

Day 4 – Dynamic Frontend Components – General - Ecommerce

Objective

Focus: Build dynamic frontend components to display marketplace data fetched from Sanity CMS or APIs while incorporating responsive design and reusable components. Enhance functionality and user experience.

Step 1: Revalidation of Day 1, 2, and 3

- Reviewed work done in the first three days, ensuring all components and schemas were correctly implemented.
- Adjusted and updated Sanity schema to align with the latest requirements for data fetching and component design.

```
• const product = {
•   name: "product",
•   title: "Product",
•   type: "document",
•   fields: [
•     {
•       name: "id",
•       title: "ID",
•       type: "string",
•     },
•     {
•       name: "name",
•       title: "Name",
•       type: "string",
•     },
•     {
•       name: "imagePath",
•       title: "Image Path",
•       type: "url",
•     },
•     {
•       name: "price",
•       title: "Price",
•       type: "number",
•     },
•   ],
• }
```

```

    {
      name: "description",
      title: "Description",
      type: "text",
    },
    {
      name: "discountPercentage",
      title: "Discount Percentage",
      type: "number",
    },
    {
      name: "isFeaturedProduct",
      title: "Is Featured Product",
      type: "boolean",
    },
    {
      name: "stockLevel",
      title: "Stock Level",
      type: "number",
    },
    {
      name: "category",
      title: "Category",
      type: "string",
    },
    {
      name: "images",
      title: "Images",
      type: "array",
      of: [
        {
          type: "image",
          options: {
            hotspot: true, // Enables image cropping
          },
        },
      ],
    },
  ],
};

export default
product;

```

```

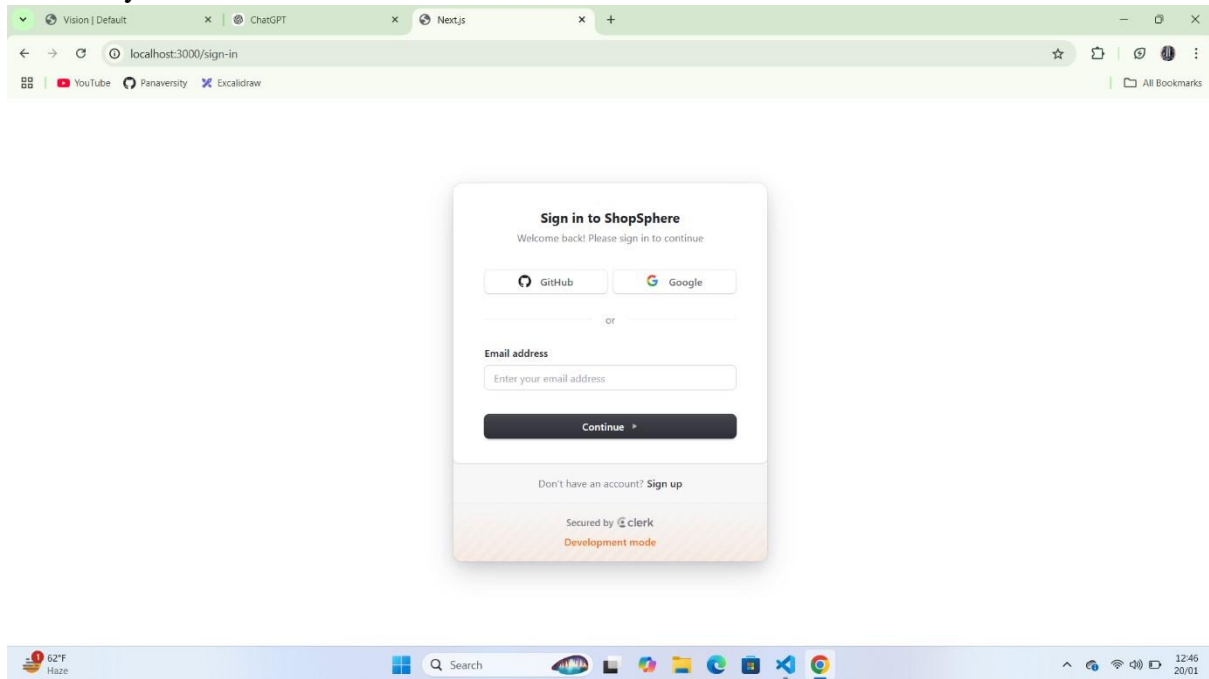
const blog = {
  name: "blogPost",
  title: "Blog Post",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Title",
      type: "string",
    },
    {
      name: "author",
      title: "Author",
      type: "string",
    },
    {
      name: "publishedAt",
      title: "Published At",
      type: "datetime",
    },
    {
      name: "category",
      title: "Category",
      type: "string",
    },
    {
      name: "excerpt",
      title: "Excerpt",
      type: "text",
      description: "A short summary of the post.",
    },
    {
      name: "image",
      title: "Image",
      type: "image",
    },
  ],
}; export default blog;

