



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de **HONORIS UNITED UNIVERSITIES**

5ÈME ANNÉE INGÉNIERIE INFORMATIQUE ET RÉSEAUX OPTION
MIAGE

RAPPORT DE PROJET ODOO

Gestion d'absence des professeurs

Réalisé par :
Mme SOUDA SABAH

Demandé par :
Pr AIT DAOUD MOHAMMED

2025–2026

Dédicace

Nous dédions ce travail à toutes les personnes qui ont marqué notre parcours académique et personnel :

À nos parents, pour leur amour inconditionnel, leurs sacrifices constants et leur soutien indéfectible. Leur confiance et leurs encouragements ont été la source de notre motivation et de notre persévérance. Sans eux, ce projet n'aurait jamais pu voir le jour.

À nos familles, pour leur présence bienveillante, leurs conseils et leur patience tout au long de cette aventure. Leur appui moral nous a permis de surmonter les moments difficiles et de garder le cap vers nos objectifs.

À nos enseignants, qui ont su éveiller en nous la passion du savoir et de la recherche scientifique. Leur rigueur, leur disponibilité et leur dévouement ont façonné notre esprit critique et notre sens de l'innovation.

À nos camarades et amis, pour leur solidarité, leur partage et leurs encouragements. Les échanges et la collaboration avec eux ont enrichi notre expérience et rendu ce parcours plus agréable.

Enfin, **à toutes les personnes**, de près ou de loin, qui ont contribué à notre réussite et à l'aboutissement de ce projet. Nous leur exprimons notre profonde gratitude et notre reconnaissance éternelle.

Remerciements

Nous tenons à exprimer nos plus sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de fin d'année.

À Professeurs Mohammed AIT DAOUD, nous adressons notre profonde gratitude pour son accompagnement bienveillant, la qualité de ses conseils, sa disponibilité et sa rigueur scientifique. Son expertise et ses orientations pertinentes ont grandement enrichi notre réflexion et ont été décisives dans l'avancement de ce travail. Grâce à son encadrement, nous avons pu mener à bien ce projet dans un cadre méthodologique structuré et formateur.

À l'ensemble du corps professoral de l'EMSI, nous exprimons notre reconnaissance pour la formation de qualité qui nous a été dispensée tout au long de notre parcours académique. Les connaissances et compétences acquises nous ont permis d'aborder ce projet avec assurance et professionnalisme.

Enfin, nous remercions toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce projet et à notre évolution personnelle et professionnelle.

Résumé

Ce projet est un mini-module développé sur Odoo permettant de gérer les absences des professeurs. Il propose une interface simple avec un tableau de bord affichant deux boutons (Espace Professeur et Espace Responsable) qui redirigent vers les vues correspondantes. Les professeurs peuvent déclarer leurs absences, tandis que les responsables accèdent à l'ensemble des absences.

Le projet repose sur l'utilisation d'Odoo pour la gestion des modules et des vues, de Python pour la logique métier, et de Docker pour la conteneurisation et le déploiement dans un environnement isolé et reproductible

Table des matières

1	Introduction Générale	6
2	Spécification et Analyse des Besoins	7
3	Architecture Technique	9
3.1	Environnement de développement	9
3.2	Structure de projet	11
4	Réalisation et Démonstration	16
4.1	Étapes de démarrage de l'application Odoo	16
4.2	Présentation des interfaces du module	18
5	Conclusion Générale	22

Table des figures

3.1	Logo de la plateforme Odoo	9
3.2	Logo du langage Python	9
3.3	Logo de Docker	10
3.4	Logo de PostgreSQL	10
3.5	Logo de GIT	11
3.6	Logo de Visual Studio Code	11
3.7	Structure de projet	12
3.8	Fichier manifest	13
3.9	Fichier init	13
3.10	Fichier models/init	13
3.11	Fichier ir.model.access	14
3.12	Fichier de Configuration	14
3.13	Fichier docker _c ompose.yml	15
4.1	État des conteneurs avec git	16
4.2	État des conteneurs avec Docker Desktop	17
4.3	Page d'accueil de l'application	17
4.4	Activer le mode développeur	17
4.5	Mettre à jour la liste des modules	18
4.6	Interface du tableau de bord principal	18
4.7	Interface de déclaration d'une absence	19
4.8	Interface de l'espace responsable avec statistiques et registre	20
4.9	Interface de gestion des professeurs	20
4.10	Interface de création d'un professeur	21
4.11	Fenêtre de validation d'une absence	21

1 Introduction Générale

Dans un contexte où la digitalisation des processus devient incontournable, les systèmes de gestion intégrés (ERP) jouent un rôle essentiel dans l'organisation et l'optimisation des activités. Odoo, en tant que plateforme open-source, offre une flexibilité remarquable pour développer des modules personnalisés adaptés aux besoins spécifiques des entreprises et des institutions.

Le projet intitulé « **Gestion des Absences des Professeurs** » s'inscrit dans cette logique. Il s'agit d'un *mini-projet académique* visant à concevoir un module simple permettant de gérer les absences des enseignants au sein d'un établissement. L'objectif n'est pas de créer une application complète avec des règles de sécurité complexes, mais plutôt de mettre en pratique les notions fondamentales de développement sur Odoo, en intégrant des vues, des menus et un tableau de bord interactif.

Ce projet repose sur trois piliers technologiques :

- **Odoo** : utilisé comme framework ERP pour la création du module, la définition des menus, des vues (liste, formulaire, tableau de bord) et l'intégration dans l'interface principale.
- **Python** : langage de programmation servant à définir la logique métier, notamment les méthodes associées aux boutons du tableau de bord qui permettent de naviguer entre les différentes vues.
- **Docker** : outil de conteneurisation permettant de déployer le projet dans un environnement isolé, reproductible et portable, garantissant une installation simple et une compatibilité entre différentes machines.

À travers ce projet, on va explorer les concepts clés du développement ERP : la structuration d'un module, la gestion des vues et des actions, ainsi que l'intégration dans l'interface utilisateur d'Odoo. L'approche choisie met l'accent sur la simplicité et la clarté, afin de fournir une solution fonctionnelle et pédagogique, adaptée à un cadre académique.

2 Spécification et Analyse des Besoins

Problématique

La gestion des absences des professeurs dans un établissement scolaire ou universitaire est souvent réalisée de manière manuelle, à travers des feuilles de présence ou des fichiers Excel. Cette approche entraîne plusieurs difficultés :

- Manque de centralisation des données, ce qui complique le suivi global.
- Risques d'erreurs liés à la saisie manuelle et à la duplication des informations.
- Absence d'une interface intuitive permettant aux professeurs de déclarer leurs absences et aux responsables de les consulter facilement.
- Difficulté à générer des rapports ou des statistiques sur les absences.

Solution proposée

Afin de répondre à ces problématiques, nous avons développé un **mini-module Odoo** intitulé *Gestion des Absences des Professeurs*. Ce module offre :

- Un **tableau de bord interactif** avec deux boutons : *Espace Professeur* et *Espace Responsable*.
- Une **vue liste et formulaire** pour enregistrer et consulter les absences.
- Une intégration directe dans le **header principal d'Odoo**, permettant un accès rapide et centralisé.
- Une architecture simple, sans gestion complexe des rôles et de la sécurité, adaptée à un projet académique.

Besoins fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités attendues du système :

- Permettre aux professeurs de déclarer leurs absences via une interface simple.
- Permettre aux responsables de consulter l'ensemble des absences.
- Fournir un tableau de bord avec des boutons de navigation clairs.
- Afficher les absences sous forme de liste et de formulaire.
- Intégrer le module dans le menu principal d'Odoo.

Besoins non fonctionnels

Les besoins non fonctionnels concernent la qualité et les contraintes du système :

- **Simplicité** : interface intuitive et facile à utiliser.
- **Portabilité** : déploiement du projet via Docker pour assurer un environnement reproducible.

- **Performance** : temps de réponse rapide pour l'affichage des vues.
- **Fiabilité** : éviter les erreurs de saisie et garantir la cohérence des données.
- **Maintenabilité** : code structuré et modulable pour faciliter les évolutions futures.

3 Architecture Technique

3.1 Environnement de développement

Le projet **Gestion des Absences des Professeurs** a été développé dans un environnement technique moderne, combinant plusieurs outils complémentaires. Chaque outil joue un rôle spécifique dans la conception, le déploiement et la documentation du module.

Odoo

Odoo est la plateforme ERP open-source utilisée pour développer le module. Elle permet de gérer les menus, les vues (liste, formulaire, tableau de bord) et l'intégration dans l'interface principale.



FIGURE 3.1 – Logo de la plateforme Odoo

Python

Python est le langage de programmation utilisé pour définir la logique métier du module, notamment les méthodes associées aux boutons du tableau de bord et la structure des modèles.



FIGURE 3.2 – Logo du langage Python

Docker

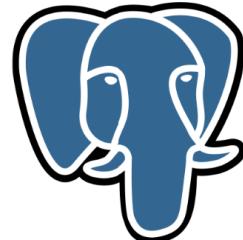
Docker est l'outil de conteneurisation utilisé pour déployer le projet dans un environnement isolé et reproductible. Il facilite la gestion des dépendances (PostgreSQL, Odoo, Python) et assure une compatibilité entre différentes machines.



FIGURE 3.3 – Logo de Docker

PostgreSQL

PostgreSQL est le système de gestion de base de données relationnelle utilisé par Odoo pour stocker les informations relatives aux absences.



