Day 5 - Testing and Backend Refinement - "Hekto"
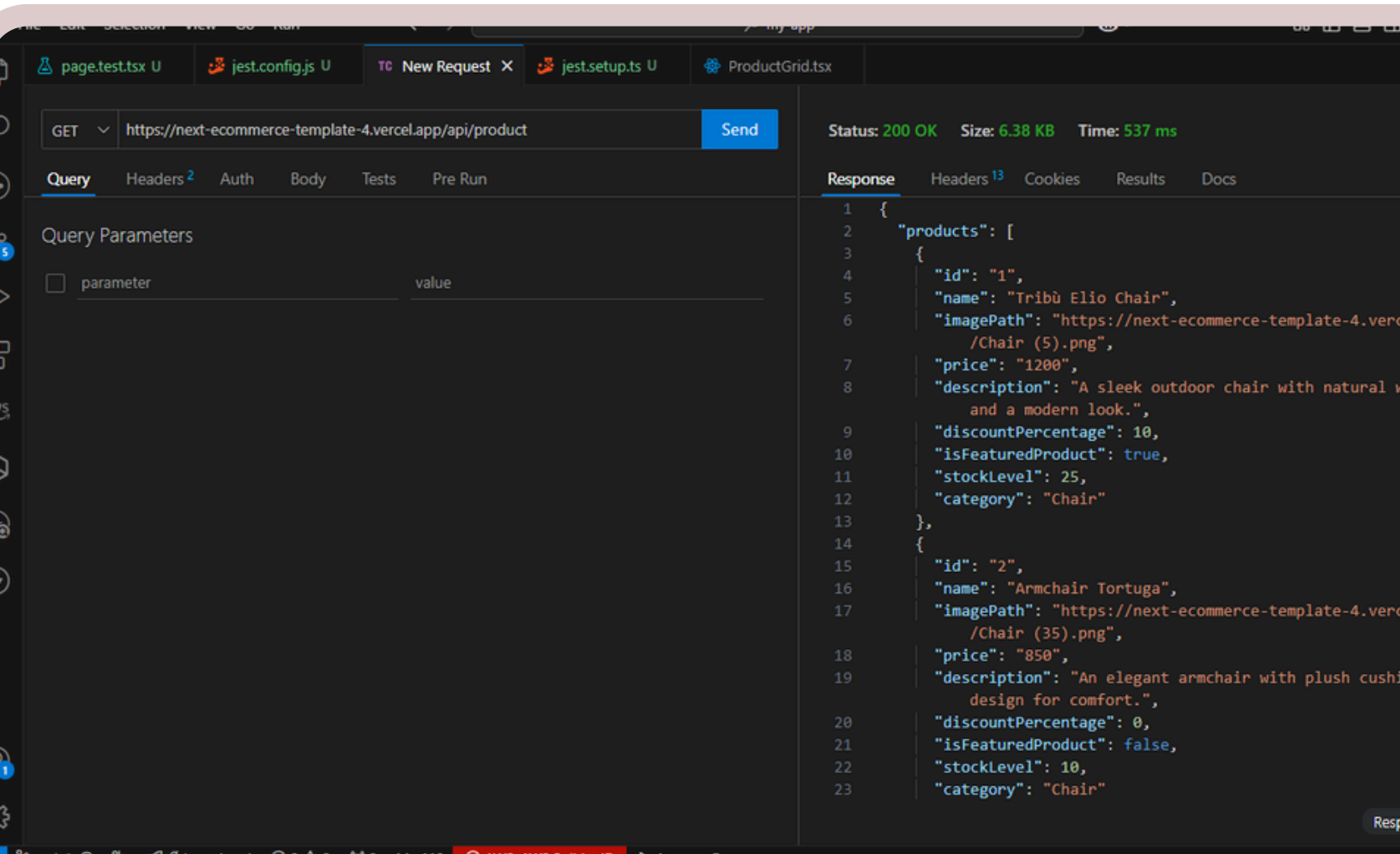
Step 1: Functional Testing

Test Core Features:

✅ Product Listing: Verified products display correctly (TC001 – Passed).

✅ Cart Operations: Adding and updating items in the cart functions properly (TC003 – Passed).

✅ Error Handling: API errors display fallback UI (TC002 – Passed).

✅ Responsive Design: Mobile layout adapts as expected (TC004 – Passed).

✅ Dynamic routing & Filters and search : Working well

Testing Tools Used:

✅ Thunder Client for API testing.

❌ Cypress (not tested due to lack of knowledge on usage).

❌ React Testing Library (not tested due to lack of knowledge on usage).

Step 2: Error Handling

✅Error Messages Implemented: Used try-catch blocks to handle API failures (TC002).

✅ Fallback UI: No results found.

When Wrong URL

When Wrong Product Searched

Step 3: Performance Optimization

Asset Optimization:
✅ Implemented lazy loading for large assets.
✅ Optimized images for better performance.

