PREPARED BY: MUHAMMAD TAHA SIDDIQUI

Hackathon Day 2

The Technical Foundation Planning

OUTLINE

My Market Place Clothing Website

- Efficient Design
- User-Friendly Interface
- Business Goals Alignment
- Advanced Features
- Reliable Performance

- Overview
- System Architecture
- Frontend Development Plan
- Backend Development Plan
- Integration and Workflow
- Conclusion

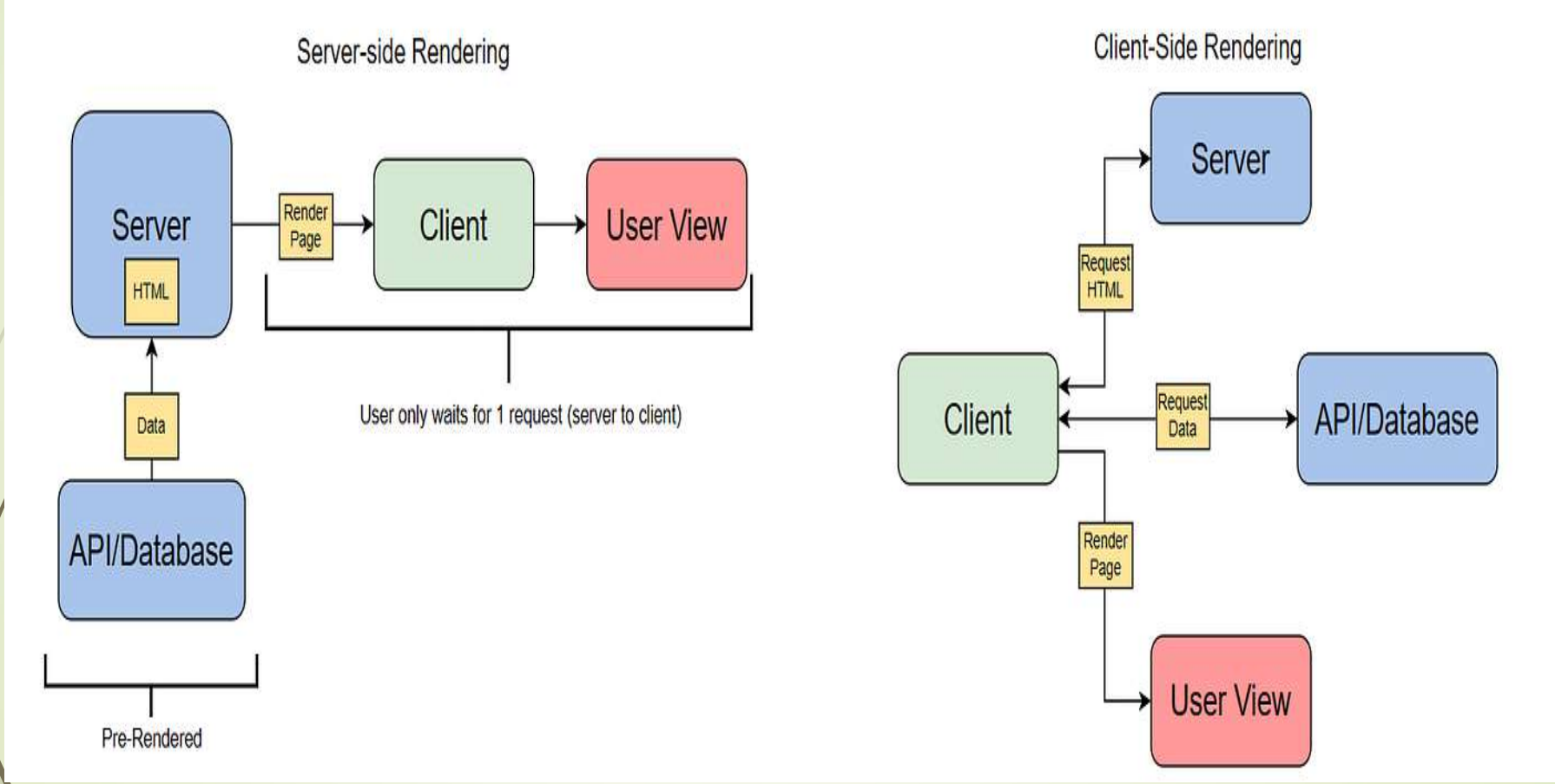
Technical Plan: Scalable Marketplace Clothing Website

