Hackathon Day 6: Bandage Marketplace Template Deployment Preparation and Staging Environment Setup

Objective: The goal of Day 6 in the Bandage project is to ensure the system is ready for deployment by setting up a staging environment, configuring hosting platforms, and preparing it for customer-facing use. The focus is on creating a production-like environment for testing, ensuring the application functions seamlessly. Additionally, the project emphasizes understanding and managing different environments such as non-production (TRN, DEV, SIT) and production (UAT, PROD, DR), following industry-standard deployment practices.

Key Learning Outcomes:

1. Build dynamic frontend components that fetch and display data from Sanity CMS or APIs.
2. Implement reusable and modular components for easier maintenance and scalability.
3. Apply state management techniques to handle data flow across components.
4. Focus on responsive design and implement UX/UI best practices.
5. Prepare for real-world client projects by replicating professional workflows.

Professional Environment Types:

1. **TRN (Training)**
   - **Purpose:** Used for onboarding new team members and providing practice environments.
   - **Key Feature:** Allows users to familiarize themselves with the system without affecting active environments.

1. **DEV (Development)**
   - **Purpose:** A dedicated environment for developers to write and test code locally.
   - **Key Feature:** Supports iterative coding and debugging without impacting production systems.
2. **SIT (System Integration Testing)**
   - **Purpose:** Validates the integration between different systems and components.
   - **Key Feature:** Ensures seamless communication and compatibility between subsystems.

3. **UAT (User Acceptance Testing)**
   - **Purpose:** Allows stakeholders to test application functionality and verify that it meets business requirements.
   - **Key Feature:** Confirms that the system is ready for production deployment by aligning with user expectations.
4. **PROD (Production)**
   - **Purpose:** The live, customer-facing environment where the application is operational for end-users.
   - **Key Feature:** Ensures high availability, performance, and security for real-world usage.
5. **DR (Disaster Recovery)**
   - **Purpose:** Serves as a backup environment for critical situations such as system failures or disasters.
   - **Key Feature:** Enables rapid recovery and minimizes downtime during emergencies.

Key Areas of Focus:

1. Deployment Strategy Planning Deployed the application on Vercel for staging and production. Integrated with Sanity CMS for dynamic content using tokens and dataset IDs

2. Environment Variable Configuration Stored sensitive data (API keys, tokens) in .env.local file. Configured environment variables securely in Vercel Dashboard for deployment.

3. Staging Environment Setup Deployed the application to Vercel and verified successful deployment. Checked content fetching from Sanity CMS. Staging Environment Testing

4. Staging Environment Testing Conducted Cypress functional tests, Postman API validation, and Lighthouse performance tests. Ensured security with HTTPS, proper data handling, and verified responsiveness across devices.

5. Documentation Updates Created a README.md file with all deployment instructions, configurations, and test results. Included all reports in the GitHub repository.

Steps for Implementation

Step 1: Hosting Platform Setup Platform Chosen: Vercel was selected for quick and easy deployment.

Connect Repository: Successfully connected the GitHub repository to Vercel for automatic deployments.

Configured build settings and added the necessary scripts for deployment in the Vercel dashboard https://github.com/RaoAsadMehmood/bandage-store

