



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

**Department of Information Technology**

## **Lab Manual**

**Practical - Web Programming**

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

**Batch - DA-I/II**

**Saniya Mallick**

**Ms.**

**A.Y. 2024 – 2025 (SEM-I)**

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

processing.

4. Design and develop a back-end application using PHP and MySQL, implementing CRUD operations to manage data effectively.

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

	<b>javascript</b> Use tools like Visual Studio Code (free).	
	<b>Self-study / Do it yourself /:</b> Practice exercises on DOM traversal and event handling.	
	<b>Experiential Learning Topics:</b> Add dynamic behavior to a webpage using DOM and events (e.g., a to-do list app).	
	<b>PBL - Project Based Learning:</b> Develop a web page with dynamic content (e.g., a task manager or interactive quiz) using DOM manipulation and event handling.	
<b>UNIT – IV</b>	<b>Server Side Scripting</b>	<b>09 Hours</b>
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		
<b>Pedagogy</b>	<b>ICT Teaching / PowerPoint Presentation and Videos:</b> <a href="https://www.coursera.org/learn/web-applications-php">https://www.coursera.org/learn/web-applications-php</a> Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management	
	<b>Self-study / Do it yourself /:</b> Practice exercises on form handling and server-side scripting with PHP.	
	<b>Experiential Learning Topics:</b> Create a basic form for data submission and handle it using PHP (e.g., feedback form).	
	<b>PBL - Project Based Learning:</b> Develop a small server-side application (e.g., a contact form with email validation and submission).	
<b>UNIT – V</b>	<b>Working with Databases and Web Application Development</b>	<b>09 Hours</b>
Module 10 - Working with databases using MySQL with PHP: MySQL database, create database, create table, primary key with AUTO_INCREMENT setting, Insert Data Into a Database Table, Select Data From a Database Table, Open or close a Connection to the MySQL Server. Module 11 - Web Application Development (Project): Develop the web application to handle client-server interactions, manage user data, and implement various web technologies to enhance the functionality of their applications. Example: Website for a Coffee Shop		
<b>Pedagogy</b>	<b>ICT Teaching / PowerPoint Presentation and Videos:</b> Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management <b>Videos:</b> <a href="https://www.coursera.org/learn/web-app">https://www.coursera.org/learn/web-app</a>	
	<b>Self-study / Do it yourself /:</b>	

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

**Text Books:**

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

**Reference Books:**

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

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

1. W3Schools HTML, CSS, JavaScript Tutorial: <https://www.w3schools.com/html/>
2. Mozilla Developer Network (MDN) Web Docs - HTML, CSS, JavaScript, DOM: [https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML](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

## Experiment No. 1

### Problem Statement

1. Create the basic structure of the **Artwave** website, including the home page layout with a header, main content area, and footer.
2. Prepare a complete project design and planning document that will guide the development of the website across all assignments.

### Objective

To design the basic structure of the Artwave website by planning its layout, content, and visual elements to ensure it meets user needs and reflects the platform's creative identity.

### Theory

## Project Design and Planning Document – Artwave

### 1. Brief Information about the Project

Artwave is an interactive web platform where users can explore and purchase music tracks and digital artworks based on different moods such as Motivation, Relaxation, Study, and Workout. The project aims to blend creativity and user experience by providing mood-based content and a seamless browsing interface. It also includes features like a user login system, shopping cart, and checkout functionality.

### 2. Set the Goals and Deliverables

#### Goals

- Build a creative, user-friendly platform to explore mood-based music and art.
- Allow users to listen to tracks, view artworks, and add items to their cart.
- Implement secure user login, registration, and checkout features.
- Ensure mobile-responsive design.

#### Deliverables

- Pages:
  - Home Page
  - About Page
  - Mood-Based Menu Pages (Motivation, Relax, Study, Workout)
  - Contact Page
  - Login & Registration Pages
  - Cart and Checkout Pages

- Functional Features:
  - Header and Footer with navigation
  - Audio and Image Display
  - Cart with quantity and subtotal
  - User authentication (Login/Register)
  - Checkout functionality

### 3. Finalize the Modules of the Project

Module	Description
Home Page	Introduces the Artwave platform with a hero section, tagline, and CTA.
About Page	Shares the story, mission, and vision of Artwave.
Mood Menu Pages	Includes categorized content (Motivation, Relax, Study, Workout) with tracks and art.
Contact Page	Allows users to send messages or feedback.
Login/Registration	Handles user authentication.
Cart & Checkout	Lets users manage their selected items and proceed with purchases.

## Website Modules

### 1. Home Page Module

#### Description:

The landing page introduces users to Artwave and sets the emotional and visual tone of the platform.

#### Features:

- Tagline and hero section showcasing the site's purpose
- Prominent “Get Started” or “Explore Moods” call-to-action
- Brief overview of what the site offers
- Navigation to other major sections
- Footer with contact links and branding

### 2. About Page Module

#### Description:

Explains the purpose and background of Artwave to establish trust and credibility.

#### Features:

- Brief story behind the Artwave project
- Mission, vision, and values (e.g., creative freedom, emotional well-being through art)
- Introduction to the creators or team behind the project

### 3. Product/Service Page (Mood Menu Pages) Module

#### Description:

Displays categorized mood pages like Motivation, Study, Relax, and Workout that contain curated tracks and artwork.

#### Features:

- Separate sections for music and artwork
- Media previews (audio and images)
- Add-to-cart option
- Quantity and pricing display
- Cart and checkout integration

#### **4. Testimonials/Review Page Module**

**Description:**

Shows user feedback and builds social proof.

**Features:**

- Short user reviews
- Star ratings or brief testimonials
- Clean, accessible layout for readability

#### **5. Contact Page Module**

**Description:**

Allows users to get in touch with Artwave for inquiries or feedback.

**Features:**

- Contact form (Name, Email, Message)
- Confirmation on successful submission
- Optional map or visual branding element

#### **6. Login & Registration Module**

**Description:**

Handles user authentication for personalized access.

**Features:**

- Secure login and registration forms
- Validation and redirection logic
- Session management

#### **7. Optional Blog Module**

**Description:**

Placeholder section for future blog posts or mood-based music/art inspiration articles.

**Features:**

- Card-style layout for blogs
- Placeholder content
- Visual balance with other site sections

### **Design Elements**

These elements help define Artwave's brand identity and make it emotionally appealing:



**1. Colors:**

- Primary: Deep Purple (#422F48)
- Accent: Soft Lavender, White, and Black contrasts

**2. Fonts and Typography:**

- Poppins for headings and body text for a modern, readable look

**3. Logo:**

- A custom logo with a symbolic connection to creativity and emotion

**4. Images and Photos:**

- Mood-based artwork visuals
- Clean, minimalist photos where needed

**4. Define the Audience**

The primary audience for Artwave includes:

- Students and working professionals seeking music or visual art tailored to their emotional or mental state.
- Individuals looking for creative inspiration or calming content based on mood.
- General users who enjoy curated media experiences with a clean interface and minimal navigation effort.

**5. Describe Pain Points & the Ideal Experience****Crafting the Ideal Experience**

To improve upon the above issues, the ideal user experience for **Artwave** should:

**a. Support Intuitive Navigation**

- Use a sticky navigation bar with clear links to pages like Home, Menu, About, Contact, Login, and Cart.
- Organize mood-based menus logically (Motivation, Relax, Study, Workout).

**b. Enable Seamless Browsing and Purchase**

- Users can preview tracks or artworks.
- Each item shows a name, type (music/art), quantity, and price.
- Cart supports both item types and includes a subtotal.

**c. Be Fully Mobile-Responsive**

- Design works smoothly across desktops, tablets, and mobile devices.
- Interactive features like audio playback and add-to-cart are touch-friendly.

**d. Provide Emotional & Creative Filters**

- Filter content by mood (Relax, Motivation, etc.) for a personal, emotional journey.
- Display curated playlists or visual themes under each mood.

**e. Add Personalization**

- Registered users can view their name after login.
- Cart saves between sessions.
- Future enhancements could include mood-based suggestions.

**f. Build Trust Through User Feedback**

- Include a dedicated Testimonials page with real user quotes.
- Use blog-style posts in the future for topics like “How Music Helps You Focus.”

#### **g. Display Clear Information and Content Previews**

- Every track and artwork shows a preview (audio or image).
- Prices, item names, and quantities are visible on both the mood pages and in the cart.

## **6. Set the Visual Direction**

### **1. Visual Design Goals**

The visual direction for the Artwave website is built to reflect creativity, emotion, and modern digital aesthetics. Its aim is to ensure users feel both inspired and understood the moment they land on the site.

Key principles:

- **Mood-Driven and Immersive:** Artwave should feel emotionally resonant and immersive to reflect its purpose of mood-based music and art.
- **Clean and Modern:** A visually clean layout that doesn't distract from the media content (music tracks and artworks).
- **Emotionally Appealing:** Use color and imagery to trigger moods — calm, motivation, focus, or relaxation — matching the intent of the platform.

### **Defining the Core Visual Elements**

**a. Color Palette** A warm and earthy color palette inspired by natural tones creates a visually consistent and soothing experience.

#### **Deep Purple**

- **Hex Code:** #4B0082
- **Usage:** Header, footer, buttons, and highlights.

#### **Soft Lavender**

- **Hex Code:** #E6E6FA
- **Usage:** Background to create calm and contrast.

#### **Charcoal Gray**

- **Hex Code:** #333333
- **Usage:** Text and important accents for legibility.

#### **Off-White**

- **Hex Code:** #F8F8FF
- **Usage:** Secondary backgrounds and subtle contrasts.

#### **Vibrant Teal**

- **Hex Code:** #008080
- **Usage:** Call-to-action buttons for energetic harmony.

## b. Typography

Fonts should be modern and easy to read while reflecting the artistic and vibrant atmosphere of the platform.

- **Primary Font:** Poppins or Montserrat (Sans-serif) – For headings and call-to-action text.
- **Secondary Font:** Open Sans or Lora – For body text and descriptions.
- **Attributes:** Use bold headings for emphasis and lighter weights for readability.

## c. Logos and Branding

A sleek, memorable logo based on the Artwave theme is essential. For instance:

- Use a stylized wave, music note, or paintbrush motif in the logo design.
- The logo should include the platform’s name “Artwave” in the selected typography.
- A monochrome version of the logo can be created for simplicity in headers or footers.

## d. Imagery and Icons

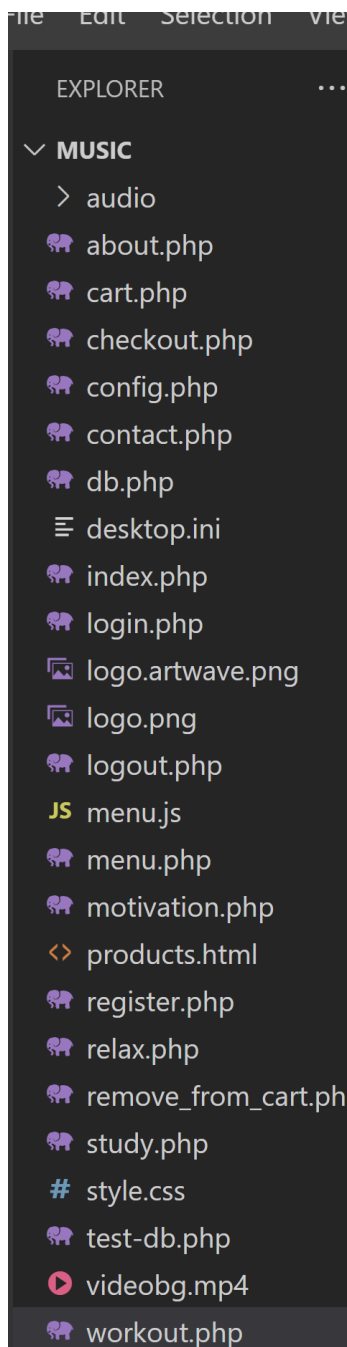
High-quality visuals make the website feel alive and engaging.

- **Photography:**
  - Pictures of musical instruments, art canvases, and creative studio spaces.
  - Showcase featured tracks, artwork, and artist portraits.
- **Icons:**
  - Minimalistic icons for categories like moods, playlists, artwork gallery, testimonials, and contact.
- **Hero Images:**
  - Use a carousel or static hero banner on the homepage featuring featured moods or trending artworks and tracks.

## 7. Map Out the Project Structure

Artwave Project Structure:

- index.php → Homepage
- about.php → About the platform
- motivation.php → Mood page: Motivation
- relax.php → Mood page: Relax
- study.php → Mood page: Study
- workout.php → Mood page: Workout
- contact.php → Contact form
- testimonial.php → Testimonials and feedback
- register.php → Registration form
- login.php → Login form
- cart.php → Cart system
- checkout.php → Checkout functionality
- blog.php (optional) → Future blog posts
- style.css → Shared stylesheet
- script.js → JavaScript interactions
- /uploads → Music/artwork storage



## 8. Plan the Content for Each Page

### 1. Home Page

#### Purpose:

- Welcome visitors.
- Highlight Artwave's unique blend of music and artwork.

#### Content Plan:

- **Header:**
  - Logo on the left.
  - Navigation menu: Home, About, Moods, Testimonials, Contact.
  - Login/Sign-Up button on the top right.
- **Hero Section:**

- A high-quality banner image featuring an artistic mood or trending track/artwork.
- Tagline: “Feel the Vibe, See the Art.”
- CTA button: “Explore Moods.”
- **Introduction Section:**
  - Brief about Artwave (one or two sentences).
  - CTA: “Learn More About Us” linking to About page.
- **Featured Moods Section:**
  - Carousel or grid of popular moods like Motivation, Study, Relax, Workout.
  - Text: “Discover Your Perfect Sound and Vision.”
- **Footer:**
  - Quick links, social media icons, contact info.

## 2. About Page

### Purpose:

- Share the story, vision, and creators behind Artwave.

### Content Plan:

- **Header:** (same as Home)
- **About Us Section:**
  - Short history and mission of Artwave.
  - Emphasis on creativity, emotion, and community.
- **Meet the Creators Section:**
  - Photos and bios of musicians, artists, and developers.
- **Special Features Section:**
  - Why choose Artwave? (Unique mood-based curation, dual art forms, personalized experience).
- **Footer:** (same as Home)

## 3. Mood Menu Pages (Motivation, Study, Relax, Workout)

### Purpose:

- Showcase curated music and artwork based on mood.

### Content Plan:

- **Header:** (same as Home)
- **Mood Section:**
  - Grid of music tracks and artworks with images, titles, and prices.
  - Hover effects for interactivity.
- **CTA Section:**
  - Button: “Add to Cart” for each item.
- **Footer:** (same as Home)

## 4. Testimonials Page

### Purpose:

- Build trust through user reviews.

### Content Plan:

- **Header:** (same as Home)

- **Customer Feedback Section:**
  - User quotes, star ratings, and avatars.
- **Submit a Testimonial:**
  - Simple form for new reviews.
- **Footer:** (same as Home)

## 5. Contact Page

### Purpose:

- Allow users to reach out for support or inquiries.

### Content Plan:

- **Header:** (same as Home)
- **Contact Form:**
  - Name, Email, Subject, Message fields with validation.
- **Location Section:**
  - Embedded Google Map with office location.
- **Operating Hours Section:**
  - Business hours displayed.
- **Footer:** (same as Home)

## 6. Login Page

### Purpose:

- Allow existing users to log in.

### Content Plan:

- **Form:**
  - Email and Password inputs.
  - Submit button.
- **Forgot Password:**
  - Link to recovery page.
- **CTA:**
  - Link to Registration page: “Don’t have an account? Sign Up Now!”

## 7. Registration Page

### Purpose:

- Allow new users to register.

### Content Plan:

- **Form Fields:**
  - Full Name, Email, Password, Confirm Password.
- **Validation:**
  - Password strength requirements.
- **Submit Button:**
  - Validates and submits data.
- **Footer:** (same as Home)

## 9. Add Ideas for Content, Images & Layout

- Use relatable mood-specific taglines and headings for each mood page.
- Include high-quality abstract or mood-themed artwork visuals.
- Match the design with audio tone—e.g., calm visuals for relaxing tracks.
- Maintain consistent image shapes, spacing, and button styles.
- Hero sections to contain minimal but impactful text.

## 10. Determine Your Site Structure / Core Website Pages

### 1. (Home Page)

#### Purpose:

To welcome visitors to Artwave and introduce them to the concept of mood-based content exploration.

#### Key Features:

- Hero section with a tagline and introductory message.
- Clear call-to-action button like “Explore Your Mood.”
- Navigation bar linking to mood pages and cart.
- Footer with branding and contact links.

### 2. (About Page)

#### Purpose:

To share the vision, mission, and background story behind the Artwave platform.

#### Key Features:

- Brief overview of how Artwave was started.
- Mission to promote emotional wellness through art and music.
- Creative team information.
- Branding and values description.

### 3. (Mood Menu Pages)

#### Purpose:

To showcase mood-specific music and artworks for user interaction and purchase.

#### Key Features (applies to each mood page):

- Separated content blocks for tracks and visual artworks.
- Media previews (audio for tracks, images for artwork).
- Add-to-cart buttons for each item.
- Display of item name, price, and quantity input.
- Consistent mood-based color theme and layout.

### 4. cart.php (Cart Page)

#### Purpose:

To allow users to review and manage items they have added to the cart.

#### Key Features:

- List of selected tracks and artworks with quantity and individual price.

- Subtotal calculation displayed dynamically.
- Option to adjust item quantities or remove items.
- Checkout button for further purchase flow.

## 5. (Login and Registration Pages)

### Purpose:

To authenticate users and allow them to create or access personal accounts.

### Key Features:

- Secure forms with input validation.
- Email and password login.
- Registration form with full name, email, and password fields.
- Redirection to mood pages or cart after login.
- Session handling for logged-in users.

## 6. (Contact Page)

### Purpose:

To provide users with a direct way to get in touch for questions, support, or feedback.

### Key Features:

- Contact form with Name, Email, and Message fields.
- Submission confirmation message.
- Clean and minimal layout to match the overall site design.

## 11. Create and Collect Design Elements

These design elements define your brand personality and help users emotionally connect with Artwave — a platform that merges music and artwork to reflect human moods.

### 1. Colors

Colors are core to mood representation on Artwave. The chosen palette is soothing yet expressive, designed to evoke creativity, calm, and emotional depth.

#### Primary Colors:

- **Deep Purple (#422F48):** Symbolizes creativity and emotional richness. Used for headers, buttons, and active states.
- **Soft Lavender (#D8C4E9):** A calming and elegant tone for backgrounds and section dividers.
- **White (#FFFFFF):** For clean, breathable layouts that support contrast and clarity.

#### Accent Colors:

- **Jet Black (#1E1E1E):** Used for high-contrast text, icons, and footer elements.
- **Pastel Pink or Sky Blue (optional):** Can be used subtly in mood tags or highlights depending on the mood page.

#### Psychological

These colors reflect serenity, inspiration, and emotional comfort — aligning with Artwave's mission to enhance mood through art and sound.

#### Impact:



## 2. Fonts and Typography

Typography on Artwave balances modern readability with expressive flair, suitable for a digital arts and music platform.

