

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



LÊ QUANG LINH - 51800423
NGUYỄN TRẦN HẢI YÊN - 51800520

**ÁP DỤNG
CÔNG NGHỆ BLOCKCHAIN VÀO
VIỆC THANH TOÁN HỌC PHÍ QUA
CRYPTO**

DỰ ÁN CÔNG NGHỆ THÔNG TIN

KỸ THUẬT PHẦN MỀM

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



LÊ QUANG LINH - 51800423
NGUYỄN TRẦN HẢI YẾN - 51800520

ÁP DỤNG
CÔNG NGHỆ BLOCKCHAIN VÀO
VIỆC THANH TOÁN HỌC PHÍ QUA
CRYPTO

DỰ ÁN CÔNG NGHỆ THÔNG TIN

KỸ THUẬT PHẦN MỀM

Người hướng dẫn
ThS. Phạm Thái Kỳ Trung

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Để hoàn thành dự án này, đầu tiên chúng tôi xin gửi lời cảm ơn sâu sắc đến Thầy Phạm Thái Kỳ Trung – người đã giúp chúng tôi từng bước hoàn thành dự án, hướng dẫn chúng tôi từ những bước đầu tiên, gợi ý cho chúng tôi những ý tưởng, lên kế hoạch thực hiện dự án, góp ý những lỗi sai, đưa ra cách giải quyết những vấn đề gặp phải trong quá trình thực hiện dự án, đến những bước cuối cùng hoàn thiện dự án. Sự tận tình của Thầy đã giúp dự án đạt được kết quả như ngày hôm nay.

Chúng tôi cũng xin gửi lời cảm ơn đến Khoa Công Nghệ Thông Tin đã giúp đỡ, hỗ trợ nhiệt tình trong quá trình xét duyệt, đăng ký môn học, giải quyết các vấn đề học vụ và các giấy tờ một cách nhanh chóng, hướng dẫn và thông báo kịp thời các vấn đề phát sinh trong môn học. Nhờ đó mà chúng tôi mới có thể hoàn thành môn học theo đúng các mốc thời gian, làm tốt các quy trình liên quan đến môn học.

Cuối cùng, chúng tôi xin cảm ơn trường Đại học Tôn Đức Thắng đã cho chúng tôi một môi trường học tập thân thiện, hiện đại và tiện nghi; dạy cho chúng tôi những kiến thức chuyên ngành và những kiến thức về xã hội; rèn luyện cho chúng tôi những tác phong chuyên nghiệp giúp ích cho chúng tôi trong quá trình học tập tại trường, cũng như giúp ích cho công việc trong tương lai. Và chúng tôi xin cảm ơn sự giúp đỡ, hỗ trợ của các anh/chị, bạn bè trong suốt những năm học đại học.

Trong quá trình thực hiện dự án và báo cáo sẽ không tránh khỏi những thiếu sót, chúng tôi rất mong nhận được sự thông cảm, góp ý và chỉ dạy tận tình từ quý Thầy/Cô.

TP. Hồ Chí Minh, ngày ... tháng ... năm 20..

Tác giả

(Ký tên và ghi rõ họ tên)

Lê Quang Linh

Nguyễn Trần Hải Yến

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của ThS. Phạm Thái Kỳ Trung. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày ... tháng ... năm 20..

Tác giả

(Ký tên và ghi rõ họ tên)

Lê Quang Linh

Nguyễn Trần Hải Yến

ÁP DỤNG CÔNG NGHỆ BLOCKCHAIN VÀO VIỆC THANH TOÁN HỌC PHÍ QUA CRYPTO

TÓM TẮT

Trong thời đại công nghệ phát triển, cuộc Cách mạng công nghiệp 4.0 vẫn đang diễn ra. Điều này đã tác động mạnh mẽ đến sự ra đời và sự phát triển vượt bậc của nhiều công nghệ cũng như áp dụng chúng vào phần lớn các lĩnh vực. Cùng với sự phát triển của các công nghệ như Dữ liệu lớn (Big Data), Trí tuệ nhân tạo (AI), công nghệ tài chính (Fintech),... Công nghệ Blockchain là một trong những công nghệ đang có những phát triển nhanh chóng, có thể coi là một trong những công nghệ tiên phong trong quá trình chuyển đổi số và xây dựng nền tảng công nghệ trong tương lai. Ứng dụng tiêu biểu nhất của công nghệ Blockchain là xây dựng nền tảng thanh toán phi tập trung nhanh chóng, tiện lợi và bảo mật. Đề tài "Áp dụng công nghệ Blockchain vào việc thanh toán học phí qua Crypto" là một ví dụ về việc áp dụng công nghệ Blockchain vào các ứng dụng thực tế.

Trong bài báo cáo này sẽ giới thiệu về một số công nghệ, nền tảng được dùng trong hệ thống như Công nghệ Blockchain, giới thiệu về Crypto, ngôn ngữ Solidity, thư viện ReactJS, NodeJS Platform,... Báo cáo cũng trình bày về quá trình triển khai tạo nên hệ thống, cách áp dụng các công nghệ vào quy trình thanh toán học phí thông qua các nền tảng, các lỗi thường gặp, những thiếu sót còn tồn tại trong hệ thống, cũng như các kết quả đã đạt được.

