

Rapport de Projet Odoo

*5ème Année en Ingénierie Informatique et
Réseaux- Groupe 9*

Création d'un module Odoo Gestion des tâches simples (sans workflow)

Réalisé par :
ILLOUL Boutaina

Encadré par :
Pr. Mohammed AITDAOUD

Remerciements

Nous tenons à exprimer nos sincères remerciements à **Mr. Mohammed AITDAOUD**, enseignant du module ERP, pour son encadrement, sa disponibilité et ses conseils précieux tout au long de la réalisation de ce projet. Ses orientations pédagogiques et ses explications claires nous ont permis de mieux comprendre le fonctionnement des ERP, en particulier Odoo, et d'acquérir des compétences pratiques en développement de modules personnalisés.

Résumé

Ce projet vise le développement d'un module Odoo dédié à la gestion des tâches simples, afin de répondre au besoin d'organisation et de suivi des activités au sein d'un environnement professionnel ou académique. Le module permet la création, la consultation, la modification et la suppression des tâches, tout en offrant une interface claire et intuitive pour les utilisateurs. Chaque tâche est caractérisée par des informations essentielles telles que le nom, le responsable, les dates et la priorité.

Le module a été conçu sans workflow, conformément aux consignes pédagogiques, afin de privilégier la simplicité et la compréhension des bases du développement de modules Odoo. Il respecte l'architecture standard d'Odoo et intègre les mécanismes fondamentaux de sécurité pour la gestion des droits d'accès.

La solution repose sur Odoo 17, avec Python pour la logique métier, XML pour la définition des interfaces utilisateur et PostgreSQL comme système de gestion de base de données. L'utilisation de Docker permet de garantir un environnement d'installation et d'exécution stable, fiable et reproductible.

Le module développé est fonctionnel, évolutif et conforme aux exigences du module ERP. Il constitue une base solide pour l'ajout futur de fonctionnalités avancées telles que des workflows, des notifications ou des tableaux de bord de suivi.

Mots-clés

Gestion des tâches, module Odoo 17, ERP, automatisation simple, tâches sans workflow, Python, XML, PostgreSQL, Docker, interface utilisateur intuitive, module évolutif.

Abstract

This project focuses on the development of a custom Odoo module dedicated to simple task management, designed to address the need for efficient organization and monitoring of activities in academic or professional environments. The module enables users to create, view, update, and delete tasks, while providing a clear and intuitive user interface. Each task is defined by essential attributes such as task name, responsible user, dates, and priority.

The module was intentionally developed without workflow implementation, in accordance with the pedagogical requirements, in order to emphasize simplicity and facilitate understanding of the fundamental principles of Odoo module development. It strictly follows the standard Odoo architecture and includes basic security mechanisms for access control.

The solution is built on Odoo 17, using Python for business logic, XML for user interface definition, and PostgreSQL as the database management system. The use of Docker ensures a reliable, reproducible, and stable development and deployment environment.

The resulting module is functional, scalable, and aligned with the objectives of the ERP course. It provides a solid foundation for future enhancements such as workflow integration, notifications, and analytical dashboards

Keywords

Task management, Odoo 17 module, ERP, simple task tracking, workflow-free module, Python, XML, PostgreSQL, Docker, intuitive user interface, scalable solution.

Table des matières

Remerciements.....	
Résumé	
Abstract	
Liste des Figures	
INTRODUCTION GÉNÉRALE	1
Chapitre 1 : Présentation d’Odoo et de l’environnement technique	3
1.1 Contexte du projet.....	3
1.2 Problématique	3
1.3 Solution proposée	3
1.4. Objectifs du TP.....	3
1.5 Environnement technique.....	3
Chapitre 2 : Installation et Réalisation du module	4
2.1. Structure du module	4
2.2 Création de l’environnement Docker	4
2.3 Modèle Principal :.....	5
2.4 Développement du modèle Python	5
2.5 Sécurité et droits d'accès.....	5
2.6 Étapes d'Installation	6
2.7 Configuration Initiale.....	6
2.8 Interfaces utilisateur :.....	8
Conclusion Générale.....	10
Références	12