- **Heading**
- **Font:**  
**Poppins (Sans-serif)** – Clean and modern with rounded elegance; used for all major headings and titles.
- **Body**
- **Font:**  
**Open Sans or Roboto** – Friendly and readable; used for descriptions, user reviews, and button text.
- **Styling Tips:**
  - Use bold for track/artwork names and CTA buttons.
  - Use lighter weights for secondary information and product descriptions.

### Impact:

This typography combination ensures accessibility, emotional tone, and aesthetic consistency across all device types.

## 3. Logo

Your logo is the visual essence of Artwave. It should symbolize both **creativity and emotional expression**.

- **Design Idea:**
  - Abstract waveforms, music notes, or a stylized brush stroke.
  - Optional elements: a heart, headphone outline, or palette stroke merged with a soundwave.
- **Color Usage:**
  - Base: Deep Purple and White.
  - Accent: Lavender or Jet Black depending on placement.
- **Placement & Scalability:**
  - Top-left in the header of every page.
  - Scalable for use on social media, mood page banners, favicons, and digital promotional material.

### Impact:

A strong logo provides brand identity and emotional recognition, especially important in a creative platform.

## 4. Imagery and Photos

Visual content is a central component of Artwave's appeal, engaging users through mood-aligned art and music previews.

### Types of Visuals:

- **Product (Track & Artwork) Images:**

- Album cover art or symbolic images representing tracks.
- Digital artwork thumbnails in grid or card layouts.
- Hover effects to preview audio or view artwork in detail.
- **Mood-Based Visual Themes:**
  - **Motivation:** Bright, dynamic visuals with energetic tones.
  - **Relax:** Calm, pastel-toned backgrounds and serene landscapes.
  - **Study:** Minimalist, focused visual themes with cool tones.
  - **Workout:** Bold, rhythmic elements with motion-based graphics.
- **Lifestyle Images (Optional):**
  - Abstract representations of people enjoying art/music.
  - Emotive stock photos or vector illustrations matching each mood.

### Impact:

These images create a visual narrative that connects the user's emotional state to Artwave's content, increasing time on site and user satisfaction.

## 5. Interactive Elements

Interactivity enhances the user experience and makes the site feel alive.

- **Add-to-Cart**
- **Buttons:**
  - Subtly animated when clicked, reinforcing interactivity.
- **Hover Effects:**
  - Glow or zoom-in on artwork or track cards.
  - Button color shifts on hover for active feedback.
- **Audio Preview (for Tracks):**
  - Play/pause button overlays.
  - Visual waveform or play timer for added engagement.
- **Quantity Selectors:**
  - Custom-styled increment/decrement buttons beside each item in cart pages.

### Impact:

Interactive design brings the platform to life, creating a smooth, immersive experience while reinforcing user trust and delight.

## 12. Design Elements Defining Brand Personality

- **Colors:**
  - Deep Purple (#422F48): Emotional depth and artistic richness
  - Soft Lavender: Calm and soothing
  - White and black: Clean contrasts and clarity
- **Fonts and Typography:**
  - Poppins: Modern, rounded, and emotionally expressive
- **Logo:**
  - A creative symbol combining sound waves and brush strokes
- **Images and Photos:**
  - Curated artwork visuals
  - Abstract or ambient imagery supporting mood themes

## Conclusion:

The **Artwave Website Project** serves as a comprehensive exercise in applying core web development concepts, visual design principles, and user experience strategies in a real-world creative context. By thoughtfully integrating music and artwork through mood-based content, this project demonstrates how a digital platform can be both functional and emotionally engaging.

Through the creation of essential pages such as **Home, About, Mood Menu Pages (Motivation, Relax, Study, Workout), Testimonials, Contact, Login/Registration, and Cart/Checkout**, the site offers users an intuitive, engaging, and purposeful browsing experience. Each page is designed to connect emotionally with users by matching visual elements and content to their current mood or need.

The visual direction — including a harmonious **color palette**, modern **typography**, expressive **logo**, and mood-specific **images** reinforces the brand identity of Artwave as a soothing and creative digital space. These elements enhance brand recognition and make the platform welcoming for both music and art lovers.

This project highlights the importance of **user-centered design**, content planning, and a well-structured development approach. By addressing the needs of a diverse audience and incorporating personalized, interactive features, **Artwave successfully bridges art, music, and mood**, providing users with a seamless and immersive digital experience.

## **Experiment No.2**

### **Problem Statement:**

#### 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.

### **Objective:**

To design a web-based platform (Artwave) that allows users to explore and purchase mood-based music and artwork using HTML. The platform promotes creativity, emotional expression, and digital engagement.

### **Introduction:**

In the modern creative economy, platforms like **Artwave** serve as essential tools for artists, musicians, and emotionally engaged audiences. Artwave curates mood-based content, offering users a tailored experience through categorized music tracks and visual artworks.

The aim is to create a functional, responsive website that merges aesthetics with usability, encouraging art discovery and supporting digital creators. The project combines front-end development skills with logical structuring to simulate a real-world creative e-commerce environment.

### **1. Home Page**

Acts as the user's first interaction with Artwave and includes:

- A hero section with a banner image and mood-driven tagline.
- Navigation bar linking to Mood Menu, About, Contact, Testimonials, Login/Register.
- Call-to-action buttons like "Explore Music", "Discover Art".
- Links to featured moods (e.g., Motivation, Relax, Study, Workout).

**Importance:**

Creates the first impression and positions Artwave as an inspiring, mood-centric platform.

**Technologies Used:**

HTML for structure, CSS for design, optional JavaScript for interactivity.

**2. Mood Menu/Product Pages**

Displays curated items based on the selected emotional theme. Each mood page includes:

- Two sections: **Music Tracks** and **Artworks**
- Previews (audio for music, thumbnails for artwork)
- Title, price, and **Add to Cart** button with quantity selection

**UX Considerations:**

Filters by category (music/art), responsive grid layout, hover effects on cards.

**Real-World Relevance:**

Encourages emotional exploration while supporting creators.

**3. Cart Page**

The cart allows users to review selected items before proceeding to checkout:

- List of items (music or artwork) with name, price, quantity, and subtotal.
- Option to remove or update quantity.
- Display of total amount with a **"Proceed to Checkout"** button.

**Optional Enhancements:**

Use of session or localStorage for cart persistence. Future expansion to include backend cart handling.

**4. About Page**

Gives insight into Artwave's vision and creative journey:

- Project origin and inspiration
- Mission to connect emotion with digital expression
- Team section with short bios of developers or creators

**Purpose:**

Builds emotional credibility and reinforces Artwave's identity.

**5. Contact Page**

Includes a contact form for user feedback or inquiries:

- Input fields: Name, Email, Message
- Submit button with validation
- Optional: Embedded Google Map or contact details (if physical studio exists)

#### **UX Impact:**

Quick communication fosters trust and improves user satisfaction.

## **6. User/Admin Registration Form**

Allows new users or admin to sign up:

- Fields: Full Name, Email, Password, Confirm Password, User Role (dropdown)
- Validations: Email format, password match, required fields

#### **Functionality:**

Lays the groundwork for personalized sessions and admin management panels.

## **7. User/Admin Login Form**

Secure login portal with form validation:

- Fields: Email/Username, Password
- Features: "Remember Me" checkbox, "Forgot Password?" link

#### **Security Measures (for future):**

- Password hashing, role-based redirects
- Separate access: Admins manage products, users browse and purchase

### **Code:**

A. Home page:

code: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Artwave | Home</title>

<link rel="stylesheet" href="style.css">

</head>

```
<body>

<header>

  <h1>Welcome to Artwave</h1>

  <nav>

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

    <a href="relax.html">Relax</a>

    <a href="motivation.html">Motivation</a>

    <a href="study.html">Study</a>

    <a href="workout.html">Workout</a>

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

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

  </nav>

</header>

<main>

  <section class="hero">

    <h2>Find Music and Art That Matches Your Mood</h2>

    <a href="relax.html">Explore Now</a>

  </section>

  <section class="introduction">

    <p>Artwave is your destination for mood-based music tracks and artwork that connect with your emotions.</p>

  </section>

</main>

<footer>

  <p>&copy; 2025 Artwave</p>

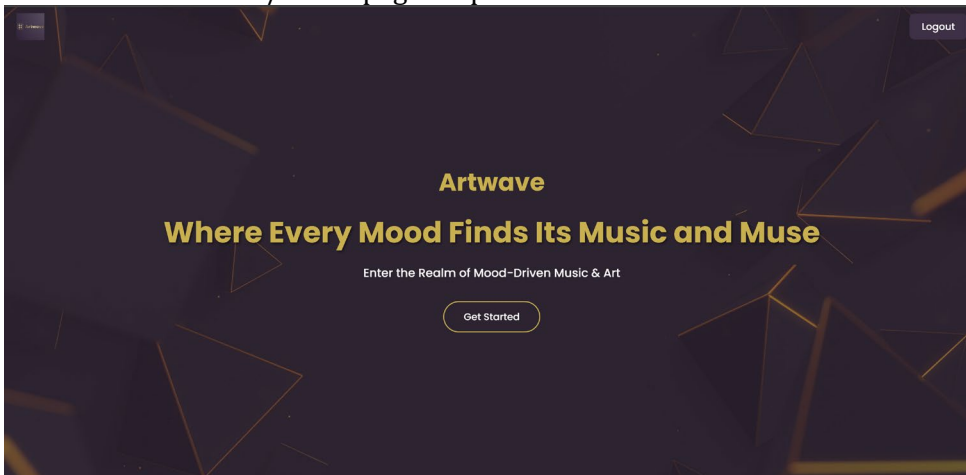
</footer>

</body>

</html>
```

**Output:**

## A. Index/Home page output:

**Code:**

## B. menu/product page:

code: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Menu - Artwave</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1>Explore All Art & Music</h1>

<nav>

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

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

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

</nav>

</header>

<main>

<section class="menu-section">



```
<h2> Music Tracks</h2>

<div class="product">

  <p><strong>Track:</strong> Mountain Echo</p>

  <p><strong>Price:</strong> ₹59</p>

  <button>Add to Cart</button>

</div>

<div class="product">

  <p><strong>Track:</strong> City Nights</p>

  <p><strong>Price:</strong> ₹49</p>

  <button>Add to Cart</button>

</div>

</section>
```

```
<section class="menu-section">

  <h2> Artworks</h2>

  <div class="product">

    <p><strong>Artwork:</strong> Silent Waves</p>

    <p><strong>Price:</strong> ₹249</p>

    <button>Add to Cart</button>

  </div>

  <div class="product">

    <p><strong>Artwork:</strong> Electric Vibe</p>

    <p><strong>Price:</strong> ₹199</p>

    <button>Add to Cart</button>

  </div>

</section>

</main>
```

```
<footer>

  <p>&copy; 2025 Artwave</p>
```

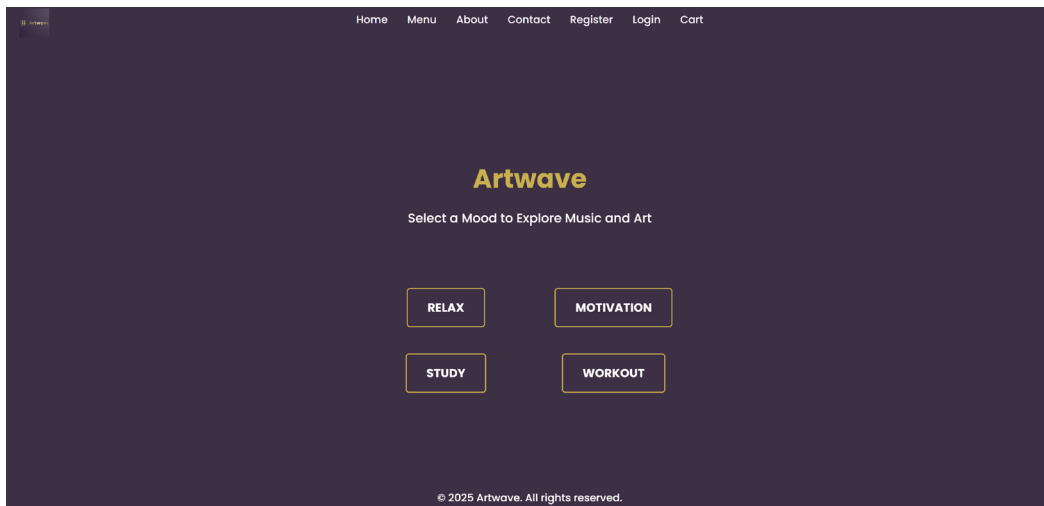
```
</footer>
```

```
</body>
```

```
</html>
```

## **Output:**

B. menu/product page output:



## **Code:**

C. cart page:

code: <!DOCTYPE html>

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

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Artwave Cart</title>
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Your Cart</h1>
```

```
<nav>
```

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

```
</nav>

</header>

<main>

  <section class="cart-items">

    <div class="cart-item">

      <p>Ocean Breeze (x1)</p>

      <p>₹49</p>

    </div>

    <div class="cart-item">

      <p>Calm Dusk (x1)</p>

      <p>₹199</p>

    </div>

  </section>

  <section class="checkout">

    <p>Subtotal: ₹248</p>

    <button>Proceed to Checkout</button>

  </section>

</main>

<footer>

  <p>© 2025 Artwave</p>

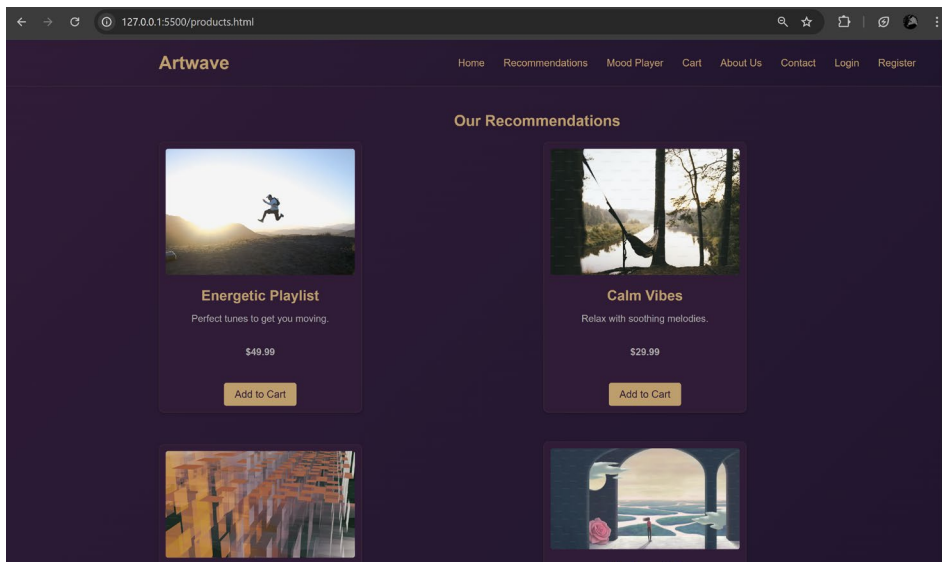
</footer>

</body>

</html>
```

**Output:**

B. cart page output:



## Code:

D. about us page:

code: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>About Artwave</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1>About Us</h1>

<nav>

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

</nav>

</header>

<main>

<section>

<h2>Our Story</h2>

<p>Artwave started as a passion project to blend creativity with emotion. We provide handpicked music tracks and original artwork to elevate your mood.</p>

</section>

<section>

<h2>Mission</h2>

<p>To inspire emotions through curated art and music experiences, customized to your mood.</p>

</section>

<section>

<h2>Our Team</h2>

<p>A creative team of designers, musicians, and developers working together to bring Artwave to life.</p>

</section>

</main>

<footer>

<p>&copy; 2025 Artwave</p>

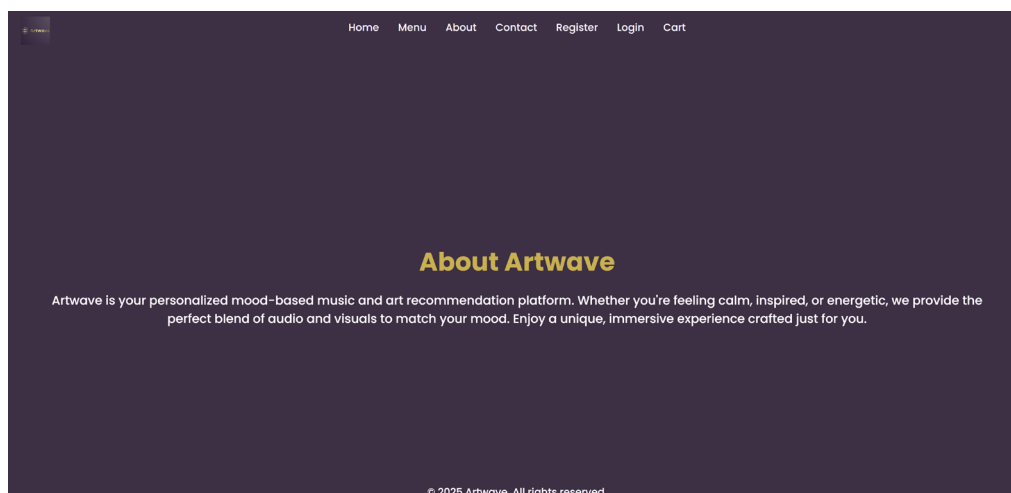
</footer>

</body>

</html>

## **Output:**

D. about us page output:



**Code:**

E. contact us page:

```
code: <!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Contact Artwave</title>
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Contact Us</h1>
```

```
<nav>
```

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

```
</nav>
```

```
</header>
```

```
<main>
```

```
<form action="#" method="post">
```

```
<label>Name: <input type="text" name="name" required></label><br>
```

```
<label>Email: <input type="email" name="email" required></label><br>
```

```
<label>Message:<br>
```

```
<textarea name="message" rows="5" required></textarea>
```

```
</label><br>
```

```
<button type="submit">Send Message</button>
```

```
</form>
```

```
</main>
```

```
<footer>
```

```
<p>&copy; 2025 Artwave</p>
```

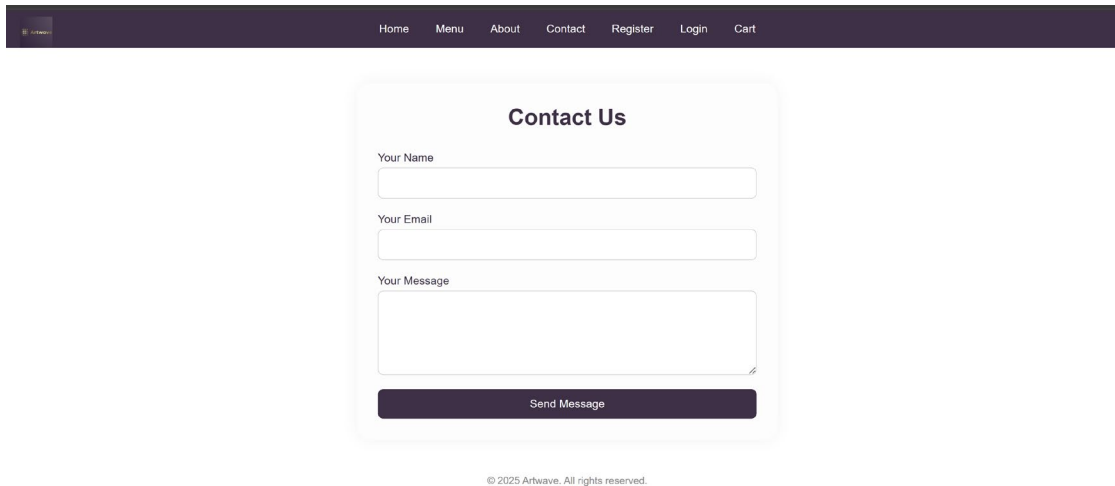
```
</footer>
```

