**1.Main class:**

**package** com.dxc.users.client;

**import** com.dxc.users.DAO.TrainingsDAOImpl;

**import** com.dxc.users.DAO.UserDAOImpl;

**public** **class** MainofAssesment {

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

**boolean** s;

UserDAOImpl u=**new** UserDAOImpl();

s=u.validate();

//if only it is validated

**if**(s) {

TrainingMenu trainingObject=**new** TrainingMenu();

trainingObject.launchMenu();

}

}

}

2. **package** com.dxc.users.client;

**import** com.dxc.users.DAO.TrainingsDAO;

**import** com.dxc.users.DAO.TrainingsDAOImpl;

**import** java.util.Scanner;

**public** **class** TrainingMenu {

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

**int** choice;

TrainingsDAO trainingDAO=**new** TrainingsDAOImpl() ;

**public** TrainingMenu() {

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

}

**public** **void** launchMenu() {

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

System.***out***.println("Menu");

System.***out***.println("1.Display All Records");

System.***out***.println("2.Display Records one by one and Update Percentage");

System.***out***.println("3.Exit");

System.***out***.println("Enter the choice");

choice=scanner.nextInt();

**switch**(choice) {

**case** 1:

System.***out***.println(trainingDAO.displayRecords());

**break**;

**case** 2:

trainingDAO.addPercent();

**break**;

**case** 3:

System.*exit*(0);

**break**;

}

}

}

}

3.DBCONNECTION

package com.dxc.users.connectdb;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class CommonDB {

public static Connection getConnection() {

try {

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

System.out.println("DriverLoaded");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection connection=null;

try {

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

} catch (SQLException e) {

e.printStackTrace();

}

System.out.println("DB Connected");

return connection;

}

}

4. **package** com.dxc.users.DAO;

**import** java.util.List;

**import** com.dxc.users.model.Trainings;

**public** **interface** TrainingsDAO {

**public** List<Trainings> displayRecords();

**public** **void** addPercent();

}

5. **package** com.dxc.users.DAO;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Scanner;

**import** com.dxc.users.connectdb.CommonDB;

**import** com.dxc.users.model.Trainings;

**import** com.mysql.jdbc.Statement;

**public** **class** TrainingsDAOImpl **implements** TrainingsDAO {

**private** **int** percentage;

**private** **static** **final** String ***DISPLAY\_ALL*** = "select \* from training";

**private** **static** **final** String ***UPDATE\_PERCENTAGE***="Update training set percentage=? where sapId=?";

Connection connection = CommonDB.*getConnection*();

List<Trainings> allt=**new** ArrayList<Trainings>();

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

@Override

**public** List<Trainings> displayRecords() {

java.sql.Statement stat=**null**;

**try** {

stat=connection.createStatement();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**try** {

ResultSet res=**null**;

res=stat.executeQuery(***DISPLAY\_ALL***);

**while**(res.next()) {

Trainings training =**new** Trainings();

training.setSap\_Id(res.getInt(1));

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

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

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

allt.add(training);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** allt;

}

@Override

**public** **void** addPercent() {

**try** {

Statement statement=(Statement) connection.createStatement();

ResultSet resultSet=statement.executeQuery(***DISPLAY\_ALL***);

PreparedStatement preparedStatement;

**while**(resultSet.next()) {

System.***out***.println("SapId: "+resultSet.getString(1));

System.***out***.println("Employee Name: "+resultSet.getString(2));

System.***out***.println("Stream: "+resultSet.getString(3));

**if**(resultSet.getInt(4)==0) {

System.***out***.println("Enter The Percentage ");

percentage=sc.nextInt();

preparedStatement=connection.prepareStatement(***UPDATE\_PERCENTAGE***);

preparedStatement.setInt(1, percentage);

preparedStatement.setString(2,resultSet.getString(1));

preparedStatement.executeUpdate();

}

**else** {

System.***out***.println("Percentage: "+resultSet.getInt(4));

}

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

6. **package** com.dxc.users.DAO;

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

**public** **boolean** validate();

**public** **boolean** confirm(String s,String m);

}

7. package com.dxc.users.DAO;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

import com.dxc.users.connectdb.\*;

public class UserDAOImpl implements UserDAO {

private String username;

private String password;

Scanner sc = new Scanner(System.in);

Connection connection = CommonDB.getConnection();

private static final String CHECK\_USERNAME = "select \* from users where userName=? and password1=?";

@Override

public boolean validate() {

boolean t=false;

System.out.println("Enter your credentials");

username = sc.next();

System.out.println("Password:");

password = sc.next();

if (confirm(username, password)) {

System.out.println("correct username and password");

t=true;

} else {

System.out.println("Wrong username and password");

t=false;

}

return t;

}

@Override

public boolean confirm(String username, String password) {

boolean exits = false;

try {

PreparedStatement ps = connection.prepareStatement(CHECK\_USERNAME);

ps.setString(1, username);

ps.setString(2, password);

ResultSet r = ps.executeQuery();

if (r.next())

exits = true;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return exits;

}

}

8. **package** com.dxc.users.model;

**public** **class** Trainings {

**private** **int** sap\_Id;

**private** String empName;

**private** String stream;

**private** **int** percentage;

**public** Trainings(**int** sap\_Id, String empName, String stream, **int** percentage) {

**super**();

**this**.sap\_Id = sap\_Id;

**this**.empName = empName;

**this**.stream = stream;

**this**.percentage = percentage;

}

**public** Trainings() {

**super**();

}

**public** **int** getSap\_Id() {

**return** sap\_Id;

}

**public** **void** setSap\_Id(**int** sap\_Id) {

**this**.sap\_Id = sap\_Id;

}

**public** String getEmpName() {

**return** empName;

}

**public** **void** setEmpName(String empName) {

**this**.empName = empName;

}

**public** String getStream() {

**return** stream;

}

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

**this**.stream = stream;

}

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

**return** percentage;

}

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

**this**.percentage = percentage;

}

@Override

**public** String toString() {

**return** "Trainings [sap\_Id=" + sap\_Id + ", empName=" + empName + ", stream=" + stream + ", percentage=" + percentage

+ "]";

}

@Override

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

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

**int** result = 1;

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

result = prime \* result + percentage;

result = prime \* result + sap\_Id;

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**;

Trainings other = (Trainings) obj;

**if** (empName == **null**) {

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

**return** **false**;

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

**return** **false**;

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

**return** **false**;

**if** (sap\_Id != other.sap\_Id)

**return** **false**;

**if** (stream == **null**) {

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

**return** **false**;

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

**return** **false**;

**return** **true**;

}

}

9. **package** com.dxc.users.model;

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

**private** String username;

**private** String password;

**public** Users(String username, String password) {

**super**();

**this**.username = username;

**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 + "]";

}

}