Day 6 - Deployment Preparation & Staging Setup

Prepared by: Mohammad Danish

Project: E-commerce Marketplace - Nike

Overview & Goal

On Day 6, the focus is on preparing the marketplace for deployment by setting up a **staging environment**, configuring hosting, and ensuring readiness for a customer-facing application. The key objectives include:

- **Deploying the application using Vercel**.
- **Building the project with npm run build** before deployment.
- **Configuring and securing environment variables** for Clerk authentication and Sanity API keys.
- **Understanding and implementing staging environments for production-like testing**.
- **Conducting thorough testing, including functional, performance, and security validation**.
- **Documenting the deployment process and test results in a structured format**.
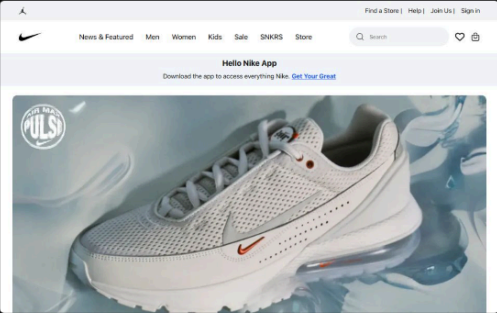-

Deployment & Staging Process

1️⃣ Hosting Platform Setup

Steps:

- Selected **Vercel** for hosting.
- Ran `npm run build` to compile and prepare the project for deployment.
- Configured **.env file** for Clerk authentication and Sanity API keys.
- Linked the **GitHub repository** to Vercel.
- Configured build and deployment settings for successful deployment.

-

2 **Configuring Environment Variables**

Steps:

- **Stored sensitive information in a .env file.**

- **Securely added environment variables to the Vercel dashboard.**
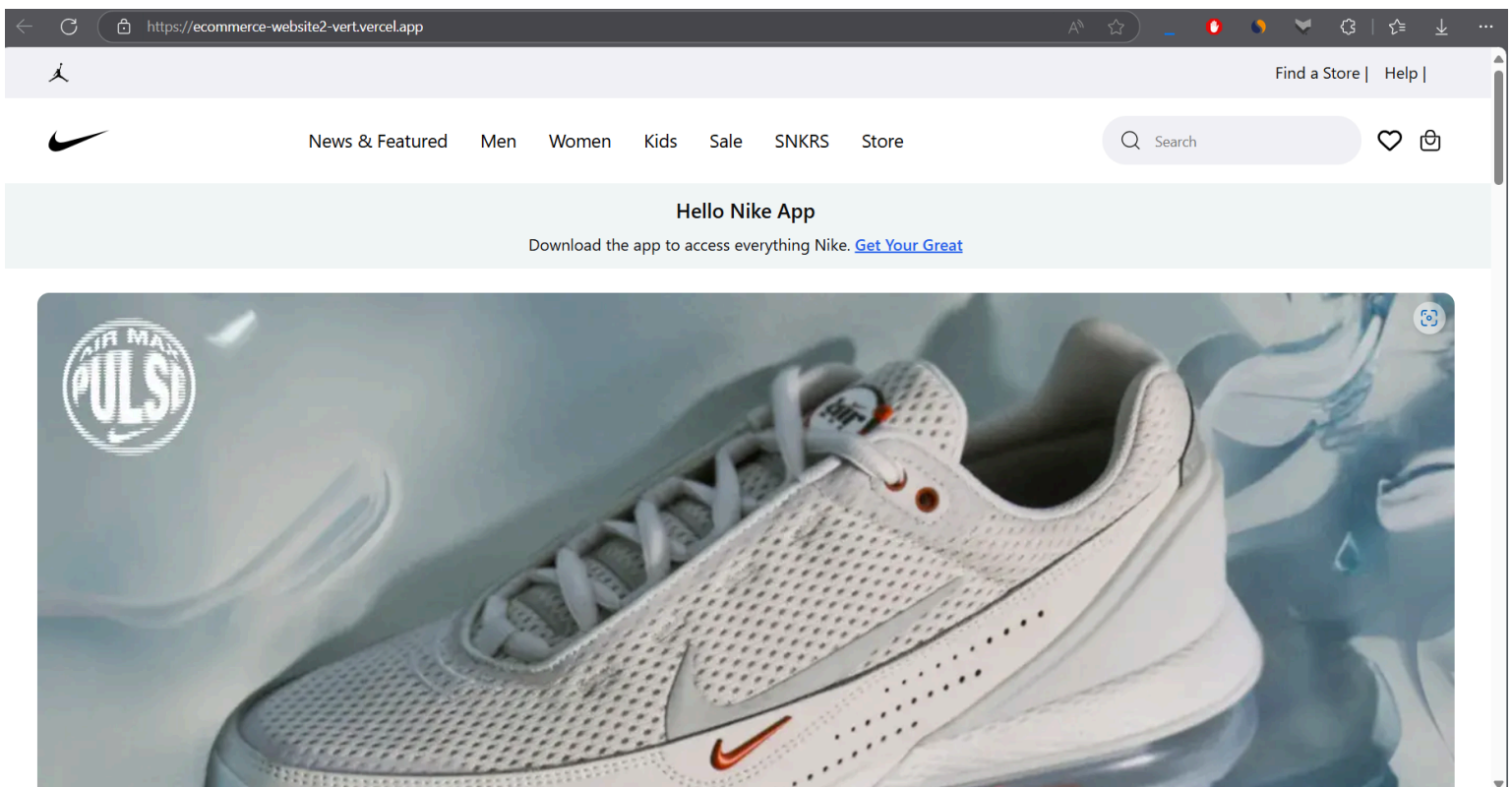
Example .env File Configuration:

```
NEXT_PUBLIC_SANITY_PROJECT_ID="6na
NEXT_PUBLIC_SANITY_DATASET="produc
NEXT_PUBLIC_SANITY_API_TOKEN=skvHA
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=
CLERK_SECRET_KEY=sk_test_ChFHWSGeH
```

