Market Place Technical Foundation :-

Step 1 : Define Technical Requirements.

) Frontend :

Using Next.js and Tailwind CSS, we design a user-friendly interface with essential pages like Home, Product Listing, Product Details, Cart, checkout & order confirmation, ensuring fast performance and responsive design.

) Sanity CMS :-

For Manage Backend data we use Sanity CMS to manage product data, customer details and order records.

) Third Party APIs :-

Integrated API to manage other backend services like shipment tracking & payment gateway.

Step 2: Design System Architecture

Key Workflow:-

① User Registration:-

- User fill the form of registration.
- User data store in Sanity.
- A confirmation email send to user.

② Product Browsing:-

- User view product category.
- Fetch product data (e.g image, description, price from Sanity.
- Display product Listing dynamically on the site.

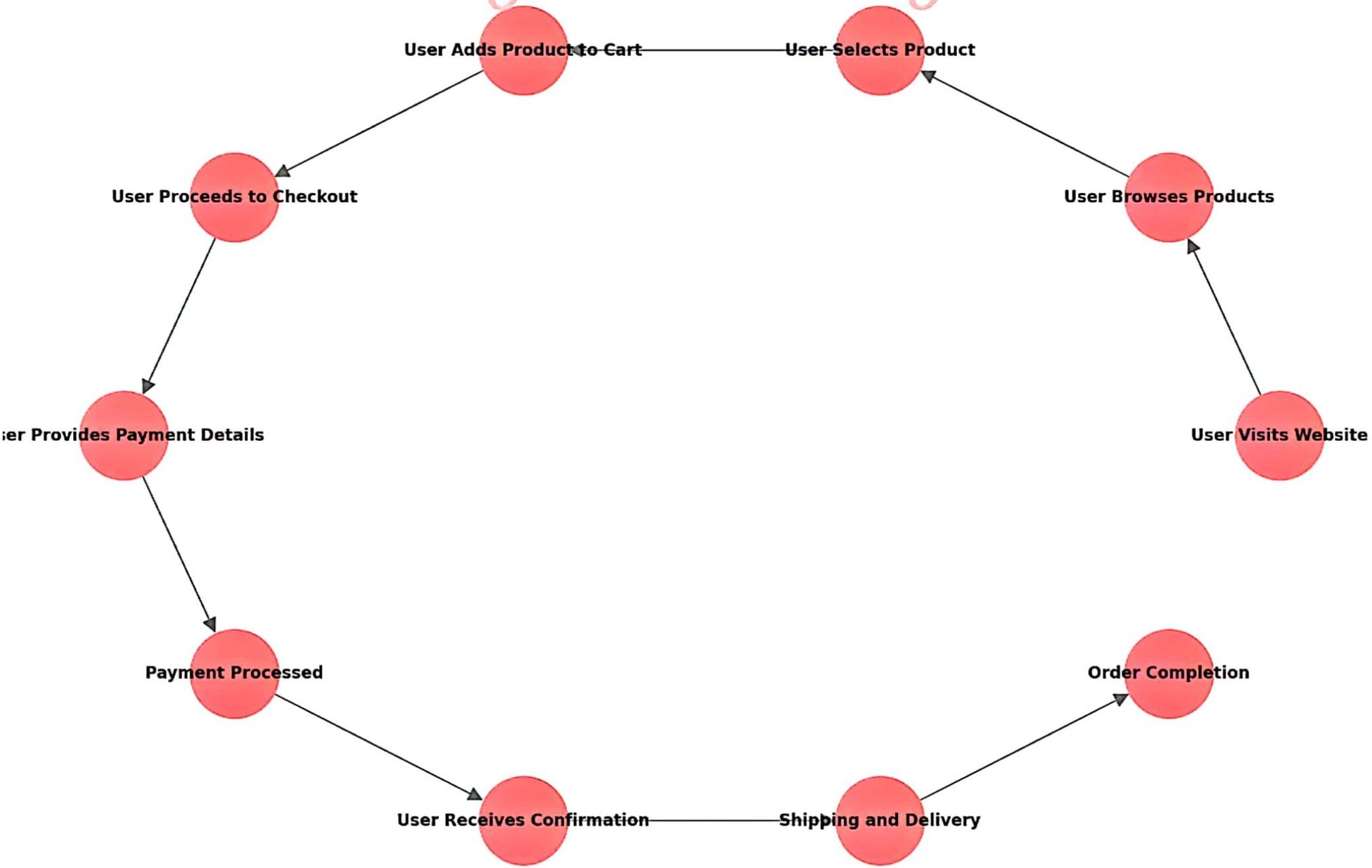
③ Order:-

- User add item to cart & checkout.
- order details (items, quantity and user information) are sent to Sanity CMS.
- Payment gateway. Securely process the payment and confirm the transaction.

④ Shipment Tracking :-

- Order status update with shipping details (tracking number) from Sanity.
- Fetches real time shipment status by shipment tracking APIs.
- Shipment status (e.g process, Delivered) display to the user.

High Level Diagram



Step 3: Plan API Requirement.

Endpoint: /Product.

Method: Get.

Description: Fetch a list of all products.

