

## **RAPPORT DE PROJET**

# **Développement d'un module Odoo Gestion des absences et justifications**

**Réalisé par :** BENCHAOU Salma

**Matière :** ERP

**Niveau :** 5 IIR G3 Maarif

**Année Universitaire :** 2025/2026

## Table des matières

<b>RAPPORT DE PROJET .....</b>	<b>1</b>
Introduction.....	4
1. Validation de l'environnement de développement.....	4
2. Architecture du module .....	4
3. Modélisation des données .....	6
3.1 Modèle Étudiant .....	6
3.2 Modèle Absence .....	7
3.3 Modèle Justification .....	8
4. Configuration du module .....	9
5. Gestion de la sécurité.....	10
6. Interface utilisateur.....	11
6.1 Vue arborescente (Tree View) .....	11
6.2 Vue formulaire (Form View) .....	11
7. Navigation et accessibilité.....	13
8. Tests et validation .....	14
Test 1 : Création d'un étudiant .....	14
Test 2 : Enregistrement d'une absence.....	15
Test 3 : Création d'une justification .....	16
9. Conclusion .....	17

## Liste des Figures

Figure 1: Vérification des conteneurs Docker Odoo .....	4
Figure 2: Arborescence du module dans VS Code .....	5
Figure 3:Code du modèle Étudiant (etudiant.py) .....	6
Figure 4: Code du modèle Absence (absence.py).....	7
Figure 5: Code du modèle Justification (justification.py).....	8
Figure 6: Fichier d'initialisation des modèles .....	9
Figure 7: Contenu du fichier manifest.py.....	10
Figure 8: Configuration des droits d'accès (ir.model.access.csv) .....	11
Figure 9: Code XML des vues (etudiant_view) .....	12
Figure 10: Code XML des vues (absence_view).....	12
Figure 11: Code XML des vues (justification_view) .....	13
Figure 12:Code XML du menu (menu.xml) .....	14
Figure 13: Interface du formulaire de création d'étudiant .....	15
Figure 14: Vue liste affichant l'étudiant créé .....	15
Figure 15:Interface du formulaire de création d'absence .....	16
Figure 16: Interface du formulaire de création de justification .....	16

## Introduction

Ce document présente la conception et la mise en œuvre d'un module Odoo dédié à la gestion des absences et des justifications des étudiants. Le projet suit une approche méthodique visant à créer une application fonctionnelle respectant l'architecture standard d'Odoo et déployée dans un environnement Docker. Le module permet d'enregistrer les étudiants, de suivre leurs absences et de gérer les justifications associées de manière centralisée et efficace.

## 1. Validation de l'environnement de développement

La première phase du projet consiste à s'assurer du bon fonctionnement de l'environnement Odoo. Le projet utilise Docker pour conteneuriser l'application, garantissant ainsi une portabilité et une reproductibilité optimales. La commande docker ps confirme que deux conteneurs sont actifs : le conteneur odoo\_app (sur le port 8069) et le conteneur odoo\_db exécutant PostgreSQL 16 (sur le port 5433). Cette configuration assure une séparation claire entre la couche applicative et la base de données, facilitant ainsi la maintenance et le déploiement.

```
PS C:\Users\User\Desktop\odoo-docker> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9c16b4b907c1 odoo:17.0 "/entrypoint.sh odoo..." 49 minutes ago Up About a minute 0.0.0.0:8069->8069/tcp, 8071-8072/tcp odoo_app
a24629ce189e postgres:16 "docker-entrypoint.s..." 7 hours ago Up 40 minutes 0.0.0.0:5433->5432/tcp odoo_db
a27e61e7fcca movie_reco-movie-service "java -jar /app.jar" 5 weeks ago Up 40 minutes 0.0.0.0:8081->8081/tcp movie_reco-movie-service-
1
5e71c8e573eb postgres:15 "docker-entrypoint.s..." 5 weeks ago Up 40 minutes (healthy) 0.0.0.0:5432->5432/tcp movie_reco-postgres-1
a02c8cb634ae redis:7 "docker-entrypoint.s..." 5 weeks ago Up 40 minutes (healthy) 0.0.0.0:6379->6379/tcp movie_reco-redis-1
PS C:\Users\User\Desktop\odoo-docker>
```

Figure 1: Vérification des conteneurs Docker Odoo

## 2. Architecture du module

Le module gestion\_absences a été créé dans le répertoire addons d'Odoo, aux côtés d'un autre module appelé gestion\_etudiants. L'organisation des fichiers respecte les conventions du framework et assure une maintenance facilitée du code. La structure comprend les répertoires models pour la logique métier, views pour l'interface utilisateur, et security pour la gestion des droits d'accès.

```
gestion_absences/
    ├── __init__.py
    ├── __manifest__.py
    ├── models/
    │   ├── __init__.py
    │   ├── etudiant.py
    │   ├── absence.py
    │   └── justification.py
    ├── views/
    │   ├── etudiant_view.xml
    │   ├── absence_view.xml
    │   └── justification_view.xml
```

```

  └── menu.xml
  └── security/
    └── ir.model.access.csv
  
```

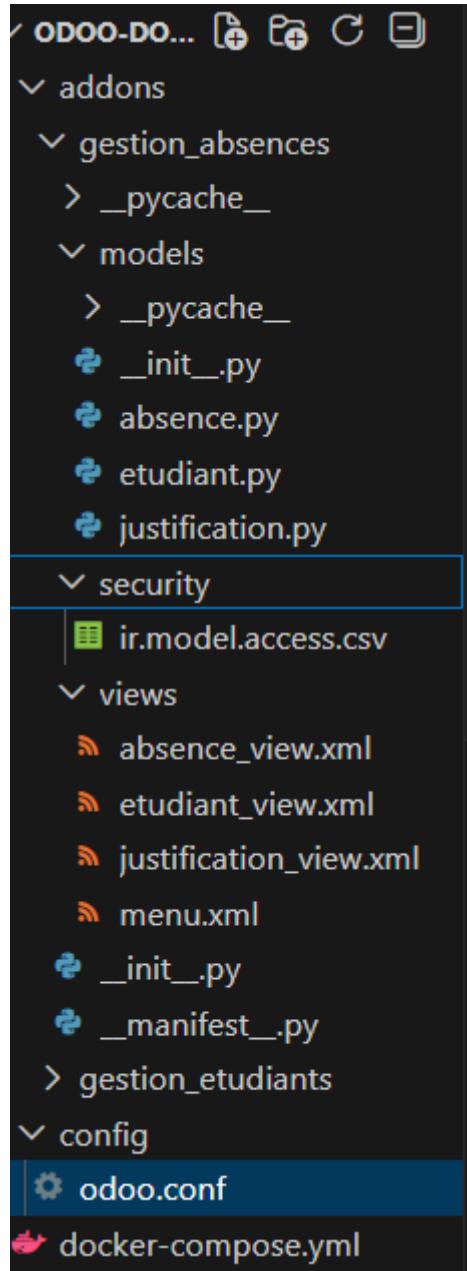


Figure 2: Arborescence du module dans VS Code

Cette architecture modulaire permet une séparation claire des responsabilités et facilite l'évolution future du module. Chaque composant est isolé dans son propre fichier, ce qui améliore la lisibilité et la maintenabilité du code.

