

**KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
TÌM HIỂU VỀ BLOCKCHAIN CƠ BẢN VÀ VÍ TIỀN ĐIỆN TỬ
HỌC KỲ I, NĂM HỌC 2025-2026**

Giảng viên hướng dẫn:
Ths. Phan Thị Phương Nam

Sinh viên thực hiện:
Họ tên: Kiều Tấn Phước
MSSV: 110122144
Lớp: DA22TTB

Vĩnh Long, tháng 12 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

[illegible]

Vĩnh Long, ngày tháng năm

Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vĩnh Long, ngày tháng năm

Thành viên hội đồng

(Ký tên và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	11
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	12
2.1 Blockchain là gì?	12
2.1.1 Các đặc điểm cốt lõi của Blockchain.....	12
2.1.2 Khối khởi thủy (Genesis Block)	14
2.2 Nguyên lý hoạt động.....	14
2.2.1 Quy trình thực hiện giao dịch	14
2.2.2 Cấu trúc khối và Hàm băm (Hash)	15
2.3 Mạng ngang hàng P2P và cơ chế đồng thuận	16
2.3.1 Khái niệm mạng ngang hàng P2P (Peer to Peer).....	16
2.3.2 Cơ chế truyền tin và Lưu trữ phân tán	17
2.4 Cơ chế đồng thuận Proof of Work (PoW)	17
2.4.1 Nguyên lý hoạt động của cơ chế PoW.....	17
2.4.2 Chức năng chính của PoW.....	18
2.5 Ví tiền điện tử (Crypto Wallet).....	19
2.5.1 Nguyên lý hoạt động của ví tiền điện tử (Crypto Wallet)	19
2.5.2 Chức năng chính ví tiền điện tử (Crypto Wallet)	19
2.5.3 Cơ chế khôi phục ví và tính bảo mật	19
2.6 Kết luận.....	20
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU.....	21
3.1 Phương pháp nghiên cứu	21
3.1.1 Tải phần mềm	21
3.1.2 Thư viện lập trình	22
3.2 Phương pháp thực hiện	22
3.3 Mô tả code và giải thích.....	27

3.3.1	Phần chuyên mục báo cáo:.....	27
3.3.2	Phần miêu tả code:	34
3.3.3	Logic xác thực số dư và chống chi tiêu gấp đôi (Double Spending).....	35
3.3.4	Chi tiết thuật toán ký số ECDSA	35
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....		48
4.1.1	Phần kết quả.....	48
4.1.2	Phần hiệu năng.....	48
4.1.3	Phần trải nghiệm	49
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		50
5.1	Kết luận.....	50
5.2	Hướng phát Triển.....	50
DANH MỤC TÀI LIỆU THAM KHẢO.....		52

LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời cảm ơn chân thành đến cô Phan Thị Phương Nam đã luôn tận tình hướng dẫn và hỗ trợ cho em trong suốt quá trình thực hiện đồ án môn học.

Cô không chỉ cung cấp những kiến thức quý báu mà còn giúp em định hướng và giải quyết những khó khăn mà em gặp phải trong việc tìm hiểu về blockchain cơ bản và ví tiền điện tử

Sự nhiệt tình và kiên nhẫn của cô đã giúp nhóm em hoàn thành đồ án chuyên ngành một cách suôn sẻ và hiệu quả hơn. Em thật sự rất biết ơn và mong rằng sẽ tiếp tục nhận được sự giúp đỡ từ cô trong những lần sau.

Sinh Viên

Kiều Tấn Phước

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1: Ứng dụng của Blockchain trong các lĩnh vực khác nhau.....	13
Hình 2: Sơ đồ hoạt động giao dịch Blockchain	15
Hình 3: Sơ đồ xử lý dữ liệu của hàm băm SHA-256	16
Hình 4: Sự khác nhau giữa client-server (mạng truyền thống) và P2P.....	17
Hình 5: Sơ đồ hoạt động cơ chế đồng thuận PoW	18
Hình 6: Phần mềm lập trình VSCODE	21
Hình 7: Phần mềm quản lý thư viện Anaconda (Python)	21
Hình 8: Phần mềm mô phỏng JUPYTER	21
Hình 9: Phần mở rộng JUPYTER.....	23
Hình 10: Chương trình chạy thành công.....	23
Hình 11: Chương trình chạy không thành công.....	24
Hình 12: Bảng lệnh command VSCODE	24
Hình 13: Nhập lệnh command “ Python Select Interpreter”.....	25
Hình 14: Chọn thư viện Python bằng bảng command	25
Hình 15: Chọn Python Venv (Python gốc)	25
Hình 16: Chọn thư viện Base (Python của Anaconda)	26
Hình 17: Phần mở rộng quản lý thư viện Python	26
Hình 18: Thư viện được tích hợp sẵn khi cài đặt.....	26
Hình 19: Giao diện Blockchain được demo bằng Flask	37
Hình 20: Kết quả tạo ví mới với địa chỉ và private key	38
Hình 21: Giao dịch ví tiền điện tử.....	39
Hình 22: Chế độ đào Block được khởi động	40
Hình 23: Thông số kỹ thuật của Block sau khi khai thác thành công.....	41
Hình 24: Thông số kỹ thuật của block khai thác thất bại.....	42
Hình 25: Chế độ tự động đào block được dừng.....	43

Hình 26: Tra cứu số dư của ví.....	44
Hình 27: Lưu danh sách ví điện tử của người dùng.....	44
Hình 28: Giao diện quản lý danh sách các khối trong Blockchain hiện tại	45
Hình 29: Lịch sử giao dịch người dùng	46
Hình 30: Giao diện mạng P2P và nhật ký hệ thống	46
Hình 31: Hiện thị các thông báo về trạng thái ký giao dịch	47

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đề tài tìm hiểu về Blockchain cơ bản và ví tiền điện tử được thực hiện nhằm giúp sinh viên nắm vững các khái niệm và tính ứng dụng cao trong lĩnh vực công nghệ thông tin. Đề tài cũng cố tập trung tìm hiểu nguyên lý hoạt động Blockchain và cơ chế liên kết các khối dữ liệu nhằm mục đích đảm bảo tính toàn vẹn và bảo mật thông tin

Trong quá trình thực hiện thì đề tài trình bày về cấu trúc của một khối (block) trong khối block bao gồm thành phần như dữ liệu, hash, mô hình mạng ngang hàng P2P (Peer-to-Peer) được phân tích làm rõ các node trong hệ thống cùng xác thực và lưu trữ dữ liệu mà không cần máy chủ trung tâm

Về mặt thực nghiệm đồ án tiến hành xây dựng chương trình mô phỏng bằng Blockchain đơn giản bằng ngôn ngữ Python. Chương trình cho phép tạo ra khối mới liên kết lại thành chuỗi, thực hiện băm dữ liệu và kiểm tra tính toàn vẹn của Blockchain. Qua đó giúp cho việc hiểu rõ hơn cách Blockchain hoạt động trong thực tế thay vì dừng lại ở lý thuyết

Thông qua đồ án này việc không chỉ củng cố lại kiến thức về blockchain và ví tiền điện tử mà còn rèn luyện kỹ năng lập trình, tư duy thuật toán và phân tích hệ thống và kết quả đạt được là nền tảng quan trọng để tiếp cận các ứng dụng nâng cao hơn của Blockchain trong các lĩnh vực tài chính, hợp đồng thông minh và hệ thống phân tán

MỞ ĐẦU

- **Lý do chọn đề tài:**

Hiện nay việc cách mạng công nghiệp 4.0 công nghệ blockchain đang được càng ngày được ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, tiền điện tử và quản lý dữ liệu, chuỗi cung ứng và các hệ thống phân tán Blockchain nổi bật với các tính năng như tính minh bạch, bảo mật cao, phi tập trung và khó bị thay đổi dữ liệu và góp phần giải quyết nhiều hạn chế của các hệ thống truyền thống khác. Tuy nhiên việc tiếp cận và hiểu rõ nguyên lý hoạt động của Blockchain đối với sinh viên vẫn còn gặp khó khăn do tính trừu tượng của công nghệ vì vậy em đã quyết định chọn đề tài “tìm hiểu về Blockchain cơ bản và ví tiền điện tử” nhằm mục đích nghiên cứu mô phỏng các cơ chế hoạt động nền tảng của Blockchain thông qua lập trình thực tế

- **Mục đích nghiên cứu:**

Mục đích nghiên cứu của đồ án là nghiên cứu làm rõ các khái niệm cơ bản của công nghệ Blockchain như cấu trúc khối, hàm băm, giao dịch, nonce, và cơ chế liên kết chuỗi khối. Đồng thời đồ án hướng đến xây dựng chương trình mô phỏng Blockchain và ví tiền điện tử đơn giản bằng ngôn ngữ Python giúp cho sinh viên dễ dàng tiếp cận hiểu rõ bản chất hoạt động của Blockchain và vận dụng kiến thức lý thuyết vào thực tiễn

- **Đối tượng và phạm vi nghiên cứu:**

- **Đối tượng nghiên cứu**

- **Công nghệ Blockchain cơ bản:** bao gồm cấu trúc block, chuỗi khối, hash, nonce, giao dịch và nguyên lý đảm bảo tính toàn vẹn dữ liệu.

- **Mạng ngang hàng P2P (Peer-to-Peer):** nghiên cứu cách các node tham gia mạng, lưu trữ và xác thực dữ liệu trong hệ thống phân tán.

- **Ví tiền điện tử:** mô phỏng cách tạo ví, quản lý địa chỉ và thực hiện giao dịch đơn giản trong Blockchain.

Chương trình mô phỏng bằng Python: xây dựng mô hình Blockchain đơn giản để minh họa trực quan cho quá trình tạo khối và xác thực dữ liệu.

- **Phạm vi nghiên cứu**

- **Xây dựng mô hình Blockchain đơn giản:** tập trung vào mô phỏng nguyên lý hoạt động, không đi sâu vào các thuật toán phức tạp như Proof of Work nâng cao hoặc các hệ thống Blockchain thương mại.

- **Ngôn ngữ lập trình Python:** sử dụng Python để triển khai chương trình mô phỏng do tính đơn giản, dễ hiểu và phù hợp cho mục đích học tập.
- **Phục vụ mục đích học tập và nghiên cứu:** đề án hướng đến việc hỗ trợ sinh viên hiểu rõ nền tảng Blockchain, làm cơ sở để nghiên cứu các ứng dụng nâng cao như tiền điện tử, hợp đồng thông minh và hệ thống phân tán.
- **Ứng dụng kiến thức công nghệ thông tin:** kết hợp kiến thức về lập trình, thuật toán và bảo mật dữ liệu nhằm tạo ra mô hình Blockchain có tính minh họa và mở rộng trong tương lai.

CHƯƠNG 1 :TỔNG QUAN

+ Trong thời đại công nghệ số phát triển mạnh mẽ việc ứng dụng công nghệ Blockchain vào nhiều lĩnh vực như tài chính tiền điện tử quản lý dữ liệu và hệ thống phân tán ngày càng trở nên phổ biến Blockchain giúp tăng cường tính minh bạch bảo mật và an toàn dữ liệu đồng thời thay đổi cách thức lưu trữ và trao đổi thông tin trong môi trường số Tiền điện tử là một trong những ứng dụng tiêu biểu của Blockchain và đang thu hút sự quan tâm lớn từ cộng đồng

+ Mục tiêu chính của đồ án là tìm hiểu và làm rõ các nguyên lý cơ bản của công nghệ Blockchain và ví tiền điện tử Đồ án tập trung nghiên cứu cấu trúc khối hàm băm nonce giao dịch và cơ chế liên kết các khối trong chuỗi Blockchain Thông qua việc xây dựng chương trình mô phỏng Blockchain đơn giản bằng ngôn ngữ Python đồ án giúp minh họa trực quan cách thức hoạt động của Blockchain từ đó nâng cao khả năng tiếp cận và hiểu biết của sinh viên về công nghệ này

+ Chương này sẽ trình bày tổng quan về bối cảnh thực hiện đề tài lý do lựa chọn Blockchain làm hướng nghiên cứu cùng với ý nghĩa lý luận và thực tiễn của việc tìm hiểu công nghệ Blockchain và tiền điện tử Đồng thời chương cũng đề cập đến vai trò của Blockchain trong quá trình chuyển đổi số và tiềm năng ứng dụng của công nghệ này trong các hệ thống thông tin hiện đại

+ Việc thực hiện đồ án không chỉ mang lại giá trị học thuật trong lĩnh vực công nghệ thông tin mà còn giúp sinh viên rèn luyện kỹ năng lập trình tư duy logic và khả năng phân tích hệ thống Nội dung chương này là cơ sở định hướng cho các chương tiếp theo đi sâu vào phân tích lý thuyết Blockchain thiết kế mô hình và triển khai chương trình mô phỏng một cách hiệu quả

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Blockchain là gì?

Blockchain (chuỗi khối) là một loại công nghệ sổ cái phân tán (Distributed Ledger Technology - DLT) cho phép lưu trữ thông tin dưới dạng các khối (block) được liên kết với nhau bằng mật mã học (cryptography) và sắp xếp theo trình tự thời gian. Khi một khối dữ liệu đã được thêm vào chuỗi, nó không thể bị thay đổi hoặc xóa bỏ (bất biến).

Ý tưởng nền tảng của Blockchain được đề xuất lần đầu vào năm 1991 bởi Stuart Haber và W. Scott Stornetta nhằm tạo ra một hệ thống nơi các tài liệu kỹ thuật số không thể bị làm giả mạo. Tuy nhiên, công nghệ này thực sự được biết đến rộng rãi vào năm 2008 khi một hoặc một nhóm người ẩn danh dưới tên Satoshi Nakamoto công bố sách trắng (whitepaper) về Bitcoin – một hệ thống tiền điện tử sử dụng Blockchain làm nền tảng cốt lõi

Về cơ bản, Blockchain có thể được xem là một cuốn sổ cái kế toán chung được chia sẻ và đồng bộ hóa trên hàng ngàn máy tính (node) trong một mạng lưới, thay vì chỉ được lưu trữ tại một máy chủ tập trung duy nhất.

