

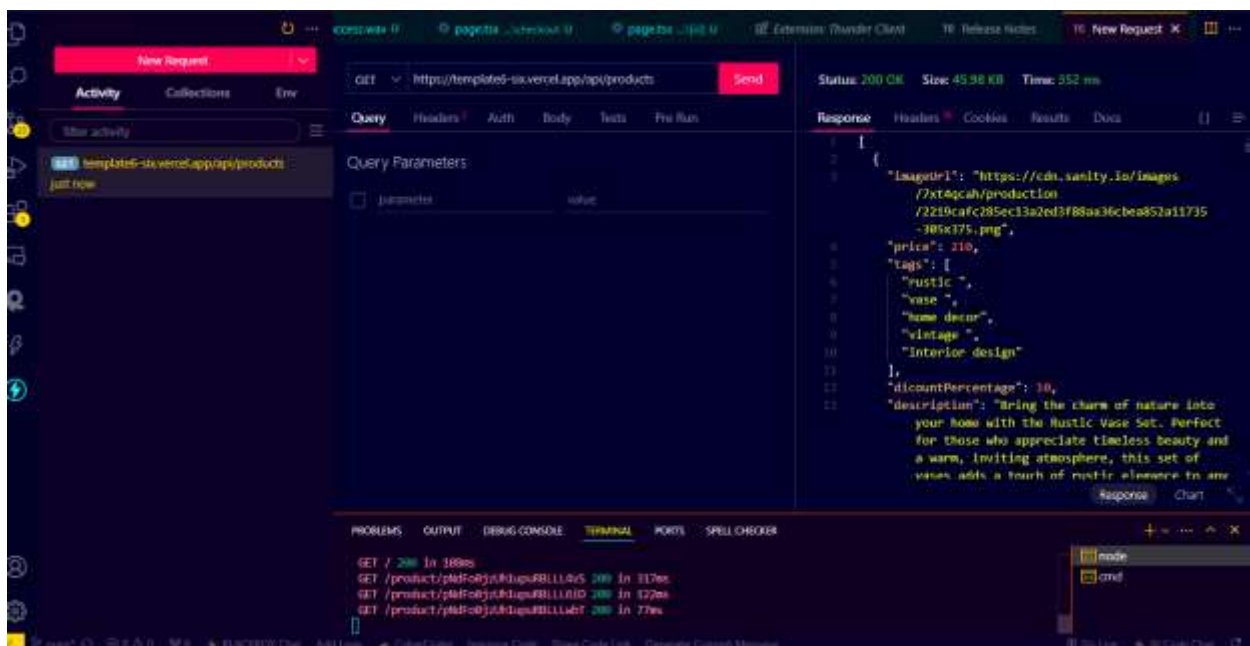
# HACKATHON DAY 5 TESTING, ERROR HANDLING AND BACKEND INTEGRATION

Step 1: functional testing

Step2: error handling

Step:3 postman api

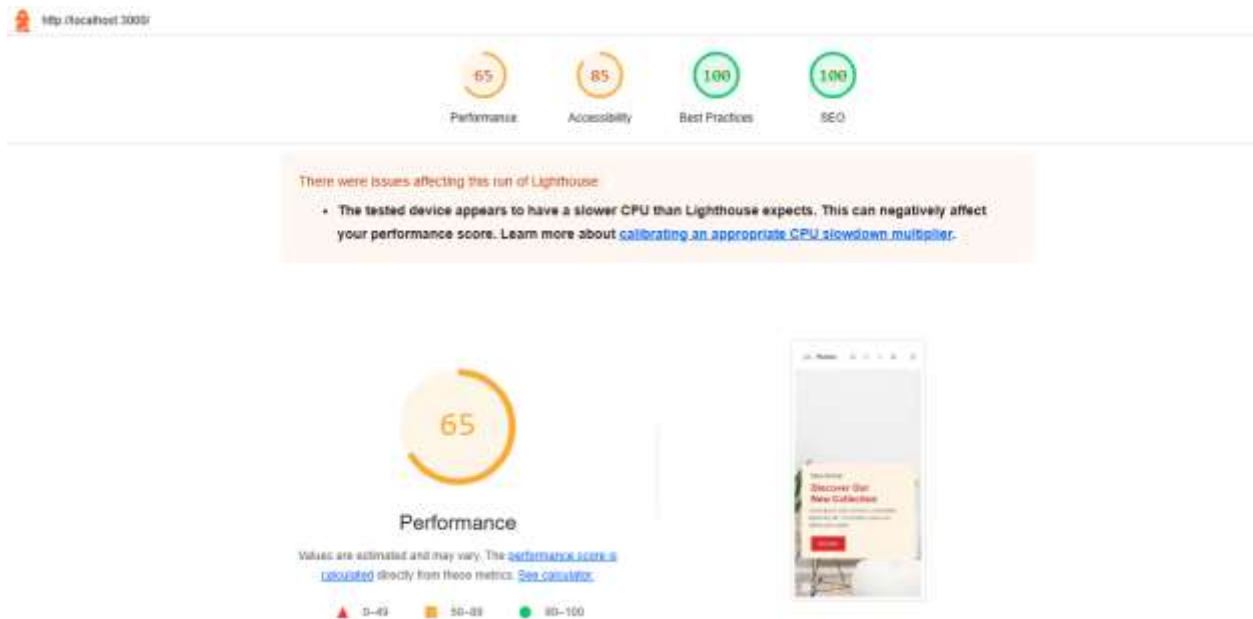
Api: <https://template6-six.vercel.app/api/products>



Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
2	TC001	Validate	Open product	Products	Products	Pass	Low	-

		product listing page	page > Verify products	displayed correctly	displayed correctly	d			issues found
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
4	TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	Works as expected
5	TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful

## Step04: Performance optimization





## Best Practices

### TRUST AND SAFETY

- |   |   |
|---|---|
| <input type="radio"/> Ensure CSP is effective against XSS attacks | ✓ |
| <input type="radio"/> Use a strong HSTS policy                    | ✓ |
| <input type="radio"/> Ensure proper origin isolation with COOP    | ✓ |

### GENERAL

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Missing source maps for large first-party JavaScript | ✓ |
|--|---|

## DIAGNOSTICS

▲ Minimize main-thread work — 17.7 s	▼
▲ Reduce JavaScript execution time — 7.8 s	▼
▲ Largest Contentful Paint element — 2,830 ms	▼
▲ Reduce unused JavaScript — Potential savings of 166 KiB	▼
▲ Page prevented back/forward cache restoration — 3 failure reasons	▼
■ Image elements do not have explicit width and height	▼
■ Minify JavaScript — Potential savings of 5 KiB	▼
■ Serve images in next-gen formats — Potential savings of 1,718 KiB	▼
■ Properly size images — Potential savings of 3,161 KiB	▼
■ Defer offscreen images — Potential savings of 2,174 KiB	▼
■ Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB	▼
■ Avoid enormous network payloads — Total size was 6,633 KiB	▼



### 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

## NAMES AND LABELS

### ▲ Buttons do not have an accessible 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

### ▲ Background and foreground colors do not have a sufficient contrast ratio.

These are opportunities to improve the legibility of your content.

## BEST PRACTICES

### ▲ Touch targets do not have sufficient size or spacing.

These items highlight common accessibility best practices.



### 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](#).

#### ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

##### ○ Structured data is valid

▼

Put these additional validators on your site to check additional SEO best practices.

#### PASSED AUDITS (8)

Show

#### NOT APPLICABLE (2)

Show



- ☐ Ensure CSP is effective against XSS attacks
- ☐ Use a strong HSTS policy
- ☐ Ensure proper origin isolation with COOP

Missing source maps for large first-party JavaScript

Single page session

Using Chromium 131.0.0.0 with cli

