



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

*Membre de*

**HONORIS UNITED UNIVERSITIES**

## École Marocaine des Sciences de l'Ingénieur (EMSI)

Matière : ERP (Enterprise Resource Planning)

### Compte rendu de projet

---

## Module de Gestion des Annonces Internes

Implémentation d'un module ERP sous Odoo

---

Réalisé par :

Khalidi Badr

Année universitaire 2024–2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Contexte et problématique</b>	<b>2</b>
<b>3</b>	<b>Objectifs du projet</b>	<b>2</b>
<b>4</b>	<b>Environnement technique</b>	<b>3</b>
<b>5</b>	<b>Architecture générale du projet</b>	<b>3</b>
<b>6</b>	<b>Structure du module</b>	<b>3</b>
<b>7</b>	<b>Modélisation des données</b>	<b>4</b>
<b>8</b>	<b>Interfaces utilisateur</b>	<b>5</b>
8.1	Menus et actions . . . . .	5
8.2	Vue liste (Tree View) . . . . .	5
8.3	Vue formulaire (Form View) . . . . .	5
<b>9</b>	<b>Illustrations et captures du projet</b>	<b>5</b>
9.1	Structure du projet dans l'explorateur . . . . .	5
9.2	Fichier manifeste du module . . . . .	6
9.3	Définition du modèle de données . . . . .	6
9.4	Configuration des vues XML . . . . .	7
9.5	Gestion des permissions d'accès . . . . .	7
9.6	Interface de liste des annonces . . . . .	8
9.7	Formulaire de création d'annonce . . . . .	8
9.8	Paramètres et configuration . . . . .	8
9.9	Page d'information du module . . . . .	9
9.10	Module dans la liste des applications . . . . .	9
<b>10</b>	<b>Résultats obtenus</b>	<b>10</b>
<b>11</b>	<b>Limites du projet</b>	<b>10</b>
<b>12</b>	<b>Perspectives d'amélioration</b>	<b>10</b>
<b>13</b>	<b>Conclusion</b>	<b>11</b>

# 1 Introduction

La communication interne est un élément crucial pour la cohésion et l'efficacité d'une organisation. Les annonces internes permettent de diffuser des informations importantes, des procédures, des événements ou des alertes à l'ensemble des collaborateurs. Cependant, cette diffusion est souvent fragmentée entre différents canaux (emails, affichages physiques, groupes de discussion), ce qui nuit à la visibilité, à la traçabilité et à l'actualisation des informations.

Ce projet vise à développer un module Odoo personnalisé pour centraliser la gestion des annonces internes. En intégrant cette fonctionnalité à l'ERP Odoo, nous offrons une solution structurée, accessible et facile à maintenir pour la publication, le suivi et l'expiration des annonces. Ce projet s'inscrit dans une démarche pédagogique visant à maîtriser le développement de modules Odoo, depuis la modélisation des données jusqu'au déploiement dans un environnement conteneurisé.

## 2 Contexte et problématique

Dans la plupart des organisations, la communication interne repose sur des outils disparates :

- Envoi d'emails groupés, souvent noyés dans les boîtes de réception
- Affichages papier dans les espaces communs, rapidement obsolètes
- Messages sur des plateformes collaboratives sans structuration
- Absence de centralisation des informations importantes

Ces pratiques entraînent plusieurs problèmes :

- Manque de visibilité sur les annonces importantes
- Difficulté à retrouver des informations anciennes
- Absence de gestion des dates de validité
- Pas de traçabilité des publications
- Risque de diffusion d'informations contradictoires

Le module de gestion des annonces internes sous Odoo répond à ces problématiques en offrant une plateforme unique, organisée et accessible à tous les collaborateurs.

## 3 Objectifs du projet

L'objectif principal est de concevoir et développer un module fonctionnel de gestion des annonces internes sous Odoo 17.

**Objectifs spécifiques :**

- Analyser les besoins en communication interne d'une organisation
- Modéliser les données relatives aux annonces (titre, contenu, dates, statut, etc.)