Liste des Figures

Figure 1:Arborescence du module gestion_taches	4
Figure 2:Configuration Docker pour Odoo et PostgreSQL.....	4
Figure 3: Modèle Python de la tâche	5
Figure 4: Sécurité et droits d'accès.....	5
Figure 5: Page d'authentification	6
Figure 6: mode développeur	7
Figure 7: Update du module.....	7
Figure 8: Activation du Module	7
Figure 9: Vue liste des tâches	8
Figure 10: Formulaire de création d'une tâche	8
Figure 11: Test du module	9
Figure 12: Résultat des tests fonctionnels du module	9

INTRODUCTION GÉNÉRALE

À l'ère de la transformation digitale, les organisations, qu'elles soient académiques ou professionnelles, sont confrontées à une augmentation constante des activités à planifier, suivre et coordonner. La multiplication des tâches quotidiennes, la répartition des responsabilités entre plusieurs acteurs et la nécessité de respecter des délais précis rendent indispensable l'utilisation d'outils numériques performants pour assurer une gestion efficace des activités.

Dans de nombreux contextes, la gestion des tâches repose encore sur des méthodes traditionnelles telles que les notes manuelles, les tableurs ou les échanges informels par courriel. Ces pratiques montrent rapidement leurs limites, notamment en termes de centralisation de l'information, de traçabilité, de fiabilité des données et de suivi des responsabilités. Elles peuvent entraîner des oublis, des incohérences et une perte de temps significative, tout en rendant difficile l'obtention d'une vision globale et structurée des tâches à réaliser.

Face à ces constats, ce projet s'inscrit dans une démarche de modernisation et de simplification de la gestion des tâches à travers le développement d'un module ERP dédié. Il vise la conception d'un module Odoo de gestion des tâches simples, permettant de centraliser l'ensemble des informations relatives aux tâches, de faciliter leur création, leur consultation, leur modification et leur suppression, tout en offrant une interface claire et intuitive aux utilisateurs. L'objectif principal est de proposer une solution intégrée, légère et efficace, adaptée à des besoins simples sans recourir à des mécanismes complexes tels que les workflows.

Sur le plan technique, le projet repose sur le framework Odoo 17, en s'appuyant sur une architecture respectant les standards de développement de la plateforme. Il combine le langage Python pour la logique métier, PostgreSQL pour la gestion des données et Docker pour garantir un environnement de déploiement stable, reproductible et maîtrisé. Cette approche permet non seulement de faciliter le développement et les tests, mais également de préparer une éventuelle évolution future du module.

Ce rapport présente l'ensemble des étapes du projet, depuis l'analyse des besoins jusqu'à la validation finale de la solution développée. Il met en évidence la méthodologie adoptée, les choix techniques effectués, les fonctionnalités implémentées ainsi que les résultats obtenus, tout en ouvrant des perspectives d'amélioration et d'enrichissement du module.

Chapitre 1 : Présentation d’Odoo et de l’environnement technique

1.1 Contexte du projet

Ce projet est réalisé dans le cadre du module ERP et vise à appliquer les concepts étudiés à travers le développement d'un module Odoo personnalisé.

1.2 Problématique

- Difficulté de suivi des tâches
- Manque de centralisation
- Absence de visibilité globale

1.3 Solution proposée

Développement d'un module Odoo **Gestion des tâches simples**, sans workflow, offrant une interface claire et des fonctionnalités CRUD complètes.

1.4. Objectifs du TP

Objectifs pédagogiques :

- Comprendre l'architecture d'Odoo
- Manipuler l'ORM
- Créer des vues XML
- Gérer la sécurité

Objectifs fonctionnels :

- Créer et gérer des tâches
- Affecter un responsable
- Gérer les priorités et les dates

1.5 Environnement technique

Le développement du module a été réalisé à l'aide des technologies suivantes :

- **Odoo 16/17**
- **Python** pour la logique métier
- **XML** pour la définition des vues
- **PostgreSQL** comme système de gestion de base de données
- **Docker & Docker Compose** pour la conteneurisation
- Tester et valider le bon fonctionnement du module

Chapitre 2 : Installation et Réalisation du module

2.1. Structure du module

Le module *gestion_taches* a été créé dans le dossier *addons* d'Odoo en respectant l'architecture standard recommandée. Cette structure permet une séparation claire entre la logique métier, les interfaces utilisateur et la sécurité.

