Day05
TESTING ERROR HANDLING AND BACKEND INTEGRATION REFINEMENT:
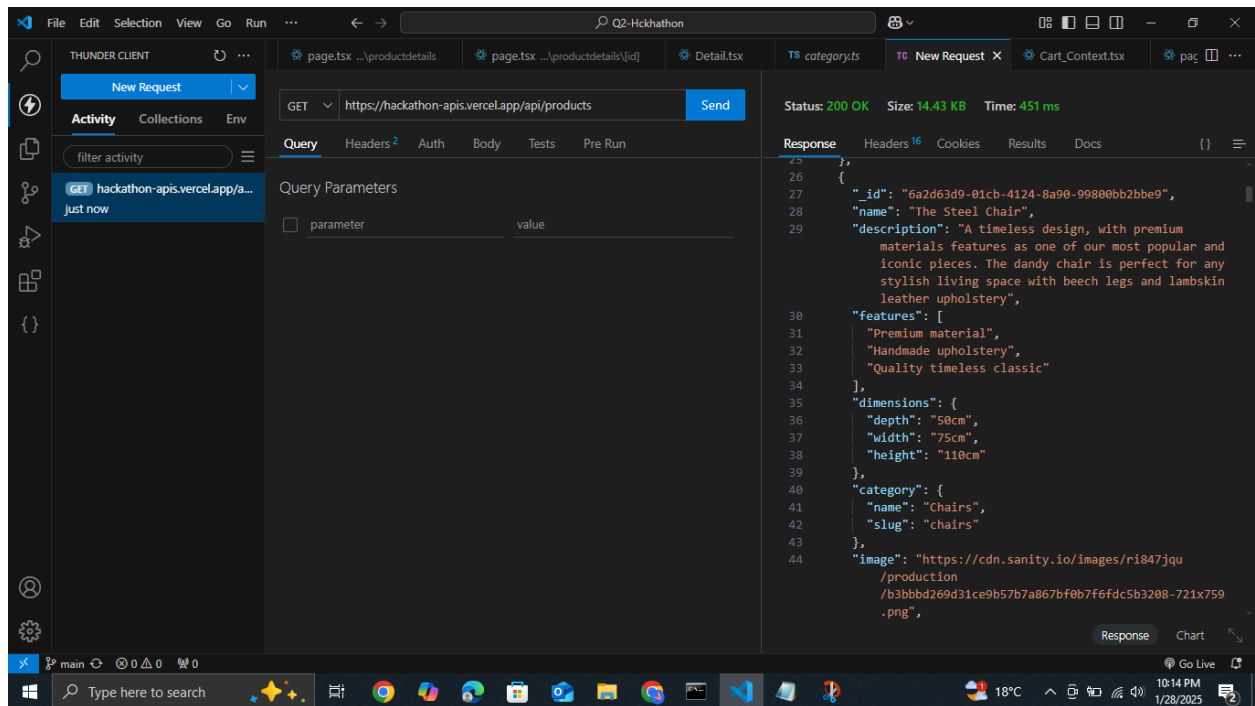
NAME:ANOUSHA BAIG
ROLLNO:00296842

● FUNCTIONAL TESTING:

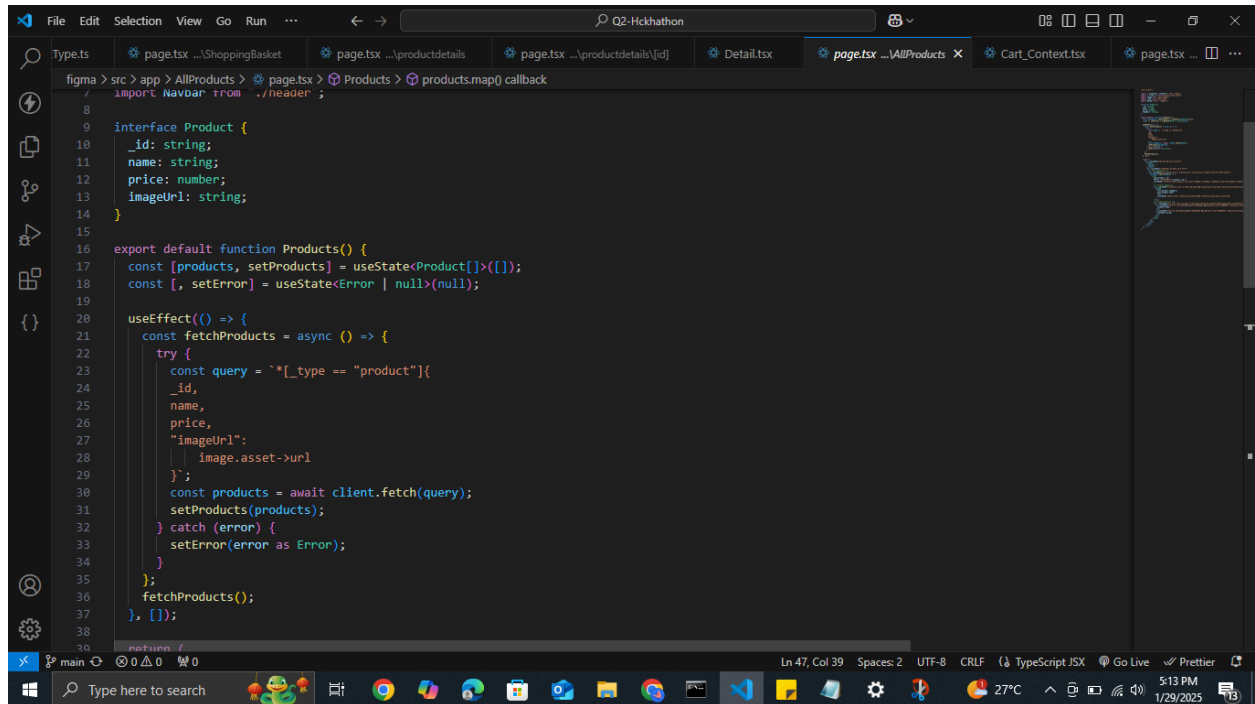- Test core functionalities like product listing, detail pages, cart operations, and user profile management.

