**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CÔNG NGHỆ BLOCKCHAIN**

**ĐỀ TÀI: Trò chơi đoán số**

**Giáo viên hướng dẫn: Nguyễn Đăng Công**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1571020148 | Hoàng văn lâm | CNTT15-01 |
| 2 | 1571020003 | Đoàn Duy Nhật Anh | CNTT15-01 |

**Hà Nội, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CÔNG NGHỆ BLOCKCHAIN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1571020148 | Hoàng văn Lâm | 19/07/2003 |  |  |
| 2 | 1571020003 | Đoàn Duy Nhật Anh | 09/05/2003 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2024**

MỤC LỤC

[I. GIỚI THIỆU 7](#_Toc199838898)

[1. Lý do chọn đề tài 7](#_Toc199838899)

[2. Mục tiêu nghiên cứu 7](#_Toc199838900)

[3. Phạm vi thực hiện 8](#_Toc199838901)

[4. Ý nghĩa thực tiễn 8](#_Toc199838902)

[II. TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc199838903)

[1.1 Bối cảnh phát triển công nghệ 8](#_Toc199838904)

[1.2 Tổng quan về trò chơi trên blockchain 9](#_Toc199838905)

[1.3 Giới thiệu về MetaMask 9](#_Toc199838906)

[1.4 Ý tưởng trò chơi đoán số tích hợp blockchain 9](#_Toc199838907)

[1.5 Mục đích của nghiên cứu tổng quan 10](#_Toc199838908)

[III. NỀN TẢNG KỸ THUẬT 10](#_Toc199838909)

[2.1 Kiến trúc tổng thể của Web3 dApp 10](#_Toc199838910)

[2.2 Công nghệ blockchain Ethereum 10](#_Toc199838911)

[2.3 Ngôn ngữ Solidity 11](#_Toc199838912)

[2.4 Ví điện tử MetaMask 11](#_Toc199838913)

[2.5 Thư viện ethers.js 12](#_Toc199838914)

[2.6 Frontend ReactJS 12](#_Toc199838915)

[2.7 Công cụ phát triển 13](#_Toc199838916)

[2.8 Mạng thử nghiệm Ethereum 13](#_Toc199838917)

[IV. PHÂN TÍCH BẢO MẬT 13](#_Toc199838918)

[3.1 Tổng quan về bảo mật trong ứng dụng Web3 13](#_Toc199838919)

[3.2 Các lỗ hổng bảo mật tiềm ẩn trong trò chơi 14](#_Toc199838920)

[3.3 Bảo mật trong kết nối MetaMask 15](#_Toc199838921)

[3.4 Kiểm thử và triển khai an toàn 16](#_Toc199838922)

[3.5 Tóm tắt các nguyên tắc bảo mật chính 16](#_Toc199838923)

[V. CẤU TRÚC HỆ THỐNG 16](#_Toc199838924)

[4.1 Kiến trúc tổng thể của hệ thống 16](#_Toc199838925)

[4.2 Các thành phần chính 17](#_Toc199838926)

[4.3 Quy trình hoạt động của hệ thống 19](#_Toc199838927)

[4.4 Triển khai và môi trường hoạt động 19](#_Toc199838928)

[4.5 Ưu điểm của cấu trúc 19](#_Toc199838929)

[VI. QUY TRÌNH NGƯỜI DÙNG 20](#_Toc199838930)

[5.1 Mục tiêu của người dùng 20](#_Toc199838931)

[5.2 Các bước thao tác của người dùng 20](#_Toc199838932)

[5.3 Trải nghiệm người dùng (UX) 21](#_Toc199838933)

[5.4 Những lưu ý dành cho người chơi 22](#_Toc199838934)

[VII. ĐÁNH GIÁ 22](#_Toc199838935)

[6.1 Đánh giá về tính năng 22](#_Toc199838936)

[6.2 Đánh giá về mặt kỹ thuật 22](#_Toc199838937)

[6.3 Đánh giá về trải nghiệm người dùng (UX) 23](#_Toc199838938)

[6.4 Điểm mạnh của hệ thống 24](#_Toc199838939)

[6.5 Hạn chế 24](#_Toc199838940)

[6.6 Đề xuất cải tiến trong tương lai 24](#_Toc199838941)

[VIII. HƯỚNG PHÁT TRIỂN 25](#_Toc199838942)

[7.1 Mở rộng quy mô trò chơi 25](#_Toc199838943)

[7.2 Tích hợp phần thưởng nâng cao 25](#_Toc199838944)

[7.3 Tăng tính ngẫu nhiên và bảo mật 25](#_Toc199838945)

[7.4 Tối ưu hóa giao diện và trải nghiệm người dùng (UX/UI) 26](#_Toc199838946)

[7.5 Hỗ trợ nhiều loại ví 26](#_Toc199838947)

[7.6 Triển khai mạng chính (mainnet) 26](#_Toc199838948)

[7.7 Khả năng thương mại hóa và cộng đồng 26](#_Toc199838949)

[7.8 Tích hợp AI hỗ trợ người chơi 27](#_Toc199838950)

[7.9 Khả năng mở rộng sang các trò chơi khác 27](#_Toc199838951)

[IX. SO SÁNH VỚI HỆ THỐNG TRUYỀN THỐNG 27](#_Toc199838952)

[8.1 Tổng quan hệ thống trò chơi truyền thống 27](#_Toc199838953)

[8.2 Tiêu chí so sánh 28](#_Toc199838954)

[8.3 Ưu điểm của hệ thống blockchain 29](#_Toc199838987)

[8.4 Hạn chế so với hệ thống truyền thống 29](#_Toc199838988)

[8.5 Tình huống áp dụng phù hợp 30](#_Toc199838989)

[8.6 Kết luận so sánh 30](#_Toc199838990)

[X. KẾT LUẬN 30](#_Toc199838991)

[9.2 Những kết quả đạt được 31](#_Toc199838992)

[9.3 Hạn chế của hệ thống 31](#_Toc199838993)

[9.4 Bài học kinh nghiệm 32](#_Toc199838994)

[9.5 Định hướng phát triển tương lai 32](#_Toc199838995)

[9.6 Kết luận chung 32](#_Toc199838996)

# GIỚI THIỆU

## 1. Lý do chọn đề tài

Trong thời đại công nghệ 4.0, Blockchain đang nổi lên như một trong những xu hướng công nghệ cách mạng có khả năng định hình lại cách thức con người xây dựng, vận hành và sử dụng các hệ thống thông tin. Từ một nền tảng ra đời nhằm hỗ trợ các loại tiền điện tử như Bitcoin hay Ethereum, blockchain đã dần phát triển thành một hạ tầng công nghệ vững chắc, mở ra tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, y tế, giáo dục, logistics, và đặc biệt là lĩnh vực giải trí và trò chơi điện tử.

