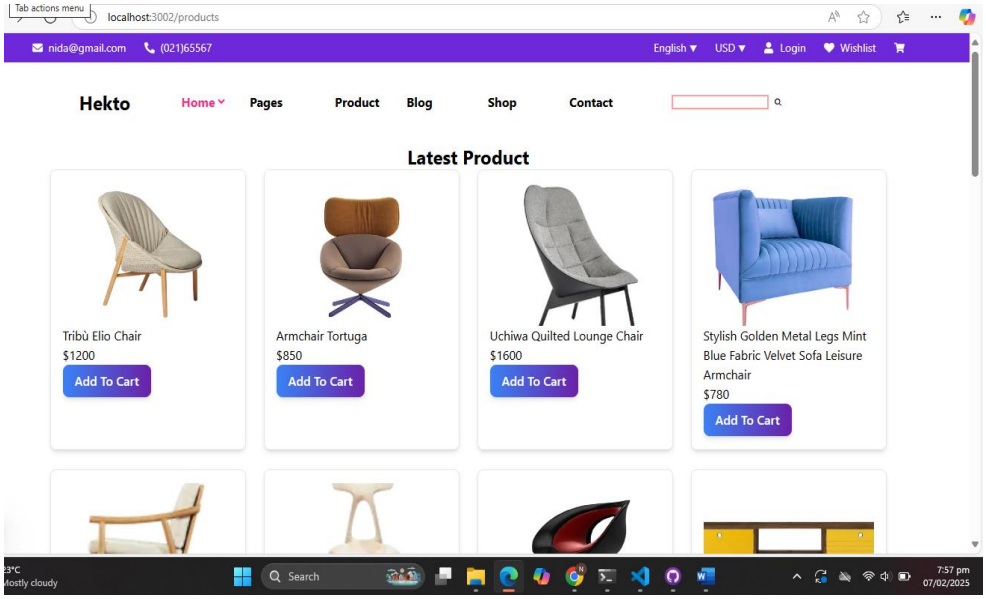
HACKATHON 03

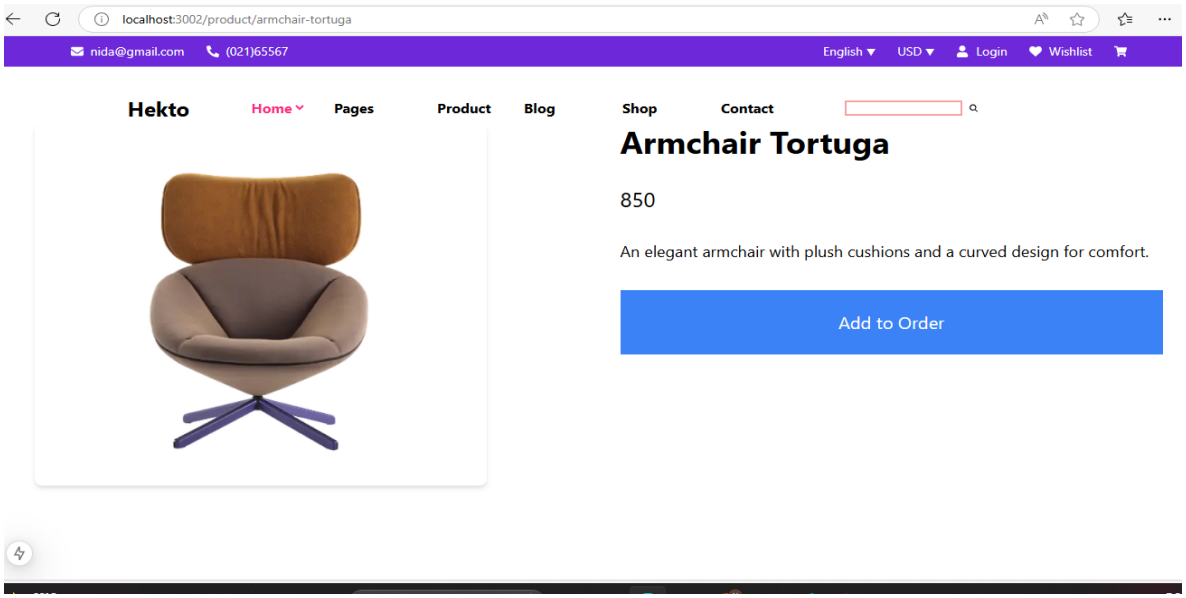
Market Place Builder

DAY 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

Step 01: Functional Testing

- Dynamic Routing
- I successfully implemented dynamic routing in my project. 🚀 The system efficiently selects optimal paths for data transmission. Overall, dynamic routing worked perfectly, ensuring smooth operations.





