UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

PROJECT REPORT

Of

Bank Management System

Subject : Advance Internet Programming Lab

Subject code : 24CAP-625

Submitted By :	**Submitted To :**
Name : Rahul Kumar	Dr Sarabjeet Kaur
UID : 24MCA20169	
Section : 4_B	

MASTERS OF COMPUTER APPLICATION

Chandigarh University, Mohali

Tabel Of Content

1.Acknowledgement

I would like to express my heartfelt gratitude to all those who helped me in completing this mini project called "Bank Management System." First, I would like to thank my respected teacher for giving me this opportunity and constantly supporting me. The teacher's valuable guidance and constant feedback helped me concentrate and ensure that I completed the project on time.

I would also like to thank my classmates and friends for sharing useful ideas with me, explaining concepts, and encouraging me throughout the project. Their support proved to be tremendously helpful, making the process easier and much more enjoyable.

I am extremely thankful to my family for their love and support during the time of the project. In all circumstances, they encouraged me and gave me the time and space needed to focus on my work.

In the end, I would like to appreciate online tutorials, forums, and documentation that helped guide me to write the code and correct errors. This project would not have been possible without all this support combined. I am very grateful to all who helped me directly or indirectly.

2.Abstract

The Bank Management System is a straightforward and user-friendly web application that offers a way for users to perform online activities related to their bank account. In this project, the user is able to create a new account, login, check their account balance, add funds, withdraw funds, and transfer funds to another account. The main goal of this project is to streamline some basic banking activities for users, making them easier and faster.

The system is built using Java, JSP, Servlets and MySQL as the backend Database. The system also utilizes Bootstrap for an organized and responsive design. Essentially, the application works by connecting web pages to a database so that users are able to securely save and manage the details of their bank accounts.

The project has the following components: user registration, user login authentication, account dashboard, transaction system, and proper feedback to the user after each action has taken place. The system also keeps a record of the transactions, so users can reference them as needed in the future.

This system is useful to learn how banking applications work in the real-world and it teaches how to use Java and databases together in an actual project implementation. Overall, the system is safe and easy to use, and will be beneficial for learning web-development techniques.

3.Introduction

Banking is a fundamental aspect of people's daily lives. Customers expect quick and secure access to their accounts, and online banking provides that access. The Bank Management System project was developed to allow users to manage their banking accounts from anywhere using the internet in a very easy way.

In the system, a user registers by entering their personal information and account information. Once registered, the user logs in securely using their account username and password. After a successful login, the user is taken to a dashboard that displays the balance of their account, as well as options to add funds to their account, withdraw money from their account, or send funds to another account. All activity is logged and stored in the SQL database.

The Bank Management System project is written in Java, and uses Servlets, JSP, and MySQL. This project provides an ideal opportunity to highlight how a real-time web application using backend and frontend technology runs. The Project exposes the student to both web development and database management skills.

In addition, it includes minimal securities, but is user-friendly. Overall, the project exemplifies how simple online banking services can be created and managed securely and systematically.

4.Design and Flow of Project

The Bank Management System is designed in a simple way so users can easily perform banking activities online. The project follows a smooth flow from login to all banking operations.

Flow of the system:

1. Registration Page: Users create a new account by entering personal details like name, email, password, and a unique account number.
2. Login Page: After registration, users can log in using their email and password.
3. Dashboard: Once logged in, users can see their balance and buttons like Add Money, Withdraw Money, View Details, Account Information , Passbook, Loan Information, Insurance, ATM Details, Contact US and Logout.
4. Add Money: User enters the amount to add, and the balance updates in the database.
5. Withdraw Money: User enters the amount to withdraw, and if the balance is enough, it deducts from the total.
6. Passbook & User Info: Shows all transaction history and account details.