Song song với sự phát triển đó, mô hình Web3 - Internet phi tập trung - cũng dần hình thành, cho phép người dùng kiểm soát dữ liệu và tài sản số của họ mà không phụ thuộc vào một bên trung gian. Trong hệ sinh thái Web3, ví điện tử phi tập trung như MetaMask đóng vai trò cầu nối quan trọng giữa người dùng và blockchain, giúp người dùng tương tác trực tiếp với các ứng dụng phi tập trung (dApp).

Trong bối cảnh đó, việc xây dựng một trò chơi đơn giản nhưng có khả năng tích hợp đầy đủ các thành phần của một dApp Web3 – như smart contract, blockchain, ví MetaMask, frontend phi tập trung – là một lựa chọn phù hợp để nghiên cứu, tìm hiểu và áp dụng công nghệ mới. Trò chơi “Đoán số tích hợp blockchain” được lựa chọn bởi tính đơn giản về mặt logic, nhưng lại có khả năng thể hiện đầy đủ quy trình vận hành một ứng dụng Web3 thực tế. Qua đề tài này, người thực hiện mong muốn có thể:

* Hiểu rõ nguyên lý hoạt động của smart contract và cơ chế lưu trữ minh bạch của blockchain.
* Biết cách xây dựng giao diện Web tích hợp với ví MetaMask để thực hiện giao dịch trên Ethereum.
* Thể hiện quy trình phát triển, triển khai và thử nghiệm một ứng dụng phi tập trung hoàn chỉnh.

## 2. Mục tiêu nghiên cứu

* Xây dựng một smart contract bằng ngôn ngữ Solidity để triển khai trò chơi đoán số trên nền tảng blockchain Ethereum.
* Phát triển giao diện người dùng bằng ReactJS và kết nối với hợp đồng thông minh thông qua thư viện ethers.js.
* Tích hợp ví MetaMask để người dùng có thể chơi game bằng cách thực hiện các giao dịch blockchain.
* Đảm bảo tính minh bạch, bảo mật và trải nghiệm thực tế của người chơi trong mô hình trò chơi đơn giản nhưng trực quan.

## 3. Phạm vi thực hiện

* Dự án tập trung trên mạng thử nghiệm Ethereum (testnet như Goerli hoặc Sepolia), không triển khai trên mạng chính.
* Trò chơi chỉ bao gồm logic đoán số ngẫu nhiên đơn giản trong phạm vi từ 1 đến 10 hoặc 1 đến 100.
* Giao diện người dùng sử dụng công nghệ React và không lưu trữ dữ liệu ở backend.
* Ví được sử dụng là MetaMask với mục đích ký và gửi giao dịch.
* Không triển khai hệ thống token hay NFT trong phạm vi phiên bản đầu tiên.

## 4. Ý nghĩa thực tiễn

Đề tài này đóng vai trò như một mô hình mẫu (prototype) minh họa cho các ứng dụng phi tập trung trong thực tế. Với quy mô nhỏ, đơn giản nhưng rõ ràng, đề tài giúp sinh viên và lập trình viên mới:

* Hiểu được quy trình hoạt động của Web3.
* Làm quen với smart contract và các công cụ phát triển ứng dụng blockchain.
* Mở rộng tư duy xây dựng dApp từ trò chơi đến các lĩnh vực khác như tài chính, bỏ phiếu, truy vết chuỗi cung ứng, và hơn thế nữa

# TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Bối cảnh phát triển công nghệ

Trong thời đại công nghệ số, sự phát triển nhanh chóng của Internet đã mở đường cho hàng loạt xu hướng mới, trong đó nổi bật là sự trỗi dậy của **Web3** – thế hệ tiếp theo của Internet, tập trung vào phi tập trung, quyền sở hữu cá nhân và bảo mật dữ liệu. Một trong những nền tảng cốt lõi của Web3 chính là **blockchain** – công nghệ chuỗi khối cho phép lưu trữ dữ liệu theo cách minh bạch, không thể thay đổi và không cần bên trung gian.

Bên cạnh tài chính phi tập trung (DeFi), một trong những lĩnh vực ứng dụng blockchain sôi động nhất hiện nay là **blockchain gaming** – nơi người chơi có thể thật sự sở hữu vật phẩm, kiếm được tài sản kỹ thuật số và tương tác minh bạch thông qua smart contract.

## 1.2 Tổng quan về trò chơi trên blockchain

Khác với trò chơi truyền thống, trò chơi trên blockchain cho phép:

* Người chơi **sở hữu tài sản kỹ thuật số** (token, NFT).
* Tính **minh bạch** vì dữ liệu được ghi lại trên blockchain.
* Sự **phi tập trung**, không phụ thuộc vào máy chủ hay công ty trung gian.
* **Giao dịch tài sản trong trò chơi** diễn ra ngang hàng giữa các người chơi.

Các trò chơi như Axie Infinity, Decentraland hay The Sandbox đã chứng minh sức hấp dẫn và tiềm năng của lĩnh vực này.

## 1.3 Giới thiệu về MetaMask

MetaMask là một ví điện tử phi tập trung (non-custodial wallet) được sử dụng phổ biến trong hệ sinh thái Ethereum. Nó giúp người dùng:

* Lưu trữ, gửi và nhận token (ETH và ERC-20).
* Kết nối với các dApp (decentralized applications).
* Dễ dàng xác thực và thực hiện các giao dịch với blockchain.

Trong các trò chơi blockchain, MetaMask đóng vai trò như "tài khoản người chơi", giúp xác thực danh tính và tương tác với smart contract.

