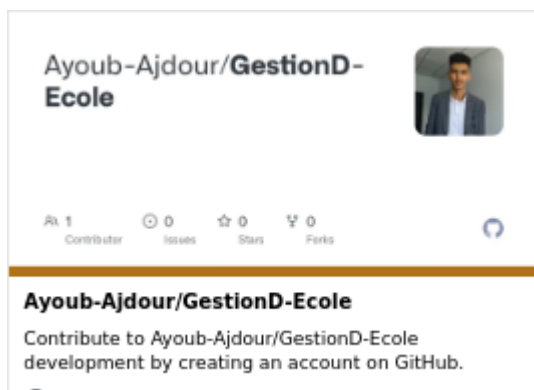




Gestion d'école

Pour poursuivre le développement de notre projet de gestion d'école, nous envisageons d'intégrer les frameworks JSF (JavaServer Faces) et Spring. Ces choix sont motivés par la nécessité d'apporter une évolutivité et une efficacité accrues à notre application. La partie du projet dédiée aux filières bénéficiera de l'utilisation de JSF, permettant ainsi des opérations CRUD fluides pour la gestion des programmes d'études. De plus, la gestion des étudiants sera optimisée grâce à l'intégration du framework Spring, qui facilitera l'ajout, la mise à jour et la suppression des profils d'étudiants. L'ensemble de ces fonctionnalités sera étroitement intégré au code source existant, disponible sur mon compte GitHub, offrant ainsi une vision complète des nouvelles structures, configurations Hibernate, JSF et Spring, ainsi que des améliorations apportées à la gestion des filières et des étudiants."



