s

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

##### **TÊN MÔN HỌC: Công nghệ BlockChain**

**ĐỀ TÀI: NFT với metadata tùy biến**

**Giảng viên hướng dẫn: ThS. Trần Dăng Công**

|  |  |
| --- | --- |
| **Họ và tên** | **Mã sinh viên** |
| **Lê Ngọc Tú** | **1571020266** |
| **Hoàng Quốc Hạnh** | **1571020087** |
| **Nguyễn Minh Đức** | **1571020068** |

**Hà Nội, 2025**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

##### **TÊN MÔN HỌC: Công nghệ BlockChain**

**ĐỀ TÀI: NFT với metadata tùy biến**

**Giảng viên hướng dẫn: ThS. Trần Dăng Công**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MSV** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1571020266** | **Lê Ngọc Tú** | **20/05/2003** |  |  |
| **1751020087** | **Hoàng Quóc Hạnh** | **06/01/2003** |  |  |
|  | **Nguyễn Minh Đức** | **04/09/2003** |  |  |

**Hà Nội, 2025**

### ****LỜI MỞ ĐẦU****

**Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu**, giáo dục là một trong những lĩnh vực chịu tác động rõ nét nhất từ sự phát triển của các công nghệ mới. Việc ứng dụng công nghệ không chỉ nhằm mục tiêu hiện đại hoá phương pháp giảng dạy và học tập, mà còn hướng đến việc cải thiện tính minh bạch, hiệu quả và cá nhân hoá trong toàn bộ quá trình giáo dục.

**Blockchain**, một công nghệ nền tảng nổi bật trong kỷ nguyên số, đã chứng minh tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, chuỗi cung ứng, y tế và gần đây là trong giáo dục. Với đặc tính **phi tập trung, bất biến và minh bạch**, Blockchain tạo điều kiện thuận lợi cho việc lưu trữ và xác thực dữ liệu học tập một cách an toàn, đáng tin cậy, và không thể bị làm giả.

**Báo cáo này trình bày một giải pháp xây dựng hệ thống website học tập tích hợp công nghệ Blockchain**, với mục tiêu cung cấp một nền tảng học trực tuyến nơi thành tích học tập của người dùng được ghi nhận và xác minh thông qua NFT (Non-Fungible Token) và metadata tuỳ biến. Cụ thể:

**NFT trong giáo dục** được sử dụng như một dạng chứng chỉ số hoá không thể thay thế, đại diện cho từng thành tích học tập cá nhân (hoàn thành khoá học, vượt qua kỳ kiểm tra, đạt chứng nhận kỹ năng...). Mỗi NFT được liên kết duy nhất với người học, đảm bảo tính sở hữu và không thể sao chép.

**Metadata tuỳ biến** đi kèm mỗi NFT sẽ chứa thông tin chi tiết như tên khóa học, đơn vị cấp chứng chỉ, điểm số, ngày hoàn thành, kỹ năng đạt được... Những dữ liệu này giúp trực quan hoá và cá nhân hoá thành tích học tập, đồng thời dễ dàng tích hợp vào hồ sơ năng lực số.

Hệ thống cũng cho phép **xác thực công khai** trên nền tảng Blockchain, giúp nhà tuyển dụng, tổ chức giáo dục hoặc các bên liên quan có thể kiểm tra độ xác thực của chứng chỉ một cách nhanh chóng và tin cậy, loại bỏ nguy cơ gian lận học thuật.

Giải pháp đề xuất không chỉ hướng đến việc nâng cao hiệu quả quản lý học tập, mà còn mở ra một hướng đi mới trong việc xây dựng hệ sinh thái giáo dục số – nơi **người học có toàn quyền sở hữu, kiểm soát và chứng minh hành trình học tập của mình một cách minh bạch và không thể chối cãi**.

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc27076)

[CHƯƠNG 1: GIỚI THIỆU 7](#_Toc16108)

[I Tổng quan về Blockchain và NFT 7](#_Toc28446)

[II Tổng quan về đề tài 7](#_Toc13238)

[1. Tên đề tài 7](#_Toc31187)

[2. Mục tiêu của đề tài 8](#_Toc29053)

[CHƯƠNG II: Nền tảng và công nghệ sử dụng 9](#_Toc23408)

[2.1. Công nghệ chính 9](#_Toc24046)

[2.2. Kiến trúc tổng thể của hệ thống 10](#_Toc15328)

[2.3. Luồng xử lý dữ liệu 10](#_Toc15991)

[2.4. Sơ đồ minh họa kiến trúc hệ thống (mô tả văn bản) 11](#_Toc15194)

[CHƯƠNG III:Thiết kế giao diện và trải nghiệm người dùng 13](#_Toc26860)

[3.1. Trang chủ (Homepage) 13](#_Toc7412)

[3.2. Trang học (Learning Page) 14](#_Toc3669)

[3.3. Trang NFT cá nhân (My Certificates / NFT Dashboard) 15](#_Toc1719)

[3.4. Giao diện kết nối ví (Wallet Integration) 16](#_Toc6445)

[CHƯƠNG IV: NFT và Metadata tùy biến trong giáo dục 17](#_Toc2918)

[4.1. Khái niệm NFT trong hệ thống 17](#_Toc8858)

[4.2. Metadata tùy biến cho NFT 17](#_Toc4945)

[4.3. Mô hình cập nhật Metadata theo điều kiện 18](#_Toc24167)

[4.4. Cơ chế cập nhật metadata 18](#_Toc9449)

[Chương V: Phân tích chức năng và mã nguồn hệ thống 20](#_Toc7107)

[5.1. Controller quản lý nội dung học tập - QuizController 20](#_Toc21249)

[5.2. Luồng hoạt động tổng quan 22](#_Toc10521)

[5.3. Xử lý nộp bài, tính điểm và cập nhật Blockchain 22](#_Toc1482)

[5.4. Hàm đánh giá Rank theo điểm số 24](#_Toc7034)

[5.5. Kiểm tra điều kiện truy cập bài tập theo Rank 24](#_Toc15639)

[5.6. Các phương thức bổ sung khác 25](#_Toc15975)

[5.7. Quản lý dữ liệu người dùng và bài kiểm tra 25](#_Toc7538)

[5.8. Cấu trúc tệp dữ liệu bài kiểm tra (quiz.json) 27](#_Toc22284)

[5.9. Cấu trúc tệp dữ liệu người dùng (user.json) 29](#_Toc26363)

[5.10. Mô hình dữ liệu gửi lên Blockchain (PlayerInputModel) 30](#_Toc32293)

[CHƯƠNG VI: Quy trình hoạt động của hệ thống 33](#_Toc11314)

[Giải thích mã nguồn HomeController.cs 34](#_Toc11805)

[1. Khai báo thư viện 34](#_Toc4117)

[2. Khai báo lớp HomeController 34](#_Toc20274)

[3. Constructor 34](#_Toc20032)

[4. Phương thức Index() 35](#_Toc6555)

[5. Phương thức Privacy() 35](#_Toc9451)

[6. Phương thức Error() 35](#_Toc9735)

[Tóm lại 36](#_Toc14332)

[6.1. Tổng kết nội dung đề tài 40](#_Toc1776)

[6.2. Hạn chế của đề tài 40](#_Toc32137)

[6.3. Bài học kinh nghiệm 41](#_Toc22927)

[6.4. Hướng phát triển tiếp theo 41](#_Toc1912)

[Kết luận 42](#_Toc29564)

[TÀI LIỆU THAM KHẢO 43](#_Toc23334)

**CHƯƠNG 1: GIỚI THIỆU**

**I Tổng quan về Blockchain và NFT**

#### **1. Blockchain**

Blockchain là một công nghệ lưu trữ dữ liệu phân tán dưới dạng chuỗi khối, nơi mỗi khối (block) chứa một tập hợp các giao dịch hoặc thông tin và được liên kết với khối trước đó bằng thuật toán mã hóa. Đặc điểm nổi bật của công nghệ này là:

**Phi tập trung**: Dữ liệu không bị kiểm soát bởi một cá nhân hay tổ chức trung gian nào mà được duy trì bởi mạng lưới các nút (nodes) ngang hàng.

