DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP-[Comfy]

Roll-No:118068

Overview

The deployment of the Comfy e-commerce website was a structured process aimed at transitioning the application from development to a live environment. Each step was carefully planned and executed to ensure the website's reliability, security, and functionality. This report details the key stages involved in the deployment process, from setting up the hosting platform to rigorous testing in the staging environment.

- **Deliverables:**
- Hosting Platform Setup
- Configuring Environment Variables
- Deploying to Staging
- Staging Environment Testing

Step 1: Hosting Platform Setup:

The deployment journey began with setting up a hosting platform.

- Created a GitHub repository named NextJS_Market_Place_Builder_Hackathon to house the source code for marketplace builder.
- Connected the repository to the deployment platform, Vercel, for seamless synchronization

This integration ensured seamless synchronization between the codebase and the hosting environment, enabling efficient deployment workflows.

Step 2: Configuring Environment Variables:

Environment variables are essential for securing sensitive information such as API keys and tokens:

- Created a .env.local file within the root of the project directory.
- Included the required variables:

NEXT_PUBLIC_SANITY_PROJECT_ID	“my projext id”
NEXT_PUBLIC_SANITY_DATASET	“production”
SANITY_API_TOKEN	“my sanity token”

Then deploy these envoirnment variables securely to the hosting platform’s dashboard using Vercel’s interface.

Step 3: Deploying to Staging:

- With the hosting setup and environment variables configured, I deployed the application to a staging environment.
- Built the application from the connected GitHub repository. The build process involved compiling the Next.js application, optimizing assets such as JavaScript and CSS files, and pre-rendering static pages to enhance performance.
- Monitored the build process to ensure it completed without errors. Any issues encountered during the build, such as missing dependencies or syntax errors, were promptly resolved.
- After deploying application to the Vercel , I added its URL in sanity CMS CORS origin.
- Verified basic functionality in the staging environment, including:
    - Browsing products.
    - Adding items to the cart.
    - Completing the checkout process.
    - Search products.

This step ensured that the application was ready for further testing and eventual production deployment.

Step 4: Staging Environment Testing:

The staging environment served as a replica of the production environment, allowing for comprehensive testing:

Testing Types:

1.Functional Testing:

Objective: Validate the core functionalities of the e-commerce marketplace to ensure they work as intended.

Test Cases:

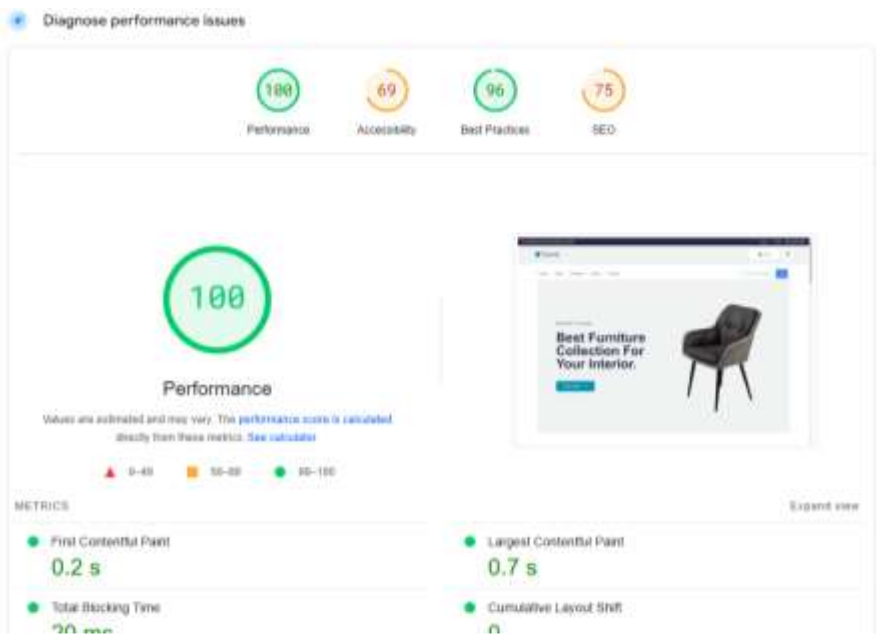
- **Product Listing:**
  - Ensure products are displayed correctly with accurate details (name, price, image).
  - Validate pagination to show the correct number of products per page.
- **Filters and Search**:
  - Verify accurate results based on user inputs for product names and categories.
  - Test multiple filter combinations for consistent behavior.
- **Cart Operations**:
  - Add items to the cart and verify updates in cart totals.
  - Update item quantities and remove items to ensure changes reflect correctly.