5.Code (Only Java and JSP)

LoginServer.java

```
package com.mycompany.bankmanagementsystem.servlet;

import com.mycompany.bankmanagementsystem.dao.DBConnection;


import jakarta.servlet.*;

import jakarta.servlet.http.*;

import jakarta.servlet.annotation.*;


import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;


@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

  protected void doPost(HttpServletRequest request, HttpServletResponse response)

      throws ServletException, IOException {


    String name = request.getParameter("name");

    String mobile = request.getParameter("mobile");

    String account_no = request.getParameter("account_no");

    String email = request.getParameter("email");

    String password = request.getParameter("password");

    try {

      Connection conn = DBConnection.getConnection();

      String sql = "SELECT * FROM user_details WHERE name=? AND mobile=? AND account_no=? AND email=? AND password=?";

      PreparedStatement ps = conn.prepareStatement(sql);

      ps.setString(1, name);

      ps.setString(2, mobile);

      ps.setString(3, account_no);

      ps.setString(4, email);

      ps.setString(5, password);

      ResultSet rs = ps.executeQuery();

      if (rs.next()) {

        HttpSession session = request.getSession();

        session.setAttribute("name", name);

        session.setAttribute("mobile", mobile);

        session.setAttribute("account_no", account_no);

        response.sendRedirect("userdetails.jsp");

      } else {

response.sendRedirect("index.jsp?error=Incorrect+Information");

      }

    } catch (Exception e) {

      e.printStackTrace();

      response.getWriter().println("Error: " + e.getMessage());

    }}}
```

RegisterServlet.java

```java
package com.mycompany.bankmanagementsystem.servlet;

import com.mycompany.bankmanagementsystem.dao.DBConnection;

import jakarta.servlet.*;

import jakarta.servlet.http.*;

import jakarta.servlet.annotation.*;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;


@WebServlet("/RegisterServlet")

public class RegisterServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        String name = request.getParameter("name");

        String mobile = request.getParameter("mobile");

        String account_no = request.getParameter("account_no");

        String email = request.getParameter("email");

        String password = request.getParameter("password");

        try {

            Connection conn = DBConnection.getConnection();

            String sql = "INSERT INTO user_details (name, mobile, account_no, email, password) VALUES (?, ?, ?, ?, ?)";

            PreparedStatement ps = conn.prepareStatement(sql);

            ps.setString(1, name);

            ps.setString(2, mobile);

            ps.setString(3, account_no);

            ps.setString(4, email);

            ps.setString(5, password);

            ps.executeUpdate();

            response.sendRedirect("index.jsp");

        } catch (Exception e) {

            e.printStackTrace();

            response.getWriter().println("Error: " + e.getMessage());

        }

    }

}
```

AddMoneyServelet

```java
package com.mycompany.bankmanagementsystem.servlet;


import com.mycompany.bankmanagementsystem.dao.DBConnection;


import jakarta.servlet.*;

import jakarta.servlet.http.*;

import jakarta.servlet.annotation.*;


import java.io.IOException;

import java.sql.*;

import java.util.logging.Level;

import java.util.logging.Logger;


@WebServlet("/AddMoneyServlet")

public class AddMoneyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession();
        String account_no = (String) session.getAttribute("account_no");
        double amount = Double.parseDouble(request.getParameter("amount"));

        // Check if amount is zero
        if (amount == 0) {
            response.sendRedirect("addmoney.jsp?error=Amount+cannot+be+zero");
            return;
        }
        if (account_no == null) {
            response.sendRedirect("index.jsp?error=Please+login+first");
            return;
        }

        // Database connection and money adding logic
        try {
            Connection conn = DBConnection.getConnection();
            String sql = "SELECT total FROM transactions WHERE account_no = ? ORDER BY date DESC LIMIT 1";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, account_no);
            ResultSet rs = ps.executeQuery();

            double currentTotal = 0;
            if (rs.next()) {
                currentTotal = rs.getDouble("total");
            }

            double newTotal = currentTotal + amount;

            // Insert the transaction into the database
            String insertQuery = "INSERT INTO transactions (account_no, credit, total) VALUES (?, ?, ?)";
            PreparedStatement insertStmt = conn.prepareStatement(insertQuery);
            insertStmt.setString(1, account_no);
            insertStmt.setDouble(2, amount);  // Debit is the added amount
            insertStmt.setDouble(3, newTotal);
            insertStmt.executeUpdate();


            response.sendRedirect("userdetails.jsp?msg=Amount+added+successfully");

        } catch (SQLException e) {
            e.printStackTrace();
            response.sendRedirect("addmoney.jsp?error=Failed+to+add+money");
        } catch (Exception ex) {
            Logger.getLogger(AddMoneyServlet.class.getName()).log(Level.SEVERE, null, ex);
        }}}
```

