

Practical Works 1 - Architecture & Modern Layout for E-Commerce

Objective: Setup a professional React environment and build a responsive "Shell" using Tailwind CSS.

Deliverables for the end of session:

1. **Vite Server:** Running on localhost:5173.
2. **Responsive Layout:** The Navbar must collapse/change on mobile, and the Product Grid must adjust from 4 columns to 1 column.
3. **Code Organization:** Components and Pages must be in their correct folders.

1. Environment Setup & Directory Control

1. **Prepare your Workspace:** Open your terminal (PowerShell, CMD, or Terminal) and navigate to where you want to keep your work (e.g., Desktop). Run these commands:

```
# Move to your Desktop or Documents
cd Desktop

# Create a parent folder for the whole module
mkdir Master_Web_2026
cd Master_Web_2026

# Create the specific folder for today's project
mkdir project-store
cd project-store
```

2. **Initialize React with Vite:** Now that you are **inside** project-store, run the command below. We use the dot . to tell Vite to install directly into the current folder rather than creating a new one.

```
npm create vite@latest . -- --template react
```

(If asked to "Proceed?", type y and press Enter. If asked to "Remove existing files?", select Yes.)

3. **Install Dependencies:**

```
npm install
```

4. **Install & Configure Tailwind CSS:** Stay inside the project-store folder and run:

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

5. **Target the Files:** Open the project in VS Code (code .). Open [tailwind.config.js](#) and edit the content line so Tailwind knows exactly which files to style:

```
JavaScript
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

6. **Activate Tailwind:** Open [src/index.css](#), delete everything inside it, and paste these three lines:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

7. **Run the Server:**

```
npm run dev
```

Keep this terminal open. Your app is now live at the URL shown (usually <http://localhost:5173>).

2. Architecture: Arborescence & Component Tree

Before coding, we must define our "Arborescence" (Site Map). In React, this translates into a **Component Tree**.

Action: Inside your [src](#) folder, create the following directory structure to keep the project organized:

- src/components/ (Small reusable UI parts)
- src/layouts/ (Structural wrappers)
- src/pages/ (The main views: Home, ProductDetails, Cart)

3. Building the Layout (The Shell)

We will create a **Main Layout** that persists across all pages.

Task A: The Navbar (Flexbox Mastery)

Create `src/components/Navbar.jsx`. Use Flexbox to align the Logo, Search, and Cart.

```
export default function Navbar() {
  return (
    <nav className="flex justify-between items-center p-4 bg-white shadow-md sticky top-0 z-50">
      <h1 className="text-2xl font-bold text-orange-600">DZ-Shop</h1>
      <div className="hidden md:flex flex-1 mx-10">
        <input type="text" placeholder="Search products..."
          className="w-full border p-2 rounded-l-md outline-none" />
        <button className="bg-orange-600 text-white px-4 rounded-r-md">Search</button>
      </div>
      <div className="flex gap-5 font-semibold">
        <span>Login</span>
        <div className="relative">
          <span>Cart</span>
          <span className="absolute -top-2 -right-2 bg-red-500 text-white text-xs rounded-full px-1">0</span>
        </div>
      </div>
    </nav>
  );
}
```

Task B: The Product Grid (CSS Grid Mastery)

Create `src/pages/Home.jsx`. Use CSS Grid to create a responsive catalog.

```
export default function Home() {
  const placeholders = [1, 2, 3, 4, 5, 6, 7, 8]; // Temporary data
  return (
    <div className="p-8">
      <h2 className="text-3xl font-bold mb-6">Latest Products</h2>
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
        {placeholders.map((id) => (
          <div key={id} className="border rounded-lg h-64 bg-gray-100 animate-pulse flex items-center justify-center">
            <span className="text-gray-400">Product Placeholder</span>
          </div>
        ))}
      </div>
    </div>
  );
}
```

4. Putting it Together

Modify `src/App.jsx` to assemble your components:

```
import Navbar from "../components/Navbar";
import Home from "../pages/Home";

function App() {
  return (
    <div className="min-h-screen bg-gray-50">
      <Navbar />
      <main>
        <Home />
      </main>
    </div>
  );
}
export default App;
```