**Detailed Design Document (LLD) – E-Commerce Application**

**Version:** 1.0  
**Prepared By:** [Insert Name]  
**Date:** [Insert Date]

**1️⃣ Document Purpose**

This document provides a detailed technical description of the components, data structures, API designs, and workflows required for developing the E-Commerce Application.

**2️⃣ System Overview**

* A full-stack e-commerce solution including customer portal, admin panel, and APIs for handling user management, product catalog, orders, and payments.

**3️⃣ Module-Wise Detailed Design**

**3.1 User Management Module**

**Components:**

* User Registration API
* Login API (with JWT token generation)
* Password Reset API
* Profile Update API

**Database Tables:**

* users (user\_id, name, email, password\_hash, phone, address, created\_at)

**APIs:**

| **API Name** | **Method** | **Endpoint** | **Description** |
| --- | --- | --- | --- |
| Register User | POST | /api/users/register | Registers a new user |
| Login | POST | /api/users/login | Authenticates and returns token |
| Update Profile | PUT | /api/users/profile | Updates user details |

**Security:**

* Passwords stored using bcrypt
* JWT token authentication

**3.2 Product Catalog Module**

**Components:**

* Product Listing API
* Product Details API
* Search, Filter, and Sort APIs

**Database Tables:**

* products (product\_id, name, description, price, category\_id, stock, image\_url)

**APIs:**

| **API Name** | **Method** | **Endpoint** | **Description** |
| --- | --- | --- | --- |
| List Products | GET | /api/products | Returns paginated product list |
| Product Details | GET | /api/products/{id} | Returns product details |
| Search Products | GET | /api/products/search | Search using keyword |
| Filter by Category | GET | /api/products/category/{id} | Filter products by category |

**3.3 Shopping Cart Module**

**Components:**

* Add to Cart API
* View Cart API
* Update/Delete Items API

**Database Tables:**

* cart (cart\_id, user\_id, product\_id, quantity)

**APIs:**

| **API Name** | **Method** | **Endpoint** | **Description** |
| --- | --- | --- | --- |
| Add to Cart | POST | /api/cart | Adds item to user’s cart |
| View Cart | GET | /api/cart | Displays current cart items |
| Update Cart Item | PUT | /api/cart/{item\_id} | Updates quantity |
| Remove Cart Item | DELETE | /api/cart/{item\_id} | Removes item from cart |

**3.4 Checkout & Payment Module**

**Components:**

* Checkout API
* Payment Gateway Integration
* Order Confirmation Notification

**Database Tables:**

* orders (order\_id, user\_id, total\_amount, payment\_status, order\_status, created\_at)
* order\_items (order\_item\_id, order\_id, product\_id, quantity, price)

**APIs:**

| **API Name** | **Method** | **Endpoint** | **Description** |
| --- | --- | --- | --- |
| Checkout | POST | /api/orders/checkout | Places an order |
| Payment Callback | POST | /api/payment/callback | Handles gateway response |
| Order Summary | GET | /api/orders/{id} | Returns order and payment details |

**Payment Integration:**

* Razorpay / PayU REST APIs
* Handle success/failure responses

**3.5 Order Management Module**

**Components:**

* Order Listing API
* Order Tracking API
* Admin Order Management

**APIs:**

| **API Name** | **Method** | **Endpoint** | **Description** |
| --- | --- | --- | --- |
| List Orders | GET | /api/orders | Lists orders for logged-in user |
| Order Tracking | GET | /api/orders/track/{id} | Shows order status |
| Admin Order Mgmt | Admin Panel |  | Update/cancel/ship orders |

**3.6 Notification Module**

* **Email Notifications** (using SMTP/Email API)
* **SMS Notifications** (using Twilio / AWS SNS)

**Triggers:**

* Order confirmation
* Payment success/failure
* Shipping/delivery updates

**3.7 Admin Panel Module**

* Dashboard view (Sales, Orders, Customers)
* Product CRUD (Create, Read, Update, Delete)
* Inventory reports
* Order management
* Role-based access control

**4️⃣ Sequence Diagrams**

Include sequence diagrams for:

* User Registration
* Checkout & Payment
* Order Processing Flow

**5️⃣ Database ER Diagram**

Include full ER diagram showing:

* Users
* Products
* Cart
* Orders
* Order Items
* Inventory
* Categories

**6️⃣ Security Design**

* JWT authentication for APIs
* HTTPS encryption (TLS 1.2+)
* Input validation & sanitization
* Role-based access control in admin modules
* PCI-DSS compliance for payments

**7️⃣ API Gateway Routing**

* All APIs routed via /api endpoint
* Microservices split (optional in Phase 2)
* Rate limiting and throttling at gateway

**8️⃣ Error Handling Design**

| **Error Type** | **HTTP Status Code** | **Message Format** |
| --- | --- | --- |
| Unauthorized Access | 401 | { "error": "Unauthorized" } |
| Invalid Request | 400 | { "error": "Bad Request" } |
| Server Error | 500 | { "error": "Internal Server Error" } |

**9️⃣ Logging and Monitoring**

* Logs stored in centralized service (CloudWatch / ELK Stack)
* Track API failures, payment errors, and security events
* Daily log rotation and archival

**🔟 Deployment Overview**

* Containerized APIs using Docker
* CI/CD pipelines using Jenkins / GitLab CI
* Hosting on AWS / Azure cloud infrastructure
* Multi-zone high availability setup

**1️⃣1️⃣ Future Enhancements**

* Progressive Web App (PWA)
* Recommendation engine using AI/ML
* Multi-vendor support
* Real-time inventory sync with ERP

**1️⃣2️⃣ Approval**

| **Name** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| [Solution Architect] | Architect |  |  |
| [Project Manager] | PM |  |  |
| [Lead Developer] | Dev Lead |  |  |