Marketplace Technical Foundation – [Trend Nest]

System Architecture:

1. The user will first signup/login and his data of the user will be stored in sanity.io CMS.
2. Then the user will be shown the Home page and can navigate between the site and can select the product.
3. User will add the product to cart.
4. From the cart page user can check out the order after and can perform the payment.
5. After payment the user will be given a confirmation notification.
6. The user will be given a link from which he can track his product.

Frontend Requirements:

User-friendly Interface:

Make an interface which a non-technical person can easily use and navigate around it.

Responsive design:

Make sure that the UI is responsive so the website is accessible from every device.

Essential Pages:

Make sure that you have the essential pages of an E-commerce website.

- Home
- Products Listing
- Product Details
- Login/Signup
- Cart
- Checkout

- Shipment tracking
- Payment

Backend Requirements:

Use Sanity.io for:

1. **Content management:** Use sanity to manage content which in our case will be product details and data.
2. **Customer data:** The customer's info will be stored in sanity when signup.
3. **Order detail:** Order's detail such as name, id, quantity will be stored in sanity when an order is placed.
4. **Payment records:** Record payment in sanity such as amount, payment status, payment history.

Schema Designs:

Make schemas which align with business goals for every requirement such as: Product, Orders, Customer data and Payments.

Third-Party APIs:

Use third-party APIs to outsource complex goals eg: Real time order tracking and Payment gateways.

API Endpoints:

Get all products from API:

- Endpoint: /products
- Method: GET
- Description: Fetches all products form sanity and displays them to frontend.
- Response:

{
"id": 1,
"name": "jackets",
"price": 50,
"stock": 200,

“description”: “Description of the product”,
“tags”: “winter collection”,
“slug”: ”jackets”,
"category": "Jackets",
“discountPrice”: 30,
"image": "jacket.jpg"
}

Creates new orders:

- Endpoint: /orders
- Method: POST
- Description: Creates a new product in sanity.
- Response:

{
"customerId": "09876453",
"customerName": "Mustafa",
"address": "Karachi, Pakistan",
"products": [
{
"productId": 1,
"quantity": 2,
"price": 50
},
{
"productId": 2,
"quantity": 1,
"price": 35
}],
"totalAmount": 85,
"paymentStatus": "Pending"
}

Track Shipments:

- Endpoint: /track_orders
- Method: GET
- Description: Track orders from third party API.
- Response:

{

"shipmentId": 1,
“orderId”: 09876543,
"expectedDate" : “03.06.2025”,
“status” : “Ware house”,
}

Payment Process:

- Endpoint: /payment
- Method: POST
- Description: Recives payments from users using third party API.
- Response:
  {
  "amount": 1000,
  “orderId”: 09876543,
  "currency" : “PKR”,
  “status” : “completed”,
  “successUrl”:”https://abc.ksmxkww/kdm”,
  “cancelationUrl”:”https://abc.ksmxkww/kdm”,
  }

Workflow Diagram: