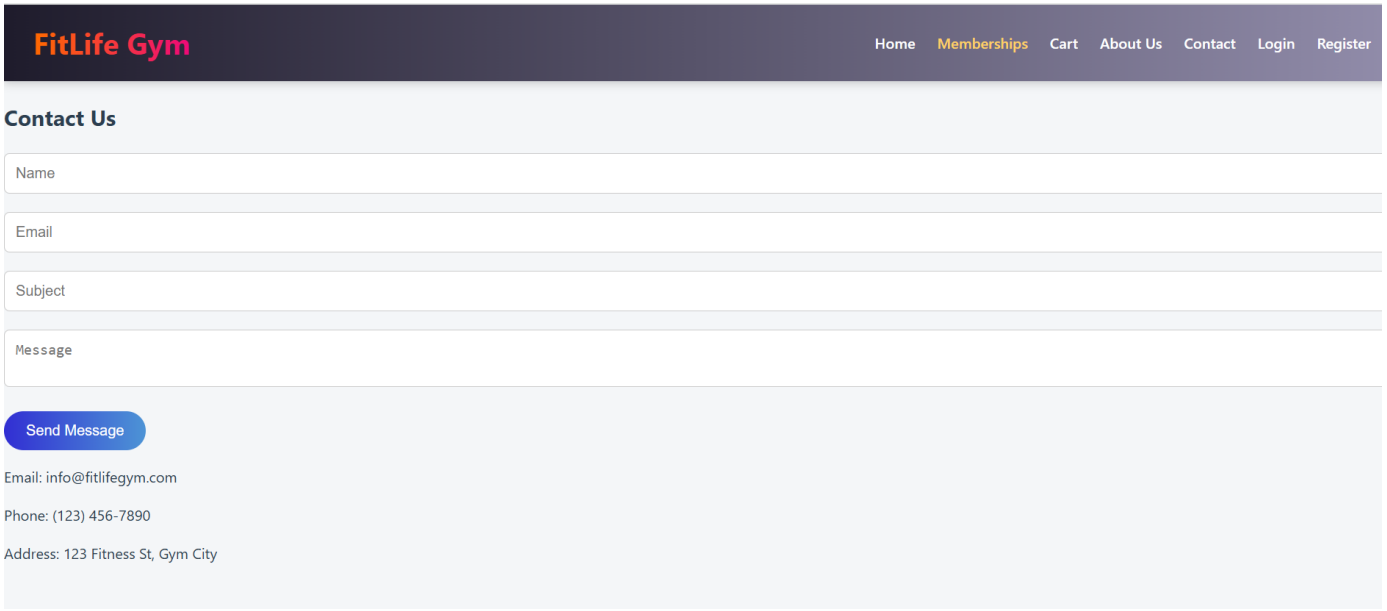


C.ContactUS

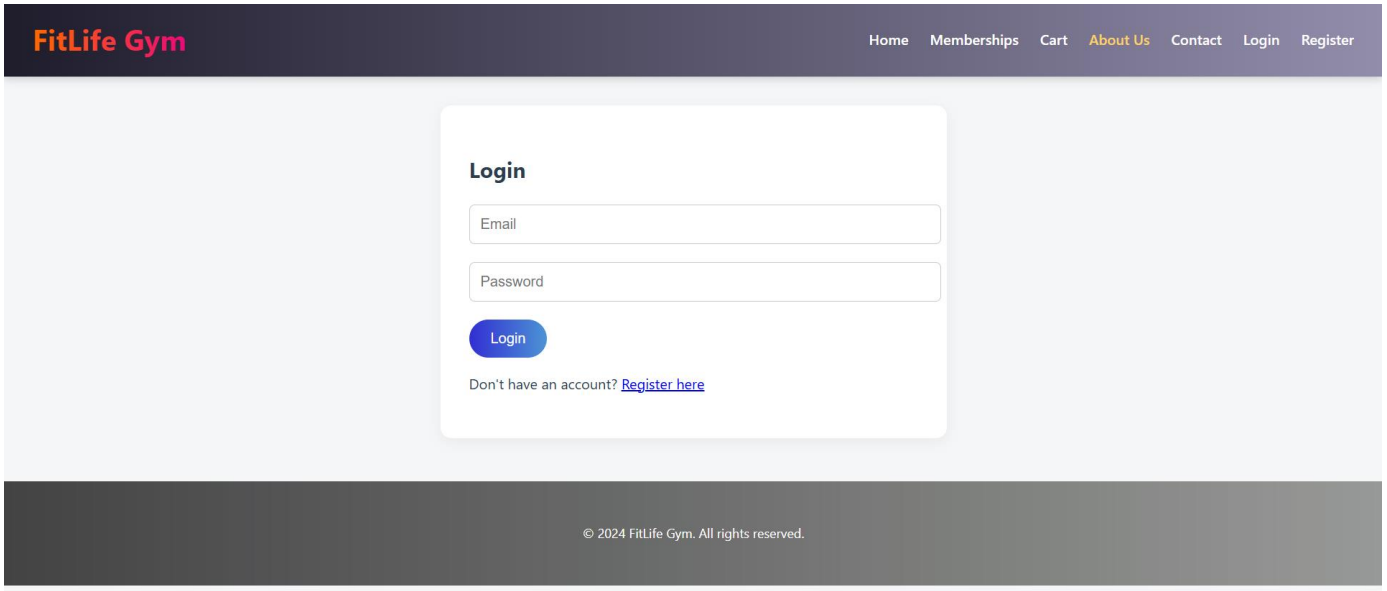
```
button:hover {
  background: linear-gradient(to right, #2e45d9, #2740bb);
  transform: translateY(-2px);
}

/* Footer */
footer {
```

```
    background: linear-gradient(to right, #434343, #696b67aa);
    color: #fff;
    text-align: center;
    padding: 2rem 1rem;
    margin-top: 3rem;
    font-size: 0.9rem;
}
```

Output



D.LoginPage



E. Register Page

Experiment No.5

JavaScript

A. Implement user registration and login forms for the Gym Management website. These forms will allow users to create an account, log in, and access personalized features, such as saving favorite items or viewing order history.

User Registration Form will allow new customers to sign up and create an account on the website. The form will capture basic user details, including the name, email address, and password (not limited to these fields).

User Login Form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.

B. Provide validations for user registration and login forms to validate the input to ensure that all required fields are filled and that the email format is valid. (**Contents beyond Syllabus)**

C. Develop cart functionality to allow users to add items, update quantities, and remove items.

Objective

To design and implement JavaScript-based front-end functionality for a Gym Management System website that:

- Enables new users to register using a form capturing essential personal and login details.
- Allows registered users to log in securely and access personalized features such as tracking progress, saving workout preferences, or managing bookings.

Theory

User Registration and Login Forms for a Gym Management Website

The development of a user registration and login system is crucial for any modern website, including a Gym Management website. This system enables customers to create personal accounts, log in to access personalized features, and enjoy a smoother user experience. In this explanation, we'll walk through the components of both the user registration and login forms, their validation processes, and the cart functionality.

User Registration Form

The **user registration form** allows new customers to sign up for the website. It captures essential details, such as the user's name, email address, password, and other information necessary for creating an account.

Key Fields in the User Registration Form:

1. **Name**: The user's full name is required to personalize the experience and for user identification.
2. **Email Address**: A unique identifier for each user. The email must be validated to ensure it follows a proper email format (e.g., user@example.com).
3. **Password**: A secure password that the user will use to log into the system. Strong password requirements such as a minimum length and a mix of letters, numbers, and special characters should be enforced for security.

4. **Confirm Password**: A field to confirm that the user has entered the correct password.

Validation for the Registration Form:

To ensure data integrity and security, the registration form should include various validations:

- **Required Fields**: Ensure that no field is left empty, including the name, email, and password fields.
- **Email Format**: The email field should only accept valid email formats (e.g., user@example.com). This can be validated using a regular expression or HTML5 input types.
- **Password Strength**: The password field should enforce security standards, such as a minimum of eight characters, at least one uppercase letter, one lowercase letter, one number, and one special character.

User Login Form

The **user login form** is where registered users authenticate themselves to access their account. It requires the user to input their email address and password to proceed. This form is often displayed when users want to access personalized features, such as viewing order history or managing favorite items.

Key Fields in the User Login Form:

1. **Email Address**: This field helps identify the user. It must be validated to ensure the input follows the correct email format.
2. **Password**: The password entered by the user must match the one stored in the system for successful login.

Validation for the Login Form:

Login form validation mainly ensures that both fields are filled and that the email is in the correct format.

- **Required Fields**: Ensure that the email and password fields are not empty.
- **Email Format**: Validate that the email input follows the correct format.