2.Performance Testing:

Check the performance of website using tools like Lighthouse and GTmetrix , to improve the performance of website which enhances user experience.

- Tested the website's performance and accessibility using Lighthouse, which provided insights into areas like page load speed, responsiveness, and SEO.
- Evaluated the website's functionality and compatibility using GTmetrix to identify and resolve potential issues across various platforms.

Lighthouse Testing Report For Comfy:



Diagnose performance issues

100	69	96	75
Performance	Accessibility	Best Practices	SEO

100

Performance

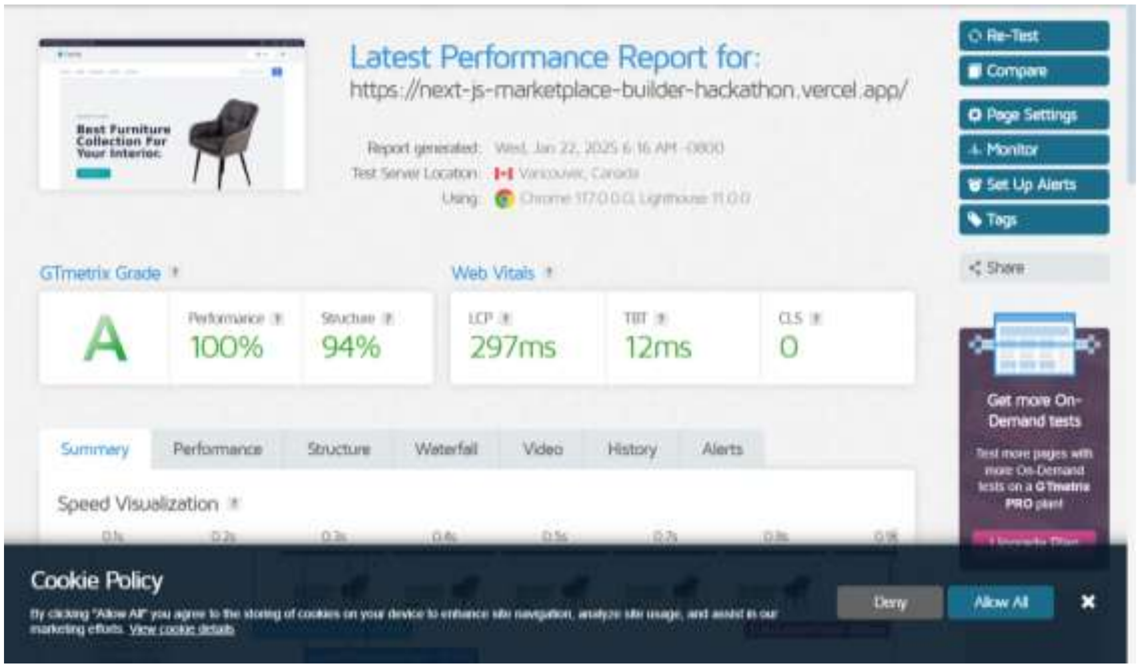
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS Expand view

● First Contentful Paint 0.2 s	● Largest Contentful Paint 0.7 s
● Total Blocking Time 20 ms	● Cumulative Layout Shift 0

GTmetrix Testing Report For Comfy:



Latest Performance Report for:
https://next-js-marketplace-builder-hackathon.vercel.app/

Report generated: Wed, Jan 22, 2025 6:16 AM -0800
Test Server Location: Vancouver, Canada
Using: Chrome 117.0.0.0, Lighthouse 11.0.0

