**E-Commerce Application on IBM Cloud Foundry**

**Phase 1: Problem Definition and Design Thinking**

Build an artisanal e-commerce platform using IBM Cloud Foundry. Connect Skilled artisans with the global audience. Showcase handmade products , from exquisite jewellery to artistic home decor. Implement secure shopping carts, Smooth payment gateways, and an intuitive checkout process.

**Project Description**

The E-Commerce Application on IBM Cloud Foundry project aims to build a secure, scalable, and high-performance e-commerce platform for [Client/Company Name]. This platform will enable customers to browse products, make purchases, and manage their accounts seamlessly. Leveraging the capabilities of IBM Cloud Foundry, we will ensure a reliable and efficient hosting environment.

**Project Objectives**

1. **Cloud-Native Architecture**: Develop the application with cloud-native principles, ensuring scalability and flexibility.

2. **User-Friendly Interface**: Create an intuitive and responsive user interface to enhance the shopping experience.

3. **Product Catalog**: Implement a comprehensive product catalog with search, filter, and sorting options.

4. **Shopping Cart and Checkout**: Allow users to add, edit, and remove items from their carts and complete secure transactions.

5. **User Authentication**: Implement user registration and authentication for personalized experiences.

6. **Order Management**: Develop an order management system for tracking and managing customer orders.

7. **Payment Integration**: Integrate secure payment gateways for seamless transactions.

8. **Admin Dashboard**: Provide administrators with a dashboard for managing products, orders, and user accounts.

9. **Security and Compliance**: Implement industry-standard security measures, including data encryption and compliance with payment card industry (PCI) standards.

10. **Scalability**: Design the application for horizontal and vertical scalability to accommodate increased traffic and product listings.

11. **Performance Optimization**: Optimize application performance for quick loading times and smooth user interactions.

12. **Testing and Quality Assurance**: Conduct thorough testing, including load testing and security testing, to ensure application reliability.

13**. Documentation**: Create comprehensive documentation covering architecture, APIs, and deployment procedures.

# **Key Features**

**Customer-Facing Features**

- User Registration and Login

- Product Browsing and Searching

- Product Details Pages

- Shopping Cart Management

- Secure Checkout

- Order History and Tracking

- Product Reviews and Ratings

- Account Management

**Admin-Facing Features**

- Product Management (CRUD)

- Order Processing and Fulfillment

- User Management

- Analytics and Reporting

- Content Management (e.g., banners and promotions)

**Technology Stack**

**Frontend Technologies**:

- HTML5, CSS3, JavaScript

- React.js (or another frontend framework of choice)

**Backend Technologies**:

- Node.js (Express.js) for server-side logic

- IBM Cloud Foundry for hosting

- IBM Cloud Databases (e.g., Db2, PostgreSQL) for data storage

- RESTful APIs for communication

**Risks and Challenges**:

1. Infrastructure Downtime Risk

2. Data Security Risk

3. Scalability Challenges

4. Third-Party Service Dependency

5. Performance Bottlenecks

6. Scope Creep

7. Budget Overruns

8. Integration Complexity

9. User Experience Challenges

10. Compliance and Legal Risks

11. Resource Availability

12. Data Backup and Recovery

13. User Traffic Spikes

14. Vendor Lock-In

**FLOW CHART:**

