

Day 6 Report:

Deployment Preparation and Staging Environment Setup:

Objective:

Day 6 focuses on preparing the marketplace application for deployment by setting up a staging environment, configuring hosting platforms, and ensuring the application is ready for a customer-facing release.

Step 1: Hosting Platform Setup:

Vercel (Recommended)

Vercel was selected as the hosting platform due to its seamless integration with Next.js and its ability to automatically deploy projects from GitHub. ***Key benefits include:***

1. Easy Integration: Native support for Next.js.
2. Automatic Deployment: Deploys automatically from GitHub.
3. Scalability: Handles varying traffic loads efficiently.
4. Serverless Functions: Enables backend logic without managing servers.

Step 2: Configure Environment Variables:

Secure API Keys, Database Credentials, and Sensitive Data Using .env Files Sensitive data like API keys and credentials should be stored securely in .env files. This prevents exposing secrets in the codebase.

1. NEXT_PUBLIC_SANITY_PROJECT_ID
2. NEXT_PUBLIC_SANITY_DATASET
3. SANITY_API_TOKEN
4. IMPORT_DATA_ON_SANITY_CHEFS
5. IMPORT_DATA_ON_SANITY_FOODS

Step 3: Deploy To Staging:

1. Deploy the Application to Staging:

- * Push the latest code to GitHub.

- *Set up a Preview Deployment in Vercel (either by linking a staging branch or using the default production branch).

- *Vercel will trigger the build and deployment process automatically.

2. Validate Deployment:

- * Check Vercel's build logs for a successful deployment.

- *Access the staging URL and test the site to ensure all features work as expected.
- o Verify the functionality of backend integrations (e.g., Sanity CMS and APIs).

3. Troubleshoot:

- *If issues arise, review build logs and verify the correct configuration of environment variables.

Step 4: Staging Environment Testing:

1. Conduct Functional Testing
2. Perform Performance Testing Lighthouse.
3. Verify Responsiveness & Error Handling.

Cart is empty.

Your Cart is empty

Coupon Code

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque diam pellentesque bibendum non.

Apply

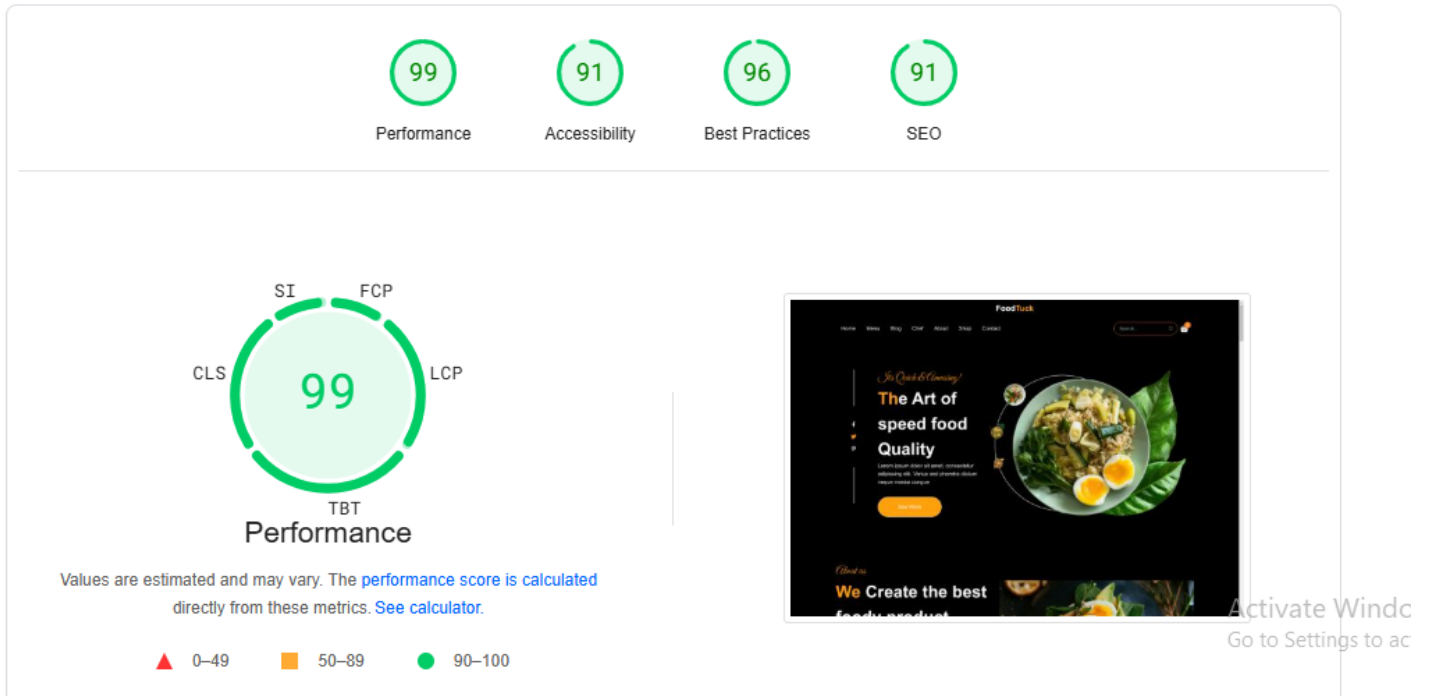
Total Bill

Cart Subtotal	\$0.00
Shipping Charge	\$0.00
Total Amount	\$0.00

Proceed to Checkout

6

4. Document Test Result.



Step 5: Documentation Updates:

1. Record all test outcomes, including performance benchmarks and functional tests.
2. List unresolved issues and prioritize fixes before production deployment.

Test Case Reporting

Documenting test cases in a structured format ensures clear tracking of testing progress and issues. Here's how you can document test cases in a CSV file:

<i>Test Case Id</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>	<i>Remarks</i>
TC001	Validate product listing	Open product page>Verify products	Products displayed	Product displayed	Passed	No issue found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback message	Fallback message shown	Passed	Handled gracefully
TC003	Check cart functionality	Add item to cart > Verify cart	Cart updates correctly	Cart updates correctly	Passed	Works as expected
TC004	Test form validation	Submit form with empty fields	Display error message	Error message displayed	Failed	Missing validation check
TC005	Verify HTTPS connection	Open site > Check HTTPS status	HTTPS enabled	HTTPS enabled	Passed	Secure connection