**Synopsis: Banking System**

**1. Introduction:**  
The banking system is a microservices-based application designed to handle core banking functionalities such as account management, transactions, and user authentication. It ensures high security, scalability, and efficiency by leveraging modern cloud technologies.

**2. Objectives:**

* Develop a secure and scalable banking system using Spring Boot.
* Implement microservices architecture to separate key functionalities.
* Use Spring Data JPA for efficient data management.
* Ensure high availability and load balancing using API Gateway and Eureka Server.
* Deploy the system on AWS using Docker and Kubernetes.

**3. System Architecture:**  
The system follows a microservices-based architecture with the following key components:

* **Authentication Service:** Handles user login, registration, and authorization.
* **Account Service:** Manages account creation, updates, and balance tracking.
* **Transaction Service:** Handles deposits, withdrawals, and fund transfers.
* **API Gateway:** Acts as a single entry point, routing requests to respective microservices.
* **Eureka Server:** Service registry for managing microservices discovery.

**4. Technologies Used:**

* **Backend:** Spring Boot, Spring Data JPA, Spring Cloud.
* **Database:** MySQL/PostgreSQL.
* **Service Discovery:** Eureka Server.
* **Containerization & Orchestration:** Docker and Kubernetes.
* **Cloud Deployment:** AWS (EKS for Kubernetes, RDS for database).

**5. Deployment Strategy:**

* Each microservice is containerized using Docker.
* Kubernetes manages deployment, scaling, and networking.
* API Gateway facilitates secure and efficient communication between services.
* AWS cloud infrastructure ensures reliability and scalability.

**6. Expected Outcomes:**

* A secure and efficient banking system with modular services.
* High availability with automated scaling and load balancing.
* Improved maintainability due to microservices architecture.
* Secure cloud deployment ensuring reliability and accessibility.