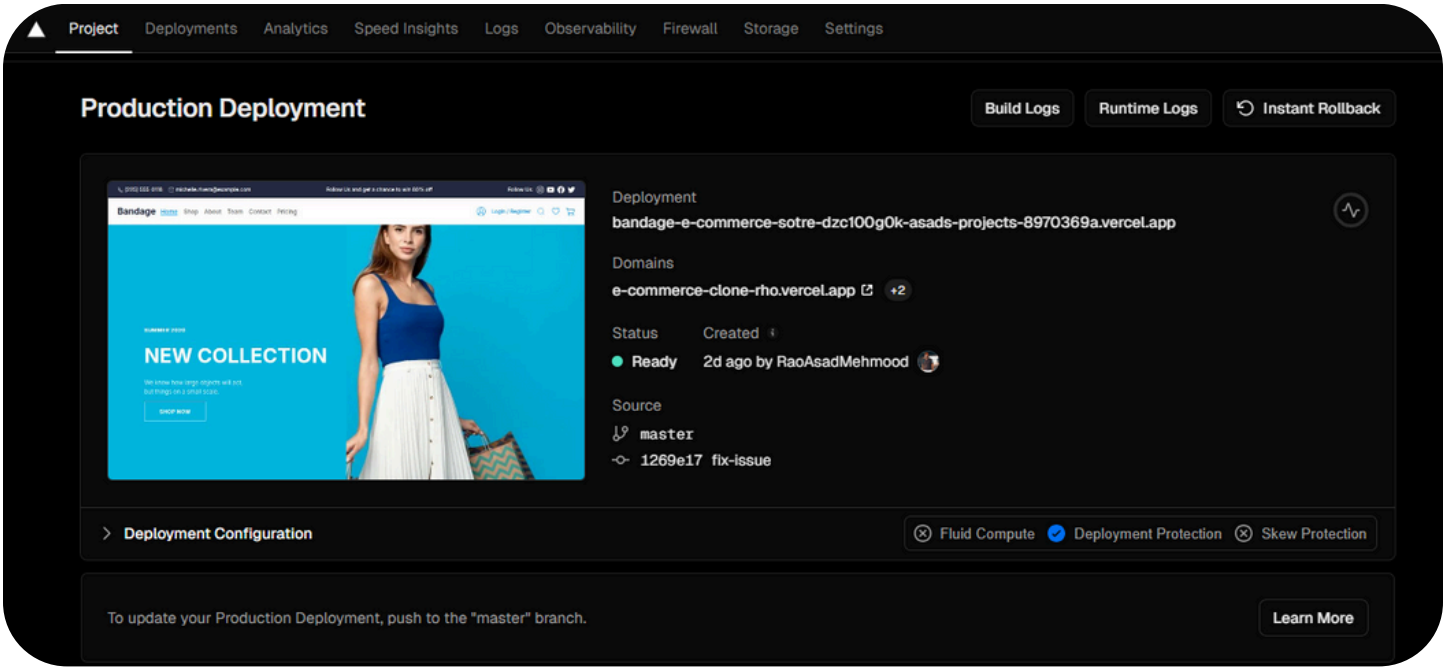
Step 2: Configure Environment Variables Create .env.local File: Created the .env.local file to store sensitive data like API keys and tokens.

```
$ .env.local
1    NEXT_PUBLIC_SANITY_PROJECT_ID="_____"
2    NEXT_PUBLIC_SANITY_DATASET="_____"
3    NEXT_PUBLIC_SANITY_TOKEN="_____"
```

Upload Variables to Vercel: Uploaded the environment variables to Vercel using the platform's dashboard for secure handling.

Step 3: Deploy to Staging Deploy Application: Deployed the application to Vercel's staging environment for testing.

Validate Deployment: Ensured the deployment build was completed without errors. Verified that the application was loading correctly, and all content was fetched properly from Sanity CMS.



Step 4: Staging Environment Testing

1. Testing Types

- **Functional Testing:**
  Verified the following features:
  - **Product Listing:** Ensured that all products were listed correctly.
  - **Product Details:** Verified that the product details page displayed the correct information.
  - **User Profile:** Checked user login, profile updates, and profile display.

- **Cart Operations:** Ensured products could be added, removed, and quantities updated in the cart.
- **Wishlist:** Validated the ability to add and remove products from the wishlist.
- **Category:** Ensured that categories displayed the correct product listings and filtered properly.
- **Dynamic Routing:** Verified that dynamic routing worked correctly for product and category pages.

