**SOURCE CODE-DEVELOPING A BACKEND ADMIN**

**FOR**

**LEARNER’S ACADEMY**

**Main Folder 🡪 To create a new Dynamic Web Project 🡪 package🡪 classes folder ,servlets folder and in WEBINF-Jsp files**

**1.com.swaraj.dao**

* **ClassesDao.java**

package com.swaraj.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.swaraj.model.Classes;

import com.swaraj.util.HibernateUtil;

public class ClassesDao

{

/\*\*

\* Save Classes

\* @param classes

\*/

public void saveClasses(Classes classes) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.save(classes);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Update Classes

\* @param classes

\*/

public void updateClasses(Classes classes) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.update(classes);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Delete Classes

\* @param id

\*/

public void deleteClasses(int id) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// Delete a classes object

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

if (classes != null) {

session.delete(classes);

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

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Get Classes By ID

\* @param id

\* @return

\*/

public Classes getClasses(int id) {

Transaction transaction = null;

Classes classes = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an classes object

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

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return classes;

}

/\*\*

\* Get all Classess

\* @return

\*/

@SuppressWarnings("unchecked")

public List<Classes> getAllClasses() {

Transaction transaction = null;

List<Classes> listOfClasses = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an classes object

listOfClasses = session.createQuery("from Classes").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return listOfClasses;

}

}

* **StudentDao.java**

package com.swaraj.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.swaraj.model.Student;

import com.swaraj.util.HibernateUtil;

public class StudentDao {

/\*\*

\* Save Student

\* @param student

\*/

public void saveStudent(Student student) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.save(student);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Update Student

\* @param student

\*/

public void updateStudent(Student student) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.update(student);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Delete Student

\* @param id

\*/

public void deleteStudent(int id) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// Delete a student object

Student student = session.get(Student.class, id);

if (student != null) {

session.delete(student);

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

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Get Student By ID

\* @param id

\* @return

\*/

public Student getStudent(int id) {

Transaction transaction = null;

Student student = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an student object

student = session.get(Student.class, id);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return student;

}

/\*\*

\* Get all Students

\* @return

\*/

@SuppressWarnings("unchecked")

public List<Student> getAllStudent() {

Transaction transaction = null;

List<Student> listOfStudent = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an student object

listOfStudent = session.createQuery("from Student").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return listOfStudent;

}

}

* **SubjectDao.java**

package com.swaraj.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.swaraj.model.Subject;

import com.swaraj.util.HibernateUtil;

public class SubjectDao {

/\*\*

\* Save Subject

\* @param subject

\*/

public void saveSubject(Subject subject) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// 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();

}

}

/\*\*

\* Update Subject

\* @param subject

\*/

public void updateSubject(Subject subject) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.update(subject);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Delete Subject

\* @param id

\*/

public void deleteSubject(int id) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// Delete a subject object

Subject subject = session.get(Subject.class, id);

if (subject != null) {

session.delete(subject);

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

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Get Subject By ID

\* @param id

\* @return

\*/

public Subject getSubject(int id) {

Transaction transaction = null;

Subject subject = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an subject object

subject = session.get(Subject.class, id);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return subject;

}

/\*\*

\* Get all Subjects

\* @return

\*/

@SuppressWarnings("unchecked")

public List<Subject> getAllSubject() {

Transaction transaction = null;

List<Subject> listOfSubject = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an subject object

listOfSubject = session.createQuery("from Subject").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return listOfSubject;

}

}

* **TeacherDao.java**

package com.swaraj.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.swaraj.model.Teacher;

import com.swaraj.util.HibernateUtil;

public class TeacherDao {

/\*\*

\* Save Teacher

\* @param teacher

\*/

public void saveTeacher(Teacher teacher) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.save(teacher);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Update Teacher

\* @param teacher

\*/

public void updateTeacher(Teacher teacher) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.update(teacher);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Delete Teacher

\* @param id

\*/

public void deleteTeacher(int id) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// Delete a teacher object

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

if (teacher != null) {

session.delete(teacher);

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

}

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Get Teacher By ID

\* @param id

\* @return

\*/

public Teacher getTeacher(int id) {

Transaction transaction = null;

Teacher teacher = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an teacher object

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

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return teacher;

}

/\*\*

\* Get all Teachers

\* @return

\*/

@SuppressWarnings("unchecked")

public List<Teacher> getAllTeacher() {

Transaction transaction = null;

List<Teacher> listOfTeacher = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an teacher object

listOfTeacher = session.createQuery("from Teacher").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return listOfTeacher;

}

}

* **UserDao.java**

package com.swaraj.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import com.swaraj.model.User;

import com.swaraj.util.HibernateUtil;

public class UserDao {

/\*\*

\* Save User

\* @param user

\*/

public void saveUser(User user) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

transaction = session.beginTransaction();

// save the student object

session.save(user);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Update User

\* @param user

\*/

public void updateUser(User user) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.update(user);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

/\*\*

\* Delete User

\* @param id

\*/

public void deleteUser(int id) {

Transaction transaction = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// Delete a user object

User user = session.get(User.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();

}

}

/\*\*

\* Get User By ID

\* @param id

\* @return

\*/

public User getUser(int id) {

Transaction transaction = null;

User user = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an user object

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

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return user;

}

public Boolean getUserByUserNameAndPassword(String userName, String password) {

Transaction transaction = null;

User user = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an user object

String hql = "FROM User U WHERE U.name = :name and U.password = :password";

Query query = session.createQuery(hql);

query.setParameter("name",userName);

query.setParameter("password",password);

List results = query.list();

if(null!=results && !results.isEmpty()){

return true;

}else{

return false;

}

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return false;

}

/\*\*

\* Get all Users

\* @return

\*/

@SuppressWarnings("unchecked")

public List<User> getAllUser() {

Transaction transaction = null;

List<User> listOfUser = null;

try {

Session session = HibernateUtil.getSession();

// start a transaction

transaction = session.beginTransaction();

// get an user object

listOfUser = session.createQuery("from User").getResultList();

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

return listOfUser;

}

}

**2.com.swaraj.model**

* **Classes.java**

package com.swaraj.model;

import java.util.ArrayList;

import java.util.Collection;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name="Classes")

public class Classes {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="class\_id")

protected int classId;

@Column(name="class\_name")

protected String className;

@Column(name="class\_section")

protected String classSection;

@ManyToMany(cascade = { CascadeType.ALL })

@JoinTable(

name = "classes\_subject",

joinColumns = { @JoinColumn(name = "class\_id") },

inverseJoinColumns = { @JoinColumn(name = "subject\_id") } )