```

Step 2: Clerk Sign-in and Sign-up Pages Implementation

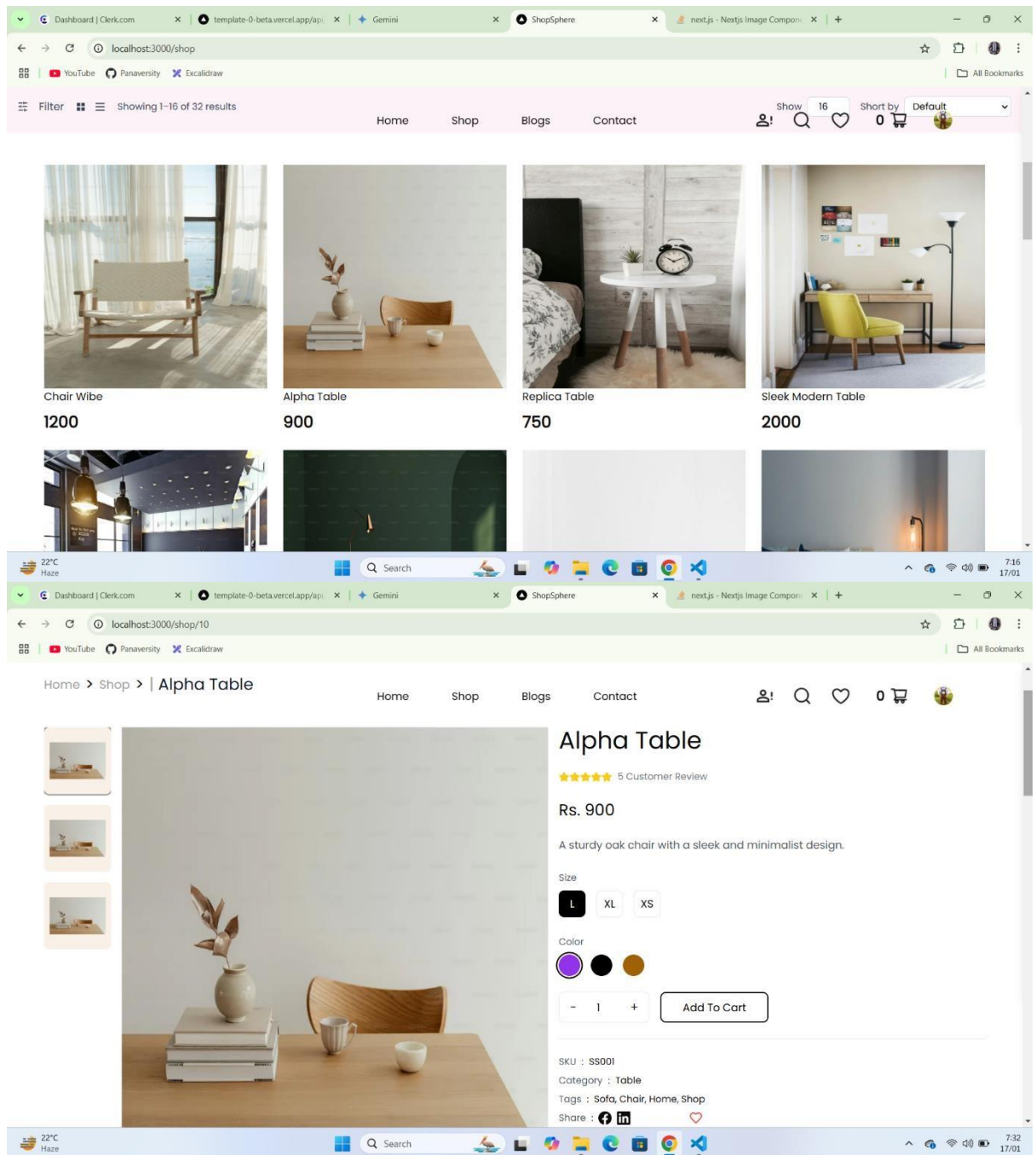
- Integrated Clerk authentication to handle user sign-in and sign-up processes seamlessly.

