



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

# **LedgerLift**

Monark (16918140 Canada Inc.)

**Projet final en génie logiciel**

**Année 2025-2026**

**Document de vision**

# 1 Introduction

---

## 1.1 À propos de Monark

---

Monark est une initiative visant à créer un écosystème Web3 collaboratif et accessible pour les étudiants universitaires, les développeurs, les entrepreneurs, les passionnés de Web3 et les communautés locales. Elle fournit une plateforme centralisée pour partager des idées de projets, collaborer sur des applications open source, et accéder à des ressources techniques adaptées.

Voir plus de détails : <https://www.monark.io/fr/about>

## 1.2 Contexte

---

Le projet Ledger Lift s'inscrit dans le contexte de l'émergence des technologies Web3 et de la blockchain, qui transforment la manière dont les données financières sont enregistrées, partagées et auditées. De plus en plus d'organisations, y compris des entreprises, des gouvernements et des projets décentralisés, utilisent des portefeuilles blockchain pour effectuer des transactions en crypto-monnaies. Cependant, ces transactions sont souvent peu lisibles pour les non-initiés et difficiles à intégrer dans des systèmes comptables traditionnels.

Ledger Lift vise à combler cet écart en développant un outil qui extrait, normalise et catégorise automatiquement les données de transaction issues des portefeuilles blockchain, afin de générer des rapports financiers compréhensibles et exploitables. Ce projet permettra aux utilisateurs (individus, comptables, entreprises) de mieux comprendre leurs activités on-chain, d'en faire le suivi, et d'exporter des données structurées pour des usages comptables ou fiscaux.

## 2 Généralités

---

### 2.1 Environnement technique de développement

---

- Plate-forme d'hébergement web (Vercel)
- Web interface using Next.js, Typescript, shadcn-ui, tailwindcss
- Hébergement de bases de données (Clickhouse, PostgreSQL ou GraphQL) et de backend (Typescript, Express)
- [The Graph](#) pour lecture de données blockchain
- Smart contract library (EVM based, [Solidity](#), [OpenZeppelin](#))
- [Docker](#) pour l'environnement du conteneur (container environment)
- Documentation API utilisant [Swagger](#) et [OpenAPI](#)

### 2.2 Éléments fournis par le client

---

Monark soutiendra le projet grâce à son expertise en matière de blockchain et de développement de contrats intelligents. L'équipe actuelle participera à l'intégration et au transfert de connaissances. Un serveur Discord dédié sera utilisé pour la communication, et des réunions bimensuelles de type Sprint avec le CTO et/ou le COO coordonneront l'avancement du projet. Le CTO fournira également des conseils en matière d'interface utilisateur et d'interface utilisateur, des outils de démarrage de projet et des outils Monark pertinents. Une documentation complète sera disponible pour aider les étudiants à apprendre la technologie de la blockchain et à livrer un projet fonctionnel. En outre, Monark fournira des portefeuilles financés par Monark pour les opérations sur la chaîne, ainsi que des modèles de départ pour le frontend et le backend et l'intégration des portefeuilles afin d'accélérer le développement et de réduire les obstacles techniques.

Si des jetons sont nécessaires pour tester le projet, ceux-ci seront fournis par Monark.

### 2.3 Estimation de l'effort

---

1. Interface web utilisant Next.js, Typescript, shadcn-ui, tailwindcss : **150 heures**
2. Authentification des utilisateurs avec [Metamask SDK](#), [Wagmi](#) + [Viem](#) : **60 heures**
3. Collecte des transactions liées aux activités standards d'un portefeuille : **145 heures**
4. Logique de normalisation pour classer les transferts, échanges, etc : **125 heures**
5. Classification de chaque transaction comme revenu, dépense, échange ou frais : **80 heures**
6. Système d'exportation des données (CSV, JSON) avec champs étiquetés : **30 heures**

7. Statistiques de synthèse : revenus totaux, dépenses, frais, gains/pertes : **80 heures**
8. (Optionnel) Recherche de prix de tokens et génération de rapports fiscaux : **60 heures**
9. Contenant docker du backend configuré avec bases de données et autres services nécessaires : **80 heures**
10. Documentation du déploiement du backend : **40 heures**
11. Documentation API complète utilisant swagger et OpenAPI : **30 heures**
12. Documentation contrat intelligent : **30 heures**
13. Vidéo démo du projet : **5 heures**
14. Vidéo explicative du projet : **10 heures**

**Total : 925 heures**

## **2.4 Soutien au projet**

---

Un représentant du client sera disponible pour rencontrer l'équipe de développement à Polytechnique Montréal et/ou à distance, selon le contexte sociosanitaire.