“

Overview:

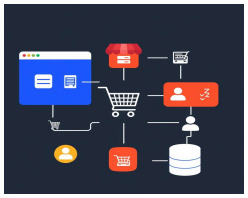
Design a scalable and user-friendly online marketplace for clothing with a focus on seamless navigation, a modern and responsive interface, and efficient performance to align with business objectives.

System Architecture Diagram



Scalable Architecture for Marketplace Clothing Platform

The architecture is thoughtfully designed to deliver scalability, performance, and security, ensuring a smooth and efficient experience for users, administrators, and third-party integrations. Below is the detailed overview:



Key Components

1.Frontend:

- **Next.js:** Powers server-side rendering (SSR) and static site generation (SSG) to provide faster load times, improved SEO, and dynamic content rendering.
- **Tailwind CSS:** A utility-first CSS framework used to craft responsive, mobile-friendly, and aesthetically appealing designs.
- **Features**:
  - User-friendly navigation with advanced filters and a powerful search functionality.
  - Personalized dashboards for account management and order tracking.
  - Enhanced cart and checkout systems with real-time validation and minimal friction.

- **Backend:**
- **Content Management (CMS)**:
  - A centralized system to manage product details, categories, promotional banners, and customer reviews.
  - Simplifies updates to product data and other content using an admin-friendly dashboard.
- **API Layer**:
  - Provides REST or GraphQL APIs for efficient communication between the frontend and backend.
  - Ensures reliable and secure data exchange across all components.
- **Order Processing**:
  - Manages order details, including shipping and billing information.
  - Facilitates a seamless purchasing workflow for a better user experience.

Sanity Database Structure for Marketplace Clothing Platform:

The database structure is designed to ensure effective content management, scalability, and quick data access. Below is the breakdown of its core components:

1.Users

- Maintains user-related information such as account details, delivery addresses, and preferences.
- **Schema Fields**: Name, email, user role, address list, and notification preferences.

2.Products

- Handles product catalog information, including inventory, pricing, descriptions, and categorization.
- **Schema Fields**: Name, description, price, stock quantity, image URLs, and category links.

3.Orders

- Keeps records of customer purchases, payment statuses, and order progress.
- **Schema Fields**: User ID, product IDs, total amount, payment status, shipping details, and order state.

4.Categories

- Organizes products into well-defined groups for easy browsing.
- **Schema Fields**: Category name, brief description, and representative image.

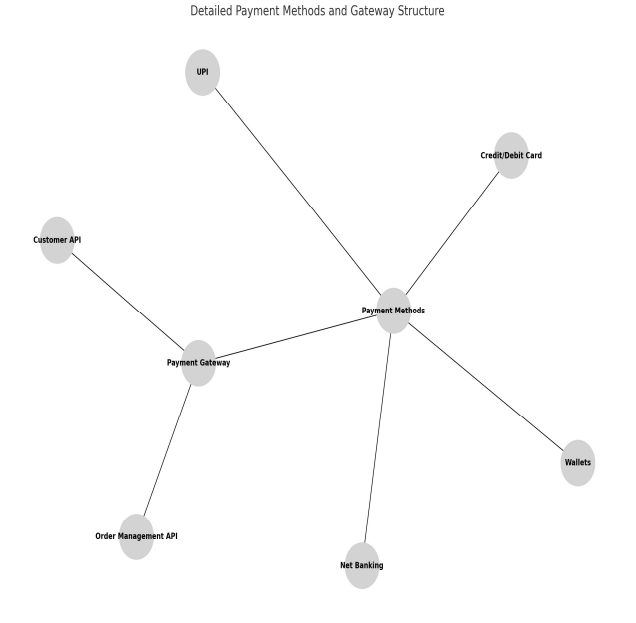
5.Product Reviews (Optional)

- Enables customers to share feedback and rate products.
- **Schema Fields**: User ID, product ID, rating score, and review text.