- Implémenter la logique métier en Python selon l'architecture Odoo
- Développer des interfaces utilisateur ergonomiques (vues liste et formulaire)
- Configurer les menus et les droits d'accès
- Déployer le module dans un environnement Dockerisé
- Tester le fonctionnement complet du module

## 4 Environnement technique

Le projet utilise les technologies suivantes :

- **Odo 17** : Plateforme ERP open-source
- **Python 3.10** : Langage pour la logique métier
- **XML** : Définition des vues et interfaces
- **PostgreSQL** : Base de données relationnelle
- **Docker et Docker Compose** : Conteneurisation et déploiement
- **CSV** : Gestion des droits d'accès

L'utilisation de Docker garantit la portabilité et la reproductibilité du projet sur différentes machines.

## 5 Architecture générale du projet

Le module suit l'architecture standard des modules Odoo :

- **Modèles (Python)** : Définition de la structure des données
- **Vues (XML)** : Interface utilisateur et présentation des données
- **Sécurité (CSV)** : Contrôle des accès aux données
- **Manifeste (Python)** : Métadonnées et configuration du module

L'architecture Docker sépare les services :

- Conteneur Odoo : Exécution de l'application
- Conteneur PostgreSQL : Stockage des données

## 6 Structure du module

Le module `gestion_annonce` est organisé comme suit :

```
gestion_annonce/
__init__.py
__manifest__.py
models/
__init__.py
annonce.py
```

```

security/
    ir.model.access.csv
views/
    projet_views.xml
(autres fichiers de configuration)

— __manifest__.py : Déclare les métadonnées, dépendances et fichiers du module
— models/annonce.py : Contient la définition du modèle de données
— views/projet_views.xml : Définit les interfaces utilisateur
— security/ir.model.access.csv : Configure les permissions

```

## 7 Modélisation des données

Le modèle principal `gestion_annonces` est défini dans `models/annonce.py` :

```

1 class Annonce(models.Model):
2     _name = "gestion_annonces"
3     _description = "Annonces internes"
4
5     name = fields.Char(string="Titre", required=True)
6     statut = fields.Selection([
7         ("brouillon", "Brouillon"),
8         ("en_cours", "En cours"),
9         ("terminé", "Terminé"),
10    ], string="Statut", default="brouillon")
11     description = fields.Text(string="Description")
12     prix = fields.Float(string="Prix")
13     date_annonce = fields.Date(string="Date de publication",
14                                 default=fields.Date.today)
15     date_expiration = fields.Date(string="Date d'expiration")
16     user_id = fields.Many2one("res.users", string="Auteur",
17                               default=lambda self: self.env.user)

```

### Explication des champs :

- `name` : Titre de l'annonce (obligatoire)
- `statut` : État de publication (brouillon/en cours/terminé)
- `description` : Contenu détaillé de l'annonce
- `date_annonce` : Date de publication (automatique)
- `date_expiration` : Date de fin de validité
- `user_id` : Auteur de l'annonce (utilisateur connecté)

## 8 Interfaces utilisateur

Les interfaces sont définies dans `views/projet_views.xml` :

### 8.1 Menus et actions

- Menu principal "Annonces" dans la barre de navigation
- Sous-menu "Gestion des Annonces" pointant vers l'action principale
- Action `action_annonce` ouvrant les vues liste et formulaire

### 8.2 Vue liste (Tree View)

Tableau structuré présentant les annonces avec colonnes complètes (Référence, Titre, Auteur, Catégorie, Priorité , Dates, Nombre de vues, Statut).

### 8.3 Vue formulaire (Form View)

Interface détaillée pour la création, modification et consultation des annonces, organisée en sections logiques.

