



# Day 6 - Deployment Preparation and Staging Environment Setup

Day 6 focuses on preparing my marketplace for deployment by setting up a staging environment, and ensuring readiness for a customer facing application. Building on the testing and optimization work from Day 5, this stage emphasizes ensuring the marketplace operates seamlessly in a production-like environment.

## 1. How did i set up my hosting platform?

- **Hosting Platform Chosen:** I chose **Vercel** for staging the application.
- **Steps Taken:**
  1. I created an account and logged into **Vercel**.
  2. I connected my **GitHub repository** to **Vercel** using the platform's integration feature.
  3. After connecting the repository, I configured the build and deployment settings to match the project's requirements.

4. I confirmed that the staging environment was successfully created and deployed.

## 2. How did i configure environment variables for deployment?

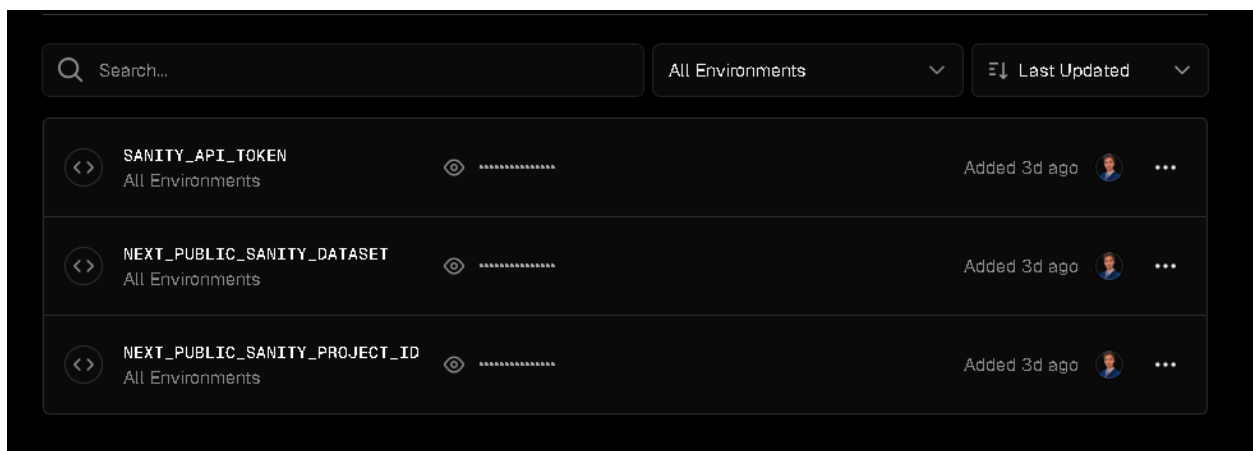
- **Environment Variables Used:** I created a `.env` file to securely store the API keys, database credentials, and other sensitive information.