Set<Subject> subjects = new HashSet<Subject>();

@ManyToMany(mappedBy = "classes")

private Set<Teacher> teachers = new HashSet<Teacher>();

@OneToMany(mappedBy="grade")

private Set<Student> students = new HashSet<Student>();

public Classes() {

super();

}

public Classes(int classId, String className, String classSection) {

super();

this.classId = classId;

this.className = className;

this.classSection = classSection;

}

public Classes(String className, String classSection) {

super();

this.className = className;

this.classSection = classSection;

}

public Classes(String className, String classSection, Set<Subject> subjects) {

super();

this.className = className;

this.classSection = classSection;

this.subjects = subjects;

}

public Classes(int classId, String className, String classSection, Set<Subject> subjects) {

super();

this.classId = classId;

this.className = className;

this.classSection = classSection;

this.subjects = subjects;

}

public int getClassId() {

return classId;

}

public void setClassId(int classId) {

this.classId = classId;

}

public String getClassName() {

return className;

}

public void setClassName(String className) {

this.className = className;

}

public String getClassSection() {

return classSection;

}

public void setClassSection(String classSection) {

this.classSection = classSection;

}

public Set<Subject> getSubjects() {

return subjects;

}

public void setSubjects(Set<Subject> subjects) {

this.subjects = subjects;

}

public Set<Teacher> getTeachers() {

return teachers;

}

public void setTeachers(Set<Teacher> teachers) {

this.teachers = teachers;

}

public Set<Student> getStudents() {

return students;

}

public void setStudents(Set<Student> students) {

this.students = students;

}

@Override

public String toString() {

return "Classes [classId=" + classId + ", className=" + className + ", classSection=" + classSection

+ ", subjects=" + subjects + ", teachers=" + teachers + ", students=" + students + "]";

}

}

* **Student.java**

package com.swaraj.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name="Student")

public class Student {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="student\_id")

protected int studentId;

@Column(name="student\_name")

private String studentName;

@Column(name="student\_mobile\_number")

private String mobileNumber;

@Column(name="student\_email")

private String studentEmail;

@ManyToOne

@JoinColumn(columnDefinition="integer", name="class\_id")

private Classes grade;

public Student() {

super();

}

public Student(int studentId, String studentName, String mobileNumber,

String studentEmail) {

super();

this.studentId = studentId;

this.studentName = studentName;

this.mobileNumber = mobileNumber;

this.studentEmail = studentEmail;

}

public Student(String studentName, String mobileNumber, String studentEmail) {

super();

this.studentName = studentName;

this.mobileNumber = mobileNumber;

this.studentEmail = studentEmail;

}

public int getStudentId() {

return studentId;

}

public void setStudentId(int studentId) {

this.studentId = studentId;

}

public String getStudentName() {

return studentName;

}

public void setStudentName(String studentName) {

this.studentName = studentName;

}

public String getMobileNumber() {

return mobileNumber;

}

public void setMobileNumber(String mobileNumber) {

this.mobileNumber = mobileNumber;

}

public String getStudentEmail() {

return studentEmail;

}

public void setStudentEmail(String studentEmail) {

this.studentEmail = studentEmail;

}

public Classes getGrade() {

return grade;

}

public void setGrade(Classes grade) {

this.grade = grade;

}

@Override

public String toString() {

return "Student [studentId=" + studentId + ", studentName=" + studentName + ", mobileNumber=" + mobileNumber

+ ", studentEmail=" + studentEmail + ", grade=" + grade + "]";

}

}

* **Subject.java**

package com.swaraj.model;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

@Entity

@Table(name="Subject")

public class Subject {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="id")

protected int id;

@Column(name="subject")

protected String subjectName;

@ManyToMany(mappedBy = "subjects")

private Set<Classes> classes = new HashSet<Classes>();

@ManyToMany(mappedBy = "subjects")

private Set<Teacher> teachers = new HashSet<Teacher>();

public Subject() {

super();

}

public Subject(int id, String subjectName) {

super();

this.id = id;

this.subjectName = subjectName;

}

public Subject(String subjectName) {

super();

this.subjectName = subjectName;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getSubjectName() {

return subjectName;

}

public void setSubjectName(String subjectName) {

this.subjectName = subjectName;

}

public Set<Classes> getClasses() {

return classes;

}

public void setClasses(Set<Classes> classes) {

this.classes = classes;

}

public Subject(int id, String subjectName, Set<Classes> classes) {

super();

this.id = id;

this.subjectName = subjectName;

this.classes = classes;

}

public Subject(String subjectName, Set<Classes> classes) {

super();

this.subjectName = subjectName;

this.classes = classes;

}

public Set<Teacher> getTeachers() {

return teachers;

}

public void setTeachers(Set<Teacher> teachers) {

this.teachers = teachers;

}

@Override

public String toString() {

return "Subject [id=" + id + ", subjectName=" + subjectName + ", classes=" + classes + ", teachers=" + teachers

+ "]";

}

}

* **Teacher.java**

package com.swaraj.model;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

@Entity

@Table(name="Teacher")

public class Teacher {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="id")

protected int id;

@Column(name="teacher\_name")

protected String teacherName;

@Column(name="teacher\_pwd")

protected String qualification;

@Column(name="teacher\_email")

protected String teacherEmail;

@ManyToMany(cascade = { CascadeType.ALL })

@JoinTable(

name = "teacher\_subject",

joinColumns = { @JoinColumn(name = "teacher\_id") },

inverseJoinColumns = { @JoinColumn(name = "subject\_id") }

)

Set<Subject> subjects = new HashSet<Subject>();

@ManyToMany(cascade = { CascadeType.ALL })

@JoinTable(

name = "teacher\_class",

joinColumns = { @JoinColumn(name = "teacher\_id") },

inverseJoinColumns = { @JoinColumn(name = "class\_id") }

)

Set<Classes> classes = new HashSet<Classes>();

public Teacher() {

super();

}

public Teacher(int id, String teacherName, String qualification, String teacherEmail) {

super();

this.id = id;

this.teacherName = teacherName;

this.qualification = qualification;

this.teacherEmail = teacherEmail;

}

public Teacher(String teacherName, String qualification, String teacherEmail) {

super();

this.teacherName = teacherName;

this.qualification = qualification;

this.teacherEmail = teacherEmail;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTeacherName() {

return teacherName;

}

public void setTeacherName(String teacherName) {

this.teacherName = teacherName;

}

public String getQualification() {

return qualification;