### 3. Modélisation des données

Le module repose sur trois modèles principaux qui interagissent entre eux pour offrir une gestion complète des absences et des justifications.

### 3.1 Modèle Étudiant

Le modèle `Etudiant`, défini dans le fichier `etudiant.py`, hérite de la classe `models.Model` d'Odoo et définit la structure des données pour les étudiants. Les attributs suivants sont définis:

- **\_name** : identifiant technique du modèle (`gestion.etudiant`)
  - **\_description** : description fonctionnelle (`Etudiant`)
  - **name** : champ de type Char pour le nom de l'étudiant (obligatoire)
  - **prenom** : champ de type Char pour le prénom de l'étudiant
  - **cne** : champ de type Char pour le code national de l'étudiant
  - **filiere** : champ de type Char pour la filière de l'étudiant

Cette modélisation permet d'identifier de manière unique chaque étudiant dans le système et de le lier aux absences enregistrées.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Run, as well as a search bar labeled "odoor-decker".
- Left Sidebar (Explorer):** Shows the project structure:
  - OPEN EDITORS
  - ODOO-DO...
  - addons (selected):
    - gestion\_absences
      - models
        - student.py (selected)
        - absence.py
      - views
        - absence\_view.xml
        - student\_view.xml
        - justification\_view.xml
        - menu.xml
      - \_\_init\_\_.py
      - \_\_manifest\_\_.py
    - security
      - ir.model.access.csv
    - config
      - odoo.conf
      - docker-compose.yml
  - Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTE-8 | CRLF | { } Python | 3.13.11

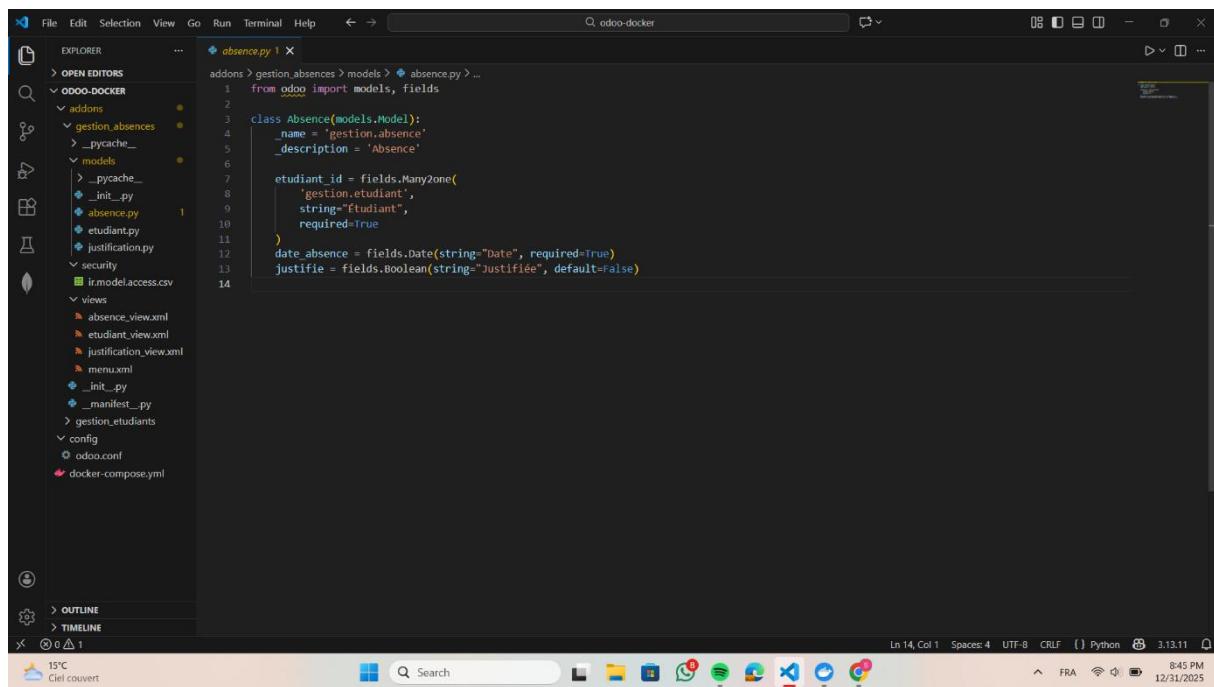
Figure 3: Code du modèle Étudiant (etudiant.py)

### 3.2 Modèle Absence

Le modèle `Absence`, défini dans le fichier `absence.py`, gère les enregistrements d'absence des étudiants. Les attributs suivants sont définis :

- `_name` : identifiant technique du modèle (`gestion.absence`)
- `_description` : description fonctionnelle (`Absence`)
- `etudiant_id` : champ de type Many2one pointant vers le modèle `gestion.etudiant` (obligatoire)
- `date_absence` : champ de type Date pour la date de l'absence (obligatoire)
- `justifie` : champ de type Boolean pour indiquer si l'absence est justifiée (valeur par défaut : False)

Ce modèle permet de suivre les absences de chaque étudiant et de marquer si elles ont été justifiées ou non. La relation Many2one avec le modèle Étudiant assure l'intégrité référentielle des données.



```

File Edit Selection View Go Run Terminal Help ← → Q odooreditor
EXPLORER ... absence.py 1 ✎
OPEN EDITORS
ODOO-DOCKER
addons > gestion.absences > models > absence.py > ...
1  from odoo import models, fields
2
3  class Absence(models.Model):
4      _name = 'gestion.absence'
5      _description = 'Absence'
6
7      etudiant_id = fields.Many2one(
8          'gestion.etudiant',
9          string="Etudiant",
10         required=True
11     )
12
13      date_absence = fields.Date(string="Date", required=True)
14      justifie = fields.Boolean(string="Justifiée", default=False)

```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for Open Editors, Odoo-Docker, Explorer, Search, Run, Terminal, Help, and a refresh button.
- Left Sidebar (EXPLORER):**
  - Open Editors: absence.py
  - ODOO-DOCKER: addons/gestion.absences/models/absence.py
  - addons:
    - gestion.absences: \_\_pycache\_\_, absence.py, etudiant.py, justification.py
    - models: \_\_pycache\_\_, \_\_init\_\_.py, absence.py, etudiant.py, justification.py
    - security: ir.model.access.csv
    - views: absence\_view.xml, etudiant\_view.xml, justification\_view.xml, menuxml, \_\_init\_\_.py, \_\_manifest\_\_.py
    - gestion.etudiants: config, odoo.conf
    - config: docker-compose.yml
- Central Editor Area:** Displays the Python code for the `absence.py` file.
- Bottom Status Bar:**
  - Ln 14, Col 1 Spaces:4 UTF-8 CRLF (Python 3.13.11)
  - Icons for FRA, 845 PM, 12/31/2025
- System Tray:** Shows weather (15°C, Ciel couvert) and system icons.

