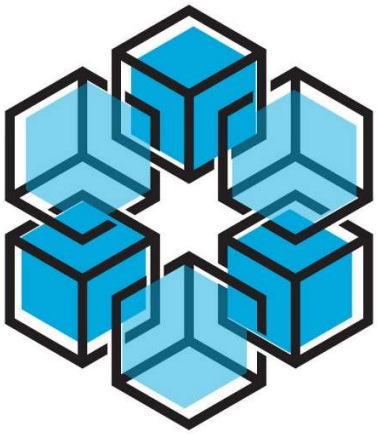
Rapport de projet : **UEMF Blockchain - Proof-of-Work Implementation**



Préparé par:

Fatima BOUYARMANE

Encadré par:

M. TMIMI

Le code source: https://github.com/FatimaBouyarmane/Projet-Blockchain/

Table de matières:

Introduction:

Ce projet consiste à créer une blockchain locale, c'est-à-dire une chaîne de blocs qui tourne sur un ordinateur personnel. Elle utilise le mécanisme de proof-of-work (PoW), où des mineurs doivent résoudre des calculs complexes pour valider les blocs et sécuriser la blockchain.

Ce travail permet de mieux comprendre le fonctionnement des blockchains, la gestion des transactions, et le rôle des mineurs dans le processus de validation via le proof-of-work.

Architecture du système:

Le serveur

Le serveur utilise Express pour gérer la blockchain et les transactions. Il :

- Stocke les blocs validés.
- Vérifie les transactions reçues (signature, solde, frais).
- Garde une liste des transactions en attente (mempool).
- Crée un nouveau bloc toutes les 10 minutes avec les transactions validées.

Le client

Le client fait deux choses :

- Miner les blocs en résolvant le proof-of-work pour valider les blocs.
- Envoyer des transactions signées au serveur.
  Il crée aussi un portefeuille avec une clé publique et une clé privée.

L'interface web

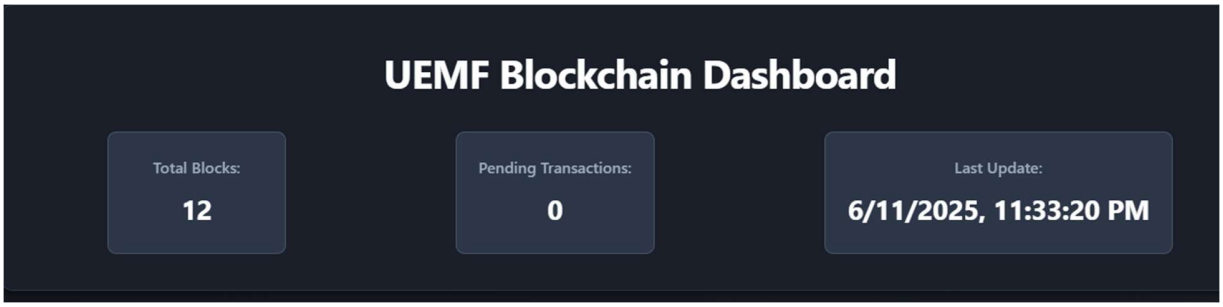
L'interface web permet de :