}

public void setQualification(String qualification) {

this.qualification = qualification;

}

public String getTeacherEmail() {

return teacherEmail;

}

public void setTeacherEmail(String teacherEmail) {

this.teacherEmail = teacherEmail;

}

public Set<Subject> getSubjects() {

return subjects;

}

public void setSubjects(Set<Subject> subjects) {

this.subjects = subjects;

}

public Set<Classes> getClasses() {

return classes;

}

public void setClasses(Set<Classes> classes) {

this.classes = classes;

}

@Override

public String toString() {

return "Teacher [id=" + id + ", teacherName=" + teacherName + ", qualification=" + qualification

+ ", teacherEmail=" + teacherEmail + ", subjects=" + subjects + ", classes=" + classes + "]";

}

}

**User.java**

**package** com.swaraj.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="UserData")

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

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

@Column(name="id")

**protected** **int** id;

@Column(name="user\_name")

**protected** String name;

@Column(name="user\_pwd")

**protected** String password;

@Column(name="user\_email")

**protected** String email;

@Column(name="user\_country")

**protected** String country;

**public** User() {

}

**public** User(String name, String password, String email, String country) {

**super**();

**this**.name = name;

**this**.password = password;

**this**.email = email;

**this**.country = country;

}

**public** User(**int** id, String name, String password, String email, String country) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.password = password;

**this**.email = email;

**this**.country = country;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**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 getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

@Override

**public** String toString() {

**return** "User [id=" + id + ", name=" + name + ", password=" + password + ", email=" + email + ", country="

+ country + "]";

}

}

**3.com.swaraj.servlets**

**ClassesServlet.java**

/\*package com.swaraj.servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

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 com.swaraj.dao.ClassesDao;

//import com.swaraj.dao.ClassesDao;

import com.swaraj.model.Classes;

//import com.swaraj.model.Classes;

\*//\*\*

\* ControllerServlet.java

\* This servlet acts as a page controller for the application, handling all

\* requests from the classes.

\* **@email** Ramesh Fadatare

\*//\*

@WebServlet("/classes")

public class ClassesServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private ClassesDao classesDao;

public void init() {

classesDao = new ClassesDao();

}

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 "/new":

showNewClassesForm(request, response);

break;

case "/insert":

insertClasses(request, response);

break;

case "/delete":

deleteClasses(request, response);

break;

case "/edit":

showClassesEditForm(request, response);

break;

case "/update":

updateClasses(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

RequestDispatcher dispatcher = request.getRequestDispatcher("class-list.jsp");

dispatcher.forward(request, response);

}

private void showNewClassesForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("class-form.jsp");

dispatcher.forward(request, response);

}

private void showClassesEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Classes existingClasses = classesDao.getClasses(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("class-form.jsp");

request.setAttribute("classes", existingClasses);

dispatcher.forward(request, response);

}

private void insertClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

//String password = "adminClasses1";

//String name, String password, String email, String country

Classes newClasses = new Classes(name, section);

classesDao.saveClasses(newClasses);

response.sendRedirect("list");

}

private void updateClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

// String password = "adminClasses1";

//String name, String password, String email, String country

Classes classes = new Classes(id,name, section);

classesDao.updateClasses(classes);

response.sendRedirect("list");

}

private void deleteClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

classesDao.deleteClasses(id);

response.sendRedirect("classes");

}

}

\*/

* **StudentServlet.java**

/\*package com.swaraj.servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

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 com.swaraj.dao.StudentDao;

import com.swaraj.model.Student;

;

\*//\*\*

\* ControllerServlet.java

\* This servlet acts as a page controller for the application, handling all

\* requests from the student.

\* **@email** Ramesh Fadatare

\*//\*

@WebServlet("/student")

public class StudentServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private StudentDao studentDao;

public void init() {

studentDao = new StudentDao();

}

protected void doPost(HttpServletRequest request,HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponseresponse)

throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/student/new":

showNewStudentForm(request, response);

break;

case "/student/insert":

insertStudent(request, response);

break;

case "/student/delete":

deleteStudent(request, response);

break;

case "/student/edit":

showEditStudentForm(request, response);

break;

case "/student/update":

updateStudent(request, response);

break;

case "/student/list":

listStudent(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Student> listStudent = studentDao.getAllStudent();

request.setAttribute("listStudent", listStudent);

RequestDispatcher dispatcher = request.getRequestDispatcher("student-list.jsp");

dispatcher.forward(request, response);

}

private void showNewStudentForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("student-form.jsp");

dispatcher.forward(request, response);

}

private void showEditStudentForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Student existingStudent = studentDao.getStudent(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("student-form.jsp");

request.setAttribute("student", existingStudent);

dispatcher.forward(request, response);

}

private void insertStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

//String password = "adminStudent1";

//String name, String password, String email, String country

Student newStudent = new Student(name, mobile,email);

studentDao.saveStudent(newStudent);

response.sendRedirect("list");

}

private void updateStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

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

//String password = "adminStudent1";

//String name, String password, String email, String country

Student student = new Student(id,name,mobile, email);

studentDao.updateStudent(student);

response.sendRedirect("list");

}

private void deleteStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

studentDao.deleteStudent(id);

response.sendRedirect("list");

}

}

\*/

* **SubjectServlet.java**

/\*package com.swaraj.servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

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 com.swaraj.dao.SubjectDao;

import com.swaraj.model.Subject;

\*//\*\*

\* ControllerServlet.java

\* This servlet acts as a page controller for the application, handlingall

\* requests from the subject.

\* **@email** Ramesh Fadatare

\*//\*

@WebServlet("/subject")

public class SubjectServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private SubjectDao subjectDao;

public void init() {

subjectDao = new SubjectDao();

}

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 "/new":

showNewForm(request, response);

break;

case "/insert":

insertSubject(request, response);

break;

case "/delete":

deleteSubject(request, response);

break;

case "/edit":

showEditForm(request, response);

break;

case "/update":

updateSubject(request, response);

break;

default:

listSubject(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listSubject(HttpServletRequest request,HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-list.jsp");

dispatcher.forward(request, response);

}

private void showNewForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-form.jsp");

dispatcher.forward(request, response);

}

private void showEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Subject existingSubject = subjectDao.getSubject(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-form.jsp");

request.setAttribute("subject", existingSubject);

dispatcher.forward(request, response);

}

private void insertSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

Subject newSubject = new Subject(name);

subjectDao.saveSubject(newSubject);

response.sendRedirect("list");

}

private void updateSubject(HttpServletRequest request,HttpServletResponse response)

throws SQLException, IOException {

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

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

//String name, String password, String email, String country

Subject subject = new Subject(id,name);

subjectDao.updateSubject(subject);

response.sendRedirect("list");

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

subjectDao.deleteSubject(id);

response.sendRedirect("list");

}

}

\*/

* **TeacherServlet.java**

/\*package com.swaraj.servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

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 com.swaraj.dao.TeacherDao;

import com.swaraj.model.Teacher;

\*//\*\*

\* ControllerServlet.java

\* This servlet acts as a page controller for the application, handling all

\* requests from the teacher.

