package com.example.Phase2.dao;  
  
import com.example.Phase2.dbConfig.SimplilearnConfig;  
import com.example.Phase2.entity.Login;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.query.Query;  
  
import java.util.List;  
  
public class LoginDao {  
  
 private SessionFactory sessionFactory;  
  
 public LoginDao() {  
 sessionFactory = SimplilearnConfig.*getSessionFactory*();  
 }  
  
 public List<Login> getLoginFromEmail(String email){  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.beginTransaction();  
  
 String sql = "SELECT \* FROM Login WHERE email = :email";  
 Query<Login> query = session.createNativeQuery(sql, Login.class);  
 query.setParameter("email", email);  
  
 List<Login> logins = query.getResultList();  
  
 transaction.commit();  
 session.close();  
  
 return logins;  
 }  
}

package com.example.Phase2.dao;  
  
import com.example.Phase2.dbConfig.SimplilearnConfig;  
import com.example.Phase2.entity.SchoolClass;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.exception.ConstraintViolationException;  
import org.hibernate.query.Query;  
  
import java.util.Collections;  
import java.util.List;  
import java.util.Objects;  
  
public class SchoolClassDao {  
 private SessionFactory sessionFactory;  
  
 public SchoolClassDao() {  
 sessionFactory = SimplilearnConfig.*getSessionFactory*();  
 }  
  
 public boolean isExist(SchoolClass schoolClass){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 SchoolClass existSchoolClass = session.find(SchoolClass.class, schoolClass);  
 return Objects.*nonNull*(existSchoolClass);  
  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
  
 public boolean insert(SchoolClass schoolClass){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 session.save(schoolClass);  
  
 transaction.commit();  
 return true;  
 }catch (ConstraintViolationException e){  
 return false;  
 }  
 }  
  
 public List<SchoolClass> getAll(){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Query<SchoolClass> query = session.createQuery("from SchoolClass", SchoolClass.class);  
 List<SchoolClass> queries = query.getResultList();  
  
 session.close();  
  
 return queries;  
  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
 }  
  
 public SchoolClass getSchoolClassById(int id){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 SchoolClass schoolClass = session.get(SchoolClass.class, id);  
  
 session.close();  
 return schoolClass;  
 }catch (Exception e){  
 return null;  
 }  
  
 }  
}

package com.example.Phase2.dao;  
  
import com.example.Phase2.dbConfig.SimplilearnConfig;  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.exception.ConstraintViolationException;  
import org.hibernate.query.Query;  
  
import java.util.Collections;  
import java.util.List;  
import java.util.Objects;  
  
public class StudentDao {  
  
 private SessionFactory sessionFactory;  
  
 public StudentDao() {  
 sessionFactory = SimplilearnConfig.*getSessionFactory*();  
 }  
  
 public boolean isExist(Student student){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 Student existStudent = session.find(Student.class, student);  
 return Objects.*nonNull*(existStudent);  
  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
  
 public boolean insert(Student student){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 session.save(student);  
  
 transaction.commit();  
 return true;  
 }catch (ConstraintViolationException e){  
 return false;  
 }  
 }  
  
 public List<Student> getAll(){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Query<Student> query = session.createQuery("from Student", Student.class);  
 List<Student> queries = query.getResultList();  
  
 session.close();  
  
 return queries;  
  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
 }  
  
 public boolean assignStudentToClass(Student student, SchoolClass schoolClass){  
 try {  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Student existStudent = session.find(Student.class, student.getStudentId());  
 SchoolClass existClass = session.find(SchoolClass.class, schoolClass.getClassId());  
  
 existStudent.setSchoolClass(existClass);  
 session.saveOrUpdate(existStudent);  
  
 transaction.commit();  
 session.close();  
 return true;  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
 public Student getStudentById(int id){  
 try{  
  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Student student = session.get(Student.class, id);  
 return student;  
 }catch (Exception e){  
 return null;  
 }  
  
 }  
  
 public List<Student> getStudentByClassId(int id){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 String studentSql = "SELECT s FROM Student s WHERE s.schoolClass.classId = :id";  
 Query<Student> studentQuery = session.createQuery(studentSql, Student.class);  
 studentQuery.setParameter("id", id);  
 List<Student> students = studentQuery.getResultList();  
  
 session.close();  
 return students;  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
  
 }  
}

package com.example.Phase2.dao;  
  
import com.example.Phase2.dbConfig.SimplilearnConfig;  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.entity.Teacher;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.exception.ConstraintViolationException;  
import org.hibernate.query.Query;  
  
import java.util.Collections;  
import java.util.List;  
import java.util.Objects;  
  
public class SubjectDao {  
  
 private SessionFactory sessionFactory;  
  
 public SubjectDao() {  
 sessionFactory = SimplilearnConfig.*getSessionFactory*();  
 }  
  
 public boolean isExist(Subject subject){  
 try (Session session = sessionFactory.openSession()) {  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 Subject existSubject = session.find(Subject.class, subject);  
 return Objects.*nonNull*(existSubject);  
  
 } catch (Exception e) {  
 return false;  
 }  
 }  
  
  
 public boolean insert(Subject subject){  
 try (Session session = sessionFactory.openSession()) {  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 session.save(subject);  
  
 transaction.commit();  
 return true;  
 } catch (ConstraintViolationException e) {  
 return false;  
 }  
 }  
  
 public List<Subject> getAll(){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Query<Subject> query = session.createQuery("from Subject", Subject.class);  
 List<Subject> queries = query.getResultList();  
  
 session.close();  
  
 return queries;  
  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
 }  
  
 public boolean assignSubjectToClass(Subject subject, SchoolClass schoolClass){  
 try {  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Subject existSubject = session.find(Subject.class, subject.getSubjectId());  
 SchoolClass existClass = session.find(SchoolClass.class, schoolClass.getClassId());  
  
 existSubject.setSchoolClass(existClass);  
 session.saveOrUpdate(existSubject);  
  
 transaction.commit();  
 session.close();  
 return true;  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
 public boolean assignSubjectToTeacher(Subject subject, Teacher teacher){  
 try {  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Subject existSubject = session.find(Subject.class, subject.getSubjectId());  
 Teacher existTeacher = session.find(Teacher.class, teacher.getTeacherId());  
  
 existSubject.setTeacher(existTeacher);  
 session.saveOrUpdate(existSubject);  
  
 transaction.commit();  
 session.close();  
 return true;  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
 public Subject getSubjectById(int id){  
 try{  
  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Subject subject = session.get(Subject.class, id);  
 return subject;  
 }catch (Exception e){  
 return null;  
 }  
  
