MAIN CLASS-  
  
**package** com.dxc.assessment.client;

**import** java.util.Scanner;

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

String a;

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

UsersApp app= **new** UsersApp();

**boolean** status=app.authenticateUserApp();

**if**(status) {

TrainingApp trainingApp= **new** TrainingApp();

trainingApp.launchTrainingApp();

}

**else** {

System.***out***.println("UserId and Password Combination is not correct");

}

}

}

Training App:-

**package** com.dxc.assessment.client;

**import** java.util.Scanner;

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

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

**public** **class** TrainingApp {

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

**public** TrainingApp() {

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

}

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

**int** sapId;

String employeeName;

String stream;

**int** percentage;

**public** **void** launchTrainingApp() {

**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");

**int** choice = 0;

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

choice = scanner.nextInt();

**switch** (choice) {

**case** 1:

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

**break**;

**case** 2:

trainingDAO.getOneByOne();

**break**;

**case** 3:

System.*exit*(0);

**default**:

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

**break**;

}

}

}

}

Users App :-

**package** com.dxc.assessment.client;

**import** java.util.Scanner;

**import** com.dxc.assessment.dao.UsersDAO;

**import** com.dxc.assessment.dao.UsersDAOImpl;

**public** **class** UsersApp {

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

**private** String username;

**private** String password;

**private** **boolean** status=**false**;

UsersDAO dao= **new** UsersDAOImpl();

**public** **boolean** authenticateUserApp() {

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

username= sc.next();

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

password=sc.next();

status= dao.authenticate(username,password);

**return** status;

}

}

Training DAO:-

package com.dxc.assessment.dao;

import java.util.List;

import com.dxc.assessment.model.Training;

public interface TrainingDAO {

public List<Training> getAllProducts();

public void getOneByOne();

}

Training DAO Impl:-

**package** com.dxc.assessment.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.assessment.dbcon.DBConnection;

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

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

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

Connection connection= DBConnection.*getConnection*();

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

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

@Override

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

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

**try** {

Statement stat= connection.createStatement();

ResultSet resultSet= stat.executeQuery(***FETCH\_TABLE\_ALL***);

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

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

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

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

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

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

allRecords.add(training);

}

} **catch** (SQLException e) {

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

e.printStackTrace();

}

**return** allRecords;

}

@Override

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

**try** {

Statement stat =

connection.createStatement(ResultSet.***TYPE\_SCROLL\_INSENSITIVE***,

ResultSet.***CONCUR\_UPDATABLE***);

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

ResultSetMetaData rsmd = res.getMetaData();

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

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

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

}

System.***out***.println("\nEnter the percentage you want to update:-");

percentage= scanner.nextInt();

res.updateInt("percentage", percentage);

res.updateRow();

}

} **catch** (SQLException e) {

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

e.printStackTrace();

}

}

}

Users DAO:-

**package** com.dxc.assessment.dao;

**public** **interface** UsersDAO {

**public** **boolean** authenticate(String username, String password);

}

Users DAO Impl:-

**package** com.dxc.assessment.dao;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** com.dxc.assessment.dbcon.DBConnection;

**public** **class** UsersDAOImpl **implements** UsersDAO {

Connection connection = DBConnection.*getConnection*();

@Override

**public** **boolean** authenticate(String username, String password) {

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

**try** {

PreparedStatement preparedStatement= connection.prepareStatement("select \* from users where username = ? and password =?");

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

ResultSet resultSet= preparedStatement.executeQuery();

**if**(resultSet.next())

**return** **true**;

}

**catch** (SQLException e) {

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

e.printStackTrace();

}

**return** **false**;

}

}

DB Connection:-

**package** com.dxc.assessment.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() {

Connection connection=**null**;

**try** {

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

} **catch** (ClassNotFoundException e1) {

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

e1.printStackTrace();

}

**try** {

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

} **catch** (SQLException e) {

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

e.printStackTrace();

}

;

**return** connection;

}

}

Training:-

**package** com.dxc.assessment.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** String toString() {

**return** "\nTraining [sapId=" + sapId + ", employeeName=" + employeeName + ", stream=" + stream + ", 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**;

}

}

Users :-

**package** com.dxc.assessment.model;

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

**private** String username;

**private** String password;

**public** Users() {

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

}

**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** String toString() {

**return** "Users [username=" + username + ", 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**;

}

}