```
</body>
```

```
</html>
```

## **Output:**

E. contact us page output:



## **Code:**

F. registration page:

code: <!DOCTYPE html>

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

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Register - Artwave</title>
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<main>
```

```
<h2>Register</h2>
```

```
<form method="post" action="#">
```

```
<label>Name: <input type="text" name="name" required></label><br>
```

```

<label>Email: <input type="email" name="email" required></label><br>
<label>Password: <input type="password" name="password" required></label><br>
<label>Confirm Password: <input type="password" name="confirm_password" required></label><br>
<button type="submit">Register</button>

</form>

</main>

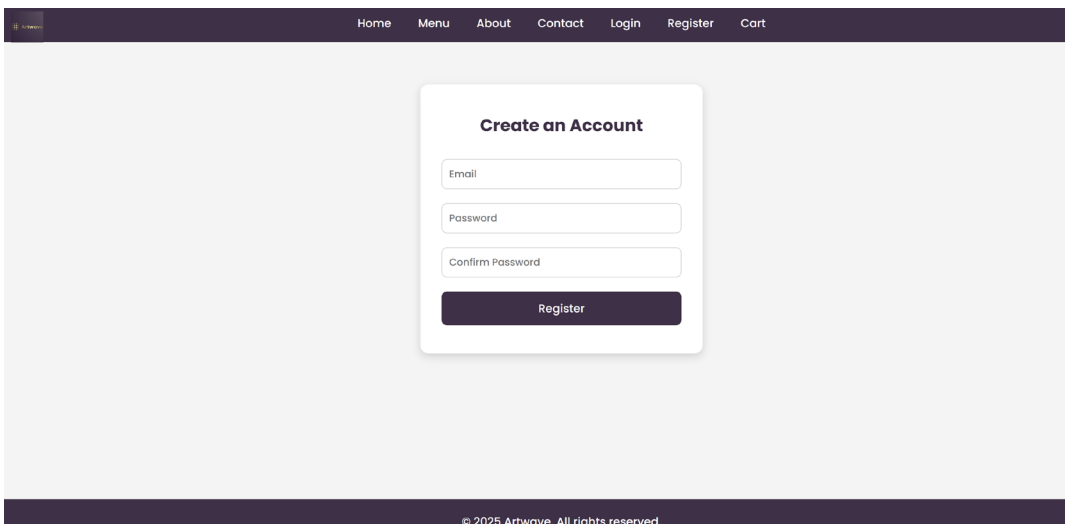
</body>

</html>

```

## **Output:**

F. registration page output:



## **Code:**

G. login page:

```

code: <!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Login - Artwave</title>

  <link rel="stylesheet" href="style.css">

</head>

```



```
<body>

<main>

  <h2>Login</h2>

  <form method="post" action="#">

    <label>Email: <input type="email" name="email" required></label><br>

    <label>Password: <input type="password" name="password" required></label><br>

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

  </form>

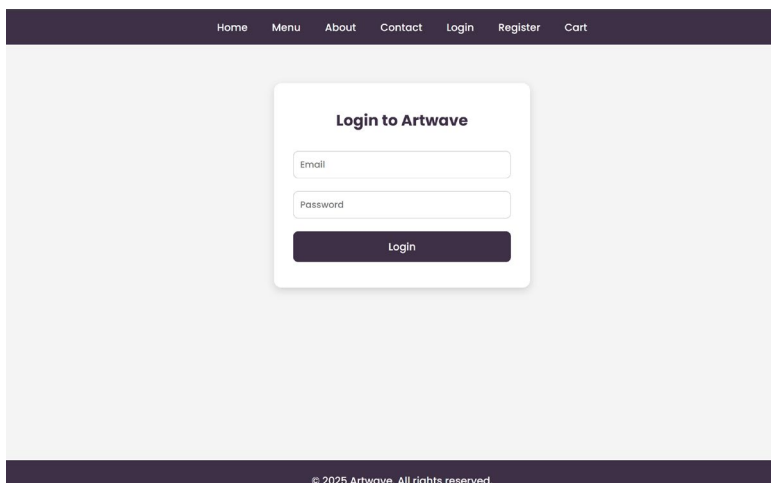
</main>

</body>

</html>
```

## **Output:**

G. login page output:



**Conclusion:** the Artwave website a platform for exploring and purchasing mood-based music tracks and digital artworks. Through this experiment, we created a responsive and user-centric interface by developing core pages including the Home, About, Menu (Product), Cart, Contact, Login, and Registration pages.

Each page was designed with clarity, usability, and visual appeal in mind, following a consistent aesthetic that reflects Artwave's creative brand. Features like categorized product listings, quantity selection, cart management, and secure form-based user interactions were implemented using HTML and CSS, laying the foundation for future integration with back-end technologies.

This experiment not only enhanced understanding of frontend web development but also emphasized the importance of user experience, visual design, and content organization in real-world digital platforms. It sets the stage for future enhancements such as JavaScript-based interactivity, PHP-based functionality, and database connectivity for a fully dynamic e-commerce system.

## Experiment No. 3

### Enhancing Layout Using CSS Grid for Artwave Website

#### Problem Statement:

- Enhance the layout of the **Artwave** homepage using CSS Grid.
- Use CSS Grid to layout the mood-based menu/product items (music tracks and artworks) in a structured, visually appealing manner.
- Style menu categories with clear headings, spacing, separators, images, descriptions, and prices.

#### Theory:

### CSS Grid for Enhancing Layout of the Artwave Website

#### Introduction to CSS Grid

CSS Grid Layout is a powerful two-dimensional CSS system designed for creating complex, responsive web layouts. Unlike Flexbox, which works primarily in one direction (row or column), CSS Grid controls layout along both rows and columns simultaneously. This makes it ideal for designing structured, visually balanced e-commerce sites or creative platforms like Artwave.

CSS Grid enables developers to:

- Create neat grids for product displays (e.g., music tracks, artworks).
- Define multiple distinct layout areas (hero, featured sections, menu categories).
- Align images, text, buttons consistently across different screen sizes.
- Build responsive designs that adjust smoothly for mobiles, tablets, and desktops.

#### Why CSS Grid for Artwave?

For Artwave, a platform showcasing mood-based music and artwork, an organized and aesthetically pleasing product display is critical for user engagement. Visitors should easily browse categories such as “Motivation Music,” “Relax Artworks,” or “Workout Tracks,” compare offerings, and add items to their cart with ease.

CSS Grid helps by:

- Arranging music tracks and artworks into flexible grids (e.g., 3 or 4 columns).
- Creating visually distinct sections for each mood category with proper headings and separators.
- Maintaining consistent spacing and alignment of images, titles, descriptions, prices, and action buttons.
- Supporting responsive layouts that adapt to various devices without complex overrides.

## 1. Home Page Layout with CSS Grid

The homepage uses CSS Grid to structure multiple sections efficiently:

- Full-width navigation header
- Hero banner featuring Artwave's tagline and call-to-action
- Multi-column highlight area showcasing mood categories or featured products
- Testimonials arranged in a clean row
- Footer with contact details and social media links

### Benefits:

- Grid areas are easy to define and position with precise control.
- Enables a harmonious balance between text, images, and buttons.
- Simplifies responsiveness and scalability.

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

The menu page displays Artwave's product catalog, grouped by mood categories such as:

- Motivation
- Study
- Relax
- Workout

Each mood category is styled with:

- Clear category headings with distinct fonts and colors
- Grid-arranged product cards featuring:
  - Product image or artwork preview
  - Title and short description
  - Price (highlighted)
  - "Add to Cart" button

### Example CSS Grid for product display:

CSS

Copy code

```
.products-grid {
```

```
display: grid;

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

grid-gap: 30px;

padding: 20px;

}
```

### Additional Styling Concepts:

- Category headings with larger font sizes and subtle background shading or underlines.
- Horizontal separators (thin lines or borders) between categories to visually divide sections.
- Hover effects on product cards and buttons to enhance interactivity.
- Responsive behavior leveraging auto-fit and minmax() to adjust columns based on screen size, minimizing media query needs.

### Code of homepage:- <!DOCTYPE html>

```
<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Artwave - Home</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<header class="site-header">

<nav>

<ul class="nav-list">

<li><a href="home.html">Home</a></li>

<li><a href="menu.html">Mood Menu</a></li>

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

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

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

</ul>

</nav>
```

```
</header>
```

```
<section class="hero">
```

```
<h1>Welcome to Artwave</h1>
```

```
<p>Your gateway to mood-based music and artworks.</p>
```

```
<button>Explore Now</button>
```

```
</section>
```

```
<section class="features">
```

```
<div class="feature">Motivation</div>
```

```
<div class="feature">Relax</div>
```

```
<div class="feature">Workout</div>
```

```
</section>
```

```
<section class="testimonials">
```

```
<article>"Artwave changed how I study — amazing tracks!" – User A</article>
```

```
<article>"Beautiful artworks to relax after work." – User B</article>
```

```
</section>
```

```
<footer class="site-footer">
```

```
<p>Contact us: info@artwave.com | Follow us on social media</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

```
Code of style.css:- * {
```

```
margin: 0;
```

```
padding: 0;
```

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

```
font-family: 'Poppins', sans-serif;
```

```
}
```

```
body, html {  
  height: 100%;  
  overflow: hidden;  
}
```

```
/* Theme colors */  
  
:root {  
  --primary-color: #412E46; /* deep purple */  
  --accent-color: #d4af37; /* golden */  
  --hover-color: #b38b2e;  
  --overlay-color: rgba(65, 46, 70, 0.6);  
}
```

```
/* Background video setup */  
  
.background-video {  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
  z-index: -2;  
}
```

```
/* Semi-transparent overlay for readability */  
  
.overlay {  
  position: fixed;
```

```
top: 0;

left: 0;

width: 100%;

height: 100%;

background: var(--overlay-color);

z-index: -1;

}
```

```
/* Logo image styling */
```

```
.logo {

  position: absolute;

  top: 0;

  left: 20px;

  z-index: 1001; /* stays above navbar */

  padding: 10px 0; /* optional spacing from the top */

}
```

```
.logo img {

  height: 44px; /* match navbar height */

  width: auto;

  display: block;

}
```

```
/* Hero content */
```

```
.hero-content {

  position: relative;

  height: 100vh;

  display: flex;

  flex-direction: column;
```



```
align-items: center;

justify-content: center;

text-align: center;

color: white;

padding: 0 2rem;

}
```

```
/* Website name heading */

.site-name {

  font-size: 2.5rem;

  font-weight: 700;

  color: var(--accent-color); /* golden color */

  margin-bottom: 1rem;

}
```

```
/* Main slogan */

.hero-content h1 {

  font-size: 3rem; /* Increased size for better visibility */

  color: var(--accent-color); /* golden color */

  margin-bottom: 1rem;

  text-shadow: 2px 4px rgba(0, 0, 0, 0.5); /* Adding shadow for better contrast */

}
```

```
/* Subtitle */

.hero-content p {

  font-size: 1.2rem;

  margin-bottom: 2rem;

}
```

```
/* Get Started Button */

.get-started-btn {

  padding: 0.75rem 2rem;

  background-color: transparent;

  border: 2px solid var(--accent-color); /* Golden border */

  color: white;

  text-decoration: none;

  font-size: 1rem;

  border-radius: 30px;

  font-weight: 500;

  transition: background-color 0.3s, color 0.3s;

}


.get-started-btn:hover {

  background-color: var(--accent-color); /* Golden background on hover */

  color: var(--primary-color); /* Dark text color */

}


/* menu css starts from here

.

.

.

*/

/* Dark Purple Background for Menu Page */

body {

  background-color: #412E46; /* Deep purple background */

  color: white; /* White text for contrast */

}
```

```
/* Menu Content */
```

```
.menu-content {  
  
  position: relative;  
  
  height: 100vh;  
  
  display: flex;  
  
  flex-direction: column;  
  
  align-items: center;  
  
  justify-content: center;  
  
  text-align: center;  
  
  color: white;  
  
  padding: 0 2rem;  
}
```

```
/* Website name heading */
```

```
.site-name {  
  
  font-size: 2.5rem;  
  
  font-weight: 700;  
  
  color: #d4af37; /* Golden color */  
  
  margin-bottom: 1rem;  
}
```

```
/* Subtitle */
```

```
.menu-content p {  
  
  font-size: 1.2rem;  
  
  margin-bottom: 2rem;  
}
```

```
/* Mood Buttons */
```

```
.mood-buttons {
```

```
display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 20px;

padding: 50px 20px;
}

.mood-button-container {

text-align: center;

width: 30%; /* 3 buttons per row */

margin: 10px;
}

.mood-button {

background-color: transparent;

border: 2px solid #d4af37; /* Golden border */

color: white;

font-size: 18px;

padding: 15px 30px;

text-transform: uppercase;

cursor: pointer;

transition: all 0.3s ease;

font-weight: bold;

border-radius: 5px;
}

/* Button Hover Effects */

.mood-button:hover {

background-color: #d4af37; /* Golden background on hover */
```

```
color: #412E46; /* Dark purple text */  
  
transform: scale(1.1); /* Slightly larger on hover */  
}
```

```
/* Footer Styling */
```

```
footer {  
  
    text-align: center;  
  
    padding: 20px;  
  
    background-color: #412E46; /* Purple footer */  
  
    color: white;  
  
    position: absolute;  
  
    width: 100%;  
  
    bottom: 0;  
}
```

```
#music-container {  
  
    text-align: center;  
  
    margin-top: 20px;  
}
```

```
#music-video {  
  
    width: 100%;  
  
    max-width: 560px;  
  
    height: 315px;  
}
```

```
/* Navbar styling */
```

```
.navbar {  
  
    display: flex;
```

```
justify-content: center;

background-color: #412E46;

padding: 15px 0;

gap: 30px;

position: sticky;

top: 0;

z-index: 999;

}
```

```
.navbar a {

text-decoration: none;

color: #fcb3ff;

font-weight: 500;

transition: color 0.3s ease;

}
```

```
.navbar a:hover {

color: #7b4b94;

}
```

```
/* Style for the logout button container */
```

```
.logout-container {

position: absolute;

top: 20px;

right: 20px;

z-index: 10; /* Make sure it's above other elements */

}
```

```
/* Style for the logout button */
```

```
.logout-btn {
```

```

background-color: #422F48;

color: white;

padding: 10px 20px;

border-radius: 8px;

text-decoration: none;

font-size: 1rem;

cursor: pointer;
}

.logout-btn:hover {

background-color: #311f36; /* Darker color on hover */
}

```

**Code of cart:-** <!DOCTYPE html>

```

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Artwave - Cart</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<header class="site-header">

<!-- Navigation -->

</header>

<main>

<h2>Your Cart</h2>

<table class="cart-table">

<thead>

```

```
<tr>

  <th>Item</th>

  <th>Price</th>

  <th>Quantity</th>

  <th>Subtotal</th>

  <th>Remove</th>

</tr>

</thead>

<tbody>

  <tr>

    <td>Motivation Track 1</td>

    <td>$1.99</td>

    <td><input type="number" value="1" min="1" /></td>

    <td>$1.99</td>

    <td><button>Remove</button></td>

  </tr>

  <!-- More items -->

</tbody>

</table>

<div class="checkout-section">

  <p>Total: $XX.XX</p>

  <button>Proceed to Checkout</button>

</div>

</main>

<footer class="site-footer">

  <!-- Footer -->

</footer>

</body>
```



```
</html>
```

```
Css for it :- .cart-table {
```

```
width: 100%;
```

```
border-collapse: collapse;
```

```
margin-bottom: 2rem;
```

```
}
```

```
.cart-table th, .cart-table td {
```

```
border: 1px solid #ddd;
```

```
padding: 1rem;
```

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

```
}
```

```
.cart-table th {
```

```
background-color: #720026;
```

```
color: white;
```

```
}
```

```
.checkout-section {
```

```
text-align: right;
```

```
font-size: 1.2rem;
```

```
font-weight: 700;
```

```
}
```

```
.checkout-section button {
```

```
background: #7f00ff;
```

```
color: white;
```

```
border: none;
```

```
padding: 0.7rem 1.5rem;  
  
border-radius: 5px;  
  
cursor: pointer;  
}
```

```
.checkout-section button:hover {  
  
  background: #9b30ff;  
  
}
```

**Code of about us:-** <!DOCTYPE html>

```
<html lang="en">  
  
<head>  
  
  <meta charset="UTF-8" />  
  
  <title>Artwave - About Us</title>  
  
  <link rel="stylesheet" href="styles.css" />  
  
</head>  
  
<body>  
  
  <header class="site-header">  
  
    <!-- Navigation -->  
  
  </header>  
  
  
  <main>  
  
    <h2>About Artwave</h2>  
  
    <section class="about-history">  
  
      <h3>Our Story</h3>  
  
      <p>Artwave started with a mission to blend music and artwork to enhance moods...</p>  
  
    </section>  
  
    <section class="about-mission">  
  
      <h3>Mission & Vision</h3>  
  
      <p>To inspire and uplift through curated mood-based content that touches the soul.</p>
```

```
</section>
```

```
<section class="team">
```

```
<h3>Meet the Team</h3>
```

```
<p>Photos and bios of founders and contributors.</p>
```

```
</section>
```

```
</main>
```

```
<footer class="site-footer">
```

```
<!-- Footer -->
```

```
</footer>
```

```
</body>
```

```
</html>
```

**Code of contact :-**<!DOCTYPE html>

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<title>Artwave - Contact</title>
```

```
<link rel="stylesheet" href="styles.css" />
```

```
</head>
```

```
<body>
```

```
<header class="site-header">
```

```
<!-- Navigation -->
```

```
</header>
```

```
<main>
```

```
<h2>Contact Us</h2>
```

```
<form class="contact-form">
```

```
<label for="name">Name</label><br />
```

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

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

```
<input type="email" id="email" name="email" required /><br />
```

```
<label for="subject">Subject</label><br />
```

```
<input type="text" id="subject" name="subject" /><br />
```

```
<label for="message">Message</label><br />
<textarea id="message" name="message" rows="5" required></textarea><br />
```

```
<button type="submit">Send Message</button>
</form>
</main>
```

```
<footer class="site-footer">
  <!-- Footer -->
</footer>
</body>
</html>
```

**Code of register:-**

```
.contact-form {
  max-width: 500px;
  margin: 2rem auto;
  display: grid;
  gap: 1rem;
}
```

```
.contact-form input,
.contact-form textarea {
  padding: 0.5rem;
  border-radius: 5px;
  border: 1px solid #ccc;
  font-family: 'Poppins', sans-serif;
}
```

```
.contact-form button {
  background: #7f00ff;
  color: white;
  border: none;
  padding: 0.7rem;
  border-radius: 5px;
  cursor: pointer;
  font-weight: 700;
}
```

```
.contact-form button:hover {
  background: #9b30ff;
```

```
}
```

