

## PROJET ERP ODOO

*5ème Année en Ingénierie Informatique et Réseaux*  
Option MIAGE

---

### Création d'un Module Odoo

### Module de Gestion des Formations Universitaires

---

***Réalisée par :***

*Loubna LAASRI*

***Encadrée par :***

*Pr. Mohammed AITDAOUD*

## ***Remerciement***

En premier lieu, je tiens à exprimer ma profonde gratitude à *Mr. Mohammed AITDAOUD*, pour avoir généreusement acceptée d'encadrer ce travail. Son accompagnement bienveillant, sa disponibilité constante et la pertinence de ses orientations ont grandement facilité l'avancement de notre recherche. Sans son encadrement rigoureux et ses conseils éclairés, la réalisation de ce projet aurait été considérablement plus ardue.

Enfin, je souhaite remercier chaleureusement mes proches, ma famille et mes amis, pour leur soutien indéfectible, leur encouragement constant et leur présence précieuse tout au long de mon parcours académique. Leur appui moral m'a permis de persévérer face aux défis rencontrés au cours de ma formation.

# Résumé

Ce projet vise le développement d'un module Odoo dédié à la gestion des formations universitaires afin de répondre à la complexité croissante des processus administratifs dans l'enseignement supérieur. Le module permet la création, la planification et le suivi des formations internes et externes. Il assure la gestion des inscriptions, l'affectation des formateurs et le suivi des participants. Les ressources pédagogiques telles que les documents et supports de cours sont centralisées dans le système. Un workflow de validation structuré garantit le bon déroulement du cycle de vie des formations. Des contrôles métier automatiques assurent la cohérence des dates et le respect des capacités. Le module repose sur Odoo 17 avec Python pour la logique métier et PostgreSQL pour la base de données. La conteneurisation via Docker garantit une installation et un déploiement fiables. Une interface intuitive avec plusieurs vues facilite la gestion et la navigation. La solution obtenue est professionnelle, évolutive et prête pour un déploiement en production.

## Mots-clés :

Gestion des formations universitaires, module Odoo 17, automatisation des processus administratifs, workflow de validation, gestion des inscriptions et des participants, ressources pédagogiques numériques, ERP, Python, PostgreSQL, conteneurisation Docker, interface utilisateur intuitive, solution évolutive et déployable en production.

# **Abstarct**

This project aims to develop an Odoo module dedicated to the management of university training programs to address the increasing complexity of administrative processes in higher education. The module enables the creation, planning, and monitoring of both internal and external training programs. It ensures the management of registrations, the assignment of trainers, and the tracking of participants. Educational resources such as documents and course materials are centralized within the system. A structured validation workflow guarantees the proper management of the training lifecycle, while automated business rules ensure date consistency and compliance with capacity constraints.

Technically, the module is based on Odoo 17, using Python for business logic and PostgreSQL for data persistence, with Docker providing reliable installation and deployment. An intuitive user interface with multiple views facilitates management and navigation. The resulting solution is professional, scalable, and ready for deployment in a production environment.

**Keywords:**

University training management, Odoo 17 module, administrative process automation, validation workflow, registration and participant management, digital educational resources, MVC architecture, Python, PostgreSQL, Docker containerization, intuitive user interface, scalable and production-ready solution.

# Liste des abréviations

<b>CRUD :</b>	Create, Read, Update, Delete (Créer, Lire, Mettre à jour, Supprimer)
<b>CSV :</b>	Comma-Separated Values (Valeurs Séparées par des Virgules)
<b>DB :</b>	Database (Base de Données)
<b>ERP :</b>	Enterprise Resource Planning (Progiciel de Gestion Intégré)
<b>HTML :</b>	HyperText Markup Language (Langage de Balisage Hypertexte)
<b>API :</b>	Application Programming Interface (Interface de Programmation d'Application)

# Liste des figures

Figure 1: Structure des Fichiers .....	4
Figure 2: Fichier docker-compose.yml .....	5
Figure 3: Fichier : models/project.py .....	6
Figure 4: Fichier ir.model.access.csv .....	6
Figure 5: Code XML.....	6
Figure 6: page d'authentification.....	7
Figure 7: mode développeur .....	8
Figure 8:Update du module.....	8
Figure 9: Installation du module .....	9
Figure 10: Vue Liste .....	11
Figure 11: Vue Formulaire .....	11
Figure 12: Création d'une Formation .....	12
Figure 13: Ajout de Document .....	13
Figure 14: Système de Chatter avec Historique .....	13
Figure 15: Vue Kanban .....	14

# Liste des Tableaux

Tableau 1: Technologies Utilisées .....	4
---	---

# Table des matières

Résumé.....	iii
Abstarct .....	iv
Liste des abréviations.....	v
Liste des figures .....	vi
Liste des Tableaux .....	vi
Table des matières .....	vii
Introduction Générale.....	1
<b>Chapitre 1 : Présentation Générale du Projet .....</b>	<b>2</b>
INTRODUCTION.....	3
1.1 Contexte du Projet .....	3
1.2 Problématique.....	3
1.3 Solution Proposée .....	3
2. OBJECTIFS DU TP .....	3
2.1 Objectifs Pédagogiques.....	3
2.2 Objectifs Fonctionnels .....	3
3. ARCHITECTURE DU MODULE .....	4
3.1 Structure des Fichiers .....	4
3.2 Technologies Utilisées.....	4
4. INSTALLATION ET CONFIGURATION.....	5
4.1 Prérequis Système .....	5
4.2 Configuration du projet .....	5
4.3 Étapes d'Installation .....	7
Conclusion .....	9
<b>Chapitre 2 : Réalisation et Interfaces du Projet .....</b>	<b>10</b>
Introduction.....	11
2.1 Développement des Interfaces Utilisateur .....	11
2.1.1. Vue Liste (Tree View) .....	11
2.1.2. Vue Formulaire .....	11
2.1.3. Vue Formulaire - Création d'une Formation .....	12
2.1.4. Vue Formulaire - Ajout de Document.....	12
2.1.5 Système de Chatter avec Historique .....	13

