📄Product Requirements Document (PRD)

Projet Semestriel – Crypto Wallet Web3

1. Résumé du Projet

L'objectif du projet est de développer une application **Crypto Wallet Web3** permettant aux utilisateurs de créer et gérer des portefeuilles cryptographiques, générer des clés privées/publiques, envoyer des transactions, effectuer des swaps entre jetons, et intégrer une implémentation blockchain basique pour la gestion des opérations. Ce wallet sera compatible avec plusieurs réseaux (ex : Ethereum, BSC, Polygon) et offrira une interface simple et sécurisée.

---

2. Objectifs Principaux

- Permettre à chaque utilisateur de générer et gérer un **wallet unique**.
- Génération automatique de **clé privée** et **clés publiques** pour chaque réseau supporté.
- Gestion des **transactions** (envoi/réception de jetons).
- Fonction **Swap** entre jetons (via smart contract ou simulateur interne).
- Implémentation d'un **mini-blockchain interne** pour tracer et valider les opérations.
- Interface simple, intuitive, et sécurisée.

---

3. Personas Utilisateurs

Utilisateur Débutant

- Veut créer un wallet facilement.
- Veut envoyer/recevoir des jetons sans complexité.

💻Utilisateur Avancé

- Comprend les concepts Web3.
- Veut visualiser les clés, explorer les transactions et utiliser le swap.

---

4. Fonctionnalités Principales

4.1 Création & Gestion de Wallet

- Génération d'une **clé privée** unique.
- Dérivation de **clés publiques** par réseau.
- Export/Backup du wallet en format sécurisé.
- Support du format de mnémonique (ex : BIP39) — optionnel.

4.2 Gestion des Réseaux (Multi-chain)

• Support initial :
• Ethereum (ETH, ERC-20)
• Binance Smart Chain (BSC, BEP-20)
• Polygon (MATIC)
• Chaque réseau possède :
• Son **adresse publique**
• Son **explorateur interne** (mini)

4.3 Transactions

• Envoi de jetons natifs (ETH, BNB, MATIC).
• Envoi de jetons tokenisés (ERC-20, BEP-20).
• Historique des transactions.
• Validation des transactions via la mini-blockchain interne.

4.4 Swap de Jetons

Deux options possibles : 1. **Simulateur interne de swap** (pas basé sur un DEX réel). 2. Swap via **smart contract interne** (AMM très simple : réserve A/B).

Fonctionnalités : - Choisir jeton source → jeton cible. - Calcul du taux via une formule simple (ex : x*y=k). - Création d'une transaction de swap enregistrée dans la blockchain.

4.5 Implémentation Mini-Blockchain

• Structure basique :
• Index
• Timestamp
• Données de la transaction
• Hash du bloc précédent
• Hash actuel
• Système de validation simple (Proof-of-Work faible ou Hash chaining).
• Explorer visuel des blocs.

---

5. Spécifications Techniques

5.1 Backend

• Langage : **Node.js / TypeScript**
• Librairies Web3 recommandées :
• `ethers.js` (principal)
• `web3.js` (optionnel)
• Smart contracts : Solidity (si swap via AMM)
• Base de données : aucune nécessaire (wallet local) ou SQLite légère pour logs.

5.2 Frontend

• Framework : **React.js** ou **React Native** (selon choix)

- UI : simple + dashboard des jetons

5.3 Blockchain interne

- Implémentation maison en TypeScript
- Stockage local (fichiers JSON)
- Validation hash chaining

---

6. Architecture Système

1. **Wallet Manager**
2. Génération clé privée
3. Dérivation addresses multi-chain
4. **Network Connector**
5. Connexion JSON-RPC aux réseaux (infura, Alchemy, ou RPC publics)
6. **Transaction Engine**
7. Construction, signature, envoi
8. **Swap Engine**
9. Simulateur ou smart contract
10. **Mini Blockchain**
11. Ajout et validation de blocs
12. **UI/UX Layer**
13. Dashboard
14. Historique
15. Exploreur blockchain

---

7. User Flow

7.1 Création du Wallet

1. L'utilisateur clique sur « Créer Wallet »
2. Génération clé privée
3. Génération adresses par réseau
4. Affichage du dashboard

7.2 Envoyer une Transaction

1. Choisir le réseau (ETH/BSC/Polygon)
2. Entrer l'adresse du destinataire
3. Entrer le montant
4. Signer → Ajouter au mini-blockchain → Envoyer au réseau

7.3 Swap

1. Sélection jeton source
2. Sélection jeton cible
3. Calcul du taux
4. Signature et validation

---

8. Sécurité

- Stockage sécurisé des clés privées (chiffrement AES local).
- Aucune transmission de la clé privée vers des serveurs.
- Signature locale obligatoirement.

---

9. Critères d'Acceptation (Acceptance Criteria)

- ✔️Un wallet peut être créé et exporté.
- ✔️Les adresses multi-chaine sont générées.
- ✔️On peut envoyer et recevoir des tokens.
- ✔️Le swap fonctionne correctement.
- ✔️La mini-blockchain valide et stocke les transactions.
- ✔️L'interface est claire et fonctionnelle.

---

10. Roadmap du Développement

Phase 1 : Wallet & Crypto Core

- Génération clé privée/publique
- Gestion multi-chain

Phase 2 : Transactions

- Envoi ETH / BNB / MATIC
- Support tokens ERC-20/BEP-20

Phase 3 : Swap

- Implémentation simulateur ou AMM

Phase 4 : Mini-Blockchain

- Structure + validation + UI explorer

Phase 5 : Finalisation UI

- Dashboard + historique + logs

---

11. Livrables

- Code source complet
- Documentation technique
- Manuel utilisateur
- Slides de présentation
- Vidéo démonstrative

---

12. Annexes (optionnel)

- Schémas d’architecture
- API documentation
- Structure de données des blocs

Fin du document PRD – Crypto Wallet Web3