Screen recording

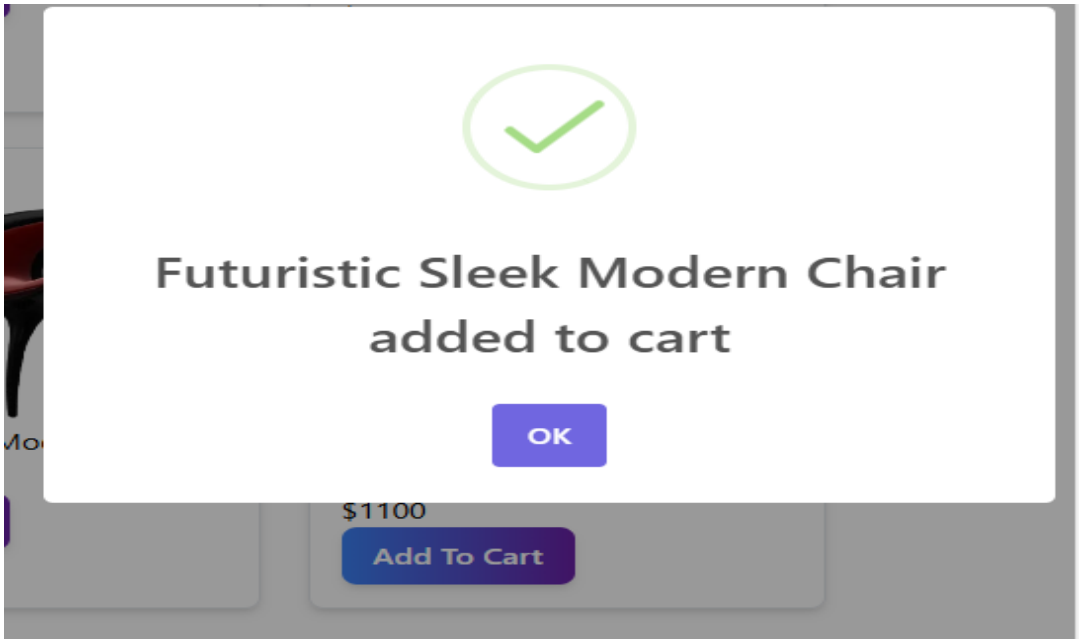
Create Next App
and 3 more pages -

- ## Add To Cart

  I successfully implemented the Add to Cart functionality, and it's working smoothly! Users can now add items effortlessly, ensuring a seamless shopping experience.

Screen shot and recording available here

Create Next App
and 3 more pages -



Step 02: Error Handling

While developing my project that involved **dynamic routing and API data fetching**, I encountered several challenges related to error handling. These issues included **failed API requests, incorrect dynamic route parameters, and handling non-existent resources**. To ensure a smooth user experience and prevent application crashes, I implemented effective error-handling strategies.

Identified Errors and Their Resolutions

1. API Data Fetching Errors

Issue:

While fetching data from an external API, I faced situations where the API request failed due to **network issues, incorrect endpoints, or unauthorized access**. This led to application crashes or empty data rendering.

Resolution:

- I implemented **proper response validation** to check if the API request was successful before processing the data.
- In case of errors like **404 (Not Found) or 500 (Server Error),** I displayed meaningful error messages instead of breaking the UI.
- I used **try-catch blocks** and **error handling functions** to gracefully manage unexpected API failures and alert users accordingly.
- Implemented **loading states** to improve user experience while fetching data dynamically.

Step 03: Performance Testing

https://next-ecommerce-template-4.vercel.app/api/product

✓ API Data display well there is no issue to GET data this is the screenshot of display

Luxury Flower Shell Sofa Chair
Price: $2500
A luxurious shell-shaped chair with gold brass metal legs.

Discount: 0%
Stock Level: 2
Category: Sofa
Cantilever Chair
Price: $780
A modern cantilever chair with a unique floating effect.