APPLYING BLOCKCHAIN TECHNOLOGY TO TUITION PAYMENT BY CRYPTO

ABSTRACT

In the era of technological development, the Industrial Revolution 4.0 is still happening. This has a strong impact on the birth and great development of many technologies as well as their application in most fields. Along with the development of technologies such as Big Data, Artificial Intelligence (AI), Fintech, ... Blockchain technology is one of the technologies that are developing rapidly, which can be considered as one of the pioneering technologies in the process of digital transformation and building technology platforms of the future. The most typical application of Blockchain technology is to build a fast, convenient and secure decentralized payment platform. The topic "Applying Blockchain Technology to tuition payment by Crypto" is an example of applying Blockchain technology to practical applications.

In this report, we will introduce some of the technologies and platforms used in the system such as Blockchain technology, introduction to Crypto, Solidity language, ReactJS Library, NodeJS Platform, ... The report also presents the implementation process that makes up the system, how to apply technologies to the tuition payment process through the platforms, common errors and shortcomings in the system, as well as as the results obtained.

MỤC LỤC

DANH MỤC HÌNH VẼ	vi
DANH MỤC BẢNG BIỂU	ix
DANH MỤC CÁC CHỮ VIẾT TẮT.....	x
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài.....	1
1.2 Mục tiêu thực hiện đề tài.....	1
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1 Blockchain.....	3
2.1.1 <i>Định nghĩa Blockchain</i>	3
2.1.2 <i>Hoạt động của Blockchain</i>	3
2.1.3 <i>Ứng dụng của Blockchain</i>	4
2.1.4 <i>Lợi ích và hạn chế của Blockchain</i>	7
2.2 Smart Contract	10
2.2.1 <i>Khái niệm</i>	10
2.2.2 <i>Đặc điểm</i>	10
2.2.3 <i>Cách thức hoạt động</i>	11
2.2.4 <i>Ưu điểm</i>	11
2.2.5 <i>Nhược điểm</i>	12
2.3 Crypto.....	12
2.3.1 <i>Định nghĩa Crypto</i>	12
2.3.2 <i>Cách thức hoạt động của Crypto</i>	13
2.3.3 <i>Phân loại Crypto</i>	13

2.3.4 Một số đồng <i>Crypto</i> phổ biến.....	14
2.3.5 Ưu điểm và nhược điểm của <i>Crypto</i>	15
2.4 Một số nền tảng hỗ trợ	16
2.4.1 <i>Solidity</i>	16
2.4.2 <i>ReactJS</i>	19
2.4.3 <i>NodeJS</i>	21
CHƯƠNG 3. PHƯƠNG PHÁP NGHIÊN CỨU	23
3.1 Sơ đồ Use case tổng quát	23
3.2 Tổng quan về mô hình ứng dụng công nghệ blockchain vào hệ thống	26
3.3 Quy trình thực hiện	28
3.3.1 Tạo smart contract với <i>Remix IDE</i> :	28
3.3.2 Deploy smartcontract lên mạng <i>Etherum testnet</i>	31
3.3.3 Xây dựng database với <i>mySQL</i>	33
3.3.4 Sử dụng <i>NodeJs</i> để tương tác với database và smart contract.....	37
3.3.5 Sử dụng <i>ReactJS</i> và <i>MUI</i> để xây dựng giao diện.....	43
CHƯƠNG 4. THỰC NGHIỆM	50
4.1 Cài đặt thực nghiệm	50
4.2 Kết quả thực nghiệm:	53
CHƯƠNG 5. KẾT LUẬN.....	63
5.1 Kết luận	63
5.2 Hướng phát triển	64
TÀI LIỆU THAM KHẢO	65

DANH MỤC HÌNH VẼ

Hình 2.1: Máy ảo Ethereum	17
Hình 2.2: Version Pragma	17
Hình 2.3: Khai báo Contract	18
Hình 2.4: Khai báo biến	18
Hình 2.5: Khai báo hàm	18
Hình 2.6: JSX	20
Hình 2.7: Virtual DOM	20
Hình 2.8: Kiến trúc NodeJS	22
Hình 3.1: Sơ đồ Use Case	23
Hình 3.2: Sơ đồ tổng quan ứng dụng Blockchain vào hệ thống	26
Hình 3.3: Compile file TDTU_Tuition.sol	31
Hình 3.4: Deploy Smart Contract.....	32
Hình 3.5: Ký Transaction.....	32
Hình 3.6: Giao diện Smart Contract.....	33
Hình 3.7: Mã JSON.....	33
Hình 3.8: Database	34
Hình 3.9: Router.....	37
Hình 3.10 API	38
Hình 3.11: Admin.....	39
Hình 3.12: Tuition	40
Hình 3.13: Wallet	41
Hình 3.14: Student	42

Hình 3.15: Validation.....	43
Hình 3.16: Interactions.....	44
Hình 3.17: Kết nối ví.....	45
Hình 3.18: Token Contract.....	46
Hình 3.19: Tiến hành thanh toán.....	48
Hình 4.1: Cài đặt thực nghiệm	50
Hình 4.2: Import Database	51
Hình 4.3: Xem tất cả các bảng	51
Hình 4.4: Chạy lệnh backend	52
Hình 4.5: Chạy lệnh Frontend	52
Hình 4.6: Trang đăng nhập.....	53
Hình 4.7: Chọn tài khoản đăng nhập.....	53
Hình 4.8: Trang xem học phí	54
Hình 4.9: Học phí đã thanh toán	54
Hình 4.10: Học chưa được thanh toán	55
Hình 4.11: Kết nối ví thanh toán	55
Hình 4.12: Chuyển VND sang USDT	56
Hình 4.13: Xác nhận ký giao dịch.....	56
Hình 4.14: Xem lịch sử giao dịch	57
Hình 4.15: Email xác nhận.....	57
Hình 4.16: Xem thống kê học phí	58
Hình 4.17: Quản lý quản trị viên.....	58
Hình 4.18: Thêm quản trị viên	59