## 9 Illustrations et captures du projet

### 9.1 Structure du projet dans l'explorateur

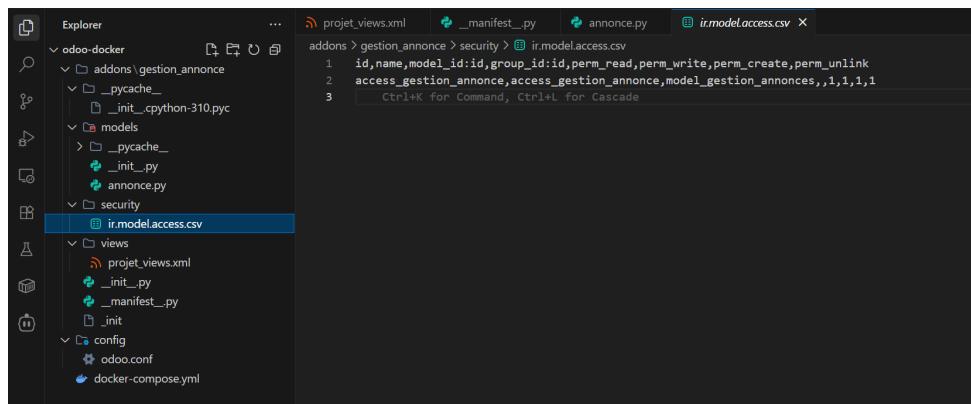
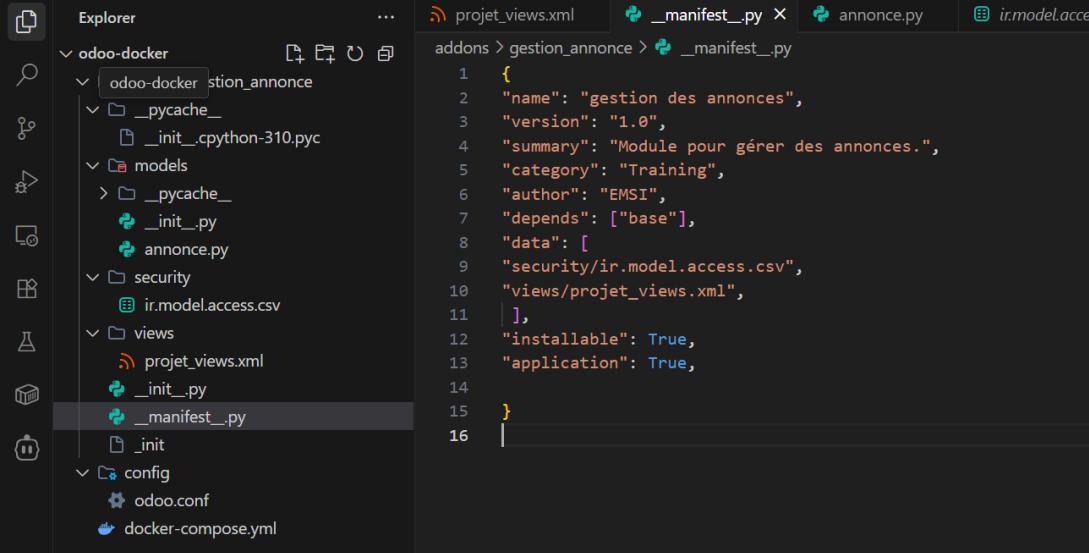


FIGURE 1 – Structure des fichiers du module dans l'explorateur

Cette capture montre l'organisation complète du module `gestion_annonce` avec tous ses répertoires et fichiers essentiels.

