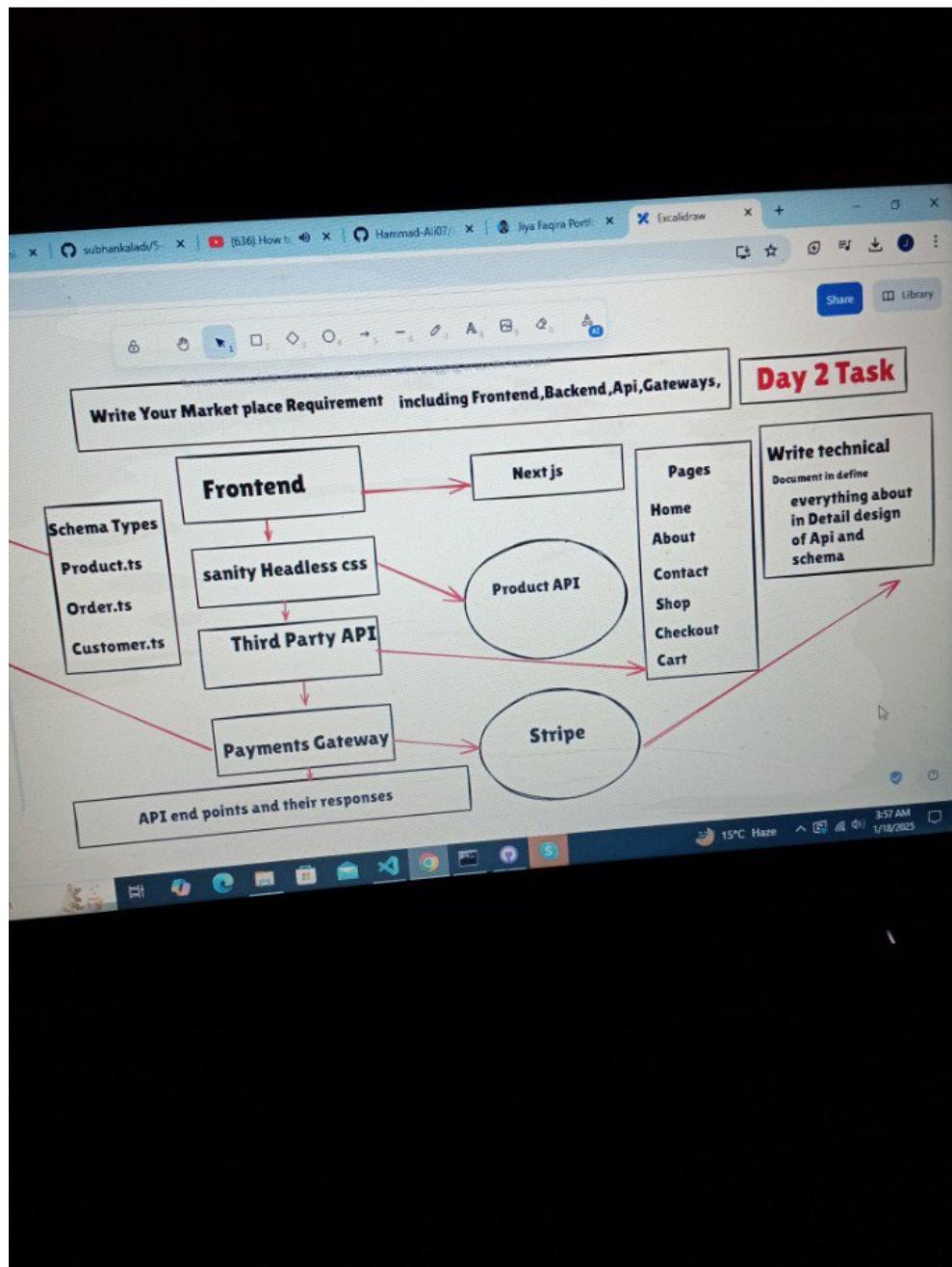




Jan 18, 2025

03:57 AM



Share



Google Lens



Edit



Delete



More



# DAY 2 TASK

## MARKET PLACE REQUIREMENTS

Create a market place involves multiple components, each with its own set of requirements for both frontend & backend development, as well as API needed for communication between them. Below is an outline of the typical requirement for building a market place:

### (1) FRONTEND REQUIREMENTS:

The frontend is what users interact with directly. It needs to be intuitive, responsive & engaging.

#### USER INTERFACE (UI):

- **Homepage:** Displays featured products, categories, search bar & user account options.
- **PRODUCT LISTINGS:** Grid or list of products with details such as images, descriptions, prices, reviews & ratings.
- **PRODUCT DETAILS PAGE:** Detailed view of a product with options to add to the cart, view seller info etc.