\* **@email** Ramesh Fadatare

\*//\*

@WebServlet("/teacher")

public class TeacherServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private TeacherDao teacherDao;

public void init() {

teacherDao = new TeacherDao();

}

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 "/teacher/new":

showNewTeacherForm(request, response);

break;

case "/teacher/insert":

insertTeacher(request, response);

break;

case "/teacher/delete":

deleteTeacher(request, response);

break;

case "/teacher/edit":

showEditTeacherForm(request, response);

break;

case "/teacher/update":

updateTeacher(request, response);

break;

case "/teacher/list":

listTeacher(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Teacher> listTeacher = teacherDao.getAllTeacher();

request.setAttribute("listTeacher", listTeacher);

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-list.jsp");

dispatcher.forward(request, response);

}

private void showNewTeacherForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher request.getRequestDispatcher("teacher-form.jsp");

dispatcher.forward(request, response);

}

private void showEditTeacherForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Teacher existingTeacher = teacherDao.getTeacher(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-form.jsp");

request.setAttribute("teacher", existingTeacher);

dispatcher.forward(request, response);

}

private void insertTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

//String name, String password, String email, String country

Teacher newTeacher = new Teacher(name,qualification,email);

teacherDao.saveTeacher(newTeacher);

response.sendRedirect("list");

}

private void updateTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

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

//String name, String password, String email, String country

Teacher teacher = new Teacher(id,name,qualification,email);

teacherDao.updateTeacher(teacher);

response.sendRedirect("list");

}

private void deleteTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

teacherDao.deleteTeacher(id);

response.sendRedirect("list");

}

}

\*/

* **UserServlet.java**

package com.swaraj.servlets;

import java.io.IOException;

import java.sql.SQLException;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

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 com.swaraj.dao.ClassesDao;

import com.swaraj.dao.StudentDao;

import com.swaraj.dao.SubjectDao;

import com.swaraj.dao.TeacherDao;

import com.swaraj.dao.UserDao;

import com.swaraj.model.Classes;

import com.swaraj.model.Student;

import com.swaraj.model.Subject;

import com.swaraj.model.Teacher;

import com.swaraj.model.User;

/\*\*

\* ControllerServlet.java

\* This servlet acts as a page controller for the application, handling all

\* requests from the user.

\* @email Ramesh Fadatare

\*/

@WebServlet("/")

public class UserServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private UserDao userDao;

private SubjectDao subjectDao;

private TeacherDao teacherDao;

private ClassesDao classesDao;

private StudentDao studentDao;

public void init() {

userDao = new UserDao();

subjectDao = new SubjectDao();

teacherDao = new TeacherDao();

classesDao = new ClassesDao();

studentDao = new StudentDao();

}

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 "/assignTeacherToAClass":

assignTeacherToAClass(request, response);

break;

case "/assignSubjectToAClass":

assignSubjectToAClass(request, response);

break;

case "/assignStudentToAClass":

assignStudentToAClass(request, response);

break;

case "/loginprocess":

doLoginValidate(request, response);

break;

case "/user/new":

showNewForm(request, response);

break;

case "/user/insert":

insertUser(request, response);

break;

case "/user/delete":

deleteUser(request, response);

break;

case "/user/edit":

showEditForm(request, response);

break;

case "/user/update":

updateUser(request, response);

break;

case "/user/list":

listUser(request, response);

break;

case "/subject/new":

showNewSubjectForm(request, response);

break;

case "/subject/insert":

insertSubject(request, response);

break;

case "/subject/delete":

deleteSubject(request, response);

break;

case "/subject/edit":

showSubjectEditForm(request, response);

break;

case "/subject/update":

updateSubject(request, response);

break;

case "/subject/list":

listSubject(request, response);

break;

case "/teacher/new":

showNewTeacherForm(request, response);

break;

case "/teacher/insert":

insertTeacher(request, response);

break;

case "/teacher/delete":

deleteTeacher(request, response);

break;

case "/teacher/edit":

showEditTeacherForm(request, response);

break;

case "/teacher/update":

updateTeacher(request, response);

break;

case "/teacher/list":

listTeacher(request, response);

break;

case "/classes/new":

showNewClassesForm(request, response);

break;

case "/classes/insert":

insertClasses(request, response);

break;

case "/classes/delete":

deleteClasses(request, response);

break;

case "/classes/edit":

showClassesEditForm(request, response);

break;

case "/classes/update":

updateClasses(request, response);

break;

case "/classes/list":

listClasses(request, response);

break;

case "/classes/report":

classReport(request, response);

break;

case "/student/new":

showNewStudentForm(request, response);

break;

case "/student/insert":

insertStudent(request, response);

break;

case "/student/delete":

deleteStudent(request, response);

break;

case "/student/edit":

showEditStudentForm(request, response);

break;

case "/student/update":

updateStudent(request, response);

break;

case "/student/list":

listStudent(request, response);

break;

case "/home":

displayMenu(request, response);

break;

case "/assignsub":

assignsub(request, response);

break;

case "/assignteacher":

assignteacher(request, response);

break;

case "/assignstudent":

assignstudent(request, response);

break;

default:

displayLogin(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void classReport(HttpServletRequest request, HttpServletResponse response) throws IOException {

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

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

Classes existingClasses = classesDao.getClasses(classId);

Set<Student> students = new HashSet<Student>();

students= existingClasses.getStudents();

request.setAttribute("listStudent", students);

Set<Teacher> teachers = new HashSet<Teacher>();

teachers=existingClasses.getTeachers();

request.setAttribute("listTeacher", teachers);

request.setAttribute("classes", existingClasses);

response.sendRedirect("class-report.jsp");

}

private void assignstudent(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("student-classes.jsp");

List<Student> listStudent = studentDao.getAllStudent();

request.setAttribute("listStudent", listStudent);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

dispatcher.forward(request, response);

}

private void assignStudentToAClass(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String studentId = request.getParameter("student");