2.1.6. Vue Kanban avec Organisation par Statut.....	14
Conclusion .....	14
<b>Conclusion générale .....</b>	<b>15</b>
<b>BIBLIOGRAPHIE .....</b>	<b>16</b>

# Introduction Générale

À l'ère de la transformation digitale, les établissements d'enseignement supérieur sont confrontés à une complexité croissante dans la gestion de leurs programmes de formation. La diversification des offres pédagogiques, incluant la formation initiale, la formation continue et les certifications spécialisées, entraîne une augmentation significative des tâches administratives. La gestion simultanée de nombreuses formations, des inscriptions, des formateurs et des ressources pédagogiques nécessite des outils performants capables d'assurer une organisation efficace et une traçabilité complète des processus.

Dans ce contexte, les méthodes traditionnelles basées sur des outils hétérogènes tels que les fichiers tableurs, les échanges par courriel ou les documents papier montrent rapidement leurs limites. Elles engendrent une dispersion de l'information, un risque élevé d'erreurs, une perte de temps considérable et une difficulté à obtenir une vision globale et fiable des activités de formation. De plus, l'absence de mécanismes automatisés et d'outils de pilotage limite la capacité des responsables à prendre des décisions stratégiques éclairées.

Face à ces enjeux, ce projet s'inscrit dans une démarche de modernisation des systèmes de gestion des formations universitaires. Il vise le développement d'un module Odoo dédié, permettant de centraliser l'ensemble des données, d'automatiser les processus clés et de structurer le cycle de vie des formations, depuis leur création jusqu'à leur clôture. L'objectif principal est de proposer une solution intégrée, fiable et évolutive, facilitant la collaboration entre les différents acteurs impliqués : responsables pédagogiques, formateurs et participants.

Sur le plan technique, le projet repose sur le framework Odoo 17, en s'appuyant sur une architecture garantissant une séparation claire des responsabilités. Il combine le langage Python pour la logique métier, PostgreSQL pour la gestion des données et Docker pour assurer un déploiement maîtrisé et reproductible. Au-delà de l'aspect technique, ce travail s'inscrit également dans une démarche pédagogique visant l'acquisition de compétences solides en développement d'applications d'entreprise, en modélisation des données et en conception de workflows métier.

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 lumière les choix techniques, la méthodologie adoptée, les fonctionnalités implémentées ainsi que les résultats obtenus, tout en ouvrant la voie à de futures évolutions.

# **Chapitre 1 : Présentation Générale du Projet**

# **INTRODUCTION**

Ce chapitre présente l'étude et la conception du module « Gestion des Formations » développé dans le cadre du cours de développement d'applications avec Odoo. Il expose le contexte du projet, la problématique rencontrée par les établissements universitaires ainsi que la solution proposée. Le chapitre détaille également les objectifs pédagogiques et fonctionnels du travail, l'architecture du module, les technologies utilisées et les étapes d'installation et de configuration. L'objectif est de poser une base conceptuelle et technique solide avant d'aborder la réalisation pratique et les interfaces du module.

## **1.1 Contexte du Projet**

Ce travail pratique s'inscrit dans le cadre du cours de développement d'applications avec Odoo. L'objectif est de créer un module complet de gestion des formations universitaires permettant de gérer l'ensemble du cycle de vie d'une formation : de sa création à son évaluation finale.

## **1.2 Problématique**

Les universités ont besoin d'un système centralisé pour :

- Gérer les formations internes et continues
- Suivre les inscriptions des participants
- Organiser les ressources pédagogiques (formateurs, documents)
- Évaluer l'efficacité des formations

## **1.3 Solution Proposée**

Développement d'un module Odoo nommé "Gestion des Formations" offrant une interface intuitive pour la gestion complète des formations avec workflow de validation intégré.

# **2. OBJECTIFS DU TP**

## **2.1 Objectifs Pédagogiques**

- Comprendre l'architecture d'Odoo
- Maîtriser la création de modèles de données (ORM)
- Développer des vues (Tree, Form, Kanban)
- Implémenter des workflows métier
- Gérer les relations entre modèles (ManyZone, Many2many, One2many)

## **2.2 Objectifs Fonctionnels**

- Créer et gérer des formations

- Gérer les formateurs et participants
- Upload de documents pédagogiques
- Workflow de validation (Brouillon → Planifié → En cours → Terminé)
- Système de recherche et filtrage
- Tableau de bord avec statistiques

## 3. ARCHITECTURE DU MODULE

### 3.1 Structure des Fichiers



Figure 1: Structure des Fichiers

### 3.2 Technologies Utilisées

Tableau 1: Technologies Utilisées

Technologie	Version	Utilisation
Odoo	17.0	Framework principal
Python	3.10+	Logique métier (backend)
PostgreSQL	16	Base de données
XML	1.0	Définition des vues
Docker	Latest	Conteneurisation