PostgreSQL

FIGURE 3.4 – Logo de PostgreSQL

GIT

Git est utilisé comme système de gestion de versions afin d'assurer le suivi des modifications apportées au code source. Il permet de conserver un historique détaillé des évolutions du projet, de revenir à des versions antérieures en cas de besoin et de travailler simultanément sur différentes fonctionnalités sans interférences.



FIGURE 3.5 – Logo de GIT

Visual Studio Code (VS Code)

Visual Studio Code est l'environnement de développement intégré (IDE) utilisé pour écrire, organiser et tester le code du module. Il offre une interface moderne et extensible, avec des fonctionnalités telles que :

- La coloration syntaxique et l'autocomplétion pour Python et XML.
- L'intégration avec Git pour le suivi des versions.
- Les extensions dédiées à Odoo et Docker pour faciliter le développement.
- Un terminal intégré permettant de lancer directement les commandes Docker ou Odoo.



FIGURE 3.6 – Logo de Visual Studio Code

Cet environnement de travail permet de combiner la puissance d'un ERP open-source, la flexibilité du langage Python et la portabilité offerte par Docker, tout en assurant une documentation claire et structurée grâce à LaTeX.

3.2 Structure de projet

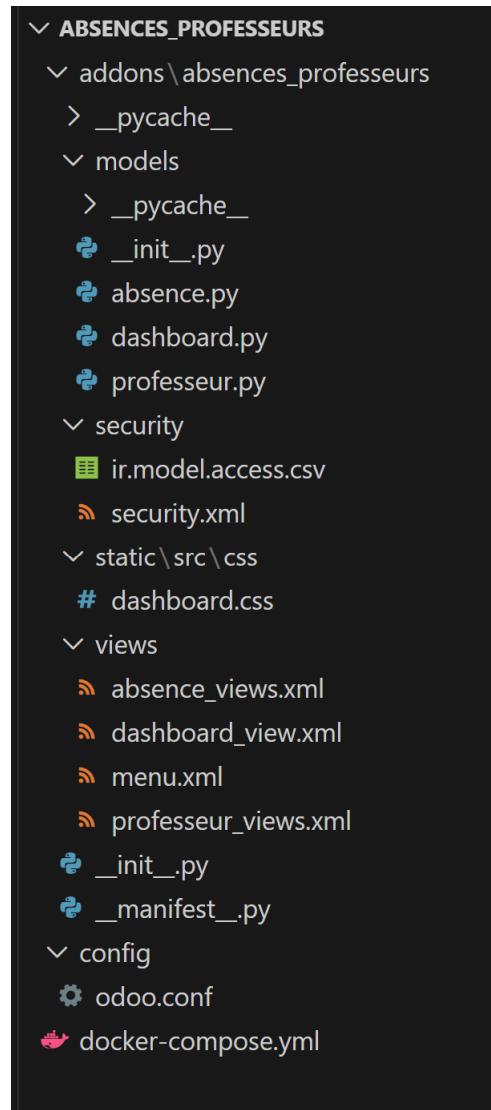


FIGURE 3.7 – Structure de projet

Le module **absences_professeurs** est organisé selon la structure standard d'un module Odoo. Voici une description des principaux fichiers et dossiers :

- **__manifest__.py** : Fichier principal du module, contenant les métadonnées (nom, version, dépendances, fichiers à charger).

```

addons > absences_professeurs > __manifest__.py
1
2  {
3      'name': 'Gestion des Absences des Professeurs',
4      'version': '1.0',
5      'summary': 'Module simple pour gérer les absences des professeurs',
6      'category': 'Education',
7      'author': 'EMSI',
8      'depends': ['base'],
9      'application': True,
10     'data': [
11         'security/ir.model.access.csv',
12         'views/professeur_views.xml',
13         'views/absence_views.xml',
14         'views/dashboard_view.xml',
15         'views/menu.xml',
16     ],
17     'assets': {
18         'web.assets_backend': [
19             'absences_professeurs/static/src/css/dashboard.css',
20         ],
21     },
22
23     'installable': True,
24
25 }
26

```

FIGURE 3.8 – Fichier manifest

- `__init__.py` : Initialise le module Python et importe les sous-modules nécessaires.

```

addons > absences_professeurs > __init__.py
1   from . import models

```

FIGURE 3.9 – Fichier init

- `models/` : Dossier contenant la logique métier du module.
- `absence.py` : Définit le modèle des absences (champs, comportements).
- `professeur.py` : Définit le modèle des professeurs.
- `dashboard.py` : Contient les méthodes du tableau de bord (actions des boutons).
- `__init__.py` : Importe les fichiers Python du dossier `models`.

```

addons > absences_professeurs > models > __init__.py
1   from . import absence
2   from . import dashboard
3   from . import professeur
4

```

FIGURE 3.10 – Fichier models/init

- **views/** : Dossier contenant les définitions des vues XML.
 - **absence_views.xml** : Définit les vues liste et formulaire pour les absences.
 - **professeur_views.xml** : Définit les vues pour les professeurs.
 - **dashboard_view.xml** : Définit le tableau de bord avec les boutons de navigation.
 - **menu.xml** : Définit les menus et sous-menus du module.
- **security/** : Dossier contenant les règles d'accès (peut être supprimé si non utilisé).
 - **security.xml** : Définit les groupes et règles de sécurité.
 - **ir.model.access.csv** : Fichier CSV définissant les droits d'accès aux modèles.

```

ir.model.access.csv ×
addons > absences_professeurs > security > ir.model.access.csv
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_absence_professeur,absence.professeur,model_absence_professeur,base.group_user,1,1,1,1
3 access_absence_dashboard,absence.dashboard,model_absence_dashboard,base.group_user,1,1,1,1
4 access_absence_professeur_master,absence.professeur.master,model_absence_professeur_master,base.group_user,1,1,1,1
5

```

FIGURE 3.11 – Fichier ir.model.access

- **static/src/css/** : Dossier contenant les fichiers CSS pour le style.
 - **dashboard.css** : Personnalisation visuelle du tableau de bord.
- **config/** : Dossier de configuration du projet.
 - **odoo.conf** : Fichier de configuration d'Odoo (base de données, ports, addons path).

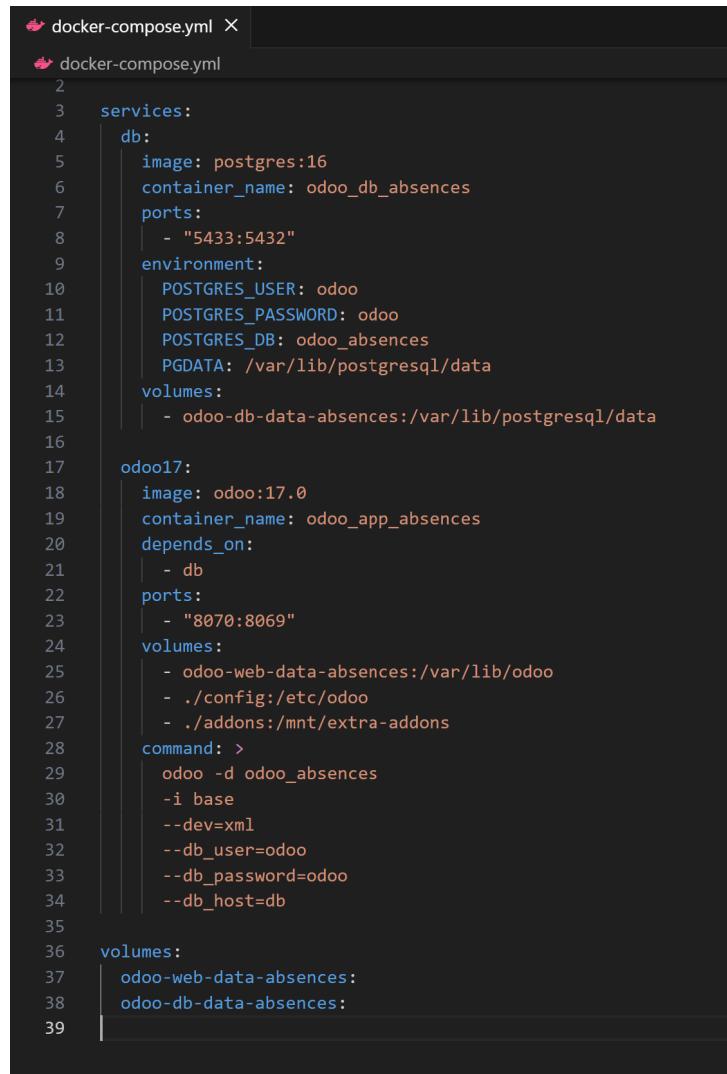
```

odoo.conf ×
config > odoo.conf
1 [options]
2 addons_path = /mnt/extra-addons
3 data_dir = /var/lib/odoo

```

FIGURE 3.12 – Fichier de Configuration

- **docker-compose.yml** : Fichier de conteneurisation pour lancer Odoo, PostgreSQL, etc.



```
 1  docker-compose.yml X
 2
 3  docker-compose.yml
 4
 5  services:
 6    db:
 7      image: postgres:16
 8      container_name: odoो_db_absences
 9      ports:
10        - "5433:5432"
11      environment:
12        POSTGRES_USER: odoो
13        POSTGRES_PASSWORD: odoō
14        POSTGRES_DB: odoō_absences
15        PGDATA: /var/lib/postgresql/data
16      volumes:
17        - odoō-db-data-absences:/var/lib/postgresql/data
18
19    odoō17:
20      image: odoō:17.0
21      container_name: odoō_app_absences
22      depends_on:
23        - db
24      ports:
25        - "8070:8069"
26      volumes:
27        - odoō-web-data-absences:/var/lib/odoō
28        - ./config:/etc/odoō
29        - ./addons:/mnt/extrā-addons
30      command: >
31        odoō -d odoō_absences
32        -i base
33        --dev=xml
34        --db_user=odoō
35        --db_password=odoō
36        --db_host=db
37      volumes:
38        odoō-web-data-absences:
39        odoō-db-data-absences:
```

FIGURE 3.13 – Fichier *dockercompose.yml*

- **__pycache__** : Dossiers générés automatiquement par Python pour la mise en cache des modules.

4 Réalisation et Démonstration

4.1 Étapes de démarrage de l'application Odoo

Le projet est conteneurisé à l'aide de Docker afin de faciliter son déploiement et garantir un environnement reproductible. Voici les étapes nécessaires pour démarrer l'application Odoo et accéder au module *Gestion des Absences des Professeurs* :

1. Cloner le projet ou accéder au dossier du module

Se placer dans le répertoire contenant le fichier `docker-compose.yml` et le dossier `addons/absences_professeurs`.

2. Lancer les conteneurs Docker

Exécuter la commande suivante dans le terminal :

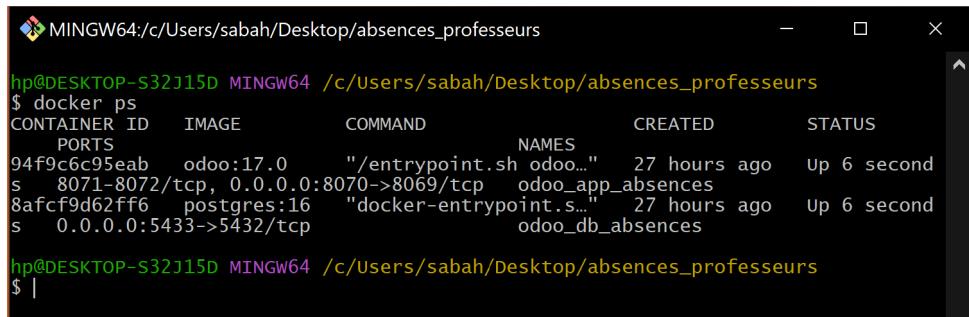
```
docker restart odoo_absences
```

Cette commande démarre les services Odoo et PostgreSQL en arrière-plan.

3. Vérifier si les conteneurs tournent

Exécuter la commande suivante dans le terminal :

```
docker ps
```



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
94f9c6c95eab	odoo:17.0	"/entrypoint.sh odoo..."	27 hours ago	up 6 second
s 8071-8072/tcp, 0.0.0.0:8070->8069/tcp	odoo_app_absences			
8afcf9d62ff6	postgres:16	"docker-entrypoint.s..."	27 hours ago	up 6 second
s 0.0.0.0:5433->5432/tcp	odoo_db_absences			

FIGURE 4.1 – État des conteneurs avec git

Ou sur Docker Desktop :

4. Accéder à l'interface Odoo

Une fois les conteneurs lancés, ouvrir un navigateur et accéder à l'URL suivante :

<http://localhost:8069>

Cette interface permet de créer une base de données et de se connecter à Odoo.

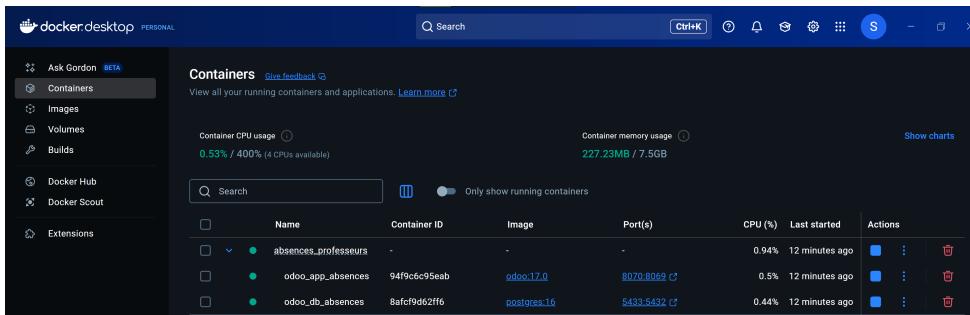


FIGURE 4.2 – État des conteneurs avec Docker Desktop

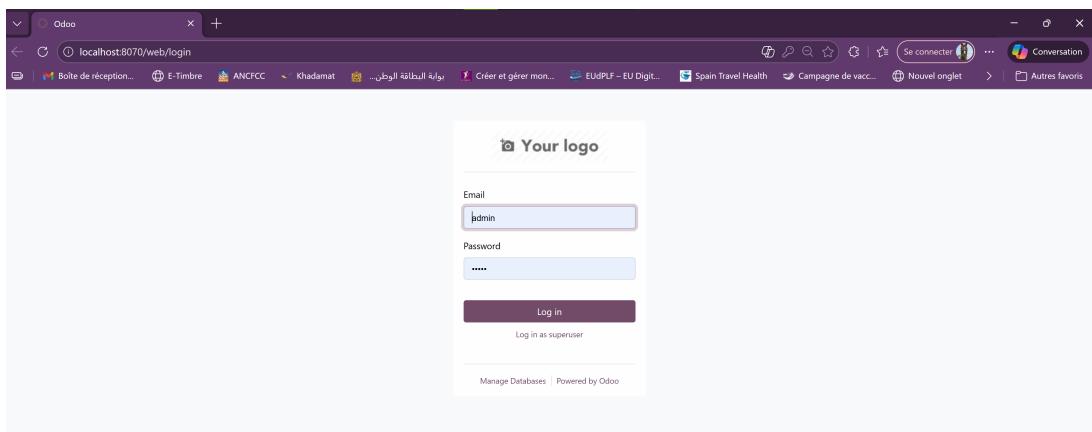


FIGURE 4.3 – Page d'accueil de l'application

5. Activer le mode développeur

Permet d'accéder aux fonctionnalités avancées d'Odoo, telles que la visualisation des identifiants techniques, la modification des vues et des menus, ainsi que le débogage du module. Cette option se trouve dans le menu Apps, cliquer sur settings puis sur Activer le mode développeur.

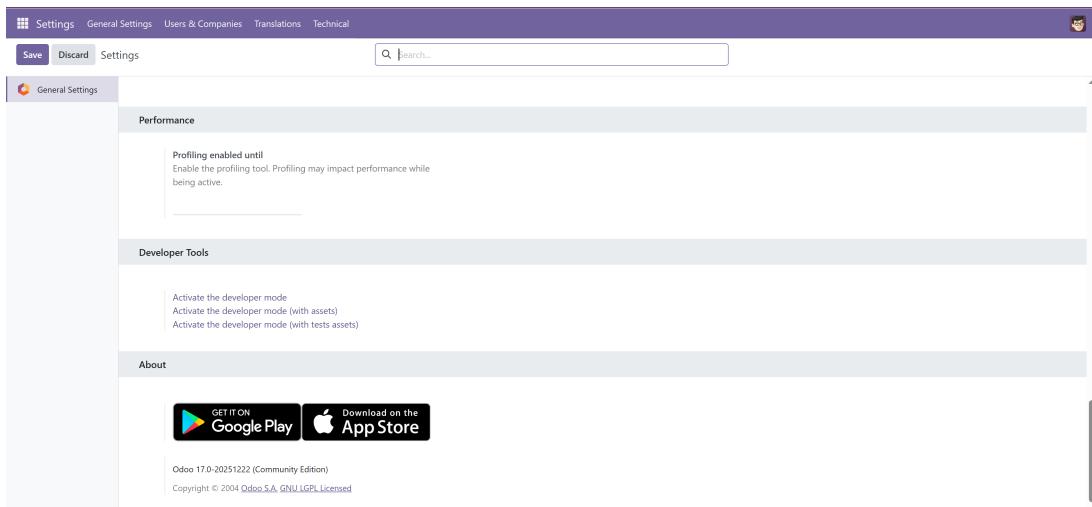


FIGURE 4.4 – Activer le mode développeur

6. Mettre à jour la liste des modules

Dans le menu Apps, cliquer sur Update Apps List pour que le module apparaisse dans la liste des applications disponibles.

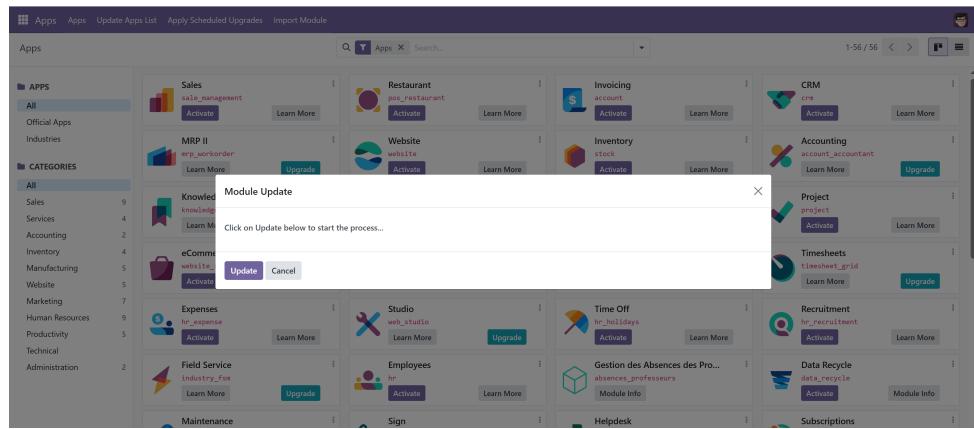


FIGURE 4.5 – Mettre à jour la liste des modules

7. Installer le module

Rechercher le module **Gestion des Absences des Professeurs**, puis cliquer sur **Install**.

