

GUIDELINES FOR E-commerce

Pages

- E-commerce dashboard
- Product Detail Page
- Cart Page
- Wishlist

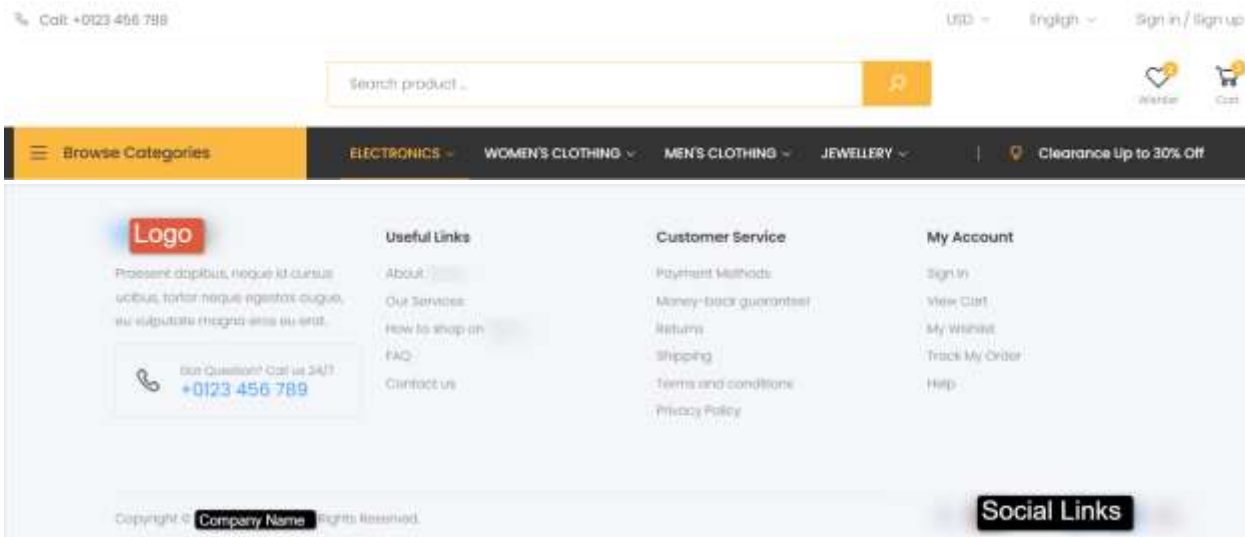
Framework / Dependencies

- ReactJS - <https://reactjs.org/>
- React-redux - <https://react-redux.js.org/>
- Redux-thunk - <https://github.com/reduxjs/redux-thunk>
- TailwindCSS - <https://tailwindcss.com/>
- react-router-dom - <https://www.npmjs.com/package/react-router-dom>
- react-icons - <https://react-icons.github.io/react-icons/>

