



BURKINA FASO

Unité-Progrès-Justice

Ministère de l'Enseignement Supérieur, Institut Supérieur des Science de la population.



APPLICATION DE GESTION DE SALLE DE CLASSE AVEC DJANGO

MEMBRES DU GROUPE5

BAYILI PEMA

COMPAORE ALEXIS A

SANKARA SAIDOU

ENSEIGNANT :

DR BARA OUSMANE

Table of Contents

INTRODUCTION	3
PREMIERE PARTIE	3
Étude comparative des Frameworks en python	3
I.FLASK	3
I.1. Définition de Flask.....	3
I.2. Comparaison entre Django vs Flask	4
I.3.Avantages de Django.....	4
I.4.Avantages de Flask.....	4
I.5.Cas d'Utilisation.....	5
I.6.Limites de Flask	5
I.7.Autres Framework Python	7
II. Django.....	7
II.1. Avantages :	9
✓ Sécurité	10
II.2. Inconvénients :	10
III.3. Description du Framework CherryPy	10
III.1. Les avantages du Framework CherryPy	13
III.2 Comparaison avec Django et Flask.....	15
DEUXIÈME PARTIE	15
IV. EXPLICATION ET MODE DU DEVELOPPEMENT DE L'APPLICATION GESTION DE SALLE DE CLASSE.	16
Interface de vs code ci-dessus on a trois parties.....	19
1. L'éditeur de code.....	19
2. La barre latérale	19
3. La barre d'activités	20
FONCTIONNEMENT DU SITE WEB	24
Conclusion	29

INTRODUCTION

Python est un langage de programmation très sophistiqué et adapté pour la création d'application de tout genre. Il permet de créer et de gérer des sites web et des applications web utilisable efficacement. Ces applications sont stockées sur des serveurs en ligne. Il faut noter que python utilise des Frameworks très performants les principaux sont DJANGO, FLASK ET CHERRYPY...la liste n'est pas exhaustive. En ce qui concerne notre application de gestion de salle de classe nous avons utiliser le Framework Django pour sa création. Car cela est pratique et facile pour le développement de ce type d'application. L'application permettra à l'utilisateur de connecter à travers un mot de passe ensuite elle permettra d'ajouter des salles de classe, de modifier et de supprimer des salles de classe tout en affichant la date de ces différentes actions dans notre magnifique et sublime application.

PREMIERE PARTIE

Étude comparative des Frameworks en python

Pour une étude comparative encore plus détaillée et adaptée aux besoins spécifiques, examinons les Frameworks Django, Flask, CherryPy et FastAPI en profondeur sur plusieurs aspects critiques : architecture, performances, extensibilité, communauté et écosystème, courbe d'apprentissage, et cas d'utilisation spécifiques.

I.FLASK

I.1. Définition de Flask.

Flask est un microframework web open-source écrit en Python. Il a été développé par Armin Ronacher et d'autres contributeurs de la communauté Python.

Flask est considéré comme un microframework car il n'a pas besoin de bibliothèques particulières pour fonctionner. Il fournit seulement les fonctionnalités de base nécessaires pour construire une application web, laissant le choix aux développeurs d'ajouter les outils et les bibliothèques dont ils ont besoin.

I.2. Comparaison entre Django vs Flask



I.3. Avantages de Django

Complet et Robuste : Django est un Framework web complet qui fournit de nombreuses fonctionnalités prêtes à l'emploi comme la gestion des utilisateurs, le routage, le ORM, etc.

Scalable : Django est conçu pour supporter des applications web de grande envergure avec de nombreuses fonctionnalités.

Communauté Active : Django bénéficie d'une large communauté de développeurs qui contribuent activement au développement du Framework et partagent des ressources.

I.4. Avantages de Flask.

Minimaliste et Léger : Flask est un micro Framework qui offre le strict nécessaire, ce qui le rend plus léger et plus facile à prendre en main.

Flexible : Flask laisse plus de liberté aux développeurs pour choisir les bibliothèques et les outils à utiliser, ce qui le rend plus flexible.

Rapide à Apprendre : Avec moins de fonctionnalités intégrées, Flask est plus simple à apprendre et à mettre en place, surtout pour les débutants.

Flask inclut un système de tests unitaires intégré, facilitant le débogage et le développement d'applications robustes.

I.5.Cas d'Utilisation.

Django est plus adapté pour des applications web complexes et évolutives, comme des sites e-commerce, des réseaux sociaux, etc.

Flask convient mieux pour des projets plus simples et des prototypes, où la flexibilité et la rapidité de développement sont essentielles.

En résumé, Django est un Framework complet et robuste, idéal pour les grands projets, tandis que Flask est un micro Framework plus léger et flexible, mieux adapté aux petits projets et aux prototypes.

I.6.Limites de Flask

Bien que très apprécié pour sa simplicité et sa flexibilité, Flask présente quelques limites :

Fonctionnalités Limitées : En tant que microframework, Flask ne fournit que les fonctionnalités de base. Les développeurs doivent ajouter eux-mêmes les bibliothèques et les outils nécessaires pour construire une application web complète.

Passage à l'Échelle Difficile : Flask est mieux adapté aux petits et moyens projets. Pour des applications web de grande envergure, d'autres Framework plus complets comme Django peuvent être plus appropriés.

Manque de Standardisation : Avec sa grande flexibilité, Flask laisse plus de liberté aux développeurs, ce qui peut parfois mener à un manque de standardisation dans les projets.

Pour pallier ses limites d'autres Frameworks ont été développés c'est le cas de Django CherryPy etc...

Applications Web Simples exemples

Sites Web Statiques : Flask peut être utilisé pour créer des sites web statiques simples, avec des pages HTML statiques.

Blogs et Sites Web de Contenu : Flask permet de créer facilement des blogs et des sites web de contenu dynamique.

Outils Internes d'Entreprise : Flask est souvent utilisé pour développer des outils internes d'entreprise, comme des tableaux de bord, des applications de gestion, etc.

Applications Web Interactives : Bien que Flask ne soit pas aussi complet que Django pour les applications web complexes, il peut être utilisé pour développer des applications web interactives avec des fonctionnalités avancées.

Applications de Visualisation de Données : Flask peut être combiné avec des bibliothèques de visualisation comme Matplotlib ou Plotly pour créer des applications web de visualisation de données.

I.7. Autres Framework Python

Voici quelques autres Framework Python populaires à connaître :

Pyramid : Un Framework web complet et modulaire, avec une bonne documentation et une communauté active.

Tornado : Un Framework asynchrone conçu pour les applications web à forte charge, avec de bonnes performances.

Bottle : Un microframework minimaliste et rapide à prendre en main, similaire à Flask.

FastAPI : Un Framework moderne et performant pour construire des API REST et GraphQL.

