INSTITUTE OF AERONAUTICAL ENGINEERING DUNDIGAL, HYDERABAD-500 043, TELANGANA, INDIA.

Front-end Web Development

COURSE CODE: ACSE04

TASK1

Title: ShopNest E-Commerce

By:

24951A6660 –

SOMEPALLI HASINI

From: CSE(AI/ML)

INSTITUTE OF AERONAUTICAL ENGINEERING
(AUTONOMOUS)
Dundigal - 500 043, Hyderabad, Telangana

TASK Self-Assessment Form

1	Name of the Student	Somepalli Hasini
2	Roll Number	24951A6660
3	Branch and Section	CSE(AI/ML)
4	Program	B. Tech
5	Course Name	Object-Oriented Programming
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	

EC	Profiles	(✓)
EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	(✓)
EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	(✓)
EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	(✓)
EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	(✓)
EC 5	Conceptualizes alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	(✓)
EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	(✓)
EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	(✓)
EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	(✓)
EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	(✓)
EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	(✓)

	EC	Profiles	(✓)
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	(✓)
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	(✓)
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	**Course Outcomes**	(✓)
	CO 1	Understand the fundamental concepts of sorting algorithms and their significance in data processing for large-scale applications.	(✓)
	CO 2	Analyze the limitations of traditional sorting algorithms when applied to big data environments.	(✓)
	CO 3	Design and implement hybrid sorting algorithms in Java that combine multiple techniques (e.g., merge-sort, quick-sort, and heap-sort) for improved performance.	(✓)
	CO 4	Evaluate the computational efficiency, time complexity, and scalability of various hybrid sorting approaches for large datasets..	(✓)
	CO 5	Apply Java-based parallel and distributed programming concepts (e.g., multithreading, Hadoop/MapReduce) to optimize hybrid sorting performance in big data contexts.	(✓)
	CO 6	Compare hybrid sorting techniques using performance metrics such as execution time, memory utilization, and throughput on different data sizes.	(✓)
	CO 7	Develop a mini-project or case study demonstrating the practical application of hybrid sorting in real-world big data scenarios such as data analytics, machine learning preprocessing, or cloud computing systems.	(✓)

9	Course ELRV Video Lectures Viewed	**Number of Videos**	**Viewing time in Hours**
10	Justify your understanding of WK1	Foundations for analysis and optimizing operations	
11	Justify your understanding of WK2 – WK9	Core to advanced concepts, tools, design, and ethics.	
12	How many WKs from WK2 to WK9 were implanted?	All 8 WKs from WK2 – WK9 are implemented in this and analysis	
	Mention them	WK2 to WK9	

[illegible]