**Tính bất biến**: Một khi dữ liệu đã được ghi vào Blockchain thì không thể sửa đổi hoặc xóa bỏ, đảm bảo tính toàn vẹn và chống gian lận.

**Minh bạch và có thể xác thực**: Tất cả các giao dịch đều có thể được kiểm tra công khai trên hệ thống, giúp tăng độ tin cậy và minh bạch.

Trong giáo dục, Blockchain mở ra khả năng lưu trữ kết quả học tập, chứng chỉ, tiến trình đào tạo... theo cách không thể làm giả, từ đó giúp xây dựng một hồ sơ học tập điện tử minh bạch và đáng tin cậy.

#### **2. NFT (Non-Fungible Token)**

NFT là một loại tài sản kỹ thuật số đặc biệt trên nền tảng Blockchain, đại diện cho một đối tượng duy nhất, không thể thay thế hoặc phân chia. Mỗi NFT có một định danh riêng và có thể chứa metadata mô tả chi tiết về nội dung, nguồn gốc, hoặc quyền sở hữu.

Trong lĩnh vực giáo dục, NFT có thể được sử dụng để:

**Cấp phát chứng chỉ học tập dạng số** dưới dạng các NFT duy nhất cho từng người học.

**Chứa metadata tùy biến** như tên khóa học, điểm số, thời gian hoàn thành, đơn vị đào tạo,...

**Cho phép xác minh công khai** các chứng chỉ mà không cần phụ thuộc vào bên thứ ba.

Sự kết hợp giữa Blockchain và NFT giúp xây dựng một hệ thống quản lý thành tích học tập minh bạch, bền vững và hoàn toàn do người học làm chủ.

**II Tổng quan về đề tài**

### ****1. Tên đề tài****

**Xây dựng Website học tập ứng dụng Blockchain, NFT và Metadata tùy biến**

### ****2. Mục tiêu của đề tài****

Đề tài hướng đến việc phát triển một nền tảng học tập trực tuyến hiện đại, ứng dụng công nghệ Blockchain kết hợp NFT và metadata tùy biến nhằm đảm bảo tính minh bạch, an toàn và cá nhân hóa trong việc ghi nhận và xác minh thành tích học tập. Cụ thể:

**Phát triển giao diện người dùng thân thiện:**  
Thiết kế một website học tập có giao diện đơn giản, dễ sử dụng và đáp ứng trải nghiệm người dùng tốt trên nhiều thiết bị (desktop, tablet, mobile).

**Tích hợp công nghệ Blockchain:**  
Áp dụng công nghệ Blockchain để lưu trữ dữ liệu học tập, đảm bảo tính minh bạch, bất biến và có thể xác thực công khai.

**Ứng dụng NFT làm chứng chỉ số hóa:**  
Sử dụng NFT (Non-Fungible Token) để đại diện cho từng chứng chỉ học tập cá nhân. Mỗi NFT sẽ là duy nhất, không thể thay thế hoặc sao chép, giúp tăng độ tin cậy của thành tích người học.

**Tùy biến metadata theo thành tích học tập:**  
Gắn kèm metadata vào từng NFT để thể hiện các thông tin chi tiết như tên khóa học, ngày hoàn thành, điểm số, kỹ năng đạt được... Các metadata này có thể cập nhật theo tiến trình học tập, phản ánh rõ ràng sự phát triển năng lực của người học.

**Tăng cường minh bạch và giá trị lâu dài:**  
Hệ thống giúp các bên liên quan như nhà tuyển dụng, cơ sở đào tạo hoặc tổ chức xác thực thông tin học tập một cách minh bạch, nhanh chóng và bền vững theo thời gian.

# CHƯƠNG ****II:**** Nền tảng và công nghệ sử dụng

### ****2.1. Công nghệ chính****

Để xây dựng hệ thống website học tập ứng dụng Blockchain, NFT và metadata tùy biến, đề tài sử dụng các công nghệ hiện đại thuộc nhiều lĩnh vực như lập trình giao diện người dùng, phát triển hợp đồng thông minh, giao tiếp với Blockchain và lưu trữ phi tập trung. Cụ thể:

**Frontend: ReactJS, HTML5, CSS3, JavaScript**  
Giao diện người dùng của hệ thống được xây dựng bằng **ReactJS**, một thư viện JavaScript phổ biến, giúp phát triển các ứng dụng web một cách linh hoạt, dễ mở rộng và có khả năng tái sử dụng các thành phần (components).  
Kết hợp với **HTML5** và **CSS3**, giao diện sẽ được thiết kế hiện đại, đáp ứng tốt trên nhiều thiết bị. **JavaScript** đóng vai trò là ngôn ngữ lập trình chính, điều khiển các tương tác động trên website.

**Smart Contract: Solidity**  
**Solidity** là ngôn ngữ lập trình chuyên dụng để phát triển các hợp đồng thông minh (smart contracts) trên các nền tảng Blockchain tương thích với **Ethereum Virtual Machine (EVM)** như Ethereum và Polygon.  
Trong dự án này, Solidity được dùng để tạo ra các hợp đồng thông minh nhằm:

Phát hành NFT làm chứng chỉ học tập

Lưu trữ các thông tin liên quan đến người học

Cập nhật metadata dựa trên tiến trình học tập

**Giao tiếp với Blockchain: Web3.js hoặc Ethers.js**  
Để frontend có thể tương tác với các smart contract trên Blockchain, ta sử dụng **Web3.js** hoặc **Ethers.js**. Đây là hai thư viện phổ biến hỗ trợ:

Kết nối ví người dùng (như MetaMask)

Gửi giao dịch đến Blockchain

Đọc dữ liệu từ hợp đồng thông minh

**Lưu trữ Metadata: IPFS (InterPlanetary File System)**  
Metadata của các NFT – bao gồm thông tin về khóa học, điểm số, kỹ năng đạt được... – sẽ được lưu trữ trên **IPFS**, một hệ thống lưu trữ phi tập trung, đảm bảo rằng dữ liệu không phụ thuộc vào máy chủ tập trung, và không thể bị chỉnh sửa sau khi đã phát hành. IPFS cũng cung cấp địa chỉ băm (hash) duy nhất, giúp liên kết trực tiếp với NFT trên Blockchain.

**Ví kết nối: MetaMask**  
**MetaMask** là một ví điện tử phổ biến, cho phép người dùng quản lý tài sản số và tương tác với các DApps (ứng dụng phi tập trung). Trong hệ thống này, MetaMask được sử dụng để:

Xác thực người dùng

Ký các giao dịch khi phát hành NFT hoặc cập nhật dữ liệu học tập

Quản lý và hiển thị NFT chứng chỉ của từng người học

### ****2.2. Kiến trúc tổng thể của hệ thống****

Hệ thống được xây dựng theo mô hình **Web3 kết hợp công nghệ phi tập trung**, gồm ba lớp chính:

#### **1. Giao diện người dùng (Frontend)**

Được phát triển bằng **ReactJS**, cho phép người học đăng nhập, tham gia khóa học, theo dõi tiến độ, nhận chứng chỉ và xem NFT của mình.

Kết nối với ví điện tử (MetaMask) để xác thực và thực hiện các giao dịch Blockchain.

#### **2. Lớp logic ứng dụng (Smart Contracts)**

Viết bằng **Solidity**, triển khai trên mạng **Ethereum hoặc Polygon**.

Chức năng:

Phát hành NFT làm chứng chỉ.