String classesId = request.getParameter("classes");

int id = Integer.parseInt(studentId);

Student existingStudent = studentDao.getStudent(id);

int classId = Integer.parseInt(classesId);

Classes existingClasses = classesDao.getClasses(classId);

Set<Student> students = new HashSet<Student>();

students= existingClasses.getStudents();

students.add(existingStudent);

existingClasses.setStudents(students);

classesDao.updateClasses(existingClasses);

existingStudent.setGrade(existingClasses);

studentDao.updateStudent(existingStudent);

RequestDispatcher dispatcher = request.getRequestDispatcher("student-classes.jsp");

List<Student> listStudent = studentDao.getAllStudent();

request.setAttribute("listStudent", listStudent);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

dispatcher.forward(request, response);

}

private void assignTeacherToAClass(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String subjectId = request.getParameter("subject");

String classesId = request.getParameter("classes");

String teacher\_Id = request.getParameter("teacher");

int id = Integer.parseInt(subjectId);

Subject existingSubject = subjectDao.getSubject(id);

int classId = Integer.parseInt(classesId);

Classes existingClasses = classesDao.getClasses(classId);

int teacherId = Integer.parseInt(teacher\_Id);

Teacher teacherObj = teacherDao.getTeacher(teacherId);

Set<Subject> subjects = new HashSet<>();

subjects= teacherObj.getSubjects();

subjects.add(existingSubject);

teacherObj.setSubjects(subjects);

Set<Classes> classes = new HashSet<>();

classes= teacherObj.getClasses();

classes.add(existingClasses);

teacherObj.setClasses(classes);

teacherDao.updateTeacher(teacherObj);

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-class-subject.jsp");

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

List<Teacher> listTeacher = teacherDao.getAllTeacher();

request.setAttribute("listTeacher", listTeacher);

dispatcher.forward(request, response);

}

private void assignteacher(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-class-subject.jsp");

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

List<Teacher> listTeacher = teacherDao.getAllTeacher();

request.setAttribute("listTeacher", listTeacher);

dispatcher.forward(request, response);

}

private void assignsub(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("assign-subject-class.jsp");

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

dispatcher.forward(request, response);

}

private void assignSubjectToAClass(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String subjectId = request.getParameter("subject");

String classesId = request.getParameter("classes");

int id = Integer.parseInt(subjectId);

Subject existingSubject = subjectDao.getSubject(id);

int classId = Integer.parseInt(classesId);

Classes existingClasses = classesDao.getClasses(classId);

Set<Subject> subjects = new HashSet<Subject>();

subjects= existingClasses.getSubjects();

subjects.add(existingSubject);

existingClasses.setSubjects(subjects);

classesDao.updateClasses(existingClasses);

RequestDispatcher dispatcher = request.getRequestDispatcher("assign-subject-class.jsp");

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

dispatcher.forward(request, response);

}

private void doLoginValidate(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

Boolean existingUser = userDao.getUserByUserNameAndPassword(userName,password);

if(existingUser){

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

// request.setAttribute("user", existingUser);

dispatcher.forward(request, response);

}else{

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

dispatcher.forward(request, response);

}

}

private void displayLogin(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

dispatcher.forward(request, response);

}

private void displayMenu(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

dispatcher.forward(request, response);

}

private void listUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<User> listUser = userDao.getAllUser();

request.setAttribute("listUser", listUser);

RequestDispatcher dispatcher = request.getRequestDispatcher("user-list.jsp");

dispatcher.forward(request, response);

}

private void showNewForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("user-form.jsp");

dispatcher.forward(request, response);

}

private void showEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

User existingUser = userDao.getUser(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("user-form.jsp");

request.setAttribute("user", existingUser);

dispatcher.forward(request, response);

}

private void insertUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

String password = "adminUser1";

//String name, String password, String email, String country

User newUser = new User(name, password,email, country);

userDao.saveUser(newUser);

response.sendRedirect("list");

}

private void updateUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

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

String password = "adminUser1";

//String name, String password, String email, String country

User user = new User(id,name, password,email, country);

userDao.updateUser(user);

response.sendRedirect("list");

}

private void deleteUser(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

userDao.deleteUser(id);

response.sendRedirect("list");

}

private void listSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Subject> listSubject = subjectDao.getAllSubject();

request.setAttribute("listSubject", listSubject);

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-list.jsp");

dispatcher.forward(request, response);

}

private void showNewSubjectForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-form.jsp");

dispatcher.forward(request, response);

}

private void showSubjectEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Subject existingSubject = subjectDao.getSubject(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("subject-form.jsp");

request.setAttribute("subject", existingSubject);

dispatcher.forward(request, response);

}

private void insertSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

Subject newSubject = new Subject(name);

subjectDao.saveSubject(newSubject);

response.sendRedirect("list");

}

private void updateSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

//String name, String password, String email, String country

Subject subject = new Subject(id,name);

subjectDao.updateSubject(subject);

response.sendRedirect("list");

}

private void deleteSubject(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

subjectDao.deleteSubject(id);

response.sendRedirect("list");

}

private void listTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Teacher> listTeacher = teacherDao.getAllTeacher();

request.setAttribute("listTeacher", listTeacher);

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-list.jsp");

dispatcher.forward(request, response);

}

private void showNewTeacherForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-form.jsp");

dispatcher.forward(request, response);

}

private void showEditTeacherForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Teacher existingTeacher = teacherDao.getTeacher(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("teacher-form.jsp");

request.setAttribute("teacher", existingTeacher);

dispatcher.forward(request, response);

}

private void insertTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

//String name, String password, String email, String country

Teacher newTeacher = new Teacher(name,qualification,email);

teacherDao.saveTeacher(newTeacher);

response.sendRedirect("list");

}

private void updateTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

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

//String name, String password, String email, String country

Teacher teacher = new Teacher(id,name,qualification,email);

teacherDao.updateTeacher(teacher);

response.sendRedirect("list");

}

private void deleteTeacher(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

teacherDao.deleteTeacher(id);

response.sendRedirect("list");

}

private void listClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Classes> listClasses = classesDao.getAllClasses();

request.setAttribute("listClass", listClasses);

RequestDispatcher dispatcher = request.getRequestDispatcher("class-list.jsp");

dispatcher.forward(request, response);

}

private void showNewClassesForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("class-form.jsp");

dispatcher.forward(request, response);

}

private void showClassesEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Classes existingClasses = classesDao.getClasses(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("class-form.jsp");

request.setAttribute("classes", existingClasses);

dispatcher.forward(request, response);