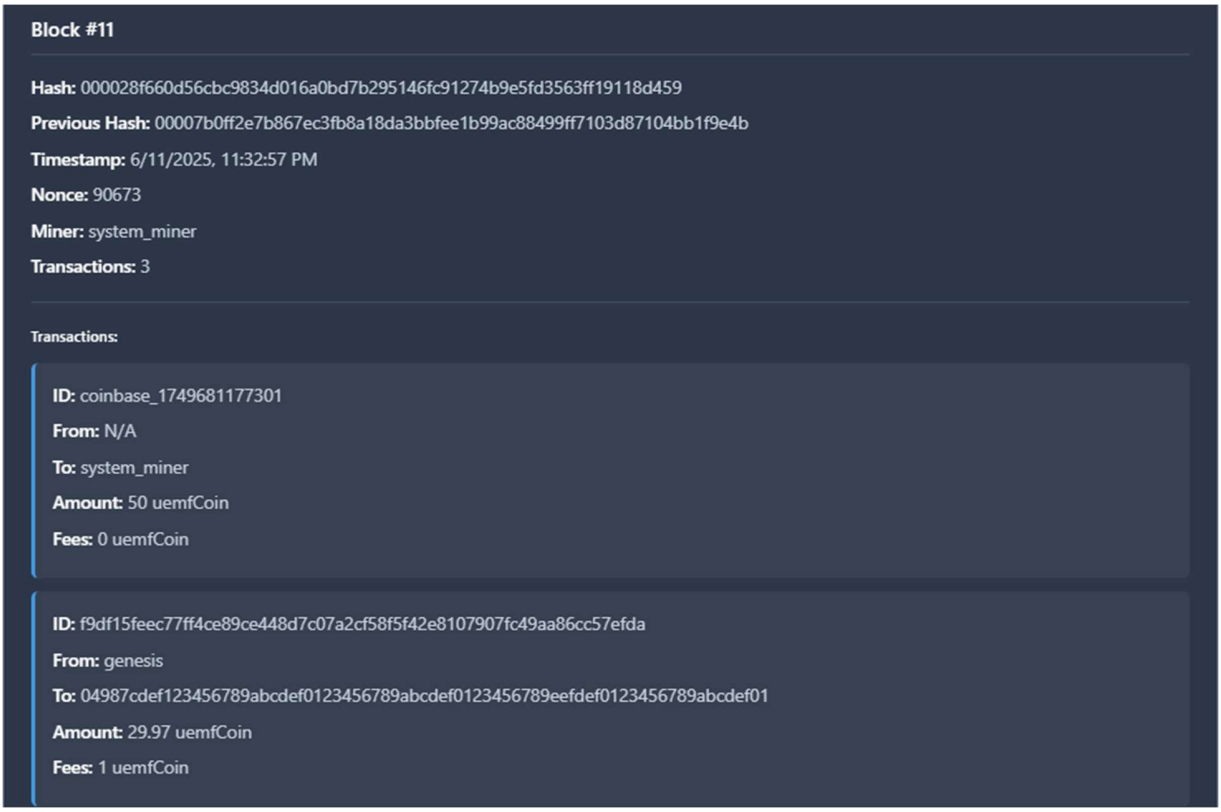
- Voir la liste des blocs validés.
- Voir les transactions en attente.
- Créer et envoyer des transactions.
- Voir le solde des portefeuilles.

