HACKATHON
DAY 6

DEPLOYMENT
PREPARATION
STAGING
ENVIRONMENT SETUP

PEPARED BY: Muhammad Mubashir Saeedi
PEPARED TO: Ameen Alam
TEACHER: Sir Bilal Khan, Sir Aneeq Khatri

Hackathon Day 6
Bandage Marketplace Template

Deployment Preparation and Staging Environment Setup

Objective

The objective of the Bandage project on Day 6 is to ensure the system is ready for deployment by setting up a staging environment, configuring hosting platforms, and preparing it for customer-facing use. The focus is on creating a production-like environment for testing and ensuring the application operates seamlessly. Additionally, the project emphasizes understanding and managing environments such as non-production (TRN, DEV, SIT) and production (UAT, PROD, DR), following industry-standard deployment practices.

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. **Apply state management techniques** to manage data flow across components.
4. **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.

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:

```
$ .env.local
1   NEXT_PUBLIC_SANITY_PROJECT_ID="q2f1sk4w"
2   NEXT_PUBLIC_SANITY_DATASET="production"
3   NEXT_PUBLIC_SANITY_TOKEN="skhTRYMbcHBCj0l2l7O73Mx3oL1lAEWqR2CiulHU31uhwziudhIbOFCrnfGaaKPKWq39ilazSXuTRla\
4
```

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.

2. Test Case Reporting

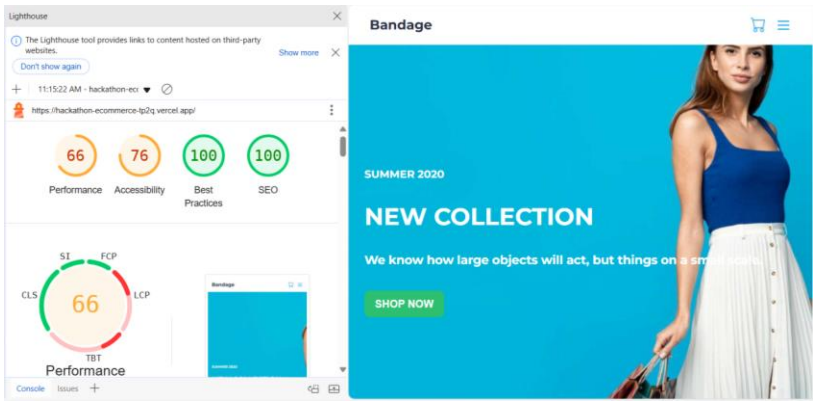
CSV Table

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Product Listing	Verify that all products are displayed on the homepage.	All products should be displayed correctly.	All products displayed as expected.	Pass	High	No issues found
TC002	Product Details	Click on any product to view its details.	The product detail page should load correctly.	Product detail page loaded without issues.	Pass	High	No issues found
TC003	Add to Cart	Click on 'Add to Cart' for a product.	Product should be added to the cart.	Product added successfully to the cart.	Pass	High	No issues found
TC004	Cart Operations	Add and remove items from the cart.	Cart should update correctly with the added/removed items.	Cart updates correctly when items are added or removed.	Pass	High	No issues found
TC005	Dynamic Routing	Click on a product to navigate to its detail page.	The correct product detail page should load.	Correct page loaded with the right details.	Pass	Medium	No issues found
TC006	Category Filter	Apply different category filters.	Products should be filtered based on selected category.	Products filtered correctly based on the selected category.	Pass	Medium	No issues found
TC007	Error Handling (Network)	Simulate a network failure and attempt to load a product.	An error message should be displayed, indicating a failure.	Error message displayed as expected.	Pass	Critical	Error handling works as expected.
TC008	Error Handling (Invalid Data)	Enter invalid data in a product search or form.	An appropriate error message should be displayed.	Error message displayed for invalid input.	Pass	High	Handled correctly with clear message.
TC009	Responsive Design	Test the website on multiple devices (desktop, tablet, mobile).	The design should adjust and be responsive on all devices.	Website design adapts correctly to various screen sizes.	Pass	Medium	No issues found on mobile/tablet.

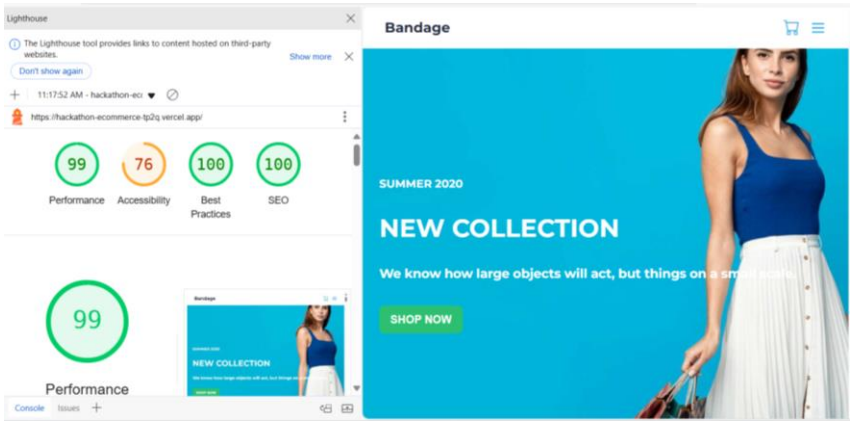
3. Performance Testing

Here is performance report generate by lighthouse tools;