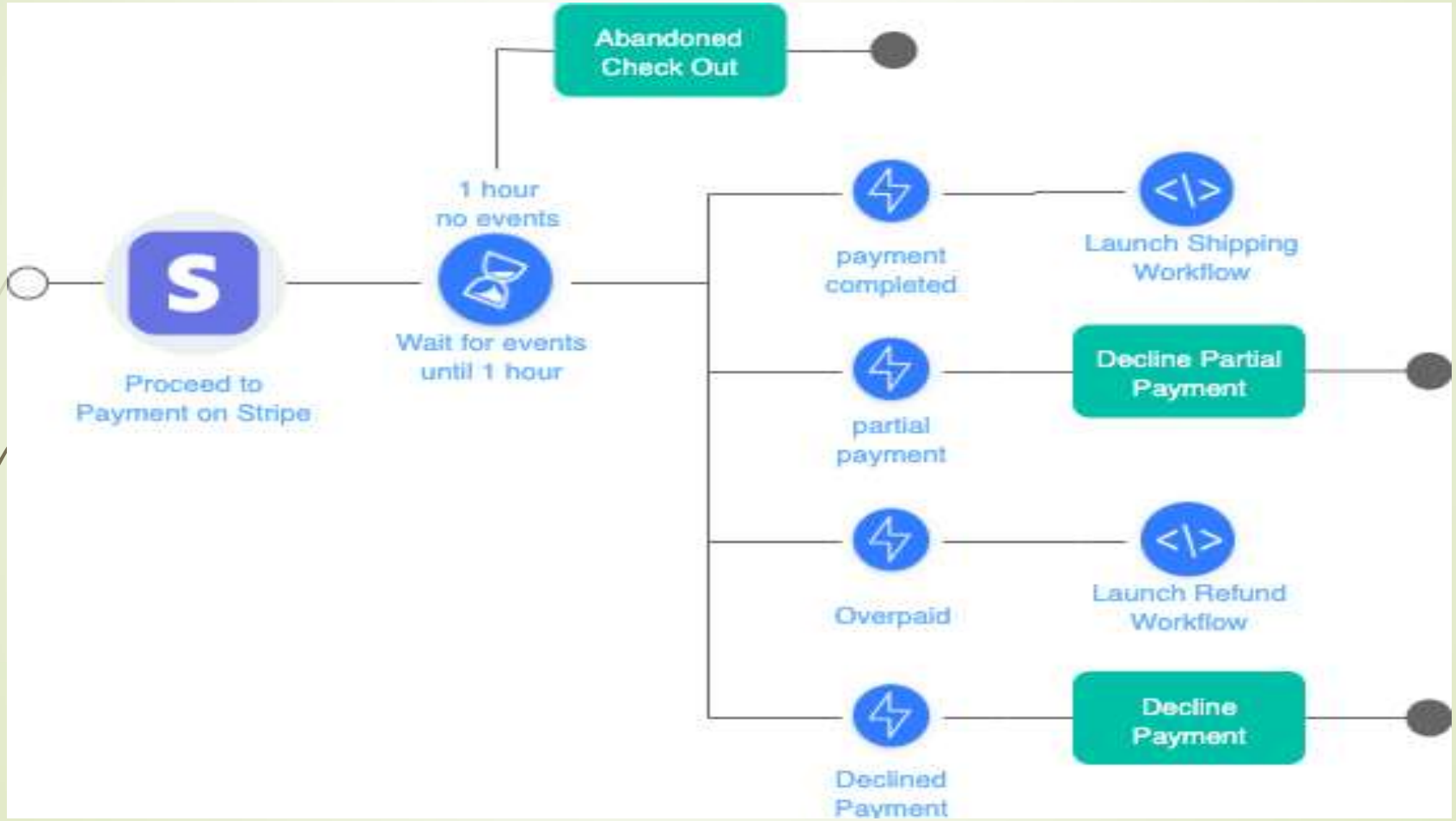
Components for Payment Gateway:

Payment Methods:

- Credit/Debit Card
- UPI
- Wallets
- Net Banking
- **User Details**:
- Email
- Mobile Number
- **Transaction Details**:
- Amount
- Currency
- Status (Pending, Successful, Failed)
- **Security and Authentication**:
- OTP Verification
- 3D Secure
- **Integration Points**:
- Interfaces with **Order Management API** to fetch order details.
- Updates the **Customer API** with transaction history.



Workflow E-Commerce Website



WORKFLOW KEYS:

Workflow 1: User Browses Products

1.The user visits the clothing store's website.
2.The frontend fetches product data from the /products API endpoint.
3.The Sanity CMS provides product details, including:
- •Product ID
- •Name
- •Price
- •Stock availability
- •Size and color options
- •Product images

4.The frontend displays the products in a responsive grid layout with options to filter by:
- •Price
- •Size
- •Color

Let me know if you'd like a visual representation or further edits!

Workflow 2: User Places an Order

1.The user selects clothing items, chooses sizes and colors, and adds them to the cart.
2.The frontend sends a POST request to the /orders API with the following details:
- •Customer
- •Product information
- •Payment

3.The Sanity CMS records the order details in the database.
4.The Payment Gateway securely processes the transaction.
5.An order confirmation email is sent to the user, including:
- •Order summary
- •Estimated delivery date
- •Payment details

