STAR LION COLLEGE OF ENGINEERING AND TECHNOLOGY

Name	A.Sridhar
Department	Computer Science And Engineering
Project title	E-commerce Application

E-commerce Application on IBM Cloud Foundry

Project Title: E-commerce App

Problem Definition:

The project is to build an artisanal e-commerce platform using IBM Cloud Foundry. The goal is to connect skilled artisans with a global audience, showcasing their handmade products and providing features like secure shopping carts, payment gateways, and an intuitive checkout process. This involves designing the e-commerce platform, implementing necessary features, and ensuring a seamless user experience.

Design Thinking:

1. Platform Design:

Creating a well-structured and intuitive layout for an e-commerce app is essential to enhance user experience and encourage seamless navigation. Here's a layout design that includes sections for product categories, individual product pages, shopping cart, checkout, and payment:

1. Home Screen
2. Product Categories
3. Individual Product Page
4. Shopping Cart
5. Checkout
6. Payment

Ensure that the design is responsive to provide a seamless experience on various devices, and prioritize user-friendly navigation and a visually appealing interface. Also, consider integrating features like saved addresses, order tracking, and customer support to enhance the overall user experience.

2. Product Showcase:

Creating a database schema for an e-commerce app involves defining the structure to store product information efficiently. We'll design a relational database schema using tables to store product details, categories, and images. Let's outline the structure:

Tables:

1. Product Table
2. Category Table
3. Image Table

Relationships:

- Each product belongs to one category, establishing a many-to-one relationship between `Product` and `Category`.
- Each product can have multiple images, forming a one-to-many relationship between `Product` and `Image`.

Database Schema (in SQL-like syntax):

SQL

```
CREATE TABLE Category (
    category_id INT PRIMARY KEY,
    name VARCHAR(255)
);
CREATE TABLE Product (
    product_id INT PRIMARY KEY,
    name VARCHAR(255),
    description TEXT,
    price DECIMAL(10, 2),
    category_id INT,
    created_at TIMESTAMP,
    FOREIGN KEY (category_id) REFERENCES Category(category_id)
);
CREATE TABLE Image (
    image_id INT PRIMARY KEY,
    product_id INT,
    url VARCHAR(255),
    is_primary BOOLEAN,
    FOREIGN KEY (product_id) REFERENCES Product(product_id)
);
```

In this schema, each product has a unique ID (`product_id`), a name, a description, a price, a reference to its category (`category_id`), and a timestamp indicating when it was created. Images related to a product are stored in the `Image` table with a reference to the corresponding product.

Ensure that you use appropriate data types and constraints based on your specific database system (e.g., MySQL, PostgreSQL) and any additional requirements for your e-commerce app.

3. User Authentication:

To implement user registration and authentication features for artisans and customers in an e-commerce app, you'll need to create functionality for users to create accounts, login, and handle authentication. Below, I'll outline the key components and steps to achieve this:

1. User Table:

Create a table to store user information, including artisans and customers.

SQL

```
CREATE TABLE User (
    user_id INT PRIMARY KEY,
    username VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    full_name VARCHAR(255),
    is_artisan BOOLEAN,
    created_at TIMESTAMP
);
```

2. User Registration:

When a user (artisan or customer) wants to register on the platform, they will provide necessary details such as username, password, email, full name, and specify if they are an artisan or customer.

SQL

```sql
INSERT INTO User (username, password, email, full_name, is_artisan, created_at)

VALUES ('username123', 'hashed_password', 'user@example.com', 'John Doe', TRUE, CURRENT_TIMESTAMP);
```

3. User Authentication:

To authenticate users during login, you'll need to validate their credentials against the database.

SQL

```sql
SELECT user_id, username, is_artisan

FROM User

WHERE username = 'username123' AND password = 'hashed_password';
```

4. Session Management.

5. User Profile:

Allow users to update their profile information, change passwords, and other relevant details.

SQL

```sql
UPDATE User

SET full_name = 'New Name', email = 'newemail@example.com'

WHERE user_id = 1;
```

6. Password Encryption.

7. Forgot Password.

8. Token-Based Authentication.

9. Access Control.

These steps outline the foundational elements of implementing user registration and authentication for both artisans and customers in an e-commerce app. Depending on your specific requirements and the technology stack you're using, the actual implementation details may vary. Always prioritize security and follow best practices to ensure a robust and secure authentication system.

4. Shopping Cart and Checkout:

Designing and developing the shopping cart functionality and a smooth checkout process is critical to providing a seamless user experience in an e-commerce app. Let's break down the steps to implement these features:

Shopping Cart Functionality:

1. **Add to Cart**
2. **View Cart**
3. **Update Cart**
4. **Calculate Total**
5. **Empty Cart**

Smooth Checkout Process:

1. **Proceed to Checkout**
2. **Login/Registration**
3. **Shipping Details**
4. **Order Summary**
5. **Payment Information**
6. **Order Confirmation**
7. **Email Confirmation**

Development Considerations:

1. **Database Updates**
2. **Integration with Payment Gateway**
3. **Session Management**
4. **Error Handling**
5. **Confirmation Page**
6. **Inventory Management**

By following these steps and considerations, you can design and develop a shopping cart functionality and a smooth checkout process that enhances the user experience and encourages successful purchases in your e-commerce app.

5.Payment Integration:

Integrating secure payment gateways is a crucial step in facilitating transactions in an e-commerce app. Payment gateways ensure secure and smooth processing of payments. Here's a general guide on how to integrate secure payment gateways into your e-commerce app:

1. **Choose a Payment Gateway.**
2. **Create Accounts and Obtain API Keys.**
3. **Set Up Server-Side Integration.**

4. Implement Client-Side Integration.

5. Implement Checkout Page.

6. Send Payment Request to Payment Gateway:

7. Handle Payment Response.

8. Update Order and Notify User.

9. Error Handling.

10. Compliance and Security.

Follow the documentation and guidelines provided by the specific payment gateway for detailed integration steps and best practices. Always prioritize security and user experience in your integration.

6.User Experience:

Creating an intuitive and visually appealing user experience is crucial for engaging both artisans and customers in your e-commerce app. Here's a step-by-step guide to achieve this goal:

1. Understand Your Users.

2. User-Friendly Interface.

3. Responsive Design.

4. Visual Hierarchy.

5. Personalization.

6. Feedback and Notifications.

7. Accessibility.

8. Gather User Feedback.

By integrating these design principles and considering the needs of both artisans and customers, you can create an appealing and intuitive e-commerce app that enhances user engagement and satisfaction.