GadgetHeaven✨

Your go-to destination for the latest and greatest gadgets

Develop an e-commerce platform for GadgetHaven, a gadget-buying website, based on a detailed Figma design. The site should feature a structured navigation bar, product categories, and individual product details pages. Key functionalities include a shopping cart, a wishlist system, and product filtering. The Context API will manage the cart and wishlist, with optional LocalStorage for data persistence. Additional pages will allow users to view and manage their cart and wishlist, sort items by price, and ensure a smooth user experience. The site must handle edge cases like price limits and prevent errors when reloading.

Requirements Update On Details Page:

💡💡Don't need to show the count of added items on the cart and wishlist in Navbar.

Navbar Section

- There will be these menu items => Logo, Brand Name, Dashboard, Stats
- Indicate Active route in the Navbar

Home Page

- **Banner Section:** According to Figma, a nice and attractive Banner will appear on the Home page. The banner will have a button that takes the user to the Dashboard page.
- **Categories Sidebar:** According to Figma, there are a few categories of gadgets, such as computers, phones, smart watches, chargers, and power banks. Categories will appear as a sidebar just like Figma
- **Gadgets Cards:** This page will default show at least 6-9 data in a card grid format.
  Each Card will have an image of the product, product name, price, and 'Details' button. Clicking the button will take you to the details page of the item.

- At least 3 categories and the first category must have 6 data sets. Other categories may have 2 or more data. (6 + 2*2 = 10 data in total). Each item's price will be at least 50 $ (not less than 50 $, could be more) and unique.
- Clicking on any category will show gadgets of that specific category on the right. Do this using nested layout
- Ensure these properties are in each data set.
- Each JSON data will contain:
    - product_id
    - product_title
    - product_image
    - category
    - price
    - description
    - Specification [ ]
    - availability (true, false)
    - rating
    - Other field (if you want to add)

Example:

```
[{
    "product_id": "P001",
    "product_title": "Apple iPhone 15 Pro Max",
    "product_image": "https://example.com/iphone15promax.jpg",
    "category": "Smartphones",
    "price": 1299,
    "description": "Experience the ultimate smartphone with the iPhone 15 Pro Max. ",
    "specification": [
      "6.7-inch Super Retina XDR display",
      "A17 Bionic chip",
      "Triple-camera system (48MP Main, 12MP Telephoto, 12MP Ultrawide)",
      "LiDAR Scanner",
      "Up to 2TB storage"
    ],
    "availability": true,
    "rating": 4.8
  }]
```

Footer Section: Make a footer section with meaningful information and content like Figma.

💡Ensure that the Navbar and Footer are showing on all the pages without Error/404 Page.

Details Page

- The “Details” page will have an image of the product, name, price, details, etc, (all the properties of the single item) according to Figma design.
- Each details page will have two buttons Add to Cart (or 🛒 icon button), and a ♥ (heart) icon button.
    - Clicking the Add to Cart / 🛒 (cart) icon button will add the item to the cart. Clicking the ♥ (heart) icon button will add the item to the

WishList.

- Disable ♥ (heart) icon button after clicking once. Don't allow adding the same items to the wish list more than once.

💡💡Don't need to show the count of added items on the cart and wishlist in Navbar.

You can implement functionalities using Context API and LocalStorage or any other way. It's up to you what you use.

You can use the NPM package React Rating star components or others for show rating.

Dashboard Page

- This page will have two tabs => Cart, and WishList.
- Cart items will appear when you click the "Cart" tab.
- Wish items will appear by clicking the "Wish List" tab.
- Items card will contain gadget name, image, price, and other info that you want to show
- Show the total price of cart items in the "Cart tab"
- There will be a "Sort by Price" button in the Cart tab, Clicking this button will sort the Cart items by price in descending order. (Costly items will appear first)

New Route/Page:

- Add 1 more routes to your website. Add relevant and meaningful content for those routes.

Miscellaneous

- Reloading the website should not show any error or 404 page.
- Add a 404 page.

- Provide documentation in the README.md file
    - Attach Live Website Link
    - Attach Requirement Document Link
    - List of React Fundamental concepts used in the project
    - What you have used for handling and managing data (context API/local storage)
    - 5 features of your website/project
- Show toast on
    - Adding item to cart
    - Adding item to wishlist
    - Show different messages in these two activities.

Challenge Tasks

- You can see that in Figma for the home page, the background color is different from the background color of the navbar on other pages. Implement this using useLocation() hook

- Clicking the Purchase button will Congratulate using Modal. Then the cart will be empty/total price will be 0, (without reloading the page or using window.href). Keep this button disabled if the cart is empty and the total price is 0. After clicking on the “Close button” of the modal it will redirect you to the home page. Do this navigation using useNavigate() hook.
- Use favicon for your website and add dynamic title for each page on your website. Resources

Optional but Recommended To Explore: (It doesn’t contain any marks)

- Limit adding items to the cart after reaching a maximum (inclusive) 1000 $. Implement removing items from the Cart. Removing items from the Cart will also decrease the total price, and then different items can be added to the cart until it reaches 1000 $ (inclusive).

- Implement removing items from the Wish List. Cards within the wishlist will also have an ‘Add to cart’ button, Clicking this button will remove the item from the wishlist, and also add it to the Cart. In case of failure (Cart exceeding total 1000 $), don’t perform this, and show an error toast.

- ### Statistics Page:

  On this page, a Composed Chart will appear. This will be a **Price vs. product name** Composed chart. That means, the Y-axis will show Prices and the X-axis will be the Product name. Area and Bar data will show according to Prices, scatter data will show according to rating.