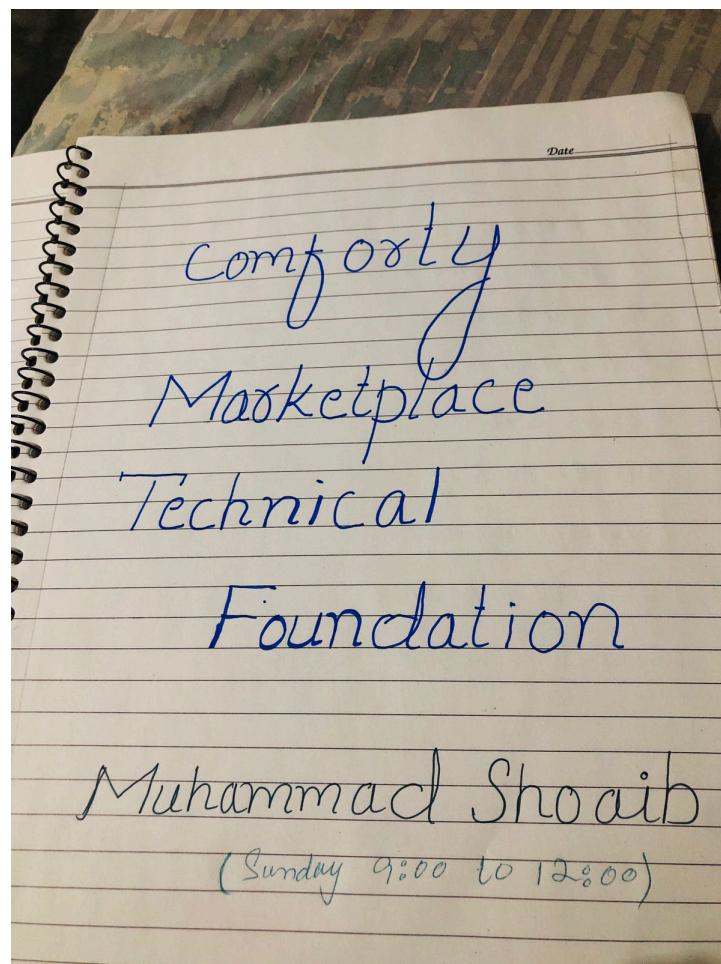


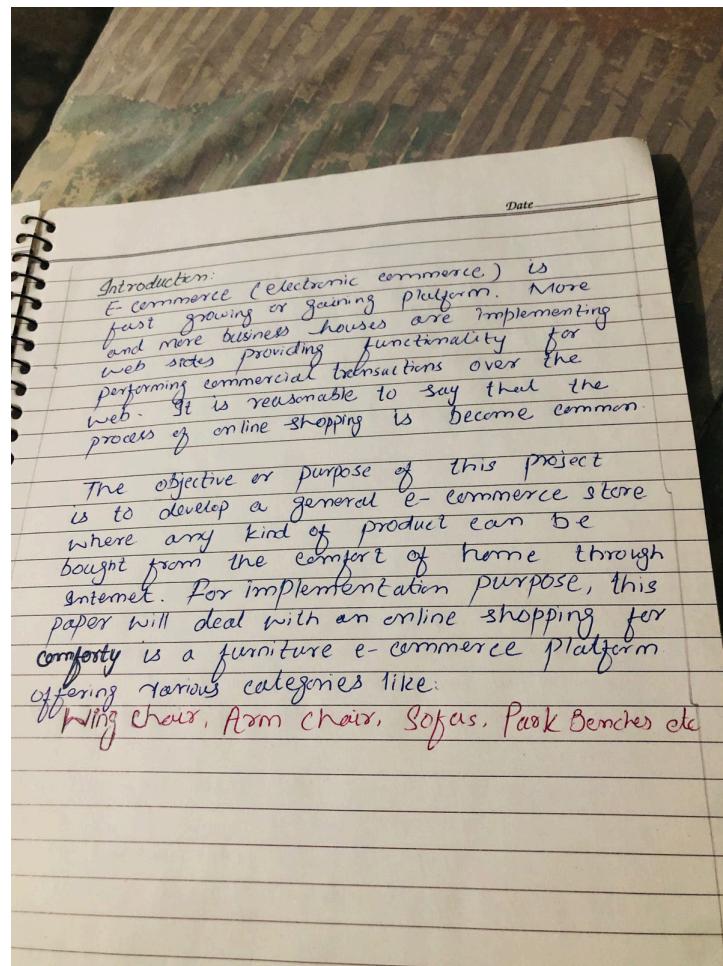
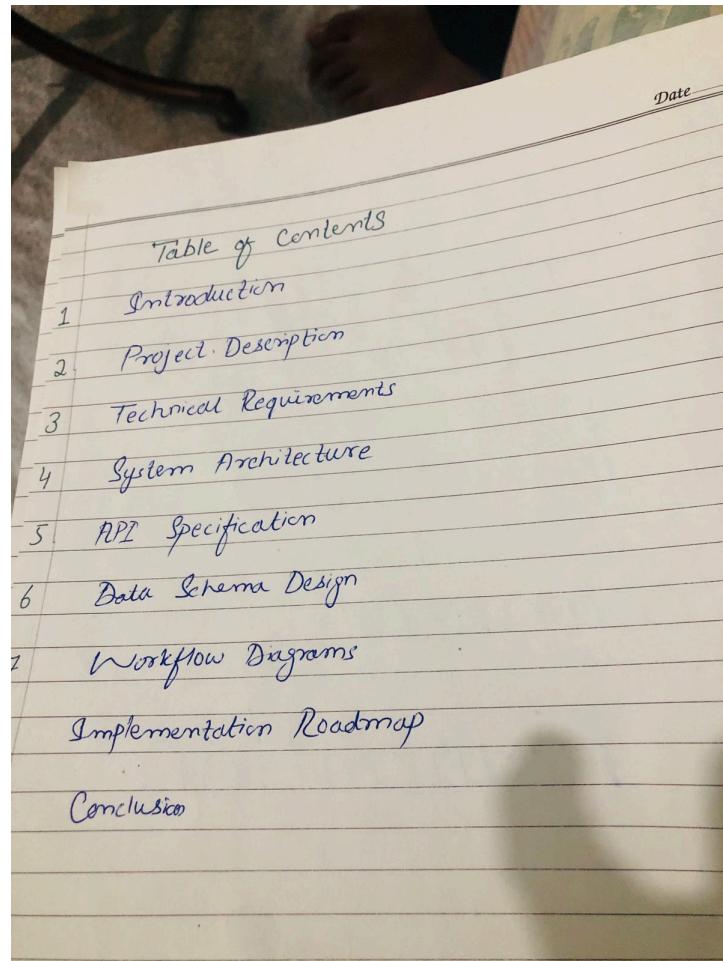
Technical Foundation for the Marketplace Journey

Hackathon 3: Day_2

Muhammad Shoaib

Sunday 9AM - 12PM





Date _____

Project Description

- * Any members can register and view available products.
- * Only registered members can purchase multiple products regardless of quantity.
- * Contact us page is available to contact admin for queries.

There are four roles available:

- > Visitor
- > User
- > Operator
- > Admin

Visitors can view, search available products.

User can view and purchase products.

Operator can excess and option to add or can view everything of admin panel.

Admin can add and edit products and also remove.

Admin can add users, edit user information and can remove users.

Admin can ship order to user based on order placed by sending confirmation mail.