Hình 4.19: Sửa thông tin quản trị viên.....	59
Hình 4.20: Xóa quản trị viên.....	60
Hình 4.21: Quản lý ví nhận tiền	60
Hình 4.22: Thêm ví nhận tiền	61
Hình 4.23: Kích hoạt ví	61
Hình 4.24: Xóa ví nhận tiền	62
Hình 4.25: Xem lịch sử giao dịch	62

DANH MỤC BẢNG BIỂU

Bảng 3.1: Tác nhân của hệ thống	23
Bảng 3.2: Use Case của hệ thống.....	24

DANH MỤC CÁC CHỮ VIẾT TẮT

API	Application Programming Interface
BNB	Binance Coin
BTC	Bitcoin
DeFi	Decentralized Finance
DNS	Domain Name System
DOM	Document Object Model
ETC	Ethereum
EVM	Ethereum Virtual Machine
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JSX	JavaScript Extension
KYC	Know Your Customer
LTH	Litecoin
MVC	Model View Controller
NFT	Non-fungible Token
UC	Use Case
USDT	Tether

XML Extensible Markup Language

XRP Ripple

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Hình thức thanh toán điện tử đang là hình thức thanh toán phổ biến và ngày càng được áp dụng rộng rãi trong các lĩnh vực ngày nay như mua sắm trực tuyến trên các sàn thương mại điện tử, thanh toán các dịch vụ giải trí, du lịch, thanh toán các loại hóa đơn, thanh toán học phí, ... Thanh toán học phí đang là vấn đề được quan tâm hiện nay, đặc biệt là các trường đại học với số lượng sinh viên lên đến hàng chục nghìn. Với sự phát triển của công nghệ và Internet, sinh viên hiện nay có xu hướng thanh toán học phí trực tuyến qua các cổng thanh toán, điều này có thể dẫn đến trường hợp quá tải cho các hệ thống ngân hàng cũng như hệ thống thu học phí của các trường đại học. Cùng với đó, việc thanh toán điện tử cũng tồn tại những rủi ro như thông tin tài khoản ngân hàng có thể bị đánh cắp, các hacker có thể tấn công để chiếm đoạt tài khoản với mục đích xấu, các khoản thanh toán có thể đến trễ nếu giao dịch vào cuối tuần, ... Chính vì vậy, mà sự ra đời của một công nghệ mang tên Blockchain có thể giải quyết phần lớn các vấn đề mà thanh toán điện tử qua các ngân hàng đang gặp phải. Với tính bảo mật cao, Blockchain được ví như "vệ sĩ mới" cho hệ thống bảo mật trong ngành tài chính ngân hàng. Blockchain đã và đang phát triển, dần trở thành một trong những công nghệ hàng đầu trong lĩnh vực tài chính ngân hàng.

Nhận thấy được thực trạng này, sau quá trình nghiên cứu về công nghệ Blockchain, Crypto, và các công nghệ liên quan, với mong muốn giải quyết được các vấn đề đã nêu trên, chúng tôi quyết định chọn đề tài "Áp dụng công nghệ Blockchain vào việc thanh toán học phí qua Crypto" cho dự án lần này.

1.2 Mục tiêu thực hiện đề tài

Crypto (hay còn gọi là tiền ảo) là một từ khóa không còn quá xa lạ với các nhà đầu tư trong những năm trở lại đây. Crypto không chịu sự quản lý của bất kỳ tổ chức nào, vì vậy mà các nhà đầu tư sẽ tránh được sự chi phối hoặc kiểm soát trong các giao dịch, chi phí giao dịch thấp (gần như là bằng 0) và thời gian xử lý giao dịch nhanh chóng. Đồng tiền Crypto không bị lạm phát và làm giả, do đó mà Crypto không bị

lạm phát như tiền pháp định. Bên cạnh những ưu điểm nổi bật, Crypto cũng có những hạn chế như gây rủi ro cho các nhà đầu tư khi xảy ra biến động giá mạnh. Dù được biết đến rộng rãi, nhưng Crypto hiện nay vẫn chưa được công nhận ở nhiều quốc gia và khu vực. Việc khai thác và đầu tư Crypto đòi hỏi cần có sự hiểu biết sâu rộng về lĩnh vực này.

