



**ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR**  
*Membre de*   
**HONORIS UNITED UNIVERSITIES**

---

# Rapport de Mini-projet Création d'un nouveau module (Gestion de projets étudiants)

---

Élaboré par :  
KHAIRY Sami (G3)

Professeur :  
Pr AIT DAOUD Mohammed

Lien Github :  
<https://github.com/Sami-khairv/Student-Project-Management---Odoo-module>

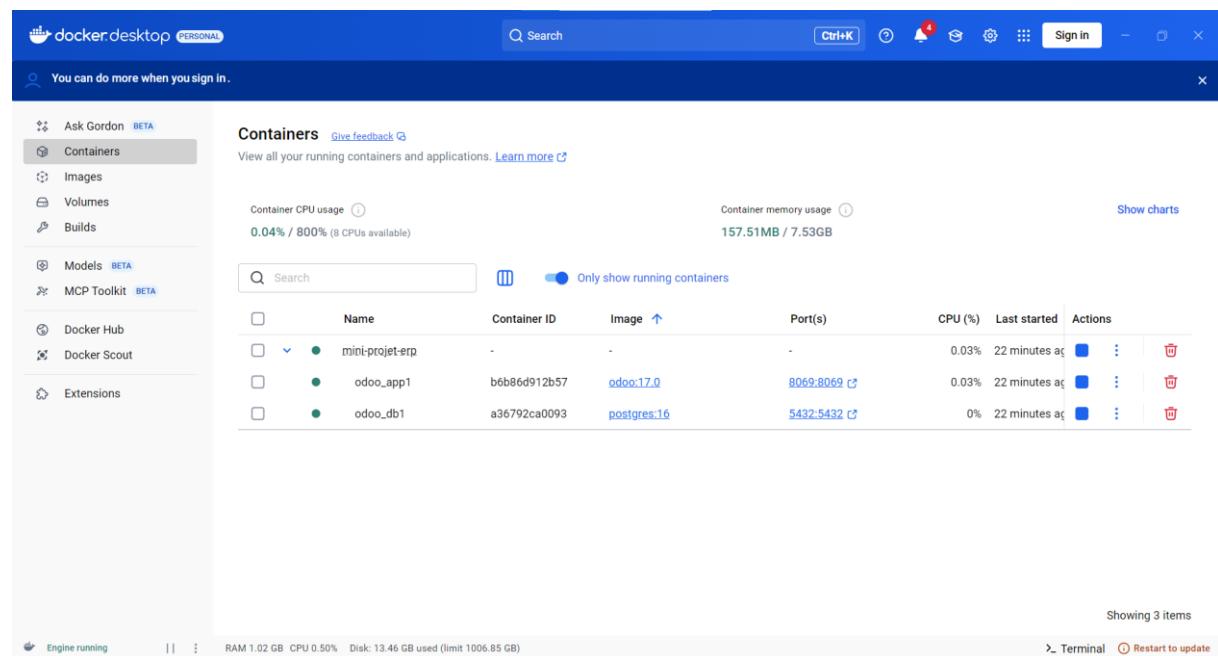
## Objectif du Module

Créer un module Odoo technique nommé `gestion_projets_etudiants` visant à digitaliser l'intégralité du processus de suivi académique. Ce module permet de :

- **Gérer le cycle de vie complet** des projets (PFE, PFA, Stages) à travers un workflow strict et validé : *Brouillon -> Validation -> En cours -> Rendu -> Noté.*
- **Suivre l'avancement technique** via un système de "**Livrables & Jalons**", intégrant une barre de progression calculée automatiquement et le dépôt des fichiers (rapports, code source).
- **Organiser les soutenances** en gérant la planification (Date, Lieu), la composition des membres du Jury, et l'envoi automatisé des **convocations par email**.
- **Appliquer une notation multicritères** précise (Rapport/10, Présentation/5, Technique/5) avec un calcul automatique et sécurisé de la note finale (/20).
- **Sécuriser l'accès aux données** grâce à trois niveaux de permissions : **Étudiant** (vue limitée aux siens), **Tuteur** (supervision et notation), et **Responsable Pédagogique** (administration).
- **Piloter l'activité** grâce à des **Tableaux de bord et Statistiques** (Vues Graphiques et Pivot) permettant d'analyser la répartition des notes et la charge des tuteurs, tout en conservant les vues Kanban et Liste pour la gestion quotidienne.

## Étape 1 : Préparation de l'environnement Docker

Nous allons utiliser un fichier `docker-compose.yml` configuré pour Odoo 17 avec une base PostgreSQL 16.



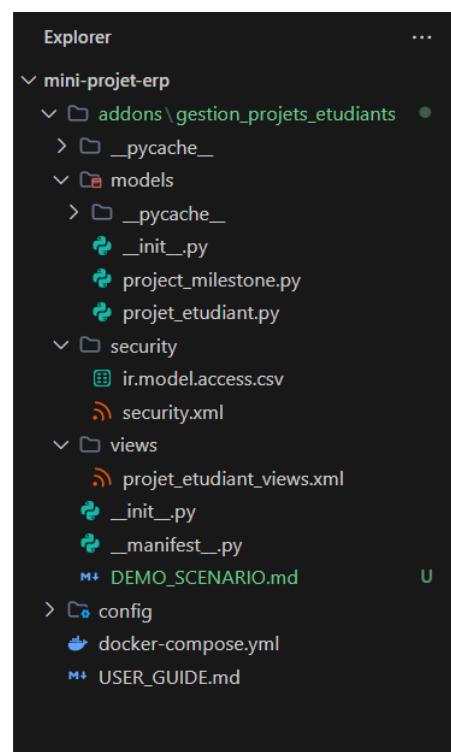
```

  docker-compose.yml X
  docker-compose.yml
1   version: "3.8"
2
3   services:
4     db:
5       image: postgres:16
6       container_name: odoo_db1
7       ports:
8         - "5432:5432"
9       environment:
10      POSTGRES_USER: odoo
11      POSTGRES_PASSWORD: odoo
12      PGDATA: /var/lib/postgresql/data/pgdata
13       volumes:
14         - odoo-db-data1:/var/lib/postgresql/data
15
16     odoo17:
17       image: odoo:17.0
18       container_name: odoo_app1
19       depends_on:
20         - db
21       ports:
22         - "8069:8069"
23       volumes:
24         - odoo-web-data1:/var/lib/odoo
25         - ./config:/etc/odoo
26         - ./addons:/mnt/extras-addons
27       command: >
28         odoo -d odoo_db1
29         -u gestion_projets_etudiants
30         --dev=xml
31         --db_user=odoo
32         --db_password=odoo
33         --db_host=db
34
35   volumes:
36     odoo-web-data1:
37     odoo-db-data1:

```

## Etape 2 : Création de la structure du module

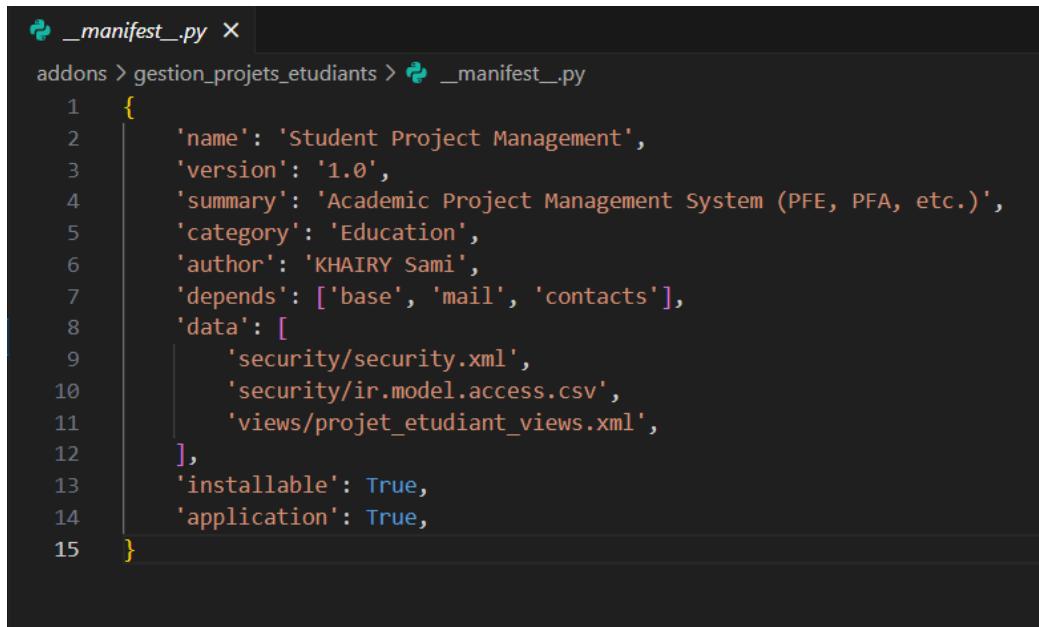
Dans le dossier addons créez l'arborescence suivante:



## Étape 3 : Définition du module (Manifeste)

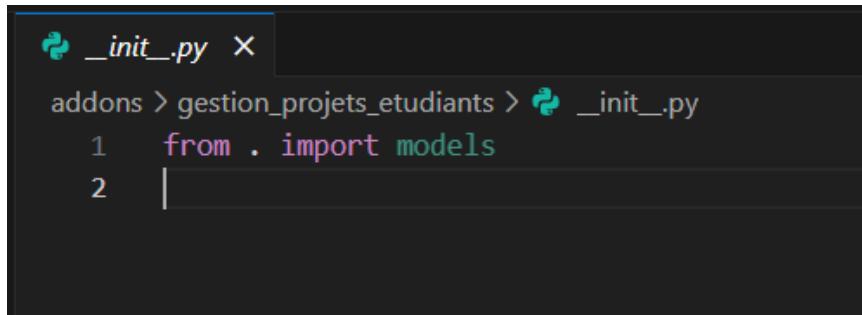
Le manifeste décrit le module et ses dépendances (mail, contacts).

**Fichier :** addons/gestion\_projets\_etudiants/\_\_manifest\_\_.py



```
__manifest__.py ×
addons > gestion_projets_etudiants > __manifest__.py
1  {
2      'name': 'Student Project Management',
3      'version': '1.0',
4      'summary': 'Academic Project Management System (PFE, PFA, etc.)',
5      'category': 'Education',
6      'author': 'KHAIRY Sami',
7      'depends': ['base', 'mail', 'contacts'],
8      'data': [
9          'security/security.xml',
10         'security/ir.model.access.csv',
11         'views/projet_etudiant_views.xml',
12     ],
13     'installable': True,
14     'application': True,
15 }
```

**Fichier :** addons/gestion\_projets\_etudiants/\_\_init\_\_.py



```
__init__.py ×
addons > gestion_projets_etudiants > __init__.py
1  from . import models
2  |
```

## Etape 4 : Le Modèle Python (Back-end)

Nous allons créer la logique métier avec les contraintes et l'héritage du système de messagerie (mail.thread).

### 1. Init du dossier models :

**Fichier :** addons/gestion\_projets\_etudiants/models/\_\_init\_\_.py

```

addons > gestion_projets_etudiants > models > __init__.py
1  from . import projet_etudiant
2  from . import project_milestone

```

**Fichier :** addons/gestion\_projets\_etudiants/models/projet\_etudiant.py

Ce fichier définit le modèle principal **StudentProject**, qui constitue le cœur de la base de données du module. Il structure l'enregistrement des projets en définissant les champs essentiels (titre, dates, liens Git) et gère les relations entre les différents acteurs (étudiants, tuteurs et membres du jury), tout en pilotant le cycle de vie du projet à travers ses différents états (brouillon, en cours, rendu, noté).

Au-delà du stockage, ce fichier contient toute la **logique métier et l'automatisation**. Il intègre des fonctions Python pour calculer dynamiquement la note finale et la progression en temps réel, vérifier la cohérence des saisies (contraintes de notes) et automatiser l'envoi des emails de convocation pour les soutenances.

```

addons > gestion_projets_etudiants > models > projet_etudiant.py > StudentProject > student_ids
1  from odoo import models, fields, api
2
3  class StudentProject(models.Model):
4      _name = "student.project"
5      _description = "Student Project Management"
6      _inherit = ['mail.thread', 'mail.activity.mixin']
7
8      name = fields.Char(string="Titre du projet", required=True, tracking=True)
9      description = fields.Html(string="Description du sujet")
10
11     project_type = fields.Selection([
12         ('pfe', 'PFE'),
13         ('pfa', 'PFA'),
14         ('mini_projet', 'Mini-projet'),
15         ('stage', 'Stage')
16     ], string="Type de Projet", default='mini_projet', tracking=True)
17
18     # Relation avec les étudiants (res.partner)
19     student_ids = fields.Many2many(
20         'res.partner',
21         string="Étudiant(s)",
22         help="Sélectionnez les étudiants assignés à ce projet."
23     )
24
25     # Relation avec le tuteur (Supervisor)
26     supervisor_id = fields.Many2one(
27         'res.partner',
28         string="Tuteur (Professeur)",
29         tracking=True
30     )
31
32     date_deadline = fields.Date(string="Date limite", tracking=True)
33     repository_url = fields.Char(string="Lien Repository (Git)")
34

```

```
projet_etudiant.py X
addons > gestion_projets_etudiants > models > projet_etudiant.py > StudentProject > student_ids
3   class StudentProject(models.Model):
4+
5       # Champs pour la Soutenance
6       defense_date = fields.Datetime(string="Date de Soutenance", tracking=True)
7       defense_location = fields.Char(string="Salle / Lieu")
8       jury_ids = fields.Many2many(
9           'res.partner',
10          'student_project_jury_rel',
11          'project_id',
12          'partner_id',
13          string="Membres du Jury",
14          help="Sélectionnez les membres du jury (en plus du tuteur)."
15      )
16
17      # Jalons & Livrables
18      milestone_ids = fields.One2many(
19          'student.project.milestone',
20          'project_id',
21          string="Livrables & Jalons"
22      )
23      progress = fields.Float(string="Progression", compute="_compute_progress", store=True)
24
25      # Évaluation Détailée
26      grade_report = fields.Float(string="Note Rapport (/10)", default=0.0, tracking=True)
27      grade_oral = fields.Float(string="Note Présentation (/5)", default=0.0, tracking=True)
28      grade_technical = fields.Float(string="Qualité Technique (/5)", default=0.0, tracking=True)
29
30      grade = fields.Float(string="Note Finale (/20)", compute="_compute_total_grade", store=True, tracking=True)
31
32      state = fields.Selection([
33          ('draft', 'Brouillon'),
34          ('validation', 'Validation'),
35          ('in_progress', 'En cours'),
36          ('submitted', 'Rendu'),
37          ('done', 'Noté')
38      ], string="État", default='draft', tracking=True, group_expand='_expand_states')
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
```

```
projet_etudiant.py X
addons > gestion_projets_etudiants > models > projet_etudiant.py > StudentProject > student_ids
3   class StudentProject(models.Model):
4+
5       def _expand_states(self, states, domain, order):
6           return [key for key, val in type(self).state.selection]
7
8       # Workflow Actions
9       def action_draft(self):
10          self.state = 'draft'
11
12          def action_validate(self):
13              self.state = 'validation'
14
15          def action_in_progress(self):
16              self.state = 'in_progress'
17
18          def action_submit(self):
19              self.state = 'submitted'
20
21          def action_done(self):
22              self.state = 'done'
23
24          def action_send_defense_invitation(self):
25              """ Envoie une convocation via le Chatter à tous les participants """
26              for record in self:
27                  if not record.defense_date:
28                      raise models.ValidationError("Veuillez définir une date de soutenance avant d'envoyer les convocations.")
29
30                  partners = record.student_ids | record.jury_ids | record.supervisor_id
31                  if not partners:
32                      raise models.ValidationError("Aucun participant (étudiant, tuteur ou jury) n'est assigné.")
33
34                  subject = f"Convocation Soutenance : {record.name}"
35                  body = f"You êtes invité à la soutenance du projet du {record.defense_date}. Lieu : {record.defense_location or 'Non spé"
36
37                  record.message_post(body=body, subject=subject, partner_ids=partners.ids)
```

```

projet_etudiant.py ×
addons > gestion_projets_etudiants > models > projet_etudiant.py > StudentProject > _check_sub_grades
  3   class StudentProject(models.Model):
104     @api.depends('milestone_ids', 'milestone_ids.is_done')
105     def _compute_progress(self):
106         for record in self:
107             if not record.milestone_ids:
108                 record.progress = 0.0
109             else:
110                 total = len(record.milestone_ids)
111                 done = len(record.milestone_ids.filtered(lambda m: m.is_done))
112                 record.progress = (done / total) * 100
113
114     @api.depends('grade_report', 'grade_oral', 'grade_technical')
115     def _compute_total_grade(self):
116         for record in self:
117             record.grade = record.grade_report + record.grade_oral + record.grade_technical
118
119     @api.constraints('grade_report', 'grade_oral', 'grade_technical')
120     def _check_sub_grades(self):
121         for record in self:
122             if not (0 <= record.grade_report <= 10):
123                 raise models.ValidationError("La note du rapport doit être entre 0 et 10.")
124             if not (0 <= record.grade_oral <= 5):
125                 raise models.ValidationError("La note orale doit être entre 0 et 5.")
126             if not (0 <= record.grade_technical <= 5):
127                 raise models.ValidationError("La note technique doit être entre 0 et 5.")
128

```

**Fichier :** addons/gestion\_projets\_etudiants/models/project\_milestone.py

Ce fichier contient la définition du modèle **ProjectMilestone**, qui représente les sous-tâches ou les livrables obligatoires liés à un projet principal. Il agit comme une table "enfant" dans la base de données, directement reliée au modèle principal student.project via une relation Many2one.

Ce fichier gère la granularité du suivi de projet en permettant de définir des étapes clés (jalons) avec leurs propres échéances et descriptions. Il implémente une sécurité de données via l'instruction `ondelete='cascade'`, assurant que si un projet est supprimé, tous ses livrables associés le sont aussi.

De plus, ce modèle intègre les fonctionnalités de soumission de travail. Grâce aux champs `is_done` (booléen) et `file_attachment` (binaire), il permet aux étudiants de marquer une étape comme terminée et de téléverser directement leurs rapports ou codes sources, ce qui alimente ensuite le calcul automatique de la progression globale du projet.

```

project_milestone.py ×
addons > gestion_projets_etudiants > models > project_milestone.py > ...
1  from odoo import models, fields, api
2
3  class ProjectMilestone(models.Model):
4      _name = "student.project.milestone"
5      _description = "Jalon / Livrable du Projet"
6      _order = "deadline, id"
7
8      project_id = fields.Many2one('student.project', string="Projet", required=True, ondelete='cascade')
9      name = fields.Char(string="Titre du Livrable", required=True)
10     description = fields.Text(string="Description")
11     deadline = fields.Date(string="Date Limite")
12     is_done = fields.Boolean(string="Terminé")
13     file_attachment = fields.Binary(string="Fichier Joint")
14     file_name = fields.Char(string="Nom du Fichier")
15

```

## Etape 5 : Sécurité et Droits d'accès

Cette étape est cruciale pour définir **qui** peut voir **quoi**.

### 1. Droits d'accès CRUD (CSV) :

**Fichier :** addons/gestion\_projets\_etudiants/security/ir.model.access.csv

Ce fichier **ir.model.access.csv** constitue la couche de sécurité essentielle du module (ACL - Access Control Lists). Il fait l'interface entre les modèles de données définis en Python et les groupes d'utilisateurs, dictant précisément les opérations CRUD (Create, Read, Update, Delete) autorisées pour chaque rôle au sein de l'application.

Le fichier configure une hiérarchie de permissions stricte pour protéger l'intégrité des données. Il accorde aux **Étudiants** des droits de travail (lecture, écriture, création) tout en leur interdisant formellement la suppression de projets ou de jalons (le 0 dans la colonne `perm_unlink`), évitant ainsi les pertes accidentnelles. À l'inverse, les **Tuteurs et Managers** se voient attribuer des droits d'administration complets (1,1,1,1), leur permettant de gérer le cycle de vie des données sans restriction.

```

ir.model.access.csv ×
addons > gestion_projets_etudiants > security > ir.model.access.csv
1  id,name,model_id:id,group_id:id,perm_create,perm_write,perm_unlink
2  access_student_project_student,student.project.student,model_student_project,group_student_project_student,1,1,1,0
3  access_student_project_tutor,student.project.tutor,model_student_project,group_student_project_tutor,1,1,1,1
4  access_student_project_manager,student.project.manager,model_student_project,group_student_project_manager,1,1,1,1
5  access_student_project_milestone_student,student.project.milestone.student,model_student_project_milestone,group_student_project_student,1,1,1,0
6  access_student_project_milestone_tutor,student.project.milestone.tutor,model_student_project_milestone,group_student_project_tutor,1,1,1,1
7  access_student_project_milestone_manager,student.project.milestone.manager,model_student_project_milestone,group_student_project_manager,1,1,1,1

```

### 2. Définition des groupes et règles :

**Fichier :** addons/gestion\_projets\_etudiants/security/security.xml

Ce fichier XML configure l'architecture des **Rôles et Groupes** utilisateurs au sein de l'interface Odoo. Il définit formellement les trois profils d'acteurs du module (Étudiant, Tuteur, Responsable Pédagogique) et établit une hiérarchie de droits par héritage (`implied_ids`), garantissant que chaque niveau supérieur possède automatiquement toutes les prérogatives du niveau inférieur.

Au-delà de la simple définition des groupes, ce script implémente les **Règles d'Enregistrement (Record Rules)**, qui assurent la confidentialité des données ligne par ligne. Il applique un filtre dynamique (`domain_force`) qui restreint la vue des étudiants uniquement aux projets où ils sont assignés, tout en accordant aux tuteurs une visibilité globale (`[(1, '=', 1)]`) sur l'ensemble des projets pour permettre leur supervision.

```
security.xml x
addons > gestion_projets_etudiants > security > security.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <odoo>
3      <data>
4          <!-- CATEGORY (Module Category in Users form) -->
5          <record model="ir.module.category" id="module_category_student_project">
6              <field name="name">Gestion des Projets</field>
7              <field name="description">Niveaux d'accès pour les projets étudiants.</field>
8              <field name="sequence" value="10"></field>
9          </record>
10
11         <!-- GROUP: STUDENT -->
12         <record id="group_student_project_student" model="res.groups">
13             <field name="name">Étudiant</field>
14             <field name="category_id" ref="module_category_student_project"/>
15             <field name="implied_ids" eval="[(4, ref('base.group_user'))]"/>
16             <field name="comment">Peut voir uniquement ses propres projets.</field>
17         </record>
18
19         <!-- GROUP: TUTOR (TEACHER) -->
20         <record id="group_student_project_tutor" model="res.groups">
21             <field name="name">Tuteur / Professeur</field>
22             <field name="category_id" ref="module_category_student_project"/>
23             <field name="implied_ids" eval="[(4, ref('group_student_project_student'))]"/>
24             <field name="comment">Peut voir tous les projets et les valider.</field>
25         </record>
26
27         <!-- GROUP: MANAGER (ADMIN) -->
28         <record id="group_student_project_manager" model="res.groups">
29             <field name="name">Responsable Pédagogique</field>
30             <field name="category_id" ref="module_category_student_project"/>
31             <field name="implied_ids" eval="[(4, ref('group_student_project_tutor'))]"/>
32             <field name="users" eval="[(4, ref('base.user_root')), (4, ref('base.user_admin'))]"/>
33         </record>
34     </data>
</odoo>
```

```

security.xml ×
addons > gestion_projets_etudiants > security > security.xml
  2   <odoo>
  3     <data>
 34
 35       <!-- RECORD RULES (Row Level Security) -->
 36
 37       <!-- Student Rule: Can only see projects where they are assigned -->
 38       <record id="rule_student_project_student_own" model="ir.rule">
 39         <field name="name">Projets Étudiant: Seulement les siens</field>
 40         <field name="model_id" ref="model_student_project"/>
 41         <field name="groups" eval="[(4, ref('group_student_project_student'))]"/>
 42         <field name="domain_force">[('student_ids', 'in', user.partner_id.id)]</field>
 43         <field name="perm_read" eval="True"/>
 44         <field name="perm_write" eval="True"/>
 45         <field name="perm_create" eval="True"/>
 46         <field name="perm_unlink" eval="False"/> <!-- Students usually can't delete -->
 47     </record>
 48
 49       <!-- Tutor Rule: See ALL projects (or refine to supervised only if needed, currently ALL for visibility) -->
 50       <record id="rule_student_project_tutor_all" model="ir.rule">
 51         <field name="name">Projets Tuteur: Tous</field>
 52         <field name="model_id" ref="model_student_project"/>
 53         <field name="groups" eval="[(4, ref('group_student_project_tutor'))]"/>
 54         <field name="domain_force">[(1, '=', 1)]</field>
 55         <field name="perm_read" eval="True"/>
 56         <field name="perm_write" eval="True"/>
 57         <field name="perm_create" eval="True"/>
 58         <field name="perm_unlink" eval="True"/>
 59     </record>
 60
 61   </data>
 62 </odoo>
 63

```

## Etape 6 : Interface Utilisateur (Vues)

Nous définissons ici les vues Kanban, Liste, Formulaire et Recherche etc.

**Fichier :** addons/gestion\_projets\_etudiants/views/projet\_etudiant\_views.xml

Ce fichier XML est l'architecte de **l'interface utilisateur (UI)** du module. Il définit l'ensemble des vues interactives (Kanban, Liste, Formulaire, Calendrier) qui permettent aux utilisateurs de gérer les projets, ainsi que la structure de navigation (Menus et Actions) nécessaire pour accéder à ces écrans.

Le code configure une expérience riche en intégrant des **composants visuels dynamiques** tels que la barre d'état du workflow, des indicateurs de couleur pour les retards et une jauge pour la note finale. De plus, il inclut des vues décisionnelles (Graphique et Pivot) qui offrent aux responsables un tableau de bord statistique instantané sur la répartition et la notation des projets.

```
project_etudiant_views.xml
addons > gestion_projets_etudiants > views > project_etudiant_views.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ocean>
3
4  <!-- SEARCH VIEW -->
5  <record id="view_student_project_search" model="ir.ui.view">
6      <field name="name">student.project.search</field>
7      <field name="model">student.project</field>
8      <field name="arch" type="xml">
9          <search>
10             <field name="name"/>
11             <field name="student_ids"/>
12             <field name="supervisor_id"/>
13             <field name="project_type"/>
14             <filter string="Mes Projets" name="my_projects" domain="[(('student_ids', 'in', uid)]"/>
15             <filter string="PFE Uniquement" name="pfe_only" domain="[(('project_type', '=', 'pfe'))]"/>
16             <filter string="Projets non notés" name="not_graded" domain="[(('grade', '=', 0), ('state', '=', 'done'))]"/>
17             <group expand="1" string="Regrouper par">
18                 <filter string="Tuteur" name="groupby_supervisor" context="{'group_by': 'supervisor_id'}"/>
19                 <filter string="Etat" name="groupby_state" context="{'group_by': 'state'}"/>
20                 <filter string="Type" name="groupby_type" context="{'group_by': 'project_type'}"/>
21             </group>
22         </search>
23     </field>
24 </record>
```

```
project_etudiant_views.xml
addons > gestion_projets_etudiants > views > project_etudiant_views.xml
2  <ocean>
26  <!-- KANBAN VIEW -->
27  <record id="view_student_project_kanban" model="ir.ui.view">
28      <field name="name">student.project.kanban</field>
29      <field name="model">student.project</field>
30      <field name="arch" type="xml">
31          <kanban default_group_by="state" class="o Kanban_Small_Column" on_create="quick_create">
32              <field name="name"/>
33              <field name="student_ids"/>
34              <field name="grade"/>
35              <field name="project_type"/>
36              <field name="state"/>
37              <field name="id"/>
38              <field name="activity_ids"/>
39              <field name="activity_state"/>
40              <templates>
41                  <t t-name="kanban-box">
42                      <div t-attrib-class="oe_kanban_global_click oe_kanban_card">
43                          <div class="oe_kanban_details">
44                              <strong class="o_kanban_record_title">
45                                  <field name="name"/>
46                              </strong>
47                              <div>
48                                  <field name="student_ids" widget="many2many_tags" options="{'color_field': 'color'}"/>
49                              </div>
50                              <div class="o_kanban_record_bottom mt-2">
51                                  <div class="oe_kanban_bottom_left">
52                                      <field name="activity_ids" widget="kanban_activity"/>
53                                      <field name="project_type" widget="label_selection"/>
54                                  </div>
55                                  <div class="oe_kanban_bottom_right">
56                                      <field name="progress" widget="progressbar"/>
57                                      <span t-if="record.grade.raw_value > 0" class="badge rounded-pill text-bg-success">
58                                          <field name="grade"/>/20
59                                      </span>
60                                  </div>
61                              </div>
62                          </div>
63                      </div>
64                  </t>
65              </templates>
66          </kanban>
67      </field>
68  </record>
```

```

projet_etudiant_views.xml ×
addons > gestion_projets_etudiants > views > projet_etudiant_views.xml
2   <odoo>
3
4     <!-- TREE VIEW -->
5     <record id="view_student_project_tree" model="ir.ui.view">
6       <field name="name">student.project.tree</field>
7       <field name="model">student.project</field>
8       <field name="arch" type="xml">
9         <tree decoration-danger="date_deadline < current_date and state != 'done'">
10          <!-- decoration-success="state == 'done'" -->
11          <!-- decoration-info="state == 'in_progress'" -->
12          <field name="name"/>
13          <field name="student_ids" widget="many2many_tags"/>
14          <field name="supervisor_id"/>
15          <field name="project_type"/>
16          <field name="date_deadline"/>
17          <field name="progress" widget="progressbar" optional="show"/>
18          <field name="defense_date" optional="hide"/>
19          <field name="grade" optional="show"/>
20          <field name="state" widget="badge">
21            <!-- decoration-success="state == 'done'" -->
22            <!-- decoration-info="state == 'in_progress'" -->
23            <!-- decoration-warning="state == 'submitted'" -->
24        </tree>
25      </field>
26    </record>
27

```

```

projet_etudiant_views.xml ×
addons > gestion_projets_etudiants > views > projet_etudiant_views.xml
2   <odoo>
3
4     <!-- FORM VIEW -->
5     <record id="view_student_project_form" model="ir.ui.view">
6       <field name="name">student.project.form</field>
7       <field name="model">student.project</field>
8       <field name="arch" type="xml">
9         <form>
10           <header>
11             <button name="action_validate" string="Valider" type="object" class="oe_highlight" invisible="state != 'draft'"/>
12             <button name="action_in_progress" string="Commencer" type="object" class="oe_highlight" invisible="state != 'validation'"/>
13             <button name="action_submit" string="Rendre le Projet" type="object" class="oe_highlight" invisible="state != 'in progress'"/>
14             <button name="action_done" string="Noter & Clôturer" type="object" class="oe_highlight" invisible="state != 'submitted'"/>
15             <button name="action_send_defense_invitation" string="Envoyer Convocation" type="object" invisible="not defense_date"/>
16             <button name="action_draft" string="Remettre en Brouillon" type="object" invisible="state == 'draft'"/>
17             <field name="state" widget="statusbar" statusbar_visible="draft,validation,in_progress,submitted,done"/>
18           </header>
19           <sheet>
20             <div class="oe_title">
21               <label for="name" string="Titre du Projet"/>
22               <h1>
23                 <field name="name" placeholder="Ex: Mise en place d'un Pipeline DevOps..."/>
24               </h1>
25             </div>
26             <group>
27               <group>
28                 <field name="student_ids" widget="many2many_tags"/>
29                 <field name="supervisor_id"/>
30                 <field name="project_type"/>
31               </group>
32               <group>
33                 <field name="date_deadline"/>
34                 <field name="repository_url" widget="url"/>
35                 <field name="grade" widget="progressbar" options="{'max_value': 20}" readonly="1"/>
36               </group>
37             </group>
38           </sheet>
39         </form>
40       </field>
41     </record>
42

```

```
projet_etudiant_views.xml
addons>gestion_projets_etudiants>views>projet_etudiant_views.xml
2   <odo>
95     <record id="view_student_project_form" model="ir.ui.view">
96       <field name="arch" type="xml">
97         <form>
98           <sheet>
99             <!--
100               <notebook>
101                 <page string="Description &amp; Détails">
102                   <field name="description" placeholder="Détails du sujet, contraintes techniques..."/>
103                 </page>
104                 <page string="Soutenance &amp; Jury" invisible="state == 'draft'">
105                   <group>
106                     <group>
107                       <field name="defense_date"/>
108                       <field name="defense_location"/>
109                     </group>
110                   <group>
111                     <field name="jury_ids" widget="many2many_tags"/>
112                   </group>
113                 </group>
114                 <group string="Evaluation" invisible="state not in ['submitted', 'done']">
115                   <group>
116                     <field name="grade_report"/>
117                     <field name="grade_oral"/>
118                   </group>
119                   <group>
120                     <field name="grade_technical"/>
121                     <field name="grade" readonly="1" decoration-bf="1" string="Total / 20" widget="gauge" style="width:120px; height:90px; cursor:pointer;"/>
122                   </group>
123                 </group>
124               </page>
125               <page string="Livrables &amp; Jalons">
126                 <group>
127                   <field name="progress" widget="progressbar"/>
128                 </group>
129             </notebook>
130           </sheet>
131         </form>
132       </field>
133     </record>
134   </odo>
```

```
projet_etudiant_views.xml
addons>gestion_projets_etudiants>views>projet_etudiant_views.xml (preview)
2   <record id="view_student_project_form" model="ir.ui.view">
95     <field name="arch" type="xml">
96       <form>
97         <sheet>
98           <!--
99             <notebook>
100               <page string="Description &amp; Détails">
101                 <field name="description" placeholder="Détails du sujet, contraintes techniques..."/>
102               </page>
103               <page string="Soutenance &amp; Jury" invisible="state == 'draft'">
104                 <group>
105                   <group>
106                     <field name="defense_date"/>
107                     <field name="defense_location"/>
108                   </group>
109                 <group>
110                   <field name="jury_ids" widget="many2many_tags"/>
111                 </group>
112               </group>
113               <group string="Evaluation" invisible="state not in ['submitted', 'done']">
114                 <group>
115                   <field name="grade_report"/>
116                   <field name="grade_oral"/>
117                 </group>
118                 <group>
119                   <field name="grade_technical"/>
120                   <field name="grade" readonly="1" decoration-bf="1" string="Total / 20" widget="gauge" style="width:120px; height:90px; cursor:pointer;"/>
121                 </group>
122               </group>
123             </page>
124             <page string="Livrables &amp; Jalons">
125               <group>
126                 <field name="progress" widget="progressbar"/>
127               </group>
128             </page>
129           </notebook>
130         </sheet>
131       </form>
132     </field>
133   </record>
134
135   <!-- CALENDAR VIEW -->
136   <record id="view_student_project_calendar" model="ir.ui.view">
137     <field name="name">student.project.calendar</field>
138     <field name="model">student.project</field>
139     <field name="arch" type="xml">
140       <calendar string="Calendrier des Soutenances" date_start="defense_date" color="project_type" mode="month">
141         <field name="name"/>
142         <field name="student_ids"/>
143         <field name="defense_location"/>
144       </calendar>
145     </field>
146   </record>
```

```
projet_etudiant_views.xml ×
addons > gestion_projets_etudiants > views > projet_etudiant_views.xml
  2   <odoo>
  3     </record>
  4
  5     <!-- GRAPH VIEW -->
  6     <record id="view_student_project_graph" model="ir.ui.view">
  7       <field name="name">student.project.graph</field>
  8       <field name="model">student.project</field>
  9       <field name="arch" type="xml">
 10         <graph string="Statistiques Projets" type="bar">
 11           <field name="project_type"/>
 12           <field name="grade" type="measure"/>
 13         </graph>
 14       </field>
 15     </record>
 16
 17     <!-- PIVOT VIEW -->
 18     <record id="view_student_project_pivot" model="ir.ui.view">
 19       <field name="name">student.project.pivot</field>
 20       <field name="model">student.project</field>
 21       <field name="arch" type="xml">
 22         <pivot string="Analyse des Projets">
 23           <field name="project_type" type="row"/>
 24           <field name="supervisor_id" type="row"/>
 25           <field name="state" type="col"/>
 26           <field name="grade" type="measure"/>
 27         </pivot>
 28       </field>
 29     </record>
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
```

```
projet_etudiant_views.xml X
addons > gestion_projets_etudiants > views > projet_etudiant_views.xml
2   <odoo>
217   </record>
218
219   <!-- ACTIONS -->
220   <record id="action_student_project" model="ir.actions.act_window">
221       <field name="name">Projets Étudiants</field>
222       <field name="res_model">student.project</field>
223       <field name="view_mode">kanban,tree,form,calendar</field>
224       <field name="help" type="html">
225           <p class="o_view_nocontent_smiling_face">
226               Créez le premier projet étudiant !
227           </p>
228       </field>
229   </record>
230
231   <!-- MENUS -->
232   <menuitem id="menu_school_root" name="Student Project" sequence="10"/>
233
234   <menuitem id="menu_student_projects"
235       name="Mes Projets"
236       parent="menu_school_root"
237       action="action_student_project"
238       sequence="10"/>
239
240   <!-- REPORTING MENU -->
241   <record id="action_student_project_reporting" model="ir.actions.act_window">
242       <field name="name">Tableau de Bord</field>
243       <field name="res_model">student.project</field>
244       <field name="view_mode">graph,pivot</field>
245       <field name="help" type="html">
246           <p class="o_view_nocontent_smiling_face">
247               Aucune donnée pour les statistiques actuellement.
248           </p>
249       </field>
250   </record>
251
252   <menuitem id="menu_student_project_reporting"
253       name="Statistiques"
254       parent="menu_school_root"
255       action="action_student_project_reporting"
256       sequence="20"/>
257
258 </odoo>
```