2.1.1 Các đặc điểm cốt lõi của Blockchain

Công nghệ Blockchain sở hữu bốn đặc điểm chính giúp nó vượt trội hơn so các hệ thống quản lý dữ liệu truyền thống

- **Phi tập trung**

- Đặc điểm tính quan trọng nhất trong hệ thống truyền thống là dữ liệu được kiểm soát bởi thực thể trung gian như ngân hàng, chính phủ, máy chủ

- Blockchain loại bỏ trung gian này dữ liệu sao chép và phân tán trên nhiều máy tính tham gia mạng lưới

- Ý nghĩa không một cá nhân hay tổ chức quyền lực tuyệt đối thay đổi dữ liệu hoặc tắt mạng lưới

- **Bất biến**

- Bất biến có nghĩa là dữ liệu được ghi vào một block và thêm vào chuỗi (noce) thì không thể bị sửa đổi hoặc xóa bỏ

- Điều này đạt được nhờ việc sử dụng hàm băm mật mã (Cryptographic Hash Function) mỗi block chứa giá trị băm (hash) của block trước đó bất kỳ thay đổi nhỏ nào trong dữ liệu của một block thì sẽ bị thay đổi Hash

chính nó, kéo theo sự thay đổi của Hash tất cả block sau đó khiến toàn bộ chuỗi bị vô hiệu hóa

- Ý nghĩa: Đảm bảo tính toàn vẹn và đáng tin cậy của dữ liệu

- **Minh bạch**

- Mặc dù danh tính thực của người dùng thường được mã hóa dưới dạng địa chỉ ví (khóa công khai) nhưng tất cả các giao dịch trên chuỗi đều được công khai

- Bất kỳ ai cũng có thể kiểm tra lịch sử giao dịch của bất kỳ địa chỉ ví nào

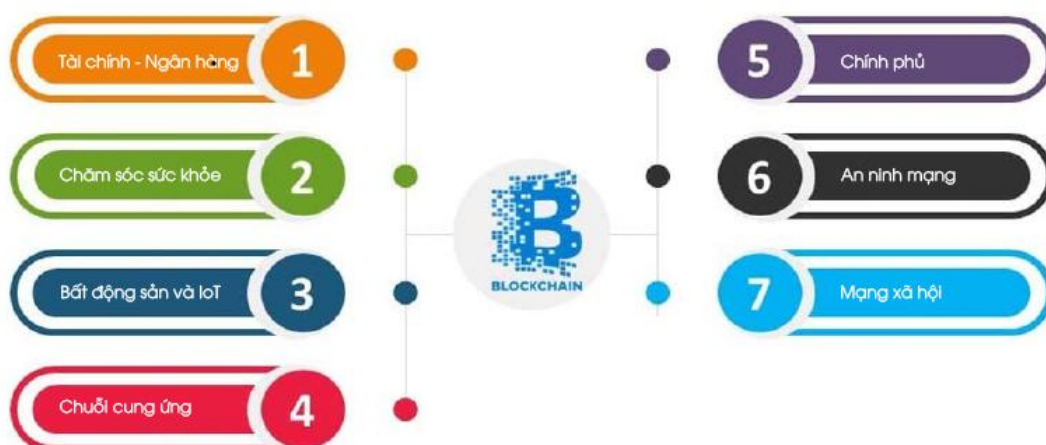
- Ý nghĩa : tạo ra hệ thống mở mọi hoạt động đều có thể xác minh mà không cần sự tin tưởng bên thứ ba

- **Bảo mật**

- Bảo mật của blockchain được đảm bảo thông qua sự kết hợp của mật mã học (để liên kết các khối) và cơ chế đồng thuận (để xác nhận tính hợp lệ của khối)

- Việc sửa đổi dữ liệu đòi hỏi việc kiểm soát hơn 51% sức mạnh tính toán của toàn bộ mạng lưới điều này gần như bất khả thi đối với các mạng lớn như Bitcoin

Ứng dụng của Blockchain



Hình 1: Ứng dụng của Blockchain trong các lĩnh vực khác nhau

2.1.2 Khối khởi thủy (Genesis Block)

Khối đầu tiên trong một chuỗi Blockchain được gọi là khối Genesis (Khối khởi thủy). Trong mã nguồn thực hiện, khối này có chỉ số (index) bằng 0. Vì không có khối nào tồn tại trước đó giá trị `previous_hash` của nó được thiết lập mặc định là một chuỗi ký tự "0". Đây là nền tảng để các khối tiếp theo liên kết vào thông qua mã băm, tạo thành một chuỗi dữ liệu bất biến.

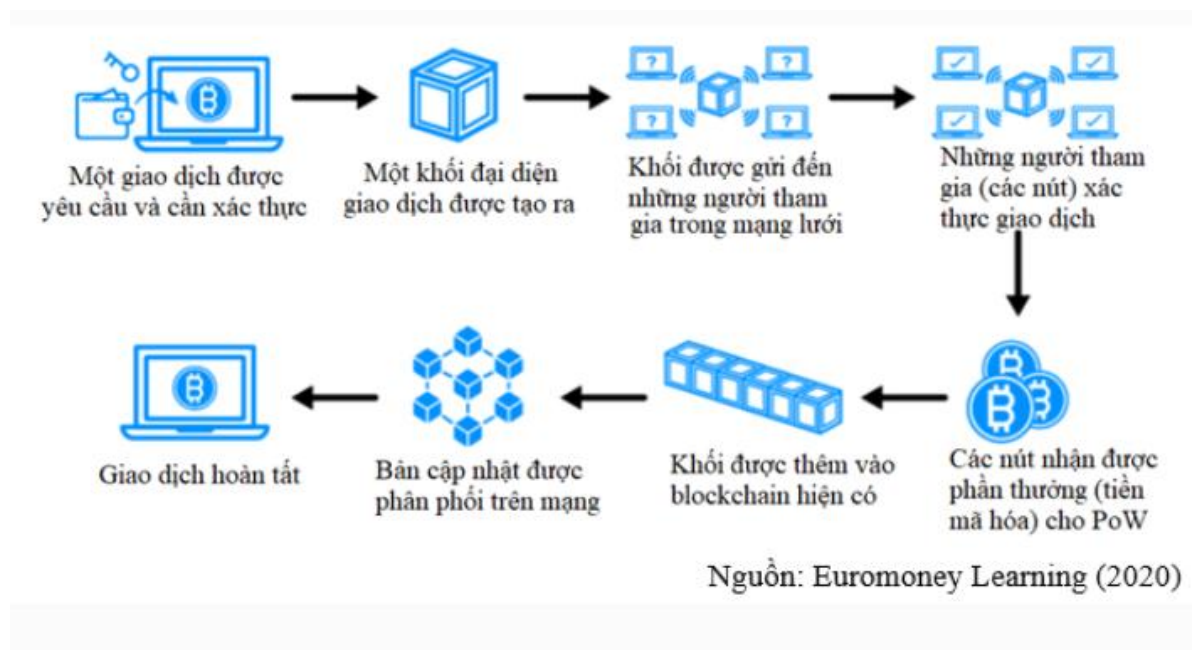
2.2 Nguyên lý hoạt động

Nguyên lý hoạt động của Blockchain là một quy trình khép kín, kết hợp giữa mạng ngang hàng (P2P), mã hóa mật mã học và cơ chế đồng thuận. Quy trình này có thể được tóm tắt qua các bước cụ thể sau:

2.2.1 Quy trình thực hiện giao dịch

Khi giao dịch người dùng xảy ra thì việc quy trình vận hành diễn ra như sau:

- **Khởi tạo giao dịch:** Người dùng tạo một yêu cầu giao dịch thông qua ví điện tử. Giao dịch này bao gồm địa chỉ người nhận số lượng và được ký bằng Khóa riêng (Private Key) của người gửi để chứng minh quyền sở hữu.
- **Phát tán (Broadcasting):** Giao dịch sau khi khởi tạo sẽ được gửi vào mạng lưới ngang hàng (P2P). Tại đây, các máy tính (node) trong mạng sẽ nhận được thông tin giao dịch này.
- **Xác thực giao dịch:** Các node kiểm tra tính hợp lệ của giao dịch (người gửi có đủ số dư không, chữ ký số có đúng không). Nếu hợp lệ, giao dịch sẽ được đưa vào hàng đợi gọi là Mempool (Memory Pool) để chờ được đóng khối.
- **Tạo khối (Mining/Forging):** Các thợ đào (Miners) sẽ gom các giao dịch từ Mempool để tạo thành một khối mới. Để khối này được chấp nhận, thợ đào phải giải một bài toán toán học phức tạp (tìm Nonce để có mã Hash hợp lệ).
- **Cập nhật vào chuỗi:** Khi một thợ đào tìm ra lời giải, khối mới sẽ được gửi đến tất cả các node khác để kiểm tra lại lần cuối. Nếu đa số các node đồng ý, khối đó sẽ được nối vào khối trước đó bằng mã Hash, và giao dịch chính thức được hoàn tất.



Hình 2: Sơ đồ hoạt động giao dịch Blockchain

2.2.2 Cấu trúc khối và Hàm băm (Hash)

1. Cấu trúc của một khối (Block Structure):

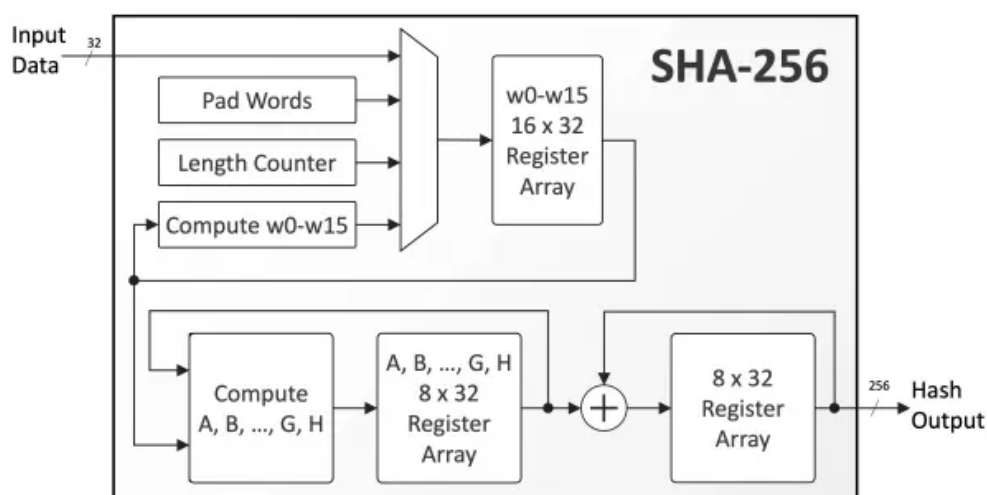
Trong Block gồm có những thành phần sau:

- **Dữ liệu giao dịch (Transaction Data):** Chứa thông tin chi tiết về các giao dịch diễn ra trong khoảng thời gian đó, bao gồm địa chỉ ví người gửi, người nhận và số lượng tài sản chuyển giao.
- **Mã Hash khối hiện tại (Current Block Hash):** Là một chuỗi ký tự duy nhất được tạo ra bằng cách băm toàn bộ thông tin trong khối. Nó đóng vai trò như "dấu vân tay" để định danh cho khối đó.
- **Mã Hash khối trước đó (Previous Block Hash):** Đây là thành phần quan trọng nhất để tạo nên chuỗi (chain). Mỗi khối (ngoại trừ khối đầu tiên) đều lưu trữ mã Hash của khối ngay trước nó, tạo thành một liên kết mật mã không thể tách rời.
- **Số Nonce (Number Used Once):** Là một giá trị số ngẫu nhiên mà các thợ đào thay đổi liên tục trong quá trình khai thác. Mục tiêu là tìm ra một giá trị Nonce sao cho khi kết hợp với dữ liệu khối, mã Hash tạo ra phải thỏa mãn độ khó của mạng lưới.
- **Dấu thời gian (Timestamp):** Ghi lại thời điểm chính xác khối được tạo ra, giúp sắp xếp các khối theo trình tự thời gian.

2. Hàm băm mật mã (SHA-256)

Hàm băm là nền tảng bảo mật của Blockchain, trong đó thuật toán SHA-256 (Secure Hash Algorithm 256-bit) là phổ biến nhất.

- **Tính định danh duy nhất:** Hàm băm chuyển đổi dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi ký tự có độ dài cố định (256 bit). Mỗi dữ liệu đầu vào chỉ cho ra một kết quả băm duy nhất.
- **Hiệu ứng thác đổ (Avalanche Effect):** Một đặc điểm quan trọng là tính nhạy cảm với thay đổi dữ liệu. Chỉ cần thay đổi một ký tự nhỏ nhất trong dữ liệu giao dịch, toàn bộ mã Hash của khối sẽ thay đổi hoàn toàn.
- **Đảm bảo tính bất biến:** Vì khối sau lưu mã Hash của khối trước, nếu dữ liệu ở khối cũ bị sửa đổi, mã Hash của nó sẽ thay đổi, dẫn đến mã Hash lưu ở khối kế tiếp bị sai lệch. Điều này khiến toàn bộ chuỗi từ điểm bị sửa đổi trở nên vô hiệu, giúp hệ thống phát hiện gian lận ngay lập tức.



Hình 3: Sơ đồ xử lý dữ liệu của hàm băm SHA-256

2.3 Mạng ngang hàng P2P và cơ chế đồng thuận

2.3.1 Khái niệm mạng ngang hàng P2P (Peer to Peer)

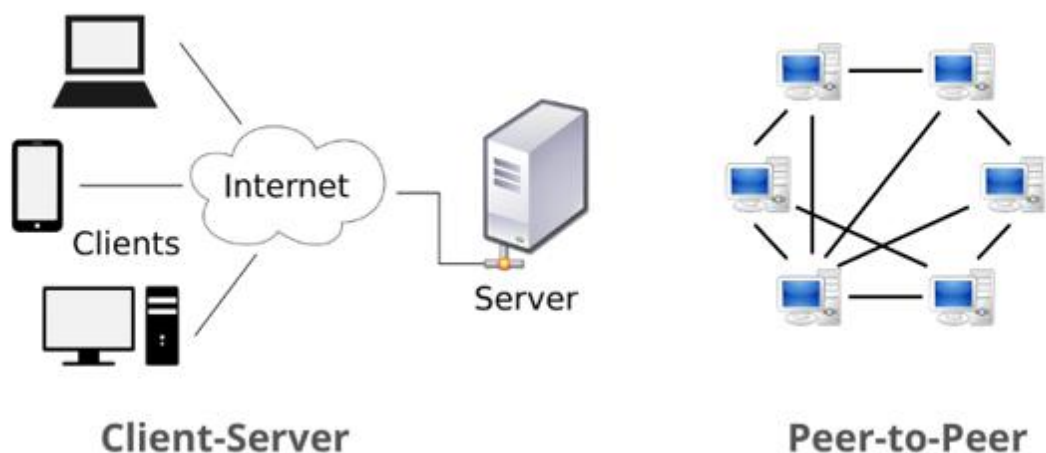
Mạng ngang hàng (P2P) là nền tảng hạ tầng của công nghệ Blockchain. Khác với mô hình Client-Server truyền thống (nơi các máy con phụ thuộc vào một máy chủ trung tâm), mạng P2P cho phép tất cả các máy tính tham gia (gọi là các Node) có quyền hạn và trách nhiệm ngang nhau.

Trong hệ thống Blockchain, mỗi Node không chỉ là một thực thể tham gia giao dịch mà còn là một "người gác cổng" lưu trữ toàn bộ lịch sử dữ liệu. Kiến trúc này tạo ra một hệ thống không có điểm yếu tập trung (Single Point of Failure).

2.3.2 Cơ chế truyền tin và Lưu trữ phân tán

Trong kiến trúc mạng ngang hàng của Blockchain, các Node (nút mạng) không chỉ đóng vai trò là thực thể thực hiện giao dịch mà còn là các máy chủ lưu trữ dữ liệu độc lập.

- **Cơ chế Lan truyền (Gossip Protocol):** Khi một giao dịch mới hoặc một khối mới được tạo ra, nó sẽ được gửi tới các Node lân cận. Các Node này sau khi kiểm tra tính hợp lệ sẽ tiếp tục chuyển tiếp cho các Node khác. Quá trình này diễn ra cực nhanh, giúp toàn bộ mạng lưới đạt được sự đồng bộ chỉ trong vài giây.
- **Sổ cái phân tán (Distributed Ledger):** Thay vì một ngân hàng giữ sổ cái, mỗi Node trong mạng P2P đều giữ một bản sao của Blockchain. Điều này đảm bảo rằng ngay cả khi 90% số Node bị sập, dữ liệu vẫn tồn tại nguyên vẹn trên 10% số Node còn lại.