Mục tiêu của đề tài "Áp dụng công nghệ Blockchain vào việc thanh toán học phí qua Crypto" nhằm mang lại những kiến thức cơ bản, giúp người đọc hiểu rõ hơn về công nghệ Blockchain, Crypto, xóa bỏ những định kiến về khái niệm tiền ảo. Để cho thấy được lợi ích của Crypto, chúng tôi quyết định xây dựng một hệ thống thanh toán học phí bằng Ctypto thông qua việc áp dụng công nghệ Blockchain. Thanh toán học phí bằng tiền ảo giúp sinh viên có thêm lựa chọn thanh toán, giúp giảm tải các giao dịch cho ngân hàng và hệ thống của trường vào những đợt thanh toán học phí. Các trường có thể chủ động, nhanh chóng trong việc quản lý dòng tiền từ nguồn tiền được gửi từ sinh viên, việc cập nhật thông tin chi tiết về các giao dịch cũng trở nên nhanh chóng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Blockchain

2.1.1 Định nghĩa Blockchain

Blockchain là cơ sở dữ liệu phân tán hoặc số cái được chia sẻ giữa các nút (node) của mạng máy tính. Chúng được biết đến nhiều nhất với vai trò quan trọng trong các hệ thống tiền điện tử để duy trì hồ sơ giao dịch an toàn và phi tập trung, nhưng chúng không giới hạn trong việc sử dụng tiền điện tử. Blockchain có thể được sử dụng để làm cho dữ liệu trong bất kỳ ngành nào trở nên bất biến - thuật ngữ dùng để mô tả tính không thể thay đổi.

Vì không có cách nào để thay đổi một khối (block) nên độ tin cậy duy nhất cần có là ở thời điểm người dùng hoặc chương trình nhập dữ liệu. Khía cạnh này làm giảm nhu cầu về các bên thứ ba đáng tin cậy, thường là kiểm toán viên hoặc những người khác làm tăng thêm chi phí và mắc sai lầm.

Kể từ khi Bitcoin được giới thiệu vào năm 2009, việc sử dụng Blockchain đã bùng nổ thông qua việc tạo ra nhiều loại tiền điện tử, ứng dụng tài chính phi tập trung (DeFi), mã thông báo không thể thay thế (NFT) và hợp đồng thông minh (Smart Contract).

Các tính chất của công nghệ Blockchain: Tính phi tập trung, tính phân tán, tính không thể thay đổi, tính bảo mật, tính minh bạch, tích hợp Smart Contract.

2.1.2 Hoạt động của Blockchain

Blockchain có phần tương tự với bảng tính hoặc cơ sở dữ liệu vì nó là nơi thông tin được nhập và lưu trữ. Nhưng điểm khác biệt chính giữa cơ sở dữ liệu hoặc bảng tính truyền thống và Blockchain là cách dữ liệu được cấu trúc và truy cập.

Một Blockchain bao gồm các chương trình được gọi là tập lệnh thực hiện các tác vụ bạn thường làm trong cơ sở dữ liệu: Nhập và truy cập thông tin cũng như lưu trữ nó ở đâu đó. Một blockchain được phân phối, có nghĩa là nhiều bản sao được lưu trên nhiều máy và tất cả chúng phải khớp nhau thì nó mới hợp lệ.

Blockchain thu thập thông tin giao dịch và nhập nó vào một khối, giống như một ô trong bảng tính chứa thông tin. Sau khi đầy, thông tin sẽ được chạy qua thuật toán mã hóa, thuật toán này tạo ra số thập lục phân gọi là hàm băm.

Sau đó, hàm băm được nhập vào khối sau và được mã hóa bằng thông tin khác trong khối. Điều này tạo ra một loạt các khối được nối với nhau.

2.1.3 Ứng dụng của Blockchain

2.1.3.1 Lĩnh vực Ngân hàng và Tài chính

Tài chính - Ngân hàng có lẽ là lĩnh vực được hưởng lợi nhiều nhất bằng việc áp dụng công nghệ Blockchain vào hệ thống. Các giao dịch thường bị trễ nếu được tiến hành vào cuối tuần. Ngay cả khi gửi tiền trong giờ làm việc, giao dịch vẫn có thể mất nhiều thời gian để xác minh do khối lượng giao dịch khổng lồ mà ngân hàng cần phải giải quyết.

Bằng cách tích hợp Blockchain vào hệ thống ngân hàng, áp dụng các phép toán ngang hàng (Peer - To - Peer), Blockchain có thể giải quyết các giao dịch gần như theo thời gian thực (Realtime), người tiêu dùng có thể thấy các giao dịch của họ được xử lý trong vài phút hoặc vài giây, giải quyết giao dịch 24/7, bất kể thời gian nào trong ngày, trong tuần, kể cả ngày lễ. Với Blockchain, các ngân hàng cũng có cơ hội trao đổi tiền giữa các tổ chức nhanh chóng và an toàn hơn, giảm các chi phí và các rủi ro trong giao dịch, tăng tính minh bạch và khả năng truy xuất nguồn gốc.

Một lĩnh vực khác có thể hưởng lợi nhiều không kém lĩnh vực Tài chính - Ngân hàng đó là lĩnh vực Tín dụng. Công nghệ Blockchain cho phép các giao dịch truyền thống được thay thế bằng các giao dịch thông minh, từ đó giảm chi phí giao dịch cho vay và tài chính kinh doanh. Hoạt động tín dụng tồn tại một số vấn đề khó khăn như chất lượng thông tin kém và dữ liệu khan hiếm, bất cập trong việc chia sẻ thông tin cho các bên liên quan, các vấn đề về quyền riêng tư và bảo mật. Blockchain có thể giải quyết các khó khăn này thông qua ứng dụng định danh khách hàng (KYC), sử dụng công nghệ mã hóa để lưu trữ thông tin trong hệ thống Blockchain. Các yêu

cầu truy vấn có thể được thực hiện qua việc thông báo cho các nhà cung cấp dữ liệu bởi Blockchain. Do đó, tất cả các bên có thể tìm kiếm dữ liệu lớn bên ngoài mà không phải tiết lộ dữ liệu kinh doanh cốt lõi.

