Project: Simulated E-Wallet and Rewards System for E-Commerce

Project Overview

This project simulates an **E-Wallet system** integrated with a **Rewards Program** for an e-commerce platform. Users can manage their wallet balance, perform simulated transactions, and earn rewards for purchases, and wallet top-ups. These points can be redeemed for discounts, cashback, or special offers.

Since this is an **academic project**, transactions will be simulated without integrating real-world payment gateways like Razorpay or PayPal. The project focuses on **database design, transaction management, schema creation, business logic**.

Roles and Responsibilities

1. Users (Customers)

- Create and manage an account.
- Add funds to their wallet using simulated transactions.
- Make purchases using the wallet balance.
- Earn and redeem reward points for cashback or discounts.
- View available balance.

2. Admin

- Manage users (view/edit user details, block/unblock accounts).
- Monitor and approve transactions (top-ups, purchases, refunds).
- Set and modify reward policies (points allocation, redemption rules).
- View, analyze, and approve/reject refunds.

3. Merchants (Sellers)

- Register as vendors on the platform.
- List products/services available for purchase.
- View order details and payment history.
- Receive payments from users' wallets (simulated).

4. Customer Support

- Resolve user queries related to failed transactions, refunds, and rewards.
- Assist users in redeeming points.
- Process transaction disputes and escalate issues if needed.

Types of Transactions (Money Transactions)

The system supports multiple **types of simulated financial transactions**:

1. Wallet Top-Up (Simulated)

- Users can add money to their wallet using a simulated payment method.
- Wallet balance is updated after a successful top-up.
- Users earn bonus points based on the amount topped up.
- **Example:** ₹500 top-up → **Earn 5 points** (1 point per ₹100).

2. Purchase Transaction

- Users buy products/services using wallet balance.
- Amount is deducted from the wallet, and reward points are granted.
- Merchants receive payment in their seller account (simulated).
- **Example:** ₹1,000 spent → **Earn 50 points (5 per ₹100 spent).**

3. Rewards Redemption

- Users can redeem reward points for discounts or cashback.
- If cashback is selected, the equivalent amount is credited to the wallet.
- **Example:** 500 points = ₹50 cashback **or** 10% discount on the next purchase.

4. Merchant Withdrawals (Simulated)

- Merchants can request withdrawal of funds from their seller accounts.
- Admin approves/rejects withdrawal requests.
- If approved, the seller account balance is reset.
- **Example:** Merchant requests withdrawal of ₹5,000 → Admin processes the request.

5. Refunds & Reversals

- Failed transactions or canceled orders trigger an automatic refund.
- Reward points earned on refunded purchases are **reversed**.
- **Example:** A ₹1,000 refund deducts 50 points earned from the original transaction.

DBMS Functionalities Used

1. Database Design & Normalization

- Multiple schemas will store structured data for **users, transactions, rewards, merchants, and logs**.
- Normalization ensures **data consistency and avoids redundancy**.

2. Stored Procedures & Triggers

- **Stored Procedures** automate **reward points calculation, refunds, and cashback processing**.
- **Triggers** automatically:
    - Grant reward points for transactions.
    - Update wallet balances.
    - Log transaction status changes.
- **Example:** A trigger automatically credits cashback when a user redeems points.

3. Referential Integrity & Foreign Keys

- **Foreign keys** enforce relationships between **users, transactions, and rewards**.
- Ensures that transactions are always linked to a valid user/merchant.

4. Indexing for Performance Optimization

- **Indexes** on **transaction history, wallet balances, and user records** will improve query speed.

5. Backup & Recovery

- The database will have scheduled **automatic backups**.
- Transactions are **logged to ensure recovery** in case of system failure.

Money Flow in the E-Wallet System

1. Wallet Top-Up (Simulated)

- **Where money comes from:** The user initiates a **simulated top-up**, adding funds to their wallet.
- **Where money goes:** The **wallet balance of the user** is increased.
- **Example:** If a user tops up ₹1,000, their **wallet balance increases by ₹1,000**.

2. Purchase Transaction (User Buys a Product)

- **Where money comes from:** The **user's wallet balance**.
- **Where money goes:** The **merchant's account** (simulated).
- **Example:** A user purchases a **₹500 item** → ₹500 is **deducted from the user's wallet** and **credited to the merchant's account**.

3. Merchant Withdraws Funds

- **Where money comes from:** The **merchant's account balance** (simulated earnings from user purchases).
- **Where money goes:** The merchant requests a **withdrawal**, and an admin **approves or processes** the request.
- **Example:** Merchant **withdraws ₹5,000** → ₹5,000 is **removed from the merchant's account** (simulated transaction).

4. Rewards Redemption (Points → Cashback)

- **Where money comes from:** The **platform's reward balance** (simulated).
- **Where money goes:** The **user's wallet** (credited cashback).
- **Example:** A user redeems **500 points for ₹50 cashback** → ₹50 is **credited to the user's wallet**.

5. Refunds & Reversals

- **Where money comes from:** The **merchant’s account balance** (if order is canceled or returned).
- **Where money goes:** The **user’s wallet balance** (credited refund).
- **Example:** A user requests a **refund for a ₹1,000 product** → ₹1,000 is **deducted from the merchant's balance** and **credited back to the user's wallet**.
- If the **reward points were granted for the purchase**, they are **automatically deducted** when the refund is processed.