Lưu trữ ID người học và liên kết với metadata.

Cập nhật tiến trình học tập bằng các hàm (functions) công khai.

#### **3. Lưu trữ dữ liệu phi tập trung**

**IPFS** được sử dụng để lưu trữ các metadata (dưới dạng JSON) chứa thông tin về chứng chỉ, thành tích, kỹ năng,...

Mỗi tệp metadata có địa chỉ IPFS duy nhất, được gắn vào NFT khi phát hành.

### ****2.3. Luồng xử lý dữ liệu****

Quy trình hoạt động cơ bản của hệ thống có thể được mô tả theo các bước sau:

**Người dùng truy cập hệ thống và kết nối ví (MetaMask)**  
→ Xác thực danh tính thông qua địa chỉ ví.

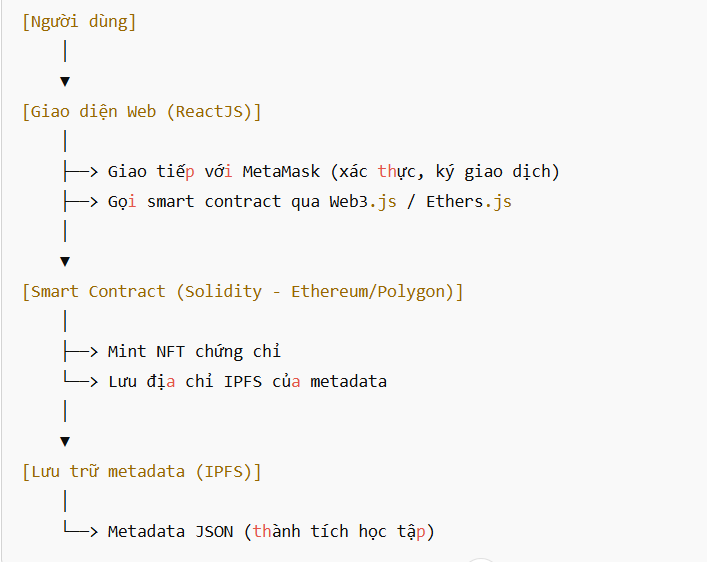
**Người dùng tham gia khóa học và hoàn thành các bài học / bài kiểm tra**  
→ Dữ liệu tiến độ được lưu tạm thời trên server (off-chain) hoặc local (nếu cần).

**Khi người học hoàn thành khóa học**  
→ Metadata chứa thông tin chứng chỉ (tên khóa học, điểm số, ngày hoàn thành...) được tạo dưới dạng file JSON và lưu trữ lên **IPFS**.

**Hệ thống gọi smart contract để mint NFT**  
→ NFT được phát hành và gắn kèm địa chỉ IPFS của metadata.  
→ NFT được gửi đến ví người học (địa chỉ MetaMask).

**Người dùng có thể xem và chia sẻ NFT chứng chỉ**  
→ NFT này có thể xác thực công khai trên Blockchain (không thể chỉnh sửa).

### ****2.4. Sơ đồ minh họa kiến trúc hệ thống (mô tả văn bản)****



**CHƯƠNG III:Thiết kế giao diện và trải nghiệm người dùng**

### ****3.1. Trang chủ (Homepage)****

**Chức năng chính:**

**Giới thiệu tổng quan** về nền tảng, lợi ích của việc ứng dụng Blockchain và NFT trong giáo dục.

Trình bày ngắn gọn về các **công nghệ chính**: Blockchain, NFT, IPFS, MetaMask, v.v.

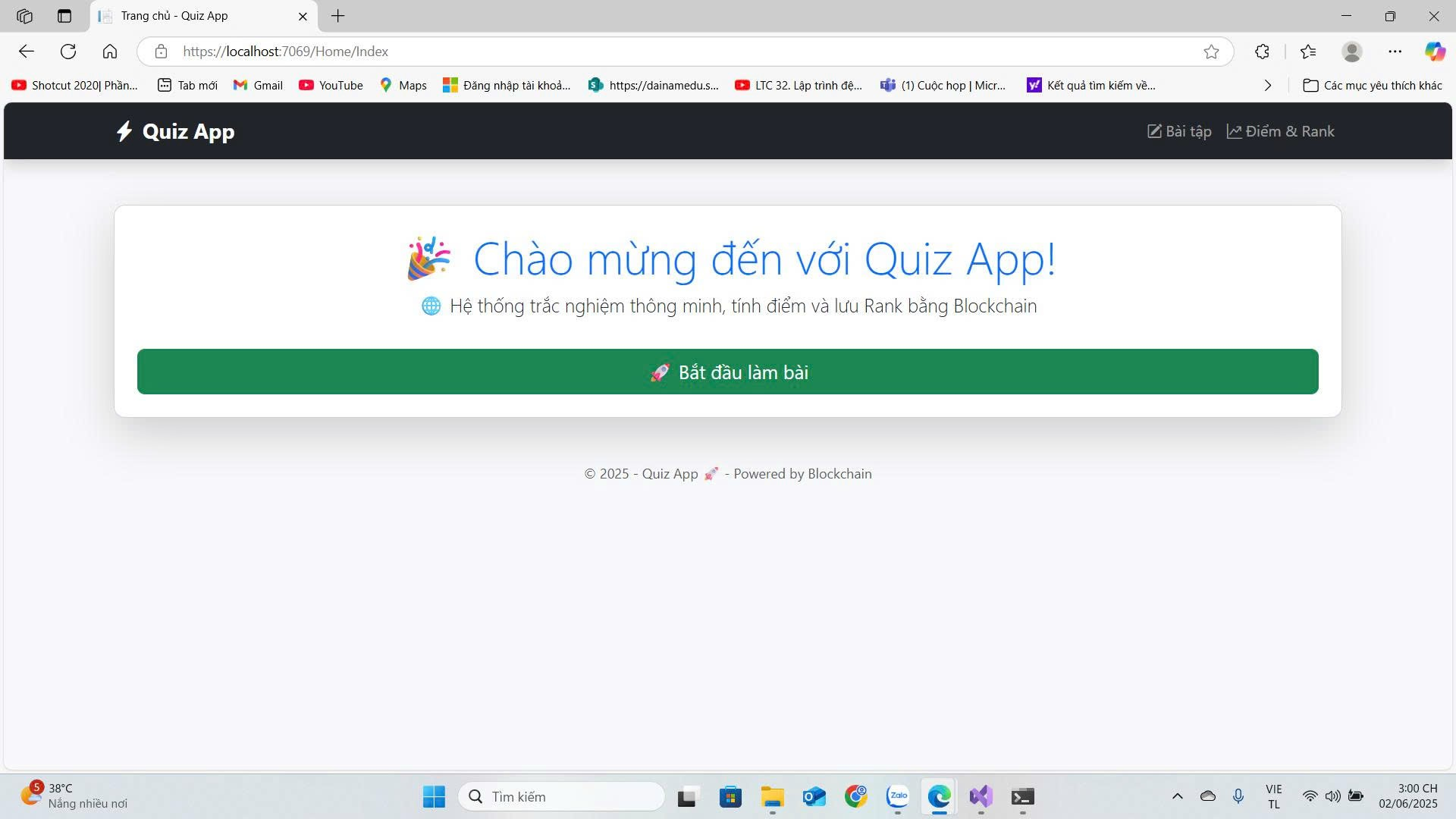
**Hiển thị danh sách khóa học nổi bật** hoặc đang được quan tâm nhiều, kèm theo hình ảnh, mô tả ngắn, và nút bắt đầu học.

**Yếu tố thiết kế:**

Thiết kế hiện đại, bố cục dễ nhìn, trực quan.

Thanh điều hướng (navbar) đơn giản, có liên kết đến các phần: Trang chủ, Khóa học, NFT cá nhân, Hướng dẫn, Đăng nhập.

