



RAPPORT

pour le projet Développement web

Master of Science 1

sur le sujet

Count Of Money

rédigé par

DAMODARANE Jean-Baptiste

TALATIZI Kamel

ZHANG Victor



Janvier 2024

TABLE DES MATIÈRES

TABLE DES MATIÈRES	2
1. Introduction	3
1.1 Contexte du projet	3
1.2 Problématique	3
1.3 Objectifs	3
1.4 Méthodologie	4
2. Analyse des besoins et spécifications fonctionnelles	4
2.1 Analyse des besoins utilisateurs	4
2.2 Spécifications fonctionnelles	4
3. Technologies et outils utilisés	5
3.1 Frontend : Angular	5
3.2 Backend : Express.js	5
3.3 Base de données : MongoDB	5
3.4 Conteneurisation avec Docker	6
3.5 Outils de gestion et de test	6
4. Architecture logicielle	6
4.1 Schéma de l'architecture générale	6
Diagramme d'architecture Logicielle	7
5. Sécurité	7
5.1 Authentification et sécurité des sessions	7
Diagramme de fonctionnement JWT	8
Chiffrement des mots de passe	8
Diagramme de fonctionnement de Hachage de mot de passe	9
5.2 Contrôle d'accès	9
Diagramme d'architecture de la sécurité des accès	9
6. Développement des fonctionnalités supplémentaires	10
6.1 Système de chat	10
6.1.1 Interface utilisateur du chat	10
6.1.2 Gestion de la sécurité dans le chat	10
7. Tests et validation	11
7.1 Tests fonctionnels	11
Conclusion	12
Perspectives d'amélioration	12

1. Introduction

1.1 Contexte du projet

Ce projet a été développé dans le cadre du module développement web du Semestre 7 du Master à l'EPITECH.

Dans un environnement de plus en plus orienté vers les technologies financières, le domaine des cryptomonnaies s'est imposé comme l'une des innovations les plus disruptives des dernières décennies. Depuis l'avènement du Bitcoin en 2009, les cryptomonnaies sont devenues un sujet central pour les investisseurs, les développeurs, et les régulateurs financiers du monde entier. Aujourd'hui, des millions d'utilisateurs s'intéressent à ces actifs numériques, et les investisseurs, qu'ils soient novices ou experts, recherchent des outils fiables pour les aider à analyser les cours et les tendances des crypto-monnaies.

Cependant, l'écosystème des crypto monnaies étant extrêmement volatil, il est crucial de disposer d'une plateforme fiable qui non seulement fournit des données en temps réel sur les crypto monnaies, mais qui offre aussi des perspectives et des actualités spécialisées pour mieux comprendre les évolutions des marchés.

1.2 Problématique

Dans un environnement aussi dynamique et fluctuant que celui des cryptomonnaies, comment peut-on fournir une information à la fois pertinente, à jour et accessible pour une large audience ?

C'est dans cette optique que le projet a vu le jour. Notre plateforme de **suivi des cryptomonnaies**, CryptoBreak, vise à répondre à cette problématique en offrant une vue d'ensemble des prix des actifs numériques tout en intégrant une revue de presse actualisée régulièrement, permettant ainsi aux utilisateurs de disposer des dernières informations du secteur.

1.3 Objectifs

Ce projet a pour objectif de développer une plateforme web complète dédiée au **suivi des cours de crypto-monnaies** et à la **revue de presse spécialisée**. Voici les fonctionnalités de notre projet :

1. **Suivre les prix** des crypto-monnaies en temps réel, avec des graphiques interactifs qui couvrent différentes périodes (1 jour, 30 jours, 90 jours, 365 jours).
2. **Accéder à des actualités spécialisées** dans le domaine des cryptomonnaies via une revue de presse actualisée fréquemment grâce à une API dédiée.
3. **Gérer leur expérience utilisateur** via des fonctionnalités de personnalisation, comme le choix des crypto-monnaies à enregistrer, ou encore la consultation des résumés d'articles.
4. **Se connecter facilement** via un système d'authentification sécurisé, y compris des connexions sociales (Google).
5. **Communiquer entre utilisateurs** via une interface de chat intégrée semblable à Discord.

1.4 Méthodologie

Le projet a suivi une approche méthodologique rigoureuse, articulée autour des phases suivantes :

- **Analyse des besoins utilisateurs** pour identifier les fonctionnalités clés à développer.
- **Choix des technologies** en fonction des exigences de scalabilité, de sécurité et de rapidité d'exécution.
- **Développement des modules frontend et backend** en utilisant une architecture modulaire et une séparation claire des responsabilités.
- **Intégration des API externes** pour la récupération des données sur les cryptomonnaies et des flux d'actualités.
- **Tests et validation** pour assurer la robustesse, la sécurité et la performance de la plateforme.

2. Analyse des besoins et spécifications fonctionnelles

2.1 Analyse des besoins utilisateurs

Pour construire une plateforme répondant aux attentes des utilisateurs, il a été essentiel de définir trois types d'utilisateurs cibles :

- **Anonymes** : Ces utilisateurs peuvent accéder aux fonctionnalités basiques de la plateforme sans créer de compte.
- **Utilisateurs enregistrés** : En créant un compte, les utilisateurs ont accès à des fonctionnalités supplémentaires telles que l'accès aux articles et l'accès à l'espace communautaire.
- **Administrateurs** : Les administrateurs disposent d'un accès privilégié pour gérer les données de la plateforme. Ils peuvent, entre autres, ajouter de nouvelles cryptomonnaies et ajuster les paramètres globaux.

2.2 Spécifications fonctionnelles

Voici un aperçu des spécifications fonctionnelles du projet :

- **Suivi des crypto monnaies** :
 - Affichage en temps réel des prix des crypto-monnaies les plus populaires (par défaut) ou celles choisies par l'utilisateur.
 - Visualisation des graphiques d'évolution des cryptomonnaies sur différentes périodes : 1 jour, 30 jours, 90 jours, 365 jours.
 - Mise à jour automatique des données via l'API **CoinGecko**.
- **Revue de presse** :
 - Intégration de l'API **newsdata.io** pour récupérer les actualités liées aux crypto-monnaies.
 - Actualisation des données fréquemment.
 - Résumés automatiques des articles pour un aperçu rapide.

- **Authentification :**
 - Authentification sécurisée avec **JWT** (JSON Web Token).
 - Connexion via Google OAuth2 pour simplifier le processus d'enregistrement.
 - Gestion des rôles utilisateurs (utilisateurs standards et administrateurs).
- **Chat intégré :**
 - Interface inspirée de **Discord** permettant aux utilisateurs connectés d'échanger des messages en temps réel.

3. Technologies et outils utilisés

Le développement de la plateforme a fait appel à plusieurs technologies et outils pour garantir la sécurité, la performance et la scalabilité du projet.

3.1 Frontend : Angular

Angular a été choisi pour le développement du frontend en raison de ses nombreux avantages : modularité, maintenabilité, et compatibilité avec les composants réactifs.

Fonctionnalités clés :

- Structure modulaire : Favorise le découpage en composants indépendants (cryptomonnaies, revue de presse, chat).
- Responsive design : Adaptation automatique de l'interface sur mobile, tablette et ordinateur.

3.2 Backend : Express.js

Pour le backend, le choix s'est porté sur **Express.js**, un framework minimaliste pour Node.js qui permet de gérer rapidement des applications web ou des APIs. Express.js est particulièrement apprécié pour sa flexibilité, son écosystème de middlewares, et sa capacité à gérer des requêtes HTTP de manière efficace.

Le backend assure la communication avec les API externes (CoinGecko et newsdata.io) et gère les interactions avec la base de données MongoDB.

- Gestion des routes via Express Router.
- Middleware pour authentification et autorisation des utilisateurs via **JWT**.
- Intégration avec MongoDB pour la persistance des données utilisateurs.

3.3 Base de données : MongoDB

Le choix de **MongoDB** s'est avéré pertinent pour gérer des données non structurées et offrir une flexibilité dans le stockage des informations sur les utilisateurs, les cryptomonnaies, et les articles de presse. MongoDB, en tant que base de données NoSQL, permet de stocker des objets JSON et de s'adapter à des structures de données évolutives.

- **Collections utilisateurs** : Stockage des informations de connexion, des préférences utilisateur, et des historiques de notifications.
- **Collections cryptomonnaies** : Conservation des données sur les crypto-monnaies suivies, avec des mises à jour en temps réel.
- **Collections articles** : Stockage des articles de presse récupérés via l'API newsdata.io.

3.4 Conteneurisation avec Docker

Pour garantir la portabilité et la reproductibilité de l'environnement de développement, nous avons utilisé **Docker**. Chaque composant de l'application (frontend, backend, base de données) est conteneurisé, ce qui permet de déployer l'application sur n'importe quel serveur sans conflit de dépendances.