## 9.2 Fichier manifeste du module



The screenshot shows the Odoo module structure in the left sidebar and the content of the `__manifest__.py` file in the main editor area.

```

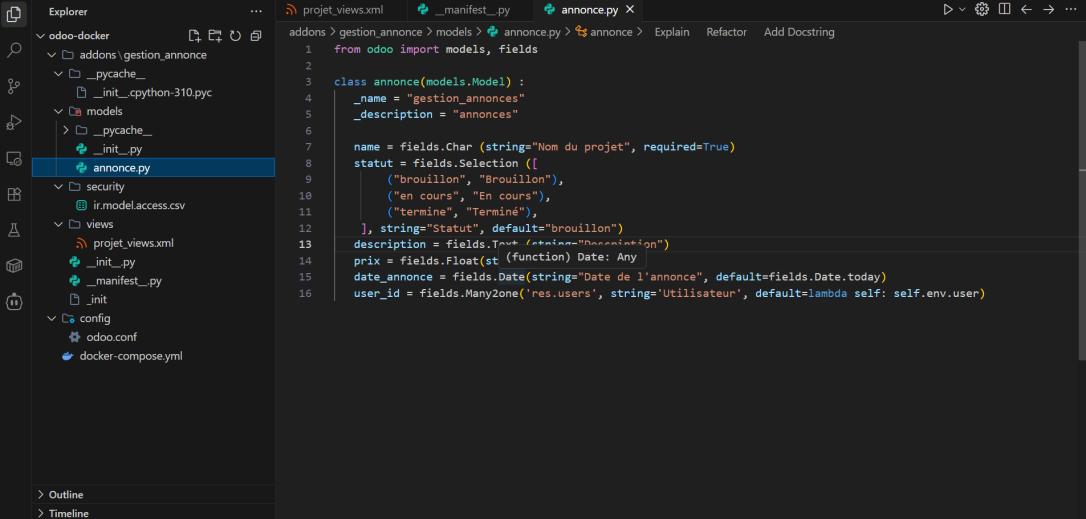
1  {
2      "name": "gestion_des annonces",
3      "version": "1.0",
4      "summary": "Module pour gérer des annonces.",
5      "category": "Training",
6      "author": "EMSI",
7      "depends": ["base"],
8      "data": [
9          "security/ir.model.access.csv",
10         "views/projet_views.xml",
11     ],
12     "installable": True,
13     "application": True,
14   }
15 }
16

```

FIGURE 2 – Contenu du fichier `__manifest__.py`

Le manifeste définit les métadonnées du module : nom, version, description, catégorie, auteur, dépendances et fichiers de données. Il est essentiel pour l'installation du module dans Odoo.

## 9.3 Définition du modèle de données



The screenshot shows the Odoo module structure in the left sidebar and the content of the `annonce.py` file in the main editor area.

```

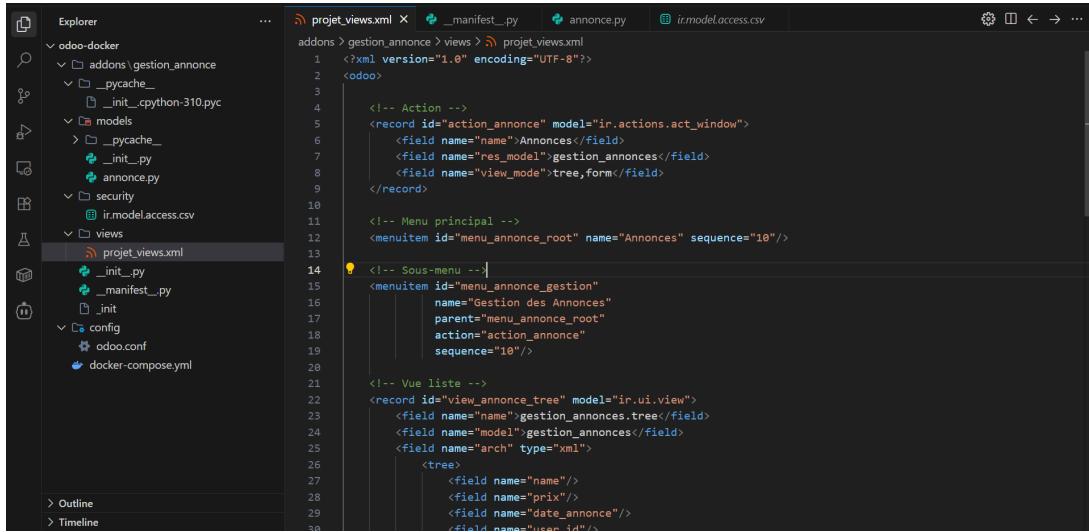
1  from odoo import models, fields
2
3  class annonce(models.Model):
4      _name = "gestion_annonces"
5      _description = "annonces"
6
7      name = fields.Char(string="Nom du projet", required=True)
8      statut = fields.Selection([
9          ("brouillon", "Brouillon"),
10         ("en cours", "En cours"),
11         ("terminé", "Terminé"),
12     ], string="Statut", default="brouillon")
13      description = fields.Text(string="Description")
14      prix = fields.Float(string="Prix", function=Date: Any)
15      date_annonce = fields.Date(string="Date de l'annonce", default=fields.Date.today)
16      user_id = fields.Many2one('res.users', string='Utilisateur', default=lambda self: self.env.user)

```

FIGURE 3 – Code Python du modèle `gestion_annonces`

Cette capture présente la classe Python définissant le modèle des annonces avec tous ses champs et leurs configurations.

