# NOTE DE SYNTHÈSE


**HAFAIDH MOHAMED**

**1 ère Année BTS SIO**

**Stage en entreprise**

**21 Mai - 21 Juin 2024**

# Sommaire

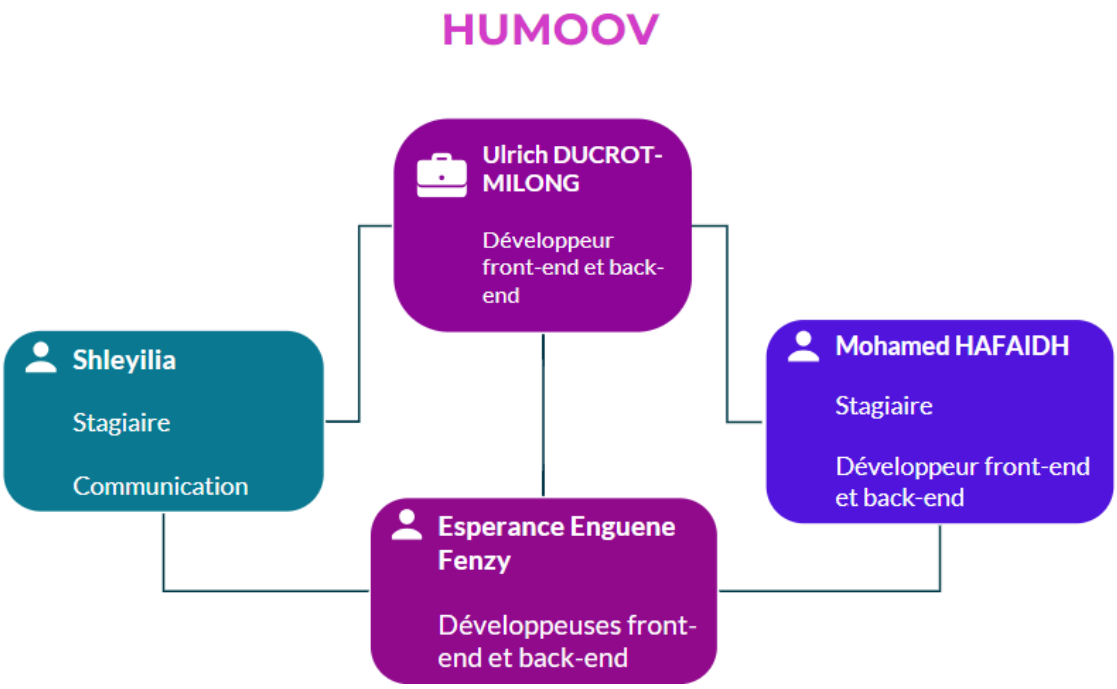
# Présentation de l’entreprise

L’entreprise Humoov, fondée le 1er mars 2018, est une entreprise innovante spécialisée dans la collecte de données personnelles, offrant aux clients une compensation financière en échange de leurs données.

Pour chaque utilisateur qui partage ses données personnelles avec Humoov, Humoov offre 10 euros par mois, soit 120 euros par an. Ce programme reflète l’engagement à valoriser les contributions des utilisateurs et à leur offrir une reconnaissance financière pour leur participation.

# Contexte et objectifs du stage

## Organigramme de l’entreprise



## I - BACK END

### - Python3

**À la fin du stage je devrais être en mesure de :**

- ouvrir et activer un environnement virtuel en python3 ;
- écrire des fonctions en python3 ;
- maîtriser les bases de la création de modèles sous python3 ;

### - Django

- sauvegarder des informations à travers les modèles ;
- être capable de créer des “vues utilisateur” sous le framework django ;
- relier ces vues à des urls navigables ;
- maîtriser la réceptions de données provenant du front-end ;
- intégrer des fichiers static (HTML, CSS, JS) ;

## II - FRONT-END

À la fin du stage le stagiaire devra être en mesure de :

### - HTML

- comprendre et modifier un formulaire en HTML ;
- adapter le formulaire à une vue utilisateur respectant l’UX design ;

### - CSS

- gérer et corriger des erreurs en .css ;
- maîtriser le maintien de la librairie des classes ;

### A - JAVASCRIPT

- maîtriser la collecte de données à travers les formulaires ;
- maîtriser l’envoi de données à travers des requêtes AJAX ;
- maîtriser la gestion de l’expérience utilisateur.