Le choix du Framework dépendra des besoins spécifiques de votre projet, de votre niveau d'expertise et des performances requises. Il est important d'évaluer les avantages et les inconvénients de chaque Framework en fonction de vos critères.

II. Django



❖ Architecture :

- ✓ **Full-stack Framework** : Django fournit une solution complète pour le développement web, y compris un ORM, un moteur de Template, un système de routage, et une interface d'administration.
- ✓ **Structure opinionnée** : Django impose une structure stricte de projet, ce qui peut aider à maintenir la cohérence dans des projets de grande envergure.

❖ Performances :

- ✓ **Pas conçu pour l'asynchrone** : Django est principalement synchrone. Bien que des outils comme Django Channels permettent de gérer des applications en temps réel, ce n'est pas aussi performant qu'une solution native asynchrone.
- ✓ **Robustesse** : Malgré cela, Django est capable de gérer des applications lourdes avec un grand nombre de requêtes grâce à des pratiques de mise en cache et de bonnes configurations de serveur.

❖ Extensibilité :

- ✓ **Applications réutilisables** : Django permet de créer des applications réutilisables facilement intégrables dans d'autres projets Django.
- ✓ **Middleware** : Django supporte un système de middleware qui permet de modifier la requête ou la réponse au niveau global.

❖ Communauté et écosystème :

- ✓ **Grande communauté** : Avec une communauté vaste et active, il y a une abondance de plugins, de bibliothèques tierces, et de ressources de support.
- ✓ **Écosystème riche** : De nombreux paquets disponibles pour étendre les fonctionnalités, allant de la gestion des paiements à l'intégration avec des services externes.

❖ Courbe d'apprentissage :

- ✓ **Documentation exhaustive** : Django a une documentation très complète, mais la complexité des fonctionnalités intégrées peut rendre l'apprentissage difficile pour les débutants.
- ✓ **Prise en main** : Pour les développeurs expérimentés, Django offre une structure claire et un ensemble d'outils puissants dès le départ.

❖ Cas d'utilisation spécifiques :

- ✓ **Projets de grande envergure** : Sites de commerce électronique, plateformes de gestion de contenu (CMS), réseaux sociaux, applications de gestion d'entreprise.
- ✓ **Sites nécessitant une interface d'administration** : Gestionnaires de contenu, applications d'administration interne.

II.1. Avantages :

- ✓ **Complexe** : Django est un Framework full-stack, fournissant tout ce dont vous avez besoin pour construire une application web complète. Cela inclut la gestion des utilisateurs, la gestion des formulaires, l'authentification, et plus encore.
- ✓ **ORM intégré** : Django inclut un ORM (Object-Relationnel Mapping) qui simplifie les interactions avec la base de données en permettant de manipuler les données en utilisant des objets Python au lieu de SQL.
- ✓ **Admin intégré** : Django fournit une interface d'administration par défaut pour gérer les modèles de données. Cette interface est extrêmement utile pour les administrateurs de site et les développeurs pour gérer le contenu et les utilisateurs.

✓ Sécurité

Protections intégrées : Django inclut des protections contre les vulnérabilités courantes comme les injections SQL, le cross-site Scripting (XSS), le cross-site request forgery (CSRF) et le clickjacking.

Meilleures pratiques de sécurité : Django encourage les meilleures pratiques de sécurité, comme l'utilisation de mots de passe hachés et salés, l'utilisation de connexions sécurisées (HTTPS), et la validation des entrées utilisateur.

Grande communauté : De nombreux développeurs utilisent Django, ce qui signifie qu'il existe une abondance de ressources, de tutoriels, de plugins et de paquets tiers pour étendre les fonctionnalités de base du Framework.

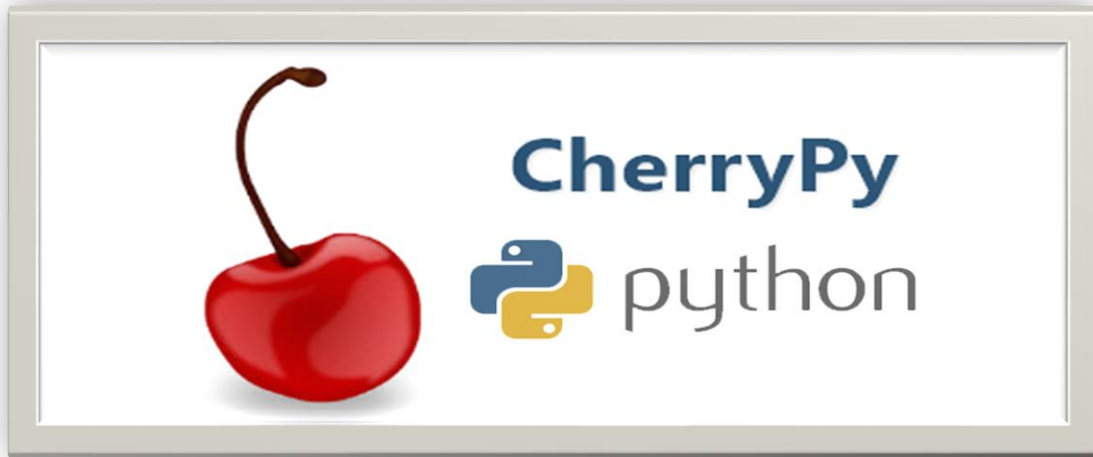
II.2. Inconvénients :

- ✓ **Complexité** : Django peut être trop lourd pour de petites applications ou des API simples. Il inclut beaucoup de fonctionnalités intégrées qui peuvent ne pas être nécessaires pour des projets plus petits.
- ✓ **Courbe d'apprentissage** : La richesse des fonctionnalités et la structure opinionnée de Django peuvent rendre l'apprentissage difficile pour les débutants.

❖ Cas d'utilisation typiques :

- ✓ Applications web complexes avec de nombreuses fonctionnalités.
- ✓ Sites de commerce électronique.
- ✓ Plateformes de gestion de contenu (CMS).
- ✓ Applications nécessitant une interface d'administration robuste.

III.3. Description du Framework CherryPy



CherryPy est un Framework web minimaliste, orientée objet qui a été créée en 2002 en Python. Il fut conçu pour rendre le développement d'applications web aussi simple que l'écriture de scripts. Il se distingue par sa flexibilité, ses performances et sa simplicité. CherryPy utilise une architecture multithreading intégrée, ce qui signifie qu'il peut gérer plusieurs requêtes simultanément en créant un thread (fils d'exécution) distinct pour chaque requête. Cela permet une meilleure utilisation des ressources du serveur et améliore les temps de réponse pour les utilisateurs finaux.

Exemple du code de création d'un serveur web minimal avec CherryPy

```
import cherrypy

class HelloWorld:
    @cherrypy.expose
    def index(self):
        return "Hello, World!"

if __name__ == '__main__':
    cherrypy.quickstart(HelloWorld())
```

En outre CherryPy est un Framework web open source écrit en Python qui permet de créer des applications web à la fois puissantes et flexibles. Il s'agit d'un Framework minimaliste, facile à utiliser et très extensible, qui utilise une approche orientée objet pour la construction de sites web.