WIthdrawMoneyServelet

```
package com.mycompany.bankmanagementsystem.servlet;
import com.mycompany.bankmanagementsystem.dao.DBConnection;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;
import java.io.IOException;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;


@WebServlet("/WithdrawMoneyServlet")
public class WithdrawMoneyServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      HttpSession session = request.getSession();
      String account_no = (String) session.getAttribute("account_no");
      double amount = ouble.parseDouble(request.getParameter("amount"));
      // Check if amount is zero
      if (amount == 0) {
         response.sendRedirect("withdraw.jsp?error=Amount+cannot+be+zero");
         return;
      }
      if (account_no == null) {
         response.sendRedirect("index.jsp?error=Please+login+first");
         return;
      }
      // Database connection and withdrawal logic
      try {
         Connection conn = DBConnection.getConnection();
         String sql = "SELECT total FROM transactions WHERE account_no = ? ORDER BY date DESC LIMIT 1";
         PreparedStatement ps = conn.prepareStatement(sql);
         ps.setString(1, account_no);
         ResultSet rs = ps.executeQuery();


         double currentTotal = 0;
         if (rs.next()) {
            currentTotal = rs.getDouble("total");
         }


         if (currentTotal < amount) {
            response.sendRedirect("withdraw.jsp?error=Insufficient+balance");
            return;
         }


         double newTotal = currentTotal - amount;


         // Insert the transaction into the database
         String insertQuery = "INSERT INTO transactions (account_no, debit, total) VALUES (?, ?, ?)";
         PreparedStatement insertStmt = conn.prepareStatement(insertQuery);
         insertStmt.setString(1, account_no);
         insertStmt.setDouble(2, amount);  // Credit is the withdrawn amount
         insertStmt.setDouble(3, newTotal);
         insertStmt.executeUpdate();



         response.sendRedirect("userdetails.jsp?msg=Amount+withdrawn+successfully");


      } catch (SQLException e) {
         e.printStackTrace();
         response.sendRedirect("withdraw.jsp?error=Failed+to+withdraw+money");
      } catch (Exception ex) {
         Logger.getLogger(WithdrawMoneyServlet.class.getName()).log(Level.SEVERE, null, ex);
      }}}
```

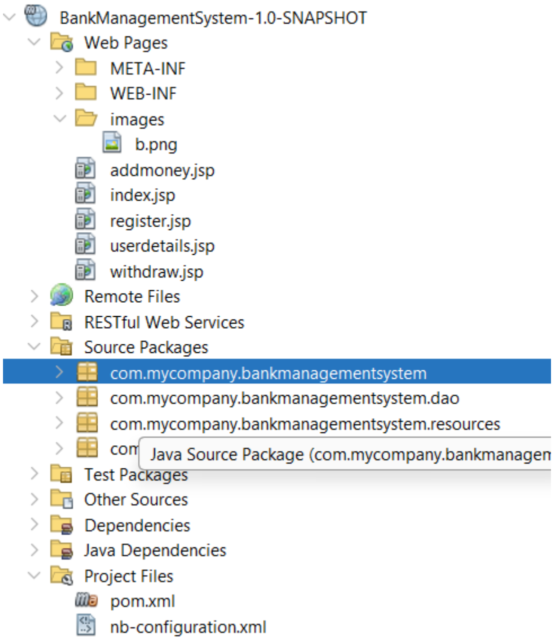
DownloadPssbookServelet

```
package com.mycompany.bankmanagementsystem.servlet;


import com.mycompany.bankmanagementsystem.dao.DBConnection;


import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.*;


import java.io.*;

import java.sql.*;


@WebServlet("/DownloadPassbookServlet")

public class DownloadPassbookServlet extends HttpServlet {


    protected void doGet(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {


        HttpSession session = request.getSession(false);

        String account_no = (String) session.getAttribute("account_no");

        String name = (String) session.getAttribute("name");


        if (account_no == null || name == null) {

            response.sendRedirect("index.jsp?error=Please+login+first");

            return;

        }


        // Create a safe file name

        String safeName = name.replaceAll("\\s+", "_");

        String fileName = "Passbook_" + safeName + "_" + account_no + ".csv";


        response.setContentType("text/csv");

        response.setHeader("Content-Disposition", "attachment; filename=" + fileName);


        try (PrintWriter writer = response.getWriter()) {

            writer.println("Date,Debit,Credit,Total");


            Connection conn = DBConnection.getConnection();

            PreparedStatement ps = conn.prepareStatement(

                "SELECT * FROM transactions WHERE account_no = ? ORDER BY date DESC"

            );

            ps.setString(1, account_no);

            ResultSet rs = ps.executeQuery();


            while (rs.next()) {

                String date = rs.getTimestamp("date").toString();

                String debit = rs.getDouble("debit") > 0 ? String.valueOf(rs.getDouble("debit")) : "-";

                String credit = rs.getDouble("credit") > 0 ? String.valueOf(rs.getDouble("credit")) : "-";

                String total = String.valueOf(rs.getDouble("total"));


                writer.println(date + "," + debit + "," + credit + "," + total);

            }


        } catch (Exception e) {

            e.printStackTrace();

        }

    }
```