- **SEARCH & FILTERS:**  
details, & read product reviews  
Filters for categories, price range, ratings, brands, etc
- **SHOPPING CART:**  
Ability to view, add, update or remove products from the cart
- **CHECKOUT PROCESS:**  
A multi-step process where users input shipping information, payment details & review their order
- **USER PROFILE:**  
Section for users to view their order history, manage personal details & payment methods.
- **SELLER DASHBOARD:**  
A dashboard for sellers to manage their products, view sales reports, & communicate with buyers.
- **ADMIN DASHBOARD:**  
Admin panel to manage users, orders, payments, & product listings

ENT

details, & read product reviews

### SEARCH & FILTERS:

Search functionality with filters for categories, price range, ratings, brand, etc

### SHOPPING CART:

Ability to view, add, update

or remove products from the cart

### CHECKOUT PROCESS:

A multi-step process where users input shipping information, payment details & review their order

### USER PROFILE:

Section for users to view their order history, manage personal details & payment methods.

### SELLER DASHBOARD:-

A dashboard for sellers to manage their products, view sales reports, & communicate with buyers.

### ADMIN DASHBOARD:

Admin Panel to manage users, orders, payments, & product listings

## • USER REGISTRATION & LOGIN:

Authenticating via emails, social media or third-party services like Google or Facebook.

## • PAYMENT

Gateways  
Processing

## (2) BACKEND REQUIREMENTS:

The backend handles data management, user authentication, order processing & more. This part of the system needs to be robust, secure & scalable.

### DATABASE:

#### • PRODUCT DATABASE:

Stores product details such as name, description, price, inventory count, category, images & seller info.

#### • USER DATABASE:

Stores user details like username, email, password hash, purchase history & preferences.

#### • ORDER DATABASE:

Maintain order details including order status, payment info, shipping details and timestamps.

#### • REVIEWS & RATINGS:

A system to store & manage user reviews & ratings for products.

(3)

### PAYMENT INTEGRATION:

Gateways like PayPal, Stripe or credit card processing services. Integration with payment services.

### AUTHENTICATION & AUTHORIZATION:

#### USER AUTHENTICATION:

Manage user login, registration & session management (using JWT tokens or session IDs)

#### ROLE-BASED ACCESS CONTROL:

Implement roles for admins, buyers & sellers to restrict access to certain features

### (3) API REQUIREMENTS:-

APIs (Application Programming interfaces) are required for communication between the frontend & backend as well as integration with third-party services.

#### AUTHENTICATION APIs:

#### LOGIN/LOGOUT:

API endpoints to allow users to log in, log out & retrieve session information

= **SOCIAL LOGIN:** API endpoints for integrating third-party login systems like Google or Facebook, allowing users to log in.

### PRODUCT APIs:

**GET PRODUCTS:** fetch a list of products with filters (e.g. category, price range)

**GET PRODUCT DETAILS:** fetch detailed information about a specific product

**CREATE/UPDATE/DELETE PRODUCT:**

to add, edit or remove products.

### ORDER & CART APIs:

#### ADD TO CART:

API to allow users to add products to their shopping cart

#### VIEW CART:

Fetch the contents of the user's cart

#### PLACE ORDER:

API to place an order once

**ORDER STATUS:**

order status

**PAYMENT:**

**PAYMENT:**

provides confirmation

**REFUND:**

needed

**REFUND:**

**REFUND:**

**REFUND:**

**REFUND:**

the user proceeds to checkout.

- ORDER STATUS:

API to get or update the order status (Payment Success, Shipped)

- PAYMENT APIs:

- PAYMENT PROCESSING:

Integration with payment providers like PayPal or Stripe to initiate & confirm payments

- REFUNDS:

API for processing refunds when needed

## REVIEW & RATING APIs:

- SUBMIT REVIEW:

API to allow users to submit reviews & ratings for products

- GET REVIEW:

Fetch reviews & rating for a specific product

- UPDATE PROFILE:

API for users to update their personal information or shipping address

- SHIPPING
- CATEGORIES

## 2. DATA BASE SCHEMA DESIGN

### USERS TABLE:

This table stores information about users (buyers, sellers, & admins). It tracks user credentials & roles.

Create Table users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(255),  
 email VARCHAR(255) UNIQUE NOT NULL,  
 password\_hash VARCHAR(255),  
 role ENUM ('buyer', 'Seller', 'admin') DEFAULT 'buyer',  
 phone\_number VARCHAR(15),  
 address TEXT,

Created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 Updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 ON UPDATE CURRENT\_TIMESTAMP);

- **USER\_ID:** Primary key, auto-incremented.
- **ROLE:** Determines if the user is buyer, seller, or admin.
- **EMAIL & PASSWORD\_HASH:** For authentication.