DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP.

***Objective**: Day 6 focuses on preparing your marketplace for deployment by setting up a staging environment, configuring hosting platforms, and ensuring readiness for a customer facing application. Building on the testing and optimization work from Day 5, this stage emphasizes ensuring the marketplace operates seamlessly in a production-like environment. Students will also learn about industry-standard practices for managing divergent environments like non-production (TRN, DEV, SIT) and production (UAT, PROD, DR).*

Step 1: Hosting Platform Setup

1. Choose a Platform:

- I used vercel app for quick deployment.

2. Connect Repository:

- Linked my GitHub repository to the hosting platform "Vercel".

- Configure build settings and add necessary scripts for deployment.

Step 2: Configure Environment Variables

1. Create a `.env` File:

Include sensitive variables like API keys and tokens.

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id

NEXT_PUBLIC_SANITY_DATASET=production

API_KEY=your_api_key
```

2. Upload Variables to Hosting Platform:

- Use the hosting platform's dashboard to securely add environment variables.

Step 3: Deploy to Staging

1. Deploy Application:

- Deploy the application to a staging environment through the hosting platform "Vercel".

2. Validate Deployment:

- Ensure the build process completes without errors.

- Verify basic functionality in the staging environment.

Step 4: Staging Environment Testing

1. Testing Types:

- Functional Testing: Verify all features, such as product listing, Filtering, search, and cart operations.

- Performance Testing: Use Lighthouse to analyze speed and responsiveness.

2. Test Case Reporting:

Document all test cases in a CSV file with fields like Test Case ID, Description, Steps, Expected Result, Actual Result, Status, and Remarks.

Example Test Cases:

Test Case ID | Description | Steps | Expected Result | Actual Result | Status | Remarks

TC001 | Validate product listing | Open product page, verify items | Products displayed | Products displayed | Passed | No issues found

TC002 | Test API error handling | Disconnect API, refresh page | Show fallback message | Fallback message shown | Passed | Handled gracefully

TC003 | Check cart functionality | Add item to cart, verify cart | Cart updates correctly | Cart updates correctly | Passed | Works as expected

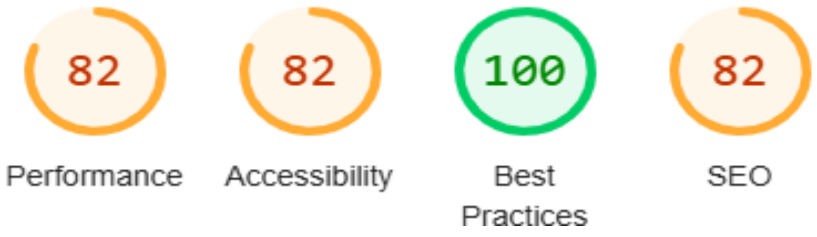
TC004 | Check Filter functionality | filter any items by their category | filter should update shop page as per customer requirement | Filter updates correctly | Passed | Works as expected

TC005 | Check Search functionality | user search items by name| Result should be related to user search | filter search correctly | Passed | Works as expected

TC004 | Test responsiveness | Resize browser window, check UI | Layout adjusts properly | Layout adjusts properly | Passed | Responsive verified.

3. Performance Testing:

- Submit a performance report generated by tools like Lighthouse in your GitHub repository.



Step 5: Documentation Updates

1. Create README.md:

- Summarize all project activities, including deployment steps and test case results.

2. Organize Project Files:

- Ensure all files from Days 1 to 6 are in a structured folder hierarchy.