

DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

- On Day 6, the focus is on getting your marketplace ready for deployment by setting up a staging environment, configuring a hosting platform, and making sure the application is fully prepared for users.
- Building on the testing and optimization work from Day 5, this step ensures that your marketplace operates smoothly in a production-like setting.

Key tasks include:

- Creating a staging environment that replicates the production setup.
- Selecting a hosting platform (e.g., Vercel, Netlify, AWS, or a custom server).
- Setting up environment variables for secure API integration.
- Verifying deployment readiness through performance testing and bug fixes.
- Ensuring a smooth user experience before the official launch.

Step 1: Set Up Hosting Platform

- **Select a Platform:** Opt for platforms like Vercel or Netlify for fast and efficient deployment.

Step 2: Configure Environment Variables

- **Create a .env File:** Add sensitive information such as API keys and tokens.
 - NEXT_PUBLIC_SANITY_PROJECT_ID = "your_project_id"
 - NEXT_PUBLIC_SANITY_DATASET = production
 - API_KEY = "your_api_key"

Step 3: Deploy to Staging

- **Deploy the Application:** Push the application to a staging environment using the chosen hosting platform.

Step 4: Test the Staging Environment

1. **Types of Testing:**
 - **Functional Testing:** Ensure all features, such as product listings, search functionality, and cart operations, work as expected.
 - **Performance Testing:** Use tools like Lighthouse or GTmetrix to evaluate speed and responsiveness.
 - **Security Testing:** Check input fields, confirm HTTPS usage, and ensure secure API communication.

Step 5: Update Documentation

- **Create a README.md:** Document all project activities, including deployment steps and test case outcomes.

TEST CASE REPORT

<u>Test Case ID</u>	<u>Test Case Description</u>	<u>Test Steps</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Status</u>	<u>Severity Level</u>	<u>Seacurity level</u>	<u>Remarks</u>
2	TC001	<u>Validate product listing page</u>	<u>Open product page > Verify products</u>	<u>Products displayed correctly</u>	<u>Products displayed correctly</u>	<u>Passed</u>	<u>Low</u>	<u>Displayed Successfully.</u>
3	TC002	<u>Test API error handling</u>	<u>Disconnect API > Refresh page</u>	<u>Show fallback UI with error message</u>	<u>Error message shown</u>	<u>Passed</u>	<u>Medium</u>	<u>Works as expected.</u>
4	TC003	<u>Check cart functionality</u>	<u>Add product to cart > Verify cart contents</u>	<u>Cart updates with added product</u>	<u>Cart updates as expected</u>	<u>Passed</u>	<u>Low</u>	<u>Display Successfully</u>
5	TC004	<u>Ensure responsiveness on mobile</u>	<u>Resize browser window > Check layout</u>	<u>Layout adjusts properly to screen size</u>	<u>Responsive layout working as intended</u>	<u>Passed</u>	<u>Medium</u>	<u>Test Successfully</u>