**Code of login :-** <!DOCTYPE html>

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<title>Artwave - Login</title>
```

```
<link rel="stylesheet" href="styles.css" />
```

```
</head>
```

```
<body>
```

```
<header class="site-header">
```

```
<!-- Navigation -->
```

```
</header>
```

```
<main>
```

```
<h2>Login to Artwave</h2>
```

```
<form class="login-form">
```

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

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

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

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

```
<button type="submit">Login</button>
```

```
</form>
```

```
</main>
```

```
<footer class="site-footer">
```

```
<!-- Footer -->
```

```
</footer>
```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<title>Artwave - Login</title>
```

```
<link rel="stylesheet" href="styles.css" />
```

```
</head>
```

```

<body>
  <header class="site-header">
    <!-- Navigation -->
  </header>

  <main>
    <h2>Login to Artwave</h2>
    <form class="login-form">
      <label for="email">Email</label>
      <input type="email" id="email" name="email" required />

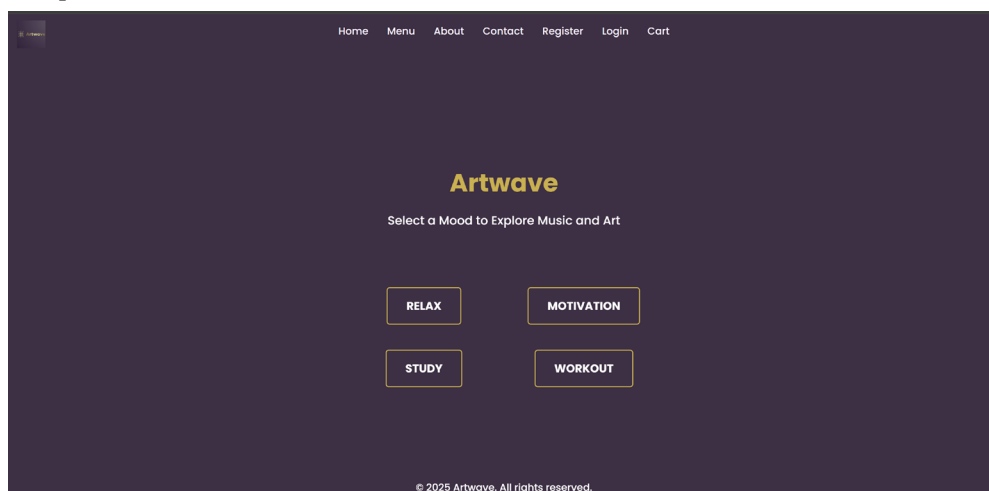
      <label for="password">Password</label>
      <input type="password" id="password" name="password" required />

      <button type="submit">Login</button>
    </form>
  </main>

  <footer class="site-footer">
    <!-- Footer -->
  </footer>
</body>
</html>

```

Output:-



**Code of motivation.html:-** <!DOCTYPE html>

```

<html lang="en">
<head>
  <meta charset="UTF-8">

```

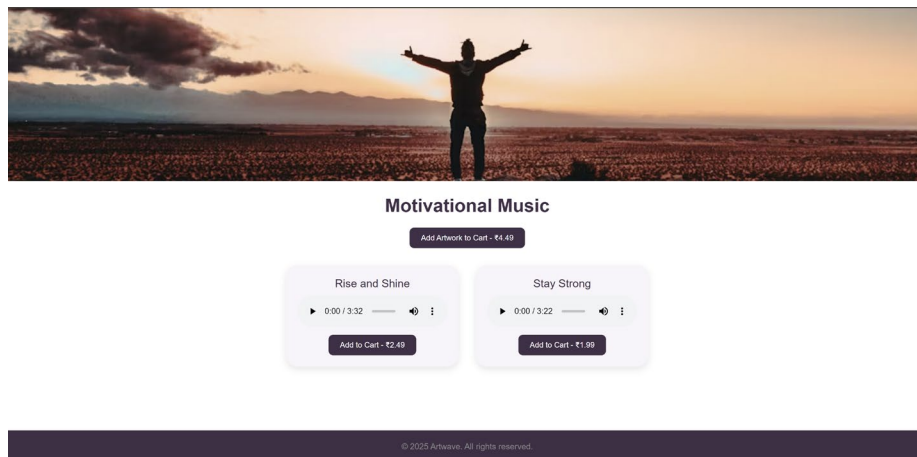
```
<title>Motivation - Artwave</title>
<link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <header>
    <h1>Motivation</h1>
    <nav>
      <a href="index.html">Home</a>
      <a href="cart.html">Cart</a>
    </nav>
  </header>

  <main class="products-grid">
    <!-- Music Track -->
    <div class="product-card" data-type="music" data-id="1" data-price="50">
      <audio controls src="assets/music/motivation1.mp3"></audio>
      <h3>Inspire Beat</h3>
      <p>Price: ₹50</p>
      <button class="add-to-cart">Add to Cart</button>
    </div>

    <!-- Artwork -->
    <div class="product-card" data-type="art" data-id="101" data-price="150">
      
      <h3>Rise & Shine</h3>
      <p>Price: ₹150</p>
      <button class="add-to-cart">Add to Cart</button>
    </div>
  </main>

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

Output:-



### Code of relax.html:- <!DOCTYPE html>

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Relax - Artwave</title>
```

```
  <link rel="stylesheet" href="assets/css/style.css">
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1>Relax</h1>
```

```
    <nav>
```

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

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

```
    </nav>
```

```
  </header>
```

```
<main class="products-grid">
```

```
  <!-- Music Track -->
```

```
  <div class="product-card" data-type="music" data-id="2" data-price="60">
```

```
    <audio controls src="assets/music/relax1.mp3"></audio>
```

```
    <h3>Peaceful Flow</h3>
```

```
    <p>Price: ₹60</p>
```

```
    <button class="add-to-cart">Add to Cart</button>
```

```
  </div>
```

```
  <!-- Artwork -->
```

```
  <div class="product-card" data-type="art" data-id="102" data-price="200">
```

```
    
```

```
    <h3>Evening Calm</h3>
```



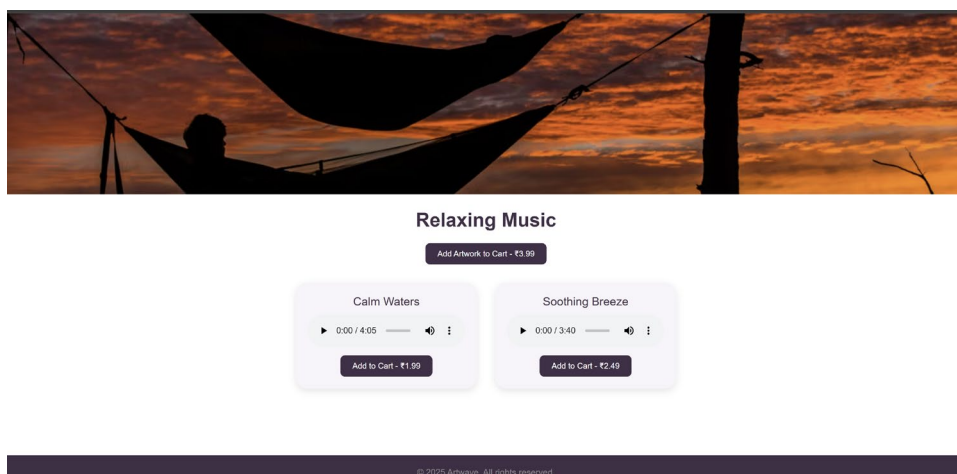
```

    <p>Price: ₹200</p>
    <button class="add-to-cart">Add to Cart</button>
  </div>
</main>

<script src="assets/js/cart.js"></script>
</body>
</html>

```

Output:-



**Code of study.html:-** <!DOCTYPE html>

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Study - Artwave</title>
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <header>
    <h1>Study</h1>
    <nav>
      <a href="index.html">Home</a>

```

```

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

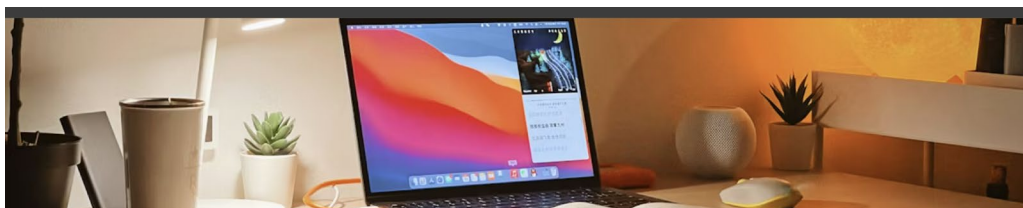
<main class="products-grid">
  <!-- Music Track -->
  <div class="product-card" data-type="music" data-id="3" data-price="70">
    <audio controls src="assets/music/study1.mp3"></audio>
    <h3>Focus Rhythm</h3>
    <p>Price: ₹70</p>
    <button class="add-to-cart">Add to Cart</button>
  </div>

  <!-- Artwork -->
  <div class="product-card" data-type="art" data-id="103" data-price="180">
    
    <h3>Concentration Space</h3>
    <p>Price: ₹180</p>
    <button class="add-to-cart">Add to Cart</button>
  </div>
</main>

<script src="assets/js/cart.js"></script>
</body>
</html>

```

Output:



### Study Music

Add Artwork to Cart - ₹3.99

Focus Beats

▶ 0:00 / 2:12 ◀ 🔊 ⋮

Add to Cart - ₹2.29

Calm Mind

▶ 0:00 / 2:07 ◀ 🔊 ⋮

Add to Cart - ₹1.79

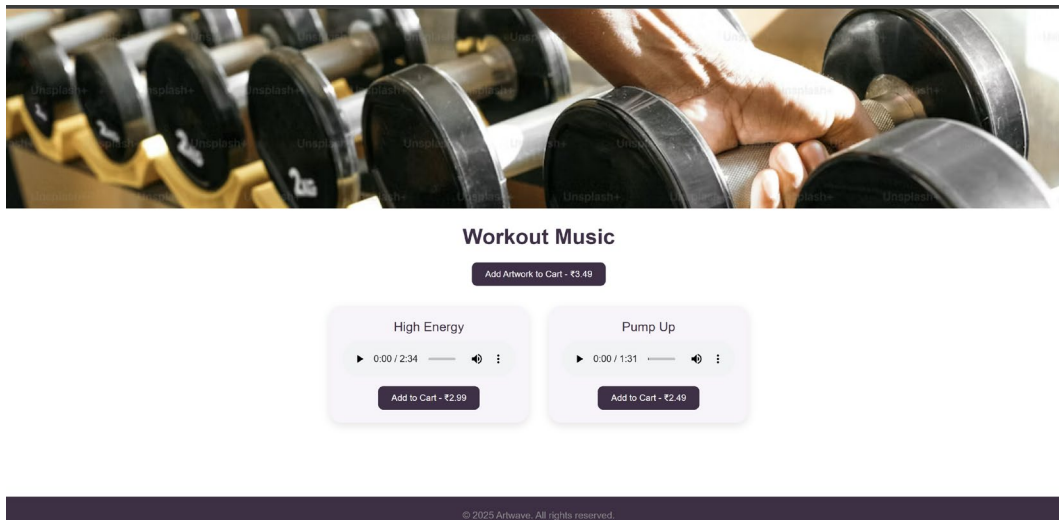
Code of workout.html:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Workout - Artwave</title>
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <header>
    <h1>Workout</h1>
    <nav>
      <a href="index.html">Home</a>
      <a href="cart.html">Cart</a>
    </nav>
  </header>

  <main class="products-grid">
    <!-- Music Track -->
    <div class="product-card" data-type="music" data-id="4" data-price="80">
      <audio controls src="assets/music/workout1.mp3"></audio>
      <h3>Power Jam</h3>
      <p>Price: ₹80</p>
      <button class="add-to-cart">Add to Cart</button>
    </div>

    <!-- Artwork -->
    <div class="product-card" data-type="art" data-id="104" data-price="220">
      
      <h3>Energy Burst</h3>
      <p>Price: ₹220</p>
      <button class="add-to-cart">Add to Cart</button>
    </div>
  </main>

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



## Conclusion :-

In this experiment, we enhanced the layout of the Artwave website using CSS Grid, improving both visual structure and user experience. By organizing homepage sections and product listings in a clean, responsive grid format, we achieved a modern, consistent, and mobile-friendly design. CSS Grid allowed precise alignment of elements like music tracks and artwork, making the site more intuitive and appealing for users. This experiment demonstrates how CSS Grid effectively transforms static content into a dynamic, well-organized layout.

## Experiment No.4

CSS Theory:-

### 1. Why CSS Styling Matters in Artwave

In creative platforms like Artwave, where users explore moods through **music and visual art**, the **styling** plays a crucial role in user perception. Aesthetic design, intuitive layouts, and readable forms enhance **emotional appeal** and **usability**, ensuring users feel inspired, relaxed, or motivated—depending on the mood they explore.

CSS enables us to:

- Create clean, immersive layouts that reflect each mood
- Improve form usability on login, registration, and contact pages
- Enhance cart readability and checkout clarity
- Ensure visual consistency across music and artwork items

## Page-wise CSS Styling Theory

### 1. Cart Page

The cart page must clearly present selected tracks and artwork, including quantity and price.

#### Styling Techniques:

- Add spacing between items (padding/margin) for visual separation
- Style quantity inputs and update/remove buttons with smooth borders and hover effects
- Highlight subtotals using bold font and soft background color

- Maintain consistent font sizes and spacing for cart item names and prices

**Result:** A clean, music/art-oriented cart experience that supports smooth checkout and encourages final purchases.

## 2. About Us Page

This page expresses Artwave's mission—promoting creativity through moods.

### Styling Techniques:

- Use **line height and padding** for comfortable reading
- Segment sections like “Our Mission” and “Our Vision” with light backgrounds or horizontal lines
- Style team photos or concept images with soft edges and spacing
- Highlight core values using boxed quotes or accent colors

**Result:** A trustworthy, artistic layout that emotionally connects visitors to Artwave's purpose.

## 3. Contact Page

A well-styled contact form builds trust for artist inquiries, music feedback, or support.

### Styling Techniques:

- Style all form fields with equal width, inner padding, and gentle border-radius
- Add focus states (e.g., glowing or color-changing borders)
- Separate fields vertically using consistent margin-bottom
- Style submit button with hover transitions and brand colors (e.g., deep purple or green)

**Result:** A friendly, artistic interface that encourages user interaction and inquiries.

## 4. Admin/User Registration Form

Registration is essential for personalized playlists, artwork uploads, and cart access.

### Styling Techniques:

- Group fields logically: personal details, login credentials
- Use labeled input fields with spacing and placeholder hints
- Round corners, add soft shadows, and use light backgrounds for the form container
- Style buttons with color transitions and visual hierarchy

**Result:** A user-friendly form that feels secure, creative, and easy to complete.

## 5. Admin/User Login Form

The login interface must feel professional and match the visual tone of the platform.

### Styling Techniques:

- Center the login box on screen with consistent padding
- Use clean borders and focused input states
- Add styling for error/success messages (color-coded)
- Incorporate subtle design accents like icon placeholders or fade-in animations

**Result:** A stylish and functional login that aligns with Artwave's artistic identity and improves accessibility.

Code of cart.html:- <!DOCTYPE html>

```
<html>
<head>
  <title>Cart - Artwave</title>
  <link rel="stylesheet" href="cart.css">
</head>
<body>
  <div class="cart-container">
    <h1>Your Cart</h1>
    <div class="cart-item">
      <span class="item-name">Relaxing Melody</span>
      <input type="number" value="1">
      <span class="item-price">₹99</span>
      <button class="remove-btn">Remove</button>
    </div>
    <div class="total">
      Subtotal: ₹99
    </div>
    <button class="checkout-btn">Proceed to Checkout</button>
  </div>
</body>
</html>
```

Css of cart:- /\* cart.css \*/

```
body {
  font-family: Poppins, sans-serif;
  background: #f8f6ff;
  padding: 20px;
}

.cart-container {
  max-width: 700px;
  margin: auto;
  background: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 12px rgba(0,0,0,0.1);
}

.cart-item {
  display: flex;
  justify-content: space-between;
  padding: 10px 0;
```

```
border-bottom: 1px solid #eee;  
}
```

```
.cart-item input {  
  width: 60px;  
  padding: 5px;  
}
```

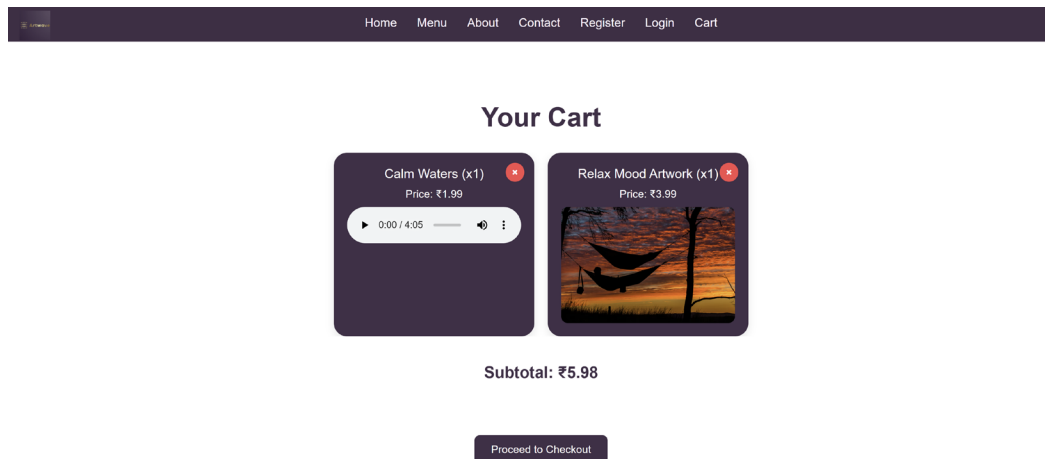
```
.item-price {  
  font-weight: bold;  
}
```

```
.remove-btn {  
  background: #ff4d4d;  
  border: none;  
  padding: 5px 10px;  
  color: #fff;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

```
.total {  
  text-align: right;  
  margin-top: 20px;  
  font-size: 18px;  
}
```

```
.checkout-btn {  
  width: 100%;  
  background: #6a0dad;  
  color: #fff;  
  padding: 12px;  
  font-size: 16px;  
  border: none;  
  border-radius: 6px;  
  margin-top: 10px;  
  cursor: pointer;  
}
```

Output of cart:-



Code of about.html:-

```
<!DOCTYPE html>
<html>
<head>
  <title>About - Artwave</title>
  <link rel="stylesheet" href="about.css">
</head>
<body>
  <div class="about-container">
    <h1>About Artwave</h1>
    <p>Artwave was founded to blend visual art and audio therapy into a single immersive platform.</p>
    <section class="mission">
      <h2>Our Mission</h2>
      <p>To help people heal, study, and express emotions through curated music and visual artworks.</p>
    </section>
    <section class="team">
      <h2>Our Team</h2>
      
    </section>
  </div>
</body>
</html>
```

