Day 6: Deployment Preparation and Staging Environment Setup

1. Staging Environment Setup

- Hosting Platform: [Insert chosen platform name, e.g., Vercel]
- **Repository Connection**: Connected GitHub repository to [Vercel/Netlify]. The project was linked via the platform's dashboard.
- Build and Deployment Settings:
 - Build settings configured as per platform requirements.
 - Ensured that the correct build commands (e.g., npm run build) are specified.
 - Deployment pipeline verified to trigger automatically after code push to main branch.

2. Environment Variables

Secure Variables:

.env file created with sensitive information:

Ex:NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
API_KEY=your_api_key

- Uploading Variables to Platform:
 - Environment variables added securely via the platform's dashboard (e.g., Vercel's Environment Variables section).

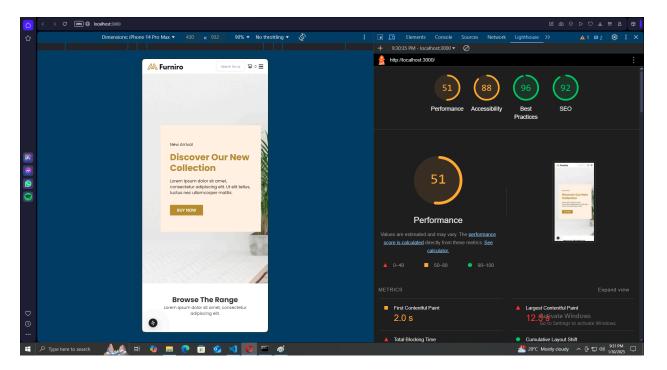
3. Staging Deployment

- Deployment Confirmation:
 - o Deployment completed successfully without errors.
 - Site loaded correctly in the staging environment.
 - Any issues faced during deployment were resolved (e.g., misconfigured environment variables).

4. Staging Environment Testing

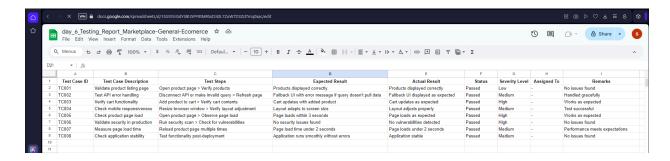
Types of Testing Conducted:

- 1. Functional Testing: Verified features such as:
 - Product listing.
 - Search functionality.
 - Cart operations.
- 2. Performance Testing:
 - Used Lighthouse to check site performance.
 - Results showed total performance score of 51% which is quite low the issue will be resolved shortly.



• Test Case Reporting:

Documented all test cases in the CSV format.



Conclusion

- The staging environment has been successfully set up and tested.
- Deployment to the staging environment is complete, and all functionalities have been verified.
- Test case reports and performance results have been documented.