 }  
  
 public List<Subject> getSubjectByClassId(int id){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 String subjectSql = "SELECT sub FROM Subject sub WHERE sub.schoolClass.classId = :id";  
 Query<Subject> subjectQuery = session.createQuery(subjectSql, Subject.class);  
 subjectQuery.setParameter("id", id);  
 List<Subject> subjects = subjectQuery.getResultList();  
  
 session.close();  
  
 return subjects;  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
  
 }  
  
}

package com.example.Phase2.dao;  
  
import com.example.Phase2.dbConfig.SimplilearnConfig;  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Teacher;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.exception.ConstraintViolationException;  
import org.hibernate.query.Query;  
  
import java.util.Collections;  
import java.util.List;  
import java.util.Objects;  
  
public class TeacherDao {  
 private SessionFactory sessionFactory;  
  
 public TeacherDao() {  
 sessionFactory = SimplilearnConfig.*getSessionFactory*();  
 }  
  
 public boolean isExist(Teacher teacher){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 Teacher existTeacher = session.find(Teacher.class, teacher);  
 return Objects.*nonNull*(existTeacher);  
  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
  
 public boolean insert(Teacher teacher){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
  
 session.save(teacher);  
  
 transaction.commit();  
 return true;  
 }catch (ConstraintViolationException e){  
 return false;  
 }  
 }  
  
 public List<Teacher> getAll(){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Query<Teacher> query = session.createQuery("from Teacher", Teacher.class);  
 List<Teacher> queries = query.getResultList();  
  
 session.close();  
  
 return queries;  
  
 }catch (Exception e){  
 return Collections.*EMPTY\_LIST*;  
 }  
 }  
  
 public boolean assignTeacherToClass(Teacher teacher, SchoolClass schoolClass){  
 try {  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Teacher existTeacher = session.find(Teacher.class, teacher.getTeacherId());  
 SchoolClass existClass = session.find(SchoolClass.class, schoolClass.getClassId());  
  
 existTeacher.setSchoolClass(existClass);  
 session.saveOrUpdate(existTeacher);  
  
 transaction.commit();  
 session.close();  
 return true;  
 }catch (Exception e){  
 return false;  
 }  
 }  
  
 public Teacher getTeacherById(int id){  
 try{  
  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 Teacher teacher = session.get(Teacher.class, id);  
 return teacher;  
 }catch (Exception e){  
 return null;  
 }  
  
 }  
  
 public List<Teacher> getTeacherByClassId(int id){  
 try{  
 Session session = sessionFactory.openSession();  
 Transaction transaction = session.getTransaction();  
 transaction.begin();  
  
 String teacherSql = "SELECT t FROM Teacher t WHERE t.schoolClass.classId = :id";  
 Query<Teacher> teacherQuery = session.createQuery(teacherSql, Teacher.class);  
 teacherQuery.setParameter("id", id);  
 List<Teacher> teachers = teacherQuery.getResultList();  
  
 session.close();  
  
 return teachers;  
 }catch (Exception e){  
 return null;  
 }  
  
 }  
}

package com.example.Phase2.dbConfig;  
  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class SimplilearnConfig {  
 public static SessionFactory getSessionFactory() {  
 Configuration con = new Configuration();  
 con.configure("META-INF/hibernate.cfg.xml");  
 return con.buildSessionFactory();  
 }  
}

package com.example.Phase2.domain;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.entity.Teacher;  
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import java.util.List;  
  
@NoArgsConstructor  
@AllArgsConstructor  
@Builder  
@Data  
public class ClassReport {  
 private SchoolClass schoolClass;  
 private List<Student> students;  
 private List<Teacher> teachers;  
 private List<Subject> subjects;  
  
}

package com.example.Phase2.entity;  
  
import lombok.Data;  
  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
  
@Data  
@Entity  
public class Login {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*) // auto\_increment  
 private int id;  
  
 private String email;  
 private String password;  
}

package com.example.Phase2.entity;  
  
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import java.io.Serializable;  
  
@AllArgsConstructor  
@NoArgsConstructor  
@Builder  
@Entity  
@Data  
@Table(name = "school\_class")  
public class SchoolClass implements Serializable {  
  
 @Id  
 @Column(name = "class\_id")  
 private int classId;  
  
 @Column(name = "class\_name")  
 private String className;  
  
}

package com.example.Phase2.entity;  
  
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
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 java.io.Serializable;  
  
@AllArgsConstructor  
@NoArgsConstructor  
@Builder  
@Data  
@Entity  
public class Student implements Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "student\_id")  
 private int studentId;  
  
 @Column(name = "student\_name")  
 private String studentName;  
  
 @Column(name = "student\_surname")  
 private String studentSurname;  
  
 @ManyToOne  
 @JoinColumn(name = "class\_id")  
 private SchoolClass schoolClass;  
}

package com.example.Phase2.entity;  
  
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
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 java.io.Serializable;  
  
@AllArgsConstructor  
@NoArgsConstructor  
@Builder  
@Data  
@Entity  
public class Subject implements Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "subject\_id")  
 private int subjectId;  
  
 @Column(name = "subject\_name")  
 private String subjectName;  
  
 @ManyToOne  
 @JoinColumn(name = "class\_id")  
 private SchoolClass schoolClass;  
  
 @ManyToOne  
 @JoinColumn(name = "teacher\_id")  
 private Teacher teacher;  
  
}

package com.example.Phase2.entity;  
  
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
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.ManyToOne;  
import javax.persistence.OneToMany;  
import java.io.Serializable;  
import java.util.List;  
  
@AllArgsConstructor  
@NoArgsConstructor  
@Builder  
@Data  
@Entity  
public class Teacher implements Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "teacher\_id")  
 private int teacherId;  
  
 @Column(name = "teacher\_name")  
 private String teacherName;  
  
 @Column(name = "teacher\_surname")  
 private String teacherSurname;  
  
 @ManyToOne  
 @JoinColumn(name = "class\_id")  
 private SchoolClass schoolClass;  
  
