Cahier de Fonctionnement

# Projet : Application de gestion universitaire

Développée avec Django

## 1. Objectif du projet

Ce système a pour objectif de centraliser et de simplifier la gestion des informations liées aux étudiants, professeurs, cours, notes, départements, classes et emplois du temps au sein d'une université. Chaque utilisateur dispose d’un espace dédié selon son rôle (étudiant, professeur, administrateur).

## 2. Acteurs et rôles

|  |  |
| --- | --- |
| Rôle | Description |
| Administrateur | Gère tous les éléments de la plateforme : étudiants, professeurs, cours, notes, emplois du temps. |
| Professeur | Peut s’inscrire, gérer ses cours, noter les étudiants, consulter leur liste, gérer son emploi du temps. |
| Étudiant | Peut s’inscrire, accéder à ses cours, ses notes et son emploi du temps. |

## 3. Pages et Fonctionnalités

Les fonctionnalités sont réparties selon les rôles définis.

### 🔐 Authentification

- Page de connexion  
- Inscription Étudiant  
- Inscription Professeur

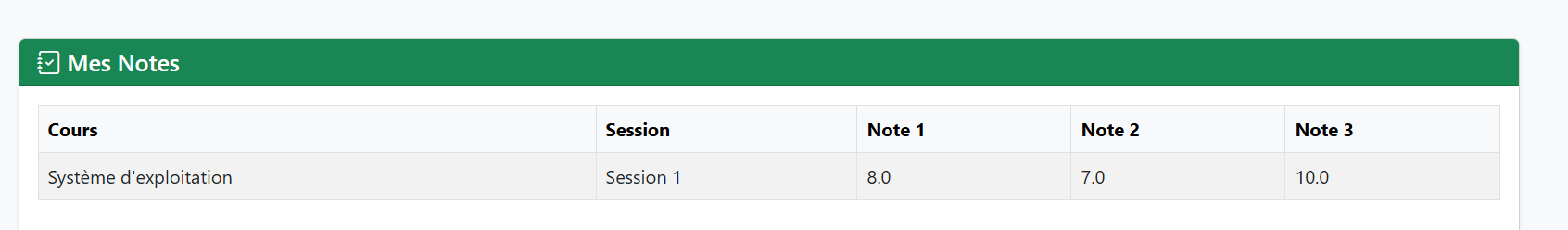
Une image contenant texte, capture d’écran, Police, logiciel

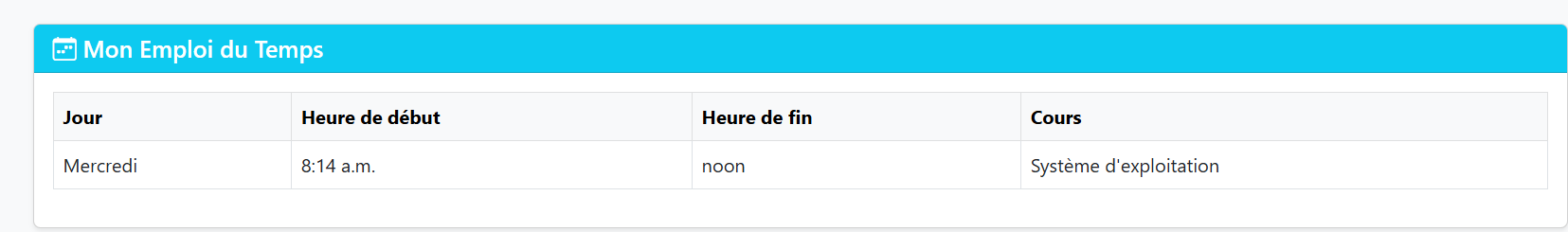
Le contenu généré par l’IA peut être incorrect.

### 👨‍🎓 Espace Étudiant

- Tableau de bord personnalisé  
- Consultation des cours  
- Visualisation des notes  
- Accès à l’emploi du temps  
- Consultation des informations personnelles







👨‍🏫 Espace Professeur

- Consultation de ses informations  
- Ajout / modification de ses cours  
- Ajout de notes pour les étudiants  
- Liste des étudiants  
- Emploi du temps personnalisé

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

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

Le contenu généré par l’IA peut être incorrect.

