

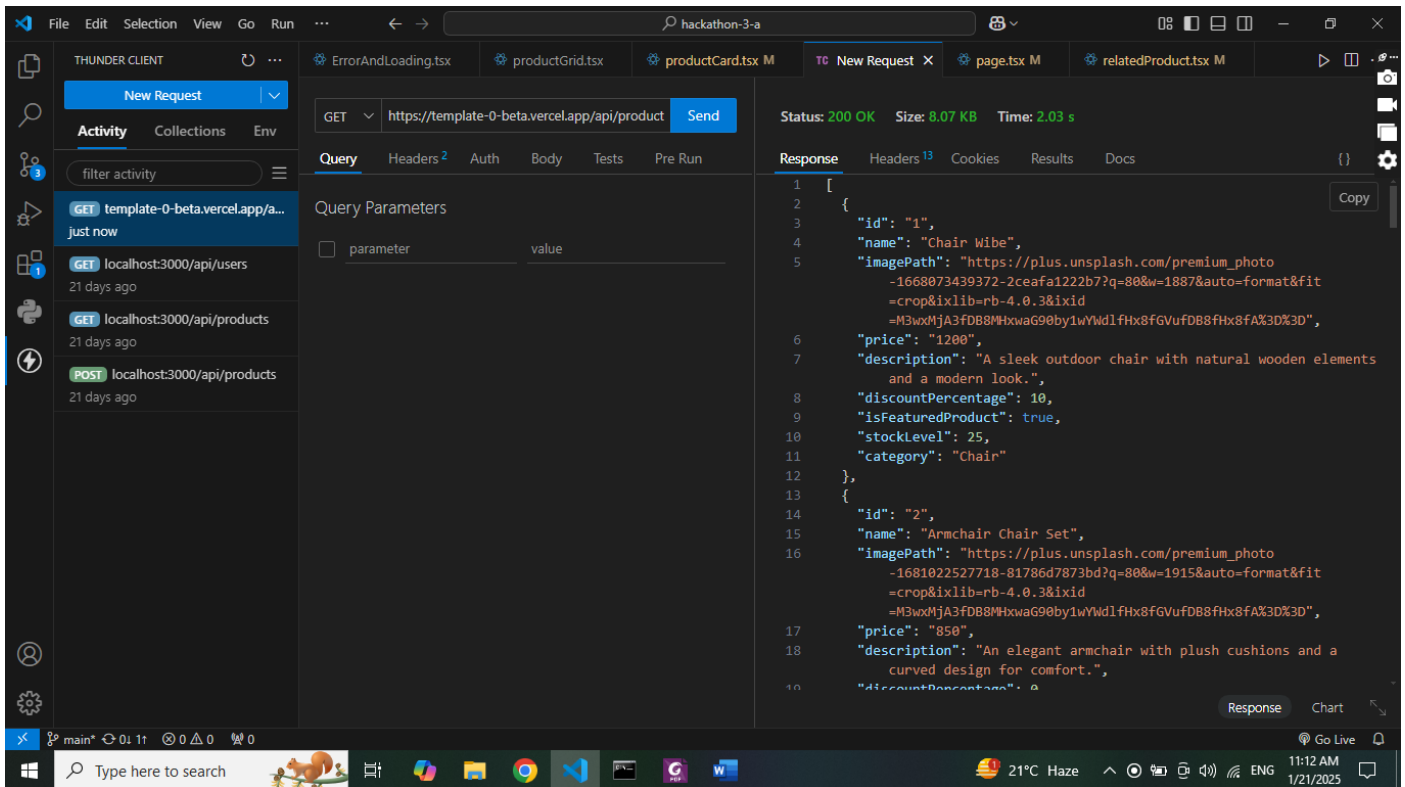
Name: Muhammad Hammad
Father Name: Muhammad Siddique
Roll No: 00413850
Days/Time: Saturday -07:00PM – 10:00PM
DAY 5 - TESTING, ERROR HANDLING, AND BACKEND
INTEGRATION REFINEMENT
Furniture E-commerce Marketplace

Objective:

Prepare the marketplace for deployment by testing backend integrations, optimizing performance, implementing error handling, and refining the user experience.