3️⃣ Deploying to Staging

Steps:

- Pushed the latest code to **GitHub**.
- Deployed successfully on **Vercel staging environment**.
- Verified that the website **loads properly and all functionalities work correctly**.

Staging Environment Testing

📌 **Functional Testing**

Tested Features:

- Product Listing & Categories

- Authentication (Login/Register with Clerk)

- Shopping Cart & Checkout

- Admin Dashboard (Orders & Customer Data Management)

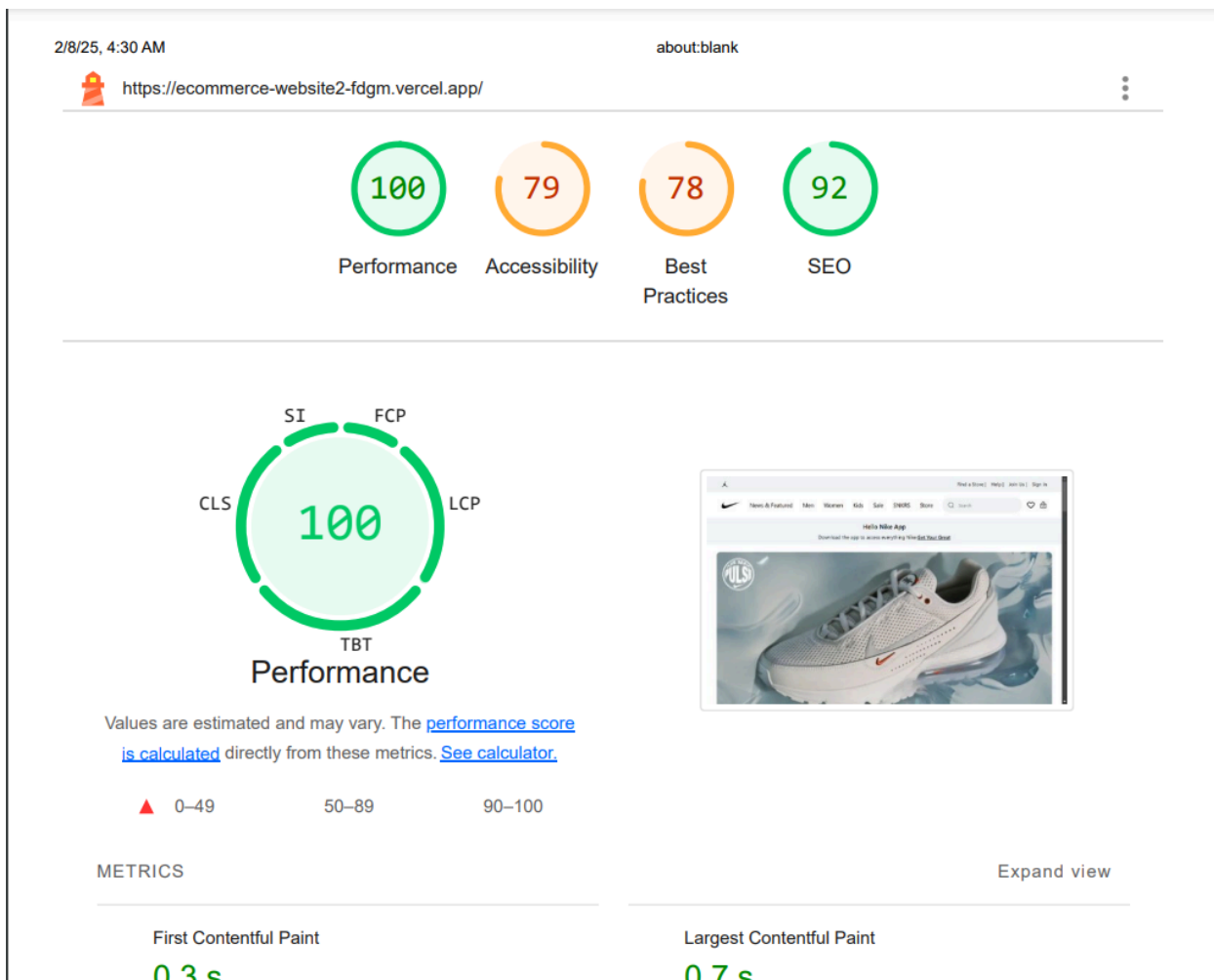- Every functionality tested and confirmed working correctly.

