**package** com.dxc1.usrval.model;

**public** **class** Training {

**private** **int** SapId;

**private** String EmployeeName;

**private** String Stream;

**private** **int** Percentage=0;

**public** Training() {

**super**();

}

**public** Training(**int** sapId, String employeeName, String stream, **int** percentage) {

**super**();

SapId = sapId;

EmployeeName = employeeName;

Stream = stream;

Percentage = percentage;

}

**public** **int** getSapId() {

**return** SapId;

}

**public** **void** setSapId(**int** sapId) {

SapId = sapId;

}

**public** String getEmployeeName() {

**return** EmployeeName;

}

**public** **void** setEmployeeName(String employeeName) {

EmployeeName = employeeName;

}

**public** String getStream() {

**return** Stream;

}

**public** **void** setStream(String stream) {

Stream = stream;

}

**public** **int** getPercentage() {

**return** Percentage;

}

**public** **void** setPercentage(**int** percentage) {

Percentage = percentage;

}

@Override

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

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

**int** result = 1;

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

result = prime \* result + Percentage;

result = prime \* result + SapId;

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

Training other = (Training) obj;

**if** (EmployeeName == **null**) {

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

**return** **false**;

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

**return** **false**;

**if** (Percentage != other.Percentage)

**return** **false**;

**if** (SapId != other.SapId)

**return** **false**;

**if** (Stream == **null**) {

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

**return** **false**;

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

**return** **false**;

**return** **true**;

}

@Override

**public** String toString() {

**return** "Training [SapId=" + SapId + ", EmployeeName=" + EmployeeName + ", Stream=" + Stream + ", Percentage="

+ Percentage + "]";

}

}

#####################################################################################

**package** com.dxc1.usrval.model;

**public** **class** UserValidation {

**private** String userName;

**private** String password;

**public** UserValidation() {

**super**();

}

**public** UserValidation(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**;

UserValidation other = (UserValidation) 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** "\nUserValidation [userName=" + userName + ", password=" + password + "]";

}

}

#####################################################################################

package com.dxc1.usrval.dbcon;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

public static Connection getConnection() {

try {

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

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection connection=null;

try {

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

} catch (SQLException e) {

e.printStackTrace();

}

return connection;

}

}

#####################################################################################

package com.dxc1.usrval.dao;

import java.util.List;

import com.dxc1.usrval.model.Training;

import com.dxc1.usrval.model.UserValidation;

public interface TrainingsDAO {

public boolean isvalidate(String UserName,String Password);

public List<Training> getallfields();

public void getTrainingmarks();

}

#####################################################################################

package com.dxc1.usrval.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

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 com.dxc1.usrval.dbcon.DBConnection;

import com.dxc1.usrval.model.Training;

import com.dxc1.usrval.model.UserValidation;

public class TrainingsDAOImpl implements TrainingsDAO {

public TrainingsDAOImpl() {

// TODO Auto-generated constructor stub

}

Connection connection = DBConnection.getConnection();

private static final String FETCH\_DETAILS = "select \* from usertable";

**private** **static** **final** String ***FETCH\_TRAINING\_DETAILS*** = "select \* from trainingstable";

public boolean isvalidate(String UserName, String Password) {

// TODO Auto-generated method stub

// UserValidation uservalidation=new UserValidation();

boolean validation = false;

PreparedStatement statement;

try {

statement = connection.prepareStatement("select\* from usertable u where UserName=? and Password=?");

statement.setString(1, UserName);

statement.setString(2, Password);

ResultSet res = statement.executeQuery();

if (res.next())

validation = true;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// statement.setInt(1, productId);

return validation;

}

public List<Training> getallfields() {

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

try {

Statement stat = connection.createStatement();

ResultSet res = stat.executeQuery(FETCH\_TRAINING\_DETAILS);

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));

allmarks.add(training);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return allmarks;

}

@Override

public void getTrainingmarks() {

// TODO Auto-generated method stub

Training training = new Training();

Statement stat;

try {

stat = connection.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_UPDATABLE);

ResultSet res = stat.executeQuery("FETCH\_TRAINING\_DETAILS");

ResultSetMetaData rsmd = res.getMetaData();

res.absolute(0);

while (res.next()) {

for (int i = 1; i <= rsmd.getColumnCount(); i++) {

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

System.out.println("Enter percentage to update");

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

#####################################################################################

**package** com.dxc1.usrval.client;

**import** java.sql.ResultSet;

**import** java.sql.ResultSetMetaData;

**import** java.sql.Statement;

**import** java.util.Scanner;

**import** com.dxc1.usrval.dao.TrainingsDAOImpl;

**import** com.dxc1.usrval.dao.TrainingsDAO;

**import** com.dxc1.usrval.model.Training;

**public** **class** TrainingsApp {

String UserName;

String Password;

TrainingsDAO trainingsDAO;

**public** TrainingsApp() {

**this**.trainingsDAO=**new** TrainingsDAOImpl();

}

**public** **void** LaunchTrainingsApp() {

Scanner scanner=**new** Scanner(System.***in***);

System.***out***.println("Please enter User Name");

UserName=scanner.next();

System.***out***.println("Please enter Password");

Password=scanner.next();

**if**(trainingsDAO.isvalidate(UserName,Password))

{

System.***out***.println("Welcome "+UserName);

**while**(**true**)

{

System.***out***.println("M E N U");

System.***out***.println("1.Display all training records");

System.***out***.println("2.Display records one by one and print the percentage");

System.***out***.println("3.E X I T");

System.***out***.println("Please enter choice between 1 to 3");

**int** choice=scanner.nextInt();

**switch**(choice) {

**case** 1:

System.***out***.println(trainingsDAO.getallfields());

**break**;

**case** 2:

**break**;

**case** 3:

System.***out***.println("Thanks for using my trainings app");

System.*exit*(0);

}

}

}

**else**

{

System.***out***.println("Sorry, username or password cannot be authenticated");

}

}

}

####################################################################################

**package** com.dxc1.usrval.client;

**import** com.dxc1.usrval.client.TrainingsApp;

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

**public** Main() {

}

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

TrainingsApp app=**new** TrainingsApp();

app.LaunchTrainingsApp();

}

}