

In this project I developed a Banking Management System which is a full-stack web application developed using Java, Spring Boot, Microservices, and various Java Full Stack technologies to provide secure and efficient banking services. The system enables users to perform essential banking operations such as account management, fund transfers, loan applications, and transaction history tracking. It follows microservices architecture to ensure scalability, security, and maintainability.

Key Features:

1. User Account Management:
 - Secure user authentication and role-based access control.
 - Supports customer registration, login, and profile management.
2. Banking Transactions:
 - Enables fund transfers (NEFT, IMPS, RTGS) between accounts.
 - Provides real-time balance updates and transaction history.
 - Implement multi-factor authentication (MFA) for secure transactions.
3. Loan & Credit Services:
 - Users can apply for personal, home, and auto loans.
 - Loan status tracking and repayment schedule management.
4. Microservices Architecture:
 - Modular services for accounts, transactions, loans, and authentication.
 - Enables easy scaling and independent deployment of services.
5. Secure API & Integrations:
 - RESTful APIs secured using JWT-based authentication.
 - Integrated with third-party payment gateways.
6. Admin Dashboard:
 - Allows bank administrators to monitor customer accounts, approve loans, and generate reports.
 - Provides insights into transactions, fraud detection, and system performance.
7. Notification System:
 - Sends SMS and email alerts for transactions, password changes, and security updates.
8. Compliance & Security:
 - Follows banking regulations like PCI-DSS for secure payments.
 - Data encryption & role-based access control (RBAC) for enhanced security.

How It Works:

1. User Registration & Authentication:
 - Users sign up with personal details and verify via OTP (SMS/Email).
 - Secure login using JWT authentication.
2. Banking Operations:
 - Users can check account balance, view statements, and make transactions.
 - Transactions are processed securely with encrypted API calls.
3. Microservices Execution:
 - The Account Service manages user details.
 - The Transaction Service handles fund transfers and payments.
 - The Loan Service processes loan applications and approvals.
 - The Notification Service sends real-time alerts.
4. Admin Control Panel:
 - Admins can approve new accounts, review loan applications, and generate reports.
5. Security & Compliance:

- Transactions are logged and monitored for fraud detection.
- Secure data encryption & role-based access for enhanced protection.

Technologies & Tools Used:

Backend:

- Java, Spring Boot, Spring MVC – Core backend framework for business logic.
- Microservices & Spring Cloud – Service-based architecture for modular banking features.
- Hibernate & JPA – ORM for efficient database interaction.
- RESTful APIs & JWT Authentication – Securing transactions and API access.

Frontend:

- Angular / React / Thymeleaf – Frontend framework for user interface.
- HTML, CSS, JavaScript, Bootstrap – Responsive UI design.

Database & Cloud Services:

- MySQL / PostgreSQL – Storing user accounts and transactions.
- Redis – Caching transaction data for performance optimization.
- AWS / Azure / Google Cloud – Cloud deployment for scalability.

Security & DevOps:

- OAuth2, Spring Security, JWT – Secure authentication & authorization.
- Docker, Kubernetes – Containerization and deployment.
- Jenkins, GitHub Actions – CI/CD pipeline for automated testing & deployment