Day 6: Deployment Preparation and Staging Environment Setup

Objective:
Day 6 focused on preparing the marketplace application for deployment by setting up a staging environment, configuring the hosting platform, and ensuring the application is ready for a customer-facing release. This step builds upon the testing and optimization efforts from Day 5, ensuring the marketplace runs smoothly in an environment that closely mirrors production.

Deployment Strategy Planning

1. Choosing a Hosting Platform: Vercel (Recommended)

Vercel was selected as the hosting platform due to its seamless integration with Next.js and its automated deployment capabilities.

Key benefits:

Easy Integration: Native support for Next.js.
Automatic Deployment: Projects deploy directly from GitHub with minimal setup.
Scalability: Efficiently handles traffic fluctuations.
Serverless Functions: Enables backend logic execution without server management.

2. Finalizing Application's Interaction with Backend Services

Before deployment, it was crucial to verify seamless integration with backend services like Sanity CMS and third-party APIs.

Sanity CMS: Ensured correct content fetching by configuring the Sanity client and API keys properly.
Third-Party APIs: Verified API calls work securely, ensuring that sensitive data (e.g., API keys) is managed through environment variables.

Professional Environment Types:

1. TRN (Training)

Purpose: Used for onboarding new team members and practice.
Key Feature: Helps users get familiar with the system without impacting active environments.

2. DEV (Development)

Purpose: Dedicated environment for developers to write and test code locally.
Key Feature: Supports iterative coding and debugging without affecting production systems.

3. SIT (System Integration Testing)

Purpose: Validates the integration between different systems and components.
Key Feature: Ensures seamless communication and compatibility between subsystems,

4. UAT (User Acceptance Testing)

Purpose: Allows stakeholders to test application functionality and validate that it meets business requirements.
Key Feature: Ensures the system is ready for production deployment by aligning with user expectations.

5. PROD (Production)

Purpose: The live, customer-facing environment where the application operates for end-users.

Key Feature: Ensures high availability, performance, and security for real-world usage.

6. DR (Disaster Recovery)

Purpose: Acts as a backup environment for critical situations such as system failures or disasters.
Key Feature: Enables quick recovery and minimizes downtime in emergencies.

Key Areas of Focus:

1. Deployment Strategy Planning

Deployed the application on Vercel for staging and production.
Integrated with Sanity CMS for dynamic content using tokens and dataset IDs2.

2.Environment Variable Configuration.

Stored sensitive data (API keys, tokens) in .env.local file.
Configured environment variables securely in Vercel Dashboard for deployment.

3. Staging Environment Setup

Deployed the application to Vercel and verified successful deployment.
Checked content fetching from Sanity CMS.

Staging Environment Testing

4. Staging Environment Testing

Conducted Cypress functional tests, Postman API validation, and Lighthouse performance tests.

Ensured security with HTTPS, proper data handling, and verified responsiveness across devices.

5. Documentation Updates

Created a README.md file with all deployment instructions, configurations, and test results.
Included all reports in the GitHub repository.

Steps for Implementation

Step 1: Hosting Platform Setup

Chosen Platform:

Vercel was selected for quick and easy deployment.

Connect Repository:

Successfully connected the GitHub repository to Vercel for automatic deployments.

Configured build settings and added the necessary scripts for deployment in the Vercel dashboard.