## Étape 7 : Tester le module

The screenshot shows a Kanban Pipeline view with four columns: Brouillon, Validation, En cours, and Rendu. The 'test' project is in the Brouillon column, and the 'test Two' project is in the Rendu column. Each project has a progress bar and a note field. A red arrow points to the print icon in the top right corner.

**Vue Kanban (Pipeline)** : Tableau de bord visuel organisant les projets par colonnes selon leur étape dans le workflow (Brouillon, En cours, Rendu...), offrant une vue d'ensemble immédiate de l'avancement global.

The screenshot shows a detailed list view of projects. It includes columns for Titre du projet, Étudiant(s), Tuteur (Professeur), Type de Projet, Date limite, Progression, Note Finale (/20), and État. The 'test' project is listed with a Mini-projet type, due on 01/06/2026, at 0% progression and 0.00 note. The 'test Two' project is listed with a PFA type, due on 01/06/2026, at 100% progression and 10.00 note. A red arrow points to the print icon in the top right corner.

**Vue Liste Déttaillée** : Affichage tabulaire complet des projets permettant de comparer rapidement les informations clés (Type, Tuteur, Date limite) et de visualiser la progression technique et la note finale via des indicateurs graphiques.

The screenshot shows a calendar for January 2026. A pop-up window for the 'test Two' project shows the date (January 8, 2026), time (11:00 - 12:00 (1 hour)), title (Titre du projet test Two), student (Étudiant(s) Azure Interior, Brandon Freeman), and location (Salle / Lieu casa). A red arrow points to the print icon in the top right corner.

**Vue Calendrier des Soutenances** : Interface de planification visuelle permettant de consulter les dates de soutenance. Le pop-up affiche les détails essentiels (horaire, étudiants concernés, lieu) sans avoir à ouvrir la fiche projet.

The screenshot shows the 'Student Project' application interface. At the top, there's a purple header bar with the title 'Student Project' and navigation links 'Mes Projets' and 'Statistiques'. On the right side of the header, there are user icons and a red arrow pointing to a dropdown menu icon.

The main area is a Kanban board divided into two columns: 'Brouillon' (left) and 'Validation' (right). In the 'Brouillon' column, there's a card for a project titled 'test' which is a 'Mini-projet' and has a progress bar at 0%. In the 'Validation' column, there's a card for a project titled 'test Two' which is a 'PFA' and has a progress bar at 100% (10.00/20).

On the left side of the board, there are several filter and grouping options:

- Filters:** Mes Projets, PFE Uniquement, Projets non notés, Add Custom Filter.
- Group By:** Tuteur, État, Type, Add Custom Group.
- Favorites:** Save current search.
- Rendu:** test Two, PFA, 100 %, 10.00/20.
- Noté:**

**Outils de Recherche Avancée :** Démonstration des capacités de tri avec la barre de recherche déployée, montrant les filtres prédefinis (ex: "PFE Uniquement") et les options de regroupement pour organiser les données efficacement.

This screenshot shows the 'Student Project' interface with a purple header bar. The 'New' button is highlighted with a red arrow. Below it, the 'Projets Étudiants' section includes a 'Import records' button with a download icon.

The Kanban board has two columns: 'Brouillon' and 'validation'. In the 'Brouillon' column, there's a card for a project titled 'test' which is a 'Mini-projet' and has a progress bar at 0%.

**Fonctionnalité d'importation :** Aperçu du menu permettant l'importation de masse (via fichiers CSV ou Excel), facilitant la création rapide de multiples fiches projets sans saisie manuelle.

This screenshot shows the 'Student Project' interface with a purple header bar. The 'New' button is highlighted with a red arrow. The main area features a 'Title' input field with a placeholder 'Title', an 'Add' button, an 'Edit' button, and a trash bin icon.

The Kanban board has two columns: 'Brouillon' and 'Validation'. In the 'Brouillon' column, there's a card for a project titled 'test' which is a 'Mini-projet' and has a progress bar at 0%.

**Quick Create (Kanban) :** Fonctionnalité permettant d'ajouter un nouveau projet à la volée directement depuis la colonne "Brouillon" du Kanban, en saisissant simplement le titre, sans ouvrir le formulaire complet.

The screenshot shows the Odoo Student Project interface. On the left, there's a form for a project titled 'test'. It includes fields for 'Titre du Projet', 'Étudiant(s)', 'Tuteur (Professeur)', 'Type de Projet', 'Date limite', 'Lien Repository (Git)', and 'Note Finale (/20)'. A progress bar indicates 0%. Below the form are tabs for 'Description & Détails' and 'Livrables & Jalons'. On the right, a 'Chatter' sidebar shows a timeline of messages from 'Mitchell Admin' and 'OdooBot'.

**Interface de Notation et Chatter :** Vue complète d'un projet "Rendu". À gauche, la grille de notation (Rapport/Oral/Technique) calculant la note finale. À droite, le "Chatter" qui historique toutes les actions et communications.

This screenshot is similar to the previous one but includes several system messages from 'OdooBot' in the Chatter sidebar. These messages include notifications about a suggestion, a request, and errors related to email and SMS sending. The project details on the left remain the same.

Affiche le menu des messages système, alertant l'utilisateur des erreurs (ex: envoi d'email échoué) ou des actions requises.

Student Project Mes Projets Statistiques

New Projets Étudiants test

**Valider** Brouillon Val

**Titre du Projet**  
test

**Étudiant(s) ?**

Tuteur (Professeur)

Type de Projet

Description & Détails

Détails du sujet, cont...

