

Fiche de procedure pour la SAE_203, notre sujet était la gestion d'absence :

Fiche de procédure : Partie Django

1. Environnement utilisé

- Python 3.12
- Django 5.2.3
- SQLite3 (base de données)

2. Étapes de développement

a. Initialisation du projet

Création du projet Django avec la commande :

```
django-admin startproject gestion_absences
```

Puis création de l'application principale :

```
python manage.py startapp absences
```

b. Création des modèles (models.py)

Définition des entités : Groupe, Étudiant, Enseignant, Cours, Absence avec relations ForeignKey pour gérer les liaisons. Ajout d'un champ pour la photo des étudiants et pour la justification des absences (fichier PDF/image).

c. Création des formulaires (forms.py)

Utilisation de ModelForm pour chaque entité afin de créer des formulaires rapidement et simplement intégrables dans les vues.

d. Création des vues (views.py)

Ajout des vues pour chaque opération CRUD pour chaque entité. Utilisation des fonctions `get_object_or_404`, `redirect`, `render`. Une vue spécifique permet de générer une fiche PDF d'absence à l'aide de `xhtml2pdf` qui sera télécharger avec la commande suivante :

```
(.venv) pip install xhtml2pdf
```

e. Templates HTML

Création de templates responsives pour chaque entité (groupes.html, etudiants.html, etc.). Ajout de boutons stylisés avec icônes pour modifier et supprimer les objets.

Utilisation de `{% url %}`, `{% csrf_token %}` et `{% for %}` pour les boucles d'affichage.

f. Configuration des routes (urls.py)

Ajout des routes pour chaque entité, en suivant la structure :

```
path('etudiant/<int:id>/modifier/', views.modifier_etudiant, name='modifier_etudiant')
```

Création de noms clairs pour chaque opération CRUD.

4. Résultat

Le site web permet :

- L'ajout/modification/suppression de groupes, enseignants, étudiants, cours, absences
- L'ajout d'un justificatif (PDF/image) pour les absences
- La génération de fiches d'absence en format PDF

- L'affichage des photos des étudiants
 - Une interface claire et stylisée**
-

MySQL

1. Création de la base et de l'utilisateur dédié

Pour commencer, connectez-vous à MariaDB en tant qu'administrateur. Créez ensuite une nouvelle base de données nommée `gestion_absences`. Une fois la base en place, créez un compte utilisateur dédié (par exemple `admin_abs`) en lui attribuant un mot de passe robuste. Enfin, accordez à cet utilisateur tous les privilèges sur la base `gestion_absences`, puis appliquez ces droits pour qu'ils prennent effet immédiatement.

2. Définition des tables et explication de leur création

- Groupes d'étudiants
On crée d'abord une table pour stocker les différents groupes d'étudiants. Chaque groupe se voit attribuer un identifiant unique (auto-incrémenté) et un libellé textuel pour son nom. Un champ supplémentaire garde trace de l'identifiant du cours auquel le groupe est lié, afin de savoir quel cours chaque groupe suit.
- Étudiants
La table des étudiants comporte, pour chaque apprenant, un identifiant unique auto-incrémenté. On y enregistre le nom, le prénom et l'adresse e-mail (cette dernière devant être unique et non vide). Un champ de secours permet de stocker le nom de groupe en doublon si nécessaire, tandis qu'un champ de type binaire (BLOB) est prévu pour stocker la photo de l'étudiant. Enfin, une clé étrangère relie chaque étudiant à son groupe d'appartenance.
- Enseignants
Les enseignants sont eux aussi identifiés par un identifiant unique auto-incrémenté. On y conserve leur nom, leur prénom et leur adresse e-mail (à nouveau unique et non vide) pour pouvoir les contacter et les lier aux cours qu'ils dispensent.
- Cours
Chaque cours dispose de son propre identifiant unique. On y note un titre descriptif, la date programmée et la durée (au format horaire). Deux liens référentiels sont établis : l'un vers l'enseignant responsable du cours, l'autre vers le groupe d'étudiants qui y assiste.
- Absences de cours
Enfin, la table des absences recense pour chaque absence un identifiant unique automatique. Elle stocke l'ID du cours concerné, celui de l'étudiant absent et celui de l'enseignant ayant constaté l'absence. Un indicateur booléen précise si l'absence est justifiée ou non, et un champ textuel permet de renseigner le motif de justification.

