E-Commerce Website – Project Workflow

1. Business Overview:

- The goal of this project is to develop a **secure and scalable E-commerce platform** for online product browsing, purchasing, and order management.
  It enables users to **shop online conveniently**, while providing an **admin dashboard** for product, order, and customer management.
  Customers, Business Stakeholders, Project Managers, QA Teams, Technical Engineers, and Third-party Vendors.

Objectives:

- Provide a seamless shopping experience.
- Enable real-time order tracking and payment security.
- Automate backend operations like order processing and inventory management.

2. Requirements:

Functional Requirements

Module	Description
User Management	Register, login, manage profiles, reset password
Product Management	Add/update/delete products, view categories
Shopping Cart & Checkout	Add items, calculate totals, confirm orders
Payment Integration	Process online payments via Stripe / Razorpay
Order Management	View, update, and track order statuses
Shipping & Delivery	Track shipments via courier APIs
Customer Support	Submit queries, complaints, and feedback
Admin Panel	Manage users, products, orders, and reports

Software Requirements:

Layer	Technology
Frontend	React.js, HTML, CSS, JavaScript
Backend	Node.js (Express) or Spring Boot (Java)
Database	MySQL / PostgreSQL
Authentication	JWT & OAuth2
Payment Gateway	Razorpay / Stripe / PayPal
Cloud Infrastructure	AWS / Render / Vercel
Message Queue (Optional)	RabbitMQ / Kafka for async tasks
Version Control	Git + GitHub
Testing Tools	Postman, Jest, JUnit
Project Management	Jira / Trello

E-Commerce Website – Project Workflow

3. Design:

UI Design

- Responsive and consistent layout with clear navigation.
- Accessible on all devices (desktop, tablet, mobile).
- Emphasis on usability and modern design principles.

Database Design

Table	Key Attributes
Users	user_id, name, email, password, role
Products	product_id, name, description, price, stock, category_id
Orders	order_id, user_id, total_amount, status, order_date
Order_Items	id, order_id, product_id, quantity, price
Payments	payment_id, order_id, method, status, date
Shipping	shipping_id, order_id, address, status, tracking_number
Cart	cart_id, user_id, product_list, total_price
Reviews	review_id, user_id, product_id, rating, comment
Admin	admin_id, username, password, role

Entity Relationships:

- One User → Many Orders (1:N)
- One Product → Many Reviews (1:N)
- One Order → One Payment (1:1)
- One Order → Many Order_Items (1:N)
- One Category → Many Products (1:N)
- One Order → One Shipping Record (1:1)

Core Classes:

- User, Product, Order, OrderItem, Payment, Shipping, Cart, Review, Admin, Category.

4. Implementation Workflow:

- **Frontend Development:**
  - Build user and admin dashboards with React components and REST API integration.
- **Backend API Creation:**
  - Develop secure REST APIs (authentication, orders, payments).
- **Database Setup:**
  - Create schema and relationships using PostgreSQL/MySQL.
- **Payment Integration:**
  - Integrate Stripe/Razorpay SDK for real-time payment processing.

E-Commerce Website – Project Workflow

- **Testing:**
  - Perform unit, integration, and system testing using Postman, Jest, or JUnit.
- **Deployment:**
  - Deploy using Docker on AWS / Render / Heroku.
- **Monitoring:**
  - Use logs, analytics, and feedback for continuous improvement.

❖ The complete workflow ensures efficient coordination between frontend, backend, and database layers, integrating payment, shipping, and admin operations to deliver a robust and scalable E-Commerce solution.

Thankyou