- Conteneurisation de l'API backend et du frontend Angular.
- MongoDB est également déployé dans un conteneur séparé.
- Utilisation de **Docker Compose** pour orchestrer les différents services.

3.5 Outils de gestion et de test

- **GitHub** : Utilisé pour la gestion du code source et la collaboration entre les membres de l'équipe. La gestion des versions a été assurée via **Git**, avec un workflow GitHub basé sur des branches pour faciliter les intégrations.
- **ClickUp** : Outil de gestion de projet pour organiser les tâches, suivre l'avancement et respecter les échéances. Les fonctionnalités de ClickUp ont permis d'assurer une collaboration fluide et d'attribuer les responsabilités à chaque membre de l'équipe.
- **Postman** : Utilisé pour tester et valider les endpoints de l'API backend. Postman a permis d'automatiser les tests et de garantir la conformité des réponses de l'API avec les spécifications fonctionnelles.

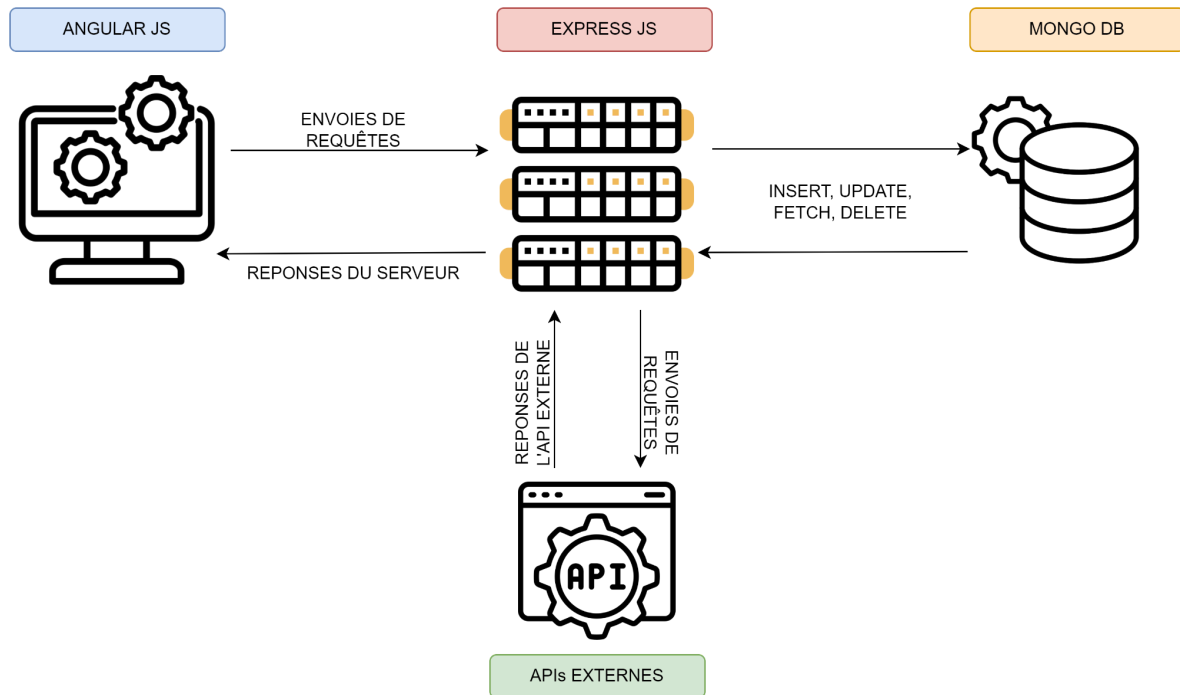
4. Architecture logicielle

4.1 Schéma de l'architecture générale

L'architecture de l'application repose sur une architecture en **microservices**, avec des interactions entre le frontend, le backend, la base de données MongoDB, et les API externes (CoinGecko, newsdata.io). Le schéma suivant illustre l'architecture générale :

- **Frontend** : Interface utilisateur développée avec Angular, communiquant avec le backend via des requêtes HTTP.
- **Backend** : API REST construite avec Express.js, gérant les requêtes utilisateurs et les appels aux API externes.
- **Base de données MongoDB** : Base NoSQL assurant la persistance des données.
- **APIs externes** : Utilisation de CoinGecko pour les données des cryptomonnaies et newsdata.io pour les flux d'actualités.

