URROOJ SAEED

Bandage

E-COMMERCE WEBSITE

DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP- Bandage

Objective:

The primary objective of Day 6 of the Bandage project is to prepare the system for deployment by establishing a staging environment, configuring hosting platforms, and ensuring it is ready for customer-facing use. The key focus is on creating a production-like environment for thorough testing, ensuring that the application functions seamlessly. Additionally, this phase emphasizes a comprehensive understanding and management of various environments, including non-production (TRN, DEV, SIT) and production (UAT, PROD, DR), while adhering to industry-standard deployment practices.

Key Learning Outcomes:

1. **Dynamic Frontend Development**:
   Construct dynamic frontend components capable of fetching and displaying data from Sanity CMS or APIs.
2. **Reusable and Modular Components**:
   Implement reusable, modular components to ensure easier maintenance and scalability across the application.
3. **State Management**:
   Apply effective state management techniques to efficiently handle data flow across different components.
4. **Responsive Design and UX/UI Practices**:
   Emphasize responsive design principles and implement user experience (UX) and user interface (UI) best practices to enhance usability.
5. **Real-World Project Preparation**:
   Prepare for real-world client projects by replicating professional workflows in a controlled development environment.

Professional Environment Types:

1. **TRN (Training)**
   - **Purpose**: Utilized for onboarding new team members and providing practice opportunities.
   - **Key Feature**: Allows users to familiarize themselves with the system without affecting live environments.
2. **DEV (Development)**
   - **Purpose**: A dedicated environment where developers can write and test code locally.
   - **Key Feature**: Facilitates iterative coding and debugging processes without disrupting production systems.

3. **SIT (System Integration Testing)**
   - **Purpose**: Validates the integration between various systems and components.
   - **Key Feature**: Ensures seamless communication and compatibility between different subsystems.
4. **UAT (User Acceptance Testing)**
   - **Purpose**: Allows stakeholders to test the application's functionality and ensure it meets business requirements.
   - **Key Feature**: Confirms that the system is ready for deployment by ensuring alignment with user expectations.
5. **PROD (Production)**
   - **Purpose**: The live environment where the application serves real users.
   - **Key Feature**: Prioritizes high availability, performance, and security to ensure smooth operation for end-users.
6. **DR (Disaster Recovery)**
   - **Purpose**: Acts as a backup environment for critical situations such as system failures or disasters.
   - **Key Feature**: Facilitates rapid recovery to minimize downtime during emergencies.

Key Areas of Focus:

1. **Deployment Strategy Planning**
   - Deployed the Bandage application on **Vercel** for both staging and production environments.
   - Integrated with **Sanity CMS** for dynamic content fetching using API tokens and dataset IDs.
2. **Environment Variable Configuration**
   - Stored sensitive information, such as API keys and tokens, in the **.env.local** file.
   - Configured environment variables securely within the **Vercel Dashboard** to ensure safe deployment.
3. **Staging Environment Setup**
   - Successfully deployed the application to **Vercel** and validated the deployment's success.
   - Ensured the application correctly fetched content from **Sanity CMS**.
4. **Staging Environment Testing**
   - Conducted functional tests using **Cypress**, validated APIs with **Postman**, and assessed performance with **Lighthouse**.
   - Ensured the system's security through **HTTPS**, verified proper data handling, and confirmed responsiveness across various devices.
5. **Documentation Updates**
   - Created a comprehensive **README.md** file detailing deployment instructions, environment configurations, and test results.
   - Included all relevant reports within the project's **GitHub repository**.