Step 1: Functional Testing

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	Low	-	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
TC003	Check "Add to Favorites" feature	Add product to favorites > Verify in favorites section	Product added to favorites section	Product added successfully	Passed	Medium	-	Works as expected
TC004	Validate cart operations	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	Functionality stable
TC005	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful
TC006	Verify dynamic routing for products	Click on a product > Verify individual product detail page loads successfully	Correct product detail page displayed	Product detail page loads correctly	Passed	High	-	No issues



Step 2: Error Handling Implementation

To ensure a smooth user experience, error handling is implemented in the product fetching logic:

- **Try-Catch for Error Handling:**
The try-catch block captures any API-related errors during the data-fetching process.
- **Fallback UI:**
If an error occurs, a user-friendly fallback message is displayed:
 - Title: "Unable to load products."
 - Message: "Please try again later."

This approach ensures the application remains functional and informative, even when backend issues arise.

Code Example:

```
let products: Products[] = [];
let hasError = false;
try {
  products = await client.fetch(query);
} catch (error) {
  console.error("Error fetching products:", error);
  hasError = true;
}
if (hasError) {
  return (
    <div className="text-center mt-5">
      <h2 className="text-xl font-bold">Unable to load products.</h2>
      <p className="text-gray-500">Please try again later.</p>
    </div>
  );
}
```

Step 3 : Image Performance Optimization

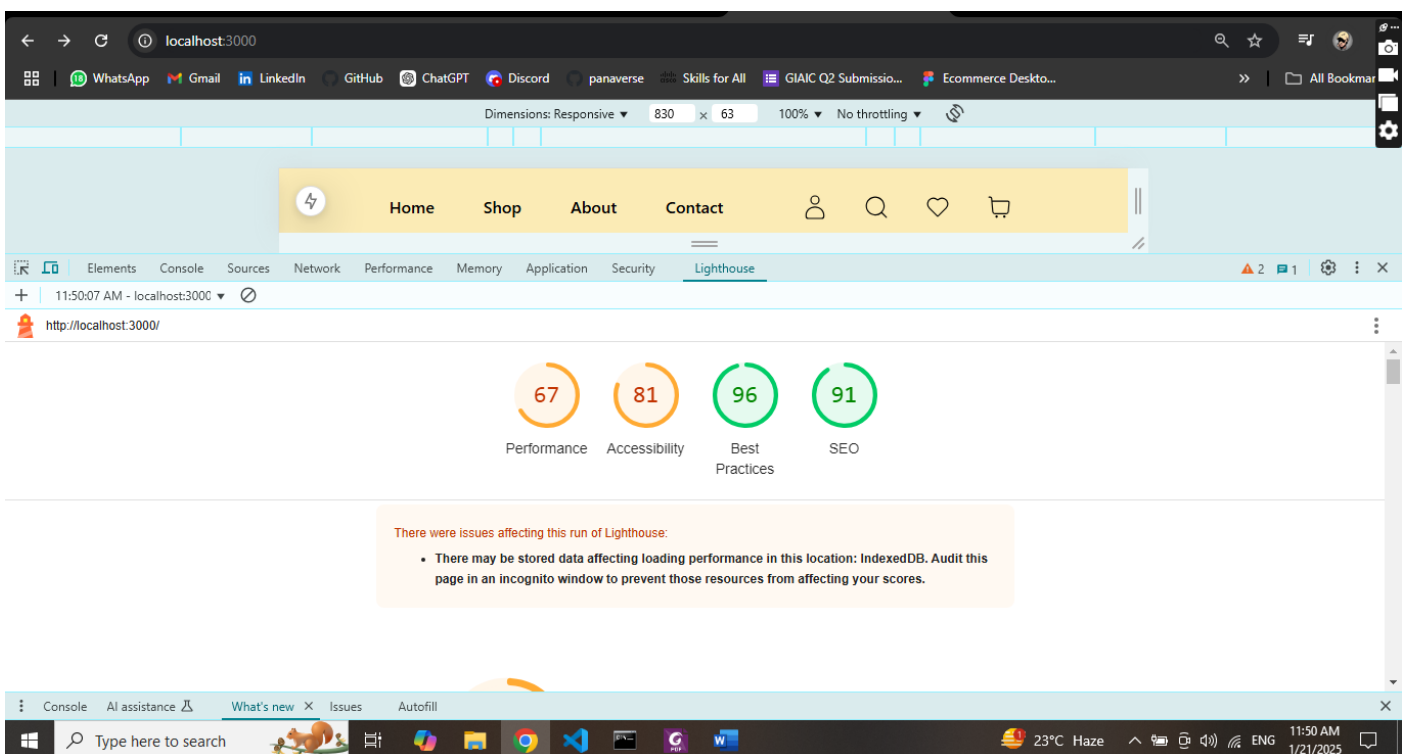
To enhance performance, images are optimized by:

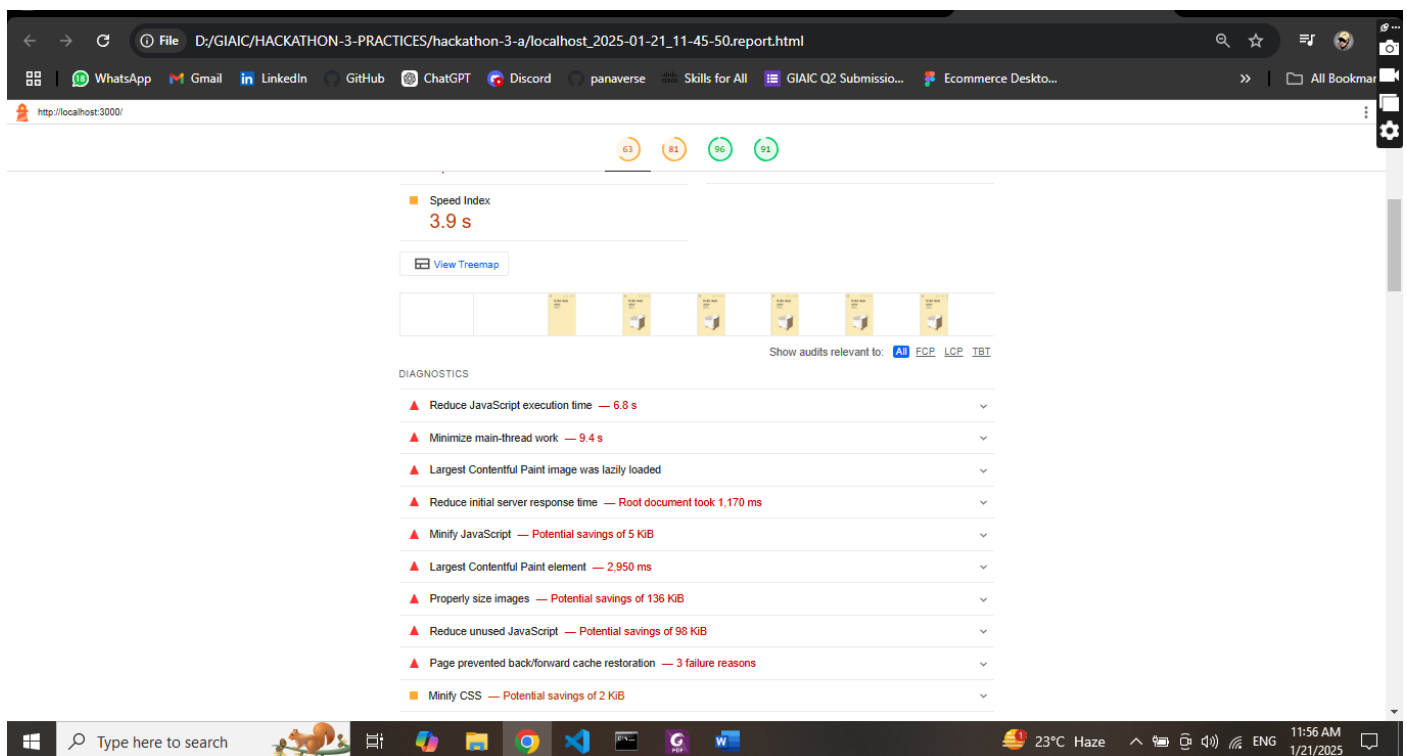
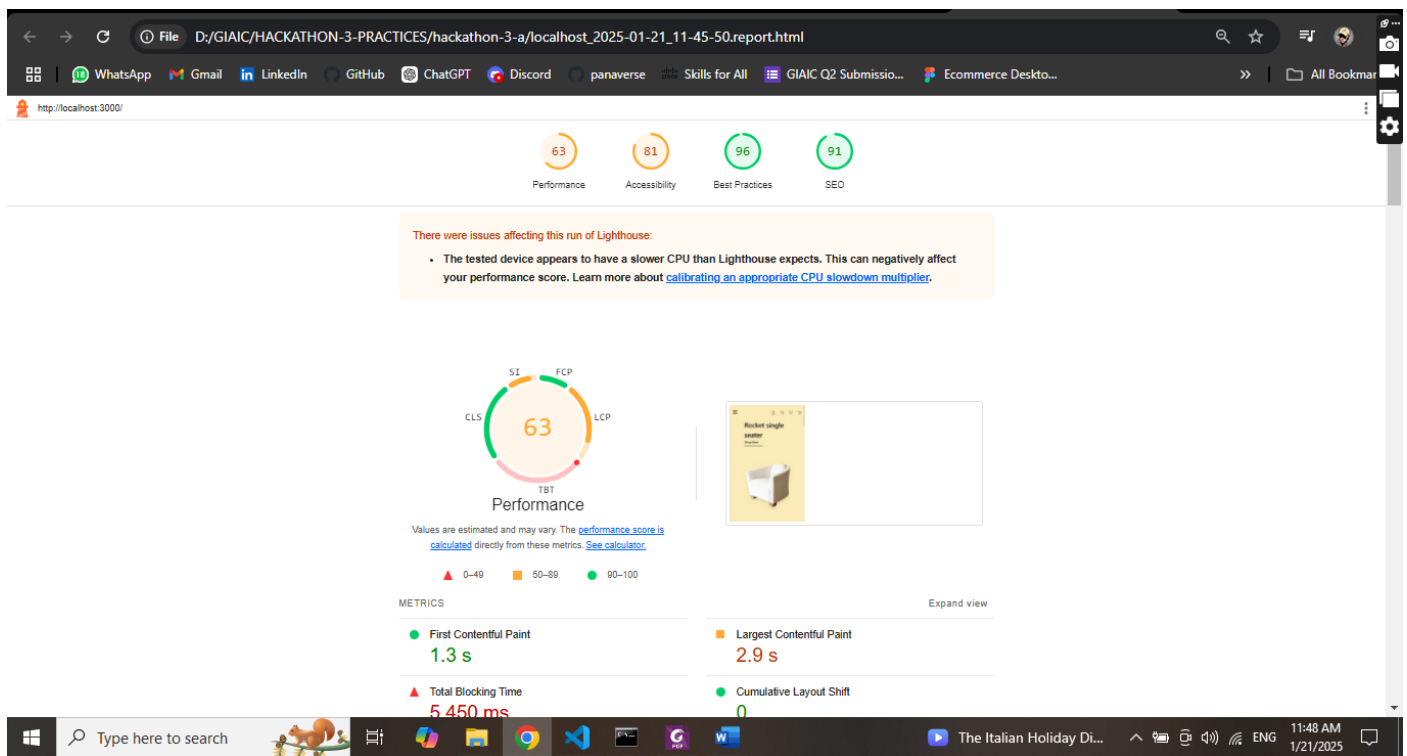
1. **Image Compression:** Using `quality={75}` for reduced file size without significant quality loss.
2. **Lazy Loading:** The `loading="lazy"` attribute defers image loading until they are visible in the viewport, speeding up the initial page load.

```
49     return {
50       <div key={_id} className="group15 flex flex-col items-start">
51         <div className="image-div mx-auto w-[200px] h-[150px] md:h-[180px] md:w-[300px] flex justify-center
52           items-center aspect-square overflow-hidden rounded-md">
53           <Link href={`~/shop/${_id}`}>
54             <Image
55               src={image}
56               width={287}
57               height={287}
58               alt={name}
59               className="object-cover"
60               quality={75} // Compress image
61               loading="lazy" // Lazy load
62             </Image>
63           </Link>
64         </div>
65       </div>
66     }
67   }
68 }
69
```