<https://github.com/Ayoub-Ajdour/GestionD-Ecole>

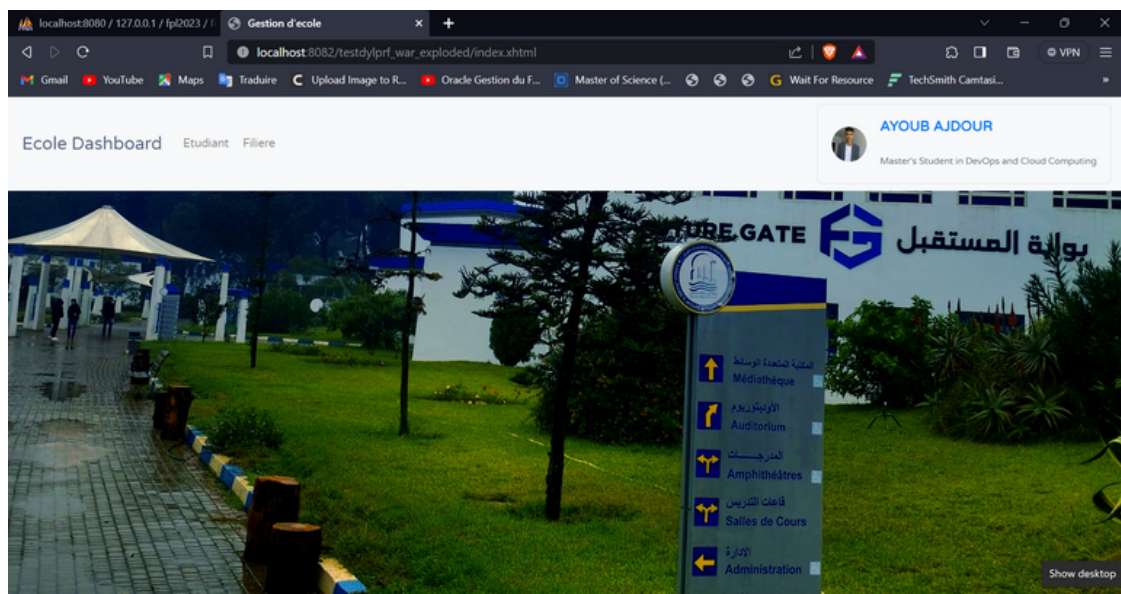


Figure 1: L'interface home

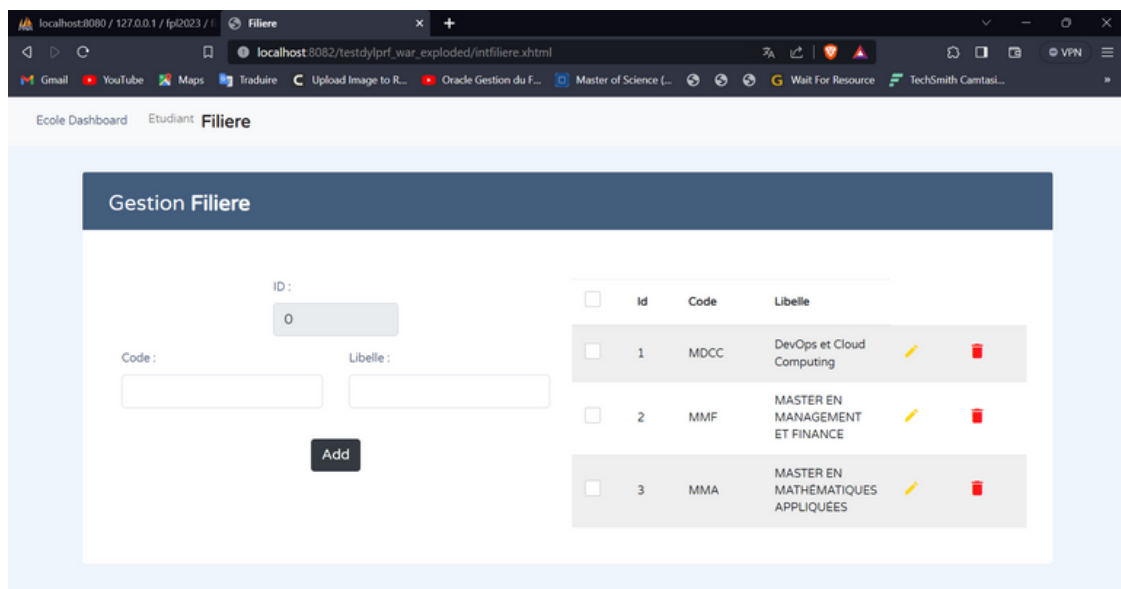


Figure 2: L'interface de gestion de Filiere

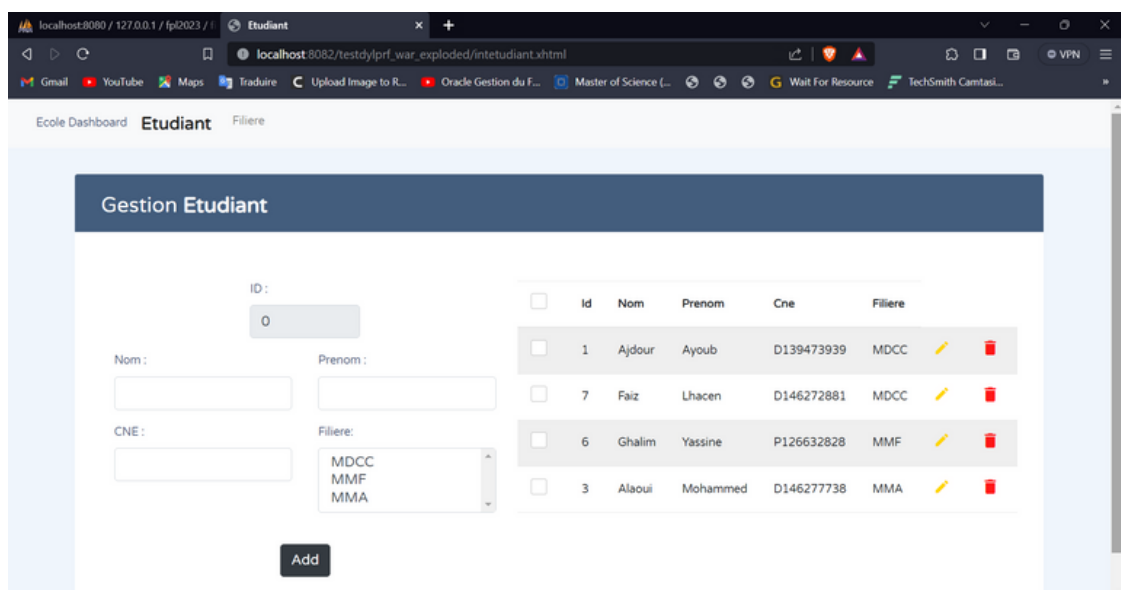


Figure 3: L'interface de gestion d'Etudiant

Class Filiere

```
package com.entities;

import javax.persistence.*;

import java.util.Collection;
import java.util.HashSet;
import java.util.Objects;
import java.util.Set;

@Entity
@Table(name = "filiere", catalog = "fp12023")
public class Filiere {

    @Id
    @Column(name = "idFiliere")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int idFiliere;
    @Basic
    @Column(name = "Code", nullable = false, length = 50)
    private String code;
    @Basic
    @Column(name = "libelle", nullable = false, length = 50)
    private String libelle;
    @OneToMany(mappedBy = "filiere")
    private Set<Etudiant> etudiantsByIdFiliere = new HashSet<Etudiant>(0);

    public int getIdFiliere() {
        return idFiliere;
    }

    public void setIdFiliere(int idFiliere) {
        this.idFiliere = idFiliere;
    }

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getLibelle() {
        return libelle;
    }

    public void setLibelle(String libelle) {
        this.libelle = libelle;
    }

    public Set<Etudiant> getEtudiantsByIdFiliere() {
        return etudiantsByIdFiliere;
    }
}
```

```

    }

    public void setEtudiantsByIdFiliere(Set<Etudiant> etudiantsByIdFiliere) {
        this.etudiantsByIdFiliere = etudiantsByIdFiliere;
    }

    @Override
    public int hashCode() {
        return Objects.hash(idFiliere, code, libelle, etudiantsByIdFiliere);
    }
}

```

Class Etudiant

```

package com.entities;

import javax.persistence.*;
import java.util.Objects;

@Entity
@Table(name = "etudiant", catalog = "fp12023")
public class Etudiant {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    @Column(name = "idEtudiant", nullable = false)
    private int idEtudiant;
    @Basic
    @Column(name = "nom", nullable = false, length = 50)
    private String nom;
    @Basic
    @Column(name = "prenom", nullable = false, length = 50)
    private String prenom;
    @Basic
    @Column(name = "cne", nullable = false, length = 50)
    private String cne;
    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "filiere", referencedColumnName = "idFiliere",
