

# ***DEPLOYMENT DAY 6: PREPARATION AND STAGING ENVIRONMENT SETUP***

## ***Objective:***

The objective of Day 6 is to prepare the marketplace for deployment by setting up a staging environment, configuring a hosting platform, and ensuring the application operates seamlessly in a production-like environment. This involves setting up environment variables, testing, and documenting all steps to ensure readiness for deployment.

## ***Steps Taken for Deployment Preparation***

### **1. Hosting Platform Selection**

We chose **Netlify** as the hosting platform for its simplicity and efficient deployment process. The following configurations were made:

**Connected GitHub Repository:** The project repository was linked to Netlify for continuous integration and deployment.

**Build and Deployment Settings:** Configured to ensure successful staging builds.

### **2. Environment Variable Configuration**

Sensitive data such as API keys and project configurations were managed securely.

Such as :

NEXT\_PUBLIC\_SANITY\_PROJECT\_ID=your\_project\_id

NEXT\_PUBLIC\_SANITY\_DATASET=production

API\_KEY=your\_api\_key

Uploaded these variables securely to Netlify through the dashboard.

### 3. Staging Environment Setup

Deployed the application to a staging environment to simulate production. This involved:

- Validating successful builds without errors.
- Verifying basic functionality, such as the homepage, navigation, and interactions.

### 4. Staging Environment Testing

We conducted various tests to ensure the staging environment is reliable:

#### Functional Testing

Verified key workflows such as:

- Product listings and search functionality.
- Cart operations, including adding/removing items.
- API interactions for real-time data updates.

#### Performance Testing

Analyzed speed and responsiveness using **GTmetrix** and **Lighthouse**. Metrics such as load time, interactivity, and layout shifts were documented. Reports from GTmetrix provided detailed insights into performance bottlenecks and optimization opportunities.

#### Security Testing

Validated the following:

- Input fields to prevent SQL injection and XSS attacks.
- Secure handling of API keys.
- HTTPS implementation.

## ***Expected Output***

- A fully functional staging environment hosted on Netlify.
- Environment variables securely configured.
- Test case and performance reports uploaded to GitHub.
- A professional README.md summarizing all project activities.