Django :

Django est un framework web open-source écrit en Python. Il fournit une structure solide et des outils pour faciliter le développement rapide et la création d'applications web robustes.

Le terme "framework" fait référence à un ensemble de composants logiciels prêts à l'emploi qui peuvent être utilisés pour créer des applications plus rapidement et plus efficacement. Dans le cas de Django, il s'agit d'un framework spécialement conçu pour le développement web.

Voici quelques concepts clés de Django :

1. Modèle-Vue-Contrôleur (MVC) : Django suit le modèle architectural MVC, mais avec une légère variante appelée Modèle-Vue-Template (MVT). Le modèle représente les données et la logique associée, la vue gère l'affichage des données à l'utilisateur et le template détermine la façon dont les données sont présentées.
2. Object-Relational Mapping (ORM) : Django inclut un ORM intégré qui permet de travailler avec une base de données relationnelle sans avoir à écrire de code SQL. Il facilite la création, la modification et la manipulation des données de la base de données.
3. Gestion de l'URL : Django utilise un système de gestion des URL pour associer des URL aux vues correspondantes. Cela permet de définir facilement les différentes pages de l'application et d'organiser la navigation entre elles.
4. Système de templates : Django propose un système de templates qui permet de séparer la logique métier de la présentation. Les templates sont des fichiers HTML avec des balises spéciales qui permettent d'insérer dynamiquement des données.
5. Sécurité : Django intègre des mécanismes de sécurité tels que la protection contre les attaques par injection SQL, les attaques XSS (Cross-Site Scripting) et la protection contre les attaques CSRF (Cross-Site Request Forgery).
6. Administration : Django fournit une interface d'administration automatique pour les modèles de base de données, ce qui permet de gérer facilement les données de l'application sans avoir à écrire de code supplémentaire.

En résumé, Django est un framework web puissant et flexible qui facilite le développement d'applications web en fournissant des fonctionnalités prêtes à l'emploi et en promouvant les bonnes pratiques de développement. En tant que développeur, vous pouvez utiliser Django pour créer rapidement des applications web dynamiques et sécurisées en utilisant le langage de programmation Python.

Pour lancer un projet avec Django:

pip install django

Cela téléchargera et installera la dernière version stable de Django.

Créez un nouveau projet Django : Une fois Django installé, il faut créer un nouveau projet en exécutant la commande suivante dans le terminal :

django-admin startproject nom_du_projet

Remplacez "nom_du_projet" par le nom qu'il faut donner au projet. Cela créera un répertoire contenant les fichiers de base pour le projet Django.

Accédez au répertoire du projet : Naviguez vers le répertoire du projet en utilisant la commande **cd** dans le terminal :

cd nom_du_projet

Lancez le serveur de développement : Pour lancer le serveur de développement de Django, exécutez la commande suivante :

python manage.py runserver

Cela démarrera le serveur de développement intégré de Django, qui permettra de visualiser l'application dans un navigateur web local.

Accédez à votre application Django : Ouvrez un navigateur web et accédez à l'URL suivante : http://localhost:8000/. Vous devriez voir la page d'accueil de votre projet Django.

Ne pas oublier que la structure et l'organisation d'un projet Django suivent certaines conventions.

Migration :

Python manage.py makemigrations -

Python manage.py migrate

makemigrations est une commande de gestion de base de données Django qui génère des fichiers de migration en fonction des modifications apportées aux modèles de votre application.

migrate est une autre commande de gestion de base de données Django qui applique les migrations pendantes, c'est-à-dire qu'elle met à jour la structure de la base de données pour refléter les modifications définies dans les fichiers de migration.

Gérer le site en super utilisateur :accès admin localhost :8000/admin

python manage.py createsuperuser

Si cela ne fonctionne pas

winpty python manage.py createsuperuser

import export :

pip install django-import-export

Dans le dossier settings.py du projet écrire dans INSTALLED_APPS ‘import_export’, puis dans admin.py mettre le nouvel objet

```
from django.contrib import admin
from import_export.admin ImportExportModelAdmin
from demo.models.personne import Personne

@admin.register(Personne)
Class PersonneAdmin(ImportExportModelAdmin):
        pass
```

import export permet d’ajouter des utilisateurs plus facilement par exemple en format xls

Patern url : dans url.py

```
Path(‘admin/’ , admin.site.urls),
Path(“accounts/”, include(“Django.contrib.auth.urls”)),
```