## 1.4 Ý tưởng trò chơi đoán số tích hợp blockchain

Trò chơi đoán số là một dạng mini-game phổ biến, có luật chơi đơn giản: người chơi chọn một số, nếu đoán trúng sẽ nhận được phần thưởng. Khi kết hợp với công nghệ blockchain, trò chơi này không chỉ mang tính giải trí mà còn thể hiện được cách các thành phần Web3 hoạt động:

* Smart contract quản lý logic trò chơi và phần thưởng.
* MetaMask dùng để xác thực và thanh toán.
* Frontend kết nối smart contract bằng thư viện như ethers.js.

## 1.5 Mục đích của nghiên cứu tổng quan

Việc nghiên cứu tổng quan giúp người thực hiện:

* Hiểu rõ bối cảnh và công nghệ nền tảng.
* So sánh giữa trò chơi Web2 và Web3.
* Xác định được tiềm năng, điểm mạnh và hạn chế của mô hình ứng dụng blockchain trong giải trí.

# NỀN TẢNG KỸ THUẬT

## 2.1 Kiến trúc tổng thể của Web3 dApp

Ứng dụng phi tập trung (decentralized application - dApp) có kiến trúc khác biệt so với ứng dụng Web truyền thống. Một dApp Web3 thường bao gồm 3 thành phần chính:

* **Frontend (ReactJS)**: Giao diện người dùng, chạy trên trình duyệt, tương tác trực tiếp với người chơi.
* **Smart Contract (Solidity)**: Chứa logic trò chơi, triển khai trên blockchain Ethereum.
* **Ví người dùng (MetaMask)**: Là cầu nối để xác thực người dùng và ký giao dịch gửi tới smart contract.

Dữ liệu không được lưu trữ trên server riêng, mà ghi trực tiếp lên blockchain, đảm bảo tính minh bạch và không thể sửa đổi.

## 2.2 Công nghệ blockchain Ethereum

Ethereum là nền tảng blockchain phổ biến hỗ trợ smart contract. Một số đặc điểm quan trọng:

* **Hỗ trợ Turing-complete** cho phép lập trình logic phức tạp trong smart contract.
* **Giao dịch được xác minh bởi các node mạng** và ghi vào sổ cái công khai.
* **Chi phí giao dịch (gas)** được tính bằng ETH, yêu cầu người dùng phải có ví ETH để tương tác.

Trong dự án này, Ethereum **testnet** (Goerli hoặc Sepolia) được sử dụng để triển khai và thử nghiệm, giúp tiết kiệm chi phí trong quá trình phát triển.

## 2.3 Ngôn ngữ Solidity

Solidity là ngôn ngữ lập trình chính thức dùng để viết smart contract trên Ethereum. Đặc điểm:

* Cú pháp gần giống JavaScript, C++.
* Hỗ trợ các kiểu dữ liệu cố định.
* Có cơ chế xác định quyền truy cập (public, private, internal, external).
* Cho phép triển khai logic trò chơi, xử lý thanh toán, kiểm tra dữ liệu nhập vào, và phát sự kiện (event).

Trong trò chơi đoán số, Solidity dùng để:

* Khởi tạo số bí mật (random).
* So sánh giá trị đoán của người chơi với số đúng.
* Gửi phần thưởng cho người thắng.

## 2.4 Ví điện tử MetaMask

MetaMask là tiện ích mở rộng trình duyệt phổ biến nhất cho người dùng Web3. Các tính năng chính:

* Lưu trữ khóa cá nhân người dùng một cách bảo mật.
* Kết nối tới mạng Ethereum (mainnet/testnet).
* Ký và gửi giao dịch blockchain.
* Hiển thị lịch sử giao dịch và token.

Trong trò chơi, MetaMask:

* Kết nối người dùng tới dApp.
* Ký giao dịch đoán số.
* Nhận phần thưởng ETH nếu đoán đúng.

## 2.5 Thư viện ethers.js

Đây là thư viện JavaScript giúp frontend kết nối và tương tác với smart contract:

* Kết nối với MetaMask để lấy địa chỉ ví, trạng thái mạng.
* Gửi giao dịch (transaction) đến smart contract.
* Đọc dữ liệu từ blockchain.
* Lắng nghe sự kiện do smart contract phát ra.

Ethers.js được lựa chọn vì:

* Nhẹ hơn web3.js.
* Cấu trúc rõ ràng, dễ dùng.
* Hỗ trợ async/await hiện đại.

## 2.6 Frontend ReactJS

React là thư viện JavaScript xây dựng giao diện người dùng, được lựa chọn nhờ:

* Tốc độ render nhanh với Virtual DOM.
* Quản lý trạng thái dễ dàng bằng hook (useState, useEffect).
* Dễ dàng tích hợp thư viện ethers.js để tương tác blockchain.

Các thành phần React trong trò chơi gồm:

* Khung nhập số đoán.
* Nút “Submit” gửi giao dịch.
* Thông báo kết quả (đúng/sai).
* Hiển thị địa chỉ ví và số dư.

## 2.7 Công cụ phát triển

|  |  |
| --- | --- |
| **Công cụ** | **Vai trò** |
| Visual Studio Code | Môi trường lập trình |
| Hardhat | Triển khai, kiểm thử, biên dịch contract |
| MetaMask | Kết nối và ký giao dịch |
| Remix IDE | Viết và kiểm thử hợp đồng nhanh chóng |
| Node.js + npm | Quản lý gói và môi trường frontend |
| Vite + React | Khởi tạo và phát triển giao diện Web |

## 2.8 Mạng thử nghiệm Ethereum

Mạng thử nghiệm như **Goerli** hoặc **Sepolia** được dùng để kiểm thử smart contract trước khi triển khai thực tế. Ưu điểm:

* Giao dịch miễn phí bằng ETH testnet.
* Môi trường gần giống mainnet.
* Giúp kiểm thử logic contract an toàn.

# PHÂN TÍCH BẢO MẬT

## 3.1 Tổng quan về bảo mật trong ứng dụng Web3

Ứng dụng Web3 vận hành trên blockchain có ưu điểm là **phi tập trung** và **minh bạch**, nhưng lại dễ trở thành mục tiêu của các cuộc tấn công nếu không được thiết kế cẩn thận. Không giống ứng dụng truyền thống có thể sửa lỗi bằng cập nhật backend, một khi **smart contract được triển khai lên blockchain**, mọi logic gần như không thể thay đổi, nên nếu có lỗi bảo mật, hậu quả sẽ rất nghiêm trọng.