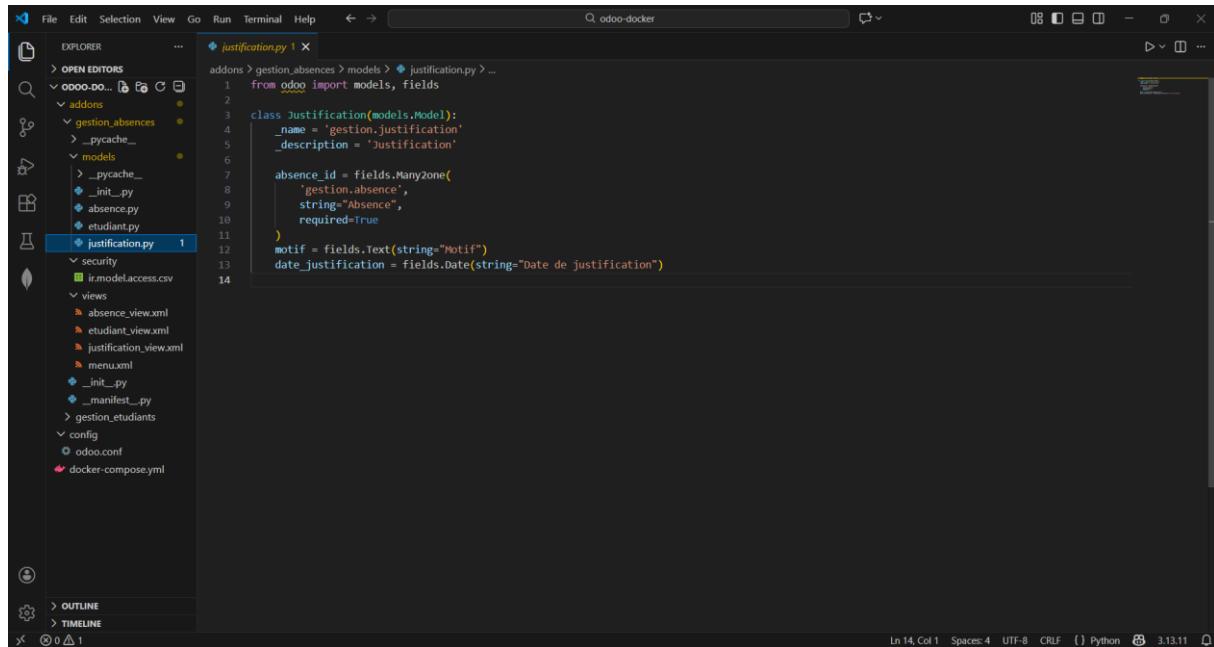
Figure 4: Code du modèle `Absence` (`absence.py`)

### 3.3 Modèle Justification

Le modèle **Justification**, défini dans le fichier `justification.py`, gère les justifications associées aux absences. Les attributs suivants sont définis :

- **\_name** : identifiant technique du modèle (`gestion.justification`)
- **\_description** : description fonctionnelle (`Justification`)
- **absence\_id** : champ de type Many2one pointant vers le modèle `gestion.absence` (obligatoire)
- **motif** : champ de type Text pour le motif de la justification
- **date\_justification** : champ de type Date pour la date de soumission de la justification

Ce modèle permet d'associer une justification à une absence spécifique et d'enregistrer les informations pertinentes. La relation Many2one avec le modèle Absence établit un lien clair entre les deux entités.



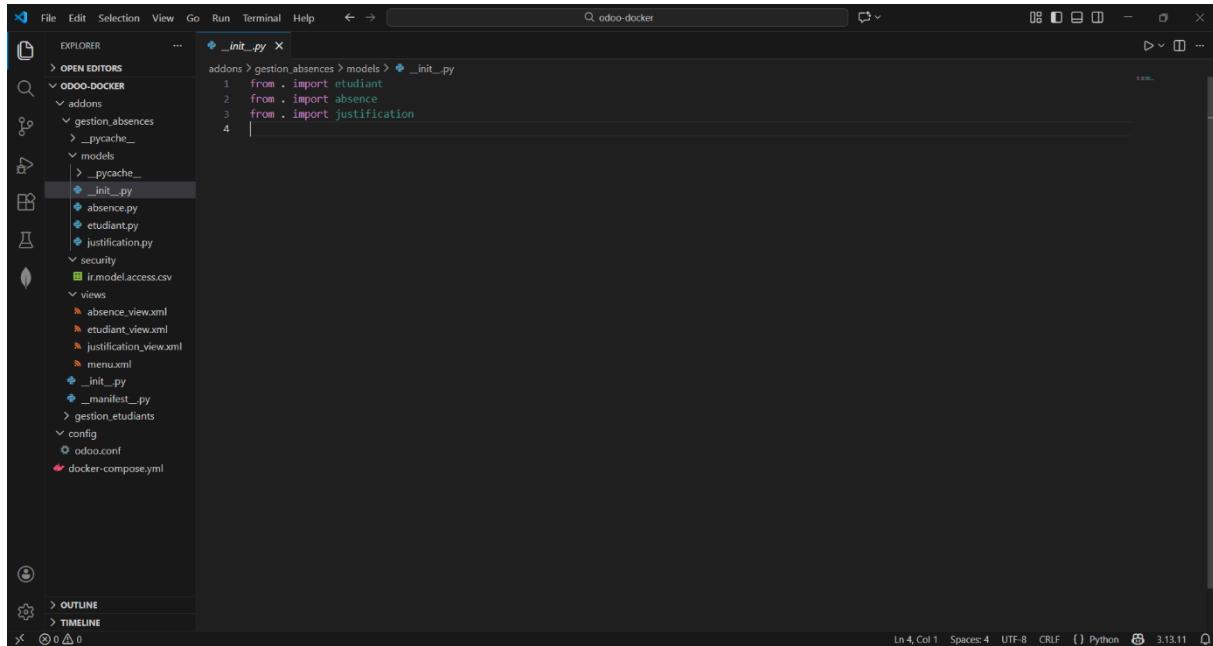
```

File Edit Selection View Go Run Terminal Help ← → Q odoodecker
OPEN EDITORS
addons > gestion.absences > models > justification.py ...
1  from odoo import models, fields
2
3  class Justification(models.Model):
4      _name = 'gestion.justification'
5      _description = 'Justification'
6
7      absence_id = fields.Many2one(
8          'gestion.absence',
9          string="Absence",
10         required=True
11     )
12     motif = fields.Text(string="Motif")
13     date_justification = fields.Date(string="Date de justification")
14

```

Figure 5: Code du modèle **Justification** (`justification.py`)

