Day 2: Marketplace Technical Foundation - [General E-Commerce Website]

Name: Mahnoor Ansari

Technical Requirements And System Architecture:

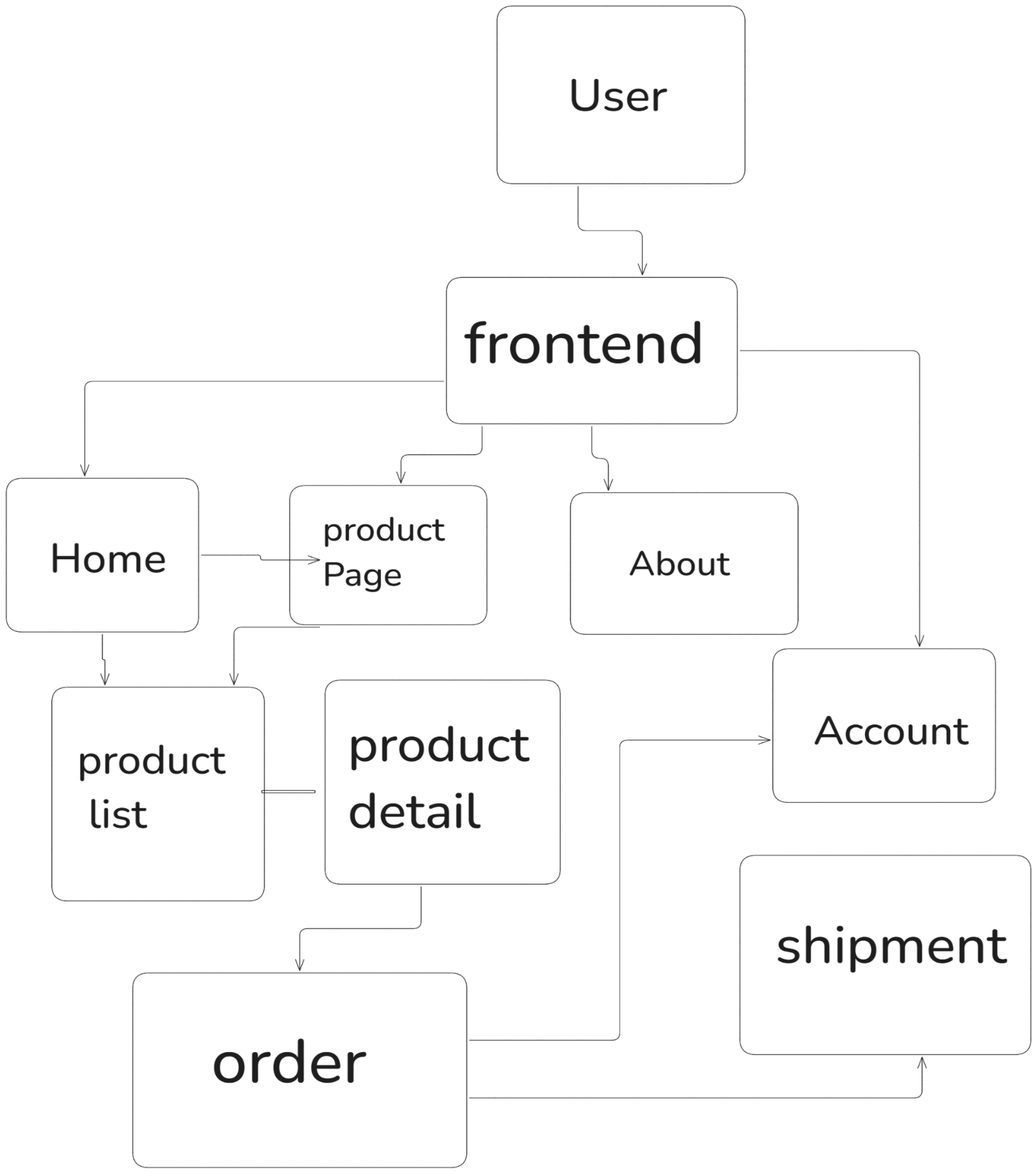
- User-friendly interface for browsing products.
- Design the website to be fully responsive and adaptable to all devices.
- The design should be simple and easy to use.
- Products should be organized into clear categories to make browsing easy for users.
- The website must include essential pages like Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation to ensure a complete shopping experience.

Essential Pages:

Home: This could be the homepage, which will be the website's landing page.
Products: This section will have a product listing page, and there will be a separate page for each product's details.
About: This page will be about the company or website. Contact: This page will have contact details and a form.
Account: This section will allow users to manage their accounts (Login, Sign Up, Dashboard). Cart: This section will display the user's shopping cart and have an option for checkout.