Test Case ID	Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate homepage product display	Open homepage > Verify product details load correctly on the page	Products should be displayed correctly, with all information such as images, prices, and descriptions	All products are correctly displayed.	Pending	High	-	Product display is correct
TC002	Test search bar functionality	Use search bar to search for a product > Verify search results match query	Search results should display matching products accurately	search work as expected.	Pending	Medium	-	Works fine.
TC003	Test add to cart functionality	Select a product > Click 'Add to Cart' > Verify product appears in cart	Product should be successfully added to the cart with correct quantity	Cart updates as expected	Pending	High	-	Works fine.
TC004	Test update cart functionality	Add a product to the cart > Update quantity of the product in cart > Verify the quantity is updated	Cart should reflect the updated quantity correctly	Works Well	Pending	High	-	Works fine.
TC005	Test dynamic product details	Click on a product > Verify dynamic details like images, descriptions, and pricing	Product details should load dynamically and display correctly	Product detail page loads correctly.	Pending	Medium	-	Dynamic Routing works good.
TC006	Test security and API data validation	Check data fetched from the API > Ensure data is accurate and secure	API data should be correct and should be securely fetched	Api Works	Pending	High	-	Security measures implemented successfully.

- ## **Testing Tools:**

- ThunderClient: For API response testing.