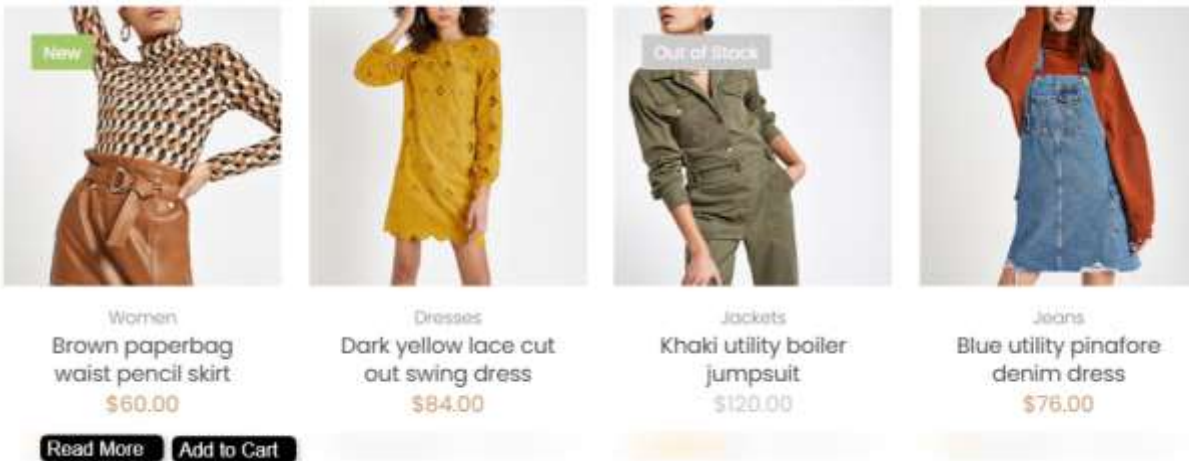
Guidelines

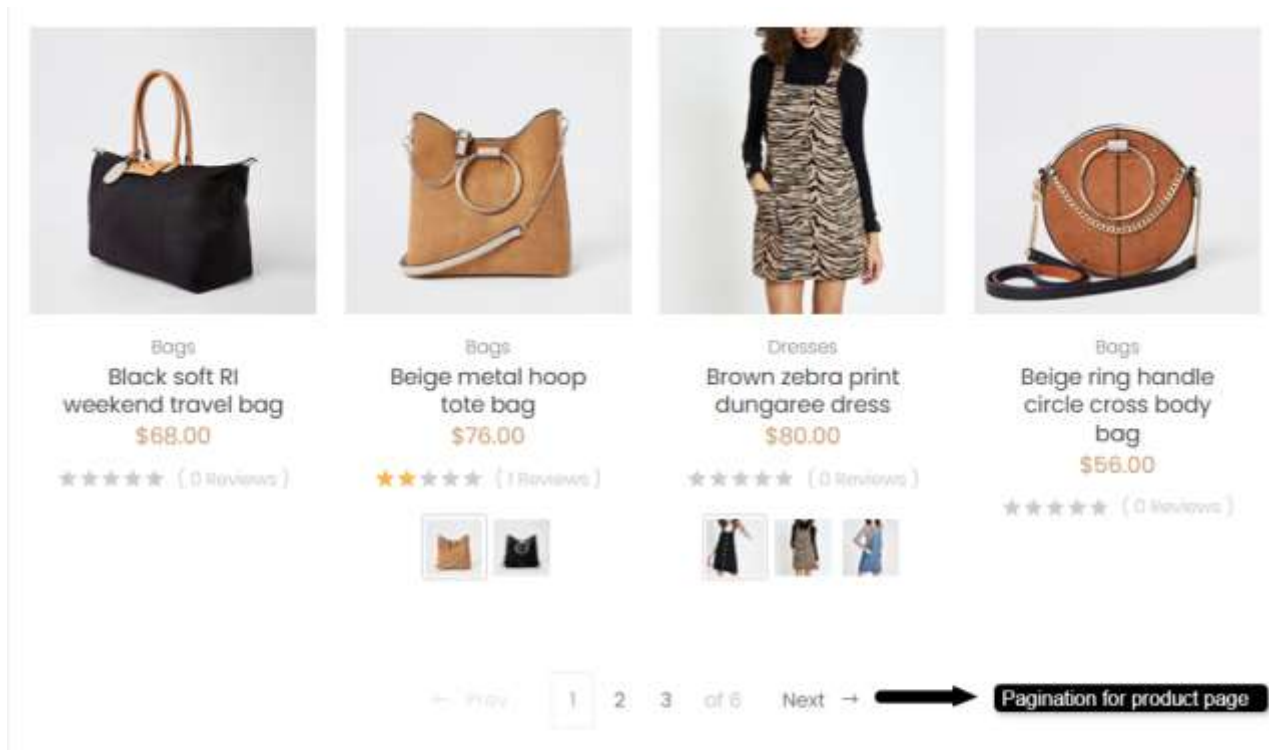
1. Website: You are going to build an e-commerce website. Here, users can browse products according to their categories, see the detail page of the relevant product, create a Wishlist for their future buying, and add the product to the cart.
2. You have to come up with an interesting Name and Logo for your E-commerce website. You can use Figma (<https://www.figma.com/>), Canva (<https://www.canva.com/>) for Logo creation.
3. Create a new project with create-react-app/vite app and install Redux, react-redux, redux-thunk, react-router and Tailwind CSS using npm
4. Your project directory must have a well-defined structure. Try to make different folders for different aspects of code. (For e.g. components folder for different components). You can learn more about react project structure here. https://youtu.be/Mm6_DIO5vvs
5. For Color Palette: Refer: <https://colorhunt.co/> for selected color palette for the website.
6. API Reference: You can use Fake Store API to fetch all product data and category wise data fetching (<https://fakestoreapi.com/>).
7. Plan Main component: Header, Footer, Banner, Product cards, Product Display Card.
8. Key Features:
 - E-commerce Dashboard: This will be the landing page of the application. Create fixed Header (Logo, cart, favorites icon, search bar Login button, notification icon), Banner section (Implement Scroll or parallax effect) and display product category and display products (will be fetched from API), Footer (Add Quick Links, Brief about company, subscribe to newsletter, contact details like number, email). Use scroll effects and animations to display products.
 - Category Filter: You can add filtration based on category, if user clicks on "Electronics", display electronics products and same for all the categories.