}

private void insertClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

//String password = "adminClasses1";

//String name, String password, String email, String country

Classes newClasses = new Classes(name, section);

classesDao.saveClasses(newClasses);

response.sendRedirect("list");

}

private void updateClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

// String password = "adminClasses1";

//String name, String password, String email, String country

Classes classes = new Classes(id,name, section);

classesDao.updateClasses(classes);

response.sendRedirect("list");

}

private void deleteClasses(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

classesDao.deleteClasses(id);

response.sendRedirect("classes");

}

private void listStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Student> listStudent = studentDao.getAllStudent();

request.setAttribute("listStudent", listStudent);

RequestDispatcher dispatcher = request.getRequestDispatcher("student-list.jsp");

dispatcher.forward(request, response);

}

private void showNewStudentForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("student-form.jsp");

dispatcher.forward(request, response);

}

private void showEditStudentForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Student existingStudent = studentDao.getStudent(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("student-form.jsp");

request.setAttribute("student", existingStudent);

dispatcher.forward(request, response);

}

private void insertStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

//String password = "adminStudent1";

//String name, String password, String email, String country

Student newStudent = new Student(name, mobile,email);

studentDao.saveStudent(newStudent);

response.sendRedirect("list");

}

private void updateStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

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

//String password = "adminStudent1";

//String name, String password, String email, String country

Student student = new Student(id,name,mobile, email);

studentDao.updateStudent(student);

response.sendRedirect("list");

}

private void deleteStudent(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

studentDao.deleteStudent(id);

response.sendRedirect("list");

}

}

**WEBINF:- JSP Files**

* **Assign-subject-class.jsp**

<%@page import=*"com.swaraj.model.Subject"*%>

<%@page import=*"com.swaraj.model.Classes"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"java.util.List"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Assign Subject to a Class</title>

</head>

<body>

<a href=*"home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<div align=*"center"*>

    <h2>Assign Subject to a Class</h2>

    <form action=*"assignSubjectToAClass"* method=*"post"*>

        Select a Subject:&nbsp;

<select name=*"subject"* style="width: *100px*;">

<%

List<Subject> subList=(ArrayList<Subject>)request.getAttribute("listSubject");

**for**(**int** i=0;i<subList.size();i++){

%>

<option value=*"*<%=subList.get(i).getId()%>*"* ><%=subList.get(i).getSubjectName()%></option>

<%} %>

<%-- <c:forEach var="subject1" items="${listSubject}">

<option value="${subject1.id}" ><c:out value="${subject1.subjectName}"/></option>

</c:forEach> --%>

</select>

  Select a Class:&nbsp;

       <select name=*"classes"* style="width: *100px*;">

<%

List<Classes> classesList=(ArrayList<Classes>)request.getAttribute("listClass");

**for**(**int** i=0;i<classesList.size();i++){

%>

<option value=*"*<%=classesList.get(i).getClassId()%>*"* ><%=classesList.get(i).getClassName()%></option>

<%} %>

</select>

        <br/><br/>

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

    </form>

</div>

</body>

</html>

* **Class-report.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Class Report</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>Class Report</h1>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

Class Report

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${classes.classId}' />*"* />

</c:if>

<tr>

<th>Class Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

value=*"*<c:out value='${classes.className}' />*"*

/>

</td>

</tr>

<tr>

<th>Class with Section </th>

<td>

<input type=*"text"* name=*"section"* size=*"45"*

value=*"*<c:out value='${classes.classSection}' />*"*

/>

</td>

</tr>

</table>

&nbsp;&nbsp;&nbsp;

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>Name</th>

<th>MobileNumber</th>

<th>Email</th>

</tr>

<c:forEach var=*"student"* items=*"*${listStudent}*"*>

<tr>

<td><c:out value=*"*${student.studentId}*"* /></td>

<td><c:out value=*"*${student.studentName}*"* /></td>

<td><c:out value=*"*${student.mobileNumber}*"* /></td>

<td><c:out value=*"*${student.studentEmail}*"* /></td>

</tr>

</br>

</c:forEach>

</table>

</div>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>Teacher Name</th>

<th>Qualification</th>

<th>Email</th>

</tr>

<c:forEach var=*"teacher"* items=*"*${listTeacher}*"*>

<tr>

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

<td><c:out value=*"*${teacher.teacherName}*"* /></td>

<td><c:out value=*"*${teacher.qualification}*"* /></td>

<td><c:out value=*"*${teacher.teacherEmail}*"* /></td>

</tr>

</c:forEach>

</table>

</div>

</div>

</body>

</html>

* **Error.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"* isErrorPage=*"true"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Error</title>

</head>

<body>

<center>

<h1>Error</h1>

<h2><%=exception.getMessage() %><br/> </h2>

</center>

</body>

</html>

* **Login.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Login</title>

</head>

<body background=*"image1.jpg"*>

<center>

<h1>Login</h1>

</center>

<div align=*"center"*>

<form action=*"loginprocess"* method=*"post"*>

<table border=*"1"* cellpadding=*"5"*>

<tr>

<th>User Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

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

/>

</td>

</tr>

<tr>

<th>User Password: </th>

<td>

<input type=*"password"* name=*"password"* size=*"45"*

value=*"*<c:out value='${user.password}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

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

&nbsp;&nbsp<a href=*"user/new"*>Register User</a>

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **Menu.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>Menu</title>

</head>

<body>

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<a href=*"home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<h1><u>Menu</u></h1>

<h2>

<ul class=*"sidebar-menu"* data-widget=*"tree"*>

<li>

<a href=*"user/list"*>Users List</a>

</li>

<li>

<a href=*"teacher/list"*>Teachers List</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"student/list"*>Students List</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"subject/list"*>Subjects List</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"classes/list"*>Classes List</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"assignsub"*>Assign Subject to a Class</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"assignteacher"*>Assign Teacher to a Class and Subject</a>

&nbsp;&nbsp;&nbsp;

</li>

<li>

<a href=*"assignstudent"*>Assign Student to a Class</a>

&nbsp;&nbsp;&nbsp;

</li>

</ul>

</h2>

</body>

</html>

* **Student-classes.jsp**

<%@page import=*"com.swaraj.model.Student"*%>

<%@page import=*"com.swaraj.model.Classes"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"java.util.List"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Assign Student to a Class</title>

</head>

<body>

<a href=*"home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<div align=*"center"*>

 <h2>Assign Student to a Class</h2>

    <form action=*"assignStudentToAClass"* method=*"post"*>

        Select a Student:&nbsp;

