Marketplace Hackathon

Marketplace Hackathon 2024 Day-2

Prepared by:

Arjumand Afreen Tabinda

Logo
Name

MARKETPLACE HACKATHON TECHNICAL ARCHITECTURE

1. Website Structure and Pages

The website comprises the following essential pages:

a. Home

- *Serves as the landing page, offering an overview of the website's objectives and showcasing key products and features.*
- *Dynamic content is managed through Sanity CMS.*

b. About

- *Provides insights into BANDAGE's mission, vision, and background.*
- *Content is easily customizable and updated via Sanity.*

c. Contact

- *Features a contact form for users to connect with the team.*
- *Form submissions can be processed via backend services.*

d. Products

- *Showcases a catalog of products along with their prices and brief descriptions.*
- *Data is dynamically retrieved from Sanity.*

e. Product Details

- *Displays comprehensive information for each product, including specifications, images, reviews, and pricing.*
- *Content is dynamically fetched from Sanity based on user selection.*

f. Cart

- *Allows users to view and manage selected products for purchase.*
- *Supports dynamic data management, enabling users to add, remove, or adjust product quantities.*

g. Login and Signup

- *Authentication and user management are facilitated through Clerk.*
- *Secure login and signup options include social login and email/password authentication.*

h. Checkout and Payment

- *Provides seamless checkout functionality integrated with Stripe for secure transactions.*
- *Supports multiple payment options.*

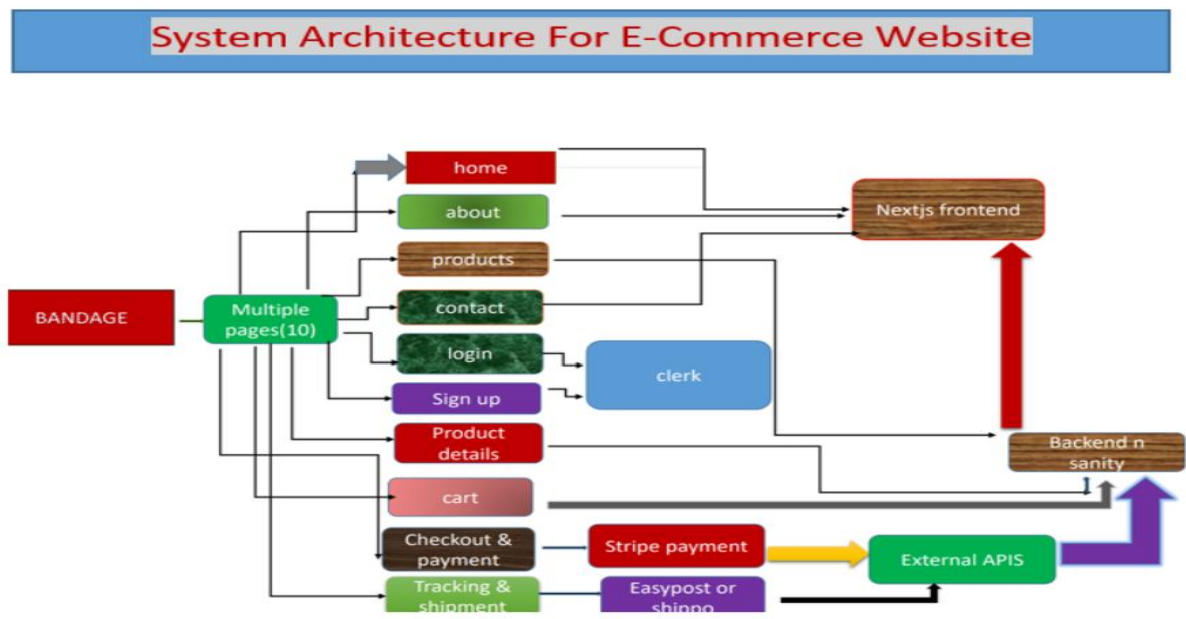*i. Shipping and Order Tracking*

- *Enables users to monitor their orders and access shipping details.*

- *Integrated with Shippo and EasyPost APIs for real-time tracking and updates.*

j. Additional Utility Pages

- *Includes error pages (e.g., 404) and loading screens to improve user experience.*

System Architecture For E-Commerce Website

2. Backend Integration with Sanity

- *Sanity functions as the primary CMS for managing and displaying dynamic content across the platform.*
- *Products, product details, banners, and homepage text are dynamically retrieved.*