        nullable = false)
    private Filiere filiere;

    public int getIdEtudiant() {
        return idEtudiant;
    }

    public void setIdEtudiant(int idEtudiant) {
        this.idEtudiant = idEtudiant;
    }

    public String getNom() {
        return nom;
    }
}

```

```
public void setNom(String nom) {
    this.nom = nom;
}

public String getPrenom() {
    return prenom;
}

public void setPrenom(String prenom) {
    this.prenom = prenom;
}

public String getCne() {
    return cne;
}

public void setCne(String cne) {
    this.cne = cne;
}

public Filiere getFiliere() {
    return filiere;
}

public void setFiliere(Filiere filiere) {
    this.filiere = filiere;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Etudiant etudiant = (Etudiant) o;
    return idEtudiant == etudiant.idEtudiant;
}

@Override
public int hashCode() {
    return Objects.hash(idEtudiant);
}
}
```

Class FiliereBean

```
package com.beans;

import java.util.List;

import javax.faces.bean.ManagedBean;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import com.entities.Filiere;
import com.service.FiliereService;

@Component
@ManagedBean
@Scope("session")
public class FiliereBean {

    @Autowired
    private FiliereService filiereService;

    private Filiere filiere;
    private boolean modifyContext;

    public FiliereBean() {

        this.filiere = new Filiere();
        this.modifyContext = false;
    }

    public Filiere getFiliere() {
        return filiere;
    }

    public void setFiliere(Filiere filiere) {
        this.filiere = filiere;
    }

    public boolean isModifyContext() {
        return modifyContext;
    }

    public void setModifyContext(boolean modifyContext) {
        this.modifyContext = modifyContext;
    }

    public List<Filiere> getFiliere() {

        return this.filiereService.getAll();
    }
}
```

```

    public String getLabel() {
        return isModifyContext() ? "Edit" : "Add";
    }

    @Transactional
    public void save() {

        this.filiereService.saveOrUpdate(this.filiere);
        this.filiere = new Filiere();
        this.setModifyContext(false);
    }

    public void modifyy(Filiere f) {

        this.filiere = f;
        this.setModifyContext(true);
    }

    @Transactional
    public void delete(Filiere f) {

        this.filiereService.delete(f);
        this.filiere = new Filiere();
    }
}

