The Source code for

Lerner’s academy

Project

Developed By: Yousif Al-Dhfeery

GitHub: <https://github.com/xxmelar-ksaxx>

Project Repository: [Lerner’s Academy](https://github.com/xxmelar-ksaxx/java_EE_STP/tree/main/Learners_Academy/src/main)

**Note:** There will be index sections for every package and class.

Table of Contents

[1- Control package 3](#_Toc77343154)

[1.0- Hibernate\_2 class 3](#_Toc77343155)

[2- DAL package 4](#_Toc77343156)

[2.0- Classes\_DAL class 4](#_Toc77343157)

[2.1- login\_DAL class 4](#_Toc77343158)

[2.2- Student\_DAL class 6](#_Toc77343159)

[2.3- Subject\_DAL class 6](#_Toc77343160)

[2.4- TC\_DAL class 8](#_Toc77343161)

[2.5- Teachers\_DAL class 8](#_Toc77343162)

[3- Entities package 10](#_Toc77343163)

[3.0- Adminstorator class 10](#_Toc77343164)

[3.1- Classes class 10](#_Toc77343165)

[3.2- Students class 12](#_Toc77343166)

[3.3- Subjects class 12](#_Toc77343167)

[3.4- Teachers\_Classes class 13](#_Toc77343168)

[3.5- Teachers class 13](#_Toc77343169)

[4- Servlets package 13](#_Toc77343170)

[4.0- ClassesServlet class 13](#_Toc77343171)

[4.1- createSubjectListner class 13](#_Toc77343172)

[4.2- loginServlet class 13](#_Toc77343173)

[4.3- StudentServlet class 13](#_Toc77343174)

[4.4- TCServlet class 13](#_Toc77343175)

[4.5- TeachersServlet class 13](#_Toc77343176)

[5- Starter package 13](#_Toc77343177)

[5.0- Database\_Initialiser class 13](#_Toc77343178)

[6- Website Pages code 13](#_Toc77343179)

[1.0- Web.xml 13](#_Toc77343180)

[1.1- HTML/JSP Pages 13](#_Toc77343181)

[1.1.0- Class\_info.jsp 13](#_Toc77343182)

[1.1.1- Classes.jsp 13](#_Toc77343183)

[1.1.2- index.jsp 13](#_Toc77343184)

[1.1.3- MainPage.jsp 13](#_Toc77343185)

[1.1.4- Students.jsp 13](#_Toc77343186)

[1.1.5- Subjects.jsp 13](#_Toc77343187)

[1.1.6- TC.jsp 13](#_Toc77343188)

[1.1.7 Teachers.jsp 13](#_Toc77343189)

# 1- Control package

## 1.0- Hibernate\_2 class

package Control;

import java.util.Properties;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.cfg.Environment;

import org.hibernate.service.ServiceRegistry;

import Entities.Adminstorator;

import Entities.Classes;

import Entities.Students;

import Entities.Subjects;

import Entities.Teachers;

import Entities.Teachers\_Classes;

public class Hibernate\_2 {

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory(String DML) {

if (sessionFactory == null) {

try {

Configuration configuration = new Configuration();

// Hibernate settings equivalent to hibernate.cfg.xml's properties

Properties settings = new Properties();

settings.put(Environment.DRIVER, "com.mysql.jdbc.Driver");

settings.put(Environment.URL, "jdbc:mysql://localhost:3306/hibernate"); // Database schema name

settings.put(Environment.USER, "jroot"); //DB User Name

settings.put(Environment.PASS, "1234"); // DB Password

settings.put(Environment.DIALECT, "org.hibernate.dialect.MySQL5Dialect");

settings.put(Environment.SHOW\_SQL, "true");

settings.put(Environment.CURRENT\_SESSION\_CONTEXT\_CLASS, "thread");

if(DML=="create")

settings.put(Environment.HBM2DDL\_AUTO, "create");

configuration.setProperties(settings);

// The Entities

configuration.addAnnotatedClass(Adminstorator.class);

configuration.addAnnotatedClass(Subjects.class);

configuration.addAnnotatedClass(Classes.class);

configuration.addAnnotatedClass(Teachers.class);

configuration.addAnnotatedClass(Teachers\_Classes.class);

configuration.addAnnotatedClass(Students.class);

ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

System.out.println("Hibernate Java Config serviceRegistry created");

sessionFactory = configuration.buildSessionFactory(serviceRegistry);

return sessionFactory;

} catch (Exception e) {

e.printStackTrace();

}

}

return sessionFactory;

}

}

# 2- DAL package

## 2.0- Classes\_DAL class

package DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Classes;

public class Classes\_DAL {

public void saveUser(Classes classes) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the Teacher object

session.save(classes);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List getAllUser() {

Transaction transaction = null;

List<Object[]> listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

String qqr="select C.id, S.name, C.roomNo, C.time from Entities.Classes C, Entities.Subjects S where C.f\_name=S.id";

Query q=session.createQuery(qqr);

listOfUser = q.getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteUser(int id) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Classes user = session.get(Classes.class, id);

if (user != null) {

session.delete(user);