## 9.4 Configuration des vues XML



```

<?xml version="1.0" encoding="UTF-8"?>
<odoo>
    <!-- Action -->
    <record id="action_annonce" model="ir.actions.act_window">
        <field name="name">Annonces</field>
        <field name="res_model">gestion_annonces</field>
        <field name="view_mode">tree,form</field>
    </record>

    <!-- Menu principal -->
    <menuitem id="menu_annonce_root" name="Annonces" sequence="10"/>

    <!-- Sous-menu -->
    <menuitem id="menu_annonce_gestion"
              name="Gestion des Annonces"
              parent="menu_annonce_root"
              action="action_annonce"
              sequence="10"/>

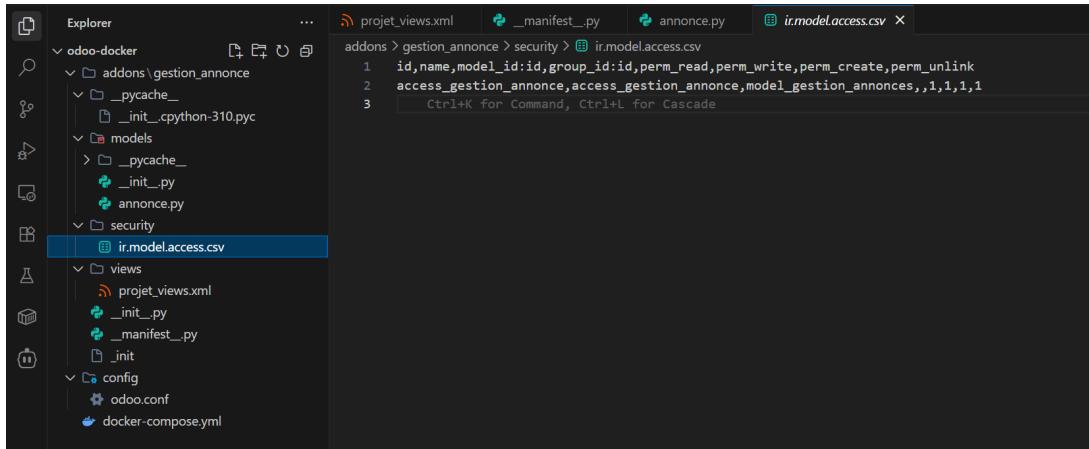
    <!-- Vue liste -->
    <record id="view_annonce_tree" model="ir.ui.view">
        <field name="name">gestion_annonces.tree</field>
        <field name="model">gestion_annonces</field>
        <field name="arch" type="xml">
            <tree>
                <field name="name"/>
                <field name="prix"/>
                <field name="date_annonce"/>
                <field name="user_id"/>
            </tree>
        </field>
    </record>
</odoo>

```

FIGURE 4 – Partie des vues XML définissant menus et actions

Le fichier XML configure les menus, les actions et les vues d’interface pour intégrer le module à l’interface Odoo.

## 9.5 Gestion des permissions d'accès



```

id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_gestion_annonce,access_gestion_annonce,model_gestion_annonces,,1,1,1,1

```

FIGURE 5 – Fichier CSV de contrôle d'accès

Ce fichier définit les droits CRUD (Create, Read, Update, Delete) sur le modèle **gestion\_annonces**.

## 9.6 Interface de liste des annonces

Reference	Titre	Auteur	Catégorie	Priorité	Date de publication	Date d'expiration	Nombre de vues	Statut
ANN00002	alerte	Mitchell Admin	Événement	★★	01/02/2026 22:04:41	01/03/2026 22:00:00	0	Publie
ANN00001	Alerte pour l'équipe	Mitchell Admin	Urgent	★★	01/01/2026 21:00:00	01/03/2026 21:00:00	0	Publie

FIGURE 6 – Vue liste des annonces internes dans Odoo

L’interface utilisateur montre le tableau listant toutes les annonces avec leurs principales informations.