Step 2: Configure Environment Variables

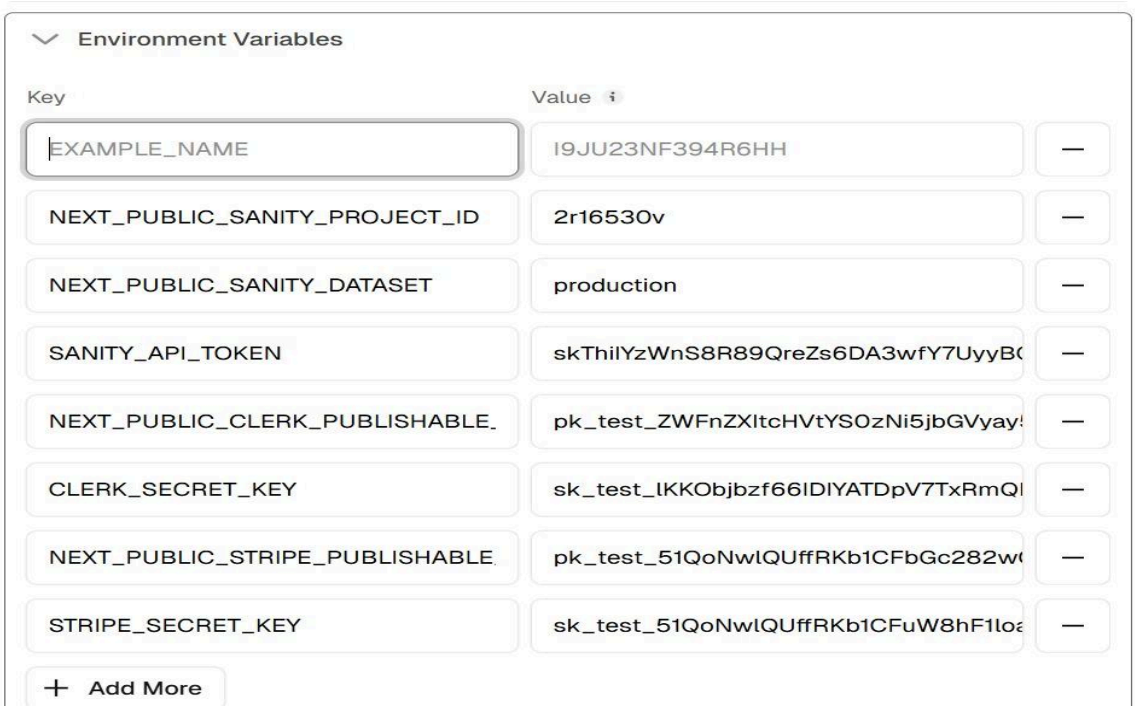
Create.env.local File:

Created the env.local file to store sensitive data like API keys and tokens.

Example:.

```
NEXT_PUBLIC_SANITY_PROJECT_ID
NEXT_PUBLIC_SANITY_DATASET
SANITY_API_TOKEN
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY
CLERK_SECRET_KEY
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY
STRIPE_SECRET_KEY
```

. **Configure Environment Variables in the Hosting Platform for Secure Deployment:** After setting up.env files in the development environment, configure these variables in Vervel to ensure secure deployment. This step is crucial for maintaining data security in production.



Upload Variables to Vercel:

Uploaded the environment variables to Vercel using the platform's dashboard for secure handling.

Step 3: Deploy to Staging

Deploy Application:

Deployed the application to Vercel's staging environment for testing.

Validate Deployment:

. Ensured the deployment build completed without errors.
.Verified that the application was loading correctly, and all content was fetched properly from Sanity CMS.

Step 4: Staging Environment Testing

1. Testing Types

Functional Testing:

Verified the following features:
.Product Listing: Ensured all products were listed correctly,
.Test API Error Handling: Verify the system's response to invalid API requests and ensure proper error messages are displayed.
.Check Cart Functionality: Ensure items can be added, removed, and updated in the cart, and verify price calculations.
.Test Form Validation: Check input fields for correct validation, error messages, and required field restrictions.
Verify Order Placement: Ensure the order process works smoothly by testing payment, order confirmation, and notifications.

Verify HTTPS Connection: Ensure all pages load securely over HTTPS, checking for valid SSL certificates and mixed content issues.

Performance Testing:

.Used Lighthouse and GTmetrix to analyze the performance, speed, and responsiveness of the application.
.Ensured the application was optimized for various devices, screen sizes, and network conditions to deliver a smooth user experience.

Security Testing:

.Validated input fields to ensure they were protected from vulnerabilities such as SQL injection and other malicious attacks.
.Ensured HTTPS was enabled for secure communication between the client and server.
.Verified that sensitive data, including API keys and user credentials, was transmitted securely and stored safely to avoid any data breaches.