Le principal objectif de CherryPy est de fournir un ensemble d'outils pour simplifier le développement d'applications web en Python. Il offre une variété de fonctionnalités, telles que la gestion de requêtes HTTP, la gestion de sessions utilisateur, la gestion de fichiers statiques, la gestion de fichiers téléchargés et bien plus encore. CherryPy permet également d'intégrer facilement des modèles de moteur de rendu de template tels que Jinja2, Mako ou encore Cheetah.

CherryPy est facile à apprendre et à utiliser, car il est basé sur des concepts familiers de la programmation orientée objet. Il utilise également une architecture modulaire, ce qui permet aux développeurs de choisir les composants qu'ils veulent utiliser et de les intégrer facilement à leur application. Le Framework est également très flexible, ce qui permet aux développeurs de

personnaliser facilement leurs applications en fonction de leurs besoins spécifiques.

CherryPy est compatible avec de nombreux serveurs web tels qu'Apache, Lighttpd, Nginx, ou encore le serveur web intégré de Python. Il prend également en charge les protocoles HTTP et HTTPS, ainsi que les cookies et les sessions utilisateur, ce qui le rend très utile pour les applications web nécessitant une authentification utilisateur.

CherryPy est un Framework web puissant et flexible pour Python, offrant une variété de fonctionnalités pour simplifier le développement d'applications web. Il est facile à apprendre et à utiliser, et il est compatible avec de nombreux serveurs web, ce qui en fait un choix idéal pour les développeurs qui cherchent à créer des applications web à la fois robustes et extensibles en Python.

Par ailleurs, **la communauté de CherryPy** est plus petite que celle de Django ou Flask, mais elle est active et fournit un support via des forums et des listes de diffusion.

III.1. Les avantages du Framework CherryPy

CherryPy est un Framework web pour Python qui offre de nombreux avantages aux développeurs qui cherchent à créer des applications web efficaces et extensibles. En effet, nous avons :

- **Facilité d'utilisation :**

CherryPy est un Framework web minimaliste et facile à utiliser qui utilise des concepts familiers de la programmation orientée objet. Les développeurs

peuvent rapidement apprendre à utiliser CherryPy pour créer des applications web de manière efficace.

- **Modularité :**

CherryPy est basé sur une architecture modulaire, ce qui permet aux développeurs de choisir les composants qu'ils souhaitent utiliser pour leur application. Cela permet une plus grande flexibilité et permet de personnaliser facilement les applications selon les besoins spécifiques de chaque projet.

- **Haute performance :**

CherryPy est conçu pour offrir des performances élevées, grâce à une implémentation optimisée des protocoles HTTP et des connexions persistantes. Cela signifie que les applications CherryPy peuvent gérer efficacement de nombreuses requêtes simultanées avec une faible latence.

- **Extensibilité :**

CherryPy peut être facilement étendu avec des plugins et des extensions tiers. Cela permet aux développeurs de bénéficier d'une large gamme de fonctionnalités prêtes à l'emploi pour leurs applications web.

- **Sécurité :**

CherryPy offre des fonctionnalités de sécurité avancées pour protéger les applications web contre les attaques de sécurité courantes telles que les injections SQL et les attaques par script intersites (XSS).

- **Documentation complète :**

CherryPy est livré avec une documentation complète et une communauté active de développeurs. Cela permet aux nouveaux utilisateurs de trouver

rapidement des réponses à leurs questions et de bénéficier de l'expertise de la communauté.

III.2 Comparaison avec Django et Flask

Le Framework CherryPy est similaire à Django et Flask. Cependant, il existe des différences entre eux. En ce qui concerne **la flexibilité**, CherryPy peut être utilisée pour construire des applications de toutes tailles. Par exemple, il peut être utilisé comme un serveur WSGI permettant d'intégrer d'autres applications WSGI ou Framework.

Par ailleurs contrairement à Django, **il n'impose pas une structure de projet stricte**, et contrairement à Flask, il fournit **un serveur http intégré** prêt à l'emploi.

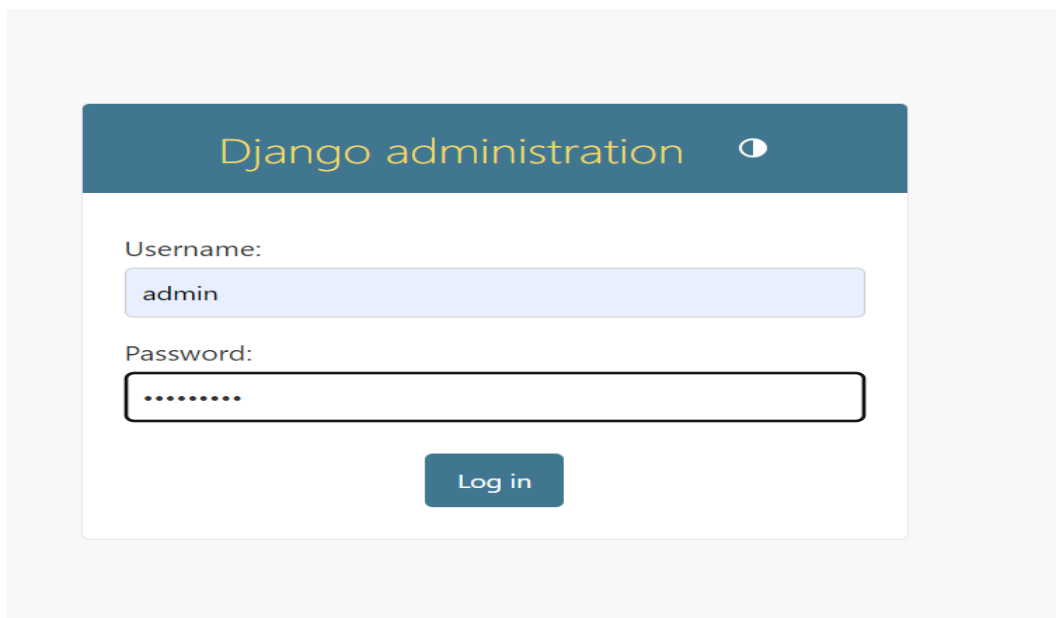
Quant à l'écosystème, celui de de CherryPy est moins vaste que celui de Django ou Flask. Cependant, il offre suffisamment de plugins et d'extensions pour couvrir les besoins courants. En outre, les développeurs peuvent intégrer facilement des outils comme Memcached ou Redis, des ORM comme SQLAlchemy, et d'autres bibliothèques, et d'autres bibliothèques Python.