System.out.println("user is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List get\_Class\_Info(String class\_id) {

Transaction transaction = null;

List<Object[]> listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

String xx="1";

String qqr="select distinct S.name, C.roomNo,T.name ,count(ST.id) from Entities.Classes C, Entities.Students ST, Entities.Subjects S, Entities.Teachers T, Entities.Teachers\_Classes TC where S.id=C.f\_name and ST.class1=C.id and ST.class1='"+class\_id+"' and C.id=TC.c\_id";

Query q=session.createQuery(qqr);

listOfUser = q.getResultList();

System.err.println("The value is: -"+class\_id+"-");

for(Object[] c: listOfUser)

{

System.out.println("###$$$ : "+c[0]);

System.out.println("###$$$ : "+c[1]);

System.out.println("###$$$ : "+c[2]);

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

}

## 

## 2.1- login\_DAL class

package DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Adminstorator;

import Entities.Subjects;

public class login\_DAL {

public boolean loginValidation(String name,String password) {

Transaction transaction = null;

boolean state=false;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

String t="Adminstorator A";

String query="select A.name, A.password from "+t;

Query q=session.createQuery(query);

List<Object[]> oa=q.list();

String qname="";

String qpass="";

for(Object[] u:oa) {

qname=(String)u[0];

qpass=(String)u[1];

}

if(qname.equals(name) && qpass.equals(password) ) {

state=true;

}

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

// return the if found

return state;

}

}

## 

## 2.2- Student\_DAL class

package DAL;

import java.util.Iterator;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Students;

import Entities.Subjects;

import Entities.Teachers;

public class Student\_DAL {

public void saveUser(Students student) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the Teacher object

session.save(student);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List getAllUser() {

Transaction transaction = null;

List<Object[]> listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// get an user object

// String shq="from Entities.Students";

// Query<Students> q = session.createQuery(shq);

// List<Students> res=q.list();

listOfUser = session.createQuery("select distinct ST.id, ST.name, ST.email, ST.jender,ST.age,ST.role, S.name from Entities.Students ST, Entities.Classes C, Entities.Subjects S where ST.class1=S.id").getResultList();

for(Object[] c:listOfUser) {

System.out.println("@@@@"+c[0]);

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteUser(int id) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Students user = session.get(Students.class, id);

if (user != null) {

session.delete(user);

System.out.println("user is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

}

## 

## 2.3- Subject\_DAL class

package DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Subjects;

public class Subject\_DAL {

public void saveUser(Subjects subject, String DML) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory(DML).openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.save(subject);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List < Subjects > getAllUser() {

Transaction transaction = null;

List < Subjects > listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("create-drop").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// get an user object

String shq="from Entities.Subjects";

Query<Subjects> q = session.createQuery(shq);

List<Subjects> res=q.list();

listOfUser = session.createQuery("from Entities.Subjects").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteUser(int id, String DML) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory(DML).openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Subjects user = session.get(Subjects.class, id);

if (user != null) {

session.delete(user);

System.out.println("user is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

//@SuppressWarnings("unchecked")

public Subjects getSubject(String name) {

Transaction transaction = null;

Subjects user=null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

String shq="select S.id from Entities.Subjects S where name='"+name+"'";

Query q = session.createQuery(shq);

List<Object> res=q.list();

int x=(Integer)res.get(0);

user = (Subjects)session.get(Subjects.class, x);

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

}

e.printStackTrace();

}

return user;

}

}

## 

## 2.4- TC\_DAL class

package DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Classes;

import Entities.Subjects;

import Entities.Teachers;

import Entities.Teachers\_Classes;

public class TC\_DAL {

public void saveUser(Teachers\_Classes TC) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the Teacher object

session.save(TC);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List getAllUser() {

Transaction transaction = null;

List<Object[]> listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

String qqr="select TC.tc\_id, T.name, S.name ,C.roomNo, C.time from Teachers T,Classes C, Teachers\_Classes TC, Subjects S where TC.t\_id=T.id and TC.c\_id=C.id and C.f\_name=S.id";

Query q=session.createQuery(qqr);

listOfUser = q.getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteUser(int id) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Teachers\_Classes tc = session.get(Teachers\_Classes.class, id);

if (tc != null) {

session.delete(tc);

System.out.println("TC is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

//@SuppressWarnings("unchecked")

public Object getTC\_E\_info(int id, String type) {

Transaction transaction = null;

Teachers teacher=null;

Classes class1=null;

Object o=null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

if(type.equals("T")) {

teacher = (Teachers)session.get(Teachers.class, id);

o=teacher;

}

else if (type.equals("C")) {

class1 = (Classes)session.get(Classes.class, id);

o=class1;

}

else {

System.out.println("!! Get TC Entity info Error !!");

}

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

}

e.printStackTrace();

}

return o;

}

}

## 

## 2.5- Teachers\_DAL class

package DAL;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import Control.Hibernate\_2;

import Entities.Subjects;

import Entities.Teachers;

public class Teachers\_DAL {

public void saveUser(Teachers teacher) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the Teacher object

session.save(teacher);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List < Teachers > getAllUser() {

