Day 6 - Deployment Preparation and Staging Environment Setup

Prepared By Aamna Ansari

Hackathon Day 6

Deployment Preparation and Staging Environment Setup

Objective:

Day 6 is dedicated to preparing your marketplace for deployment by establishing a staging environment, configuring hosting platforms, and ensuring the application is ready for customer-facing use. Building on the testing and optimization efforts from Day 5, this phase focuses on ensuring the marketplace functions smoothly in a production-like setting. Students will also gain insights into industry-standard practices for managing various environments, including non-production environments (such as TRN, DEV, and SIT) and production environments (such as UAT, PROD, and DR).

Key Learning Outcomes:

1. Build dynamic frontend components that fetch and display data from Sanity CMS or APIs.

2. Implement reusable and modular components for easier maintenance and scalability.

3. Focus on responsive design and implement UX/UI best practices.

Prepare for real-world client projects by replicating professional workflows.

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 IDs

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.

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

I choose Vercel for quick and easy deployment.

https://furnir02byamna.vercel.app/

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.

https://github.com/AamnaAnsari/furnir02byamna

Step 2: Configure Environment Variables

Create. env.local File:

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

Upload Variables to Vercel:

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

Step 3: Deploy to Staging

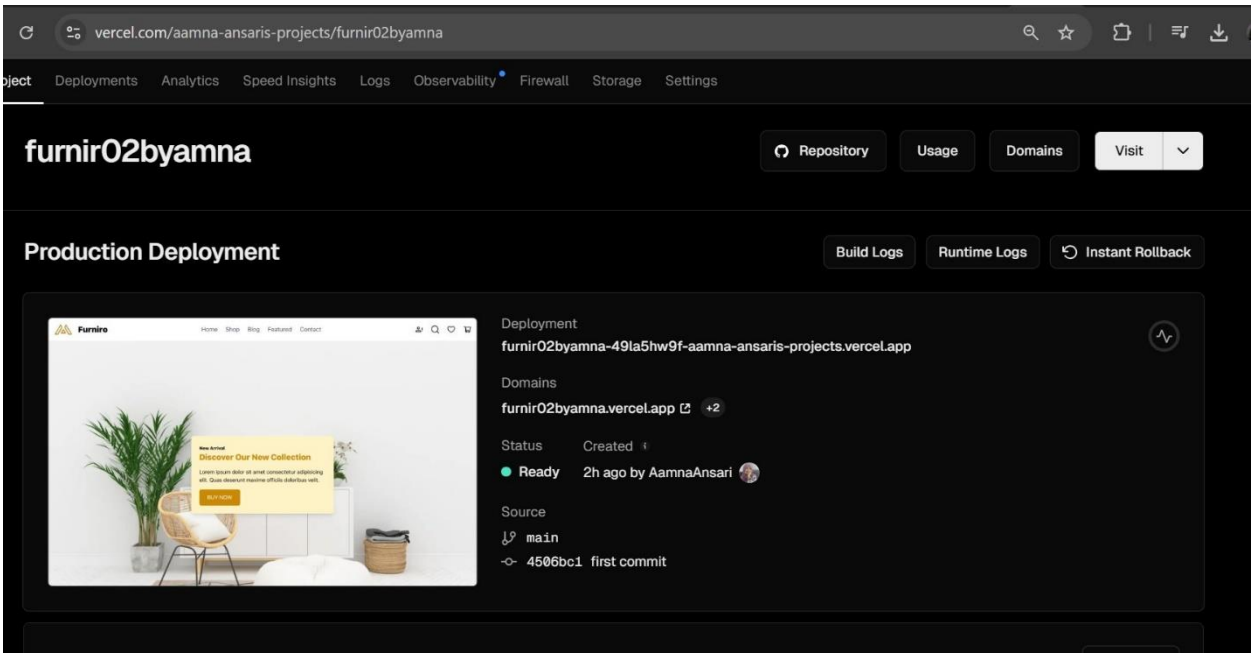
Deploy Application:

 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.