DBConnection.java

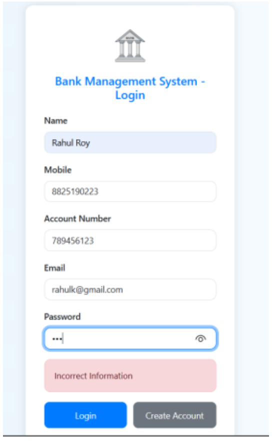
```
package com.mycompany.bankmanagementsystem.dao;

import java.sql.Connection;

import java.sql.DriverManager;



public class DBConnection {

    public static Connection getConnection() throws Exception {

        Class.forName("com.mysql.cj.jdbc.Driver");

        return DriverManager.getConnection("jdbc:mysql://localhost:3306/bank_db", "root", "rahul");

    }}
```

JSP file and Images file



```
BankManagementSystem-1.0-SNAPSHOT
  Web Pages
    META-INF
    WEB-INF
    images
      b.png
    addmoney.jsp
    index.jsp
    register.jsp
    userdetails.jsp
    withdraw.jsp
  Remote Files
  RESTful Web Services
  Source Packages
    com.mycompany.bankmanagementsystem
    com.mycompany.bankmanagementsystem.dao
    com.mycompany.bankmanagementsystem.resources
    com  Java Source Package (com.mycompany.bankmanagem
  Test Packages
  Other Sources
  Dependencies
  Java Dependencies
  Project Files
    pom.xml
    nb-configuration.xml
```