Css:- /\* about.css \*/

```
body {
  font-family: Poppins, sans-serif;
  background: #fffafc;
  padding: 20px;
  color: #333;
}
```



```
.about-container {
  max-width: 800px;
  margin: auto;
}
```

```
h1, h2 {
  color: #6a0dad;
}
```

```
p {
  line-height: 1.6;
  margin-bottom: 15px;
}
```

```
.team img {
  width: 100%;
  max-width: 400px;
  border-radius: 8px;
  margin-top: 10px;
}
```

Code of contact.html:- <!DOCTYPE html>

```
<html>
<head>
  <title>Contact - Artwave</title>
  <link rel="stylesheet" href="contact.css">
</head>
<body>
  <div class="contact-form">
    <h1>Contact Us</h1>
    <form>
      <input type="text" placeholder="Your Name" required>
      <input type="email" placeholder="Your Email" required>
      <textarea placeholder="Your Message"></textarea>
      <button type="submit">Send</button>
    </form>
  </div>
</body>
</html>
```

Css :- /\* contact.css \*/

```
body {
  font-family: Poppins, sans-serif;
  background: #f3efff;
  padding: 40px;
```

```

}

.contact-form {
  background: #fff;
  max-width: 600px;
  margin: auto;
  padding: 25px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

.contact-form input, .contact-form textarea {
  width: 100%;
  margin-bottom: 15px;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #ccc;
}

.contact-form button {
  background-color: #6a0dad;
  color: white;
  border: none;
  padding: 10px 15px;
  border-radius: 5px;
  cursor: pointer;
}

```

```

Code of register.html:- <!DOCTYPE html>
<html>
<head>
  <title>Register - Artwave</title>
  <link rel="stylesheet" href="register.css">
</head>
<body>
  <div class="form-box">
    <h1>Create Account</h1>
    <form>
      <input type="text" placeholder="Full Name" required>
      <input type="email" placeholder="Email" required>
      <input type="password" placeholder="Password" required>
      <input type="password" placeholder="Confirm Password" required>
      <button type="submit">Register</button>
    </form>
  </div>
</body>
</html>

```

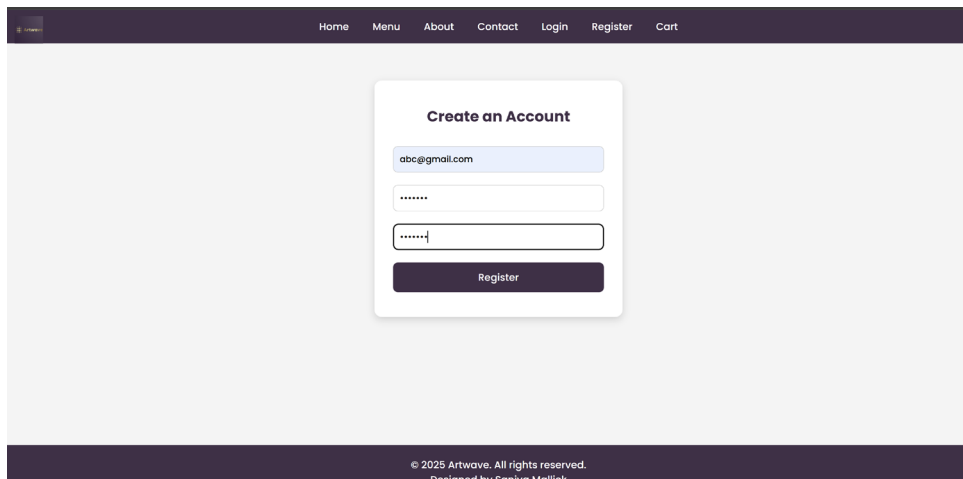
```
</div>
</body>
</html>
/* register.css */
body {
  font-family: Poppins, sans-serif;
  background: #f0f0ff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.form-box {
  background: #fff;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 0 12px rgba(0,0,0,0.1);
  width: 300px;
}

.form-box input {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

.form-box button {
  width: 100%;
  background: #6a0dad;
  color: #fff;
  padding: 10px;
  border: none;
  border-radius: 5px;
}
```

Output of register:-



Code of login.html:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Login - Artwave</title>
  <link rel="stylesheet" href="login.css">
</head>
<body>
  <div class="login-form">
    <h1>Login</h1>
    <form>
      <input type="email" placeholder="Email" required>
      <input type="password" placeholder="Password" required>
      <button type="submit">Login</button>
    </form>
  </div>
</body>
</html>
```

Css :-- /\* login.css \*/

```
body {
  font-family: Poppins, sans-serif;
  background: #e8e2f0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

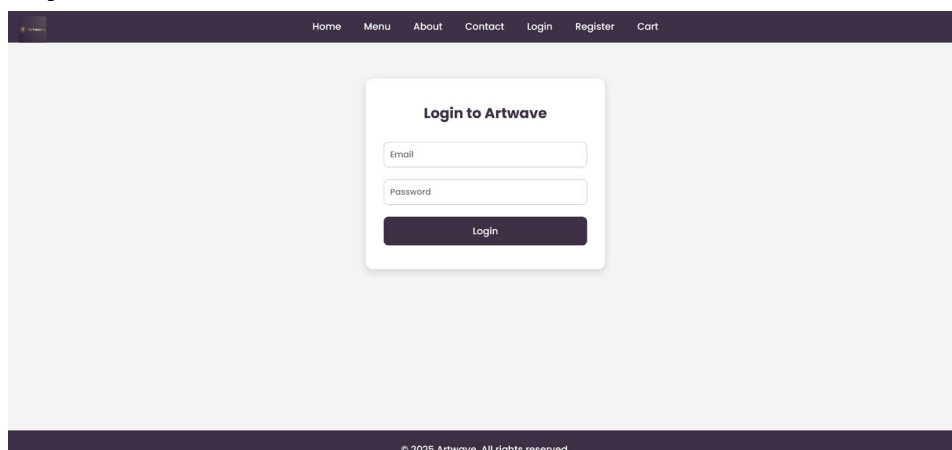
```
.login-form {
  background: white;
  padding: 30px;
  border-radius: 10px;
```

```
box-shadow: 0 0 15px rgba(0,0,0,0.1);  
width: 300px;  
}
```

```
.login-form input {  
width: 100%;  
padding: 10px;  
margin-bottom: 15px;  
border-radius: 5px;  
border: 1px solid #ccc;  
}
```

```
.login-form button {  
background-color: #6a0dad;  
color: white;  
width: 100%;  
padding: 10px;  
border: none;  
border-radius: 5px;  
}
```

Output:-



## Conclusion :

**Experiment No.4** focused on improving the user interface of key Artwave pages using CSS. By implementing padding, alignment, typography, and responsive design principles, the pages like Cart, Contact, About, Register, and Login now offer a cleaner, user-friendly, and more professional appearance. This experiment demonstrated how structured styling enhances user experience and engagement.

## Experiment No.5

### JavaScript Theory: User Registration, Login, Validation, and Cart Functionality for Artwave Website

#### Introduction

In the context of the **Artwave** website—a platform where users can explore mood-based music and visual artwork—JavaScript plays a vital role in enhancing interactivity and user experience. This experiment focuses on implementing JavaScript functionality to support essential features like **user registration, login, form validation, and cart behavior**.

Interactive client-side scripting not only improves usability but also helps in data validation before interacting with backend services, reducing errors and improving performance.

---

#### 1. User Registration and Login Forms

These forms are fundamental for establishing user identity and enabling personalized interactions such as saving favorite tracks, purchasing artwork, or tracking orders.

##### Registration Form

Users can create an Artwave account by submitting details such as:

- Full Name
- Email Address
- Password and Confirm Password

##### JavaScript Responsibilities:

- Prevent empty form submissions
- Validate email format using regular expressions
- Enforce password criteria (e.g., minimum 8 characters, mix of characters)
- Ensure password and confirm password fields match
- Provide real-time validation messages (e.g., “Password too short”, “Email invalid”)

##### Login Form

Returning users log in using their credentials.

##### JavaScript Responsibilities:

- Ensure both email and password fields are filled
- Validate email format
- Match credentials against previously stored data (local storage or backend in future)
- Redirect to the user dashboard or mood page upon successful login
- Display error messages for incorrect login attempts

#### 2. JavaScript Form Validations

Form validation is key to ensuring the correctness of input data and providing users with immediate feedback, avoiding repeated backend requests.

##### Key Validation Tasks in Artwave:

- Mandatory fields must not be left blank
- Email must match a standard format (e.g., name@example.com)
- Password must meet strength rules (length, character type)

- Passwords must match
- Display contextual error messages beneath each field

### Why It

#### Matters:

On the Artwave site, real-time validation ensures that users have a smooth onboarding process, improving trust and reducing user frustration.

### 3. Cart Functionality

The **Artwave Cart** allows users to manage purchases of **music tracks** and **artworks**. JavaScript ensures the cart updates instantly and remains responsive.

#### JavaScript Cart Features:

- Dynamically add selected items to the cart
- Display all cart items with details: name, quantity, price
- Update quantity with recalculated subtotal in real time
- Allow item removal from cart
- Store cart state using localStorage for session persistence
- Enable checkout preparation based on selected items
- Use arrays of objects to structure cart data (e.g., { id, name, type, price, qty })

#### DOM Manipulation Examples:

- Generating cart rows dynamically
- Listening for quantity change events
- Updating total cost display
- Managing button states (Add to Cart → In Cart)

## CODES:-

#### Register.js:-

```
document.getElementById("registerForm").addEventListener("submit", function (e) {
  e.preventDefault();
```

```
const name = document.getElementById("name").value.trim();
const email = document.getElementById("email").value.trim();
const password = document.getElementById("password").value;
const confirmPassword = document.getElementById("confirmPassword").value;
```

```
if (!name || !email || !password || !confirmPassword) {
  alert("Please fill out all fields.");
  return;
}
```

```
const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (!emailRegex.test(email)) {
  alert("Please enter a valid email.");
  return;
}
```

```
if (password.length < 6) {
```

```

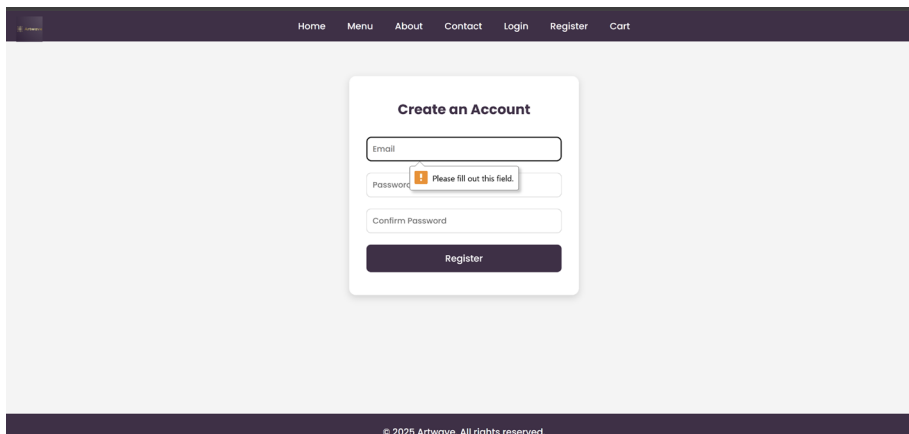
    alert("Password must be at least 6 characters.");
    return;
}

if (password !== confirmPassword) {
    alert("Passwords do not match.");
    return;
}

alert("Registration successful!");
// Save data or send to server
});

```

Output:-



The screenshot displays a web page with a dark purple header containing navigation links: Home, Menu, About, Contact, Login, Register, and Cart. The main content area features a light gray background with a central white box titled "Create an Account". Inside this box are three input fields: "Email", "Password", and "Confirm Password". The "Password" field is currently empty and has a small orange warning icon and the text "Please fill out this field." next to it. Below the input fields is a dark purple button labeled "Register". At the bottom of the page, a dark purple footer contains the text "© 2025 Artwave. All rights reserved."

login.js:-

```

document.getElementById("loginForm").addEventListener("submit", function (e) {
    e.preventDefault();

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

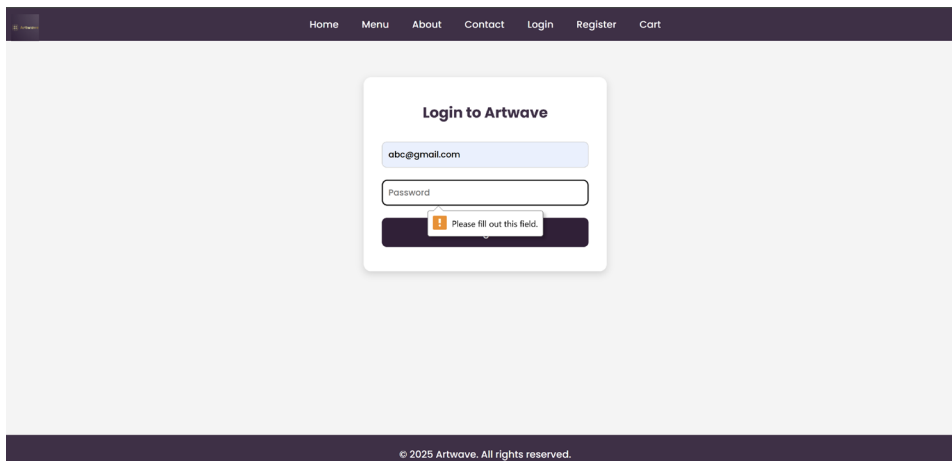
    if (!email || !password) {
        alert("Please enter both email and password.");
        return;
    }

    // Placeholder check; replace with actual validation logic
    if (email === "test@artwave.com" && password === "123456") {
        alert("Login successful!");
        window.location.href = "home.html";
    } else {
        alert("Invalid credentials.");
    }
});

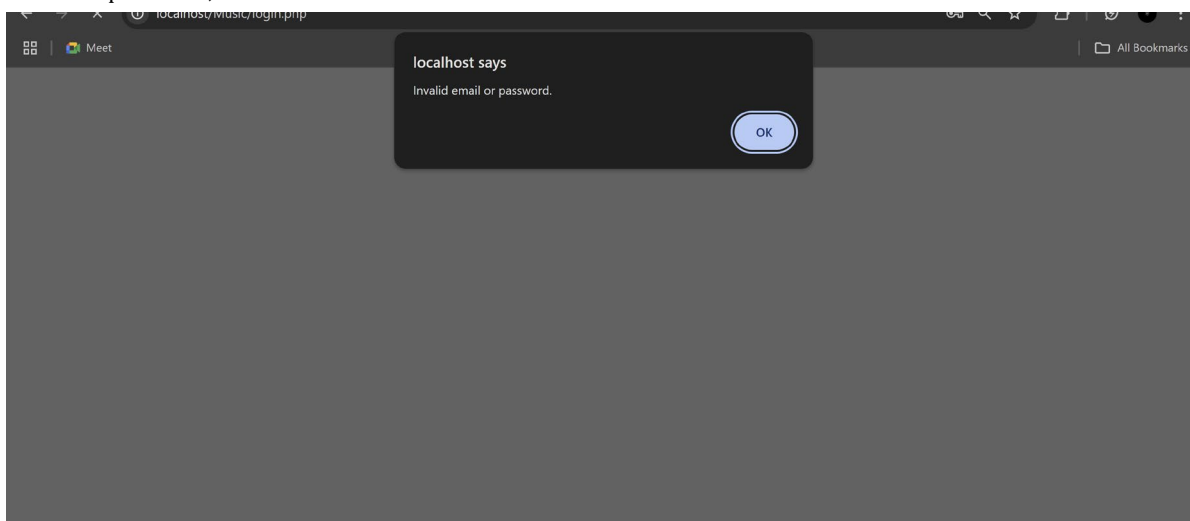
```

Output:-





If invalid password;-



cart.js:-

```
let cart = [];
```

```
function addToCart(itemId, itemName, itemPrice) {
  const existingItem = cart.find(item => item.id === itemId);
  if (existingItem) {
    existingItem.quantity += 1;
  } else {
    cart.push({ id: itemId, name: itemName, price: itemPrice, quantity: 1 });
  }
  updateCart();
}
```

```
function removeFromCart(itemId) {
  cart = cart.filter(item => item.id !== itemId);
  updateCart();
}
```

```
function updateCart() {
  const cartContainer = document.getElementById("cartItems");
  cartContainer.innerHTML = "";
```

```
let subtotal = 0;
cart.forEach(item => {
  const itemTotal = item.price * item.quantity;
```

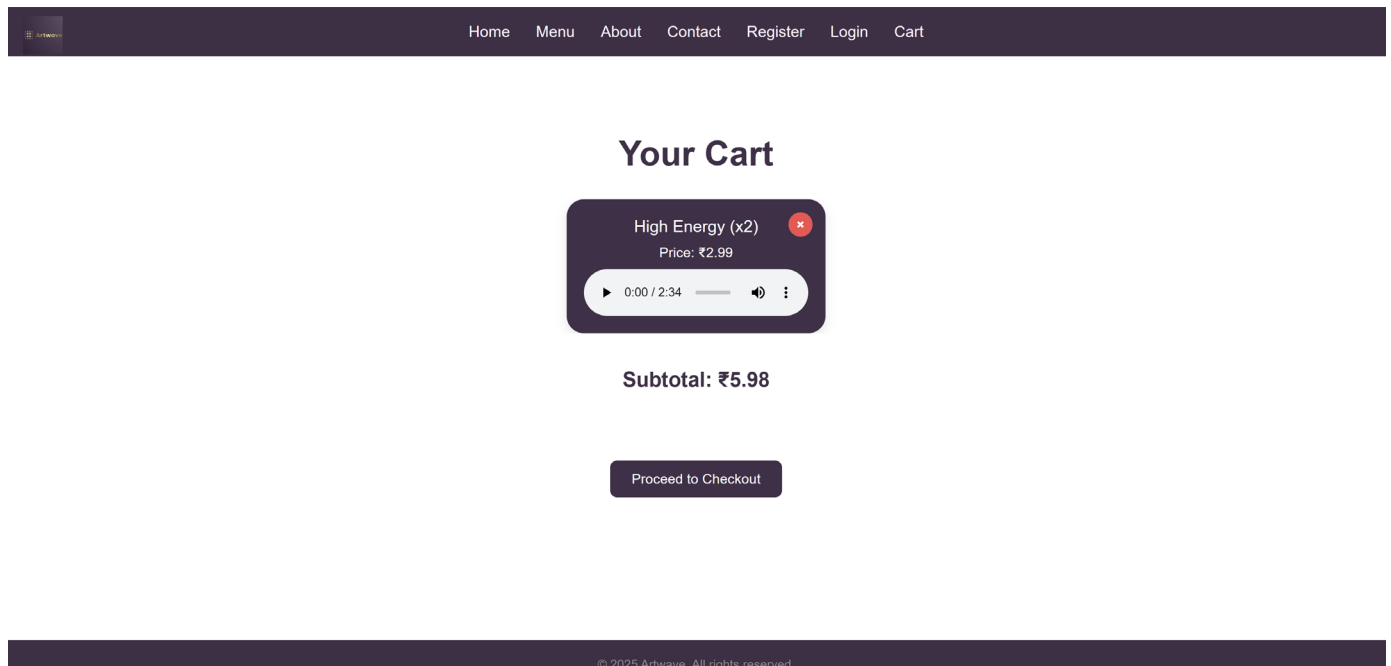
```
subtotal += itemTotal;
```

```
const cartRow = `
<div class="cart-item">
  <span>${item.name}</span>
  <input type="number" value="${item.quantity}" onchange="updateQuantity('${item.id}', this.value)" />
  <span>₹${itemTotal}</span>
  <button onclick="removeFromCart('${item.id}')">Remove</button>
</div>
`;
cartContainer.innerHTML += cartRow;
});
```

```
document.getElementById("cartSubtotal").innerText = `Subtotal: ₹${subtotal}`;
}
```

```
function updateQuantity(itemId, newQty) {
  const item = cart.find(item => item.id === itemId);
  if (item && newQty > 0) {
    item.quantity = parseInt(newQty);
    updateCart();
  }
}
```

Output:-



### Conclusion :

This experiment successfully demonstrated the use of JavaScript for enhancing the interactivity and functionality of core user operations in the Artwave website. By implementing user registration and login with form validations, and building a dynamic shopping cart system, we achieved a responsive and functional user interface that improves the overall user experience and maintains input accuracy.

