Day 6 Hackathon Report: Deployment Preparation and Staging Environment Setup 🚀

1. Deployment Strategy Planning

Hosting Platform Selection:

- The application was deployed on **Vercel**, chosen for its seamless deployment process and high scalability.
- **GitHub** was integrated as the version control system, ensuring smooth CI/CD workflows with Vercel.

Backend Service Interaction:

- Successfully integrated **Sanity CMS** for efficient content management.
- Validated seamless communication with third-party APIs to ensure smooth data retrieval and processing.

---

2. Environment Variable Configuration

Securing Sensitive Data:

- A local **.env file** was created to store API keys, database credentials, and other sensitive information.
- Environment variables were securely uploaded to **Vercel's dashboard**, preventing exposure in the codebase.

Hosting Platform Configuration:

- Configured and verified all required environment variables within **Vercel**.
- Ensured sensitive information was **not exposed** in logs or the public repository.

---

3. Staging Environment Setup

Deployment:

- Successfully deployed the application to a **staging environment** using Vercel.
- Verified that the deployment **build process completed without errors**.
- Ensured all pages rendered correctly and application functionalities operated as expected.

Staging Environment Testing:

- **Functional Testing:**
  - Employed **Cypress** to validate user interactions and workflows.
  - Tested core functionalities such as **product listing, cart operations, and API responses** – all passed successfully.
- **Performance Testing:**
  - Conducted **Lighthouse performance analysis**.
  - Load time optimized to **~1.2 seconds**.
  - Achieved a **performance score of 96** with enhanced responsiveness.
- **Security Testing:**
  - Used **Postman** to validate API responses and confirm secure request handling.
  - Ensured **input sanitization** to mitigate risks like SQL injection and XSS attacks.
  - Verified **HTTPS enforcement** for secure data transmission.
  - Checked the correct handling of **API keys and environment variables**.

Documentation of Test Results:

- Documented all functional, performance, and security test findings.
- No critical issues detected – the system was ready for production deployment.

---

4. Steps for Implementation

Step 1: Hosting Platform Setup

1. **Platform Selection:**
   - Chose **Vercel** for its streamlined deployment and automated builds.
2. **Repository Connection:**
   - Linked the **GitHub repository** with Vercel.
   - Configured **build settings and deployment scripts**.

Step 2: Configure Environment Variables

1. Created a **.env file** containing essential environment variables.
2. Uploaded variables securely to **Vercel's environment settings**.

Step 3: Deploy to Staging

1. **Deploy Application:**

    - Successfully deployed the application to the **staging environment** via Vercel.

2. **Validate Deployment:**
    - Ensured the **build completed without errors**.
    - Conducted comprehensive functionality and performance tests in the staging environment.

Final Thoughts

- **Secure deployment processes** were implemented to protect sensitive data.
- **Automated testing and performance analysis** ensured the application met quality benchmarks before the final release.
- The **staging environment closely resembled production**, providing an accurate testing ground for validation before the official launch.

🚀 The application is now fully optimized and deployment-ready!