- **Performance Testing:**
  - Used **Lighthouse** and **GTmetrix** to analyze the performance, speed, and responsiveness of the application.
  - Ensured the application was optimized for various devices, screen sizes, and network conditions to deliver a smooth user experience.
- **Security Testing:**
  - Validated input fields to protect against vulnerabilities like SQL injection and other malicious attacks.
  - Ensured that HTTPS was enabled for secure communication between the client and server.
  - Verified that sensitive data, including API keys and user credentials, was transmitted securely and stored safely to prevent data breaches.

2. Test Case Reporting

CSV Format

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Product Listing	Verify that all products are displayed on the homepage.	All products should be displayed correctly.	All products displayed as expected.	Pass	High	No issues found
TC002	Product Details	Click on any product to view its details.	The product detail page should load correctly.	Product detail page loaded without issues.	Pass	High	No issues found

TC003	Add to Cart	Click on 'Add to Cart' for a product.	Product should be added to the cart.	Product added successfully to the cart.	Pass	High	No issues found
TC004	Cart Operations	Add and remove items from the cart.	Cart should update correctly with the added/removed items.	Cart updates correctly when items are added or removed.	Pass	High	No issues found
TC005	Dynamic Routing	Click on a product to navigate to its detail page.	The correct product detail page should load.	Correct page loaded with the right details.	Pass	Medium	No issues found
TC006	Category Filter	Apply different category filters.	Products should be filtered based on selected category.	Products filtered correctly based on the selected category.	Pass	Medium	No issues found
TC007	Error Handling (Network)	Simulate a network failure and attempt to load a product.	An error message should be displayed, indicating a failure.	Error message displayed as expected.	Pass	Critical	Error handling works as expected.
TC008	Error Handling (Invalid Data)	Enter invalid data in a product search or form.	An appropriate error message should be displayed.	Error message displayed for invalid input.	Pass	High	Handled correctly with clear message.
TC009	Responsive Design	Test the website on multiple devices (desktop, tablet, mobile).	The design should adjust and be responsive on all devices.	Website design adapts correctly to various screen sizes.	Pass	Medium	No issues found on mobile/tablet.

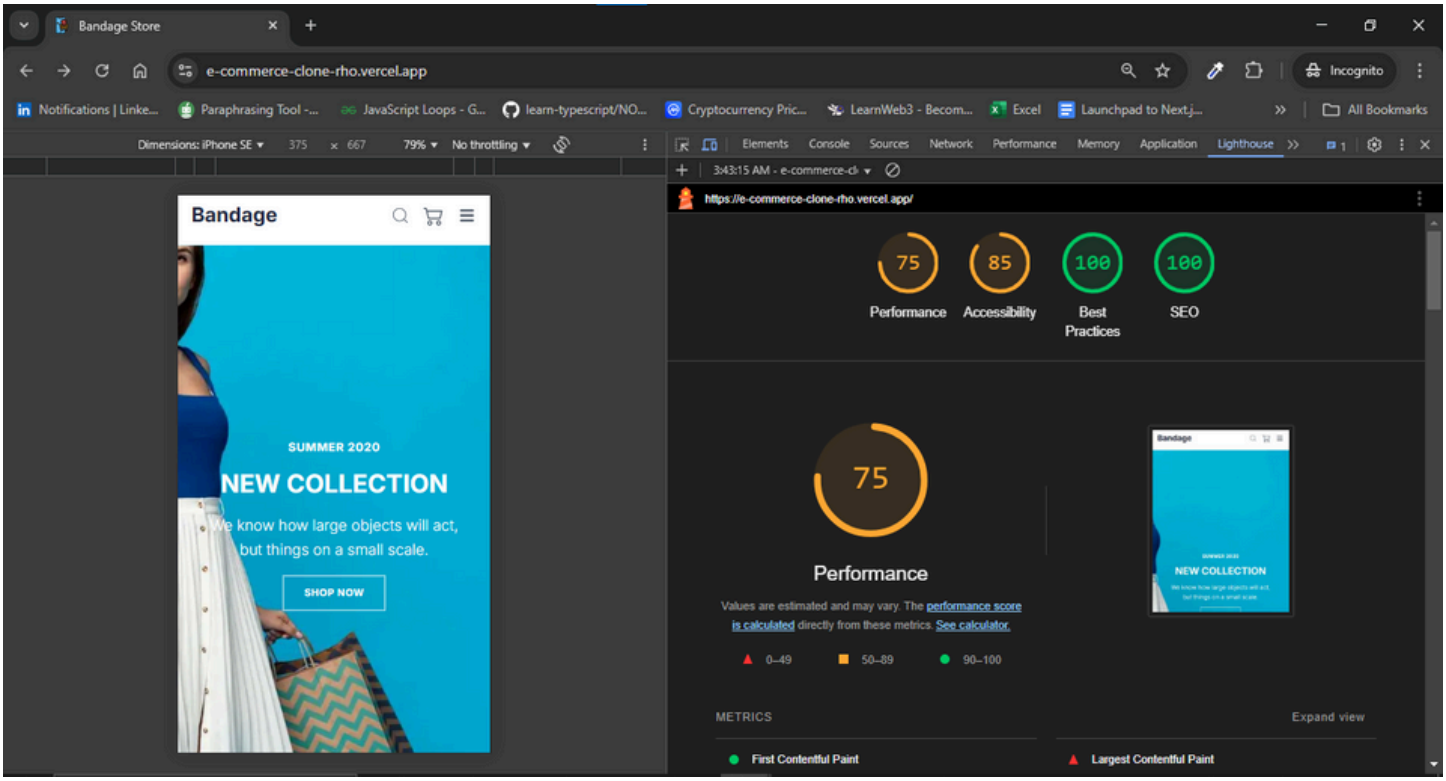
CSV Table

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Product Listing	Verify that all products are displayed on the h	All products should be displa	All products displayed a	Pass	High	No issues found
TC002	Product Details	Click on any product to view its details.	The product detail page shou	Product detail page loac	Pass	High	No issues found
TC003	Add to Cart	Click on 'Add to Cart' for a product.	Product should be added to t	Product added successfu	Pass	High	No issues found
TC004	Cart Operations	Add and remove items from the cart.	Cart should update correctly	Cart updates correctly w	Pass	High	No issues found
TC005	Dynamic Routing	Click on a product to navigate to its detail pag	The correct product detail pa	Correct page loaded wit	Pass	Medium	No issues found
TC006	Category Filter	Apply different category filters.	Products should be filtered b	Products filtered correct	Pass	Medium	No issues found
TC007	Error Handling (Network)	Simulate a network failure and attempt to loa	An error message should be d	Error message displayed	Pass	Critical	Error handling works as expected.
TC008	Error Handling (Invalid Data	Enter invalid data in a product search or form.	An appropriate error message	Error message displayed	Pass	High	Handled correctly with clear message.
TC009	Responsive Design	Test the website on multiple devices (deskto	The design should adjust and	Website design adapts c	Pass	Medium	No issues found on mobile/tablet.

