Day 6 - Deployment Preparation and Staging Environment Setup

Objective:

Day 6 focuses on preparing your marketplace for deployment. This involves setting up a staging environment, configuring hosting platforms, and testing the application in a production-like environment. This process will help make your marketplace ready for production.

Key Steps We Implemented:

Step 1: Hosting Platform Setup

- **Choosing the Platform:**
  We selected Vercel because it offers simple and fast deployment and has seamless integration with Next.js.
- **Connecting GitHub Repository:**
  We linked our GitHub repository with Vercel. This step was completed easily, and we configured the build setup for our repository.
- **Configuring Build Settings and Deployment:**
  We configured the build and deployment settings correctly to ensure successful staging builds. This included setting up build scripts.

Step 2: Environment Variables Configuration

- **Creating .env File:**
  We securely stored sensitive data such as API keys and tokens in a .env file. The file included variables such as:
  - NEXT_PUBLIC_SANITY_PROJECT_ID
  - NEXT_PUBLIC_SANITY_DATASET
  - API_KEY
- **Uploading Variables to Hosting Platform:**
  We uploaded our environment variables securely through the hosting platform's dashboard to ensure secure deployment in the production environment.

Step 3: Deploying the Application to Staging

- **Deployment:**
  We deployed the application to Vercel's staging environment.

- **Deployment Validation:**
  After the deployment, we verified that the build process completed successfully without errors, and the application loaded correctly.

Step 4: Staging Environment Testing

- **Functional Testing:**
  We tested major features of the application, such as product listing, cart functionality, and search features. We used tools like Cypress to test workflows.
- **Performance Testing:**
  We used Lighthouse and GTmetrix to analyze the speed and responsiveness of the application.
- **Security Testing:**
  We ensured that HTTPS was enabled, API keys were handled securely, and input fields were protected against SQL injection.
- **Responsiveness Testing:**
  We made the layout responsive and ensured it displayed correctly on both mobile and desktop screens.
- **Test Case Reporting:**
  We documented the test cases in a CSV file, which included the following information:
  - **Test Case ID**
  - **Description**
  - **Steps**
  - **Expected Result**
  - **Actual Result**
  - **Status**
  - **Remarks**