Concernant la communauté, celle de CherryPy est plus petite que celle de Django ou Flask, mais elle est active et fournit un support via des forums et des listes de diffusion.

DEUXIÈME PARTIE

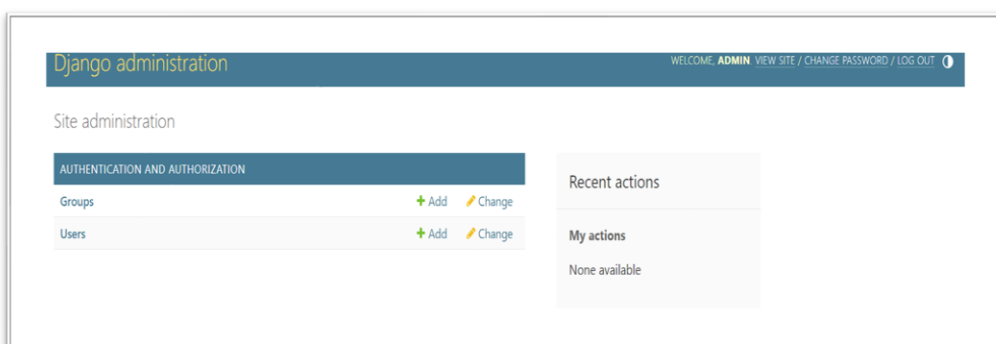
IV. EXPLICATION ET MODE DU DEVELOPPEMENT DE L'APPLICATION GESTION DE SALLE DE CLASSE.

Page de connexion dès qu'on au serveur 127.0.0.1 :8000



The screenshot shows the Django administration login interface. At the top, there is a dark blue header with the text "Django administration" in yellow and a moon icon on the right. Below the header, the form has two input fields: "Username:" with the value "admin" and "Password:" with masked characters. A "Log in" button is positioned below the password field.

Page de connexion dès qu'on au serveur 127.0.0.1 :8000/admin



The screenshot shows the Django administration dashboard after a successful login. The top header is dark blue with "Django administration" on the left and "WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. Below the header, the page is titled "Site administration". There are two main sections: "AUTHENTICATION AND AUTHORIZATION" and "Recent actions". The "AUTHENTICATION AND AUTHORIZATION" section has a table with two rows: "Groups" and "Users", each with "+ Add" and "Change" links. The "Recent actions" section has a "My actions" sub-section with the text "None available".

The screenshot shows the Django administration interface for changing a password. The breadcrumb trail is 'Home > Password change'. The left sidebar shows 'AUTHENTICATION AND AUTHORIZATION' with links to 'Groups' and 'Users', both with '+ Add' buttons. The main content area is titled 'Password change' and contains the instruction: 'Please enter your old password, for security's sake, and then enter your new password twice so we can verify you typed it in correctly.' There are three input fields: 'Old password:', 'New password:', and 'New password confirmation:'. Below the 'New password:' field, there are four lines of password requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.' At the bottom of the form is a button labeled 'CHANGE MY PASSWORD'.

On a la possibilité de changer son mot de passe en cliquant sur « change password »

On a la possibilité d'ajouter un autre utilisateur

The screenshot shows the Django administration interface for adding a new user. The breadcrumb trail is 'Home > Authentication and Authorization > Users > Add user'. The left sidebar shows 'AUTHENTICATION AND AUTHORIZATION' with links to 'Groups' and 'Users', both with '+ Add' buttons. The main content area is titled 'Add user' and contains the instruction: 'First, enter a username and password. Then, you'll be able to edit more user options.' There are three input fields: 'Username:', 'Password:', and 'Password confirmation:'. Below the 'Username:' field, there is a note: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.' Below the 'Password:' field, there are four lines of password requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.' Below the 'Password confirmation:' field, there is a note: 'Enter the same password as before, for verification.' At the bottom of the form are three buttons: 'SAVE', 'Save and add another', and 'Save and continue editing'.

```
System check identified no issues (0 silenced).
July 10, 2024 - 21:00:40
Django version 5.0.7, using settings 'MonprojetDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

```
[10/Jul/2024 21:01:36] "GET / HTTP/1.1" 200 10629
Not Found: /favicon.ico
[10/Jul/2024 21:01:37] "GET /favicon.ico HTTP/1.1" 404 2119
[10/Jul/2024 21:01:47] "GET /admin/ HTTP/1.1" 302 0
```

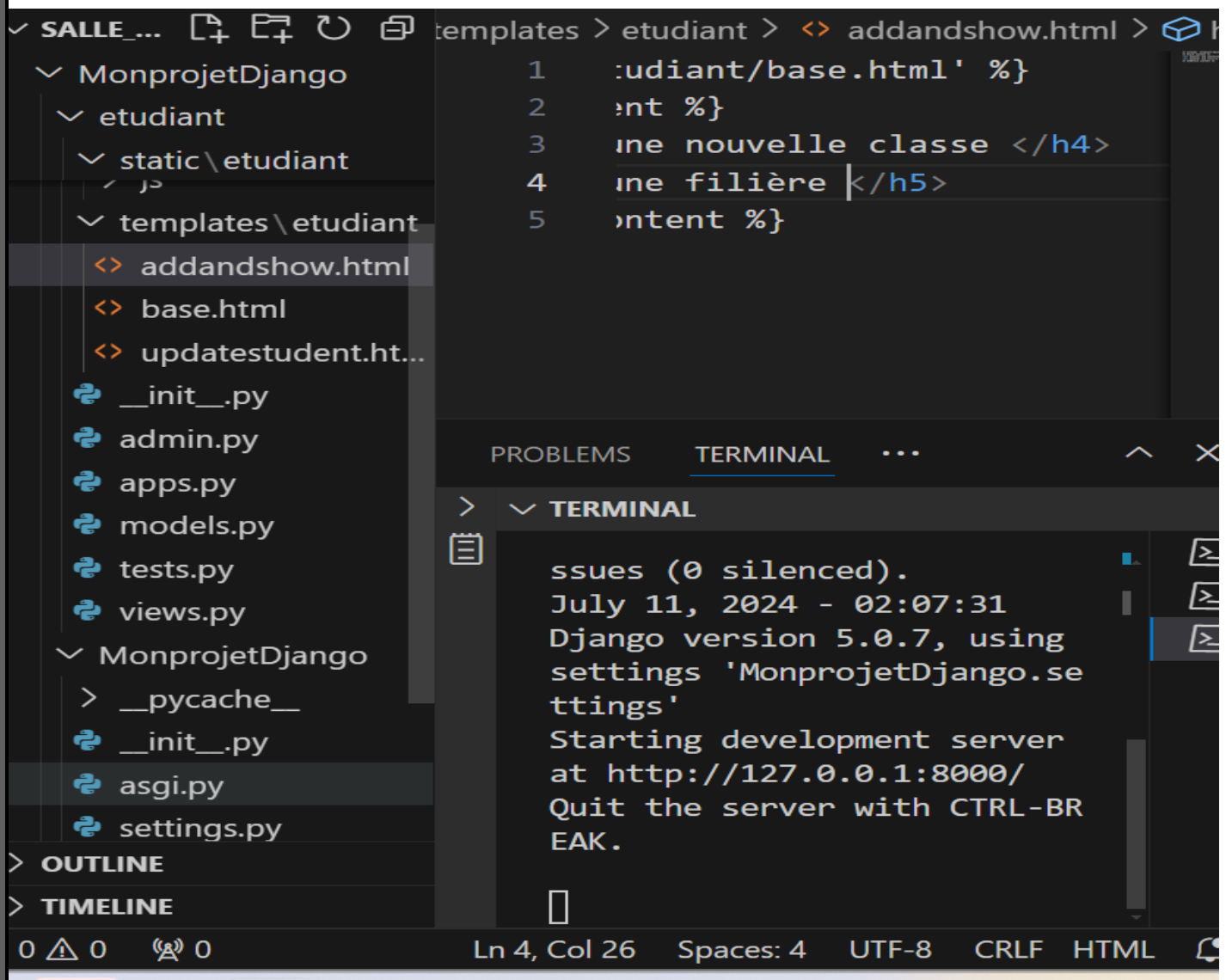
Après avoir exécuter python manage.py runserver. Un lien du serveur apparait et on clique là-dessus.