Transaction transaction = null;

List < Teachers > listOfUser = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// get an user object

String shq="from Entities.Teachers";

Query<Subjects> q = session.createQuery(shq);

List<Subjects> res=q.list();

listOfUser = session.createQuery("from Entities.Teachers").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

//transaction.rollback();

System.out.println("transaction closed!!");

}

e.printStackTrace();

}

return listOfUser;

}

public void deleteUser(int id) {

Transaction transaction = null;

try (Session session = Hibernate\_2.getSessionFactory("none").openSession()) {

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

Teachers user = session.get(Teachers.class, id);

if (user != null) {

session.delete(user);

System.out.println("user is deleted");

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

}

# 3- Entities package

## 3.0- Adminstorator class

package Entities;

import javax.persistence.\*;

@Entity

public class Adminstorator {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String password;

public Adminstorator() {

System.out.println("Admnistorator defult constructure runs..");

}

public Adminstorator(String name, String password) {

super();

this.name = name;

this.password = password;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Adminstorator [id=" + id + ", name=" + name + ", password=" + password + "]";

}

}

## 

## 3.1- Classes class

package Entities;

import java.util.Set;

import javax.persistence.\*;

@Entity

@Table(name ="classes")

public class Classes {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@ManyToOne

@JoinColumn(name="subject\_id",referencedColumnName="id")

private Subjects f\_name; // foreign key of the subject entity id

private int roomNo;

private int time;

// for @ManyToOne relation

@OneToMany(mappedBy = "c\_id")

Set<Teachers\_Classes> registerd\_classes;

public Classes() {

System.out.println("Classes defult constructure runs..");

}

public Classes(Subjects f\_name, int roomNo, int time) {

super();

this.f\_name = f\_name;

this.roomNo = roomNo;

this.time = time;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public Subjects getF\_name() {

return f\_name;

}

public void setF\_name(Subjects f\_name) {

this.f\_name = f\_name;

}

public int getRoomNo() {

return roomNo;

}

public void setRoomNo(int roomNo) {

this.roomNo = roomNo;

}

public int getTime() {

return time;

}

public void setTime(int time) {

this.time = time;

}

@Override

public String toString() {

return "Classes [id=" + id + ", f\_name=" + f\_name + ", roomNo=" + roomNo + ", time=" + time + "]";

}

}

## 

## 3.2- Students class

package Entities;

import javax.persistence.\*;

@Entity

public class Students {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String email;

private String jender;

private int age;

private String role;

private int class1;

public Students() {

System.out.println("Students defult constructure runs..");

}

public Students(String name, String email, String jender, int age, String role, int class1) {

super();

this.name = name;

this.email = email;

this.jender = jender;

this.age = age;

this.role = role;

this.class1 = class1;

}

public Students(String name, String role, int class1) {

super();

this.name = name;

this.role = role;

this.class1 = class1;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getJender() {

return jender;

}

public void setJender(String jender) {

this.jender = jender;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public int getClass1() {

return class1;

}

public void setClass1(int class1) {

this.class1 = class1;

}

@Override

public String toString() {

return "Students [id=" + id + ", name=" + name + ", email=" + email + ", jender=" + jender + ", age=" + age

+ ", role=" + role + ", class1=" + class1 + "]";

}

}

## 

## 3.3- Subjects class

package Entities;

import javax.persistence.\*;

@Entity

@Table(name = "subjects")

public class Subjects {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private int credits;

public Subjects() {

System.out.println("Subjects defult constructure runs..");

}

public Subjects(String name, int credits) {

super();

this.name = name;

this.credits = credits;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getCredits() {

return credits;

}

public void setCredits(int credits) {

this.credits = credits;

}

@Override

public String toString() {

return "Subjects [id=" + id + ", name=" + name + ", credits=" + credits + "]";

}

}

## 

## 3.4- Teachers\_Classes class

package Entities;

import javax.persistence.\*;

@Entity

public class Teachers\_Classes {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int tc\_id;

@ManyToOne

@JoinColumn(name = "teachers\_id")

private Teachers t\_id;

@ManyToOne

@JoinColumn(name = "class\_id")

private Classes c\_id;

public Teachers\_Classes() {

System.out.println("Teachers\_Classes defult constructure runs..");

}

public Teachers\_Classes(Teachers t\_id, Classes c\_id) {

super();

this.t\_id = t\_id;

this.c\_id = c\_id;

}

public Teachers getT\_id() {

return t\_id;

}

public void setT\_id(Teachers t\_id) {

this.t\_id = t\_id;

}

public Classes getC\_id() {

return c\_id;

}

public void setC\_id(Classes c\_id) {

this.c\_id = c\_id;

}

@Override

public String toString() {

return "Teachers\_Classes [t\_id=" + t\_id + ", c\_id=" + c\_id + "]";

}

}

## 

## 3.5- Teachers class

package Entities;

import java.util.Set;

import javax.persistence.\*;

@Entity

public class Teachers {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String email;

private String jender;

private int age;

private String role;

// for @ManyToOne relation

@OneToMany(mappedBy = "t\_id")