```

Class EtudiantBean

```

package com.beans;

import com.entities.Etudiant;
import com.entities.Filiere;
import com.service.EtudiantService;
import com.service.FiliereService;
import com.service.impl.EtudiantServiceImpl;
import com.service.impl.FiliereServiceImpl;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import javax.faces.bean.ManagedBean;
import java.util.List;

@Component
@ManagedBean
@Scope("session")
public class EtudiantBean {
    @Autowired
    private EtudiantService etudiantService;

    private Etudiant etudiant;
    private boolean modifyContext;
    private int filiere;
}

```

```

@Autowired
private FiliereService filiereService;

public int getFiliere() {
    return filiere;
}

public void setFiliere(int filiere) {
    this.filiere = filiere;
}

public EtudiantBean() {

    this.etudiant = new Etudiant();
    this.modifyContext = false;
    this.etudiantService = new EtudiantServiceImpl();
    this.filiereService = new FiliereServiceImpl();
}

public Etudiant getEtudiant() {
    return etudiant;
}

public void setEtudiant(Etudiant etudiant) {
    this.etudiant = etudiant;
}

public boolean isModifyContext() {
    return modifyContext;
}

public void setModifyContext(boolean modifyContext) {
    this.modifyContext = modifyContext;
}

public List<Etudiant> getEtudiants() {

    return this.etudiantService.getAll();
}

public String getLabel() {
    return isModifyContext() ? "Edit" : "Add";
}

//@Transactional
public void save() {
    this.etudiant.setFiliere(filiereService.getById(this.filiere));
    this.etudiantService.saveOrUpdate(this.etudiant);
    this.etudiant = new Etudiant();
    this.setModifyContext(false);
}

public void modifyy(Etudiant f) {

```



```

        this.etudiant = f;
        this.setModifyContext(true);
    }

    //@Transactional
    public void delete(Etudiant f) {
        this.etudiantService.delete(f);
        this.etudiant = new Etudiant();
    }
}

```

Class FiliereDAOImpl

```

package com.dao.impl;

import java.util.List;

import org.hibernate.SessionFactory;
import org.hibernate.criterion.DetachedCriteria;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.support.HibernateDaoSupport;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import com.dao.FiliereDAO;
import com.entities.Filiere;

@Repository
public class FiliereDaoImpl extends HibernateDaoSupport implements FiliereDAO
{

    @Autowired
    public void setUpSessionFactory(SessionFactory sf) {
        this.setSessionFactory(sf);
    }

    @Override
    @Transactional
    public void saveOrUpdate(Filiere f) {

        this.getHibernateTemplate().saveOrUpdate(f);
    }

    @Override
    @Transactional
    public void delete(Filiere f) {

        this.getHibernateTemplate().delete(f);
    }

    @Override
    public Filiere getById(int id) {

```

```

        return this.getHibernateTemplate().get(Filiere.class, id);
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<Filiere> getAll() {

        return (List<Filiere>) this.getHibernateTemplate().findByCriteria(
            DetachedCriteria.forClass(Filiere.class));
    }
}