Date:12/12/25

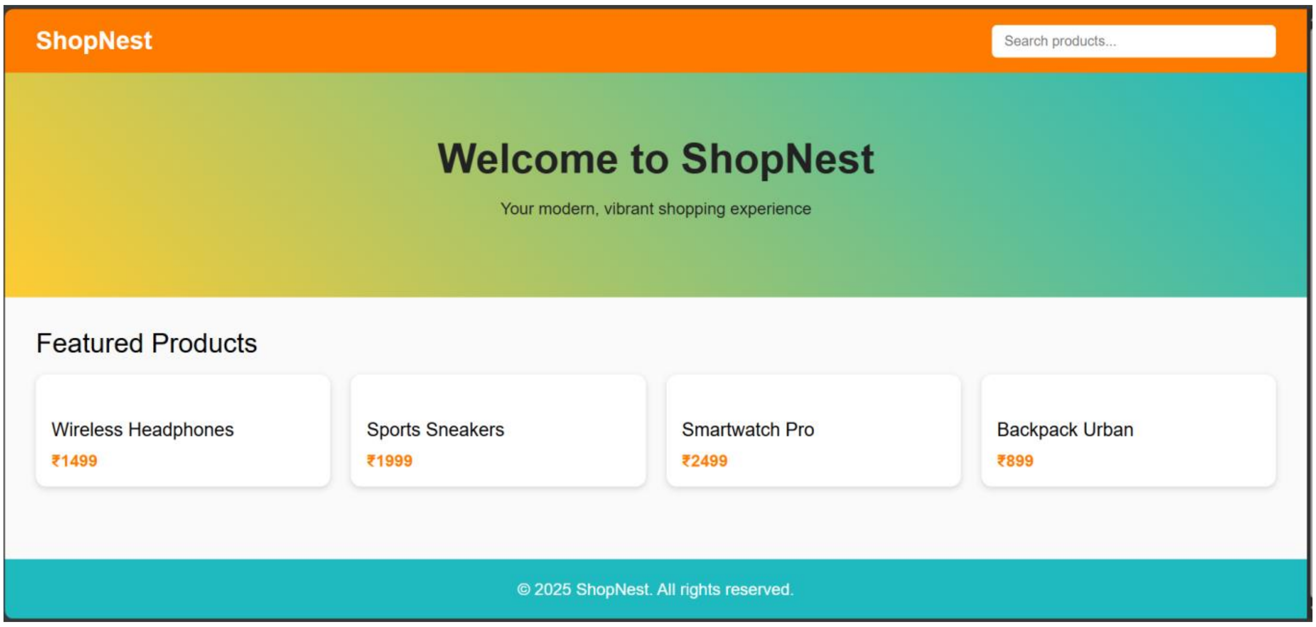
Signature of the Student

Project Report – ShopNest E-Commerce Frontend

1. Title and Domain

Title: *ShopNest – A Modern Responsive E-Commerce Frontend*
Domain: *Web Development / User Interface & Frontend Engineering*

ShopNest is a frontend-only web application developed using **HTML**, **CSS**, and **JavaScript**, designed to simulate a modern online shopping interface. The platform falls under the domain of **e-commerce web development**, focusing specifically on **UI/UX design**, **responsive layouts**, and **interactive elements** that enhance the user's browsing experience. The website uses a vibrant visual theme built around **yellow, orange, and teal tones** to give the interface a fresh and energetic aesthetic suitable for product-oriented platforms.



Welcome to ShopNest

Your modern, vibrant shopping experience

Featured Products

Wireless Headphones
₹1499

Sports Sneakers
₹1999

Smartwatch Pro
₹2499

Backpack Urban
₹899

© 2025 ShopNest. All rights reserved.

CODE:

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

    <style>
        /* ---------- GLOBAL THEME ---------- */
        :root {
            --yellow: #ffcc33;
            --orange: #ff7b00;
            --teal: #1fbabf;
            --dark: #222;
            --light: #fff;
        }

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

        /* ---------- NAVBAR ---------- */
        nav {
            background: var(--orange);
            padding: 15px 30px;
            display: flex;
            justify-content: space-between;
            align-items: center;
            color: var(--light);
        }

        nav .logo {
            font-size: 24px;
            font-weight: bold;
        }

        nav input {
            padding: 8px 12px;
            border-radius: 5px;
            border: none;
            width: 250px;
        }

        /* ---------- HERO / BANNER ---------- */
        .banner {
            background: linear-gradient(45deg, var(--yellow), var(--teal));
            color: var(--dark);
            padding: 60px 20px;
            text-align: center;
        }

        .banner h1 {
            margin: 0;
            font-size: 40px;
        }

        /* ---------- PRODUCT GRID ---------- */
        .products-section {
            padding: 30px;
        }

        .products-title {
            font-size: 26px;
            margin-bottom: 15px;
        }

        .products-grid {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
            gap: 20px;
        }

        .product-card {
            background: var(--light);
            border-radius: 10px;
            padding: 15px;
            box-shadow: 0 2px 6px rgba(0,0,0,0.15);
            transition: 0.3s;
        }

        .product-card:hover {
            transform: translateY(-6px);
        }

        .product-card img {
            width: 100%;
            border-radius: 10px;
        }

        .product-name {
            font-size: 18px;
            margin: 10px 0;
        }

        .product-price {
            color: var(--orange);
            font-weight: bold;
        }

        /* ---------- FOOTER ---------- */
        footer {
            text-align: center;
            padding: 20px;
            background: var(--teal);
            margin-top: 40px;
            color: var(--light);
        }
    </style>

</head>
<body>

    <!-- NAVBAR -->
    <nav>
        <div class="logo">ShopNest</div>
        <input type="text" id="search" placeholder="Search products...">
    </nav>

    <!-- HERO SECTION -->
    <section class="banner">
        <h1>Welcome to ShopNest</h1>
        <p>Your modern, vibrant shopping experience</p>
    </section>

