



Profil des clients:

Les clients de BitChest sont des utilisateurs inscrits et ayant approvisionné leur compte BitChest avec des fonds depuis leur compte bancaire personnel. Ils sont des investisseurs intéressés par les crypto-monnaies et souhaitent gérer leur portefeuille de manière efficace grâce à l'interface d'administration privée.

Fonctionnalités essentielles pour les clients:

- Gestion des données personnelles :
 - Les clients doivent pouvoir mettre à jour leurs informations personnelles, telles que leur nom, adresse e-mail, etc.
 - Gestion du portefeuille :
 - Les clients doivent avoir accès à une liste de toutes les crypto-monnaies qu'ils possèdent dans leur portefeuille.
 - Pour chaque crypto-monnaie, ils doivent pouvoir consulter les détails des achats antérieurs, y compris la date, la quantité et le cours.
 - La plus-value actuelle pour chaque crypto-monnaie doit être affichée, indiquant les gains potentiels en cas de vente.
 - Solde en euro :
 - Le solde en euros doit être toujours visible pour que les clients aient une idée claire de la valeur totale de leur portefeuille en devise fiat.
 - Vente de crypto-monnaie :
 - Les clients doivent être en mesure de vendre la totalité de leur détention d'une crypto-monnaie.
 - Les fonds en euros correspondants doivent être crédités sur leur compte BitChest
-

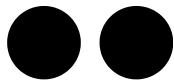


- Consultation des cours des crypto-monnaies :
 - Les clients doivent pouvoir consulter les cours actuels de toutes les crypto-monnaies prises en charge par BitChest.
- Liste des crypto-monnaies :
 - Une liste des crypto-monnaies prises en charge (Bitcoin, Ethereum, Ripple, etc.) doit être affichée.
- Courbes de progression :
 - Les clients doivent pouvoir visualiser la courbe de progression de chaque crypto-monnaie au fil du temps, pour évaluer leur performance.
- Achat de crypto-monnaies :
 - Les clients doivent pouvoir acheter une quantité de crypto-monnaie au cours actuel.

Restrictions importantes pour les clients:

- Les clients ne peuvent pas vendre une quantité arbitraire de crypto-monnaies ; ils ne peuvent vendre que la totalité de leur détention.

En résumé, l'interface d'administration des clients de BitChest doit fournir des fonctionnalités complètes pour gérer leur portefeuille de crypto-monnaies, y compris la visualisation des performances, la gestion des ventes, la consultation des cours et la possibilité d'acheter de nouvelles crypto-monnaies. Il est essentiel de maintenir une visibilité constante sur le solde en euros pour une gestion financière transparente.



Semaine 1 – Planification et Conception :

- Réunion de lancement du projet avec l'équipe.
-
- Définition des objectifs et des exigences du projet.
- Conception de l'architecture de base de la plateforme.
- Création du dépôt GitHub et configuration initiale.
- Sélection des technologies clés à utiliser.
- Création d'une maquette initiale de l'interface utilisateur.
- Définition des rôles et des responsabilités de l'équipe.

Semaine 2 – Développement initial :

- Mise en place de l'environnement de développement.
- Développement du backend de base pour la gestion des comptes utilisateurs.
- Intégration d'une API de cotation de crypto-monnaies pour afficher les prix en temps réel.
- Développement du processus d'achat et de vente initial de crypto-monnaies.
- Création de la page d'accueil de la plateforme.
- Tests unitaires et d'intégration initiaux.

Semaine 3 – Amélioration de l'interface utilisateur et Sécurité :

- Affinement de l'interface utilisateur (UI/UX).
 - Ajout de fonctionnalités pour gérer les portefeuilles de crypto-monnaies.
 - Mise en place de mesures de sécurité, notamment l'authentification à deux facteurs.
 - Tests de sécurité et de performance.
 - Documentation du code existant et mise en place d'un plan de documentation continue.
 - Début de la rédaction de tests automatisés.
-



Semaine 4 – Finalisation et Lancement :

- Correction de bugs et optimisation du code.
- Finalisation de la documentation utilisateur.
- Mise en place de l'hébergement en production.
- Configuration des outils de surveillance et de suivi des erreurs.
- Tests de dernière minute et gestion des problèmes.
- Lancement de la plateforme en version bêta restreinte.
- Planification de la phase de bêta-test et collecte des commentaires des utilisateurs.

Modélisation

crypto_wallets		currencies		currency_histories		personal_access_tokens	
id	bigint NN	id	bigint NN	id	bigint NN	id	bigint NN
created_at	timestamp	crypto_name	varchar(255) NN	quoting	int NN	tokenable_type	varchar(255) NN
updated_at	timestamp	created_at	timestamp	date	date NN	tokenable_id	bigint NN
sell_at	timestamp	updated_at	timestamp	currency_id	bigint NN	name	varchar(255) NN
capital_gain	decimal(8,2)					token	varchar(64) NN
quantity	bigint NN					abilities	text
purchase_date	timestamp					last_used_at	timestamp
purchase_price	decimal(10,2)					expires_at	timestamp
sale_price	decimal(10,2)					created_at	timestamp
profit	decimal(10,2)					updated_at	timestamp
user_id	bigint NN						
currency_id	bigint NN						

users		wallets	
id	bigint NN	id	bigint NN
email	varchar(255) NN	quantity	bigint NN
role	users_role_enum NN	created_at	timestamp
password	varchar(255) NN	updated_at	timestamp
remember_token	varchar(100)	user_id	bigint NN
created_at	timestamp		
updated_at	timestamp		