```

Class EtudiantDAOImpl

```

package com.dao.impl;

import com.dao.EtudiantDAO;
import com.entities.Etudiant;
import com.entities.Filiere;
import org.hibernate.SessionFactory;
import org.hibernate.criterion.DetachedCriteria;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.support.HibernateDaoSupport;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Repository
public class EtudiantDaoImpl extends HibernateDaoSupport implements
EtudiantDAO {
    @Autowired
    public void setUpSessionFactory(SessionFactory sf) {
        this.setSessionFactory(sf);
    }

    @Override
    @Transactional
    public void saveOrUpdate(Etudiant f) {
        this.getHibernateTemplate().saveOrUpdate(f);
    }

    @Override
    @Transactional
    public void delete(Etudiant f) {
        this.getHibernateTemplate().delete(f);
    }

    @Override
    public Etudiant getById(int id) {

        return this.getHibernateTemplate().get(Etudiant.class, id);
    }

    @Override

```

```

@SuppressWarnings("unchecked")
public List<Etudiant> getAll() {
    return (List<Etudiant>) this.getHibernateTemplate().findByCriteria(
        DetachedCriteria.forClass(Etudiant.class));
}
}

```

Class FiliereServiceImpl

```

package com.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import com.dao.FiliereDAO;
import com.entities.Filiere;
import com.service.FiliereService;

@Component
public class FiliereServiceImpl implements FiliereService {

    @Autowired
    private FiliereDAO filiereDao ;

    @Override
    @Transactional
    public void saveOrUpdate(Filiere f) {
        this.filiereDao.saveOrUpdate(f);
    }

    @Override
    @Transactional
    public void delete(Filiere f) {
        this.filiereDao.delete(f);
    }

    @Override
    public Filiere getById(int id) {
        return this.filiereDao.getById(id);
    }

    @Transactional
    public List<Filiere> getAll() {
        List<Filiere> filieres = this.filiereDao.getAll();
        return filieres;
    }
}

```

Class EtudiantService

```
package com.service.impl;

import com.dao.EtudiantDAO;
import com.dao.FiliereDAO;
import com.entities.Etudiant;
import com.entities.Filiere;
import com.service.EtudiantService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Component
public class EtudiantServiceImpl implements EtudiantService {

    @Autowired
    private EtudiantDAO etudiantDAO ;
    @Override
    @Transactional
    public void saveOrUpdate(Etudiant f) {
        this.etudiantDAO.saveOrUpdate(f);
    }

    @Override
    @Transactional
    public void delete(Etudiant f) {
        this.etudiantDAO.delete(f);
    }

    @Override
    public Etudiant getById(int id) {
        return this.etudiantDAO.getById(id);
    }

    @Override
    @Transactional
    public List<Etudiant> getAll() {
        List<Etudiant> etudiants = this.etudiantDAO.getAll();

        //      System.out.println("test ==> "+
        filieres.get(0).getEtudiants().size());

        return etudiants;
    }
}
```

filiere.xhtml

```
<div class="container-xl" >
  <div class="table-responsive">
    <div class="table-wrapper">
      <div class="table-title">
        <div class="row">
          <div class="col-sm-6">
            <h2>Gestion <b>Filiere</b></h2>
          </div>
        </div>
      </div>
    </div>
    <div class="container-fluid">
      <div id="alert" class="alert font-weight-bold text-center"></div>
      <div class="row">
        <div class="section">
          <h:form>
            <div class="col-sm-12 col-md-6 col-lg-4 mx-auto">
              <div class="form-group">
                <label>ID :</label>
                <h:inputText
value="#{filiereBean.filiere.idFiliere}"
                class="form-control"
                readonly="true" />
              </div>
            </div>
            <div class="container">
              <div class="row">
                <div class="col-sm-12 col-md-6 mx-auto">
                  <div class="form-group">
                    <label>Code :</label>
                    <h:inputText
value="#{filiereBean.filiere.code}"
                    class="form-control "
                    required="true" />
                  </div>
                </div>
                <div class="col-sm-12 col-md-6 mx-auto">
                  <div class="form-group">
                    <label>Libelle :</label>
                    <h:inputText
value="#{filiereBean.filiere.libelle}"
                    class="form-control "
                    required="true" />
                  </div>
                </div>
              </div>
            </div>
            <div class="form-group text-center">
              <br></div>
            </div>
          </h:form>
        </div>
      </div>
    </div>
  </div>

```

[illegible]

```

title="Delete" style="color: #ff0000;">&#xE872;</i>
        </h:commandLink>

    </td>
</tr>
</ui:repeat>
</tbody>
</table>
</h:form>
</div>
</div>
</div>
</div>
</div>
</div>

