

Cahier des Charges

Nom du projet : **FinTrack (Application de Gestion Financière)**

1. Présentation générale du projet

L'application **FinTrack** vise à offrir aux utilisateurs une solution simple et efficace pour gérer leurs finances personnelles. Elle permet de suivre les revenus et les dépenses, de fixer des objectifs financiers, et de visualiser des graphiques comparant les dépenses aux revenus. De plus, l'application intègre des rappels pour aider les utilisateurs à respecter leurs échéances et à ne pas dépasser un seuil de dépenses défini. Le projet sera développé avec **React** pour l'interface utilisateur, **PostgreSQL** pour la gestion des données, et **Tailwind CSS** pour le design.

2. Objectifs du projet

- Fournir aux utilisateurs un outil simple et intuitif pour gérer leurs finances.
 - Permettre le suivi des revenus et des dépenses, ainsi que la budgétisation.
 - Offrir la possibilité de définir et de suivre des objectifs financiers.
 - Implémenter des fonctionnalités de rappels automatiques pour les dépenses et les dettes à venir.
 - Intégrer un système de notifications en cas de dépassement de seuil de dépenses.
 - Générer des rapports financiers basés sur l'historique des revenus et des dépenses.
-

3. Fonctionnalités attendues

3.1. Authentification et gestion des utilisateurs

- Inscription (création de compte avec un mot de passe sécurisé).
- Connexion via un système d'authentification basé sur PostgreSQL (utilisation de JWT pour la sécurité).
- Gestion des informations personnelles : nom, email, photo de profil.

3.2. Gestion des revenus

- Ajouter, modifier, et supprimer des sources de revenus.
- Affichage d'un historique des revenus.
- Calcul automatique du revenu total sur une période donnée.

3.3. Gestion des dépenses

- Ajouter, modifier, et supprimer des dépenses.
- Catégorisation des dépenses (logement, alimentation, loisirs, etc.).
- Alertes en cas de dépassement du seuil de dépenses fixé par l'utilisateur.

3.4. Gestion des objectifs financiers

- Permettre à l'utilisateur de définir des objectifs financiers (ex : épargne, remboursement de dette).
- Suivi de la progression des objectifs.
- Notification lorsque l'objectif est atteint ou en retard.

3.5. Rappels et alertes

- Ajouter des rappels pour les échéances de paiement (dettes, factures, etc.).
- Notifications automatiques via mail pour alerter l'utilisateur des échéances importantes.
- Rappels de dépenses à venir (exemple : loyer mensuel).

3.6. Budgétisations

- Définir des budgets à court et au long terme.
- Visualiser les budgets définis.

3.7. Rapports financiers

- Générer des rapports graphiques des revenus et des dépenses.
 - Comparaison des dépenses par catégorie sur une période donnée.
 - Afficher les tendances et proposer des conseils pour améliorer la santé financière.
-

4. Technologies utilisées

4.1. Frontend :

- **React.js** : pour la construction de l'interface utilisateur.
- **Tailwind CSS** : pour le design et la mise en page.
- **FontAwesome** : pour les icônes utilisés dans l'interface.

4.2. Backend :

- **Node.js** avec **Express** : pour gérer les API et les interactions côté serveur.
- **PostgreSQL** : comme base de données relationnelle pour stocker les données des utilisateurs et des transactions.

4.3. Authentification :

- **JWT** (JSON Web Token) : pour sécuriser les sessions utilisateurs.
- **PostgreSQL** : pour stocker les informations d'identification des utilisateurs.

4.4. Notifications :

- **Service Worker** : pour implémenter les notifications de dépenses et rappels.
-

5. Interface utilisateur

L'interface utilisateur doit être :

- **Responsive** : s'adapter aux écrans de différentes tailles (desktop, tablette, mobile).
- **Ergonomique** : facile à naviguer avec une mise en page claire et intuitive.
- **Personnalisée** : permettre à l'utilisateur de voir ses informations personnelles dès l'accueil.

Pages principales :

1. **Accueil** : Vue générale avec un aperçu des revenus, des dépenses, et des objectifs.
2. **Revenus** : Gestion des sources de revenus (ajout, suppression, modification).
3. **Dépenses** : Gestion des dépenses avec des alertes en cas de dépassement de seuil.
4. **Budgets** : Création et suivi de budgets mensuels.
5. **Rappels** : Gestion des rappels de paiements et des échéances financières.
6. **Objectifs Financiers** : Suivi des objectifs financiers à court et long terme.
7. **Rapports** : Vue analytique des tendances financières de l'utilisateur (graphiques et tableaux).
8. **Conseils** : visualiser des conseils prédéfinis pour mieux gérer ces finances.

6. Livrables

- **Code source** : Hébergé sur GitHub ou GitLab, bien documenté.
- **Documentation utilisateur** : Guide d'utilisation pour les fonctionnalités principales.
- **Documentation technique** : Documentation pour les développeurs expliquant l'architecture du système, la configuration de la base de données, et les API.

7. Maintenance et évolutions futures

- Mise en place d'un système de suivi des bugs et des fonctionnalités futures (ex : GitHub Issues ou Jira).
- Prévoir des évolutions possibles comme l'intégration de services bancaires via API, ou des outils d'investissement.

9. Conclusion

Le projet **FinTrack** vise à offrir une solution moderne et efficace pour aider les utilisateurs à gérer leurs finances personnelles. Les fonctionnalités telles que la gestion des revenus, des dépenses, des objectifs financiers, ainsi que des rappels et des notifications, permettront aux utilisateurs d'avoir une vue d'ensemble sur leurs finances et d'éviter des situations financières difficiles.