8. Accéder au module

Une fois installé, le module apparaît dans le **menu principal (header)** sous le nom **Absences**. En cliquant dessus, l'utilisateur accède au tableau de bord avec les boutons **Espace Professeur** et **Espace Responsable**.

Ces étapes permettent de démarrer rapidement l'application Odoo dans un environnement isolé, tout en garantissant une installation simple et reproductible grâce à Docker.

4.2 Présentation des interfaces du module

Le module **Gestion des Absences des Professeurs** propose une série d'interfaces ergonomiques et intuitives, conçues pour faciliter la déclaration, la validation et le suivi des absences dans un cadre académique.

Tableau de bord principal

Le tableau de bord d'accueil est divisé en deux espaces :

- **Mon Espace (Professeur)** : permet aux enseignants de déclarer une absence rapidement via un bouton dédié.
- **Espace Responsable** : offre aux responsables pédagogiques un accès aux statistiques globales et à la gestion des demandes.



FIGURE 4.6 – Interface du tableau de bord principal

Déclaration d'absence

Cette vue permet de consulter les détails d'une absence déclarée :

- Nom du professeur
- Matière concernée
- Date d'absence
- Motif (ex. : maladie)
- État de la demande (en attente, validée, refusée)

Un bouton permet de confirmer l'absence ou de la refuser.

The screenshot shows a web-based application interface for managing absences. At the top, there's a purple header bar with the text "Gestion des Absences" and "Accueil". Below this is a white header bar with a "New" button, a dropdown menu showing "absence.dashboard,103", a gear icon, and a refresh icon. A blue bar below the white one contains the text "Confirmer l'Absence". The main content area has a light gray background and is titled "Déclaration d'Absence". It lists several pieces of information in a table-like format:

Professeur	SOUDA
Matière	AI
Date d'absence	01/01/2026
Motif	Maladie
État	En attente

FIGURE 4.7 – Interface de déclaration d'une absence

Espace Responsable

Cette interface affiche un résumé statistique :

- **Total des absences** : nombre global
- **En attente** : absences non encore traitées
- **Validées** : absences approuvées
- **Refusées** : absences rejetées

Un registre des absences est également affiché sous forme de tableau, avec les colonnes : Professeur, Matière, Date, Motif, État.

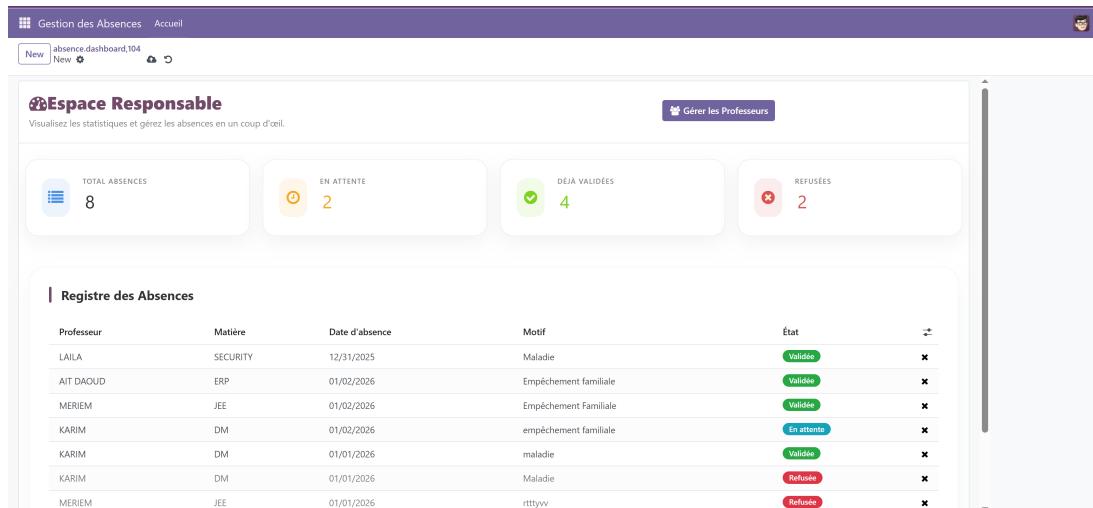


FIGURE 4.8 – Interface de l'espace responsable avec statistiques et registre

Gestion des Professeurs

Cette vue permet de visualiser la liste des enseignants, leur matière respective et le nombre total d'absences. Un bouton permet d'ajouter une ligne pour enregistrer un nouveau professeur.