3. Mise en place des relations

Pour garantir l'intégrité des données, chaque étudiant est relié à son groupe via une clé étrangère ; chaque cours est lié à son enseignant et à son groupe ; et chaque enregistrement d'absence renvoie clairement au cours, à l'étudiant et à l'enseignant concernés. Cette organisation assure une structure cohérente, facile à interroger et à maintenir.

Déploiement Django + Gunicorn + Nginx avec MariaDB distante (Debian 12)

Infrastructure

- **Server-web (Django + Gunicorn + Nginx)** : 192.168.66.230
- **Server-DB (MariaDB)** : 192.168.66.70

1. Serveur MariaDB (Serveur-DB)

Installer MariaDB

```
sudo apt update
sudo apt install mariadb-server
sudo mariadb
```

Puis j'ai importer le fichier .sql

```
mysql -u root -p nom_de_la_base < /home/toto/gestion_absences.sql
```

```
CREATE USER 'admin'@'192.168.66.230' IDENTIFIED BY 'toto';
GRANT ALL PRIVILEGES ON myproject.* TO 'admin'@'192.168.66.230';
FLUSH PRIVILEGES;
EXIT;
```

Autoriser les connexions distantes

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Modifier :

```
bind-address = 0.0.0.0
```

Redémarrer le service :

```
sudo systemctl restart mariadb
```

2. Serveur Django (Server-web)

Installer les paquets nécessaires

```
sudo apt update
sudo apt install python3-pip python3-dev libmariadb-dev build-essential libssl-dev libffi-dev python3-venv
```

Préparer le projet

Importation de l'application django dans /home/toto

```
cd ~/gestion_absences
python3 -m venv venv
source venv/bin/activate
```

Installer Django et les dépendances

```
(.venv) pip install django gunicorn mysqlclient
```

Si erreur pour mysqlclient :

```
sudo apt install default-libmysqlclient-dev
```

Configurer settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'myproject',
        'USER': 'django_user',
        'PASSWORD': 'your_password',
        'HOST': '192.168.6.70',
        'PORT': '3306',
    }
}

ALLOWED_HOSTS = ['192.168.66.230']
```

Migrer la base de données

```
(.venv) python manage.py migrate
```

3. Gunicorn

Tester Gunicorn

```
gunicorn --workers 3 --bind 127.0.0.1:8000 myproject.wsgi:application
```

Créer le service systemd

```
sudo nano /etc/systemd/system/gunicorn.service
```

Contenu :

```
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=your_username
Group=www-data
WorkingDirectory=/home/your_username/gestion_absences
ExecStart=/home/toto/gestion_absences/.venv/bin/gunicorn --workers 3 --bind
unix:/home/toto/gestion_absences/gestion_absences.sock gestion_absences.wsgi:application

[Install]
WantedBy=multi-user.target
```

Puis :

```
sudo systemctl start gunicorn
sudo systemctl enable gunicorn
```

4. Nginx

Installer Nginx

```
sudo apt install nginx
```

Créer la configuration du site

```
sudo nano /etc/nginx/sites-available/gestion_absences
```

Contenu :

```
server {
    listen 80;
    server_name 192.168.66.230;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/toto/gestion_absences;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/home/toto/gestion_absences/gestion_absences.sock;
    }
}
```

Activer et redémarrer Nginx

```
sudo ln -s /etc/nginx/sites-available/gestion_absences /etc/nginx/sites-enabled
sudo nginx -t
sudo systemctl restart nginx
```

Test final

Ouvre ton navigateur sur :

```
http://192.168.66.230
```

Notre application Django s'affiche et est fonctionnel