

Governor Sindh Initiative AI Course

Name:Mehwish Sheikh

Roll number:223558

Hackathon Day:2

PLANNING THE TECHNICAL FOUNDATION

General E-Commerce Website:Comforty

Project Overview: Comforty E-Commerce Website:

Project Title: Comforty - A Modern E-Commerce Platform

Objective: Build a sleek, responsive, and feature-rich e-commerce website using modern technologies such as Next.js,Tailwind CSS, Sanity, Figma, and third-party APIs for seamless product management, user authentication, and shipping.

Project Goals:

User Experience: Create an intuitive and visually appealing user interface for browsing and purchasing products.

Responsive Design: Ensure compatibility across devices using mobile-first design principles.

Content Management: Utilize Sanity as a CMS for easy product and content updates.

API Integration: Integrate third-party APIs for dynamic data (products, orders, and shipping).

Performance: Leverage Next.js features like static site generation (SSG) and server-side rendering (SSR) for fast load times.

System Architecture for E-Commerce Platform:

Frontend (Client-Side):

Built with Next.js and styled using Tailwind CSS.

Handles:

User interface (UI) and experience (UX).

Dynamic routing for pages (e.g., product, cart, checkout).

Data fetching and rendering (SSG/SSR/CSR).

Integration with APIs for data display and user actions.

Backend:

Sanity CMS:

Governor Sindh Initiative AI Course

Name: Mehwish Sheikh

Roll number: 223558

Role: Manages product data, categories, and promotional content.

Integration:

Fetch data using Sanity's GROQ queries or client SDK.

Example: Fetch product details for rendering product pages.

Custom API (Next.js API Routes):

Endpoints:

/api/cart: Handle cart operations (add, update, remove items).

/api/orders: Create and manage orders.

/api/users: Manage user data (e.g., profile, order history).

Third-Party API Orchestration:

Integrates payment gateway and shipping API.

Example:

Payment API: Validate payments before order creation.

Shipping API: Calculate shipping costs and generate labels.

Third-Party Integrations:

Payment Gateway (e.g., Easypaisa, Jazzcash, HBL etc):

Secure payment processing.

Tokenization of sensitive payment data.

Shipping API (e.g., ShipEngine):

Real-time shipping rates.

Label generation.

Shipment tracking.

Detailed System Architecture:

Frontend (Next.js + Tailwind CSS)

Components:

Reusable UI components (e.g., Navbar, ProductCard, Footer).

Pages for Home, Product Listing, About, Contact, FAQs, Login/Signup, Account, Admin, Product Details, Cart, and Checkout.

State Management:

Client-side state for cart and user session (e.g., using Context API or Zustand).

Data Fetching:

SSG: Pre-render static pages (e.g., product catalog) for performance.

SSR: Fetch dynamic data (e.g., personalized recommendations).

Governor Sindh Initiative AI Course

Name:Mehwish Sheikh

Roll number:223558

CSR: Use client-side fetching for actions like adding to the cart.

API Integration:

Fetch data from Sanity and custom APIs using `getStaticProps`, `getServerSideProps`, or `useSWR`.

Backend

Sanity CMS:

Role: Manages product data, categories, and promotional content.

Integration:

Fetch data using Sanity's GROQ queries or client SDK.

Example: Fetch product details for rendering product pages.

Custom API (Next.js API Routes):

Endpoints:

`/api/cart:Method:GET` -> Handle cart operations (add, update, remove items).

`/api/orders:Method:Post` -> Create and manage orders.

`/api/users:Method:GET` -> Manage user data (e.g., profile, order history).

Third-Party API Orchestration:

Integrates payment gateway and shipping API.

Example:

Payment API: Validate payments before order creation.

Shipping API: Calculate shipping costs and generate labels.

Third-Party Integrations

Payment Gateway:

Example: Easyapaise or Jazzcash.

Flow:

Frontend collects payment details via a secure payment form.

Sends payment data to the custom API.

API interacts with the payment gateway for processing.

Returns success or failure response to the frontend.

Shipping API:

Example: ShipEngine.

Flow:

Frontend sends shipping details (e.g., address, cart weight) to the custom API.

Custom API queries ShipEngine for rates and labels.

Returns shipping options and tracking info to the frontend.

Data Flow Diagram:

Governor Sindh Initiative AI Course

Name: Mehwish Sheikh

Roll number: 223558

[Frontend (Next.js + Tailwind CSS)]

```
|  
|--> Fetch Data (Sanity CMS via GROQ or SDK)  
|  
|--> API Calls (Custom API)  
|  
|   |--> Payment Gateway (e.g., Easypaisa)  
|   |  
|   |--> Shipping API (e.g., ShipEngine)  
|  
|<-- Return Data/Responses
```

Deployment and Hosting:

Frontend: Deployed on Vercel (optimized for Next.js).

Sanity CMS: Hosted on Sanity's platform.

Custom API: Deployed alongside the frontend on Vercel or separately on a Node.js server.

Third-Party APIs: Managed by their respective providers (e.g., Easypaisa, ShipEngine).