Analyze Performance with Lighthouse

```
D:\GIAIC\HACKATHON-3-PRACTICES\hackathon-3>npm install -g lighthouse
added 164 packages in 46s
11 packages are looking for funding
  run `npm fund` for details
D:\GIAIC\HACKATHON-3-PRACTICES\hackathon-3>npm fund
@typescript-eslint/eslint-plugin@8.20.0, @typescript-eslint/scope-manager@8.20.0, @typescript-eslint/types@8.20.0, @typescript-eslint/type-utils@8.20.0, @typescript-eslint/typescript-estree@8.20.0, @typescript-eslint/utils@8.20.0, @typescript-eslint/visitor-keys@8.20.0, @typescript-eslint/parser@8.20.0
https://opencollective.com/units/projects/eslint-import-resolver-ts
eslint-import-resolver-typescript@3.7.0
https://github.com/sponsors/antonk52
lilconfig@3.1.3
D:\GIAIC\HACKATHON-3-PRACTICES\hackathon-3>lighthouse http://localhost:3000 --view
```





Step 4: Cross-Browser and Device Testing

Developer Resources :

The screenshot shows a web browser at `localhost:3000/checkout`. The page contains a checkout form with the following fields:

- Country / region: Pakistan
- Street Address: House # B-99, Sector B Bhattai Colony K
- Town / City: Karachi
- Province: Select a province
- Zip Code: 75190
- Phone: (empty)

The Chrome DevTools Developer Resources panel is open, showing a list of resources. The 'Issues' tab is selected, displaying a table of resource status and URLs.

Status	URL	Initiator
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
failure	http://localhost:3000/_next/static/chunks/webpack.js?v=1737442238210	http://lo
failure	http://localhost:3000/_next/static/chunks/webpack.js?v=1737442238210	http://lo
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/__nextjs_source-map?filename=webpack-inten	
failure	http://localhost:3000/_next/static/chunks/webpack.js?v=1737442238210	http://lo
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/__nextjs_source-map?filename=webpack-inten	
failure	http://localhost:3000/_next/static/chunks/webpack.js?v=1737442238210	http://lo
success	/FwiQWRkIHRvIENhcnRciAGiFwiT3V0IG9mIFNK	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	
success	/json;charset=utf-8;base64,eyJ2ZXJzaW9uIjozLC	

Performance Monitor:

The screenshot shows a web browser at `localhost:3000/shop/D24vb8qimRcTY37EypHPU5`. The page displays a product card for the 'Rapson Thirty-Nine Sofa' with the following details:

- Size: MD, LG, XL
- Color: Black, Red, White, Blue
- Quantity: 3
- SKU: SS001
- Category: Sofa

The Chrome DevTools Performance Monitor panel is open, showing a timeline of performance metrics. The 'Performance' tab is selected, displaying a table of metrics and their values.

Metric	Value
CPU usage	14.2%
JS heap size	57.0 MB
DOM Nodes	1,902
JS event listeners	
Documents	
Document Frames	
Layouts / sec	
Style recalcs / sec	

Responsiveness:

