Building the Technical Foundation of -Furnishly Marketplace

Defining Business Goals, Database Schemas, Technical Requirements, API Endpoints, and Clean Flowcharts

(Furnishly - By Wania A.)

1. Business Goals

- **Increase Market Share:** Capture a significant portion of the online market in the target niche.
- **Customer Satisfaction:** Provide a seamless and enjoyable shopping experience, leading to high customer retention.
- **Operational Efficiency:** Streamline order processing, inventory management, and customer support to minimize costs and maximize efficiency.
- **Revenue Growth:** Drive consistent revenue growth through increased sales volume and higher average order value.
- **Brand Building:** Establish a strong brand identity and build trust with customers.

2. Database Schema

- **Users:**
    - user_id (Primary Key)
    - username
    - email
    - password_hash
    - first_name
    - last_name
    - phone_number
    - address
    - role (e.g., customer, seller, admin)
- **Products:**
    - product_id (Primary Key)
    - name
    - description
    - price

    - category
    - seller_id (Foreign Key referencing Users)
    - image_url
    - stock_quantity
- **Orders:**
    - order_id (Primary Key)
    - customer_id (Foreign Key referencing Users)
    - order_date
    - total_amount
    - status (e.g., pending, processing, shipped, delivered, canceled)
- **Order_Items:**
    - order_item_id (Primary Key)
    - order_id (Foreign Key referencing Orders)
    - product_id (Foreign Key referencing Products)
    - quantity
    -
    - unit_price
- **Reviews:**
    - review_id (Primary Key)
    - customer_id (Foreign Key referencing Users)
    - product_id (Foreign Key referencing Products)
    - rating
    - comment
- **Carts:**
    - cart_id (Primary Key)
    - customer_id (Foreign Key referencing Users)
- **Cart_Items:**
    - cart_item_id (Primary Key)
    - cart_id (Foreign Key referencing Carts)
    - product_id (Foreign Key referencing Products)
    - quantity

3. Technical Requirements

- **Frontend:**
    - **Framework:** React, Angular, or Vue.js
    - **Features:** Product browsing, search, filtering, product pages, shopping cart, checkout, order tracking, user accounts, reviews, etc.
- **Backend:**
    - **Framework:** Node.js with Express, Python with Django/Flask, or Ruby on Rails
    - **Database:** PostgreSQL, MySQL, or MongoDB
    - **API Development:** RESTful APIs for product management, order processing, user authentication, payment integration, etc.
- **Cloud Hosting:** AWS, Azure, or Google Cloud Platform

- **Payment Gateway Integration:** Stripe, PayPal, or local payment providers
- **Search Functionality:** Elasticsearch or Algolia for fast and relevant search results
- **Security:** Robust authentication and authorization, data encryption, and protection against common vulnerabilities (e.g., SQL injection, cross-site scripting)
- **Scalability:** The system should be able to handle increasing traffic and data volumes as the business grows.

4. API Endpoints

- **Product APIs:**
  - `GET /products`: Retrieve all products
  - `GET /products/{product_id}`: Retrieve a specific product
  - `POST /products`: Create a new product
  - `PUT /products/{product_id}`: Update a product
  - `DELETE /products/{product_id}`: Delete a product
- **User APIs:**
  - `POST /users`: Create a new user (registration)
  - `POST /users/login`: User login
  - `GET /users/profile`: Retrieve user profile
  - `PUT /users/profile`: Update user profile
- **Order APIs:**
  - `POST /orders`: Create a new order
  - `GET /orders/{order_id}`: Retrieve order details
  - `GET /orders/user`: Retrieve orders for a specific user
- **Cart APIs:**
  - `GET /carts`: Retrieve user's cart
  - `POST /carts/add`: Add a product to the cart
  - `PUT /carts/update`: Update product quantity in the cart
  - `DELETE /carts/remove`: Remove a product from the cart
- **Review APIs:**
  - `POST /reviews`: Create a new product review
  - `GET /reviews/{product_id}`: Retrieve reviews for a specific product

5. Flowcharts

- **Order Placement Flowchart:**

  1. Customer browses products.
  2. Customer adds products to the cart.
  3. Customer proceeds to checkout.
  4. Customer enters shipping and billing information.

5. Customer selects payment method.
6. Payment is processed.
7. Order is created and confirmed.
8. Order is processed and shipped.
9. Customer receives order.

- **Product Search Flowchart:**

1. User enters search query.
2. Search query is processed by the search engine.
3. Relevant products are retrieved from the database.
4. Search results are displayed to the user.

Prepared By Wania A.

00382982