Group Capstone Project — E-Commerce Website

Title: Group Capstone Project: E-Commerce Website

Due Date: 17 November 2025

Short description:

Build a pixel-perfect e-commerce frontend from a Figma design using React, Tailwind CSS, React Router for navigation, and Redux for state management. All data lives client-side (no backend). This project demonstrates component design, routing, global state, forms, derived state, and UI/UX fidelity.

Figma Reference: E-Commerce Store — Melsoft Academy

Tech stack:

- React (functional components + hooks)
- Tailwind CSS (utility-first styling)
- react-router-dom (routing)
- Redux Toolkit (state management)
- Optional dev tools: Redux DevTools, ESLint, Prettier, Git/GitHub

---

Project Goals

1. Recreate the provided Figma design faithfully (layout, spacing, typography, colors).
2. Implement a working shopping flow: browse → view product → add to cart → checkout.

3. Demonstrate sound React architecture: reusable components, clear folder structure, and predictable global state via Redux.
4. Validate forms (shipping/payment) and compute derived values (cart totals).
5. Produce clean, readable code with basic tests and clear documentation.

High-level Feature List

- Fully responsive **Home** page with product listings
- **Menu Sidebar** (toggleable)
- **Search** bar (local filtering)
- **Product** detail page
- **Cart** overview (quick and full cart page)
- **Checkout** forms (shipping + payment) and **Order Summary**
- **Order Successful** confirmation page

Pages & Routes (react-router-dom)

- `/` — Home (product grid, sidebar, search, cart overview)
- `/product/:id` — Product Page (details + add to cart)
- `/cart` — Cart Page (list, modify qty, remove)
- `/checkout` — Checkout (shipping, payment, summary)
- `/order-success` — Order Successful (static confirmation)

Routing notes:

- Use `Link`/`NavLink` for navigation.
- Pass product object via location state when navigating to Product Page for simplicity, and fallback to finding product by id from local product array.

Core Components (suggested)

- `App` — top-level router + layout
- `Navbar` — top navigation, cart icon, search input
- `Sidebar` — menu items (toggle with `useState`)
- `ProductCard` — reusable product preview (image, name, price, Add button)
- `ProductList` — maps product data to `ProductCard`
- `ProductDetail` — product details layout
- `CartSummary` — small cart widget (used in Navbar/Home)
- `CartPage` — full cart with quantity controls
- `CheckoutForm` — shipping and payment subcomponents
- `OrderSuccess` — static success page
- `UI` atoms — `Button`, `Input`, `Badge`, `Modal` (optional)

Data & Local Storage

- Keep product catalog as a local JSON/JS array (`/src/data/products.js`).
- Cart state is stored in Redux; persist cart to `localStorage` so a refresh keeps items.

- Example product shape:

```JavaScript
{
  id: 'p123',
  name: 'Product Name',
  price: 299.99,
  category: 'Shoes',
  description: 'Long description...',
  image: '/assets/img.jpg'
}
```

Redux Structure (Redux Toolkit recommended)

- **Slices:**
  - `cartSlice` — actions: `addItem`, `removeItem`, `increaseQty`, `decreaseQty`, `clearCart`.
  - (Optional) `uiSlice` — actions: `toggleSidebar`, `openModal`, etc.
- **Selectors:**
  - `selectCartItems`, `selectCartTotalQty`, `selectCartTotalPrice` (derived)
- **Store:** configure with `configureStore` and include Redux DevTools.
- **Persistence:** use a middleware or `subscribe` to save cart to `localStorage` on change.

State Management Plan

- Local component state (`useState`) for ephemeral UI (sidebar open/close, search input, controlled form inputs).
- Redux for global, shared state (cart contents, possibly user session if added later).
- Derived values (total cost) computed via selectors or `useMemo` in components.

Styling & Design

- Use Tailwind CSS to match Figma styles. Create a small design token file (colors, spacing, fonts) via `tailwind.config.js` to match Figma.
- Build reusable component classes (e.g., `btn-primary`, `card`) using `@apply` in a CSS file to keep markup tidy.

Development Roadmap (suggested milestones)

1. **Setup & Skeleton**
   - Create React app, install Tailwind, react-router-dom, Redux Toolkit.
   - Add project structure and global layout (Navbar + Footer).
2. **Static UI**
   - Build `ProductCard`, `ProductList`, `Sidebar`, and other static components to match Figma.
3. **Routing**
   - Configure routes and navigation between Home and Product pages.
4. **Cart (Redux)**
   - Implement `cartSlice`, cart UI, and persistence to `localStorage`.

5. **Cart Page & Quantity Controls**
6. **Checkout Forms & Validation**
7. **Polish & Responsiveness**
    - Tailwind tweaks, mobile layout, and accessibility checks.
8. **Testing & Cleanup**
    - Manual testing, fix bugs, refactor code for clarity.

Note: adjust days to your team schedule — this roadmap is a baseline.

Deliverables

- GitHub repo with clear README (how to run, tech choices, known issues)
- Fully navigable React app matching Figma
- Redux state with `cart` persisted

Stretch / Optional Enhancements

- Add product filtering & categories
- Add sorting (price, popularity)
- Integrate a payment simulator (Stripe test keys) for a fuller checkout demo

---

Collaboration Tips for a Group Project

- Use feature branches and PRs on GitHub; review each other's code.

- Agree on a component naming convention and folder structure early.
- Keep tasks small and assign responsibilities (e.g., UI, data, Redux, forms).