

UR Mobile Shopee Website Documentation

1. Project Overview

This is the frontend of an eCommerce website built with React and Vite. It includes features such as product listing, cart management, checkout, and user authentication. The UI is styled using TailwindCSS, and state management is handled via local state and context API.

2. Tech Stack

- React (JSX-based UI components)
- Vite (Fast build tool for development)
- TailwindCSS (For styling)
- React Router (For navigation)
- Axios/Fetch API (For backend communication)

3. Folder Structure

```
extracted_project/  
  .gitignore  
  package-lock.json  
  tailwind.config.js  
  eslint.config.js  
  index.html  
  package.json  
  README.md  
  vite.config.js  
  src/  
    App.css  
    App.jsx  
    index.css  
    main.jsx  
    assets/ (icons, images, styles)  
    components/  
      AddAddressModal/  
      Category/  
      Checkout/  
        Address.jsx  
        Bag.jsx  
        CheckoutPage.jsx  
        Payment.jsx
```

4. Routing & Navigation

The project uses React Router for page navigation. Example routes:

- / - Home Page
- /shop - Product Listing
- /cart - Shopping Cart
- /checkout - Checkout Process
- /login - User Authentication

5. State Management

- Local state is managed using `useState` and `useEffect`.
- Context API is used for global state management (e.g., user authentication, cart state).

6. API Communication

- Uses `Axios` or `Fetch` API for making backend requests.
- Expected API Endpoints:
 - `GET /products` - Fetch all products
 - `POST /login` - User authentication
 - `POST /checkout` - Process order

7. Authentication Flow

- Users can log in using a form on `/login`.
- Authentication tokens are stored in `localStorage/sessionStorage`.
- Protected routes are managed via `PrivateRoute` components.

8. Deployment & Build Process

- Uses `Vite` for development and build (`npm run build`).
- The output files are optimized for production and can be deployed to `Vercel`, `Netlify`, or any hosting service.

9. Future Enhancements (Optional)

- Adding a Wishlist feature
- Implementing Payment Gateway (Stripe, Razorpay)
- Enhancing UI with animations