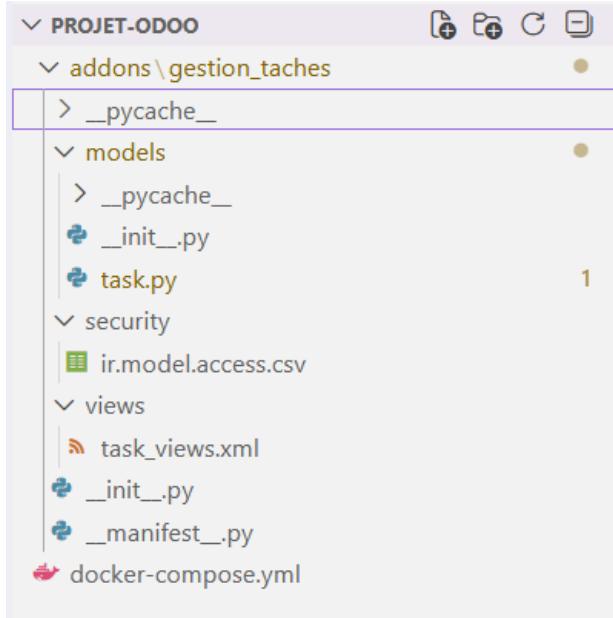


Figure 1:Arborescence du module *gestion_taches*

2.2 Crédation de l'environnement Docker

Afin de faciliter le déploiement et l'exécution d'Odoo, Docker a été utilisé. Le fichier *docker-compose.yml* permet de lancer automatiquement deux conteneurs : Odoo et PostgreSQL.

The image shows a code editor displaying the 'docker-compose.yml' file. The file defines two services: 'db' and 'odoo'. The 'db' service uses the 'postgres:13' image and sets environment variables for POSTGRES_DB, POSTGRES_USER, and POSTGRES_PASSWORD, and maps a volume 'odo-db-data' to '/var/lib/postgresql/data'. The 'odoo' service uses the 'odoo:16' image, depends on the 'db' service, and maps port 8069. It also maps a volume 'odo-db-data' to '/mnt/extras-addons' and sets environment variables for HOST, USER, and PASSWORD. The code is as follows:

```
version: '3.1'

services:
  db:
    image: postgres:13
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=odo
      - POSTGRES_PASSWORD=odo
    volumes:
      - odo-db-data:/var/lib/postgresql/data

  odoo:
    image: odoo:16
    depends_on:
      - db
    ports:
      - "8069:8069"
    volumes:
      - ./addons:/mnt/extras-addons
    environment:
      - HOST=db
      - USER=odo
      - PASSWORD=odo
    volumes:
      - odo-db-data:
```

Figure 2:Configuration Docker pour Odoo et PostgreSQL

2.3 Modèle Principal :

Fichier : gestion_taches/__manifest__.py

```
PROJET-ODOO
├── addons > gestion_taches
│   ├── __init__.py
│   ├── models
│   │   └── task.py
│   ├── security
│   │   └── ir.model.access.csv
│   └── views
│       └── task_views.xml
└── docker-compose.yml

addons > gestion_taches > __manifest__.py
1 1 {
2     'name': 'Gestion des tâches simples',
3     'version': '1.0',
4     'summary': 'Gestion des tâches sans workflow',
5     'description': 'Module Odoo pour gérer des tâches simples',
6     'category': 'Productivité',
7     'author': 'Anssame',
8     'depends': ['base'],
9     'data': [
10         'security/ir.model.access.csv',
11         'views/task_views.xml',
12     ],
13     'application': True,
14 }
15 }
```

Fichier : gestion_taches/__init__.py

```
from . import models
```

Fichier : gestion_taches/models/__init__.py

```
from . import task
```

2.4 Développement du modèle Python

Le modèle Python *gestion.tache* représente l'entité principale du module. Il définit les attributs essentiels d'une tâche tels que le nom, le responsable, les dates et la priorité.

```
from odoo import models, fields

class Task(models.Model):
    _name = 'gestion.tache'
    _description = 'Tâche'

    name = fields.Char(string="Nom de la tâche", required=True)
    description = fields.Text(string="Description")
    date_debut = fields.Date(string="Date de début")
    date_fin = fields.Date(string="Date de fin")
    responsable = fields.Many2one('res.users', string="Responsable")
    priorite = fields.Selection([
        ('basse', 'Basse'),
        ('moyenne', 'Moyenne'),
        ('haute', 'Haute')
    ], string="Priorité", default='moyenne')
```