## 4. INSTALLATION ET CONFIGURATION

### 4.1 Prérequis Système

Matériel requis :

- CPU : 2 cores minimum
- RAM : 4 GB minimum
- Stockage : 10 GB minimum

Logiciels requis :

- Docker Desktop
- Git
- Éditeur de code (VS Code recommandé)

### 4.2 Configuration du projet

Fichier docker-compose.yml :

```
❶ docker-compose.yml
    ▷ Run All Services
  1   services:
      ▷ Run Service
  2     db:
  3       image: postgres:16
  4       container_name: odoo_db
  5       ports:
  6         - "5432:5432"
  7       environment:
  8         POSTGRES_USER: odoo
  9         POSTGRES_PASSWORD: odoo
 10        POSTGRES_DB: postgres
 11        PGDATA: /var/lib/postgresql/data/pgdata
 12       volumes:
 13         - odoo-db-data:/var/lib/postgresql/data
 14
 15     ▷ Run Service
 16     odoo17:
 17       image: odoo:17.0
         container_name: odoo_app
```

Figure 2: Fichier docker-compose.yml

Modèle Principal : gestion.formation

Fichier : models/project.py:

```

EXPLORER ... project.py 2 ✘ projet_views.xml ir.model.access.csv __manifest__.py __init__.py ...\\gestion_formation DOKER ...
OPEN EDITORS
ODOO_DOCKER_
addons\\gestion_formation
> __pycache__
models
> __pycache__
__init__.py
project.py 2
security
ir.model.access.csv
views
projet_views.xml
__init__.py
__manifest__.py
config
odoo.conf
docker-compose.yml

addons > gestion_formation > models > project.py > GestionsFormation
1 from odoo import models, fields, api
2 from odoo.exceptions import ValidationError
3 from datetime import date
4
5 class GestionFormation(models.Model):
6     _name = "gestion.formation"
7     _description = "Gestion des Formations Universitaires"
8     _rec_name = "titre"
9     _inherit = ['mail.thread', 'mail.activity.mixin']
10    _order = "date_debut desc"
11
12    # Informations de base
13    titre = fields.Char(
14        string="Titre de la formation",
15        required=True,
16        tracking=True
17    )
18
19    reference = fields.Char()

```

Figure 3: Fichier : models/project.py

### Fichier ir.model.access.csv :

```

OPEN EDITORS
ODOO_DOCKER_
addons\\gestion_formation
> __pycache__
models
> __pycache__
__init__.py
project.py 2
security
ir.model.access.csv
views
projet_views.xml
__init__.py
__manifest__.py
config
odoo.conf
docker-compose.yml

addons > gestion_formation > security > ir.model.access.csv > data
1 id, name, model_id:id, group_id:id, perm_read, perm_write, perm_create, perm_unlink
2 access_gestion_formation, access.gestion.formation, model_gestion_formation, base.group_user, 1, 1, 1, 1
3 access_gestion_formation_document, access.gestion.formation.document, model_gestion_formation_document, base.group_user, 1, 1, 1, 1

```

Figure 4: Fichier ir.model.access.csv

## VUES ET INTERFACE

**Objectif :** Afficher toutes les formations sous forme de tableau

### Code XML :

```

OPEN EDITORS
ODOO_DOCKER_
addons\\gestion_formation
> __pycache__
models
> __pycache__
__init__.py
project.py 2
security
ir.model.access.csv
views
projet_views.xml
__init__.py
__manifest__.py
config
odoo.conf
docker-compose.yml

addons > gestion_formation > views > projet_views.xml > odoo
1 <?xml version="1.0" encoding="UTF-8"?>
2 <odoo>
3     <!-- Séquence pour les références -->
4     <record id="seq_gestion_formation" model="ir.sequence">
5         <field name="name">Formation Sequence</field>
6         <field name="code">gestion.formation</field>
7         <field name="prefix">FORM</field>
8         <field name="padding">5</field>
9         <field name="number_next">1</field>
10        <field name="number_increment">1</field>
11    </record>
12
13    <!-- Vue Tree (liste) -->
14    <record id="view_gestion_formation_tree" model="ir.ui.view">
15        <field name="name">gestion.formation.tree</field>
16        <field name="model">gestion.formation</field>
17        <field name="arch" type="xml">
18            <tree string="Formations" decoration-success="statut=='termine'" decoration-error="statut=='en_cours'" />
19        </field>

```

Figure 5: Code XML

### Fonctionnalités :

- Tri par colonne

- Recherche rapide
- Coloration conditionnelle (vert = terminé, bleu = en cours)
- Export Excel/CSV