2.1.3.2 Tiền tệ

Tiền tệ của người dùng phụ thuộc vào ngân hàng hoặc chính phủ quốc gia của họ. Người dùng có thể bị đánh cắp thông tin cá nhân nếu ngân hàng của họ bị tấn công, hoặc người dùng sống ở một quốc gia có chính phủ không ổn định, giá trị đồng tiền của người dùng có thể gặp rủi ro. Đây là những lí do thúc đẩy sự ra đời của đồng tiền điện tử.

Blockchain có thể nói là nền tảng cho các loại tiền điện tử như Bitcoin (BTC), Ethereum (ETC), Binance Coin (BNB), ... Bằng cách trải rộng hoạt động của mình trên một mạng máy tính, Blockchain cho phép các loại tiền điện tử hoạt động mà không cần cơ quan trung ương. Điều này không chỉ làm giảm rủi ro mà còn giảm phí xử lý và giao dịch. Blockchain cung cấp một loại tiền tệ ổn định hơn với nhiều ứng dụng, một mạng lưới rộng lớn của các cá nhân và tổ chức kinh doanh trong nước và quốc tế cho những người dùng ở các quốc gia có tiền tệ không ổn định. Công dân của các quốc gia bị chiến tranh tàn phá hoặc có chính phủ thiếu cơ sở hạ tầng cung cấp nhận dạng có thể không có quyền truy cập vào tài khoản tiết kiệm, không thể lưu trữ tài sản một cách an toàn. Ví điện tử cho tài khoản tiết kiệm như một phương tiện giúp ích cho việc thanh toán đối với những người không có nhận dạng nhà nước.

2.1.3.3 Y tế

Blockchain có thể được các nhà cung cấp dịch vụ chăm sóc sức khỏe tận dụng để lưu trữ hồ sơ y tế của bệnh nhân một cách an toàn. Blockchain cung cấp cho bệnh nhân bằng chứng và sự tự tin rằng hồ sơ không thể thay đổi bằng cách viết vào chuỗi khỏi các hồ sơ y tế đã được tạo và ký. Blockchain cũng đảm bảo quyền riêng tư bằng cách mã hóa và lưu trữ các hồ sơ sức khỏe cá nhân trên Blockchain bằng khóa riêng.

2.1.3.4 Hồ sơ tài sản

Tại Văn phòng Ghi chép địa phương, nhân viên chính phủ phải nhập thủ công các chứng thư thực tế hay ghi lại quyền sở hữu tài sản vào cơ sở dữ liệu trung tâm. Trong trường hợp xảy ra tranh chấp tài sản, các yêu cầu về tài sản phải được đối chiếu với cơ sở dữ liệu. Quá trình này khá tốn kém và mất thời gian, còn dễ xảy ra lỗi của con người, mỗi sai sót đều khiến việc theo dõi quyền sở hữu tài sản trở nên kém hiệu quả hơn. Nhu cầu quét tài liệu và theo dõi các tệp vật lý trong văn phòng ghi chép địa phương có thể được loại bỏ bởi Blockchain. Chủ sở hữu có thể tin tưởng rằng chứng thư của họ là chính xác và được ghi lại vĩnh viễn, nếu quyền sở hữu tài sản được lưu trữ và xác minh trên Blockchain.

Việc chứng minh quyền sở hữu tài sản gần như là không thể ở các quốc gia hoặc khu vực bị chiến tranh tàn phá có ít hoặc không có cơ sở hạ tầng chính phủ hoặc tài chính và không có Văn phòng Lưu trữ. Các mốc thời gian minh bạch và rõ ràng về quyền sở hữu tài sản có thể được thiết lập bằng cách sử dụng Blockchain bởi những người sống ở khu vực đó.

2.1.3.5 Smart Contract (Hợp đồng thông minh)

Hợp đồng thông minh cũng giống như một hợp đồng pháp lý truyền thống nhưng được ghi lại dưới dạng ngôn ngữ máy tính. Hợp đồng thông minh mô tả khả năng tự đưa ra các điều khoản và thực thi thỏa thuận trên cơ sở hệ thống máy tính bằng công nghệ Blockchain.

Các doanh nghiệp tự động hóa các quy trình được cài đặt trước, giảm chi phí hoạt động bằng cách sử dụng hợp đồng thông minh. Thực hiện tự động toàn bộ quá trình mà không có sự can thiệp từ bên ngoài, không cần gặp mặt trực tiếp, hạn chế sự tương tác giữa con người với dữ liệu của công ty. Vì thế mà đảm bảo được tính minh bạch, có thể truy xuất và giảm khả năng đánh cắp hoặc làm mất dữ liệu.

2.1.3.6 Chuỗi cung ứng

Hàng hóa thường được di chuyển từ nơi này đến nơi khác ở quy mô trên toàn thế giới. Người tiêu dùng rất quan tâm đến nguồn gốc và các thông tin khác liên quan

đến sản phẩm. Với các phương pháp lưu trữ dữ liệu truyền thống, khó có thể tìm ra được nguồn gốc của vấn đề, chẳng hạn như hàng kém chất lượng đến từ nhà cung cấp nào.

Công nghệ Blockchain sẽ cung cấp thông tin về nguồn gốc hàng hóa, quá trình từ khi thành phẩm cho đến tay người tiêu dùng. Thông tin rõ ràng, minh bạch của chuỗi cung ứng được cải thiện dẫn đến việc nâng cao sự tin tưởng của người tiêu dùng.