2. Test Case Reporting.

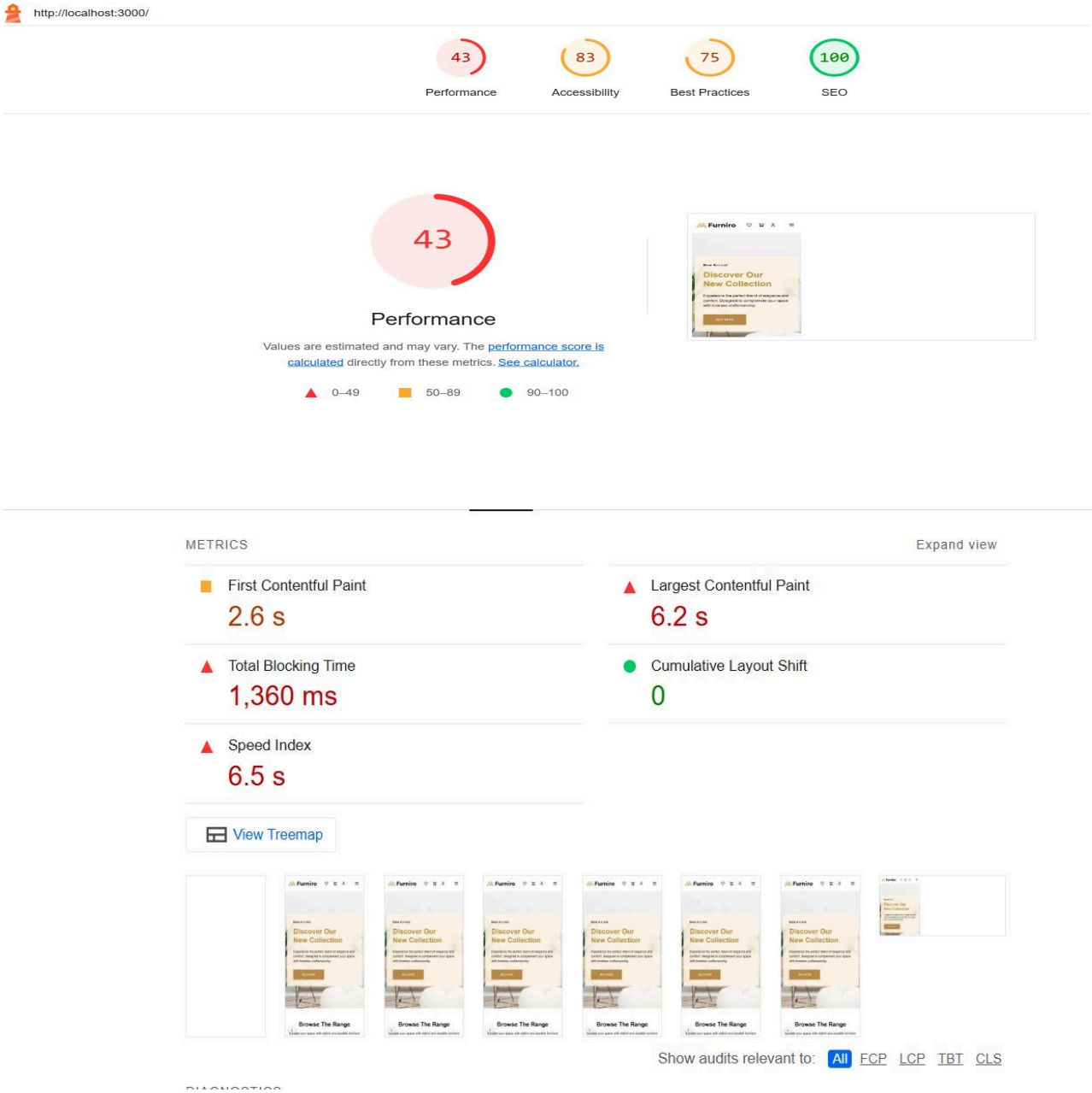
CSV Table

A2

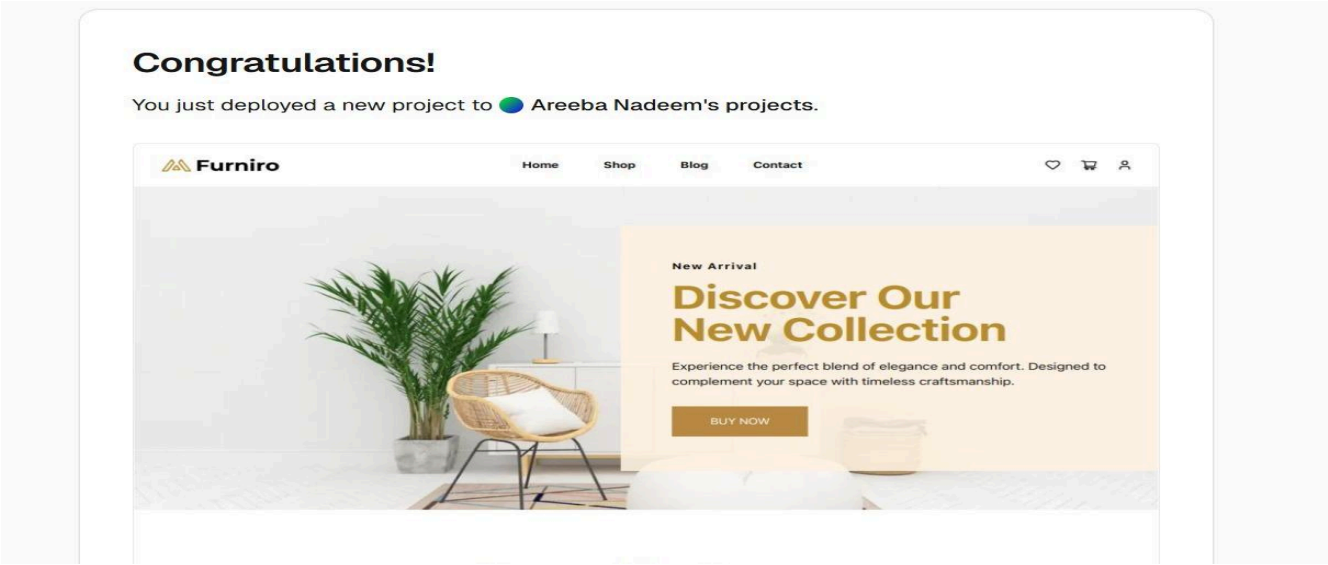
Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks
TC001	Validate product listing	Open shop page > Verify products	Products displayed	Products displayed	Passed	No issues found
TC002	Test API error handling	Test API error handling	Disconnect API > Refresh page	Show fall back message	Passed	Handled gracefully
TC003	Check cart functionality	Check cart functionality	Add item to cart > Verify cart	Cart Update correctly	Passed	Works as expected
TC004	Test form validation	Test form validation	Submit form with empty fields	Display error message	Failed	Missing validation check
TC005	Verify HTTPS connection	Verify HTTPS connection	Open site > Check HTTPS status	HTTPS enabled	Passed	Secure connection

