

**MINISTRE DE L'ENSEIGNEMENT
SUPERIEUR DE LA RECHERCHE
SCIENTIFIQUE ET DE L'INNOVATION**

**BURKINA FASO
UNITE-PROGRES-JUSTICE**

**UNIVERSITE JOZEPH KI-ZERBO
INSTITUT SUPERIEUR DES SCIENCES DE LA POPULATION
LICENCE PROFESSIONNELLE EN ANALYSE STATISTIQUE**



Année universitaire 2023-2024

**ETUDE COMPARATIVE DES FRAMEWORK WEB POUR LE
DEVELOPPEMENT D'APPLICATIONS WEB EN PYTHON**

Réalisé par groupe2:

Enseignant :

Bationo Martinien

Dr BARA Ousmane

Cissé Oumarou

ZANGRE Madi

Table des matières

Introduction	3
LA PREMIERE PARTIE	3
I. Étude Comparative des Framework Web Python	3
Pour Django	3
Pour Pyramide	6
Pour Flask	8
Exigences Répondant aux Besoins Communes	8
Avantages de Flask	8
Pour CherryPy	10
Exigences Répondant aux Besoins Communes	10
Pour TurboGears	12
Exigences Répondant aux Besoins Communes	12
Avantages de TurboGears	13
LA DEUXIEME PARTIE	16
II. II. Pratique : Création d'une Application Web avec Django pour la gestion des salles de classe	16
Conclusion	20

Introduction

De nos jours avec l'évolution rapide du développement web, choisir le bon Framework est essentiel pour créer des applications web efficaces, robustes et évolutives. Les Framework web en Python sont particulièrement populaires en raison de la simplicité et de la lisibilité de ce langage, ainsi que de son écosystème riche en bibliothèques. Il existe donc un grand nombre de Framework Web Python disponibles. Ils ont tendance à se distinguer par leurs objectifs, leur structure, et/ou leur fonction principale. Ainsi l'utilisation des Framework prend en compte la taille et la complexité du projet. Lorsqu'on cherche à développer un système de grande taille, riche en fonctionnalités et en exigences, un Framework complet pourrait être le bon choix. Lorsque l'application est plus petite et plus simple, vous devriez probablement envisager un micro Framework. Un Framework complet est une solution tout-en-un avec des bibliothèques configurées pour travailler ensemble de manière transparente. Il prend en charge le développement de services dorsaux, d'interfaces frontales et de bases de données.

LA PREMIERE PARTIE

I. Étude Comparative des Framework Web Python



Pour Django

Date de sortie : En 2005, Django est devenu l'un des Framework web les plus populaires pour le développement d'applications web en Python. Il propose une architecture MVC (Modèle-Vue-Contrôleur) robuste et suit les principes du

développement rapide d'applications (RAD). L'objectif de Django est de permettre de concevoir des applications web complètes.

Exigences Répondant aux Besoins Communes

Création d'utilisateurs avec mot de passe : Django propose un système d'authentification intégré qui facilite la création d'utilisateurs avec mot de passe. Il permet la gestion des utilisateurs, des groupes et des autorisations directement dans Django.

Création, modification et suppression de cours :

Création d'un modèle de données pour représenter les cours, y compris le titre du cours et la date de publication.

Création de vues (fonctions ou classes) pour afficher, modifier et supprimer ces cours.

Django fournit des formulaires pour faciliter la validation des données entrées par les utilisateurs.

En résumé ces caractéristiques font de Django un outil très évolutif, vraiment rapide et polyvalent.

Avantages de Django

ORM (Object-Relational Mapping) :

Django utilise son ORM pour mapper des objets sur des tables de base de données. Le même code fonctionne avec différentes bases de données et peut être transféré d'une base de données à l'autre. Principales bases de données prises en charge : PostgreSQL, MySQL, SQLite et Oracle, avec des pilotes tiers pour d'autres bases de données.

Flexibilité et Évolutivité :

Avec Django, vous pouvez créer n'importe quelle application web, du projet à petite échelle au site web complexe. Grâce à sa flexibilité, Django est également utilisé pour la création de MVP (Minimum Viable Product), permettant aux start-ups d'optimiser leur temps et leur budget.

