

Technical Foundation Plan for Marketplace Builder

1. Features & Functionalities

Core Features

- User Registration and Authentication:- Allow user to sign up, log in.

and manage their profiles.

- Search and Listings:- Seller e

user can search for products using keywords and filter ~~description~~ by categories, price or ratings.

- Shopping Cart: users can add items to their cart for purchase.

- Checkout Process:- A streamlined process to review orders and confirm payments.

- Order Tracking:- Buyers can track the status of their orders.

2. Data Schema

users Table:

- ID: unique identifier

- Name: user's full name

- Email: user's email address

- Password: Encrypted password

- Role: Buyer or Seller.

Product Table :-

- ID : unique identifier
- Name : Product name
- Description : Product details
- Price : Product price
- Category : Product category
- Seller-ID : Reference the seller from the users Table

Order Table :-

- ID : unique identifier
- User ID : Reference the buyer from the users table
- Product IDs : List of product IDs
- Status : Current order status (e.g., pending, shipped, delivered)
- Total price : Total cost of the order.

Cart Table :-

- ID : unique identifier
- user-ID : Reference the buyer from the users Table
- Product-IDs : List of product IDs in the cart.

3 System Flow .

1) Homepage:

- Displays featured products and categories.
- Search bar for product discovery.

2) Product Details:

- Shows detailed information about a single product.
- Includes an option to add the product to the Cart.

3) Product Listing:

- Display a grid or list view of products with filters.

4) Cart and Checkout:-

- User review their cart and proceed to payment.

- User Backend calculates total cost and validates inventory.

5) Order Tracking:

- Updates the user on the status of their purchase.

6) Store and Manage Content:

Product, categories, and other data will be managed through Sanity's intuitive dashboard.

- Integration with Frontend: use Sanity's APIs to fetch and display data dynamically in the Next.js frontend.

7) Real-time Collaborations:-

- Enable team members to update content collaboratively in real-time.

