1. Architecture Documentation (architecture/)

Purpose:

Provide a comprehensive overview of the system's architecture, design principles, component interactions, and infrastructure setup. This section ensures that all stakeholders have a clear understanding of how the system is structured and how different components interact.

Contents:

a. System Overview (system-overview.md):

- **Introduction:**
  - Brief description of **FinPay** and its primary objectives.
  - Overview of the problem it solves and the value it provides to users.
- **Key Components:**
  - List of all microservices (e.g., Authentication Service, User Service, Payment Service).
  - Brief description of each service's responsibilities.
- **Technology Stack:**
  - Summary of technologies used (e.g., Java Spring Boot, Angular, Kafka).

b. Architecture Diagrams (diagrams/):

- **High-Level Architecture Diagram (high-level-architecture.png):**
  - Visual representation of all microservices, their interactions, external integrations (e.g., payment gateways), databases, and infrastructure components.
- **Component Diagrams (component-diagram.png):**
  - Detailed diagrams showing the internal structure of each microservice.
- **Data Flow Diagrams (data-flow-diagram.png):**
  - Illustrate how data moves through the system, including interactions between services and external APIs.
- **Deployment Diagram (deployment-diagram.png):**
  - Shows how services are deployed across infrastructure (e.g., Kubernetes clusters, Docker containers).

c. Design Decisions (design-decisions.md):

- **Microservices Architecture:**
  - Rationale for choosing microservices over monolithic architecture.
- **Technology Choices:**
  - Reasons for selecting specific technologies and frameworks.
- **Database Selection:**
  - Justification for using PostgreSQL, MongoDB, etc., for different services.
- **Communication Protocols:**
  - Decisions on using RESTful APIs, Kafka, RabbitMQ for inter-service communication.
- **Security Measures:**
  - Overview of security protocols like OAuth2, JWT, RBAC.

d. Infrastructure Setup (infrastructure-setup.md):

- **Cloud Provider Selection:**
  - Details about the chosen cloud provider (e.g., Azure) and reasons for selection.
- **Network Configuration:**
  - VPCs, subnets, firewalls, and other networking components.
- **Kubernetes Cluster Setup:**
  - Steps to provision and configure Kubernetes clusters.
- **Database Provisioning:**
  - Setup and configuration of databases (PostgreSQL, MongoDB).

e. Security Architecture (security-architecture.md):

- **Authentication & Authorization:**
  - Detailed explanation of how authentication and authorization are handled across services.
- **Data Encryption:**
  - Methods used to encrypt data at rest and in transit.
- **Vulnerability Management:**

    - Strategies for identifying and mitigating security vulnerabilities.

f. Scalability & Resilience Strategies (scalability-resilience.md):

- **Horizontal Scaling:**
    - How services can scale independently based on demand.
- **Fault Isolation:**
    - Techniques to ensure that failures in one service do not cascade to others.
- **Load Balancing:**
    - Mechanisms in place to distribute traffic evenly across service instances.

g. Monitoring & Logging Architecture (monitoring-logging.md):

- **Monitoring Tools:**
    - Overview of tools like Prometheus and Grafana used for monitoring.
- **Logging Strategy:**
    - Use of the ELK Stack for centralized logging and analysis.
- **Alerting Mechanisms:**
    - How alerts are configured and managed to notify the team of issues.