Day 06

Deployment Preparation And Staging Environment Setup

By Zahida Imran

Deployment Report

1. Deployment Strategy Planning:

Hosting Platform: Vercel

- Selected due to its effortless compatibility with contemporary frameworks like Next.js, rapid deployment capabilities, and robust features like serverless functions.
- Configured to enable automated deployment pipelines by linking the GitHub repository.

Backend Interaction

- Connected to Sanity CMS for efficient content management and updates.
- Leveraged third-party APIs for product data and additional functionalities, ensuring all API endpoints were thoroughly tested for smooth frontend communication.

2. Environment Variable Configuration:

Local .env File

Included sensitive variables:

- NEXT_PUBLIC_SANITY_PROJECT_ID
- SANITY_API_TOKEN
- NEXT_PUBLIC_SANITY_DATASET

Vercel Dashboard Configuration

- Added the specified variables to the "Environment Variables" section within the Vercel Project Settings.
- Verified that the variables were assigned to the appropriate scope, such as Production, Preview, or Development.

3. Staging Environment Setup:

Initial Deployment

- Deployed the application to the staging environment utilizing Vercel's automated build system.
- Confirmed that the deployment log was error-free and the application was accessible via the staging URL.

Validation Check

- Verified that static assets, dynamic routes, and backend data loaded as expected.

4. Staging Environment Testing:

Performance Testing

Tools Used: Lighthouse

Key Metrics:

- Page Load Time: Achieved sub-2 second load times for most pages.
- Performance Score: Scored 85+ on Lighthouse.

Security Testing

Validations: Environment variables were used to secure API keys.

Responsiveness Testing

Verified cross-device compatibility by testing on:

- Mobiles
- Tablets
- Desktops

Error Handling

Ensured proper handling of 404 errors using custom error pages.