Matthias GIL - DA Python

Livrable : Projet 7 Grandpy Bot, le papy robot

Lien vers le site internet : https://oc-projet7-matthiasgil.herokuapp.com/
Lien vers le depository GitHub : https://github.com/herodote2242/projet7_grandpy
Lien vers le tableau Trello : https://trello.com/b/szFRgXmn/p7

Ce document explique la démarche choisie, les difficultés rencontrées et les solutions trouvées.

La première étape a été de travailler sur le "parser". C'est-à-dire, créer le script sentence_manipulator.py qui vient découper la question posée par l'utilisateur en une suite de mots, et supprimer tout ce qui n'est pas nécessaire pour la suite (accents, ponctuation, mots communs). Ne restent que les mots-clés qui sont ensuite envoyés à un second script, maps.py, qui utilise cette liste de mots pour rechercher grâce à l'API Google Maps, l'adresse et les coordonnées gps correspondant aux mots-clés. Enfin, les coordonnées gps sont transmises au troisième script wikidata.py, qui s'en sert pour récupérer sur l'API de Wikipédia le résumé de l'article correspondant au lieu. Finalement, un dernier script app.py se charge de compiler l'agencement des trois précédents pour obtenir une réponse exploitable.

Pour chacun de ces quatre scripts, et afin de suivre les recommandations de TDD, quatre scripts "miroirs" ont été créés, afin d'effectuer des mocks sur les APIs et des tests unitaires sur les fonctions Python. Si l'utilité des tests unitaires me paraît évidente, sa mise en application la première fois l'est moins. J'ai eu du mal à transformer cette nécessité en test, et à comprendre la manière d'utiliser les mocks. Mon mentor a pris du temps pour m'expliquer comment les agencer correctement, car j'ai trouvé le cours d'OC sur le sujet trop superficiel.

Pour la seconde étape, il s'agissait d'implémenter flask sur le projet. Cette fois encore, je trouve que le cours OC est trop éloigné des besoins du projet, avec son exemple sur Facebook. Surtout que j'y suis devenu allergique depuis un moment. J'ai donc suivi le tutoriel de Miguel Grindberg, extrêmement bien expliqué, pour créer une couche minimaliste du futur site, avec l'utilisation d'une base et d'une vue qui en hérite. Flask est très intéressant, et de ce que j'en ai vu, il permet de construire des petits projets relativement aisément. Contrairement à Django qui devient tout indiqué dès qu'un projet prend un minimum d'ampleur. Flask est le genre de framework qui a l'air taillé sur mesure pour faire des petits projets personnels. Je me pencherai de nouveau dessus une fois mon parcours fini, ou si j'ai besoin de faire des projets personnels en attendant de mieux connaître Django.

La troisième étape a été de travailler sur le front-end, c'est à dire la partie visible du futur site internet. J'ai créé une structure html et css relativement simple, avec un fond d'écran trouvé sur unsplash.com représentant deux grand-pères. C'est en arrivant à la partie JavaScript que j'ai rencontré la plus grande difficulté de ce projet. Si le premier cours "apprendre à coder avec JavaScript" m'a beaucoup plu au niveau de la syntaxe et des ressemblances avec Python, le second

"créez des pages web interactives" m'a complètement perdu. Etant bloqué sur l'avancement du projet, j'ai pris plus d'un mois pour travailler sur ce second cours, en faisant les exercices et activités de manière approfondie. Un long moment de découragement plus tard, et une fois que les trois activités étaient validées et les notions assimilées, je me suis senti plus à l'aise pour reprendre le cours du projet et implémenter la partie JavaScript et les requêtes AJAX sur mon site de Grandpy. La fin de la partie JavaScript a soulevé différents problèmes, tel que le non affichage de la carte et du marqueur, alors qu'aucune erreur côté serveur ou côté JavaScript n'y faisait référence. Enfin, une dernière mise en style des différents éléments constitutifs des questions réponses a été nécessaire afin de rendre le site un minimum attrayant. Je ne suis pas graphiste et je pense qu'il est très facile de faire plus joli au niveau du design, mais je préfère les interfaces épurées et sobres. Le côté tout en nuances de gris, façon noir et blanc du début du 20è siècle, collait parfaitement au personnage de Grandpy, qui cherche dans ses souvenirs pour raconter ses anecdotes.

Concernant la quatrième et dernière étape, le déploiement sur un serveur Heroku, c'était sans aucun doute la partie la plus facile et la plus rapide. Malheureusement, une erreur de Pipfile.lock, avec la présence d'une ligne qui ne devait pas exister, résultat d'une mauvaise manipulation de ma part, m'a pris plusieurs heures à résoudre. Sans compter ce problème, il aurait suffit de peu de temps pour déployer. Pour ce faire, j'ai suivi une fois de plus les instructions données par le site de Miguel Grindberg, car je n'ai pas trouvé dans les explications de base d'Héroku de mentions concernant la création et le contenu du fichier Procfile. Et même si dans un premier temps l'application était déployée, l'url saisie dans mon explorateur renvoyait "application error".
C'est un outil très agréable que je découvre en tant que débutant, et que je ne manquerai pas d'utiliser pour mes prochains projets personnels. Je pense déjà à faire une première version de ce que je souhaite utiliser pour le p13 (une application pour un club sportif), mais dans une version site internet + Flask, déployé sur Heroku.

CONCLUSION :
C'est un projet qui m'a apporté son lot de nouveautés et de découvertes, mais surtout de grands questionnements. Pendant le long mois de remise à niveau sur JavaScript, je me suis demandé plus d'une fois si j'avais choisi le bon parcours. Un choix de parcours plus court, tel Développeur Web Junior, n'aurait-il pas été plus pertinent ? A force de réflexion, je suis tout de même revenu à mon point de départ. C'est bien le parcours Python que je veux valider, peu importe si c'est beaucoup plus long que prévu. Peu importe si j'ai plus de difficultés que je ne pouvais imaginer avant de commencer le parcours. Cette remise en question m'a permis de me rendre compte que j'ai encore énormément de choses à approfondir. En Python bien sûr, mais aussi en JavaScript, que j'aimerais maîtriser plus largement. C'est devenu un de mes projets prioritaires post-parcours, afin d'élargir mes compétences plus vers du full-stack (même si ce terme est voué à disparaître car plus vraiment adapté à la réalité, mais qui à mon sens englobe l'aspect front-end et back-end de manière intuitive). Et puis en Flask, finalement. Tout comme j'imagine que le projet 8 me fera envie de me pencher encore plus sur Django à la fin du parcours.