Steps for Implementation:

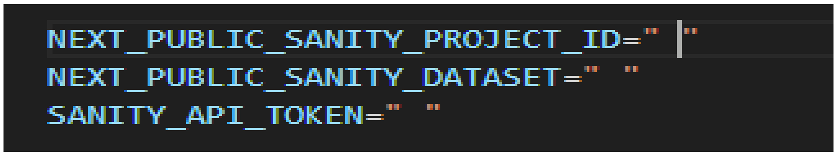
Step 1: Hosting Platform Setup

- **Platform Chosen**: Vercel
- **Repository Connection**: GitHub

Step 2: Configure Environment Variables

Create .env.local File

- A **.env.local** file was created to store sensitive data such as API keys and tokens securely. This file ensures that sensitive credentials are kept private and not exposed in the codebase.



```
NEXT_PUBLIC_SANITY_PROJECT_ID=" "
NEXT_PUBLIC_SANITY_DATASET=" "
SANITY_API_TOKEN=" "
```

Step 3: Deploy to Staging

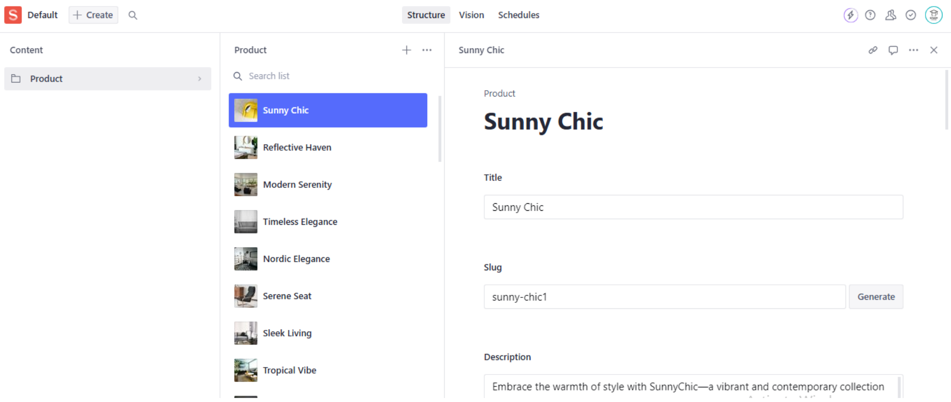
Deploy Application

- The Bandage application was successfully deployed to **Vercel's staging environment** for thorough testing before moving to production.

Validate Deployment

- The deployment build completed without errors, ensuring a successful setup.
- After deployment, it was verified that the application loaded correctly and all content was successfully fetched from **Sanity CMS**.

Products



Product

- Sunny Chic
- Reflective Haven
- Modern Serenity
- Timeless Elegance
- Nordic Elegance
- Serene Seat
- Sleek Living
- Tropical Vibe

Sunny Chic

Product

Sunny Chic

Title: Sunny Chic

Slug: sunny-chic1 Generate

Description: Embrace the warmth of style with SunnyChic—a vibrant and contemporary collection

Deployment

Step 4: Staging Environment Testing

1. *Testing Types*

Functional Testing

- **Product Listing**: Ensured all products were listed correctly.
- **Product Details**: Verified that the product details page displayed accurate information.
- **User Profile**: Checked user login functionality, profile updates, and profile display.
- **Cart Operations**: Verified the ability to add, remove, and update quantities of products in the cart.
- **Wish list**: Ensured users could add and remove products from their wish list.
- **Category**: Validated that categories displayed the correct product listings and filtered products accordingly.
- **Dynamic Routing**: Confirmed that dynamic routing worked properly for product and category pages.

Performance Testing

- **Lighthouse** tool was used to analyze the performance, speed, and responsiveness of the application.

Security Testing

- Input fields were tested for vulnerabilities like SQL injection and other attacks to ensure secure interactions.
- HTTPS was enabled for secure client-server communication.
- Sensitive data, including API keys and user credentials, was securely transmitted and stored to prevent data breaches.

