Hackathon Day 6

Bandage Marketplace Template

Deployment Preparation and Staging Environment Setup

Objective

The objective of the Bandage project on Day 6 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 and ensuring the application operates seamlessly. Additionally, the project emphasizes understanding and managing 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 manage 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)**
    a. **Purpose**: Used for onboarding new team members and practice.
    b. **Key Feature**: Helps users get familiar with the system without impacting active environments.
2. **DEV (Development)**

    a. **Purpose**: Dedicated environment for developers to write and test code locally.
    b. **Key Feature**: Supports iterative coding and debugging without affecting production systems.

3. **SIT (System Integration Testing)**
    a. **Purpose**: Validates the integration between different systems and components.
    b. **Key Feature**: Ensures seamless communication and compatibility between subsystems.
4. **UAT (User Acceptance Testing)**
    a. **Purpose**: Allows stakeholders to test application functionality and validate that it meets business requirements.
    b. **Key Feature**: Ensures the system is ready for production deployment by aligning with user expectations.
5. **PROD (Production)**
    a. **Purpose**: The live, customer-facing environment where the application operates for end-users.
    b. **Key Feature**: Ensures high availability, performance, and security for real-world usage.
6. **DR (Disaster Recovery)**
    a. **Purpose**: Acts as a backup environment for critical situations such as system failures or disasters.
    b. **Key Feature**: Enables quick recovery and minimizes downtime in emergencies.

Key Areas of Focus:

1. **Deployment Strategy Planning**
    a. Deployed the application on Vercel for staging and production.
    b. Integrated with Sanity CMS for dynamic content using tokens and dataset IDs.
2. **Environment Variable Configuration**
    a. Stored sensitive data (API keys, tokens) in `.env.local` file.
    b. Configured environment variables securely in the Vercel Dashboard for deployment.
3. **Staging Environment Setup**
    a. Deployed the application to Vercel and verified successful deployment.

   b. Checked content fetching from Sanity CMS.

4. **Staging Environment Testing**
   a. Conducted Cypress functional tests, Postman API validation, and Lighthouse performance tests.
   b. Ensured security with HTTPS, proper data handling, and verified responsiveness across devices.
5. **Documentation Updates**
   a. Created a `README.md` file with all deployment instructions, configurations, and test results.
   b. 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.
  - Deployed URL: Hackathon Marketplace.
- **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.
  - GitHub Repository: Hackathon E-commerce.

Step 2: Configure Environment Variables

- **Create `.env.local` File**:
  - Created the `.env.local` file to store sensitive data like API keys and tokens.
- **Example**:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=your_dataset
```

- **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 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**
    a. **Functional Testing**:
        i. Verified the following features:
            1. Product Listing: Ensured all products were listed correctly.
            2. Product Details: Verified product details page displayed the correct information.
            3. User Profile: Checked user login, profile update, and profile display.
            4. Cart Operations: Ensured products could be added, removed, and quantities updated in the cart.
            5. Wishlist: Validated the ability to add and remove products from the wishlist.
            6. Category: Ensured categories displayed correct product listings and filtered accordingly.
            7. Dynamic Routing: Verified that dynamic routing worked properly for product and category pages.
    b. **Performance Testing**:
        i. Used Lighthouse and GTmetrix to analyze the performance, speed, and responsiveness of the application.
        ii. Ensured the application was optimized for various devices, screen sizes, and network conditions to deliver a smooth user experience.
    c. **Security Testing**:
        i. Validated input fields to ensure they were protected from vulnerabilities such as SQL injection and other malicious attacks.

ii. Ensured HTTPS was enabled for secure communication between the client and server.

iii. Verified that sensitive data, including API keys and user credentials, was transmitted securely and stored safely to avoid any data breaches.

2. Test Case Reporting

Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate Product Listing	Open product page > Verify products are listed correctly.	Products displayed correctly with accurate details.	Products displayed correctly.	Passed	High	Developer	No issues found
TC002	Test Product Details	Click on a product > Verify the details are displayed correctly.	Product details page shows accurate information.	Product details shown correctly.	Passed	High	Developer	No issues found
TC003	Test Cart Functionality	Add item to cart > Verify cart updates correctly.	Cart updates with the correct item and quantity.	Cart updated correctly.	Passed	High	Developer	Works as expected
TC004	Test Wishlist Functionality	Add item to wishlist > Verify wishlist updates.	Item is added to the wishlist.	Item added to wishlist.	Passed	Medium	Developer	Works as expected
TC005	Test Category Filter	Apply category filter > Verify filtered products.	Only products in the selected category are displayed.	Correct category products displayed.	Passed	Medium	Developer	No issues found

TC006	Test Dynamic Routing	Click on a product > Verify dynamic routing to product details.	Product detail page loads correctly.	Product page loaded correctly.	Passed	High	Developer	Works as expected
TC007	Test User Profile Management	Login > Update profile > Verify updated information.	User profile is updated and displayed correctly.	Profile updated successfully.	Passed	High	Developer	No issues found
TC008	Test Responsive Design	Resize browser window > Check layout adjustment.	Layout adjusts properly on different screen sizes.	Layout adjusted correctly.	Passed	High	Developer	Responsive verified

3. **Performance Testing**
    a. **Performance Metrics for Mobile**:
        i. Details generated using Lighthouse tools.
    b. **Performance Metrics for Desktop**:
        i. Details generated using Lighthouse tools.

Conclusion for Deployment Preparation and Staging Steps

Day 6 focused on setting up a staging environment for deployment, including configuring environment variables, testing functionality, and updating documentation. This ensures a smooth and secure transition to the live platform, minimizing risks and enhancing readiness for production.