# Présentation de mon Poste de Travail

**En tant que stagiaire en développement web. Aujourd'hui, je vais vous présenter mon poste de travail ainsi que les projets qui me sont confiés.**

**Sachant que pendant le stage la première semaine ainsi que la deuxième j'étais en formation pour apprendre Python ainsi que le Framework Django, cela me permettra d'être à l'aise dans le vrais projet de l'entreprise pour les semaines à venir.**

## Projets principales : apprendre django


**https://www.djangoproject.com/**

1. **Installer et configurer un environnement Django**
2. **Développer un site web avec un formulaire pour la collecte de données**

**Installation et configuration de l'environnement Django.**

**Django est un framework web puissant et flexible en Python. Voici les étapes pour installer et le configurer :**

```
// Mise à jour des paquets
sudo apt update

// Installation de Python3 :
sudo apt install python3 python3-pip
```

```
// Installation de virtualenv
pip3 install virtualenv

// Création d'un environnement virtuel

mkdir myproject
cd myproject
virtualenv venv


// Activation de l'environnement virtuel
source venv/bin/activate

// Installation de Django

pip install django
```

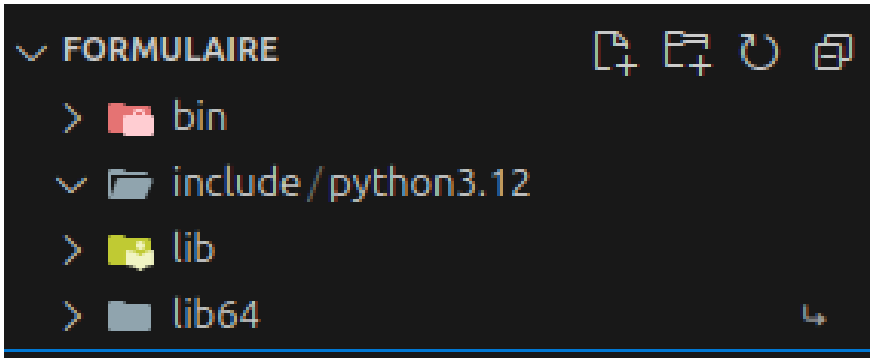
**En résumé, j'ai suivi les différentes étapes cruciales pour installer Django, un framework web basé sur Python, sur un système Linux.**

**Après l'installation de Django, il faut utiliser un logiciel qui interprète votre projet, par exemple Visual Studio Code par Microsoft ou Fleet par JetBrains, ses deux logiciels sont gratuits.**



Une fois que le virtualenv (l'environnement virtuel) est créé nous avons 4

fichiers bin, lib, include et lib64.




**bin** (ou **Scripts** sous Windows) :

- Contient les exécutables et scripts nécessaires pour utiliser l'environnement virtuel.

**lib** :

- Contient tous les paquets Python installés dans l'environnement virtuel.

**include** :

- Contient les fichiers d'en-tête C nécessaires pour compiler certains modules Python.

**lib64** :

- (ce dossier n'est pas toujours présent, il dépend du système et de la version de Python)

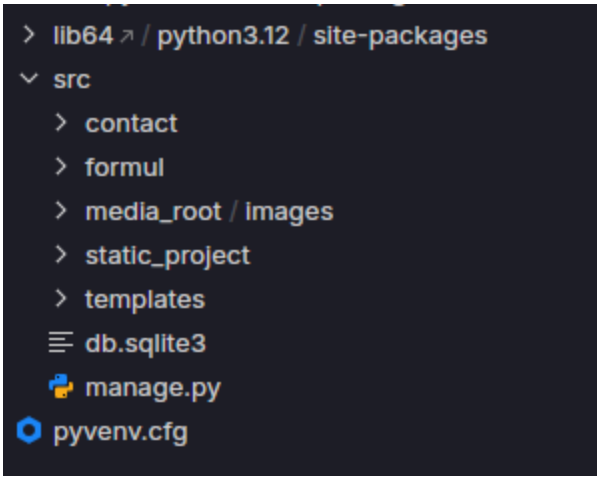
Une fois que l'environnement virtuel est créé, on doit créer un projet pour qu'on puisse travailler sur notre application WEB. Pour cela