Do đó, việc **phân tích và đánh giá bảo mật** là vô cùng quan trọng trong quá trình phát triển dApp.

## 3.2 Các lỗ hổng bảo mật tiềm ẩn trong trò chơi

**3.2.1 Lộ số ngẫu nhiên (Randomness vulnerability)**

Một trong những điểm yếu phổ biến nhất trong các trò chơi blockchain là sử dụng hàm sinh số ngẫu nhiên không đủ an toàn. Ví dụ:

uint256 randomNumber = uint256(keccak256(abi.encodePacked(block.timestamp, msg.sender))) % 10;

Hàm trên có thể bị đoán hoặc khai thác vì:

* block.timestamp và msg.sender là thông tin **công khai**.
* Những người khai thác (miner/validator) có thể **dự đoán hoặc thao túng** kết quả.

**Giải pháp:**

* Sử dụng dịch vụ tạo số ngẫu nhiên có thể xác minh (Verifiable Randomness) như **Chainlink VRF**.
* Hoặc tách quy trình chơi và quy trình trả thưởng thành **hai bước** để giảm khả năng lộ kết quả.

**3.2.2 Tái nhập (Reentrancy attack)**

Reentrancy là lỗi mà hacker có thể gọi lại hợp đồng thông minh trước khi hoàn thành giao dịch ban đầu, từ đó **rút tiền nhiều lần**. Đây là lỗi đã gây ra vụ hack DAO nổi tiếng trên Ethereum.

**Giải pháp:**

* Sử dụng **mẫu Checks-Effects-Interactions** để cập nhật trạng thái **trước khi** gọi external contract.
* Dùng từ khóa reentrancyGuard trong Solidity (OpenZeppelin library).

**3.2.3 Xác thực quyền hạn người dùng (Access control)**

Nếu smart contract không kiểm tra đúng quyền của người dùng (ví dụ: ai được đặt lại số may mắn), thì có thể bị người ngoài truy cập và làm thay đổi luật chơi.

**Giải pháp:**

* Sử dụng từ khóa onlyOwner, require(msg.sender == owner) để giới hạn quyền truy cập.
* Kế thừa từ thư viện Ownable.sol để quản lý quyền hiệu quả.

**3.2.4 Phí gas bất thường (Gas griefing / denial-of-service)**

Một kẻ tấn công có thể cố tình gửi nhiều giao dịch đoán sai để:

* Làm tăng chi phí gas.
* Làm nghẽn mạng và gây **từ chối dịch vụ (DoS)** cho người dùng thật.

**Giải pháp:**

* Giới hạn số lần đoán mỗi người trong một khoảng thời gian.
* Dùng cơ chế cooldown hoặc stake ETH để tránh spam.

**3.2.5 Giao diện giả mạo (Phishing frontend)**

Do frontend là mã nguồn mở, kẻ xấu có thể sao chép giao diện, thay đổi địa chỉ contract hoặc ví nhận tiền để lừa người chơi.

**Giải pháp:**

* Công bố địa chỉ contract **chính thức** trên trang chủ và mạng xã hội.
* Cảnh báo người dùng kiểm tra kỹ domain, contract và ví trước khi chơi.

## 3.3 Bảo mật trong kết nối MetaMask

* MetaMask sẽ hiển thị cảnh báo rõ ràng khi một trang web yêu cầu kết nối ví.
* Người dùng nên **xác minh contract** và thông tin giao dịch trước khi nhấn “Xác nhận”.
* Luôn yêu cầu người dùng xác thực quyền truy cập (eth\_requestAccounts) đúng cách.
* Không lưu trữ khóa riêng hay dữ liệu nhạy cảm trong trình duyệt.

## 3.4 Kiểm thử và triển khai an toàn

* Tất cả smart contract nên được **kiểm thử tự động** với Hardhat, chai/mocha hoặc Foundry.
* Có thể dùng dịch vụ **kiểm toán smart contract** như OpenZeppelin hoặc Certik nếu triển khai sản phẩm thực tế.
* Khuyến nghị **triển khai thử nghiệm trên testnet** trong thời gian dài trước khi đưa lên mainnet.

## 3.5 Tóm tắt các nguyên tắc bảo mật chính

|  |  |
| --- | --- |
| **Nguy cơ** | **Biện pháp phòng ngừa** |
| Số ngẫu nhiên bị đoán | Dùng Chainlink VRF hoặc sinh số an toàn |
| Reentrancy | Mẫu Checks-Effects-Interactions, Guard |
| Truy cập trái phép | Sử dụng onlyOwner hoặc phân quyền rõ ràng |
| Tấn công spam hoặc DoS | Giới hạn số lần chơi, cooldown |
| Giao diện giả mạo | Xác minh domain, contract công khai |

# CẤU TRÚC HỆ THỐNG

## 4.1 Kiến trúc tổng thể của hệ thống

Hệ thống trò chơi đoán số tích hợp blockchain là một ứng dụng phi tập trung (decentralized application – dApp), gồm ba thành phần chính:

* **Frontend (ReactJS)**: Giao diện người dùng chạy trên trình duyệt, cho phép người chơi nhập số, kết nối ví và tương tác với smart contract.
* **Smart Contract (Solidity)**: Chạy trên mạng Ethereum, lưu trữ logic trò chơi và xử lý các giao dịch.
* **Ví điện tử MetaMask**: Là cầu nối giữa người dùng và blockchain, giúp ký giao dịch và gửi yêu cầu đến smart contract.

Dưới đây là sơ đồ mô tả kiến trúc hệ thống:

[Người chơi]

⇅

[Trình duyệt Web - ReactJS frontend]

⇅ ethers.js

[MetaMask - Ví Ethereum]

⇅

[Smart Contract - Blockchain Ethereum]

## 4.2 Các thành phần chính

**4.2.1 Frontend (Giao diện người dùng)**

Được phát triển bằng ReactJS + Vite, phần frontend chịu trách nhiệm:

* Hiển thị giao diện trò chơi cho người dùng.
* Kết nối với MetaMask để lấy địa chỉ ví, kiểm tra mạng.
* Gửi số đoán của người dùng lên smart contract.
* Hiển thị kết quả đoán (đúng/sai), thông báo phần thưởng, số dư ví.