Fonctionnement

Création des blocs toutes les 10 minutes

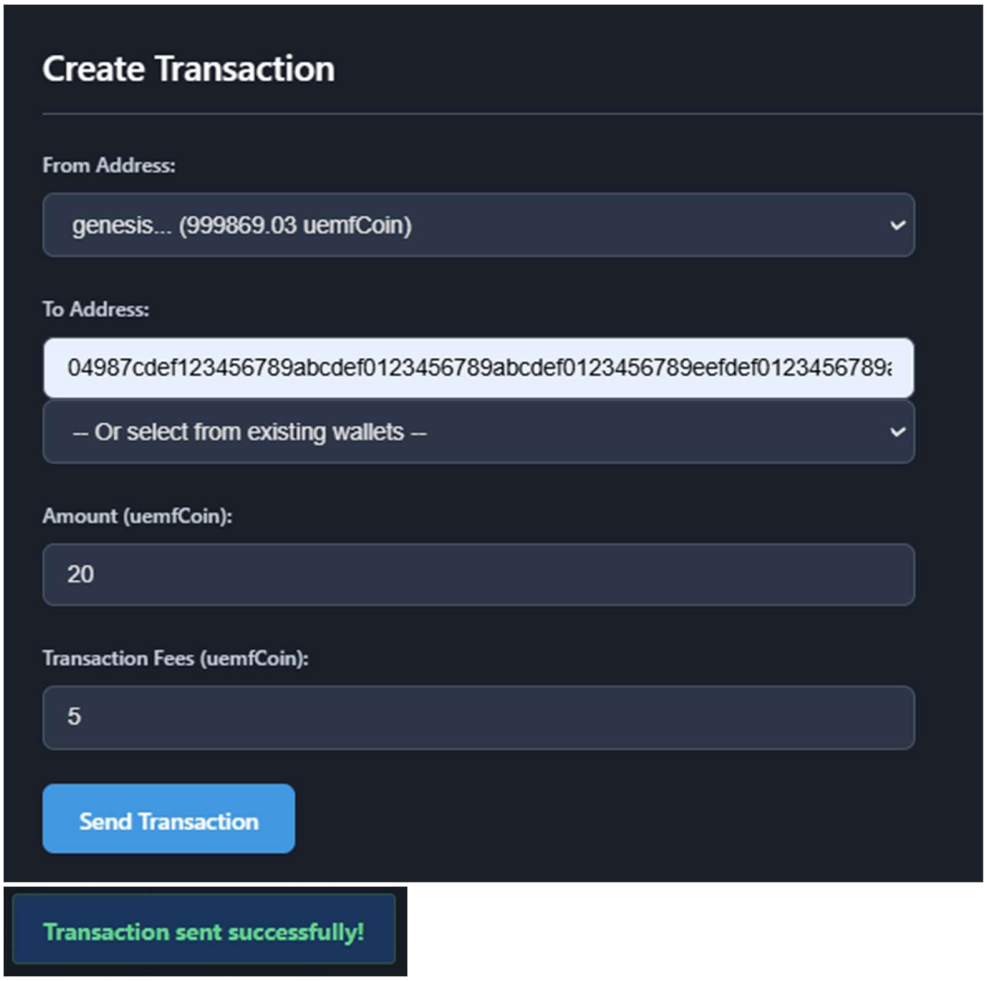
Le serveur Node.js crée un nouveau bloc à intervalle régulier, ici toutes les 10 minutes. Ce processus automatise la validation et l'ajout des transactions validées dans un bloc immuable.



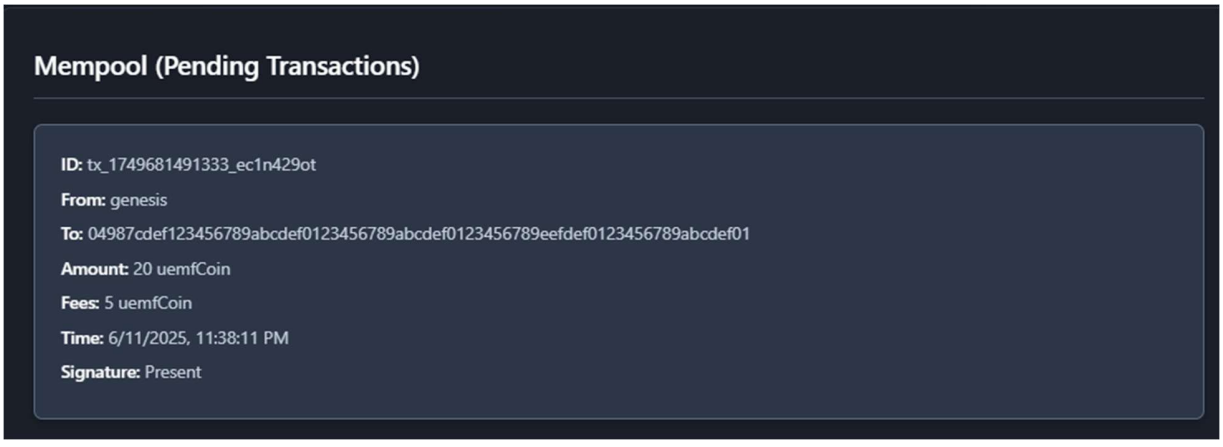


Gestion des transactions : création, validation, ajout au mempool

Les utilisateurs peuvent créer des transactions via un formulaire dans l'interface web. Chaque transaction est signée manuellement via un programme Node.js pour garantir son authenticité. Le serveur vérifie la validité (signature, solde suffisant, frais) avant d'ajouter la transaction au mempool en attente d'inclusion dans un bloc.



