
Day # 6 Deployment Preparation and Staging Environment Setup [E-Commerce(HELLO NIKE)]

Name: Alishba Shabbir

Introduction:

This guide outlines the key steps required to set up a staging environment, configure secure deployment settings, and conduct thorough testing before the final deployment. It ensures your marketplace project runs smoothly and securely in a production-like setting.

1. Hosting Platform Setup

Choosing the Right Platform:

For fast deployment and seamless GitHub integration, Vercel is the best choice. It ensures smooth workflows and efficient project deployment. Vercel offers a reliable and developer-friendly experience.

2. Configuring Environment Variables

Managing API Keys, Database Credentials, and Sensitive Data:

Securing sensitive information (such as API keys and database credentials) is crucial in web development. Instead of embedding them directly into the source code, use `.env` files for enhanced security.

Why Use `.env` Files?

- **Security** – Prevents unauthorized access to sensitive data.
- **Flexibility** – Supports different configurations for development, staging, and production environments.
- **Portability** – Allows easy collaboration without exposing secrets.

Setting Up Environment Variables in Vercel:

1. Navigate to **Settings > Environment Variables** in your Vercel dashboard.
2. Add the necessary keys, such as:
 - NEXT_PUBLIC_SANITY_PROJECT_ID
 - API_KEY
3. Redeploy the application to apply changes.

Best Practices for Secure Deployment:

- Never hardcode sensitive data in your codebase.
- Regularly update and rotate API keys.
- Utilize secret management tools (e.g., AWS Secrets Manager, HashiCorp Vault).
- Restrict access to environment variables based on user roles.
- Monitor logs to detect potential security threats.

3. Deploying the Staging Environment

Setting up a staging environment enables comprehensive testing under conditions similar to production before the final release. This helps identify bugs, optimize performance, and ensure all functionalities work as expected.

4. Staging Environment Testing

Thorough testing is critical to ensure website stability, security, and efficiency. The following testing methods were applied:

1. Functional Testing:

Ensures that all features perform correctly, including navigation, forms, and interactive elements.

Key Areas:

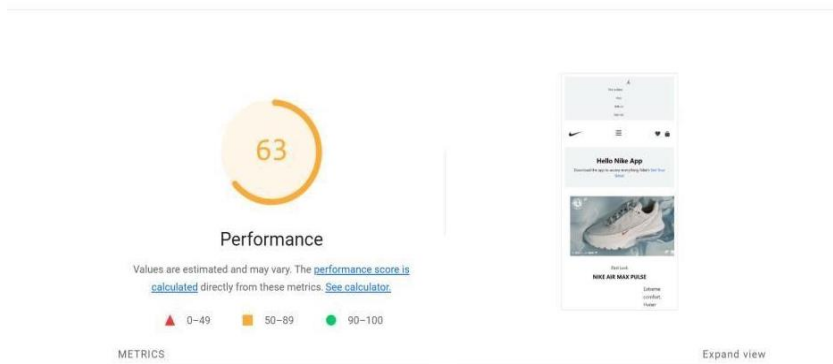
- Navigation links and button responsiveness
- Product listings and search functionality
- Shopping cart operations (adding/removing items)
- Form validation and error handling
- Responsive design across various devices

2. Performance Testing:

Evaluates website speed and responsiveness under different conditions to ensure a seamless user experience.

Key Areas:

- Page load speed analysis using **Lighthouse**
- Handling multiple simultaneous user requests
- Optimizing images and static assets



3. Security Testing:

Identifies vulnerabilities and ensures data protection against threats.

Key Areas:

- Encryption and secure data storage
- Protection against common cyber threats (e.g., SQL Injection, XSS attacks)
- HTTPS implementation and secure API communication

