

DAY:2 Planning the technical foundation

1) Technical Foundation Schema for Marketplace

1. Frontend Requirements

A responsive and user-friendly interface enabling users to:

Frontend Details

Pages to Create:

1. Home Page

a. Features:

- i. Hero section with a welcome banner and featured products.
- ii. Navigation bar with links to key pages (e.g., Home, Products, About, Contact Us).

b. Buttons:

- i. "Shop Now" button for featured products.
- ii. "View Details" button for individual products.
- iii. Navigation buttons (e.g., Home, Products).

c. Code (Tailwind):

2. Product Listing Page

a. Features:

- i. Grid layout to display products with an image, name, and price & description.

b. Buttons:

- i. "Add to Cart" button.
- ii. "View Details" button.

3. Product Details Page

a. Features:

- i. Large product image, description, price...
- ii. Dropdown for selecting size or color (if applicable).

b. Buttons:

- i. "Add to Cart" button.
- ii. "Buy Now" button.

c. Libraries:

- i. Use @headlessui/react for dropdowns.
- ii. Use react-rating-stars-component for reviews.

4. Cart Page

a. Features:

- i. List of items added to the cart with quantity selectors.
- ii. Display total price and a "Proceed to Checkout" button.

b. Buttons:

- i. "Remove Item" button.
- ii. "Proceed to Checkout" button.

5. Checkout Page

a. Features:

- i. Form for entering shipping and payment details.
- ii. Display order summary.

b. Buttons:

- i. "Place Order" button.
- ii. "Edit Cart" button.

6. Order Confirmation Page

a. Features:

- i. Display order ID, summary, and estimated delivery date.

b. Buttons:

- i. "Back to Home" button.
- ii. "View Order Details" button.

2. Backend Requirements

- **Sanity CMS:**

- Manage product data.
- Store customer details.
- Maintain order records.

Order Schema:

```
export default {  
  orderId: "string",  
  userId: "string",  
  productId: "string",  
  quantity: "number",  
  totalPrice: "number",  
  orderDate: "string", // ISO date string or Date object  
  shippingAddress: "string",  
  status: "string", // e.g., "pending", "shipped", "delivered"  
}
```

Product Schema:

```
export default {  
  productId: "string",  
  name: "string",  
  type: "document", // Type could refer to the product category or format  
  price: "number",  
  imageUrl: "string",  
  description: "string",  
}
```

API Endpoints:

1. User Management Endpoints:

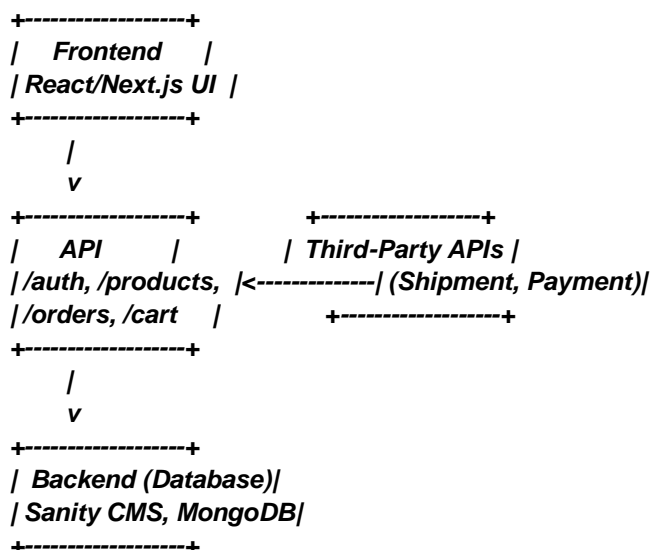
- **POST /api/auth/register:** Register a new user.
- **POST /api/auth/login:** Authenticate a user and return a token.
- **GET /api/user/:id:** Retrieve the user's profile based on user ID (e.g., to view their order history).

2. Product Management Endpoints:

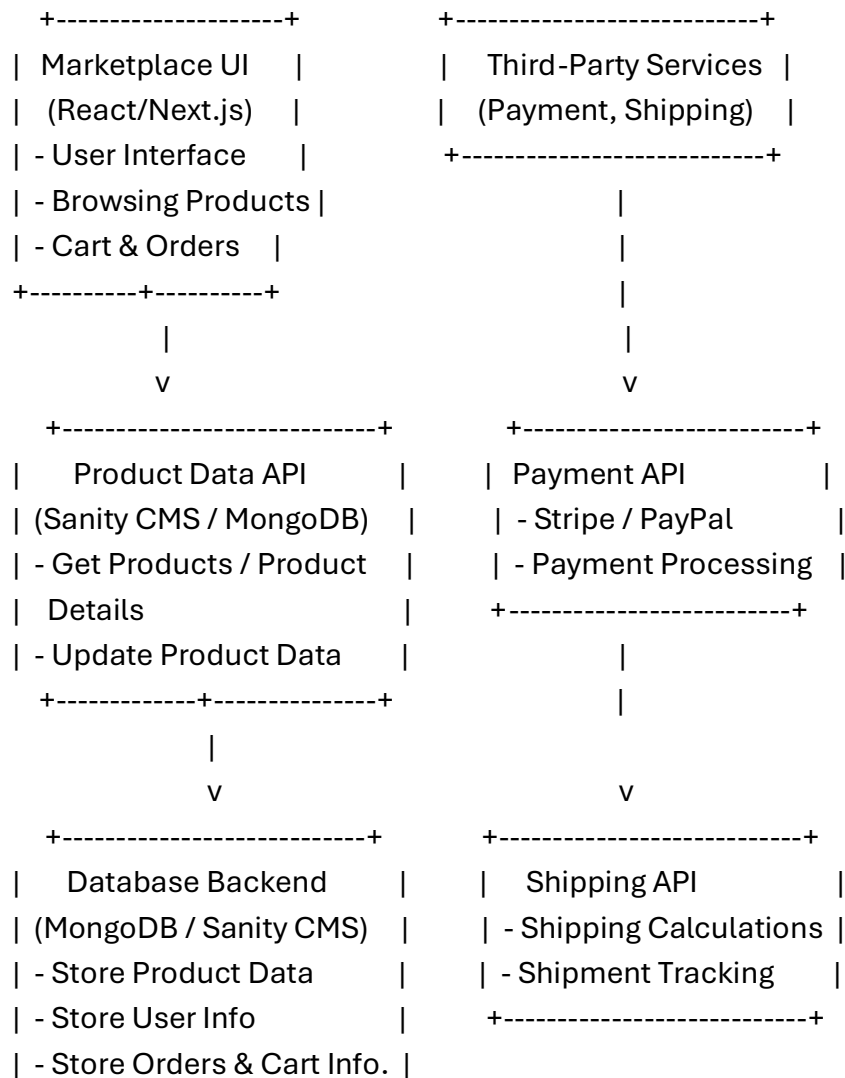
- **GET /api/products:** Fetch all products (can include filtering parameters).
- **GET /api/products/:id:** Fetch a single product's details by its ID.
- **POST /api/products:** Admin adds a new product to the marketplace
- **PUT /api/products/:id:** Admin updates an existing product's details (e.g., price, description).

3. Order Management Endpoints:

- **POST /api/orders:** Create a new order when a user proceeds to checkout.
- **GET /api/orders/:userId:** Retrieve all orders made by a specific user.
- **GET /api/cart/:userId:** Fetch the user's cart details.
- **PUT /api/cart/:userId:** Update the user's cart (e.g., add or remove products).



High-Level System Architecture Diagram:



MARKETPLACE NAME: “ThriftTreasures”