Set<Teachers\_Classes> registerd\_classes;

public Teachers() {

System.out.println("Teachers defult constructure runs..");

}

public Teachers(String name, String email, String jender, int age, String role) {

super();

this.name = name;

this.email = email;

this.jender = jender;

this.age = age;

this.role = role;

}

public Teachers(String name, String role) {

super();

this.name = name;

this.role = role;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getJender() {

return jender;

}

public void setJender(String jender) {

this.jender = jender;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

@Override

public String toString() {

return "Teachers [id=" + id + ", name=" + name + ", email=" + email + ", jender=" + jender + ", age=" + age

+ ", role=" + role + "]";

}

}

# 4- Servlets package

## 4.0- ClassesServlet class

package Servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import DAL.Classes\_DAL;

import DAL.Student\_DAL;

import DAL.Subject\_DAL;

import Entities.Classes;

import Entities.Students;

import Entities.Subjects;

public class ClassesServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

Classes\_DAL cDal;

Subject\_DAL sDal;

public void init() {

cDal = new Classes\_DAL();

sDal = new Subject\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/insert\_class":

insertUser(request, response);

break;

case "/list\_classes":

//listUser(request, response);

getSubjectsnames(request, response);

break;

case "/delete\_class":

deleteSubject(request, response);

break;

case "/get\_subject\_names":

getSubjectsnames(request, response);

break;

case "/class\_info":

getClassInfo(request, response);

break;

case "/class\_info\_refrech":

getClassInfo\_refresh(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

String name = request.getParameter("subject\_name");

String room = request.getParameter("room\_no");

String time = request.getParameter("class\_time");

if( name==null || room==null || time==null) {

request.setAttribute("TheState", false);

}

else {

Subjects subject=sDal.getSubject(name);

Classes classes = new Classes(subject,Integer.parseInt(room),Integer.parseInt(time));

cDal.saveUser(classes); // 2nd argument for the DML type (CRUD)

}

try {

//listUser(request, response);

getSubjectsnames(request, response);

} catch (Exception e) {

System.out.println("list teachers (try block) error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Object[]> listClasses = cDal.getAllUser();

request.setAttribute("listClasses", listClasses);

RequestDispatcher dispatcher = request.getRequestDispatcher("Classes.jsp");

dispatcher.forward(request, response);

//response.sendRedirect("Classes.jsp");

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

int id = Integer.parseInt(request.getParameter("id"));

cDal.deleteUser(id);

try {

getSubjectsnames(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

private void getSubjectsnames(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Subjects > listSubjects = sDal.getAllUser();

request.setAttribute("listSubjects", listSubjects);

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

private void getClassInfo(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Object[]> listClasses = cDal.getAllUser();

request.setAttribute("lci", listClasses);

RequestDispatcher dispatcher = request.getRequestDispatcher("Class\_info.jsp");

dispatcher.forward(request, response);

}

private void getClassInfo\_refresh(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

String name = request.getParameter("class\_inf\_id");

System.out.println(" $$$ prameter value= -"+name+"-");

List class\_info = cDal.get\_Class\_Info(name);

request.setAttribute("class\_info", class\_info);

try {

getClassInfo(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

}

## 

## 4.1- createSubjectListner class

package Servlets;

import java.io.IOException;

import java.sql.SQLException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.List;

import DAL.Subject\_DAL;

import Entities.Subjects;

public class createSubjectListner extends HttpServlet {

private static final long serialVersionUID = 1L;

private Subject\_DAL sDal;

public void init() {

sDal = new Subject\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("1");

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("2");

String action = request.getServletPath();

System.out.println(action);

try {

switch (action) {

case "/insert":

insertUser(request, response);

break;

case "/list":

listUser(request, response);

break;

case "/delete":

deleteSubject(request, response);

break;

case "/test\_1":

test\_1(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

System.out.println("3");

String name = request.getParameter("subject\_input");

String email = request.getParameter("credits\_input");

if(name.equals("") || email.equals("")) {

request.setAttribute("TheState", false);

}

else {

Subjects newUser = new Subjects(name,Integer.parseInt(email));

System.out.println("4");

sDal.saveUser(newUser,"update"); // 2nd argument for the DML type (CRUD)

}

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list subjects error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Subjects > listUser = sDal.getAllUser();

request.setAttribute("listUser", listUser);

RequestDispatcher dispatcher = request.getRequestDispatcher("Subjects.jsp");

dispatcher.forward(request, response);

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

System.out.println("in delete ");

int id = Integer.parseInt(request.getParameter("id"));

sDal.deleteUser(id, "none");

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list subjects error!!");

}

}

private void test\_1(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

System.out.println("in s test 1 ");

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list subjects error!!");

}

}

}

## 

## 4.2- loginServlet class

package Servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import DAL.login\_DAL;

public class loginServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private login\_DAL loginDLA;

public void init() {

loginDLA = new login\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

System.out.println("Inside the Login Servlet");

String name = request.getParameter("username");

String password = request.getParameter("password");

boolean state=loginDLA.loginValidation(name, password);