- *Enables administrators to update content without altering the website's code.*

3. User Authentication via Clerk

- *Clerk manages authentication for login and signup.*
- *Features:*
  - *Secure handling of user data.*
  - *Password reset, multi-factor authentication, and social login options.*
- *Seamlessly integrates with Next.js for real-time authentication.*

4. Payment Processing with Stripe

- *The Stripe API ensures secure transaction handling.*
- *Features include:*
  - *Support for various payment methods, including credit cards and digital wallets.*
  - *Manages payment success and failure for a smooth checkout experience.*

5. Shipping and Order Tracking via Shippo and EasyPost

- *Shippo and EasyPost APIs streamline shipment management and tracking.*
- *Shippo:*
  - *Generates shipping labels.*
  - *Manages carrier accounts and shipping rates.*

- *EasyPost:*
  - *Provides live tracking updates.*
  - *Allows users to check the status and location of their shipments.*

6. Tech Stack

a. Next.js

- *Powers a scalable, SEO-optimized, and high-performance website.*
- *Supports server-side rendering (SSR) and static site generation (SSG) for enhanced speed and efficiency.*

b. Sanity

- *Functions as a headless CMS for handling dynamic content, including product information and other website components.*
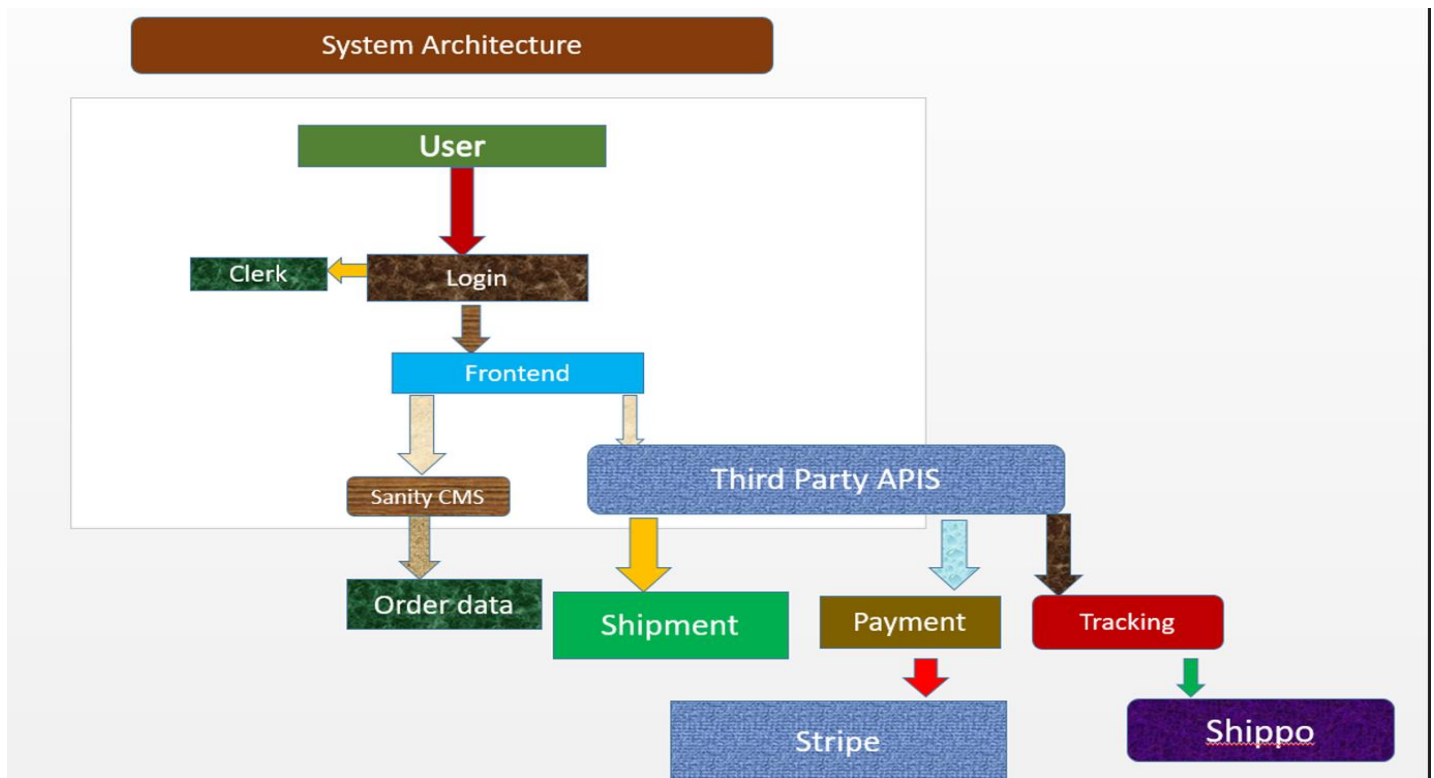
c. Clerk