Cart Functionality

In addition to user registration and login features, the Gym Management website will require cart functionality. This feature allows users to select and purchase coffee or other products, view the items in their cart, update quantities, and remove items.

Key Components of the Cart:

1. **Add Items to Cart**: Users can select coffee or other items and add them to their cart. Each item will typically have a name, price, and quantity.
2. **Update Quantities**: Users should be able to adjust the quantity of an item in the cart. For example, a user can add more cups of coffee or reduce the quantity.
3. **Remove Items**: Users should have the option to remove an item from their cart if they no longer want it.

CONCLUSION

Implementing user registration and login forms along with cart functionality is essential for a

seamless and personalized experience on a Gym Management website. The registration form ensures that users can create accounts securely, while the login form allows them to access personalized features. Additionally, cart functionality makes it easy for users to add, update, and remove items, which enhances the shopping experience. Proper input validation ensures that users' data is accurate, secure, and prevents issues such as incorrect email formats or mismatched passwords. This system improves both user engagement and customer retention for the Gym Management website.

Javascript

```
// ===== Cart System =====
let cart = JSON.parse(localStorage.getItem('cart')) || [];
// Add to Cart Functionality
document.addEventListener('DOMContentLoaded', function() {
 // Add to Cart Buttons
 document.querySelectorAll('.add-to-cart').forEach(button => {
  button.addEventListener('click', function(e) {
   const productCard = e.target.closest('.product-card');
   const product = {
    id: productCard.dataset.id,
    name: productCard.querySelector('h3').textContent,
    price: productCard.querySelector('.price').textContent,
    quantity: 1
   };
   addToCart(product);
   updateCartCount();
   showCartNotification(product.name);
  });
 });

 // Update Cart Display if on Cart Page
 if (document.querySelector('.cart-items')) {
```

```
    renderCartItems();
    setupCartControls();
  }
  // Theme Toggle
  const themeToggle = document.getElementById('themeToggle');
  if (themeToggle) {
    themeToggle.addEventListener('click', () => {
      document.body.setAttribute('data-theme',
        document.body.getAttribute('data-theme') === 'light' ? 'dark' : 'light'
      );
    });
  }
});
// Cart Functions
function addToCart(product) {
  const existingItem = cart.find(item => item.id === product.id);
  if (existingItem) {
    existingItem.quantity += 1;
  } else {
    cart.push(product);
  }
  saveCart();
}
function removeFromCart(productId) {
  cart = cart.filter(item => item.id !== productId);
  saveCart();
  renderCartItems();
}
function updateQuantity(productId, newQuantity) {
```

```
  const item = cart.find(item => item.id === productId);
  if (item) {
    item.quantity = parseInt(newQuantity);
    if (item.quantity <= 0) {
      removeFromCart(productId);
    } else {
      saveCart();
      renderCartItems();
    }
  }
}
function saveCart() {
  localStorage.setItem('cart', JSON.stringify(cart));
}
function getCartTotal() {
  return cart.reduce((total, item) => {
    return total + (parseFloat(item.price.replace(/[^0-9.]/g, '')) * item.quantity);
  }, 0);
}
// UI Updates
function renderCartItems() {
  const cartItemsContainer = document.querySelector('.cart-items');
  const cartSummary = document.querySelector('.cart-summary');
  if (!cartItemsContainer) return;
  cartItemsContainer.innerHTML = '';
  if (cart.length === 0) {
    cartItemsContainer.innerHTML = '<p class="empty-cart">Your cart is empty</p>';
    cartSummary.innerHTML = '<p>Subtotal: $0.00</p>';
    return;
```

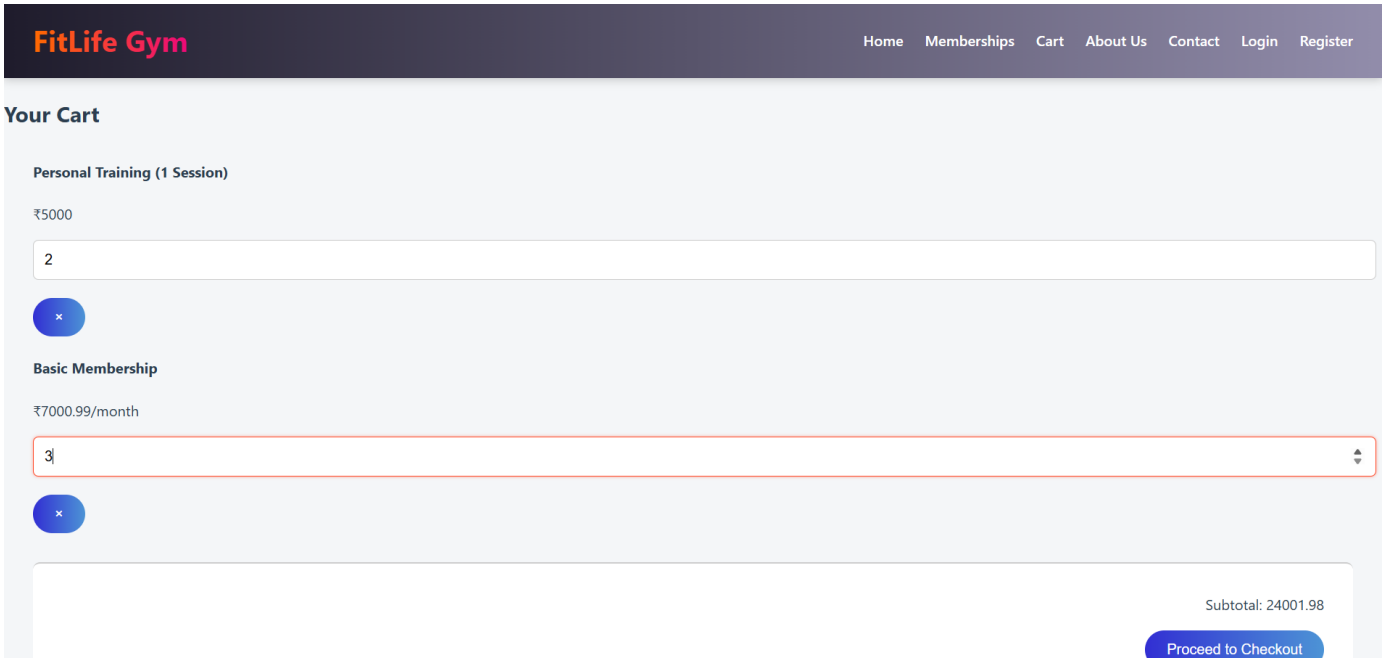
```
  }
  cart.forEach(item => {
    const cartItem = document.createElement('div');
    cartItem.className = 'cart-item';
    cartItem.innerHTML = `
      <div class="item-info">
        <h4>${item.name}</h4>
        <p>${item.price}</p>
      </div>
      <div class="item-controls">
        <input type="number" min="1" value="${item.quantity}"
              data-id="${item.id}" class="quantity-input">
        <button class="remove-item" data-id="${item.id}">×</button>
      </div>
    `;
    cartItemsContainer.appendChild(cartItem);
  });
  cartSummary.innerHTML = `
    <p>Subtotal: ${getCartTotal().toFixed(2)}</p>
    <button class="checkout-btn">Proceed to Checkout</button>
  `;
}
function setupCartControls() {
  // Quantity changes
  document.querySelectorAll('.quantity-input').forEach(input => {
    input.addEventListener('change', (e) => {
      updateQuantity(e.target.dataset.id, e.target.value);
    });
  });
```

```
  // Remove buttons
  document.querySelectorAll('.remove-item').forEach(button => {
    button.addEventListener('click', (e) => {
      removeFromCart(e.target.dataset.id);
    });
  });
}
function updateCartCount() {
  const cartCount = document.getElementById('cart-count');
  if (cartCount) {
    const totalItems = cart.reduce((sum, item) => sum + item.quantity, 0);
    cartCount.textContent = totalItems;
    cartCount.style.display = totalItems > 0 ? 'flex' : 'none';
  }
}

function showCartNotification(productName) {
  const notification = document.createElement('div');
  notification.className = 'cart-notification';
  notification.innerHTML = `
    <p>${productName} added to cart!</p>
  `;
  document.body.appendChild(notification);

  setTimeout(() => {
    notification.classList.add('show');
  }, 10);

  setTimeout(() => {
```

```
    notification.classList.remove('show');
    setTimeout(() => {
      notification.remove();
    }, 300);
  }, 3000);
}

// Form Submissions
document.getElementById('contact-form')?.addEventListener('submit', (e) => {
  e.preventDefault();
  alert('Message sent!');
  e.target.reset();
});

document.getElementById('register-form')?.addEventListener('submit', (e) => {
  e.preventDefault();
  alert('Registration successful!');
  window.location.href = 'login.html';
});

document.getElementById('login-form')?.addEventListener('submit', (e) => {
  e.preventDefault();
  alert('Login successful!');
  window.location.href = 'index.html';
});
// Add this at the end of the DOMContentLoaded event listener
document.addEventListener('DOMContentLoaded', function() {
    // ... existing code ...
```

```
  // Initialize cart count
  updateCartCount();
});
```

OUTPUT



Experiment No.6

JavaScript

A. The user login form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.

B. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus)**

C. Use localStorage or sessionStorage to store authentication data, such as the user's email and login status. This ensures that once a user is logged in, they remain authenticated even after the page reloads or when they visit the site again. (**Contents beyond Syllabus)**

D. Save the cart data to local storage when items are added, updated, or removed. Retrieve and load the cart data from local storage when the page loads. (**Contents beyond Syllabus)**

Objective

To implement a front-end JavaScript solution for managing user login state and shopping cart data in a Gym Management System website that:

- Authenticates users using email and password via the login form.
- Stores login session details using localStorage or sessionStorage to persist authentication state.