System.out.println("the login state is: "+state);

if(state==true) {

RequestDispatcher dispatcher = request.getRequestDispatcher("MainPage.jsp");

dispatcher.forward(request, response);

}

else {

request.setAttribute("TheState", state);

RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");

dispatcher.forward(request, response);

}

}

}

## 

## 4.3- StudentServlet class

package Servlets;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import DAL.Classes\_DAL;

import DAL.Student\_DAL;

import DAL.Teachers\_DAL;

import Entities.Classes;

import Entities.Students;

import Entities.Subjects;

import Entities.Teachers;

public class StudentServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

Student\_DAL sDal;

Classes\_DAL cDal;

public void init() {

sDal = new Student\_DAL();

cDal = new Classes\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/insert\_student":

insertUser(request, response);

break;

case "/list\_student":

//listUser(request, response);

getclassesnames(request, response);

break;

case "/delete\_student":

deleteSubject(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

System.out.println("3");

String name = request.getParameter("name");

String email = request.getParameter("email");

String gender = request.getParameter("jender");

String age = request.getParameter("age");

String role = request.getParameter("role");

String class1 = request.getParameter("class1");

if( name.equals("") || age.equals("") ) {

request.setAttribute("TheState", false);

}

else {

Students student = new Students(name,email,gender,Integer.parseInt(age),role,Integer.parseInt(class1));

sDal.saveUser(student); // 2nd argument for the DML type (CRUD)

}

try {

getclassesnames(request, response);

} catch (Exception e) {

System.out.println("list teachers (try block) error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Object[]> listUser = sDal.getAllUser();

request.setAttribute("listUser", listUser);

RequestDispatcher dispatcher = request.getRequestDispatcher("Students.jsp");

dispatcher.forward(request, response);

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

System.out.println("in delete ");

int id = Integer.parseInt(request.getParameter("id"));

sDal.deleteUser(id);

try {

getclassesnames(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

private void getclassesnames(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Classes> listclasses = cDal.getAllUser();

request.setAttribute("listclasses", listclasses);

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

}

## 

## 4.4- TCServlet class

package Servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import DAL.Classes\_DAL;

import DAL.Subject\_DAL;

import DAL.TC\_DAL;

import DAL.Teachers\_DAL;

import Entities.Classes;

import Entities.Subjects;

import Entities.Teachers;

import Entities.Teachers\_Classes;

/\*\*

\* Servlet implementation class TCServlet

\*/

public class TCServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

Classes\_DAL cDal;

Teachers\_DAL tDal;

TC\_DAL tcDal;

public void init() {

cDal = new Classes\_DAL();

tDal = new Teachers\_DAL();

tcDal = new TC\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/inserst\_TC":

insertUser(request, response);

break;

case "/list\_TC":

//listUser(request, response);

getTCnames(request, response);

break;

case "/delete\_tc":

deleteSubject(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

String t\_name\_id = request.getParameter("t\_name");

String class1\_id = request.getParameter("c\_name");

if(t\_name\_id==null || class1\_id==null) {

request.setAttribute("TheState", false);

//response.sendRedirect("list\_TC");

}

else {

Object teacher=tcDal.getTC\_E\_info(Integer.parseInt(t\_name\_id),"T");

Object class1=tcDal.getTC\_E\_info(Integer.parseInt(class1\_id),"C");

Teachers\_Classes teachers = new Teachers\_Classes((Teachers)teacher, (Classes)class1);

tcDal.saveUser(teachers); // 2nd argument for the DML type (CRUD)

}

try {

//listUser(request, response);

getTCnames(request, response);

} catch (Exception e) {

System.out.println("list teachers (try block) error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Object[]> list\_tc = tcDal.getAllUser();

request.setAttribute("list\_tc", list\_tc);

RequestDispatcher dispatcher = request.getRequestDispatcher("TC.jsp");

dispatcher.forward(request, response);

//response.sendRedirect("Classes.jsp");

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

int id = Integer.parseInt(request.getParameter("id"));

tcDal.deleteUser(id);

try {

getTCnames(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

private void getTCnames(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Teachers > listTnames = tDal.getAllUser();

request.setAttribute("listteachers", listTnames);

List < Classes > listCnames = cDal.getAllUser();

request.setAttribute("listclasses", listCnames);

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

}

## 

## 4.5- TeachersServlet class

package Servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.Arrays;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import DAL.Teachers\_DAL;

import Entities.Subjects;

import Entities.Teachers;

public class TeachersServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

Teachers\_DAL tDal;

public void init() {

tDal = new Teachers\_DAL();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/insert\_teacher":

insertUser(request, response);

break;

case "/list\_teacher":

listUser(request, response);

break;

case "/delete\_teacher":

deleteSubject(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

System.out.println("3");

String name = request.getParameter("name");

String email = request.getParameter("email");

String gender = request.getParameter("gender");

String age = request.getParameter("age");

String role = request.getParameter("role");

if( name.equals("") || age.equals("") ) {

request.setAttribute("TheState", false);

}

else {

Teachers teacher = new Teachers(name,email,gender,Integer.parseInt(age),role);

tDal.saveUser(teacher); // 2nd argument for the DML type (CRUD)