<select name=*"student"* style="width: *100px*;">

<%

List<Student> subList=(ArrayList<Student>)request.getAttribute("listStudent");

**for**(**int** i=0;i<subList.size();i++){

%>

<option value=*"*<%=subList.get(i).getStudentId()%>*"* ><%=subList.get(i).getStudentName()%></option>

<%} %>

<%-- <c:forEach var="subject1" items="${listSubject}">

<option value="${subject1.id}" ><c:out value="${subject1.subjectName}"/></option>

</c:forEach> --%>

</select>

  Select a Class:&nbsp;

       <select name=*"classes"* style="width: *100px*;">

<%

List<Classes> classesList=(ArrayList<Classes>)request.getAttribute("listClass");

**for**(**int** i=0;i<classesList.size();i++){

%>

<option value=*"*<%=classesList.get(i).getClassId()%>*"* ><%=classesList.get(i).getClassName()%></option>

<%} %>

</select>

        <br/><br/>

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

    </form>

</div>

</body>

</html>

* **Teacher-class-subject.jsp**

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

pageEncoding=*"ISO-8859-1"*%>

<%@page import=*"com.swaraj.model.Subject"*%>

<%@page import=*"com.swaraj.model.Classes"*%>

<%@page import=*"com.swaraj.model.Teacher"*%>

<%@page import=*"java.util.\*"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Assign teacher to a class for the subject</title>

</head>

<body>

<a href=*"home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<div align=*"center"*>

    <h2>Assign teacher to a class for the subject</h2>

    <form action=*"assignTeacherToAClass"* method=*"post"*>

        Select a Teacher:&nbsp;

<select name=*"teacher"* style="width: *100px*;">

<%

List<Teacher> teachersList=(ArrayList<Teacher>)request.getAttribute("listTeacher");

**for**(**int** i=0;i<teachersList.size();i++){

%>

<option value=*"*<%=teachersList.get(i).getId()%>*"* ><%=teachersList.get(i).getTeacherName()%></option>

<%} %>

</select>

  Select a Class:&nbsp;

       <select name=*"classes"* style="width: *100px*;">

<%

List<Classes> classesList=(ArrayList<Classes>)request.getAttribute("listClass");

**for**(**int** i=0;i<classesList.size();i++){

%>

<option value=*"*<%=classesList.get(i).getClassId()%>*"* ><%=classesList.get(i).getClassName()%></option>

<%} %>

</select>

Select a Subject:&nbsp;

        <select name=*"subject"* style="width: *100px*;">

<%

List<Subject> subList=(ArrayList<Subject>)request.getAttribute("listSubject");

**for**(**int** i=0;i<subList.size();i++){

%>

<option value=*"*<%=subList.get(i).getId()%>*"* ><%=subList.get(i).getSubjectName()%></option>

<%} %>

</select>

        <br/><br/>

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

    </form>

</div>

</body>

</html>

**Main.Webapp.classes**

* **class-form.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Class Management Application</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>Class Management</h1>

<h2>

<a href=*"new"*>Add New Class</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Classes</a>

</h2>

</center>

<div align=*"center"*>

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

<form action=*"update"* method=*"post"*>

</c:if>

<c:if test=*"*${class == null}*"*>

<form action=*"insert"* method=*"post"*>

</c:if>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

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

Edit Class

</c:if>

<c:if test=*"*${class == null}*"*>

Add New Class

</c:if>

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${class.classId}' />*"* />

</c:if>

<tr>

<th>Class Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

value=*"*<c:out value='${class.className}' />*"*

/>

</td>

</tr>

<tr>

<th>Class with Section </th>

<td>

<input type=*"text"* name=*"section"* size=*"45"*

value=*"*<c:out value='${class.classSection}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

<input type=*"submit"* value=*"Save"* />

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **class-list.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>Class Management Applicationn</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<center>

<h1>Class Management</h1>

<h2>

<a href=*"new"*>Add New Class</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Classes</a>

</h2>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>ClassName</th>

<th>Section</th>

<th>Actions</th>

</tr>

<c:forEach var=*"class"* items=*"*${listClass}*"*>

<tr>

<td><c:out value=*"*${class.classId}*"* /></td>

<td><c:out value=*"*${class.className}*"* /></td>

<td><c:out value=*"*${class.classSection}*"* /></td>

<td>

<a href=*"edit?id=*<c:out value='${class.classId}' />*"*>Edit</a>

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

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

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

<a href=*"report?id=*<c:out value='${class.classId}' />*"*>Class Report</a>

</td>

</tr>

</br>

</c:forEach>

</table>

</div>

</body>

</html>

**Main.Webapp.student**

* **student-form.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Student Management Application</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>Student Management</h1>

<h2>

<a href=*"new"*>Add New Student</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Students</a>

</h2>

</center>

<div align=*"center"*>

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

<form action=*"update"* method=*"post"*>

</c:if>

<c:if test=*"*${student == null}*"*>

<form action=*"insert"* method=*"post"*>

</c:if>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

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

Edit Student

</c:if>

<c:if test=*"*${student == null}*"*>

Add New Student

</c:if>

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${student.studentId}' />*"* />

</c:if>

<tr>

<th>Student Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

value=*"*<c:out value='${student.studentName}' />*"*

/>

</td>

</tr>

<tr>

<th>Student Mobile: </th>

<td>

<input type=*"text"* name=*"mobile"* size=*"15"*

value=*"*<c:out value='${student.mobileNumber}' />*"*

/>

</td>

</tr>

<tr>

<th>Student Email: </th>

<td>

<input type=*"text"* name=*"email"* size=*"45"*

value=*"*<c:out value='${student.studentEmail}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

<input type=*"submit"* value=*"Save"* />

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **student-list.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>Student Management Application</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<center>

<h1>Student Management</h1>

<h2>

<a href=*"new"*>Add New Student</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Students</a>

</h2>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>Name</th>

<th>MobileNumber</th>

<th>Email</th>

<th>Actions</th>

</tr>

<c:forEach var=*"student"* items=*"*${listStudent}*"*>

<tr>

<td><c:out value=*"*${student.studentId}*"* /></td>

<td><c:out value=*"*${student.studentName}*"* /></td>

<td><c:out value=*"*${student.mobileNumber}*"* /></td>

<td><c:out value=*"*${student.studentEmail}*"* /></td>

<td>

<a href=*"edit?id=*<c:out value='${student.studentId}' />*"*>Edit</a>

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

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

</td>

</tr>

</br>

</c:forEach>

</table>

</div>

</body>

</html>

**Main.Webapp.subject**

* **subject-form.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Manage Subjects</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>Manage Subjects</h1>