Theory

Implementing a User Login System and Cart Functionality Using localStorage or sessionStorage

In modern web development, enhancing user experience by retaining user login status and cart data across page reloads and future visits is crucial. By utilizing the localStorage or sessionStorage objects in JavaScript, we can store essential data on the client-side, allowing for seamless interactions between users and the website. This process involves handling user login, saving authentication data, and managing cart data efficiently.

User Login Form

The **user login form** allows registered users to log into their accounts, typically requiring an email address and password. After a successful login, the user should be redirected to a homepage or a personalized dashboard. This functionality ensures that users can securely access their accounts without having to repeatedly enter credentials during each visit.

Key Components of the User Login Process:

1. **Email and Password Authentication**: The login form must accept the user's email and password. The server or local storage (if pre-saved credentials are used) verifies these details. If the entered credentials match the stored records, the login process is successful.

2. **Redirection After Login**: Once the login credentials are validated, the user should be redirected to a personalized page, such as the homepage or dashboard, where they can access personalized features like order history, saved favorites, and more.
3. **Session Management**: To maintain a user's login state after a page refresh or subsequent visits, data can be stored in either localStorage or sessionStorage. The major difference between the two is that localStorage persists data across sessions, while sessionStorage stores data only for the duration of the page session. For user login status, localStorage is typically preferred because it allows the user to remain logged in even after closing and reopening the browser.

Using localStorage to Store Authentication Data:

In the login process, once the user logs in successfully, we can store their email address and login status (true or false) in the localStorage. Here's an example of how to implement this:

```
function loginUser() {
 let email = document.getElementById('email').value;
 let password = document.getElementById('password').value;

 // Example login credentials stored locally for demo purposes
 let storedEmail = "user@example.com";
 let storedPassword = "password123";

 // Simple authentication
 if (email === storedEmail && password === storedPassword) {
  // Store authentication data in localStorage
  localStorage.setItem('userEmail', email);
  localStorage.setItem('isLoggedIn', 'true');

  // Redirect to homepage or user dashboard
  window.location.href = "/dashboard";
 } else {
  alert("Invalid email or password.");
 }
```

}

Cart Functionality with localStorage

Alongside user authentication, the cart functionality is an integral part of an e-commerce website. Users should be able to add, update, and remove items from their cart, and this data should be saved across sessions so that users don't lose their cart content when they reload the page or return to the website.

Key Components of Cart Functionality:

1. **Add Items to Cart**: When a user adds an item to their cart, the system should update the cart data stored in localStorage with the new item.
2. **Update Cart Items**: Users should be able to update the quantity of items in their cart. This update should be reflected in the localStorage to ensure that the changes persist across page reloads.
3. **Remove Items from Cart**: If a user decides to remove an item, this change should also be saved to localStorage to maintain consistency when the user revisits the cart.
4. **Retrieve Cart Data on Page Load**: When the page loads, the cart data should be retrieved from localStorage, ensuring that the user sees the correct items and quantities in their cart.

Storing and Retrieving Cart Data:

Here's an example of how to implement cart functionality using localStorage:

```
// Add item to cart
function addToCart(item) {
 let cart = JSON.parse(localStorage.getItem('cart')) || [];

 // Check if item already exists in cart
 let existingItem = cart.find(cartItem => cartItem.id === item.id);
 if (existingItem) {
  existingItem.quantity += 1;  // Increase quantity if item exists
 } else {
  cart.push(item);  // Add new item
 }

 // Save updated cart to localStorage
```

```
  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

// Remove item from cart
function removeFromCart(itemId) {
  let cart = JSON.parse(localStorage.getItem('cart')) || [];
  cart = cart.filter(item => item.id !== itemId);  // Remove item
  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

// Update item quantity
function updateCartItemQuantity(itemId, newQuantity) {
  let cart = JSON.parse(localStorage.getItem('cart')) || [];
  let item = cart.find(cartItem => cartItem.id === itemId);
  if (item) {
    item.quantity = newQuantity;  // Update quantity
    localStorage.setItem('cart', JSON.stringify(cart));
  }
  displayCart();
}

// Display cart
function displayCart() {
  let cart = JSON.parse(localStorage.getItem('cart')) || [];
  let cartDisplay = document.getElementById('cartDisplay');
  cartDisplay.innerHTML = '';
```

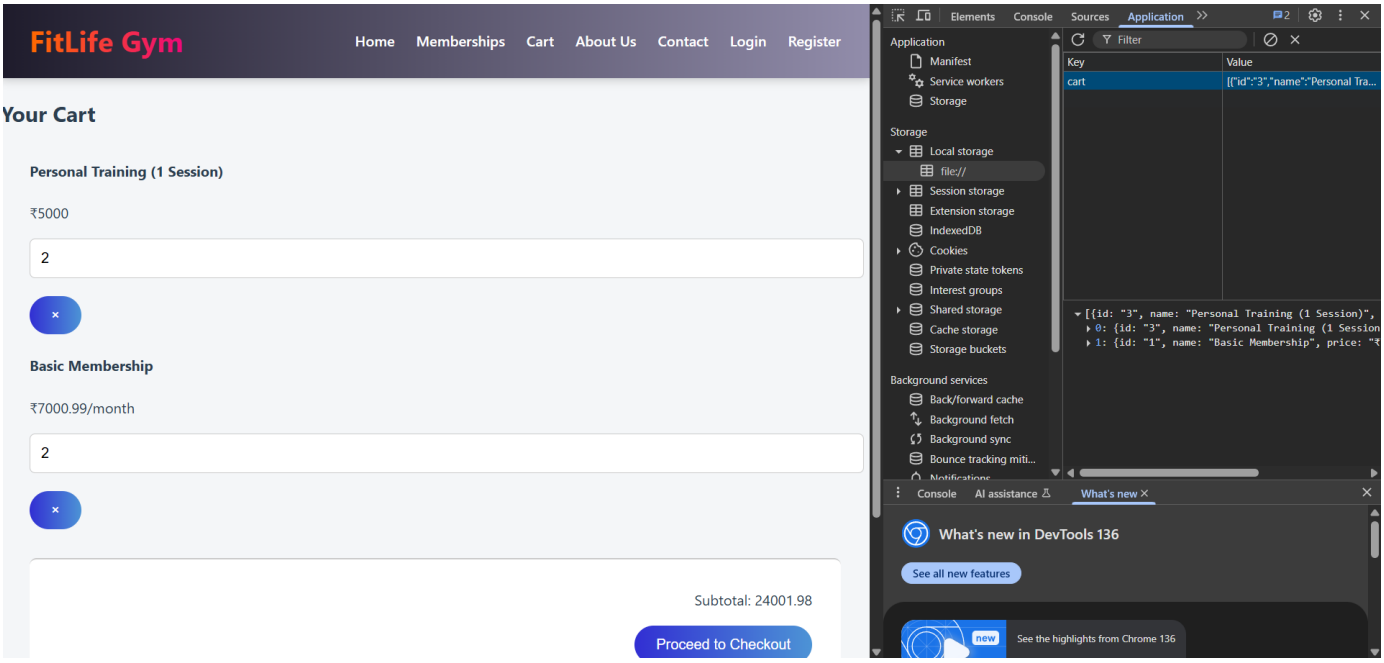
```
  cart.forEach(item => {
    cartDisplay.innerHTML += `<div>${item.name} - ${item.quantity} x $${item.price}</div>`;
  });
}

// Load cart data on page load
window.onload = function() {
  displayCart();
};
```