3. Performance Testing

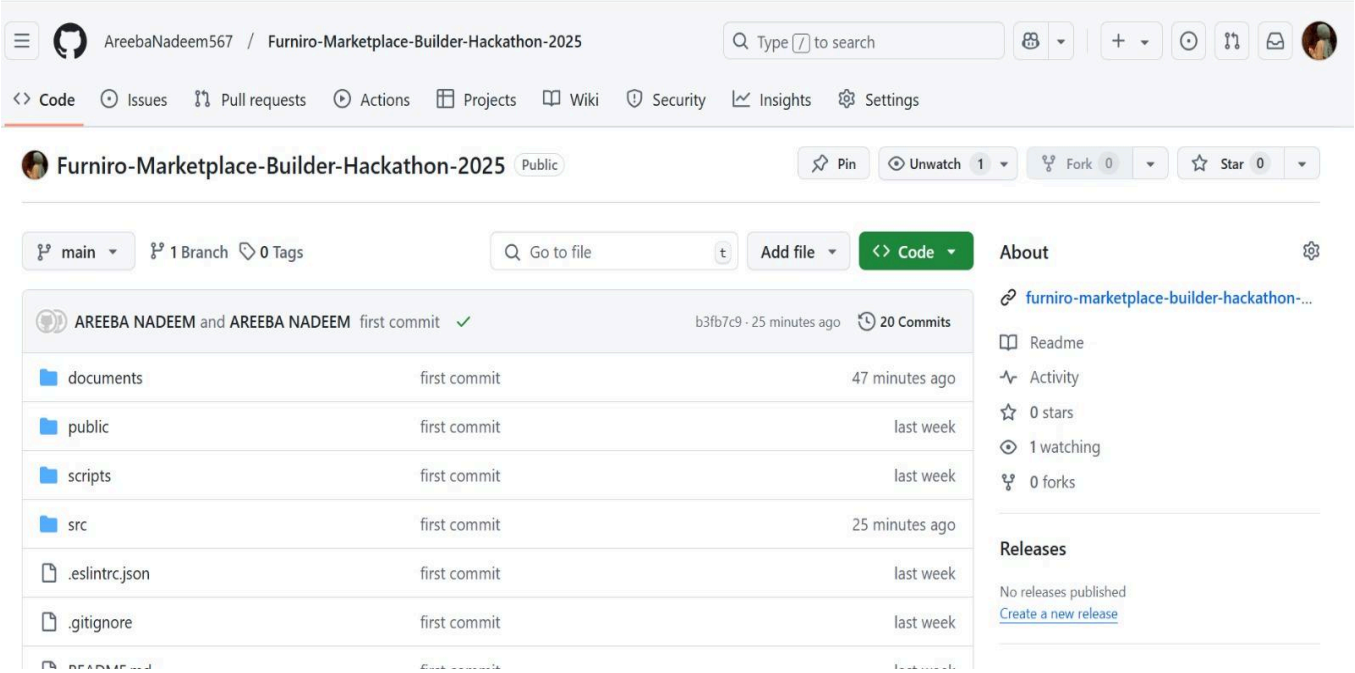
Here is performance report generate by lighthouse tools;



Vercel



Github



Conclusion for Deployment Preparatio and Staging Step:

Day 6 focused on setting up a staging environment for deployment, including configuring environment variables, testing functionality, and updating documentation. This ensures a smooth and secure transition to the live platform, minimizing risks and enhancing readiness for production.