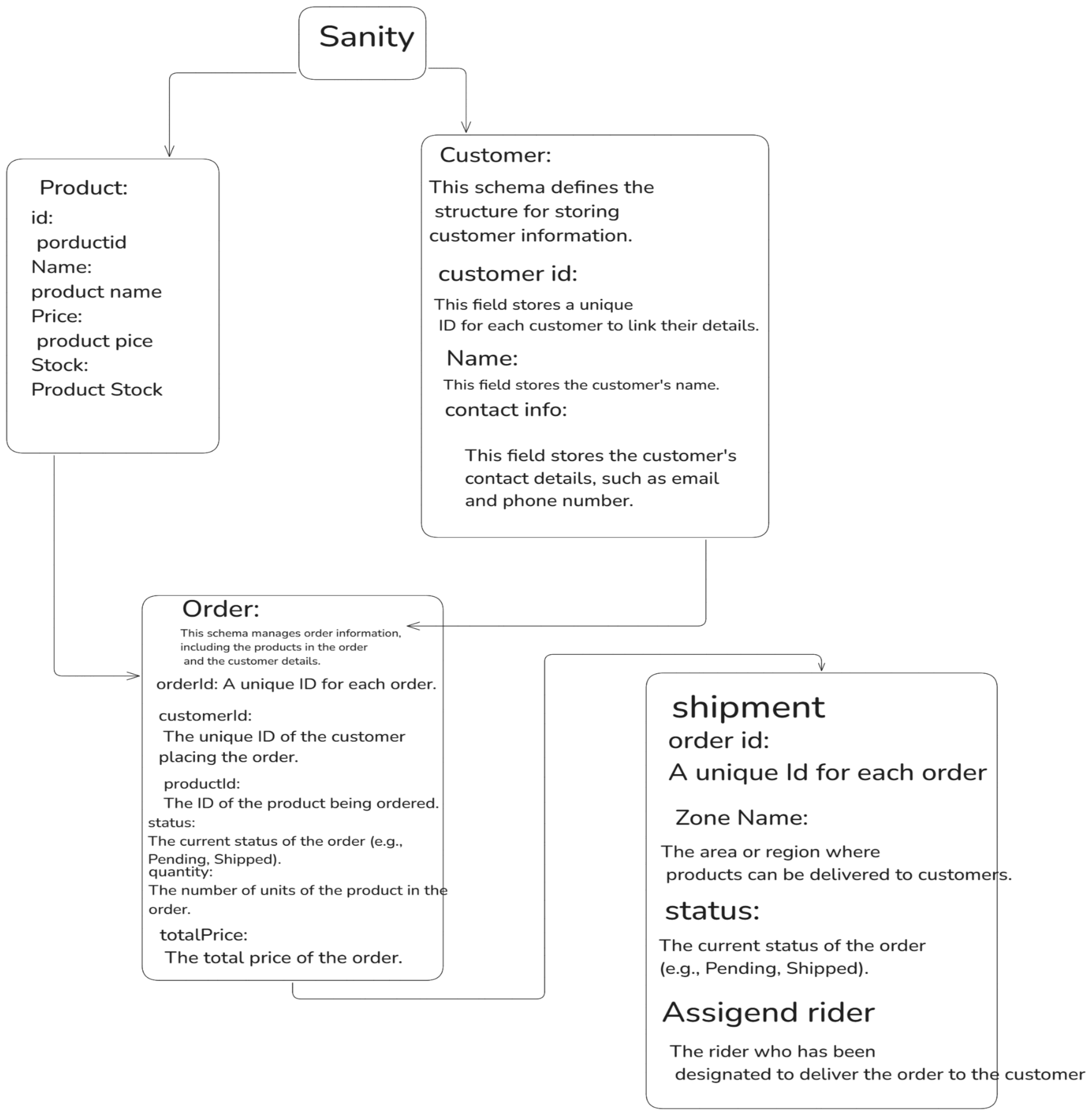
Shipment: This section will manage shipping details, such as address, shipping method, and payment.

Order: This page will show users their previous orders and order history.



