

Day 6

DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

Step 1: Hosting Platform Setup

1. Choose a Platform:

- I have chosen **Vercel** for hosting due to its seamless integration with Next.js and automatic deployments.

2. Connect Repository:

- I have connected my **GitHub repository** to Vercel.
- Configured the necessary build settings with `npm run build`.
- Added environment variables for secure operations.

Step 2: Configure Environment Variables

1. Create a .env File:

- I have created a `.env` file with the required sensitive variables:

2. Upload Variables to Hosting Platform:

- I have securely added the environment variables in **Vercel's dashboard** under the **Environment Variables section** to ensure they are available during deployment.

Step 3: Deploy to Staging

1. Deploy Application:

- I have deployed the application to a **staging environment** using **Vercel** for pre-production testing.

2. Validate Deployment:

- Ensured the build process completed successfully without any errors.
- Verified the basic functionality, including:
 - Navigation across pages
 - API integrations

- Environment variable configurations
- Responsiveness on different screen sizes

Step 4: Staging Environment Testing

1. Testing Types:

- **Functional Testing:**
 - Verified features such as:
 - Product listing display
 - Search functionality
 - Cart operations and checkout process
- **Performance Testing:**
 - Used **Lighthouse** and **GTmetrix** to analyze:
 - Page load speed
 - Time to interactive
 - Overall performance score
- **Security Testing:**
 - Validated input fields for vulnerabilities
 - Ensured the use of **HTTPS**
 - Verified secure API communications and protected environment variables

2. Test Case Reporting:

- I have documented the test cases in a **CSV file** in the day 5 directory do check it out:

3. Performance Testing:

- I conducted performance testing using **Lighthouse** to analyze metrics such as:
 - Performance score
 - First Contentful Paint (FCP)
 - Time to Interactive (TTI)
 - Cumulative Layout Shift (CLS)
- The generated **Lighthouse performance report** has been uploaded to my **day 5 directory** for documentation and reference.

Step 5: Documentation Updates

1. Create **README.md:**

- I have created a **README.md** file with the following sections:
 - **Project Overview:** Brief summary of the application
 - **Deployment Steps:** Detailed instructions for deployment
 - **Environment Variables:** Instructions for setting up **.env** variables

- **Test Case Results:** Summary of functional, performance, and security testing
- **Performance Report:** Lighthouse report details and improvements

2. Organize Project Files:

- Ensured all files from **Days 1 to 6** are well-structured in the folder hierarchy in my repo.

Checklist for Day 6:

Deployment Preparation	✓
Staging Environment Testing	✓
Documentation	✓
Form Submission	✓
Final Review	✓