Nút **kết nối ví MetaMask** đặt nổi bật, dễ tìm.



### ****3.2. Trang học (Learning Page)****

**Chức năng chính:**

**Phát nội dung học tập**: bao gồm video bài giảng, nội dung tương tác và tài liệu đính kèm dạng PDF.

Sau mỗi bài học có **câu hỏi trắc nghiệm** để đánh giá mức độ tiếp thu của người học.

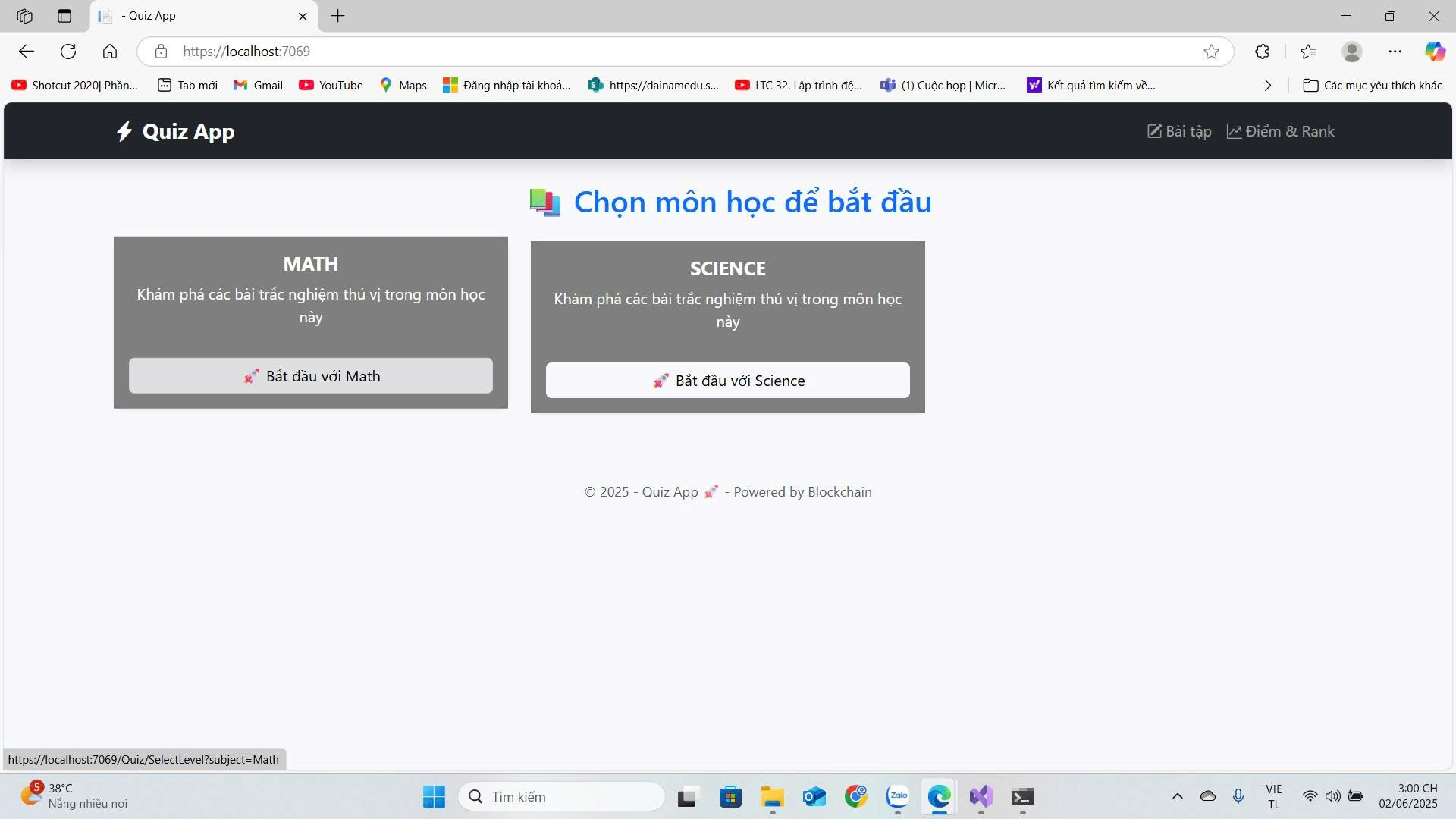
**Hiển thị tiến trình học tập** bằng thanh phần trăm hoặc sơ đồ bước (step indicator), cập nhật theo thời gian thực.

**Yếu tố thiết kế:**

Bố cục chia hai cột: bên trái là nội dung bài học, bên phải là mục lục khóa học và tiến trình học.

Giao diện trắc nghiệm đơn giản, phản hồi ngay sau khi nộp bài.

Thiết kế responsive, tương thích cả trên máy tính và điện thoại.



### ****3.3. Trang NFT cá nhân (My Certificates / NFT Dashboard)****

**Chức năng chính:**

Hiển thị **danh sách các NFT chứng nhận** người học đã đạt được.

Mỗi NFT đại diện cho một khóa học hoàn thành, với hình ảnh, tiêu đề, mô tả.

Nhấn vào từng NFT để xem thông tin chi tiết:

Tên khóa học

Điểm số

Ngày hoàn thành

Cấp độ/chứng chỉ đạt được

Nút kiểm tra trạng thái **Blockchain** (xem trên blockchain explorer)

Metadata đính kèm (truy cập qua IPFS)

**Yếu tố thiết kế:**

Giao diện dạng **thẻ (card layout)** cho từng NFT.

Dùng màu sắc để phân loại cấp độ (ví dụ: cơ bản, trung cấp, nâng cao).

Cho phép chia sẻ đường dẫn chứng chỉ công khai.

### ****3.4. Giao diện kết nối ví (Wallet Integration)****

**Chức năng chính:**

**Kết nối với MetaMask** để xác thực người dùng dựa trên địa chỉ ví.

Khi người dùng đăng nhập, hệ thống yêu cầu **ký một thông báo xác thực (message signing)** thay vì sử dụng tên đăng nhập/mật khẩu truyền thống.

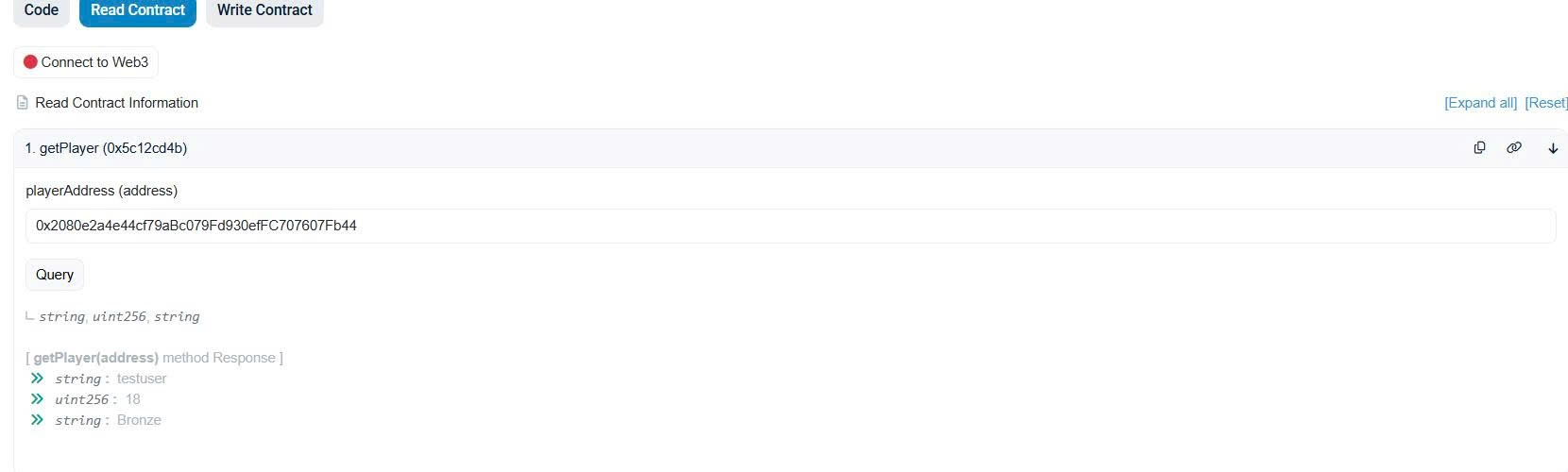
Sau khi xác minh thành công, giao diện cập nhật theo trạng thái đăng nhập (hiển thị ví, NFT, tiến trình học,...).