Figure 3: Modèle Python de la tâche

2.5 Sécurité et droits d'accès

Un fichier *ir.model.access.csv* a été ajouté afin d'autoriser les opérations de création, lecture, modification et suppression des tâches.

```
PROJET-ODOO
├── addons > gestion_taches
│   ├── __init__.py
│   ├── models
│   │   └── task.py
│   └── security
│       └── ir.model.access.csv
└── docker-compose.yml

addons > gestion_taches > security > ir.model.access.csv
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_gestion_tache,access_gestion_tache,model_gestion_tache,,1,1,1,1
3 |
```

Figure 4: Sécurité et droits d'accès

2.6 Étapes d'Installation

Étape 1 :

Démarrer Docker docker-compose up -d

Étape 2 : Créer la base de données

- ✓ Accéder à <http://localhost:8069>
- ✓ Email admin : admin
- ✓ Mot de passe : admin

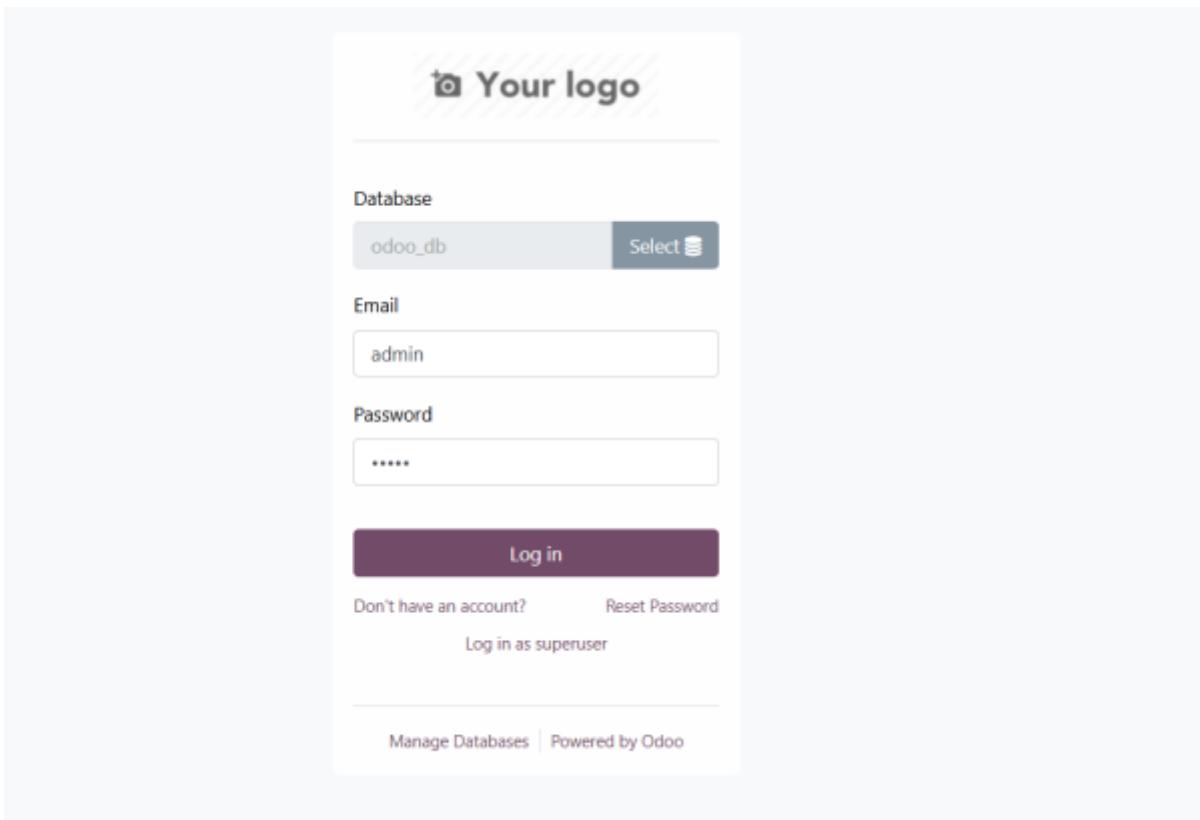


Figure 5: Page d'authentification

2.7 Configuration Initiale

➤ Activation du mode développeur :