- **Steps Taken:**

1. I created the `.env` file in the root of the project folder.
2. I added variables such as:

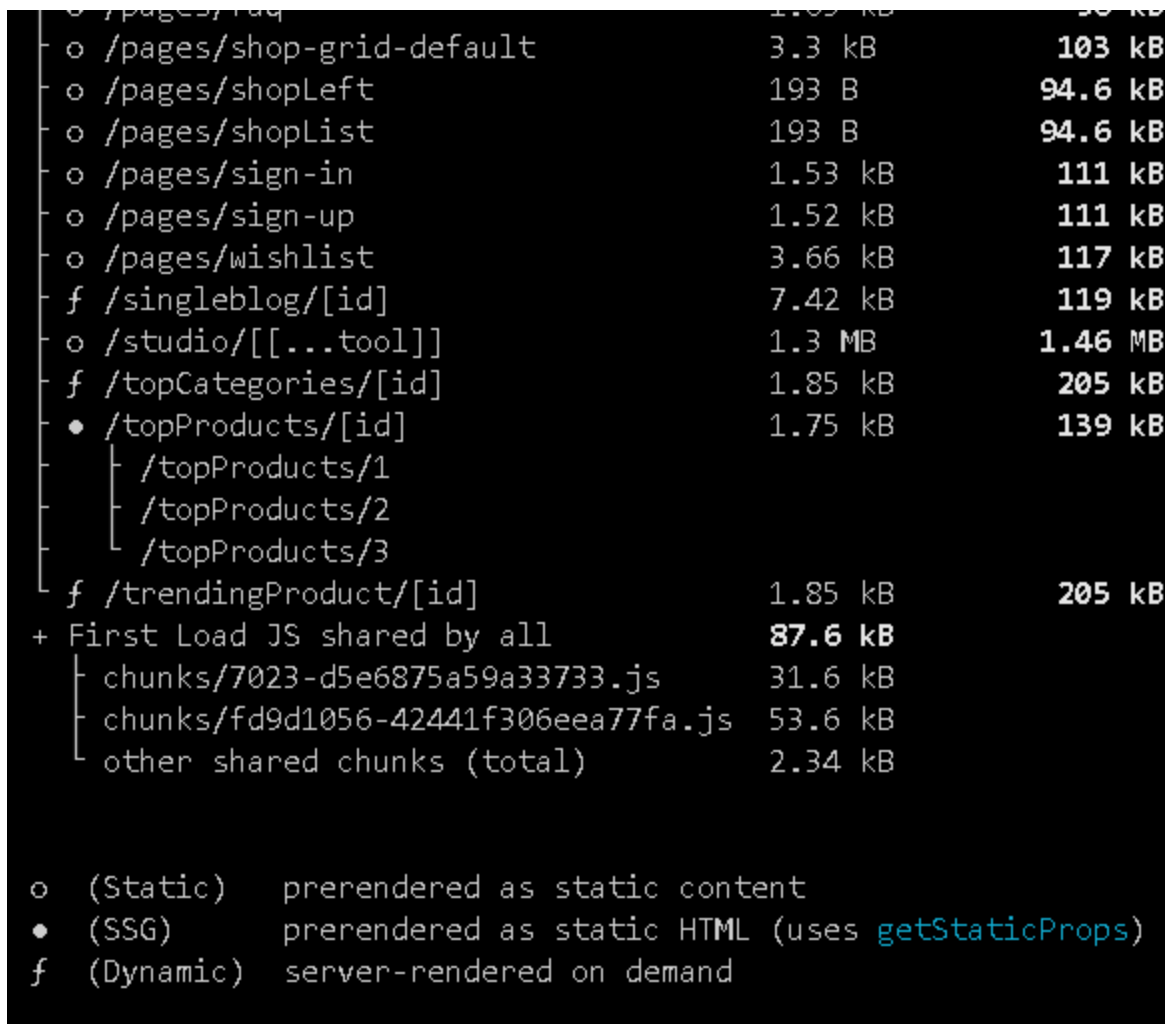
```
NEXT_PUBLIC_SANITY_PROJECT_ID=my_project_id
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN=my_api_key
```

3. I uploaded the environment variables through **Vercel's dashboard** to ensure secure deployment.



## 3. How did I deploy the application to the staging environment ?

first i run the command in terminal `npm run build` to see if there is any error or not , once i satisfied , I triggered the deployment by clicking the **Deploy** button on **Vercel**.



o /pages/faq	1.05 kB	105 kB
o /pages/shop-grid-default	3.3 kB	103 kB
o /pages/shopLeft	193 B	94.6 kB
o /pages/shopList	193 B	94.6 kB
o /pages/sign-in	1.53 kB	111 kB
o /pages/sign-up	1.52 kB	111 kB
o /pages/wishlist	3.66 kB	117 kB
f /singleblog/[id]	7.42 kB	119 kB
o /studio/[[...tool]]	1.3 MB	1.46 MB
f /topCategories/[id]	1.85 kB	205 kB
• /topProducts/[id]	1.75 kB	139 kB
/topProducts/1		
/topProducts/2		
/topProducts/3		
f /trendingProduct/[id]	1.85 kB	205 kB
+ First Load JS shared by all	87.6 kB	
chunks/7023-d5e6875a59a33733.js	31.6 kB	
chunks/fd9d1056-42441f306eea77fa.js	53.6 kB	
other shared chunks (total)	2.34 kB	
o (Static)	prerendered as static content	
• (SSG)	prerendered as static HTML (uses <code>getStaticProps</code> )	
f (Dynamic)	server-rendered on demand	