La transaction ensuite est ajoutée au mempool en attendant qu'elle soit confirmée:



Minage : résoudre un puzzle pour valider un bloc et recevoir une récompense

Le processus de minage consiste à résoudre un puzzle cryptographique lié à la difficulté définie. Le premier mineur à résoudre ce puzzle soumet le bloc au serveur, qui l'ajoute à la blockchain et crédite la récompense au mineur.

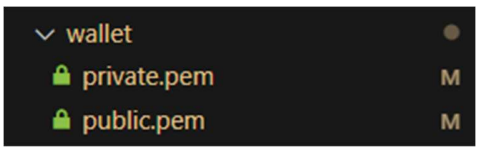
Wallets et signatures

Chaque client génère une paire de clés cryptographiques : la clé publique sert d'adresse pour recevoir les fonds, et la clé privée est utilisée pour signer les transactions. La sécurité repose sur la confidentialité de la clé privée.

```
const crypto = require("crypto");
const fs = require("fs");
const path = require("path");

function generateKeys() {
  const { publicKey, privateKey } = crypto.generateKeyPairSync("rsa", {
    modulusLength: 2048,
    publicKeyEncoding: { type: "spki", format: "pem" },
    privateKeyEncoding: { type: "pkcs8", format: "pem" }
  });

  const walletDir = path.join(__dirname, "client", "wallet");
  if (!fs.existsSync(walletDir)) {
    fs.mkdirSync(walletDir, { recursive: true });
  }

  fs.writeFileSync(path.join(walletDir, "private.pem"), privateKey);
  fs.writeFileSync(path.join(walletDir, "public.pem"), publicKey);

  console.log("Keys generated and saved to client/wallet/");
}

generateKeys();
```

```
PS C:\Users\Fbouy\Music\Projet Blockchain> node .\generateKeys.js
Keys generated and saved to client/wallet/
```

Ces deux fichiers sont générés à partir de script en dessus.

