Task Report

Name: Makula Pravallika

Email-id: makulapravallika02@gmail.com

Task Title: ATM interface Project

Task Description:

In this Project, we develop an ATM interface program using Java.This Project will simulate the functionalities of a real ATM machine providing users with a secure and ituitive way to perform banking operations.

Steps Taken:

1. I create two java classes.
2. First class is Operations.java In this program I write all the methods implementations here that are deposit(),withdraw(),balanceenquiry() and userAuthentication().
3. Another class is bankingapplication.java It is user implementation class. Here we read input from the user that is userid and pin then by calling userAuthentication() method we authenticate the user if user entered details is correct then display ATM operations otherwise display invalid credentials.
4. To store the user data I take one table in backend for this I use postgresql.
5. I create one table bankdata name with attributes uid,pin,acno,name,balance,emailid.
6. By running the bankingapplication.java file it will run.

Solution:

- Table creation in postgresql-
  create table bankdata(uid text Primary key,pin int,acno int unique,name text,balance int,email text);
- Insert data into that table-
  insert into bankdata
  values('Pravallika30',12345,101,'pravallika',50,'makulapravallika02@gmail.com');
- Create java classes

1. **Operations.java**

```
package org.example;

import java.sql.*;
import java.util.Scanner;

public class Operations {
    static Scanner sc=new Scanner(System.in);
    public static void main(String[] args) {
        Connection conn;
        try {
            conn=
```

```
DriverManager.getConnection("jdbc:postgresql://localhost:5432/demo","pos
tgres","Pravallika@501");
            System.out.println("connection successfully");

        } catch (SQLException e) {
            System.out.println("connection failed");
            //e.printStackTrace();
        }
    }
    public static boolean userAuthenticate(Connection conn){
        System.out.println("enter userid:");
        String uid=sc.next();
        System.out.println("enter pin:");
        int pin=sc.nextInt();
        String q="select *from bankdata where uid=?";
        try {
            PreparedStatement ps=conn.prepareStatement(q);
            ps.setString(1,uid);
            ResultSet rs=ps.executeQuery();
            while(rs.next()){
               String userid=rs.getString(1);
               int pinp=rs.getInt(2);
               if(uid.equalsIgnoreCase(userid) && pin==pinp){
                   return true;
               }
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
        return false;

    }
    public static void deposit(Connection conn){
        System.out.println("enter account number:");
        int ac=sc.nextInt();
        System.out.println("enter amount to deposit");
        int amount=sc.nextInt();
        if(amount>0) {
            try {
                String que = "update bankdata set balance=balance+?
where acno=?";
                PreparedStatement ps = conn.prepareStatement(que);
                ps.setInt(1, amount);
                ps.setInt(2, ac);
                int row=ps.executeUpdate();
                if(row>0) System.out.println("Amount is Deposited
successfully");
                else System.out.println("Deposite not successful:Invalid
details");
                conn.close();
            } catch (SQLException e) {
                throw new RuntimeException(e);
            }
        }else System.out.println("Deposite not successful:Invalid
details");

    }
    public static void withdraw(Connection conn){
        int bal=0;
        System.out.println("enter account number:");
        int ac=sc.nextInt();
```

```
        System.out.println("enter amount to withdraw");
        int amount=sc.nextInt();
        if(amount<=0) {
            System.out.println("Invalid amount");
            throw new RuntimeException();
        }
        try {
            String q="select *from bankdata where acno=?";
            String que="update bankdata set balance=balance-? where
acno=?";
            PreparedStatement p=conn.prepareStatement(q);
            PreparedStatement ps=conn.prepareStatement(que);
            p.setInt(1,ac);
            ResultSet rs=p.executeQuery();
            if(rs.next())
                bal=rs.getInt("balance");
            if(bal>amount) {
                ps.setInt(1, amount);
                ps.setInt(2, ac);
                ps.executeUpdate();
                System.out.println("Amount is withdraw successfully");
                conn.close();
            }
            else System.out.println("With draw not possible");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
    public static void balanceEnquiry(Connection conn){
        System.out.println("enter account number:");
        int ac=sc.nextInt();
        try {
            String que="select *from bankdata where acno=?";
            PreparedStatement ps=conn.prepareStatement(que);
            ps.setInt(1,ac);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                int a = rs.getInt(3);
                String s = rs.getString(4);
                int t = rs.getInt(5);
                System.out.println("Account no:" + a + "\n Account
Holdername:" + s + "\n Balance:" + t);
                conn.close();
            }
        } catch (SQLException e) {
            //throw new RuntimeException(e);
            System.out.println("invalid");
        }
    }
}
```

2.BankingApplication.java

```
package org.example;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;
```

```
public class BankApplication {
    static Connection conn;
    public static void main(String[] args){
        try {
            Class.forName("org.postgresql.Driver");
            //System.out.println("class loaded successfully");
        } catch (ClassNotFoundException e) {
            System.out.println("class is not found");
            throw new RuntimeException(e);
        }
        try {
            conn=
DriverManager.getConnection("jdbc:postgresql://localhost:5432/demo","postgr
es","Pravallika@501");
            System.out.println("connection successfully");

        } catch (SQLException e) {

            System.out.println("connection failed");
            e.printStackTrace();
        }

        Scanner sc=new Scanner(System.in);
        boolean user=Operations.userAuthenticate(conn);
        if(user) {
            System.out.println("Welcome to our bank..!!");
            System.out.println("1.Deposit\n2.Withdraw\n3.Balance
Enquiry\n4.Exit");
            System.out.println("Enter your choice");
            int n=sc.nextInt();
            switch (n) {
                case 1:
                    Operations.deposit(conn);
                    break;
                case 2:
                    Operations.withdraw(conn);
                    break;
                case 3:
                    Operations.balanceEnquiry(conn);
                    break;
                case 4:
                    return;
                default:
                    System.out.println("Invalid choice");
            }
        }
        else{
            System.out.println("Invalid credentials");
        }
    }
}
```

Output:

connection successfully

enter userid:

Pravallika30

enter pin:

12345

Welcome to our bank..!!

1.Deposit

2.Withdraw

3.Balance Enquiry

4.Exit

Enter your choice

1

enter account number:

101

enter amount to deposit

3000

Amount is Deposited successfully.

Challenges faced:

1. To connect jdbc I face some issues.
2. To implement withdraw and balance enquiry methods.

Learnings:

1. JDBC Connection.
2. Maven and dependencies

Project Update:

Mostly completed but without GUI means by using console and jdbc connection I develop this project.