PROJET – Module Blockchain

Titre du projet : SafeClub – Trésorerie sécurisée d'un club étudiant sur Ethereum
Type : Projet de groupe 2 ou 4 personnes maximum

1. Objectif du projet

Réaliser un **smart contract Ethereum** permettant à un club étudiant de :

- gérer sa trésorerie (coffre-fort en ETH),
- créer des **propositions de dépenses**,
- faire **voter les membres**,
- exécuter les paiements uniquement si les règles définies sont respectées,
- en appliquant les **bonnes pratiques de sécurité** vues en cours.

---

2. Travail demandé

Les étudiants doivent :

1. Concevoir et implémenter un smart contract "SafeClub" en **Solidity** avec :
    - gestion des membres (ajout/suppression, liste),
    - réception d'ETH (vault),
    - création de propositions de dépenses (montant, destinataire, description, deadline),
    - votes des membres (pour/contre, un seul vote par membre),
    - exécution sécurisée d'une proposition acceptée (vérifications + transfert).
2. Intégrer des mécanismes de **sécurité** :
    - protection contre la reentrancy,
    - contrôle d'accès (owner, membres, éventuellement trésorier),

- validation des montants et des états (deadline, pas de double exécution, etc.).

3. Tester et analyser le contrat :
    - tests (scripts ou scénarios) pour les cas principaux,
    - utilisation d'un outil d'analyse statique et commentaire des warnings importants.

---

3. Livrables

À la fin du projet, chaque groupe rend :

1. **Code source du smart contract** (+ scripts / instructions de déploiement).
2. **Mini rapport de sécurité (5–8 pages)** :
    - description du contrat,
    - modèle de menaces (3–5 scénarios d'attaque),
    - vulnérabilités considérées et contre-mesures,
    - résultats d'analyse statique (warnings + réponses).
3. **Documentation courte (3–5 pages)** :
    - structures de données,
    - fonctions principales,
    - règle de décision (acceptation d'une proposition).

Optionnel mais valorisé :
4. **Petite interface Web** permettant de voir le solde, lister les propositions, voter, exécuter.

---

4. Outils recommandés

- **Solidity 0.8.x**

- Remix ou Hardhat / Truffle + Ganache / Hardhat Network
- MetaMask pour les tests
- OpenZeppelin (`Ownable`, `ReentrancyGuard`, etc.)
- Outil d'analyse : Slither ou équivalent

SÉANCE FINALE DE VALIDATION

La séance finale est une **séance obligatoire de validation du projet**.
Chaque groupe dispose de **10–15 minutes** (présentation + questions).

Déroulement

1. **Présentation rapide (3–5 minutes)**
    - Rappel du but du projet.
    - Architecture globale du contrat (rôles, propositions, votes).
2. **Démonstration (5–7 minutes)**
    - Déploiement du contrat (ou contrat déjà déployé).
    - Scénario complet :
        - dépôt de fonds dans le coffre-fort,
        - création d'une proposition de dépense,
        - votes de membres (pour/contre),
        - exécution de la proposition acceptée.
3. **Sécurité & questions (3–5 minutes)**
    - Explication de 2–3 menaces identifiées et des protections mises en place.
    - Comment vous avez géré :
        - la reentrancy,

        - le contrôle d’accès,
        - la validation des montants et des états.
    - Questions de l’enseignant sur le code, les choix techniques et les limites.

Critères de validation

- Fonctionnalités principales opérationnelles.
- Contrat déployable et utilisable.
- Présence réelle de mécanismes de sécurité (pas seulement sur le papier).
- Capacité à expliquer le fonctionnement et les risques.
- Qualité du mini rapport de sécurité et de la documentation.