

Énoncé 4A IBC : Projet Annuel

Projet : Création d'un échange décentralisé sur Ethereum

Descriptif :

Ce projet a pour but de tester les compétences des étudiants de 4ème année à créer une Decentralized App en utilisant leurs connaissances sur le protocole Ethereum, sur un framework web et en infrastructure IT.

Modalités :

Ce projet est à rendre par groupes de 3.

Le rendu doit être sous le format d'un repository Git **public**.

Le rendu doit être uploadé sur Github avant la date butoir du projet.

Chaque étape du projet sera individuellement notée en fonction de l'apport de chaque étudiant.

Enoncé:

Le but de ce projet annuel est de créer une dApp complète simulant une plateforme d'échange décentralisée d'actifs numériques sur Ethereum (Layer 1 ou Layer 2, au choix).

Cette plateforme doit proposer des fonctionnalités pour les populations suivantes :

- La capacité aux utilisateurs de visualiser la liste des actifs sur la plateforme et de choisir un actif spécifique à consulter (prix, volume, nombre de trades).
- La capacité aux utilisateurs d'échanger un actif pour un autre (token swap) depuis une page dédiée.
- La capacité aux utilisateurs d'ajouter des actifs dans une base de liquidité (Liquidity Pool) qui va régir le prix d'échange des actifs. Un ajout d'actifs ne peut se faire que par pair (ETH/MATIC par exemple).
- La capacité aux utilisateurs d'ajouter des actifs dans une épargne d'actifs (staking) afin de gagner périodiquement un certain pourcentage de leur mise en ce même actif.
- La capacité des administrateurs à gérer de manière globale la plateforme (paramétrer les frais de la plateforme, bannissement d'une adresse, donation d'actifs gratuitement).
- La capacité des administrateurs à livrer des statistiques de la plateforme sur un Dashboard public.

Une fois déployée, la dApp aura comme fonctions aux utilisateurs de :

- Permettre à un nouvel utilisateur de se connecter via son wallet,
- Permettre à un utilisateur connecté de parcourir les actifs disponibles,
- Permettre à un utilisateur connecté d'échanger un actif selon les paires disponibles,
- Permettre à un utilisateur connecté d'ajouter dans une Liquidity Pool ses actifs selon les paires disponibles,
- Permettre à un utilisateur connecté d'épargner ses actifs dans des Staking Pool selon les actifs disponibles.

Une fois déployée, la dApp aura comme fonctions d'administration de :

- Permettre à l'administrateur de se connecter (au Dashboard Admin),
- Permettre à l'administrateur d'ajouter de nouveaux administrateurs (depuis le Dashboard),
- Permettre à l'administrateur d'alimenter la balance du SC en ETH,
- Permettre à l'administrateur de retirer depuis la balance du SC en ETH (les fees de trading),
- Permettre à l'administrateur de retirer un actif du site,
- Permettre à l'administrateur de bannir un utilisateur,

- Permettre à l'administrateur de voir les statistiques des différents utilisateurs,
- Permettre à l'administrateur de voir les statistiques des différents d'actifs.

Le smart-contract Ethereum devra avoir comme fonctionnalité:

- Enregistrer les utilisateurs de l'application,
- Héberger chaque actif sur la plateforme.
- Transférer les actifs d'une part et d'autre.

Un oracle est à prévoir pour les tâches suivantes :

- Gestion des statistiques indexées,
- Gestion du cours ETH en Euro et/ou à Tokens.

Le smart-contract sera codé en Solidity et à déployer sur un Testnet Ethereum.

La dApp doit avoir un front web propulsé par un framework de votre choix entre React ou Vue (Next et Nuxt sont permis).

La dApp doit aussi gérer les comptes utilisateurs via une base de données PostgreSQL.

Toutes les données **non identifiantes** doivent être stockées sur le SC Ethereum.

Toutes les données **identifiantes** doivent être stockées sur la base de données PostgreSQL.

Le lien entre les deux se fait via un ID Unique à attribuer à l'utilisateur lors de chaque première connexion.

La dApp doit être Dockerisée afin de faciliter son déploiement, des volumes persistants pour la BDD sont à prévoir.

Annexes :

<https://ethereum.org/en/developers/docs/networks/>

<https://sepolia.dev/>

<https://docs.soliditylang.org/en/v0.8.23/>

<https://react.dev/learn>

<https://fr.vuejs.org/guide/introduction.html>

<https://fr.vuejs.org/ecosystem/themes.html>

<https://docs.docker.com/get-started/>

<https://docs.docker.com/compose/>

<https://www.postgresql.org/docs/current/tutorial.html>

<https://www.sushi.com/academy>

<https://www.sushi.com/swap>

<https://app.uniswap.org/>

<https://uniswap.org/whitepaper-v3.pdf>