Nous allons résumer les caractéristiques clés de Django dans un tableau

Tableau 1 : les caractéristiques clés de Django

Caractéristique	Description
Architecture MVT (Model-View-Template)	Sépare les préoccupations en trois composants : modèle, vue et template, facilitant le développement et la maintenance.
ORM (Object-Relational Mapping)	Facilite les interactions avec la base de données en mappant les modèles Python aux tables de la base de données.
Administration Automatique	Génère automatiquement une interface d'administration basée sur les modèles définis, permettant une gestion facile des utilisateurs, permissions et données.
Sécurité Intégrée	Protection contre les attaques courantes comme CSRF, XSS et SQL Injection. Gère l'authentification et les autorisations des utilisateurs.
Système de Templates	Permet la séparation du code HTML et de la logique Python, offrant des balises et des filtres pour manipuler les données affichées.
Formulaires	Offre des formulaires pour faciliter la validation des données entrées par les utilisateurs.
Support Multibase de Données	Supporte plusieurs SGBD (PostgreSQL, MySQL, SQLite, Oracle) et permet le transfert du code d'une base de données à une autre sans modifications majeures.
Moteur de Routage des URLs	Permet de définir des URLs propres et lisibles, mappant les URLs aux vues correspondantes.
Flexibilité et Évolutivité	Capable de gérer des projets à petite échelle jusqu'aux sites web complexes, et adapté pour la création de MVPs (Minimum Viable Products).
Communauté Active et Documentation	Bénéficie d'une communauté active et d'une documentation complète, facilitant la résolution des problèmes et l'apprentissage.
Extensions et Plug-ins	Large éventail d'extensions et de plug-ins disponibles pour ajouter des fonctionnalités supplémentaires sans avoir à les développer à partir de zéro.
Support de l'Internationalisation (i18n)	Facilite la création d'applications multi-langues grâce à ses outils intégrés pour l'internationalisation et la localisation.

Pour Pyramid



Pyramid

Pyramid est un framework web léger et flexible pour Python, sorti initialement en 2010. Conçu pour être simple à utiliser et à comprendre, Pyramid suit le principe de minimalisme avec la possibilité d'ajouter des fonctionnalités au fur et à mesure des besoins. Pyramid permet aux développeurs de créer des applications web robustes et évolutives avec une grande liberté architecturale.

Exigences Répondant aux Besoins Communes

Création d'utilisateurs avec mot de passe : Pyramid ne propose pas un système d'authentification intégré comme Django, mais il permet l'intégration facile de bibliothèques externes comme `AuthTktAuthenticationPolicy` pour gérer l'authentification des utilisateurs avec mot de passe. Il est flexible pour utiliser divers systèmes d'authentification et de gestion des utilisateurs.

Création, modification et suppression de cours : Avec Pyramid, on peut créer des modèles de données en utilisant des bibliothèques comme `SQLAlchemy` pour représenter les cours, y compris le titre du cours et la date de publication. Il est possible de faire la création de vues (fonctions ou classes) pour afficher, modifier et supprimer ces cours. Il y'a également l'utilisation de formulaires et de validation de données à l'aide de bibliothèques comme `Deform`.

Avantages de Pyramid

Flexibilité et Minimalisme :

Pyramid suit une approche minimaliste, permettant aux développeurs d'ajouter seulement les composants nécessaires à leur projet. Il a une grande liberté architecturale pour structurer les applications web selon les besoins spécifiques.

Support de Diverses Bases de Données :

Pyramid ne vient pas avec un ORM intégré, mais il s'intègre facilement avec des ORM populaires comme SQLAlchemy. Support pour diverses bases de données via les ORM compatibles.

Scalabilité et Performances :

Conçu pour être performant et capable de gérer des applications web de grande envergure. Il est adaptable pour les petites applications ainsi que pour les projets complexes.