 Product Details: Verified product details page displayed the correct information.

 User Profile: Checked user login, profile update, and profile display.

 Cart Operations: Ensured products could be added, removed, and quantities updated in the cart.

 Wishlist: Validated the ability to add and remove products from the wishlist.

 Category: Ensured categories displayed correct product listings and filtered accordingly.

 Dynamic Routing: Verified that dynamic routing worked properly for product and category pages.

 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.

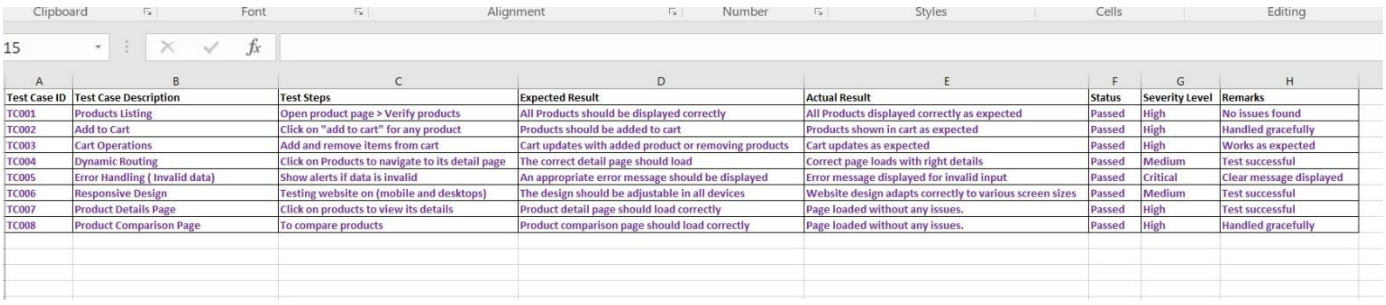
 Security Testing:

 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.

CSV Table:

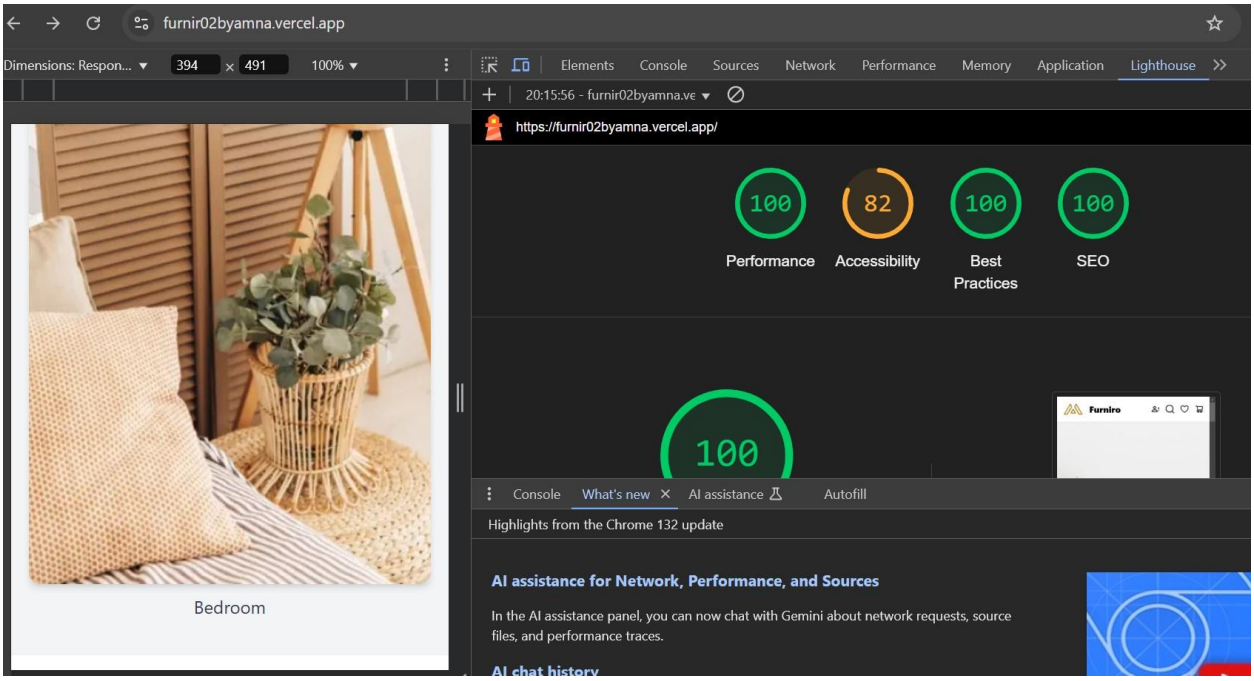


Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Products Listing	Open product page > Verify products	All Products should be displayed correctly	All Products displayed correctly as expected	Passed	High	No issues found
TC002	Add to Cart	Click on "add to cart" for any product	Products should be added to cart	Products shown in cart as expected	Passed	High	Handled gracefully
TC003	Cart Operations	Add and remove items from cart	Cart updates with added product or removing products	Cart updates as expected	Passed	High	Works as expected
TC004	Dynamic Routing	Click on Products to navigate to its detail page	The correct detail page should load	Correct page loads with right details	Passed	Medium	Test successful
TC005	Error Handling (Invalid data)	Show alerts if data is invalid	An appropriate error message should be displayed	Error message displayed for invalid input	Passed	Critical	Clear message displayed
TC006	Responsive Design	Testing website on (mobile and desktops)	The design should be adjustable in all devices	Website design adapts correctly to various screen sizes	Passed	Medium	Test successful
TC007	Product Details Page	Click on products to view its details	Product detail page should load correctly	Page loaded without any issues.	Passed	High	Test successful
TC008	Product Comparison Page	To compare products	Product comparison page should load correctly	Page loaded without any issues.	Passed	High	Handled gracefully