Latest Product

Luxury Flower Shell Sofa Chair
Price: $2500
A luxurious shell-shaped chair with gold brass metal legs.
Discount: 0%
Stock Level: 2
Category: Sofa

Cantilever Chair
Price: $780
A modern cantilever chair with a unique floating effect.
Discount: 15%
Stock Level: 5
Category: Chair

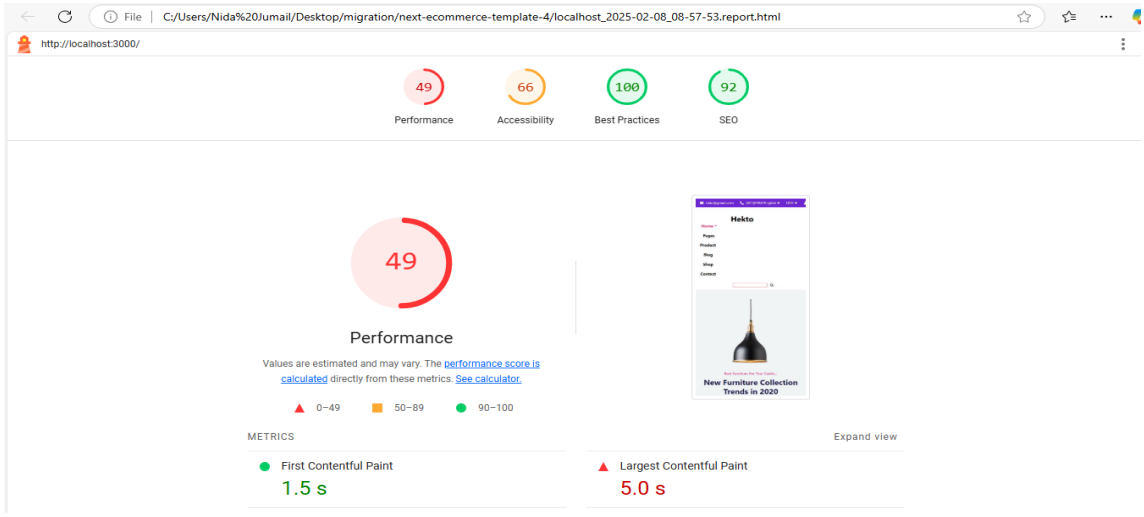
Nordic Net Red Chair
Price: $320
An acrylic dining chair with a sleek and minimalist Nordic design.
Discount: 10%
Stock Level: 20
Category: Chair

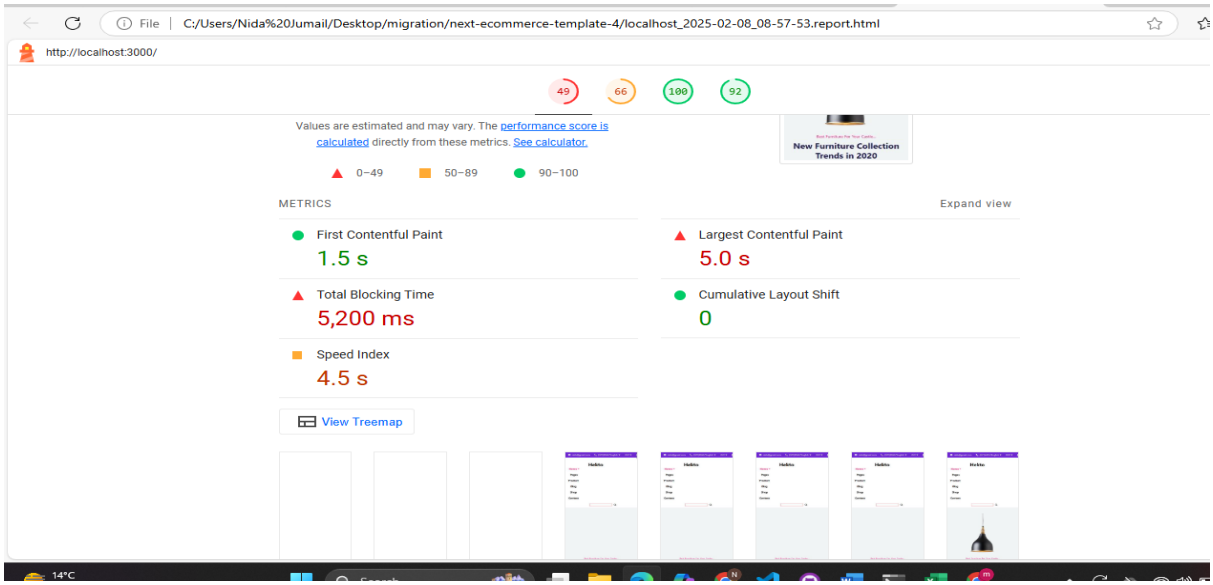
Sobuy Blue Folding Chair Wooden Padded
Price: $120
A foldable wooden chair with a padded seat for extra comfort.
Discount: 12%
Stock Level: 30

