

*A Report Submitted in  
Partial Fulfilment for  
award of Bachelor of Technology/Master of Integrated Technology*

**In  
DBMS (BCSE0452)  
COMPUTER SCIENCE & ENGINEERING**

**By  
Drishay Chauhan  
(2301330100084)**

**Under the Supervision of  
Ms. Neeti Taneja  
Assistant Professor**



**Department of Computer Science & Engineering  
School of Computer Science and Information Technology  
NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY  
GREATER NOIDA  
(An Autonomous Institute)  
Affiliated to  
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**

## **DECLARATION**

I hereby declare that the work presented in this report entitled “Secured Online Banking System”, was carried out by me. I have not submitted the matter embodied in this report for the award of any degree or diploma of any other University or Institute.

**Name :** Drishay Chauhan

---

*(Candidate's Signature)*

## **CERTIFICATE**

Certified that has carried out the project work presented in this Project Report in partial fulfillment of the requirements for the award of **Bachelor of Technology, CSE 2<sup>nd</sup> Year, 4<sup>th</sup> Sem** from **Noida Institute of Engineering & Technology** under my supervision.

Signature: \_\_\_\_\_

Ms. Neeti Taneja  
Assistant Professor  
NIET, Gr. Noida

Signature: \_\_\_\_\_

Dr. Kumud Saxena  
Dean SCSIT, HOD  
Department of CSE  
NIET, Gr. Noida

## **ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to **Ms. NEETI TANEJA** for their invaluable guidance, support, and constant supervision throughout the development of this project. Their expertise and encouragement have been pivotal in shaping the project.

I would also like to extend my thanks and appreciation to our esteemed **HOD (Dr. Kumud Saxena)** for their continuous motivation and support. Their encouragement has been instrumental in completing this project successfully.

Finally, I am deeply thankful to my peers and everyone who directly or indirectly contributed to this project with their knowledge, advice, and assistance.

## **ABSTRACT**

This project presents a Secured Online Bank Management System that modernizes traditional banking software. Built using a Java backend with MySQL and a React-based frontend, the system provides secure login, account management, and transaction functionalities. Key security measures, including password hashing and role-based authentication, ensure protection against unauthorized access. The solution addresses the limitations of legacy terminal-based systems, offering improved usability, scalability, and data integrity, and lays the foundation for further enhancements in online banking.

# TABLE OF CONTENTS

	<b>Page No.</b>
Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Overview	
1.2 Objective And Scope	
<b>CHPATER 2: LITERATURE REVIEW</b>	<b>2</b>
2.1 Overview of Existing Systems	
2.2 Limitations of Existing Systems	
<b>CHAPTER 3: REQUIREMENTS</b>	<b>3-4</b>
3.1 Functional Requirements	
3.2 Non-Functional Requireme	
3.3 Software and Hardware Requirements	
<b>CHAPTER 4: SYSTEM DESGIN AND IMPLEMENTATION</b>	<b>5-6</b>
4.1 System Architecture (Flow diagram)	
4.2 GUI Design	
4.3 Database Design	
4.4 Implementation Details	
<b>CHAPTER 5: TESTING</b>	<b>8</b>
5.1 Testing Methodology	
5.2 Test Cases and Results	
<b>CHAPTER 6: CONCLUSION AND FUTURE WORK</b>	<b>9</b>
6.1 Conclusion	
6.2 Future Work	
<b>APPENDIX</b>	<b>10</b>

# **INTRODUCTION**

## **1.1 Overview**

The project aims to develop an online banking platform that replaces outdated, terminal-based bank management systems. The system integrates a robust Java backend and a dynamic React frontend to deliver a secure and efficient banking experience. Emphasis is placed on safeguarding sensitive customer data and streamlining day-to-day banking operations.

## **1.2 Objective And Scope**

The primary objective is to design and implement a bank management system that ensures high security, scalability, and user-friendliness.

- **Objectives:**

- Enable secure login and password recovery through encryption and hashing.
- Provide features such as account details, balance inquiry, deposits, withdrawals, and fund transfer.
- Offer role-based access for customers, employees, and managers.

- **Scope:**

- The project focuses on creating a web-based platform using Java, MySQL, and React.
- It is limited to the core functionalities required for online banking and does not include external integrations like real-time trading or loan processing.

# **LITERATURE REVIEW**

## **2.1 Overview of Existing Systems**

Existing bank management systems are usually built on legacy technologies and often operate as terminal-based applications. These systems typically offer basic functionalities but are hindered by clunky interfaces and outdated security measures. Recent studies indicate that traditional systems are prone to vulnerabilities, difficult to scale, and challenging to maintain.

## **2.2 Limitations of Existing Systems**

- **Security Issues:** Inadequate methods for user authentication and data encryption increase the risk of unauthorized access.
- **User Interface:** Terminal-based interfaces are not user-friendly, especially when compared to modern web applications.
- **Maintenance & Scalability:** Legacy systems suffer from poor scalability and high maintenance costs due to outdated technology stacks.
- **Integration Difficulties:** Limited support for integrating new technologies leads to challenges in adapting to market needs.