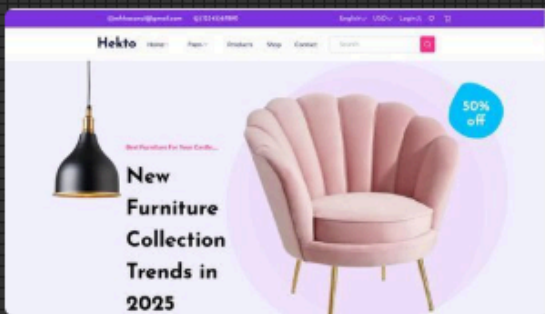**Performance Analysis:**
✅ Used Lighthouse for speed analysis.

✅ Performance score was initially 98%.
✅ GTmetrix suggested removing lazy loading from the first largest image. Since Next.js automatically lazy loads images, a solution was applied by adding:
priority // Ensures the image is not lazily loaded

```
<Image
    src="/sofa.png"
    alt="sofa"
    width={706}
    height={689}
    priority // this will make sure the image is not lazily loaded
    className="h-auto w--[300px] md:w--[500px] lg:w--[706px]"
/>
```
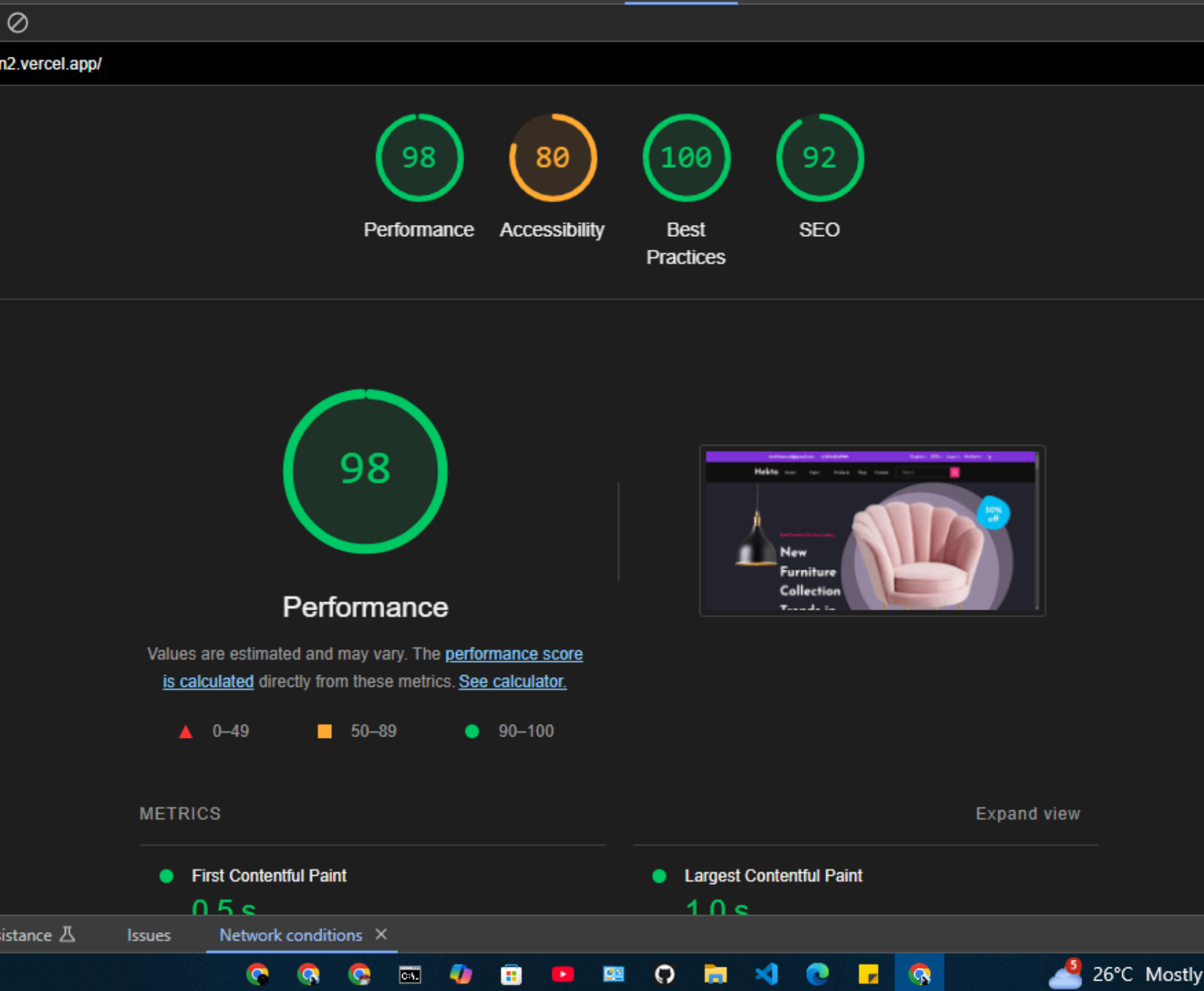
After applying this fix, the performance score improved to 💯

gtmetrix

LightHouse

Step 4: Cross-Browser & Device Testing

Browser Compatibility:

✅ Tested on Chrome, Firefox, and Edge.