Hình 4: Sự khác nhau giữa client-server (mạng truyền thống) và P2P

2.4 Cơ chế đồng thuận Proof of Work (PoW)

2.4.1 Nguyên lý hoạt động của cơ chế PoW

Proof of Work (PoW) là cơ chế đồng thuận được sử dụng trong Blockchain nhằm giúp tất cả các máy tính (node) trong mạng thống nhất với nhau về dữ liệu giao dịch mà không cần bên trung gian quản lý. Cơ chế này đảm bảo rằng mọi giao dịch được ghi vào Blockchain đều hợp lệ và không bị chỉnh sửa.

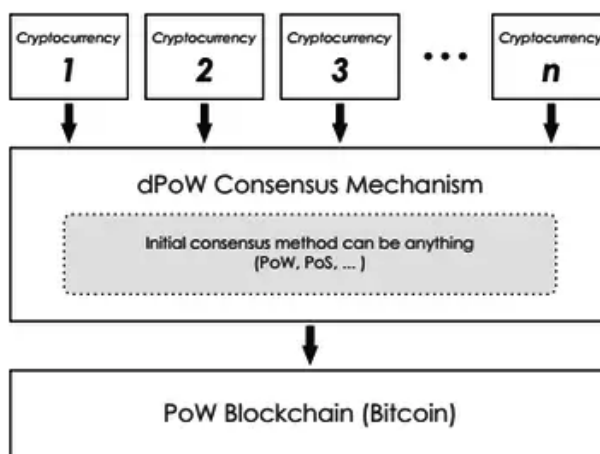
Trong PoW, các thợ đào (miner) tham gia vào quá trình khai thác bằng cách sử dụng sức mạnh tính toán của máy tính để giải một bài toán mật mã. Cụ thể, thợ đào phải tìm ra một giá trị gọi là *Nonce* sao cho khi kết hợp với dữ liệu của khối và đưa qua hàm băm sẽ tạo ra một giá trị thỏa mãn điều kiện mà mạng lưới đặt ra. Quá trình này đòi hỏi phải thử rất nhiều lần nên tiêu tốn nhiều tài nguyên tính toán.

Khi một thợ đào tìm được Nonce hợp lệ, khối mới sẽ được tạo ra và phát tán đến toàn bộ mạng Blockchain. Các node khác trong mạng sẽ tiến hành kiểm tra lại khối, bao gồm việc xác minh các giao dịch và giá trị băm của khối. Nếu khối được xác nhận là hợp lệ, nó sẽ được thêm vào chuỗi khối và trở thành một phần của sổ cái phân tán.

Cơ chế PoW giúp đảm bảo tính bảo mật cho Blockchain vì việc thay đổi dữ liệu trong một khối đã được xác nhận sẽ yêu cầu phải tính toán lại toàn bộ các khối phía sau, điều này gần như không khả thi. Để khuyến khích các thợ đào tham gia duy trì mạng lưới, người tạo ra khối hợp lệ đầu tiên sẽ nhận được phần thưởng bằng tiền điện tử, chẳng hạn như Bitcoin, cùng với phí giao dịch.

2.4.2 Chức năng chính của PoW

- **Khai thác (Mining):** Các thợ đào sử dụng sức mạnh tính toán để tìm giá trị Nonce phù hợp nhằm tạo ra một khối mới.
- **Xác thực khối:** Khi một thợ đào tìm được *Nonce* đúng, khối sẽ được gửi cho các node khác trong mạng để kiểm tra lại tính hợp lệ trước khi thêm vào Blockchain.
- **Đảm bảo an toàn:** Việc phải tiêu tốn nhiều tài nguyên tính toán giúp ngăn chặn các hành vi gian lận và tấn công mạng.
- **Phần thưởng:** Thợ đào tạo được khối hợp lệ sẽ nhận phần thưởng là tiền điện tử (ví dụ: Bitcoin) và phí giao dịch.



Hình 5: Sơ đồ hoạt động cơ chế đồng thuận PoW

2.5 Ví tiền điện tử (Crypto Wallet)

2.5.1 Nguyên lý hoạt động của ví tiền điện tử (Crypto Wallet)

Ví tiền điện tử là công cụ được sử dụng để quản lý và thực hiện các giao dịch tiền điện tử trong hệ thống Blockchain. Ví không lưu trữ tiền trực tiếp mà lưu trữ các thông tin cần thiết để chứng minh quyền sở hữu tài sản số của người dùng.

Mỗi ví tiền điện tử chứa một cặp khóa bao gồm khóa công khai (Public Key) và khóa riêng (Private Key). Khóa công khai được dùng để tạo địa chỉ ví, cho phép người khác gửi tiền điện tử đến. Địa chỉ ví có thể được chia sẻ công khai mà không ảnh hưởng đến tính an toàn của tài sản.

Khóa riêng đóng vai trò rất quan trọng vì nó được sử dụng để ký số và xác nhận các giao dịch chuyển tiền. Khi người dùng muốn gửi tiền điện tử, ví sẽ dùng khóa riêng để tạo chữ ký số, chứng minh rằng giao dịch đó là hợp lệ và được thực hiện bởi chủ sở hữu thực sự của tài sản. Do đó, việc bảo mật khóa riêng là yếu tố then chốt trong việc bảo vệ tiền điện tử.

Ví tiền điện tử giúp người dùng dễ dàng theo dõi số dư, gửi và nhận tiền điện tử, đồng thời kiểm soát hoàn toàn tài sản của mình mà không cần thông qua ngân hàng hay tổ chức trung gian. Tuy nhiên, nếu người dùng làm mất khóa riêng, quyền truy cập vào tài sản số sẽ bị mất vĩnh viễn và không thể khôi phục.

2.5.2 Chức năng chính ví tiền điện tử (Crypto Wallet)

- **Lưu trữ khóa:** Ví lưu trữ khóa công khai (*Public Key*) và khóa riêng (*Private Key*), không lưu trữ tiền trực tiếp.
- **Tạo địa chỉ ví:** Từ khóa công khai, ví tạo ra địa chỉ để người khác có thể gửi tiền điện tử vào.
- **Thực hiện giao dịch:** Ví dùng khóa riêng để ký số và xác nhận các giao dịch chuyển tiền.
- **Bảo mật tài sản:** Người dùng kiểm soát hoàn toàn tài sản của mình thông qua khóa riêng; nếu mất khóa riêng sẽ mất quyền truy cập vào tiền điện tử.

2.5.3 Cơ chế khôi phục ví và tính bảo mật

Ví tiền điện tử hoạt động dựa trên cặp khóa bất đối xứng: khóa công khai (Public Key) và khóa riêng (Private Key). Khóa công khai được dùng để tạo địa chỉ ví, trong khi khóa riêng được dùng để ký số giao dịch. Nếu người dùng làm mất khóa riêng, quyền truy cập vào tài sản sẽ bị mất vĩnh viễn. Trong thực tế, các hệ thống hiện đại thường sử dụng cụm từ gợi nhớ (Mnemonic) để giúp người dùng sao lưu khóa riêng một cách dễ dàng hơn.

2.6 Kết luận

Thông qua quá trình nghiên cứu chi tiết tại Chương 2, chúng ta đã xây dựng được một bức tranh toàn cảnh và sâu sắc về nền tảng lý thuyết của công nghệ Blockchain. Đây không chỉ là một công nghệ lưu trữ dữ liệu thông thường mà là một sự kết hợp tinh vi giữa toán học mật mã, lý thuyết mạng ngang hàng và các học thuyết kinh tế về sự tin cậy.

Thứ nhất, về cấu trúc và tính toàn vẹn: Nghiên cứu đã chỉ rõ cách thức các khối (blocks) được liên kết chặt chẽ thông qua hàm băm SHA-256. Việc mỗi khối lưu trữ mã băm của khối trước đó tạo nên một cấu trúc "chuỗi" bất biến. Bất kỳ một nỗ lực xâm nhập hay chỉnh sửa dữ liệu nào dù là nhỏ nhất ở một khối đơn lẻ cũng sẽ dẫn đến hiệu ứng "thác đổ", làm thay đổi toàn bộ các mã băm phía sau và ngay lập tức bị mạng lưới đào thải. Điều này thiết lập một tiêu chuẩn mới về tính minh bạch và bảo mật dữ liệu.

Thứ hai, về cơ chế vận hành phi tập trung: Việc nghiên cứu mạng ngang hàng (P2P) đã cho thấy ưu điểm vượt trội của Blockchain so với mô hình máy chủ tập trung truyền thống. Bằng cách phân tán quyền kiểm soát và lưu trữ sổ cái cho mọi nút mạng (Nodes), hệ thống đã loại bỏ được rủi ro từ "điểm yếu tập trung", đảm bảo tính sẵn sàng cao và khả năng chống lại các cuộc tấn công từ bên ngoài một cách hiệu quả.

Thứ ba, về kỹ thuật số: Cơ chế đồng thuận Proof of Work (PoW) và hệ thống ví tiền điện tử dựa trên cặp khóa công khai - riêng tư (Public/Private Key) đã giải quyết triệt để bài toán về lòng tin trong môi trường số. Người dùng có thể thực hiện giao dịch trực tiếp với nhau một cách an toàn mà không cần đến bất kỳ một tổ chức trung gian tài chính nào đứng ra bảo đảm.

Tóm lại những nội dung lý thuyết về cấu trúc khối, hàm băm, mạng P2P và cơ chế bảo mật khóa chính cho toàn bộ đề án. Những kiến thức vững chắc này không chỉ giúp chúng ta hiểu rõ bản chất của công nghệ Blockchain mà còn là nền tảng thực tiễn quan trọng để triển khai phần thực nghiệm tại

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Phương pháp nghiên cứu

3.1.1 Tải phần mềm

- **VISUAL STUDIO CODE:** phần mềm để lập trình cho python



Hình 6: Phần mềm lập trình VSCODE

- **ANACONDA:** là phần mềm quản lý môi trường cho phép các gói thư viện mạnh mẽ được sử dụng phổ biến trong khoa học phân tích dữ liệu học máy và lập trình Python.



Hình 7: Phần mềm quản lý thư viện Anaconda (Python)

- **JUPYTER NOTEBOOK:** là một ứng dụng mã nguồn mở được sử dụng để viết và thực thi mã lập trình theo cách tương tác. Đây là công cụ cực kỳ phổ biến trong khoa học dữ liệu, học máy, và phân tích dữ liệu



Hình 8: Phần mềm mô phỏng JUPYTER

3.1.2 Thư viện lập trình

- **Flask:** tạo web server và các API endpoints cho ứng dụng blockchain
- **Werkzeug:** chạy Flask server trong môi trường Jupyter Notebook
- **Ecdsa:** tạo ví tiền điện tử, ký và xác thực chữ ký số bằng thuật toán SECP256k1 (giống Bitcoin)

- **Công cụ phát triển:**

- **Python (phiên bản 3.13.3) :** ngôn ngữ lập trình chính
- **Jupyter Notebook :** môi trường chạy và trình bày code
- **Web Browser:** hiển thị giao diện web demo

CÔNG NGHỆ WEB (nhúng trong Python):

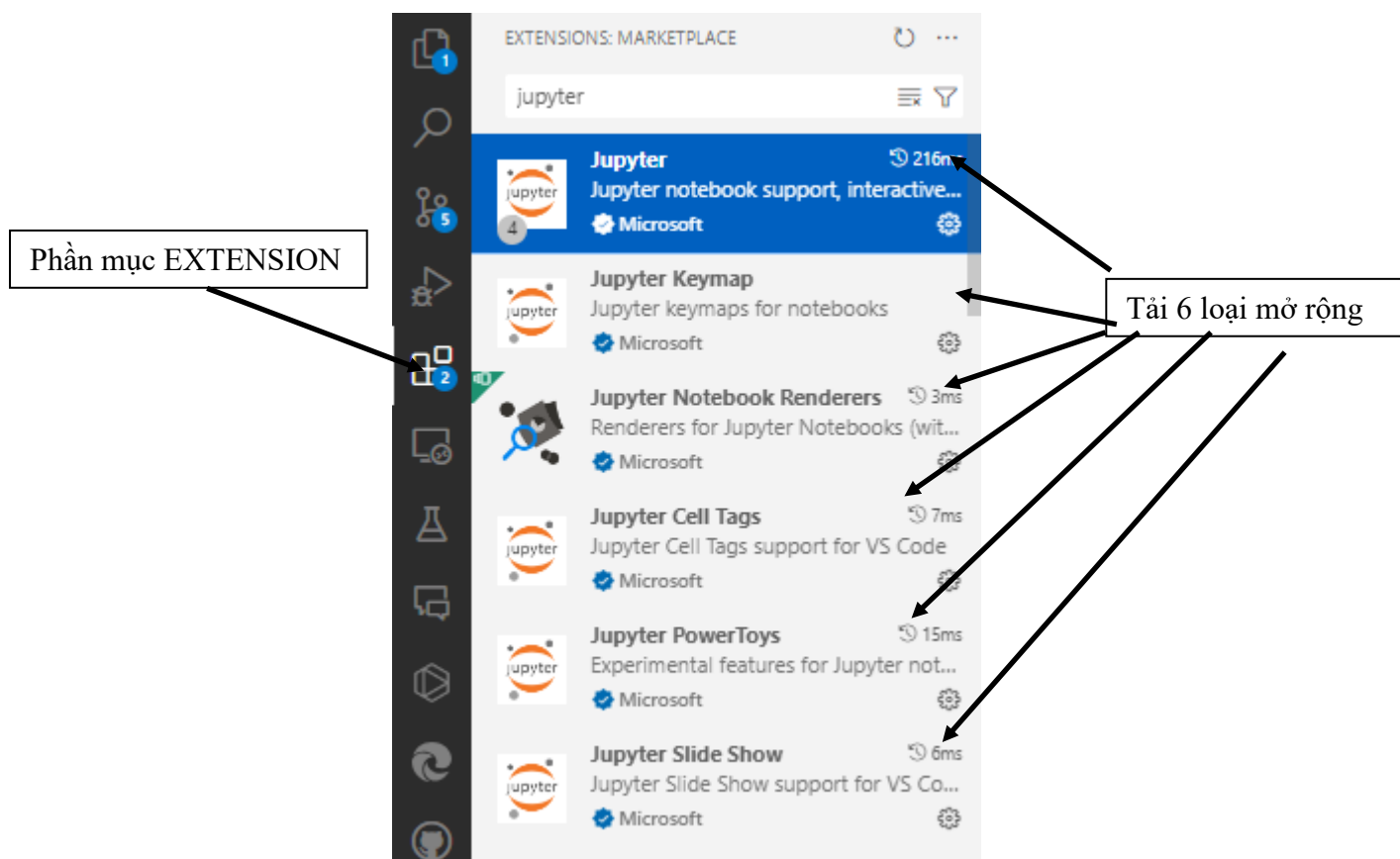
- **HTML5:** cấu trúc giao diện web
- **CSS3:** tạo style, gradient và hiệu ứng cho giao diện
- **JavaScript:** xử lý tương tác người dùng, gọi API bất đồng bộ

3.2 Phương pháp thực hiện

Để làm được các thuật toán mô phỏng thì trước tiên chúng ta phải cài đặt phần mềm quản lý ANACONDA để liên kết qua VISUAL STUDIO CODE