- *Facilitates user authentication and account management, ensuring a secure and seamless login/signup process.*

d. Stripe

- *Manages secure transactions during the checkout process.*

e. Shippo and EasyPost

- *These APIs streamline shipment processing and enable users to track their orders in real time.*

System Architecture

7. User Experience Highlights

- *Dynamic Content: Pages such as Products and Product Details retrieve and showcase real-time data from the Sanity CMS.*
- *Secure and Seamless Authentication: Clerk provides a smooth and hassle-free login and registration experience.*
- *Streamlined Payments: Stripe ensures secure and effortless transactions for users.*

- *Live Order Tracking: Shippo and EasyPost improve the post-purchase journey by offering real-time shipment updates.*

8. Scalability and Maintenance

- *The integration of Sanity, Clerk, Stripe, Shippo, and EasyPost allows the BANDAGE website to scale efficiently as the business expands.*
- *Content updates happen dynamically, while external APIs minimize maintenance efforts for core functionalities like authentication, payments, and shipping.*

9. Steps Involved in My Website

Step 1: User Visits the Website

Frontend (Next.js):

- *The user lands on the homepage.*
- *Featured products and banners are dynamically retrieved from Sanity CMS via API calls and displayed accordingly.*
- *If the user is already logged in, Clerk Authentication securely manages their session and greets them with personalized details.*

Step 2: User Logs In/Signs Up

Authentication via Clerk:

- *Users can register or log in using:*
  - *Email and password.*
  - *Third-party services like Google, GitHub, or other providers supported by Clerk.*
- *Clerk Authentication:*
  - *Handles user sessions and securely stores authentication details.*

- *Restricts access to certain actions (e.g., cart or checkout) to authenticated users only.*
- *Users who are logged in can:*
  - *Access their profile page.*
  - *View their past orders.*
  - *Modify account preferences.*

Step 3: Exploring Products

Products Page:

- *The user navigates to the products section, where a catalog of items is displayed.*
- *Sanity CMS:*
  - *Stores essential product details such as name, price, images, and descriptions.*

- An API retrieves this information and presents it dynamically on the page.
- *For logged-in users:*
  - They can bookmark items or add them to a wishlist (potential future feature).

Step 4: Viewing Product Details

Product Details Page:

- *When a user selects a product, they are directed to a detailed view.*
- *Sanity CMS:*
  - Retrieves specifications, pricing, and stock availability through an API.
  - This data is dynamically presented on the Product Details page.

Step 5: Adding a Product to the Cart

Frontend (Next.js):

- *When the user selects Add to Cart:*
  - The product details (ID, quantity, price) are temporarily stored in the session or managed via a state management tool like Redux or Context API.
  - If the user is not logged in:
    - They receive a prompt to sign in or register before proceeding to checkout.

- *Once authenticated, the cart is saved and linked to the user's account for seamless access.*

Step 6: Proceeding to Checkout

Checkout Process:

- *The user selects Checkout to review their order before making a payment.*
- *Stripe Payment Integration:*
  - *Calculates the total cost of items in the cart.*
  - *Redirects the user to Stripe's secure payment portal, where they enter payment details (e.g., credit card).*

Step 7: Payment Confirmation

Stripe API Integration:

- *After successful payment:*
  - *Stripe sends a confirmation response back to the website through its API.*
  - *Sanity CMS:*
    - *Stores order details (e.g., user data, purchased products, payment status) for tracking and record-keeping.*

Step 8: Shipping and Tracking Orders

Third-Party APIs (Shippo/EasyPost):

- *The order is forwarded to shipping services like Shippo or EasyPost.*
- *These APIs:*
  - *Generate a unique tracking number.*
  - *Continuously update the shipping status.*
- *The user can check order status and tracking updates on the Order Tracking Page.*