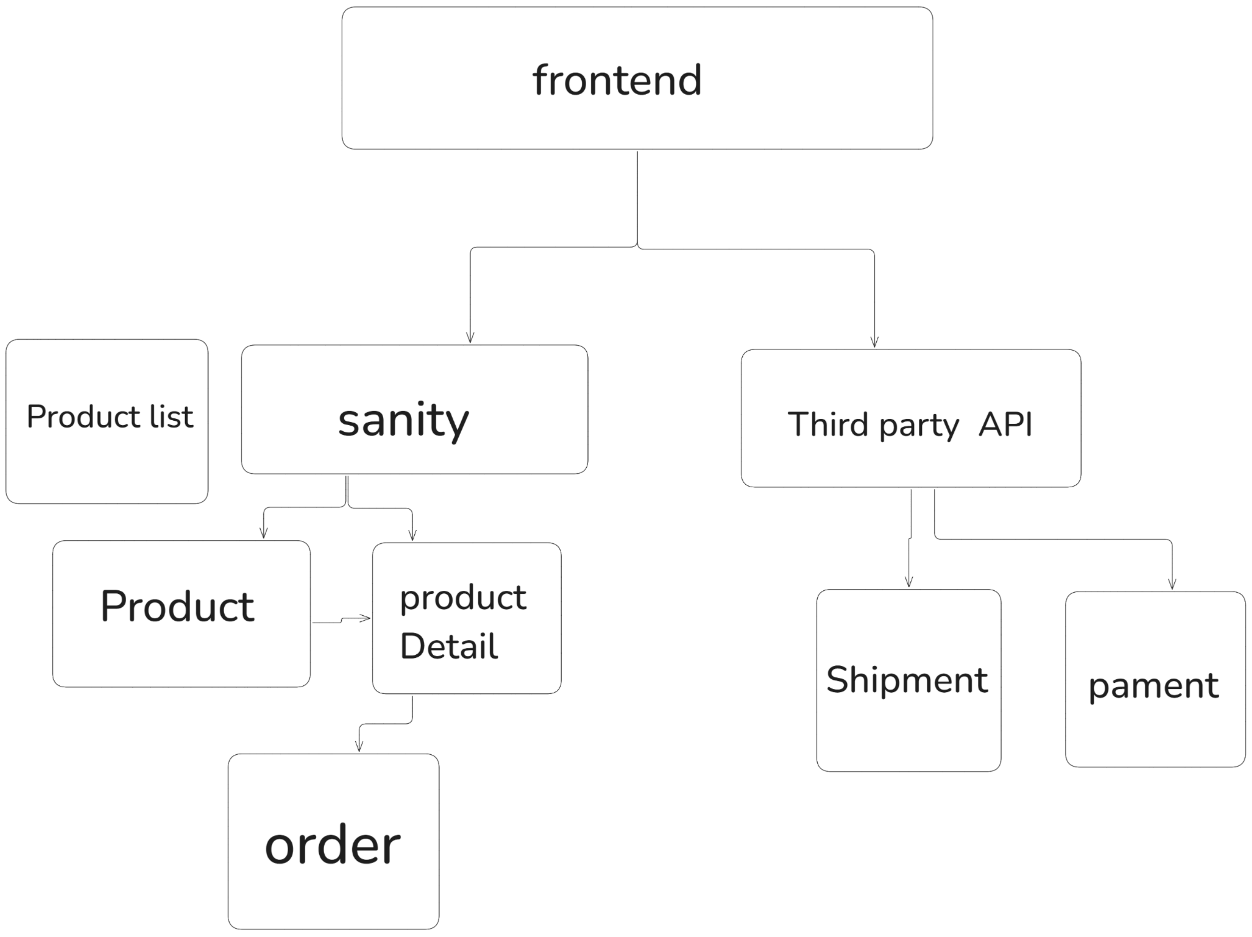
Sanity as Backend:

Sanity CMS will be used to manage product data, customer details, and order records for the marketplace. Sanity will act as the database, storing all the necessary information for smooth operations.


Sanity
Product:
id:
porductid
Name:
product name
Price:
product pice
Stock:
Product Stock
Customer:
This schema defines the structure for storing customer information.
customer id:
This field stores a unique ID for each customer to link their details.
Name:
This field stores the customer's name.
contact info:
This field stores the customer's contact details, such as email and phone number.
Order:
This schema manages order information, including the products in the order and the customer details.
orderId: A unique ID for each order.
customerId:
The unique ID of the customer placing the order.
productId:
The ID of the product being ordered.
status:
The current status of the order (e.g., Pending, Shipped).
quantity:
The number of units of the product in the order.
totalPrice:
The total price of the order.
shipment
order id:
A unique Id for each order
Zone Name:
The area or region where products can be delivered to customers.
status:
The current status of the order (e.g., Pending, Shipped).
Assigend rider
The rider who has been designated to deliver the order to the customer

Third Party API:

Use third-party APIs to handle shipment tracking, process payments, and manage other necessary backend operations. This will improve the overall performance and capabilities of your platform.



Endpoint	Method	Description	Request Body	Response
/api/products	GET	Fetch all products	None	[{id, name, image, price, description}]
/api/products/:id	GET	Get product by ID	None	{id, name, image, price, description}
/api/cart	POST	Add item to cart	{productId,productname, quantity, price}	{cartId, items: [{productId, productname, price quantity}]}
/api/orders	POST	Create a new order	{cartId, customerid,productId, paymentInfo}	{orderId, customerid, productID, quantity, paymentinfo, status}