```

etudiant.xhtml

```

<div class="container-xl" >
    <div class="table-responsive">
        <div class="table-wrapper">
            <div class="table-title">
                <div class="row">
                    <div class="col-sm-6">
                        <h2>Gestion <b>Etudiant</b></h2>
                    </div>
                </div>
            </div>
            <div class="container-fluid">
                <div id="alert" class="alert font-weight-bold text-
center"></div>
                <div class="row">
                    <div class="section">
                        <h:form>
                            <div class="col-sm-12 col-md-6 col-lg-4 mx-
auto">
                                <div class="form-group">
                                    <label>ID :</label>
                                    <h:inputText
value="#{etudiantBean.etudiant.idEtudiant}"
                                class="form-control"
                                readonly="true" />
                                </div>
                            </div>
                            <div class="container">
                                <div class="row">
                                    <div class="col-sm-12 col-md-6 mx-
auto">
                                        <div class="form-group">
                                            <label>Nom :</label>
                                            <h:inputText
value="#{etudiantBean.etudiant.nom}"

```

```

class="form-control "
required="true" />
    </div>
</div>
<div class="col-sm-12 col-md-6 mx-
auto">
    <div class="form-group">
        <label>Prenom :</label>
        <h:inputText
value="#{etudiantBean.etudiant.prenom}"
class="form-control "
required="true" />
    </div>
</div>
<div class="col-sm-12 col-md-6 mx-
auto">
    <div class="form-group">
        <label>CNE :</label>
        <h:inputText
value="#{etudiantBean.etudiant.cne}"
class="form-control "
required="true" />
    </div>
</div>
<div class="col-sm-12 col-md-6 mx-
auto">
    <div class="form-group">
        <h:outputLabel
value="Filiere:" />
        <h:selectOneListbox
value="#{etudiantBean.filiere}" class="form-control">
<!--
            <f:selectItems
value="#{filiereBean.filiere}" var="f" itemLabel="#{f.code}"
itemValue="#{f.idFiliere}" class="form-control"/>
        </h:selectOneListbox>
    </div>
</div>
</div>
<div class="form-group text-center">
    <br></br>
    <h:commandButton class="btn btn-dark"
action="#{etudiantBean.save()}"
value="${etudiantBean.label}" />
</div>

```



```

        </h:form>
    </div>
    <div class="section">
        <h:form>
            <table class="table table-striped table-
hover" id="eeee">
                <thead>
                    <tr>
                        <th>
                            <span class="custom-checkbox">
                                <input type="checkbox"
id="selectAll"></input>
                            <label for="selectAll"></label>
                        </span>
                        </th>
                        <th>Id</th>
                        <th>Nom</th>
                        <th>Prenom</th>
                        <th>Cne</th>
                        <th>Filiere</th>
                    </tr>
                </thead>
                <tbody>
                    <ui:repeat
value="\${etudiantBean.etudiants}" var="e">
                        <tr>
                            <td>
                                <span class="custom-checkbox">
                                    <input type="checkbox" id="checkboxx1"
name="options[]" value="1"></input>
                                <label for="checkboxx1"></label>
                            </span>
                            </td>
                            <td>
                                \${e.idEtudiant}
                            </td>
                            <td>
                                \${e.nom}
                            </td>
                            <td>
                                \${e.prenom}
                            </td>
                            <td>
                                \${e.cne}
                            </td>
                            <td>
                                \${e.filiere.code}
                            </td>
                            <td>
                                <h:commandLink
action="\#{etudiantBean.modifyy(e)}"><i class="material-icons" title="Edit"
style="color: #FFD700;">&#xE254;</i></h:commandLink>
                            </td>
                            <td>
                                <h:commandLink
action="\#{etudiantBean.delete(e)}">

```

```

<i class="material-icons"
title="Delete" style="color: #ff0000;">&#xE872;</i>
</h:commandLink>
</td>
</tr>
</ui:repeat>
</tbody>
</table>
</h:form>
</div>
</div>
</div>
</div>
</div>

```