# 1. Introduction

*IntranetComplet* est un projet Django complet destiné à la gestion de l'intranet d'entreprise. Il intègre :

* Authentification via AD/LDAP et/ou fichier JSON local
* Gestion des utilisateurs et des groupes
* Intégration avec Moodle
* Intégration avec Nextcloud
* Génération automatisée de documents RH
* Interface moderne (responsive, thème clair/sombre)

# 2. Architecture Générale

Une image contenant texte, capture d’écran, Police, diagramme

Le contenu généré par l’IA peut être incorrect.

# 3. Prérequis

* Python ≥ 3.10 (testé avec 3.12)
* pip
* Accès réseau à Moodle, Nextcloud, AD/LDAP
* git

# 4. Installation et Configuration

## 4.1 Clonage du dépôt

git clone https://votre-repo/intranetcomplet.git

cd intranetcomplet

## 4.2 Environnement virtuel et dépendances

python -m venv venv

source venv/bin/activate # sous Linux/macOS

venv\Scripts\activate # sous Windows

pip install -r requirements.txt

## 4.3 Fichier .env

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## 4.4 Migrations et superutilisateur

python moodle/manage.py migrate

python moodle/manage.py createsuperuser

Affecter au groupe admin :

from django.contrib.auth.models import User, Group

user = User.objects.get(username="votre\_user")

admin\_group = Group.objects.get(name="admin")

user.groups.add(admin\_group)

user.save()

## 4.5 Tests de connexion Nextcloud

python test\_nextcloud\_simple.py

Résultat attendu : code HTTP 207 ou 200.

## 4.6 Lancement du serveur

nohup python moodle/manage.py runserver 0.0.0.0:8000 > server.log 2>&1 &

Accès : http://<votre\_serveur>:8000/

# 5. Configuration des Services Intégrés

## 5.1 Active Directory / LDAP

* Variables .env : AD\_SERVER, AD\_DOMAIN, AD\_SEARCH\_BASE
* Authentification via UserService.authenticate()
* Synchronisation dans auth.User et UserProfile

## 5.2 API Moodle

Parfait ! Voici le tableau prêt à être copié-collé directement dans Word. Il est optimisé pour un affichage clair avec colonnes bien alignées. Tu peux le coller dans la section **5.2 API Moodle** de ton document Word sans perdre la mise en forme.

**📋 Fonctions Web Services utilisées**

| **Fonction** | **Description** | **Capabilités associées** |
| --- | --- | --- |
| core\_course\_create\_categories | Créer des catégories de cours | moodle/category:manage |
| core\_course\_create\_courses | Créer de nouveaux cours | moodle/course:create, moodle/course:visibility |
| core\_course\_delete\_categories | Supprimer des catégories de cours | moodle/category:manage |
| core\_course\_delete\_courses | Supprimer des cours | moodle/course:delete |
| core\_course\_get\_categories | Obtenir les détails des catégories | moodle/category:viewhiddencategories |
| core\_course\_get\_contents | Obtenir le contenu des cours | moodle/course:update, moodle/course:viewhiddencourses |
| core\_course\_get\_courses | Obtenir les détails des cours | moodle/course:view, moodle/course:update, moodle/course:viewhiddencourses |
| core\_course\_get\_courses\_by\_field | Rechercher des cours par champ | moodle/course:view, moodle/course:update |
| core\_course\_update\_categories | Mettre à jour des catégories | moodle/category:manage |
| core\_course\_update\_courses | Mettre à jour des cours | moodle/course:update, moodle/course:changecategory, moodle/course:changefullname, moodle/course:changeshortname, moodle/course:changeidnumber, etc. |
| core\_enrol\_get\_enrolled\_users | Obtenir les utilisateurs inscrits à un cours | moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update, moodle/site:accessallgroups |
| core\_role\_assign\_roles | Assigner des rôles manuellement | moodle/role:assign |
| core\_role\_unassign\_roles | Retirer des rôles manuellement | moodle/role:assign |
| core\_user\_get\_users | Rechercher des utilisateurs | moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update |
| core\_webservice\_get\_site\_info | Obtenir des infos sur le site et l’utilisateur | moodle/webservice:createtoken |
| enrol\_manual\_enrol\_users | Inscrire manuellement des utilisateurs | enrol/manual:enrol |
| enrol\_manual\_unenrol\_users | Désinscrire manuellement des utilisateurs | enrol/manual:unenrol |
| local\_ajouturl\_add\_url | Ajouter un lien URL dans un cours | moodle/course:manageactivities |
| local\_ajouturl\_delete\_url | Supprimer un lien URL d’un cours | moodle/course:manageactivities |
| local\_wsmanagesections\_create\_sections | Créer des sections dans un cours | moodle/course:update |
| local\_wsmanagesections\_delete\_sections | Supprimer des sections d’un cours | moodle/course:update |

💡 **Astuce Word** : une fois collé, tu peux sélectionner le tableau, cliquer sur “Disposition du tableau” > “Ajuster automatiquement” > “Ajuster au contenu” pour que tout soit bien lisible.

Souhaites-tu aussi que je t’aide à styliser ce tableau avec des couleurs ou bordures spécifiques ?

Plugins nécessaires :

* local\_ajouturl
* local\_wsmanagesections

## 5.3 API Nextcloud (Capdrive)

* Connexion via WebDAV et OCS
* Fonctions : list\_nc\_dir(path), création de partages
* Variables .env : NEXTCLOUD\_WEBDAV\_URL, NEXTCLOUD\_SHARE\_URL, etc.

# 6. Utilisation de l'Intranet

## 6.1 Authentification et rôles

* Sources : JSON local et AD/LDAP
* Rôles : admin, user
* Vue : caplogy\_app.views.login

## 6.2 Navigation et fonctionnalités

* Accueil, catégories, cours
* Ajout/Suppression d’enseignants
* Gestion des logos (SchoolImage)

## 6.3 Génération de documents (AutoDocs)

* App Django intégrée
* Templates HTML pour RH
* Formulaires de saisie

## 6.4 Interface d'administration Django

* URL : /admin
* Modèles : User, Group, SchoolImage, etc.

# 7. Détails des Composants

## 7.1 caplogy\_project

* Fichiers : settings.py, urls.py, WSGI/ASGI
* Chargement .env via python-dotenv

## 7.2 caplogy\_app

* Modèles : UserProfile, SchoolImage
* Services : UserService, MoodleAPI, NextcloudAPI
* Vues et formulaires : views.py, forms.py
* Templates : base.html, login.html, etc.

## 7.3 App homepage

* Vue simple, template home.html

## 7.4 Projet AutoDocs

* App Django indépendante
* Commande create\_roles
* Templates RH dans templates\_docs

# 8. Tests et Qualité

* Script : test\_nextcloud\_simple.py

# 9. Déploiement

1. Proxy : Nginx
2. SSL : certbot/LetsEncrypt
3. Sécurité : désactiver DEBUG, protéger .env

# 10. Dépannage et FAQ

* LDAP : vérifier AD\_SEARCH\_BASE, certificats
* Moodle : activer web services, vérifier token
* Nextcloud : vérifier identifiants, URL DAV vs OCS et path dans nc\_dir

# 11. Annexes

* requirements.txt : Django, requests, ldap3, etc.