Conclusion

By using localStorage or sessionStorage, we can create a more seamless and personalized user experience. The login system ensures that users remain authenticated across sessions, while the cart functionality allows users to add, update, and remove items without losing their cart data between visits. Both localStorage and sessionStorage are invaluable tools in modern web development, enabling persistent data storage on the client side. While localStorage offers persistent data storage across browser sessions, sessionStorage is more transient, making it ideal for temporary data storage during a single browsing session. By combining these features, we can build efficient, user-friendly, and reliable web applications.

```
<!-- Login Page -->
<form id="loginForm">
    <input type="email" id="email" required>
    <input type="password" id="password" required>
    <button type="submit">Login</button>
</form>

<!-- Registration Page -->
<form id="registerForm">
    <input type="text" id="name" required>
    <input type="email" id="email" required>
```

```
    <input type="password" id="password" required>
    <select id="role">
        <option value="member">Member</option>
        <option value="admin">Admin</option>
    </select>
    <button type="submit">Register</button>
</form>

<!-- Navigation (All Pages) -->
<nav class="auth-links">
    <!-- Dynamically populated -->
</nav>
<span class="cart-count"></span>

<!-- Cart Page -->
<div class="cart-items"></div>
<div id="cart-total"></div>
```



Experiment No.7

PHP

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

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

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

Objective

To develop a secure and user-friendly registration system for the Gym Management website using PHP and MySQL, that:

- Collects necessary user details such as name, email, and password.
- Validates input data for accuracy and completeness.

Theory

Developing a PHP Script for User Registration on a Gym Management Website

User registration is a crucial component of any web application, enabling personalized experiences and account management. In the context of a **Gym Management website**, user registration allows gym members to create accounts, manage subscriptions, track fitness progress, and access other personalized services. This PHP script will guide the process of **user registration**, handling form inputs, validation, error reporting, and providing user feedback after successful registration.

The registration process can be broken down into **three major steps**: accepting user input, implementing error handling and validation, and providing feedback after registration.

A. Developing the PHP Script to Handle User Registration

When creating a registration system for the Gym Management website, we will need a **user registration form** that gathers the user's name, email, password, and possibly additional information like membership type and phone number. The form will send the data to a PHP script where the data will be processed and validated before being stored.

1. HTML Registration Form

Here's an example of an HTML form to gather user information:

```
<form action="register.php" method="POST">
  <label for="name">Full Name:</label>
```

```
    <input type="text" id="name" name="name" required><br><br>
    <label for="email">Email Address:</label>
    <input type="email" id="email" name="email" required><br><br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required><br><br>
    <label for="confirm_password">Confirm Password:</label>
    <input type="password" id="confirm_password" name="confirm_password" required><br><br>
    <button type="submit" name="submit">Register</button>
</form>
```

In this form:

- The user must enter their **name**, **email**, **password**, and **confirm password**.
- The form submits data to register.php using the **POST** method.

2. Handling Form Data in PHP

The PHP script register.php will receive the form data and process it. The script will access form data through the $_POST superglobal array. Here's an example:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = trim($_POST['name']);
    $email = trim($_POST['email']);
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
}
```

Once the data is captured, the next step is to validate it and ensure it meets certain criteria before proceeding to the database.

B. Implementing Error Handling and Validation

Validation is crucial to ensure that the user provides the correct information during registration. If validation fails, the script should provide informative error messages to guide the user in correcting their input.

1. Validating User Input

Here are the validation steps we need to implement:

- **Name Validation**: Ensure the name field is not empty.

- **Email Validation**: Ensure the email is in a valid format and is not already registered in the system.
- **Password Validation**: Ensure the password meets minimum length requirements and that the password and confirm password match.

Here's an example of how to validate the user input:

```
$errors = [];

if (empty($name)) {
    $errors[] = "Full name is required.";
}

// Validate email
if (empty($email)) {
    $errors[] = "Email is required.";
} elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errors[] = "Invalid email format.";
} else {
    // Check if email already exists in the database
    $query = "SELECT email FROM users WHERE email = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        $errors[] = "Email is already registered.";
    }
}

if (empty($password)) {
    $errors[] = "Password is required.";
} elseif (strlen($password) < 8) {
    $errors[] = "Password must be at least 8 characters long.";
}
```

```
// Confirm password match
if ($password !== $confirm_password) {
   $errors[] = "Passwords do not match.";
}
```

In this validation block:

- The filter_var() function checks if the email is in a valid format.
- A query is executed to check if the email already exists in the database to avoid duplicate entries.
- Password length is checked to ensure a secure password.
- The password and confirm password fields are checked for equality.

2. Handling Database Errors

If the input is valid, the data will be inserted into the database. If the database connection fails or an issue occurs during insertion, the script should handle errors gracefully. For example, using prepared statements helps prevent SQL injection attacks.

```
// Hash the password for security
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
$query = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";
$stmt = $conn->prepare($query);
$stmt->bind_param("sss", $name, $email, $hashed_password);

if (!$stmt->execute()) {
   $errors[] = "Error occurred while registering the user.";
}
```

C. Providing Feedback to the User

Once the data has been successfully validated and inserted into the database, it's time to **provide feedback to the user**. This can be done by either displaying a success message or redirecting the user to the login page.

1. Successful Registration

If the registration is successful, the user should be notified, and a **confirmation message** should be displayed, or they should be redirected to the login page:

```
if (empty($errors)) {
   echo "Registration successful!";
   // Redirect to the login page after 2 seconds
```

```
    header("refresh:2;url=login.php");

    exit();

}
```

2. Redirecting to Login Page

Alternatively, if the registration is successful, redirect the user to the login page for them to log in with the credentials they just created. Here's an example of redirecting after successful registration:

```
header("Location: login.php");  // Redirect to the login page

exit();
```

3. Error Feedback

If there are any validation errors, those should be displayed to the user so they can correct the issues. For example:

Conclusion

Creating a user registration system for a **Gym Management website** using PHP involves accepting and validating user input, handling errors, securely storing user data, and providing appropriate feedback after the registration process.

- **Input Validation**: Ensures the data is correct (valid email format, password strength, matching passwords, etc.).
- **Error Handling**: Handles validation errors like empty fields, invalid email format, and password mismatches, displaying user-friendly messages.
- **Feedback**: After successful registration, the user should either receive a confirmation message or be redirected to the login page.

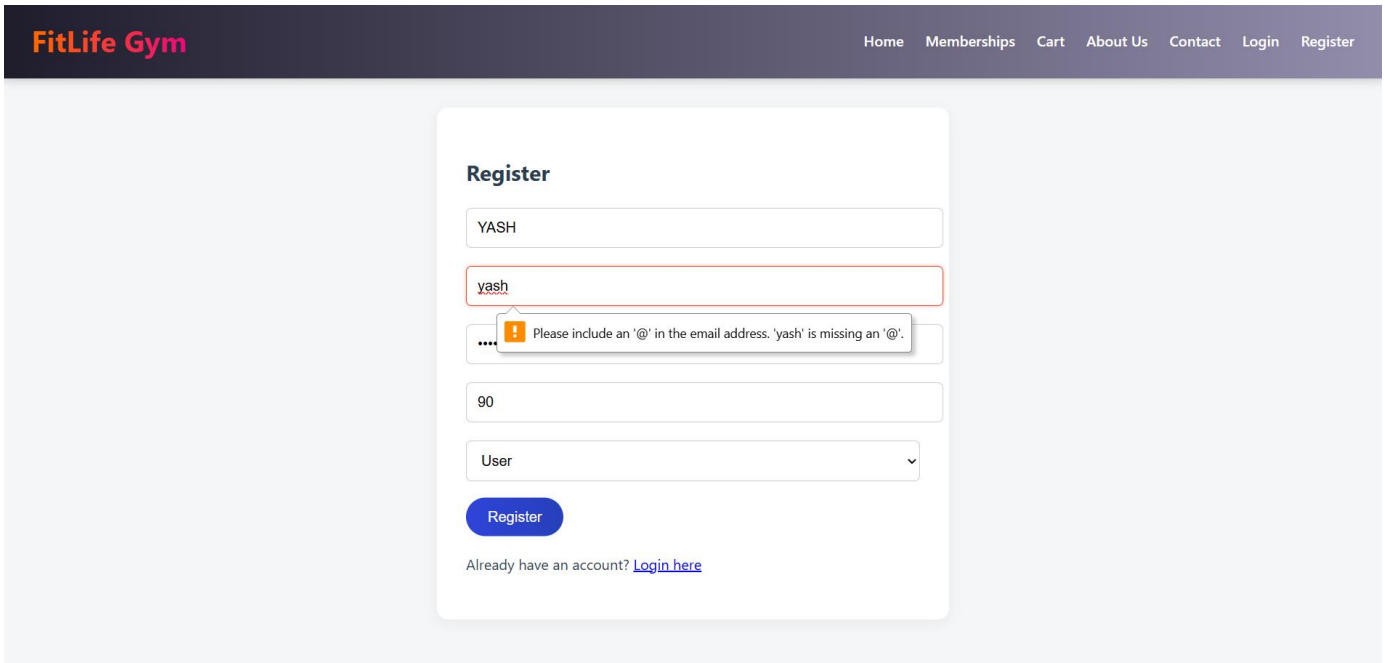
By carefully following these steps, you ensure that the registration process is secure, user-friendly, and easy to navigate for new gym members.

PYTHON