}

package com.example.Phase2.service;  
  
  
import com.example.Phase2.dao.LoginDao;  
import com.example.Phase2.dao.SchoolClassDao;  
import com.example.Phase2.dao.StudentDao;  
import com.example.Phase2.dao.SubjectDao;  
import com.example.Phase2.dao.TeacherDao;  
  
import java.util.logging.Logger;  
  
public class BaseService {  
 protected static Logger *logger* = Logger.*getLogger*(BaseService.class.getName());  
 protected LoginDao loginDao = new LoginDao();  
 protected SchoolClassDao schoolClassDao = new SchoolClassDao();  
 protected SubjectDao subjectDao = new SubjectDao();  
 protected TeacherDao teacherDao = new TeacherDao();  
 protected StudentDao studentDao = new StudentDao();  
}

package com.example.Phase2.service;  
  
import com.example.Phase2.entity.Login;  
  
import java.util.List;  
  
public class LoginService extends BaseService{  
  
 public boolean checkUserEmail(String email){  
 *logger*.info("[checkUserEmail] started!");  
 try {  
 List<Login> logins = loginDao.getLoginFromEmail(email);  
 return !logins.isEmpty();  
 }catch (Exception e){  
 *logger*.info("[checkUserEmail] exception occurred!"+ e);  
 return false;  
 }  
 }  
  
 public boolean checkUserPassword(String email, String password){  
 try{  
 *logger*.info("[checkUserPassword] started!");  
 List<Login> logins = loginDao.getLoginFromEmail(email);  
 return logins.get(0).getPassword().equals(password);  
 }catch (Exception e){  
 *logger*.info("[checkUserPassword] exception occurred!"+ e);  
 return false;  
 }  
 }  
}

package com.example.Phase2.service;  
  
import com.example.Phase2.domain.ClassReport;  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.entity.Teacher;  
  
import java.util.List;  
  
public class SchoolClassService extends BaseService {  
  
 public boolean addNewSchoolClass(int id, String name) {  
 try {  
 SchoolClass schoolClass = SchoolClass.*builder*().classId(id).className(name).build();  
 if (!schoolClassDao.isExist(schoolClass)) {  
 boolean isInserted = schoolClassDao.insert(schoolClass);  
 return isInserted;  
 }  
 return false;  
 } catch (Exception e) {  
 //todo: change log status  
 *logger*.info("Exception occurred! " + e);  
 return false;  
 }  
 }  
  
 public List<SchoolClass> getAllClass() {  
 return schoolClassDao.getAll();  
 }  
  
 public ClassReport getClassReportById(int id) {  
  
 SchoolClass schoolClass = schoolClassDao.getSchoolClassById(id);  
 List<Student> students = studentDao.getStudentByClassId(id);  
 List<Teacher> teachers = teacherDao.getTeacherByClassId(id);  
 List<Subject> subjects = subjectDao.getSubjectByClassId(id);  
  
 return ClassReport.*builder*().schoolClass(schoolClass).subjects(subjects).teachers(teachers).students(students).build();  
 }  
}

package com.example.Phase2.service;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Teacher;  
  
import java.util.List;  
import java.util.Objects;  
  
public class StudentService extends BaseService{  
 public boolean addNewStudent(String studentName, String studentSurname) {  
 try{  
 Student student = Student.*builder*().studentName(studentName).studentSurname(studentSurname).build();  
 if(!studentDao.isExist(student)){  
 boolean isInserted = studentDao.insert(student);  
 return isInserted;  
 }  
 return false;  
 }catch (Exception e){  
 //todo: change log status  
 *logger*.info("Exception occurred! " +e);  
 return false;  
 }  
 }  
  
 public List<Student> getAllStudent() {  
 return studentDao.getAll();  
 }  
  
  
 public boolean assignStudentToClass(int studentId, int classId){  
 Student student = studentDao.getStudentById(studentId);  
 SchoolClass schoolClass = schoolClassDao.getSchoolClassById(classId);  
  
 if(Objects.*nonNull*(student) && Objects.*nonNull*(schoolClass)){  
 return studentDao.assignStudentToClass(student, schoolClass);  
 }  
 return false;  
 }  
}

package com.example.Phase2.service;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.entity.Teacher;  
  
import java.util.List;  
import java.util.Objects;  
  
public class SubjectService extends BaseService {  
  
 public boolean addNewSubject(String subjectName) {  
 try{  
  
 Subject subject = Subject.*builder*().subjectName(subjectName).build();  
 if(!subjectDao.isExist(subject)){  
 boolean isInserted = subjectDao.insert(subject);  
 return isInserted;  
 }  
 return false;  
 }catch (Exception e){  
 //todo: change log status  
 *logger*.info("Exception occurred! " +e);  
 return false;  
 }  
 }  
  
 public List<Subject> getAllSubject() {  
 return subjectDao.getAll();  
 }  
  
 public boolean assignSubjectToClass(int subjectId, int classId){  
 Subject subject = subjectDao.getSubjectById(subjectId);  
 SchoolClass schoolClass = schoolClassDao.getSchoolClassById(classId);  
  
 if(Objects.*nonNull*(subject) && Objects.*nonNull*(schoolClass)){  
 return subjectDao.assignSubjectToClass(subject, schoolClass);  
 }  
 return false;  
 }  
  
 public boolean assignSubjectToTeacher(int subjectId, int classId){  
 Subject subject = subjectDao.getSubjectById(subjectId);  
 Teacher teacher = teacherDao.getTeacherById(classId);  
  
 if(Objects.*nonNull*(subject) && Objects.*nonNull*(teacher)){  
 return subjectDao.assignSubjectToTeacher(subject, teacher);  
 }  
 return false;  
 }  
}

package com.example.Phase2.service;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Teacher;  
  
import java.util.List;  
import java.util.Objects;  
  
public class TeacherService extends BaseService{  
  
 public boolean addNewTeacher(String teacherName, String teacherSurname) {  
 try{  
 Teacher teacher = Teacher.*builder*().teacherName(teacherName).teacherSurname(teacherSurname).build();  
 if(!teacherDao.isExist(teacher)){  
 boolean isInserted = teacherDao.insert(teacher);  
 return isInserted;  
 }  
 return false;  
 }catch (Exception e){  
 *logger*.info("Exception occurred! " +e);  
 return false;  
 }  
 }  
  
 public List<Teacher> getAllTeacher() {  
 return teacherDao.getAll();  
 }  
  
 public boolean assignTeacherToClass(int teacherId, int classId){  
 Teacher teacher = teacherDao.getTeacherById(teacherId);  
 SchoolClass schoolClass = schoolClassDao.getSchoolClassById(classId);  
  
 if(Objects.*nonNull*(teacher) && Objects.*nonNull*(schoolClass)){  
 return teacherDao.assignTeacherToClass(teacher, schoolClass);  
 }  
 return false;  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.service.SchoolClassService;  
import com.example.Phase2.service.StudentService;  
  