Security Considerations:

API Authentication:

Use API keys for Sanity, payment gateway, and shipping API.

Store sensitive credentials securely (e.g., in environment variables).

Frontend Security:

Use HTTPS and Content Security Policies (CSP).

Avoid exposing sensitive data (e.g., API keys) in the client-side code.

Payment:

Use PCI-compliant gateways like Stripe.

Never store sensitive payment information on your servers.

Tools and Libraries:

Frontend:

Next.js, Tailwind CSS, SWR (data fetching).

Backend:

Next.js API Routes, Sanity SDK, Axios (API requests).

Third-Party:

Easypaisa (Payment), ShipEngine (Shipping).

System Workflows: User and Admin:

Governor Sindh Initiative AI Course

Name: Mehwish Sheikh

Roll number: 223558

1. User Workflow

Step 1: User Login/Signup

Frontend:

User accesses the login/signup page.

Inputs email/password or uses social login (e.g., Google).

Backend:

Sends credentials to the authentication service (e.g., Firebase Auth).

Verifies credentials and returns a JWT or session token.

Post-login:

Token is stored in the client (e.g., HTTP-only cookie or localStorage).

User is redirected to the homepage or dashboard.

Step 2: Browse Products

Frontend:

User navigates to the homepage or product listing page.

Filters and sorts products (e.g., by category, price, rating).

Backend:

Fetches product data from Sanity CMS.

Returns filtered product list to the frontend.

Frontend:

Renders the product grid dynamically.

Step 3: View Product Details

Frontend:

User clicks on a product to view its details.

Product details page is dynamically loaded using the product ID.

Backend:

Fetches product details from Sanity CMS.

Returns product description, price, reviews, and availability.

Frontend:

Displays product information and "Add to Cart" button.

Step 4: Add to Cart

Frontend:

User selects quantity and clicks "Add to Cart."

Cart state is updated locally or via an API call.

Backend:

If using server-side cart:

API endpoint (/api/cart) updates the cart in the database.

Returns the updated cart data to the frontend.

Frontend:

Updates the cart UI.

Step 5: Checkout

Frontend:

Governor Sindh Initiative AI Course

Name: Mehwish Sheikh

Roll number: 223558

User proceeds to the checkout page.

Inputs shipping details and selects a shipping method.

Backend:

Calls the shipping API (e.g., ShipEngine) to calculate shipping costs.

Returns available shipping options and costs.

Frontend:

Displays the total cost, including shipping.

Step 6: Payment

Frontend:

User inputs payment details or selects a saved payment method.

Sends payment data securely to the custom API.

Backend:

Custom API interacts with the payment gateway (e.g., Easypaisa) to process payment.

Returns payment confirmation or error.

Frontend:

Displays a success message and order confirmation.

Step 7: Order Tracking

Frontend:

User navigates to the "Order History" page.

Selects an order to view its status.

Backend:

Fetches order and tracking details from the database and shipping API.

Returns the tracking status.

Frontend:

Displays order details and shipment tracking link.

2. Admin Workflow

Step 1: Admin Login

Frontend:

Admin accesses the admin login page.

Inputs credentials.

Backend:

Authenticates credentials using Firebase Auth or a custom admin system.

Returns a session token with admin privileges.

Post-login:

Admin is redirected to the admin dashboard.

Step 2: Manage Products

Frontend:

Admin navigates to the "Products" section.

Views a list of all products.

Adds, edits, or deletes a product.

Backend:

Governor Sindh Initiative AI Course

Name:Mehwish Sheikh

Roll number:223558

Fetches product data from Sanity CMS for display.

Handles create, update, or delete operations via Sanity's API.

Frontend:

Reflects changes in the product list.

Step 3: View and Manage Orders

Frontend:

Admin navigates to the "Orders" section.

Views a list of all orders, with statuses (e.g., pending, shipped).

Selects an order to update its status.

Backend:

Fetches order data from the database.

Updates order status (e.g., marked as shipped) and notifies the user.

Frontend:

Updates the order status in the admin panel.

Step 4: Manage Users

Frontend:

Admin navigates to the "Users" section.

Views a list of all registered users.

Can block/unblock users or reset their passwords.

Backend:

Fetches user data from the authentication service or database.

Handles user updates (e.g., status changes).

Frontend:

Reflects changes in the user list.

Step 5: Monitor Analytics

Frontend:

Admin accesses the "Analytics" dashboard.

Views key metrics (e.g., sales, orders, user activity).

Backend:

Fetches aggregated data from the database or analytics service.

Returns metrics to the frontend.

Frontend:

Displays charts and insights dynamically.

WorkFLOW Summary:



Governor Sindh Initiative AI Course

Name:Mehwish Sheikh

Roll number:223558

BROWSE PRODUCT



VIEW PRODUCT DETAILS



ADD TO CART



CHECKOUT



PAYMENT



ORDER TRACKING

MANAGE PRODUCT



MANAGE ORDERS



MANAGE USERS



MONITOR ANALYTICS