Date _____

Technical Requirements

1. **Frontend Requirements:**

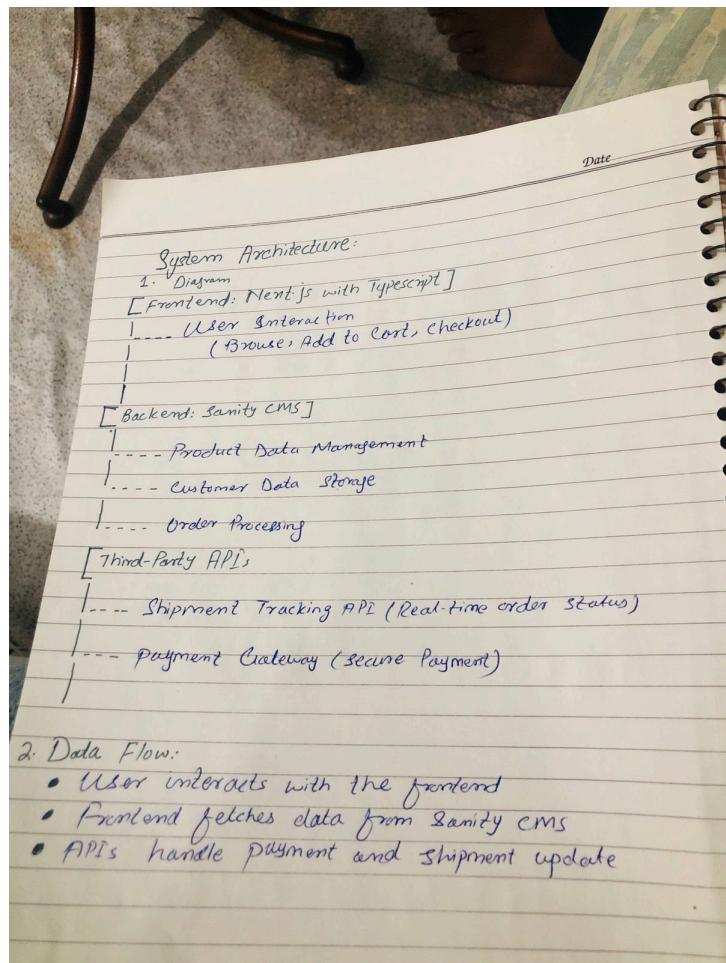
- Next.js framework with typescript, tailwind.
- Responsive design for mobile and desktop.
- App Routers
- Home, Product listing, product details, Cart, checkout, Order confirmation.

2. **Backend Requirements:**

- Sanity CMS for managing
- Product Data (categories, Price, Stock)
- Customer Data (Profiles, Order History)
- Order Data (Products, Quantities, Status)

Third-Party APIs:

- Shipment tracking API for real-time delivery status
- Payment Gateway for secure transactions



Date _____

API Specification

1. Products API

Endpoint: /products	Method: Get
Purpose: Retrieve all available products	
Request: No parameters required	
Response:	{
	{id: "091",
	{name: "Stool chair",
	{price: 320,
	{category: "Furniture",
	{stock: 25,
	{image: "url-to-image",
	{description: "Clip Stool chair",
	}

2. same as more objects

Orders API

Endpoint: /orders	Method: POST
Purpose: Create a new order	
Request payload:	"206",
	{ "customerId": "206",
	"Products": [
	{"productId": "303", "quantity": 10},
	{"productId": "304", "quantity": 20}
], "total Price": 920,
	"Delivery Address": "ABC main street"

Response:	205+1
	{
	"orderId": "3051",
	"status": "order paid",
	"estimated Delivery": "2021-06-15"
	}