Tableau 2 : les caractéristiques clés de Pyramid

Caractéristique	Description
Flexibilité Architecturale	Permet une grande liberté dans la structure des applications, en choisissant les composants et les bibliothèques nécessaires.
Minimalisme	Suit une approche minimaliste, fournissant un framework léger auquel on peut ajouter des fonctionnalités au besoin.
Support des ORM	Compatible avec des ORM populaires comme SQLAlchemy pour gérer les interactions avec les bases de données.
Sécurité	Offre des politiques d'authentification et d'autorisation flexibles, intégrant facilement des bibliothèques externes pour renforcer la sécurité.
Moteur de Templates	Compatible avec plusieurs moteurs de templates comme Jinja2 et Chameleon, offrant une grande flexibilité dans le choix des outils de présentation.
Routage Sophistiqué	Utilise un système de routage flexible pour mapper les URLs aux vues correspondantes, avec un support pour les vues basées sur les classes et les fonctions.
Support des Transactions	Intégration avec les gestionnaires de transactions pour assurer la cohérence et la fiabilité des opérations de base de données.
Modularité	Permet de structurer les applications en modules réutilisables, facilitant la maintenance et l'évolutivité.
Documentation Complète	Bénéficie d'une documentation complète et de haute qualité, facilitant l'apprentissage et la mise en œuvre des meilleures pratiques.
Communauté Active	Supporté par une communauté active et croissante, offrant des extensions et des solutions aux problèmes courants rencontrés par les développeurs.
Support de l'Internationalisation (i18n)	Facilite la création d'applications multi-langues grâce à ses outils intégrés pour l'internationalisation et la localisation.
Performance	Conçu pour être performant et capable de gérer des applications web de grande envergure, tout en étant adapté pour des projets plus modestes.

Pour Flask



Flask est un micro-framework web pour Python, sorti initialement en 2010. Il est conçu pour être léger, simple à utiliser et extensible. Flask suit une philosophie minimaliste, permettant aux développeurs de choisir et d'intégrer uniquement les composants dont ils ont besoin. Grâce à sa flexibilité et à sa simplicité, Flask est idéal pour les projets de petite à moyenne envergure et pour les développeurs qui préfèrent une approche plus personnalisée du développement web.

Exigences Répondant aux Besoins Communes

Création d'utilisateurs avec mot de passe : Flask n'intègre pas de système d'authentification par défaut, mais il permet l'utilisation de bibliothèques externes comme `Flask-Login` pour gérer l'authentification des utilisateurs avec mot de passe.

Flexibilité pour intégrer divers systèmes d'authentification et de gestion des utilisateurs.

Création, modification et suppression de cours :

Avec Flask, on peut créer des modèles de données en utilisant des bibliothèques comme `SQLAlchemy` pour représenter les cours, y compris le titre du cours et la date de publication.

Création de vues (fonctions ou classes) pour afficher, modifier et supprimer ces cours.

Utilisation de formulaires et de validation de données à l'aide de bibliothèques comme `WTForms`.

Avantages de Flask

Flexibilité et Minimalisme :

Flask suit une approche minimaliste, permettant aux développeurs d'ajouter seulement les composants nécessaires à leur projet.

Grande liberté architecturale pour structurer les applications web selon les besoins spécifiques.

Support de Diverses Bases de Données :Flask ne vient pas avec un ORM intégré, mais il s'intègre facilement avec des ORM populaires comme SQLAlchemy.Support pour diverses bases de données via les ORM compatibles.

Extensibilité et Personnalisation :Permet d'ajouter des extensions pour des fonctionnalités supplémentaires sans complexifier la base du framework.Large écosystème d'extensions disponibles pour répondre à divers besoins.

Tableau 3 : les caractéristiques clés de Flask

caractéristiques	Description
Flexibilité Architecturale	Permet une grande liberté dans la structure des applications, en choisissant les composants et les bibliothèques nécessaires.
Minimalisme	Suit une approche minimaliste, fournissant un framework léger auquel on peut ajouter des fonctionnalités au besoin.
Support des ORM	Compatible avec des ORM populaires comme SQLAlchemy pour gérer les interactions avec les bases de données.
Sécurité	Offre des politiques d'authentification et d'autorisation flexibles, intégrant facilement des bibliothèques externes pour renforcer la sécurité.
Moteur de Templates	Utilise Jinja2 comme moteur de templates par défaut, offrant une grande flexibilité et des fonctionnalités puissantes pour la présentation des données.
Routage Simplifié	Utilise un système de routage simple et intuitif pour mapper les URLs aux vues correspondantes.
Modularité	Permet de structurer les applications en modules réutilisables, facilitant la maintenance et l'évolutivité.
Documentation Complète	Bénéficie d'une documentation complète et de haute qualité, facilitant l'apprentissage et la mise en œuvre des meilleures pratiques.
Communauté Active	Supporté par une communauté active et croissante, offrant des extensions et des solutions aux problèmes courants rencontrés par les développeurs.
Support de l'Internationalisation (i18n)	Facilite la création d'applications multi-langues grâce à ses outils intégrés pour l'internationalisation et la localisation.
Performance	Conçu pour être performant et capable de gérer des applications web de grande envergure, tout en étant adapté pour des projets plus modestes.
Écosystème d'Extensions	Large écosystème d'extensions disponibles pour ajouter des fonctionnalités supplémentaires sans complexifier la base du framework.

Pour CherryPy



CherryPy

A Minimalist Python Web Framework

CherryPy est un framework web minimaliste pour Python, sorti initialement en 2002. Il permet de construire des applications web de manière simple et rapide en utilisant des techniques orientées objet. CherryPy est un framework léger qui peut être utilisé pour développer des applications web robustes et performantes sans imposer de structure particulière.

Exigences Répondant aux Besoins Communes

Création d'utilisateurs avec mot de passe : CherryPy ne propose pas de système d'authentification intégré, mais il permet l'utilisation de bibliothèques externes pour gérer l'authentification des utilisateurs avec mot de passe. Flexibilité pour intégrer divers systèmes d'authentification et de gestion des utilisateurs.

Création, modification et suppression de cours :

- CherryPy, on peut créer des modèles de données en utilisant des bibliothèques comme SQLAlchemy pour représenter les cours, y compris le titre du cours et la date de publication.
- Création de vues (fonctions ou classes) pour afficher, modifier et supprimer ces cours. Utilisation de formulaires et de validation de données à l'aide de bibliothèques externes.

Flexibilité et Minimalisme :

- CherryPy suit une approche minimaliste, permettant aux développeurs d'ajouter seulement les composants nécessaires à leur projet.
- Grande liberté architecturale pour structurer les applications web selon les besoins spécifiques.

Support de Diverses Bases de Données :

- CherryPy ne vient pas avec un ORM intégré, mais il s'intègre facilement avec des ORM populaires comme SQLAlchemy.
- Support pour diverses bases de données via les ORM compatibles.

Extensibilité et Personnalisation :

- Permet d'ajouter des extensions pour des fonctionnalités supplémentaires sans complexifier la base du framework.
- Large écosystème d'extensions disponibles pour répondre à divers besoins.

Tableau 4 :Caractéristiques Clés de CherryPy

Caractéristique	Description
Flexibilité Architecturale	Permet une grande liberté dans la structure des applications, en choisissant les composants et les bibliothèques nécessaires.
Minimalisme	Suit une approche minimaliste, fournissant un framework léger auquel on peut ajouter des fonctionnalités au besoin.
Support des ORM	Compatible avec des ORM populaires comme SQLAlchemy pour gérer les interactions avec les bases de données.
Sécurité	Offre des politiques d'authentification et d'autorisation flexibles, intégrant facilement des bibliothèques externes pour renforcer la sécurité.
Moteur de Templates	CherryPy ne vient pas avec un moteur de templates intégré, mais il s'intègre facilement avec des moteurs de templates comme Jinja2 et Mako.
Routage Simplifié	Utilise un système de routage simple et intuitif pour mapper les URLs aux vues correspondantes.
Modularité	Permet de structurer les applications en modules réutilisables, facilitant la maintenance et l'évolutivité.
Documentation Complète	Bénéficie d'une documentation complète et de haute qualité, facilitant l'apprentissage et la mise en œuvre des meilleures pratiques.
Communauté Active	Supporté par une communauté active et croissante, offrant des extensions et des solutions aux problèmes courants rencontrés par les développeurs.
Support de l'Internationalisation (i18n)	Facilite la création d'applications multi-langues grâce à ses outils intégrés pour l'internationalisation et la localisation.
Performance	Conçu pour être performant et capable de gérer des applications web de grande envergure, tout en étant adapté pour des projets plus modestes.
Écosystème d'Extensions	Large écosystème d'extensions disponibles pour ajouter des fonctionnalités supplémentaires sans complexifier la base du framework.

Pour TurboGears



TurboGears est un framework web full-stack pour Python, lancé initialement en 2005. Il combine les meilleurs éléments de plusieurs frameworks pour fournir une solution complète de développement web. TurboGears permet de développer des applications web de manière rapide et efficace, tout en offrant une grande flexibilité pour s'adapter à des besoins variés.

Exigences Répondant aux Besoins Communes

Création d'utilisateurs avec mot de passe :

- TurboGears intègre `repoze.what` et `repoze.who` pour la gestion de l'authentification et des autorisations.
- Offre une solution complète pour créer des utilisateurs avec mot de passe, gérer les sessions et les autorisations.

Création, modification et suppression de cours :

- TurboGears utilise SQLAlchemy comme ORM par défaut, permettant de créer des modèles de données pour représenter les cours, y compris le titre du cours et la date de publication.
- Fournit des vues basées sur des contrôleurs pour afficher, modifier et supprimer ces cours.
- Utilisation de `ToscaWidgets` pour les formulaires et la validation des données.

Ces caractéristiques font de TurboGears un outil puissant et polyvalent pour le développement d'applications web.

Avantages de TurboGears

Full-Stack et Modulaire :

- TurboGears peut être utilisé comme un framework full-stack ou un micro-framework, en fonction des besoins du projet.
- Intégration de composants comme SQLAlchemy, ToscaWidgets, et Genshi pour fournir une solution complète de développement.

Flexibilité et Extensibilité :

- Permet aux développeurs de choisir et de remplacer les composants par défaut selon leurs préférences.
- Extensible grâce à un grand nombre de plug-ins et d'extensions.

Scalabilité et Performances :

- Conçu pour gérer des applications de grande envergure tout en maintenant des performances élevées.
- Adapté pour des projets de petite à grande échelle.

Tableau 5 : Caractéristiques Clés de TurboGears

Caractéristique	Description
Full-Stack et Modulaire	Peut être utilisé comme un framework full-stack ou micro-framework, intégrant des composants comme SQLAlchemy, ToscaWidgets, et Genshi.
Support des ORM	Utilise SQLAlchemy comme ORM par défaut pour gérer les interactions avec les bases de données.
Sécurité	Intègre repoze.what et repoze.who pour la gestion de l'authentification et des autorisations.
Moteur de Templates	Utilise Genshi ou Kajiki pour les templates, offrant une grande flexibilité dans la présentation des données.
Routage Sophistiqué	Utilise un système de routage flexible pour mapper les URLs aux contrôleurs correspondants.
Widgets et Formulaires	Intègre ToscaWidgets pour créer et valider des formulaires complexes.
Modularité	Permet de structurer les applications en modules réutilisables, facilitant la maintenance et l'évolutivité.
Documentation Complète	Bénéficie d'une documentation complète et de haute qualité, facilitant l'apprentissage et la mise en œuvre des meilleures pratiques.

Communauté Active	Supporté par une communauté active et croissante, offrant des extensions et des solutions aux problèmes courants rencontrés par les développeurs.
Support de l'Internationalisation (i18n)	Facilite la création d'applications multi-langues grâce à ses outils intégrés pour l'internationalisation et la localisation.
Performance	Conçu pour être performant et capable de gérer des applications web de grande envergure, tout en étant adapté pour des projets plus modestes.
Écosystème d'Extensions	Large écosystème d'extensions disponibles pour ajouter des fonctionnalités supplémentaires sans complexifier la base du framework.

En résumé Lorsqu'il s'agit de choisir un framework web Python, les développeurs ont à leur disposition une variété d'options, chacune avec ses propres avantages et inconvénients. Voici un résumé des principales caractéristiques et points forts des frameworks Django, Flask, Pyramid, CherryPy, et TurboGears

Tableau récapitulatif

Frame-work	Date de sortie	Approche	Points forts	Utilisation idéale
Django	2005	Full-Stack	Architecture MVC, ORM intégré, système d'authentification intégré, admin automatique, communauté active, documentation complète	Applications complexes nécessitant une structure rigide et de nombreuses fonctionnalités prêtes à l'emploi
Flask	2010	Micro-Framework	Flexibilité, simplicité, extensibilité, large écosystème d'extensions, Jinja2 pour les templates	Projets de petite à moyenne envergure, ou lorsqu'une personnalisation et une flexibilité maximales sont requises
Pyramid	2010	Minimaliste	Grande flexibilité, support des ORM comme SQLAlchemy, modularité, routage sophistiqué	Applications de toute taille nécessitant une liberté architecturale et une personnalisation poussée
CherryPy	2002	Minimaliste	Facilité d'utilisation, orienté objet, léger, performances élevées	Projets légers à modérés où la simplicité et les performances sont prioritaires
TurboGears	2005	Full-Stack/Modulaire	Combinaison de plusieurs composants puissants, flexibilité pour être utilisé comme un micro-framework ou full-stack	Applications nécessitant une approche full-stack mais avec la flexibilité d'un micro-framework si nécessaire

En résumé Django se distingue par sa richesse fonctionnelle et sa structure rigide, ce qui le rend idéal pour les grandes applications nécessitant une organisation stricte et de nombreux outils prêts à l'emploi. Sa courbe d'apprentissage est légèrement plus raide, mais il offre une solution complète dès la sortie de la boîte.

