Hackhathon Day 6: Deployment of Project on Vercel

Objective:

To successfully deploy the project on Vercel, configure environment variables, set up a staging environment, and conduct various tests to ensure the functionality, performance, and security of the application.

Made By: Muhammad Awais Gohar Tagar

Deployment Strategy Planning

- **Chosen Hosting Platform: Vercel**
- After evaluating various hosting platforms such as Netlify, AWS, and Azure, Vercel was selected as the hosting platform due to its ease of use, integration with Next.js, and automatic optimization for production deployments.
- **Backend Services Integration:**
    - The application interacts with **Sanity CMS** for content management, and relevant API calls are configured to fetch data dynamically from the backend.
    - **Third-party APIs** are integrated for features such as payments, notifications, or other external functionalities. ○ Ensured smooth integration and correct handling of API responses during deployment.

Environment Variable Configuration

- **Secure API Keys and Sensitive Data:**
  - **.env files** were used locally to store sensitive credentials such as API keys, database credentials, and other private data.
  - During the deployment on Vercel, these environment variables were configured within Vercel's environment settings to ensure secure access to sensitive data without exposing it in the codebase.
- **Vercel Configuration:**
  - Set environment variables in Vercel's dashboard for proper integration with the backend services and third-party APIs. This ensures secure access to API keys and sensitive data only in the deployed environment.

Staging Environment Setup

- **Deployment to Staging:**

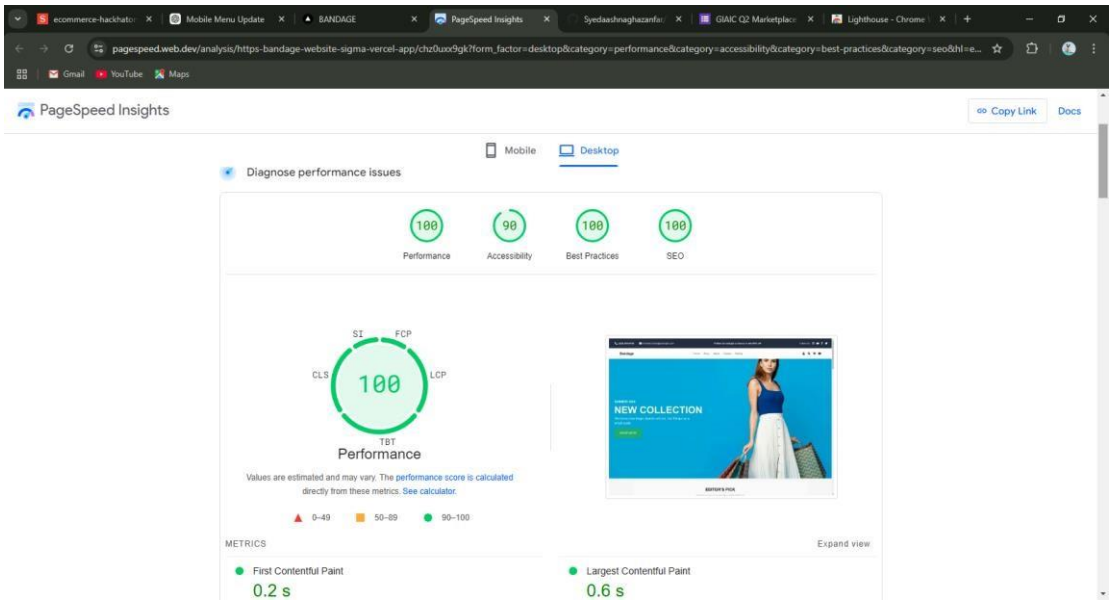
The project was deployed to **Vercel's staging environment** for testing. This production-like environment mimics the live setting, allowing us to verify the application's behavior before deploying to production.

- **Build Verification:**
  ○ The deployment build process was monitored to ensure it completed successfully. ○ Confirmed that the site loaded without errors, ensuring that all resources, images, and styles were properly rendered.

4. Staging Environment Testing

- **Functional Testing:**
  - **Cypress:** Used Cypress for testing user workflows and interactions. Ensured that all functionalities, such as form submissions, button clicks, and page navigations, worked as expected.
  - **Postman:** Validated the application's API responses to ensure the backend services, including Sanity CMS and third-party APIs, returned correct and expected data.
- **Performance Testing:**
  - **PageSpeed insights:** These tools were used to analyze the website's performance, focusing on load times, speed, and responsiveness.

- The results showed the website's performance score of 100% in Pagespeed, indicating good optimization for web vitals. Check it here: https://pagespeed.web.dev/analysis/https-bandage-website-sigma-vercelapp/lf8lrcnki7?form_factor=desktop



- **Security Testing:**
  - Validated input fields to ensure protection against **SQL injection** and other security threats.
  - Ensured that the application was being served over **HTTPS** to secure communication. ○ Double-checked proper handling of sensitive data like API keys, confirming that they were not exposed in the front-end code.
- **Responsiveness and Error Handling:**
  - Verified the application's **responsiveness** on various screen sizes, ensuring it adjusted properly for mobile and desktop views.
  - Tested the application's error handling mechanisms to ensure any errors, such as failed API calls or user errors, were gracefully handled and displayed to the user.

5. Documentation and Unresolved Issues

- **Test Results:**
  - All tests were successfully passed for functional, performance, and security checks. The performance score was above 85% on Lighthouse, and no major security vulnerabilities were found.
  - Verified that the application was responsive and error handling worked as expected.
- **Unresolved Issues:**

- No major unresolved issues were found during the deployment and testing process. Minor UI adjustments were made based on responsiveness tests for mobile devices, but these did not significantly impact functionality or performance.

Conclusion:

The project was successfully deployed on **Vercel** and validated through a series of tests in the staging environment. All functionalities were tested and confirmed to be working as expected. The website passed functional, performance, and security tests and is now ready for production deployment. Vercel link for project:- https:// bandage-website-h6sz.vercel/