Lý do: việc cài python từ thư viện chính của python sẽ thiếu đi thư viện mô phỏng và khi cài đặt thư viện từ python sẽ nặng dung lượng hơn nên việc chúng ta sẽ cài phần mềm ANACONDA để tích hợp các thư viện có sẵn từ thư viện này mà không đầy bộ nhớ dung lượng và nhẹ hơn

Về phần JUPYTER NOTEBOOK thì chúng ta sẽ cài đặt trong VISUAL STUDIO CODE bằng vào EXTENSION → chọn 6 loại mở rộng chính của JUPYTER như hình sau:



Hình 9: Phần mở rộng JUPYTER

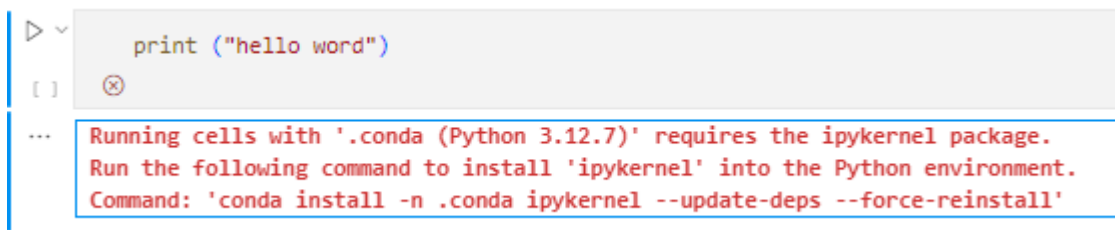
Sau khi tải xong các phần EXTENSION về thì tiếp theo chúng ta tạo 1 file JUYTER xem có chạy được hay chưa nếu chạy chương trình thành công thì nó sẽ hiện lên còn không thì chúng ta sẽ ngược lại:

- **Hình ảnh:**

```

▶  print ("Hello world")
[1]  ✓  0.0s
...  Hello world
  
```

Hình 10: Chương trình chạy thành công



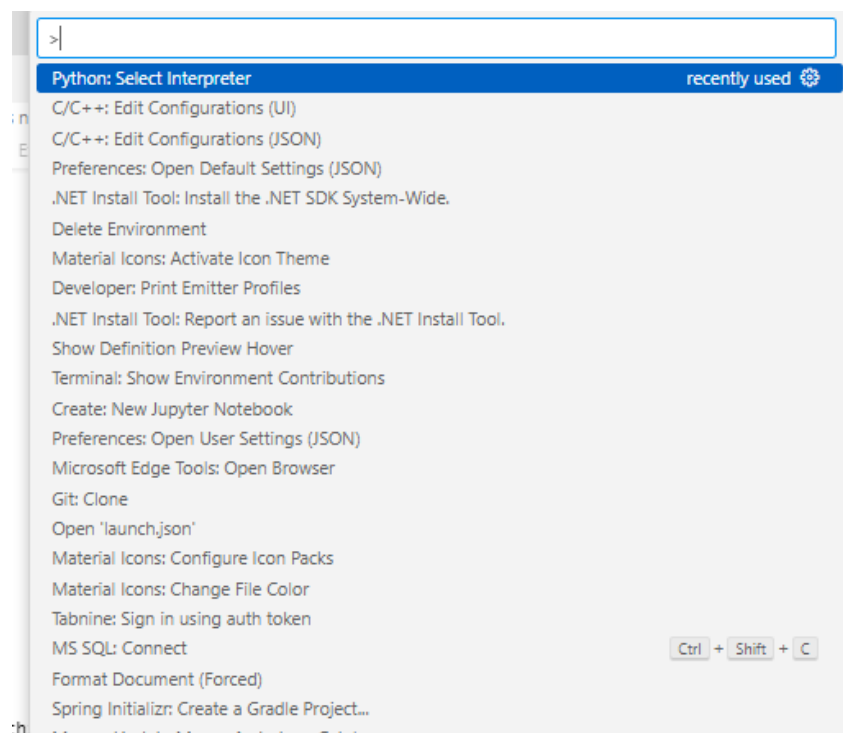
Hình 11: Chương trình chạy không thành công

Trong trường hợp lỗi này thì chúng ta sẽ giải quyết bằng cách thực hiện như sau:

B1: chúng ta sẽ sử dụng đối với các máy tính windows thì chúng ta sẽ sử dụng tổ hợp phím CTRL + SHIFT + P để mở Command lên sao đó nhập bảng lệnh

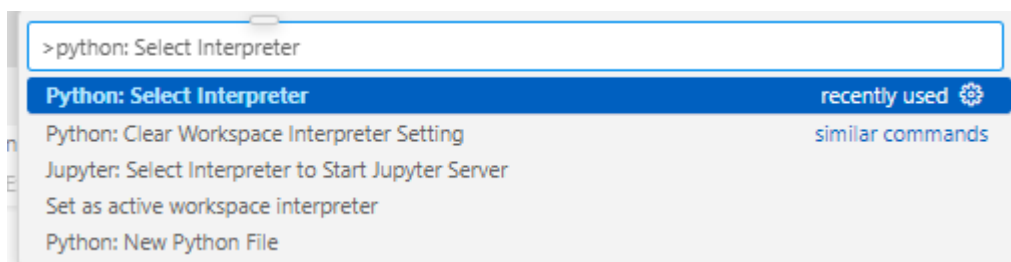
Đối với các máy MACOS thì chúng ta sẽ sử dụng Command + Shift + P để vào Command của visual studio code

Hình ảnh Command của **VISUAL STUDIO CODE**:



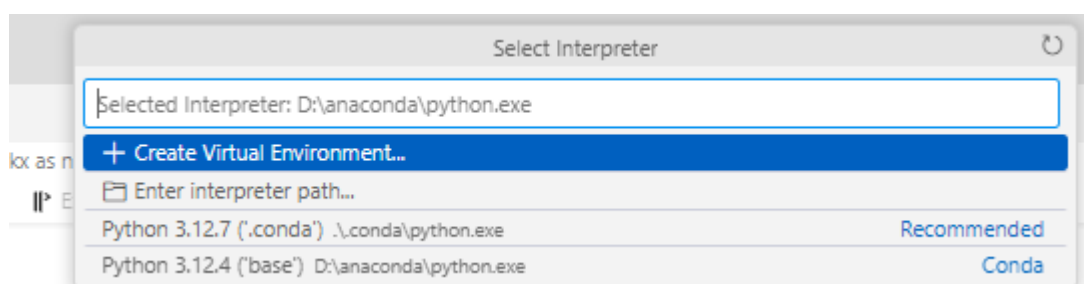
Hình 12: Bảng lệnh command VSCODE

Lúc này chúng ta sẽ nhập lệnh Python: Select Interpreter



Hình 13: Nhập lệnh command “ Python Select Interpreter”

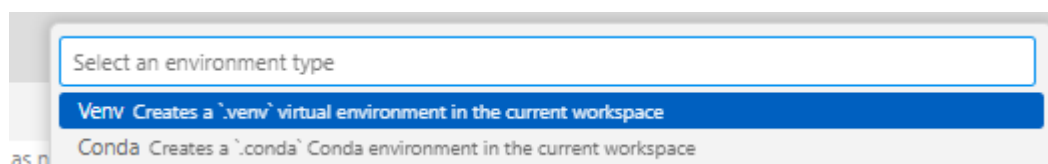
Về bước này chúng ta sẽ kiểm tra xem nó có tích hợp hay chưa nếu có thì nó sẽ hiện lên:



Hình 14: Chọn thư viện Python bằng bảng command

Hình ảnh 11: tạo môi trường thư viện python

Còn không hiện lên thì chúng ta sẽ đến phần tiếp theo đó là tạo 1 thư viện “CREATE VIRTUAL ENVIRONMENT”



Hình 15: Chọn Python Venv (Python gốc)

- Mô tả

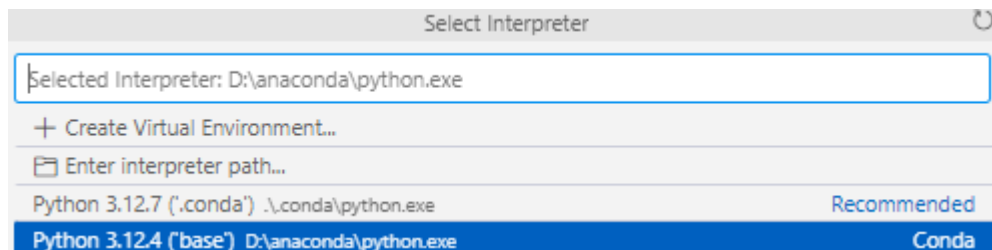
Ở đây có có 2 phần thuộc tính của thư viện:

Venv: thư viện này là thư viện để cho mục Python gốc khi tải xuống và hiện lên trên phần này

Conda: đây là thư viện của ANACONDA nên nó sẽ tích hợp sẵn cho chúng ta ở phần này

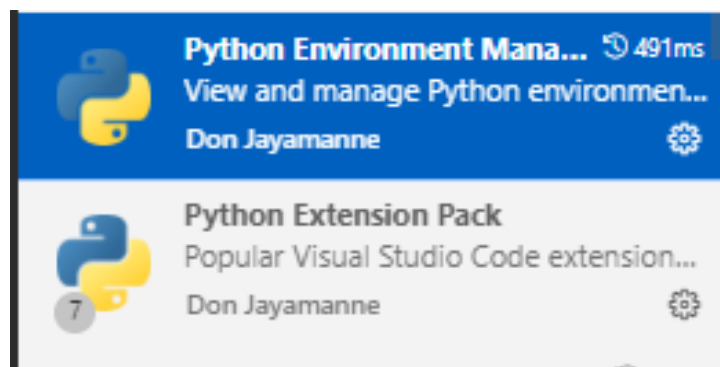
Trong phần này thì chúng ta sẽ chọn Conda vì khi chúng ta chạy chương trình VISUAL STUDIO CODE sẽ tự động liên kết qua thư viện python do đó việc sử dụng trở nên dễ dàng hơn và tiện lợi hơn.

Khi bạn chọn xong thì VISUAL STUDIO CODE sẽ tự động liên kết đồng bộ qua ANACONDA



Hình 16: Chọn thư viện Base (Python của Anaconda)

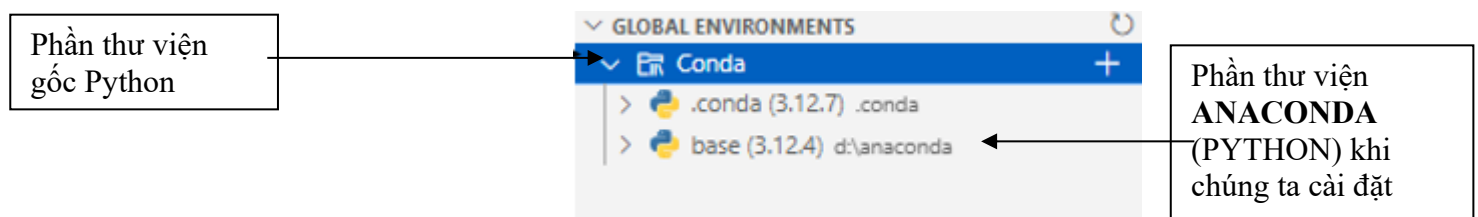
Phần tiếp theo chúng ta sẽ quay lại phần tải mở rộng lúc này chúng ta sẽ phải tải phần mở rộng quản lý thư viện Python



Hình 17: Phần mở rộng quản lý thư viện Python

- Mô tả:

2 phần mở rộng này để giúp cho chúng ta có thể xem được các cái thư viện để cài đặt và di chuyển thư viện tùy theo sở thích sử dụng thư viện nào trong Python.



Hình 18: Thư viện được tích hợp sẵn khi cài đặt

3.3 Mô tả code và giải thích

3.3.1 Phần chuyên mục báo cáo:

- Phần chuyên mục bao gồm:

- Miêu tả code
- Hình ảnh code
- Kết thúc

➤ **Danh mục code:**

- **CLASS BLOCK - Cấu trúc một khối trong blockchain:**

```
class Block:

    def __init__(self, index, transactions, previous_hash, nonce=0):

        self.index = index          # Số thứ tự của block

        self.transactions = transactions  # Danh sách giao dịch

        self.previous_hash = previous_hash  # Hash của block trước

        self.timestamp = time.time()      # Thời gian tạo block

        self.nonce = nonce                # Số dùng để đào

        self.hash = self.calculate_hash()  # Hash của block này


    def calculate_hash(self):

        # Gom tất cả dữ liệu block thành một chuỗi JSON

        block_data = {

            "index": self.index,

            "transactions": self.transactions,

            "previous_hash": self.previous_hash,

            "timestamp": self.timestamp,

            "nonce": self.nonce

        }
```

```
block_string = json.dumps(block_data, sort_keys=True).encode()

# Áp dụng hàm băm SHA-256 để tạo hash
return hashlib.sha256(block_string).hexdigest()

def mine_block(self, difficulty):

    # Tạo target: chuỗi N số 0 (N = difficulty)
    target = "0" * difficulty

    # Lặp cho đến khi tìm được hash bắt đầu bằng N số 0
    while self.hash[:difficulty] != target:

        self.nonce += 1          # Tăng nonce

        self.hash = self.calculate_hash() # Tính lại hash

    return self.hash
```

- **CLASS BLOCKCHAIN - Chuỗi khối:**

```
class Blockchain:

    def __init__(self):

        # Khởi tạo chain với Genesis Block (block đầu tiên)
        self.chain = [Block(0, [], "0")]

        self.difficulty = 3          # Độ khó mining (3 số 0)

        self.pending_transactions = [] # Giao dịch chờ xử lý

        self.mining_reward = 50      # Phần thưởng cho thợ đào

    def add_transaction(self, sender, receiver, amount):

        # Thêm giao dịch mới vào hàng đợi

        self.pending_transactions.append({

            "sender": sender,
```

```
        "receiver": receiver,

        "amount": amount

    })

def mine_pending_transactions(self, miner_address):

    # Tạo block mới chứa các giao dịch đang chờ

    new_block = Block(

        len(self.chain),          # Index = số block hiện có

        self.pending_transactions.copy(), # Copy giao dịch pending

        self.chain[-1].hash      # Hash của block cuối

    )

    new_block.mine_block(self.difficulty) # Đào block

    self.chain.append(new_block)         # Thêm vào chain

    # Reset pending và thêm phần thưởng cho thợ đào

    self.pending_transactions = [{

        "sender": "Network",

        "receiver": miner_address,

        "amount": self.mining_reward

    }]

    return new_block

def get_balance(self, address):

    # Tính số dư bằng cách duyệt qua tất cả giao dịch

    balance = 0

    for block in self.chain:
```

```
for tx in block.transactions:

    if tx.get("receiver") == address:

        balance += tx.get("amount", 0) # Cộng tiền nhận

    if tx.get("sender") == address:

        balance -= tx.get("amount", 0) # Trừ tiền gửi

return balance
```

- **CLASS WALLET - Ví tiền điện tử với ECDSA**

```
class Wallet:

    def __init__(self, name=""):

        self.name = name

        # Tạo cặp khóa bất đối xứng bằng thuật toán SECP256k1

        self._private_key =
ecdsa.SigningKey.generate(curve=ecdsa.SECP256k1)

        self._public_key = self._private_key.get_verifying_key()

    @property
    def public_key(self):

        # Public key dùng làm địa chỉ ví (công khai)

        return binascii.hexlify(self._public_key.to_string()).decode()

    @property
    def private_key(self):

        # Private key dùng để ký giao dịch (bí mật)

        return binascii.hexlify(self._private_key.to_string()).decode()
```

```
def sign_transaction(self, receiver, amount):

    # Tạo dữ liệu giao dịch

    tx_data = {

        "sender": self.public_key,

        "receiver": receiver,

        "amount": amount

    }

    tx_string = json.dumps(tx_data, sort_keys=True).encode()

    # Ký giao dịch bằng private key

    signature = self._private_key.sign(tx_string)

    return {**tx_data, "signature": binascii.hexlify(signature).decode()}


@staticmethod

def verify_transaction(transaction):

    # Khôi phục public key từ địa chỉ người gửi

    sender_key = ecdsa.VerifyingKey.from_string(

        binascii.unhexlify(transaction["sender"]),

        curve=ecdsa.SECP256k1

    )

    tx_data = {

        "sender": transaction["sender"],

        "receiver": transaction["receiver"],

        "amount": transaction["amount"]

    }

    tx_string = json.dumps(tx_data, sort_keys=True).encode()

    signature = binascii.unhexlify(transaction["signature"])
```

```
# Xác thực chữ ký bằng public key  
return sender_key.verify(signature, tx_string)
```

- **CLASS P2PNODE - Mô phỏng mạng P2P:**

```
class P2PNode:  
  
    def __init__(self, name):  
        self.name = name  
  
        self.blockchain = Blockchain() # Mỗi node có bản sao blockchain  
  
        self.peers = [] # Danh sách các node kết nối  
  
    def connect_to_peer(self, peer):  
        # Kết nối 2 chiều giữa các nodes  
  
        if peer not in self.peers and peer != self:  
            self.peers.append(peer)  
            peer.peers.append(self)  
  
    def broadcast_block(self, block):  
        # Gửi block mới đến tất cả peers  
  
        for peer in self.peers:  
            peer.receive_block(block)  
  
    def receive_block(self, block):  
        # Xác thực block nhận được  
  
        if block.hash == block.calculate_hash():  
            # Kiểm tra liên kết với block cuối  
  
            if block.previous_hash == self.blockchain.chain[-1].hash:
```

```
self.blockchain.chain.append(copy.deepcopy(block))
```

- **CLASS AUTOMINER - Tự động đào block:**

```
class AutoMiner:

    def __init__(self, blockchain):

        self.blockchain = blockchain

        self.is_running = False      # Trạng thái chạy

        self.timer = None             # Timer để lập lịch

        self.interval = 10            # Khoảng cách giữa các block (giây)

        self.miner_address = "Miner"  # Địa chỉ nhận thưởng


    def start(self, miner_address, interval):

        self.miner_address = miner_address

        self.interval = interval

        self.is_running = True

        self._schedule_next_mine()


    def _schedule_next_mine(self):

        if self.is_running:

            # Đặt timer để đào block sau interval giây

            self.timer = threading.Timer(self.interval, self._do_mine)

            self.timer.daemon = True

            self.timer.start()


    def _do_mine(self):

        if not self.is_running:
```

```
return
```

```
self.blockchain.mine_pending_transactions(self.miner_address)
```

```
self._schedule_next_mine()
```

3.3.2 Phần miêu tả code:

Để chúng ta tạo mô phỏng hoàn chỉnh thì chúng ta sẽ sử dụng phương pháp và thư viện như sau:

❖ Tạo hash bằng hashlib:

- Sử dụng thư viện hashlib để tạo các hash SHA-256 cho block.
- Hàm sha256() nhận dữ liệu đầu vào (chuỗi JSON chứa thông tin block) và trả về chuỗi hash 64 ký tự.
- Hash đảm bảo tính toàn vẹn dữ liệu: thay đổi 1 bit dữ liệu sẽ tạo ra hash hoàn toàn khác.

❖ Tính toán Proof of Work (Mining):

- Sử dụng phương pháp brute-force để tìm nonce thỏa mãn điều kiện độ khó.
- Vòng lặp while liên tục tăng nonce và tính lại hash cho đến khi hash bắt đầu bằng N số 0.
- Độ khó (difficulty) quyết định số lượng số 0 cần có ở đầu hash.

❖ Ký và xác thực giao dịch bằng ecdsa:

- Sử dụng thư viện ecdsa với đường cong SECP256k1 (giống Bitcoin) để tạo cặp khóa bất đối xứng.
- Private key dùng để ký giao dịch, public key dùng để xác thực chữ ký.
- Hàm sign() tạo chữ ký số, hàm verify() kiểm tra tính hợp lệ của chữ ký.

❖ Mô phỏng mạng P2P:

- Tạo class P2PNode đại diện cho mỗi node trong mạng.
- Các node kết nối với nhau qua danh sách peers và broadcast block mới đến tất cả nodes.
- Mỗi node xác thực block trước khi thêm vào blockchain của mình.

❖ Tạo giao diện web bằng Flask:

- Sử dụng Flask để tạo web server và các API endpoints.
- HTML/CSS/JavaScript được nhúng trực tiếp trong code Python để tạo giao diện người dùng.
- Các API như /api/chain, /api/tx, /api/balance phục vụ cho việc tương tác với blockchain.

❖ Đào block tự động bằng threading:

- Sử dụng threading.Timer để lập lịch đào block theo khoảng thời gian cố định.
- AutoMiner chạy trong background thread, không ảnh hưởng đến giao diện người dùng.
- Mô phỏng giống Bitcoin thực tế với block time cố định (~10 giây trong demo).

3.3.3 Logic xác thực số dư và chống chi tiêu gấp đôi (Double Spending)

Để đảm bảo tính toàn vẹn của hệ thống tài chính mô phỏng, thuật toán tính số dư được thực hiện bằng cách duyệt qua toàn bộ lịch sử các khối trong chuỗi.

- Hệ thống duyệt qua từng khối (block) và từng giao dịch (transaction) bên trong.
- Nếu địa chỉ ví đóng vai trò là người nhận (receiver), số dư sẽ được cộng thêm số tiền tương ứng.
- Nếu địa chỉ ví đóng vai trò là người gửi (sender), số dư sẽ bị trừ đi số tiền đó. Cơ chế này giúp ngăn chặn việc một người dùng gửi số tiền vượt quá số dư họ hiện có trong ví

3.3.4 Chi tiết thuật toán ký số ECDSA

Đồ án sử dụng thư viện ecdsa với đường cong SECP256k1, đây chính là tiêu chuẩn mà mạng Bitcoin đang sử dụng.

- **Quá trình ký:** Dữ liệu giao dịch bao gồm người gửi, người nhận và số tiền được chuyển thành chuỗi JSON. Khóa riêng sẽ được dùng để tạo ra một chữ ký số (signature) duy nhất cho dữ liệu này.
- **Quá trình xác thực:** Các node khác trong mạng sẽ dùng khóa công khai của người gửi để kiểm tra chữ ký. Nếu dữ liệu bị thay đổi dù chỉ 1 bit, chữ ký sẽ không còn khớp và giao dịch bị từ chối.

Kết Luận chương 3:

Qua quá trình hiện thực hóa nghiên cứu trong Chương 3, chúng ta đã xây dựng thành công một hệ thống mô phỏng blockchain hoàn chỉnh bằng ngôn ngữ Python. Hệ thống này bao gồm đầy đủ các thành phần cốt lõi của một blockchain thực tế như Bitcoin, từ cấu trúc block, thuật toán băm SHA-256, cơ chế đồng thuận Proof of Work, cho đến ví tiền điện tử với chữ ký số ECDSA.

Về mặt kỹ thuật, chương trình đã triển khai thành công các chức năng quan trọng. Đầu tiên là cấu trúc Block với đầy đủ các thành phần bao gồm index (số thứ tự), timestamp (mốc thời gian), transactions (danh sách giao dịch), previous_hash (hash của block trước), nonce (số dùng để đào) và hash (hash của block hiện tại). Các thành phần này được liên kết chặt chẽ với nhau nhằm đảm bảo tính toàn vẹn và bất biến của dữ liệu.

Tiếp theo là thuật toán băm SHA-256 được sử dụng để tạo "dấu vân tay số" cho mỗi block. Đây là hàm một chiều nghĩa là không thể tìm lại dữ liệu gốc từ hash. Bất kỳ thay đổi nhỏ nào trong dữ liệu đều tạo ra hash hoàn toàn khác giúp phát hiện mọi hành vi giả mạo.

Cơ chế Proof of Work mô phỏng quá trình đào block trong Bitcoin. Thợ đào phải tìm một số nonce sao cho hash của block bắt đầu bằng N số 0 với N là độ khó. Quá trình này tiêu tốn tài nguyên tính toán nhằm đảm bảo tính bảo mật và phi tập trung của mạng lưới.

Ví tiền điện tử sử dụng thuật toán ECDSA (Elliptic Curve Digital Signature Algorithm) với đường cong SECP256k1 là cùng loại mà Bitcoin sử dụng. Mỗi ví có cặp khóa bất đối xứng gồm private key để ký giao dịch và public key làm địa chỉ ví. Chữ ký số đảm bảo chỉ chủ sở hữu private key mới có thể thực hiện giao dịch và không ai có thể giả mạo.

Mô phỏng mạng P2P (Peer-to-Peer) thể hiện tính phi tập trung của blockchain. Mỗi node trong mạng lưu trữ bản sao đầy đủ của blockchain và khi có block mới nó được broadcast đến tất cả các nodes. Các nodes xác thực block trước khi thêm vào chain của mình nhằm đảm bảo sự đồng thuận trong toàn mạng.

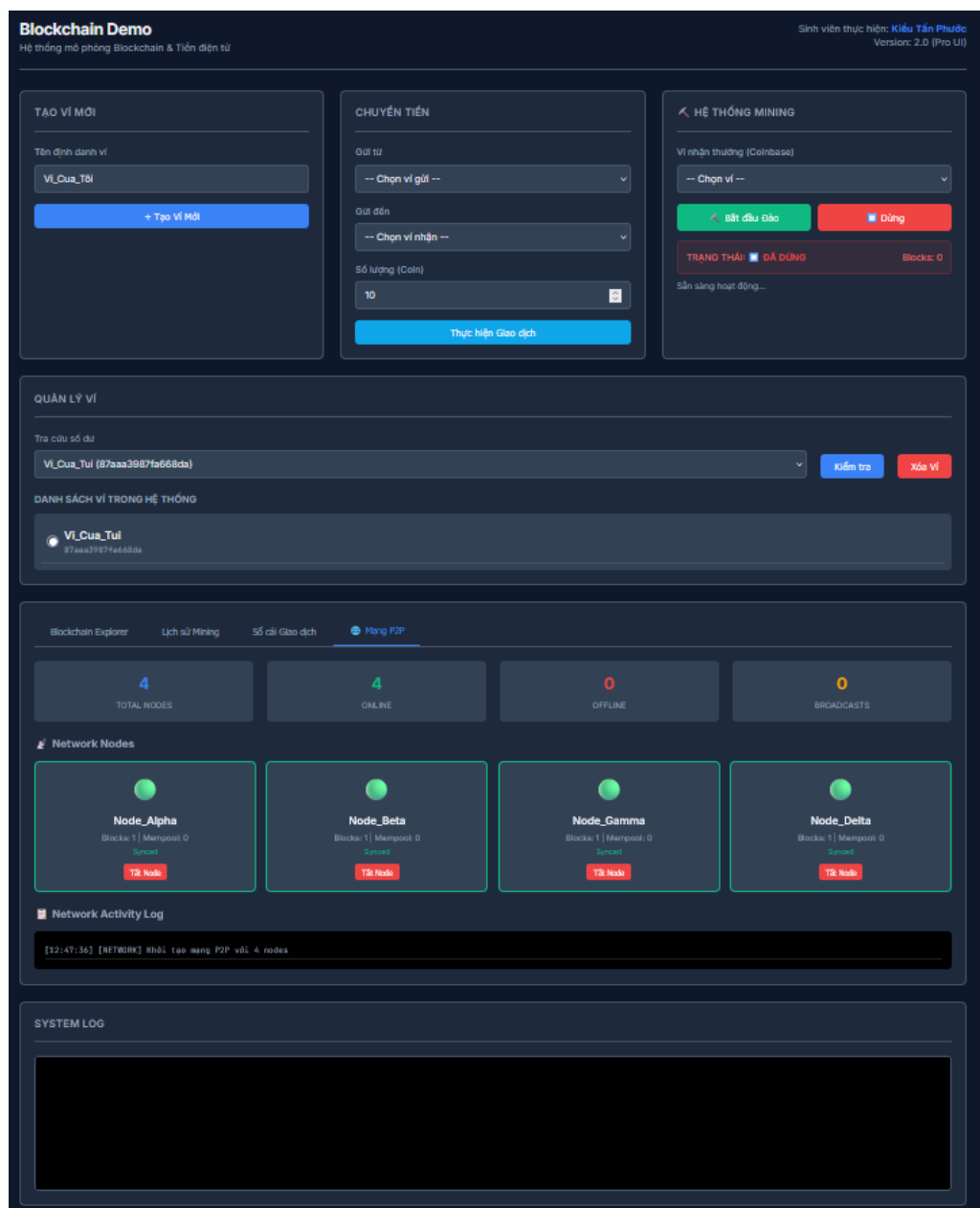
Giao diện web được xây dựng bằng Flask framework cho phép người dùng tương tác trực quan với blockchain thông qua trình duyệt. Giao diện hỗ trợ các chức năng như tạo ví mới, gửi giao dịch có chữ ký, tự động đào block, tra cứu số dư và xem trạng thái blockchain theo thời gian thực.

Về mặt ứng dụng thực tiễn, chương trình mô phỏng này giúp người học hiểu rõ cơ chế hoạt động của blockchain và tiền mã hóa. Thay vì chỉ đọc lý thuyết trừu tượng, người dùng có thể trực tiếp thao tác tạo ví, ký giao dịch, quan sát quá trình đào block và thấy được cách các block liên kết với nhau thông qua hash.

Tuy nhiên chương trình vẫn còn một số hạn chế so với blockchain thực tế. Mạng P2P chỉ mô phỏng đơn giản với vài nodes và chưa có cơ chế xử lý xung đột chain (fork). Độ khó mining cố định và chưa tự động điều chỉnh như Bitcoin. Dữ liệu chưa được lưu trữ vĩnh viễn và sẽ mất khi khởi động lại chương trình.

Tóm lại Chương 3 đã hoàn thành mục tiêu xây dựng một chương trình mô phỏng blockchain đơn giản nhưng đầy đủ các thành phần cốt lõi. Chương trình không chỉ minh họa được lý thuyết đã trình bày ở Chương 2 mà còn cung cấp công cụ thực hành giúp người học nắm vững kiến thức về blockchain và ví tiền điện tử một cách trực quan và hiệu quả.

Danh mục hình ảnh sau khi: chạy thử nghiệm và hoàn thành:



Hình 19: Giao diện Blockchain được demo bằng Flask

❖ Giải thích hình ảnh:

Hình 19 thể hiện giao diện chính của ứng dụng web mô phỏng Blockchain được xây dựng bằng Flask. Giao diện được thiết kế trực quan với các thành phần chức năng được phân chia rõ ràng.

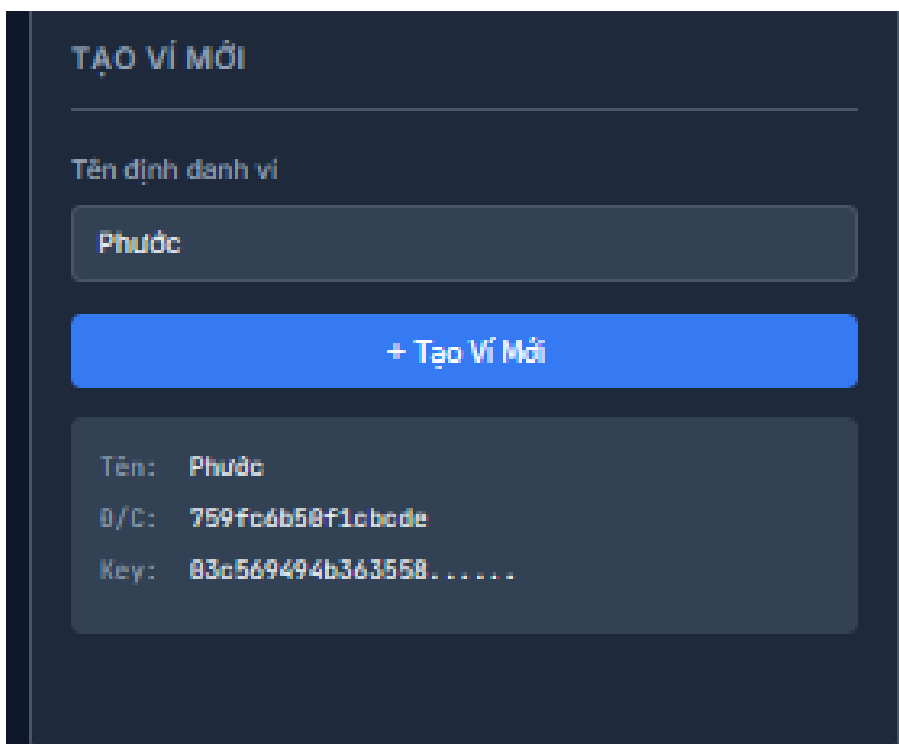
Phần trên cùng là tiêu đề "BLOCKCHAIN" với dòng mô tả ngắn gọn về ứng dụng. Bên dưới là hai khối chức năng chính được đặt cạnh nhau. Khối bên trái màu xanh dương là form "GIAO DỊCH MỚI" cho phép người dùng nhập thông tin giao dịch bao gồm người gửi, người nhận và số tiền cần chuyển. Khối bên phải màu cam là phần "TỰ ĐỘNG ĐÀO" mô phỏng quá trình mining của Bitcoin với nút bắt đầu và dừng đào.

Phía dưới là khối "TRA CỨU SỐ DƯ" màu xanh lá cho phép người dùng chọn ví và xem số dư hiện tại. Tiếp theo là các tab chuyển đổi giữa các chế độ xem gồm Blockchain, Lịch sử đào và Giao dịch.

Phần cuối hiển thị bảng thống kê blockchain với các chỉ số quan trọng. Số 1 thể hiện số lượng block hiện có trong chain. Số 0 ở cột giao dịch cho thấy chưa có giao dịch nào được thực hiện. Cột đang chờ hiển thị số giao dịch pending. Số 3 là độ khó mining hiện tại nghĩa là hash của block phải bắt đầu bằng 3 số 0.

Giao diện sử dụng màu sắc phân biệt rõ ràng giữa các chức năng giúp người dùng dễ dàng thao tác và theo dõi trạng thái của blockchain theo thời gian thực.

- **Phần kết quả chạy mô phỏng:**



TẠO VÍ MỚI

Tên định danh ví

Phước

+ Tạo Ví Mới

Tên: Phước

Đ/C: 759fc6b50f1cb0de

Key: 03c569494b363558.....

Hình 20: Kết quả tạo ví mới với địa chỉ và private key

❖ Giải thích hình ảnh:

Hình 20 thể hiện kết quả sau khi người dùng tạo ví tiền điện tử mới trong ứng dụng mô phỏng blockchain. Giao diện bao gồm một form đơn giản với ô nhập tên ví và nút "Tạo Ví Mới" màu hồng.

Sau khi nhấn nút tạo ví, hệ thống sử dụng thuật toán ECDSA với đường cong SECP256k1 để tự động sinh ra cặp khóa bất đối xứng cho người dùng. Kết quả hiển thị bao gồm ba thông tin quan trọng. Tên ví là tên do người dùng đặt để dễ nhận diện. Địa chỉ ví là 16 ký tự đầu của public key được dùng để nhận tiền và công khai cho mọi người biết. Private key là khóa bí mật được hiển thị một phần với dấu ba chấm và phải được giữ kín tuyệt đối vì đây là chìa khóa để ký giao dịch và chứng minh quyền sở hữu ví.

Đây là bước đầu tiên bắt buộc trước khi người dùng có thể thực hiện các giao dịch trong hệ thống blockchain. Việc tạo ví hoàn toàn tự động và không cần kết nối đến bất kỳ server nào, phản ánh đúng bản chất phi tập trung của blockchain.

Hình 21: Giao dịch ví tiền điện tử

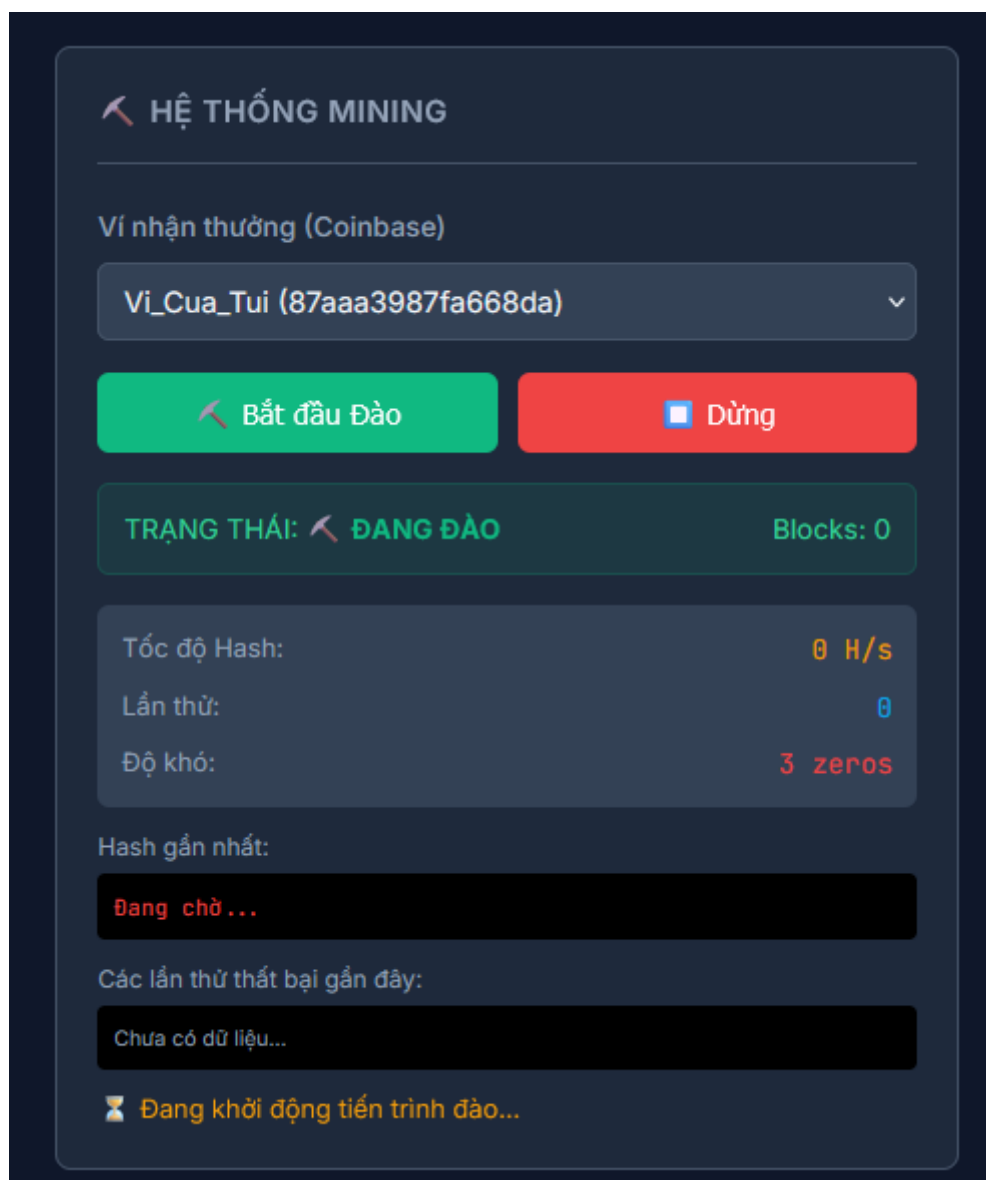
❖ Giải thích hình ảnh

Hình ảnh 21 giao diện thực hiện giao dịch chuyển tiền giữa hai ví điện tử trong hệ thống Blockchain Demo.

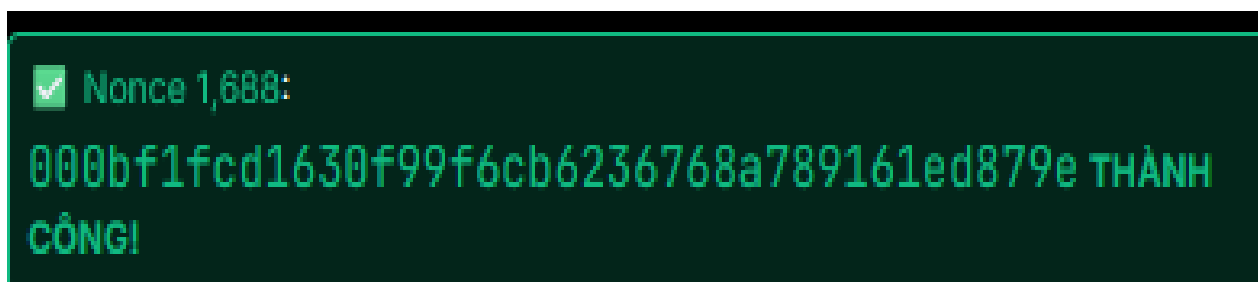
Người dùng chọn ví gửi là Alice với địa chỉ ví aacfd1fb17036146, chọn ví nhận là Ví_Của_Tôi với địa chỉ e3d8fbca7959cff3, và nhập số tiền cần chuyển là 10.02 coin.

Khi bấm nút "Ký & Gửi Giao dịch", hệ thống sẽ sử dụng khóa riêng (Private Key) của ví Alice để ký số lên giao dịch bằng thuật toán ECDSA, đảm bảo rằng chỉ chủ sở hữu ví mới có thể thực hiện giao dịch. Sau khi ký thành công, giao dịch được gửi vào Mempool (bộ nhớ chờ) và chờ được đưa vào block tiếp theo khi thợ đào thực hiện đào block.

Thông báo "Giao dịch đã ký và gửi thành công!" màu xanh lá cho biết giao dịch đã được hệ thống xác nhận và đang chờ xử



Hình 22: Chế độ đào Block được khởi động



Hình 23: Thông số kỹ thuật của Block sau khi khai thác thành công

❖ **Giải thích hình ảnh**

Hình 23 minh họa kết quả của quá trình khai thác (mining) một block thành công trong mạng blockchain. Khi một thợ đào (miner) tìm được giá trị nonce phù hợp, hệ thống sẽ tạo ra một block hợp lệ với các thông số kỹ thuật được hiển thị.

- **Phân tích các thành phần:**

- **Hash của Block**

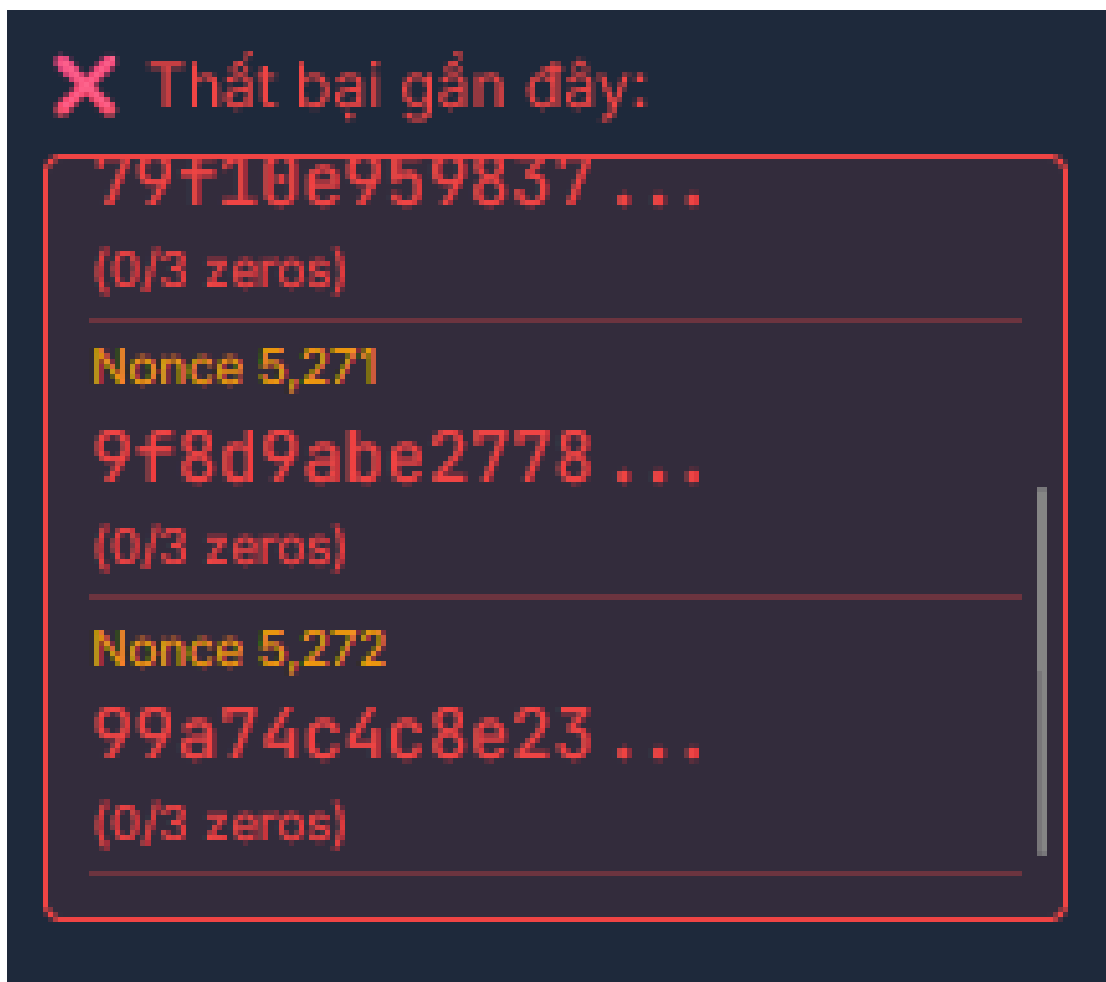
Giá trị hash được hiển thị 0000f1fc0105df9f1c0d02507d08f0101e00f9e là kết quả của việc áp dụng thuật toán băm SHA-256 lên toàn bộ dữ liệu của block. Điểm đặc biệt của hash này là nó bắt đầu bằng các số "0000" - đây chính là bằng chứng cho thấy block đã đáp ứng được yêu cầu về độ khó (difficulty) của mạng. Việc một hash bắt đầu bằng nhiều số 0 liên tiếp là cực kỳ khó tìm và đòi hỏi thợ đào phải thử hàng nghìn, thậm chí hàng triệu giá trị nonce khác nhau.