2.1.3.7 Bỏ phiếu

Công nghệ Blockchain góp phần xây dựng một hệ thống bỏ phiếu hiện đại. Việc bỏ phiếu bằng Blockchain làm giảm khả năng gian lận trong bầu cử và tăng tỷ lệ cử tri đi bỏ phiếu, phiếu bầu gần như không thể bị giả mạo. Tính minh bạch trong quá trình bầu cử cũng sẽ được nâng cao, giảm nhân sự cần thiết trong quá trình bầu cử và cung cấp cho các quan chức kết quả gần như ngay lập tức bởi việc sử dụng Blockchain. Điều này sẽ loại bỏ nhu cầu kiểm lại phiếu hoặc bất kỳ mối lo ngại về việc gian lận có thể đe dọa đến cuộc bầu cử.

2.1.4 Lợi ích và hạn chế của Blockchain

2.1.4.1 Lợi ích

Độ chính xác của chuỗi: Giao dịch trên Blockchain loại bỏ sự tham gia của con người khỏi quá trình xác minh, dẫn đến ít lỗi do con người gây ra hơn và ghi lại thông tin chính xác hơn. Trường hợp một máy tính trên mạng mắc lỗi tính toán, lỗi sẽ chỉ xảy ra với một bản sao của chuỗi khối và không được phần còn lại của mạng chấp nhận.

Giảm chi phí: Blockchain loại bỏ nhu cầu xác minh của bên thứ ba và các chi phí liên quan.

Phân cấp: Blockchain không lưu trữ thông tin ở vị trí trung tâm. Thay vào đó, Blockchain được sao chép và lan truyền trên mạng máy tính. Khi một khối mới được thêm vào chuỗi khối, mọi máy tính trên mạng sẽ cập nhật chuỗi khối để phản ánh sự thay đổi. Vì điều này mà Blockchain trở nên khó giả mạo hơn.

Giao dịch hiệu quả: Các giao dịch có thể được hoàn thành nhanh chóng trong vài phút. Điều này đặc biệt hữu ích cho các giao dịch xuyên biên giới, thường mất nhiều thời gian hơn do vấn đề về múi giờ và thực tế là tất cả các bên đều phải xác nhận xử lý thanh toán.

Giao dịch riêng tư: Một số mạng Blockchain hoạt động như cơ sở dữ liệu công cộng, tức là bất kỳ ai có kết nối internet đều có thể xem danh sách lịch sử giao dịch của mạng. Người dùng có thể truy cập chi tiết giao dịch nhưng không thể truy cập thông tin nhận dạng về người dùng thực hiện các giao dịch đó.

Giao dịch an toàn: Sau khi giao dịch được ghi lại, mạng Blockchain xác minh tính xác thực của nó. Sau khi giao dịch được xác thực, nó sẽ được thêm vào khối Blockchain. Mỗi khối trên Blockchain chứa hàm băm duy nhất của nó và hàm băm duy nhất của khối trước nó. Do đó, các khối không thể bị thay đổi sau khi mạng xác nhận chúng.

Tính minh bạch: Phần lớn các Blockchain đều là phần mềm mã nguồn mở. Do đó mọi người có thể xem mã của nó. Các loại tiền điện tử có thể được xem xét bởi các kiểm toán viên để đảm bảo an toàn. Điều này cũng có nghĩa là không có thẩm quyền thực sự nào về việc ai kiểm soát mã Bitcoin hoặc cách nó được chỉnh sửa. Vì vậy bất kỳ ai cũng có thể đề xuất thay đổi hoặc nâng cấp hệ thống. Bitcoin có thể được cập nhật nếu phần lớn người dùng mạng đồng ý với phiên bản mới của mã.

Ngân hàng thay thế: Một lợi ích khác của Blockchain và tiền điện tử là tạo điều kiện cho bất kỳ ai, không phân biệt sắc tộc, giới tính, địa lý, ... đều có thể sử dụng được tiền điện tử. Ở các quốc gia đang phát triển, nền kinh tế còn yếu kém, một số người trưởng thành không có tài khoản ngân hàng hoặc phương tiện cất giữ tài sản. Do đó, những người này phụ thuộc phần lớn vào tiền mặt. Điều này khiến việc lưu trữ, cất giữ tài sản không được đảm bảo an toàn, dễ dàng bị trộm cắp. Tiền điện tử có tính an toàn hơn, gây khó khăn cho những người có ý định đánh cắp tài sản. Các Blockchain đang tìm kiếm các giải pháp không chỉ để trở thành một đơn vị lưu trữ tài sản mà còn để lưu trữ hồ sơ y tế, quyền tài sản và nhiều hợp đồng pháp lý khác trong tương lai.

2.1.4.2 Hạn chế

Chi phí công nghệ: Mặc dù Blockchain có thể giúp người dùng tiết kiệm chi phí giao dịch nhưng công nghệ này không hoàn toàn miễn phí. Hệ thống mà Bitcoin sử dụng để xác thực các giao dịch, tiêu thụ một lượng lớn sức mạnh tính toán. Dẫn đến việc xuất hiện những giải pháp cho vấn đề này như thiết lập các trang trại khai thác Bitcoin sử dụng nguồn năng lượng từ mặt trời, khí đốt tự nhiên dư thừa từ các trang web fracking, năng lượng từ trạng trại gió.