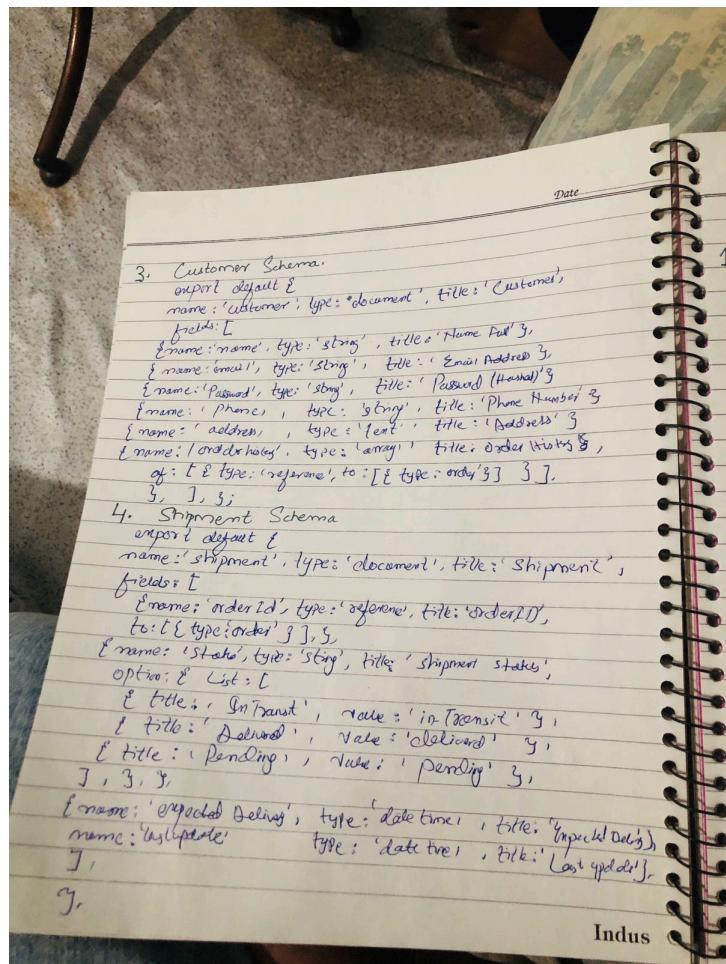
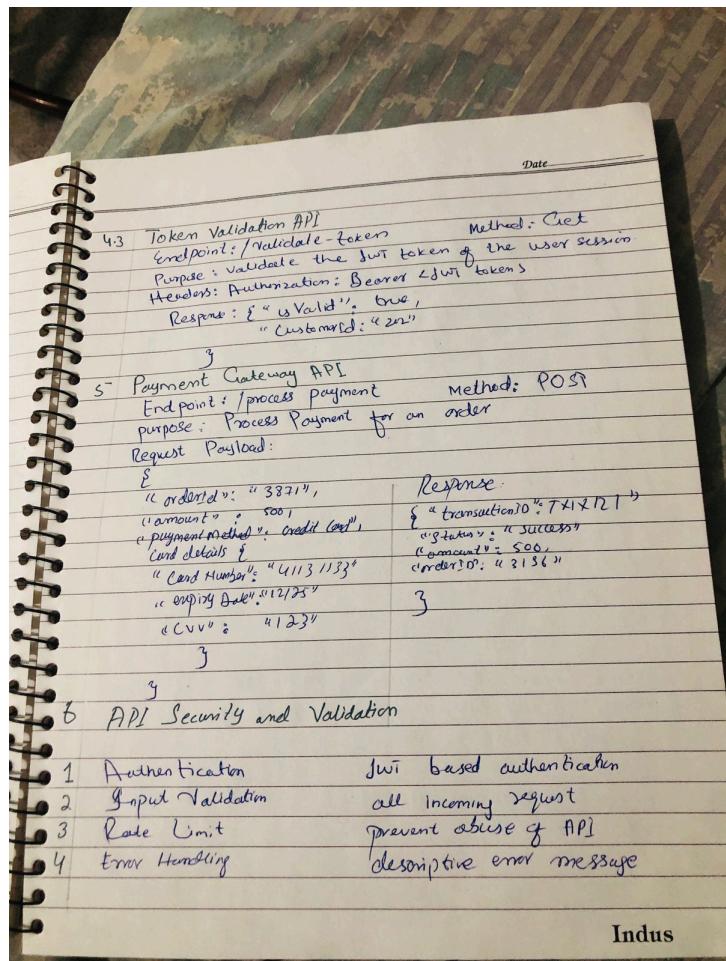
Indus