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.util.List;  
  
public class AssignStudentToClassServlet extends HttpServlet {  
  
 private StudentService studentService = new StudentService();  
 private SchoolClassService schoolClassService = new SchoolClassService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Student> students = studentService.getAllStudent();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 request.setAttribute("students", students);  
 request.setAttribute("schoolClasses", schoolClasses);  
 request.getRequestDispatcher("assignStudentToClass.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 int studentId = Integer.*parseInt*(request.getParameter("studentId"));  
 int classId = Integer.*parseInt*(request.getParameter("classId"));  
  
 boolean isAdded = studentService.assignStudentToClass(studentId, classId);  
 List<Student> students = studentService.getAllStudent();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Student assigned!");  
 request.setAttribute("students", students);  
 request.setAttribute("schoolClasses", schoolClasses);  
 } else {  
 request.setAttribute("errorMessage", "Student cannot be assigned!");  
 request.setAttribute("students", students);  
 request.setAttribute("schoolClasses", schoolClasses);  
 }  
 request.getRequestDispatcher("assignStudentToClass.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.service.SchoolClassService;  
import com.example.Phase2.service.StudentService;  
import com.example.Phase2.service.SubjectService;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.IOException;  
import java.util.List;  
  
public class AssignSubjectToClassServlet extends HttpServlet {  
  
 private SubjectService subjectService = new SubjectService();  
 private SchoolClassService schoolClassService = new SchoolClassService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Subject> subjects = subjectService.getAllSubject();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("schoolClasses", schoolClasses);  
 request.getRequestDispatcher("assignSubjectToClass.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 int subjectId = Integer.*parseInt*(request.getParameter("subjectId"));  
 int classId = Integer.*parseInt*(request.getParameter("classId"));  
  
 boolean isAdded = subjectService.assignSubjectToClass(subjectId, classId);  
 List<Subject> subjects = subjectService.getAllSubject();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Subject assigned!");  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("schoolClasses", schoolClasses);  
 } else {  
 request.setAttribute("errorMessage", "Subject cannot be assigned!");  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("schoolClasses", schoolClasses);  
 }  
 request.getRequestDispatcher("assignSubjectToClass.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.entity.Teacher;  
import com.example.Phase2.service.SubjectService;  
import com.example.Phase2.service.TeacherService;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.IOException;  
import java.util.List;  
  
public class AssignSubjectToTeacherServlet extends HttpServlet {  
  
 private SubjectService subjectService = new SubjectService();  
 private TeacherService teacherService = new TeacherService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Subject> subjects = subjectService.getAllSubject();  
 List<Teacher> teachers = teacherService.getAllTeacher();  
  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("teachers", teachers);  
 request.getRequestDispatcher("assignSubjectToTeacher.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 int subjectId = Integer.*parseInt*(request.getParameter("subjectId"));  
 int teacherId = Integer.*parseInt*(request.getParameter("teacherId"));  
  
 boolean isAdded = subjectService.assignSubjectToTeacher(subjectId, teacherId);  
 List<Subject> subjects = subjectService.getAllSubject();  
 List<Teacher> teachers = teacherService.getAllTeacher();  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Subject assigned!");  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("teachers", teachers);  
 } else {  
 request.setAttribute("errorMessage", "Subject cannot be assigned!");  
 request.setAttribute("subjects", subjects);  
 request.setAttribute("teachers", teachers);  
 }  
 request.getRequestDispatcher("assignSubjectToTeacher.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Teacher;  
import com.example.Phase2.service.SchoolClassService;  
import com.example.Phase2.service.StudentService;  
import com.example.Phase2.service.TeacherService;  
  
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.util.List;  
  
public class AssignTeacherToClassServlet extends HttpServlet {  
  
 private TeacherService teacherService = new TeacherService();  
 private SchoolClassService schoolClassService = new SchoolClassService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Teacher> teachers = teacherService.getAllTeacher();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 request.setAttribute("teachers", teachers);  
 request.setAttribute("schoolClasses", schoolClasses);  
 request.getRequestDispatcher("assignTeacherToClass.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 int teacherId = Integer.*parseInt*(request.getParameter("teacherId"));  
 int classId = Integer.*parseInt*(request.getParameter("classId"));  
  
 boolean isAdded = teacherService.assignTeacherToClass(teacherId, classId);  
 List<Teacher> teachers = teacherService.getAllTeacher();  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Teacher assigned!");  
 request.setAttribute("teachers", teachers);  
 request.setAttribute("schoolClasses", schoolClasses);  
 } else {  
 request.setAttribute("errorMessage", "Teacher cannot be assigned!");  
 request.setAttribute("teachers", teachers);  
 request.setAttribute("schoolClasses", schoolClasses);  
 }  
 request.getRequestDispatcher("assignTeacherToClass.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.domain.ClassReport;  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Teacher;  
import com.example.Phase2.service.SchoolClassService;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import java.io.IOException;  
import java.util.List;  
  
public class ClassReportServlet extends HttpServlet {  
 private SchoolClassService schoolClassService = new SchoolClassService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 request.setAttribute("schoolClasses", schoolClasses);  
 request.getRequestDispatcher("classReport.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {  
 response.setContentType("text/html");  
 int classId = Integer.*parseInt*(request.getParameter("classId"));  
  
 ClassReport schoolClassReport= schoolClassService.getClassReportById(classId);  
  
 request.setAttribute("schoolClassReport", schoolClassReport);  
 request.getRequestDispatcher("report.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.service.SchoolClassService;  
  
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.util.List;  
  
public class ClassServlet extends HttpServlet {  
  
 private SchoolClassService schoolClassService = new SchoolClassService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<SchoolClass> schoolClasses = schoolClassService.getAllClass();  
  
 request.setAttribute("schoolClasses", schoolClasses);  
 request.getRequestDispatcher("viewClass.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {  
 response.setContentType("text/html");  
 String className = request.getParameter("className");  
 int classId = Integer.*parseInt*(request.getParameter("classId"));  
  
 boolean isAdded = schoolClassService.addNewSchoolClass(classId, className);  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Class added!");  
 } else {  
 request.setAttribute("errorMessage", "Class cannot be added!");  
 }  
 request.getRequestDispatcher("addClass.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.service.LoginService;  
  
import java.io.\*;  
import javax.servlet.ServletException;  
import javax.servlet.http.\*;  
  
public class LoginServlet extends HttpServlet {  
  
 private static final long *serialVersionUID* = 1L;  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 LoginService loginService = new LoginService();  
  
