

CAHIER DE CHARGES POUR LE PROJET DU SITE WEB *LA-TRAVERSÉE*

Version 1.0

31 août 2024

Fait par :

Ana Isnardis

Paris

France

SOMMAIRE

Présentation du projet.....	3
Notice d'installation.....	4
a) Introduction.....	4
b) Étape 1 : Cloner le projet.....	4
c) Étape 2 : Créer et activer un environnement virtuel.....	5
d) Étape 3 : Installer les dépendances.....	5
e) Étape 4 : Configurer la base de données.....	5
Choix des libraires.....	7
Algorithmes utilisés.....	7
Structure de données.....	8
a) Modèle Video.....	8
b) Modèle Poeme.....	9
c) Modèle Memoire.....	10

Présentation du projet

Ce projet vise la création d'un site web du projet artistique intitulé *la-traversée*. Celui-ci est un projet pluridisciplinaire réalisé à partir d'un mémoire de recherche soutenu en août 2023. Après la rédaction des poèmes et des vidéos, ce projet a l'objectif de la création d'un spectacle autour de l'histoire des femmes portugaises pendant la colonisation brésilienne.

L'intérêt de ce site web est de présenter l'idée du spectacle, ainsi que les créations déjà réalisées : les poèmes, les vidéos et le mémoire. Il est destiné à toute personne intéressée dans le projet et aux structures qui seront éventuellement contactées dans la recherche d'un financement. L'objectif est que les personnes aient des informations claires à propos du projet et puissent accéder facilement aux créations réalisées.

Pour le faire, les vidéos seront disponibles directement sur le site web, ainsi que les poèmes et le mémoire. Les personnes pourront filtrer les vidéos par langue (portugais, français ou bilingues), par environnement (vidéos enregistrées à l'intérieur ou à l'extérieur) et par des vidéos avec musique et sans musique. Les poèmes pourront aussi être filtrés par langue.

Dans la home page du site web, il sera possible d'accéder à quatre pages : *poèmes*, où seront présentés les poèmes, *mémoire*, où le mémoire sera disponible directement dans la page, *registres*, où seront les vidéos et *à propos*, où il aura un texte de présentation du projet.

Notice d'installation

a) Introduction

Cette notice d'installation a comme objectif d'écrire les étapes nécessaires pour configurer et déployer le projet sur votre machine, qu'elle soit Mac ou Windows. L'objectif est de garantir que toutes les dépendances sont installées correctement, que la base de données est configurée et que le serveur de développement est opérationnel. Avant de commencer l'installation, assurez-vous que les éléments suivants sont installés sur votre machine :

Python 3.x : Le projet nécessite Python 3 pour fonctionner. Vous pouvez vérifier si Python 3 est installé en utilisant la commande :

```
python3 --version
```

Pip : Pip est le gestionnaire de packages pour Python et est nécessaire pour installer les dépendances du projet. Il est généralement inclus avec Python 3. Vérifiez la version avec :

```
pip3 --version
```

Virtualenv : Un environnement virtuel isolé pour installer les dépendances du projet. Il permet de garder les dépendances spécifiques au projet séparées des autres projets Python.

Git : Pour cloner le projet depuis un dépôt Git.

b) Étape 1 : Cloner le projet

Commencez par cloner le dépôt Git sur votre machine locale en utilisant la commande :

```
git clone https://github.com/isnardis/la-traversee.git
```

```
cd la-traversee
```

c) Étape 2 : Créer et activer un environnement virtuel

Vous devez créer un environnement virtuel pour installer les dépendances du projet. Pour cela, faites la commande :

```
python3 -m venv venv
```

Pour activez l'environnement virtuel sur Mac, faites la commande :

```
source venv/bin/activate
```

Si vous êtes sur Windows, la commande est la suivante :

```
.\venv\Scripts\activate
```

Vous devriez maintenant voir (venv) au début de votre invite de commande, indiquant que l'environnement virtuel est actif.