```
Password (again):
Superuser created successfully.
PS C:\Users\USER\Desktop\SALLE_DE_CLASSE\MonprojetDjango> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 10, 2024 - 21:00:40
Django version 5.0.7, using settings 'MonprojetDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Sqlite3 est le système de gestion de base de données de Django qui lui est associé

Interface de vs code ci-dessus on a trois parties.



1. L'éditeur de code

L'éditeur de code est la partie centrale de VS Code où vous écrivez et modifiez votre code. Il offre des fonctionnalités avancées d'édition comme la coloration syntaxique, l'auto-complétions, la navigation dans le code et la détection d'erreurs. L'éditeur prend en charge de nombreux langages de programmation.

2. La barre latérale

La barre latérale contient différents panneaux qui vous permettent de gérer vos fichiers, vos extensions et vos paramètres. Les principaux panneaux sont :

- Explorateur de fichiers : pour naviguer et gérer vos fichiers et dossiers
- Recherche : pour rechercher et remplacer du texte dans vos fichiers
- Extensions : pour installer, configurer et gérer vos extensions
- Débogage : pour déboguer votre code.

3. La barre d'activités

La barre d'activités se trouve sur le côté gauche et permet d'accéder rapidement aux principaux panneaux de VS Code :

Explorateur : pour accéder à l'explorateur de fichiers

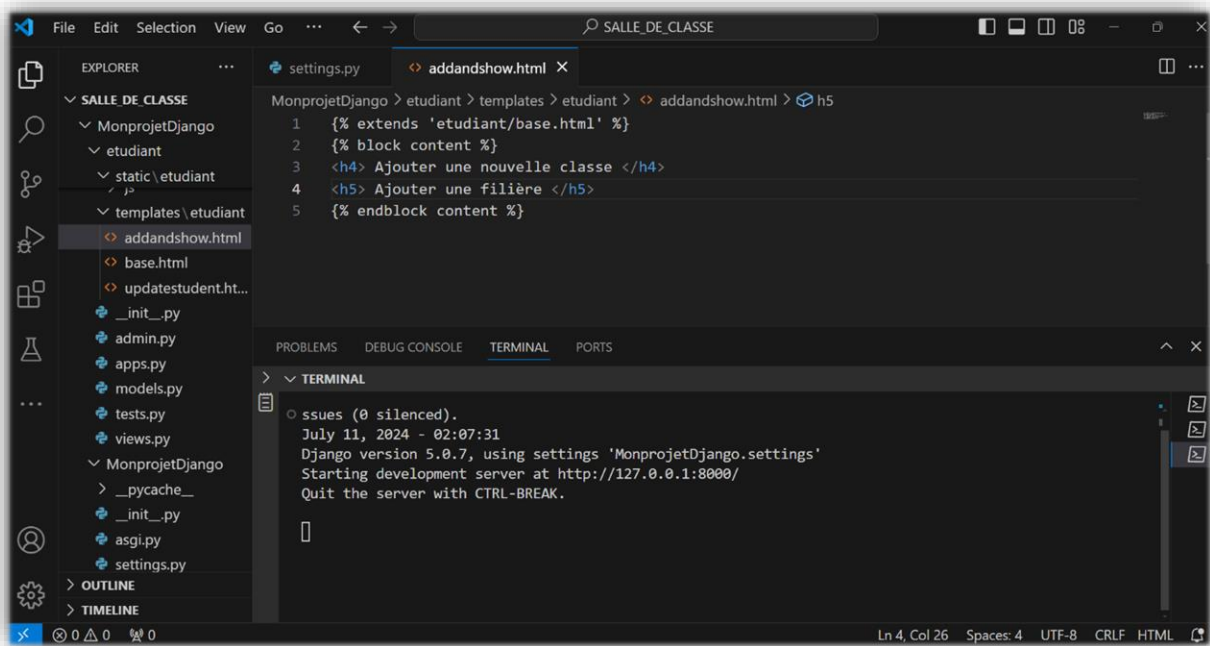
Recherche : pour ouvrir le panneau de recherche

Débogage : pour accéder aux outils de débogage

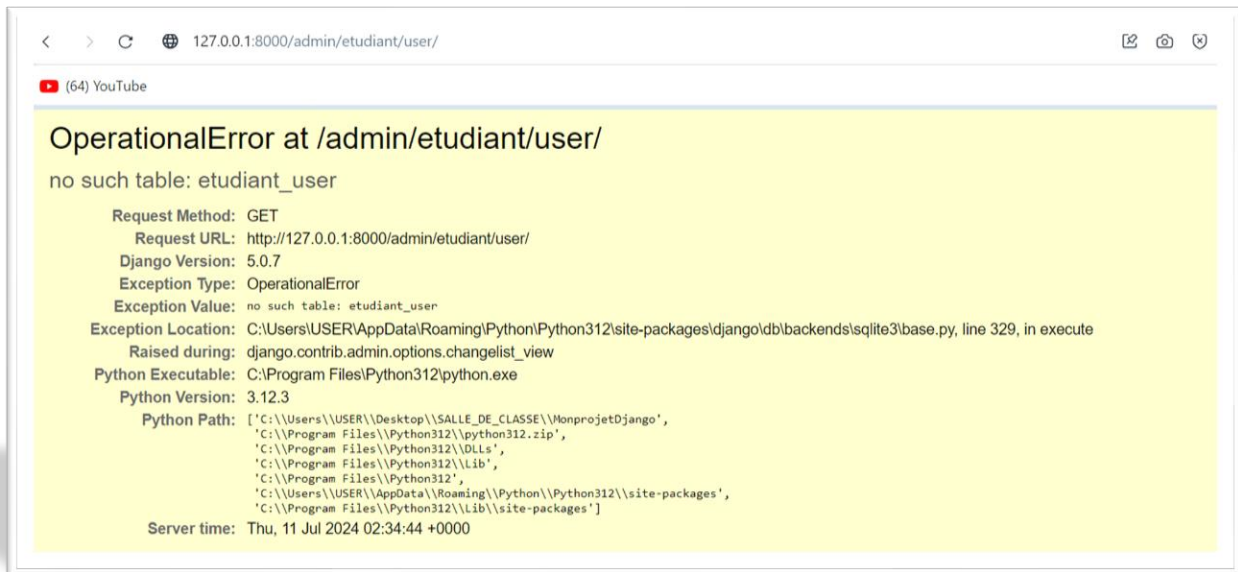
Extensions : pour ouvrir le panneau des extensions

