**package** com.dxc.user.client;

**public** **class** Main {

**public** Main() {

}

**public** **static** **void** main(String[] args) {

DxcApp app=**new** DxcApp();

app.launchDXCApp();

}

}

package com.dxc.user.client;

import java.util.Scanner;

import com.dxc.training.dao.TrainingDAO;

import com.dxc.training.dao.TrainingDAOImpl;

import com.dxc.user.dao.UserDAO;

import com.dxc.user.dao.UserDAOimpl;

public class DxcApp {

UserDAO userDAO =new UserDAOimpl();

TrainingDAO trainingDAO=new TrainingDAOImpl();

String userName;

String passWord;

Scanner sc=new Scanner(System.in);

public DxcApp() {

}

public void launchDXCApp()

{

while(true)

{

System.out.println("menu");

System.out.println("1. search users");

System.out.println("2.get all trainee details");

System.out.println("3.update percentage");

System.out.println("enter choice");

int choice=sc.nextInt();

switch(choice)

{

case 1:System.out.println("please enter username and passwrod to search");

userName=sc.next();

passWord=sc.next()

if(userDAO.isUserExists(userName, passWord))

{

System.out.println("User successfully authenticated ");

}

else

{

System.out.println("sorry cannot be authenticated");

}

break;

case 2:System.out.println("details of all the trainee are displayed below");

System.out.println(trainingDAO.getAllTrainee());

break;

case 3:trainingDAO.getTrainee();

}

}

}

}

**package** com.dxc.user.model;

**public** **class** Users {

**private** String userName;

**private** String passWord;

**public** Users() {

// **TODO** Auto-generated constructor stub

}

**public** Users(String userName, String passWord) {

**super**();

**this**.userName = userName;

**this**.passWord = passWord;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getPassWord() {

**return** passWord;

}

**public** **void** setPassWord(String passWord) {

**this**.passWord = passWord;

}

@Override

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + ((passWord == **null**) ? 0 : passWord.hashCode());

result = prime \* result + ((userName == **null**) ? 0 : userName.hashCode());

**return** result;

}

@Override

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (getClass() != obj.getClass())

**return** **false**;

Users other = (Users) obj;

**if** (passWord == **null**) {

**if** (other.passWord != **null**)

**return** **false**;

} **else** **if** (!passWord.equals(other.passWord))

**return** **false**;

**if** (userName == **null**) {

**if** (other.userName != **null**)

**return** **false**;

} **else** **if** (!userName.equals(other.userName))

**return** **false**;

**return** **true**;

}

@Override

**public** String toString() {

**return** "Users [userName=" + userName + ", passWord=" + passWord + "]";

}

}

package com.dxc.user.dbcon;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

public DBConnection() {

// TODO Auto-generated constructor stub

}

public static Connection getConnection()

{

try {

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

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Connection connection=null;

try {

connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/dxc","root","root");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return connection;

}

}

**package** com.dxc.user.dao;

**import** com.dxc.user.model.Users;

**public** **interface** UserDAO {

//public void getUser(Users users);

**public** **boolean** isUserExists(String userName, String passWord);

//public Users getUser(Users users);

}

package com.dxc.user.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.dxc.user.dbcon.DBConnection;

public class UserDAOimpl implements UserDAO {

Connection connection=DBConnection.getConnection();

public UserDAOimpl() {

// TODO Auto-generated constructor stub

}

@Override

public boolean isUserExists(String userName, String passWord) {

boolean userExists=false;

PreparedStatement preparedStatement;

try {

preparedStatement=connection.prepareStatement("select\* from users where userName=? and passWord=?");

preparedStatement.setString(1, userName);

preparedStatement.setString(2, passWord);

ResultSet res=preparedStatement.executeQuery();

if(res.next())

{

userExists=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return userExists;

}

}

package com.dxc.training.dao;

import java.util.List;

import com.dxc.training.model.Training;

public interface TrainingDAO

{

public List<Training> getAllTrainee();

public void getTrainee();

}

package com.dxc.training.dao;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import com.dxc.training.model.Training;

import com.dxc.user.dbcon.DBConnection;

public class TrainingDAOImpl implements TrainingDAO {

Connection connection=DBConnection.getConnection();

public TrainingDAOImpl() {

// TODO Auto-generated constructor stub

}

@Override

public List<Training> getAllTrainee()

{

List<Training> allTrainee=new ArrayList<Training>();

try {

Statement stat=connection.createStatement();

ResultSet res=stat.executeQuery("select \* from training");

while(res.next())

{

Training training=new Training();

training.setSapId(res.getInt(1));

training.setEmployeeName(res.getString(2));

training.setStream(res.getString(3));

training.setPercentage(res.getInt(4));

allTrainee.add(training);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return allTrainee;

}

public void getTrainee()

{

try {

Statement stat =connection.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

ResultSet res = stat.executeQuery("select \* from training");

//ResultSetMetaData rsmd = res.getMetaData();

res.beforeFirst();

while(res.next())

{

for (int i = 1; i <= 3; i++)

{

System.out.println(res.getString(i) + " ");

}

Scanner sc=new Scanner(System.in);

System.out.println("enter percentage");

int percentage=sc.nextInt();

res.updateInt(4, percentage);

res.updateRow();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}