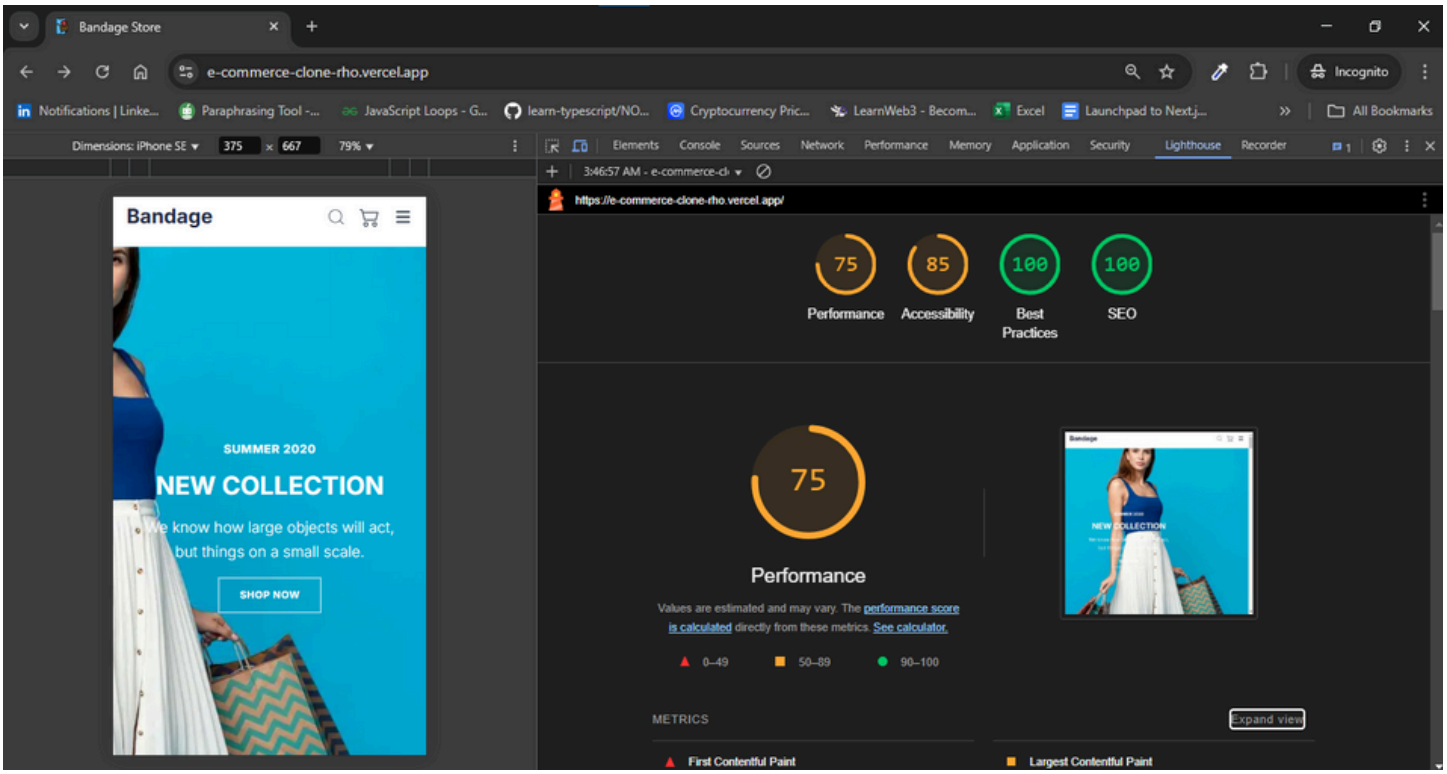
3. Performance Testing

Here is the performance report generated by lighthouse tools.