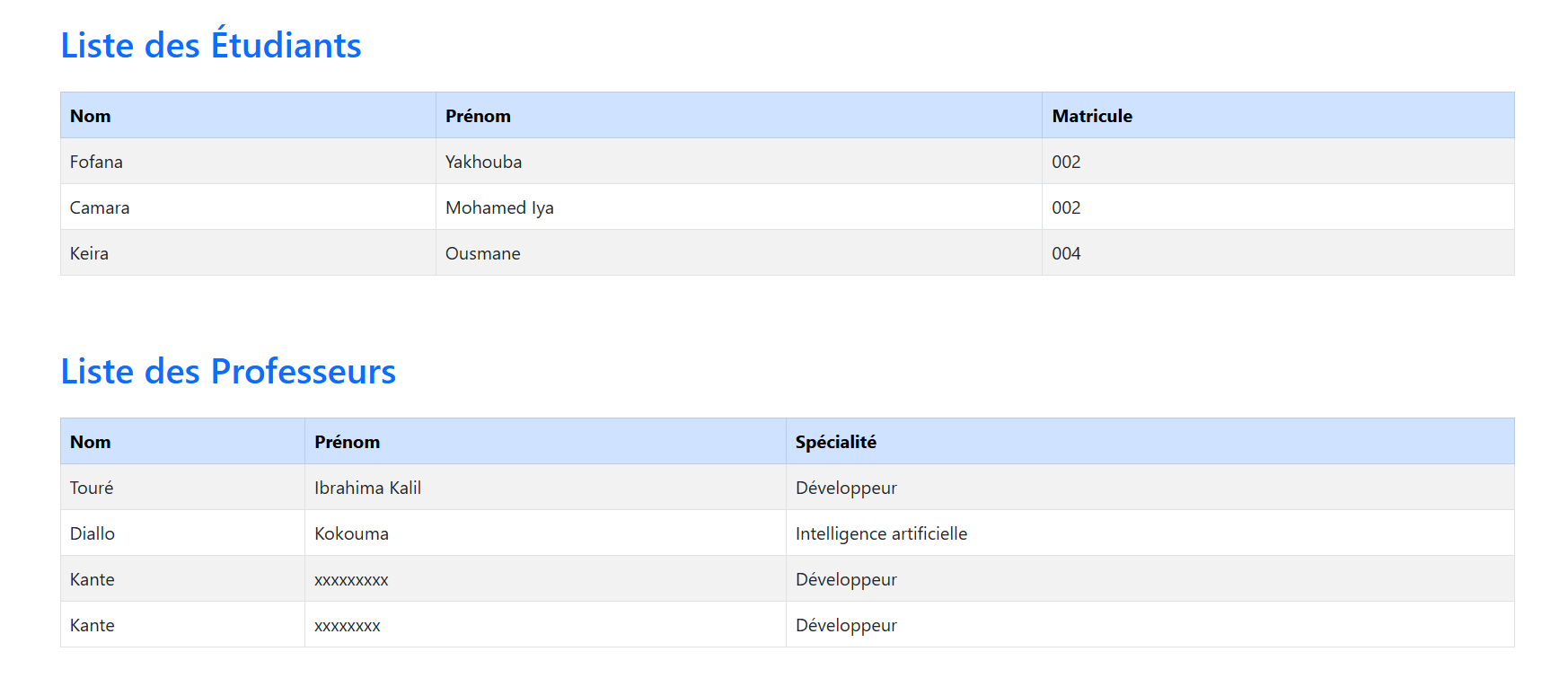
Une image contenant texte, ligne, Police, nombre

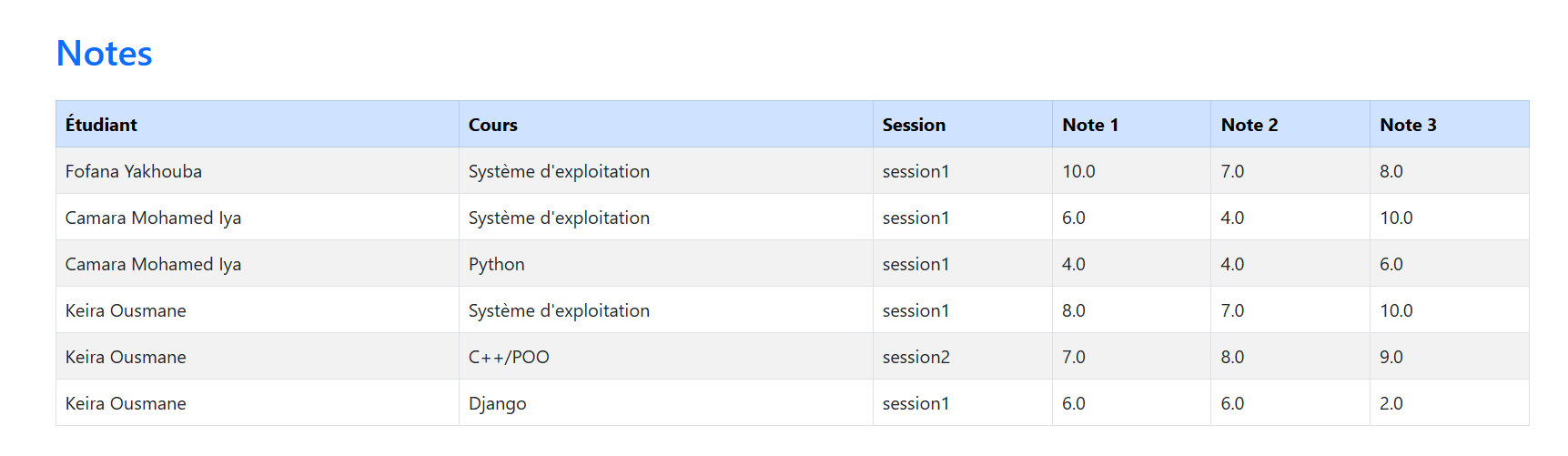
Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, logiciel, nombre, capture d’écran

Le contenu généré par l’IA peut être incorrect.

