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

**package** com.dxc.training.client;

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

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

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

ValidationApp validationApp = **new** ValidationApp();

validationApp.launchApp();

}

}

**package** com.dxc.training.client;

**import** java.util.Scanner;

**import** com.dxc.training.dao.TrainingDAO;

**import** com.dxc.training.dao.TrainingDAOImpl;

**import** com.dxc.training.dao.ValidateDAO;

**import** com.dxc.training.dao.ValidateDAOImpl;

**import** com.dxc.training.model.User;

**public** **class** ValidationApp {

ValidateDAO validateDAO = **new** ValidateDAOImpl();

TrainingDAO trainingDAO = **new** TrainingDAOImpl();

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

String userName;

String password;

**int** choice = 0;

**public** **void** launchApp() {

User user = takeInput();

**if** (validateDAO.isUserExists(user)) {

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

**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 the percentage");

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

System.***out***.println("Please enter your choice : (1-6)");

choice = scanner.nextInt();

**switch** (choice) {

**case** 1:

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

**break**;

**case** 2:

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

trainingDAO.getOneByOne();

**break**;

**case** 3:

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

System.*exit*(0);

**break**;

**default**:

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

}

}

} **else** {

System.***out***.println(": User name cannot be authenticated");

}

}

**private** User takeInput() {

System.***out***.println("Enter your Credentials");

System.***out***.println("Enter Username :");

userName = scanner.next();

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

password = scanner.next();

User user = **new** User(userName, password);

**return** user;

}

}

**package** com.dxc.training.dao;

**import** java.util.List;

**import** com.dxc.training.model.Training;

**public** **interface** TrainingDAO {

**public** List<Training> getAllDetails();

**public** **void** getOneByOne();

}

**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.dbcon.DbConnection;

**import** com.dxc.training.model.Training;

**public** **class** TrainingDAOImpl **implements** TrainingDAO {

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

Connection connection = DbConnection.*getConnection*();

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

@Override

**public** List<Training> getAllDetails() {

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

**try** {

Statement stat = connection.createStatement();

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

**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));

allTraining.add(training);

}

} **catch** (SQLException e) {

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

e.printStackTrace();

}

**return** allTraining;

}

**public** **void** getOneByOne() {

Training training = **new** Training();

Statement stat;

**try** {

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

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

ResultSetMetaData rsmd = resultSet.getMetaData();

**int** columnsNumber = rsmd.getColumnCount();

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

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

System.***out***.print(resultSet.getString(i) + " ");

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

System.***out***.print(resultSet.getString(i) + " ");

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

**if** (i == 4) {

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

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

resultSet.updateInt(4, percentage);

resultSet.updateRow();

}

}

}

} **catch** (SQLException e) {

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

e.printStackTrace();

}

}

}

**package** com.dxc.training.dao;

**import** java.util.List;

**import** com.dxc.training.model.User;

**public** **interface** ValidateDAO {

**public** **boolean** isUserExists(User user);

}

**package** com.dxc.training.dao;

**import** com.dxc.training.dbcon.DbConnection;

**import** com.dxc.training.model.User;

**import** java.sql.Statement;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**public** **class** ValidateDAOImpl **implements** ValidateDAO {

**private** **static** **final** String ***FETCH\_USER\_ALL*** = "select \* from user where userName=? and password=?";

Connection connection = DbConnection.*getConnection*();

**public** ValidateDAOImpl() {

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

}

@Override

**public** **boolean** isUserExists(User user) {

**boolean** userExists = **false**;

PreparedStatement preparedStatement;

**try** {

preparedStatement = connection.prepareStatement(***FETCH\_USER\_ALL***);

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

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

ResultSet res = preparedStatement.executeQuery();

**if** (res.next()) {

userExists = **true**;

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** userExists;

}

}

package com.dxc.training.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) {

// 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.training.model;

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

**private** **int** sapId;

**private** String employeeName;

**private** String stream;

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

**public** Training() {

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

}

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

+ percentage + "]";

}

}

**package** com.dxc.training.model;

**public** **class** User {

**private** String userName;

**private** String password;

**public** User() {

**super**();

}

**public** User(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**;

User other = (User) 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** "User [\nuserName=" + userName + ", password=" + password + "]";

}

}