- Search Functionality: Implement search functionality. Once the user clicks on the search icon, then displays the search bar, otherwise only the search icon. Users can search the product based on the product name. Display only the products which matches the search string.

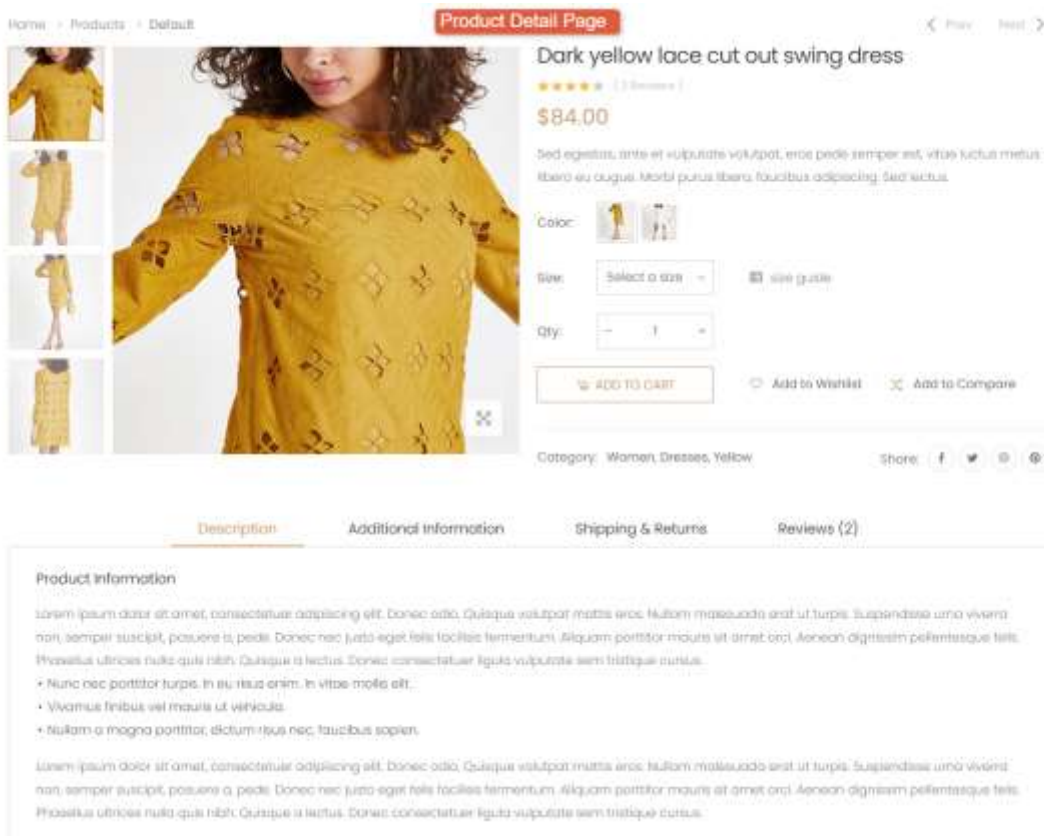


- Login: When a user clicks on the Login page, display the Login page. Add Email and Password as Input field and do relevant Validation.
- Product Card: Here, Display product Name, Image, discounted and original price, add to cart button, add to favorites icon, review star.



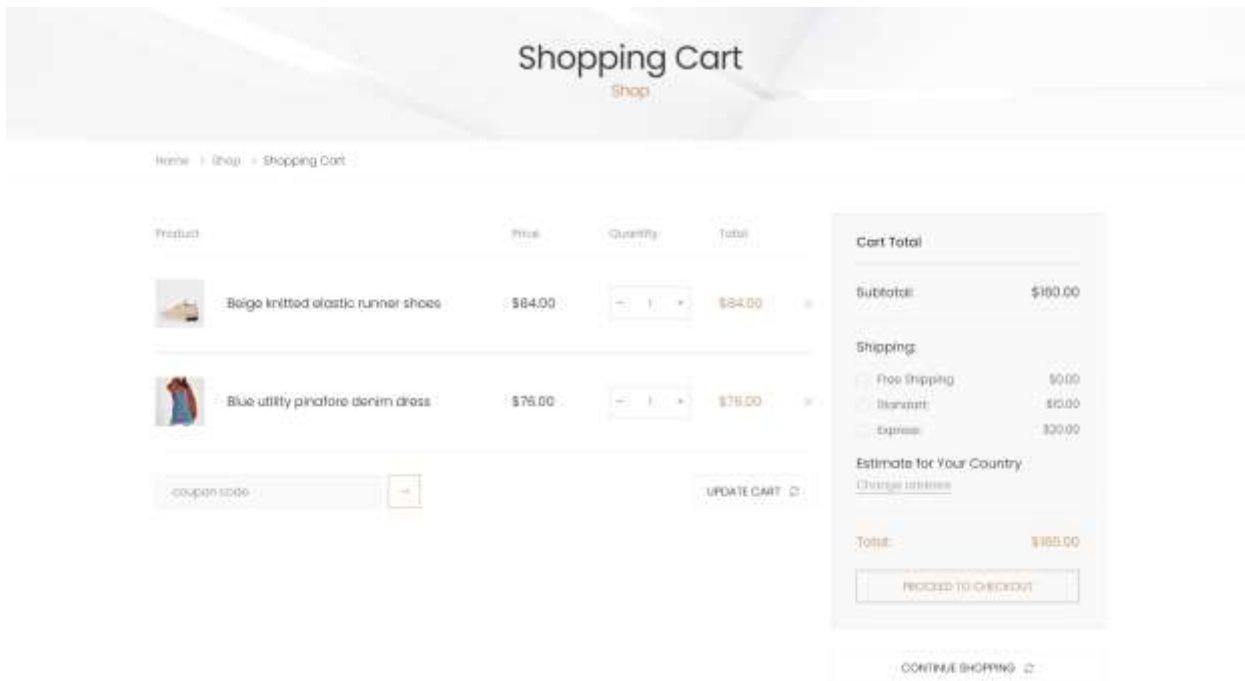


-
- **Product Details Page:** When a user clicks on any product, the product details page will open for that product. Here, display the details of the product e.g. Image, price, units, name, reviews, payment options, offers, delivery by, in details information about products, key features of product, add to cart button.