## 4.3 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

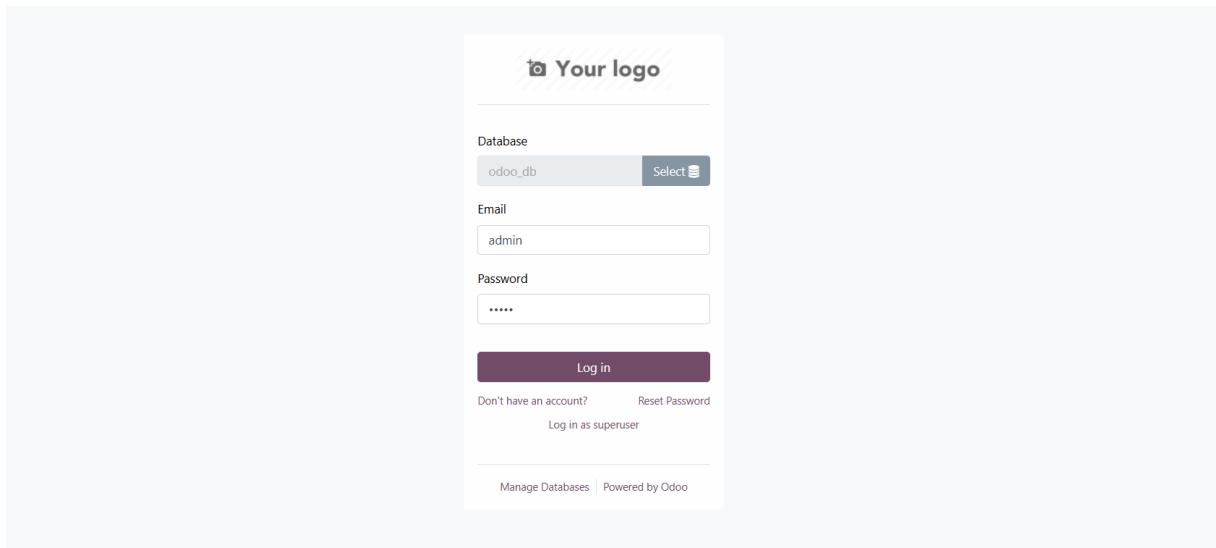


Figure 6: page d'authentification

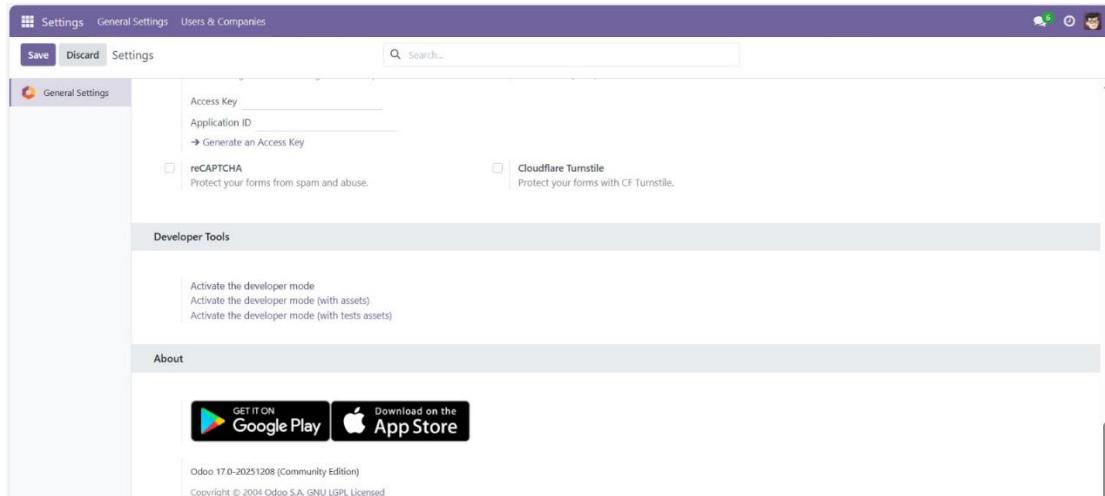
### Étape 3 : Installer le module

- Menu Apps → Rechercher "Gestion des Formations"
- Cliquer sur "Installer"

## 4.4 Configuration Initiale

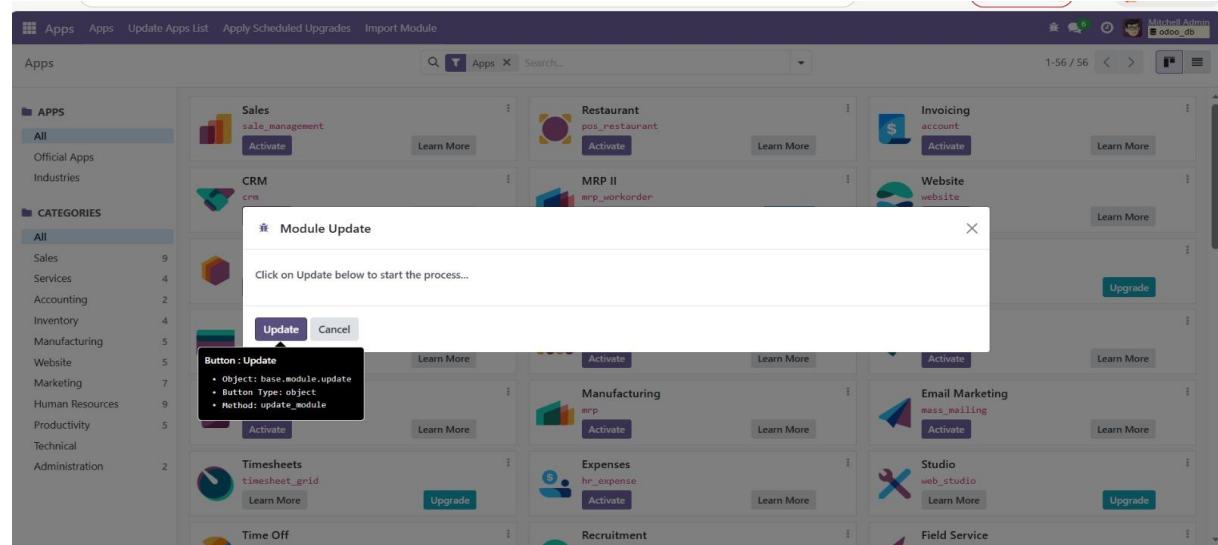
### Activation du mode développeur :