Flask, en revanche, offre une approche minimaliste et flexible, permettant aux développeurs de choisir et d'ajouter uniquement les composants dont ils ont besoin. Il est parfait pour les petites à moyennes applications où la personnalisation est cruciale.

Pyramid combine flexibilité et puissance, permettant de développer des applications de toute taille avec une liberté architecturale considérable. Son système de routage sophistiqué et sa compatibilité avec divers ORM en font un choix polyvalent pour de nombreux projets.

CherryPy est un framework léger et performant qui facilite le développement d'applications orientées objet. Sa simplicité et ses performances le rendent adapté aux projets de taille modérée où la rapidité et l'efficacité sont essentielles.

TurboGears offre une approche hybride, permettant une utilisation full-stack ou comme micro-framework selon les besoins. Il combine les avantages de plusieurs composants pour fournir une solution robuste et flexible, adaptée à des applications de toute taille.

LA DEUXIEME PARTIE


II. II. Pratique : Création d'une Application Web avec Django pour la gestion des salles de classe

L'image suivante montre un peu l'aperçu de l'accueil de notre application web. Sur cette page, on peut trouver les différentes informations sur chaque salle de classe.

Salles ISSP

[Accueil](#) [A propos](#) [Contact](#) [Administration](#) [Rechercher](#)

Bienvenue sur notre application de gestion de salle de classes




Salle 1

La salle des étudiants en 1ère année de ...

Publiée depuis le 10 juillet 2024

Modifié depuis le 13 juillet 2024 12:45




Salle 2

La salle des étudiants en Master 1

Publiée depuis le 1 juillet 2024

Modifié depuis le 13 juillet 2024 12:45




Salle 3

La salle des étudiants en LPAS 2

Publiée depuis le 13 juillet 2024

Modifié depuis le 13 juillet 2024 12:35




Salle 4

La salle des étudiants en ISE 2

Publiée depuis le 13 juillet 2024

Modifié depuis le 13 juillet 2024 12:35



Salle 5


La salle

Publiée depuis le15 juillet 2024

Modifié depuis le 13 juillet 2024 12:46

Disponible: oui

Lire la suite



Salle 6


Salle de travaux de groupe

Publiée depuis le31 juillet 2024

Modifié depuis le 13 juillet 2024 12:46

Disponible: oui

Lire la suite



Salle 7


Salle de cours

Publiée depuis le10 juillet 2024

Modifié depuis le 13 juillet 2024 12:40

Disponible: non

Lire la suite



Salle 8

Salle de groupe

Publiée depuis le1 juillet 2024

Modifié depuis le 13 juillet 2024 12:42

Disponible: oui

Lire la suite

Tout droit réservé, développé par le Groupe 2 en LPAS2

Les onglets A propos et contact fournissent des explications détaillées sur les concepteurs de l'application web c'est-à-dire leurs noms, leurs photos, et adresse électronique en cas de besoin. Nous avons ajouté ces onglets dans le but d'embellir et de rendre particulier l'application web. A l'extrême gauche nous avons l'onglet rechercher qui permet de simplifier la recherche de classe par un visiteur.

L'onglet administrateur nous envoi vers la partie privée seulement accessible par les administrateurs avec nom d'utilisateur et mot de passe.

Administration de Django

Nom d'utilisateur :

Mot de passe :

Connexion

Pour se connecter il faut obligatoirement remplir ces deux rectangles ci-dessus et ensuite appuyer sur le bouton connexion. Cela nous permettra d'accéder la page suivante ci-dessous.

En cliquant sur le bouton Salles, nous accédons à la page suivante qui offre la possibilité d'ajouter une nouvelle salle.

Administration de Django

BIENVENUE, ADMIN. VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION

Accueil > Gestion > Salles

Écrivez ici pour filtrer...