Các thành phần chính trong frontend:

* App.jsx: Giao diện chính.
* GuessForm.jsx: Form nhập số đoán và xử lý sự kiện.
* wallet.js: Kết nối với MetaMask và blockchain.
* contract.js: Tương tác với smart contract thông qua ethers.js.

**4.2.2 Smart Contract (Hợp đồng thông minh)**

Smart contract được viết bằng Solidity, triển khai trên Ethereum testnet. Nó chịu trách nhiệm:

* Lưu trữ số bí mật cần đoán (hoặc mã hóa).
* So sánh số người chơi nhập vào với số cần đoán.
* Phân phối phần thưởng cho người chơi đoán đúng.
* Ghi nhận lịch sử lượt chơi hoặc phát sự kiện.

Ví dụ chức năng chính:

* guessNumber(uint number): So sánh số đoán và trả kết quả.
* rewardWinner(): Gửi phần thưởng nếu đoán đúng.

**4.2.3 MetaMask (Ví người dùng)**

MetaMask là thành phần trung gian giúp người chơi:

* Kết nối với Ethereum testnet (Goerli hoặc Sepolia).
* Ký giao dịch gửi đến smart contract.
* Quản lý tài sản và nhận phần thưởng (ETH testnet).

MetaMask không lưu dữ liệu trò chơi, nhưng là **chìa khóa để xác thực danh tính** người chơi và kích hoạt các hành động trên blockchain.

**4.2.4 Thư viện ethers.js**

Đây là thư viện JavaScript giúp kết nối frontend với blockchain:

* Tạo provider để đọc dữ liệu từ blockchain.
* Tạo signer để gửi giao dịch đã ký.
* Tương tác với smart contract bằng ABI.

Cấu trúc tương tác thông qua ethers.js như sau:

const provider = new ethers.providers.Web3Provider(window.ethereum);

const signer = provider.getSigner();

const contract = new ethers.Contract(contractAddress, abi, signer);

## 4.3 Quy trình hoạt động của hệ thống

1. **Người dùng truy cập website.**
2. **Frontend yêu cầu kết nối ví MetaMask.**
3. **MetaMask yêu cầu người dùng chấp thuận kết nối.**
4. **Người dùng nhập số đoán vào form.**
5. **Số đoán được gửi dưới dạng giao dịch đến smart contract.**
6. **Smart contract xử lý logic, so sánh số và phát phần thưởng nếu đúng.**
7. **Frontend hiển thị kết quả (thắng/thua).**

## 4.4 Triển khai và môi trường hoạt động

* **Frontend** được chạy trên môi trường localhost (npm run dev) hoặc có thể triển khai lên nền tảng như Vercel/Netlify.
* **Smart contract** được triển khai lên Ethereum testnet bằng Hardhat.
* **Ví người dùng** sử dụng MetaMask đã cấu hình mạng testnet và nạp ETH thử nghiệm từ faucet.

## 4.5 Ưu điểm của cấu trúc

* **Phi tập trung**: Không phụ thuộc vào máy chủ riêng, mọi dữ liệu lưu trên blockchain.
* **Minh bạch**: Người chơi có thể kiểm tra code hợp đồng và lịch sử giao dịch.
* **Bảo mật**: Dữ liệu và logic không thể thay đổi sau khi triển khai contract.
* **Tương tác dễ dàng**: MetaMask là ví phổ biến, giúp người dùng dễ tiếp cận Web3.

# QUY TRÌNH NGƯỜI DÙNG

## 5.1 Mục tiêu của người dùng

Người dùng truy cập ứng dụng với mục tiêu:

* Kết nối ví cá nhân (MetaMask).
* Nhập một số để tham gia trò chơi dự đoán.
* Gửi giao dịch đến blockchain để kiểm tra kết quả.
* Nhận thông báo thắng/thua và nhận thưởng nếu đoán đúng.

## 5.2 Các bước thao tác của người dùng

**Bước 1: Truy cập website trò chơi**

Người dùng mở trình duyệt và truy cập đường dẫn frontend (ví dụ: <http://localhost:5173> nếu chạy cục bộ hoặc đường dẫn deploy thực tế). Trang chủ sẽ hiển thị giao diện trò chơi đơn giản.

**Bước 2: Kết nối ví MetaMask**

* Người dùng nhấn vào nút “Kết nối ví”.
* Trình duyệt sẽ mở cửa sổ MetaMask yêu cầu người dùng cho phép truy cập ví.
* Sau khi chấp thuận, địa chỉ ví Ethereum sẽ được hiển thị trên giao diện.

Nếu người dùng chưa cài MetaMask, hệ thống có thể hiển thị thông báo hướng dẫn cài đặt tiện ích này.

**Bước 3: Nhập số cần đoán**

* Sau khi kết nối thành công, người dùng sẽ thấy một ô nhập số từ 0 đến 9 (hoặc giới hạn tùy theo cấu hình).
* Người dùng nhập vào số họ đoán là số “bí mật” mà hệ thống đã tạo trước đó trên smart contract.

**Bước 4: Gửi giao dịch**

* Khi nhấn nút “Gửi dự đoán”, frontend sẽ gọi smart contract thông qua ethers.js.
* MetaMask hiện cửa sổ xác nhận giao dịch: người dùng xác nhận gửi giao dịch và trả một khoản phí gas (ETH testnet).
* Giao dịch được gửi lên Ethereum testnet.

**Bước 5: Nhận kết quả**

* Sau khi giao dịch được xác nhận trên blockchain (khoảng vài giây đến 1 phút tùy mạng), hệ thống sẽ:
  + Hiển thị thông báo kết quả là “Đoán đúng” hoặc “Đoán sai”.
  + Nếu đoán đúng, người dùng sẽ nhận một phần thưởng là một lượng ETH testnet được gửi từ hợp đồng đến ví.
  + Nếu đoán sai, không có phần thưởng, nhưng có thể chơi lại.

**Bước 6: Tiếp tục chơi hoặc rời khỏi**

* Người dùng có thể tiếp tục đoán số khác hoặc ngắt kết nối ví.
* Dữ liệu trò chơi (kết quả, phần thưởng) đều được ghi nhận vĩnh viễn trên blockchain và có thể truy xuất qua trình khám phá block như Etherscan.

