

Project Implementation Plan

1. TỔNG QUAN

- **Dự án:** Web3 Crowdfunding
- **Người thực hiện:** Lê Đức Thiện – B22DCCN823
- **Mục tiêu:** Xây dựng một nền tảng gọi vốn phi tập trung cơ bản sử dụng blockchain để đảm bảo tính minh bạch và lòng tin của cộng đồng dành cho người kêu gọi vốn.
- **Ý nghĩa trong học tập:**
 - Hiểu cách blockchain thay thế các hệ thống tập trung truyền thống.
 - Làm quen với quy trình phát triển ứng dụng phi tập trung (DApp).

2. KIẾN THỨC CỐT LÕI

2.1 Phi tập trung (Decentralization)

- Dữ liệu và logic không được lưu trữ hoặc kiểm soát bởi một thực thể trung tâm mà được phân phối trên mạng lưới các nút blockchain.
- Không cần trung gian quản lý tiền quỹ (như Kickstarter), mọi giao dịch được ghi trực tiếp trên blockchain Ethereum.
- Người dùng tự quản lý ví của mình (ví dụ: MetaMask) và tương tác trực tiếp với hợp đồng thông minh.
- **Lợi ích:** Loại bỏ phí trung gian, tăng tính tự chủ cho người dùng.

2.2 Hợp đồng thông minh (Smart Contracts)

- Là các đoạn mã lập trình chạy trên blockchain, tự động thực thi các điều kiện được định trước mà không cần sự can thiệp của con người.
- **Lợi ích:** Giảm thiểu rủi ro gian lận, tăng tính minh bạch và hiệu quả.

2.3 Minh bạch và bất biến (Transparency & Immutability)

- Minh bạch: Mọi giao dịch đều được ghi công khai trên blockchain.
- Bất biến: Một khi dữ liệu được ghi lên blockchain, nó không thể bị thay đổi.
- **Lợi ích:** Xây dựng niềm tin cho người ủng hộ.

3. MÔ TẢ SẢN PHẨM

3.1 Tính năng chính:

- **Tạo chiến dịch:** Người tạo mô tả dự án, nhập mục tiêu (số ETH) và thời hạn kết thúc kêu gọi.
- **Đóng góp:** Người ủng hộ gửi ETH vào hợp đồng thông minh.
- **Kết thúc chiến dịch:**
 - Nếu đạt mục tiêu trước thời hạn: tiền chuyển cho người tạo.
 - Nếu hết thời hạn mà không đạt: tiền hoàn lại người ủng hộ.

3.2 Công nghệ sử dụng:

- Blockchain: Ethereum (testnet Sepolia để không tốn ETH thật).
- Smart Contract: Solidity, Foundry.
- Backend: Node.js, Express.js, Web3.js/Ethers.js.
- Frontend: React.js, Tailwind CSS.
- Ví điện tử: MetaMask.

4. KẾ HOẠCH THỰC HIỆN

Tuần 1: Học kiến thức blockchain cơ bản

- Nghiên cứu về lịch sử, nguyên lý hoạt động của blockchain Bitcoin, Ethereum, ...
- Tìm hiểu cơ chế đồng thuận (Proof of Work, Proof of Stake).
- Cài đặt MetaMask trên trình duyệt và thiết lập tài khoản.

- Kết nối MetaMask với testnet Sepolia.
- Tìm hiểu cách hoạt động của Sepolia faucet, nhận ETH thử nghiệm và thực hiện các giao dịch gửi/nhận ETH giữa các ví.
- Tìm hiểu về block explorer (Etherscan) để theo dõi giao dịch trên blockchain.

Tuần 2: Học Solidity cơ bản & viết smart contract đơn giản

- Nghiên cứu cú pháp Solidity: khai báo biến, kiểu dữ liệu, cấu trúc điều kiện, vòng lặp, modifier, event.
- Viết các hợp đồng thông minh đơn giản:
 - Smart contract lưu trữ biến số và thay đổi giá trị.
 - Smart contract cho phép gửi và nhận ETH.
- Tìm hiểu Remix IDE, viết và biên dịch hợp đồng trên Remix.
- Kiểm thử hợp đồng bằng cách thực hiện các giao dịch trên Remix và kiểm tra kết quả trên blockchain testnet.
- Tìm hiểu về gas fee và tối ưu hóa gas cho các smart contract.

Tuần 3: Học Foundry cơ bản

- Cài đặt Foundry và thiết lập môi trường phát triển.
- Nghiên cứu cấu trúc thư mục dự án Foundry, cách tạo và biên dịch hợp đồng thông minh.
- Viết các bài kiểm thử tự động bằng Foundry để kiểm tra tính đúng đắn của smart contract:
 - Kiểm tra tính hợp lệ của giao dịch gửi/nhận ETH.
 - Kiểm tra xử lý lỗi khi gửi quá số dư khả dụng.
- Tìm hiểu cách debug hợp đồng với Foundry để tối ưu logic.

Tuần 4: Xây dựng smart contract crowdfunding

- Thiết kế kiến trúc smart contract:
 - Cấu trúc dữ liệu lưu thông tin chiến dịch gọi vốn.
 - Các điều kiện cần thiết để một chiến dịch hợp lệ.

- Chức năng đóng góp ETH, kiểm tra số dư.
- Cơ chế hoàn tiền nếu chiến dịch không đạt mục tiêu.
- Cơ chế xác minh và rút tiền khi đạt mục tiêu.
- Viết hợp đồng trên Solidity, triển khai trên testnet Sepolia.
- Kiểm thử toàn diện với Foundry để phát hiện lỗi logic.
- Chạy thử nghiệm các tình huống thực tế trên testnet.

Tuần 5-6: Phát triển backend & frontend

- Backend:
 - Xây dựng API với Node.js và Express.js.
 - Kết nối API với smart contract sử dụng Web3.js hoặc Ethers.js.
 - Thiết lập MongoDB để lưu trữ dữ liệu người dùng và chiến dịch.
 - Xây dựng hệ thống xác thực người dùng bằng địa chỉ ví.
- Frontend:
 - Thiết kế UI bằng React.js và Tailwind CSS.
 - Tạo trang đăng nhập, danh sách chiến dịch, chi tiết chiến dịch.
 - Tích hợp MetaMask để kết nối với blockchain.
 - Hiển thị số dư, tiến độ chiến dịch và giao dịch thành công/thất bại.

Tuần 7-8: Kiểm thử và triển khai

- Kiểm thử hệ thống:
 - Thử nghiệm các chức năng từ đăng nhập, tạo chiến dịch, đóng góp, rút tiền.
 - Kiểm tra khả năng xử lý lỗi khi người dùng nhập sai thông tin hoặc không đủ ETH.
 - Kiểm tra bảo mật hợp đồng thông minh.
- Triển khai:
 - Triển khai frontend trên Vercel hoặc Netlify.

- Triển khai backend trên MongoDB Atlas.

5. KẾT QUẢ MONG ĐỢI

- Hoàn thành DApp crowdfunding phi tập trung trên testnet Sepolia.
- Đạt được hiểu biết thực tế về phát triển ứng dụng Web3.
- Viết báo cáo hoàn chỉnh về quy trình thực hiện và kết quả thử nghiệm.