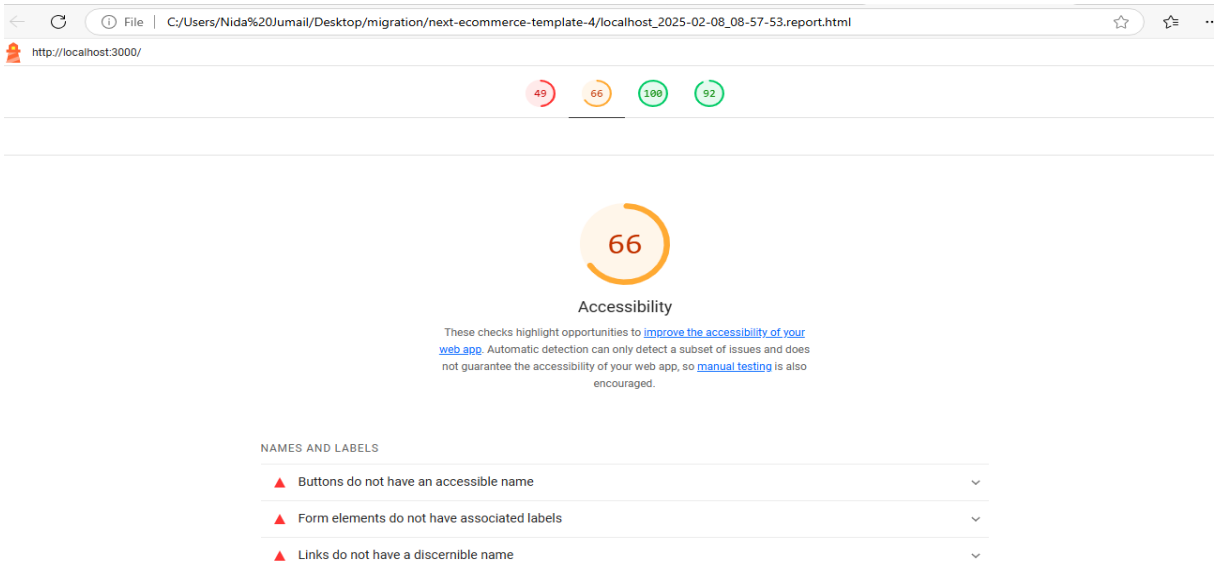
Lighthouse Tool:

npm install -g lighthouse

Lighthouse http://localhost:3000 - -view



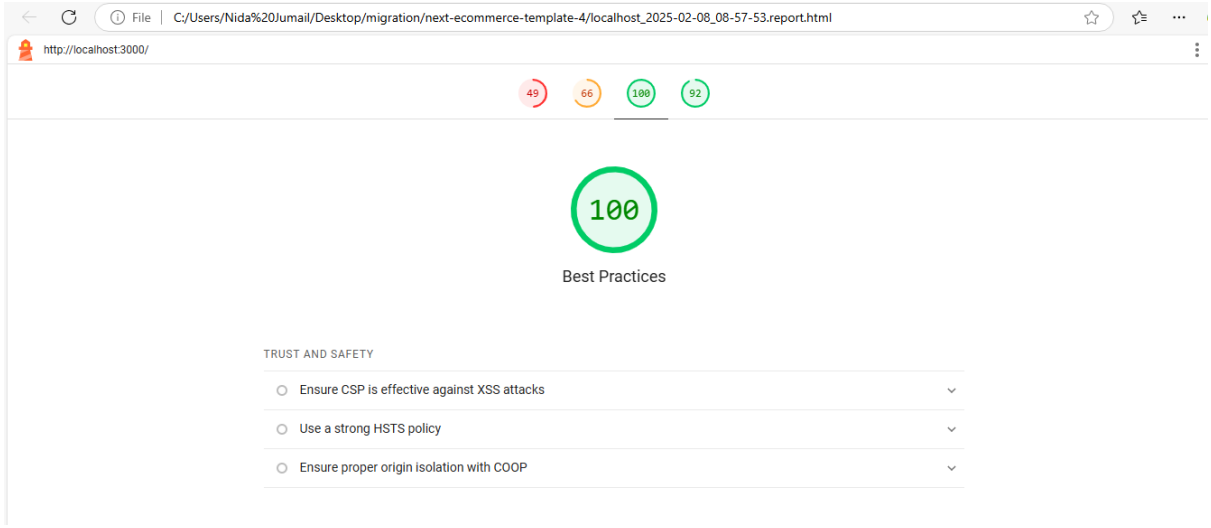




 Best Practices (100/100) – Your website follows industry standards, ensuring security, modern coding techniques, and best development practices.

 SEO (92/100) – Your site is well-optimized for search engines, which helps improve visibility and reach more customers.

 Good Accessibility (66/100) – Your website is fairly accessible, meaning it's usable for a wide range of users, including those with disabilities.



Trust and Safety

Ensure CSP is effective against XSS attacks

Use a strong HSTS policy

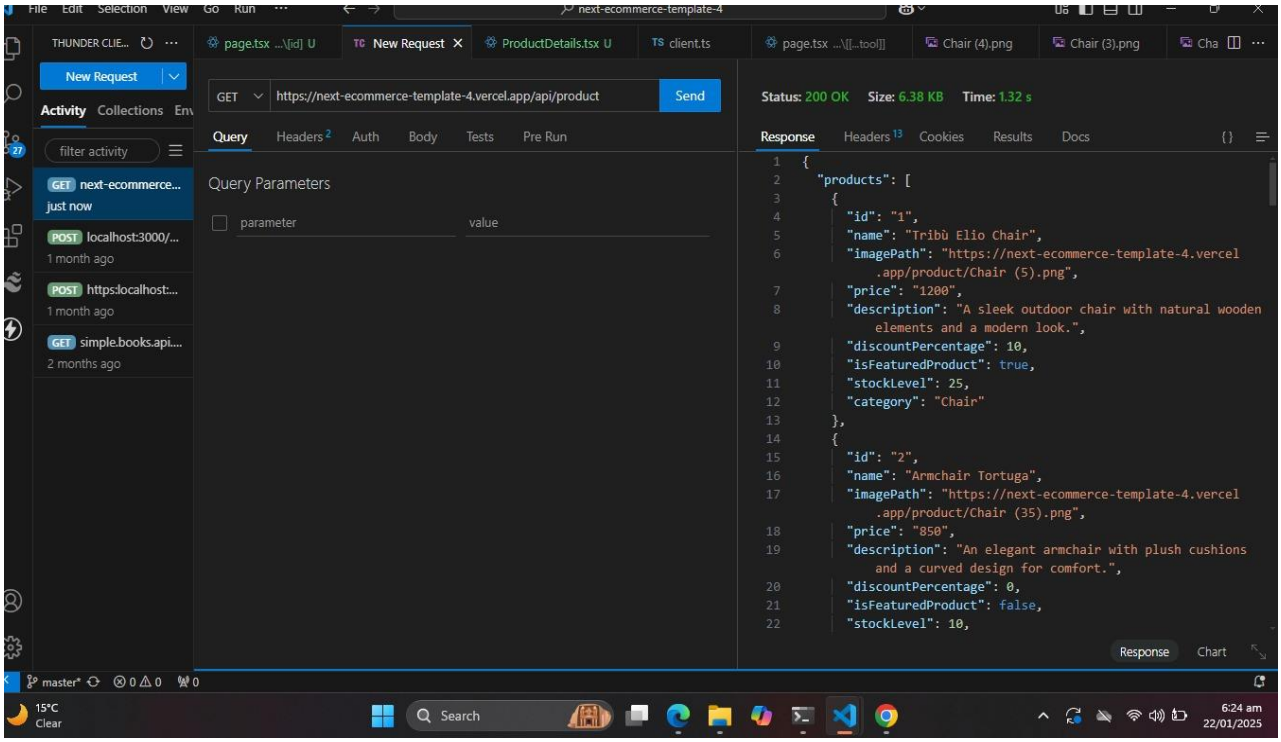
Ensure proper origin isolation with COOP