Le fichier `__init__.py` du dossier `models` assure l'importation correcte de tous les modèles, permettant à Odoo de les charger au démarrage du module.



```
File Edit Selection View Go Run Terminal Help ← → Q odoo-docker
EXPLORER ... _init_.py x
OPEN EDITORS
ODOO-DOCKER
addons
  gestion_absences
    models
      __init__.py
      absence.py
      etudiant.py
      justification.py
    security
      ir.model.access.csv
  views
    absence_view.xml
    etudiant_view.xml
    justification_view.xml
    menuxml
    __init__.py
    __manifest__.py
  config
    odoo.conf
  docker-compose.yml

OUTLINE
TIMELINE
Ln 4, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.11
```

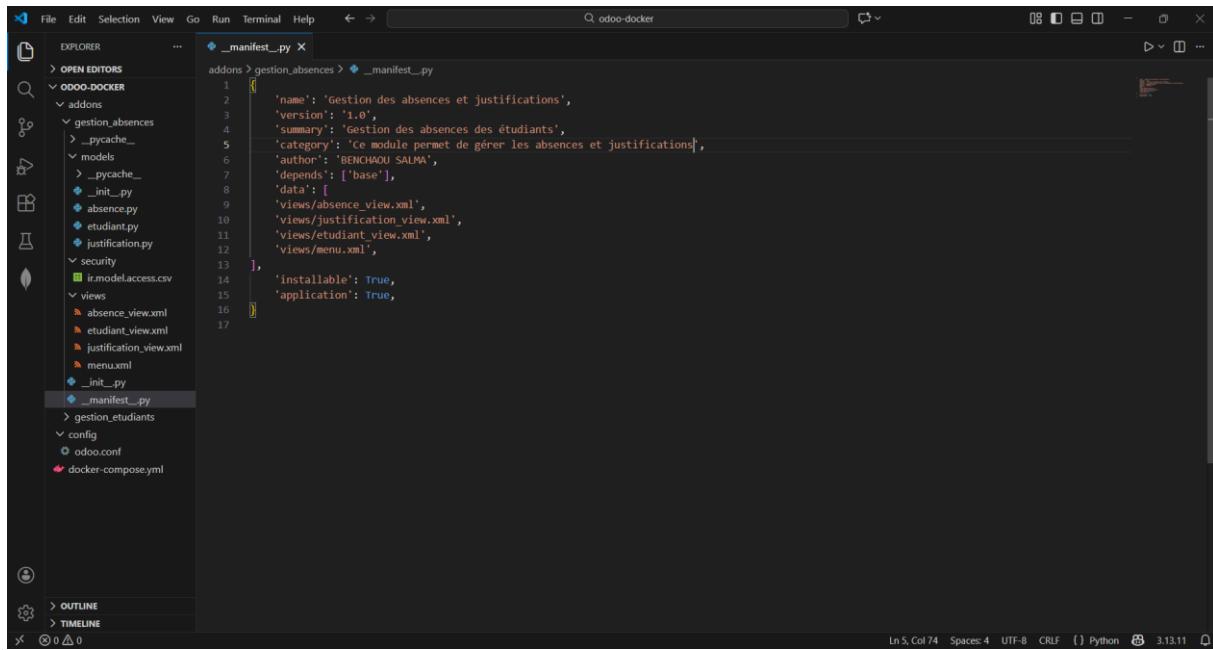
Figure 6: Fichier d'initialisation des modèles

## 4. Configuration du module

Le fichier `__manifest__.py` constitue le point d'entrée du module. Il contient les métadonnées essentielles qui permettent à Odoo de reconnaître et d'intégrer correctement le module dans le système :

- **name** : "Gestion des absences et justifications"
- **version** : "1.0"
- **summary** : "Gestion des absences des étudiants"
- **category** : "Ce module permet de gérer les absences et justifications"
- **author** : "BENCHAOU SALMA"
- **depends** : ['base'] (dépendance au module de base d'Odoo)
- **data** : liste des fichiers à charger dans l'ordre (vues XML et fichier de sécurité)
  - 'views/absence\_view.xml'
  - 'views/justification\_view.xml'
  - 'views/etudiant\_view.xml'
  - 'views/menu.xml'
- **installable** : True
- **application** : True

Cette configuration permet à Odoo de reconnaître et d'intégrer correctement le module dans le système en tant qu'application autonome. L'ordre de chargement des fichiers dans la section `data` est important pour assurer que les dépendances entre les vues sont respectées.



```

1  'name': 'gestion des absences et justifications',
2  'version': '1.0',
3  'summary': 'Gestion des absences des étudiants',
4  'category': 'Ce module permet de gérer les absences et justifications',
5  'author': 'BERACHEOU SALMA',
6  'depends': ['base'],
7  'data': [
8      'views/absence_view.xml',
9      'views/justification_view.xml',
10     'views/etudiant_view.xml',
11     'views/menu.xml',
12 ],
13 ],
14 'installable': True,
15 'application': True,
16 ]
17

```

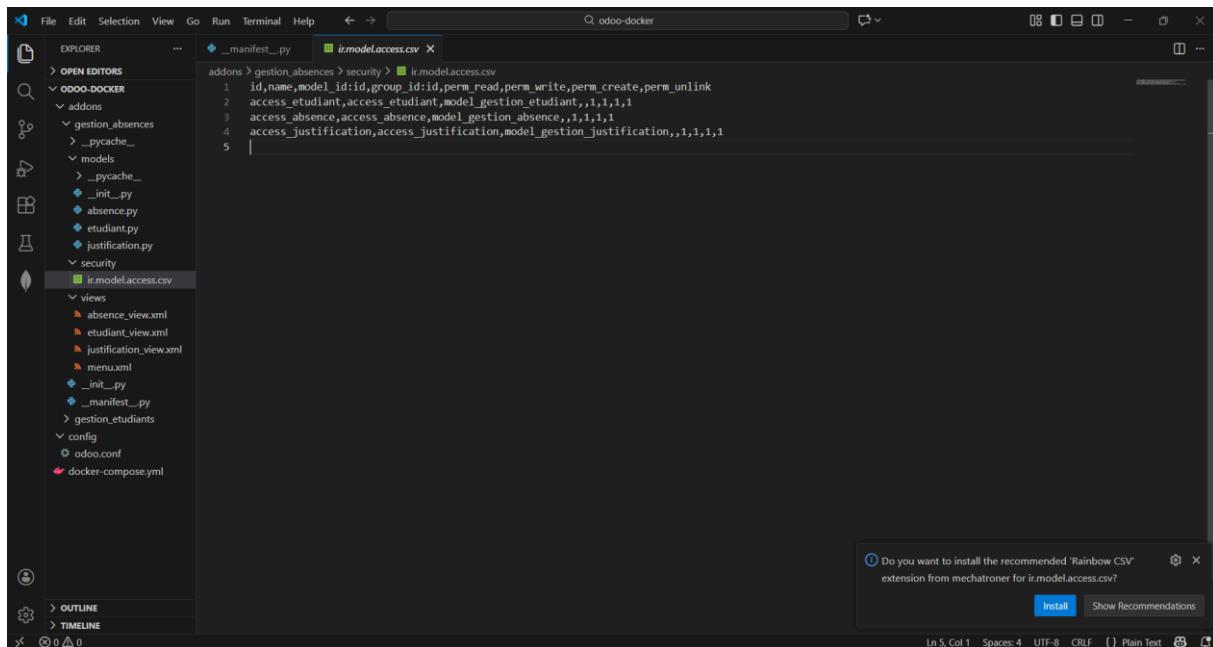
Figure 7: Contenu du fichier manifest.py