📌 Performance Testing

Optimization Techniques Used:

- **Lazy Loading & Image Compression**

- **Minified CSS & JavaScript Bundles**

- **Implemented Caching Strategies**

🔍 **Performance Testing Tools Used:** Google Chrome Lighthouse 📈 **Results: 100 Score on Desktop, 78 Score on Mobile** 🚀

📌 Security Testing

Steps Taken:

- **Validated Input Fields** using **React Hook Form & Zod validation**.
- **Ensured HTTPS usage** for secure data transmission.
- **Checked API security** and avoided exposing sensitive data in the frontend

📌 Challenges Faced & Solutions

Challenge	Issue Faced	Solution Implemented
Authentication	Clerk login session issues	Adjusted session handling and token validation
API Errors	Some API requests failed	Implemented request retries and caching
Build Errors	Deployment failed initially	Debugged logs and fixed configuration settings
Performance	Mobile score lower than expected	Optimized assets and improved rendering speed

📌Final Deliverables & Submission:

- ✅ **Fully deployed on Vercel staging environment**

- ✅ **Comprehensive error handling mechanisms**

- ✅ **Optimized performance with high Lighthouse scores**

- ✅ **Security validation completed with input validation**

- ✅ **Updated GitHub repository with structured documentation**

- ✅ **CSV-based functional testing report created**

Test Case ID	Component	Description	Expected Result	Actual Result	Status
TC-001	Categories	Ensure categories load dynamically	Categories appear	Categories appear	✅ Pass
TC-002	All Products	Verify fetched products from Sanity display correctly	Products visible	Products visible	✅ Pass
TC-003	Cart Page & Functionality	Test add/remove/update items in cart	Cart updates correctly	Cart updates correctly	✅ Pass
TC-004	Checkout Page & Functionality	Ensure checkout process completes successfully	Order placed	Order placed	✅ Pass
TC-005	Order Data Pushed to Sanity	Confirm orders are saved in Sanity CMS	Orders saved	Orders saved	✅ Pass
TC-006	Best of All	Check if best-selling products section displays correctly	Best products displayed	Best products displayed	✅ Pass
TC-007	Featured Page	Verify featured products load correctly	Featured products displayed	Featured products displayed	✅ Pass
TC-008	Header	Ensure header navigation works properly	Navigation works	Navigation works	✅ Pass
TC-009	Featured	Verify featured section displays correct data	Featured content displayed	Featured content displayed	✅ Pass
TC-010	Get Help Page	Check if help page loads correctly	Help content displayed	Help content displayed	✅ Pass
TC-011	Hero Component	Ensure homepage hero section loads properly	Hero content visible	Hero content visible	✅ Pass
TC-012	Pagination	Test pagination for products	Next/Prev pages load correctly	Next/Prev pages load correctly	✅ Pass
TC-013	Product Card	Ensure product card UI and interactions work	Product details and actions work	Product details and actions work	✅ Pass
TC-014	Sign-in Page	Verify React Hook Form and Zod validation	Validations work properly	Validations work properly	✅ Pass
TC-015	Sign-up Page	Check sign-up functionality with form validation	User successfully registers	User successfully registers	✅ Pass
TC-016	Authentication with Clerk	Ensure authentication and session handling works	User logs in/logs out properly	User logs in/logs out properly	✅ Pass
TC-017	Slug Component (Dynamic Products)	Test dynamic routing for product pages	Correct product page loads	Correct product page loads	✅ Pass

📌 Final Thoughts & Looking Forward

This journey of **deploying and optimizing** the marketplace has been an **enriching learning experience**. From configuring hosting to testing every functionality, each step has enhanced my skills. I am grateful to my instructors for their guidance and support throughout this process.

This is just the beginning—I look forward to applying these skills in **real-world projects** and pushing my knowledge further! 🚀

Name: Mohammad Danish
ID : 00037323