**Yếu tố thiết kế:**

Hiển thị rõ trạng thái ví: đã kết nối / chưa kết nối.

Nếu chưa có MetaMask, hiển thị liên kết hướng dẫn cài đặt.

Trạng thái xác minh hiển thị dưới dạng popup hoặc toast thông báo.



**CHƯƠNG IV: NFT và Metadata tùy biến trong giáo dục**

### ****4.1. Khái niệm NFT trong hệ thống****

**NFT (Non-Fungible Token)** là một loại tài sản số duy nhất, không thể thay thế lẫn nhau, được phát hành trên nền tảng Blockchain. Mỗi NFT có một mã định danh riêng biệt và có thể chứa thông tin mô tả thông qua metadata.

Trong hệ thống này, **NFT được sử dụng để cấp chứng chỉ học tập**, đại diện cho sự hoàn thành khóa học của người học. Mỗi NFT chứng chỉ là duy nhất và có thể xác minh công khai trên mạng Blockchain.

Lợi ích của việc sử dụng NFT làm chứng chỉ:

Không thể làm giả hoặc chỉnh sửa.

Dễ dàng xác thực nguồn gốc và thành tích.

Người học sở hữu hoàn toàn chứng chỉ trong ví cá nhân.

### ****4.2. Metadata tùy biến cho NFT****

**Metadata** là phần thông tin mô tả gắn liền với NFT, thường được lưu trữ dưới dạng file .json trên hệ thống phi tập trung như **IPFS**. Trong đề tài này, metadata bao gồm:

Tên khóa học

Điểm số

Cấp độ người học

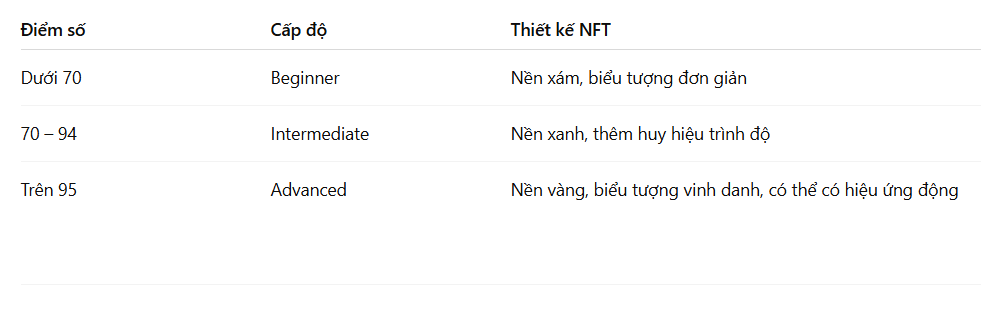
Ngày hoàn thành

Hình ảnh đại diện cho chứng chỉ (dựa trên cấp độ)

Mô tả chi tiết về nội dung học

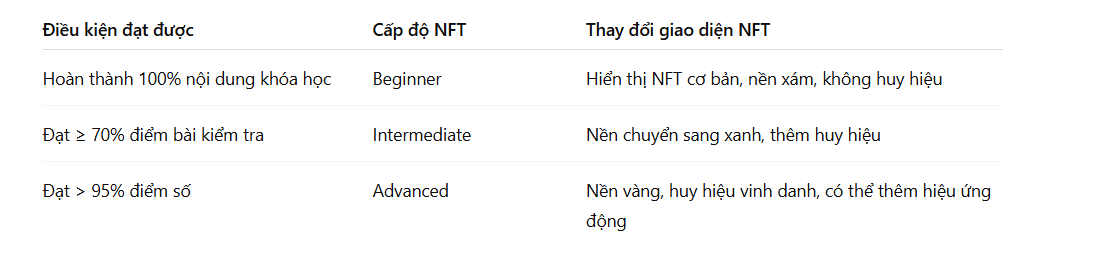
#### **Tùy biến theo thành tích người học**

Metadata được cập nhật linh hoạt dựa trên hiệu suất học tập. Giao diện NFT được thiết kế sinh động để phản ánh trình độ người học qua màu sắc, biểu tượng và huy hiệu:



### ****4.3. Mô hình cập nhật Metadata theo điều kiện****

Việc cập nhật metadata không chỉ dựa trên điểm số mà còn theo tiến trình hoàn thành khóa học. Dưới đây là bảng mô tả điều kiện và ảnh hưởng đến giao diện NFT:



### ****4.4. Cơ chế cập nhật metadata****

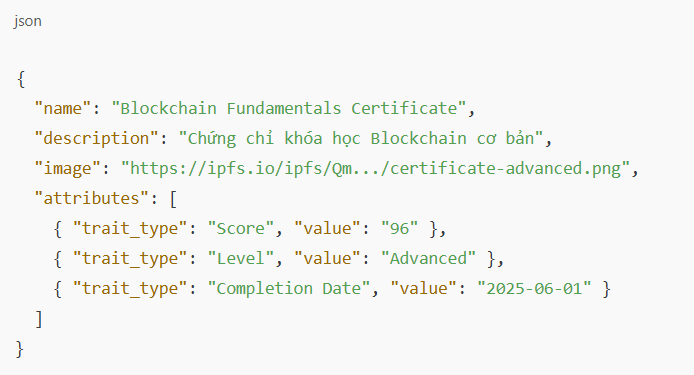
NFT trong hệ thống sử dụng **hàm** tokenURI(uint256 tokenId) của hợp đồng thông minh để xác định đường dẫn metadata tương ứng. Khi người học đạt điều kiện mới (ví dụ: cải thiện điểm số), hệ thống thực hiện:

Tạo file .json metadata mới với thông tin cập nhật.

Upload file lên **IPFS**, nhận về **CID (Content Identifier)**.

Gọi hàm smart contract để cập nhật tokenURI mới, trỏ đến địa chỉ IPFS mới.

Ví dụ đoạn metadata JSON:



## ****Chương V: Phân tích chức năng và mã nguồn hệ thống****

### ****5.1. Controller quản lý nội dung học tập -**** QuizController

Controller QuizController chịu trách nhiệm xử lý các thao tác chính liên quan đến việc học, làm bài kiểm tra, phân cấp người học và tích hợp Blockchain.

#### **5.1.1. Khai báo thư viện**

using Microsoft.AspNetCore.Mvc;using btlquiz.Data;using btlquiz.Models;using btlquiz.Services;

Các namespace được sử dụng:

btlquiz.Data: Chứa dữ liệu bài tập, người dùng (tĩnh hoặc giả lập).

btlquiz.Models: Mô hình dữ liệu như Quiz, User.

btlquiz.Services: Chứa lớp BlockchainService để tương tác với Smart Contract.

#### **5.1.2. Khởi tạo** BlockchainService

private readonly BlockchainService \_blockchainService;

public QuizController()

{

\_blockchainService = new BlockchainService();

}

Khởi tạo dịch vụ để xử lý các chức năng liên quan đến Blockchain như mint NFT, cập nhật metadata,...