```
<?php

// php/register.php

include 'db_connect.php';

// Check if form submitted

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

    $name = trim($_POST['name']);

    $email = trim($_POST['email']);
```

```
$password = trim($_POST['password']);
$errors = [];
// Basic validation
if (empty($name)) {
    $errors[] = "Name is required.";
}
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errors[] = "Invalid email format.";
}
if (strlen($password) < 6) {
    $errors[] = "Password must be at least 6 characters.";
}
// If no validation errors, proceed
if (empty($errors)) {
    // Hash password
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

    // Check for existing email
    $stmt = $conn->prepare("SELECT id FROM users WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->store_result();
    if ($stmt->num_rows > 0) {
        echo "<p>Email already registered. Try logging in.</p>";
    } else {
        // Insert user
        $stmt = $conn->prepare("INSERT INTO users (name, email, password) VALUES (?, ?, ?)");
        $stmt->bind_param("sss", $name, $email, $hashedPassword);
        if ($stmt->execute()) {
```

```
            header("Location: ../login.html?registered=success");
            exit();
        } else {
            echo "<p>Error during registration. Please try again.</p>";
        }
    }
    $stmt->close();
  } else {
    // Display all validation errors
    foreach ($errors as $error) {
        echo "<p style='color:red;'>$error</p>";
    }
  }
}
?>
```



Experiment No.8

PHP

A. Develop a PHP script to handle user login for the Gym Management website. The script should accept input from users for their login credentials. (all required fields for login).

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

C. Implement error handling to notify users of login failures due to incorrect credentials or other errors.

D. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page.

Objective

To develop a secure and efficient login system using PHP and MySQL for the Gym Management System that:

- Allows registered users to log in using their email and password.
- Authenticates user credentials securely against stored records.
- Provides real-time feedback on successful or failed login attempts.
- Redirects authenticated users to a personalized dashboard or homepage.
- Ensures session-based access control for further website functionality.

Theory

PHP Script for User Login in a Gym Management Website

A secure and efficient login system is a core component of any Gym Management website. It enables registered users, such as gym members or staff, to access their personalized dashboards where they can manage profiles, track workouts, book training sessions, or monitor progress. The implementation of a login system using **PHP** involves collecting credentials from the user, verifying them against the stored data, managing session data, and providing clear feedback for both success and failure cases.

This explanation is divided into four major components:

- **A. Developing the PHP script to handle user login**
- **B. Providing feedback upon successful login**
- **C. Implementing error handling for login failures**
- **D. Delivering personalized feedback or redirection upon success**

A. Developing the PHP Script to Handle User Login

The login system starts with an **HTML form** that collects the user's email address and password. The PHP script then processes these credentials and checks them against the records stored in the database.

HTML Form Example

```
<form action="login.php" method="POST">
  <input type="email" name="email" placeholder="Email Address" required>
  <input type="password" name="password" placeholder="Password" required>
  <button type="submit" name="login">Login</button>
</form>
```

This form uses the POST method to securely send user input to the PHP script (login.php).

B. Providing Feedback upon Successful Login

If the email exists and the password matches the stored hashed password, the user is successfully authenticated. PHP's password_verify() function is used to compare the plaintext password with the hashed one stored in the database.

C. Implementing Error Handling for Login Failures

In the event of incorrect input or invalid credentials, the login system should provide helpful and secure feedback without exposing sensitive data.

By using clear messages, users can understand what went wrong without revealing whether an email exists (which can be a security concern if handled improperly).

D. Delivering Personalized Feedback or Redirection

A personalized experience helps build user trust and engagement. Once authenticated, the system can:

- Display a **welcome message** including the user's name.
- Redirect the user to their **dashboard or homepage**.
- Log session data for features like viewing class schedules, progress reports, or messaging trainers.

This ensures that only logged-in users can access gym features such as profile updates, class bookings, or feedback forms.

Conclusion

A user login system for a Gym Management website is essential for secure member access and personalized services. Using PHP, the system should:

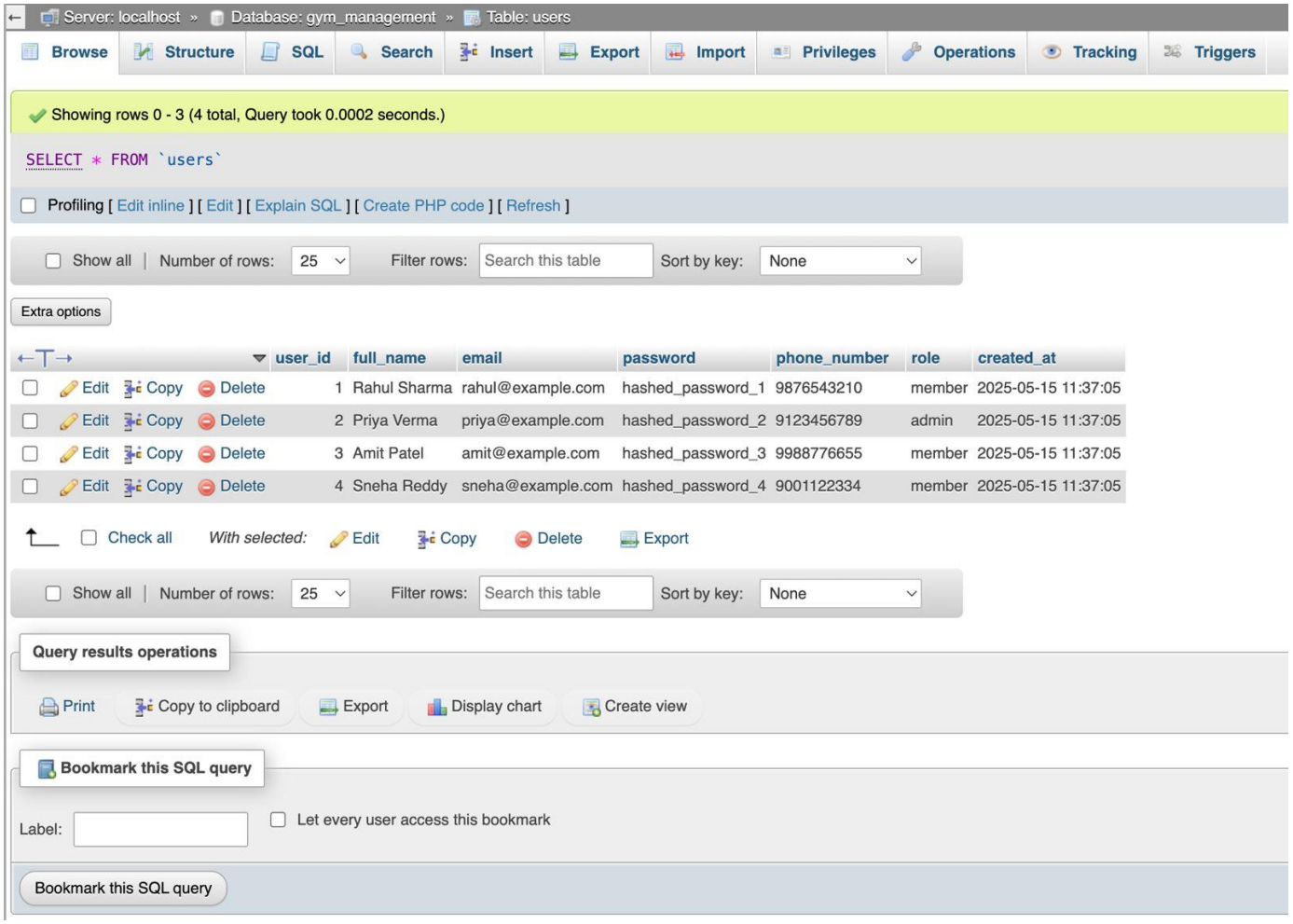
- Collect login credentials securely via a form.

- Verify the credentials against a hashed password in the database.
- Initiate user sessions upon successful login.
- Redirect users to their homepage or provide a welcome message.
- Handle and display errors when login fails.

With a well-implemented login system, the website becomes a reliable and secure portal for managing gym memberships, bookings, and communications. This system enhances user experience while ensuring data protection and accessibility.