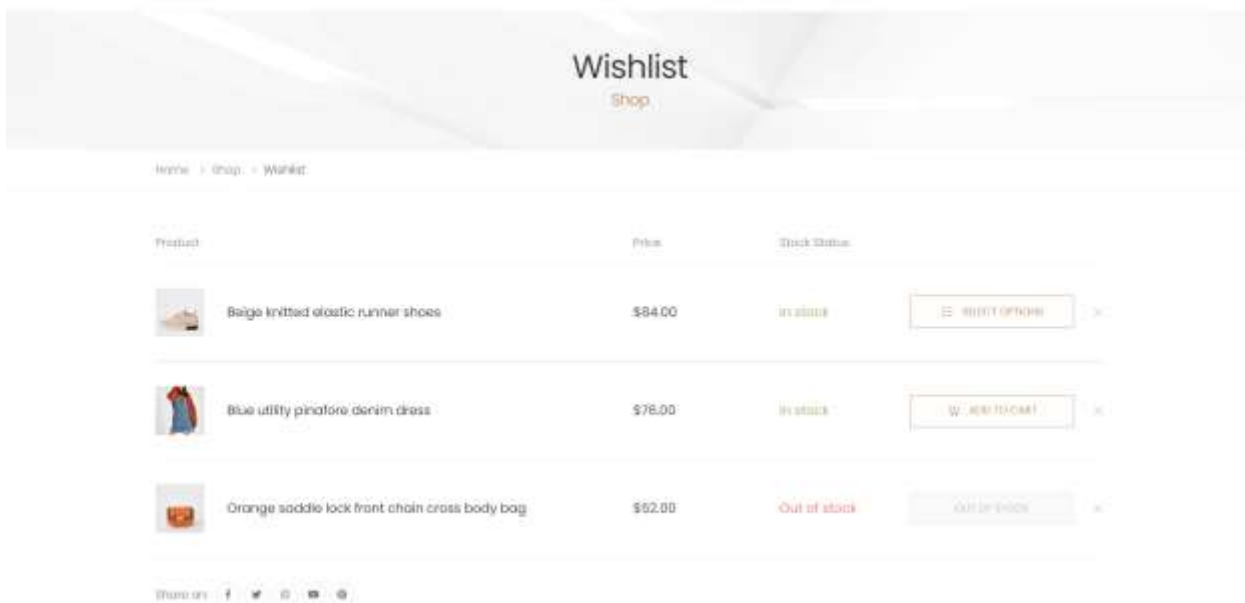


- **Cart:** When a user clicks on the “Add to Cart” button on any product, add that product to cart and display on cart page when user clicks on cart icon and change button display from “Add to cart” to “Go to Cart”. If a user deletes

or adds any other product then displays Total price accordingly. Display number of items added to cart on the cart icon present on header.



- Favorites/Wishlist: When a user clicks on the “Add to Favorites”/ “Later” icon on any product, add that product to Wishlist and display on the Wishlist page when the user clicks on the Wishlist icon. Display number of items added to Wishlist on the Wishlist icon present on header.



- Add what we sell section: You can add briefs about company policies like Free Returns, Free Shipping, 24/7 Support on the landing page.



- Add Privacy and T&C section (Optional): Here you can use modal to display privacy and T&C.

8. Dev Requirements:

- Develop the web application using SPA technology and state management library. React + Redux, Tailwind CSS.
- As the data stored in react state is not persistent, you can use local storage API in JavaScript to store your cart and favorite data and make it persistent. You can learn more about it at - https://www.w3schools.com/jsref/prop_win_localstorage.asp
- Product Data should be dynamic, even blogs (if added) should come from API.
- All input fields should be validated.
- Folder Structure and File naming Convention should be proper.
- Implement carousel, card hovering, accordion for FAQ (Frequently Asked Questions) etc.
- It should be responsive.
- UI should be user friendly, even a non-technical person should be able to navigate throughout the website easily.
- Implement Unit testing to ensure bug free application using React-Testing Library.

9. Hosting Requirements:

- App Service should be integrated with your version control platform (e.g. GitHub) (<https://github.com/>) to pull the changes when made.
- Host your app on Heroku cloud or any other hosting provider like Vercel (<https://vercel.com/>), Netlify(<https://www.netlify.com/>).