#### **5.1.3. Lựa chọn môn học –** SelectSubject()

public IActionResult SelectSubject()

{

var subjects = DataService.GetAllSubjects();

return View(subjects);

}

Truy xuất danh sách các môn học có sẵn từ lớp DataService.

Hiển thị ra view để người dùng lựa chọn môn học muốn làm bài tập.

#### **5.1.4. Lựa chọn cấp độ –** SelectLevel(string subject)

public IActionResult SelectLevel(string subject)

{

...

var allowedLevels = new List<string>();

foreach (var quiz in quizzes)

{

if (IsRankSufficient(user.Rank, quiz.Level))

{

allowedLevels.Add(quiz.Level);

}

}

...

}

Sau khi chọn môn học, hệ thống lọc các bài kiểm tra theo level (cấp độ).

Chỉ hiển thị các bài phù hợp với rank của người dùng (ví dụ: Beginner, Intermediate, Advanced).

#### **5.1.5. Làm bài trắc nghiệm –** TakeQuiz(GET)

[HttpGet]public IActionResult TakeQuiz(string subject, string level)

Hiển thị nội dung câu hỏi trắc nghiệm theo môn học và cấp độ.

Kiểm tra người dùng có đủ rank để làm bài hay không. Nếu không đủ thì thông báo từ chối truy cập.

#### **5.1.6. Nộp bài và cập nhật Blockchain –** TakeQuiz(POST)

[HttpPost]public async Task<IActionResult> TakeQuiz(string subject, string level, Dictionary<int, int> answers)

Nhận kết quả làm bài của người dùng dưới dạng Dictionary<int, int>: ID câu hỏi → ID đáp án đã chọn.

Tính điểm dựa trên đáp án đúng.

Cập nhật điểm, rank và lưu thông tin lên Blockchain thông qua BlockchainService.

Có thể mint NFT mới hoặc cập nhật metadata của NFT đã tồn tại, tùy theo tiến độ người học.

### ****5.2. Luồng hoạt động tổng quan****

Người dùng chọn môn học → SelectSubject.

Chọn cấp độ phù hợp với rank → SelectLevel.

Làm bài trắc nghiệm → TakeQuiz (GET).

Gửi kết quả, tính điểm, cập nhật thông tin → TakeQuiz (POST).

Gửi metadata lên IPFS, gọi smart contract cập nhật NFT.

Nếu bạn muốn mình viết tiếp phân tích phần BlockchainService hay các model Quiz, User, hoặc logic tính điểm – chỉ cần gửi thêm code hoặc nói rõ là mình sẽ viết tiếp vào chương 3 hoặc chương khác tùy bạn muốn. Bạn muốn tiếp theo phân tích phần nào?

### ****5.3. Xử lý nộp bài, tính điểm và cập nhật Blockchain****

Sau khi người dùng hoàn tất phần làm bài trắc nghiệm, hệ thống sẽ xử lý kết quả thông qua phương thức TakeQuiz (POST). Dưới đây là các bước chi tiết:

#### **5.3.1. Tính điểm**

int score = 0;

foreach (var q in quiz.Questions.Select((value, index) => new { value, index }))

{

if (answers != null && answers.ContainsKey(q.index) && answers[q.index] == q.value.CorrectIndex)

score++;

}

Duyệt qua từng câu hỏi của bài kiểm tra.

Kiểm tra đáp án người dùng chọn với đáp án đúng (CorrectIndex).

Nếu đúng, tăng score.

#### **5.3.2. Cập nhật người dùng**

user.Score += score;

user.Rank = CalculateRank(user.Score);

DataService.SaveUser(user);

Cộng thêm điểm vào tổng điểm của người dùng.

Cập nhật Rank dựa trên điểm mới bằng hàm CalculateRank().

Lưu lại thông tin người dùng thông qua DataService.

#### **5.3.3. Cập nhật lên Blockchain**

var txHash = await \_blockchainService.SetPlayerAsync(user.Username, user.Score, user.Rank);

Gọi đến BlockchainService để ghi dữ liệu người học lên Blockchain (bao gồm: tên người dùng, điểm, rank).

Kết quả trả về là txHash – mã giao dịch đã gửi lên chuỗi khối.

#### **5.3.4. Thông báo kết quả**

ViewBag.Message = $"Bạn làm đúng {score} trên {quiz.Questions.Count} câu.<br/>" +

$"Tổng điểm hiện tại: {user.Score}.<br/>" +

$"Rank hiện tại: {user.Rank}.<br/>" +

$"Giao dịch đã gửi lên blockchain, TxHash: {txHash}";

Hiển thị thông tin kết quả điểm số, rank mới đạt được và mã giao dịch Blockchain.

Đây là một điểm nổi bật trong việc minh bạch hóa quá trình học thông qua công nghệ Blockchain.

### ****5.4. Hàm đánh giá Rank theo điểm số****

private string CalculateRank(int score)

{

if (score >= 80) return "Platinum";

if (score >= 50) return "Gold";

if (score >= 20) return "Silver";

return "Bronze";

}

Hệ thống phân chia thứ hạng (Rank) người học thành 4 cấp độ:

**Platinum**: từ 80 điểm trở lên.

**Gold**: từ 50–79 điểm.

**Silver**: từ 20–49 điểm.

**Bronze**: dưới 20 điểm.

Các rank này sẽ ảnh hưởng đến việc người dùng có thể truy cập các bài kiểm tra cấp độ cao hay không.

### ****5.5. Kiểm tra điều kiện truy cập bài tập theo Rank****

private bool IsRankSufficient(string userRank, string quizLevel)

Mỗi rank và level có giá trị số tương ứng:

Rank: Bronze (1), Silver (2), Gold (3), Platinum (4)

Level: Easy (1), Medium (2), Hard (3)

Hệ thống chỉ cho phép người dùng làm bài nếu userRankValue >= quizLevelValue.

### ****5.6. Các phương thức bổ sung khác****

#### **Hiển thị danh sách môn học:**

public IActionResult Subjects()

#### **Hiển thị danh sách bài tập theo môn học:**

public IActionResult Quizzes(string subject)

Hiển thị danh sách bài kiểm tra theo từng môn học, có lọc theo cấp độ phù hợp với rank.

### ****5.7. Quản lý dữ liệu người dùng và bài kiểm tra****

Toàn bộ dữ liệu của hệ thống được quản lý dưới dạng tệp tin JSON, thông qua lớp tĩnh DataService. Việc này đơn giản hóa quá trình phát triển hệ thống, đặc biệt phù hợp với mô hình MVP (Minimum Viable Product) hoặc dự án học tập.

#### **5.7.1. Dữ liệu bài kiểm tra (**quiz.json**)**

private static readonly string quizFile = Path.Combine(Directory.GetCurrentDirectory(), "Data", "quiz.json");

Đường dẫn tới tệp quiz.json chứa danh sách các bài kiểm tra (quiz).

Mỗi đối tượng Quiz gồm các thông tin như: Subject, Level, Questions, trong đó mỗi Question có Text, Options[], CorrectIndex.

##### **Đọc tất cả quiz:**

public static List<Quiz> GetAllQuizzes()

Đọc toàn bộ nội dung quiz.json và chuyển thành danh sách các đối tượng Quiz.

Nếu không có tệp, trả về danh sách rỗng.

##### **Lấy danh sách môn học duy nhất:**

public static List<string> GetAllSubjects()

Lấy danh sách Subject duy nhất từ các bài quiz có trong hệ thống.

Phục vụ cho bước người dùng chọn môn học trước khi làm bài.