```
<?php
// php/login.php
session_start();
include 'db_connect.php'; // Make sure this file contains your DB connection

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = trim($_POST['email'] ?? '');
    $password = trim($_POST['password'] ?? '');

    // Input validation
    if (empty($email) || empty($password)) {
        echo "<p style='color:red;'>Both email and password are required.</p>";
        exit();
    }

    // Fetch user from DB
    $stmt = $conn->prepare("SELECT id, name, password FROM users WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    // Verify credentials
```

```
    if ($result->num_rows === 1) {
        $user = $result->fetch_assoc();
        if (password_verify($password, $user['password'])) {
            // Set session
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_name'] = $user['name'];

            // Redirect or display welcome
            header("Location: ../dashboard.php?login=success");
            exit();
        } else {
            echo "<p style='color:red;'>Incorrect password. Try again.</p>";
        }
    } else {
        echo "<p style='color:red;'>No user found with that email address.</p>";
    }

    $stmt->close();
    $conn->close();
}
?>
```


Server: localhost » Database: gym_management » Table: users
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)
SELECT * FROM `users`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
user_id full_name email password phone_number role created_at
Edit Copy Delete 1 Rahul Sharma rahul@example.com hashed_password_1 9876543210 member 2025-05-15 11:37:05
Edit Copy Delete 2 Priya Verma priya@example.com hashed_password_2 9123456789 admin 2025-05-15 11:37:05
Edit Copy Delete 3 Amit Patel amit@example.com hashed_password_3 9988776655 member 2025-05-15 11:37:05
Edit Copy Delete 4 Sneha Reddy sneha@example.com hashed_password_4 9001122334 member 2025-05-15 11:37:05
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query

Experiment No.9

PHP and MySQL

A. Develop a PHP script that allows users to manage their shopping cart for an e-commerce website (e.g., a Gym Management store). The script should allow users to add items to their cart, view their cart contents, and remove items if needed.

B. Develop a PHP script to manage the shopping cart for an e-commerce website (e.g., a Gym Management 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.

Objective

To design and implement a PHP-based cart management system for a Gym Management website that enables users to:

- Add gym-related items (e.g., memberships, supplements, training programs) to their shopping cart.
- View the contents of their cart at any time.
- Remove individual items or clear the cart.
- Persist cart data in a MySQL database to maintain user selections across sessions.

Theory

A shopping cart system is crucial for an e-commerce website, such as a Gym Management site that sells products like gym equipment, apparel, supplements, or memberships. Allowing users to add, view, and remove items from their cart enhances the shopping experience, and using **PHP and MySQL** for server-side management ensures that data persists across sessions. Storing cart data in the database ensures that the user's cart is available even if they log out or visit from another device.

In this theory, we will explore how to develop a shopping cart system for a Gym Management website, using **PHP** for the server-side logic and **MySQL** for database storage. The system will allow users to:

- **Add items to the cart**
- **View cart contents**
- **Remove items from the cart**
- **Store cart data in MySQL to persist across sessions**

A. Database Design for the Shopping Cart

The first step in building a shopping cart system is to design the database schema. We need at least two tables: one for storing **product** details and another for managing the **shopping cart** itself.

1. Products Table

This table stores the details of the gym products available for purchase, such as gym equipment, apparel, and supplements.

Field	Type	Description
product_id	INT (PK)	Unique ID for the product
name	VARCHAR(255)	Name of the product
price	DECIMAL(10,2)	Price of the product
description	TEXT	Description of the product
stock	INT	Available stock of the product

Example SQL to create the products table:

sql

CopyEdit

```
CREATE TABLE products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    description TEXT,
    stock INT DEFAULT 0
);
```

2. Shopping Cart Table

This table stores the user's cart items. The cart is typically associated with a user, so it includes a user_id field. It will store product quantities, as well as the associated user and product IDs.

Field	Type	Description
cart_id	INT (PK)	Unique ID for each cart item
user_id	INT	ID of the user who owns the cart
product_id	INT	ID of the product added to the cart
quantity	INT	Quantity of the product in the cart
date_added	TIMESTAMP	The time when the product was added to the cart

Example SQL to create the shopping cart table:

```
CREATE TABLE cart (
    cart_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL DEFAULT 1,
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
```

```
);
```

B. Adding Items to the Cart

When a user adds an item to their cart, the PHP script must check if the item is already in the cart. If it is, the quantity is updated; otherwise, a new record is created.

Example PHP Script for Adding Items:

```
php
CopyEdit
session_start(); // Start the session to store user-specific data
include('db_connection.php'); // Include database connection

if (isset($_POST['add_to_cart'])) {
    $user_id = $_SESSION['user_id']; // Get the user ID from session
    $product_id = $_POST['product_id']; // Get the product ID from the form
    $quantity = $_POST['quantity']; // Get the quantity from the form

    // Check if the product already exists in the cart
    $query = "SELECT * FROM cart WHERE user_id = ? AND product_id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("ii", $user_id, $product_id);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // Update quantity if product is already in the cart
        $query = "UPDATE cart SET quantity = quantity + ? WHERE user_id = ? AND product_id = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("iii", $quantity, $user_id, $product_id);
        $stmt->execute();
    } else {
        // Add a new product to the cart
        $query = "INSERT INTO cart (user_id, product_id, quantity) VALUES (?, ?, ?)";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("iii", $user_id, $product_id, $quantity);
        $stmt->execute();
    }

    echo "Item added to cart.";
```

```
}
```

This script checks whether the item is already in the cart and either updates the quantity or adds a new item.

C. Viewing Cart Contents

To view the contents of the cart, the user's user_id is used to retrieve all products associated with that user from the **cart** table. The details of each product, including the product name, price, and quantity, are fetched from the **products** table.

Example PHP Script to View Cart:

```
php
CopyEdit
session_start();
include('db_connection.php');

$user_id = $_SESSION['user_id']; // Get user ID from session

// Fetch cart items for the user
$query = "SELECT c.cart_id, p.name, p.price, c.quantity
        FROM cart c
        JOIN products p ON c.product_id = p.product_id
        WHERE c.user_id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();

while ($row = $result->fetch_assoc()) {
    echo "Product: " . $row['name'] . " | Price: $" . $row['price'] . " | Quantity: " . $row['quantity'] . "<br>";
}
```

This script retrieves all the items in the user's cart and displays their details.

D. Removing Items from the Cart

To remove an item from the cart, the user can delete it by specifying the cart_id. This can be done using a simple PHP script that deletes the corresponding entry from the **cart** table.

Example PHP Script for Removing Items:

```
php
CopyEdit
session_start();
```

```
include('db_connection.php');

if (isset($_GET['remove_item'])) {
    $cart_id = $_GET['cart_id']; // Get cart item ID from URL parameter

    // Remove the item from the cart
    $query = "DELETE FROM cart WHERE cart_id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $cart_id);
    $stmt->execute();