- Customized the pages for a user-friendly interface while maintaining a responsive layout.



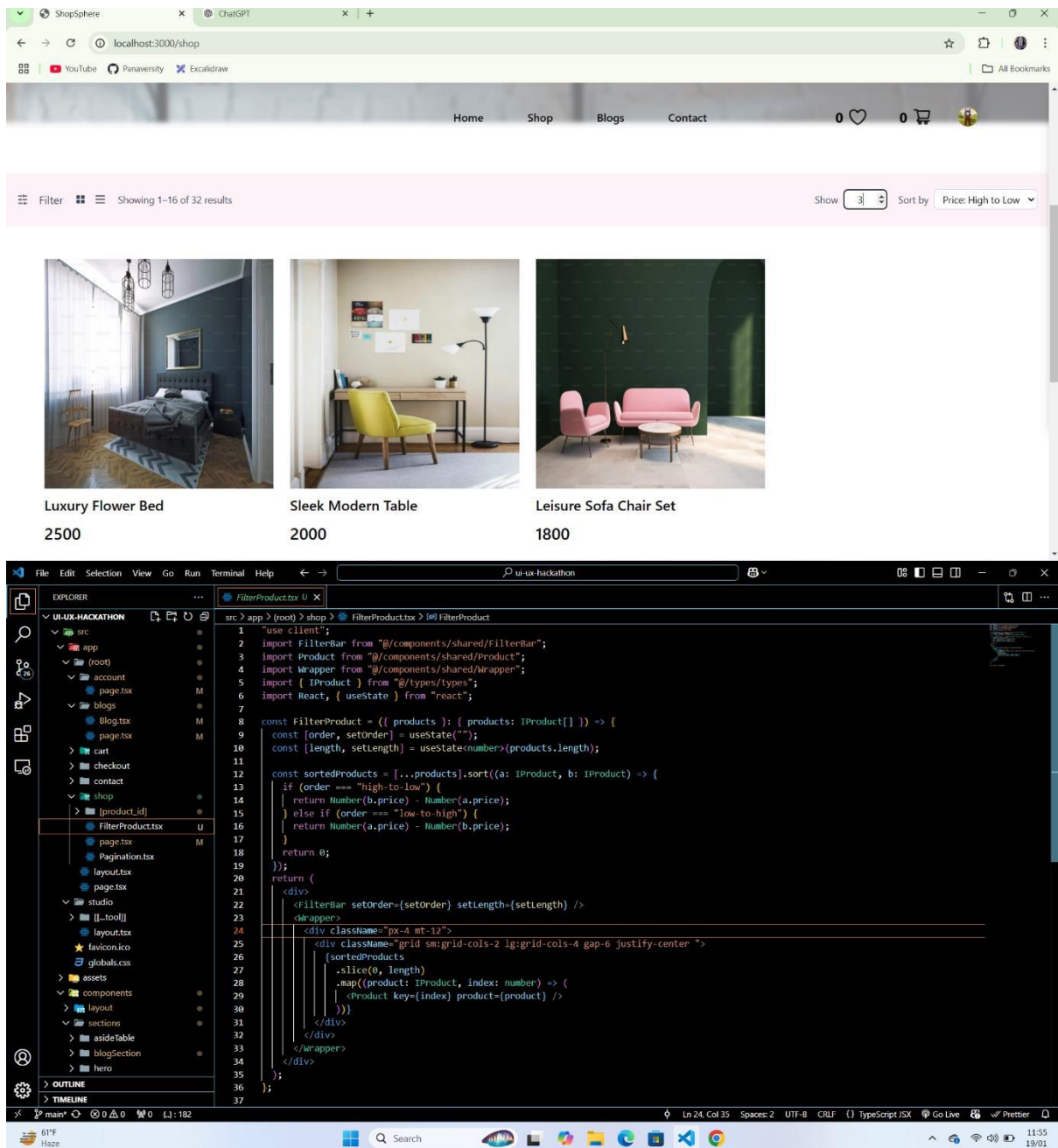
Step 3: Key Functionalities Implementation