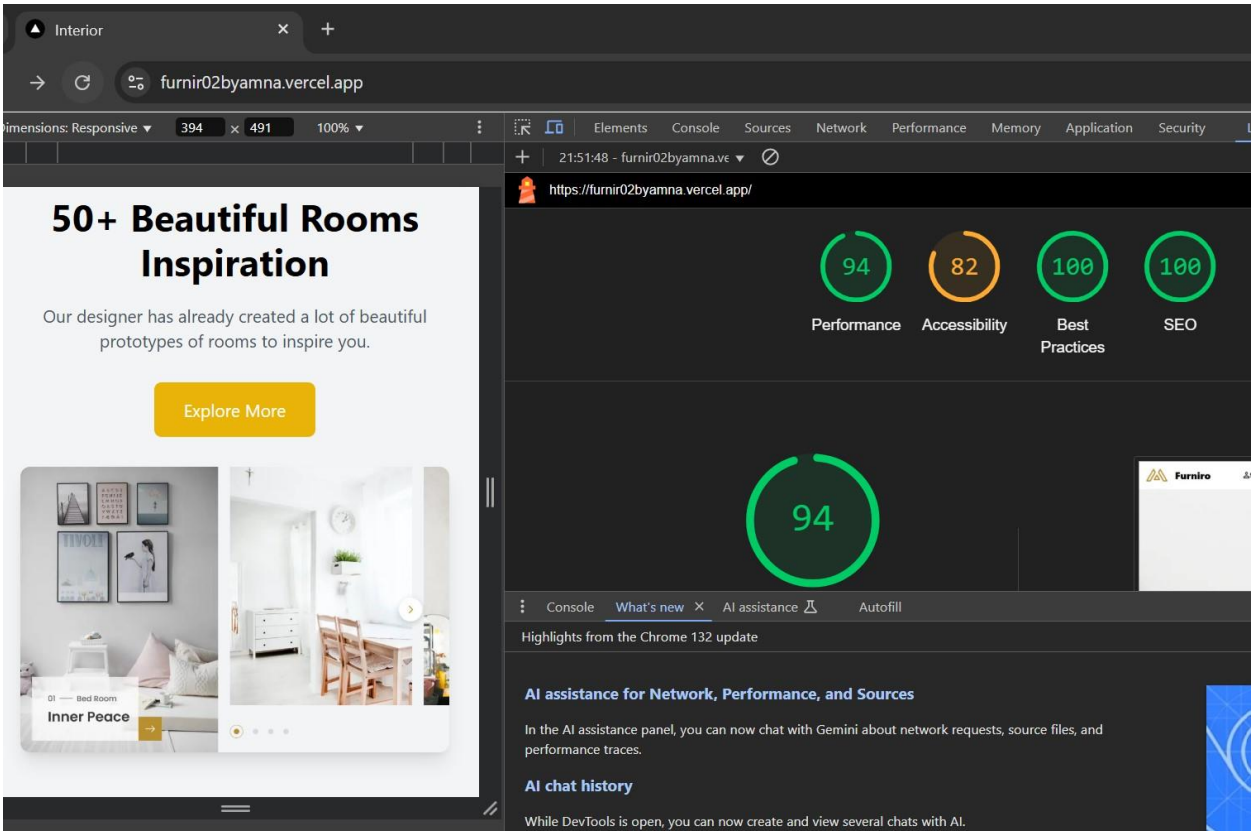
3. Performance Testing

Here is performance report generate by lighthouse tools.

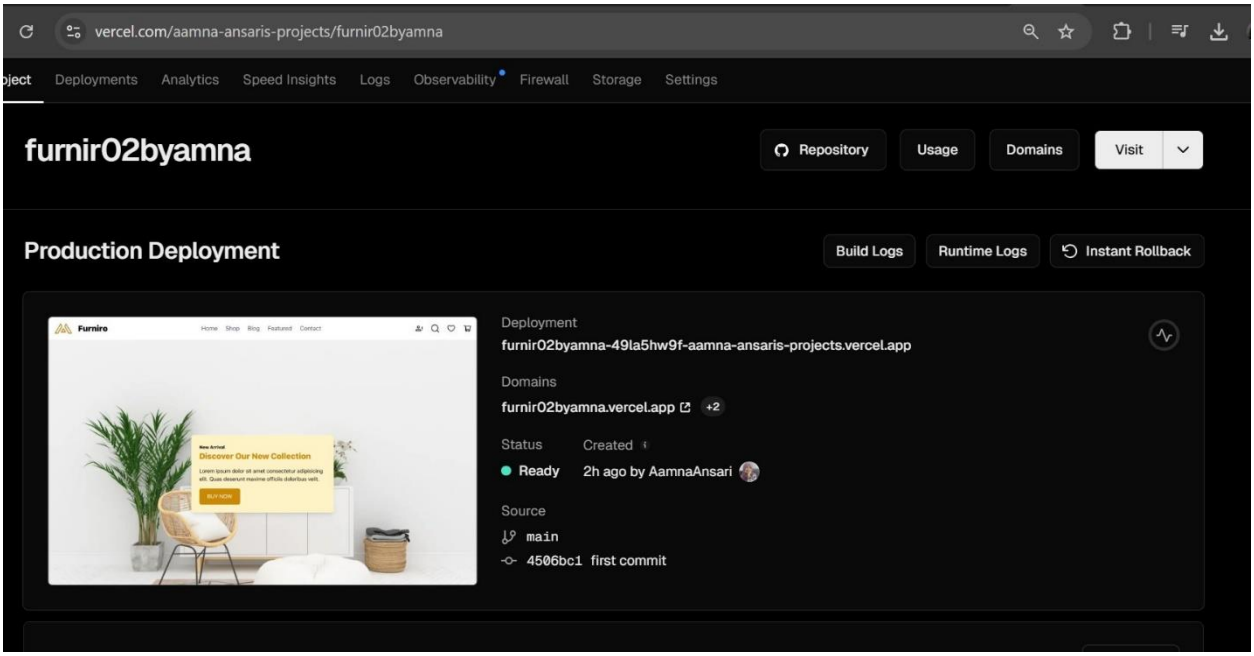
Performance Metrics for Desktop:



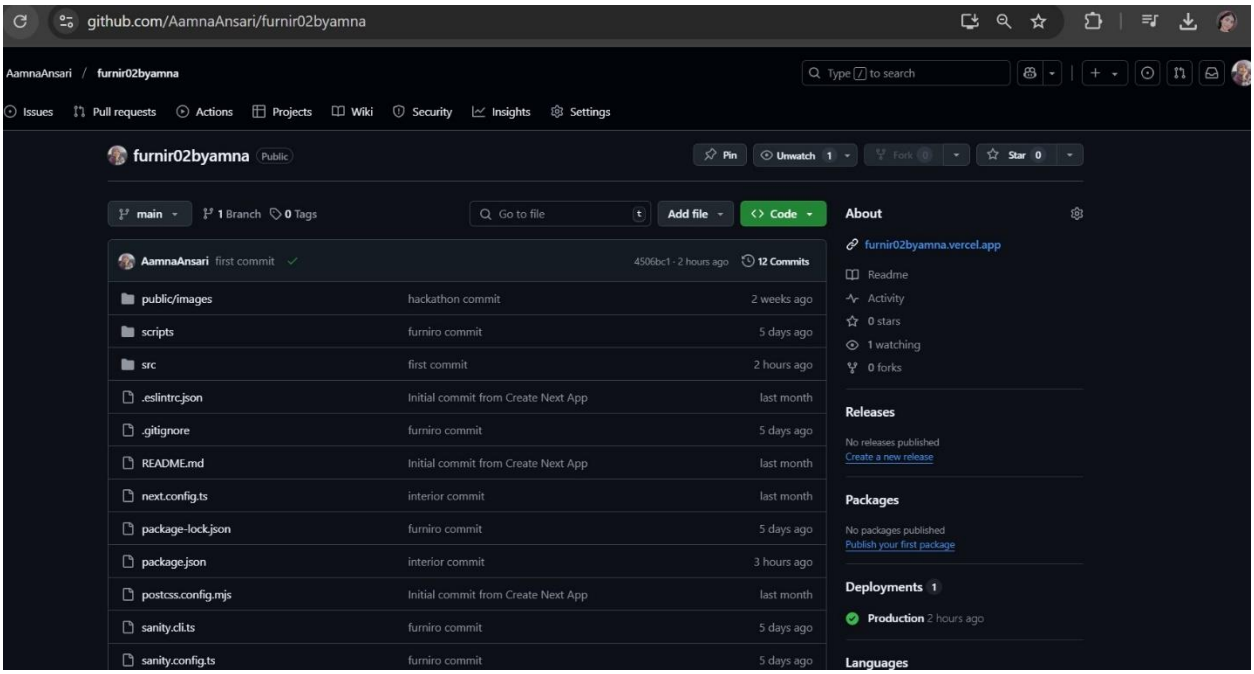
Performance Metrics for Mobile:



Vercel:



Github:



Conclusion:

Day 6 centered on preparing for deployment by setting up a staging environment, configuring essential environment variables, testing key functionalities, and updating documentation. This comprehensive approach ensures a seamless and secure transition to the live platform, mitigating potential risks and optimizing the marketplace's readiness for production.

SELF VALIDATION CHECKLIST

1. Deployment Preparation ☐
2. Staging Environment Testing ☐
3. Documentation ☐
4. Form Submisision ☐
5. Final Review ☐

💼 **Prepared By: Aamna Ansari**

💼 **Prepared To: Ameen Alam**

💼 **Class Teacher: Sir Ali Aftab Sheikh**

💼 **Batch : Tuesday (2 to 5)**

💼 **Date: 26 Jan, 2025**