    echo "Item removed from cart.";
}
```

This script deletes the cart item corresponding to the cart_id passed via the URL.

E. Session and User Integration

By using PHP sessions, each user's cart can be uniquely identified and stored. The user_id is retrieved from the session and is used to associate the cart with the specific user. This ensures that users can manage their cart data persistently, even if they log out and log back in.

Conclusion

In a Gym Management website, a shopping cart system enables users to purchase products like gym equipment, memberships, and apparel. By using **PHP and MySQL**:

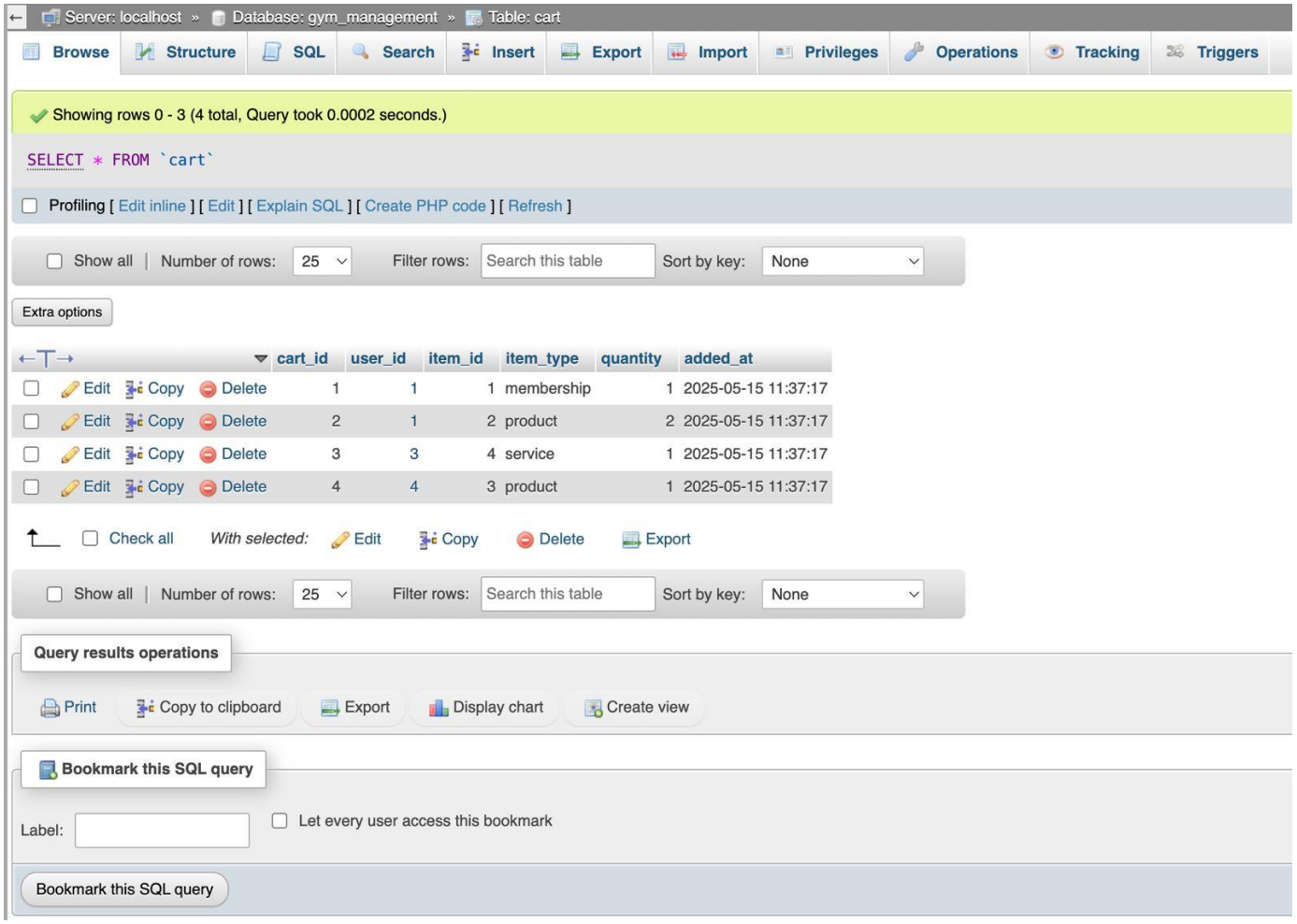
- The **products table** stores the available items.
- The **cart table** manages the cart data for individual users.
- PHP scripts enable users to **add, view, and remove items** from their cart.
- The use of **sessions** ensures that each user's cart is unique and persistent across sessions.

Code

```
<?php

// php/gym_cart.php

session_start();

include 'db_connect.php';


if (!isset($_SESSION['user_id'])) {

   echo json_encode(["error" => "Please log in to manage your cart."]);
```

```
    exit();
}

$user_id = $_SESSION['user_id'];
$action = $_GET['action'] ?? '';

switch ($action) {
    case 'add':
        $item_id = intval($_POST['item_id']);
        $quantity = max(1, intval($_POST['quantity']));

        $stmt = $conn->prepare("INSERT INTO gym_cart (user_id, item_id, quantity)
                                VALUES (?, ?, ?)
                                ON DUPLICATE KEY UPDATE quantity = quantity + VALUES(quantity)");
        $stmt->bind_param("iii", $user_id, $item_id, $quantity);
        $stmt->execute();
        echo json_encode(["success" => true, "message" => "Item added to your gym cart."]);
        break;

    case 'view':
        $stmt = $conn->prepare("SELECT p.name, p.price, c.quantity
                                FROM gym_cart c
                                JOIN gym_products p ON c.item_id = p.id
                                WHERE c.user_id = ?");
        $stmt->bind_param("i", $user_id);
        $stmt->execute();
        $result = $stmt->get_result();

        $cart = [];
```

```
        while ($row = $result->fetch_assoc()) {
            $cart[] = $row;
        }

        echo json_encode($cart);
        break;

    case 'remove':
        $item_id = intval($_POST['item_id']);
        $stmt = $conn->prepare("DELETE FROM gym_cart WHERE user_id = ? AND item_id = ?");
        $stmt->bind_param("ii", $user_id, $item_id);
        $stmt->execute();
        echo json_encode(["success" => true, "message" => "Item removed from your cart."]);
        break;

    default:
        echo json_encode(["error" => "Invalid action."]);
        break;
}

$conn->close();
?>
```


Server: localhost » Database: gym_management » Table: cart
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)
SELECT * FROM `cart`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
cart_id user_id item_id item_type quantity added_at
Edit Copy Delete 1 1 1 membership 1 2025-05-15 11:37:17
Edit Copy Delete 2 1 2 product 2 2025-05-15 11:37:17
Edit Copy Delete 3 3 4 service 1 2025-05-15 11:37:17
Edit Copy Delete 4 4 3 product 1 2025-05-15 11:37:17
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query

Experiment No.10

PHP and MySQL: Handling the Checkout Process for an E-Commerce Website

The checkout process is a critical step in any e-commerce website, including a Gym Management Website, where users are ready to complete their purchase of products like gym equipment, apparel, memberships, or supplements. A smooth and secure checkout process enhances user experience, builds trust, and ensures transactions are successfully completed. This process involves validating user input, processing the cart data, calculating totals (including taxes, discounts, and shipping), storing order details in the database, and providing feedback to the user.

The following theory will explore the steps involved in creating a PHP and MySQL script to handle the checkout process, including:

- A. Validating User Input
- B. Processing Cart Data
- C. Storing Order Information in MySQL
- D. Providing Feedback to the User
- E. Finalizing the Order

A. Validating User Input

Before proceeding with the checkout, it is essential to validate the user's input. Common fields that need validation include:

1. Shipping Address: Ensure that the user has provided a valid and complete shipping address.
2. Payment Information: For security purposes, validate the format of the payment method (credit card number, expiration date, CVV, etc.).
3. Cart Validation: Verify that the user's cart contains items and that they have enough stock available for each product.

Example Validation for Address

```
if (empty($shipping_address)) {
   $errors[] = "Please provide a shipping address.";
}
```

Example Validation for Payment Information

```
if (!preg_match("/^[0-9]{16}$/", $card_number)) {
   $errors[] = "Invalid credit card number format.";
}
```

B. Processing Cart Data

Once the user has entered the necessary information, the next step is to process the cart data. The cart data stored in the database should be retrieved and used to calculate the final price.

1. Retrieving Cart Items

The script must fetch the user's cart contents from the cart table using the user's user_id to ensure that the correct items are checked out.

```
session_start();

include('db_connection.php');


$user_id = $_SESSION['user_id']; // Get user ID from session


$query = "SELECT c.cart_id, p.name, p.price, c.quantity
        FROM cart c
        JOIN products p ON c.product_id = p.product_id
        WHERE c.user_id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();
```

2. Calculating the Total Price

While retrieving the cart data, calculate the total price by multiplying the product price by the quantity. Additionally, apply any discounts, taxes, or shipping fees if applicable.

```
$total_price = 0;
while ($row = $result->fetch_assoc()) {
   $total_price += $row['price'] * $row['quantity'];
}
```

C. Storing Order Information in MySQL

Once the cart data has been validated and processed, it's time to store the order details in the orders table and clear the cart.

1. Creating the Orders Table

The orders table will store the user's order information, such as the products purchased, the total price, shipping address, and payment status.

Example SQL to create the orders table:

```
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    total_price DECIMAL(10,2) NOT NULL,
    shipping_address TEXT NOT NULL,
    payment_status ENUM('Pending', 'Completed', 'Failed') DEFAULT 'Pending',
    date_ordered TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

2. Inserting Order Data

After calculating the total price and verifying payment, store the order data in the orders table.

```
$query = "INSERT INTO orders (user_id, total_price, shipping_address, payment_status)
        VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($query);
$stmt->bind_param("idss", $user_id, $total_price, $shipping_address, $payment_status);
$stmt->execute();
$order_id = $stmt->insert_id; // Get the ID of the newly inserted order
```

3. Storing Order Items

For each item in the cart, insert the details (product, quantity, price) into the order_items table.

```
CREATE TABLE order_items (
    item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
```

```
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

Inserting items into the order_items table:

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


foreach ($cart_items as $item) {
    $stmt->bind_param("iiid", $order_id, $item['product_id'], $item['quantity'], $item['price']);
    $stmt->execute();
}
```

D. Providing Feedback to the User

Once the order is successfully placed, the user should receive feedback. This could be a confirmation message or a redirect to an order summary page.