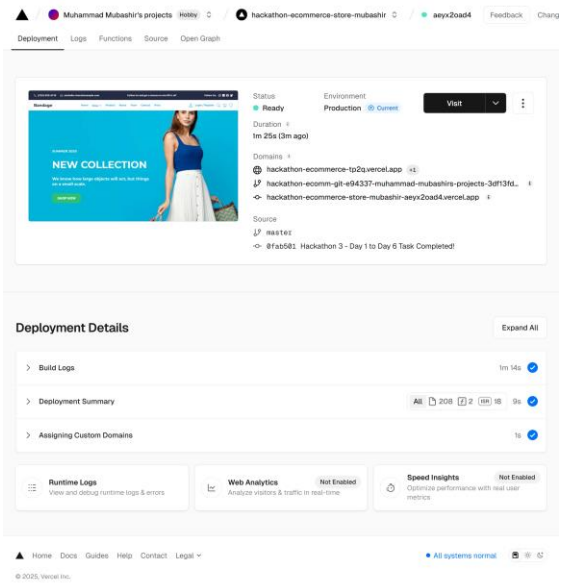
Mobile Performance



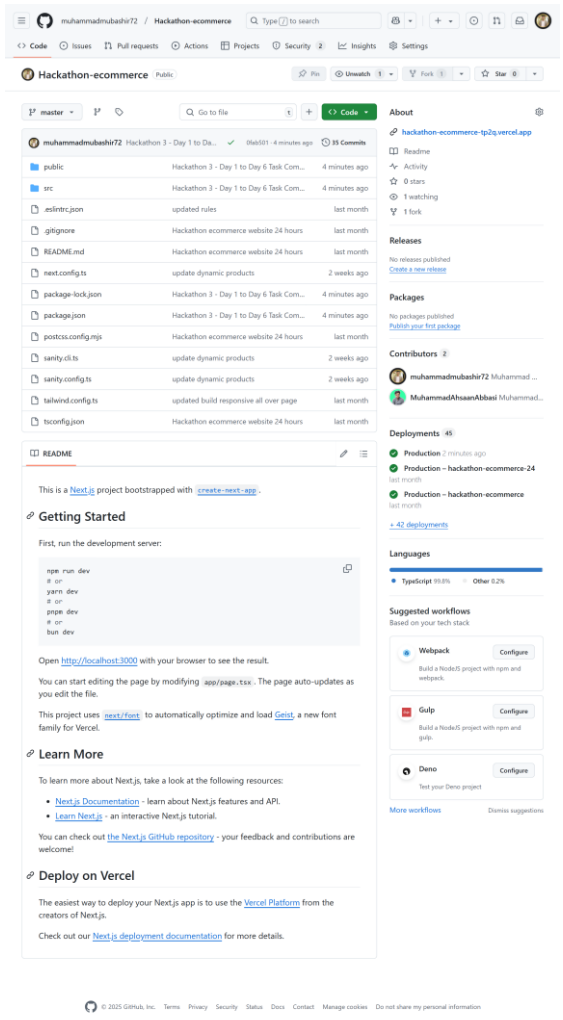
Desktop Performance



Vercel

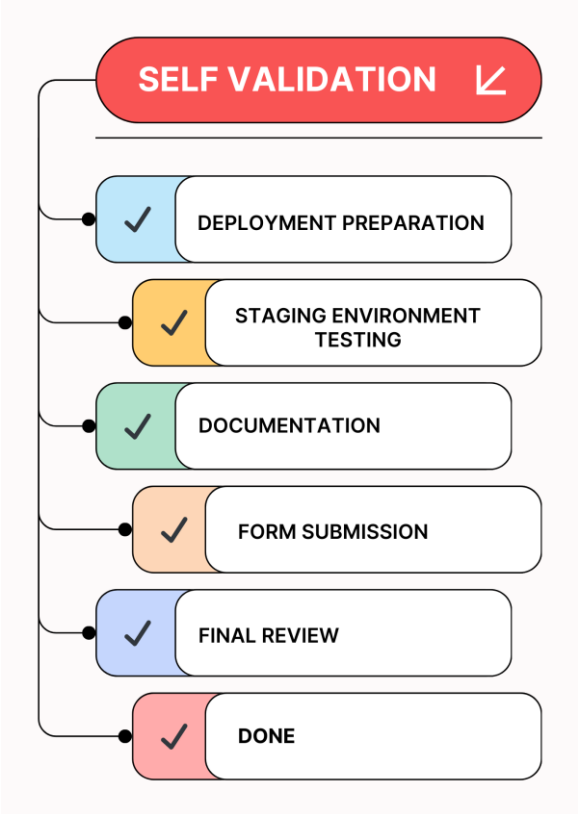


Github



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. ✓



Prepared By: Muhammad Mubashir Saeedi

Prepared To: Ameen Alam

Class Teacher: Sir Bilal Khan And Sir Aneeq Khatri

Date: 22 Jan, 2025