## Experiment No.6

### JavaScript Theory: Persistent Login and Cart Functionality using Web Storage API (Artwave Website)

#### Introduction

For an interactive and seamless user experience on a music and artwork platform like **Artwave**, preserving user sessions and cart items is essential. JavaScript's **Web Storage API** provides simple yet powerful methods—`localStorage` and `sessionStorage`

to store data on the client-side. These enable functionalities such as **remembering user login** and **maintaining cart contents** even after a page reload or tab closure.

#### 1. Persistent Login Using `localStorage`/`sessionStorage`

The login system allows users to securely enter their credentials to access their Artwave account. Once logged in, we store their login status and identity in the browser using:

- **localStorage** – Persistent even after the browser is closed and reopened.
- **sessionStorage** – Data lasts only for the session (until the tab or browser is closed).

##### Key Features:

- On successful login, store values like:
  - `userEmail`
  - `isLoggedIn`
- On each page load:
  - Check for these flags and auto-redirect to dashboard if still logged in.
- On logout:
  - Clear all stored values using `localStorage.removeItem()`.

##### Benefits:

- Saves time by avoiding repetitive logins.
- Delivers a more fluid browsing experience for the user.
- Ideal for small-to-medium web apps or academic prototypes like Artwave.

#### 2. Persistent Cart Using `localStorage`

For Artwave, where users browse and purchase music tracks or artworks, retaining cart data ensures a smoother user flow. This is done by saving the cart structure as a serialized JSON string in `localStorage`.

##### Implementation Features:

- Every cart update (add, remove, quantity change):
  - Convert the updated cart (array of objects) to JSON.
  - Store in `localStorage` with a key (e.g., `artwaveCart`).
- On page load:
  - Check for cart data in `localStorage`.
  - If found, parse and restore the cart items dynamically in the DOM.

##### Benefits:

- Prevents cart loss during accidental reloads or navigation.
- Encourages users to return and complete purchases.
- Improves user satisfaction on long browsing sessions.

## Use Cases Beyond the Syllabus

This experiment explores practical applications often used in real-world websites:

- Managing session state on the client side.
- Working with JavaScript objects, arrays, and JSON.
- Using browser storage for personalized, persistent user experiences.
- Creating e-commerce-like behavior without a backend.

## Codes:

login.js:-

```
document.getElementById("loginForm").addEventListener("submit", function (e) {
  e.preventDefault();
```

```
  const email = document.getElementById("loginEmail").value.trim();
  const password = document.getElementById("loginPassword").value;
```

```
  if (!email || !password) {
    alert("Please fill in all fields.");
    return;
  }
```

```
  // Simulated login check
  if (email === "user@artwave.com" && password === "123456") {
    localStorage.setItem("isLoggedIn", "true");
    localStorage.setItem("userEmail", email);
    alert("Login successful!");
    window.location.href = "home.html";
  } else {
    alert("Invalid credentials.");
  }
});
```

Logout funvntion:-

```
function logout() {
  localStorage.removeItem("isLoggedIn");
  localStorage.removeItem("userEmail");
  alert("You have been logged out.");
  window.location.href = "login.html";
}
```

Cart.js:

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

```
function addToCart(id, name, price) {
```

```

const existing = cart.find(item => item.id === id);
if (existing) {
  existing.quantity += 1;
} else {
  cart.push({ id, name, price, quantity: 1 });
}
saveCart();
renderCart();
}

```

```

function removeFromCart(id) {
  cart = cart.filter(item => item.id !== id);
  saveCart();
  renderCart();
}

```

```

function updateQuantity(id, qty) {
  const item = cart.find(i => i.id === id);
  if (item && qty > 0) {
    item.quantity = parseInt(qty);
    saveCart();
    renderCart();
  }
}

```

```

function saveCart() {
  localStorage.setItem("artwaveCart", JSON.stringify(cart));
}

```

```

function renderCart() {
  const container = document.getElementById("cartItems");
  container.innerHTML = "";
  let subtotal = 0;

```

```

  cart.forEach(item => {
    const total = item.price * item.quantity;
    subtotal += total;

```

```

  container.innerHTML += `
    <div class="cart-item">
      <span>${item.name}</span>
      <input type="number" value="${item.quantity}" onchange="updateQuantity('${item.id}', this.value)" />
      <span>₹${total}</span>
      <button onclick="removeFromCart('${item.id}')">Remove</button>

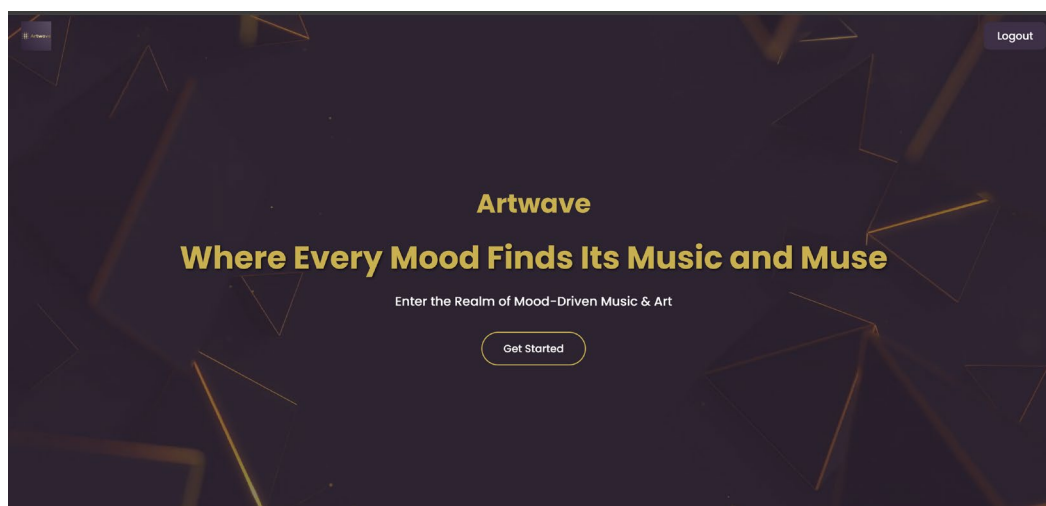
```

```
</div>
`
});

document.getElementById("cartSubtotal").innerText = `Subtotal: ₹${subtotal}`;
}

// Initialize cart display on page load
window.onload = renderCart;
```

output of homepage/index page:



### Conclusion:

This experiment applied JavaScript's Web Storage API to implement persistent login and cart functionality for the Artwave website. By storing user login status and cart data using localStorage, we ensured that users stay signed in and do not lose their cart items after page reloads or browser restarts. This improves convenience, enhances the overall user experience, and makes the site feel more professional and seamless.

## Experiment No.7

### PHP Theory: User Registration Handling for the Artwave Website

#### Introduction

User registration is a key feature of any interactive website, especially for platforms like **Artwave**, where users explore and purchase music or artwork. In this experiment, a **PHP script** is developed to handle the server-side logic of the registration process—capturing, validating, and securely storing user data in a MySQL database.

PHP is ideal for form handling due to its ability to securely process inputs, manage sessions, and interact with databases like MySQL. This enables Artwave to onboard users effectively while maintaining strong data integrity and security.

#### Core Components of PHP-Based Registration

##### 1. Form Handling with \$\_POST

Once the registration form is submitted, the PHP script retrieves data using `$_POST`. Fields like full name, email, and password are collected for further processing.

##### 2. Input Validation

To ensure accurate and safe data:

- All required fields (name, email, password) must be filled.
- Email format is validated using `filter_var()`.
- Passwords are checked for length and match (if confirmation is used).

##### 3. Password Hashing

For security, raw passwords are not stored directly. PHP's `password_hash()` function is used to securely encrypt the password before storing it in the database. This prevents password exposure even if the database is compromised.

##### 4. Database Interaction (MySQL)

Using **MySQLi** or **PDO**, the validated and sanitized user data is stored into a **users table** in the database. SQL **prepared statements** are used to prevent SQL injection attacks.

##### 5. Error Handling

The script checks for issues like:

- Empty required fields
- Invalid email format
- Duplicate email (already registered)
- Database connection or insertion failures

Clear and user-friendly error messages are displayed accordingly.

##### 6. User Feedback and Redirection

Upon successful registration, the user:

- Receives a success message (Registration successful!)
- May be redirected to the **login page** to sign in and access personalized features

#### Security Considerations

- All inputs are sanitized using `htmlspecialchars()` or similar methods to avoid XSS attacks.
- Passwords are hashed using `password_hash()` (not stored in plain text).
- Prepared statements are used to prevent SQL injection.

## Result and Impact on Artwave

This functionality ensures that **users can register securely**, enabling:

- Personalized user experiences (saved tracks, artwork purchases)
- Cart persistence and history tracking
- Admin-managed user database for analysis or updates

Codes:-

Regiter.php:- <?php

// Start the session

session\_start();

// If the user is already logged in, redirect to the index page

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

header("Location: index.php");

exit(); // Stop further execution

}

include 'config.php'; // Database connection file

if (\$\_SERVER["REQUEST\_METHOD"] == "POST") {

\$email = \$\_POST["email"];

\$password = \$\_POST["password"];

\$confirmPassword = \$\_POST["confirmPassword"];

// Check if passwords match

if (\$password != \$confirmPassword) {

echo "<script>alert('Passwords do not match.');

</script>";

} else {

// Check if user already exists

\$stmt = \$conn->prepare("SELECT \* FROM users WHERE email = ?");

\$stmt->bind\_param("s", \$email);

\$stmt->execute();

\$result = \$stmt->get\_result();

if (\$result->num\_rows > 0) {

echo "<script>alert('Account with this email already exists.');

</script>";

} else {

// Create new user

\$hashedPassword = password\_hash(\$password, PASSWORD\_DEFAULT);

\$stmt = \$conn->prepare("INSERT INTO users (email, password) VALUES (?, ?)");

\$stmt->bind\_param("ss", \$email, \$hashedPassword);

if (\$stmt->execute()) {

echo "<script>alert('Registration successful!'); window.location.href='login.php';</script>";



```

    } else {
        echo "<script>alert('Something went wrong. Please try again.');

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Register - Artwave</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            display: flex;
            flex-direction: column;
            align-items: center;
            min-height: 100vh;
        }

        .navbar {
            background-color: #422F48;
            padding: 15px;
            width: 100%;
            text-align: center;
        }

        .navbar a {
            color: white;
            margin: 0 4px;
            text-decoration: none;
            font-weight: 500;
        }
    </style>

```

```
.register-container {  
  background: white;  
  margin-top: 60px;  
  padding: 40px 30px;  
  border-radius: 12px;  
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
  max-width: 400px;  
  width: 90%;  
  text-align: center;  
}
```

```
.register-container h2 {  
  color: #422F48;  
  margin-bottom: 20px;  
}
```

```
.register-container input {  
  width: 100%;  
  padding: 10px;  
  margin: 10px 0;  
  border: 1px solid #ccc;  
  border-radius: 8px;  
  font-size: 14px;  
}
```

```
.register-container button {  
  background-color: #422F48;  
  color: white;  
  border: none;  
  padding: 12px 20px;  
  border-radius: 8px;  
  font-size: 16px;  
  cursor: pointer;  
  margin-top: 10px;  
  width: 100%;  
}
```

```
.register-container button:hover {  
  background-color: #341f39;  
}
```

```
footer {  
  margin-top: auto;  
  padding: 20px;
```

```
text-align: center;
background-color: #422F48;
color: white;
width: 100%;
}
</style>
</head>
<body>
  <!-- Logo -->
  <div class="logo">
    
  </div>

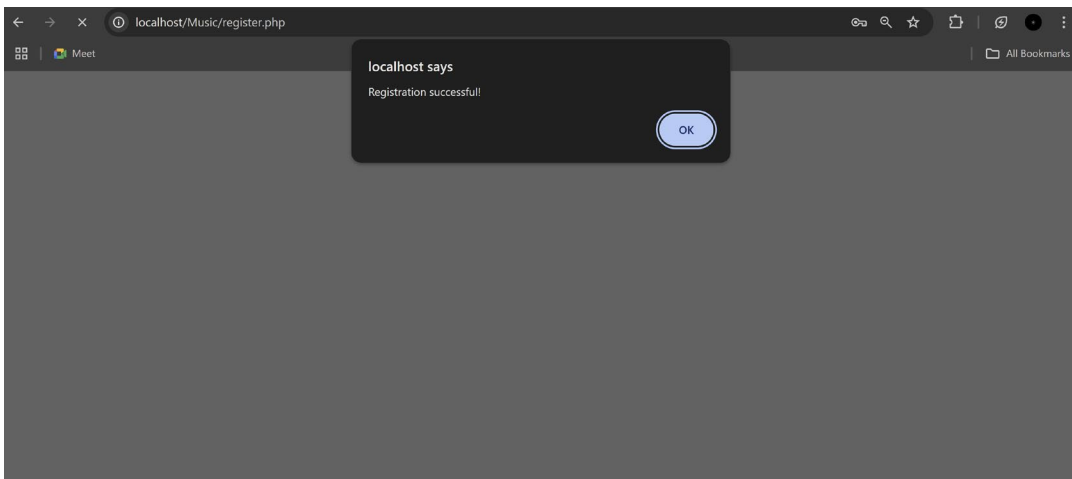
  <!-- Navbar -->
  <nav class="navbar">
    <a href="index.php">Home</a>
    <a href="menu.php">Menu</a>
    <a href="about.html">About</a>
    <a href="contact.php">Contact</a>
    <a href="login.php">Login</a>
    <a href="register.php">Register</a>
    <a href="cart.php">Cart</a>
  </nav>

  <!-- Registration Form -->
  <div class="register-container">
    <h2>Create an Account</h2>
    <form id="registerForm" method="POST" action="register.php">
      <input type="email" name="email" placeholder="Email" required />
      <input type="password" name="password" placeholder="Password" required />
      <input type="password" name="confirmPassword" placeholder="Confirm Password" required />
      <button type="submit">Register</button>
    </form>
  </div>

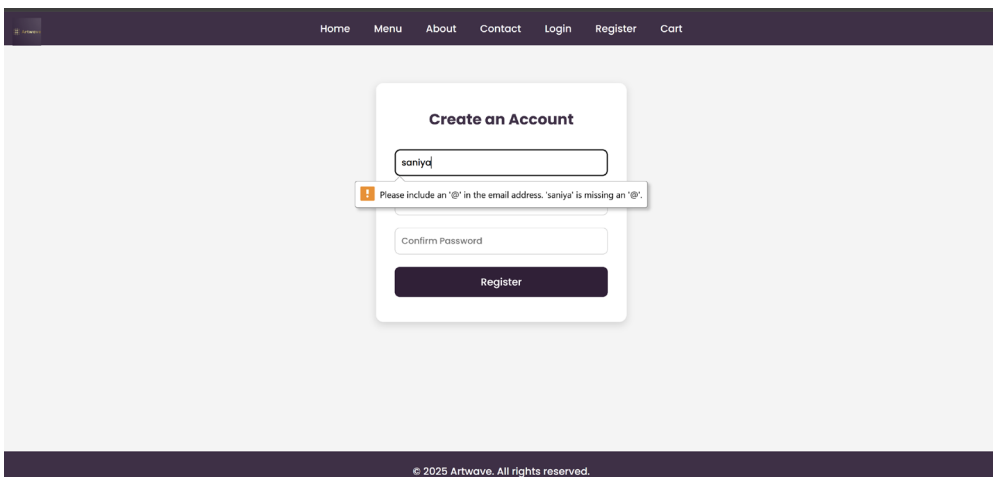
  <!-- Footer -->
  <footer>
    <p>&copy; 2025 Artwave. All rights reserved. <br> Designed by Saniya Mallick</p>
  </footer>

</body>
</html>
```

### Output of register successful attempt:-



### Validation error :-



### Congif.php:

```
<?php
```

```
$host = 'localhost';
```

```
$username = 'root';
```

```
$password = '';
```

```
$dbname = 'music_db';
```

```
$port = 3307;
```

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

```
if ($conn->connect_error) {
```

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

```
}
```

```
?>
```

Music.db.sql:-

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3307:3307
-- Generation Time: May 15, 2025 at 11:21 PM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.2.12
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `music_db`
--
```

```
-- -----
```

```
--
-- Table structure for table `cart`
--
```

```
CREATE TABLE `cart` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `track_id` varchar(255) NOT NULL,
  `track_name` varchar(255) NOT NULL,
  `track_url` varchar(255) NOT NULL,
  `added_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `quantity` int(11) NOT NULL DEFAULT 1,
  `type` enum('music','artwork') NOT NULL DEFAULT 'music',
```

```

`price` decimal(10,2) NOT NULL DEFAULT 0.00
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```
-----
```

```

--
-- Table structure for table `contacts`
--

```

```

CREATE TABLE `contacts` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `message` text NOT NULL,
  `submitted_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```
-----
```

```

--
-- Table structure for table `users`
--

```

```

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `users`
--

```

```

INSERT INTO `users` (`id`, `username`, `email`, `password`) VALUES
(1, ' ', 'abc@gmail.com', '$2y$10$UqC5kcDrAxX/VMp2ONDrtd6iVX6uHOBpzk0D2aSZ6WaJXF1lOURS');

```

```

--
-- Indexes for dumped tables
--

```

```
--  
-- Indexes for table `cart`  
--  
ALTER TABLE `cart`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `user_id` (`user_id`);  
  
--  
-- Indexes for table `contacts`  
--  
ALTER TABLE `contacts`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- Indexes for table `users`  
--  
ALTER TABLE `users`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- AUTO_INCREMENT for dumped tables  
--  
  
--  
-- AUTO_INCREMENT for table `cart`  
--  
ALTER TABLE `cart`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=22;  
  
--  
-- AUTO_INCREMENT for table `contacts`  
--  
ALTER TABLE `contacts`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT for table `users`  
--  
ALTER TABLE `users`
```

```
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

```
--
```

```
-- Constraints for dumped tables
```

```
--
```

```
--
```

```
-- Constraints for table `cart`
```

```
--
```

```
ALTER TABLE `cart`
```

```
ADD CONSTRAINT `cart_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;
```

```
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Login.php:-

```
<?php
```

```
// Start the session
```

```
session_start();
```

```
include 'config.php'; // Ensure database connection file is included
```

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

