



Rapport Technique — mini-projet-python-portfolio-GitHub

1. Introduction

Ce rapport technique présente une application développée en Python permettant d'analyser automatiquement les dépôts GitHub et de générer un portfolio professionnel sous forme de site web statique (HTML + CSS). L'objectif est de fournir une vitrine claire et lisible des projets GitHub, incluant des informations clés telles que le nombre de commits, les langages utilisés, les étoiles (stars), les forks, ainsi qu'un score de valorisation calculé automatiquement.

2. Objectifs du mini projet

Les objectifs principaux du projet sont :

- Récupérer les repositories publics via l'API GitHub.
- Extraire des métriques utiles : commits, langages, stars, forks.
- Calculer un score de projet afin de faciliter la mise en avant des dépôts les plus pertinents.
- Générer automatiquement un fichier HTML de portfolio à partir d'un template Jinja2.
- Appliquer un style CSS responsive basé sur une grille et des cartes (cards).

3. Architecture du projet

L'application est organisée selon une structure simple et maintenable, séparant la logique de récupération des données, l'analyse et la génération HTML.

Dossier/Fichier	Rôle
assets/style.css	Feuille de style CSS (design, grille responsive, cartes).
templates/portfolio.html	Template Jinja2 utilisé pour générer le portfolio HTML.
output/portfolio.html	Fichier HTML final généré (résultat).
config.py	Configuration : BASE_URL et GITHUB_TOKEN.
github_api.py	Accès à l'API GitHub : repos, commits, langages.
analyzer.py	Analyse des repos et calcul du score.
html_generator.py	Rendu Jinja2 et génération du fichier HTML final.
main.py	Point d'entrée : orchestration de l'exécution.

4. Fonctionnement de l'application

Le fonctionnement global suit un pipeline en quatre étapes :

- 1) Saisie du nom d'utilisateur GitHub.
- 2) Récupération des repositories via l'API GitHub.
- 3) Analyse des repositories (métriques + score).
- 4) Génération du portfolio HTML dans le dossier **output/**.

5. Interaction avec l'API GitHub

Le module **github_api.py** effectue des requêtes HTTP vers l'API GitHub

(BASE_URL = <https://api.github.com>). Les endpoints utilisés sont :

- **/users/{username}/repos** : liste des repositories.
- **/repos/{username}/{repo}/commits** : liste des commits (utilisée pour compter les commits).
- **/repos/{username}/{repo}/languages** : répartition des langages (clé/valeur).

6. Analyse des repositories et calcul du score

Le module **analyzer.py** analyse chaque repository et calcule un score selon la formule suivante :

$$\text{Score} = (\text{commits} \times 2) + (\text{stars} \times 5) + (\text{forks} \times 3)$$

Cette pondération permet de valoriser à la fois l'activité (commits) et la popularité (stars/forks). Les informations analysées sont ensuite envoyées au moteur de template pour la génération HTML.

7. Génération du portfolio HTML

Le module **html_generator.py** utilise **Jinja2** pour charger le template **templates/portfolio.html**, remplacer les variables (username et repos) et générer le fichier final : **output/portfolio.html**

8. Mise en forme (CSS)

Le fichier **assets/style.css** applique un style simple et professionnel :

- Fond gris clair, police sans-serif,
- Affichage en grille responsive (*grid*),
- Cartes blanches avec ombre et bord arrondi,
- Liens GitHub mis en évidence.

9. Résultat obtenu

Après exécution, le portfolio est accessible via **output/portfolio.html**. La page affiche les repositories de l'utilisateur avec les métriques calculées et un rendu visuel sous forme de cartes.

Portfolio GitHub

hassaoui-labs

mini-projet-python-cloud-collaboratif-g03

Commits : 26

Langages : Jupyter Notebook

★ Stars : 0

🍴 Forks : 0

🏆 Score : 52

[Voir le repository](#)

mini-projet-python-debugging-collaboratif-g01

Commits : 5

Langages : Python

★ Stars : 0

🍴 Forks : 0

🏆 Score : 10

[Voir le repository](#)

mini-projet-python-gestion-versions-g02

Commits : 2

Langages :

★ Stars : 0

🍴 Forks : 0

🏆 Score : 4

[Voir le repository](#)

mini-projet-python-portfolio-github-g04

Commits : 4

Langages : HTML, Python, CSS

★ Stars : 1

🍴 Forks : 0

🏆 Score : 13

[Voir le repository](#)

10. Conclusion

Cette application Python permet de générer automatiquement un portfolio GitHub en HTML à partir des données de l'API GitHub. La structure du projet est claire (séparation API / analyse / génération), le rendu HTML est lisible et le CSS offre une présentation professionnelle.