**Activer l'Environnement Virtuel :**

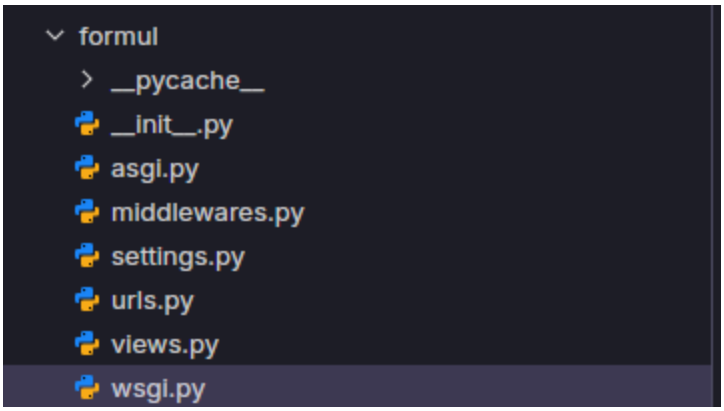

Installer le projet django :

La commande : django-admin startproject formul


Après l'installation nous avons un fichier 'src' à l'intérieur se trouve notre projet formul.




`formul/`: Il s'agit du dossier principal de votre projet Django. Ce dossier contient tous les fichiers de configuration et les composants de votre application web.




Chaque fichier joue un rôle spécifique dans le fonctionnement et la configuration de l'application Django.

| |
|---|
| **__pycache__/:** Ce dossier est automatiquement généré par Python pour stocker les fichiers compilés des modules Python. Il n'est pas nécessaire de s'en préoccuper directement. |
| **__init__.py:** Ce fichier indique à Python que ce dossier doit être traité comme un module. Il est souvent vide, mais il peut être utilisé pour exécuter du code d'initialisation pour le module. |
| **asgi.py:** Ce fichier est utilisé pour déployer l'application Django avec ASGI (Asynchronous Server Gateway Interface), qui permet de gérer les requêtes asynchrones. C'est utile pour les applications en temps réel ou nécessitant des communications WebSocket. |
| **middlewares.py:** Ce fichier contient les middlewares de votre application. Les middlewares sont des composants logiciels qui s'exécutent entre le serveur web et votre application Django. Ils peuvent traiter les requêtes et réponses avant qu'elles ne soient gérées par les vues. |
| **settings.py:** Ce fichier contient toutes les configurations et paramètres de votre projet Django, comme les configurations de la base de données, les applications installées, les paramètres de sécurité, etc. |
| **urls.py:** Ce fichier définit les URL que votre application va gérer et les vues correspondantes. Il fait le lien entre les URL saisies par l'utilisateur et les fonctions de vue qui génèrent les réponses. |

