**Référence du PCD :** PCD/2425/73

**Université de la Manouba**

**Ecole Nationale des Sciences de l'Informatique**



Cahier des charges du Projet de Conception et de Développement

**ID :232 version : 02**

**Sujet**

**Une plateforme d’e-commerce utilisant la blockchain**

**pour Les transactions Sécurisées**

**Réalisé par**

Yahya Hamed

Oussema Aissaoui

**Encadré par**

Prof. Wissem Ben Saied

Année Universitaire 2024/2025

**1.Introduction**

Ce projet vise à développer une plateforme e-commerce décentralisée qui permet aux acheteurs et aux vendeurs d'échanger des biens et services en toute sécurité grâce à la technologie blockchain. La blockchain garantit la transparence, l'immuabilité et la décentralisation des transactions. Chaque transaction est enregistrée sous forme de blocs sécurisés, empêchant toute modification sans l'accord du réseau.

****1.1. Objectifs principaux :****

* Fournir une plateforme sécurisée et transparente pour les transactions e-commerce.
* Utiliser des smart contracts pour automatiser les paiements et les enregistrements de commandes.
* Assurer la traçabilité des commandes via un hash unique dans la blockchain.
* Offrir une interface utilisateur intuitive et une expérience utilisateur fluide.

**2. Travail demandé**

**2.1. Acteurs**

****Acheteurs****:

* Utilisateurs qui parcourent le catalogue de produits, effectuent des achats et suivent leurs commandes.

****Vendeurs****:

* Utilisateurs qui listent leurs produits, gèrent leurs stocks et suivent les commandes passées.

****Administrateurs****:

* Gestionnaires de la plateforme qui supervisent les transactions, gèrent les utilisateurs et assurent le bon fonctionnement du système.

**2.2. Besoins fonctionnels**

**Acheteur :**

* Parcourir le catalogue de produits.
* Effectuer des achats via une interface de paiement blockchain.
* Suivre l'état des commandes en temps réel.
* Consulter l'historique des transactions.

****Vendeurs :****

* Lister des produits avec des descriptions détaillées.
* Gérer les stocks et les prix des produits.
* Suivre les commandes et les paiements reçus.
* Consulter l'historique des ventes.

****Administrateurs :****

* Gérer les utilisateurs (acheteurs et vendeurs).
* Superviser les transactions et les enregistrements de commandes.
* Assurer la maintenance et la sécurité de la plateforme.
* Citez les considérations et les contraintes dont vous devez tenir en compte lors de votre réalisation

**2.3. Besoins non fonctionnels**

* **Sécurité** : Les transactions doivent être sécurisées grâce à la blockchain, avec une protection contre les fraudes et les modifications non autorisées.
* **Performance** : La plateforme doit être capable de gérer un grand nombre de transactions simultanées sans perte de performance.
* **Scalabilité** : La plateforme doit être capable de s'adapter à une augmentation du nombre d'utilisateurs et de transactions.
* **Disponibilité** : La plateforme doit être disponible 24/7 avec un temps d'arrêt minimal.
* **Expérience utilisateur** : L'interface doit être intuitive et facile à utiliser pour tous les types d'utilisateurs.

**2.4. Besoins du domaine**

* **Transparence** : Toutes les transactions doivent être enregistrées de manière transparente et immuable dans la blockchain.
* **Décentralisation** : La plateforme doit fonctionner de manière décentralisée, sans dépendre d'une autorité centrale.
* **Traçabilité** : Chaque commande doit être traçable via un hash unique dans la blockchain.

**3. Environnements du travail**

**3.1. Environnement Matériel**

* Serveurs dédiés pour l'hébergement de la plateforme.
* Matériel de test pour les développeurs (ordinateurs performants, accès à des réseaux blockchain de test).
* Infrastructure cloud pour la scalabilité et la disponibilité.

**3.2. Environnement Logiciel**

****Frontend (Interface Utilisateur):****

* **Langages de programmation** :
  + **HTML** : Pour la structure des pages web.
  + **CSS** : Pour la mise en forme et le style des éléments de l'interface.
  + **JavaScript** : Pour l'interactivité et la dynamique des pages.
* **Framework/Librairies** :
  + **Bootstrap 4** : Pour le design responsive et les composants UI prêts à l'emploi (grilles, boutons, formulaires, etc.).
* **Outils de développement** :
  + **Visual Studio Code** : Environnement de développement intégré (EDI) pour le codage et le débogage.
  + **Git** : Pour le contrôle de version et la collaboration entre développeurs.
  + Remix IDE : Pour la compilation des codes Solidity
* **Outils de test** :
  + **Jest** : Pour les tests unitaires du code JavaScript.
  + **Cypress** : Pour les tests d'intégration et les tests end-to-end de l'interface utilisateur.

##### ****Backend (API & Smart Contracts)****

* **Langages de programmation** :
  + **JavaScript/Node.js** : Pour le développement du backend et des API.
  + **Solidity** : Pour le développement des smart contracts sur Ethereum ou Binance Smart Chain.
* **Frameworks** :
  + **Web3.js** : Pour l'interaction avec la blockchain (lecture/écriture des données, gestion des transactions).
* **Outils de développement** :
  + **Remix** : Pour le développement, le test et le déploiement des smart contracts.
  + **Ganache** : Pour simuler un réseau blockchain local pendant le développement.
  + **Postman** : Pour tester les API backend.

##### ****Blockchain:****

**Réseaux blockchain supportés** :

* + **Ethereum** : Pour les smart contracts et les transactions en Ether ou tokens ERC-20.

**Tokens de paiement** :

* + **Stablecoins** :  **USDT (Tether).**

**Outils de monitoring** :

* + **Etherscan** : Pour suivre les transactions sur Ethereum.
  + Metamask : Pour tester les transaction et créer des portfeuils virtuels

##### **Autres outils :**

* **Google Analytics** : Pour le suivi des utilisateurs et l'analyse du trafic sur la plateforme.
* **Outils de reporting** : Pour générer des rapports sur les transactions, les ventes et les performances de la plateforme.

1. **Chronogramme prévisionnel du projet**

|  |  |  |  |
| --- | --- | --- | --- |
| Période | Semaine | Tâches principales |  |
| **Février** | 1-2 | • Analyse des besoins fonctionnels et non fonctionnels |  |
|  |  | • Définition de l'architecture de la plateforme |  |
|  |  | • Choix des technologies et outils |  |
|  | 3-4 | • Conception des wireframes et maquettes UI/UX |  |
|  |  | • Rédaction du cahier des charges |  |
| **Mars** | 1-2 | • Développement du frontend (mise en place du projet, pages principales) |  |
|  |  | • Intégration du design et tests UI |  |
|  | 3-4 | • Développement du backend (API et gestion des utilisateurs) |  |
|  |  | • Déploiement d'une base de données initiale |  |
| **Avril** | 1-2 | • Implémentation des smart contracts et tests sur Ganache |  |
|  |  | • Intégration Web3.js et interactions avec la blockchain |  |
|  | 3-4 | • Tests unitaires et correction des bugs |  |
|  |  | • Optimisation des performances et sécurisation |  |
| **Mai** | 1-2 | • Tests finaux et validation |  |
|  |  | • Documentation technique et utilisateur |  |
|  |  | • Présentation et déploiement final de la plateforme |  |

**Remarque** : La rédaction du rapport se fait tout au long de cette période afin de documenter l'évolution du projet et ses différentes phases.