#### **5.7.2. Dữ liệu người dùng (**user.json**)**

private static readonly string userFile = Path.Combine(Directory.GetCurrentDirectory(), "Data", "user.json");

Lưu trữ thông tin của người dùng đang sử dụng hệ thống, bao gồm:

Username

Score: tổng điểm tích lũy

Rank: thứ hạng hiện tại (Bronze, Silver, Gold, Platinum)

##### **Đọc thông tin người dùng:**

public static UserInfo GetUser()

Kiểm tra xem tệp user.json có tồn tại hay không.

Nếu có, đọc và deserialize thành đối tượng UserInfo.

Nếu không có, trả về một người dùng mặc định với Username = Unknown, Score = 0, Rank = Bronze.

##### **Lưu thông tin người dùng:**

public static void SaveUser(UserInfo user)

Serialize đối tượng UserInfo thành JSON.

Ghi đè nội dung vào tệp user.json.

#### **5.7.3. Tính đơn giản và mở rộng**

Việc sử dụng tệp JSON để lưu trữ dữ liệu mang lại các lợi ích:

Dễ quản lý và thao tác.

Không cần kết nối cơ sở dữ liệu phức tạp.

Phù hợp cho các ứng dụng demo, học thuật hoặc ứng dụng nhỏ.

Tuy nhiên, nếu triển khai thực tế ở quy mô lớn hơn, hệ thống nên được mở rộng sử dụng cơ sở dữ liệu như:

**SQLite** cho ứng dụng desktop nhỏ.

**SQL Server / MySQL** cho hệ thống web lớn.

Có thể tích hợp cùng **EF Core** để thao tác dữ liệu theo mô hình ORM.

### ****5.8. Cấu trúc tệp dữ liệu bài kiểm tra (****quiz.json****)****

Tệp quiz.json lưu trữ danh sách các bài kiểm tra (quizzes) theo định dạng JSON. Mỗi bài kiểm tra bao gồm 3 thuộc tính chính:

Subject: tên môn học (ví dụ: "Math", "Science").

Level: độ khó của bài kiểm tra (ví dụ: "Easy", "Hard").

Questions: danh sách các câu hỏi thuộc bài kiểm tra đó.

Mỗi câu hỏi (Question) gồm:

Text: nội dung câu hỏi.

Options: danh sách các phương án lựa chọn.

CorrectIndex: chỉ số phương án đúng trong mảng Options (bắt đầu từ 0).

#### **Ví dụ nội dung** quiz.json

[

{

"Subject": "Math",

"Level": "Easy",

"Questions": [

{

"Text": "2 + 2 = ?",

"Options": ["3", "4", "5", "6"],

"CorrectIndex": 1

},

{

"Text": "5 \* 1 = ?",

"Options": ["1", "5", "10", "0"],

"CorrectIndex": 1

}

]

},

{

"Subject": "Math",

"Level": "Hard",

"Questions": [

{

"Text": "√81 = ?",

"Options": ["7", "8", "9", "10"],

"CorrectIndex": 2

},

{

"Text": "12 \* 12 = ?",

"Options": ["144", "154", "124", "120"],

"CorrectIndex": 0

}

]

},

{

"Subject": "Science",

"Level": "Easy",

"Questions": [

{

"Text": "Water freezes at \_\_\_ °C",

"Options": ["0", "100", "50", "25"],

"CorrectIndex": 0

}

]

}]

#### **Phân tích:**

**Tính tổ chức:** Dữ liệu được phân nhóm rõ ràng theo môn học và độ khó.

**Tính mở rộng:** Có thể dễ dàng thêm môn mới, cấp độ mới hoặc câu hỏi mới mà không ảnh hưởng đến các cấu trúc hiện tại.

**Tương thích:** Mô hình dữ liệu phù hợp để ánh xạ trực tiếp sang các lớp C# (Quiz, Question) trong hệ thống ASP.NET MVC.

### ****5.9. Cấu trúc tệp dữ liệu người dùng (****user.json****)****

Tệp user.json được dùng để lưu thông tin cá nhân của người dùng trong quá trình sử dụng hệ thống học tập, bao gồm các thuộc tính:

Username: tên đăng nhập của người dùng.

Score: tổng điểm hiện tại người dùng đã đạt được thông qua các bài kiểm tra.

Rank: cấp bậc (rank) của người dùng, được phân loại dựa trên tổng điểm.

#### **Ví dụ nội dung** user.json

{

"Username": "testuser",

"Score": 18,

"Rank": "Bronze"}

#### **Giải thích:**

**Username**: định danh duy nhất người dùng trong hệ thống. Ví dụ: "testuser".

**Score**: tổng điểm tích lũy, được cập nhật sau mỗi bài kiểm tra. Trong ví dụ trên, người dùng hiện có 18 điểm.

**Rank**: cấp độ học tập tương ứng với điểm số. Hệ thống sử dụng hàm CalculateRank(int score) để xác định rank như sau:

Bronze: dưới 20 điểm.

Silver: từ 20 đến dưới 50 điểm.

Gold: từ 50 đến dưới 80 điểm.

Platinum: từ 80 điểm trở lên.

#### **Ý nghĩa:**

Tệp user.json giúp hệ thống ghi nhớ tiến độ học tập và kiểm soát truy cập vào các bài kiểm tra theo cấp độ khó (Easy, Medium, Hard).

Thông tin này cũng được ghi nhận trên blockchain (qua BlockchainService) nhằm đảm bảo tính minh bạch và bảo mật.

### ****5.10. Mô hình dữ liệu gửi lên Blockchain (****PlayerInputModel****)****

Để ghi nhận thông tin người chơi lên blockchain, hệ thống sử dụng lớp PlayerInputModel nhằm chuẩn hóa dữ liệu đầu vào cho dịch vụ BlockchainService.

#### **Định nghĩa lớp**

public class PlayerInputModel

{

public string Username { get; set; }

public int Score { get; set; }

public string Rank { get; set; }

}

#### **Các thuộc tính:**

Username: tên người dùng cần ghi nhận lên blockchain.

Score: tổng điểm người dùng đạt được.

Rank: cấp bậc hiện tại, phản ánh trình độ học tập của người dùng.

#### **Chức năng và mục đích:**

Lớp này hoạt động như một DTO (Data Transfer Object), giúp đóng gói thông tin người dùng trước khi truyền qua các phương thức xử lý blockchain, điển hình là:

await \_blockchainService.SetPlayerAsync(user.Username, user.Score, user.Rank);

PlayerInputModel tách biệt dữ liệu với logic xử lý, giúp tăng tính bảo trì, tái sử dụng và dễ dàng tích hợp với API hoặc hệ thống phân tán.

#### **Ví dụ:**

Giả sử người dùng "testuser" có 65 điểm, hệ thống sẽ tạo một đối tượng như sau:

var model = new PlayerInputModel

{

Username = "testuser",

Score = 65,

Rank = "Gold"

};

Đối tượng này sau đó sẽ được chuyển thành một giao dịch và gửi lên blockchain thông qua lớp BlockchainService.

# ****CHƯƠNG VI:**** Quy trình hoạt động của hệ thống

using System.Diagnostics;

using btlquiz.Models;

using Microsoft.AspNetCore.Mvc;

namespace btlquiz.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

## ****Giải thích mã nguồn**** HomeController.cs

### ****1. Khai báo thư viện****

using System.Diagnostics;using btlquiz.Models;using Microsoft.AspNetCore.Mvc;

System.Diagnostics: Dùng để truy cập thông tin hệ thống như Activity (giúp truy dấu request).

btlquiz.Models: Gọi tới mô hình ErrorViewModel (thường dùng để truyền dữ liệu lỗi sang View).

Microsoft.AspNetCore.Mvc: Cung cấp các lớp và thuộc tính để xây dựng controller và view trong ASP.NET Core MVC.

### ****2. Khai báo lớp**** HomeController

public class HomeController : Controller

Lớp này kế thừa từ Controller để sử dụng các chức năng MVC như trả về View(), xử lý HTTP request,...