Tốc độ và dữ liệu kém hiệu quả: Quá trình Blockchain xét duyệt các giao dịch phụ thuộc vào một mạng lớn hơn, do đó bị giới hạn về tốc độ di chuyển. Ví dụ, Bitcoin chỉ có thể xử lý 4,6 giao dịch trên mỗi giây, còn Visa thì lên đến 1.700 giao dịch trên mỗi giây. Sự gia tăng số lượng giao dịch có thể gây ra các vấn đề về tốc độ mạng. Một vấn đề cấp bách khác là về kích thước khối và khả năng mở rộng chuỗi khối, do mỗi khối chỉ có thể chứa một lượng dữ liệu nhất định.

Hoạt động bất hợp pháp: Blockchain bảo vệ người dùng khỏi bị hack và bảo vệ quyền riêng tư bởi tính bảo mật, nhưng nó cũng có thể bị lợi dụng bởi các tội phạm để thực hiện các hoạt động và giao dịch bất hợp pháp trên mạng Blockchain. Các giao dịch bất hợp pháp trên Blockchain sẽ khó theo dõi hơn so với theo dõi các giao dịch ở ngân hàng. Ví dụ được trích dẫn nhiều nhất về việc sử dụng Blockchain cho các giao dịch bất hợp pháp có lẽ là Con đường tơ lụa, một thị trường rửa tiền và ma túy bất hợp pháp trên web đen trực tuyến hoạt động từ tháng 2 năm 2011 đến tháng 10 năm 2013, khi FBI đóng cửa nó.

Quy định: Quy định của chính phủ đối với tiền điện tử vẫn là một mối lo ngại đối với nhiều người dùng trong lĩnh vực tiền điện tử. Mạng lưới phi tập trung của Bitcoin ngày càng phát triển, việc chấm dứt Bitcoin ngày một khó khăn hơn, nhưng các chính phủ có thể xem việc sở hữu tiền điện tử là bất hợp pháp.

Rủi ro mất tài sản: Tài sản kỹ thuật số được bảo mật bằng khóa mật mã, ví dụ như tiền điện tử trong ví Blockchain. Không thể khôi phục khóa mật mã cho phép truy cập vào tài sản nếu chủ sở hữu tài sản kỹ thuật số làm mất nó, vì hệ thống phi

tập trung, không thể khôi phục hay yêu cầu lấy lại quyền truy cập từ cơ quan trung ương hoặc ngân hàng, do đó tài sản sẽ bị mất vĩnh viễn.

2.2 Smart Contract

2.2.1 Khái niệm

Hợp đồng thông minh (Smart Contract) là một chương trình thực hiện tự động hóa các hành động cần thiết trong một thỏa thuận hoặc hợp đồng. Sau khi hoàn thành, các giao dịch có thể được theo dõi và không thể đảo ngược.

Hợp đồng thông minh cho phép các giao dịch và thỏa thuận đáng tin cậy được thực hiện giữa các bên khác nhau, ẩn danh mà không cần cơ quan trung ương, hệ thống pháp luật hoặc cơ chế thực thi bên ngoài. Hợp đồng thông minh không chứa ngôn ngữ pháp lý, điều khoản hoặc thỏa thuận, chỉ có mã thực thi các hành động khi đáp ứng các điều kiện cụ thể.

2.2.2 Đặc điểm

Cũng giống với hợp đồng truyền thống, hợp đồng thông minh đưa ra các điều khoản thỏa thuận. Các điều khoản này được viết bằng ngôn ngữ lập trình dựa trên công nghệ Blockchain.

Một hợp đồng thông minh gồm có 4 yếu tố:

- Chủ thể hợp đồng: Là các bên tham gia trực tiếp vào hợp đồng. Các bên được cấp quyền truy cập, theo dõi quá trình thực thi hợp đồng.
- Điều khoản hợp đồng: Là các điều khoản, thỏa thuận được đặt ra và chấp thuận bởi các bên tham gia. Các điều này được quy định ở dạng chuỗi và lập trình đặc biệt.
- Chữ ký số: Khi tham gia hợp đồng thông minh, các bên phải thỏa thuận về chữ ký số và thực hiện các thao tác thông qua chữ ký số đó.
- Nền tảng phân quyền: Hợp đồng thông minh phải được tải lên Blockchain khi bước vào giai đoạn hoàn tất. Chuỗi Blockchain tiếp tục phân phối dữ liệu về các nút và lưu dữ liệu lại, không thể điều chỉnh.

2.2.3 Cách thức hoạt động

Hệ thống hợp đồng thông minh tuân theo các câu lệnh “If/when ... then ...”, nếu các điều kiện nhất định được thỏa mãn, hợp đồng thông minh sẽ thực thi một tác vụ cụ thể. Trong giai đoạn đầu, các điều khoản trong hợp đồng thông minh được viết dưới dạng ngôn ngữ lập trình. Sau đó, chúng sẽ được mã hóa và đánh dấu bằng một địa chỉ rồi được chuyển vào một khối thuộc Blockchain. Các nút đang hoạt động trên Blockchain sẽ phân phối và sao chép lại các dữ liệu được chuyển vào khối. Các dữ liệu lưu trữ trên Blockchain sẽ chờ đợi các điều kiện được kích hoạt. Khi được kích hoạt, các hợp đồng thông minh sẽ hoạt động và thực thi một số điều khoản đã được thỏa thuận trước đó và tự động kiểm tra quá trình thực hiện những cam kết được nêu trong hợp đồng.