Azure Interior  
Azure Interior, Brandon Freeman  
Azure Interior, Colleen Diaz  
Azure Interior, Nicole Ford  
Deco Addict  
Deco Addict, Addison Olson  
Deco Addict, Douglas Fletcher  
Deco Addict, Floyd Steward  
Search More...  
Start typing...

Date limite

Lien Repository (Git)  
Note Finale (/20)

Démonstration du champ de sélection multiple (Many2many) permettant d'assigner facilement un groupe d'étudiants au projet.

Student Project Mes Projets Statistiques

New Projets Étudiants test

**Valider** Brouillon Validation En cours Rendu Noté

**Titre du Projet**  
test

**Étudiant(s) ?**

Tuteur (Professeur)

Type de Projet Mini-projet

Description & Détails Livrables & Jalons

Progression 0 %

Ter... Titre du Livrable Date Lim... Fichier Joint

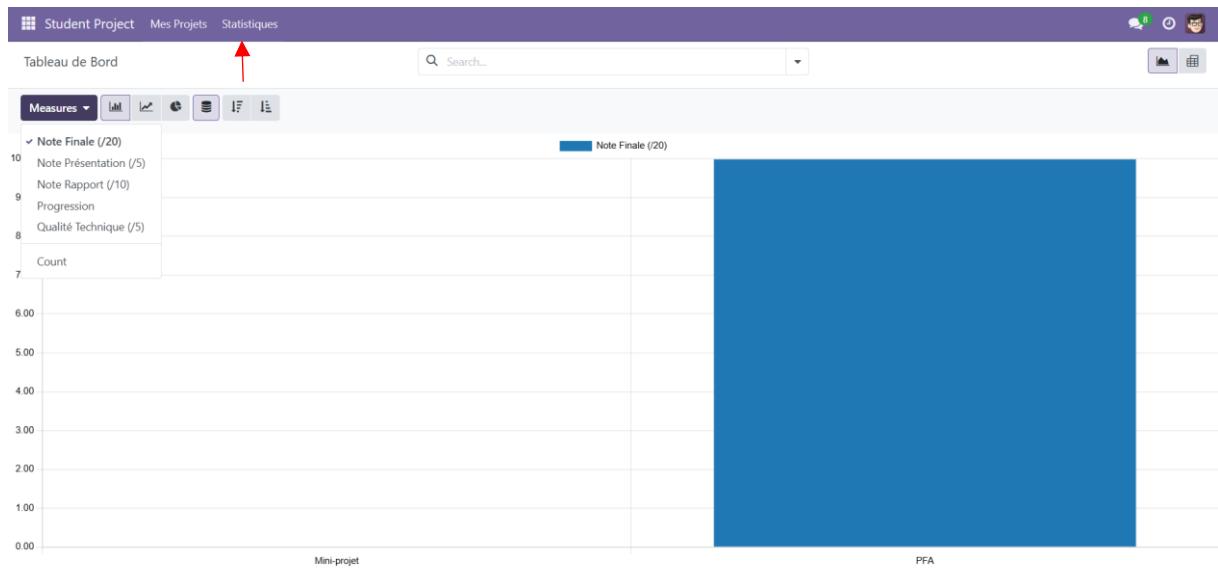
Add a line

Upload your file

**Onglet Livrables & Jalons :** Interface permettant de définir les étapes techniques du projet. C'est ici que les étudiants peuvent téléverser leurs fichiers, ce qui met à jour automatiquement la barre de progression globale.

The screenshot shows the 'Student Project' application interface. On the left, a project details page is displayed for a project titled 'test Two'. It includes fields for 'Étudiant(s)', 'Tuteur (Professeur)', 'Type de Projet', 'Date limite', 'Lien Repository (Git)', 'Note Finale (/20)', and various tabs for 'Description & Détails', 'Soutenance & Jury', and 'Livrables & Jalons'. On the right, a Chatter feed titled 'Today' shows a history of activity from 'Mitchell Admin' with various status updates and note changes.

L'écran permet de gérer toute la logistique de la soutenance (date, lieu, membres du jury) et d'accéder à la **grille d'évaluation**. On y voit le calcul détaillé de la note (Rapport/Oral/Technique) synthétisé par une jauge visuelle dynamique. Sur la droite, le **Chatter** (historique) joue un rôle crucial de traçabilité, enregistrant chaque modification de note, changement d'état et l'envoi automatique des emails de convocation aux participants.



Présente l'outil de reporting visuel, ici un diagramme en barres comparant la "Note Finale" par "Type de Projet", avec un menu permettant de changer les mesures analysées.

The screenshot shows a dashboard titled "Tableau de Bord". At the top, there are buttons for "Measures" and various filters. Below is a grid showing average final grades (Note Finale /20) for different categories. A red arrow points to a small icon in the top right corner of the dashboard area.

	Note Finale (/20)	Note Finale (/20)	Note Finale (/20)
⊕ Total	10.00	10.00	
⊕ Mini-projet			
⊕ None			
⊕ PFA	10.00	10.00	
⊕ Azure Interior, Brandon Freeman	10.00	10.00	

Montre la vue tableau croisé dynamique, idéale pour les responsables pédagogiques afin d'analyser les performances (notes moyennes) regroupées par tuteur, type de projet et état d'avancement.

The screenshot shows a list of users under the "Settings" tab. It includes columns for Name, Login, Language, Latest authentication, and Status. The status column uses color-coded buttons: blue for "Never Connected" and green for "Confirmed". A red arrow points to the "Settings" tab at the top left.

Name	Login	Language	Latest authentication	Status
Marc Demo	demo	English (US)		Never Connected
Mitchell Admin	admin	English (US)	01/01/2026 23:38:09	Confirmed
Professeur1	professeur1@gmail.com	English (US)		Never Connected
student1	student1@gmail.com	English (US)	01/01/2026 23:37:10	Confirmed

Affiche la liste des utilisateurs du système, permettant à l'administrateur de gérer les comptes (étudiants, professeurs) et de vérifier leur statut de connexion.

The screenshot shows the "Settings / Users" page for a user named "Professeur1". It includes fields for Name, Email Address, and Access Rights. Under "OTHER", there is a dropdown menu for "Gestion des Projets" with options: Étudiant, Tuteur / Professeur, and Responsable Pédagogique. A red arrow points to the "Send an Invitation Email" button at the top left.

Illustre la fiche de configuration d'un utilisateur où l'on attribue son niveau de droits spécifique pour le module (Étudiant, Tuteur ou Responsable Pédagogique).