Step 9: Viewing Profile & Order History

Profile Section:

- *Logged-in users can visit their Profile Page to:*
  - *Check their past orders.*
  - *Monitor the progress of their current orders.*
  - *Edit personal details such as contact information and addresses.*
- *Sanity CMS:*
  - *Fetches order history from the Sanity database dynamically and displays it on the Profile Page.*

10. API Endpoints

Endpoint	Method	Function	Example Response
/products	GET	Retrieves all available products.	{ "id": 1, "name": "Product X", "price": 150 }
/orders	POST	Creates a new order and stores it in the database.	{ "orderId": 987, "status": "Order Placed" }
/order-status	GET	Retrieves the status of a specific order.	{ "orderId": 987, "status": "Processing", "ETA": "3 days" }
/shipment-status	GET	Checks shipment status via third-party shipping APIs.	{ "shipmentId": 456, "status": "Dispatched", "location": "Hub A" }
/express-shipment	GET	Provides live updates for express deliveries.	{ "orderId": 321, "status": "Out for Delivery", "ETA": "30 mins" }

Endpoint	Method	Function	Example Response
/user/cart	GET	Retrieves the items currently stored in the user's cart.	{ "cartItems": [{ "id": 1, "name": "Product X", "quantity": 2 }], "totalPrice": 300 }
/user/cart/add	POST	Adds a new product to the shopping cart.	{ "message": "Product added successfully", "cartItems": [{ "id": 1, "name": "Product X", "quantity": ...

API Endpoints

Endpoint	Method	Function	Example Response
/user/cart/remove	DELETE	Deletes a product from the shopping cart.	{ "message": "Product removed successfully", "cartItems": [{ "id": 2, "name":

Endpoint	Method	Function	Example Response
			"Product Y", "quantity": 1 }] }
/reviews	POST	Allows users to submit feedback for a product.	{ "productId": 1, "reviewId": 765, "message": "Review submitted successfully" }
/track/order	GET	Fetches comprehensive tracking information for an order.	{ "orderId": 987, "status": "Shipped", "lastLocation": "City Hub", "ETA": "2 days", "trackingDetails": [{ "date": "2025-01-16", "status": "Dispatch" }] }

11. Quick Summary

🔑 Key Pages:

- **Home 🏠 – Showcases products, features, and dynamic content via Sanity CMS.**
- **About 📖 – Shares BANDAGE's mission, vision, and background.**
- **Contact 📞 – Includes a form for user inquiries.**
- **Products 🛍️ – Displays available products with prices and brief descriptions.**
- **Product Details 🛒 – Provides detailed specifications, images, reviews, and pricing.**
- **Cart 🛒 – Allows users to manage selected items before checkout.**
- **Login/Signup 🔑 – User authentication handled by Clerk, including social login options.**
- **Checkout & Payment 💳 – Ensures secure payments via Stripe.**
- **Shipment & Tracking 🚚 – Offers real-time tracking with Shippo and EasyPost.**

🔧 Backend Integration:

- **Sanity CMS 🌐 – Manages dynamic content for products and pages.**

- *Clerk Authentication 🔐 – Handles secure login and user sessions.*
- *Stripe Payment 💳 – Supports multiple payment methods for seamless checkout.*

Shipping & Tracking

🎁 Shippo & EasyPost – Real-time order tracking and label generation.

Technology Stack

- *Next.js 🚀 – High-performance, SEO-optimized frontend framework.*
- *Sanity 🎁 – Headless CMS for flexible content management.*
- *Clerk 🔑 – Secure user authentication and account management.*
- *Stripe 💳 – Reliable payment gateway for transactions.*
- *Shippo/EasyPost 🚚 – Real-time shipment tracking and logistics.*

💡 Enhanced User Experience

- *Dynamic Content 📈 – Product pages and details seamlessly retrieved from Sanity CMS.*

- *Smooth Authentication 🔐 – Hassle-free login and signup via Clerk.*

- *Secure Payments 💳 – Fast and safe transactions through Stripe.*

- *Live Order Tracking 🚚 – Stay updated on shipments using Shippo/EasyPost.*

📈*Scalability & Maintenance*

- *Designed to scale effortlessly as the business expands.*

- *Easy content updates via Sanity CMS with minimal upkeep required for third-party integrations.*

A big thank you to Sir Ameen Alam for his invaluable guidance and support in successfully implementing these features!