# ShopEase – E-Commerce Application

## 1. Project Title

**ShopEase – E-Commerce Application**
A full-stack e-commerce backend system designed to manage products, users, carts, and orders with a scalable and real-world architecture.

---

## 2. Project Objective

The goal of ShopEase is to build a secure, scalable, and production-ready e-commerce platform that supports multiple user roles and complete shopping workflows.

### User Roles:

- **Customers:** Browse products, manage cart, place orders, manage addresses
- **Administrators:** Manage products, categories, users, and orders

### Key Objectives:

- Provide secure authentication and role-based access
- Support complete shopping flow from product browsing to order placement
- Maintain data integrity using relational database design
- Build a modular backend suitable for frontend (React) integration
- Enable easy future enhancements like payments and notifications

---

## 3. Project Scope

### Backend Responsibilities:

- User registration and authentication
- Role-based access control (Admin / User)
- Product and category management
- Cart and order processing
- Address management
- REST API development
- Database interaction and transaction handling

### Frontend Responsibilities (Planned):

- User-friendly shopping interface
- Product listing and category filtering
- Cart and checkout flow

• Order history and profile management
• Admin dashboards for product and category management

---

## 4. Full-Stack Architecture

### 1. Frontend (React.js – Planned):

• Components: Login, Register, Product List, Product Details, Cart, Orders, Admin Panel
• State Management: Context API / Redux
• Routing: React Router
• API Integration: Axios

### 2. Backend (Spring Boot):

• Controller Layer: REST APIs
• Service Layer: Business logic
• Repository Layer: JPA / Hibernate
• Security Layer: Spring Security with JWT

### 3. Database (MySQL):

• Normalized schema with relational mappings
• Transaction management
• Indexing for performance

### 4. Communication:

• REST APIs using JSON
• Secure JWT-based authentication

---

## 5. Technology Stack

| Layer | Technology |
|---|---|
| Frontend | React.js (Planned) |
| Backend | Java, Spring Boot |
| Security | Spring Security, JWT |
| Database | MySQL |
| ORM | JPA / Hibernate |
| Build Tool | Maven |
| Documentation | Swagger |

| Layer | Technology |
| --- | --- |
| Testing | Postman |

---

## 6. User Roles & Access Control

### Roles:

- **Admin:** Full control over products, categories, and users
- **User:** Browse products, manage cart, place orders

### Access Control:

- Role-based API authorization
- JWT token validation for secured endpoints
- Protected admin APIs

---

## 7. Functional Modules

### 7.1 Authentication & Account Management

- User registration and login
- Encrypted password storage
- JWT token generation and validation
- Role-based access control

### 7.2 User Management (Admin)

- View all users
- Assign roles
- Activate / deactivate users

### 7.3 Product Management

- Add, update, delete products (Admin)
- View all products (User)
- Category-based product browsing

### 7.4 Category Management

- Create and manage categories
- Assign products to categories

### 7.5 Cart Management

- One cart per user
- Add products to cart
- Update quantities
- Remove items
- View cart

### 7.6 Order Management

- Place orders
- Order status tracking
- View order history
- Order-item mapping

### 7.7 Address Management

- Add multiple addresses
- Update address details
- Select address during checkout

---

## 8. Database Design

### Tables:

- Users
- Products
- Categories
- Cart
- CartItems
- Orders
- OrderItems
- Addresses

### Relationships:

- One User → One Cart
- One User → Many Orders
- One User → Many Addresses
- One Category → Many Products
- One Cart → Many CartItems
- One Order → Many OrderItems

---

## 9. Backend Package Structure

```
com.app
 ├── controller
 │    ├── AuthController.java
 │    ├── ProductController.java
 │    ├── CategoryController.java
 │    ├── CartController.java
 │    ├── OrderController.java
 │    └── UserController.java
 ├── service
 ├── repository
 ├── entity
 ├── security
 ├── config
 ├── payload
 └── ShopEaseApplication.java
```

---

## 10. Security Considerations

- JWT-based authentication
- Encrypted password storage
- Role-based endpoint protection
- Input validation to prevent SQL Injection and XSS

---

## 11. REST API Overview

### Authentication APIs:

- POST `/api/auth/register`
- POST `/api/auth/login`

### Product APIs:

- POST `/api/admin/products`
- GET `/api/products`
- GET `/api/products/{id}`

### Cart APIs:

- POST `/api/cart/add`
- GET `/api/cart`

- DELETE `/api/cart/remove`

**Order APIs:**

- POST `/api/orders`
- GET `/api/orders`

---

## 12. Testing & Quality Assurance

**Backend Testing:**

- Unit testing with JUnit
- API testing using Postman

**Documentation:**

- Swagger UI for API testing and documentation

---

## 13. Future Enhancements

- Payment gateway integration (Razorpay / Stripe)
- Product reviews and ratings
- Wishlist functionality
- Search and pagination
- Email notifications
- Admin analytics dashboard

---

## 14. Conclusion

ShopEase is a robust and scalable e-commerce backend system inspired by real-world applications. With secure authentication, strong database relationships, and modular architecture, the platform is well-suited for enterprise-level expansion and frontend integration.