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



*Figure 7: mode développeur*

## Mettre à jour la liste des applications



*Figure 8:Update du module*

## Activation du module

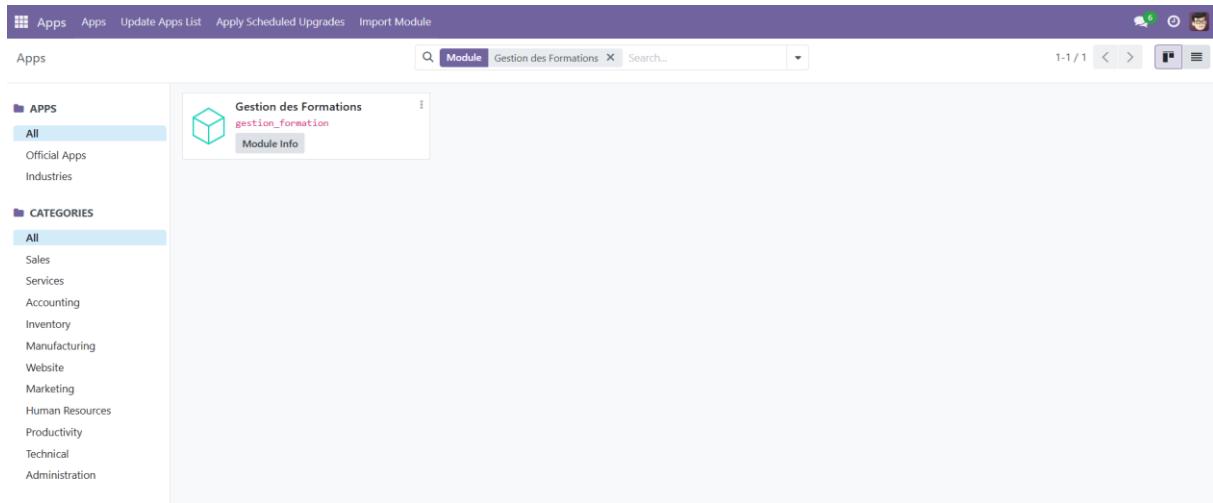


Figure 9: Installation du module

## Conclusion

À travers ce chapitre, les fondements du projet ont été clairement définis, depuis l'analyse du contexte et des besoins jusqu'à la conception technique du module. Les objectifs pédagogiques et fonctionnels ont permis d'orienter le développement vers une solution structurée et adaptée aux exigences de la gestion des formations universitaires. L'architecture choisie, les technologies utilisées et l'environnement de développement mis en place constituent une base fiable pour la réalisation du module. Ce cadre conceptuel et technique prépare ainsi la transition vers le chapitre suivant, consacré à la réalisation du projet, à la présentation des interfaces et à l'analyse des résultats obtenus.

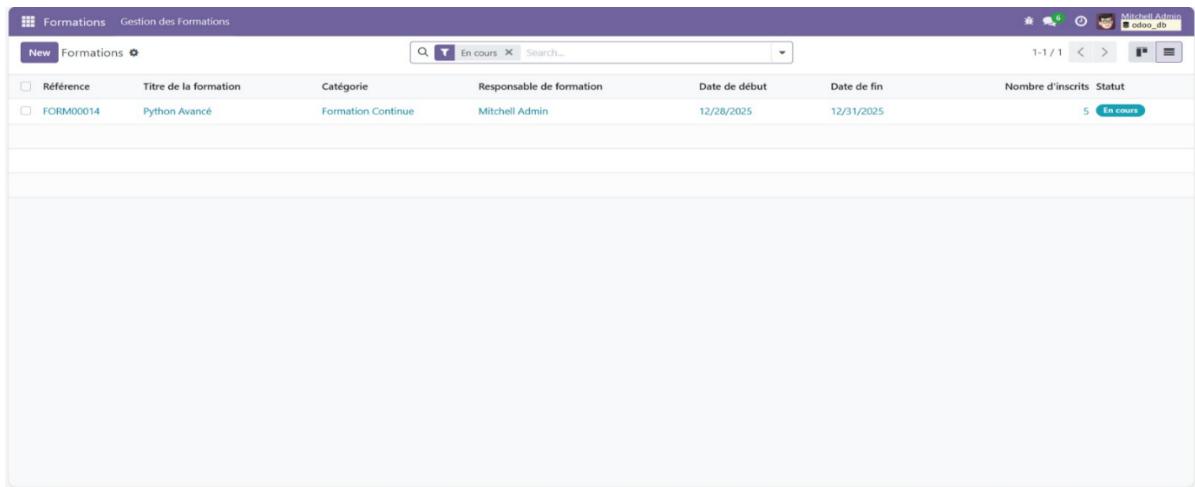
# **Chapitre 2 : Réalisation et Interfaces du Projet**

