Name: *Noor Ul Sehar*

Roll No: 00349943

Class: Friday (9 AM - 12 PM)

Date: 02-feb-2025

Nike Clone General E-commerce Platform

Day 1: Problem Definition & Audience Identification

Objective: Define the purpose of the platform and identify the target audience. **Key Tasks:**

1. **Problem Statement:**
   - ○ Identified the need for a seamless e-commerce platform for authentic products.
2. **Audience Research:** ○ Targeted tech-savvy users who value convenience and security.
3. **Scope Definition:**
   - ○ Outlined core features: product browsing, cart management, secure checkout, and order tracking.

Outcome:

✅ Clear problem statement | ✅ Defined target audience | ✅ Scope finalized

---

Day 2: Tech Stack & System Design

Objective: Plan the technical architecture and tools for the platform. **Key Tasks:**

1. **Tech Stack Selection:**
   - ○ Frontend: Next.js (dynamic, SEO-friendly). ○ Backend: Sanity CMS (content management).

    - APIs/Tools: Stripe (payments), ShipEngine (shipping).

2. **System Design:**
    - Created schemas for products, users, and orders in Sanity CMS.
    - Mapped API endpoints for product listings, orders, payments, and tracking.

Outcome:

✅ Architecture blueprint | ✅ Sanity CMS setup | ✅ Third-party API integration plan

---

Day 3: Data Integration & Migration

Objective: Fetch, store, and display dynamic product data. **Key Tasks:**

1. **Data Fetching:** ○ Pulled product details (name, price, brand, images) from external API.
2. **Sanity CMS Setup:**
    - Migrated data to Sanity and updated schema:
        - Added inventory, colors, and status fields.
        - Used GROQ queries to fetch structured data.
3. **Frontend Integration:**
    - Built dynamic product grids with filters (category, price). ○ Designed a clean UI for seamless browsing.

Outcome:

✅ Data migrated to Sanity | ✅ Schema optimized | ✅ Dynamic product display

---

Day 4: Frontend Component Development

Objective: Build modular, reusable components for a responsive UI. **Key Tasks:**

1. **Core Components Developed:** ○ Product Listings: Grid layout with filters. ○ Product Details Page: Dynamic routing with Next.js. ○ Cart & Wishlist: State management via React Context API and localStorage. ○ Checkout Flow: Multi-step form with validation.
    - Search Bar, Filters, Pagination: Enhanced user navigation.
2. **Integrations:**
    - Clerk for authentication.
    - ShipEngine for order tracking.
    - Stripe for secure payments.

Outcome:

✅ 18+ components built | ✅ Responsive design | ✅ Real-world workflows replicated

---

Day 5: Testing, Error Handling & Backend Refinement

Objective: Validate functionality, optimize performance, and prepare for deployment.
Key Tasks:

1. **Functional Testing:**
    - Verified product listings, filters, cart operations (add/update/remove), dynamic routing, and API responses (Postman).
    - Tested components with React Testing Library.
2. **Error Handling:**
    - Implemented try-catch blocks for API calls.
    - Added user-friendly error messages and fallback UIs (e.g., "No items found").
3. **Performance Optimization:**
    - Used Lighthouse/GTmetrix to optimize load times (image compression, code splitting).
    - Achieved 90+ performance scores.
4. **Security Testing:**
    - Validated input sanitization and HTTPS compliance.
    - Scanned vulnerabilities with OWASP ZAP.

5. **Cross-Browser/Device Testing:**
    - Confirmed responsiveness on Chrome, Firefox, Safari, Edge, and mobile devices.
6. **Documentation:**
    - Logged 15+ test cases and resolved errors (e.g., image/price display issues).

Outcome:

✅ All core features validated | ✅ Error-free UX | ✅ Deployment-ready backend

---

Day 6: Deployment Preparation & Staging Setup

Objective: Simulate production-like environment and finalize deployment. **Key Tasks:**

1. **Hosting Setup:**
    - Chose **Vercel** for Next.js optimization, global CDN, and automatic scaling.
    - Connected GitHub repo and configured build commands.
2. **Environment Variables:** ○ Securely stored API keys (Sanity, Stripe, ShipEngine) in Vercel's dashboard.
3. **Staging Deployment:**
    - Deployed to yourproject-name.vercel.app and validated functionality (product listings, cart, auth).
4. **Staging Testing:**
    - Tested search, filters, checkout flow, and user sessions.
    - Ensured responsive design on PC/mobile.
5. **Performance Checks:**
    - 72 Lighthouse score.

Outcome:

✅Staging environment live | ✅Security/performance validated | ✅Documentation updated

---

Day 7: Final Deployment & Launch

Objective: Go live and ensure smooth post-launch operations. **Key Tasks:**

1. **Final Checks:**
   - Re-validated environment variables and API endpoints.
   - Tested checkout flow with Stripe test mode.
2. **Production Deployment:**
   - Merged code to `main` branch; triggered Vercel deployment.
   - Configured custom domain (if applicable) and enforced HTTPS.
3. **Post-Launch Monitoring:**
   - Set up Vercel Analytics for traffic/performance tracking.
   - Enabled error logging via Sentry.
4. **Submission:**
   - Prepared GitHub repo link: https://github.com/nh7220869/hakathon-3 ○ Shared live URL: https://hakathon-3-1ihb-git-main-noor-ul-sehars-projects.vercel.app/ ○ Submitted hackathon documentation (Days 1–7).

Final Outcome:

Marketplace Successfully Launched!

- Users can browse authentic products, add to cart, and checkout securely.
- Real-time order tracking via ShipEngine.
- Responsive on all devices and compliant with security standards.

Progress Summary (Days 1–7):

- **Day 1:** Defined the why (problem) and who (audience).
- **Day 2:** Finalized the how (tech stack and architecture).

- **Day 3:** Brought data to life (Sanity CMS + dynamic UI).
- **Day 4:** Built a user-centric frontend (components + integrations).
- **Day 5:** Validated functionality and optimized performance.
- **Day 6:** Prepared for deployment with staging setup.
- **Day 7:** Successfully launched the platform.

Self-Assessment:

- All test cases passed.
- Deployment checklist completed.
- Ready for user onboarding and scaling!