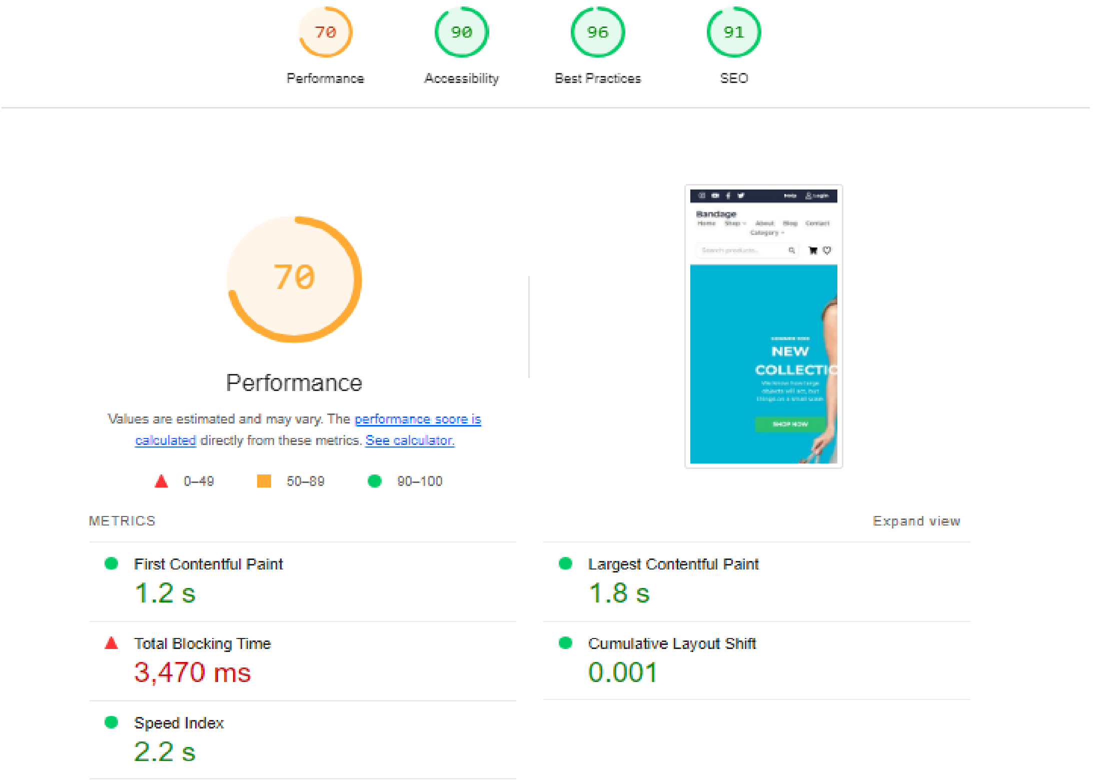
2. *Test Case Reporting*

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
1	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	No issues found
2	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with	Error message shown	Passed	Medium	Handled gracefully

				error message				
3	TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	Low	Works as expected
4	TC004	Cross Browsing, Open our website on different Browsers such as Yahoo	Working Correctly on every browser	Working Correctly on every browser	Responsive layout working as intended	Passed	Medium	Test successful
5	TC005	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	High	Test successful

3. ***Performance Testing***

Here is performance report generate by lighthouse tools;



70 Performance

90 Accessibility

96 Best Practices

91 SEO

70

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint
1.2 s

● Largest Contentful Paint
1.8 s

▲ Total Blocking Time
3,470 ms

● Cumulative Layout Shift
0.001

● Speed Index
2.2 s

Conclusion

Through the deployment of the Bandage application in a staging environment and subsequent rigorous testing, we have ensured that the system is secure, functional, and optimized for performance across various platforms. This comprehensive testing process, including functional, performance, and security assessments, has validated that all critical features are working as expected and that the application meets the highest standards of reliability and security. By configuring the staging environment, deploying the application seamlessly, and optimizing for multiple devices and conditions, we have created a robust foundation for the application's launch. This process has laid the groundwork for a smooth and successful transition to production, ensuring a user-friendly, secure, and high-performance platform for end users. Moving forward, the application is ready for further refinement and scaling as it progresses toward full deployment and customer-facing use.

SELF VALIDATION

- ☑ Deployment Preparation
- ☑ Staging Environment Testing
- ☑ Documentation
- ☑ Form Submission
- ☑ Final Review
- ☑ Done