**package** userModel;

**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** String toString() {

**return** "\nUser [username=" + username + ", password=" + password + "]";

}

}

**package** userModel;

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

**int** sapId;

String employeeName;

String stream;

**int** percentage;

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

**super**();

**this**.sapId = sapId;

**this**.employeeName = employeeName;

**this**.stream = stream;

**this**.percentage = percentage;

}

**public** Training() {

**super**();

}

@Override

**public** String toString() {

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

}

}

package 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** dao;

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

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

}

package dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import dbcon.DBconnection;

public class UserDAOImpl implements UserDAO{

Connection connection = DBconnection.getConnection();

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

private static final String FETCH\_USER = "select\* from user where username=? AND password=?";

@Override

public boolean validate(String username,String password) {

boolean userExists = false;

PreparedStatement prepareStatement;

try {

prepareStatement = connection.prepareStatement(FETCH\_USER);

prepareStatement.setString(1, username);

prepareStatement.setString(2, password);

ResultSet res = prepareStatement.executeQuery();

if (res.next()) {

userExists = true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return userExists;

}

}

package dao;

import java.util.List;

import userModel.Training;

public interface TrainingDAO {

public List<Training> displayRecords();

public void updateRecord(int sapId, int percentage);

}

package dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import dbcon.DBconnection;

import userModel.Training;

public class TrainingDAOImpl implements TrainingDAO{

Connection connection = DBconnection.getConnection();

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

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

private static final String RECORD\_UPDATE\_QUERY = "update training set percentage=? where sapId=?";

@Override

public List<Training> displayRecords() {

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

ResultSet res;

try {

Statement stat = connection.createStatement();

res = stat.executeQuery(FETCH\_RECORD\_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));

allRecords.add(training);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return allRecords;

}

@Override

public void updateRecord(int sapId, int percentage) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(RECORD\_UPDATE\_QUERY);

preparedStatement.setInt(1, percentage);

preparedStatement.setInt(2, sapId);

preparedStatement.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

package client;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Scanner;

import dao.TrainingDAO;

import dao.TrainingDAOImpl;

import dao.UserDAO;

import dao.UserDAOImpl;

import userModel.Training;

public class App {

UserDAO userDAO;

String username;

String password;

int sapId;

String employeeName;

String stream;

int percentage;

int choice;

Scanner scanner = new Scanner(System.in);

public void launchApp() {

UserDAO userDAO = new UserDAOImpl();

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

username = scanner.next();

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

password = scanner.next();

if (userDAO.validate(username, password)) {

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

} else {

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

System.exit(0);

}

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 ");

Scanner scanner = new Scanner(System.in);

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

choice = scanner.nextInt();

TrainingDAO trainingDAO = new TrainingDAOImpl();

switch (choice) {

case 1:

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

break;

case 2:

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

record=trainingDAO.displayRecords();

Iterator<Training>iterator=record.iterator();

while(iterator.hasNext()) {

Training training =new Training();

training=iterator.next();

System.out.println(training.toString());

if(training.getPercentage()==0) {

System.out.println("enter percentage");

percentage=scanner.nextInt();

trainingDAO.updateRecord(training.getSapId(),percentage);

}

else {

System.out.println("percentage already entered");

}

}

}

}

}

}

**package** client;

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

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

App app = **new** App();

app.launchApp();

}

}