Example Success Feedback:

```
echo "Thank you for your purchase! Your order number is: " . $order_id;
```

If the payment fails, provide an error message:

```
echo "There was an error processing your payment. Please try again.";
```

E. Finalizing the Order

After processing the order:

1. Clear the Cart: Once the user's cart is successfully converted into an order, clear the items from the cart.

```
$query = "DELETE FROM cart WHERE user_id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $user_id);
$stmt->execute();
```

Conclusion

A well-implemented checkout process on a Gym Management website ensures that users can securely complete their purchases for gym equipment, memberships, or supplements. The PHP and MySQL integration allows the following:

1. User Input Validation: Ensures that user details such as shipping address and payment information are correct.

2. Cart Data Processing: Retrieves cart data, calculates the total price, and ensures product availability.
3. Order Information Storage: The order and its items are stored in the database, keeping track of the transaction.
4. Feedback: Users receive confirmation or error messages based on the transaction outcome.
5. Finalization: The cart is cleared, and product stock is updated to reflect the purchase.

By integrating PHP with MySQL, the shopping experience on a Gym Management website is streamlined, secure, and persistent, ensuring a smooth transition from cart to purchase.

PHP and MySQL

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.

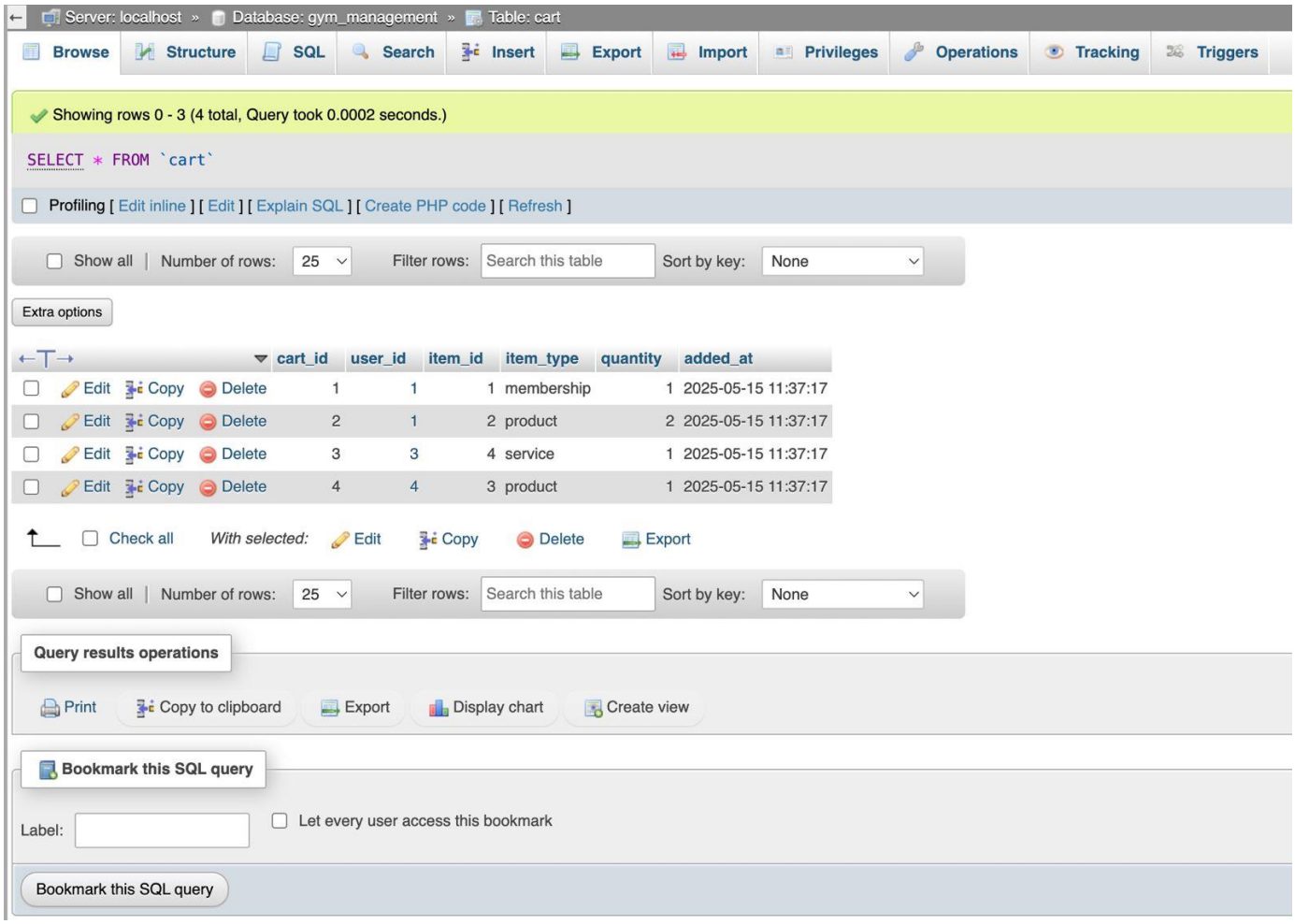
B. Develop a PHP script that processes the checkout process for users who are ready to complete their purchase, integrating the MySQL database for handling user and order information. The script should validate user input, process the cart data, and provide feedback upon successful or failed checkout.

Code

```
<?php

// php/gym_checkout.php

include 'db_connect.php';

session_start();


if (!isset($_SESSION['user_id'])) {

    echo json_encode(["error" => "You must be logged in to checkout."]);

    exit();

}


$user_id = $_SESSION['user_id'];

$address = trim($_POST['address'] ?? '');

$payment_method = trim($_POST['payment_method'] ?? '');
```

```
if (empty($address) || empty($payment_method)) {
    echo json_encode(["error" => "Address and payment method are required."]);
    exit();
}

$conn->begin_transaction();

try {
    // Insert order
    $stmt = $conn->prepare("INSERT INTO gym_orders (user_id, address, payment_method, created_at) VALUES (?, ?, ?, NOW())");
    $stmt->bind_param("iss", $user_id, $address, $payment_method);
    $stmt->execute();
    $order_id = $stmt->insert_id;

    // Get cart items
    $stmt = $conn->prepare("SELECT item_id, quantity FROM gym_cart WHERE user_id = ?");
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $result = $stmt->get_result();

    while ($item = $result->fetch_assoc()) {
        $stmt_item = $conn->prepare("INSERT INTO gym_order_items (order_id, item_id, quantity) VALUES (?, ?, ?)");
        $stmt_item->bind_param("iii", $order_id, $item['item_id'], $item['quantity']);
        $stmt_item->execute();
    }
```

```
    // Clear the cart
    $stmt = $conn->prepare("DELETE FROM gym_cart WHERE user_id = ?");
    $stmt->bind_param("i", $user_id);
    $stmt->execute();

    $conn->commit();
    echo json_encode(["success" => "Checkout successful. Your order has been placed."]);
} catch (Exception $e) {
    $conn->rollback();
    echo json_encode(["error" => "Checkout failed. Please try again."]);
}
?>
```



Conclusion

The Gym Management System project successfully integrates multiple web technologies including **HTML, CSS, JavaScript, PHP, and MySQL** to provide a full-stack, dynamic, and interactive user experience. Throughout the development of this system, we implemented key features that reflect a real-world e-commerce structure, adapted to fitness and gym-related services.

We began with secure user **registration and login modules**, ensuring data validation and session-based authentication. Using PHP and MySQL, we created a **shopping cart system** where users can add, view, and remove gym products such as memberships, supplements, or training programs. This cart system is designed to persist data across sessions, ensuring a smooth and personalized experience.

Furthermore, a robust **checkout module** was developed, enabling users to finalize their purchases with input validation and transactional accuracy. All operations are handled through well-structured SQL queries and PHP scripts, ensuring security, integrity, and reliability of user and order data.

Additionally, the use of **sessionStorage/localStorage** in JavaScript ensures persistent login and cart states on the front-end, improving UX.

Overall, the project demonstrates the practical application of core programming concepts in a modular, scalable, and maintainable way—laying a strong foundation for future enhancements such as payment gateway integration, admin dashboards, or mobile responsiveness.