- **ERROR HANDLING:**

```tsx
import NavBar from "./header ";

interface Product {
  _id: string;
  name: string;
  price: number;
  imageUrl: string;
}

export default function Products() {
  const [products, setProducts] = useState<Product[]>([]);
  const [, setError] = useState<Error | null>(null);

  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const query = `*[_type == "product"]{
          _id,
          name,
          price,
          "imageUrl":
            image.asset->url
        }`;
        const products = await client.fetch(query);
        setProducts(products);
      } catch (error) {
        setError(error as Error);
      }
    };
    fetchProducts();
  }, []);

  return (
```
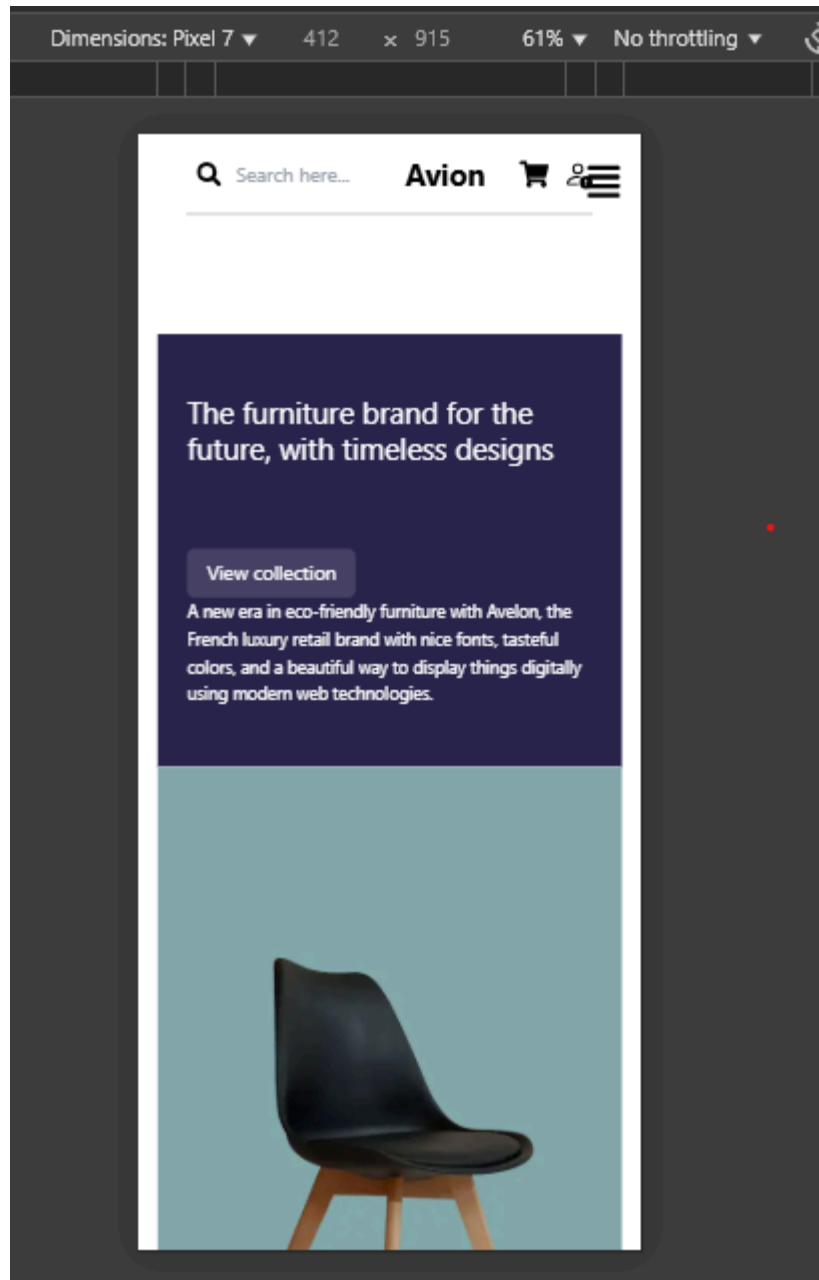
```
useEffect(() => {
  const fetchCategory = async () => {
    try {
      const categoryData = await client.fetch(
        `*[_type == "category" && _id == $categoryId]{
          _id,
          name,
          slug
        }`,
        { categoryId }
      );
      setCategory(categoryData[0]);
    } catch (error) {
      console.error("Error fetching category:", error);
    }
  };
  fetchCategory();
}, [categoryId]);

// Fetch products of the category based on category ID
useEffect(() => {
  const fetchProducts = async () => {
    try {
      const productsData = await client.fetch(
```
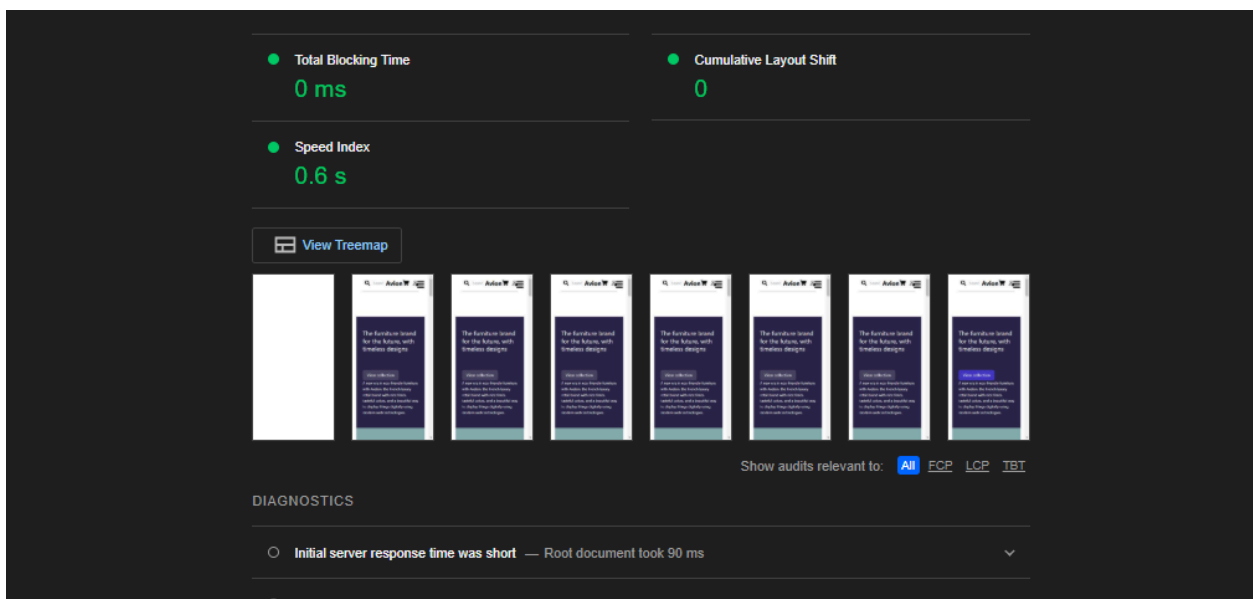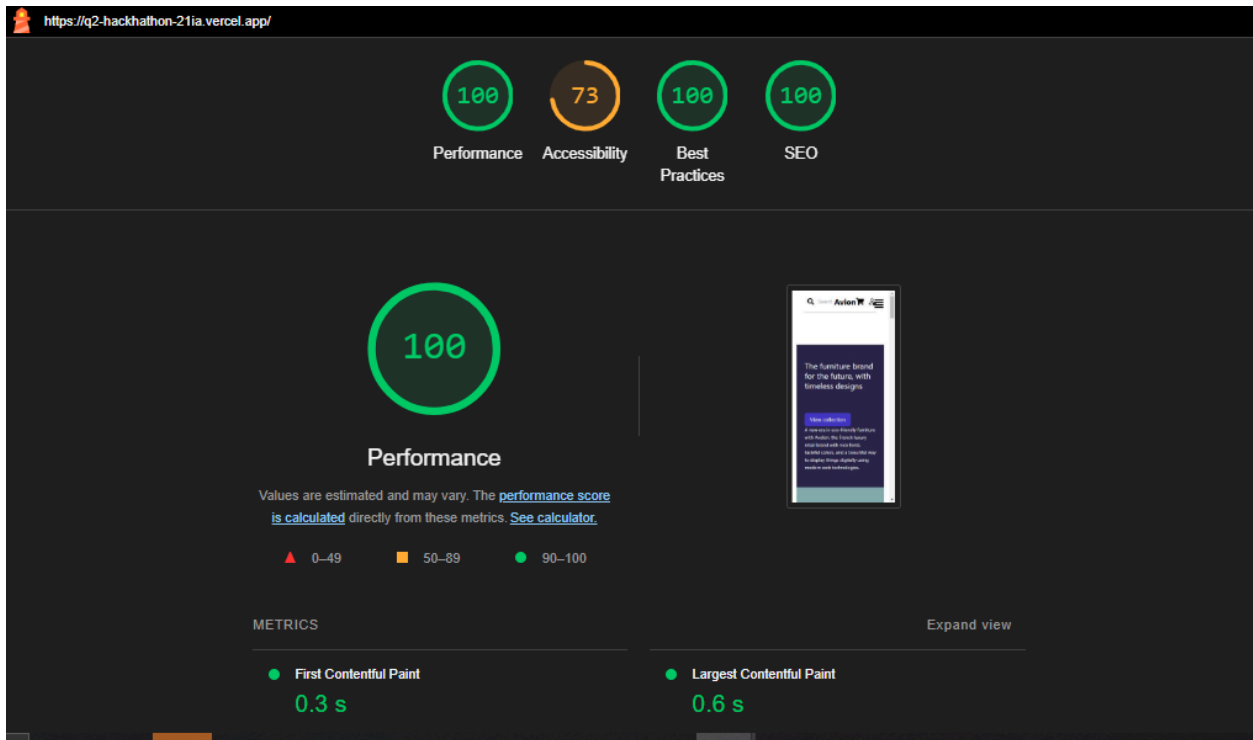
- **Cross-Browser and Device Testing:**