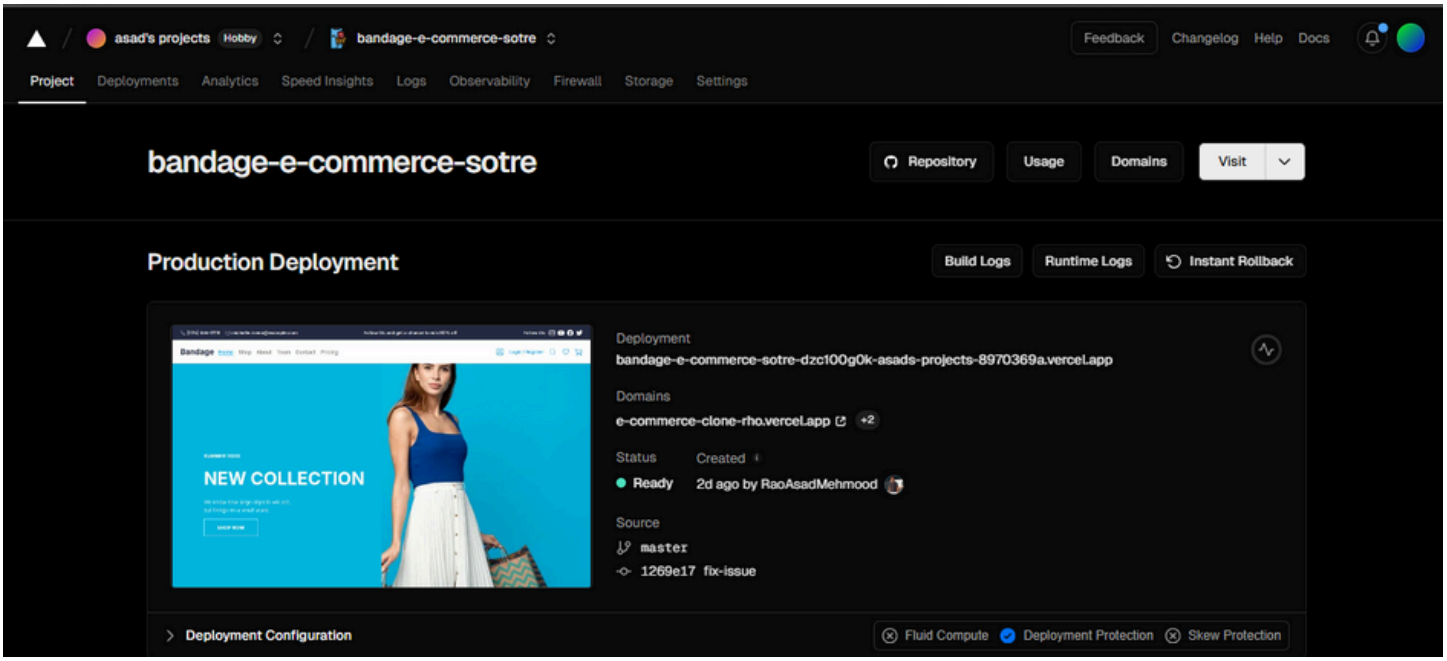
Performance Metrics for Mobile



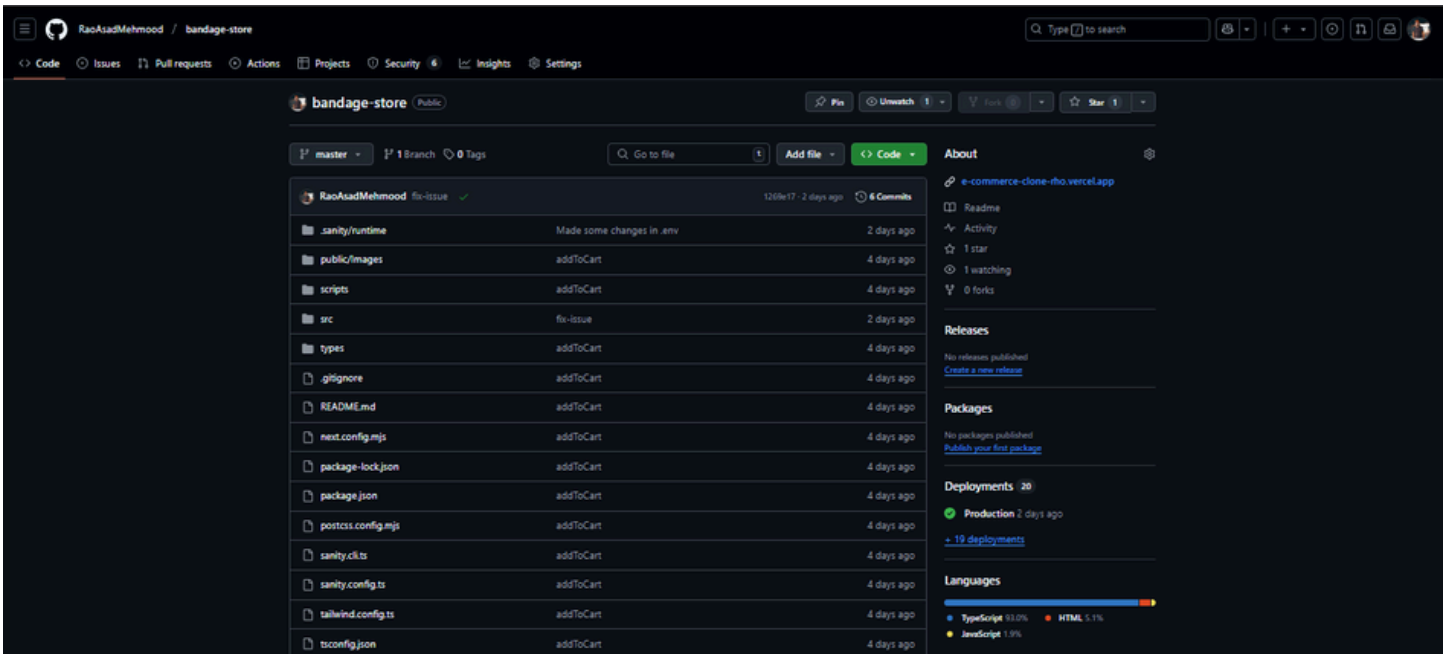
Performance Metrics for Desktop



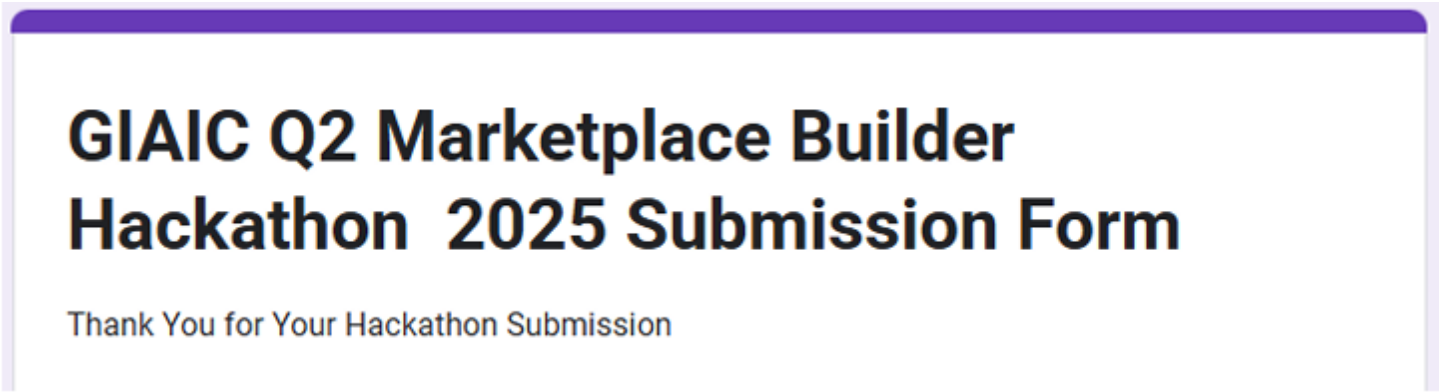
Vercel



GitHub



Form Submission



Conclusion for Deployment Preparation and Staging Step:

Day 6 was dedicated to setting up the staging environment for deployment, which involved configuring environment variables, testing functionality, and updating documentation. This process ensures a smooth and secure transition to the live platform, minimizing potential risks and enhancing the system's readiness for production deployment.

Prepared By: Rao Asad Mehmood.