By Muhammad Abdullah

Day 6 - Deployment Preparation and Staging Environment Setup

Introduction

- Today's focus is on deployment preparation by setting up a staging environment.
- We will configure hosting, manage environment variables securely, and conduct staging tests.
- Ensuring a smooth transition to a production-like environment is the goal.

---

1. Selecting a Hosting Platform

- Vercel is the recommended platform due to its ease of deployment and automatic integrations.
- It allows seamless linking with GitHub repositories, enabling quick and continuous deployments.

---

2. Connecting GitHub Repository to Vercel

- **Step 1**: Sign in to Vercel and create a new project.
- **Step 2**: Select the GitHub repository containing your marketplace project.
- **Step 3**: Configure build settings and specify necessary scripts for a smooth deployment.
- **Step 4**: Set up automatic deployments so that any future code changes reflect in staging.

---

3. Configuring Environment Variables Securely

- **Step 1**: Create an .env.local file in your project with required keys.
- **Step 2**: Add NEXT_PUBLIC_SANITY_PROJECT_ID, API_KEY, and other secrets.
- **Step 3**: Navigate to Vercel's dashboard > Project settings > Environment Variables.
- **Step 4**: Enter variables securely to ensure sensitive data is not exposed in the codebase.

---

4. Deploying to a Staging Environment

- **Step 1**: Push the latest code changes to GitHub.
- **Step 2**: Deploy the application using Vercel's automatic build process.
- **Step 3**: Ensure the deployment completes without any errors.
- **Step 4**: Test the basic functionality of the application in the staging environment.

---

5. Staging Environment Testing

a) Functional Testing (Already done in a previous milestone)

- Verify key features such as product listing, search functionality, and cart operations.
- Use Cypress for automated UI testing and Postman for API validation.

b) Performance Testing

- Used GTmetrix(in a previous milestone) to analyze page load speed, responsiveness, and efficiency.
- Optimize images, minimize render-blocking resources, and ensure fast response times.

c) Security Testing

- Ensure **HTTPS is enabled** and input fields are validated.
- Check for secure handling of **API keys and authentication data**.

---

6. Organizing Project Files and Documentation

- Maintain a **clear folder structure**:
  - `DOCUMENTS/` – Contains all planning and testing documentation.
  - `furniro/` – Source code of the application.
  - `furniro/src/sanity` – Sanity schema and client files
  - `furniro/src/app` – Layout and main page, routes and dynamic routes
  - `furniro/src/components` – Reusable components and simple components
  - `README.md` – A complete guide summarizing the project and setup instructions.
- Document all test cases in a CSV file with details on expected vs. actual results.
- Submit performance reports generated from Lighthouse or GTmetrix.

---

7. Finalizing Deployment Strategy

- Ensure the project is ready for production by resolving any staging issues.
- Maintain a professional GitHub repository with well-structured documentation.

Conclusion

- By completing Day 6, we have successfully deployed and tested the application in a staging environment.
- The marketplace is now production-ready, with all necessary optimizations and security measures in place.
- The final step is ensuring the GitHub repository is properly structured and submitting the required documents.

Checklist for Day 6:

Deployment Preparation: ✔

Staging Environment Testing: ✔

Documentation: ✔

Form Submission: ✔

Final Review: ✔