Response:


```
[ { "id": 1,
"name": "Product Name",
 "description": "Product Description",
"price": 99.99,
"category": "Category Name",
 "stock": 100 }
]
```

Endpoint: /api/orders
Method: POST
Description: Place a new order.
Request Body Example:

```
{
  "userId": 101,
  "items": [
    {
      "productId": 1,
      "quantity": 2
    },
    {
      "productId": 3,
      "quantity": 1
    }
  ]
}
```

Response Example


```
{
  "id": 3,
  "userId": 101,
  "items": [
    {
      "productId": 1,
      "quantity": 2
},
    {
      "productId": 3,
      "quantity": 1
    }
  ],
  "totalPrice": 149.98,
  "status": "pending",
  "createdAt": "2025-01-15T08:00:00Z"
}
```

Endpoint: `/api/shipments/{id}`

Method: GET

Description: Retrieve details of a single shipment by its ID.

Response Example:

```
{
  "id": 1,
  "orderId": 101,
  "trackingNumber": "TRACK12345",
  "status": "in-transit",
  "currentLocation": "Chicago, IL",
  "estimatedDelivery": "2025-01-20"
}
```

Step 4: Documentation.

System Architecture Overview:-

) Frontend :

Using Next.js and Tailwind CSS, we design a user-friendly interface with essential pages like Home, Product Listing, Product Details, Cart, checkout & Order Confirmation, ensuring fast performan and responsive design.

) Sanity CMS:-

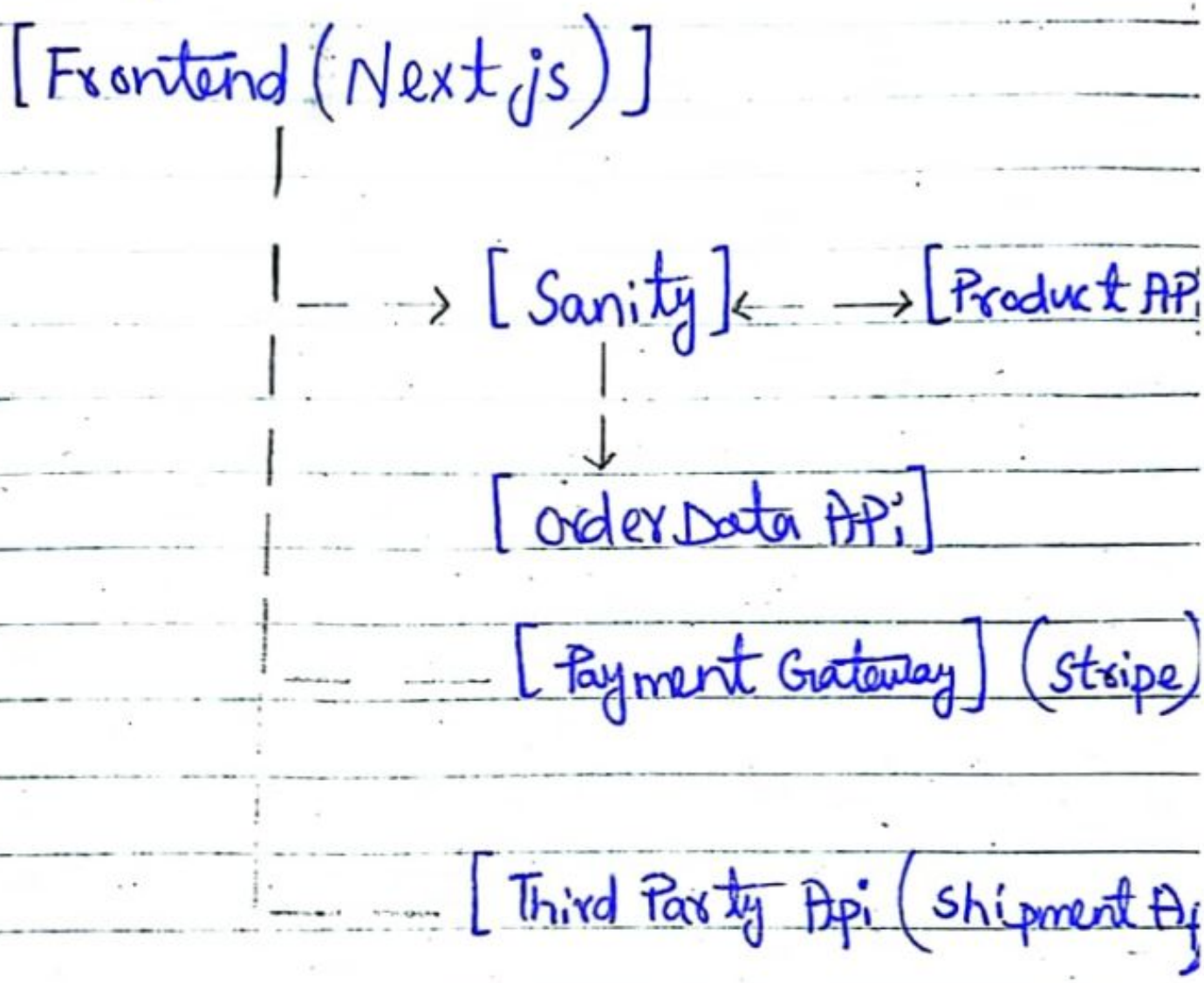
For Manage Backend data we use Sanity CMS to manage product data, customer details and order records.

) Third Party APIs:-

Integrated API to manage other backend services like shipment tracking & payment gateway.

Step 4: Documentation.

Diagram:-



Sanity Schema Example:

Step 4

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productId',
      title: 'Product ID',
      type: 'string',
    },
    {
      name: 'name',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'stockQuantity',
      title: 'Stock Quantity',
      type: 'number',
      validation: Rule => Rule.required().min(0)
    }
```