## 9.7 Formulaire de création d’annonce

The screenshot shows the 'New' creation form for an internal announcement. It includes fields for Reference (set to 'New'), Title ('Alerte pour l'équipe'), Author ('Mitchell Admin'), Creation Date ('01/02/2026 21:50:19'), and Content/Pieces jointes. A 'Catégorie' dropdown menu is open, showing categories like Urgent, Information, Événement, Urgent, Politique, Formation (which is selected), Général, and Autre. Status buttons at the top right include 'Brouillon', 'Publie', and 'Archivée'.

FIGURE 7 – Formulaire de création/modification d’une annonce

Cette vue permet de saisir ou modifier toutes les informations d’une annonce interne.

## 9.8 Paramètres et configuration

The screenshot shows the configuration form for an existing announcement (Reference: ANN00001). It displays fields for Title ('Alerte pour l'équipe'), Author ('Mitchell Admin'), Creation Date ('01/02/2026 21:52:42'), and Content/Pieces jointes. A 'Catégorie' dropdown menu is open, showing categories like Urgent, Information, Événement, Urgent, Politique, Formation (selected), Général, and Autre. Status buttons at the top right include 'Archiver', 'Remettre en brouillon', 'Brouillon', 'Publie', and 'Archivée'. A note at the bottom left says 'faire le rapport'.

FIGURE 8 – Interface de paramétrage du module

Les paramètres permettent de configurer différents aspects du module selon les besoins de l'organisation.

## 9.9 Page d'information du module

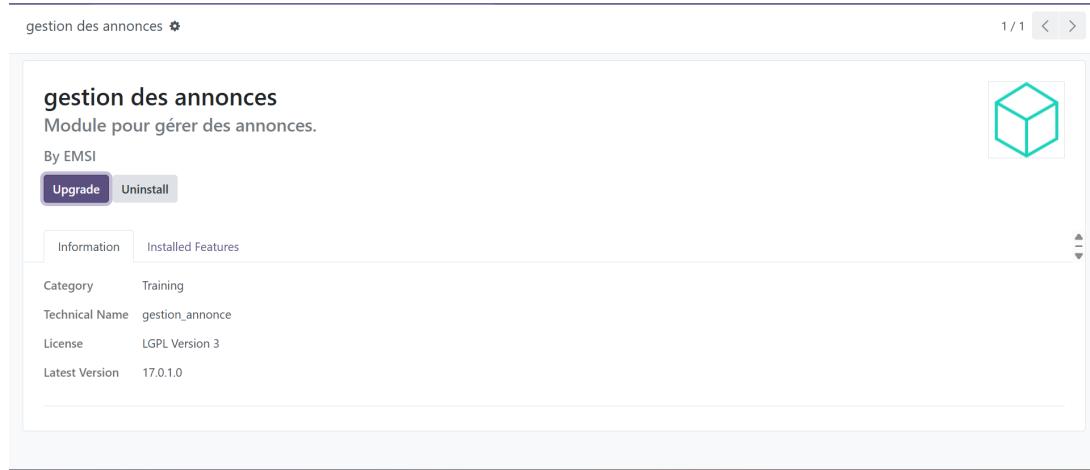


FIGURE 9 – Page d'information du module dans les applications Odoo

Cette page présente les informations techniques du module et permet son activation/-désactivation.