- Device Testing:

- **Lighthouse Performance**

100 **Performance**

73 **Accessibility**

100 **Best Practices**

100 **SEO**

100

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● **First Contentful Paint**
0.3 s

● **Largest Contentful Paint**
0.6 s

● **Total Blocking Time**
0 ms

● **Cumulative Layout Shift**
0

● **Speed Index**
0.6 s

⊞ View Treemap

Show audits relevant to: **All** FCP LCP TBT

DIAGNOSTICS

○ **Initial server response time was short** — Root document took 90 ms ⌄

○ **Avoids enormous network payloads** — Total size was 323 KiB ⌄

73

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

NAMES AND LABELS

⚠ Buttons do not have an accessible name ⌄

⚠ Links do not have a discernible name ⌄

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

100

Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks   ⌄

PASSED AUDITS (14)   Show

NOT APPLICABLE (3)   Show

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks   ⌄

PASSED AUDITS (14)   Show

NOT APPLICABLE (3)   Show

100

SEO

These checks ensure that your page is following basic search engine
optimization advice. There are many additional factors Lighthouse does not
score here that may affect your search ranking, including performance on
Core Web Vitals. Learn more about Google Search Essentials.

- **Checklist Day-05**

Checklist item	status
Functional Testing:	☑
Error Handling:	☑
Performance Optimization:	☑
Cross-Browser and Device Testing:	☑
Security Testing:	☑
Documentation:	☑
Final Review:	☑