<h2>

<a href=*"new"*>Add New Subject</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Subjects</a>

</h2>

</center>

<div align=*"center"*>

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

<form action=*"update"* method=*"post"*>

</c:if>

<c:if test=*"*${subject == null}*"*>

<form action=*"insert"* method=*"post"*>

</c:if>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

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

Edit Subject

</c:if>

<c:if test=*"*${subject == null}*"*>

Add New Subject

</c:if>

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${subject.id}' />*"* />

</c:if>

<tr>

<th>Subject Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

value=*"*<c:out value='${subject.subjectName}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

<input type=*"submit"* value=*"Save"* />

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **subject-list.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>Manage Subjects</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<center>

<h1>Manage Subjects</h1>

<h2>

<a href=*"new"*>Add New Subjcet</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Subjects</a>

</h2>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>Name</th>

<th>Actions</th>

</tr>

<c:forEach var=*"subject"* items=*"*${listSubject}*"*>

<tr>

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

<td><c:out value=*"*${subject.subjectName}*"* /></td>

<td>

<a href=*"edit?id=*<c:out value='${subject.id}'/>*"*>Edit</a>

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

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

</td>

</tr>

</br>

</c:forEach>

</table>

</div>

</body>

</html>

**Main.Webapp.teacher**

* **teacher-form.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>Teacher Management</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>Teacher Management</h1>

<h2>

<a href=*"new"*>Add New Teacher</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Teachers</a>

</h2>

</center>

<div align=*"center"*>

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

<form action=*"update"* method=*"post"*>

</c:if>

<c:if test=*"*${teacher == null}*"*>

<form action=*"insert"* method=*"post"*>

</c:if>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

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

Edit User

</c:if>

<c:if test=*"*${teacher == null}*"*>

Add New Teacher

</c:if>

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${teacher.id}' />*"* />

</c:if>

<tr>

<th>Teacher Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

value=*"*<c:out value='${teacher.teacherName}' />*"*

/>

</td>

</tr>

<tr>

<th>Teacher Qualification: </th>

<td>

<input type=*"text"* name=*"qualification"* size=*"15"*

value=*"*<c:out value='${teacher.qualification}' />*"*

/>

</td>

</tr>

<tr>

<th>Teacher Email: </th>

<td>

<input type=*"text"* name=*"email"* size=*"45"*

value=*"*<c:out value='${teacher.teacherEmail}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

<input type=*"submit"* value=*"Save"* />

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **teacher -list.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>Teacher Management</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<center>

<h1>Teacher Management</h1>

<h2>

<a href=*"new"*>Add New Teacher</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Teachers</a>

</h2>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

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

<tr>

<th>ID</th>

<th>Teacher Name</th>

<th>Qualification</th>

<th>Email</th>

<th>Actions</th>

</tr>

<c:forEach var=*"teacher"* items=*"*${listTeacher}*"*>

<tr>

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

<td><c:out value=*"*${teacher.teacherName}*"* /></td>

<td><c:out value=*"*${teacher.qualification}*"* /></td>

<td><c:out value=*"*${teacher.teacherEmail}*"* /></td>

<td>

<a href=*"edit?id=*<c:out value='${teacher.id}' />*"*>Edit</a>

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

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

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

**Main.Webapp.user**

* **user-form.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<html>

<head>

<title>User Management Application</title>

</head>

<body>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>User Management</h1>

<h2>

<a href=*"new"*>Add New User</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Users</a>

</h2>

</center>

<div align=*"center"*>

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

<form action=*"update"* method=*"post"*>

</c:if>

<c:if test=*"*${user == null}*"*>

<form action=*"insert"* method=*"post"*>

</c:if>

<table border=*"1"* cellpadding=*"5"*>

<caption>

<h2>

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

Edit User

</c:if>

<c:if test=*"*${user == null}*"*>

Add New User

</c:if>

</h2>

</caption>

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

<input type=*"hidden"* name=*"id"* value=*"*<c:out value='${user.id}' />*"* />

</c:if>

<tr>

<th>User Name: </th>

<td>

<input type=*"text"* name=*"name"* size=*"45"*

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

/>

</td>

</tr>

<tr>

<th>User Email: </th>

<td>

<input type=*"text"* name=*"email"* size=*"45"*

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

/>

</td>

</tr>

<tr>

<th>Country: </th>

<td>

<input type=*"text"* name=*"country"* size=*"15"*

value=*"*<c:out value='${user.country}' />*"*

/>

</td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*>

<input type=*"submit"* value=*"Save"* />

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

* **user-list.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

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

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

<%@page import = *"java.util.logging.Logger"* %>

<html>

<head>

<title>User Management Application</title>

</head>

<body>

<% Logger logger = Logger.getLogger(**this**.getClass().getName());%>

<a href=*"../home"*>HOME PAGE</a>

&nbsp;&nbsp;&nbsp

<center>

<h1>User Management</h1>

<h2>

<a href=*"new"*>Add New User</a>

&nbsp;&nbsp;&nbsp;

<a href=*"list"*>List All Users</a>

</h2>

</center>

<div align=*"center"*>

<table border=*"1"* cellpadding=*"5"*>

<caption><h2>List of Users</h2></caption>

<tr>

<th>ID</th>

<th>Name</th>

<th>Password</th>

<th>Email</th>

<th>Country</th>

<th>Actions</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.password}*"* /></td>

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

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

<td>

<a href=*"edit?id=*<c:out value='${user.id}' />*"*>Edit</a>

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

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

</td>

</tr>

</br>

</c:forEach>

</table>

</div>

</body>

</html>

**METAINF 🡪Maven🡪SchoolManagement🡪com.swaraj🡪pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>SchoolManagement</groupId>

<artifactId>com.swaraj</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>com.swaraj Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.10.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-commons-annotations -->

<!-- <dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-commons-annotations</artifactId>

<version>3.2.0.Final</version>

</dependency> -->

<!-- https://mvnrepository.com/artifact/javax.persistence/javax.persistence-api -->

<dependency>

<groupId>javax.persistence</groupId>

<artifactId>javax.persistence-api</artifactId>

<version>2.2</version>

</dependency>

<dependency>

<groupId>javax.activation</groupId>

<artifactId>javax.activation-api</artifactId>

<version>1.2.0</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

<version>8.0.13</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.10</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<finalName>com.swaraj</finalName>

</build>

</project>