### ****3. Constructor****

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

ILogger<HomeController>: Đối tượng ghi log để theo dõi và ghi lại các sự kiện, lỗi khi chạy ứng dụng.

Constructor này nhận đối tượng logger từ hệ thống dependency injection.

### ****4. Phương thức**** Index()

public IActionResult Index()

{

return View();

}

Trả về giao diện chính (Index.cshtml trong thư mục Views/Home).

Thường là trang chủ của ứng dụng.

### ****5. Phương thức**** Privacy()

public IActionResult Privacy()

{

return View();

}

Trả về giao diện thông tin bảo mật (Privacy.cshtml).

Đây là trang mặc định được tạo bởi ASP.NET Core khi khởi tạo một project mẫu.

### ****6. Phương thức**** Error()

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

Dùng để xử lý và hiển thị lỗi khi có vấn đề xảy ra trong hệ thống.

ErrorViewModel chứa RequestId giúp theo dõi lỗi theo từng yêu cầu.

ResponseCache: Ngăn trình duyệt và máy chủ lưu cache lỗi.

### ****Tóm lại****

HomeController là controller mặc định dùng để xử lý các yêu cầu đến trang chính.

Cung cấp các phương thức Index(), Privacy(), và Error() để điều hướng người dùng tới giao diện tương ứng.

Việc tách logic xử lý và giao diện giúp ứng dụng dễ bảo trì, mở rộng.

using Microsoft.AspNetCore.Mvc;

using btlquiz.Data;

using btlquiz.Models;

using btlquiz.Services;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using btlquiz.Models;

namespace btlquiz.Controllers

{

public class QuizController : Controller

{

private readonly BlockchainService \_blockchainService;

public QuizController()

{

\_blockchainService = new BlockchainService();

}

// Hiển thị danh sách môn học để chọn

public IActionResult SelectSubject()

{

var subjects = DataService.GetAllSubjects();

return View(subjects);

}

// Hiển thị danh sách bài tập theo môn đã chọn, kèm theo check rank

public IActionResult SelectLevel(string subject)

{

if (string.IsNullOrEmpty(subject))

return RedirectToAction("SelectSubject");

var quizzes = DataService.GetAllQuizzes()

.Where(q => q.Subject == subject).ToList();

var user = DataService.GetUser();

// Lọc những bài tập user đủ rank mới được làm

var allowedLevels = new List<string>();

foreach (var quiz in quizzes)

{

if (IsRankSufficient(user.Rank, quiz.Level))

{

allowedLevels.Add(quiz.Level);

}

}

ViewBag.Subject = subject;

ViewBag.AllowedLevels = allowedLevels;

return View(quizzes);

}

// GET: Hiển thị câu hỏi bài tập theo môn và level

[HttpGet]

public IActionResult TakeQuiz(string subject, string level)

{

if (string.IsNullOrEmpty(subject) || string.IsNullOrEmpty(level))

return RedirectToAction("SelectSubject");

var quizzes = DataService.GetAllQuizzes();

var quiz = quizzes.FirstOrDefault(q => q.Subject == subject && q.Level == level);

if (quiz == null)

return NotFound("Không tìm thấy bài tập.");

var user = DataService.GetUser();

if (!IsRankSufficient(user.Rank, level))

return Content("Rank của bạn chưa đủ để làm bài tập này.");

return View(quiz);

}

// POST: Nhận kết quả làm bài, tính điểm, cập nhật user và gửi lên blockchain

[HttpPost]

public async Task<IActionResult> TakeQuiz(string subject, string level, Dictionary<int, int> answers)

{

if (string.IsNullOrEmpty(subject) || string.IsNullOrEmpty(level))

return RedirectToAction("SelectSubject");

var quizzes = DataService.GetAllQuizzes();

var quiz = quizzes.FirstOrDefault(q => q.Subject == subject && q.Level == level);

if (quiz == null)

return NotFound("Không tìm thấy bài tập.");

### ****6.1. Tổng kết nội dung đề tài****

Trong khuôn khổ đề tài “Lập trình xe AlphaBot thu thập thông tin trong nông trại”, nhóm đã thiết kế và triển khai thành công một hệ thống giám sát môi trường ứng dụng công nghệ IoT, với các nội dung chính:

Lập trình điều khiển xe AlphaBot sử dụng ESP32 để di chuyển trong môi trường giả lập nông trại.

Tích hợp các cảm biến đo nhiệt độ, độ ẩm, gia tốc, giúp thu thập dữ liệu môi trường tại nhiều vị trí khác nhau.

Gửi dữ liệu thời gian thực về nền tảng điện toán đám mây Firebase, tạo tiền đề cho việc giám sát từ xa.

Kiểm thử hệ thống trong thực tế và thu được kết quả ổn định, chính xác.

Đề tài không chỉ đạt được mục tiêu kỹ thuật mà còn thể hiện tính ứng dụng cao trong sản xuất nông nghiệp hiện đại.

### ****6.2. Hạn chế của đề tài****

Phạm vi thử nghiệm còn nhỏ, chỉ mới giả lập trong môi trường đơn giản.

Cảm biến sử dụng chưa đa dạng, chưa phản ánh đầy đủ các yếu tố môi trường như ánh sáng, độ ẩm đất, pH.

Xe chưa được tích hợp tính năng tránh vật cản và điều hướng tự động.

### ****6.3. Bài học kinh nghiệm****

Làm quen với quy trình thiết kế hệ thống IoT hoàn chỉnh từ phần cứng, lập trình, kết nối đến lưu trữ dữ liệu.

Rèn luyện kỹ năng xử lý cảm biến, lập trình vi điều khiển và giao tiếp dữ liệu qua mạng.

Nâng cao tư duy giải quyết vấn đề thực tiễn bằng cách ứng dụng công nghệ vào đời sống.

### ****6.4. Hướng phát triển tiếp theo****

Nâng cấp hệ thống để hoạt động trong môi trường thực tế với quy mô lớn hơn.

Tích hợp thêm cảm biến và hệ thống điều khiển thông minh như AI, camera nhận diện, GPS, tự động tránh chướng ngại vật.

Phát triển ứng dụng điều khiển và hiển thị trên di động, cung cấp giao diện thân thiện và tiện lợi cho người dùng.

# ****Kết luận****

Đề tài **“Xây dựng Website học tập ứng dụng Blockchain và NFT tùy biến”** đã thành công trong việc phát triển một hệ thống học tập trực tuyến hiện đại, ứng dụng các công nghệ tiên tiến để số hóa và minh bạch hóa quá trình giáo dục.

Việc sử dụng **NFT làm chứng chỉ học tập** không chỉ đảm bảo tính xác thực, bảo mật mà còn tạo ra minh chứng trực quan, sinh động về sự tiến bộ và nỗ lực của người học trong suốt quá trình học tập.

Đặc biệt, khả năng **cập nhật metadata theo thành tích học tập** mở ra một hướng đi mới cho việc đánh giá, chứng nhận năng lực một cách linh hoạt, cá nhân hóa và phù hợp với xu thế phát triển của thời đại kỹ thuật số.

Hệ thống không chỉ đáp ứng được các yêu cầu kỹ thuật và trải nghiệm người dùng mà còn mang tính ứng dụng thực tiễn cao, góp phần thúc đẩy chuyển đổi số trong lĩnh vực giáo dục.

# ****TÀI LIỆU THAM KHẢO****

Tài liệu kỹ thuật AlphaBot – Waveshare

Datasheet cảm biến DHT11, MPU6050

Tài liệu ESP32: https://docs.espressif.com

Firebase Documentation – https://firebase.google.com/docs

MQTT Protocol – <https://mqtt.org>

Internet of Things: Principles and Paradigms – Rajkumar Buyya

·