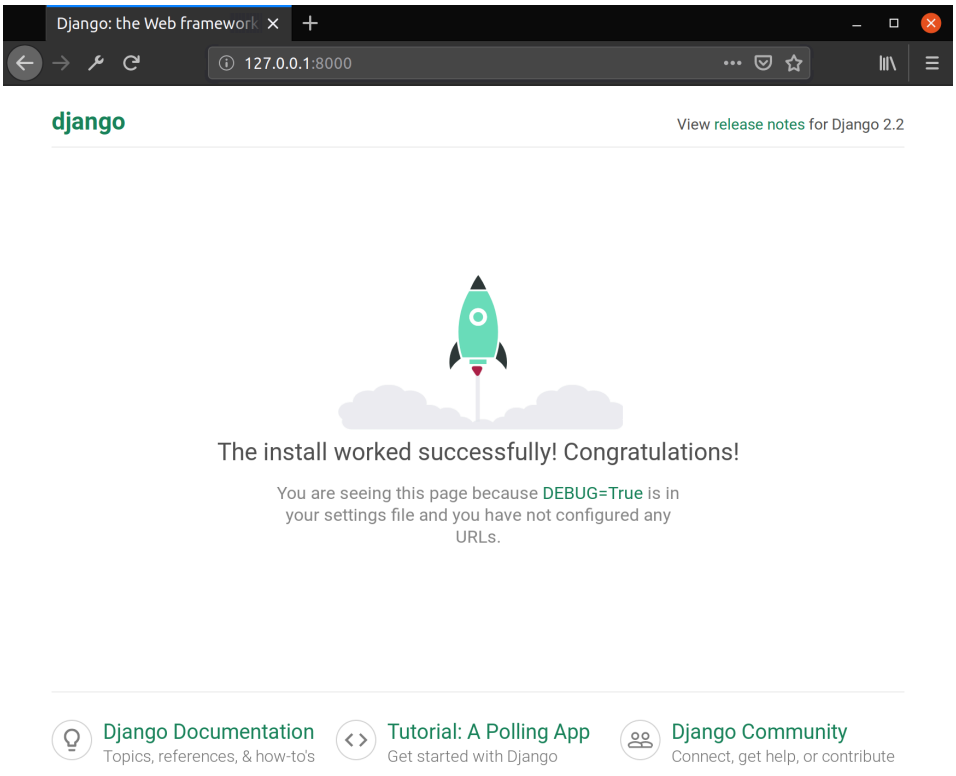
| |
|---|
| **views.py:** Ce fichier contient les fonctions de vue, qui gèrent la logique de traitement des requêtes et génèrent les réponses pour les utilisateurs. Les vues peuvent rendre des pages HTML, rediriger vers d'autres pages, etc. |
| **wsgi.py:** Ce fichier est utilisé pour déployer l'application Django avec WSGI (Web Server Gateway Interface), qui est le standard pour les communications entre les serveurs web et les applications web Python. |

Une fois le projet créé, on doit utiliser la commande **python manage.py runserver** pour démarrer notre serveur.

```
(formulaire) anonymat@anonymat-GS60:~/Bureau/formulaire/src$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 07, 2024 - 16:14:21
Django version 5.0.6, using settings 'formul.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Puis on tombe sur notre application **WEB** grâce à **l'IP** localhost sur le navigateur.

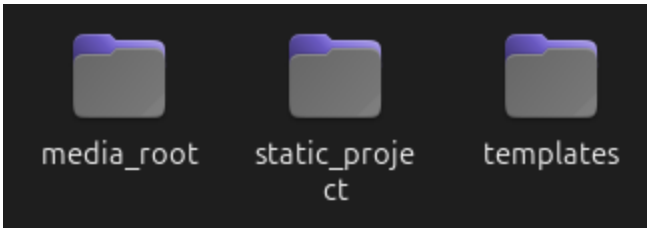



Une fois que

l'application web fonctionne correctement, nous devons cree des dossier pour intégrer notre projet pour cela faut faire des modifications dans le dossier **src/**, les modifications vont nous permettre d'intégrer notre **html,css** ainsi que le **javascript** grâce au fichier qui se trouve dans **src/formul** nommé **settings.py.**

Pour intégrer nos fichiers html,css et javascript nous devons créer 3 dossiers dans le **'src'** qui vont nous permettre de stocker les fichiers de ses 3 extensions.

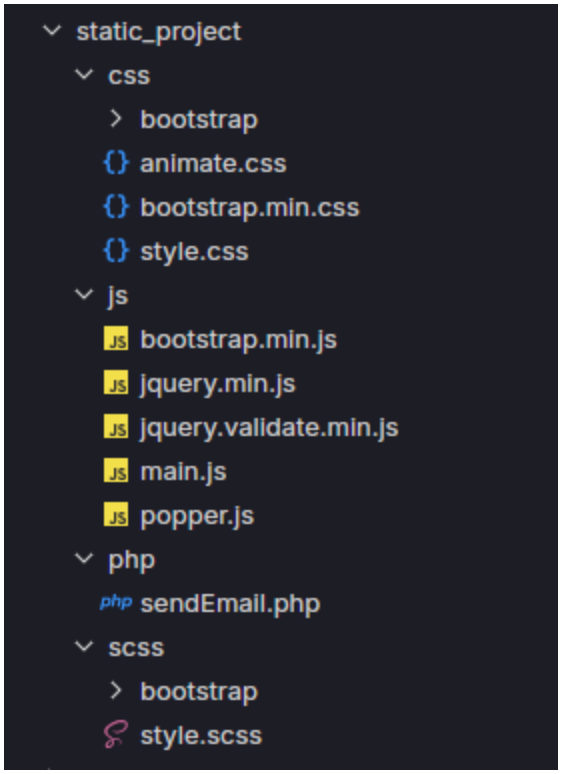
Ses dossier doivent avoir un nom **unique** :




Dans le dossier **media_root** il faut mettre que les images par exemple :

**src/media_root/image/image.png**

Dans le dossier **static_project** il faut mettre les fichier **css, js, php** et **scss** par exemple :



Dans le dossier **templates** il faut mettre les fichiers html par exemple :