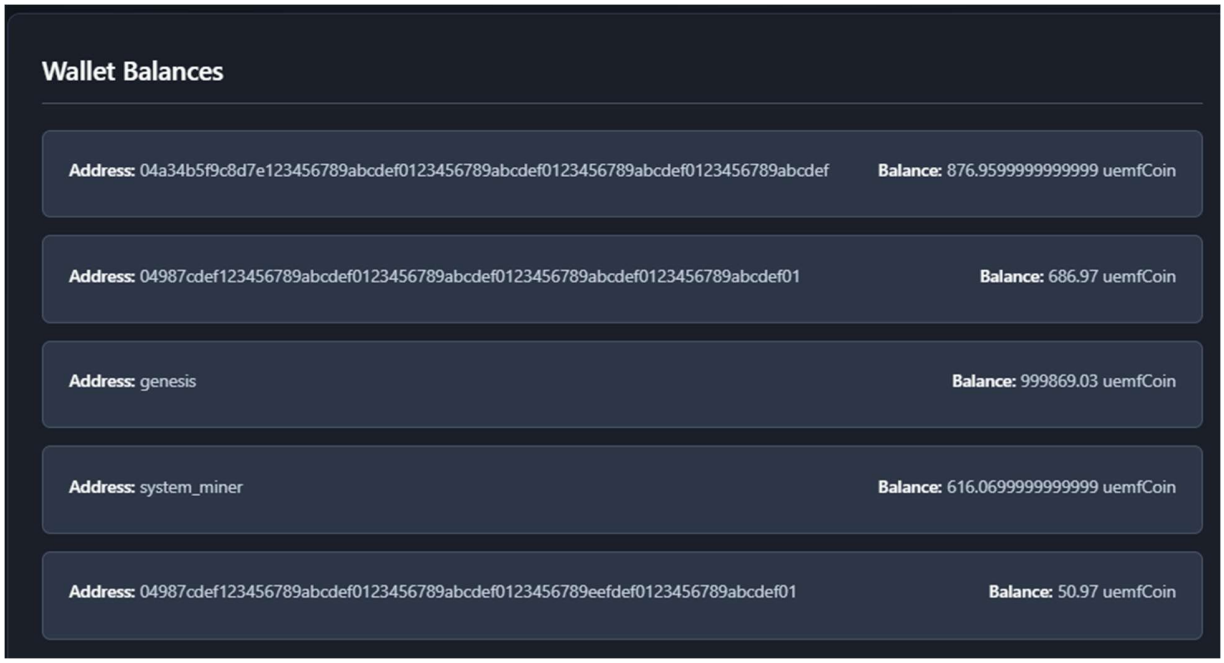


Signature manuelle des transactions avec un programme Node.js

Pour assurer la sécurité, la signature des transactions est réalisée manuellement via un script Node.js, qui chiffre les données de la transaction avec la clé privée pour produire une signature vérifiable par le serveur.

Affichage des soldes dans l'interface

L'interface web affiche les soldes des portefeuilles en temps réel, calculés à partir des transactions confirmées dans la blockchain.



Interface web et serveur

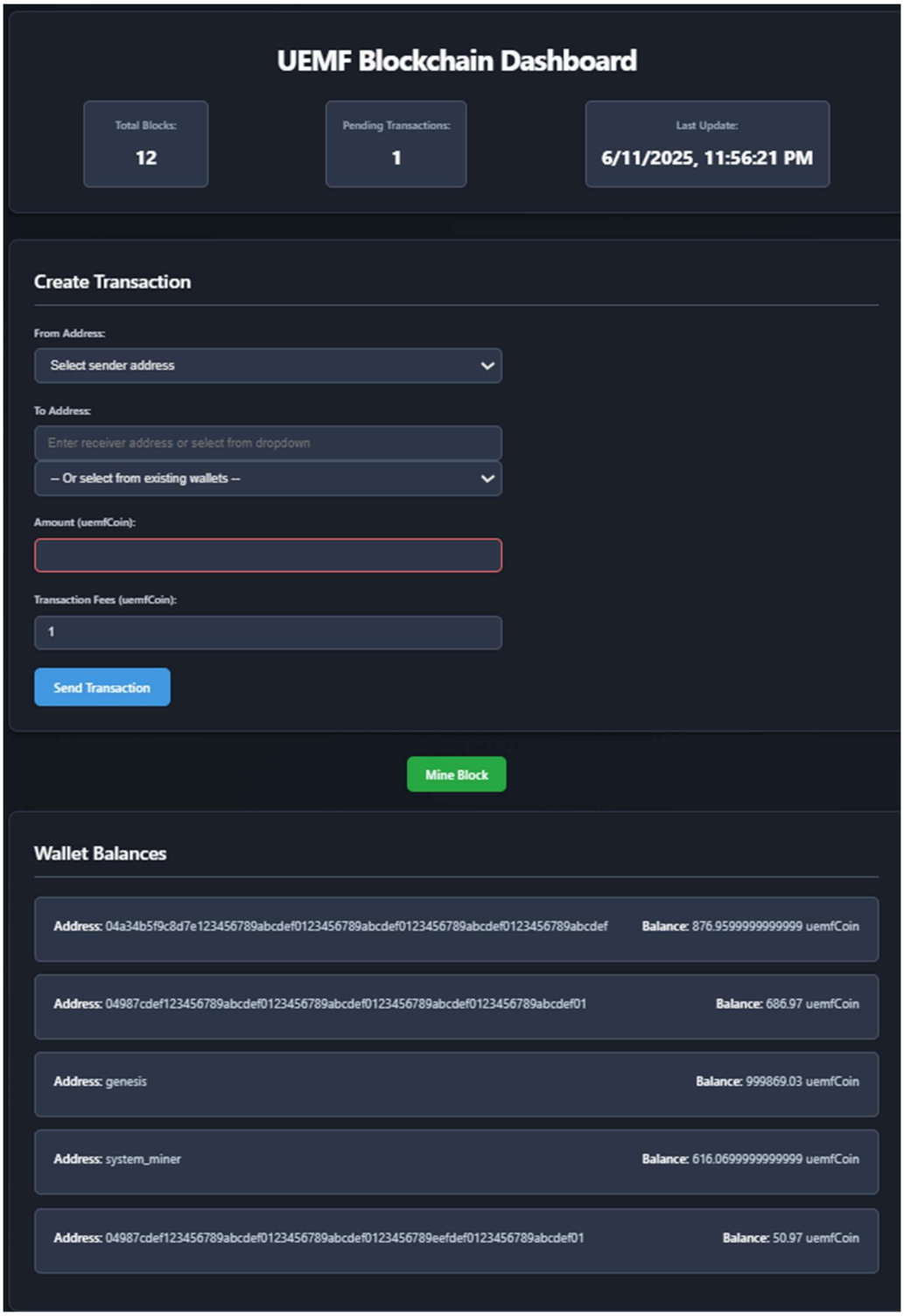
Serveur Express pour gérer les requêtes et servir l'interface

