* ***Create a app called Training Assessment App***

*Create table Users in mysql*

*Username : varchar(20)*

*Password : varchar(20)*

*Insert three records.*

*The aim of this app is to keep track of the assessment scores of the participants.*

*The app should be secured and only authorized users can access this app.*

*See the same output below:*

*Enter your credentials:*

*Username: neha*

*Password: sample*

*[ These credentials should be validated against Users table in mysql]*

1. *Scenario 1: User name cannot be authenticated*
2. *Scenario 1: User successfully authenticated.*

*M E N U*

1. *Display All Training Records*
2. *Display Records one by One and update the percentage*
3. *E X I T*

* *Create a table called* ***Training*** *in MySQL with the following fields,*
  + *Sap\_ID*
  + *Employee\_name*
  + *Stream*
  + *Percentage*
* *Enter some data into the table without Percentage.*

*Write a JDBC program to achieve the following things :*

1. *Write Java code using JDBC driver to display records all the records.*
2. *Write Java code to display records one by one. Each time the record is displayed, prompt user to enter the Percentage and update the record.*
3. If the record already have percentage data (Hint: other than 0) then no updating of the percentage should take place.

NB:

1. Use DAO Design Pattern
2. Please make sure you use Collection Framework classes wherever required.

**CODE**

**MAIN**

**package** com.cms.app.client;

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

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

// **TODO** Auto-generated method stub

TrainingAssessmentApp app = **new** TrainingAssessmentApp();

app.launchTrainingAssessmentApp();

}

}

**TrainingAssessmentApp**

package com.cms.app.client;

import java.util.Scanner;

import com.cms.app.dao.TrainingDAO;

import com.cms.app.dao.UserDAO;

import com.cms.app.dao.impl.TrainingDAOImpl;

import com.cms.app.dao.impl.UserDAOImpl;

import com.cms.app.model.Users;

public class TrainingAssessmentApp {

UserDAO userDAO = new UserDAOImpl();

String password;

String userName;

int sapId;

String employeeName;

String stream;

int percentage;

Scanner scanner = new Scanner(System.in);

public TrainingAssessmentApp() {

// TODO Auto-generated constructor stub

}

public void launchTrainingAssessmentApp() {

Users user = takeUserInput();

if(userDAO.validateUser(user)) {

System.out.println("Valid");

TrainingDAO trainingDAO = new TrainingDAOImpl();

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 update percentage");

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

int choice = 0;

System.out.println("Please enter your choice: (1-3)");

choice = scanner.nextInt();

switch (choice) {

case 1:

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

break;

case 2:

System.out.println("Please update percentage: ");

trainingDAO.getTrainingOneByOne();

System.out.println("Update successfully");

case 3:

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

System.exit(0);

default:

System.out.println("Please enter (1-3)");

}

}

}

else

{

System.out.println("Invalid");

}

}

private Users takeUserInput() {

// TODO Auto-generated method stub

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

userName = scanner.next();

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

password = scanner.next();

Users user = new Users(userName, password);

return user;

}

}

**TrainingDAO**

package com.cms.app.dao;

import java.util.List;

import com.cms.app.model.Training;

public interface TrainingDAO {

public List<Training> getAllTraining();

public List<Training> getTrainingOneByOne();

}

**UserDAO**

*package com.cms.app.dao;*

*import java.util.List;*

*import com.cms.app.model.Training;*

*public interface TrainingDAO {*

*public List<Training> getAllTraining();*

*public List<Training> getTrainingOneByOne();*

*}*

**TrainingDAOImpl**

package com.cms.app.dao.impl;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import com.cms.app.dao.TrainingDAO;

import com.cms.app.dbconnection.DBConnection;

import com.cms.app.model.Training;

public class TrainingDAOImpl implements TrainingDAO {

Connection connection = DBConnection.getConnection();

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

@Override

public List<Training> getAllTraining() {

// TODO Auto-generated method stub

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

try {

Statement statement = connection.createStatement();

ResultSet resultSets = statement.executeQuery(FETCH\_ALL);

while(resultSets.next()) {

Training training = new Training();

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

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

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

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

allTraining.add(training);

}

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

return allTraining;

}

Scanner sc = new Scanner(System.in);

@Override

public List<Training> getTrainingOneByOne() {

// TODO Auto-generated method stub

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

try {

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

ResultSet resultSets = statement.executeQuery(FETCH\_ALL);

while(resultSets.next()) {

System.out.println("Please enter percentage for : " + resultSets.getString(2));

int percentage = sc.nextInt();

resultSets.updateInt(4, percentage);

resultSets.updateRow();

}

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

return updateTraining;

}

}

**UserDAOImpl**

package com.cms.app.dao.impl;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.cms.app.dao.UserDAO;

import com.cms.app.dbconnection.DBConnection;

import com.cms.app.model.Users;

public class UserDAOImpl implements UserDAO {

Connection connection = DBConnection.getConnection();

private static final String VALIDATE\_User = "select \* from users where username = ? and password = ?";

@Override

public boolean validateUser(Users users) {

// TODO Auto-generated method stub

boolean validation = false;

PreparedStatement preparedStatement;

try {

preparedStatement = connection.prepareStatement(VALIDATE\_User);

preparedStatement.setString(1, users.getUserName());

preparedStatement.setString(2, users.getPassword());

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

validation = true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return validation;

}

}

**DBConnection**

package com.cms.app.dbconnection;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnection {

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 (Exception e) {

// TODO: handle exception

}

return connection;

}

}

**Training**

**package** com.cms.app.model;

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

**public** **int** sapId;

**public** String employeeName;

**public** String stream;

**public** **int** percentage;

**public** Training() {

**super**();

}

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

**super**();

**this**.sapId = sapId;

**this**.employeeName = employeeName;

**this**.stream = stream;

**this**.percentage = percentage;

}

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

**return** sapId;

}

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

**this**.sapId = sapId;

}

**public** String getEmployeeName() {

**return** employeeName;

}

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

**this**.employeeName = employeeName;

}

**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** **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** "\n Training [sapId=" + sapId + ", employeeName=" + employeeName + ", stream=" + stream + ", percentage="

+ percentage + "]";

}

}

**Users**

**package** com.cms.app.model;

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

**public** String userName;

**public** String password;

**public** Users() {

**super**();

}

**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** "\n Users [userName=" + userName + ", password=" + password + "]";

}

}