AUTHENTIFICATION ET AUTORISATION

Groupes + Ajouter

Utilisateurs + Ajouter

GESTION

Salles + Ajouter

Sélectionnez l'objet salle à changer

Action : ----- Envoyer 0 sur 8 sélectionné

- ☐ SALLE
- ☐ Salle 8
- ☐ Salle 7
- ☐ Salle 6
- ☐ Salle 5
- ☐ Salle 4
- ☐ Salle 3
- ☐ Salle 2
- ☐ Salle 1

8 salles

AJOUTER SALLE +

En cliquant sur l'une des salles nous y trouverons les différents onglets à savoir :

- l'onglet historique qui permet de connaître l'historique de la salle en question

Administration de Django

BIENVENUE, CISSE. VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION

Site d'administration

AUTHENTIFICATION ET AUTORISATION

Groupes + Ajouter Modification

Utilisateurs + Ajouter Modification

GESTION

Salles + Ajouter Modification

Actions récentes

Mes actions

- ✗ vdbgd Salle
- martinien Utilisateur
- martinien Utilisateur
- + martinien Utilisateur
- + vdbgd Salle
- ✗ Salle 3 Salle
- + ISE Salle
- ✗ Salle 4 Salle
- ✗ Salle 4 Salle

- l'onglet enregistrer qui permet de sauvegarder les modifications et l'ajout d'une information.
- L'onglet enregistrer et ajouter un nouveau, comme son nom l'indique cet onglet permet de modifier et offre la possibilité d'ajouter un nouvel objet.
- L'onglet Enregistrer et continuer les modifications. Par exemple en étant au niveau de la salle 2, on peut modifier les informations sur cette salle l'enregistrer et continuer la modification.

✓ L'objet salle « Salle 2 » a été modifié avec succès. Vous pouvez l'éditer à nouveau ci-dessous.

Modification de salle

Salle 2 HISTORIQUE

Nom salle :

Description :

La salle des étudiants en Master 1

Image : Actuellement: media/Salle_2.jpg
Modification: Aucun fichier n'a été sélectionné

Date publication : Aujourd'hui |

Disponible :

Ce bouton permet de supprimer la
salle

Site d'administration

The screenshot shows the Django Admin interface. On the left, there are two main sections: 'AUTHENTIFICATION ET AUTORISATION' and 'GESTION'. Under 'AUTHENTIFICATION ET AUTORISATION', there are links for 'Groupes' and 'Utilisateurs', each with '+ Ajouter' and 'Modification' buttons. Under 'GESTION', there is a link for 'Salles' with '+ Ajouter' and 'Modification' buttons. On the right, there is a sidebar titled 'Actions récentes' with a section 'Mes actions' listing recent actions like 'Salle 2' and 'Salle 1'. Three blue arrows point from text boxes below to specific buttons: one from 'Ce bouton permet de voir les utilisateurs' to the 'Utilisateurs' link, one from 'Ce bouton permet d'ajouter les utilisateur' to the '+ Ajouter' button for 'Utilisateurs', and one from 'Ce bouton permet aussi d'ajouter un groupe ou encore un utilisateur' to the '+ Ajouter' button for 'Groupes'.

AUTHENTIFICATION ET AUTORISATION	
Groupes	+ Ajouter Modification
Utilisateurs	+ Ajouter Modification

GESTION	
Salles	+ Ajouter Modification

Actions récentes

Mes actions

- Salle 2
- Salle 2
- Salle 2
- Salle 2
- Salle 2
- Salle 5
- Salle 6
- Salle 2
- Salle 1
- Salle 1

Ce bouton permet de voir les utilisateurs

Ce bouton permet d'ajouter les utilisateur

Ce bouton permet aussi d'ajouter un groupe ou encore un utilisateur

Conclusion

Choisir le bon framework web Python pour votre projet dépend de plusieurs facteurs comme la taille du projet, la flexibilité, et les fonctionnalités nécessaires. Django est idéal pour les grandes applications web grâce à sa structure robuste et ses nombreuses fonctionnalités intégrées. Flask convient mieux aux petits à moyens projets où la flexibilité et la personnalisation sont cruciales. Pyramid offre une grande liberté architecturale pour des projets de toute taille. CherryPy est simple, léger et performant, parfait pour des projets de taille modérée. TurboGears combine les avantages de plusieurs composants, offrant une solution flexible et puissante adaptée à différents types de projets.

Votre nom d'utilisateur pour accéder à la partie administration est [Dr_BARA_Ousmane](#) avec pour mot de passe [prof_python](#).

Dans la conception de notre application, nous avons eu à télécharger des dossiers html avec les stylisations Bootstrap sur internet, avoir donc une connexion internet avant de lancer l'application est donc un atout pour pouvoir jouir de l'application dans tous ses états.