Diagramme d'architecture Logicielle



5. Sécurité

La sécurité d'une plateforme dédiée au suivi des cryptomonnaies est primordiale, compte tenu des risques élevés d'attaques sur les données sensibles des utilisateurs. Voici les détails des mécanismes de sécurité que nous avons mis en place dans ce projet.

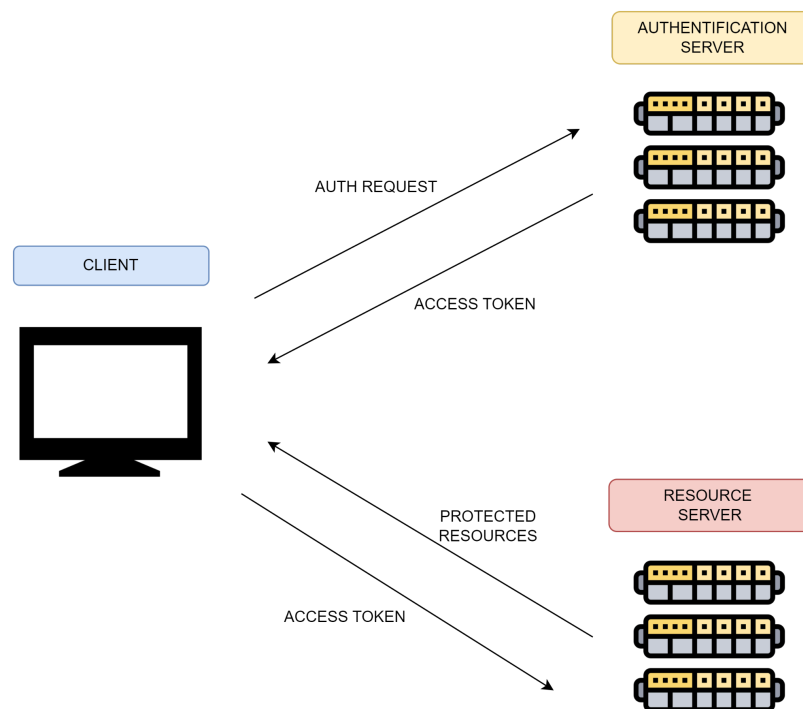
5.1 Authentification et sécurité des sessions

L'authentification des utilisateurs repose sur l'utilisation de **JWT (JSON Web Tokens)**, un standard ouvert permettant de sécuriser les sessions utilisateurs sur la plateforme. Le processus suit les étapes suivantes :

- **Création du Token** : Lorsqu'un utilisateur se connecte, un token est généré. Ce token contient des informations sur l'utilisateur (ID, rôle, etc.) et est signé avec une clé secrète pour empêcher toute modification frauduleuse.
- **Stockage du Token** : Le token JWT est envoyé au client (navigateur ou application mobile) et est stocké dans le **localStorage** ou dans un **cookie sécurisé**. Ce choix dépend des configurations de sécurité du client.
- **Validation du Token** : À chaque requête faite par l'utilisateur vers le serveur (par exemple, pour accéder à ses préférences ou à des articles sauvegardés), le token est envoyé dans les en-têtes HTTP et vérifié côté serveur pour s'assurer qu'il est valide.

- **Expiration du Token** : Les tokens sont configurés pour expirer après une certaine période (généralement 1 à 2 heures). Une fois expiré, l'utilisateur doit se reconnecter pour recevoir un nouveau token.
- **Révocation des Tokens** : En cas de connexion suspecte ou d'une tentative de piratage, les administrateurs peuvent révoquer manuellement les tokens pour bloquer l'accès aux comptes compromis.