6.Output



Bank Management System - Login

Name: Rahul Roy

Mobile: 8825190223

Account Number: 789456123

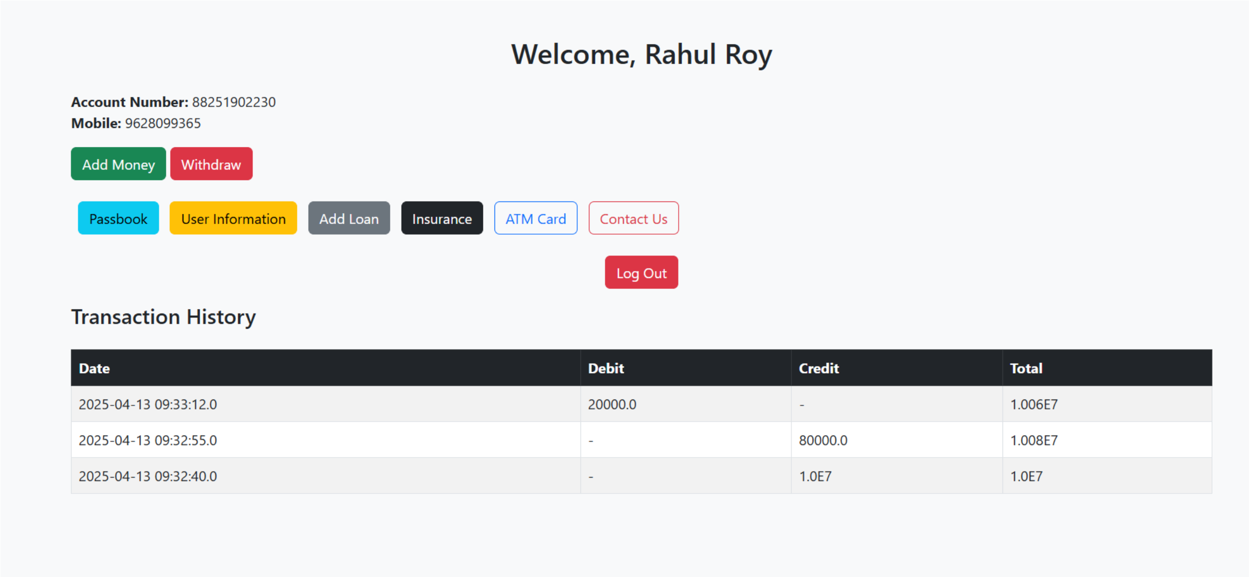
Email: rahulk@gmail.com

Password: •••

Incorrect Information

After Click Create Account Button

Dashboard

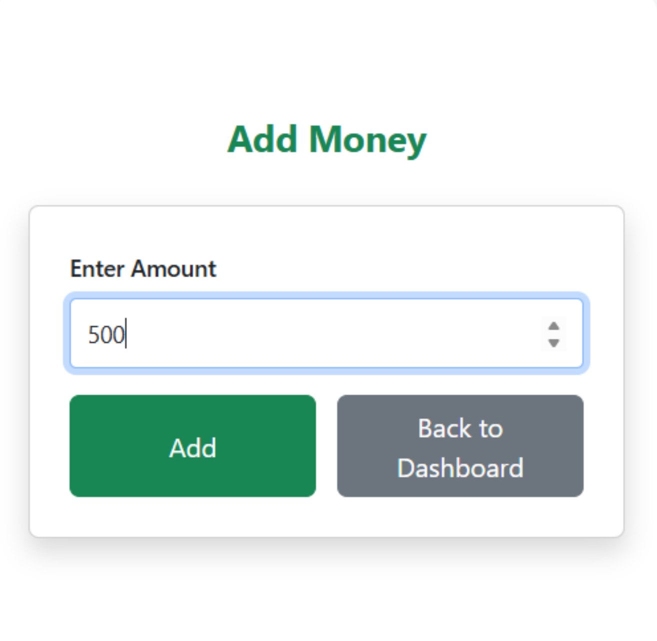

Welcome, Rahul Roy

Account Number: 88251902230
Mobile: 9628099365

Transaction History

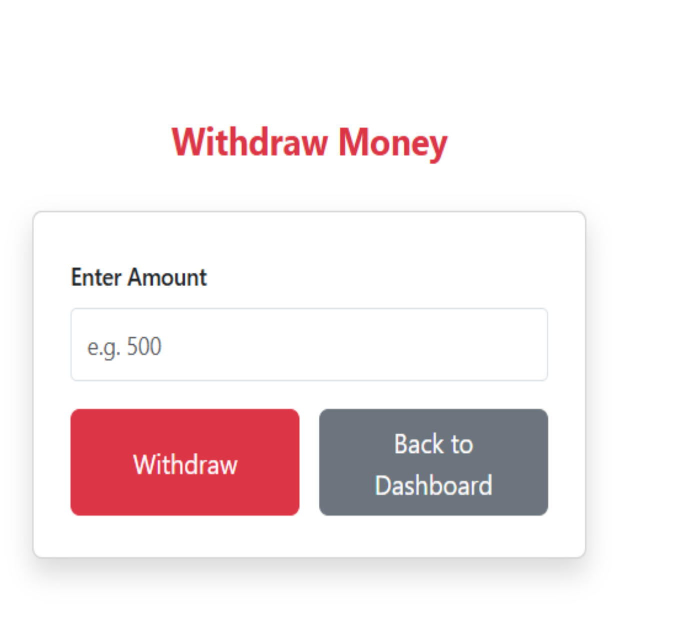
Date	Debit	Credit	Total
2025-04-13 09:33:12.0	20000.0	-	1.006E7
2025-04-13 09:32:55.0	-	80000.0	1.008E7
2025-04-13 09:32:40.0	-	1.0E7	1.0E7

