Marketplace Technical Foundation – Syed Ali Kazim

1. Technical Requirements

Frontend Requirements:

- Easy to use User Interface for browsing products.
- Responsive design using Tailwind utility classes for mobile and desktop users.

Essential pages:

- Home: All pages are accessible using Navigation menu for both Desktop and Mobile.
- Product Page: Displays products by category or search.
- Product Details: Displays detailed information about selected products.
- Cart: Shows items added by the user.
- Checkout: Handles payment and shipping details.
- Order Confirmation: Displays the order summary and confirmation.

Backend Requirements (Sanity CMS):

- Product Management: Manage product details, inventory, and categories.

- Order Management: Store order details, customer info, and payment status.

Schemas:

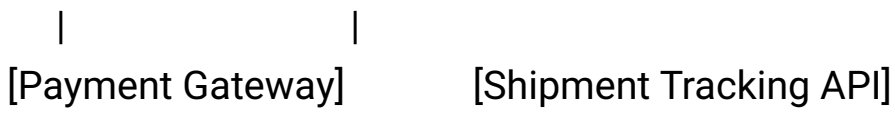
- Product Schema: Includes fields like id, name, price, image, stock, and description.

- Order Schema: Includes customer details, product details, payment status, and order status.

2. System Architecture

Architecture Overview:

This architecture diagram demonstrates the interactions between the frontend, backend, and APIs.



Key Workflows:

1. User Browsing Products:

User accesses the site and goes to products page→ API fetches products → Data displayed dynamically.

2. Placing an Order:

User adds items to cart → Proceeds to checkout → Data sent to Sanity CMS.

3. Tracking Shipments:

Real-time updates will be fetched via Shipment Tracking API.

4. Payment Processing:

Payment Gateway will handle payment securely → Sanity CMS will record payment info.

3. API Requirements

General Endpoints

- **Fetch Products:**

Endpoint: /products

Method: GET

Response: { "id": , "name": "Product ", "price": }

- **Create Order:**

Endpoint: /orders

Method: POST

Payload: { "customer": {}, "products": [], "paymentStatus": "paid" }

Response: { "orderId": , "status": "" }

- **Track Shipment:**

Endpoint: /shipment

Method: GET

Response: { "orderId": , "status": "", "ETA": "" }

---

4. Sanity Schema Example

- **Product Schema**

```
export default {
  name: '',
```

```
  type: '',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'image', type: 'image', title: 'Product Image' },
    { name: 'description', type: 'text', title: 'Description' }
  ]
};
```

- **Order Schema**

```
export default {
  name: '',
  type: '',
  fields: [
    { name: '', type: 'object', fields: [ /* customer fields */ ] },
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }]
},
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'orderStatus', type: 'string', title: 'Order Status' }
  ]
};
```

5. Roadmap and Flowcharts

1. Setup and Configuration:

Successfully Installed and configured Next.js.

Successful Setup Sanity CMS with necessary schemas.

Configuring APIs for payments and shipment tracking.

2. Frontend Development:

Built essential pages (Home, Product Listing, etc.).

Implemented responsive design.

Integrated Sanity API for dynamic product display.

3. Backend Integration:

Connected frontend to Sanity CMS.

Implement payment gateway and shipment tracking APIs.

4. Testing:

Tested API integrations, Tested UI responsiveness, and workflows.

5. Deployment:

Deployed to Vercel or similar platform.

Monitoring performance and user feedback.

6. Roadmap for User

User Opens Site
|
v
Homepage Displays Featured Products
|
v
User Navigates to Product Listing
|
API Request -> Fetch Products from Sanity CMS
|
v
Products Displayed on Frontend
|
v
User Selects Product -> Product Details Page
|
v
User Adds Product to Cart
|
v
User Proceeds to Checkout
|
API Request -> Create Order in Sanity CMS
|
v
Payment Gateway -> Process Payment
|
v

API Request -> Record Payment in Sanity CMS

|

v

Order Confirmation Displayed

|

API Request -> Fetch Shipment Status

|

v

Real-Time Shipment Tracking Displayed

This structured roadmap and technical planning will serve as a comprehensive guide for our marketplace project.