Hackathon Day:-2

Market place Technical Foundation
Furniture website

The Technical Foundation :-

• Frontend Req:-
i) Basic Structure and attractive design.
ii) Should be User-friendly.
iii) Add Responsiveness.
iv) Dynamic routing (if necessary)
v) Generating dynamic pages for each product.
vi) Add to cart functionality.

• User Authentication:-
i) Adding functionality and user authentication.
ii) Get User info, as well as assign unique id to that User.

• Inventory Management:-
i) Firstly, get the API for products.
ii) Work on basic admin Setup, where it can manage quantity of products. Where it can be also get order details.
iii) Checkout and billing.
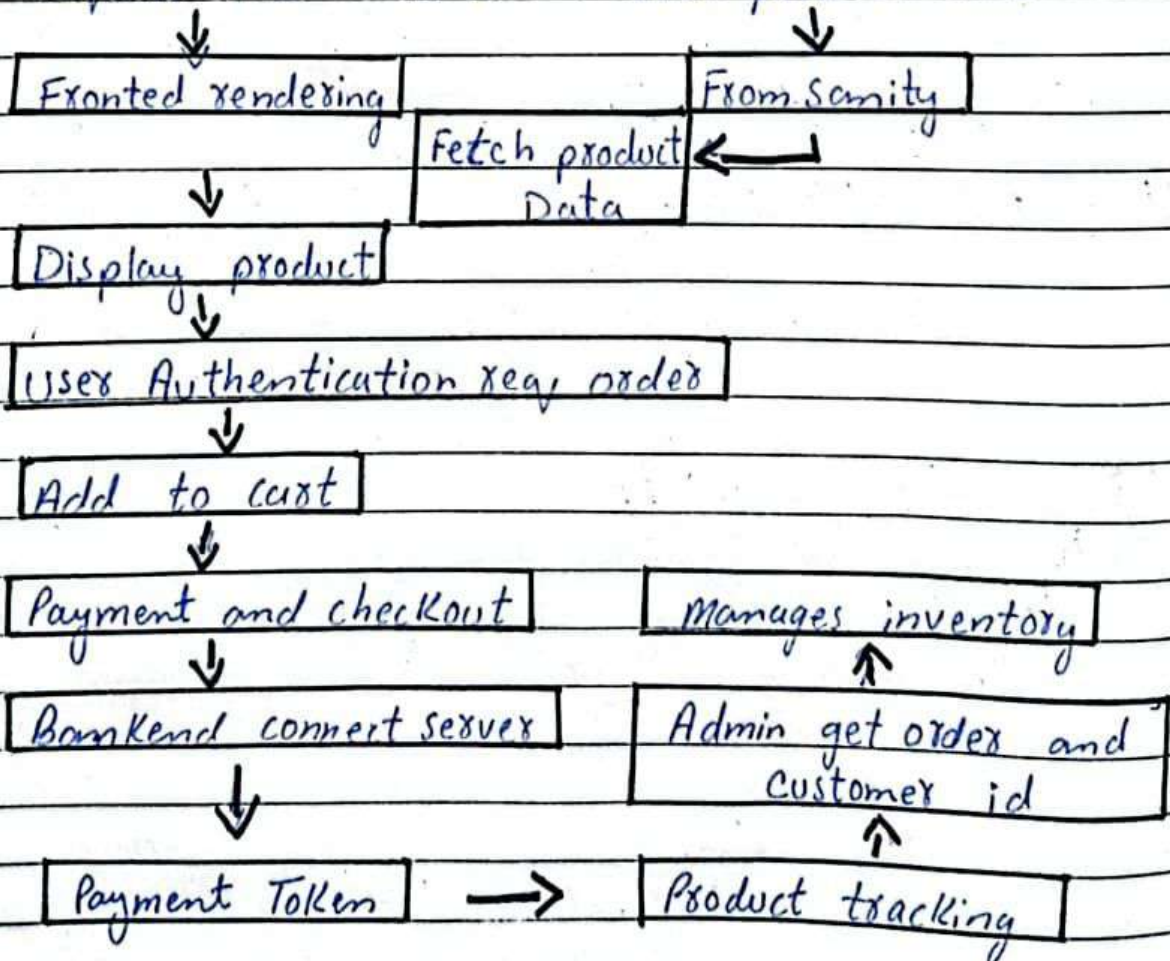iv) Manage all backend, working and getting API from Sanity.

Order Tracking :-

i) Using platforms like Shipengine to track the product.

Required tools :-

→ Next.JS environment.
→ Tailwind CSS.
→ Sanity or (For API)
→ ShipEngine.
→ Stripe

Website Architecture :-

```
User request the website ──→ fetch product API
          ↓                          ↓
  ┌──────────────────┐      ┌──────────────────┐
  │ Fronted rendering │      │   From sanity    │
  └──────────────────┘      └──────────────────┘
                    ┌──────────────┐ ↖
                    │ Fetch product │←──┘
          ↓         │    Data       │
  ┌──────────────┐  └──────────────┘
  │ Display product │
  └──────────────┘
          ↓
  ┌──────────────────────────┐
  │ User Authentication req order │
  └──────────────────────────┘
          ↓
  ┌──────────────┐
  │ Add to cart  │
  └──────────────┘
          ↓
  ┌──────────────────────┐      ┌──────────────────┐
  │ Payment and checkout │      │ Manages inventory │
  └──────────────────────┘      └──────────────────┘
          ↓                              ↑
  ┌──────────────────────┐      ┌──────────────────────┐
  │ Bankend connect server │      │ Admin get order and   │
  └──────────────────────┘      │    customer id        │
          ↓                      └──────────────────────┘
  ┌──────────────┐                       ↑
  │ Payment Token │ ──→  ┌──────────────────┐
  └──────────────┘       │ Product tracking  │
                         └──────────────────┘
```

Data Schema:-

i) Product_Data (with example):-
   { Product_Id : 19601,
     Product_name : "timber",
     Slug : slug,
     Price : 29,
     Stock : 200 }

ii) Order_Data
    { order_id : 00152,
      Customer_id : 019,
      Customer_email : "abc@gmail.com",
      Customer_contact : 03151221218,
      Customer_address : "......, Karachi" }

iii) Customer_Data {
      Customer_id : 019,
      Customer_name : "Abc",
      Customer_email : "abc@gmail.com",
      Customer_contact : 03151221218,
      Customer_address : "......, Karachi" }

iv) Delivery_Zone {
     Zone_id : 71,
     Zone_name : ['malir', 'Gulshan-e-Iqbal']
     Cover_area : "All over Karachi"

-: Relations of these :-

1) Product → Connect to order through product ID.
2) orders → Connect to Customer through Customer ID.
3) Customer → Connect to Zone ID.