d) Étape 3 : Installer les dépendances

Une fois l'environnement virtuel activé, il est nécessaire d'installer les dépendances du projet à partir du fichier requirements.txt. Pour le faire, saisissez la commande suivante :

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

e) Étape 4 : Configurer la base de données

Le projet utilise une base de données pour stocker les informations. Vous devez appliquer les migrations pour configurer sa base de données. Pour le faire, suivez les étapes suivantes :

1. Appliquer les migrations

```
python3 manage.py migrate
```

2. Créer un superutilisateur (cela vous permettra d'accéder à l'interface d'administration)

```
python3 manage.py createsuperuser
```

Ensuite, suivez les instructions pour définir un nom d'utilisateur, un mot de passe et un e-mail pour le superutilisateur.

3. Lancer le serveur de développement

Une fois les étapes précédentes terminées, vous pouvez lancer le serveur de développement Django pour lancer votre site en production :

```
python3 manage.py runserver
```

Accédez à votre navigateur et tapez `http://127.0.0.1:8000/` pour voir le site web.

Votre projet Django devrait maintenant être installé et fonctionnel. Vous pouvez commencer à développer, tester, ou déployer le projet en fonction de vos besoins. Si vous rencontrez des problèmes, vérifiez que toutes les étapes ont été suivies correctement et consultez la documentation officielle de Django pour des informations supplémentaires.

Choix des libraires

Django est le framework web principal utilisé dans ce projet. Sa sélection est justifiée par plusieurs facteurs clés. D'abord, Django offre une architecture modulaire, connu par MVC : modèle-vue-contrôleur. Cela veut dire que chaque module peut fonctionner de façon indépendante, nous permettant donc de modifier la logique métier et les composants visuels d'une application sans affecter l'autre. Grâce à cette architecture bien structurée, Django permet de gérer efficacement des projets de toute taille, en offrant la possibilité d'augmenter la taille du site web si souhaité.

Ce framework connaît aussi une importante sécurité, car il gère automatiquement plusieurs aspects, comme la prévention des injections SQL. Finalement, Django bénéficie aussi d'une documentation complète, en facilitant la résolution des problèmes mais aussi l'implémentation de nouvelles fonctionnalités.

Algorithmes utilisés

Dans le cadre du développement d'applications web, en particulier celles construites avec Django, la sélection des algorithmes est cruciale pour assurer la performance et la scalabilité du projet. Ce chapitre présente les algorithmes clés utilisés dans le projet, justifie leur choix en fonction des exigences fonctionnelles et non fonctionnelles et explique comment ils contribuent à la réalisation des objectifs du projet.

Dans ce projet, les pages registres (vidéos) et poèmes nécessitent la possibilité de filtrer les contenus en fonction de plusieurs critères, tels que la langue, le lieu, et la présence ou non de musique. L'algorithme utilisé est celui de filtrage basé sur des requêtes SQL ou ORM (Object-Relational Mapping) de Django.

Cet algorithme permet d'exécuter des requêtes directement sur la base de données, minimisant ainsi la quantité de données chargées en mémoire et réduisant le temps de traitement. Grâce à l'ORM, il est possible de construire dynamiquement des requêtes complexes. L'implémentation des filtres avec des méthodes comme *filter()*, *exclude()*, ou *annotate()* de Django permet de gérer facilement les conditions de filtrage sans écrire de code SQL.

Le projet nécessite aussi l'utilisation de templates pour générer dynamiquement des pages web à partir des données stockées dans la base de données. Il été utilisé donc des

algorithmes de mise en cache et de préchargement des données. En mettant en cache les requêtes fréquentes ou les fragments de page, on évite de ré-exécuter des calculs ou des requêtes coûteuses, ce qui améliore la réactivité de l'application.

Le cache permet aussi de gérer efficacement un grand nombre d'utilisateurs simultanés en réduisant la charge sur le serveur de base de données, en permettant aussi d'optimiser l'expérience utilisateur en réduisant les temps d'attente.