```
    $email = $_POST["email"];
```

```
    $password = $_POST["password"];
```

```
    // Prepare SQL query to fetch user from database
```

```
    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
```

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

```
    $stmt->execute();
```

```
    $result = $stmt->get_result();
```

```
    if ($result->num_rows > 0) {
```

```
        // User exists, fetch the user
```

```
        $user = $result->fetch_assoc();
```

```
        // Verify password
```

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



```

// Password is correct, start session and redirect
$_SESSION['user_id'] = $user['id'];
$_SESSION['email'] = $user['email'];

// Redirect to index.php after successful login
header("Location: index.php");
exit(); // Stop further execution after redirection
} else {
    echo "<script>alert('Invalid email or password.');

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Login - Artwave</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />

```

```

    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">

```

```

<style>
body {
    font-family: 'Poppins', sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    min-height: 100vh;
}

```

```

.navbar {
    background-color: #422F48;

```

```
padding: 15px;  
width: 100%;  
text-align: center;  
}
```

```
.navbar a {  
  color: white;  
  margin: 0 4px;  
  text-decoration: none;  
  font-weight: 500;  
}
```

```
.login-container {  
  background: white;  
  margin-top: 60px;  
  padding: 40px 30px;  
  border-radius: 12px;  
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
  max-width: 400px;  
  width: 90%;  
  text-align: center;  
}
```

```
.login-container h2 {  
  color: #422F48;  
  margin-bottom: 20px;  
}
```

```
.login-container input {  
  width: 100%;  
  padding: 10px;  
  margin: 10px 0;  
  border: 1px solid #ccc;  
  border-radius: 8px;  
  font-size: 14px;  
}
```

```
.login-container button {  
  background-color: #422F48;  
  color: white;
```

```
border: none;
padding: 12px 20px;
border-radius: 8px;
font-size: 16px;
cursor: pointer;
margin-top: 10px;
width: 100%;
}
```

```
.login-container button:hover {
  background-color: #341f39;
}
```

```
footer {
  margin-top: auto;
  padding: 20px;
  text-align: center;
  background-color: #422F48;
  color: white;
  width: 100%;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <!-- Logo -->
```

```
  <div class="logo">
```

```
    
```

```
  </div>
```

```
  <!-- Navbar -->
```

```
  <nav class="navbar">
```

```
    <a href="index.php">Home</a>
```

```
    <a href="menu.php">Menu</a>
```

```
    <a href="about.php">About</a>
```

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

```
    <a href="login.php">Login</a>
```

```
    <a href="register.php">Register</a>
```

```
    <a href="cart.php">Cart</a>
```

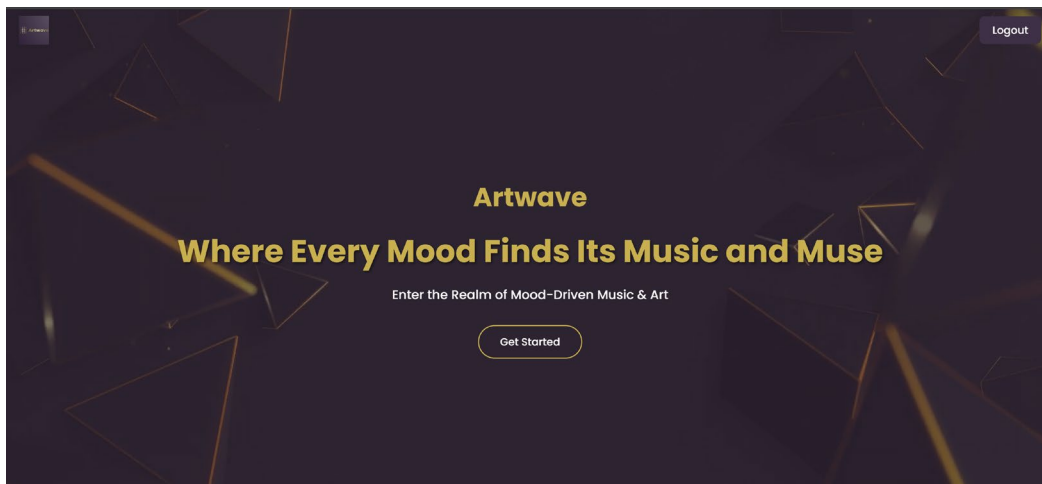
```
  </nav>
```

```
<!-- Login Form -->
<div class="login-container">
  <h2>Login to Artwave</h2>
  <form method="POST" action="login.php">
    <input type="email" name="email" placeholder="Email" required />
    <input type="password" name="password" placeholder="Password" required />
    <button type="submit">Login</button>
  </form>
</div>

<!-- Footer -->
<footer>
  <p>&copy; 2025 Artwave. All rights reserved. <br> Designed by Saniya Mallick</p>
</footer>

</body>
</html>
```

### Output of after successful attempt to login page :-



### Conclusion:-

In this experiment, we successfully implemented a user registration system using PHP. The registration form collects user information, which is then validated and securely stored in the database with hashed passwords to ensure security. Proper error handling provides users with immediate feedback on form submission, enhancing the user experience. This foundational functionality is essential for managing user accounts and personalizing interactions on the Artwave platform. The experiment highlights the importance of server-side scripting in handling data securely and reliably in web applications.

## Experiment No.8

### PHP Theory: User Login System for Artwave Website

- A.** Develop a PHP script to handle user login for the Artwave website. The script should accept input from users for their login credentials (email and password).
- B.** Provide feedback to the user upon successful login, either through a confirmation message or by redirecting to a welcome or home 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, such as displaying a personalized welcome message with the user's name.

### Theory: PHP Login System

A user login system is a critical feature for e-commerce websites like Artwave, enabling secure access to personalized content such as shopping carts, order history, and user profiles.

Key components of a PHP login system include:

- **Form Handling:** Capture login credentials (email and password) submitted via POST.
- **Input Validation:** Ensure both email and password fields are filled and formatted correctly.
- **Credential Verification:** Compare the submitted credentials against stored user data in a MySQL database.
- **Password Security:** Passwords are stored as hashes during registration using `password_hash()`. During login, `password_verify()` is used to securely compare the entered password with the stored hash.
- **Session Management:** Upon successful login, PHP sessions maintain the user's authenticated state across pages.
- **User Feedback:** Provide clear confirmation on login success and appropriate error messages on failure.

### Security Aspects

- **Password Hashing & Verification:** This prevents storing plain-text passwords, enhancing user data protection.
- **Session Handling:** Sessions ensure that only authenticated users can access restricted areas, improving security and user experience.

### Codes:-

#### Login.php:

```
<?php
// Start the session
session_start();
include 'config.php'; // Ensure database connection file is included
```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];

    // Prepare SQL query to fetch user from database
    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // User exists, fetch the user
        $user = $result->fetch_assoc();

        // Verify password
        if (password_verify($password, $user['password'])) {
            // Password is correct, start session and redirect
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['email'] = $user['email'];

            // Redirect to index.php after successful login
            header("Location: index.php");
            exit(); // Stop further execution after redirection
        } else {
            echo "<script>alert('Invalid email or password.');

```

```
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">
<style>
body {
  font-family: 'Poppins', sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
  display: flex;
  flex-direction: column;
  align-items: center;
  min-height: 100vh;
}

.navbar {
  background-color: #422F48;
  padding: 15px;
  width: 100%;
  text-align: center;
}

.navbar a {
  color: white;
  margin: 0 4px;
  text-decoration: none;
  font-weight: 500;
}

.login-container {
  background: white;
  margin-top: 60px;
  padding: 40px 30px;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
  max-width: 400px;
  width: 90%;
  text-align: center;
}
```

```
.login-container h2 {  
  color: #422F48;  
  margin-bottom: 20px;  
}  
  
.login-container input {  
  width: 100%;  
  padding: 10px;  
  margin: 10px 0;  
  border: 1px solid #ccc;  
  border-radius: 8px;  
  font-size: 14px;  
}  
  
.login-container button {  
  background-color: #422F48;  
  color: white;  
  border: none;  
  padding: 12px 20px;  
  border-radius: 8px;  
  font-size: 16px;  
  cursor: pointer;  
  margin-top: 10px;  
  width: 100%;  
}  
  
.login-container button:hover {  
  background-color: #341f39;  
}  
  
footer {  
  margin-top: auto;  
  padding: 20px;  
  text-align: center;  
  background-color: #422F48;  
  color: white;  
  width: 100%;  
}  
</style>  
</head>
```



```
<body>
  <!-- Logo -->
  <div class="logo">
    
  </div>

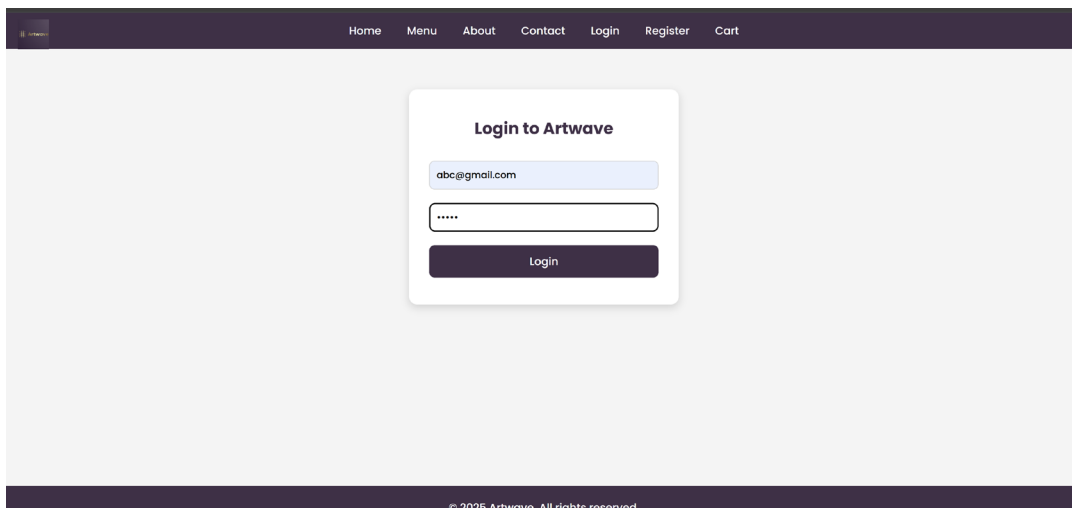
  <!-- Navbar -->
  <nav class="navbar">
    <a href="index.php">Home</a>
    <a href="menu.php">Menu</a>
    <a href="about.php">About</a>
    <a href="contact.php">Contact</a>
    <a href="login.php">Login</a>
    <a href="register.php">Register</a>
    <a href="cart.php">Cart</a>
  </nav>

  <!-- Login Form -->
  <div class="login-container">
    <h2>Login to Artwave</h2>
    <form method="POST" action="login.php">
      <input type="email" name="email" placeholder="Email" required />
      <input type="password" name="password" placeholder="Password" required />
      <button type="submit">Login</button>
    </form>
  </div>

  <!-- Footer -->
  <footer>
    <p>&copy; 2025 Artwave. All rights reserved. <br> Designed by Saniya Mallick</p>
  </footer>

</body>
</html>
```

Output of login page :-



### Logout.php:

```
<?php
session_start();      // Start the session
session_unset();      // Unset all session variables
session_destroy();    // Destroy the session
```

```
// Redirect to login or homepage
header("Location: login.php");
exit();
```

### index.php:

```
<?php
session_start();

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
?>
```

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

```

<title>Artwave</title>
<link rel="stylesheet" href="style.css"/>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap"
rel="stylesheet">
</head>
<body>

<!-- Logo -->
<div class="logo">
  
</div>

<!-- Video Background -->
<video autoplay muted loop playsinline class="background-video">
  <source src="videobg.mp4" type="video/mp4" />
  Your browser does not support the video tag.
</video>

<!-- Overlay -->
<div class="overlay"></div>

<!-- Hero Content -->
<section class="hero-content">
  <h2 class="site-name">Artwave</h2>
  <h1>Where Every Mood Finds Its Music and Muse</h1>
  <p>Enter the Realm of Mood-Driven Music & Art</p>

  <!-- Get Started Button -->
  <a href="menu.php" class="get-started-btn">Get Started</a>
</section>

<!-- Logout Button -->
<div class="logout-container">
  <a href="logout.php" class="logout-btn">Logout</a>
</div>

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

```

**Conclusion:**

In this experiment, we successfully implemented a secure user login system for the Artwave website using PHP. By validating user credentials against the database and using password hashing and verification, we ensured safe authentication. Session management was used to maintain user login states, enabling personalized experiences and protected content access. Proper error handling enhanced usability by providing clear feedback on login successes or failures. This login system forms a crucial foundation for user interaction and personalized features in the Artwave platform.

**Experiment No.9****PHP Shopping Cart System for Artwave (Music and Artwork Platform)****A. Session-Based Shopping Cart (Without MySQL)**

This approach uses PHP sessions to temporarily hold the cart data while the user is browsing your Artwave website. It is ideal for quick development and does not require database interaction.

Key Features:

- Cart data is saved in \$\_SESSION.
- Data persists only during the user's current browsing session.
- No login is required to use the cart.
- Cart contents are lost when the session expires or the browser closes.

Operations Supported:

- Add to Cart: Store track/artwork ID, name, quantity, and price in the session.
- View Cart: Show the cart contents saved in the session.
- Remove from Cart: Delete items by their ID from the session array.

Advantages:

- Simple and fast to implement.
- No database overhead.

Limitations:

- Cart data is lost after the session ends.
- Not suitable for persistent or multi-device cart management.

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

For a professional and persistent cart system, the cart data is stored in a MySQL database linked to the logged-in user's account. This approach suits the Artwave platform's need to save users' music tracks and artwork selections persistently.

Key Features:

- Each user's cart is linked to their unique user ID.
- Cart contents are stored in database tables (cart and cart\_items).
- Requires user login to associate the cart with a user.

- Cart persists across sessions, devices, and logins.

#### Operations Supported:

- Add to Cart: Insert or update the selected tracks/artwork in the database cart tables.
- View Cart: Fetch all cart items for the logged-in user from the database.
- Remove from Cart: Delete items from the database cart tables by item ID.

#### Advantages:

- Persistent, user-specific cart experience.
- Works across multiple sessions and devices.
- Enables tracking and analytics of user cart behavior.

#### Limitations:

- Requires database setup and connection management.
- Requires a secure login system to associate carts with users.

#### Code:-

##### Cart.php:

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_db";

$conn = new mysqli($servername, $username, $password, $dbname, 3307);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$user_id = $_SESSION['user_id'];

// Handle item removal
if (isset($_POST['remove_track_name']) && isset($_POST['remove_type'])) {
    $remove_name = $conn->real_escape_string($_POST['remove_track_name']);
    $remove_type = $conn->real_escape_string($_POST['remove_type']);
    $stmt_remove = $conn->prepare("DELETE FROM cart WHERE user_id = ? AND track_name = ? AND type = ?
LIMIT 1");
    $stmt_remove->bind_param("iss", $user_id, $remove_name, $remove_type);
    $stmt_remove->execute();
}
```

```

$stmt_remove->close();
header("Location: cart.php");
exit();
}

// Fetch cart items with price
$stmt = $conn->prepare("SELECT track_name, track_url, quantity, type, price FROM cart WHERE user_id = ?");
$stmt->bind_param("i", $user_id);
$stmt->execute();
$stmt->store_result();
$stmt->bind_result($name, $url, $quantity, $type, $price);

$music_items = [];
$artwork_items = [];
$subtotal = 0;

while ($stmt->fetch()) {
    $item_total = $quantity * $price;
    $subtotal += $item_total;
    $item = ['name' => $name, 'url' => $url, 'quantity' => $quantity, 'price' => $price];
    if ($type === 'music') {
        $music_items[] = $item;
    } elseif ($type === 'artwork') {
        $artwork_items[] = $item;
    }
}

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

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<title>Your Cart - Artwave</title>
<link rel="stylesheet" href="style.css" />
<style>
body {
    font-family: 'Poppins', sans-serif;
    background-color: #ffffff;
    color: #422F48;
    margin: 0;

```

```
padding: 0;  
text-align: center;  
}
```

```
.cart-container {  
  margin-top: 60px;  
  padding: 30px 20px;  
}
```

```
.navbar {  
  display: flex;  
  justify-content: center;  
  background-color: #412E46;  
  padding: 18px 0;  
  gap: 30px;  
  position: sticky;  
  top: 0;  
  z-index: 999;  
}
```

```
.navbar a {  
  text-decoration: none;  
  color: #fcf3ff;  
  font-weight: 500;  
  font-size: 18px;  
  transition: color 0.3s ease;  
}
```

```
.navbar a:hover {  
  color: #7b4b94;  
}
```

```
.cart-header {  
  font-size: 2.5rem;  
  color: #422F48;  
}
```

```
.cart-items {  
  display: flex;  
  justify-content: center;  
  gap: 20px;  
  margin-top: 30px;  
  flex-wrap: wrap;  
  padding: 0 20px;
```

```
max-height: 500px;  
overflow-y: auto;  
}
```

```
.cart-items::-webkit-scrollbar {  
  width: 10px;  
}
```

```
.cart-items::-webkit-scrollbar-track {  
  background: #f1f1f1;  
  border-radius: 10px;  
}
```

```
.cart-items::-webkit-scrollbar-thumb {  
  background-color: #422F48;  
  border-radius: 10px;  
}
```

```
.cart-item {  
  background-color: #422F48;  
  border-radius: 20px;  
  padding: 20px;  
  width: 300px;  
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  color: white;  
  position: relative;  
}
```

```
.cart-item-title {  
  font-size: 1.2rem;  
  margin-bottom: 10px;  
}
```

```
audio, img.artwork {  
  width: 100%;  
  margin-top: 10px;  
  border-radius: 10px;  
}
```

```
.remove-btn {  
  position: absolute;
```



```
    top: 15px;
    right: 15px;
    background: #ff4c4c;
    border: none;
    border-radius: 50%;
    width: 28px;
    height: 28px;
    color: white;
    font-weight: bold;
    cursor: pointer;
    line-height: 28px;
    font-size: 16px;
  }

.checkout-btn {
  padding: 12px 25px;
  background-color: #422F48;
  color: rgb(255, 255, 255);
  border: none;
  border-radius: 8px;
  cursor: pointer;
  margin-top: 40px;
  font-size: 1rem;
}

.checkout-btn:hover {
  background-color: #311f36;
}

footer {
  margin-top: 60px;
  padding: 20px;
  font-size: 0.9rem;
  color: #888;
}
</style>
</head>
<body>

<!-- Logo -->
<div class="logo">
  
</div>
```

```

<!-- Navigation Bar -->
<nav class="navbar">
  <a href="index.php">Home</a>
  <a href="menu.php">Menu</a>
  <a href="about.php">About</a>
  <a href="contact.php">Contact</a>
  <a href="register.php">Register</a>
  <a href="login.php">Login</a>
  <a href="cart.php">Cart</a>
</nav>