}

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list teachers (try block) error!!");

}

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List < Teachers > listUser = tDal.getAllUser();

request.setAttribute("listUser", listUser);

RequestDispatcher dispatcher = request.getRequestDispatcher("Teachers.jsp");

dispatcher.forward(request, response);

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

System.out.println("in delete ");

int id = Integer.parseInt(request.getParameter("id"));

tDal.deleteUser(id);

try {

listUser(request, response);

} catch (Exception e) {

System.out.println("list teachers error!!");

}

}

}

# 5- Starter package

## 5.0- Database\_Initialiser class

package Starter;

import org.hibernate.Session;

import org.hibernate.Transaction;

import Control.Hibernate\_2;

import Entities.Adminstorator;

import Entities.Classes;

import Entities.Subjects;

public class Database\_Initialiser {

static Session session = Hibernate\_2.getSessionFactory("create").openSession();

public void createAdminstorator(String name,String password) {

session.save(new Adminstorator(name, password));

}

public Subjects createSubjects(String name,int credits) {

Subjects s=new Subjects(name, credits);

session.save(s);

return s;

}

public void createclasses(Subjects name,int roomno,int time) {

session.save(new Classes(name, roomno, time));

}

public void send() {

Transaction transaction = null;

try {

transaction = session.beginTransaction();

//session.save(a);

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

public static void main(String[] args) {

Database\_Initialiser dba=new Database\_Initialiser();

dba.createAdminstorator("yousif", "1234"); // The Admin Credentials

dba.send();

}

}

# 6- Website Pages code

## 1.0- Web.xml

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd" id="WebApp\_ID" version="3.0">

<display-name>Learners\_Academy</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<description></description>

<display-name>Subjects\_Servlet</display-name>

<servlet-name>S\_S</servlet-name>

<servlet-class>Servlets.createSubjectListner</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>S\_S</servlet-name>

<url-pattern>/insert</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>S\_S</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>S\_S</servlet-name>

<url-pattern>/delete</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>S\_S</servlet-name>

<url-pattern>/test\_1</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>Login\_Servlet</display-name>

<servlet-name>L\_S</servlet-name>

<servlet-class>Servlets.loginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>L\_S</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>Teachers\_Servlet</display-name>

<servlet-name>T\_S</servlet-name>

<servlet-class>Servlets.TeachersServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>T\_S</servlet-name>

<url-pattern>/insert\_teacher</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>T\_S</servlet-name>

<url-pattern>/list\_teacher</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>T\_S</servlet-name>

<url-pattern>/delete\_teacher</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>Student\_Servlet</display-name>

<servlet-name>Student\_S</servlet-name>

<servlet-class>Servlets.StudentServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Student\_S</servlet-name>

<url-pattern>/insert\_student</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Student\_S</servlet-name>

<url-pattern>/list\_student</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Student\_S</servlet-name>

<url-pattern>/delete\_student</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>Classes\_Servlet</display-name>

<servlet-name>Classes\_S</servlet-name>

<servlet-class>Servlets.ClassesServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/insert\_class</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/list\_classes</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/delete\_class</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/get\_subject\_names</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/class\_info</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Classes\_S</servlet-name>

<url-pattern>/class\_info\_refrech</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>TC\_Servlet</display-name>

<servlet-name>TC\_S</servlet-name>

<servlet-class>Servlets.TCServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>TC\_S</servlet-name>

<url-pattern>/inserst\_TC</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>TC\_S</servlet-name>

<url-pattern>/list\_TC</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>TC\_S</servlet-name>

<url-pattern>/delete\_tc</url-pattern>

</servlet-mapping>

</web-app>

## 1.1- HTML/JSP Pages

### 1.1.0- Class\_info.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>Class informations</h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="class\_info\_refrech">

Class Number:<br/>

<select id="Class" name="class\_inf\_id">

<c:if test="${lci != null}">

<c:forEach var="user" items="${lci}">

<option value="${user[0]}"><c:out value="${user[1]} / RoomNo: ${user[2]}" /></option>

</c:forEach>

</c:if>

</select>

<input type="submit" value="Refresh\_1">

</form>

<div align="center">

<table border="1" cellpadding="10">

<caption><h2>Class information</h2></caption>

<tr>

<th>Subject</th>

<th>Room No.</th>

<th>Teacher</th>

<th>No of Students</th>

</tr>

<c:forEach var="user" items="${class\_info}">

<tr>

<td><c:out value="${user[0]}" /></td>

<td><c:out value="${user[1]}" /></td>

<td><c:out value="${user[2]}" /></td>

<td><c:out value="${user[3]}" /></td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

### 

### 1.1.1- Classes.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>List/Add Classes </h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="insert\_class" >

<fieldset>

Subject:<br/>

<select id="Class" name="subject\_name">

<c:if test="${listSubjects != null}">

<c:forEach var="user" items="${listSubjects}">

<option><c:out value="${user.name}" /></option>

</c:forEach>

</c:if>

</select>

<br/>

Room Number:<br/>

<select id="Class" name="room\_no">

<option value="01">01</option>

<option value="02">02</option>

<option value="03">03</option>

<option value="04">04</option>

</select>

<br/>

Class Time:<br/>

<select id="Class" name="class\_time">

<option >10</option>

<option >12</option>

<option >02</option>

<option >04</option>

</select>

</fieldset>

<br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:red'>Empty selection!!</span><%