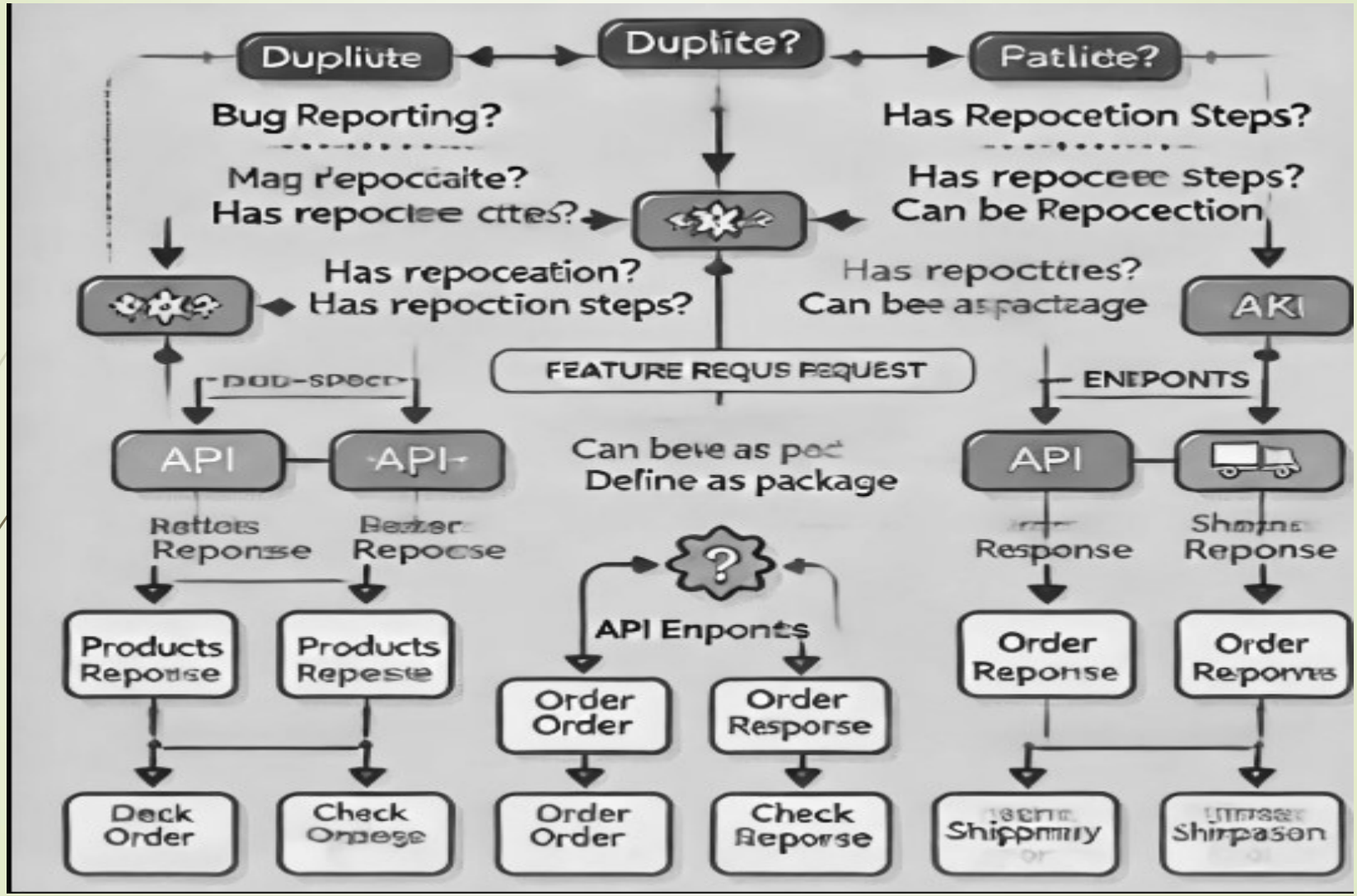
Workflow 3: User Tracks Shipment

1.The user visits the "Track Order" page on the website.
2.The frontend sends a GET request to the /shipment API with the order ID.
3.The shipment tracking API responds with the following details:
- •Shipment ID
- •Current status (e.g., shipped, out for delivery, delivered)
- •Expected delivery date

4.The frontend displays the real-time tracking information, including:
- •Shipment status
- •Estimated delivery timeline

API Integration End Point Api

Endpoint	Method	Description	Payload/Query Parameters	Response Example
/api/products	GET	Fetch a list of all available clothing products.	None	[{"id": "101", "name": "T-shirt", "price": 10.99, "stock": 50, "image": "url-to-image"}]
/api/orders	POST	Create a new order with customer and product details.	{ "customer": {"name": "John Doe", "email": "john@example.com"}, "products": [{"id": "101", "quantity": 2}], "paymentStatus": "Paid" }	{ "orderId": "ORD78910", "status": "Confirmed" }
/api/shipment	GET	Retrieve the shipment status for a specific order.	orderId=ORD78910	{ "shipmentId": "SHIP12345", "orderId": "ORD78910", "status": "Dispatched", "expectedDeliveryDate": "2025-01-25" }


Duplite?
Bug Reporting?
Has Repocetion Steps?
Has repoceee steps?
Can be Repocection
Has repoceation?
Has repoction steps?
Has repoctres?
Can bee aspactaage
AKI
FEATURE REQUS REQUEST
ENEPONTS
API
API
Can bene as pot
Define as package
API
Response
Reponse
Products Reponse
Products Repeste
API Enponts
Order Order
Order Response
Order Reponse
Order Reponses
Deck Order
Check Order
Order Order
Check Reporse
Shippmmy
Shirpason

Conclusion

An e-commerce website for clothing serves as a dynamic platform for customers to explore, select, and purchase apparel conveniently from anywhere. By integrating features like a user-friendly interface, detailed product descriptions, size guides, filters, and secure payment options, the website ensures a seamless shopping experience. It also provides tools such as personalized recommendations, discounts, and a hassle-free return policy to enhance customer satisfaction. Effective inventory management and APIs for real-time order tracking further streamline operations. Ultimately, the platform bridges the gap between fashion brands and consumers, fostering growth and accessibility in the clothing industry.