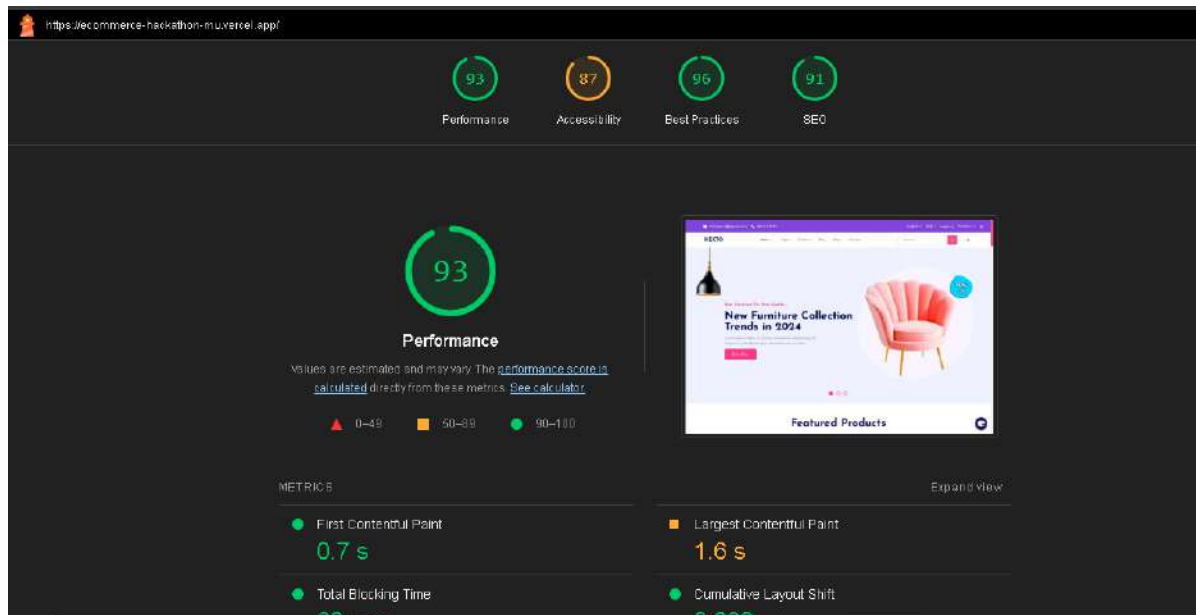
## 4. How did you test the staging environment?