}

%>

<br/>

<input type="submit" value="Create">

</form>

<div align="center">

<form action="list\_classes" >

<input type="submit" value="Refresh">

</form>

<table border="1" cellpadding="10">

<caption><h2>List of Classes</h2></caption>

<tr>

<th>ID</th>

<th>Subject</th>

<th>RoomNo</th>

<th>Time</th>

</tr>

<c:forEach var="user" items="${listClasses}">

<tr>

<td><c:out value="${user[0]}" /></td>

<td><c:out value="${user[1]}" /></td>

<td><c:out value="${user[2]}" /></td>

<td><c:out value="${user[3]}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete\_class?id=<c:out value='${user[0]}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

### 

### 1.1.2- index.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

<style type="text/css">

body{

background-color: #540B0E;

}

div{

text-align: center;

}

#span1{

display: inline-block;

text-align: left;

color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;

padding: 2px 10px;

padding-bottom: 20px;

}

input[type=text],input[type=password]{

width: 200px;

padding: 3px 10px;

margin: 8px 0;

box-sizing: border-box;

font-family: Segoe UI;

font-size: 15px;

}

input[type=button], input[type=submit], input[type=reset] {

background-color: #197278;

border: none;

color: #EDDDD4;

padding: 3px 30px;

text-decoration: none;

margin: 4px 2px;

cursor: pointer;

font-family: Segoe UI;

font-size: 20px;

}

h1{

text-align: center; color: #EDDDD4;

font-family: Segoe UI;

font-size: 40px;

}

hr.new5 {

margin-top: -30px;

margin-left: 20%;

margin-right: 20%;

border: 1px solid #EDDDD4;

border-radius: 10px;

}

#myinfo{

text-align: left;

color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;

margin-top: 10%;

}

</style>

</head>

<body>

<h1>Adminstrator Login</h1><br/>

<hr class="new5">

<div style="margin-top: 8%">

<form action="login" method="post" >

<span id="span1">

User Name <br/>

<input type="text" name="username" placeholder="Enter username" value="yousif"><br/>

Password <br/>

<input type="password" name="password" placeholder="Enter password" value="1234"><br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:#EDDDD4; text-shadow: 2px 2px #000000;'>Invalid Credentials!!</span><%

}

%>

</span>

<div style="text-align: center; margin-top: -13px">

<input type="submit" value="Login">

</div>

</form>

</div>

<div id="myinfo">

@ Developed By: Yousif Al-Dhfeery <br/>

@ GitHub: https://github.com/xxmelar-ksaxx

</div>

</body>

</html>

### 

### 1.1.3- MainPage.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

<style>

body{

background-color: #540B0E;

}

a:link, a:visited {

background-color: #197278;

color: #EDDDD4;

padding: 7px 1px;

text-align: center;

text-decoration: none;

display: inline-block;

cursor: pointer;

font-family: Segoe UI;

font-size: 20px;

width: 17em;

}

a:hover, a:active {

background-color: #14868E;

color: #EDDDD4;

}

h1{

text-align: center; color: #EDDDD4;

font-family: Segoe UI;

font-size: 40px;

}

hr.new5 {

margin-top: -30px;

margin-left: 20%;

margin-right: 20%;

border: 1px solid #EDDDD4;

border-radius: 10px;

}

#myinfo{

text-align: left;

color: #EDDDD4;

font-family: Segoe UI;

font-size: 20px;

}

</style>

</head>

<body>

<div style="text-align: center;">

<h1>Main Menu</h1> <br/>

<hr class="new5">

<br/>

<h2><a href="list" > List/Add Subject </a><br/></h2>

<h2><a href="get\_subject\_names" > List/Add Classes </a><br/></h2>

<h2><a href="list\_teacher" > List/Add Teachers </a><br/></h2>

<h2><a href="list\_TC" > List/Assign The Teachers Classes </a><br/></h2>

<h2><a href="list\_student" > List/Add Student </a><br/></h2>

<h2><a href="class\_info\_refrech" > Class information </a><br/></h2>

</div>

<div id="myinfo">

@ Developed By: Yousif Al-Dhfeery <br/>

@ GitHub: https://github.com/xxmelar-ksaxx

</div>

</body>

</html>

### 

### 1.1.4- Students.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>List/Add Student</h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="insert\_student">

<fieldset>

Name:<br/> <input type="text" name="name" placeholder="inter name">

<br/>

Email:<br/> <input type="email" name="email" placeholder="inter email">

<br/>

Gender:

<br/>

<select name="jender">

<option value="male">male</option>

<option value="female">female</option>

</select>

<br/>

Age:<br/> <input type="text" name="age" placeholder="age"><br/>

Class Number:<br/>

<select id="Class" name="class1">

<c:if test="${listclasses != null}">

<c:forEach var="user" items="${listclasses}">

<option value="${user[0]}"><c:out value="${user[1]} / RoomNo: ${user[2]}" /></option>