Diagramme de fonctionnement JWT

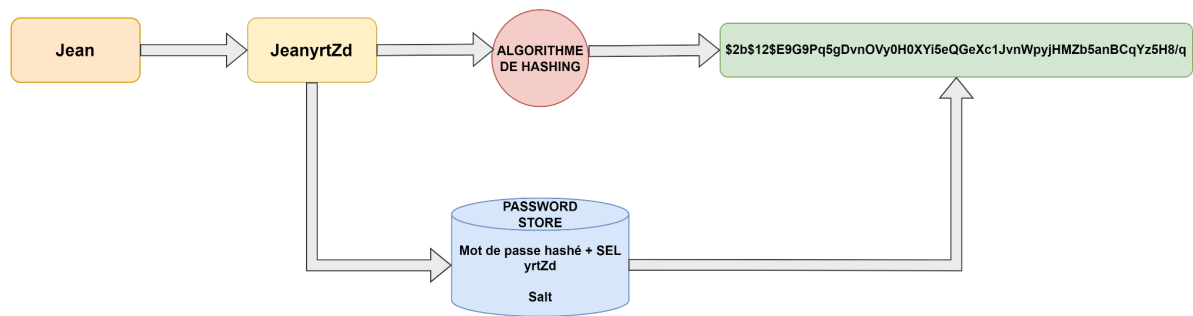


Chiffrement des mots de passe

Les mots de passe des utilisateurs sont chiffrés avant d'être stockés dans la base de données. Nous utilisons l'algorithme **bcrypt**, un algorithme de hachage bien établi, pour garantir que les mots de passe sont protégés. Voici un résumé du processus :

1. Lors de l'enregistrement, le mot de passe de l'utilisateur est "haché" avec une **valeur de sel** aléatoire.
2. Ce hachage est ensuite stocké dans la base de données.
3. À chaque tentative de connexion, le mot de passe fourni est de nouveau haché, et ce hachage est comparé à celui enregistré.

Diagramme de fonctionnement de Hachage de mot de passe

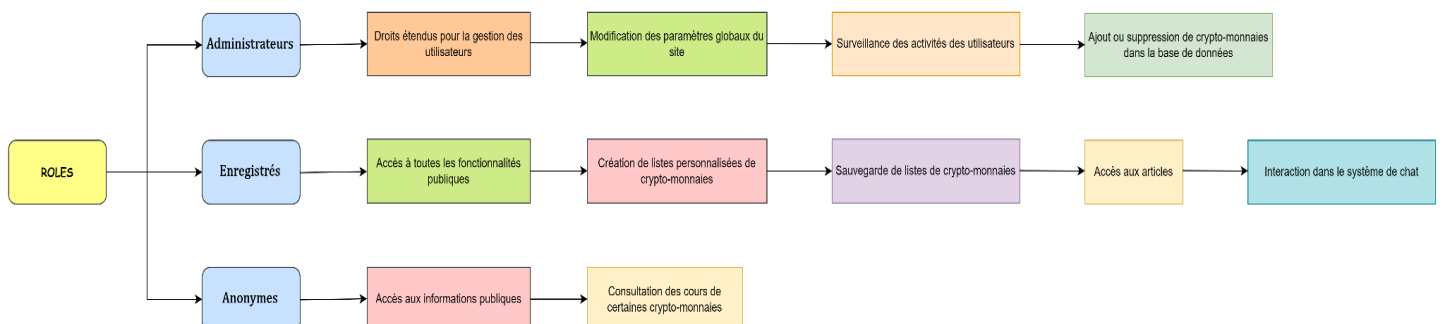


5.2 Contrôle d'accès

Pour assurer une gestion sécurisée des données et des fonctionnalités, nous avons implémenté un système de **contrôle d'accès basé sur les rôles (Role-Based Access Control - RBAC)**. Les différents types d'utilisateurs (anonymes, utilisateurs enregistrés, administrateurs) disposent de privilèges distincts.

1. **Utilisateurs anonymes** : Ils peuvent accéder aux informations publiques telles que les cours de certaines crypto-monnaies.
2. **Utilisateurs enregistrés** : Ces utilisateurs peuvent accéder à toutes les fonctionnalités publiques et personnalisées telles que la création et la sauvegarde de listes de crypto-monnaies et l'accès aux articles. Ils peuvent également interagir dans le système de chat.
3. **Administrateurs** : Les administrateurs ont des droits étendus, leur permettant de gérer les utilisateurs, de modifier les paramètres globaux du site, et de surveiller quelques activités. Ils peuvent également ajouter ou supprimer des crypto-monnaies dans la base de données.

Diagramme d'architecture de la sécurité des accès



6. Développement des fonctionnalités supplémentaires

6.1 Système de chat

L'ajout d'un système de chat intégré permet aux utilisateurs de discuter entre eux en temps réel. Cette fonctionnalité, n'étant pas complètement implémentée, permet de renforcer l'aspect communautaire de la plateforme et offre un espace de discussion sur les tendances des crypto-monnaies.

6.1.1 Interface utilisateur du chat

L'interface du chat est inspirée de la plateforme Discord, offrant une **navigation par canaux** pour faciliter la communication. Les utilisateurs peuvent :

- Envoyer des messages en temps réel.
- Partager des liens et des articles pertinents sur les crypto-monnaies.