- **Testing Methods:**

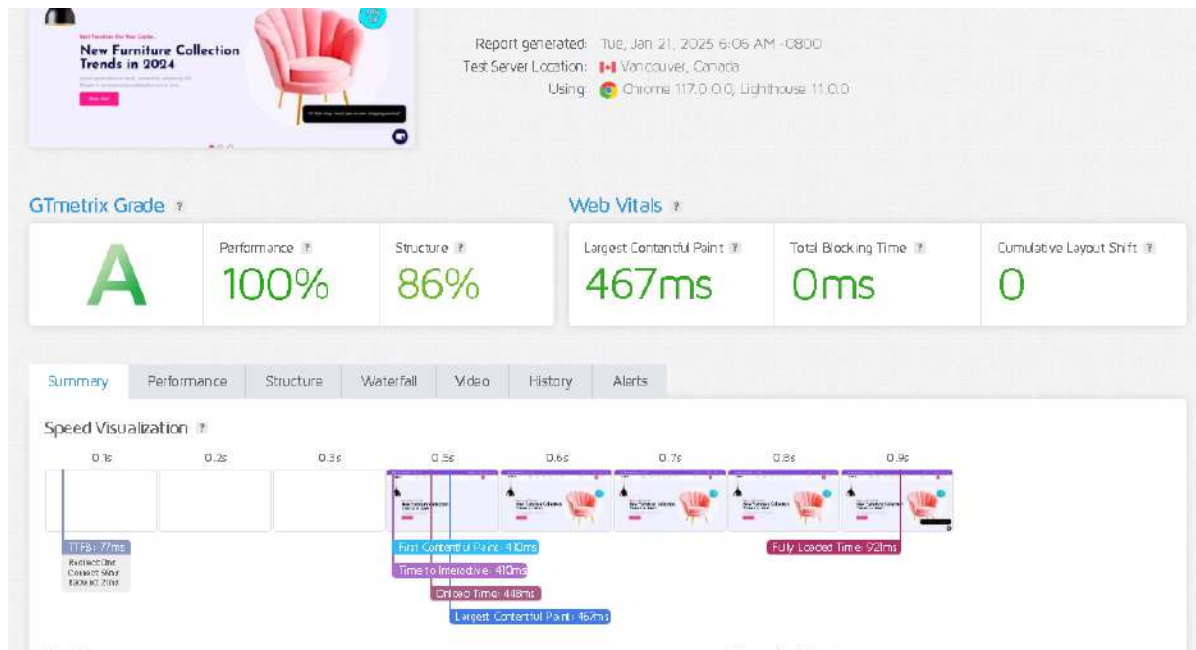
1. **Functional Testing:** I used **manual testing** to verify features like the product listing page, cart functionality etc .

2. **Performance Testing:** I used **Lighthouse** to analyze the app's speed and responsiveness and **GTmetrix** for additional performance insights.

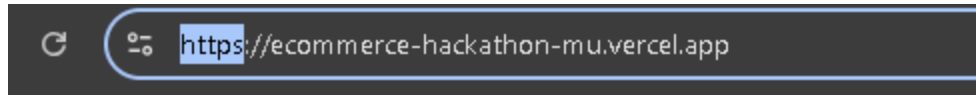
## Lighthouse



## GTMETRIX



1. **Security Testing:** I validated input fields to ensure there were no vulnerabilities like, confirmed **HTTPS** was enabled, and checked if sensitive data was properly encrypted.



## 5. What results did you find during staging testing?

**Answer:**

- **Test Results:**
  - **Functional Testing:** All major features, including the product page and cart functionality, worked as expected.
  - **Performance Testing:** The website passed performance tests with no significant issues. It loaded within an acceptable time frame, as confirmed by **Lighthouse & GTMetrix**
  - **Security Testing:** All input fields were secure, **HTTPS** was active, and no security vulnerabilities were found

## 6. What documentation did you create for this process?

- **Documentation Created:**
  1. **Test Case Report:** I documented all test cases, including descriptions, steps, expected results, and actual results.
  2. **Performance Report:** I added **Lighthouse** performance test reports.
  3. **Deployment Documentation:** I updated the **README.md** file with the deployment process, testing results, and steps for running the app in different environments.
  4. **File Organization:** All project files are organized in a structured folder hierarchy. The `documents/` folder contains all test cases and performance reports.

## 7. How did you structure your project and ensure a professional setup?

## Answer:

- **Project Structure:**

- The project is organized into the following folders:
  - `src/` for source code files.
  - `public/` for static files like images and assets.
  - `documents/` for all project documentation (including test reports, performance data, etc.).
- The **README.md** file contains a summary of the project and the steps for setting up, deploying, and testing the app

## Conclusion

By setting up a staging environment and ensuring all essential configurations were in place, I established a strong foundation for a customer-facing application. Through testing and optimization, I confirmed the application's functionality, performance, and security met the required standards for production.

The deployment process was streamlined with Vercel, and environment variables were securely managed for seamless integration. Testing through Lighthouse and GTmetrix provided valuable insights into performance, while manual and security testing ensured the application's reliability. Comprehensive documentation, including test case reports, performance results, and deployment guidelines, was created to support ongoing maintenance and collaboration.

With a well-structured project setup, this stage reinforces the readiness of the marketplace to transition into a live production environment, ensuring a smooth experience for users and stakeholders alike.