index.html, main.html tous les fichiers contenant l'extension .html.

Une fois cela fait on doit s'y rendre dans le fichier **settings.py** qui se trouve dans **/src/formul/settings.py.** Une fois dans ce fichier nous devons lui indiquer le chemin pour qu'il puisse récupérer les dossier créé précédemment, pour cela nous devons descendre toute en bas du fichier puis lui indiquer le chemin en langage python :


STATICFILES_DIRS :

- os.path.join(BASE_DIR, "static_project") : Cela indique à Django de rechercher les fichiers statiques dans le répertoire static_project situé à la racine du projet.

STATIC_ROOT :

- Cette ligne définit le chemin absolu où les fichiers statiques seront collectés pour la mise en production. Lors de l'exécution de la commande collectstatic, Django rassemblera tous les fichiers statiques des répertoires spécifiés et les placera dans ce répertoire.

Par exemple, cela pourrait être utilisé pour servir des fichiers statiques à partir d'un serveur web comme Nginx.

MEDIA_URL = '/media/' :

- Cette ligne définit l'URL relative où les fichiers médias (comme les images téléchargées par les utilisateurs) seront accessibles depuis le navigateur.

  Par exemple, une image téléchargée serait accessible à l'URL http://votre-site/media/nom-de-l-image.jpg.

MEDIA_ROOT :

- Cette ligne définit le chemin absolu où les fichiers médias téléchargés par les utilisateurs seront stockés. Par exemple, si un utilisateur télécharge une image, elle sera stockée dans le répertoire media_root à la racine du projet.

En résumé, ce code configure où Django doit rechercher et stocker les fichiers statiques et médias, ainsi que les URL à utiliser pour les rendre accessibles via un navigateur web.

Une fois cela fait nous avons faire quelque modification dans le fichier **views.py :**

wsgi.py × views.py ×