## 5.3 Trải nghiệm người dùng (UX)

Ứng dụng được thiết kế đơn giản, dễ sử dụng:

* Giao diện trực quan, thao tác chỉ qua vài nút bấm.
* Không yêu cầu đăng ký tài khoản.
* Việc sử dụng MetaMask giúp xác thực an toàn và hạn chế gian lận.
* Giao dịch được hiển thị công khai nên người dùng có thể tin tưởng vào tính minh bạch của trò chơi.

## 5.4 Những lưu ý dành cho người chơi

* Đảm bảo đã cài đặt MetaMask và nạp sẵn ETH testnet từ faucet trước khi chơi.
* Kiểm tra kỹ mạng testnet đang sử dụng (ví dụ: Sepolia, Goerli).
* Giao dịch sẽ tiêu tốn một lượng nhỏ ETH testnet dưới dạng phí gas.
* Người chơi cần kiên nhẫn chờ xác nhận từ blockchain sau khi gửi giao dịch.

# ĐÁNH GIÁ

## 6.1 Đánh giá về tính năng

Hệ thống trò chơi đoán số được đánh giá là hoàn chỉnh về mặt chức năng cơ bản của một ứng dụng phi tập trung (dApp). Các tính năng đã được xây dựng và kiểm thử bao gồm:

* **Kết nối ví MetaMask thành công**.
* **Gửi số dự đoán đến smart contract trên blockchain**.
* **Nhận phản hồi từ hợp đồng thông minh (đúng/sai)**.
* **Phát phần thưởng nếu đoán đúng**.
* **Hiển thị kết quả và quản lý trải nghiệm người dùng mượt mà**.

Những chức năng cốt lõi này đều hoạt động ổn định trong môi trường testnet (Goerli hoặc Sepolia) và đáp ứng đúng yêu cầu ban đầu của đề tài.

## 6.2 Đánh giá về mặt kỹ thuật

**a. Tính phi tập trung**

Ứng dụng sử dụng blockchain để xử lý logic cốt lõi (so sánh số đoán và phát thưởng), đảm bảo tính phi tập trung và minh bạch. Không có server trung gian, dữ liệu và kết quả đều được ghi trên blockchain, không thể bị chỉnh sửa.

**b. Bảo mật**

* Giao dịch được xác thực qua ví MetaMask, đảm bảo mỗi hành động đều do người dùng thực hiện thật sự.
* Smart contract không lưu trữ dữ liệu nhạy cảm.
* Dữ liệu không thể bị thay đổi sau khi ghi vào blockchain.

Tuy nhiên, nếu số cần đoán được khai báo tĩnh trong hợp đồng (hardcoded), người dùng có thể đọc được từ Etherscan => đây là một điểm có thể bị khai thác nếu không được mã hóa hoặc sinh ngẫu nhiên phù hợp.

**c. Mức tiêu thụ tài nguyên**

Hệ thống tiêu thụ tài nguyên thấp, chỉ cần trình duyệt và ví MetaMask. Tuy nhiên, vì dựa vào blockchain nên vẫn bị ảnh hưởng bởi:

* Thời gian xác nhận giao dịch (từ vài giây đến 1 phút).
* Phí gas dù nhỏ nhưng vẫn có thể khiến trải nghiệm kém hơn nếu mạng nghẽn.

## 6.3 Đánh giá về trải nghiệm người dùng (UX)

* Giao diện đơn giản, dễ hiểu, phù hợp với người chưa từng dùng blockchain.
* Việc kết nối MetaMask thuận tiện, các thông báo rõ ràng.
* Trò chơi dễ chơi, gần gũi, tạo sự hấp dẫn cho người dùng mới làm quen Web3.

Tuy nhiên, vì đây là bản thử nghiệm nên:

* Chưa có bảng xếp hạng, thống kê hoặc lịch sử chơi => thiếu tính cạnh tranh.
* Giao diện chưa tối ưu cho di động.
* Không có hướng dẫn tích hợp bên trong game => người dùng mới có thể gặp khó khăn nếu không quen dùng ví.

## 6.4 Điểm mạnh của hệ thống

* Ứng dụng công nghệ blockchain thực tế trong một trò chơi đơn giản.
* Giao diện ReactJS hiện đại, thân thiện.
* Kết nối MetaMask mượt mà, đáng tin cậy.
* Logic minh bạch, có thể kiểm tra trực tiếp qua Etherscan.
* Có tiềm năng mở rộng: phần thưởng thực, NFT, bảng điểm, staking,…

## 6.5 Hạn chế

* Số cần đoán chưa được mã hóa hoặc tạo ngẫu nhiên phức tạp.
* Giao diện đơn giản, chưa tối ưu UX/UI hoàn toàn.
* Phụ thuộc vào MetaMask, chưa hỗ trợ nhiều ví (WalletConnect, TrustWallet…).
* Hệ thống mới chỉ chạy trên testnet, chưa triển khai mạng chính (mainnet).

## 6.6 Đề xuất cải tiến trong tương lai

* Áp dụng kỹ thuật **commit-reveal** hoặc **zero-knowledge proof** để tăng tính ngẫu nhiên và bảo mật cho số cần đoán.
* Tích hợp các loại ví phổ biến khác ngoài MetaMask.
* Bổ sung **bảng xếp hạng**, **thống kê lượt chơi**, và hệ thống phần thưởng đa dạng (NFT, điểm tích lũy,…).
* Triển khai trên mạng chính (Ethereum mainnet hoặc Layer 2 như Arbitrum, Polygon).
* Cải thiện giao diện UI/UX, hỗ trợ đa nền tảng (web + mobile).

# HƯỚNG PHÁT TRIỂN

## 7.1 Mở rộng quy mô trò chơi

Hiện tại trò chơi chỉ đơn giản là đoán một con số trong phạm vi từ 0 đến 9. Trong tương lai, để tăng tính hấp dẫn và tính cạnh tranh, có thể mở rộng quy mô như:

* Tăng phạm vi đoán (ví dụ từ 0–99 hoặc 0–999).
* Thiết kế các cấp độ khó với phần thưởng tương ứng.
* Thêm chế độ “chơi nhiều người” – người nào đoán đúng nhanh nhất sẽ nhận toàn bộ phần thưởng.

