Planning the technical Foundation

1. ***Technical Requirement:***

It's an E-commerce website contains pages like Home, about, cart, Product Detail , Product listing.

Frontend Requirement:

- ### **User Friendly Interface:**

Homepage contains hero section , products featured.

Product Listing shows all the product dynamic card

Cart section where user shows there products and checkout from here

Product Detail displays products name, price , description , image and quantity

Responsive designs on all devices like mobile , tablets , laptop, larger screens.

- ### **Sanity CMS as Backend:**

Sanity CMS stores data of products detail like product name, product image, description , price , quantity.

Sanity CMS stores orders data like customers details and shipment data.

- ### **Designing Schema:**

Stores details of each product name, image, description, price, and quantity.

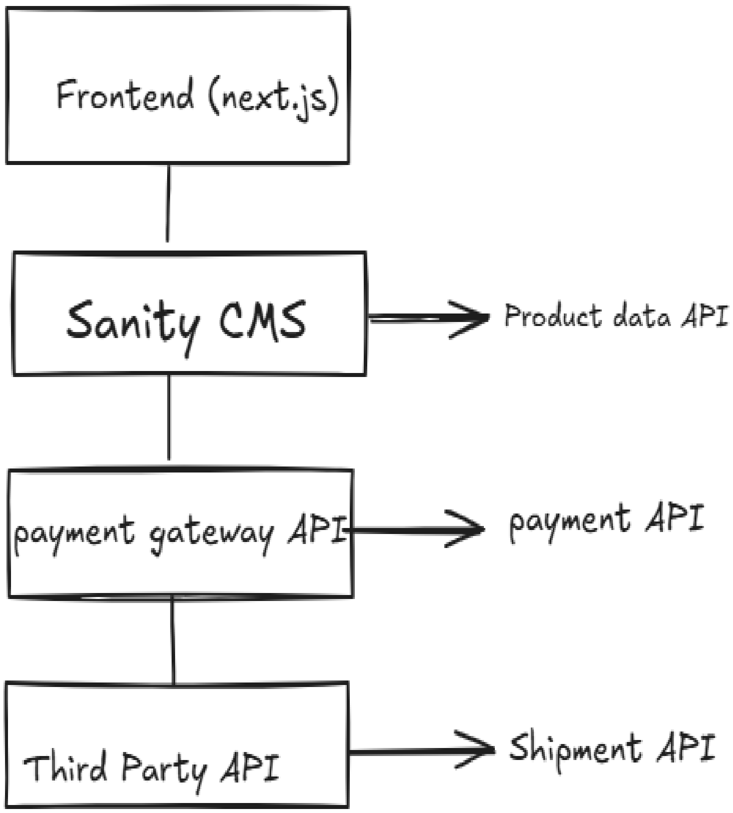
Captures customer details, shipping address, and the list of products in the order.

- ### **Third Party API:**

Payment Gateway API : Stripe, Lemon Squizy

Shipment API : ShipEngine , Shippo etc

2. ***System Architecture:***



Key Workflows:

1. User registration:

User sign up in the website ,the data of the user stored in sanity after that user got a confirmation mail.

2.Product Browsing:

Frontend makes request to product API from sanity to pass product data.

3.Order Placement:

User adds items to cart , proceeds to checkout order details saved in sanity

4.Shipment Tracking:

Shipment tracking from third party API shows status of tracking order and shows user real time updates.

3. API Requirement:

Endpoint Name	Method	Description	Response
/products	GET	Fetch all products from sanity	Product detail { “id”:1, “name”:”product1”, “price”:90$ }
/orders	POST	Creates new order in sanity	{“orderId”:1, “customerName”:”john”, “productId”:2}
/shipment	GET	Track order from shipment API	{“shipmentId”:”df32f65dg”, “status”:”successful”}

4. ***Sanity Schema:***

1. /Products:

```
{ name: 'product',
 type: 'document',
 title: 'Product',
 fields: [ { name: 'name',
 type: 'string',
 title: 'Product Name' },
{ name: 'price',
type: 'number',
title: 'Price'},
],
 };
```

5. /Orders:

```
export default {
  name: 'order',
  type: 'document',
  title: 'Order',
  fields: [
    {
      name: 'customerName',
      type: 'string',
      title: 'Customer Name',
    },
    {
      name: 'productId',
      type: 'reference',
      to: [{ type: 'product' }],
      title: 'Product',
    },
  ],
};
```

3. /Shipment Schema:

```
export default {
  name: 'shipment',
  type: 'document',
  title: 'Shipment',
  fields: [
    {
      name: 'shipmentId',
      type: 'string',
      title: 'Shipment ID',
    },
    {
      name: 'status',
      type: 'string',
      title: 'Shipment Status',
    },
  ],
};
```