### 🛠️ Espace Administrateur

- Tableau de bord  
- Gestion complète : étudiants, professeurs, cours, classes, départements, notes, emploi du temps

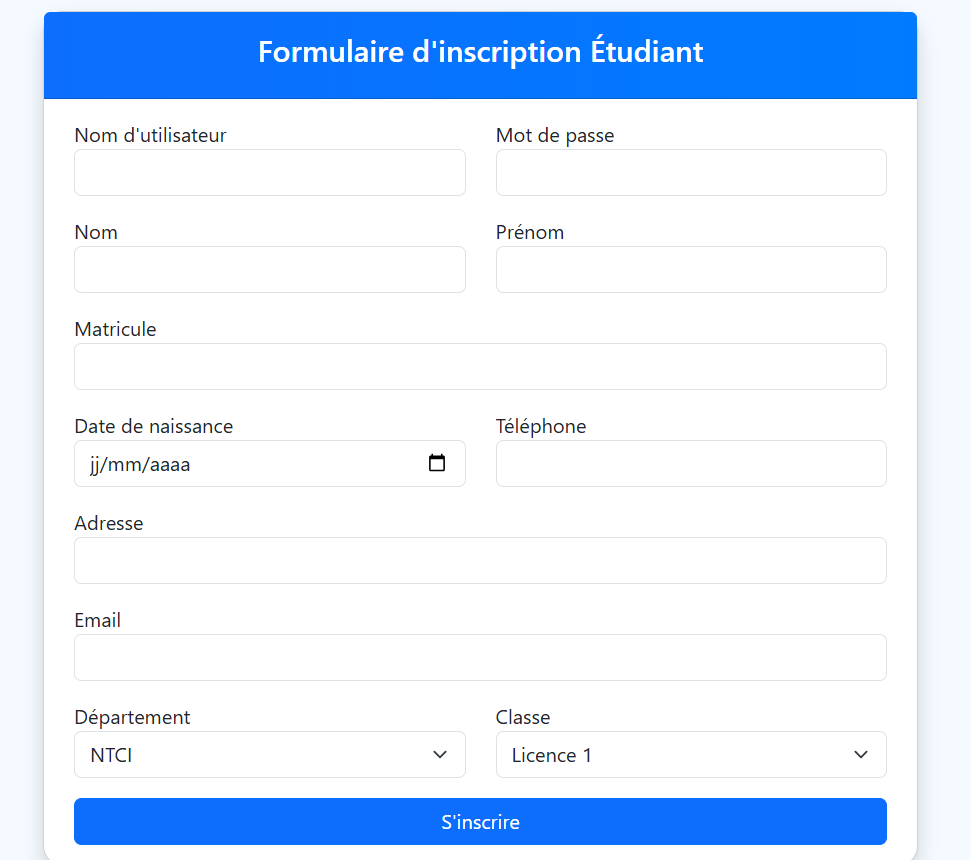


## 4. Flux de fonctionnement général

Voici quelques exemples de processus utilisateur.

* ✔️ Inscription Étudiant :

1. Accès au formulaire  
   2. Remplissage des champs requis  
   3. Création d’un compte Utilisateur (rôle = étudiant)  
   4. Création du profil Étudiant lié à l’utilisateur  
   5. Redirection vers la page de connexion



* ✔️ Ajout d’une note par un professeur :

1. Connexion au tableau de bord  
2. Sélection d’un étudiant, d’un cours et d’une session  
3. Remplissage des champs de note  
4. Sauvegarde dans le modèle Note

## 5. Structure technique

Modèles : Utilisateur, Étudiant, Enseignant, Cours, Note, Classe, Département, EmploiDuTemps

## 6. Technologies utilisées

- Django (framework web Python)  
- SQLite (base de données par défaut)  
- Bootstrap (interface graphique responsive)  
- HTML, CSS, Django Templates  
- Authentification personnalisée avec AbstractUser

## 7. Organisation des fichiers

- models.py → définition des modèles  
- views.py → logique métier et affichage  
- urls.py → routes  
- templates/ → interface utilisateur (etudiant, professeur, admin)