Nom : Dominic Fournier

Titre : CTO

Téléphone : 438-820-4765

Courriel : [dominic@monark.io](mailto:dominic@monark.io)

### 3 Description du produit

---

[LedgerLift](#) est un projet d'analyse de blockchain centré sur le back-end destiné à la création de rapports financiers et à la préparation d'audits pour les individus, les DAO et les entreprises de crypto-monnaies. L'objectif est d'extraire l'historique des transactions, les frais de gaz, les échanges de tokens, les flux de revenus, et l'activité du portefeuille, puis de convertir ces données en formats lisibles par l'homme pour la comptabilité, la conformité, ou les rapports de trésorerie.

La plateforme se connecte aux portefeuilles fournis par les utilisateurs, récupère les données de transaction à l'aide d'API publiques ou de requêtes de sous-graphe, et classe les entrées (par exemple, dépense, revenu, transfert). Les étudiants peuvent ajouter des fonctionnalités comme l'étiquetage automatique, l'étiquetage des portefeuilles, ou l'agrégation de plusieurs portefeuilles. Les formats de sortie incluent des fichiers CSV, des résumés financiers, ou des données prêtes à être intégrées dans des outils de comptabilité comme QuickBooks ou Excel.

Ce projet offre une forte exposition à la normalisation des données, à l'analyse des métadonnées et aux heuristiques de classification. Des fonctionnalités supplémentaires peuvent inclure la récupération du prix des tokens au moment de la transaction, l'agrégation des coûts en gaz ou l'estimation du lot fiscal. En travaillant sur LedgerLift, les étudiants apprennent à combler le fossé technique entre les données de blockchain et les processus financiers hors chaîne.

Le système aide à résoudre l'un des besoins les plus urgents des équipes Web3 et des indépendants : rendre la finance native de la blockchain lisible et légalement utilisable dans un monde de réglementation traditionnelle.

Voici un lien vers une **maquette** du projet envisagé par Monark : <https://ledgerlift.monark.io/>

**Livrables & Fonctionnalités Souhaitées :**

1. Interface web utilisant Next.js, Typescript, shadcn-ui, tailwindcss
  - a. Créer un outil de génération de rapports financiers pour les transactions en crypto.
2. Authentification des utilisateurs avec [Wagmi](#) + [Viem](#)
  - a. Les utilisateurs se connectent avec leurs portefeuilles pour consulter et extraire l'historique de leurs transactions.
3. Collecte des transactions liées aux activités standards d'un portefeuille
  - a. Système de récupération des transactions effectuées sur un portefeuille crypto standard.
4. Logique de normalisation pour classier les transferts, échanges, etc
  - a. Organisation automatique des transactions selon leur type : transferts, échanges (swaps), frais de réseau.
5. Classification de chaque transaction comme revenu, dépense, échange ou frais
  - a. Chaque transaction est catégorisée pour faciliter les calculs financiers.
6. Système d'exportation des données (CSV, JSON) avec champs étiquetés
  - a. Permet d'exporter facilement les données structurées pour les comptables et les outils spécialisés.
7. Statistiques de synthèse : revenus totaux, dépenses, frais, gains/pertes
  - a. Affiche les revenus, dépenses, frais et profits/pertes sur une période donnée.
8. (Optionnel) Recherche de prix de tokens et génération de rapports fiscaux
  - a. Intégration d'un API pour récupérer les prix des tokens et générer des rapports fiscaux automatisés.
9. Contenant docker du backend configuré avec bases de données et autres services nécessaires
10. Documentation du déploiement du backend
11. Documentation API complète utilisant swagger et OpenAPI
12. Documentation contrat intelligent
13. Vidéo démo du projet
14. Vidéo explicative du projet