## Projet 7 « Créez GrandPy Bot, le papy-robot »

Olena ANDROSOVYCH

# Introduction et annonce du plan

Lors de Projet 7, on développe un bot « GrandPy » qui fournit l’adresse de l’endroit que l’utilisateur cherche aussi bien que « raconte » une histoire par rapport au sujet.

Dans une première partie de ce document je présenterai la démarche choisie.

Dans la deuxième partie je citerai les difficultés rencontrées et les solutions trouvées.

Dans la dernière partie on trouvera le lien vers le code du programme (sur Github) et l'adresse du site web déployé sur Heroku.

# I. La démarche choisie

1. Lors du projet 7, tout d’abord, j’ai créé un plan du travail et découpé le projet par les tâches et sous- tâches. Pour l’organisation de ces modules j’ai utilisé *trello.com*.
2. Dans la deuxième étape, j’ai installé le *flask* et *pytest*, ai créé la structure du projet *flask* et réalisé un premier test unitaire qui vérifie un code de réponse HTTP qui doit être « 200 OK ».
3. Par la suite, j’ai créé une interface utilisateur en utilisant *bootstrap* (les fichiers: « main.html », « bootstrap.min.css », «style.css») et ai vérifié la responsivité de l'interface.
4. Dans l'étape suivante, j'ai créé un parseur c'est-a-dire une méthode qui analyse la question d'un utilisateur afin de trouver dans le phrase un nom de l’endroit recherché (le fichier « parseur.py »).
5. Puis, j'ai installé *requests* et ai réalisé des requêtes à l'API Google Maps afin de trouver l'adresse de l'endroit recherché et ses coordonnés, grâce auxquelles on conçoit la carte (les fichiers « location.py », « map.js »).
6. De même façon, on utilise les mots clés reçus du parseur afin de retirer l'information de l'API Wikipédia. On utilise *monkeypatch* pour mocker la méthode *requests.get()* (le fichier « wiki\_info.py »).
7. Lors de mon projet, j'ai exploité l'approche *ajax* afin d'envoyer des questions de l'utilisateur et afficher des réponses du bot sur la page web (le fichier « ajax.js »).
8. L'étape finale consiste au déploiement de l'application sur web à l'aide du Heroku.

# II. Les difficultés rencontrées et les solutions trouvées

Lors du développement du projet 7 j'ai eu plusieurs difficultés.

1. L'utilisation d'*ajax*. Après l'apprentissage de la partie théorique, au début, il était difficile à adopter cette approche pour mon projet.

La solution : j’ai regardé des vidéos explicatifs avec des exemples différents et ai demandé mon mentor.

1. La création d’une carte à cause de la limite d’un nombre des requêtes par jour à l’API Google Maps.

La solution : créer un comte payant qui propose les services gratuits pour le période d'essai.

1. La page html n'est pas mise à jour après le changement du code. La problème est lié au cash du browser.

La solution : choisir l'option « disable cash » dans les outils de développement des navigateurs.

1. Déploiement de l'application sur Heroku. Les *errors* a été concerné un variable d'environnement.

La solution : définir Config Vars dans les settings sur le site du Heroku.

# III. Le code du programme et l'adresse du site web de l'application

Le lien vers le code du programme : <https://github.com/OlenaChe/Project7_GP>

L'adresse du site déployé sur l'Internet <https://project7-grandpy.herokuapp.com/>