 String email = request.getParameter("email");  
 String password = request.getParameter("password");  
  
 if(loginService.checkUserEmail(email)){  
 if(loginService.checkUserPassword(email, password)){  
 request.getRequestDispatcher("home.jsp").forward(request, response);  
 }  
 else{  
 request.setAttribute("errorMessage", "Invalid password!");  
 request.getRequestDispatcher("login.jsp").forward(request, response);  
 }  
 }else{  
 request.setAttribute("errorMessage", "Invalid email!");  
 request.getRequestDispatcher("login.jsp").forward(request, response);  
 }  
  
 }  
  
 public void destroy() {  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.Student;  
import com.example.Phase2.entity.Teacher;  
import com.example.Phase2.service.StudentService;  
  
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.util.List;  
  
public class StudentServlet extends HttpServlet {  
  
 private StudentService studentService = new StudentService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Student> students = studentService.getAllStudent();  
  
 request.setAttribute("students", students);  
 request.getRequestDispatcher("viewStudent.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 String studentName = request.getParameter("studentName");  
 String studentSurname = request.getParameter("studentSurname");  
  
 boolean isAdded = studentService.addNewStudent(studentName, studentSurname);  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Student added!");  
 } else {  
 request.setAttribute("errorMessage", "Student cannot be added!");  
 }  
 request.getRequestDispatcher("addStudent.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.Subject;  
import com.example.Phase2.service.SubjectService;  
  
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.util.List;  
  
public class SubjectServlet extends HttpServlet {  
  
 private SubjectService subjectService = new SubjectService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Subject> subjects = subjectService.getAllSubject();  
  
 request.setAttribute("subjects", subjects);  
 request.getRequestDispatcher("viewSubject.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 String subjectName = request.getParameter("subjectName");  
  
 boolean isAdded = subjectService.addNewSubject(subjectName);  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Subject added!");  
 } else {  
 request.setAttribute("errorMessage", "Subject cannot be added!");  
 }  
 request.getRequestDispatcher("addSubject.jsp").forward(request, response);  
  
 }  
}

package com.example.Phase2.servlet;  
  
import com.example.Phase2.entity.SchoolClass;  
import com.example.Phase2.entity.Teacher;  
import com.example.Phase2.service.TeacherService;  
  
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.util.List;  
  
public class TeacherServlet extends HttpServlet {  
  
 TeacherService teacherService = new TeacherService();  
  
 @Override  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 List<Teacher> teachers = teacherService.getAllTeacher();  
  
 request.setAttribute("teachers", teachers);  
 request.getRequestDispatcher("viewTeacher.jsp").forward(request, response);  
  
 }  
  
 @Override  
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 String teacherName = request.getParameter("teacherName");  
 String teacherSurname = request.getParameter("teacherSurname");  
  
 boolean isAdded = teacherService.addNewTeacher(teacherName, teacherSurname);  
  
 if (isAdded) {  
 request.setAttribute("infoMessage", "Teacher added!");  
 } else {  
 request.setAttribute("errorMessage", "Teacher cannot be added!");  
 }  
 request.getRequestDispatcher("addTeacher.jsp").forward(request, response);  
  
 }  
}

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>  
 <property name="hibernate.connection.password">2022\_password</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/simplilearn?useUnicode=true&amp;serverTimezone=GMT</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.hbm2ddl.auto">update</property>  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>  
 <mapping class="com.example.Phase2.entity.Login"/>  
 <mapping class="com.example.Phase2.entity.SchoolClass"/>  
 <mapping class="com.example.Phase2.entity.Student"/>  
 <mapping class="com.example.Phase2.entity.Subject"/>  
 <mapping class="com.example.Phase2.entity.Teacher"/>  
 </session-factory>  
</hibernate-configuration>

Add.css

.center {  
 margin: auto;  
 width: 60%;  
 border: 3px solid black;  
 padding: 20px;  
 position: absolute;  
 top: 40%;  
 left: 20%;  
}  
div{  
 border-radius: 3px;  
 padding : 10px;  
 text-align: center;  
 background-color: antiquewhite;  
 font-size: 15px;  
}  
body {  
 text-align: center;  
}

Home.css

.center {  
 margin: auto;  
 width: 60%;  
 border: 3px solid black;  
 padding: 10px;  
 position: absolute;  
 top: 30%;  
 left: 20%;  
 border-radius: 3px;  
 text-align: center;  
 background-color: antiquewhite;  
 font-size: 15px;  
}  
body {  
 text-align: center;  
}  
.center a.aToButton {  
 display: inline-block;  
 padding: 10px 20px;  
 background-color: #7a6b5d;  
 color: #fff;  
 text-align: center;  
 text-decoration: none;  
 font-size: 16px;  
 border-radius: 5px;  
 border: gray;  
 cursor: pointer;  
 margin: 2px;  
}

Login.css

.center {  
 margin: auto;  
 width: 60%;  
 border: 3px solid black;  
 padding: 10px;  
 position: absolute;  
 top: 40%;  
 left: 20%;  
}  
div{  
 border-radius: 3px;  
 padding : 10px;  
 text-align: center;  
 background-color: antiquewhite;  
 font-size: 15px;  
}  
body {  
 text-align: center;  
}

viewClass.css

.center {  
 margin: auto;  
 width: 60%;  
 border: 3px solid black;  
 padding: 10px;  
 position: absolute;  
 top: 30%;  
 left: 20%;  
 border-radius: 3px;  
 text-align: center;  
 background-color: antiquewhite;  
 font-size: 15px;  
}  
  
div table{  
 margin: auto;  
 width: 50%;  
}  
div th, td{  
 border-radius: 5px;  
 background-color: #7a6b5d;  
 color: #fff;  
 padding: 2px;  
 text-align: center;  
 margin: auto;  
}  
  
body {  
 text-align: center;  
}

<%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 16:45  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Add Class</title>  
 <link rel="stylesheet" href="stylesheets/addClass.css">  
</head>  
<body>  
 <h1>You can add class in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
 <form action="/addClass" method="post">  
 <label>Class Id:</label>  
 <input name="classId"/>  
 <br><br>  
 <label>Class Name: </label>  
 <input name="className"/>  
 <br><br>  
 <button name="add-class" type="submit">Add Class</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 16:45  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Add Student</title>  
 <link rel="stylesheet" href="stylesheets/addClass.css">  
</head>  
<body>  
 <h1>You can add student in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
  