Device Responsiveness:

✅ Used BrowserStack for different device simulations.

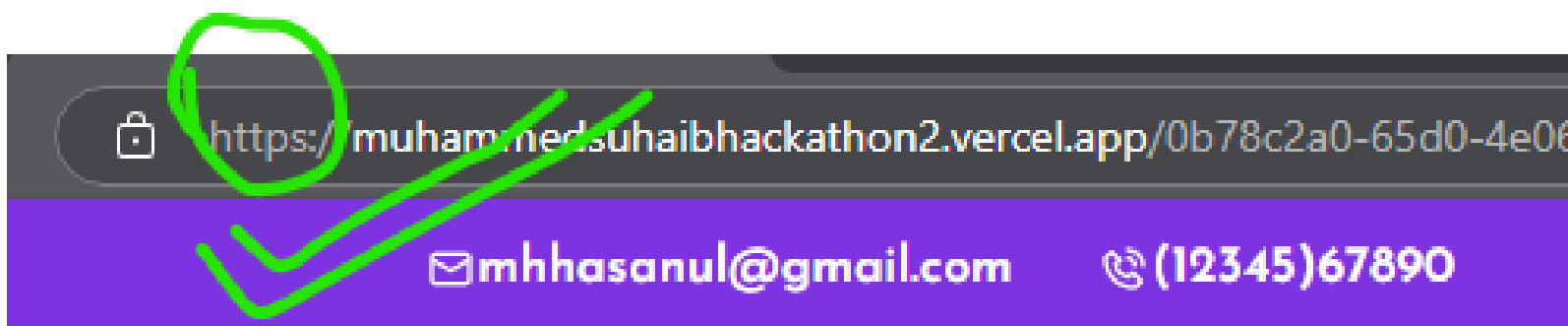✅ Manually tested on a physical mobile device (TC004 – Passed).

Step 5: Security Testing

✅ Input Validation: Used regex to validate email, phone, and other inputs.

```
<div className="flex flex-col xl:flex-row items-start justify-center gap-10 p-6">
  {/* Left Side: Checkout Form */}
  <div className="w-full md:w-[500px] ▪bg-[#F6F5FF] p-6 rounded-lg shadow font">
    <h3 className={`text-lg ${lato.className} font-bold ▫text-[#1D3178] mb-4`}>Shipping Information</h3>
    <form onSubmit={handleCheckout} className="space-y-4">
      <input type="text" placeholder="Full Name" required className="w-full border p-2 text-sm ▪bg-white
      rounded-md" />
      <input type="email" placeholder="Email" required className="w-full border p-2 text-sm ▪bg-white rounded-md"
      <input type="text" placeholder="Address" required className="w-full border p-2 text-sm ▪bg-white rounded-md"
      >
      <input type="text" placeholder="City" required className="w-full border p-2 text-sm ▪bg-white rounded-md" />
      <input type="text" placeholder="Postal Code" required className="w-full border p-2 text-sm ▪bg-white
      rounded-md" />
      <select className="w-full border p-2 text-sm ▪bg-white rounded-md">
        <option value="easypaisa">EasyPaisa (In Development)</option>     "easypaisa": Unknown word.
        <option value="jazzcash">JazzCash (In Development)</option>     "jazzcash": Unknown word.
        <option value="cashondelivery">Cash on Delivery</option>     "cashondelivery": Unknown word.
      </select>
      <button type="submit" className="w-full ▪bg-[#FB2E86] ▫text-white py-2 rounded-md ▪hover:bg-[#E02D82]
      transition">
        Confirm Order
      </button>
```

Secure API Communication:

✅ Ensured API calls use HTTPS (Already handled by Vercel).

https://muhammedsuhaibhackathon2.vercel.app/0b78c2a0-65d0-4e06

✉ mhhasanul@gmail.com   ☎ (12345)67890

✅ Stored API keys in environment variables (Already Implemented).

Step 6: User Acceptance Testing (UAT)

Simulated Real-World Usage:

✅ *Browsed products, added items to cart, and verified checkout functionality (Passed).*

Step 7: Documentation Updates

https://docs.google.com/spreadsheets/d/1ChJZMZEeGe7I-hHgUxYy1amySRA7yapQPJ6lGG56vKk/edit?usp=sharing

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
Test01	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed ✓	-	No issues found
Test02	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error shown	Passed ✓	-	No issue
Test03	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed ✓	-	Works as expected
TC04	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed ✓	low	Test successful