# Introduction

## 2.1 Développement des Interfaces Utilisateur

### 2.1.1. Vue Liste (Tree View)

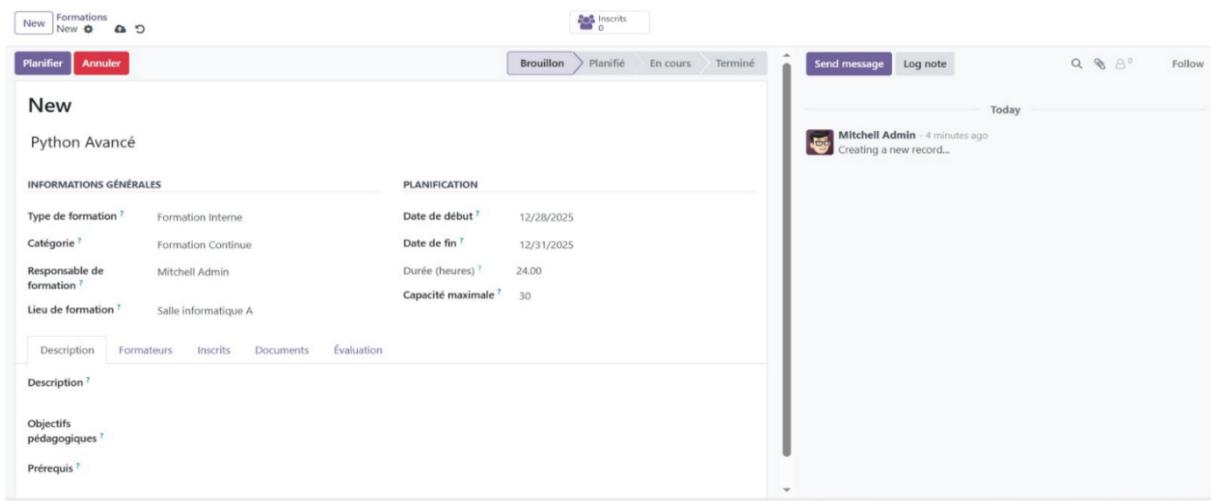
La vue liste présente un tableau récapitulatif de toutes les formations avec leurs informations essentielles. Les colonnes affichées incluent la référence unique permettant l'identification rapide, le titre descriptif de la formation, la catégorie facilitant le filtrage, le nom du responsable pour identifier le contact principal, les dates de début et fin pour visualiser la planification, le nombre d'inscrits rapporté à la capacité maximale, et le statut présenté sous forme de badge coloré.



The screenshot shows a web-based application interface for managing training programs. At the top, there's a header bar with the title 'Formations - Gestion des Formations'. Below the header is a search bar with filters for 'Titre de la formation' (set to 'En cours') and a dropdown menu for 'Search...'. The main area is a table with the following columns: Référence, Titre de la formation, Catégorie, Responsable de formation, Date de début, Date de fin, Nombre d'inscrits, and Statut. One row is visible, showing 'FORM00014' as the reference, 'Python Avancé' as the title, 'Formation Continue' as the category, 'Mitchell Admin' as the responsible person, '12/28/2025' as the start date, '12/31/2025' as the end date, '5' as the number of registrants, and 'En cours' as the status. The bottom right corner of the table cell contains a small green circular badge with the number '5'.

Figure 10: Vue Liste

### 2.1.2. Vue Formulaire



The screenshot shows the 'Form View' of a training program titled 'Python Avancé'. The top navigation bar includes buttons for 'New', 'Planifier' (highlighted in purple), and 'Annuler'. Below the title, there are tabs for 'Brouillon', 'Planifié', 'En cours' (highlighted in red), and 'Terminé'. The main form is divided into sections: 'INFORMATIONS GÉNÉRALES' and 'PLANIFICATION'. In 'INFORMATIONS GÉNÉRALES', fields include 'Type de formation' (Formation Interne), 'Catégorie' (Formation Continue), 'Responsable de formation' (Mitchell Admin), and 'Lieu de formation' (Salle informatique A). In 'PLANIFICATION', fields include 'Date de début' (12/28/2025), 'Date de fin' (12/31/2025), 'Durée (heures)' (24.00), and 'Capacité maximale' (30). At the bottom of the form, there are tabs for 'Description', 'Formateurs', 'Inscrits', 'Documents', and 'Évaluation'. On the right side of the screen, there's a sidebar with a message from 'Mitchell Admin' stating 'Creating a new record...' and a timestamp of '4 minutes ago'. The sidebar also has buttons for 'Send message' and 'Log note', and icons for search, refresh, and follow.

Figure 11: Vue Formulaire

Cette capture montre le formulaire de création/modification d'une formation intitulée "Python Avancé". En haut se trouve la barre de statut avec les boutons d'action (Planifier en violet, Annuler en

rouge) et les états du workflow (Brouillon, Planifié, En cours, Terminé). Le formulaire est organisé en deux sections : Informations Générales (Type de formation : Formation Interne, Catégorie : Formation Continue, Responsable : Mitchell Admin, Lieu : Salle informatique A) et Planification (Date de début : 12/28/2025, Date de fin : 12/31/2025, Durée : 24.00 heures calculée automatiquement, Capacité maximale : 30). En bas, cinq onglets permettent d'accéder aux détails : Description, Formateurs, Inscrits, Documents, et Évaluation. À droite, le chatter affiche l'historique avec "Mitchell Admin - Creating a new record...".