 <form action="/addStudent" method="post">  
 <label>Student Name: </label>  
 <input name="studentName"/>  
 <br><br>  
 <label>Student Surname: </label>  
 <input name="studentSurname"/>  
 <br><br>  
 <button name="add-student" type="submit">Add Student</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 16:45  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Add Subject</title>  
 <link rel="stylesheet" href="stylesheets/addClass.css">  
</head>  
<body>  
 <h1>You can add subject in here!</h1>  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
 <form action="/addSubject" method="post">  
  
 <label>Subject: </label>  
 <input name="subjectName"/>  
 <br><br>  
 <button name="subject-class" type="submit">Add Subject</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 16:45  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Add Teacher</title>  
 <link rel="stylesheet" href="stylesheets/addClass.css">  
</head>  
<body>  
 <h1>You can add teacher in here!</h1>  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
  
 <form action="/addTeacher" method="post">  
 <label>Teacher Name: </label>  
 <input name="teacherName"/>  
 <br><br>  
 <label>Teacher Surname: </label>  
 <input name="teacherSurname"/>  
 <br><br>  
 <button name="add-teacher" type="submit">Add Teacher</button>  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="com.example.Phase2.entity.SchoolClass" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 3.03.2023  
 Time: 17:14  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Assign Student To Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<h1>You can assign student to class in here!</h1>  
<body>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
 <br><br>  
 <form action="/assignStudent" method="post">  
  
 <label>Student: </label>  
 <select name="studentId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<Student> students = (List<Student>) request.getAttribute("students");  
 for(Student student : students){ %>  
 <option value="<%=student.getStudentId()%>"><%=student.getStudentName() + " " + student.getStudentSurname()%></option>  
 <%}%>  
 </select>  
  
 <br><br>  
 <label>Class Name: </label>  
 <select name="classId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<SchoolClass> schoolClasses = (List<SchoolClass>) request.getAttribute("schoolClasses");  
 for(SchoolClass schoolClass : schoolClasses){ %>  
 <option value="<%=schoolClass.getClassId()%>"><%=schoolClass.getClassName()%></option>  
 <%}%>  
 </select>  
 <br><br>  
 <button name="assign-student-to-class" type="submit">Assign Student To Class</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="com.example.Phase2.entity.Subject" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 3.03.2023  
 Time: 17:14  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Assign Subject To Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<h1>You can assign subject to class in here!</h1>  
<body>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
  
 <br><br>  
 <form action="/assignSubject" method="post">  
  
 <label>Subject: </label>  
 <select name="subjectId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<Subject> subjects = (List<Subject>) request.getAttribute("subjects");  
 for(Subject subject : subjects){ %>  
 <option value="<%=subject.getSubjectId()%>"><%=subject.getSubjectName()%></option>  
 <%}%>  
 </select>  
  
 <br><br>  
 <label>Class Name: </label>  
 <select name="classId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<SchoolClass> schoolClasses = (List<SchoolClass>) request.getAttribute("schoolClasses");  
 for(SchoolClass schoolClass : schoolClasses){ %>  
 <option value="<%=schoolClass.getClassId()%>"><%=schoolClass.getClassName()%></option>  
 <%}%>  
 </select>  
 <br><br>  
 <button name="assign-subject-to-class" type="submit">Assign Subject To Class</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="com.example.Phase2.entity.Subject" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 3.03.2023  
 Time: 17:14  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Assign Subject To Teacher</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<h1>You can assign subject to teacher in here!</h1>  
<body>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
  
 <br><br>  
 <form action="/assignSubjectTeacher" method="post">  
  
 <label>Subject: </label>  
 <select name="subjectId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<Subject> subjects = (List<Subject>) request.getAttribute("subjects");  
 for(Subject subject : subjects){ %>  
 <option value="<%=subject.getSubjectId()%>"><%=subject.getSubjectName()%></option>  
 <%}%>  
 </select>  
  
 <br><br>  
 <label>Teacher Name: </label>  
 <select name="teacherId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<Teacher> teachers = (List<Teacher>) request.getAttribute("teachers");  
 for(Teacher teacher : teachers){ %>  
 <option value="<%=teacher.getTeacherId()%>"><%=teacher.getTeacherName() + " " + teacher.getTeacherSurname()%></option>  
 <%}%>  
 </select>  
 <br><br>  
 <button name="assign-subject-to-teacher" type="submit">Assign Subject To Teacher</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 3.03.2023  
 Time: 17:14  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Assign Teacher To Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<h1>You can assign teacher to class in here!</h1>  
<body>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
  
 <br><br>  
 <form action="/assignTeacher" method="post">  
  
 <label>Teacher: </label>  
 <select name="teacherId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<Teacher> teachers = (List<Teacher>) request.getAttribute("teachers");  
 for(Teacher teacher : teachers){ %>  
 <option value="<%=teacher.getTeacherId()%>"><%=teacher.getTeacherName() + " " + teacher.getTeacherSurname()%></option>  
 <%}%>  
 </select>  
  
 <br><br>  
 <label>Class Name: </label>  
 <select name="classId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<SchoolClass> schoolClasses = (List<SchoolClass>) request.getAttribute("schoolClasses");  
 for(SchoolClass schoolClass : schoolClasses){ %>  
 <option value="<%=schoolClass.getClassId()%>"><%=schoolClass.getClassName()%></option>  
 <%}%>  
 </select>  
 <br><br>  
 <button name="assign-teacher-to-class" type="submit">Assign Teacher To Class</button>  
  
 <br><br>  
 <%  
 Object message = request.getAttribute("infoMessage");  
 if (message != null) { %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 <%  
 Object errorMessage = request.getAttribute("errorMessage");  
 if (errorMessage != null) { %>  
 <span style="color:red;"><%=errorMessage.toString()%></span>  
 <%}%>  
  
 </form>  
 </div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view class report in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
  
 <form action="/viewReport" method="post">  
 <label>Class Name: </label>  
 <select name="classId" required>  
 <option value="" disabled selected>Select an option</option>  
 <%List<SchoolClass> schoolClasses = (List<SchoolClass>) request.getAttribute("schoolClasses");  
 for(SchoolClass schoolClass : schoolClasses){ %>  
 <option value="<%=schoolClass.getClassId()%>"><%=schoolClass.getClassName()%></option>  
 <%}%>  
 </select>  
 <br><br>  
 <button type="submit">Get Class Report</button>  
 </form>  
  