## 9.10 Module dans la liste des applications

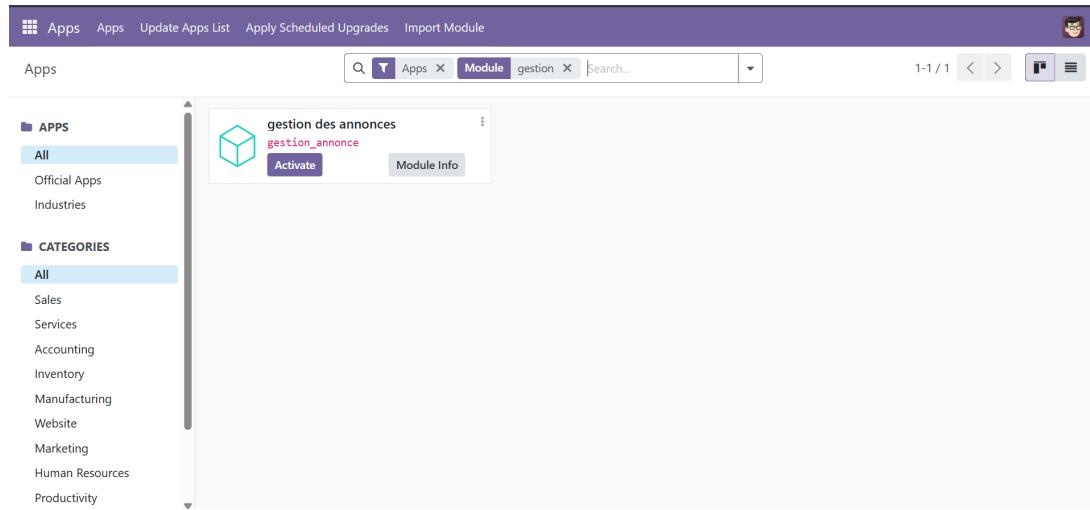


FIGURE 10 – Module visible dans la liste des applications Odoo

Le module apparaît correctement dans la liste des applications disponibles pour installation.

## 10 Résultats obtenus

Le module de gestion des annonces internes a été développé avec succès et répond à tous les objectifs initiaux :

- **Fonctionnalités implémentées :**
  - Création, modification et suppression d'annonces
  - Gestion des statuts (brouillon, en cours, terminé)
  - Attribution automatique de l'auteur
  - Dates de publication et d'expiration utilisateur complète (liste et formulaire)
  - Intégration parfaite dans le menu Odoo
- **Technique :**
  - Module correctement structuré selon les standards Odoo
  - Code Python propre et documenté
  - Vues XML fonctionnelles et ergonomiques
  - Permissions de sécurité configurées
  - Déploiement Docker opérationnel
- **Utilisateur :**
  - Interface intuitive et facile à utiliser
  - Accès rapide aux annonces
  - Gestion simplifiée du cycle de vie des annonces
  - Centralisation de la communication interne

## 11 Limites du projet

Certaines limitations ont été identifiées :

- **Fonctionnalités de base :** Le module couvre les besoins essentiels mais pourrait être enrichi
- **Gestion des pièces jointes :** Absence de système d'upload de fichiers
- **Notifications :** Pas de système d'alerte pour les nouvelles annonces
- **Reporting :** Absence de rapports statistiques sur les annonces
- **Personnalisation avancée :** Options limitées pour le formatage du contenu

## 12 Perspectives d'amélioration

Plusieurs évolutions pourraient enrichir le module :

- **Notifications automatiques :**
  - Alertes par email pour les nouvelles annonces
  - Rappels avant expiration
  - Notifications dans le chatter Odoo

- **Enrichissement du contenu :**
  - d'images et documents de formatage riche (éditeur WYSIWYG) de liens hypertextes
- **Gestion avancée :**
  - Catégories d'annonces personnalisables
  - Destinataires spécifiques (groupes, départements)
  - Workflow d'approbation pour publication
- **Reporting et analyse :**
  - Statistiques de consultation
  - Rapports PDF des annonces
  - Tableau de bord des publications
- **Intégrations :**
  - Connexion avec le module Employés
  - Synchronisation avec le calendrier Odoo /Android pour consultation mobile

## 13 Conclusion

Ce projet a permis de développer un module Odoo 17 fonctionnel pour la gestion des annonces internes, répondant au besoin de centralisation de la communication organisationnelle. Sur le plan technique, il a consolidé la maîtrise du développement Odoo (ORM Python, vues XML, sécurité, déploiement Docker) et des technologies web modernes. Bien que perfectible, le module constitue une base solide et démontre la capacité d'Odoo à supporter des solutions métier sur mesure. Cette expérience prépare à des développements ERP professionnels plus avancés.