1. **Efficient Data Display** ○ Fetched and displayed marketplace data from Sanity CMS using optimized GROQ queries.
 - Ensured efficient data rendering for a smooth user experience.
2. **Product Details Page** ○ Developed a product details page to display comprehensive information about selected products. ○ Included fields such as product name, price, description, and images.



Step 4: Product Filtering Component

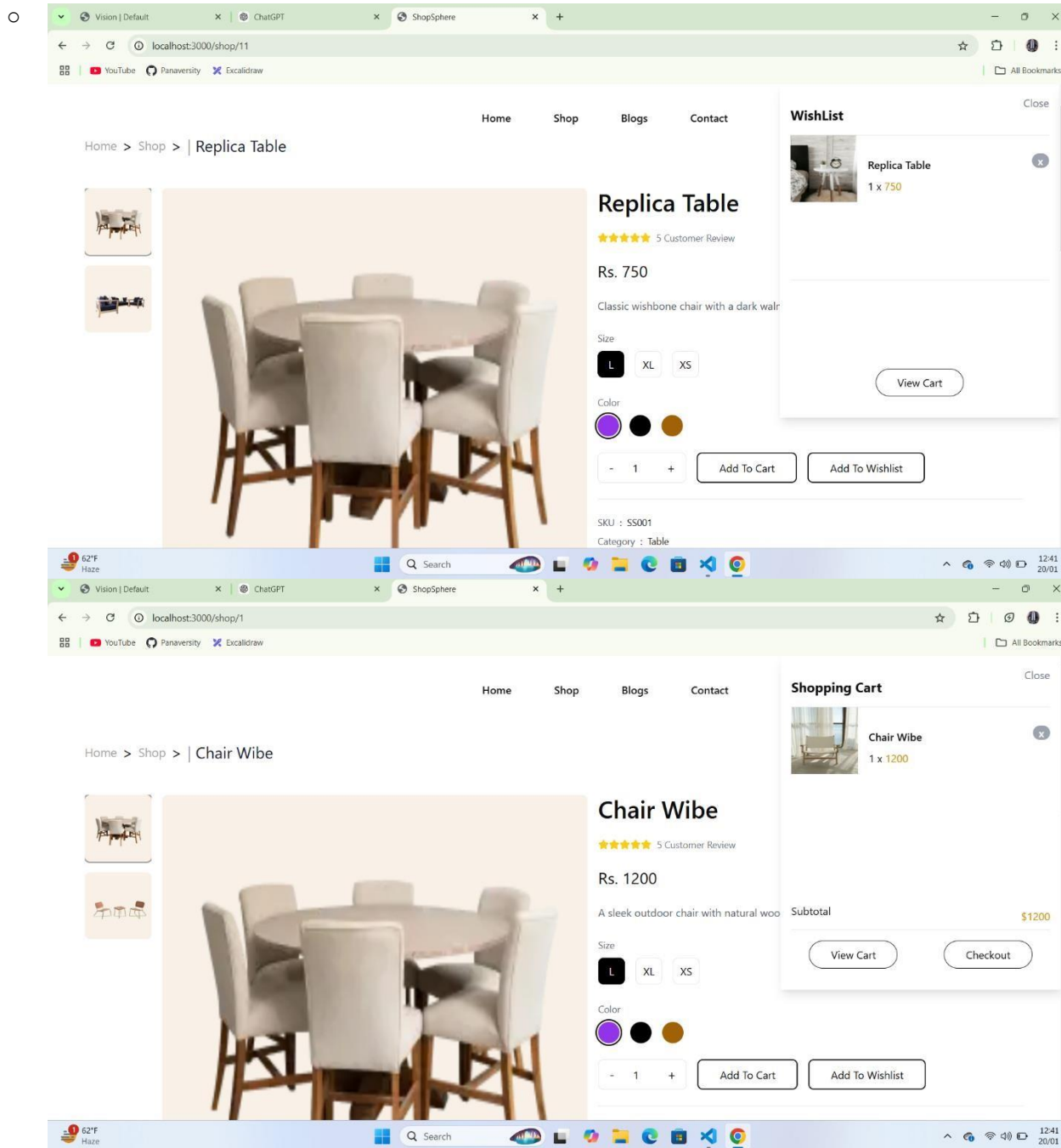
- Implemented filtering functionality to sort products by price:
 - **High to Low**
 - **Low to High**
- Added a filter to show products by number (e.g., 10, 20, 50 products per page).