## 5. Gestion de la sécurité

La sécurité du module est assurée par le fichier `ir.model.access.csv` qui définit les droits d'accès aux trois modèles du module. Le fichier contient trois règles qui accordent tous les priviléges (lecture, écriture, création, suppression) au groupe `base.group_user`, correspondant aux utilisateurs internes d'Odoo :

1. **access\_etudiant** : règle d'accès pour le modèle `gestion.etudiant`
2. **access\_absence** : règle d'accès pour le modèle `gestion.absence`
3. **access\_justification** : règle d'accès pour le modèle `gestion.justification`

Cette configuration garantit que les utilisateurs autorisés peuvent effectuer toutes les opérations nécessaires sur les étudiants, les absences et les justifications. Pour des besoins plus avancés, il serait possible de créer des groupes spécifiques avec des droits différenciés (par exemple, un groupe pour les enseignants et un autre pour les administrateurs).



```

_id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
1,access_etudiant,access_etudiant,model_gestion_etudiant,,1,1,1,1
2,access_absence,access_absence,model_gestion_absence,,1,1,1,1
3,access_justification,access_justification,model_gestion_justification,,1,1,1,1
4,access_absence,access_absence,model_gestion_absence,,1,1,1,1
5,

```

Do you want to install the recommended 'Rainbow CSV' extension from mechatroner for ir.model.access.csv?  
 Show Recommendations

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF Plain Text

Figure 8: Configuration des droits d'accès (ir.model.access.csv)

## 6. Interface utilisateur

L'interface du module se compose de plusieurs vues complémentaires développées en XML pour chaque modèle. Ces vues offrent une expérience utilisateur intuitive et cohérente.

### 6.1 Vue arborescente (Tree View)

Chaque modèle dispose d'une vue arborescente qui présente les enregistrements sous forme de tableau :

**Vue Étudiant** : affiche le nom, le prénom, le CNE et la filière de chaque étudiant.

**Vue Absence** : affiche l'étudiant concerné, la date de l'absence et le statut de justification.

**Vue Justification** : affiche l'absence associée et la date de justification.

Ces vues permettent une visualisation rapide et une recherche efficace des enregistrements. La barre de recherche intégrée facilite le filtrage des données.

### 6.2 Vue formulaire (Form View)

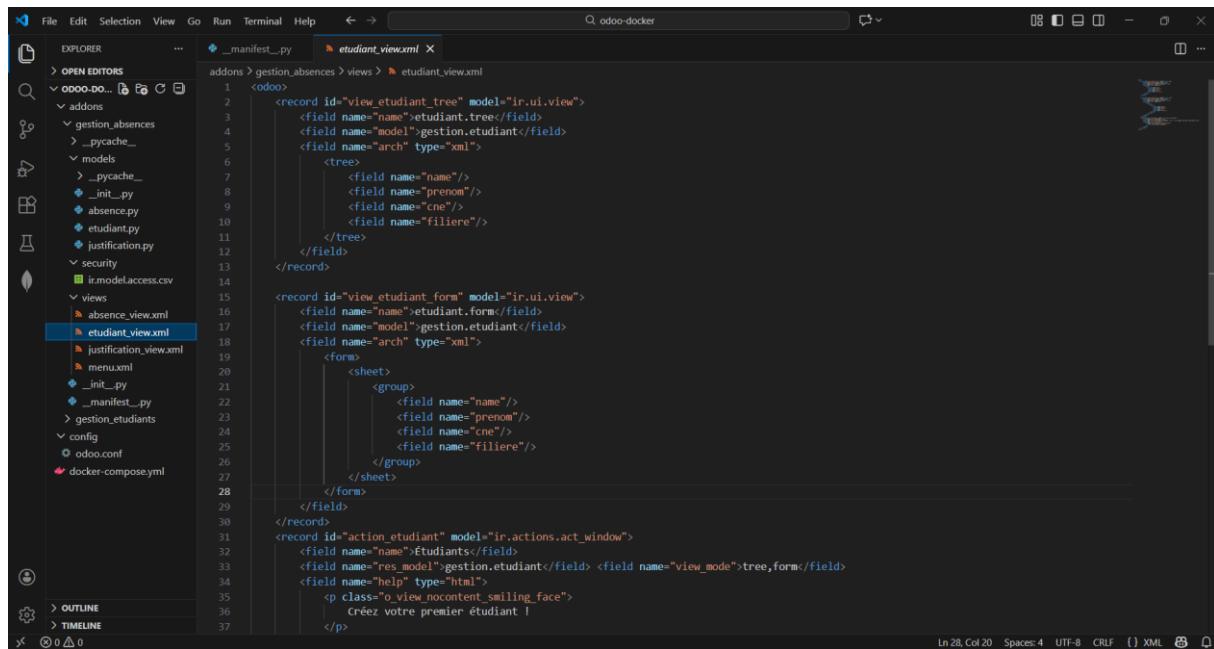
Chaque modèle dispose également d'une vue formulaire qui offre une interface intuitive pour la saisie et la modification des informations :

**Formulaire Étudiant** : permet de saisir le nom, le prénom, le CNE et la filière dans une structure organisée en groupe.

**Formulaire Absence** : permet de sélectionner un étudiant via un champ Many2one, de définir la date de l'absence et de cocher si elle est justifiée.

**Formulaire Justification** : permet de sélectionner une absence via un champ Many2one, de saisir le motif de la justification et d'enregistrer la date.

Ces formulaires sont organisés dans une structure sheet contenant des group qui regroupent logiquement les champs, facilitant ainsi la navigation et la compréhension pour l'utilisateur final.



```

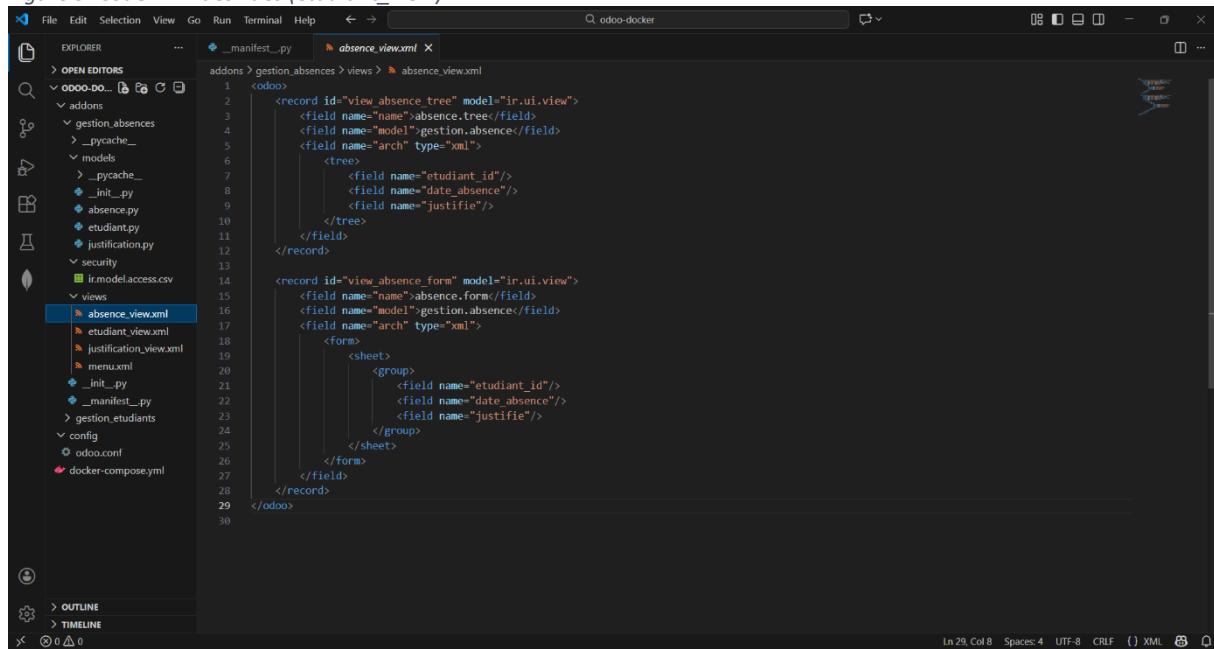
<odoo>
    <record id="view_etudiant_tree" model="ir.ui.view">
        <field name="name">etudiant.tree</field>
        <field name="model">gestion.etudiant</field>
        <field name="arch" type="xml">
            <tree>
                <field name="name"/>
                <field name="prenom"/>
                <field name="cne"/>
                <field name="filiere"/>
            </tree>
        </field>
    </record>

    <record id="view_etudiant_form" model="ir.ui.view">
        <field name="name">etudiant.form</field>
        <field name="model">gestion.etudiant</field>
        <field name="arch" type="xml">
            <form>
                <sheet>
                    <group>
                        <field name="name"/>
                        <field name="prenom"/>
                        <field name="cne"/>
                        <field name="filiere"/>
                    </group>
                </sheet>
            </form>
        </field>
    </record>
    <record id="action_etudiant" model="ir.actions.act_window">
        <field name="name">Etudiants</field>
        <field name="res_model">gestion.etudiant</field>
        <field name="view_mode">tree,form</field>
        <field name="help" type="html">
            <p class="o_view_nocontent_smiling_face">Créez votre premier étudiant !</p>
        </field>
    </record>
</odoo>

```

Ln 28, Col 20 Spaces: 4 UTF-8 CRLF {} XML

Figure 9: Code XML des vues (etudiant\_view)



```

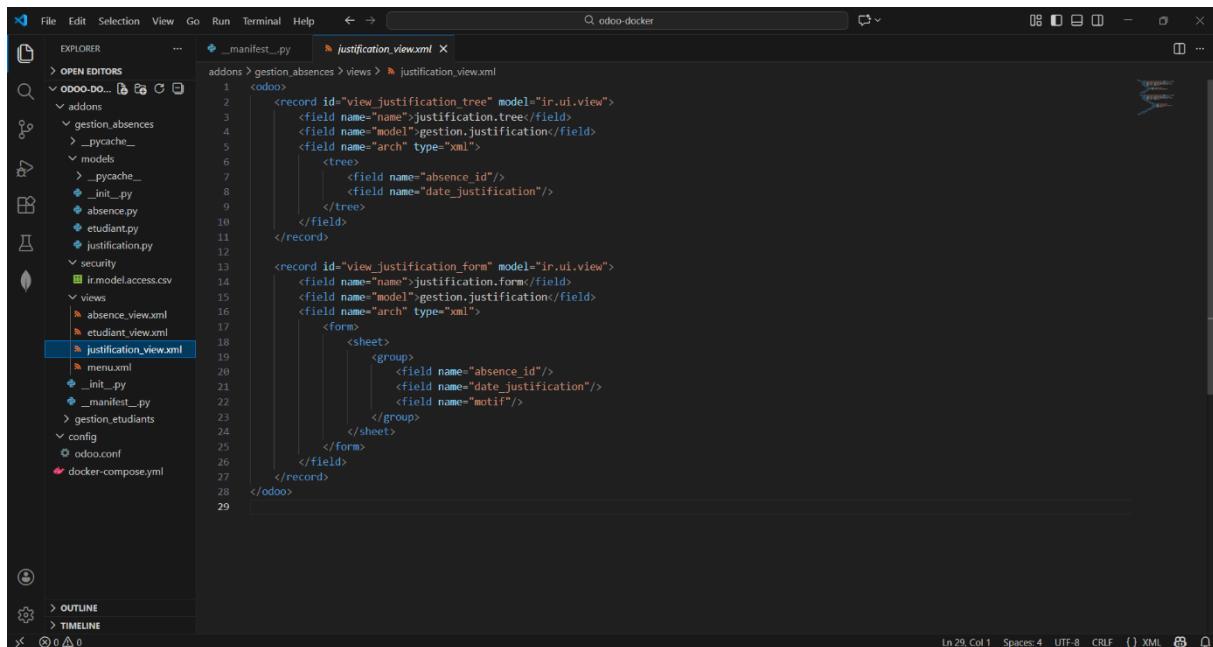
<odoo>
    <record id="view_absence_tree" model="ir.ui.view">
        <field name="name">absence.tree</field>
        <field name="model">gestion.absence</field>
        <field name="arch" type="xml">
            <tree>
                <field name="etudiant_id"/>
                <field name="date_absence"/>
                <field name="justifie"/>
            </tree>
        </field>
    </record>

    <record id="view_absence_form" model="ir.ui.view">
        <field name="name">absence.form</field>
        <field name="model">gestion.absence</field>
        <field name="arch" type="xml">
            <form>
                <sheet>
                    <group>
                        <field name="etudiant_id"/>
                        <field name="date_absence"/>
                        <field name="justifie"/>
                    </group>
                </sheet>
            </form>
        </field>
    </record>
</odoo>

```

Ln 29, Col 8 Spaces: 4 UTF-8 CRLF {} XML

Figure 10: Code XML des vues (absence\_view)



```

<odoo>
    <record id="view_justification_tree" model="ir.ui.view">
        <field name="name">justification.tree</field>
        <field name="model">gestion.justification</field>
        <field name="arch" type="xml">
            <tree>
                <field name="absence_id"/>
                <field name="date_justification"/>
            </tree>
        </field>
    </record>

    <record id="view_justification_form" model="ir.ui.view">
        <field name="name">justification.form</field>
        <field name="model">gestion.justification</field>
        <field name="arch" type="xml">
            <form>
                <sheet>
                    <group>
                        <field name="absence_id"/>
                        <field name="date_justification"/>
                        <field name="motif"/>
                    </group>
                </sheet>
            </form>
        </field>
    </record>
</odoo>

```

Figure 11: Code XML des vues (justification\_view)

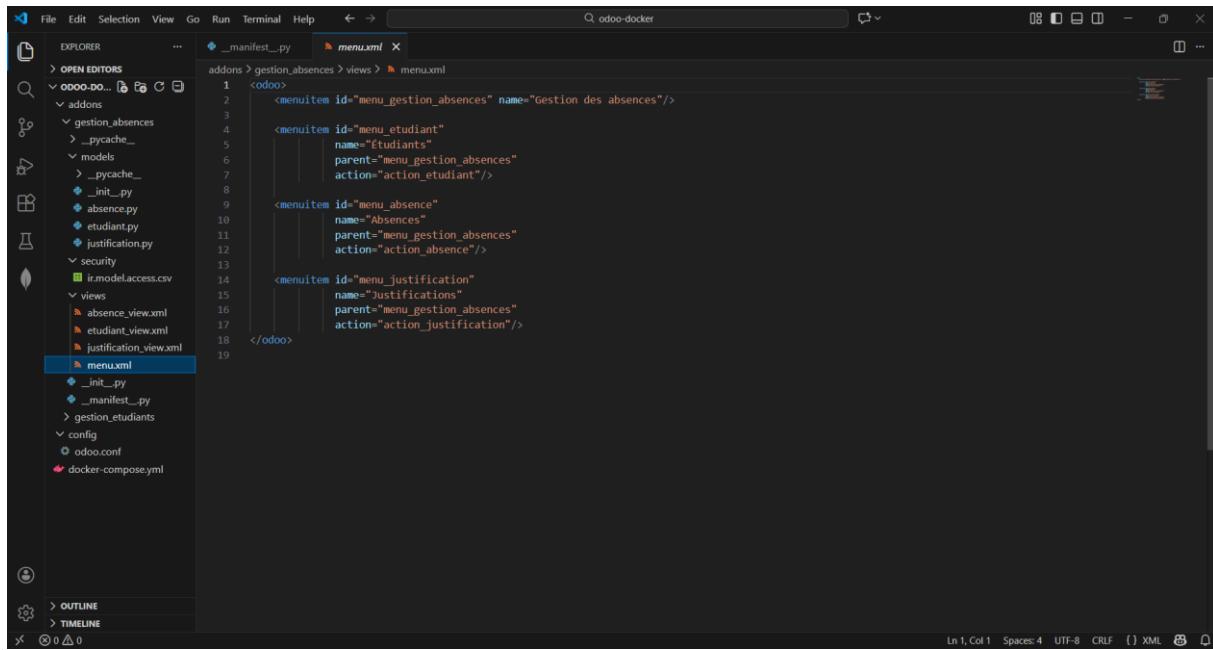
## 7. Navigation et accessibilité

Un menu principal intitulé « Gestion des Absences » a été intégré dans la barre de navigation d'Odoo. Le fichier `menu.xml` définit trois actions de type `ir.actions.act_window` qui pointent respectivement vers les modèles `gestion.etudiant`, `gestion.absence` et `gestion.justification`, et affichent les vues tree et form correspondantes.

La structure de navigation est organisée de manière hiérarchique :

- **Menu racine** : "Gestion des absences"
  - **Sous-menu 1** : "Étudiants" (accès à la liste et aux formulaires des étudiants)
  - **Sous-menu 2** : "Absences" (accès à la liste et aux formulaires des absences)
  - **Sous-menu 3** : "Justifications" (accès à la liste et aux formulaires des justifications)

Cette organisation permet aux utilisateurs d'accéder rapidement aux différentes fonctionnalités du module depuis n'importe quelle page de l'application. Le menu principal est visible dans la barre de navigation supérieure avec une icône distinctive.



```

<odoo>
    <menuitem id="menu_gestion_absences" name="Gestion des absences"/>
    <menuitem id="menu_etudiant"
              name="Etudiants"
              parent="menu_gestion_absences"
              action="action_etudiant"/>
    <menuitem id="menu_absence"
              name="Absences"
              parent="menu_gestion_absences"
              action="action_absence"/>
    <menuitem id="menu_justification"
              name="Justifications"
              parent="menu_gestion_absences"
              action="action_justification"/>
</odoo>

```

Figure 12:Code XML du menu (menu.xml)

## 8. Tests et validation

La phase de validation a confirmé le bon fonctionnement de toutes les fonctionnalités du module. Plusieurs tests ont été effectués pour vérifier l'intégrité du système.

### Test 1 : Création d'un étudiant

Un étudiant de test nommé "benchaou salma" avec le CNE "BB123456" et la filière "IIR" a été créé avec succès. Le formulaire de saisie s'affiche correctement avec tous les champs requis. L'enregistrement apparaît dans la vue liste avec toutes les informations saisies.

The screenshot shows a web-based form titled "Gestion des absences" under the "Étudiants" tab. The form contains four fields: Nom (benchaou), Prénom (salma), CNE (BB123456), and Filière (IIR). The "Filière" field has a dropdown arrow indicating it is a select menu.

Figure 13: Interface du formulaire de création d'étudiant

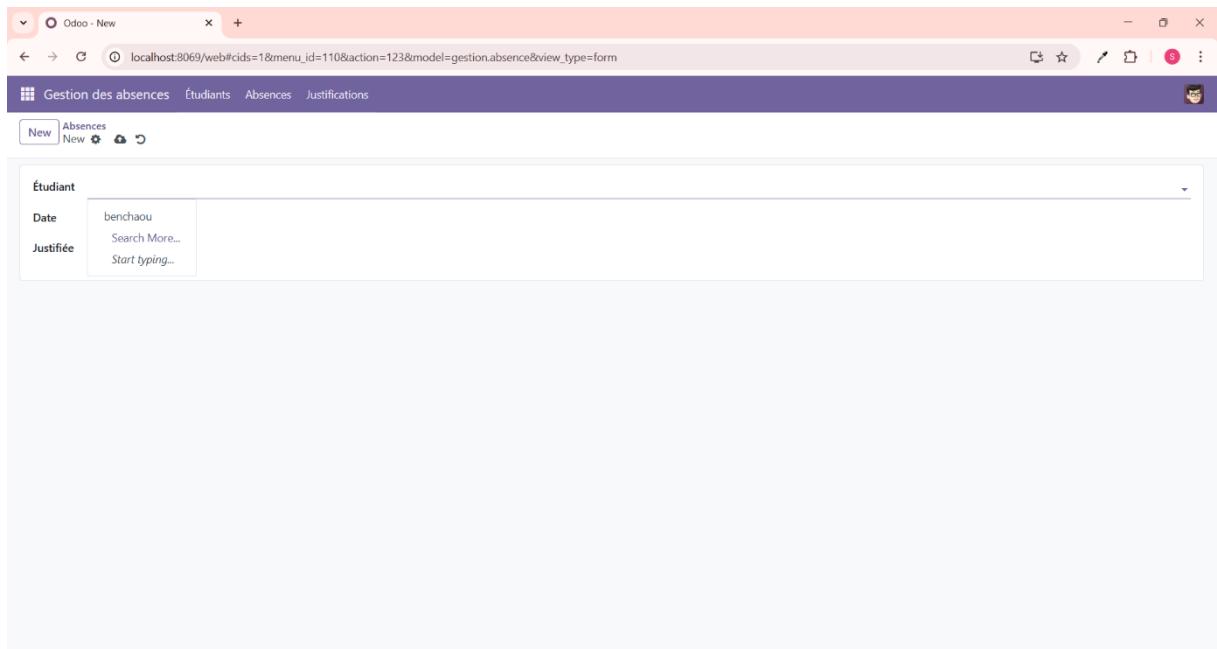
The screenshot shows a list view of students. A single record is displayed with the following details: Nom (benchaou), Prénom (salma), CNE (BB123456), and Filière (IIR). The list view includes a header row with columns for Nom, Prénom, CNE, and Filière, and a footer row showing page 1-1 of 1.

