

NETBANKING PLATFORM

Final Year / Major Project Report

A Secure, Scalable & Industry-Ready Online Banking System

1. Introduction

The banking industry has undergone a massive digital transformation over the last decade. NetBanking platforms enable customers to perform banking activities remotely with high security and reliability.

This project focuses on designing and implementing a NetBanking platform that reflects real-world banking systems while remaining suitable for academic and internship-level evaluation.

The emphasis of this project is not only functionality but also architecture, scalability, security, and deployment readiness.

2. Problem Statement

Traditional banking systems require physical presence for most operations, leading to inefficiency and inconvenience.

Existing small-scale digital banking applications often lack proper architecture, security planning, and scalability.

The problem addressed is the design of a robust NetBanking system that supports secure transactions, auditability, and future growth.

3. Project Objectives

To design a secure and scalable NetBanking platform.

To implement core banking functionalities such as authentication, fund transfer, and transaction history.

To apply industry-standard architectural principles.

To ensure deployment readiness and extensibility.

4. Scope of the Project

The project covers online banking functionalities for individual users.

Admin operations and core banking backend integration are considered out of scope.

The system is designed to simulate real banking operations in a controlled academic environment.

5. System Architecture

The system follows a layered service-oriented architecture.

Each layer has a specific responsibility, ensuring separation of concerns.

The architecture supports horizontal scaling and fault isolation.

6. Technology Stack

Frontend: HTML, CSS, JavaScript / React (optional).

Backend: PHP / Node.js / Java (conceptual).

Database: Relational Database (MySQL / PostgreSQL).

Security: HTTPS, token-based authentication.

7. Functional Requirements

User registration and authentication.

Account balance inquiry.

Fund transfer between accounts.

Transaction history and statement generation.

Beneficiary management.

8. Non-Functional Requirements

High availability and reliability.

Secure handling of sensitive data.

Scalability for increased user load.

Maintainability and modularity.

9. Data Flow Diagrams

DFD Level 0 illustrates the system as a single process.

DFD Level 1 decomposes the system into major modules.

DFD Level 2 focuses on fund transfer operations.

10. Database Design

The database is normalized to reduce redundancy.

ACID properties ensure transactional integrity.

Audit logs are maintained for compliance.

11. Security Design

Passwords are stored using secure hashing algorithms.

All communications occur over encrypted channels.

Role-based access control restricts unauthorized access.

12. API Design

RESTful APIs are used for communication.

Endpoints are versioned for backward compatibility.

Input validation and error handling are enforced.

13. DevOps & Deployment

Separate environments are maintained for development, testing, and production.

CI/CD pipelines automate testing and deployment.

Monitoring and logging ensure system observability.

14. Testing Strategy

Unit testing validates individual components.

Integration testing ensures module interaction.

Security testing identifies vulnerabilities.

15. Limitations

The platform does not connect to real banking networks.

Advanced fraud detection is not implemented.

Mobile application support is limited.

16. Future Enhancements

Integration with real payment gateways.

Two-factor authentication.

Mobile banking application.

Microservices migration.

17. Conclusion

The NetBanking platform demonstrates a strong understanding of system design principles.

The project balances academic requirements with industry relevance.

It serves as a solid foundation for further enhancements.