Day 6 - Deployment Preparation and Staging Environment Setup Report Prepared

by: Ammara Rajput

Objective

The main goal for today was to prepare the marketplace for deployment by setting up a staging environment. This included configuring hosting platforms, securing environment variables, conducting staging environment testing, and updating project documentation to ensure a smooth deployment process.

Key Outcomes

1. Fully configured and deployed a staging environment.
2. Conducted functional, performance, and security testing.
3. Documented all steps, issues, and results.
4. Updated the project GitHub repository with organized files and a README.md summarizing the project.

Steps I Followed

Step 1: Deployment Strategy Planning

- Platform Selection: Choose Vercel as the hosting platform for its seamless integration with Next.js.
- Backend Services Integration: Configured Sanity CMS and third-party APIs to ensure they interacted correctly with the application.

Step 2: Hosting Platform Setup

1. **Connected GitHub Repository:**
   - Linked the repository to Vercel and configured build settings.
   - Added deployment scripts for staging.
2. **Environment Variable Configuration:**
   - Created a .envfile containing sensitive information like API keys and tokens:
   - NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
   - NEXT_PUBLIC_SANITY_DATASET=production
   - API_KEY=your_api_key
   - Uploaded these variables securely to Vercel’s dashboard.

Step 3: Staging Environment Deployment

- Deployed Application: Successfully deployed the marketplace application to the staging environment.
- Validation: Ensured that the build was completed without errors and that the staging site loaded as expected.

Step 4: Staging Environment Testing

1. **Functional Testing:**
   - Used Cypress to verify workflows such as product listing, cart operations, and search functionality.
   - Validated API responses using Postman.
2. **Performance Testing:**
   - Ran Lighthouse to analyze speed, responsiveness, and load times.
   - Results: Performance score was 92/100, indicating excellent optimization.
3. **Security Testing:**
   - Just to let you know, I assure you that HTTPS was enabled.
   - Validated secure handling of API keys and input fields to prevent SQL injection.
4. **Test Case Reporting:**
   - Documented test cases in CSV format. Sample: Expected Result.pdf

Step 5: Documentation Updates

1. **README.md Creation:**
   - Summarized all six days of activities and deployment steps.
   - Highlighted the folder structure and included links to key resources.
2. **Organized Files:**
   - Structured project files into a clear hierarchy, including:
     - documents/ folder for all reports and plans.
     - src/ for source code.
     - public/ for static assets.
     - Test reports and performance results.

Deliverables and Submission

1. **Deployed Staging Environment:**
   - Link: Staging Environment
2. **GitHub Repository:**
   - Github repo
   - Documents folder with all Day 1 to Day 6 reports.
   - Test case reports in CSV format.
   - Performance testing results.
   - Organized project files.

Challenges I Faced

1. **Environment Variable Issues:**

    - Resolved by carefully checking variable names and ensuring they matched the application’s configuration.
2. **Testing in Staging:**
    - Minor discrepancies in API responses were debugged using Postman and resolved.

Best Practices Followed

1. **Environment Security:**
    - Secured all sensitive information in .env files and hosted dashboards.
2. **Clean Code Practices:**
    - Used modular utility functions for API calls.
    - Added comprehensive comments for clarity.
3. **Version Control:**
    - Made frequent Git commits with clear messages.
4. **Data Validation**:
    - Validated all data types and documented unresolved issues.

What I Achieved Today

- Successfully deployed a staging environment for the marketplace.
- Conducted thorough testing and documented results.
- Organized all project files professionally.