## 7.2 Tích hợp phần thưởng nâng cao

Thay vì chỉ trả ETH làm phần thưởng, hệ thống có thể:

* Phát hành token ERC-20 riêng cho trò chơi để sử dụng làm phần thưởng.
* Tặng **NFT** độc quyền cho người thắng nhiều lần, có thể giao dịch hoặc lưu giữ như kỷ niệm.
* Áp dụng cơ chế “staking”: người dùng đặt cược một lượng token để chơi, nếu đoán đúng sẽ nhận nhiều hơn.

## 7.3 Tăng tính ngẫu nhiên và bảo mật

Hiện tại số cần đoán được đặt sẵn trong smart contract (hardcoded) hoặc đặt bằng tay. Trong tương lai có thể:

* Tích hợp **Chainlink VRF (Verifiable Random Function)** để tạo số ngẫu nhiên công khai, minh bạch và không thể đoán trước.
* Áp dụng **kỹ thuật commit-reveal**: người chơi gửi hash của số đoán và sau đó mới reveal để tránh lộ thông tin trước khi kiểm tra.
* Sử dụng **Zero-Knowledge Proofs** để xác minh dự đoán đúng mà không lộ dữ liệu.

## 7.4 Tối ưu hóa giao diện và trải nghiệm người dùng (UX/UI)

* Cải tiến giao diện với đồ họa bắt mắt hơn, thân thiện với người dùng mới (onboarding MetaMask, hướng dẫn).
* Hỗ trợ nền tảng **mobile** và **tablet**.
* Hiển thị **lịch sử chơi, thống kê thắng thua**, điểm tích lũy cá nhân.

## 7.5 Hỗ trợ nhiều loại ví

Hiện tại chỉ hỗ trợ MetaMask. Trong tương lai, có thể mở rộng ra:

* WalletConnect: để kết nối ví Trust Wallet, Rainbow, Coinbase Wallet,...
* Tích hợp hệ thống xác thực người chơi thông qua email, Google hoặc mạng xã hội (kết hợp Web2 & Web3).

## 7.6 Triển khai mạng chính (mainnet)

Sau khi thử nghiệm kỹ lưỡng, có thể triển khai hệ thống lên mạng chính Ethereum hoặc các giải pháp Layer 2 như:

* **Polygon**: phí rẻ, tốc độ nhanh, tương thích EVM.
* **Arbitrum/Optimism**: giữ được bảo mật từ Ethereum nhưng chi phí thấp hơn.
* **Base**: Layer 2 của Coinbase, thân thiện với người mới.

Việc triển khai mainnet sẽ cho phép trò chơi có phần thưởng thực tế, hấp dẫn cộng đồng người dùng Web3 lớn hơn.

## 7.7 Khả năng thương mại hóa và cộng đồng

* Tổ chức các giải đấu (tournament) đoán số theo mùa.
* Kêu gọi tài trợ từ các dự án blockchain để tạo giải thưởng thật.
* Phát triển cộng đồng Discord/Telegram để người chơi giao lưu, trao đổi chiến thuật, góp ý sản phẩm.

## 7.8 Tích hợp AI hỗ trợ người chơi

Trong tương lai xa hơn, có thể tích hợp AI để:

* Gợi ý số đoán dựa trên thống kê.
* Chatbot giải thích luật chơi, hướng dẫn dùng MetaMask.
* Hệ thống chống gian lận sử dụng machine learning để phát hiện bất thường.

## 7.9 Khả năng mở rộng sang các trò chơi khác

Mô hình hiện tại là cơ bản và có thể được mở rộng thành nền tảng mini-game blockchain:

* Trò chơi may rủi khác: tung đồng xu, rút thăm, quay số.
* Các trò chơi chiến lược: đấu giá, đoán xu hướng giá token,...
* Xây dựng hẳn một **sàn trò chơi Web3** tích hợp nhiều mini-game dùng chung token nội bộ.

# SO SÁNH VỚI HỆ THỐNG TRUYỀN THỐNG

## 8.1 Tổng quan hệ thống trò chơi truyền thống

Hệ thống trò chơi truyền thống (Web2) thường hoạt động trên nền tảng tập trung. Một server (máy chủ) trung tâm sẽ xử lý toàn bộ logic trò chơi, xác định kết quả và lưu trữ thông tin người dùng. Các ví dụ phổ biến bao gồm:

* Trò chơi dự đoán trúng thưởng qua website hoặc ứng dụng.
* Trò chơi nhỏ (mini-game) tích hợp trong các app di động.
* Các nền tảng chơi game “may rủi” có hệ thống phần thưởng nội bộ.

Mặc dù những trò chơi này hoạt động nhanh và phổ biến, nhưng chúng tồn tại nhiều điểm hạn chế về minh bạch, bảo mật và quyền sở hữu dữ liệu người dùng.

## 8.2 Tiêu chí so sánh

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Hệ thống truyền thống (Web2) | Hệ thống blockchain tích hợp ví MetaMask |
| Minh bạch | Thấp – người dùng không thể kiểm chứng kết quả | Cao – kết quả được ghi công khai trên blockchain |
| Bảo mật | Phụ thuộc vào server, dễ bị tấn công | Bảo mật cao, không có điểm trung tâm tấn công |
| Quyền sở hữu dữ liệu | Do nhà phát triển kiểm soát | Dữ liệu thuộc về người dùng (trên ví của họ) |
| Tính tin cậy | Dễ thay đổi kết quả nếu server bị thao túng | Kết quả không thể chỉnh sửa sau khi ghi nhận |
| Tính phi tập trung | Không có – phụ thuộc hoàn toàn vào một bên | Có – hợp đồng thông minh tự động xử lý và lưu trữ |
| Chi phí vận hành | Rẻ hơn vì không tốn phí gas | Có chi phí gas cho mỗi giao dịch trên blockchain |
| Tốc độ phản hồi | Nhanh vì xử lý cục bộ trên server | Chậm hơn do cần xác nhận giao dịch trên blockchain |
| Tính mở rộng | Cao nhưng khó tích hợp công nghệ mới | Tốt – dễ tích hợp DeFi, NFT, staking,... |
| Tính minh chứng/kiểm tra lại | Không thể hoặc rất khó | Dễ dàng – có thể kiểm tra mọi giao dịch trên Etherscan |
| Phần thưởng thực tế | Có thể bị hạn chế hoặc không có giá trị | Có thể là token, NFT – giao dịch được trên toàn cầu |

## 8.3 Ưu điểm của hệ thống blockchain

* **Minh bạch tuyệt đối**: Mỗi giao dịch, mỗi lần đoán đều được ghi nhận công khai, giúp người chơi tin tưởng hệ thống.
* **Không thể gian lận**: Kết quả được xử lý bởi hợp đồng thông minh, không ai có thể can thiệp chỉnh sửa.
* **Tương thích tài chính phi tập trung (DeFi)**: Có thể kết hợp với staking, airdrop, NFT, lending...
* **Tài sản thật sự thuộc về người chơi**: Token hoặc phần thưởng được lưu trữ trong ví cá nhân, không thể bị thu hồi hoặc thay đổi.