Structure de données

La structure de données est un aspect fondamental dans la conception et le développement de tout projet logiciel. Elle détermine comment les données sont organisées, stockées, et manipulées au sein de l'application. Ce chapitre présente la structure de données utilisée dans le projet Django, décrivant les modèles de données, leurs relations, ainsi que la manière dont ces structures soutiennent les exigences fonctionnelles du projet.

Dans ce projet Django, les principales entités sont les vidéos, les poèmes, et le texte du mémoire de master. Chaque entité est représentée par un modèle Django, qui définit la structure des données, les types de champs et les relations entre les entités. Ces modèles sont stockés dans le fichier *models.py* de l'application *core*.

a) Modèle *Video*

Le modèle *Video* représente les vidéos disponibles sur la page *Registres*. Chaque vidéo possède plusieurs attributs qui permettent de les filtrer selon les critères définis. Voici leurs attributs principaux :

- titre : Le titre de la vidéo.
- description : Une brève description du contenu de la vidéo.
- langue : La langue de la vidéo (e.g., 'français', 'portugais', 'bilingue').
- lieu : Le lieu où la vidéo a été filmée (e.g., 'intérieur', 'extérieur').
- musique : Un booléen indiquant si la vidéo contient de la musique ou non.
- fichier : Le chemin vers le fichier vidéo stocké sur le serveur.

Les vidéos sont autonomes et ne dépendent d'aucune autre entité. Elles peuvent être filtrées et triées selon les attributs décrits ci-dessus. Voici la définition du modèle :

```
from django.db import models
```



```

class Video(models.Model):
    LANGUES = [
        ('fr', 'Français'),
        ('pt', 'Portugais'),
        ('bi', 'Bilingue'),
    ]

    LIEUX = [
        ('int', 'Intérieur'),
        ('ext', 'Extérieur'),
    ]

    titre = models.CharField(max_length=200)
    description = models.TextField()
    langue = models.CharField(max_length=2, choices=LANGUES)
    lieu = models.CharField(max_length=3, choices=LIEUX)
    musique = models.BooleanField(default=False)
    fichier = models.FileField(upload_to='videos/')

    def __str__(self):
        return self.titre

```

b) Modèle *Poeme*

Le modèle *Poeme* représente les poèmes disponibles sur la page *Poèmes*. Comme pour les vidéos, les poèmes peuvent être filtrés par langue. Les poèmes sont également autonomes et peuvent être filtrés selon la langue spécifiée. Voici leurs attributs principaux, suivie de la définition de ce modèle.

- titre : Le titre du poème.
- contenu : Le texte du poème.
- langue : La langue du poème (e.g., 'français', 'portugais', 'bilingue').

```

class Poeme(models.Model):
    LANGUES = [
        ('fr', 'Français'),
        ('pt', 'Portugais'),
        ('bi', 'Bilingue'),
    ]

    titre = models.CharField(max_length=200)
    contenu = models.TextField()
    langue = models.CharField(max_length=2, choices=LANGUES)

    def __str__(self):
        return self.titre

```

c) Modèle *Memoire*

Le modèle *Memoire* représente un mémoire de master disponible sur la page Mémoire. Ce modèle est plus simple, car il n'a besoin que de stocker un document unique. Ce modèle n'entretient pas de relations complexes avec d'autres modèles. Ces attributs principaux sont :

- titre : Le titre du mémoire.
- description : Une description du contenu du mémoire.
- document : Le fichier PDF ou Word du mémoire stocké sur le serveur.

