CLASS TRAINING APP:

package com.dxcass.user.client;

import java.util.Scanner;

import com.dxcass.user.dao.TrainingDAO;

import com.dxcass.user.dao.TrainingDAOImpl;

import com.dxcass.user.model.Training;

public class TrainingApp {

Scanner scanner=new Scanner(System.in);

Training training;

TrainingDAO trainingDAO = new TrainingDAOImpl();

int sapId;

String employeeName;

String stream;

int percentage;

int choice=0;

public TrainingApp() {

// TODO Auto-generated constructor stub

this.trainingDAO=new TrainingDAOImpl();

}

public void launchapp()

{

while(true)

{

System.out.println("Select one");

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

System.out.println("2.Display one by one");

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

System.out.println("Enter the choice");

choice=scanner.nextInt();

switch(choice) {

case 1:

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

break;

case 2:

trainingDAO.updatePercentage();

break;

case 3:

System.exit(0);

break;

}

}

}

}

TRAINING MAIN

package com.dxcass.user.client;

public class TrainingMain {

public static void main(String[] args) {

TrainingApp app=new TrainingApp();

app.launchapp();

}

}

Training DAO:

package com.dxcass.user.dao;

import java.util.List;

import com.dxcass.user.model.Training;

public interface TrainingDAO {

public List<Training>getAllRecords();

public void updatePercentage();

}

USER DAO:

package com.dxcass.user.dao;

import com.dxcass.user.model.User;

public interface UserDAO {

boolean validate = false;

public boolean validate(User user);

}

TrainingDAOImpl:

package com.dxcass.user.dao;

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.dxcass.user.dbcon.DBConnection;

import com.dxcass.user.model.Training;

import java.sql.PreparedStatement;

import java.sql.Connection;

public class TrainingDAOImpl implements TrainingDAO {

int percentage;

Scanner sc=new Scanner(System.in);

Connection connection = DBConnection.getConnection();

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

private static final String UPDATE\_USER = "Update training set sapId=?,employeeName=?,stream=?,percentage=?";

public List<Training> getAllRecords() {

// TODO Auto-generated method stub

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

try {

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(FETCH\_USERS\_ALL);

while (resultSet.next()) {

Training training = new Training(0, null, null, 0);

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

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

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

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

getAllRecords.add(training);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return getAllRecords;

}

public void updatePercentage() {

// TODO Auto-generated method stub

try {

Statement statement=connection.createStatement();

ResultSet resultSet=statement.executeQuery(FETCH\_USERS\_ALL);

PreparedStatement preparedStatement;

while(resultSet.next()) {

System.out.println("SapId: "+resultSet.getString(1));

System.out.println("Employee Name: "+resultSet.getString(2));

System.out.println("Stream: "+resultSet.getString(3));

if(resultSet.getInt(4)==0) {

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

percentage=sc.nextInt();

preparedStatement=connection.prepareStatement(UPDATE\_USER);

preparedStatement.setInt(1, percentage);

preparedStatement.setString(2,resultSet.getString(1));

preparedStatement.executeUpdate();

}

else {

System.out.println("Percentage: "+resultSet.getInt(4));

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

UserDAOImpl:

package com.dxcass.user.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.dxcass.user.dbcon.DBConnection;

import com.dxcass.user.model.User;

public class UserDAOImpl implements UserDAO{

Connection connection=DBConnection.getConnection();

private static final String Fetch\_User="Select \* from user where username=? and password =?";

public boolean validate(User user) {

// TODO Auto-generated method stub

boolean validate=false;

PreparedStatement preparedStatement;

try {

preparedStatement=connection.prepareStatement(Fetch\_User);

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

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

ResultSet resultSet=preparedStatement.executeQuery();

if(resultSet.next()) {

validate=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return validate;

}

}

DBConnection:

package com.dxcass.user.dbcon;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

public 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");

System.out.println("connected");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Return connection;

}

}

Training:

package com.dxcass.user.model;

public class Training {

private int sapId;

private String employeeName;

private String stream;

private 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 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 [sapId=" + sapId + ", employeeName=" + employeeName + ", stream=" + stream + ", percentage="

+ percentage + "]";

}

}

User:

package com.dxcass.user.model;

public class User {

private String userName;

private String password;

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 + "]";

}

}