After Click Add Money Button


Add Money

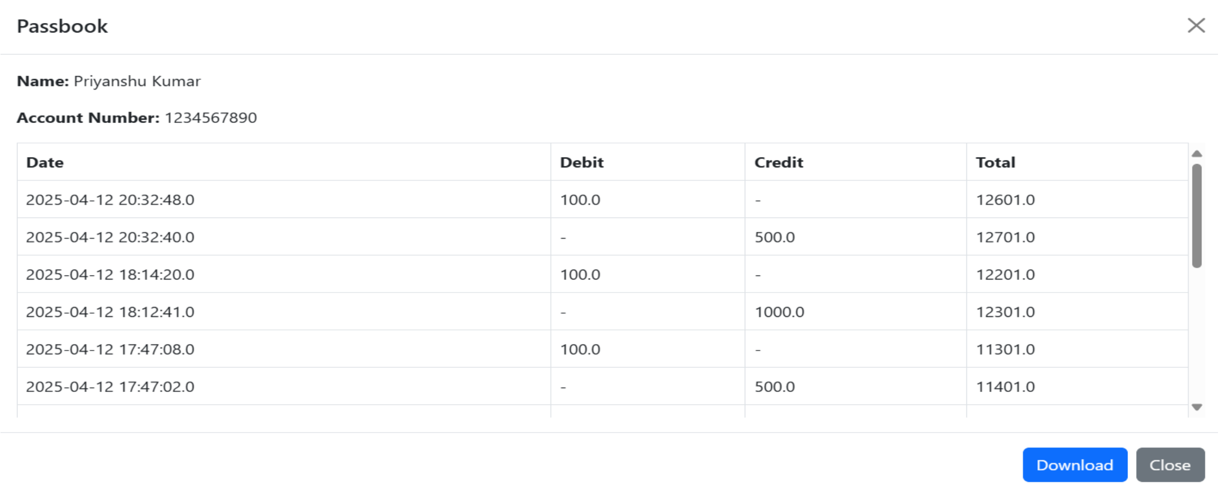
Enter Amount: 500

After Click Widhdraw Button


Withdraw Money

Enter Amount: e.g. 500

After Click Passbook Button

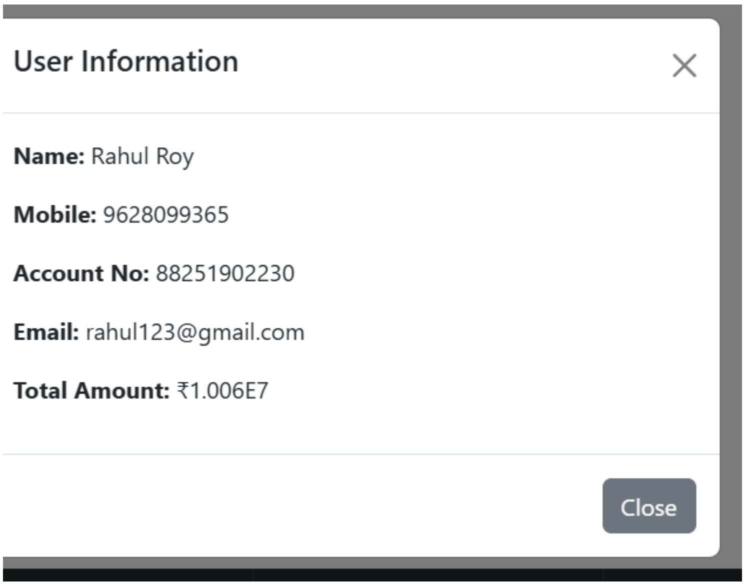

Passbook

Name: Priyanshu Kumar

Account Number: 1234567890

Date	Debit	Credit	Total
2025-04-12 20:32:48.0	100.0	-	12601.0
2025-04-12 20:32:40.0	-	500.0	12701.0
2025-04-12 18:14:20.0	100.0	-	12201.0
2025-04-12 18:12:41.0	-	1000.0	12301.0
2025-04-12 17:47:08.0	100.0	-	11301.0
2025-04-12 17:47:02.0	-	500.0	11401.0

After Click Account Information Button



User Information

Name: Rahul Roy

Mobile: 9628099365

Account No: 88251902230

Email: rahul123@gmail.com

Total Amount: ₹1.006E7

7.Future Work

The current version of the Banking Management System includes essential features such as user login, adding money, withdrawing money, and applying for loans. However, there is significant scope for improvement and expansion to make the system more robust, user-friendly, and secure.

