

**Roll No:**00477608

**Date:** 4-2-25

**Name:** Syeda Hafiza Bibi Amna

**Class:** Friday (9 AM - 12 PM)

# Sit & Style Studio

---

## Day 1: Problem Definition & Audience Identification

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

**Key Tasks:**

**Problem Statement:**

- Recognized the demand for a smooth e-commerce platform offering genuine products.

**Audience Research:**

- Focused on tech-savvy individuals who prioritize convenience and security.

**Scope Definition:**

- Defined essential features: product exploration, shopping cart management, secure payment processing, and order tracking.

**Outcome:**

- ☒ Clear problem definition
- ☒ Target audience identified
- ☒ Scope established

## 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-optimized)

- Backend: Sanity CMS (content management)
- APIs/Tools: Stripe (payment processing), ShipEngine (shipping)
- 2. System Design:**
  - Designed schemas for products, users, and orders within Sanity CMS
  - Mapped API endpoints for product listings, orders, payments, and tracking

**Outcome:**

-  Architecture blueprint established
-  Sanity CMS setup completed
-  Third-party API integration plan developed.




## Day 3: Data Integration & Migration

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

**Key Tasks:**

- 1. Data Fetching:**
  - Retrieved 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 retrieve structured data.
- 3. Frontend Integration:**
  - Developed dynamic product grids with filters (category, price).
  - Designed a sleek UI for effortless browsing.

**Outcome:**

-  Data successfully migrated to Sanity
-  Schema optimized for efficiency
-  Dynamic product display implemented

## 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 using 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
- Stripe for secure payments

### Outcome:

-  18+ components successfully built
-  Responsive design implemented
-  Real-world workflows effectively 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 (using 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 to optimize load times (image compression, code splitting)
- Achieved great 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 optimal Next.js performance, global CDN, and automatic scaling.
  - Connected GitHub repository 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, authentication).
4. **Staging Testing:**
  - Tested search, filters, checkout flow, and user sessions.
  - Ensured responsive design on both PC and mobile.
5. **Performance Checks:**
  - Achieved a 72 Lighthouse score.

### Outcome:

-  Staging environment live
-  Security and performance validated
-  Documentation updated

## Day 7: Final Deployment & Launch

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

### Key Tasks:

#### Final Checks:

- Re-validated environment variables and API endpoints.
- Tested checkout flow using Stripe test mode.

#### Production Deployment:

- Merged code into the main branch and triggered Vercel deployment.
- Configured a custom domain (if applicable) and enforced HTTPS.

#### Post-Launch Monitoring:

- Set up Vercel Analytics for tracking traffic and performance.
- Enabled error logging via Sentry.

### Submission:

- Prepared the GitHub repository link for submission: [https://github.com/SyedaHafizaBibiAmna/Marketplace\\_Builder\\_Hackathon.git](https://github.com/SyedaHafizaBibiAmna/Marketplace_Builder_Hackathon.git)
- Shared live URL: <https://sit-and-style-studio.vercel.app/>
- Submitted hackathon documentation (Days 1–7).




## Final Outcome:

- **Marketplace Successfully Launched!**
  - Users can browse authentic products, add to the cart, and securely checkout.
  - Real-time order tracking through ShipEngine.
  - Fully responsive across 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