## 8.4 Hạn chế so với hệ thống truyền thống

* **Phụ thuộc vào ví**: Người chơi phải cài MetaMask hoặc ví tương thích mới có thể tham gia.
* **Phí giao dịch (gas fee)**: Mỗi hành động đều phải trả một khoản phí nhỏ (dù trên testnet thì phí vẫn tồn tại).
* **Trải nghiệm ban đầu phức tạp**: Người mới có thể gặp khó khăn trong việc hiểu blockchain, kết nối ví,...
* **Tốc độ xử lý chậm hơn**: So với game truyền thống xử lý trong milliseconds, blockchain cần vài giây hoặc lâu hơn.

## 8.5 Tình huống áp dụng phù hợp

* **Trò chơi có yếu tố phần thưởng giá trị**: Đảm bảo minh bạch và công bằng cho mọi người chơi.
* **Hệ thống muốn gây dựng cộng đồng Web3**: Thu hút người dùng crypto thông qua mô hình chơi để kiếm (Play-to-Earn).
* **Nhu cầu kiểm chứng và minh bạch cao**: Ví dụ các trò chơi tổ chức giải đấu, cần chứng minh công bằng.

## 8.6 Kết luận so sánh

Mặc dù còn một số rào cản kỹ thuật và trải nghiệm người dùng, hệ thống trò chơi đoán số sử dụng công nghệ blockchain đã chứng minh được nhiều ưu điểm vượt trội về tính minh bạch, bảo mật và khả năng mở rộng. Đây là xu hướng mới của các trò chơi phi tập trung trong kỷ nguyên Web3, đặc biệt phù hợp với những ai muốn ứng dụng công nghệ blockchain vào lĩnh vực giải trí và tài chính

# KẾT LUẬN

**9.1 Tóm tắt nội dung đã thực hiện**

Trong khuôn khổ đề tài, nhóm đã nghiên cứu, thiết kế và xây dựng thành công một ứng dụng trò chơi đoán số đơn giản, vận hành trên nền tảng blockchain Ethereum, kết hợp với ví điện tử MetaMask để người dùng tương tác và nhận phần thưởng. Hệ thống sử dụng smart contract viết bằng ngôn ngữ Solidity, triển khai trên mạng testnet và tích hợp frontend bằng ReactJS (Vite) để đảm bảo trải nghiệm người dùng trực quan, mượt mà.

Dự án không chỉ giúp nhóm nắm bắt kiến thức về cách hoạt động của blockchain, ví phi tập trung, hợp đồng thông minh, mà còn thực hành triển khai một mô hình Web3 thực tế – từ backend phi tập trung cho đến giao diện người dùng hiện đại.

## 9.2 Những kết quả đạt được

* Hiểu rõ quy trình xây dựng một ứng dụng phi tập trung (DApp) từ A đến Z.
* Thiết kế và triển khai thành công hợp đồng thông minh (Smart Contract) có thể tự động xử lý logic trò chơi.
* Tích hợp MetaMask thành công giúp người dùng tương tác trực tiếp với blockchain.
* Xây dựng giao diện người dùng thân thiện, trực quan bằng ReactJS.
* Thử nghiệm, kiểm tra và đánh giá hiệu quả hoạt động của hệ thống trên mạng testnet.

## 9.3 Hạn chế của hệ thống

Dù đã hoàn thiện ở mức cơ bản, hệ thống vẫn còn một số hạn chế nhất định:

* Giao diện đơn giản, chưa có nhiều tính năng mở rộng.
* Chưa tích hợp hệ thống tạo số ngẫu nhiên an toàn (Chainlink VRF).
* Người dùng mới vẫn cần có kiến thức nền tảng về ví và blockchain.
* Hiện tại chỉ hoạt động trên mạng thử nghiệm, chưa có cơ chế phân phối token thực.

## 9.4 Bài học kinh nghiệm

Thông qua quá trình thực hiện, nhóm đã rút ra được nhiều bài học giá trị:

* **Blockchain không chỉ là tài chính**, mà còn có thể ứng dụng vào game, giải trí, giáo dục...
* **Bảo mật là ưu tiên hàng đầu** khi làm việc với smart contract vì mọi thao tác là không thể hoàn tác.
* **Giao diện người dùng là cầu nối quan trọng** giúp công nghệ blockchain trở nên dễ tiếp cận hơn.
* Việc kết hợp nhiều công nghệ đòi hỏi sự phối hợp tốt giữa frontend, backend và tư duy hệ thống tổng thể.

## 9.5 Định hướng phát triển tương lai

* Mở rộng trò chơi theo hướng đa dạng hóa gameplay và cơ chế phần thưởng.
* Tích hợp hệ thống tạo số ngẫu nhiên minh bạch như Chainlink VRF.
* Xây dựng mô hình kinh tế token (tokenomics) và hệ thống NFT.
* Mở rộng sang mạng chính và triển khai các chiến dịch tương tác cộng đồng (Discord, Telegram, airdrop...).

## 9.6 Kết luận chung

Đề tài “Trò chơi đoán số tích hợp công nghệ blockchain và ví MetaMask” tuy đơn giản về mặt ý tưởng, nhưng đã thể hiện rõ sức mạnh của công nghệ Web3 trong việc mang lại tính minh bạch, phi tập trung và quyền sở hữu thực sự cho người dùng. Đây không chỉ là một trò chơi, mà còn là một bước thử nghiệm và tiền đề cho những ứng dụng blockchain lớn hơn trong tương lai. Dự án khẳng định tiềm năng của các mô hình trò chơi phi tập trung (GameFi) và sự cần thiết của việc kết hợp giữa kỹ thuật phần mềm với tư duy kinh tế số hiện đại.