

GUIDE DÉBUTANT - E-MENU TOGO

Ce guide vous accompagne dans la configuration et le démarrage du projet E-Menu Togo, une plateforme de gestion de restaurants et de menus.

PRÉREQUIS

Avant de commencer, assurez-vous d'avoir installé :

1. Python 3.8 ou supérieur
 - Téléchargez depuis : <https://www.python.org/downloads/>
 - Vérifiez l'installation : `python --version`
2. Un éditeur de code (recommandé : VS Code, PyCharm, ou Sublime Text)

CONFIGURATION INITIALE

ÉTAPE 1 : Extraire le projet

1. Décompressez le fichier ZIP du projet dans un dossier de votre choix
2. Ouvrez un terminal/invite de commande dans ce dossier

ÉTAPE 2 : Créer un environnement virtuel

Windows

```
python -m venv venv  
venv\Scripts\activate
```

macOS/Linux

```
python3 -m venv venv  
source venv/bin/activate
```

ÉTAPE 3 : Installer les dépendances

```
pip install -r requirements.txt
```

CONFIGURATION BASE DE DONNÉES

ÉTAPE 1 : Choisir votre base de données

Le projet utilise PostgreSQL.

Si vous voulez utiliser MySQL, modifiez le fichier :

E_Menu/settings.py

ÉTAPE 3 : Configuration PostgreSQL

1. Installez PostgreSQL sur votre système
2. Créez une base de données
3. Modifiez E_Menu/settings.py :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'votre_nom_db',
        'USER': 'votre_utilisateur',
        'PASSWORD': 'votre_mot_de_passe',
        'HOST': 'localhost',
        'PORT': '5432',
        'OPTIONS': {
            'charset': 'utf8',
        },
    }
}
```

ÉTAPE 4 : Configuration MySQL (optionnel)

1. Installez MySQL sur votre système
2. Créez une base de données
3. Installez le driver : `pip install mysqlclient`
4. Modifiez `E_Menu/settings.py` :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'votre_nom_db',
        'USER': 'votre_utilisateur',
        'PASSWORD': 'votre_mot_de_passe',
        'HOST': 'localhost',
        'PORT': '3306',
        'OPTIONS': {
            'charset': 'utf8mb4',
        },
    }
}
```

MIGRATIONS

ÉTAPE 1 : Créer les migrations

`python manage.py makemigrations`

Cette commande analyse vos modèles et crée les fichiers de migration nécessaires.

ÉTAPE 2 : Appliquer les migrations

`python manage.py migrate`

Cette commande applique toutes les migrations et crée les tables dans la base de données.

ÉTAPE 3 : Vérifier les migrations

`python manage.py showmigrations`

Cette commande affiche l'état de toutes les migrations.

CRÉATION D'UN SUPERUTILISATEUR

Créez un compte administrateur pour accéder à l'interface d'administration :

`python manage.py createsuperuser`

Suivez les instructions pour créer votre compte administrateur.

DÉMARRAGE DU SERVEUR

ÉTAPE 1 : Lancer le serveur de développement

`python manage.py runserver`

Le serveur démarre sur `http://127.0.0.1:8000/`

ÉTAPE 2 : Accéder à l'application

- Interface principale : `http://127.0.0.1:8000/`

- Interface d'administration : `http://127.0.0.1:8000/admin/`

STRUCTURE DU PROJET

E_Menu/ # Configuration principale Django

├── settings.py # Paramètres du projet

├── urls.py # URLs principales

└── wsgi.py # Configuration WSGI

accounts/ # Gestion des utilisateurs

├── models.py # Modèle utilisateur

├── views.py # Vues d'authentification

└── templates/ # Templates d'authentification

structures/ # Gestion des restaurants

├── models.py # Modèle Structure

├── views.py # Vues des structures

└── templates/ # Templates des structures

plats/ # Gestion des plats

├── models.py # Modèle Plat

├── views.py # Vues des plats

└── templates/ # Templates des plats

menus/ # Gestion des menus
├── models.py # Modèle Menu
├── views.py # Vues des menus
└── templates/ # Templates des menus

avis/ # Système d'avis
├── models.py # Modèle Avis
├── views.py # Vues des avis
└── templates/ # Templates des avis

static/ # Fichiers statiques (CSS, JS, images)
├── css/ # Styles CSS
├── js/ # Scripts JavaScript
└── images/ # Images

templates/ # Templates principaux
├── base.html # Template de base
└── index.html # Page d'accueil

FONCTIONNALITÉS PRINCIPALES

1. GESTION DES UTILISATEURS

- Inscription/Connexion
- Profils utilisateurs
- Gestion des permissions

2. GESTION DES STRUCTURES (RESTAURANTS)

- Création de restaurants
- Gestion des informations
- Système de notation

3. GESTION DES PLATS

- Ajout de plats
- Catégorisation
- Système de promotions
- Gestion des prix

4. GESTION DES MENUS

- Création de menus
- Association plats-menus
- Gestion des disponibilités

5. SYSTÈME D'AVIS

- Notation des plats
- Commentaires
- Calcul des moyennes

6. RECHERCHE ET FILTRAGE

- Recherche par nom
- Filtrage par catégorie
- Filtrage par disponibilité
- Filtrage par promotion

COMMANDES UTILES

Gestion des migrations

```
python manage.py makemigrations  
python manage.py migrate  
python manage.py showmigrations
```

Gestion des utilisateurs

```
python manage.py createsuperuser  
python manage.py changepassword
```

Serveur de développement

```
python manage.py runserver  
python manage.py runserver 8080 # Port personnalisé
```

Shell Django

```
python manage.py shell
```

Collecte des fichiers statiques

```
python manage.py collectstatic
```

Vérification du projet

```
python manage.py check
```

Création d'une application

```
python manage.py startapp nom_app
```

DÉPANNAGE

PROBLÈME : "No module named 'django'"

SOLUTION : Activez votre environnement virtuel et installez les dépendances

PROBLÈME : "Database connection failed"

SOLUTION : Vérifiez la configuration de votre base de données dans settings.py

PROBLÈME : "Migration conflicts"

SOLUTION : Supprimez le dossier migrations/ et refaites les migrations

PROBLÈME : "Static files not found"

SOLUTION : Exécutez python manage.py collectstatic

PROBLÈME : "Port already in use"

SOLUTION : Utilisez un autre port : python manage.py runserver 8080