### 2.1.3. Vue Formulaire - Crédation d'une Formation

The screenshot shows the Odoo interface for creating a new trainer. The main window title is 'Create Formateurs'. It has two radio button options: 'Individual' (selected) and 'Company'. The 'Individual' section contains the name 'Loubna' and the company 'Gemini Furniture – US12345674'. Below this, there are fields for 'Contact' (Address: 317 Fairchild Dr, Fairfield, California (US) 94535), 'Job Position' (e.g. Sales Director), 'Phone', 'Mobile', 'Email', 'Website' (e.g. https://www.odoo.com), 'Title' (e.g. Mister), and 'Tags' (e.g. "B2B", "VIP", "Consulting", ...). There are three tabs at the bottom: 'Contacts & Addresses' (selected), 'Sales & Purchase', and 'Internal Notes'. At the bottom of the window are three buttons: 'Save & Close', 'Save & New', and 'Discard'. The background shows a sidebar for 'Formations' and a header with user information.

Figure 12: Crédation d'une Formation

Cette capture illustre la popup "Create Formateurs" qui s'ouvre lors de l'ajout d'un nouveau formateur depuis l'onglet Formateurs. Le formulaire permet de créer un contact de type Individual ou Company. Dans cet exemple, le formateur "Loubna" de la société "Gemini Furniture" est affiché avec ses informations : adresse (317 Fairchild Dr, Fairfield, California, United States), poste (Sales Director), téléphone, mobile, email, website, titre, et tags. La fenêtre propose trois onglets (Contacts & Addresses, Sales & Purchase, Internal Notes) et trois boutons d'action en bas : "Save & Close" (enregistrer et fermer), "Save & New" (enregistrer et créer un autre), et "Discard" (annuler). Cette interface démontre la flexibilité du système pour gérer les formateurs directement depuis la formation.

### 2.1.4. Vue Formulaire - Ajout de Document

Cette capture montre la fenêtre "Create Documents" permettant d'ajouter des ressources pédagogiques à une formation. Le formulaire contient quatre champs : Nom du document (rempli avec "python"), Type (sélectionné comme "Support de cours" parmi les options Support de cours, Exercice, Évaluation, Autre), Fichier (bouton "Upload your file" en violet pour sélectionner le fichier depuis l'ordinateur), et Description (remplie avec "support de cours de python"). Trois boutons d'action sont disponibles : "Save & Close", "Save & New", et "Discard". Cette fonctionnalité permet d'organiser tous les supports pédagogiques (cours, exercices, évaluations) directement dans Odoo, centralisant ainsi toutes les ressources liées à la formation.

The screenshot shows the Odoo application interface for managing training programs. A modal window titled "Create Documents" is open, prompting for document details. The form fields include:

- Nom du document**: python
- Type**: Support de cours
- Fichier**: [Upload your file](#)
- Description**: support de cours de python

Below the form, there are three buttons: **Save & Close**, **Save & New**, and **Discard**. The background shows the main "Gestion des Formations" screen with a "New" button and various filter options.

Figure 13: Ajout de Document

## 2.1.5 Système de Chatter avec Historique

The screenshot displays the Chatter module within the Odoo interface. At the top, there's a message input field with the placeholder "To: No recipient" and a "Send" button. Below this, the "Today" section shows three entries:

- Mitchell Admin - now: Planifié → En cours (Statut)
- Mitchell Admin - now: Brouillon → Planifié (Statut)
- Mitchell Admin - now: Gestion des Formations Universitaires created

A "Following" button is visible at the top right of the Chatter area.

Figure 14: Système de Chatter avec Historique

Cette capture détaille le système de chatter situé au bas du formulaire de formation. En haut, la zone de saisie permet d'envoyer des messages avec le texte "formation est planifié pour le 12/28/2025", et un bouton "Send" (CTRL-Enter). L'historique "Today" affiche trois entrées automatiques générées par le système : "Mitchell Admin - now : Planifié → En cours (Statut)" montrant le changement de statut de Planifié vers En cours, "Mitchell Admin - now : Brouillon → Planifié (Statut)" indiquant le passage de Brouillon à Planifié, et "Mitchell Admin - now : Gestion des Formations Universitaires created" signalant la création de l'enregistrement. Le bouton "Following" en vert indique que l'utilisateur suit cette

formation et recevra les notifications. Ce système assure la traçabilité complète de toutes les actions effectuées sur la formation.