Problèmes : pour afficher les erreurs et les avertissements dans votre code

Compte : pour se connecter à son compte Microsoft



Avant la création des migrations voilà à quoi ressemblait le serveur lorsqu'on essayait de le réactualiser un message nous dit qu'il ne voit pas la table étudiante.



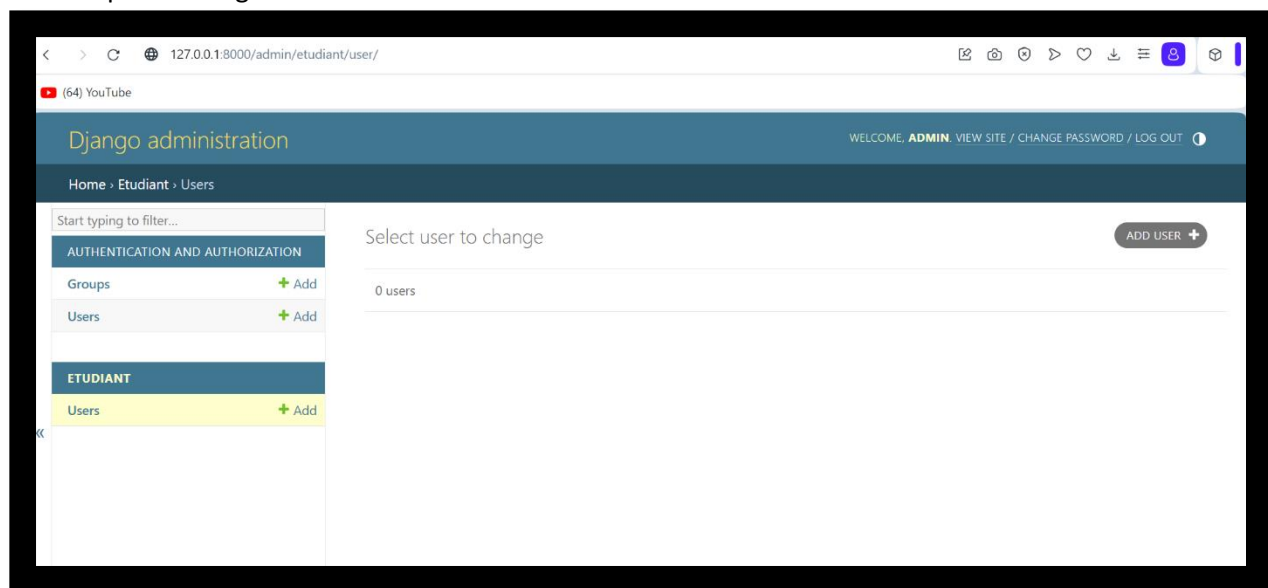
la capture ci-dessus montre une migration des dossiers de migrations vers la base de l'application

```

PS C:\Users\USER\Desktop\SALLE_DE_CLASSE\MonprojetDjango> python manage.py makemigrations
Migrations for 'etudiant':
  etudiant\migrations\0001_initial.py
    - Create model User
PS C:\Users\USER\Desktop\SALLE_DE_CLASSE\MonprojetDjango> python manage.py migration
Unknown command: 'migration'. Did you mean showmigrations?
Type 'manage.py help' for usage.
PS C:\Users\USER\Desktop\SALLE_DE_CLASSE\MonprojetDjango> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, etudiant, sessions
Running migrations:
  Applying etudiant.0001_initial... OK
PS C:\Users\USER\Desktop\SALLE_DE_CLASSE\MonprojetDjango>

```

la capture ci-dessus nous montre qu'on a bien configuré et réactualisé notre serveur d'application après les migrations.



Il faut préciser que le fichier template ne prennent que des fichier html.

Nous allons par la suite créer un fichier « forms.py » dans le fichier « étudiant qui est le nom de notre application ».

Ci-dessus la partie password input permet de sécuriser en rendant le mot de de passe crypté.

```

MonprojetDjango > etudiant > forms.py > StudentRegistration > Meta
>
6 class StudentRegistration(forms.ModelForm):
7     class Meta:
8         model = User
9         fields = ['name', 'email', 'password']
10        widgets= {
11            'name': forms.TextInput(attrs={'class': 'form-control'}),
12            'email': forms.EmailInput(attrs={'class': 'form-control'}),
13            'password': forms.PasswordInput(attrs={'class': 'form-control'}),
14        }

```

La partie mot de passe est maintenant invisible à la saisie ce qui permet de renforcer la sécurité de notre application.

Ajouter une nouvelle classe

Name:

Email:

Password:

FONCTIONNEMENT DU SITE WEB

Pour faire fonctionner le site, il faut au préalable fixer le chemin de travail avant d'exécuter la commande « `python manage.py runserver` ». Le lien du site s'affiche donc et nous conduit vers la page d'accueil du site. Mais avant, pour un affichage complet du site, il est recommandé d'utiliser une connexion internet.


```

Invite de commandes - pytho x + v
Microsoft Windows [version 10.0.22631.3880]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\USER>cd "C:\Users\USER\Desktop\Projet_Djan"

C:\Users\USER\Desktop\Projet_Djan>MonProjDjango
'MonProjDjango' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

C:\Users\USER\Desktop\Projet_Djan>cd MonProjDjango

C:\Users\USER\Desktop\Projet_Djan\MonprojDjango>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
?: (staticfiles.W004) The directory 'C:\Users\USER\Desktop\Projet_Djan\MonprojDjango\static' in the STATICFILES_DIRS set
ting does not exist.

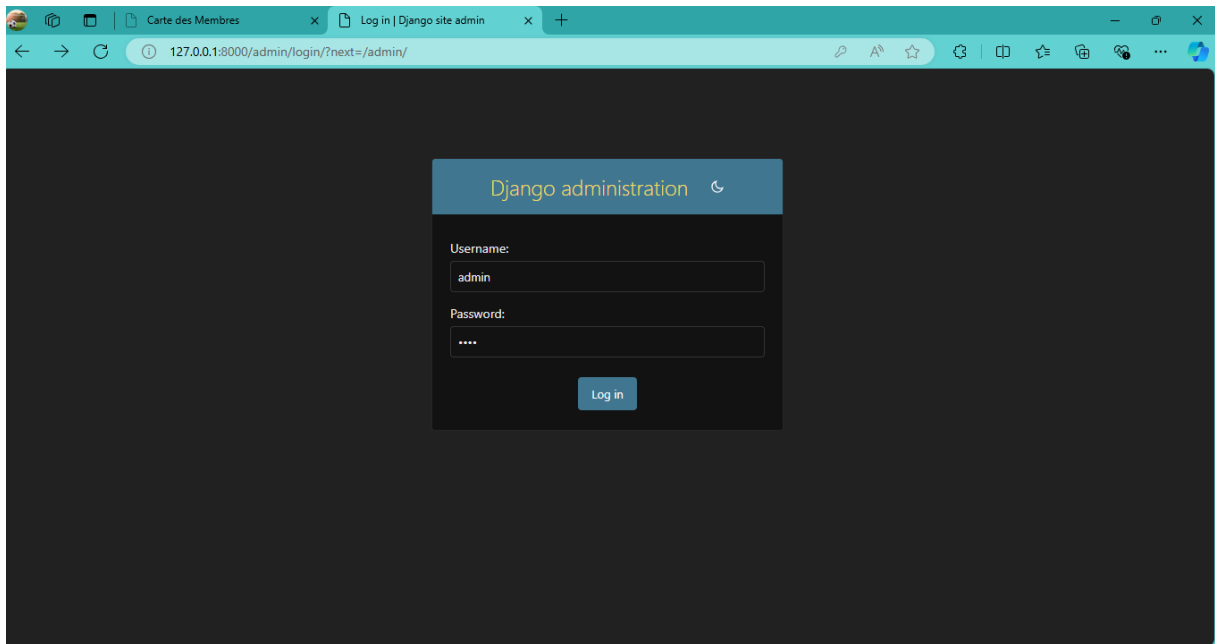
System check identified 1 issue (0 silenced).
July 13, 2024 - 14:05:07
Django version 5.0.6, using settings 'MonprojDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

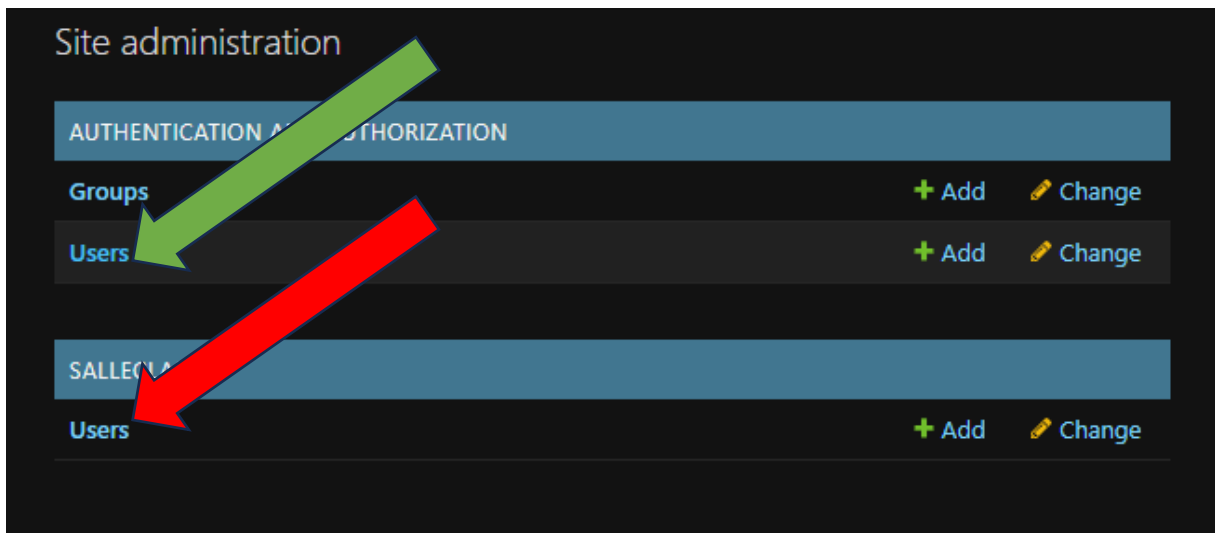
En étant sur la page d'accueil du site, nous pouvons visualiser la liste des salles de classe disponible dans notre base qui s'actualise automatique lors de l'ajout ou de la suppression de salle de classe. Nous pouvons parcourir les autres onglets About, Contact ou encore (le plus important d'ailleurs) l'onglet Espace administrateur pour accéder à la partie backend réservée aux administrateurs où il est possible de créer, modifier ou supprimer des salles de classes.

ID	Nom	Date	Code de la classe	Prof Principal
21	MPSE1	24-02-2016	SQD36	Laurent Gourou
22	MPSE3	12-01-2023	1234	BVNNBNFG
23	LPAS1	12-01-2014	5468	BASSINGA Hervé
24	ISE1	23-03-2024	3659	Réné ZEMBA
25	MPSE 2	13-07-2024	6987mpse	SANKARA Saïdou

L'accès a la partie backend est conditionnée par l'entrée du nom d'utilisateur (admin) et du mot de passe (1234)



Cette image nous montre la partie backend. Ici nous pouvons accéder aux salles de classe (flèche rouge) et à la liste des administrateurs (flèche verte)