    <!-- PRODUCTS -->
    <section class="products-section">
        <div class="products-title">Featured Products</div>

        <div class="products-grid" id="productList">
            <!-- JS renders products here -->
        </div>
    </section>

    <footer>© 2025 ShopNest. All rights reserved.</footer>

    <script>
        /* ---------- SIMPLE PRODUCT DATA ---------- */
        const products = [
            { name: "Wireless Headphones", price: "₹1499", img:
"https://via.placeholder.com/250x180" },
            { name: "Sports Sneakers", price: "₹1999", img:
"https://via.placeholder.com/250x180" },
            { name: "Smartwatch Pro", price: "₹2499", img:
```

```
"https://via.placeholder.com/250x180" },
            { name: "Backpack Urban", price: "₹899", img:
"https://via.placeholder.com/250x180" },
        ];

        const container = document.getElementById("productList");

        function renderProducts(items) {
            container.innerHTML = "";
            items.forEach(p => {
                container.innerHTML += `
                    <div class="product-card">
                        <img src="${p.img}" alt="">
                        <div class="product-name">${p.name}</div>
                        <div class="product-price">${p.price}</div>
                    </div>
                `;
            });
        }

        renderProducts(products);

        /* ---------- SEARCH FUNCTIONALITY ---------- */
        document.getElementById("search").addEventListener("input", function() {
            const keyword = this.value.toLowerCase();
            const filtered = products.filter(p =>
p.name.toLowerCase().includes(keyword));
            renderProducts(filtered);
        });
    </script>

</body>
</html>
```

2. Concepts Applied

This project integrates a range of **frontend development concepts**, combining structural design, aesthetic styling, and dynamic behavior to build a cohesive shopping interface. The major concepts applied include:

2.1 HTML Semantic Structure

The project uses semantic HTML elements to create a readable, accessible structure:

- `<nav>` for navigation bar
- `<section>` for banner and product listings
- `<footer>` for end-of-page information
- `<div>` containers for product cards and layout structure

Semantic tags help search engines and browsers interpret the purpose of each section. The main code layout begins with:

```
<nav>
    <div class="logo">ShopNest</div>
    <input type="text" id="search" placeholder="Search
products...">
</nav>
```

This shows structural hierarchy: the navigation contains branding and an interactive search bar. Using semantic elements reduces clutter and clarifies the flow of the webpage.

2.2 CSS Styling and Theme Design

The website uses CSS variables defined in `:root` to maintain a consistent theme of **yellow, orange, and teal**. Example:

```
:root {
    --yellow: #ffcc33;
    --orange: #ff7b00;
    --teal: #1fbabf;
    --dark: #222;
    --light: #fff;
}
```

This ensures:

- Faster color updates across the website
- Easy theme management
- Cleaner, reusable code

The project applies **responsive design** techniques such as:

- CSS Grid (`display: grid`) for product cards
- `auto-fit` and `minmax()` to automatically adjust columns for different screen sizes
- `@media` properties implied through flexible sizing

Each product card is styled with modern UI properties such as border-radius, shadows, and transitions to create a professional interface:

```
.product-card {
    background: var(--light);
    border-radius: 10px;
    padding: 15px;
    box-shadow: 0 2px 6px rgba(0,0,0,0.15);
    transition: 0.3s;
}
```

Hover animations (`:hover`) give interactivity and visual feedback.

2.3 JavaScript for Interactivity

JavaScript is used to dynamically display products and handle search functionality.

Product Rendering

The products are stored as a JavaScript array:

```
const products = [
    { name: "Wireless Headphones", price: "₹1499", img:
"..." },
    ...
];
```

The `renderProducts()` function loops through this list and injects each product card into the DOM using template literals:

```
function renderProducts(items) {
    container.innerHTML = "";
    items.forEach(p => {
        container.innerHTML += `
            <div class="product-card">
                <img src="${p.img}" alt="">
                <div class="product-
name">${p.name}</div>
                <div class="product-
price">${p.price}</div>
            </div>
        `;
    });
}
```

This demonstrates:

- DOM manipulation
- Template-based UI generation
- Logic separation (data vs UI)

Search Input Functionality

A search bar filters products as the user types:

```
document.getElementById("search").addEventListener("inp
ut", function() {
    const keyword = this.value.toLowerCase();
    const filtered = products.filter(p =>
p.name.toLowerCase().includes(keyword));
    renderProducts(filtered);
});
```

Concepts applied:

- Event listeners
- Real-time filtering
- Functional array methods (`filter()` and `includes()`)
- Case-insensitive matching using `.toLowerCase()`

This achieves an interactive, dynamic search experience without the need for a backend.

2.4 Responsive Grid Layout

The product section uses:

```
grid-template-columns: repeat(auto-fit, minmax(220px,
1fr));
```

This ensures:

- Automatic adjustment of product columns
- Optimal display regardless of device width
- Smooth transitions between desktop, tablet, and mobile views

The website’s design prioritizes **fluid resizing**, avoiding rigid widths that break on small screens.

2.5 UI/UX Principles

Every part of the interface uses design concepts such as:

- **Whitespace balance** to reduce clutter
- **Color psychology** (yellow/orange for energy, teal for balance)
- **Hierarchy** (large banner heading, medium card headings)
- **Consistency** in fonts, spacing, and shapes

This produces a clean, user-friendly browsing flow.

3. Features Implemented

3.1 Product Listing Interface

A grid-based product listing allows users to quickly browse multiple items at once. Each card includes:

- Product image
- Title
- Price
- Clean layout

The cards include hover effects to mimic modern e-commerce behavior.

3.2 Search Functionality

The search bar allows instant filtering. As users type, the list dynamically adjusts. This ensures:

- Faster navigation
- More personalised browsing

- Reduced cognitive load

This feature works entirely on the frontend through JavaScript filtering.

3.3 Responsive Layout

The website looks consistent across:

- Desktop monitors
- Tablets
- Mobile phones

The grid layout and flexible units (`minmax`, `auto-fit`, `padding`, and percentage-based widths) ensure adaptability.

3.4 Modern Color-Themed UI

The interface follows a consistent color palette:

- **Yellow & Orange** → energetic, modern
- **Teal** → soothing contrast
- **Light white backgrounds** for readability

This gives the site a distinctive branded identity.

3.5 Interactive Navigation & Banner

The navbar contains:

- Branding
- Search bar
  The banner includes:

- Attention-grabbing gradient
- Clear call-to-action message

This area highlights the site's purpose instantly.

4. Challenges Faced and Their Solutions

4.1 Challenge: Designing a Clean and Modern UI

Issue: Balancing bright colors with readability is difficult, as yellow and orange can overpower text.
Solution:

- Used bright colors mainly for accents and backgrounds
- Kept text dark (`#222`) for readability
- Added white card backgrounds for contrast

This maintained clarity while preserving the theme.

4.2 Challenge: Making the Layout Fully Responsive

Issue: Product cards became too small or misaligned on different devices.
Solution:

- Implemented CSS Grid with `auto-fit`
- Used `minmax()` to ensure minimum readable size
- Allowed automatic column collapse for mobile

This created a scalable layout that adjusts without media queries.

4.3 Challenge: Implementing Product Search Efficiently

Issue: A beginner-friendly code structure was needed for filtering products.
Solution:

- Used JavaScript's `.filter()` for clean logical flow
- Lowercased both search input and product names to avoid mismatches
- Re-rendered only the filtered list

This resulted in smooth filtering with minimal code.

4.4 Challenge: Rendering Dynamic Content Without a Backend

Issue: The project required dynamic content but only frontend technologies.
Solution:

- Stored product data in a JavaScript array
- Generated DOM elements using template strings
- Handled updates through re-rendering functions

This provided dynamic functionality without needing a database or server.

4.5 Challenge: Maintaining UI Consistency

Issue: Multiple components risked inconsistent spacing, font sizes, and colors.
Solution:

- Created CSS variables
- Used uniform padding and card structure
- Applied consistent border radii

This resulted in clean visual coherence across the site.

Conclusion

ShopNest successfully demonstrates the application of modern frontend development techniques to build a visually appealing and responsive e-commerce interface. Using HTML for structure, CSS for design, and JavaScript for interactivity, the project showcases essential skills required in real-world web development. The final interface is lightweight, interactive, visually consistent, and user-friendly, fulfilling the primary objectives of the project.