# **REQUIREMENTS**

## **3.1 Functional Requirements**

- **User Authentication:** Secure login, password recovery, and role-based access.
- **Account Management:** Viewing account details, balance inquiry, deposit, withdrawal, and fund transfer functionalities.
- **Data Integrity:** Ensure that all financial transactions are accurate and recorded securely.

## **3.2 Non-Functional Requirements**

- **Security:** Use of encryption, password hashing, and secure connection protocols.
- **Performance:** The system must handle multiple simultaneous transactions with minimal response time.
- **Usability:** Intuitive user interface for smooth navigation and ease of operation.
- **Scalability:** Ability to expand features and accommodate a growing user base without significant changes in infrastructure.

## **3.3 Software and Hardware Requirements**

- **Software:**
  - **Backend:** Java (JDK 22), Apache Tomcat, MySQL.
  - **Frontend:** Node.js, React, HTML/CSS/JavaScript.

- Development Tools: VS Code or Eclipse/IntelliJ, Git for version control.
- Hardware:
  - A standard desktop or server machine with 4GB RAM minimum.
  - Adequate storage for database and application files.

# SYSTEM DESIGN AND IMPLEMENTATION

## 4.1 System Architecture (Flow Diagram)

The system follows a three-tier architecture:

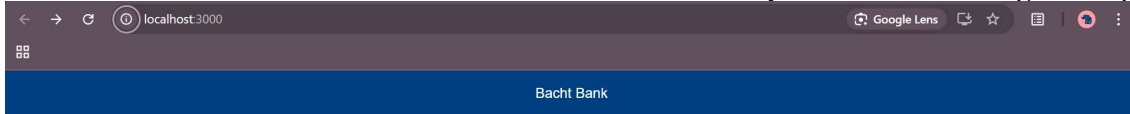
- **Presentation Layer:** Developed using React, it provides a dynamic and responsive user interface.
- **Application Layer:** A Java backend that handles business logic, authentication, and transaction processing via Servlets.
- **Data Layer:** MySQL database managing persistent storage and ensuring data integrity.

*Include a flowchart diagram here that shows the data flow from user interaction through the frontend, to the backend processing, and finally database updates.*

## 4.2 GUI Design

The graphical user interface is designed for clarity and ease of use. Key features include:

- A landing page with three options (Customer, Employee, Manager).
- A role-based login screen that adapts to the user type.
- Dashboard pages for different roles containing user details, a navigation menu for various operations, and a central area for executing transactions.



Customer

Employee

Manager

© Copyright reserved to Bacht Bank

### 4.3 Database Design

The database schema includes tables such as:

- customer\_details: Stores customer login and account data.
- staff\_details: Maintains employee login data.
- transaction\_history: Records deposits, withdrawals, and transfers.

*An ER diagram should be included to show relationships such as “Customer has many Transactions”.*

### 4.4 Implementation Details

The system’s core is built using:

- Java Servlets: Handling user authentication, secure transactions, and interaction with the MySQL database.
- MySQL: Backend storage for user data, account balances, and transaction records.
- React: Frontend framework providing a responsive user interface.

Key implementation features:

- Passwords are hashed using a custom algorithm.

- Role-based authentication is enforced by the backend.
- Web services expose APIs (e.g., POST /api/login) that the React frontend consumes.

# TESTING

## 5.1 Testing Methodology

Testing was carried out in multiple phases:

- Unit Testing: Individual modules (e.g., authentication, transaction processing) were tested using sample data.
- Integration Testing: Ensured the backend API correctly communicates with the database and the React frontend.
- User Acceptance Testing: A limited group of users tested the system to validate usability and functionality.

## 5.2 Test Cases and Results

Examples of test cases include:

- Login: Valid credentials result in successful login; invalid credentials show appropriate error messages.
- Fund Transfer: The system accurately deducts the transfer amount from the sender and adds it to the recipient while maintaining transaction logs.
- Data Consistency: Simultaneous transactions do not lead to data loss or inconsistent account balances. The tests confirmed that all functional and non-functional requirements are met with satisfactory performance.

## CONCLUSION AND FUTURE WORK

### 6.1 Conclusion

The Secured Online Bank Management System successfully provides a modern, secure, and efficient solution to the limitations of legacy banking systems. The integration of a Java backend, MySQL, and a React frontend creates a robust platform that enhances user experience and system security. The project demonstrates the viability of transitioning from traditional systems to contemporary, web-based applications.

### 6.2 Future Work

Future enhancements may include:

- **Enhanced Security:** Incorporating multi-factor authentication and advanced encryption methods.
- **Auto-Trading Features:** Integrating market analytics for automated transaction signals.
- **Mobile Application:** Developing a mobile app version for greater accessibility.
- **Performance Optimization:** Further tuning the system for high-frequency transactions and increased concurrent users.

## **APPENDIX**

### **Source Code**

### **GitHub Repository Link:**

<https://github.com/Drishay/CollegeProjects/tree/main/DBMS-Project>