- **Thông báo "THÀNH CÔNG"**

Dòng chữ màu xanh lá cây "THÀNH CÔNG" (SUCCESS) xác nhận rằng block này đã được khai thác thành công và sẵn sàng được thêm vào chuỗi blockchain. Block này đã vượt qua tất cả các tiêu chí xác thực của mạng, bao gồm: hash hợp lệ, cấu trúc dữ liệu đúng định dạng, và liên kết chính xác với block trước đó thông qua previous hash.

- **Ý nghĩa trong cơ chế Proof of Work (PoW)**

Quá trình khai thác block thành công chứng minh rằng thợ đào đã thực hiện một lượng công việc tính toán đáng kể. Điều này đảm bảo tính bảo mật của mạng blockchain vì để giả mạo hoặc thay đổi một block, kẻ tấn công sẽ phải thực hiện lại toàn bộ công việc tính toán này, đồng thời phải nhanh hơn tất cả các thợ đào trung thực khác trong mạng.



Hình 24: Thông số kỹ thuật của block khai thác thất bại

❖ **Giải thích hình ảnh**

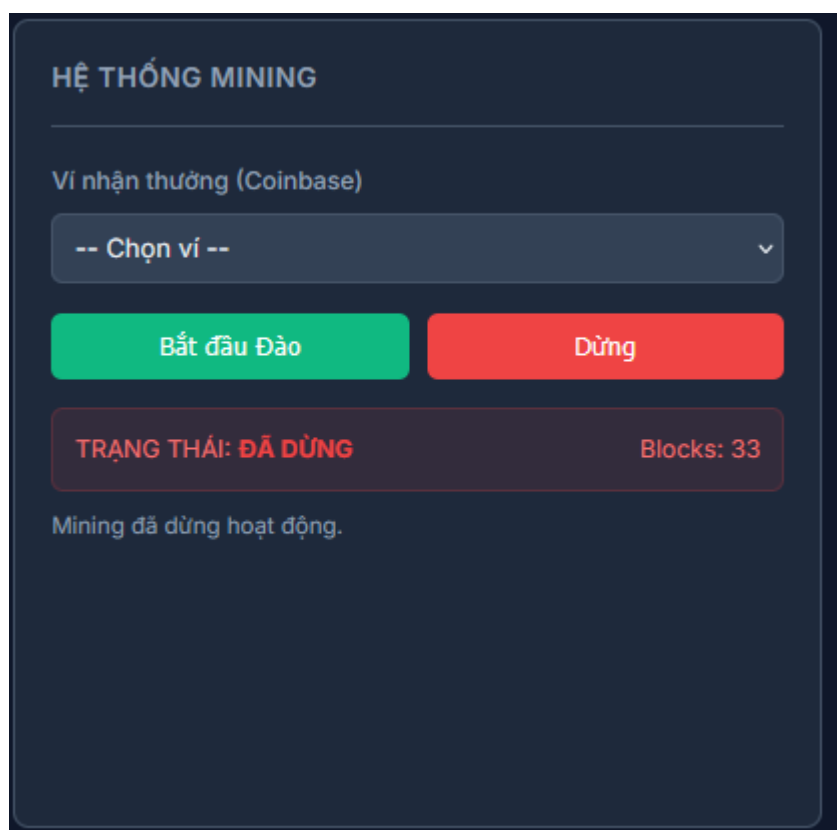
Hình 24 trình bày quá trình khai thác blockchain với các lần thử thất bại, minh họa cách thức hoạt động của thuật toán Proof of Work khi chưa tìm được hash đáp ứng yêu cầu.

- **Phân tích các thành phần:**

- **Thất bại gần đây (Recent Failed Attempts):** Giao diện hiển thị danh sách các lần thử khai thác gần nhất không thành công. Mỗi lần thử đại diện cho một nỗ lực tìm kiếm giá trị nonce phù hợp của thợ đào.
- **Giá trị Nonce:** Nonce (Number used ONCE) là một giá trị số nguyên mà thợ đào liên tục thay đổi để tạo ra các hash khác nhau. Trong hình, các giá trị nonce 5,271 và 5,272 cho thấy thợ đào đang ở lần thử thứ hơn 5,000.
- **Kết quả Hash không hợp lệ:** Các hash được tạo ra như 79f10e959837..., 9f8d9abe2778..., 99a74c4c8e23... đều không bắt đầu bằng số 0. Ký hiệu (0/3 zeros) cho biết hash không có số 0 nào ở đầu, trong khi yêu cầu độ khó là 3 số 0 liên tiếp. Do đó, tất cả các lần thử này đều thất bại.

- **Ý nghĩa trong thuật toán Proof of Work:**

Quá trình này minh họa bản chất xác suất của việc khai thác blockchain. Thợ đào phải liên tục thử các giá trị nonce khác nhau cho đến khi may mắn tìm được một giá trị tạo ra hash có đủ số 0 ở đầu theo yêu cầu. Với độ khó là 3 số 0, xác suất tìm được hash hợp lệ trong mỗi lần thử chỉ khoảng $1/4096$ (16^3). Điều này có nghĩa là trung bình thợ đào phải thử hàng nghìn lần trước khi thành công, đảm bảo rằng việc tạo block mới đòi hỏi nỗ lực tính toán thực sự và ngăn chặn việc spam hoặc giả mạo block trong mạng blockchain.



Hình 25: Chế độ tự động đào block được dừng

Giải thích hình ảnh

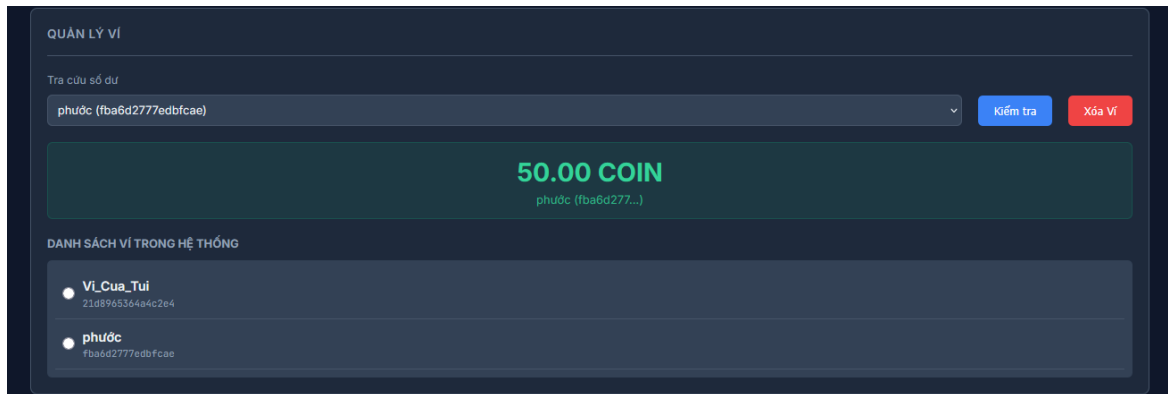
- **Mô phỏng quá trình đào Proof-of-Work**
Phản ánh tiến trình đào mô phỏng chính xác cơ chế Proof-of-Work như Bitcoin thực tế. Giao diện bao gồm các thông số chi tiết về quá trình đào.
- **Bảng thông số Mining:** hiển thị tốc độ Hash (số lần thử hash mỗi giây, ví dụ: 84,933 H/s) tổng số lần thử để tìm block hiện tại, và độ khó (số lượng số 0 ở đầu hash cần đạt được, ví dụ: 3 zeros nghĩa là hash phải bắt đầu bằng "000...").
- **Hash gần nhất hiển thị hash đang được tính toán với mã màu trực quan:** các số 0 ở đầu hiển thị màu xanh lá (đạt yêu cầu), phần còn lại hiển thị màu đỏ nếu chưa đủ số zeros theo yêu cầu.

- Khu vực hiển thị kết quả được chia thành hai cột.

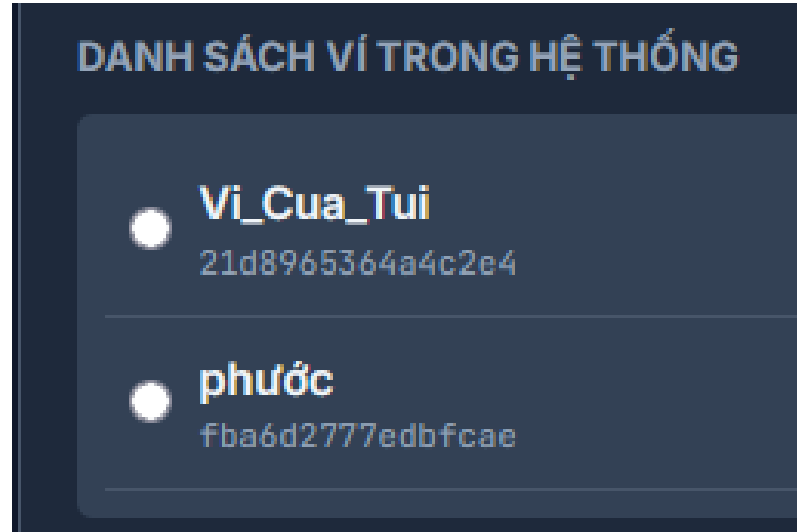
Cột bên trái "Thất bại gần đây": có viền đỏ và nền đỏ nhạt, hiển thị 8 lần thử gần nhất không đạt yêu cầu. Mỗi dòng bao gồm giá trị Nonce, phần Hash đã tính, và số zeros đạt được so với yêu cầu (ví dụ: 2/3 zeros).

Cột bên phải "Thành công": có viền xanh và nền xanh nhạt, lưu lại các block đã đào thành công. Mỗi dòng hiển thị thông báo "Block Found!", giá trị Nonce thành công, và Hash hợp lệ.