6.1.2 Gestion de la sécurité dans le chat

Pour sécuriser le système de chat, nous avons mis en place une mesure très importante :

- **Authentification obligatoire** : Seuls les utilisateurs enregistrés peuvent participer au chat.

7. Tests et validation

La phase de tests et de validation est cruciale pour garantir que l'application fonctionne comme prévu et qu'elle peut gérer la montée en charge prévue. Plusieurs types de tests ont été effectués.

7.1 Tests fonctionnels

Les tests fonctionnels ont été réalisés pour s'assurer que chaque fonctionnalité de l'application fonctionne correctement, y compris l'authentification, le suivi des cryptomonnaies, la revue de presse et le chat. Voici les outils et méthodes utilisés :

1. Postman :

Pour tester les requêtes API, nous avons utilisé Postman, qui permet de simuler les requêtes et de vérifier les réponses du serveur.

- **Test de l'authentification** : Validation des processus de création de comptes, de connexion et de gestion des tokens JWT.
- **Test du suivi des cryptomonnaies** : Vérification de la récupération des données en temps réel via l'API CoinGecko et l'affichage correct des informations sur le frontend.
- **Test de la revue de presse** : Validation du filtrage des articles, de l'intégration de l'API newsdata.io et de la fonctionnalité de résumé automatique.
- **Test du chat** : Simulation de plusieurs utilisateurs en simultané pour s'assurer que le système de chat en temps réel fonctionne comme prévu.

2. Tests de performance:

Des évaluations de performance ont été réalisées en invitant plusieurs utilisateurs à interagir avec l'application sur un environnement local. Ces tests ont permis d'observer la réactivité de l'application dans des scénarios d'utilisation variés, en simulant différents comportements d'utilisateurs actifs. Les retours des utilisateurs ont été recueillis pour évaluer comment l'application se comportait lors de l'utilisation simultanée par plusieurs personnes. Les résultats ont été analysés afin de garantir que les temps de réponse demeurent acceptables et que l'expérience utilisateur ne soit pas compromise, même lors des pics d'utilisation. Cette approche a permis d'identifier les points à améliorer pour assurer une performance optimale lors de l'utilisation réelle de l'application.

Conclusion

Le projet de développement d'une plateforme de suivi des crypto monnaies a permis de réaliser une application web robuste, sécurisée et fonctionnelle, capable de répondre aux besoins des utilisateurs en matière d'information et d'interaction sur le marché dynamique des crypto monnaies. Grâce à une architecture bien pensée, l'application intègre des fonctionnalités essentielles telles que l'authentification sécurisée par JWT, un système de chat inspiré des meilleures pratiques des plateformes sociales, et des notifications personnalisables qui permettent aux utilisateurs de rester informés des évolutions de leurs actifs numériques.

L'accent mis sur la sécurité, notamment avec le chiffrement des mots de passe et le contrôle d'accès basé sur les rôles, garantit que les données des utilisateurs sont protégées contre les menaces potentielles. De plus, les tests fonctionnels et de performance effectués tout au long du développement ont assuré que chaque fonctionnalité répond aux exigences établies et peut supporter un grand nombre d'utilisateurs simultanés.

La mise en place de systèmes innovants tels qu'une revue de presse en temps réel et un système de notifications adaptatives enrichit l'expérience utilisateur, tout en rendant la plateforme plus interactive et engageante. L'intégration de ces fonctionnalités répond non seulement aux attentes actuelles des utilisateurs, mais aussi à leurs besoins futurs dans un environnement technologique en constante évolution.

Perspectives d'amélioration

En vue d'améliorer encore cette application, plusieurs pistes ont été identifiées :

1. **Notifications avancées** : Améliorer le système de notifications pour inclure des alertes en fonction de l'analyse des tendances du marché et des mouvements significatifs des prix, augmentant ainsi l'engagement des utilisateurs.
2. **Intégration de nouvelles API** : Explorer l'intégration d'autres API pour enrichir les sources d'informations disponibles, permettant aux utilisateurs d'accéder à des analyses de marché et des conseils d'experts.
3. **Améliorations de l'interface utilisateur** : Recueillir les retours des utilisateurs pour améliorer continuellement l'interface utilisateur et l'expérience globale, en veillant à ce qu'elle reste intuitive et agréable.