Cette image montre la partie qui permet de modifier, ajouter ou supprimer les salles de classe. Pour modifier ou supprimer, il faut cliquer sur l'identifiant (flèche orange). Pour, ajouter des salles, il faut cliquer sur add (flèche bleue)

Home > Salleclasse > Users

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SALLECLASSE

Users + Add

Select user to change ADD USER +

Action: Go 0 of 5 selected

ID	NAME	DATE	PASSWORD	PRINCIPAL
25	MPSE 2	13-07-2024	6987mpse	SANKARA Saidou
24	ISE1	23-03-2024	3659	Réné ZEMBA
23	LPAS1	12-01-2014	5468	BASSINGA Hervé
22	MPSE3	12-01-2023	1234	BVNNBNFG
21	MPSE1	24-02-2016	SQD36	Laurent Gourou

5 users

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SALLECLASSE

Users + Add

Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username: ?
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:
Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:
Enter the same password as before, for verification.

SAVE Save and add another Save and continue editing

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SALLECLASSE

Users + Add

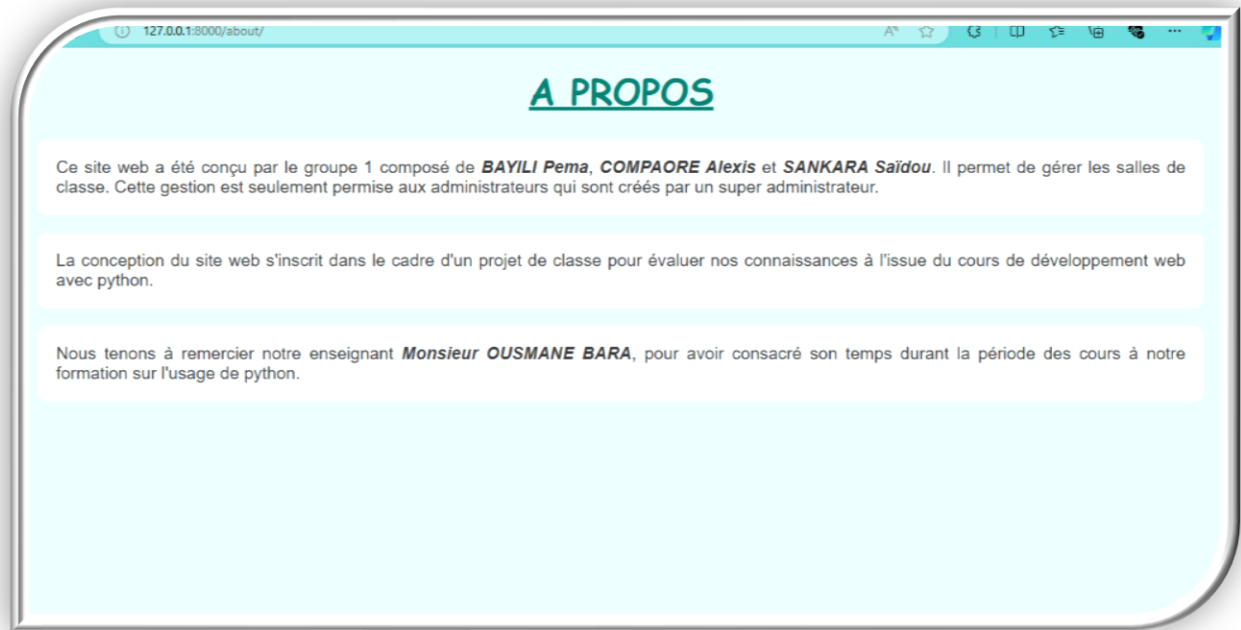
Select user to change ADD USER +

Action: Go 2 of 5 selected

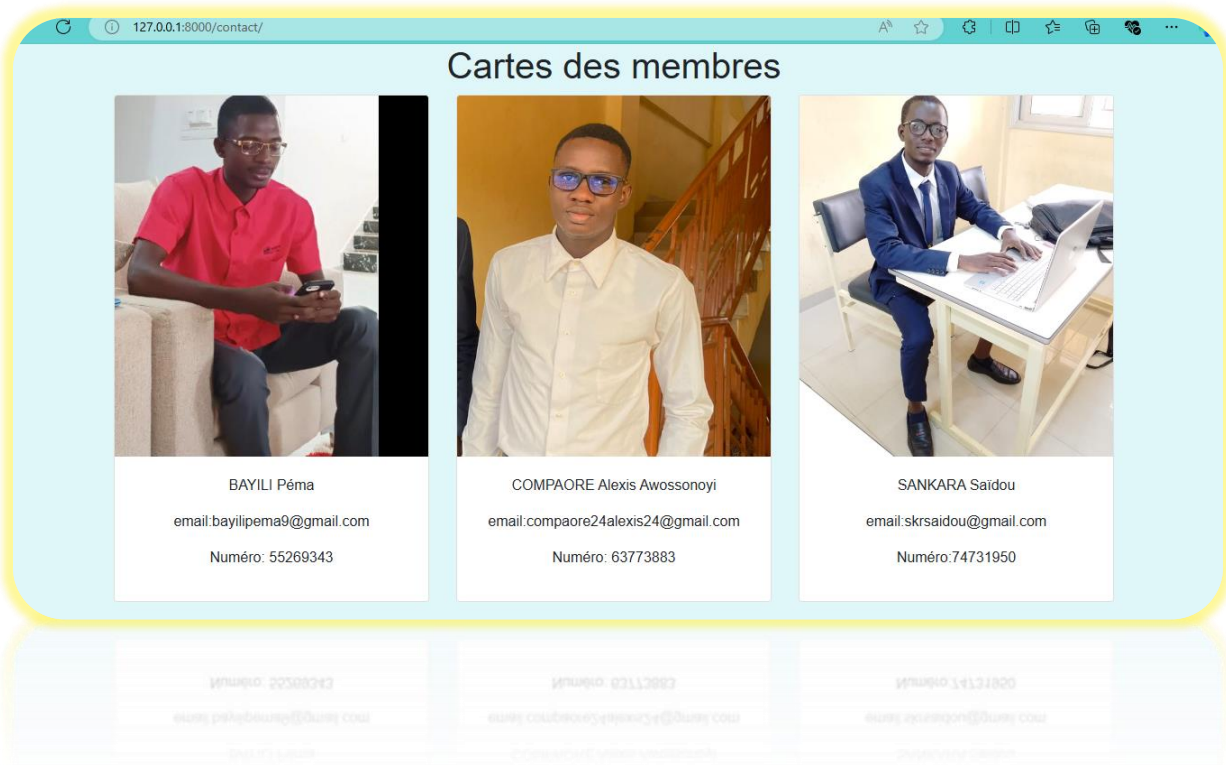
ID	NAME	DATE	PASSWORD	PRINCIPAL
<input checked="" type="checkbox"/> 25	MPSE 2	13-07-2024	6987mpse	SANKARA Saidou
<input checked="" type="checkbox"/> 24	ISE1	23-03-2024	3659	Réné ZEMBA
<input type="checkbox"/> 23	LPAS1	12-01-2014	5468	BASSINGA Hervé
<input type="checkbox"/> 22	MPSE3	12-01-2023	1234	BVNNBNFG
<input type="checkbox"/> 21	MPSE1	24-02-2016	SQD36	Laurent Gourou

5 users

En cliquant sur l'onglet à propos, nous arrivons sur la page dédiée à la contextualisation de la page et à la partie des remerciements du professeur



L'onglet contact permet d'afficher les images et les contacts des membres du groupe, concepteurs du site web.



Conclusion

En épilogue, l'application Django de gestion de salle classe nous a été d'une très grande importance car elle nous a permis de nous familiariser avec le langage python et plus précisément le Framework. L'application gestion de salle est une application très utile pouvant permettre de répondre aux soucis de gestion de salle dans les écoles et universités pour la programmation efficace des cours. Sa manipulation est très aisée et facile d'accès pour tout utilisateur.