```python
#Import Python

#Import Django

from django.shortcuts import render, get_object_or_404
from django.http import JsonResponse, HttpResponse

#Formulaire Import

def public_views (request):

    #Get me as the user
    me = request.user

    #Variable
    hello = "Bienvenue sur le formulaire"

    #Context
    context = {
        'bonjour' : hello
```

- **django.shortcuts.render** : Cette fonction permet de générer une réponse en utilisant un modèle HTML.
- **django.shortcuts.get_object_or_404** : Cette fonction permet de récupérer un objet de la base de données, et retourne une erreur 404 si l'objet n'existe pas.
- **django.http.JsonResponse** : Cette classe permet de renvoyer une réponse HTTP avec du contenu JSON.
- **django.http.HttpResponse** : Cette classe permet de renvoyer une réponse HTTP classique.

def public_views(request) :

**public_views** : C'est le nom de la fonction de vue qui sera appelée lorsqu'une requête correspondante est reçue.

**request** : C'est l'objet qui représente la requête HTTP reçue par le serveur.

me = request.user :

**request.user** : Cet attribut de l'objet `request` représente l'utilisateur actuellement connecté. S'il n'y a pas d'utilisateur connecté, il sera égal à `votre user (le nom de votre PC vu que on est en localhost)`

return render(request, "index.html", context) :

**render** : Cette fonction prend trois arguments :

- **request** : L'objet de la requête HTTP.
- **"index.html"** : Le nom du modèle HTML à utiliser pour générer la réponse.
- **context** : Le dictionnaire de contexte contenant les données à passer au modèle.

**Conclusion de la mission :**

Cette vue Django simple récupère l'utilisateur actuel, définit un message de

bienvenue, et rend une page HTML (`index.html`) avec ce message. C'est une structure typique d'une vue dans Django, illustrant comment traiter des requêtes et renvoyer des réponses en utilisant des modèles HTML et des variables de contexte.

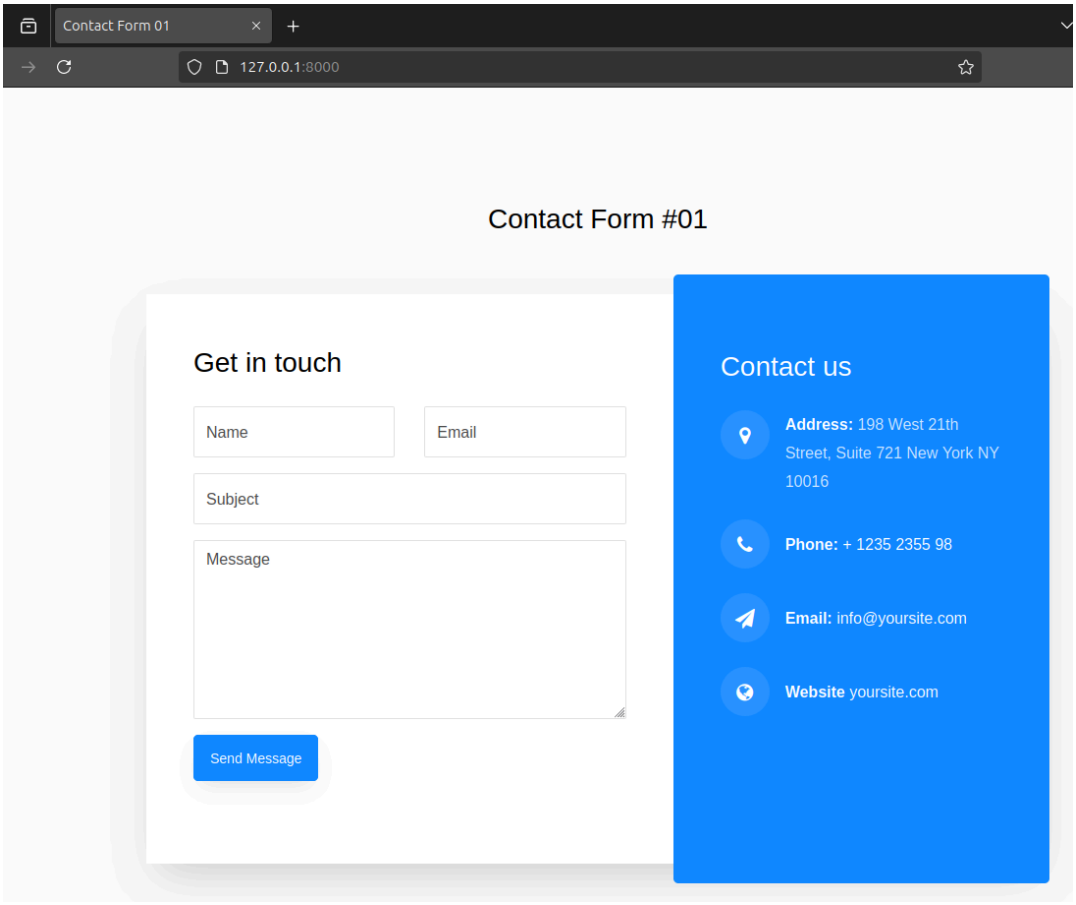
Une fois cela fait nous pouvons lancer notre serveur pour faire fonctionner notre application web :

Avant cela nous devons faire dans la console de notre application web la commande suivante :

```
(formulaire) anonymat@anonymat-GS60:~/Bureau/formulaire/src$ python3 manage.py makemigrations
No changes detected
(formulaire) anonymat@anonymat-GS60:~/Bureau/formulaire/src$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contact, contenttypes, sessions
Running migrations:
  No migrations to apply.
(formulaire) anonymat@anonymat-GS60:~/Bureau/formulaire/src$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 15, 2024 - 09:44:53
Django version 5.0.6, using settings 'formul.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Aller dans votre navigateur web et vous taper l'IP suivante donc 127.0.0.1:8000 pour information c'est L'IP localhost, une fois cela fait vous avez accès à votre page web :