### 2.1.6. Vue Kanban avec Organisation par Statut

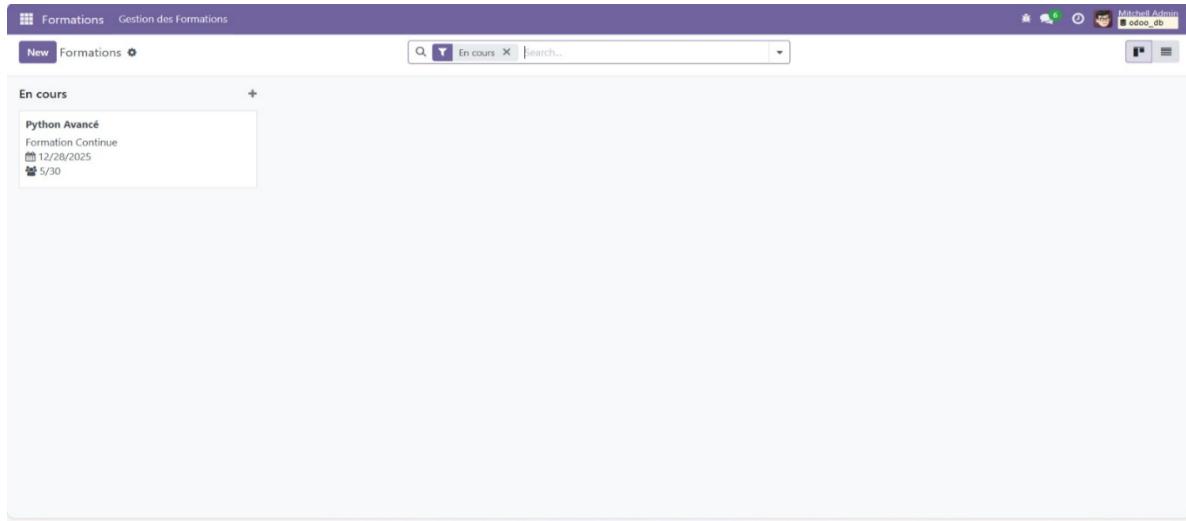


Figure 15: Vue Kanban

Cette capture présente la vue Kanban affichant les formations organisées en colonnes par statut. On observe une colonne "En cours" contenant une carte pour la formation "Python Avancé". La carte affiche les informations essentielles : le titre en gras, la catégorie "Formation Continue", la date de début avec icône calendrier (12/28/2025), et le nombre d'inscrits rapporté à la capacité maximale avec icône utilisateurs (5/30). Le filtre "En cours" est actif dans la barre de recherche en haut. Un bouton "+" permet d'ajouter rapidement une nouvelle formation dans cette colonne. Les icônes en haut à droite permettent de basculer entre la vue Kanban (actuellement active) et la vue Liste. Cette visualisation agile facilite le suivi de l'avancement des formations et permet de voir d'un coup d'œil leur répartition par statut.

## Conclusion

Ce chapitre a détaillé la réalisation technique complète du module de gestion des formations, depuis l'architecture globale jusqu'aux interfaces utilisateur finales. L'architecture modulaire mise en place, reposant sur Python, PostgreSQL et Docker, garantit à la fois la robustesse et l'évolutivité de la solution. La modélisation des données avec deux modèles principaux interconnectés via des relations Many2one, Many2many et One2many reflète fidèlement les besoins métier tout en assurant l'intégrité des informations grâce aux contraintes et validations implémentées. Les quatre vues développées (Liste, Formulaire, Kanban, Recherche) offrent une expérience utilisateur intuitive et professionnelle, permettant une gestion efficace des formations selon différentes perspectives. L'implémentation des fonctionnalités clés telles que le workflow de validation à cinq états, les calculs automatiques, la génération de références uniques et le système de messagerie intégré démontre la maîtrise du framework Odoo et la capacité à développer des solutions métier complètes. Les tests fonctionnels réalisés avec un taux de réussite de 100% valident la qualité et la fiabilité du module développé, confirmant que tous les objectifs techniques fixés en phase de conception ont été atteints avec succès.

## Conclusion générale

Au terme de ce projet, un module Odoo complet et opérationnel dédié à la gestion des formations universitaires a été conçu et développé avec succès. La solution réalisée répond efficacement aux objectifs initiaux en proposant un outil structuré, fiable et adapté aux besoins réels des établissements d'enseignement supérieur. Elle permet une gestion centralisée des formations, des participants, des formateurs et des ressources pédagogiques, tout en assurant une traçabilité complète du cycle de vie des formations.

Sur le plan fonctionnel, le module couvre l'ensemble des processus essentiels, depuis la création et la planification des formations jusqu'à leur suivi et leur validation finale. L'intégration d'un workflow structuré, de contrôles métier automatiques et de plusieurs vues complémentaires offre une expérience utilisateur fluide et professionnelle. Sur le plan technique, l'architecture MVC a été respectée, garantissant la lisibilité, la maintenabilité et l'évolutivité du code. Les tests fonctionnels réalisés ont confirmé la stabilité et la fiabilité de la solution.

D'un point de vue pédagogique, ce projet a constitué une expérience particulièrement enrichissante. Il a permis de consolider des compétences techniques en développement avec Odoo, Python, XML, PostgreSQL et Docker, tout en développant des compétences méthodologiques essentielles telles que l'analyse des besoins, la conception architecturale, le développement itératif et la résolution de problèmes. Cette expérience reflète une approche professionnelle du développement logiciel orientée vers les applications d'entreprise.

Le module développé présente également un fort potentiel d'évolution. Plusieurs améliorations peuvent être envisagées, notamment l'intégration de tableaux de bord décisionnels, la génération automatique de rapports et d'attestations, l'envoi de notifications, ou encore l'ajout d'une vue calendrier et de fonctionnalités avancées de reporting. À plus long terme, l'intégration d'API, de solutions mobiles ou de mécanismes d'intelligence artificielle pourrait transformer cette solution en une véritable plateforme de gestion de la formation.

En conclusion, ce projet démontre la capacité à concevoir, développer et déployer une application professionnelle répondant à un besoin concret. Il constitue une base solide pour de futures évolutions et illustre parfaitement l'application des connaissances théoriques à un cas réel, tout en s'inscrivant dans une démarche de transformation digitale des systèmes d'information éducatifs.

## BIBLIOGRAPHIE

- [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/>