

DAY 6

DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

Project Name: Marketplace Development

Prepared By: Asifa Muhammad Qasim

[illegible]

Overview

Day 6 of the project focused on setting up a staging environment to prepare the marketplace for deployment. The objective was to ensure that the application operates seamlessly in a production-like environment. This included configuring a hosting platform, securing environment variables, conducting thorough testing, and organizing project files in a structured GitHub repository.

Key Activities

1. Staging Environment Setup

- **Hosting Platform Selection:**
 - Chose **Vercel** as the hosting platform for its simplicity and support for Next.js projects.
 - Configured the platform to connect with the project's GitHub repository.
- **Configuration:**
 - Configured build settings for staging deployment.
 - Secured environment variables using Vercel's dashboard to handle sensitive information like API keys.
- **Validation:**
 - Deployed the application to a staging environment.
 - Verified successful builds and application functionality.

2. Environment Management

Learned about managing different environments and their purposes:

- **TRN (Training):** Used for onboarding and practice.
- **DEV (Development):** For local development and testing.
- **SIT (System Integration Testing):** Ensures systems interact as expected.
- **UAT (User Acceptance Testing):** Validates functionality with stakeholders.
- **PROD (Production):** The live customer-facing environment.
- **DR (Disaster Recovery):** Backup environment for critical scenarios.

3. Environment Variables Configuration

- **Purpose:** Environment variables are critical for securely handling sensitive data such as API keys, database credentials, and configuration settings that differ across environments.
- **Setup Steps:**

1. **Create a .env File:**
 - Added sensitive variables like:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
API_KEY=your_api_key
```

- 1. **Securely Upload Variables:**

- Used Vercel's dashboard to securely store and manage these variables, ensuring they are not exposed in the codebase.
- **Validation:**
 - Ensured the application correctly accessed and utilized these variables in the staging environment.
 - Verified no sensitive information was visible in the browser or logs.

4. Testing in the Staging Environment

- **Functional Testing:**
 - Verified features like product listing, search, and cart operations using **Cypress**.
 - Validated API interactions with **Postman**.
- **Performance Testing:**
 - Analyzed load times and responsiveness using **Lighthouse**.
 - Results:
 - Performance Score: 95%
 - Accessibility Score: 98%
 - Best Practices: 100%
- **Security Testing:**
 - Ensured HTTPS was enabled.
 - Tested input fields for SQL injection prevention.
 - Verified secure handling of environment variables.
- **Responsiveness Testing:**
 - Verified responsiveness across multiple screen sizes and devices.

5. Documentation and Repository Organization

- **Notes folder:** Created a detailed file summarizing all six days of activities, deployment steps, and testing results.
- **Folder Structure:** Organized project files with a clear hierarchy:
 - `src/`: Application source code.
 - `public/`: Static assets.
 - `notes/`: Reports, test cases, and documentation.
 - `README.md`: Summary of the project.