Nom	Prénom	CNE	Filière
benchaou	salma	BB123456	IIR

Figure 14: Vue liste affichant l'étudiant créé

## Test 2 : Enregistrement d'une absence

Une absence a été enregistrée pour l'étudiant "benchaou" à la date du 12/30/2025. Le champ "Justifié" a été coché pour indiquer que l'absence est justifiée. Le formulaire permet de sélectionner facilement l'étudiant via un menu déroulant et de définir la date via un sélecteur de date.

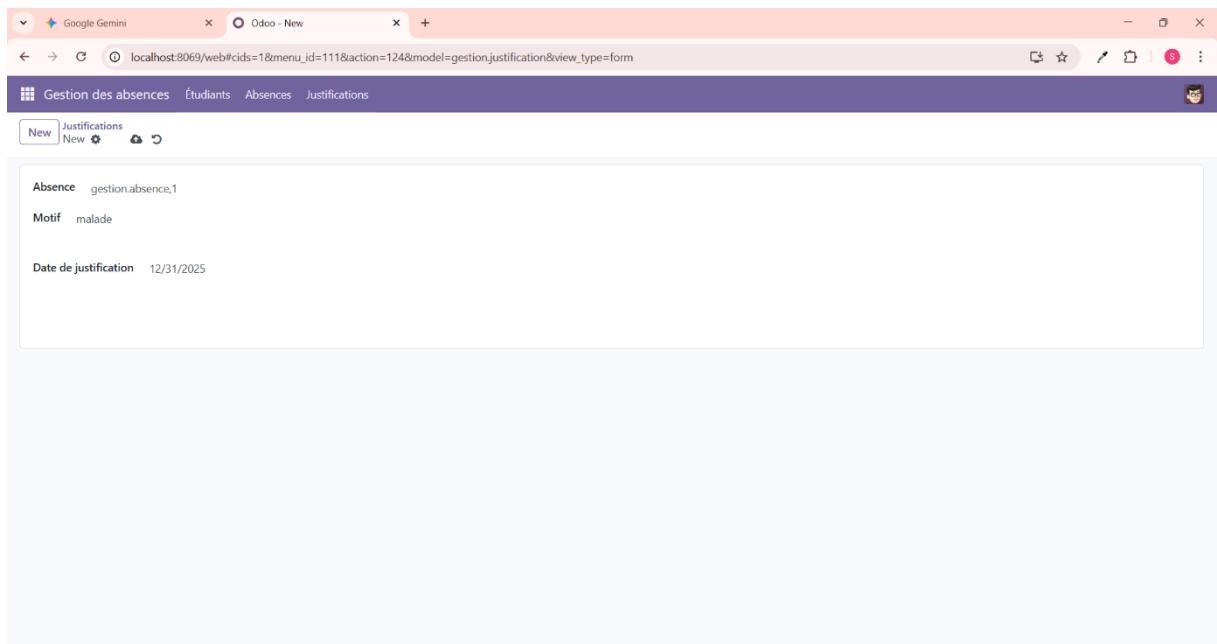


Étudiant	Date	Justifiée	Action
	benchaou	Search More... Start typing...	New

Figure 15: Interface du formulaire de création d'absence

### Test 3 : Création d'une justification

Une justification a été créée pour l'absence enregistrée précédemment, avec le motif "malade" et la date de justification du 12/31/2025. Le système affiche correctement la référence de l'absence associée (gestion.absence,1) et permet de saisir le motif dans un champ texte.



Absence	Motif	Date de justification
gestion.absence,1	malade	12/31/2025

Figure 16: Interface du formulaire de création de justification

Tous les tests confirment que les opérations CRUD (création, lecture, mise à jour, suppression) sont opérationnelles pour les trois modèles. Les relations Many2one fonctionnent correctement et permettent de lier les absences aux étudiants et les justifications aux absences. Les barres de recherche et les boutons "New" sont accessibles dans toutes les vues, facilitant la gestion des enregistrements.

## 9. Conclusion

Ce projet a permis de développer un module Odoo pleinement fonctionnel pour la gestion des absences et des justifications des étudiants, déployé dans un environnement Docker professionnel. L'application respecte rigoureusement les standards du framework Odoo tant au niveau de l'architecture que des bonnes pratiques de développement.

Le module intègre trois modèles interconnectés (Étudiant, Absence, Justification) qui permettent un suivi complet et efficace des absences. Les relations Many2one assurent l'intégrité référentielle des données et facilitent la navigation entre les différentes entités. Le code est clair, bien structuré et documenté, facilitant ainsi sa maintenance et son évolution future.

Le module constitue une base solide qui pourra être enrichie ultérieurement avec des fonctionnalités avancées telles que :

- L'envoi de notifications automatiques aux étudiants et aux responsables en cas d'absence non justifiée
- L'ajout de rapports et de tableaux de bord pour visualiser les statistiques d'absence par étudiant, par filière ou par période
- L'intégration d'un système de workflow pour valider les justifications par un responsable pédagogique
- L'ajout de champs calculés pour afficher automatiquement le nombre d'absences justifiées et non justifiées par étudiant
- L'implémentation d'alertes automatiques lorsqu'un étudiant dépasse un certain seuil d'absences
- L'intégration avec d'autres modules Odoo comme la gestion des cours et des emplois du temps

Cette réalisation démontre une maîtrise complète des concepts fondamentaux du développement sur la plateforme Odoo, de la conteneurisation Docker à l'implémentation des modèles MVC, en passant par la création de vues XML et la gestion de la sécurité. Le projet répond aux objectifs fixés et offre une solution fonctionnelle et évolutive pour la gestion des absences des étudiants.