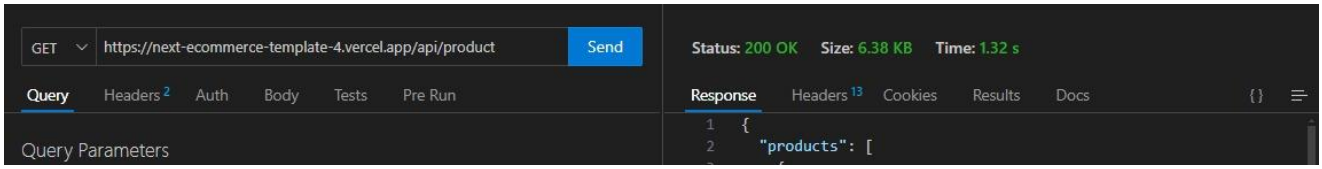
csv link

https://drive.google.com/file/d/1ZpYkuaNMQfO8nciAy73ajp37vG4y8O02/view?usp=drive_link

1	Test Case ID	Test Case	Test Steps	Expected Result	Actual Result	Status	Severity Level	Asg To	Remarks
2	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	-	No 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

✓ Tools: Thunder Client (I used Thunder Client to get data)

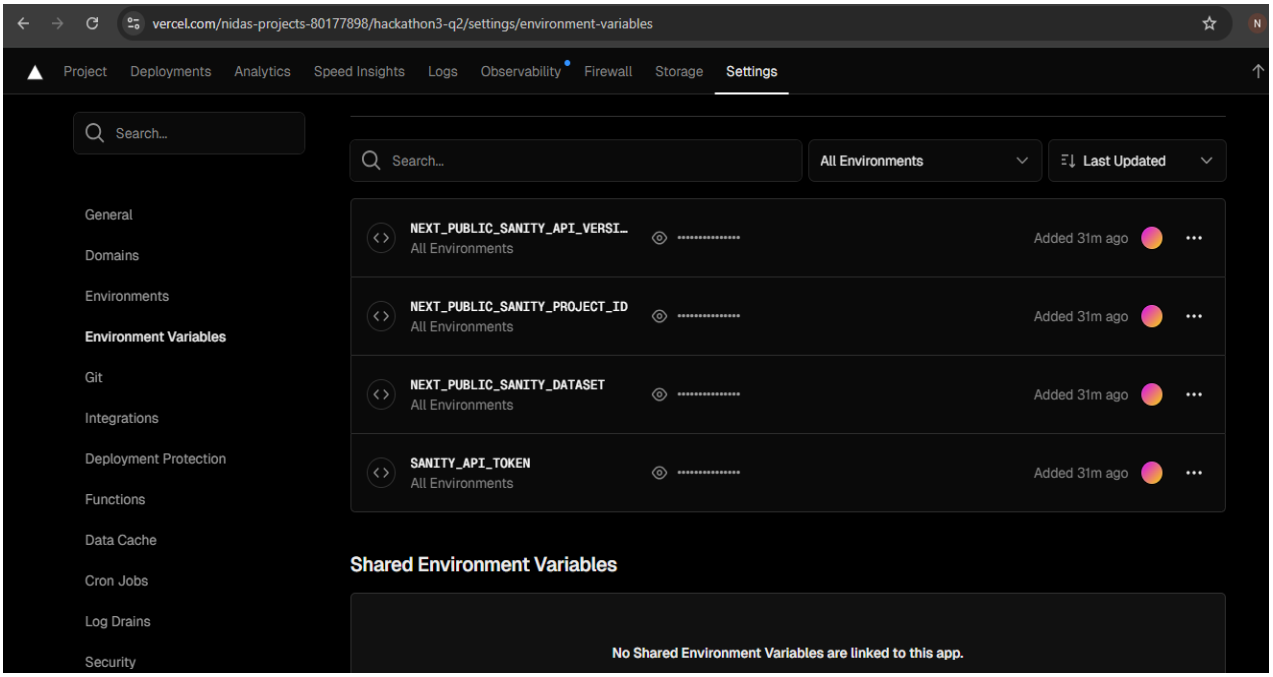




Hosting platform variables

- I used **Vercel** for hosting my e-commerce website, and the deployment process was smooth and efficient. Its **automatic builds, fast CDN, and seamless integration with Next.js** made it the perfect choice for my project.
  Create env file on hosting