1. Paramètres → Activer le mode développeur
2. Ou via URL : <http://localhost:8069/web?debug=1>

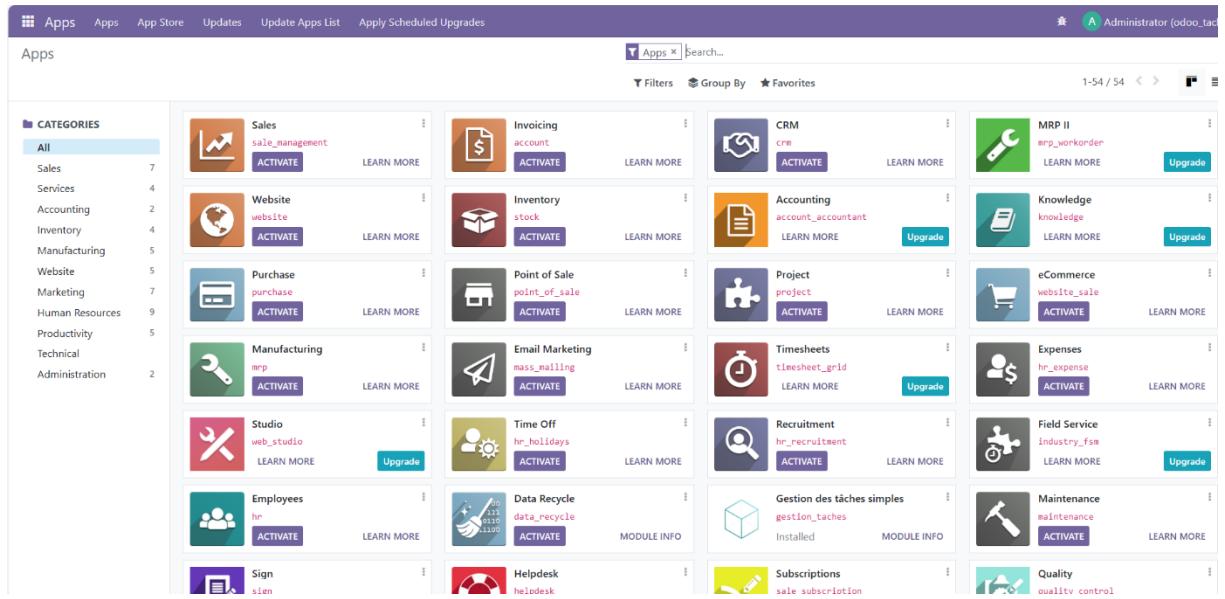


Figure 6: mode développeur

➤ Mettre à jour la liste des applications

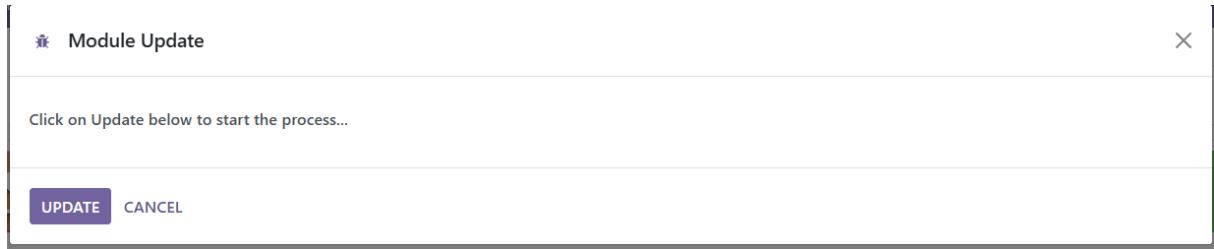


Figure 7: Update du module

➤ Activation du module

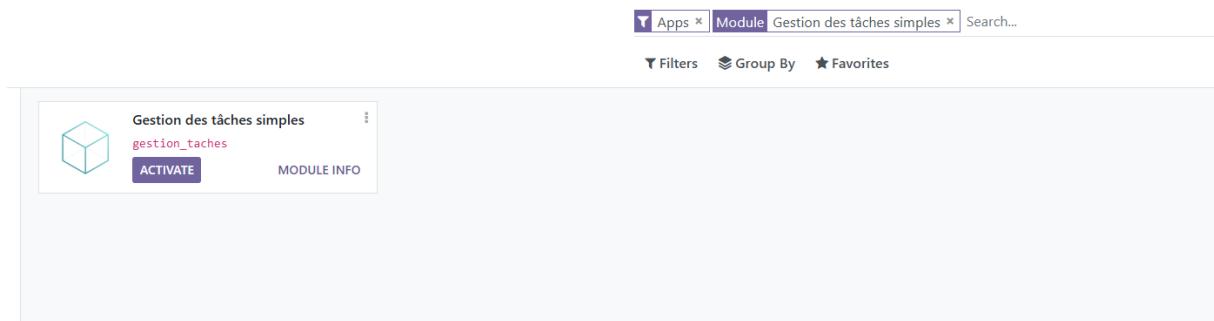


Figure 8: Activation du Module

2.8 Interfaces utilisateur :

➤ Accès au module « Gestion des tâches »

Cette interface représente l'accès principal au module Gestion des tâches simples. Elle affiche la vue liste initiale du module, permettant à l'utilisateur de visualiser l'ensemble des tâches enregistrées dans le système. À partir de cette interface, l'utilisateur peut effectuer des actions de recherche, de filtrage ou créer une nouvelle tâche via le bouton New.

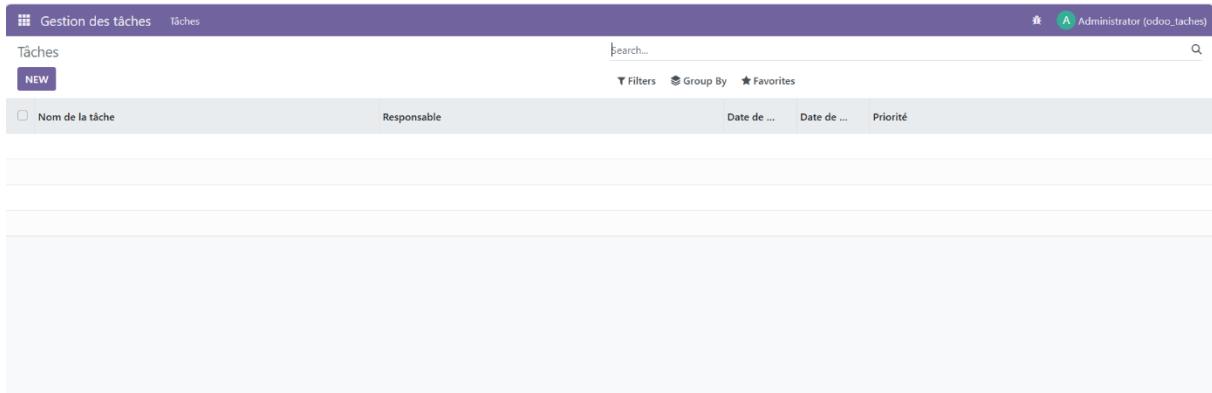


Figure 9: Vue liste des tâches

➤ Formulaire de création d'une nouvelle tâche

Cette interface correspond au formulaire de création d'une nouvelle tâche. Elle permet à l'utilisateur de saisir les informations essentielles telles que le nom de la tâche, le responsable, la priorité, les dates de début et de fin ainsi qu'une description détaillée. Cette vue formulaire assure une saisie structurée et intuitive des données.

A screenshot of the Odoo Task Management module's creation form. The top navigation bar shows 'Gestion des tâches' and 'Tâches', with 'Tâches / New' selected. On the right, there are 'Action' and 'New' buttons. The main area contains fields for 'Nom de la tâche' (with a required indicator), 'Responsable' (with a dropdown arrow), 'Priorité' (set to 'Moyenne'), 'Date de début' (with a calendar icon), 'Date de fin' (with a calendar icon), and 'Description' (with a required indicator). The form is enclosed in a light gray border.