Step 5: Wishlist and Cart Management Functionality

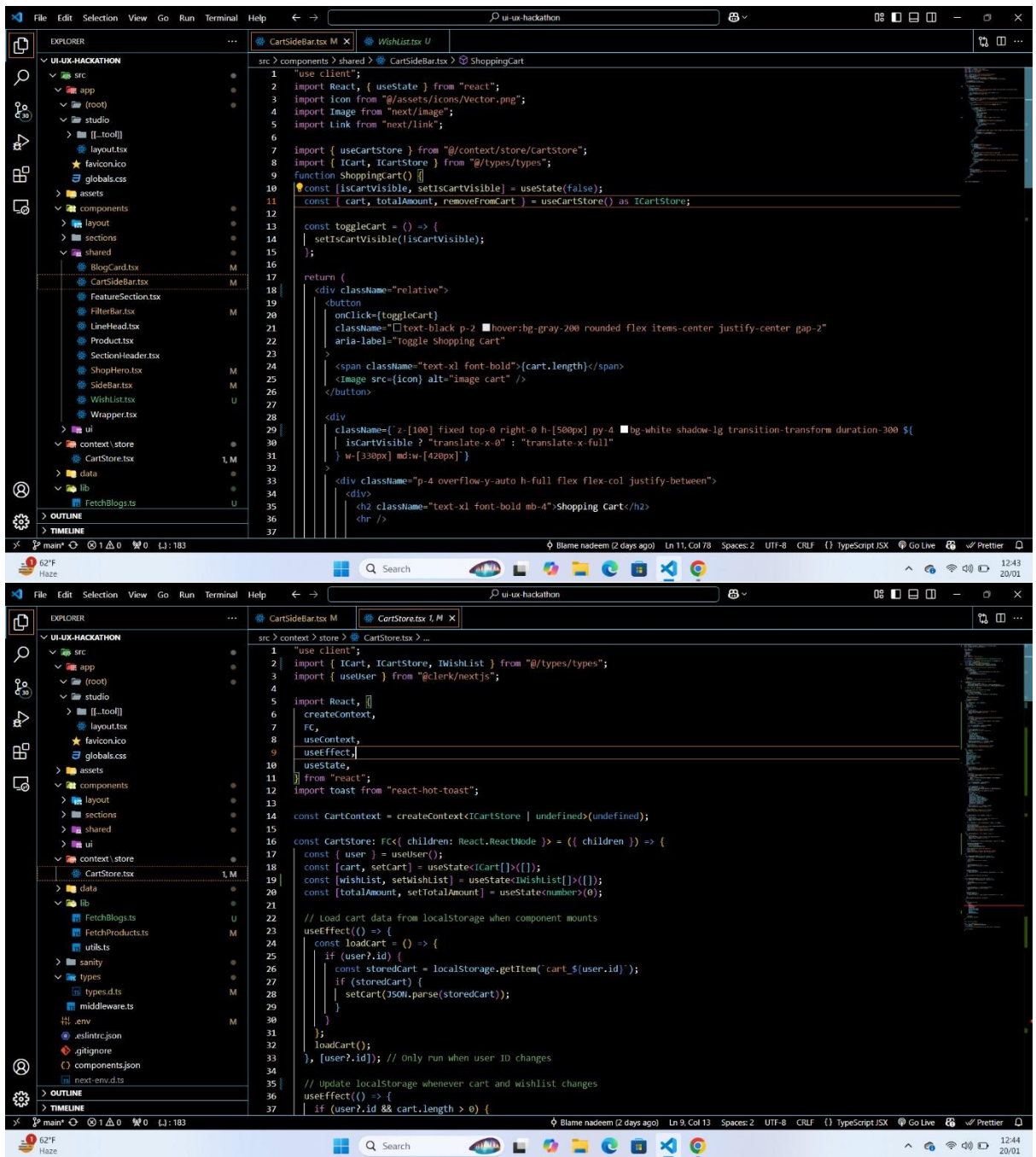
1. Wishlist Management

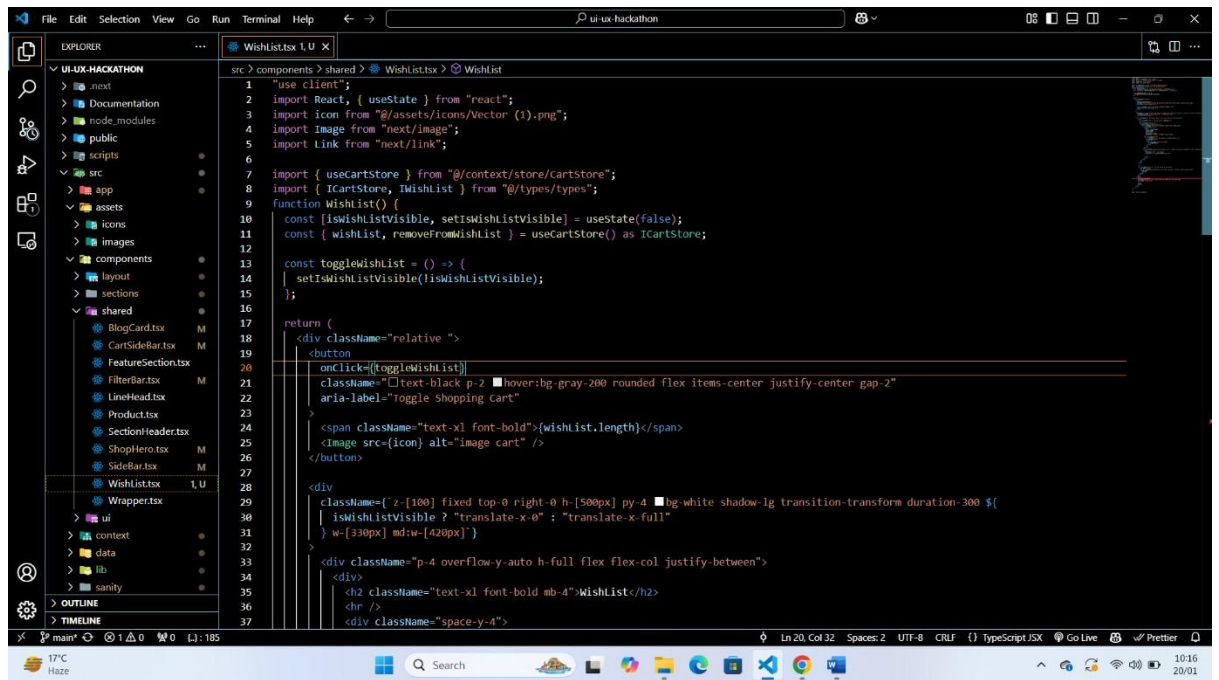
- Utilized local storage to manage the wishlist feature.
- Enabled adding and removing products from the wishlist.



Added functionality for:

- ✦ Adding products to the cart.
- ✦ Incrementing and decrementing product quantity.
- ✦ Removing products from the cart.
- Leveraged local storage to ensure persistence of cart data.





Nadeem Khan