<div class="cart-container">
  <h1 class="cart-header">Your Cart</h1>

  <div class="cart-items">
    <?php
      if (empty($music_items) && empty($artwork_items)) {
        echo "<p>Your cart is empty!</p>";
      }

      foreach ($music_items as $track) {
        echo "<div class='cart-item'>";
          echo " <div class='cart-item-title'>" . htmlspecialchars($track['name']) . " (x" .
htmlspecialchars($track['quantity']) . ")</div>";
          echo "<p>Price: ₹" . number_format($track['price'], 2) . "</p>";
          echo "<audio controls><source src='" . htmlspecialchars($track['url']) . "' type='audio/mpeg'></audio>";
          echo "<form method='post' style='position:absolute; top:0; right:0;'>";
            echo "<input type='hidden' name='remove_track_name' value='" . htmlspecialchars($track['name'],
ENT_QUOTES) . "'>";
            echo "<input type='hidden' name='remove_type' value='music'>";
            echo "<button type='submit' class='remove-btn' title='Remove item'>&times;</button>";
          echo "</form>";
        echo "</div>";
      }

      foreach ($artwork_items as $art) {
        echo "<div class='cart-item'>";
          echo " <div class='cart-item-title'>" . htmlspecialchars($art['name']) . " (x" .
htmlspecialchars($art['quantity']) . ")</div>";
          echo "<p>Price: ₹" . number_format($art['price'], 2) . "</p>";
          echo "<img class='artwork' src='" . htmlspecialchars($art['url']) . "' alt='Artwork'>";
          echo "<form method='post' style='position:absolute; top:0; right:0;'>";
            echo "<input type='hidden' name='remove_track_name' value='" . htmlspecialchars($art['name'],
ENT_QUOTES) . "'>";

```

```
        echo "<input type='hidden' name='remove_type' value='artwork'>";
        echo "<button type='submit' class='remove-btn' title='Remove item'>&times;</button>";
        echo "</form>";
        echo "</div>";
    }
?>
</div>

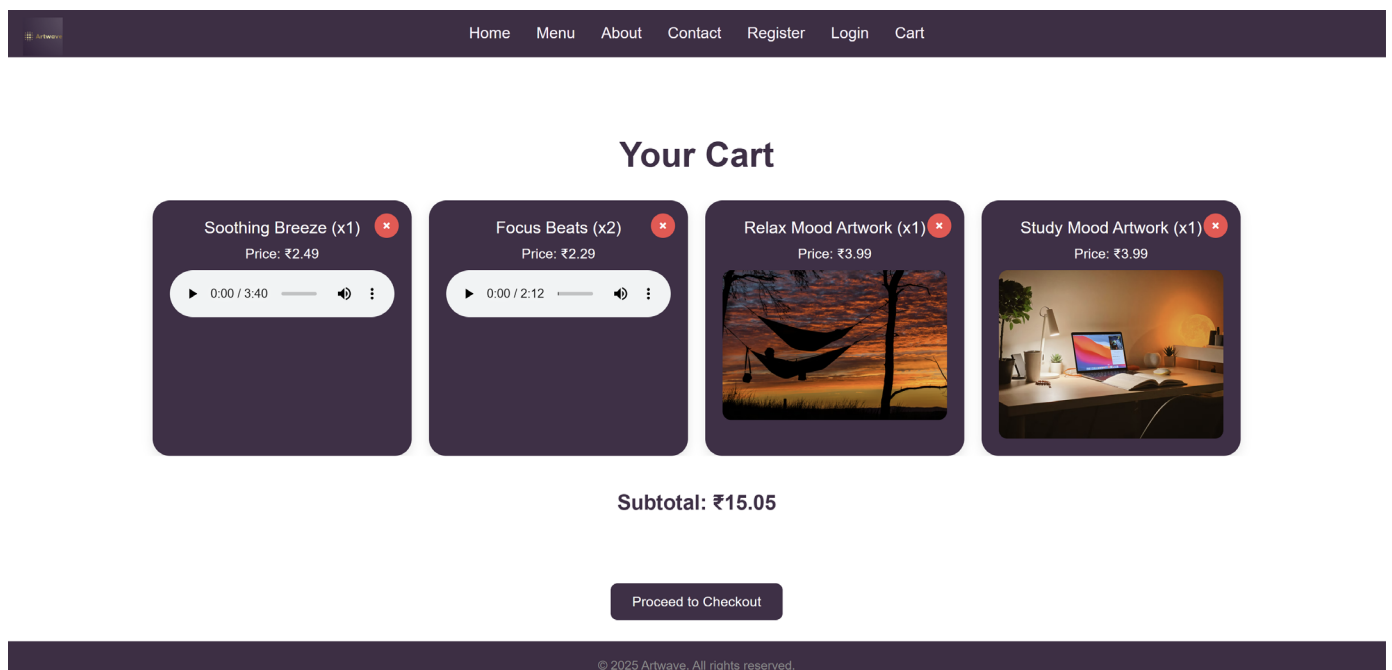
<?php if ($subtotal > 0): ?>
    <h2 style="margin-top: 40px; color: #422F48;">Subtotal: ₹<?php echo number_format($subtotal, 2);
?></h2>
<?php endif; ?>

<form action="checkout.php" method="post" style="margin-top:40px;">
    <button type="submit" class="checkout-btn">Proceed to Checkout</button>
</form>
</div>

<footer>
    &copy; 2025 Artwave. All rights reserved.
</footer>

</body>
</html>
```

### Output of cart page :



Checkout.php:-

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_db";

$conn = new mysqli($servername, $username, $password, $dbname, 3307);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$user_id = $_SESSION['user_id'];

// Fetch cart items including price
$stmt = $conn->prepare("SELECT track_name, quantity, price FROM cart WHERE user_id = ?");
$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();

$cart_items = $result->fetch_all(MYSQLI_ASSOC);
$stmt->close();

if (empty($cart_items)) {
    echo "<p>Your cart is empty. Please add items before checkout.</p>";
    echo "<p><a href='cart.php'>Back to Cart</a></p>";
    exit();
}

$total_price = 0;
foreach ($cart_items as $item) {
    $total_price += $item['quantity'] * $item['price'];
}
```

```
// Simulate order success by clearing the cart
$del_stmt = $conn->prepare("DELETE FROM cart WHERE user_id = ?");
$del_stmt->bind_param("i", $user_id);
$del_stmt->execute();
$del_stmt->close();

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Checkout Complete - Artwave</title>
  <style>
    body {
      font-family: 'Poppins', sans-serif;
      text-align: center;
      padding: 40px;
      background-color: #fff;
      color: #422F48;
    }
    .message {
      font-size: 1.8rem;
      margin-bottom: 20px;
    }
    .item-list {
      max-width: 500px;
      margin: 0 auto 30px;
      text-align: left;
    }
    .item-list ul {
      list-style-type: none;
      padding: 0;
    }
    .item-list li {
      padding: 8px 0;
      border-bottom: 1px solid #ddd;
    }
  </style>
```

```

.total {
    font-size: 1.5rem;
    margin-top: 20px;
    color: #311f36;
}
a.button {
    text-decoration: none;
    padding: 12px 30px;
    background-color: #422F48;
    color: white;
    border-radius: 8px;
    font-size: 1rem;
}
a.button:hover {
    background-color: #311f36;
}
</style>
</head>
<body>
    <div class="message">
        <p>Thank you for your purchase! Your order has been placed successfully.</p>
    </div>

    <div class="item-list">
        <h3>Order Summary:</h3>
        <ul>
            <?php foreach ($cart_items as $item): ?>
                <li>
                    <?php echo htmlspecialchars($item['track_name']); ?> (x<?php echo $item['quantity']; ?>)
                    - ₹<?php echo number_format($item['quantity'] * $item['price'], 2); ?>
                </li>
            <?php endforeach; ?>
        </ul>

        <div class="total">
            Total Paid: ₹<?php echo number_format($total_price, 2); ?>
        </div>
    </div>

    <a href="index.php" class="button">Back to Home</a>

```

&lt;/body&gt;

&lt;/html&gt;

Output of checkout page :

Thank you for your purchase! Your order has been placed successfully.

**Order Summary:**

Soothing Breeze (x1) - ₹2.49

Relax Mood Artwork (x1) - ₹3.99

Focus Beats (x2) - ₹4.58

Study Mood Artwork (x1) - ₹3.99

Total Paid: ₹15.05

[Back to Home](#)**Remove from cart page :-** <?php

session\_start();

include 'config.php';

// Get the cart\_id to remove

\$data = json\_decode(file\_get\_contents("php://input"), true);

\$cart\_id = \$data['cart\_id'];

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

\$user\_id = \$\_SESSION['user\_id'];

\$stmt = \$conn-&gt;prepare("DELETE FROM cart WHERE id = ? AND user\_id = ?");

\$stmt-&gt;bind\_param("ii", \$cart\_id, \$user\_id);

\$stmt-&gt;execute();

echo json\_encode(['success' =&gt; true]);

} else {

echo json\_encode(['success' =&gt; false, 'message' =&gt; 'User not logged in']);

}

?&gt;

**Update in cart quantity code :-**

```
<?php
session_start();

// Check if update quantity form is submitted
if (isset($_POST['update_cart'])) {
    $productId = $_POST['product_id'];
    $newQuantity = intval($_POST['quantity']);

    // Validate quantity (minimum 1)
    if ($newQuantity < 1) {
        $newQuantity = 1;
    }

    // Update quantity in session cart array
    if (isset($_SESSION['cart'][$productId])) {
        $_SESSION['cart'][$productId]['quantity'] = $newQuantity;
    }

    // Redirect to cart page to avoid resubmission
    header("Location: cart.php");
    exit();
}
?>
```

**Conclusion:**

This PHP script enables users to update the quantity of items in their shopping cart by modifying the session data. It ensures that the quantity is always at least one, preventing invalid entries. By storing the cart in the session, the cart data persists during the user's browsing session. The script also redirects back to the cart page after updating to prevent duplicate form submissions and provide a seamless user experience. This approach is simple and effective for managing cart quantities in a session-based shopping cart system.



## Experiment 10

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

### Theory: PHP Checkout Process for Artwave

The checkout process is the final and most critical step in any e-commerce platform, including Artwave—a website offering music tracks and artworks for sale. It converts the user's cart into an official order by capturing necessary details such as billing, shipping, and payment information, and then recording it securely in the database for further processing.

Since Artwave deals with unique digital and physical products, a robust checkout system ensures accurate order handling, stock management, and customer satisfaction.

### Two Approaches to Checkout

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

In this simpler approach:

- Cart data is stored temporarily in PHP sessions (\$\_SESSION['cart']).
- On checkout, a confirmation message is displayed to the user.
- Suitable for quick demos or prototype versions of Artwave.

#### Workflow:

1. Retrieve cart details from session storage.
2. Validate essential checkout fields such as user name, email, and contact information.
3. Show a success or error message based on input validation.
4. Clear the cart session data after successful checkout.

#### Advantages:

- Easy and fast to implement.
- Minimal setup and no database dependency.

#### Limitations:

- Data is lost after session ends; no persistent order records.
- Not suitable for production or real-world use.
- No history or analytics of orders.

#### B. MySQL-Based Checkout System

This is the professional and scalable checkout solution for Artwave:

- Order and user data are stored in MySQL database tables.

- Supports user login integration, persistent order records, and backend order management.
- Enables order tracking, analytics, and inventory control.

### Workflow:

1. Confirm user login/session status.
2. Retrieve cart items either from session or database.
3. Validate checkout inputs such as shipping address, payment details, and contact information.
4. Insert the order data into an orders table and item details into an order\_items table.
5. Display feedback (success or failure) to the user.
6. Clear the cart session or database entries upon successful order placement.

### Codes:-

#### checkout.php:

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_db";

$conn = new mysqli($servername, $username, $password, $dbname, 3307);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$user_id = $_SESSION['user_id'];

// Fetch cart items including price
$stmt = $conn->prepare("SELECT track_name, quantity, price FROM cart WHERE user_id = ?");
$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();

$cart_items = $result->fetch_all(MYSQLI_ASSOC);
$stmt->close();

if (empty($cart_items)) {
    echo "<p>Your cart is empty. Please add items before checkout.</p>";
}
```

```

    echo "<p><a href='cart.php'>Back to Cart</a></p>";
    exit();
}

$total_price = 0;
foreach ($cart_items as $item) {
    $total_price += $item['quantity'] * $item['price'];
}

// Simulate order success by clearing the cart
$del_stmt = $conn->prepare("DELETE FROM cart WHERE user_id = ?");
$del_stmt->bind_param("i", $user_id);
$del_stmt->execute();
$del_stmt->close();

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>Checkout Complete - Artwave</title>
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            text-align: center;
            padding: 40px;
            background-color: #fff;
            color: #422F48;
        }
        .message {
            font-size: 1.8rem;
            margin-bottom: 20px;
        }
        .item-list {
            max-width: 500px;
            margin: 0 auto 30px;
            text-align: left;
        }
        .item-list ul {
            list-style-type: none;
            padding: 0;
        }
    </style>

```

```

.item-list li {
    padding: 8px 0;
    border-bottom: 1px solid #ddd;
}
.total {
    font-size: 1.5rem;
    margin-top: 20px;
    color: #311f36;
}
a.button {
    text-decoration: none;
    padding: 12px 30px;
    background-color: #422F48;
    color: white;
    border-radius: 8px;
    font-size: 1rem;
}
a.button:hover {
    background-color: #311f36;
}
</style>
</head>
<body>
    <div class="message">
        <p>Thank you for your purchase! Your order has been placed successfully.</p>
    </div>

    <div class="item-list">
        <h3>Order Summary:</h3>
        <ul>
            <?php foreach ($cart_items as $item): ?>
                <li>
                    <?php echo htmlspecialchars($item['track_name']); ?> (x<?php echo $item['quantity']; ?>)
                    - ₹<?php echo number_format($item['quantity'] * $item['price'], 2); ?>
                </li>
            <?php endforeach; ?>
        </ul>

        <div class="total">
            Total Paid: ₹<?php echo number_format($total_price, 2); ?>
        </div>
    </div>

    <a href="index.php" class="button">Back to Home</a>

```

&lt;/body&gt;

&lt;/html&gt;

**music.db database:-**


The screenshot shows the phpMyAdmin interface for the 'music\_db' database. The left sidebar displays the database structure with 'music\_db' selected, showing tables: 'cart', 'contacts', and 'users'. The main panel shows the 'Structure' tab for 'music\_db'. It lists three tables: 'cart' (1 row, 32.0 KiB), 'contacts' (0 rows, 16.0 KiB), and 'users' (2 rows, 16.0 KiB). Below the table list, there is a 'Create new table' button and a form to create a new table with a name and number of columns (set to 4). The 'Console' tab is visible at the bottom.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> cart		1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> contacts		0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> users		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<b>3 tables</b>	<b>Sum</b>	<b>3</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>64.0 KiB</b>	<b>0 B</b>

**If theres no checkout products ;-**

The screenshot shows a web browser window at the URL 'localhost/Music/checkout.php'. The page displays a message: 'Your cart is empty. Please add items before checkout.' Below the message is a link labeled 'Back to Cart'.

If theres products in the cart:-



HomeMenuAboutContactRegisterLoginCart

## Your Cart

Rise and Shine (x1)✕

Price: ₹2.49

▶ 0:00 / 3:32 🔊 ⋮


High Energy (x1)✕

Price: ₹2.99

▶ 0:00 / 2:34 🔊 ⋮

Motivation Mood Artwork (x1)✕




Price: ₹4.49



**Subtotal: ₹9.97**

Proceed to Checkout

© 2025 Artwave. All rights reserved.

 |  Meet |  All Bookmarks

Thank you for your purchase! Your order has been placed successfully.

**Order Summary:**

Rise and Shine (x1) - ₹2.49

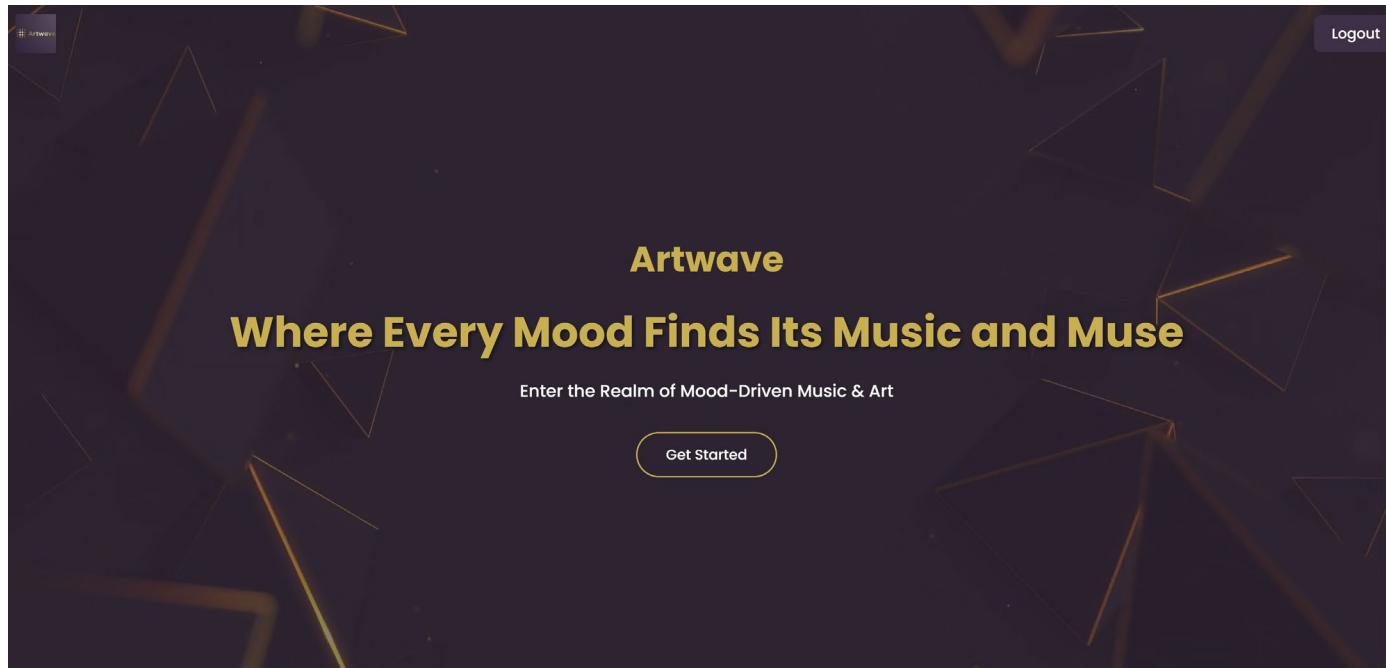
Motivation Mood Artwork (x1) - ₹4.49

High Energy (x1) - ₹2.99

**Total Paid: ₹9.97**

Back to Home

It goes back to the homepage :-



## Conclusion

In this final experiment, the checkout process for the Artwave website was successfully developed and implemented, marking a crucial milestone in building a fully functional e-commerce platform for music tracks and artworks. By integrating PHP with MySQL, the system enables users to securely finalize their purchases while ensuring order data is accurately recorded and maintained for future processing.

Two approaches were explored: a session-based checkout suitable for quick prototypes, and a robust MySQL-based checkout that supports persistent order management, user authentication, and data integrity. This comprehensive implementation not only enhances user experience by providing seamless order placement and feedback but also lays a strong foundation for scalable features such as order tracking, inventory control, and analytics.

Overall, this experiment completes the core e-commerce workflow on Artwave, transforming browsing and cart management into a streamlined purchase experience, essential for real-world online retail success.