

# Day 6 - Deployment Preparation and Staging Environment Setup Documentation

## Introduction

Day 6 focused on preparing the marketplace for deployment by setting up a staging environment, configuring hosting platforms, and ensuring readiness for a customer-facing application. This phase builds on testing and optimization from Day 5, with emphasis on professional environment management and seamless production-like functionality.

---

## Objectives

1. Set up and configure a staging environment for the marketplace.
  2. Understand professional environment management stages (TRN, DEV, SIT, UAT, PROD).
  3. Conduct comprehensive staging environment testing.
  4. Organize all project files in a structured GitHub repository.
  5. Document deployment steps and testing results professionally.
- 

## Key Areas of Focus

### 1. Hosting Platform Setup

- **Platform Selected:** Vercel for its ease of use and seamless integration with GitHub.
- **Repository Connected:** GitHub repository linked to Vercel.
- **Environment Variables Configured:** Sensitive data (API keys, tokens) securely added to the platform.

### 2. Staging Environment Deployment

- **Deployment Process:**
  - Application deployed successfully to a staging environment.
  - Build and deployment settings validated for error-free staging.
- **Validation:** Ensured all core features (product listing, cart operations, etc.) work seamlessly in the staging environment.

### 3. Staging Environment Testing

#### Functional Testing

- Tested workflows including product listing, search functionality, and checkout.
- Verified dynamic routing for product detail pages.

Performance Testing

- **Tools Used:** Lighthouse and GTmetrix.
- **Results:**
  - Page load time: Under 2 seconds.
  - Lighthouse performance score: 90+.

Security Testing

- Validated input fields to prevent SQL injection and XSS attacks.
- Ensured HTTPS for secure communication.
- Verified handling of sensitive data securely.

Responsiveness Testing

- Tested responsiveness across devices (desktop, tablet, mobile) using BrowserStack.
- Ensured consistent performance on Chrome, Firefox, Safari, and Edge.

---

Documentation Updates

Test Case Reporting

Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks
TC001	Validate product listing	Open homepage	Products displayed	Products displayed	Passed	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Fallback message shown	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 responsiveness layout	Resize browser window > Check layout	Layout adjusts properly	Layout adjusts properly	Passed	Responsive verified

Performance Report

- Lighthouse and GTmetrix reports uploaded to GitHub repository under `documents/performance/`.

## Project File Organization

- **Folder Structure:**
    - `src/` - Application source code.
    - `public/` - Public assets.
    - `documents/` - Reports, deployment instructions, and testing results.
    - `README.md` - Summary of project structure and deployment steps.
- 

## Final Deliverables

- **Deployed Staging Environment:** Fully functional staging link.
  - **GitHub Repository:**
    - All project files organized in a structured hierarchy.
    - `documents/` folder containing reports and deployment instructions.
    - A professional `README.md` summarizing project activities.
  - **Testing Reports:**
    - CSV-based test case report.
    - Performance analysis results.
- 

## Conclusion

This stage ensured that the marketplace operates seamlessly in a production-like environment. With robust testing, secure configurations, and professional documentation, the project is deployment-ready and meets industry standards.

---

## Submission Checklist

- ☒ Deployment Preparation Completed.
- ☒ Staging Environment Deployed.
- ☒ Testing Conducted and Documented.
- ☒ All Files Organized in GitHub Repository.
- ☒ Submission Form Completed.