  
 </div>  
</body>  
</html>

<%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 23.02.2023  
 Time: 23:41  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>HomePage</title>  
 <link rel="stylesheet" href="stylesheets/home.css">  
</head>  
<body>  
 <div class="center">  
 <a href="addClass.jsp" class="aToButton">Add Class</a>  
 <a href="/viewClass" class="aToButton">View Class</a>  
 <br>  
 <a href="addSubject.jsp" class="aToButton">Add Subject</a>  
 <a href="/viewSubject" class="aToButton">View Subject</a>  
 <br>  
 <a href="addTeacher.jsp" class="aToButton">Add Teacher</a>  
 <a href="/viewTeacher" class="aToButton">View Teacher</a>  
 <br>  
 <a href="addStudent.jsp" class="aToButton">Add student</a>  
 <a href="/viewStudent" class="aToButton">View Student</a>  
  
 <br>  
 <hr>  
  
 <a href="/assignStudentToClass" class="aToButton">Assign Student To Class</a>  
 <br>  
 <a href="/assignTeacherToClass" class="aToButton">Assign Teacher To Class</a>  
 <br>  
 <a href="/assignSubjectToClass" class="aToButton">Assign Subject To Class</a>  
 <br>  
 <!-- <a href="/assignSubjectToTeacher" class="aToButton">Assign Subject To Teacher</a>  
 <br>-->  
 <a href="/viewClassReport" class="aToButton">Class Report</a>  
 </div>  
  
</body>  
</html>

<%@ **page** import="org.hibernate.type.ObjectType" %>  
<%@ **page** contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html>  
<head>  
 <title>Student Portal</title>  
 <link rel="stylesheet" href="stylesheets/login.css">  
</head>  
<body>  
<h1><%= "Welcome to Student Portal!" %></h1>  
<hr>  
</form>  
<hr/>  
<form action="/loginServlet" method="post">  
  
 <div class="center">  
 <label>Email: </label>  
 <input type="email" name="email"/><br/>  
  
 <br>  
  
 <label>Password: </label>  
 <input type="password" name="password"/><br/>  
  
 <br>  
  
 <input type="submit" value="submit"/>  
 <input type="reset" value="reset"/>  
 <br>  
 <%  
 Object message = request.getAttribute("errorMessage");  
 if(message!= null){ %>  
 <span style="color:red;"><%=message.toString()%></span>  
 <%}%>  
 </div>  
  
</form>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.domain.ClassReport" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="com.example.Phase2.entity.Subject" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view class report in here!</h1>  
  
<a href="home.jsp">  
 Back To Home Page  
</a>  
<div class="center">  
  
 <%ClassReport classReport = (ClassReport) request.getAttribute("schoolClassReport");%>  
 <label>Class Id: </label>  
 <span><%=classReport.getSchoolClass().getClassId()%></span>  
 <br><br>  
 <label>Class Name: </label>  
 <span><%=classReport.getSchoolClass().getClassName()%></span>  
 <br><br>  
 <label>Students: </label>  
 <%  
 for(Student student : classReport.getStudents()){%>  
 <span><%=student.getStudentName() + " " + student.getStudentSurname()%></span>  
 <%}%>  
 <br><br>  
 <label>Teachers: </label>  
 <%  
 for(Teacher teacher : classReport.getTeachers()){%>  
 <span><%=teacher.getTeacherName() + " " + teacher.getTeacherSurname()%></span>  
 <%}%>  
 <br><br>  
 <label>Subjects: </label>  
 <%  
 for(Subject subject : classReport.getSubjects()){%>  
 <span><%=subject.getSubjectName() %></span>  
 <%}%>  
  
</div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>View Teacher</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>Student Added!</h1>  
  
 <a href="home.jsp">  
 Back To Student Assign Page  
 </a>  
 <div class="center">  
 <table>  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Surname</th>  
 </tr>  
 </thead>  
 <tbody>  
 <% List<Student> students = (List<Student>) request.getAttribute("students");  
 for (Student student : students) { %>  
 <tr>  
 <td><%= student.getStudentId() %>  
 </td>  
 <td><%= student.getStudentName() %>  
 </td>  
 <td><%= student.getStudentSurname() %>  
 </td>  
 </tr>  
 <% } %>  
 </tbody>  
 </table>  
 </div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>View Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view class in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
 <table>  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 </tr>  
 </thead>  
 <tbody>  
 <% List<SchoolClass> schoolClasses = (List<SchoolClass>) request.getAttribute("schoolClasses");  
 for (SchoolClass schoolClass : schoolClasses) { %>  
 <tr>  
 <td><%= schoolClass.getClassId() %>  
 </td>  
 <td><%= schoolClass.getClassName() %>  
 </td>  
 </tr>  
 <% } %>  
 </tbody>  
 </table>  
 </div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %>  
<%@ **page** import="com.example.Phase2.entity.Student" %>  
<%@ **page** import="java.util.Objects" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>View Teacher</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view student in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
 <table>  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Surname</th>  
 <th>Class</th>  
 </tr>  
 </thead>  
 <tbody>  
 <% List<Student> students = (List<Student>) request.getAttribute("students");  
 for (Student student : students) { %>  
 <tr>  
 <td><%= student.getStudentId() %>  
 </td>  
 <td><%= student.getStudentName() %>  
 </td>  
 <td><%= student.getStudentSurname() %>  
 </td>  
 <%if(Objects.*nonNull*(student.getSchoolClass())){%>  
 <td><%= student.getSchoolClass().getClassName() %></td>  
 <%}else{%>  
 <td></td>  
 <%}%>  
 </tr>  
 <% } %>  
 </tbody>  
 </table>  
 </div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Subject" %>  
<%@ **page** import="java.util.Objects" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>View Class</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view subjects in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
 <div class="center">  
 <table>  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Class</th>  
 <th>Teacher</th>  
 </tr>  
 </thead>  
 <tbody>  
 <% List<Subject> subjects = (List<Subject>) request.getAttribute("subjects");  
 for (Subject subject : subjects) { %>  
 <tr>  
 <td><%= subject.getSubjectId() %>  
 </td>  
 <td><%= subject.getSubjectName() %>  
 </td>  
 <%if(Objects.*nonNull*(subject.getSchoolClass())){%>  
 <td><%= subject.getSchoolClass().getClassName() %></td>  
 <%}else{%>  
 <td></td>  
 <%}%>  
 <%if(Objects.*nonNull*(subject.getTeacher())){%>  
 <td><%= subject.getTeacher().getTeacherName() + " " + subject.getTeacher().getTeacherSurname() %></td>  
 <%}else{%>  
 <td></td>  
 <%}%>  
 </tr>  
 <% } %>  
 </tbody>  
 </table>  
 </div>  
</body>  
</html>