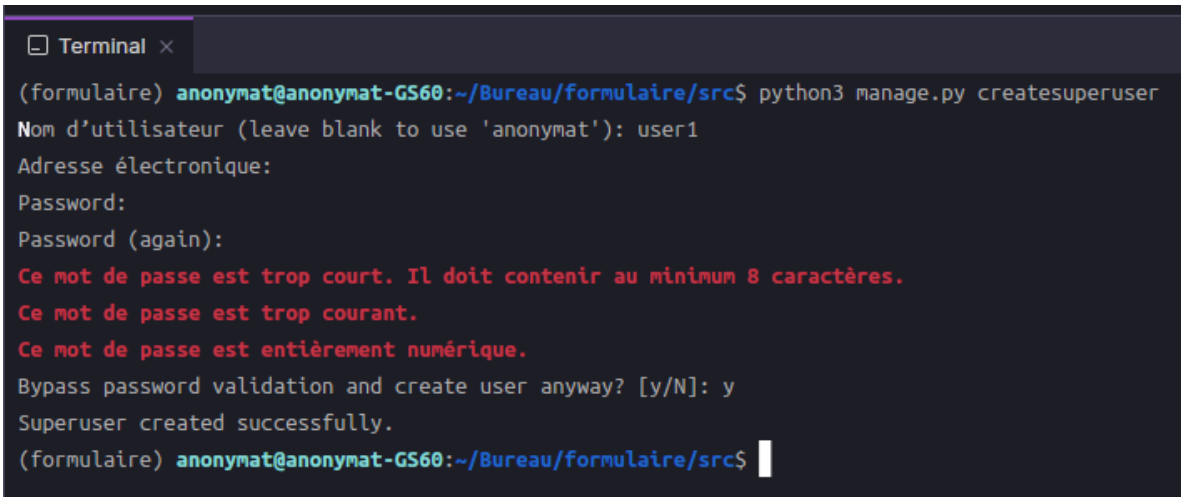

Voila notre **framework django** fonctionne correctement, maintenant nous allons passer à l'étape où nous devons tout relier, c'est à dire relier les champs **Name, Email, Subject** ainsi que **message** pour que **l'utilisateur** puisse saisir ses champs. Puis nous devons créer la partie **admin** pour les développeurs, grâce à cela ça va nous permettre de récolter **les données saisies** par **l'utilisateur**.

Pour la création de l'admin il faut taper les command suivante :

python3 manage.py createsuperuser

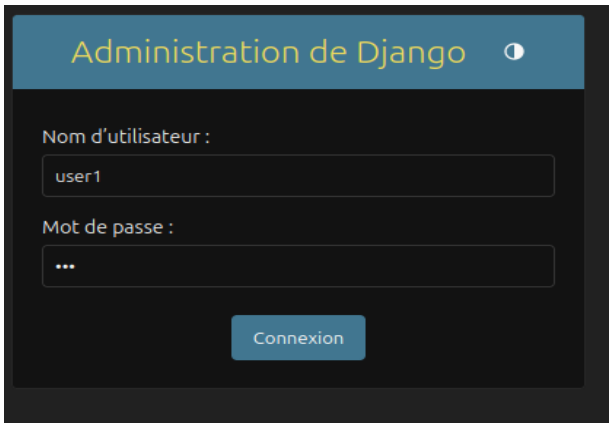
Cela permet de générer un compte avec des privilèges administratifs pour

accéder à l'interface d'administration de Django.



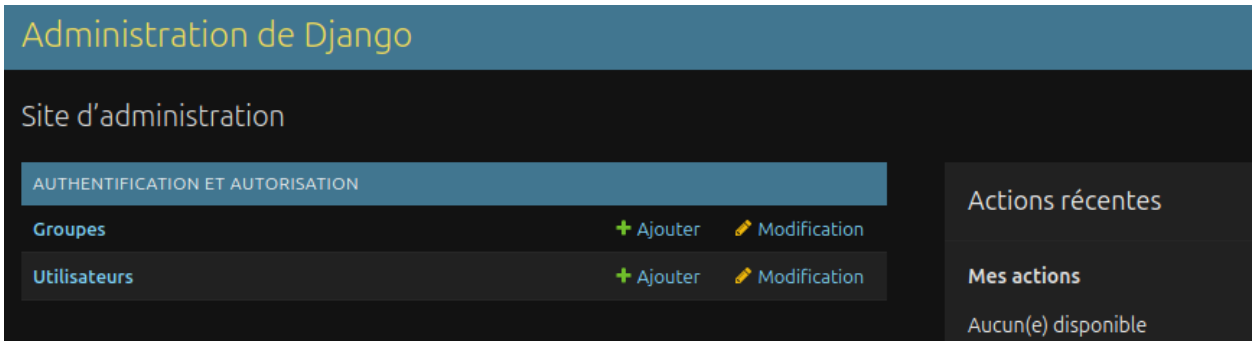

Puis nous devons taper dans l' url de votre adresse IP localhost suivie d'un /admin/.

