Comsats University Islamabad

Semester End Project Proposal

Database Systems

Submitted to

Ms. Nimra Iqram

Submitted by

Muhammad Ahmad

(FA23-BCE-113)

Computer Engineering

Introduction

This proposal outlines the development of a next-generation e-commerce platform, E-Commerce, utilizing a modern technology stack: Angular for the frontend, Node.js/Express for the backend, and Microsoft SQL Server for the database. The project incorporates advanced features to ensure security, scalability, automation, and personalized user experience, positioning it as an enterprise-grade solution suitable for real-world deployment.

Objective

The primary objectives are to:

- Deliver a secure e-commerce system supporting multiple roles (buyers, vendors, managers).
- Automate business processes and maintain data integrity across all layers.
- Provide real-time analytics and personalized dashboards for users.
- Enable rapid future expansion and integration with external services.

Solution

The solution involves a robust database schema and advanced SQL Server features to support the E-Commerce platform, comprising 13 normalized tables with primary keys (PK), foreign keys (FK), and constraints. Key tables include:

- **USER**: (id PK, email_address UNIQUE, password_hash, is_active).
- **ROLE**: (id PK, name UNIQUE, CHECK 'buyer', 'vendor', 'manager').
- **USER_ROLE**: (user_id PK FK → USER, role_id PK FK → ROLE).
- **BUYER**: (id PK, user_id UNIQUE FK → USER, preferences CHECK 'male', 'female', 'children').
- **VENDOR**: (id PK, user_id UNIQUE FK → USER, vendor_name UNIQUE).
- **PRODUCT**: (id PK, vendor_id FK → VENDOR, price CHECK ≥ 0, stock_quantity CHECK ≥ 0).
- **CART**: (id PK, user_id FK → USER, product_id FK → PRODUCT, quantity CHECK > 0).
- **SHOP_ORDER**: (id PK, user_id FK → USER, total_amount CHECK ≥ 0, status CHECK 'Pending', 'Delivered', 'Rejected').
- **ORDER_ITEMS**: (id PK, order_id FK → SHOP_ORDER, product_id FK → PRODUCT, quantity CHECK > 0).
- **PRODUCT_REVIEW**: (id PK, buyer_id FK → BUYER, product_id FK → PRODUCT, rating CHECK 1-5).

Relationships: 1:N (e.g., USER to CART, PRODUCT to ORDER_ITEMS) and 1:1 (e.g., USER to BUYER).
Constraints: ON DELETE CASCADE (e.g., user_id in CART), ON DELETE NO ACTION (e.g., product_id in ORDER_ITEMS).

Advanced Features

1. Automated Business Logic with Advanced SQL Server Features

Stored Procedures

- **Atomic Multi-Step Transactions**: All critical operations such as checkout, user registration, and product management will be encapsulated within stored procedures to ensure atomicity and secure business logic execution—minimizing data inconsistencies.
- **Dynamic Stock & Order Management**: Inventory validation, stock deduction, and order processing will be handled automatically by procedures, eliminating manual intervention and reducing error rates.
- **Role-Aware Registration**: Upon user sign-up, procedures will handle role assignment and initial profile setup in a single secure transaction.

Triggers

- **Automated Data Consistency**: Database triggers will enforce rules like default address assignment, cart cleanup, and review eligibility without relying solely on application logic.
- **Self-Healing Mechanisms**: Triggers will detect and resolve constraint violations, ensuring data consistency and seamless user experience in edge cases.

Views

- **Personalized Dashboards**: Role-specific dashboards for buyers and vendors will be powered by SQL views, which will serve as a bridge between the business logic and the UI. These views will provide real-time access to relevant data, minimizing backend processing. By leveraging the separation of concerns, the dashboard's data will be processed and served efficiently, allowing role-specific customization with minimal load on the backend.
- **Advanced Analytics**: Views will offer pre-aggregated insights into sales trends, order statuses, and product performance.
- **Optimized Reporting**: Complex business queries will be encapsulated into reusable, performance-tuned views for frontend consumption.

2. Security & Compliance

- **Authentication Security**: JWT-based authentication will be implemented to securely manage user credentials and session handling, ensuring stateless, scalable, and tamper-proof access control.
- **Granular Role Management**: Multi-role architecture with strict access control will be enforced both at the API and database layers for enhanced security.
- **Data Encryption**: Sensitive fields like password_hash are encrypted using SQL Server's built-in encryption functions, protecting user credentials against breaches.

3. Performance Optimization

- **Indexing**: Clustered indexes on PRODUCT.id and non-clustered indexes on SHOP_ORDER.order_date optimize high-traffic queries, reducing response times for order retrieval and inventory checks.
- **Query Execution Plans**: Analyzed and optimized using SQL Server Management Studio to ensure efficient execution of stored procedures and view queries.

Additional Enhancements

- **Personalization**: AI-ready schema supports future recommendation engines, with real-time updates via Angular.
- **Extensibility**: Modular design allows integration of promotions, payment gateways, and cloud deployment.
- **Cloud-Ready Deployment**: The system will be structured for easy deployment to cloud platforms with support for horizontal scalability and high availability.

Deliverables

- Fully implemented E-Commerce database schema with 13 tables, constraints, and advanced features.
- Functional stored procedures, triggers and views.
- Secure RESTful APIs with role-based access and JWT authentication.
- Responsive Angular frontend with personalized dashboards.
- Comprehensive documentation of the schema, implementation, and testing process.

4-Week Development Plan

- **Week 1**: Database Design and Setup
- **Week 2**: Advanced Features Implementation
- **Week 3**: Backend and API Development
- **Week 4**: Frontend Development and Testing