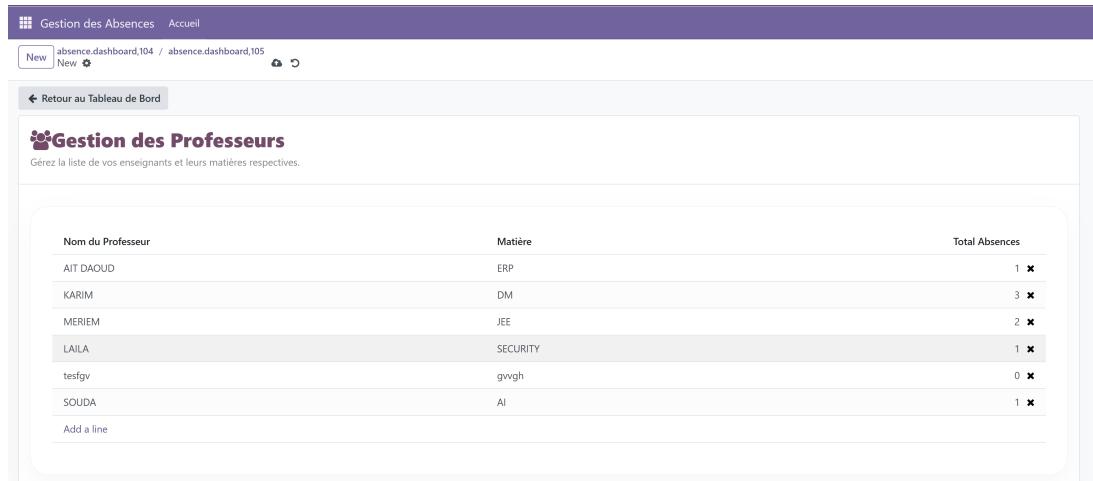


FIGURE 4.9 – Interface de gestion des professeurs

Fiche Professeur

L'interface de création d'un professeur permet de saisir son nom, sa matière. Une section dédiée affiche le nombre total d'absences enregistrées, avec des boutons pour enregistrer ou annuler la saisie.

Create Professeurs Masters

Nom du Professeur	SOUDA	Matière	AI		
Absences	Professeur	Matière	Date d'a...	Motif	État

Nombre d'absences 0

Save & Close **Save & New** **Discard**

FIGURE 4.10 – Interface de création d'un professeur

Validation d'une absence

Lorsqu'un responsable consulte une absence, une fenêtre s'ouvre avec les détails et les boutons **Valider** ou **Refuser**. Cette interface permet un traitement rapide et clair des demandes.

Gestion des Absences Accueil

New absence.dashboard,104

TOTAL ABSENCES	8	EN ATTENTE	2	DÉJÀ VALIDÉES	4	REFUSÉES	2
Registre des Absences							
Professeur	Matière						
LAILA	SECUR						
AIT DAOUD	ERP						
MERIEM	JEE						
KARIM	DM						
KARIM	DM						
KARIM	DM						
MERIEM	JEE						
SOUDA	AI						
Add a line							

Open: Absences

Valider **Refuser**

Créé par: Mitchell Admin

En attente Validée Refusée

Professeur: SOUDA
Matière: AI
Date d'absence: 01/01/2026

MOTIF DE L'ABSENCE

Maladie

Save **Discard**

Maladie **Refuser** **En attente**

FIGURE 4.11 – Fenêtre de validation d'une absence

Ces interfaces ont été conçues pour offrir une navigation fluide, une séparation claire des rôles, et une gestion simplifiée des absences dans un environnement éducatif.

5 Conclusion Générale

Le projet **Gestion des Absences des Professeurs** a permis de mettre en pratique les concepts fondamentaux du développement sur Odoo, en intégrant des modèles, des vues et un tableau de bord interactif. Bien qu'il s'agisse d'un mini-projet académique, il illustre clairement la puissance et la flexibilité offertes par les ERP open-source pour répondre à des besoins spécifiques.

L'utilisation combinée de **Python**, **Odoo** et **Docker** a permis de construire une solution simple, reproductible et portable. Python a assuré la logique métier, Odoo a fourni le cadre ERP pour la gestion des absences, et Docker a facilité le déploiement dans un environnement isolé. De plus, la rédaction du rapport en **LaTeX** a garanti une présentation professionnelle et structurée.

Ce projet démontre l'importance de la modularité et de la conteneurisation dans le développement logiciel moderne. Il constitue une base solide pour des évolutions futures, telles que l'ajout de fonctionnalités avancées (statistiques, notifications, gestion des rôles), tout en restant un excellent exercice académique pour comprendre l'intégration des outils technologiques dans un cadre pratique.