Figure 10: Formulaire de création d'une tâche

➤ Détails d'une tâche créée

Cette capture illustre l'affichage détaillé d'une tâche après sa création. Les informations saisies sont clairement présentées, notamment le nom de la tâche, le responsable, la priorité, les dates et la description. Cette vue permet également la modification ou la suppression de la tâche, confirmant la prise en charge complète des opérations CRUD.

The screenshot shows a task record in the Odoo Task Management module. The task details are as follows:

- Nom de la tâche**: Préparer le rapport du projet Odoo
- Responsable**: Administrator
- Priorité**: Haute
- Date de début**: 12/29/2025
- Date de fin**: 01/04/2026
- Description**: Rédiger le rapport du module Gestion des tâches simples

Figure 11: Test du module

➤ Vue liste après l'ajout d'une tâche

Cette vue liste présente l'ensemble des tâches enregistrées après l'ajout d'une nouvelle tâche. Elle offre une vision globale et synthétique des informations clés telles que le responsable, les dates et la priorité. Cette interface facilite le suivi et la consultation rapide des tâches existantes.

The screenshot shows a list of tasks in the Odoo Task Management module. The table has the following columns:

Nom de la tâche	Responsable	▼ Date de début	Date de fin	Priorité
Préparer le rapport du projet Odoo	Administrator	12/29/2025	01/04/2026	Haute

Figure 12: Résultat des tests fonctionnels du module

Conclusion Générale

Au terme de ce projet, un module Odoo fonctionnel et opérationnel dédié à la gestion des tâches simples a été conçu et développé avec succès. La solution réalisée répond pleinement aux objectifs initiaux en proposant un outil structuré, fiable et facile à utiliser, adapté aux besoins d'organisation et de suivi des activités dans un contexte académique ou professionnel. Elle permet une gestion centralisée des tâches, des responsables et des échéances, tout en offrant une visibilité claire sur l'ensemble des activités.

Sur le plan fonctionnel, le module couvre les opérations essentielles de gestion des tâches, notamment la création, la consultation, la modification et la suppression des enregistrements. Le choix de développer le module sans workflow a permis de privilégier la simplicité, la clarté des fonctionnalités et la compréhension des mécanismes fondamentaux d'Odoo. L'utilisation de vues complémentaires, telles que la vue liste et la vue formulaire, assure une navigation fluide et une expérience utilisateur intuitive. Les tests fonctionnels réalisés ont confirmé la stabilité et la fiabilité du module développé.

Sur le plan technique, le projet respecte l'architecture standard d'Odoo et s'appuie sur une conception claire et maintenable. L'utilisation du langage Python pour la logique métier, de XML pour la définition des interfaces utilisateur et de PostgreSQL pour la gestion des données garantit une solution cohérente et évolutive. De plus, l'intégration de Docker a permis de mettre en place un environnement de développement et de déploiement stable, reproductible et maîtrisé.

D'un point de vue pédagogique, ce projet a constitué une expérience enrichissante, permettant de consolider des compétences techniques en développement de modules Odoo ainsi qu'en conteneurisation avec Docker. Il a également favorisé le développement de compétences méthodologiques essentielles telles que l'analyse des besoins, la structuration d'une solution logicielle et la validation fonctionnelle d'une application ERP.

Le module développé présente un potentiel d'évolution important. Plusieurs améliorations peuvent être envisagées, telles que l'ajout d'un workflow de suivi des tâches, l'intégration de notifications automatiques, l'introduction de tableaux de bord de suivi ou encore l'ajout de vues calendrier et de fonctionnalités avancées de reporting. À plus long terme, l'intégration

de services externes ou de fonctionnalités intelligentes pourrait enrichir davantage cette solution.

En conclusion, ce projet démontre la capacité à concevoir, développer et déployer un module Odoo professionnel répondant à un besoin concret. Il constitue une base solide pour de futures évolutions et illustre efficacement l'application des connaissances théoriques acquises dans le cadre du module ERP à un cas pratique et réel.

Références

[1] Odoo S.A. (2024). Odoo 17.0 Documentation Officielle. Disponible en ligne :

<https://www.odoo.com/documentation/17.0/>

[2] Odoo S.A. (2024). Odoo ORM API Reference. Disponible en ligne :

<https://www.odoo.com/documentation/17.0/developer/reference/backend/orm.html>

[3] Docker Inc. (2024). Docker Documentation. Disponible en ligne :

<https://docs.docker.com/>