Lien vers le code source sur Github : <https://github.com/GuillaumeM92/GrandPy>

Lien vers le tableau Trello : <https://trello.com/b/gulEQvsF/grandpy>

**Démarche choisie :**

* **Méthodologie de projet choisie :**
  + J’ai d’abord rempli un tableau Trello avec les étapes clés à réaliser pour mener à bien le projet selon la méthodologie de projet agile. Au fur et à mesure du développement des modules, j’ai pu visualiser le travail effectué, et le travail restant à faire.
  + Conformément aux consignes de ce projet 7 j’ai adopté une démarche de « test-driven development » pour l’écriture des modules de mon application web. C’est-à-dire que j’ai commencé par la rédaction des tests sous pytest à l’aide du mot clé « assert » et de la condition à remplir. Les tests s’affichaient alors en rouge.  
    Puis j’ai écrit en python le code permettant le fonctionnement des 3 modules du dossier « utils », en faisant des tests à l’aide de mocks récupérés via les API de Google Maps et de Wikipedia. Une fois que les tests se sont affichés en vert, je me suis penché sur le code de mes modules afin de le refactoriser et de le rendre plus DRY et efficient.

**Difficultés rencontrées :**

* **Développement Web :**
* C’était la première fois que je faisais du développement web, la difficulté principale a donc été d’apprendre toutes les bases nécessaires à ce travail, telles que :
  + **HTML :**
    - Apprentissage de la syntaxe, des tags, des différents de contenu, et des conventions utilisées pour le web moderne.
  + **CSS :**
    - Apprentissage des fonctionnalités de mise en page CSS, notamment des fonctionnalités modernes telles que les  « flexbox ».
  + **JavaScript :** 
    - Apprentissage du langage Javascript et de se différences de syntaxe par rapport à Python. Compréhension du ciblage des éléments HTML grâce au DOM.
  + **jQuery :**
    - Apprentissage de la syntaxe JavaSript simplifiée. Compréhension du fonctionnement des opérations asynchrones avec AJAX.
  + **Heroku :**
    - Apprentissage du fonctionnement d’heroku via la documentation en ligne. Des difficultés notamment pour la configuration du « procfile » nécessaire pour définir le point d’entrée vers l’application.
  + **Google Maps API :**
    - Apprentissage des paramètres de l’API google Maps, récupération et utilisation de l’adresse formatée et des coordonnées au format JSON. Affichage de la carte intégrée dans le site.  
      Mise en place d’une clé API et de paramètres pour restreindre son utilisation uniquement au site web.
  + **Wikipedia API :**
    - Apprentissage du fonctionnement et des paramètres de l’API Wiki Media, via une documentation parfois un peu complexe car étoffée.  
      J’ai opté pour une recherche basée sur les coordonnées GPS récupérées dans l’API Maps, ce qui permet d’obtenir des anecdotes pertinentes extraites de Wikipedia pour la plupart des adresses demandées.
* **Nouveaux modules Python :**
* Pour mener à bien ce projet, il m’a fallu apprendre à utiliser de nouveaux modules de Python, tels que :
  + **Flask :**
    - Apprentissage du rôle de flask dans un projet web, et de son fonctionnement, notamment la notion de routes qui permettent de traiter les requêtes HTML « Get » et « Post ».
  + **Pytest :**
    - Apprentissage du fonctionnement du module pytest, et de son utilité dans le cadre du test des fonctionnalités d’un projet, via l’utilisation de tests unitaires.

Découpage de mon code en module, de manière à pouvoir les tester individuellement de façon autonome.

* **Regular Expressions :**
  + Utilisation du module Regex (RE) dans Python afin de concevoir le parser des input utilisateur. Les regex ont été utilisées uniquement pour filtrer les inputs de manière à ne conserver que les caractères alphanumériques.

La majorité du travail du parser est en fait effectué grâce à une liste de « stopwords » préconçue et personnalisée, mise à disposition par Peter Graham.