<%@ **page** import="com.example.Phase2.entity.SchoolClass" %>  
<%@ **page** import="java.util.List" %>  
<%@ **page** import="com.example.Phase2.entity.Teacher" %>  
<%@ **page** import="java.util.Objects" %><%--  
 Created by IntelliJ IDEA.  
 User: ozge  
 Date: 1.03.2023  
 Time: 23:57  
 To change this template use File | Settings | File Templates.  
--%>  
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>View Teacher</title>  
 <link rel="stylesheet" href="stylesheets/viewClass.css">  
</head>  
<body>  
<h1>You can view teacher in here!</h1>  
  
 <a href="home.jsp">  
 Back To Home Page  
 </a>  
  
 <div class="center">  
 <table>  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Name</th>  
 <th>Surname</th>  
 <th>Class</th>  
 </tr>  
 </thead>  
 <tbody>  
 <% List<Teacher> teachers = (List<Teacher>) request.getAttribute("teachers");  
 for (Teacher teacher : teachers) { %>  
 <tr>  
 <td><%= teacher.getTeacherId() %>  
 </td>  
 <td><%= teacher.getTeacherName() %>  
 </td>  
 <td><%= teacher.getTeacherSurname() %>  
 </td>  
 <%if(Objects.*nonNull*(teacher.getSchoolClass())){%>  
 <td><%= teacher.getSchoolClass().getClassName() %></td>  
 <%}else{%>  
 <td></td>  
 <%}%>  
 </tr>  
 <% } %>  
 </tbody>  
 </table>  
 </div>  
</body>  
</html>

<?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" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" id="WebApp\_ID" version="2.5">  
 <display-name>SimpleServletApp</display-name>  
 <welcome-file-list>  
 <welcome-file>login.jsp</welcome-file>  
 </welcome-file-list>  
 <servlet>  
 <description></description>  
 <display-name>Login</display-name>  
 <servlet-name>LoginServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.LoginServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>LoginServlet</servlet-name>  
 <url-pattern>/loginServlet</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>AddClass</display-name>  
 <servlet-name>AddClassServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.ClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>AddClassServlet</servlet-name>  
 <url-pattern>/addClass</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>ViewClass</display-name>  
 <servlet-name>ViewClassServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.ClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>ViewClassServlet</servlet-name>  
 <url-pattern>/viewClass</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>AddSubject</display-name>  
 <servlet-name>AddSubjectServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.SubjectServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>AddSubjectServlet</servlet-name>  
 <url-pattern>/addSubject</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>ViewSubject</display-name>  
 <servlet-name>ViewSubjectServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.SubjectServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>ViewSubjectServlet</servlet-name>  
 <url-pattern>/viewSubject</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>AddTeacher</display-name>  
 <servlet-name>AddTeacherServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.TeacherServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>AddTeacherServlet</servlet-name>  
 <url-pattern>/addTeacher</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>ViewTeacher</display-name>  
 <servlet-name>ViewTeacherServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.TeacherServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>ViewTeacherServlet</servlet-name>  
 <url-pattern>/viewTeacher</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>AddStudent</display-name>  
 <servlet-name>AddStudentServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.StudentServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>AddStudentServlet</servlet-name>  
 <url-pattern>/addStudent</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>viewStudent</display-name>  
 <servlet-name>viewStudentServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.StudentServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>viewStudentServlet</servlet-name>  
 <url-pattern>/viewStudent</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignStudentToClass</display-name>  
 <servlet-name>assignStudentToClassServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignStudentToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignStudentToClassServlet</servlet-name>  
 <url-pattern>/assignStudentToClass</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignStudent</display-name>  
 <servlet-name>assignStudentServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignStudentToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignStudentServlet</servlet-name>  
 <url-pattern>/assignStudent</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignTeacherToClass</display-name>  
 <servlet-name>assignTeacherToClassServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignTeacherToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignTeacherToClassServlet</servlet-name>  
 <url-pattern>/assignTeacherToClass</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignTeacher</display-name>  
 <servlet-name>assignTeacherServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignTeacherToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignTeacherServlet</servlet-name>  
 <url-pattern>/assignTeacher</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignSubjectToClass</display-name>  
 <servlet-name>assignSubjectToClassServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignSubjectToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignSubjectToClassServlet</servlet-name>  
 <url-pattern>/assignSubjectToClass</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignSubject</display-name>  
 <servlet-name>assignSubjectServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignSubjectToClassServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignSubjectServlet</servlet-name>  
 <url-pattern>/assignSubject</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignSubjectToTeacher</display-name>  
 <servlet-name>assignSubjectToTeacherServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignSubjectToTeacherServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignSubjectToTeacherServlet</servlet-name>  
 <url-pattern>/assignSubjectToTeacher</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>assignSubjectTeacher</display-name>  
 <servlet-name>assignSubjectTeacherServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.AssignSubjectToTeacherServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>assignSubjectTeacherServlet</servlet-name>  
 <url-pattern>/assignSubjectTeacher</url-pattern>  
 </servlet-mapping>  
  
 <servlet>  
 <description></description>  
 <display-name>viewClassReport</display-name>  
 <servlet-name>viewClassReportServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.ClassReportServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>viewClassReportServlet</servlet-name>  
 <url-pattern>/viewClassReport</url-pattern>  
 </servlet-mapping>  
  
  
 <servlet>  
 <description></description>  
 <display-name>viewReport</display-name>  
 <servlet-name>viewReportServlet</servlet-name>  
 <servlet-class>com.example.Phase2.servlet.ClassReportServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>viewReportServlet</servlet-name>  
 <url-pattern>/viewReport</url-pattern>  
 </servlet-mapping>  
  
  
</web-app>

<?xml version="1.0" encoding="UTF-8"?>  
<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/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>Phase2</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <name>Phase2</name>  
 <packaging>war</packaging>  
  
 <properties>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <junit.version>5.7.0</junit.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.4.25.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate.validator</groupId>  
 <artifactId>hibernate-validator</artifactId>  
 <version>6.1.6.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.16</version>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <version>1.18.26</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.3.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>