

# Clothes Marketplace Project: Full Documentation (Days 1-6)

## Overview

The clothes 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 clothes.
- **Business Goals:**
  - Promote small businesses and entrepreneurship.
  - Provide a platform to easily buy/sell clothes 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

### Key Achievements:

- **Tech Stack:**
  - Frontend: Next.js with Tailwind CSS for styling.

- Backend: Sanity CMS for content management.
- Database: MongoDB for storing sensitive data and authentication.
- APIs: ShipEngine for order tracking and Stripe for payment processing.
- **API Requirements:**
  - User management: /register, /login, and /verify-route.
  - 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, slug, 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 clothes dynamically fetched from Sanity.
- **Filters and Sorting:**
  - Implemented filters for categories and price ranges.

- Sorting options included price and popularity.
- **Reusable Components:**
  - ProductCard: Displayed product images, titles, and prices.
  - FilterSidebar: Sidebar for filtering and sorting.
  - PaginationControls: Enabled page navigation for large datasets.

## 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.

- **CSV-Based Testing Report:**

### Test Case Table

Test Case ID	Description	Expected Result	Actual Result	Status	Severity	Remarks

TC001	Verify navigation links	Links navigate correctly	All links function correctly	Pas	Low	None
TC002	Check product listing display	Products display as expected	Products displayed correctly	Pas	Medium	None
TC003	Test shopping cart operations	Items add, update, and remove	Cart functionality works as expected	Pas	High	None
TC004	Validate contact form submission	Form submits successfully	Submission works with valid data	Pas	Medium	None
TC005	Analyze performance metrics	Achieve Performance $\geq 90$	Performance: 92	Pas	Medium	Optimizations for images implemented
TC006	Verify accessibility features	Accessibility score $\geq 90$	Accessibility: 96	Pas	Medium	Addressed contrast issues
TC007	Validate best practices	Best Practices score $\geq 90$	Best Practices: 96	Pas	Low	Minor improvements in image ratios noted
TC008	Optimize SEO	SEO score $\geq 90$	SEO: 100	Pas	Low	Structured data validated successfully

## CSV Content

Test Case ID,Description,Expected Result,Actual Result,Status,Severity,Remarks

TC001,Verify navigation links,Links navigate correctly,All links

function correctly,Pass,Low,None  
TC002,Check product listing display,Products display as expected,Products displayed correctly,Pass,Medium,None  
TC003,Test shopping cart operations,Items add, update, and remove,Cart functionality works as expected,Pass,High,None  
TC004,Validate contact form submission,Form submits successfully,Submission works with valid data,Pass,Medium,None  
TC005,Analyze performance metrics,Achieve Performance  $\geq$  90,Performance: 92,Pass,Medium,Optimizations for images implemented  
TC006,Verify accessibility features,Accessibility score  $\geq$  90,Accessibility: 96,Pass,Medium,Addressed contrast issues  
TC007,Validate best practices,Best Practices score  $\geq$  90,Best Practices: 96,Pass,Low,Minor improvements in image ratios noted  
TC008,Optimize SEO,SEO score  $\geq$  90,SEO: 100,Pass,Low,Structured data validated successfully

## 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 .env File:  
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.
- **Documentation:**
  - Created a README.md summarizing the project structure and deployment steps.
  - Organized the GitHub repository with folders for `src/`, `public/`, and `documents/`.

## Conclusion

Over the six days, the clothes 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 clothes Marketplace hackathon project! 🎉