</c:forEach>

</c:if>

</select>

<br/>

Role:<br/>

<select name="role">

<option value="Student">Studnt</option>

</select>

</fieldset><br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:red'>Empty fields!!</span><%

}

%>

<br/>

<input type="submit" value="save">

</form>

<div align="center">

<form action="list\_student" >

<input type="submit" value="Refresh">

</form>

<table border="1" cellpadding="10">

<caption><h2>List of Students</h2></caption>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Gender</th>

<th>Age</th>

<th>Role</th>

<th>Class</th>

</tr>

<c:forEach var="user" items="${listUser}">

<tr>

<td><c:out value="${user[0]}" /></td>

<td><c:out value="${user[1]}" /></td>

<td><c:out value="${user[2]}" /></td>

<td><c:out value="${user[3]}" /></td>

<td><c:out value="${user[4]}" /></td>

<td><c:out value="${user[5]}" /></td>

<td><c:out value="${user[6]}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete\_student?id=<c:out value='${user[0]}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

### 

### 1.1.5- Subjects.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>List/Add Subject</h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="insert" method="post" style="border-style: solid; border-color:black; padding: 2%;">

Subject Name: <input type="text" name="subject\_input"><br/>

Subject Credits: <input type="number" name="credits\_input"><br/><br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:red'>Empty fields!!</span><%

}

%>

<br/>

<input type="Submit" value="Save" style="margin-left: 10%">

</form>

<br/>

<br/>

<div align="center">

<form action="list" >

<input type="submit" value="Refresh">

</form>

<table border="1" cellpadding="10">

<caption><h2>List of Subjects</h2></caption>

<tr>

<th>ID</th>

<th>Name</th>

<th>Credits</th>

</tr>

<c:forEach var="user" items="${listUser}">

<tr>

<td><c:out value="${user.id}" /></td>

<td><c:out value="${user.name}" /></td>

<td><c:out value="${user.credits}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete?id=<c:out value='${user.id}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

### 

### 1.1.6- TC.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>List/Assign The Teachers Classes</h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="inserst\_TC">

<fieldset>

<br/>

Teacher Name:<br/>

<select id="Teacher" name="t\_name">

<c:if test="${listteachers != null}">

<c:forEach var="user" items="${listteachers}">

<option value="${user.id}"><c:out value="${user.name}" /></option>

</c:forEach>

</c:if>

</select>

<br/>

Class :<br/>

<select id="Class" name="c\_name">

<c:if test="${listclasses != null}">

<c:forEach var="user" items="${listclasses}">

<option value="${user[0]}"><c:out value="${user[1]} / Room ${user[2]}" /></option>

</c:forEach>

</c:if>

</select>

</fieldset>

<br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:red'>Empty selection!!</span><%

}

%>

<input type="submit" value="Submit">

</form>

<div align="center">

<form action="list\_TC" >

<input type="submit" value="Refresh">

</form>

<table border="1" cellpadding="15">

<caption><h2>List of assigned Teachers Classes</h2></caption>

<tr>

<th>Teacher</th>

<th>Subject</th>

<th>RoomNo</th>

<th>Time</th>

</tr>

<c:forEach var="user" items="${list\_tc}">

<tr>

<td><c:out value="${user[1]}" /></td>

<td><c:out value="${user[2]}" /></td>

<td><c:out value="${user[3]}" /></td>

<td><c:out value="${user[4]}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete\_tc?id=<c:out value='${user[0]}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

### 

### 1.1.7 Teachers.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>List/Add Teachers</h1>

<h2>

<a href="MainPage.jsp">Home Page</a>

</h2>

<form action="insert\_teacher">

<fieldset>

Name:<br/><input type="text" name="name" placeholder="inter name">

<br/>

Email:<br/> <input type="email" name="email" placeholder="inter email">

<br/>

Gender:

<br/>

<select name="gender">

<option value="male">male</option>

<option value="female">female</option>

</select>

<br/>

Age:

<br/>

<input type="text" name="age" placeholder="age">

<br/>

Role:

<br/>

<select name="role">

<option value="Teacher">Teacher</option>

</select>

</fieldset><br/>

<%

if(request.getAttribute("TheState") == (Object)false ) {

%><span id="e" style='color:red'>Empty fields!!</span><%

}

%>

<input type="submit" value="save" >

</form>

<div align="center">

<form action="list\_teacher" >

<input type="submit" value="Refresh">

</form>

<table border="1" cellpadding="10">

<caption><h2>List of Teachers</h2></caption>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Gender</th>

<th>Age</th>

<th>Role</th>

</tr>

<c:forEach var="user" items="${listUser}">

<tr>

<td><c:out value="${user.id}" /></td>

<td><c:out value="${user.name}" /></td>

<td><c:out value="${user.email}" /></td>

<td><c:out value="${user.jender}" /></td>

<td><c:out value="${user.age}" /></td>

<td><c:out value="${user.role}" /></td>

<td>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="delete\_teacher?id=<c:out value='${user.id}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>