In the future, additional functionalities such as **transaction history**, **account statements**, and **real-time notifications** (via SMS or email) can be integrated to enhance user experience and transparency. Implementation of **two-factor authentication (2FA)** and **biometric login** will improve system security. Features like **credit score monitoring** and **loan eligibility calculators** can help users make informed financial decisions.

Moreover, an **admin dashboard** can be introduced for managing users, monitoring transactions, approving/rejecting loan applications, and generating reports. Integration with **third-party payment gateways** (like UPI, Paytm, or Razorpay) will allow seamless money transfer and bill payment facilities.

To improve scalability, the project can be migrated to a **cloud-based infrastructure**. Adding support for **multiple languages** and **voice assistance** will make the system more inclusive. A mobile app version of the system will ensure better accessibility and ease of use for users on the go.

Finally, the system can be enhanced by applying **machine learning models** for fraud detection, predicting loan defaults, and offering personalized financial recommendations. These future enhancements will transform the project from a basic banking application into a smart, secure, and user-centric banking management platform.

8.Conclusion

The Banking Management System project has been developed to simplify and automate essential banking operations such as user login, adding money, withdrawing money, and applying for loans. The system provides a user-friendly interface, allowing customers to perform transactions efficiently and securely. It also helps reduce manual errors and increases the speed and reliability of banking services.

This project not only covers the core functionalities of digital banking but also lays the foundation for future expansion and integration of advanced features. Throughout the development process, key programming concepts, database management, and secure authentication practices were applied to ensure a functional and scalable system.

Overall, the project has improved understanding of real-world banking operations and how technology can be used to solve financial service challenges. It demonstrates how digital systems can enhance user experience while ensuring data accuracy and security.

This project serves as a solid base for building a more advanced and fully-featured banking application in the future.

9.References

- **Bank Management System Report PDF – Scribd**

A comprehensive project report detailing the development of a bank management system using Java, including system design, implementation, and features.
View Document

- **Project Report on Online Banking – Scribd**

This document outlines the development of an online banking system using Java Server Pages, focusing on online transactions and customer convenience.
View Document

- **Bank Management System in Java and MySQL Report – SlideShare**

A detailed report covering the implementation of a bank management system using Java and MySQL, including design and architecture.
View Presentation

- **Bank Management Project in Java using JSP, Servlet, and MySQL – Codebun**

A practical project demonstrating the development of a bank management system with source code and implementation details.
View Project

- **Online Bank Management System using JSP and Servlet – YouTube Playlist**

A series of video tutorials guiding through the creation of an online bank management system using JSP and Servlets.
Watch Playlist

- **JSP Servlet Projects with Source Code Free Download – Java Guides**

A collection of over 100 free Java/Java EE projects developed using JSP, Servlet, JDBC, Hibernate, and MySQL for learning purposes.
Explore Projects

- **Online Banking Project – Tpoint Tech**

A project detailing the development of an online banking system using JSP, JDBC, JavaScript, AJAX, and Oracle, suitable for NetBeans IDE.
View Project

- **Internet Banking Management System – GitHub Repository**

A fully functional Java web application for internet banking management using JSP, Servlet, and JDBC.
View Repository

- **Bank Management System Report – SlideShare**

This document describes a project to develop a bank management system using Java, including functional and non-functional requirements, code implementation, and sample outputs.
View Presentation

- **Banking Management System (SRS Report) – ResearchGate**

A Software Requirements Specification report detailing the functionalities, design diagrams, and future scope of a banking management system.
View Report

- **Banking Management System – Studocu**

A project report submitted for the Master of Computer Applications, covering system documentation and implementation details.
View Document

- **Bank Management System – IRJMETS**

A peer-reviewed journal article demonstrating the implementation of core banking functionalities through a web-based application.
View Article