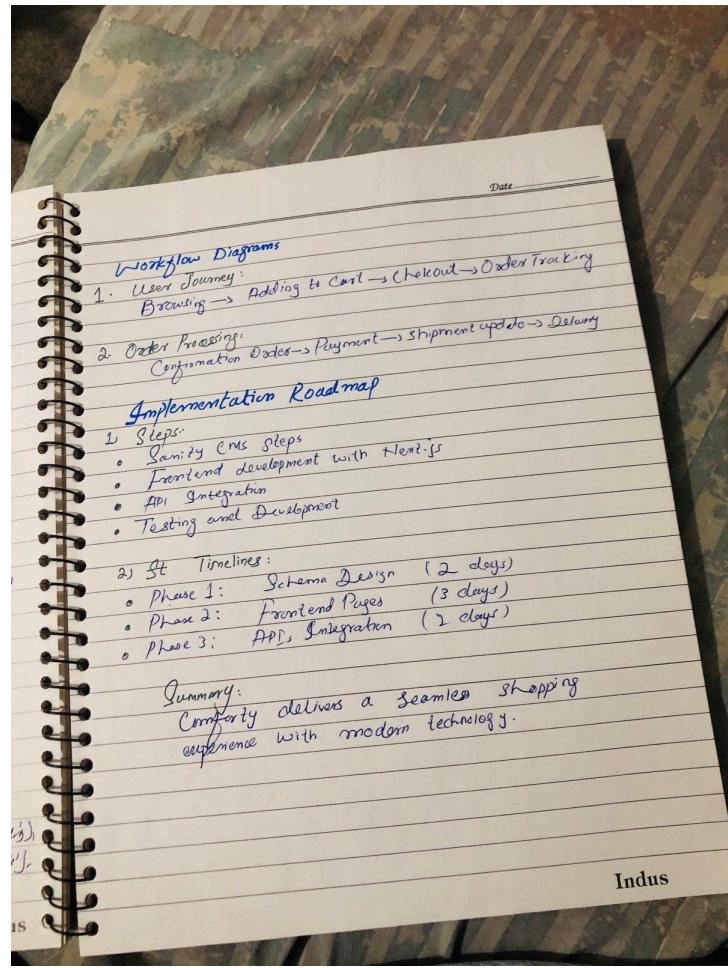
Date

3 Shipment Tracking API
 Endpoint: /shipment/status Method: GET
 Purpose: Retrieve the shipment status of an order
 Request Parameters: orderId: Unique ID of the order
 Response:
 { "orderId": "98765",
 "status": "In Transit",
 "expectedDelivery": "2025-1-20",
 "lastUpdate": "2024-01-05 14:35:00" }
 4 Customer Authentication APIs
 4.1 Sign-up API
 Endpoint: /signup Method: POST
 Purpose: Register a new customer
 Request Payload:
 { "name": "M. Sheekh",
 "email": "sheekhshab056@gmail.com",
 "password": "*****",
 "phone": "+91 23456789",
 "address": "ABC Komachi" }
 Response:
 { "message": "Successfully Registered!",
 "customerId": "101" }
 4.2 Sign-in API
 Endpoint: /signin
 Purpose: Authenticate an existing customer
 Request Payload:
 { "email": "sheekhshab056@gmail.com",
 "password": "123456" }
 Response:
 { "token": "JABCB291",
 "customerId": "303" }
 4:

Date

2 Order Schema
 export default {
 name: 'order', type: 'document', title: 'Order',
 fields: [{
 name: 'customerId', type: 'reference', title: 'Customer ID',
 to: { type: 'customer' } },
 { name: 'products', type: 'array', title: 'Products',
 of: [{ type: 'object',
 fields: [{
 name: 'productId', type: 'reference', title: 'Product ID',
 to: { type: 'product' } },
 { name: 'quantity', type: 'number', title: 'Quantity' }] }] },
 { name: 'totalPrice', type: 'number', title: 'Total Price' },
 { name: 'deliveryAddress', type: 'string', title: 'Delivery Address' },
 { name: 'status', type: 'string', title: 'Order Status' },
 options: [{ title: 'Pending', value: 'pending' },
 { title: 'In Progress', value: 'in progress' },
 { title: 'Delivered', value: 'delivered' }],
 { name: 'createdAt', type: 'date-time', title: 'Order Place At' }] }
 3, 4:
 Indus





This document lays the groundwork for the marketplace journey.