Environment Variables



Staging environment setting

- Functional Testing

  Cart

  Price listing

  Order placement

  Search functionality

- Add Read.me file collectively 3-7 day

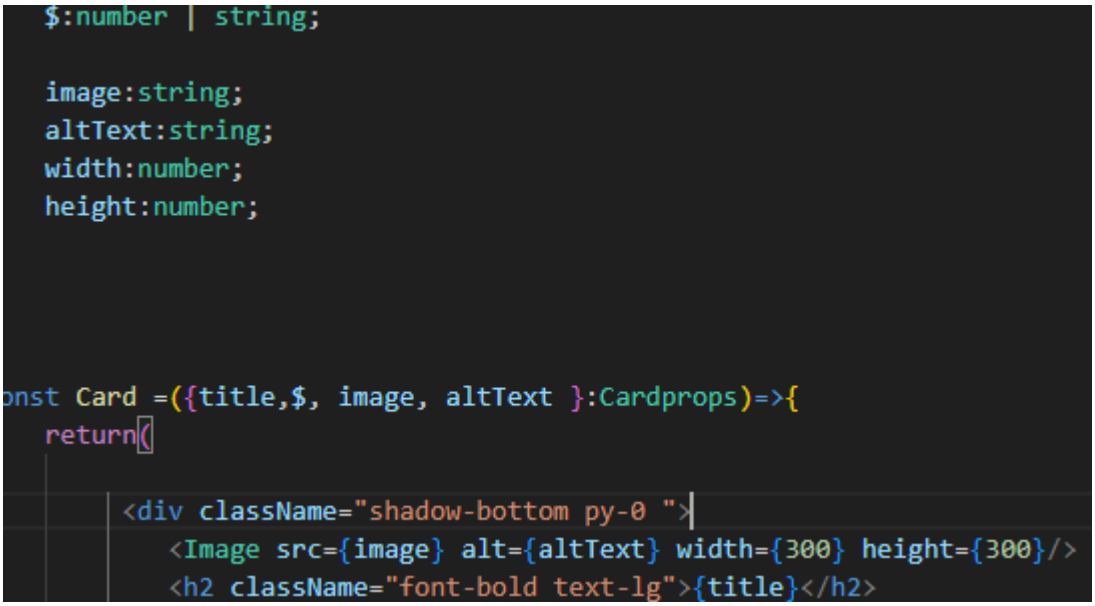
- READ.Me link dashboard

  https://raw.githubusercontent.com/nidajumail/admin-dashboard/refs/heads/main/README.md
- READ.Me UI/UX Project

https://raw.githubusercontent.com/nidajumail/hackathon3-q2/refs/heads/main/README.md

- Performance optimization

<Image/>



- **Sentry**

  const { withSentryConfig } = require('@sentry/nextjs');

  const moduleExports = {

  };

```
const SentryWebpackPluginOptions = {
silent: true,};
module.exports = withSentryConfig(moduleExports,
SentryWebpackPluginOptions) ;
```

- ## Backup Disaster Recovery

  Here we are using vercel which has automatic feature of backup

Share your experience

- We faced multiple hurdles, especially in **dynamic routing, data fetching, and optimizing performance** using tools like **Lighthouse**.
- **Deployment Issues:** Resolved environment-specific errors . by properly handling .env files.

Additional Comments;

- Working on this e-commerce website has been an incredible learning journey. Since this was my first project of this scale, I faced many challenges but also gained valuable experience along the way.
- Optimized images using Next.js <Image> component and **lazy loading**.
- Learned about protecting API routes and handling authentication tokens properly.
- Error handling (404 pages, API failures, loading states) improves UX.
-
- Faced several deployment issues while pushing the project to Vercel.
-
- Learned how to properly configure environment variables (.env.local) to avoid errors.