Leur définition dans le projet c'est la suivante :

```
class Memoire(models.Model):  
    titre = models.CharField(max_length=200)  
    description = models.TextField()  
    document = models.FileField(upload_to='memoires/')  
  
    def __str__(self):  
        return self.titre
```

La structure de données du projet a été conçue pour être simple, efficace et évolutive. Elle permet de stocker et de gérer efficacement les informations relatives aux vidéos, aux poèmes et au mémoire, en garantissant une flexibilité suffisante pour répondre aux besoins de filtrage et d'affichage des contenus. Cette architecture assure une base pour les développements futurs et pour l'intégration de nouvelles fonctionnalités.

Manuel utilisateur

Ce manuel utilisateur a pour objectif de guider les utilisateurs à travers les fonctionnalités principales du site web créé avec Django. Il explique comment naviguer sur le site, utiliser les filtres disponibles, et interagir avec les différents contenus proposés, tels que les vidéos, les poèmes, et les mémoires. Ce manuel est destiné aux utilisateurs finaux qui souhaitent tirer le meilleur parti de l'application.

Pour accéder au site, ouvrez votre navigateur web de préférence et entrez l'URL suivante :

`http://127.0.0.1:8000/`

Cela vous dirigera vers la page d'accueil du site. Le site comporte plusieurs sections principales, chacune accessible depuis la page d'accueil ou via le menu de navigation situé en haut de chaque page. Les principales sections du site sont les suivantes :

1. Page d'Accueil (Home)
2. Vidéos (intitulé Registres)
3. Poèmes
4. Mémoire
5. À Propos

La page d'accueil est la première page que vous verrez en accédant au site. Elle contient un texte d'introduction qui présente le site et son contenu. Vous pouvez revenir à la page d'accueil à tout moment en cliquant sur le lien "Home" dans le menu de navigation. La page des vidéos vous permet de visualiser et de filtrer les vidéos disponibles sur le site. Les vidéos peuvent être filtrées selon plusieurs critères pour vous aider à trouver celles qui vous intéressent. Les vidéos sont affichées sous forme de liste ou de vignettes, avec des informations de base comme le titre et une brève description. Les filtres disponibles sont :

- **Langue** : Choisissez entre les vidéos en français, en portugais, ou bilingues.
- **Lieu** : Sélectionnez les vidéos filmées à l'intérieur ou à l'extérieur.
- **Musique** : Filtrez les vidéos avec ou sans musique.

Pour utiliser les filtres, sélectionnez les options de filtrage souhaitées dans les menus déroulants ou les cases à cocher. Ensuite, cliquez sur le bouton "Appliquer les filtres" pour mettre à jour la liste des vidéos affichées.

La page des poèmes vous permet de lire et de filtrer les poèmes disponibles sur le site. Les poèmes sont affichés sous forme de liste avec leur titre et un extrait du texte. Choisissez entre les poèmes en français, en portugais, ou bilingues. Pour le faire, sélectionnez la langue souhaitée dans le menu déroulant. Ensuite, cliquez sur "Appliquer les filtres" pour afficher les poèmes correspondant à vos critères.

La page de mémoire est dédiée à la présentation d'un mémoire de master. Cette page permet de télécharger ou de visualiser le mémoire directement depuis le site. Un lien pour télécharger ou lire le document est disponible, accompagné d'une description du contenu. Cliquez sur le lien "Télécharger le Mémoire" ou "Lire le Mémoire" pour accéder au document. Finalement, la page À Propos fournit des informations sur le projet, son objectif, et les personnes ou organisations qui y sont impliquées. Elle possède un texte explicatif sur les objectifs et la mission du projet et donne des détails sur la manière de contacter les responsables du site.

Chaque section du site est conçue pour permettre une interaction simple et intuitive avec le contenu. Cliquer sur les titres permet de voir plus de détails ou d'accéder à des pages spécifiques. et utiliser les filtres affine la recherche et facilite l'accès aux contenus pertinents. Ce manuel utilisateur vise à vous fournir toutes les informations nécessaires pour naviguer efficacement sur le site et utiliser ses fonctionnalités. Si vous avez besoin d'assistance supplémentaire, n'hésitez pas à consulter la page À Propos pour obtenir des informations de contact.