Spécifications techniques

Sommaire

Structure du projet

Le code se découpera en deux grandes parties indépendantes pour la répartition des tâches : le back-end et le front-end.

Back-end

Le back-end aura les responsabilités suivantes :

1. Récupérer les informations à partir de différentes sources
2. Enregistrer les informations de l'utilisateur
3. Traiter les données puis les envoyer au front-end dans un format adapté

API et base de données

Pour envoyer l'information au back-end, on a plusieurs choix possibles :

- framework Django
- framework Flask
- librairie FastAPI

Les deux premiers sont des frameworks, Django étant plus complet et lourd que Flask. Le rendu visuel sera affecté au front-end donc ces frameworks restent trop complexes pour la tâche à réaliser. On choisit finalement **FastAPI**.

Data Science

Il existe beaucoup de librairies sur le Machine Learning. La tâche est une simple régression linéaire. La librairie adaptée est **scikit-learn**. On couplera cette dernière avec Pandas et numpy.

Front-end

Intéractions

Pour la partie front-end, on souhaite pouvoir gérer les intéractions de l'utilisateur (zoom, filtre, focus sur une donnée,...) de manière optimale. On utilisera ainsi un framework Javascript, soit un des choix suivants :

- React
- Vue
- Angular
- Svelte
- Solid
- Alpine

Pour une question de préférence, on partira sur **Solid**. La librairie a déjà été utilisée par un des membres du projet : bonne performance, rapide à utiliser, peu complexe dans l'usage, s'apparente à React dans la structure du code.

Représentations visuelles

On couplera ainsi le framework avec la librairie **D3.js**, qui nous permet de créer des visuels pour les données.
Exemple de visuels :

- https://observablehq.com/@d3/stacked-bar-chart/2
- https://observablehq.com/@d3/icelandic-population-by-age-1841-2019
- https://observablehq.com/@d3/bar-chart-race
- https://observablehq.com/@d3/index-chart/2
- https://observablehq.com/@d3/world-tour
- https://observablehq.com/@d3/radial-area-chart

Conception

Fonctionnalités	Implémentations
Enregistrer son profil d'investisseur (prudent, courageux,...) et proposer un preset de paramètres cohérents	
Enregistrer les investissements déjà réalisés	
Afficher les ressources prévisionnelles de l'utilisateur	
Afficher l'état des marchés boursiers	- Back : récupérer les principaux index boursiers à partir de l'API - Front : afficher l'évolution des index sous formes de Line Chart
Rechercher des valeurs sur les marchés	- Back : récupérer les cotations trouvés à partir de la recherche via l'API - Front : afficher leur évolution dans le Line Chart précédent
Enregistrer l'investissement initial	- Back : enregistrer la donnée dans une base de données - Front : afficher un champ de formulaire pour enregistrer l'information
Afficher les performances du portefeuille en fonction de la méthode d'investissement	- Back : calculer le DCA par type de périodicité et le Lump Sum - Front : cf. affichage du rendement du portefeuille
Afficher le rendement du portefeuille en considérant les tendances à la baisse	
Idem en considérant les tendances à la hausse	
Idem en considérant la tendance moyenne	- Back : calculer la valeur de l'investissement et les intérêts composés (cf. CAGR) - Front : Afficher sous forme de Line Chart
Représenter la volatilité d'une cotation	

Représenter la volatilité du portefeuille	- Back : calculer la volatilité moyenne de toutes les cotations du portefeuille - Front : afficher la répartition des titres sous forme de camembert
Proposer d'autres cotations pour reconstituer le portefeuille	
Proposer un ensemble cohérent de valeurs selon les informations de l'utilisateur	
Afficher l'ensemble des indicateurs liés au portefeuille	
Enregistrer la devise du foyer fiscal	- Back : enregistrer la donnée dans une base de données - Front : afficher un champ de formulaire pour enregistrer l'information, en proposant une liste déroulante des devises
Enregistrer les frais de gestion	- Back : enregistrer les frais pour chaque achat et vente, en - Front : afficher un champ de formulaire pour enregistrer l'information

Lancement du code

Il faut au préalable avoir Python et Node.js sur la machine pour lancer le code.

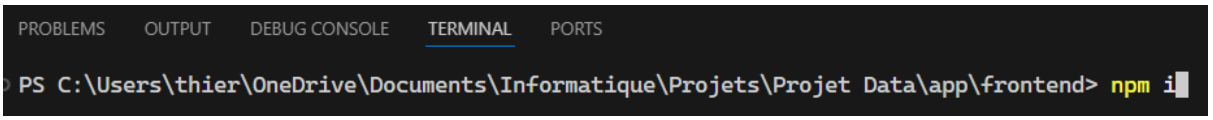
API

Installation des librairies Python :

Lancement du serveur backend :

Front-end

Installation des librairies Javascript :



Lancement du serveur frontend :