Nike Marketplace Project: Full Documentation (Days 1–6):

Overview:

The **Nike Marketplace** is an e-commerce platform aimed at empowering small businesses and individuals by providing a seamless and secure online shopping experience. Over the course of six days, the project evolved from brainstorming ideas to deploying a staging environment. Each day introduced specific tasks that contributed to the overall development.

Day 1: Conceptualization and Marketplace Design:

Key Achievements:

Defined the marketplace type as a general e-commerce platform for **Nike** products.

Business Goals:

Promote small businesses and entrepreneurship.

- Provide a platform to easily buy/sell **Nike** products online.

Data Schema Design:

Entities: Products, Orders, Customers, and Delivery Zones.

Relationships:

Customers place orders that reference products.

    - Delivery zones are assigned to drivers for fulfillment.
  -

Day 2: Technical Planning:

Tech Stack:

Frontend: Next.js with Tailwind CSS for styling.

Backend: Sanity CMS for content management.

- **APIs**: ShipEngine for order tracking and Stripe for payment processing.

API Requirements:

Product management: /products, /product/:id.

- **Orders**: /orders [POST] and /shipment/:id [GET].

Deployment Plan:

  - Frontend on Vercel and backend on AWS Lambda with serverless architecture.

Day 3: Data Migration

Key Achievements:

Custom Migration Code:

Data from Sanity CMS was migrated to Next.js using GROQ queries.

- Example GROQ Query: *[_type == "product"] {title, description, price, image}

Schema Definition:

- **Products** schema included fields for title, id description, price, and image.

Client Integration:

  - Fetched and displayed data dynamically on the homepage.

Day 4: Building Dynamic Frontend Components

Key Achievements:

Dynamic Product Listings:

- Created a ProductList component to display **Nike** products dynamically fetched from Sanity.

Filters and Sorting:

Implemented filters for categories and price ranges.

- Sorting options included price and popularity.

Reusable Components:

ProductPage: Displayed product images, titles, and prices.

Day 5: Testing and Backend Refinement

Key Achievements:

Testing Types:

Functional Testing:

- Verified workflows like product listings, cart operations, and API interactions.

Performance Testing:

- Used Lighthouse to analyze load times and responsiveness.

Security Testing:

  - Validated input fields, secure API keys, and HTTPS implementation.

Day 6: Deployment Preparation and Staging Environment Setup

Key Achievements:

Deployment Strategy:

Hosted the application on Vercel for quick deployment.

- **Integrated GitHub repository for CI/CD.**

Environment Variables:

- **Configured sensitive variables (e.g., API keys) in .env and uploaded them securely to Vercel.**

Staging Environment:

Deployed a staging build to validate functionality in a production-like environment.

Example .envFile:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id

NEXT_PUBLIC_SANITY_DATASET=production

API_KEY=your_api_key
```

Staging Testing:

Functional Testing: Verified key workflows like product listings and checkout.

Performance Testing: Used GTmetrix for analyzing speed and responsiveness.

- **Security Testing**: Validated HTTPS, input handling, and secure API calls.

Conclusion:

Over the six days, the Nike Marketplace project progressed from concept to deployment, integrating robust features and ensuring a seamless user experience. With a well-structured GitHub repository, dynamic components, and comprehensive testing, the project is now ready for live deployment in a production environment.

The next steps include:

1. **Addressing any unresolved issues documented in the staging tests.**
2. **Monitoring the live environment for user feedback and performance metrics.**
3. **Scaling the platform to include advanced features like multi-language support and predictive analytics.**

This marks the successful completion of the Nike Marketplace hackathon project!