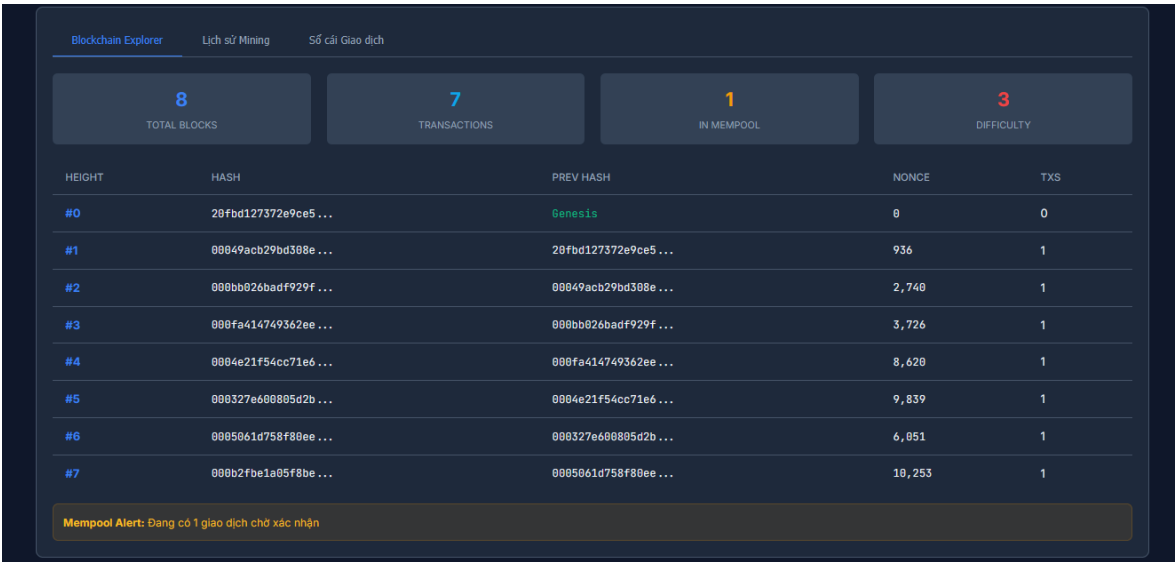
❖ **Phần còn lại của chức năng:**



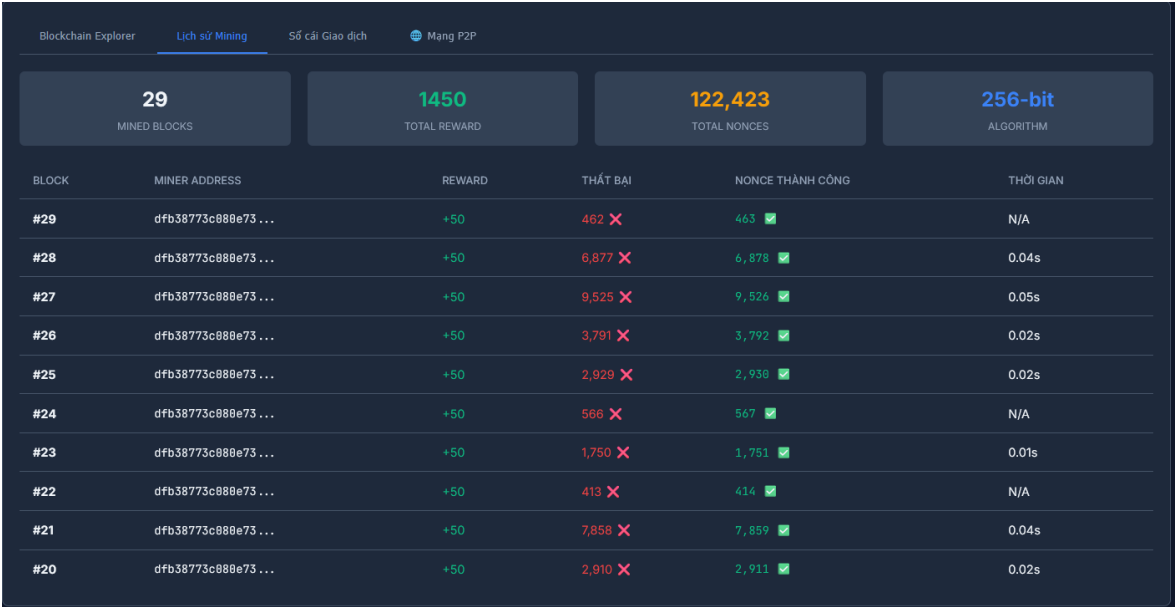
Hình 26: Tra cứu số dư của ví



Hình 27: Lưu danh sách ví điện tử của người dùng



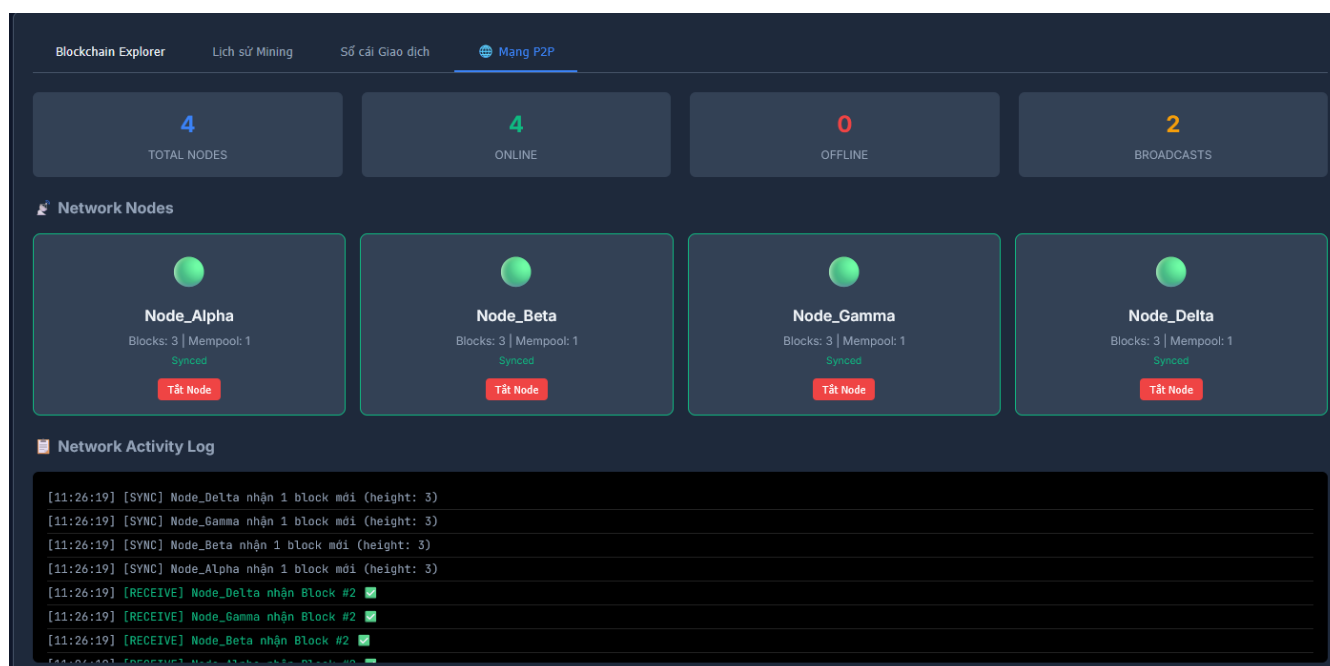
Hình 28: Giao diện quản lý danh sách các khối trong Blockchain hiện tại



Hình 28: Giao diện truy xuất lịch sử đào khối và dữ liệu thợ đào trên hệ thống.

BLOCK	SENDER	RECEIVER	AMOUNT	SIGNED?
#31	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#31	Network ...	0fb12a5c4c15 ...	50.00	None
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	0fb12a5c4c15 ...	b26e46fd3712 ...	10.08	Verified
#30	dffb38773c888 ...	0fb12a5c4c15 ...	10.08	Verified

Hình 29: Lịch sử giao dịch người dung



Hình 30: Giao diện mạng P2P và nhật ký hệ thống

Hình 30 trình bày giao diện quản lý mạng ngang hàng (Peer-to-Peer) trong ứng dụng blockchain, cho phép theo dõi và quản lý các node tham gia vào mạng lưới phân tán. Mạng P2P là kiến trúc mạng phi tập trung, trong đó mỗi máy tính (node) đóng vai trò vừa là máy khách (client) vừa là máy chủ (server), cho phép chia sẻ tài nguyên và dữ liệu trực tiếp với nhau mà không cần thông qua một máy chủ trung tâm. Trong bối cảnh blockchain, mạng P2P đảm bảo rằng tất cả các node trong mạng đều lưu trữ một bản sao đầy đủ của sổ cái (ledger), từ đó tạo nên tính bất biến và khả năng chống giả mạo của hệ thống.

Giao diện này cung cấp cho người dùng cái nhìn tổng quan về trạng thái hoạt động của từng node, các kết nối mạng hiện có, cũng như nhật ký ghi lại toàn bộ hoạt động khai thác và đồng bộ dữ liệu trong thời gian thực.

- **Phân tích các thành phần:**

- **Network Nodes (Các Node Mạng):**

Giao diện hiển thị 4 node: Node_Alpha, Node_Beta, Node_Gamma, và Node_Delta. Mỗi node được biểu diễn bằng một thẻ riêng biệt với đèn trạng thái màu xanh lá cây cho thấy đang hoạt động và kết nối bình thường. Người dùng có thể thêm mới hoặc xóa node thông qua các nút "Add" và "Delete".

- **Network Connections (Kết nối Mạng):**

Quản lý các kết nối giữa các node trong mạng. Trong kiến trúc P2P, mỗi node có thể kết nối trực tiếp với nhiều node khác mà không cần thông qua server trung tâm, tạo nên tính phi tập trung (decentralization) - đặc điểm cốt lõi của công nghệ blockchain.

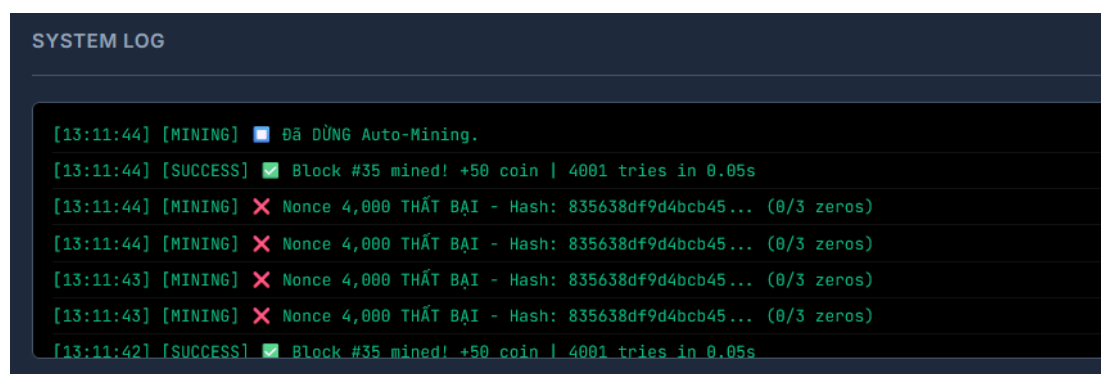
- **System Log (Nhật ký Hệ thống):**

Ghi lại toàn bộ hoạt động của mạng bao gồm thông báo khai thác thành công (Block #35 với phần thưởng 50 coin sau 4,091 lần thử trong 8.95 giây), các lần thử thất bại, và trạng thái của chế độ Auto-Mining.

Ý nghĩa của mạng P2P trong Blockchain:

Mạng ngang hàng P2P là nền tảng của công nghệ blockchain, mang lại những ưu điểm quan trọng:

- **Tính phi tập trung:** Không có điểm kiểm soát trung tâm, loại bỏ rủi ro từ single point of failure.
- **Tính minh bạch:** Mọi node đều có bản sao của blockchain và có thể xác minh giao dịch.
- **Tính chống kiểm duyệt:** Không ai có thể đơn phương chặn hoặc thay đổi giao dịch.
- **Tính đồng thuận:** Các node phối hợp để đạt được sự đồng thuận về trạng thái của blockchain thông qua cơ chế Proof of Work.



Hình 31: Hiển thị các thông báo về trạng thái ký giao dịch

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Kết quả Nghiên cứu

4.1.1 Phần kết quả

Sau quá trình nghiên cứu lý thuyết và tiến hành hiện thực hóa bằng ngôn ngữ Python, đồ án đã đạt được các kết quả thực nghiệm sau:

- **Xây dựng cấu trúc dữ liệu chuỗi khối:** Hệ thống đã triển khai thành công lớp Block với đầy đủ các thuộc tính quan trọng như index, timestamp, transactions, previous_hash và nonce. Các khối được liên kết chặt chẽ, đảm bảo rằng nếu bất kỳ dữ liệu nào trong khối bị thay đổi, mã băm hash sẽ thay đổi ngay lập tức, làm mất tính hợp lệ của toàn bộ các khối phía sau.
- **Cơ chế đào tự động (Auto-mining):** Thông qua việc sử dụng thư viện threading, hệ thống đã mô phỏng thành công tiến trình đào khối chạy ngầm. Kết quả thực nghiệm tại Hình 22 và Hình 23 cho thấy hệ thống có khả năng tìm kiếm giá trị nonce thỏa mãn độ khó thiết lập, sau đó cấp phát phần thưởng khối (Mining Reward) cho thợ đào theo đúng quy tắc của mạng Blockchain.
- **Hệ thống ví và chữ ký số:** Đồ án đã tích hợp thành công thuật toán ECDSA để tạo cặp khóa bất đối xứng. Người dùng có thể tạo ví mới, nhận địa chỉ ví công khai và thực hiện ký giao dịch bằng khóa riêng một cách an toàn. Các giao dịch này được kiểm tra tính hợp lệ trước khi đưa vào hàng đợi Mempool.

4.1.2 Phần hiệu năng

Qua các đợt thử nghiệm thực tế trên môi trường Jupyter Notebook, hiệu năng của hệ thống được đánh giá qua các chỉ số sau:

- **Độ ổn định của thuật toán băm:** việc sử dụng hàm SHA-256 giúp việc tính toán mã băm diễn ra cực nhanh (dưới 0.01 giây cho một lần băm), nhưng lại cực kỳ bảo mật vì tính chất một chiều và hiệu ứng thác đổ.
- **Kiểm soát độ khó (Mining Difficulty):** với mức độ khó được thiết lập là 3, máy tính mất trung bình từ 5 đến 15 giây để tìm ra một khối mới. Điều này chứng minh thuật toán Proof of Work hoạt động hiệu quả, tiêu tốn tài nguyên tính toán một cách có kiểm soát để ngăn chặn các cuộc tấn công spam giao dịch.
- **Tính toàn vẹn dữ liệu:** khi thử nghiệm thay đổi dữ liệu của một khối đã đào xong, hệ thống ngay lập tức phát hiện sự sai lệch mã băm và từ chối đồng bộ khối đó vào chuỗi, minh chứng cho tính bất biến của Blockchain.

4.1.3 Phần trải nghiệm

- **Tính trực quan:** Giao diện Web được xây dựng trên nền tảng Flask cung cấp cái nhìn tổng thể về mạng lưới. Tại tab "Blockchain hiện tại", người dùng có thể theo dõi sự phát triển của chuỗi theo thời gian thực, bao gồm cả các giao dịch đang nằm trong trạng thái chờ (Pending).
- **Sự tiện lợi trong thao tác:** Các chức năng như "Ký & Gửi giao dịch" hay "Tra cứu số dư" được thiết kế đơn giản, giúp người dùng không cần am hiểu sâu về kỹ thuật vẫn có thể tương tác với hệ thống. Nhật ký hoạt động (Log) hiển thị chi tiết các sự kiện diễn ra, giúp việc quản lý và xử lý lỗi trở nên dễ dàng hơn

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau thời gian nghiên cứu và thực hiện đề tài "Tìm hiểu về Blockchain cơ bản và ví tiền điện tử", em đã hoàn thành các mục tiêu đề ra và đạt được những kết luận sau:

- **Về mặt lý thuyết:** Đã làm rõ được các khái niệm cốt lõi về công nghệ sổ cái phân tán, cơ chế đồng thuận Proof of Work và vai trò của mật mã học trong việc bảo mật dữ liệu.
- **Về mặt thực hiện:** Xây dựng thành công một ứng dụng mô phỏng hoàn chỉnh bằng Python, đáp ứng đầy đủ các tính năng của một Blockchain sơ khai như: băm dữ liệu, khai thác khối, quản lý ví và thực hiện giao dịch có chữ ký số.
- **Về giá trị học thuật:** Đồ án không chỉ dừng lại ở mức độ lý thuyết mà còn cung cấp một công cụ trực quan để quan sát cách các khối dữ liệu liên kết với nhau, cách thợ đào nhận phần thưởng và cách ví điện tử bảo vệ tài sản người dùng thông qua cặp khóa công khai - riêng tư.

5.2 Hướng phát triển

Mặc dù đã đạt được những kết quả khả quan, tuy nhiên hệ thống vẫn còn một số hạn chế cần được cải thiện trong tương lai:

- **Nâng cấp cơ chế đồng thuận:** Nghiên cứu và triển khai các cơ chế hiện đại hơn như Proof of Stake (PoS) để tiết kiệm năng lượng hơn so với Proof of Work.
- **Xử lý xung đột mạng lưới:** Phát triển thuật toán giải quyết tranh chấp khi có nhiều thợ đào cùng tìm ra khối một lúc (Forking), đảm bảo tính thống nhất dựa trên quy tắc chuỗi dài nhất.
- **Lưu trữ dữ liệu bền vững:** Hiện tại dữ liệu vẫn lưu trên RAM, hướng phát triển tới sẽ tích hợp cơ sở dữ liệu như SQLite hoặc MongoDB để lưu trữ chuỗi khối vĩnh viễn.
- **Mở rộng ứng dụng:** Tích hợp các hợp đồng thông minh (Smart Contracts) đơn giản để thực hiện các giao dịch tự động có điều kiện, mở rộng khả năng ứng dụng vào các lĩnh vực như quản lý chuỗi cung ứng hoặc bầu cử điện tử.
- **Tích hợp Hợp đồng thông minh (Smart Contracts):** Cho phép thực hiện các giao dịch tự động khi thỏa mãn các điều kiện lập trình sẵn, mở rộng ứng dụng vào bầu cử điện tử hoặc quản lý chuỗi cung ứng.

- **Nâng cấp cơ chế đồng thuận:** Chuyển đổi từ Proof of Work (PoW) sang Proof of Stake (PoS) để giảm thiểu tiêu thụ năng lượng và tăng tốc độ xác nhận giao dịch.
- **Cơ sở dữ liệu bền vững:** Thay vì lưu trữ trên RAM (dữ liệu mất khi tắt chương trình), hệ thống cần tích hợp SQLite hoặc MongoDB để lưu trữ chuỗi khối vĩnh viễn.
- **Xử lý phân nhánh (Forking):** Phát triển thuật toán giải quyết tranh chấp khi nhiều thợ đào tìm ra khối cùng lúc, đảm bảo tính thống nhất dựa trên quy tắc chuỗi dài nhất.

DANH MỤC TÀI LIỆU THAM KHẢO

Sau đây là phần nguồn tài liệu tham khảo từ các trang web khác nhau:

1. <https://viblo.asia/p/xay-dung-mot-mang-blockchain-tu-dau-bang-python-07LKXJbJIV4>
2. <https://codluck.com/vi/tu-xay-dung-blockchain-don-gian-bang-python/>
3. <https://funix.edu.vn/chia-se-kien-thuc/xay-dung-lap-trinh-blockchain-voi-python/>
4. <https://codluck.com/vi/tu-xay-dung-blockchain-don-gian-bang-python/>
5. <https://funix.edu.vn/chia-se-kien-thuc/xay-dung-lap-trinh-blockchain-voi-python/>
6. <https://www.geeksforgeeks.org/python/create-simple-blockchain-using-python/>
7. <https://webissoft.com/articles/how-to-create-blockchain-in-python/>
8. <https://github.com/RyanDsilva/blockchain-simulation>
9. <https://stackoverflow.com/questions/34451214/how-to-sign-and-verify-signature-with-ecdsa-in-python>