Par exemple : [http://127.0.0.1:8000/admin/](http://127.0.0.1:8000/admin/)



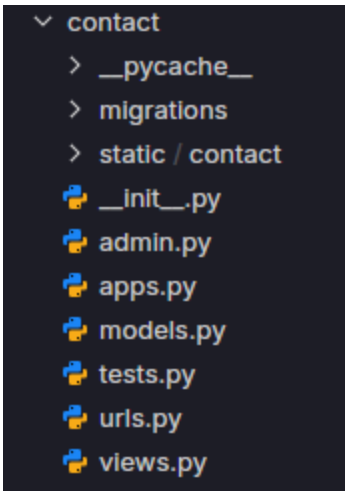

Une fois accéder à cette page vous devez taper les identifiants créés précédemment pour vous connecter en admin.

Une fois connecté vous devez tomber sur une page d'admin :

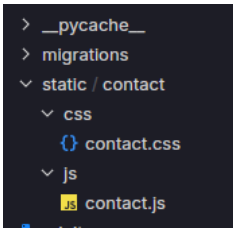



Maintenant nous devons installer une application pour notre formulaire contact

pour cela nous devons taper la commande python3 manage.py startapp contact. Une fois taper dans le terminal de commande nous avons un dossier contact qui s'est créé :




Comme vous pouvez le voir nous avons les même fichier que le dossier formul. Dans le dossier contact nous devons créer un dossier static et à l'intérieur on crée un dossier contact une fois dans contact nous devons créer 2 dossiers nommé js et css. Dans le dossier js on crée un fichier contact.js et dans le dossier css on crée un fichier contact.css. Le fichier js va nous permettre de créer la partie logique de notre formulaire et le css va permettre d'afficher l'animation quand l'utilisateur a bien valider son formulaire.



# Remerciements

Je tiens à exprimer ma profonde gratitude à tous ceux qui ont contribué au succès de mon stage et à mon apprentissage.

Tout d'abord, je remercie chaleureusement Mr Ducrot-Milong Ulrich, pour m'avoir transmis une multitude de connaissances précieuses au cours de ma formation. Son expertise et sa patience ont été déterminantes dans mon développement professionnel. Chaque jour passé à ses côtés a été une véritable source d'inspiration et d'apprentissage.

Enfin, je remercie toute l'équipe Humoov et leur accueil chaleureux et leur soutien tout au long de cette expérience.

# *Conclusion globale*

Ce stage a été une expérience enrichissante et déterminante pour mon développement personnel et professionnel.

D'un point de vue personnel, cette expérience m'a permis de renforcer plusieurs compétences essentielles. J'ai développé une meilleure gestion du temps, une plus grande capacité d'adaptation face à des situations imprévues, et une communication plus efficace avec mes collègues. Travailler dans un environnement professionnel m'a également permis de gagner en confiance et en autonomie.

Sur le plan professionnel, j'ai acquis de solides compétences en développement informatique. J'ai appris à maîtriser de nouveaux outils et méthodes de travail spécifiques à mon domaine. Le partage des connaissances et la guidance m'ont permis de comprendre les subtilités du métier et d'améliorer significativement mes compétences pratiques.

Cette expérience a clarifié et consolidé mes ambitions professionnelles. Je me sens désormais prêt à relever de nouveaux défis et à m'engager dans des projets futurs avec une perspective plus éclairée et une approche plus stratégique.

Ce stage a été une étape cruciale dans mon parcours, me permettant de grandir tant sur le plan personnel que professionnel. Je suis reconnaissant pour les opportunités qui m'ont été offertes et je suis impatient de mettre en pratique tout ce que j'ai appris dans mes futurs projets.