GTmetrix Grade

	Performance	Structure
A	100%	94%

Web Vitals

LCP	TBT	CLS
297ms	12ms	0

Get more On-Demand tests

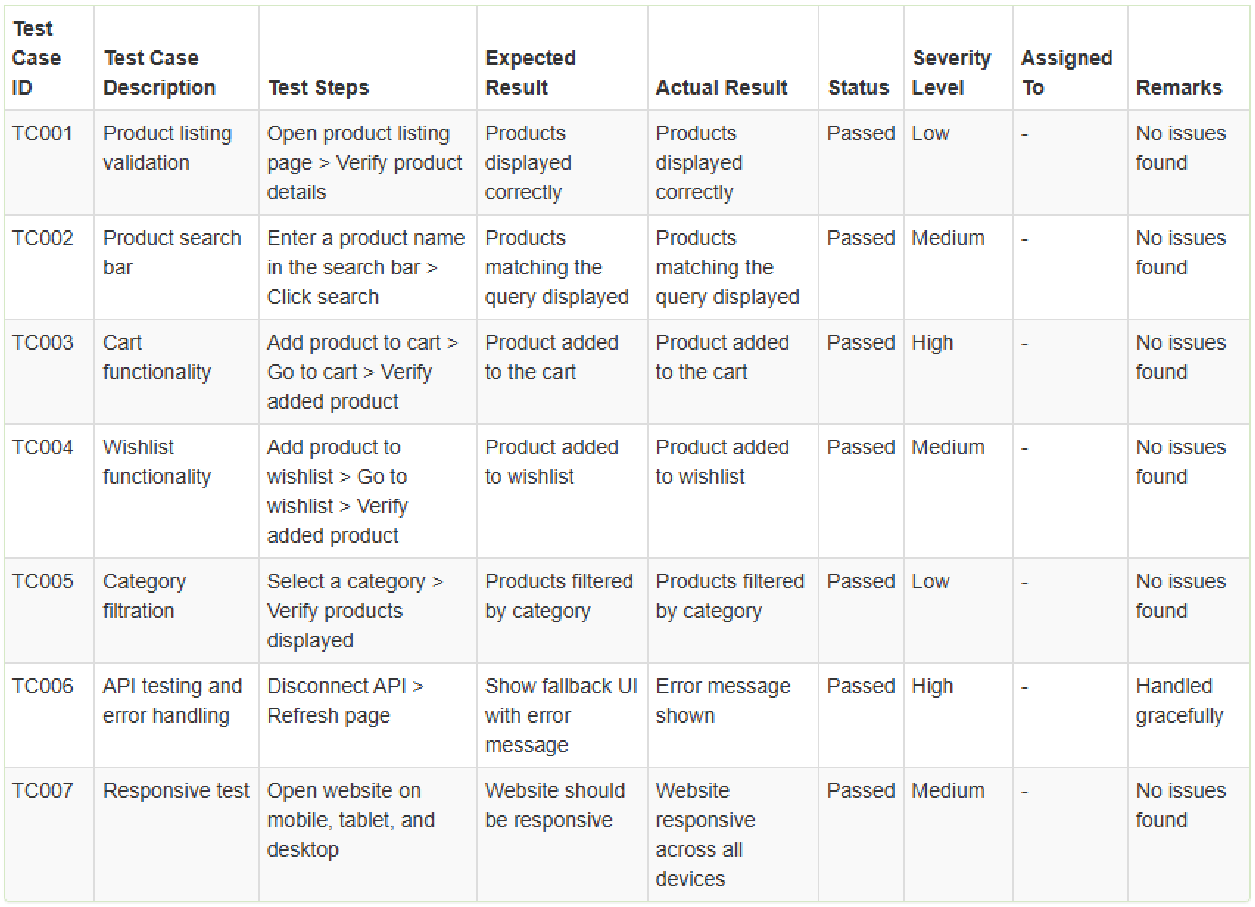
Test more pages with more On-Demand tests on a GTmetrix PRO plan!

Cookie Policy

By clicking "Allow All" you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. View cookie details

Test Case Reporting:

The tests validate core functionalities such as product listing, search, cart, and wishlist operations. They also assess advanced features like category filtering, API error handling, and responsiveness across devices. All tests passed successfully, with no issues found except for API error handling, which was gracefully managed. The detailed report ensures the system adheres to functional and user experience standards.



Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Product listing validation	Open product listing page > Verify product details	Products displayed correctly	Products displayed correctly	Passed	Low	-	No issues found
TC002	Product search bar	Enter a product name in the search bar > Click search	Products matching the query displayed	Products matching the query displayed	Passed	Medium	-	No issues found
TC003	Cart functionality	Add product to cart > Go to cart > Verify added product	Product added to the cart	Product added to the cart	Passed	High	-	No issues found
TC004	Wishlist functionality	Add product to wishlist > Go to wishlist > Verify added product	Product added to wishlist	Product added to wishlist	Passed	Medium	-	No issues found
TC005	Category filtration	Select a category > Verify products displayed	Products filtered by category	Products filtered by category	Passed	Low	-	No issues found
TC006	API testing and error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	High	-	Handled gracefully
TC007	Responsive test	Open website on mobile, tablet, and desktop	Website should be responsive	Website responsive across all devices	Passed	Medium	-	No issues found