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

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

**public** Main() {

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

}

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

UserApp app=**new** UserApp();

app.launchUserApp();

}

}

package com.dxc.training.client;

import java.util.Scanner;

import com.dxc.training.dao.TrainingDAO;

import com.dxc.training.dao.TrainingDAOImpl;

public class TrainingApp {

TrainingDAO trainingDAO= new TrainingDAOImpl();

int choice = 0;

Scanner scanner = new Scanner(System.in);

public TrainingApp() {

// TODO Auto-generated constructor stub

}

public void launchTrainingApp() {

while(true) {

System.out.println("MENU");

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

System.out.println("2. Display Records One by One and update the percentage");

System.out.println("3. EXIT");

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

choice=scanner.nextInt();

switch (choice) {

case 1:

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

break;

case 2:

trainingDAO.display();

break;

case 3:

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

System.exit(0);

break;

default:

System.out.println("Please select option from the menu (1-3)");

}

}

}

}

package com.dxc.training.client;

import java.util.Scanner;

import com.dxc.training.dao.UserDAO;

import com.dxc.training.dao.UserDAOImpl;

import com.dxc.training.model.User;

public class UserApp {

UserDAO userDAO = new UserDAOImpl();

String username;

String password;

Scanner scanner=new Scanner(System.in);

public UserApp() {

// TODO Auto-generated constructor stub

}

public void launchUserApp() {

User user=new User();

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

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

username=scanner.next();

user.setUserName(username);

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

password=scanner.next();

user.setPassword(password);

boolean validated=userDAO.validateUserAndPassword(user);

if(validated) {

System.out.println("Welcome "+username+"!, User successfully authenticated");

TrainingApp trainingApp=new TrainingApp();

trainingApp.launchTrainingApp();

}

else {

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

}

}

}

package com.dxc.training.dao;

import java.util.List;

import com.dxc.training.model.Training;

public interface TrainingDAO {

public List<Training> displayAllRecords();

public void display();//display one by one and update percentage

}

package com.dxc.training.dao;

import java.sql.Connection;

import java.sql.ResultSet;

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{

Connection connection=DBConnection.getConnection();

Scanner scanner=new Scanner(System.in);

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

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

public TrainingDAOImpl() {

// TODO Auto-generated constructor stub

}

@Override

public List<Training> displayAllRecords() {

try {

Statement stat=connection.createStatement();

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

while(res.next()) {

Training data=new Training();

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

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

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

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

allRecords.add(data);

}

}catch(SQLException e) {

e.printStackTrace();

}

return allRecords;

}

@Override

public void display() {

try {

Statement stat1=connection.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet res=stat1.executeQuery(FETCH\_ALL\_RECORDS);

int count=1;

while(res.next()) {

System.out.println("Enter "+count+" Percentage");

int percentage=scanner.nextInt();

res.updateInt(4,percentage);

res.updateRow();

count++;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

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

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

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

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

}

package com.dxc.training.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.dxc.training.dbcon.DBConnection;

import com.dxc.training.model.User;

public class UserDAOImpl implements UserDAO{

Connection connection=DBConnection.getConnection();

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

public UserDAOImpl() {

// TODO Auto-generated constructor stub

}

@Override

public boolean validateUserAndPassword(User user) {

boolean authentic=false;

try {

PreparedStatement stat=connection.prepareStatement(FETCH);

ResultSet res=stat.executeQuery();

while(res.next())

{

if(res.getString(1).equals(user.getUserName()) && res.getString(2).equals(user.getPassword())) {

authentic=true;

break;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return authentic;

}

}

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

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

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

**private** String employeeName;

**private** String stream;

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

**public** Training() {

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

}

**private** 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** "\nTraining [sapId=" + sapId + ", employeeName=" + employeeName + ", stream=" + stream + ", percentage="

+ percentage + "]";

}

}

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

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

**private** String userName;

**private** String password;

**public** User() {

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

}

**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 [userName=" + userName + ", password=" + password + "]";

}

}