Le serveur Express gère toutes les requêtes API pour la création de transactions, le minage, et l'affichage des données. Il sert également les pages HTML/CSS du site web qui permettent aux utilisateurs d'interagir avec la blockchain locale.

Fonctionnalités de l'interface : voir la blockchain, créer des transactions, voir les soldes

L'interface utilisateur permet :

- de visualiser la chaîne de blocs complète,
- de créer et soumettre des transactions,
- de consulter le mempool,
- de voir le solde de chaque portefeuille.


UEMF Blockchain Dashboard
Total Blocks:
12
Pending Transactions:
1
Last Update:
6/11/2025, 11:56:21 PM
Create Transaction
From Address:
Select sender address
To Address:
Enter receiver address or select from dropdown
-- Or select from existing wallets --
Amount (uemfCoin):
Transaction Fees (uemfCoin):
1
Send Transaction
Mine Block
Wallet Balances
Address: 04a34b5f9c8d7e123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
Balance: 876.9599999999999 uemfCoin
Address: 04987cdef123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef01
Balance: 686.97 uemfCoin
Address: genesis
Balance: 999869.03 uemfCoin
Address: system_miner
Balance: 616.0699999999999 uemfCoin
Address: 04987cdef123456789abcdef0123456789abcdef0123456789eefdef0123456789abcdef01
Balance: 50.97 uemfCoin

Mempool (Pending Transactions)

ID: tx_1749681491333_ec1n429ot
From: genesis
To: 04987cdef123456789abcdef0123456789abcdef0123456789eefdef0123456789abcdef01
Amount: 20 uemfCoin
Fees: 5 uemfCoin
Time: 6/11/2025, 11:38:11 PM
Signature: Present

Blockchain

Block #0

Hash: 000000genesis
Previous Hash: N/A
Timestamp: 6/11/2025, 1:33:08 PM
Nonce: 0
Miner: genesis-miner
Transactions: 0

Block #1

Hash: 0000b895f04e5ec7c0d5537e1d86295d17be4b1202542e510cd50a61688b9d06
Previous Hash: 000000genesis
Timestamp: 6/11/2025, 3:25:53 PM
Nonce: 105776
Miner: system_miner
Transactions: 3

Transactions:

ID: coinbase_1749651953482
From: N/A
To: system_miner
Amount: 50 uemfCoin
Fees: 0 uemfCoin

ID: 2bc031b00bf3b5e3c9bb8983dea90b280ac80bdbf61d15d6e3e10cfa48d39fc7
From: 04a34b5f9c8d7e123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
To: 04987cdef123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef01
Amount: 10 uemfCoin
Fees: 1.07 uemfCoin

Conclusion

Ce projet m'a permis d'apprendre à construire une blockchain locale en Node.js avec preuve de travail. J'ai compris comment gérer les transactions, les vérifier, les ajouter au mempool, et comment le minage permet de créer des blocs valides. J'ai aussi appris à générer des portefeuilles (clés publique/privée) et à signer manuellement les transactions.

Grâce à ce projet, j'ai mieux compris les principes de fonctionnement d'une blockchain : la sécurité par signature numérique, le rôle du minage dans la validation des blocs, la gestion des soldes, et l'importance d'une interface claire pour interagir avec le système. Ce travail m'a donné une vision concrète de concepts souvent théoriques.

Le code source: https://github.com/FatimaBouyarmane/Projet-Blockchain/