Đối với mạng lưới Ethereum, trong khi những người dùng đang tiến hành các giao dịch và tương tác với nhau, các hợp đồng thông minh sẽ đảm nhiệm vai trò thực thi và quản lý những hoạt động diễn ra trên blockchain.

Trên thực tế, so với các nền tảng Blockchain khác, hợp đồng thông minh trên nền tảng Ethereum có nhiều điểm nổi trội hơn. Hợp đồng thông minh trên mạng lưới này gồm có một mã hợp đồng và hai Public Key. Public Key thứ nhất do người tạo hợp đồng cung cấp, Public Key thứ hai đại diện cho hợp đồng, có vai trò như một mã định danh kỹ thuật số duy nhất cho một hợp đồng thông minh riêng biệt.

2.2.4 Ưu điểm

Tương tự như ưu điểm của công nghệ Blockchain, ưu điểm của hợp đồng thông minh là loại bỏ sự tham gia của bên thứ ba. Một số ưu điểm khác của hợp đồng thông minh như:

Nhanh chóng và hiệu quả: Tốc độ thực hiện hợp đồng thông minh được tăng một cách đáng kể. Bởi vì quá trình thực hiện được tự động hóa, hợp đồng thông minh sẽ được thực hiện ngay lập tức nếu một điều kiện được đáp ứng.

Độ chính xác: Không có bên thứ ba tham gia vào quá trình thực hiện hợp đồng thông minh, do đó không thể xuất hiện lỗi của con người. Cùng với đó, các bản ghi

giao dịch sẽ được mã hóa và chia sẻ công khai giữa các bên tham gia. Vì vậy, hạn chế được ảnh hưởng của lợi ích cá nhân đối với hợp đồng.

Tính bảo mật: Đối với hợp đồng thông minh, người dùng chỉ có thể truy cập xem thông tin nhưng không thể chỉnh sửa bất cứ thông tin nào. Điều này cũng giúp bảo mật những dữ liệu quan trọng.

Tiết kiệm: Vì không cần đến sự giám sát của bên thứ ba, không phải tốn các chi phí in ấn, chuyển phát, lưu kho như hợp đồng truyền thống, hợp đồng thông minh có thể giảm đi các chi phí này trong quá trình vận hành. Hợp đồng thông minh cũng góp phần tiết kiệm chi phí hoạt động và nâng cao hiệu quả thông qua các chương trình phi tập trung và tự thực hiện (self – executing).

2.2.5 *Nhược điểm*

Không thể thay đổi: Do tính bảo mật được dựa trên công nghệ Blockchain, các dữ liệu chỉ có thể được xem bởi người dùng, rất khó hoặc không thể chỉnh sửa dữ liệu nếu xảy ra các sai sót, hay thay đổi các nội dung, thỏa thuận của hợp đồng.

Lỗi hỏng: Một số lỗi vẫn còn tồn tại bởi vì hợp đồng thông minh được con người viết ra bằng bộ mã máy tính. Do đó, hợp đồng thông minh có thể xảy ra các sự cố trong lúc vận hành hoặc có khả năng bị tấn công bởi những tội phạm với mục đích phi pháp.

Pháp lý: Một số quốc gia chưa có chính sách pháp lý rõ ràng cũng như cách thức quản lý hợp đồng thông minh, các bên chủ thể sẽ không được đảm bảo quyền lợi nếu có lỗi xảy ra.

2.3 Crypto

2.3.1 *Định nghĩa Crypto*

Crypto (Cryptocurrency) còn được gọi là tiền điện tử, là bất kỳ dạng tiền tệ nào tồn tại dưới dạng kỹ thuật số hoặc ảo và sử dụng mật mã để bảo mật các giao dịch. Tiền điện tử không có cơ quan quản lý hoặc phát hành trung tâm, thay vào đó sử dụng hệ thống phi tập trung để ghi lại các giao dịch và phát hành các đơn vị mới.

Tiền điện tử là một hệ thống ngang hàng có thể cho phép bất kỳ ai ở bất kỳ đâu gửi và nhận thanh toán. Thanh toán bằng tiền điện tử tồn tại hoàn toàn dưới dạng các mục nhập kỹ thuật số vào cơ sở dữ liệu trực tuyến mô tả các giao dịch cụ thể. Khi thanh toán bằng tiền điện tử, các giao dịch sẽ được ghi lại trong sổ cái công khai. Tiền điện tử được lưu trữ trong ví kỹ thuật số.

Tiền điện tử xác minh các giao dịch bằng cách sử dụng mã hóa để xác minh. Điều này có nghĩa là mã hóa nâng cao liên quan đến việc lưu trữ và truyền dữ liệu tiền điện tử giữa các ví và tới sổ cái công khai. Mục đích của mã hóa là hướng đến sự an toàn và bảo mật.

2.3.2 Cách thức hoạt động của Crypto

Tiền điện tử hoạt động dựa trên nền tảng Blockchain hay còn gọi là sổ cái công khai, đây là một bản ghi tất cả các giao dịch được cập nhật và nắm giữ bởi những người sở hữu tiền tệ.

Quá trình khai thác bao gồm việc sử dụng sức mạnh máy tính để giải các bài toán phức tạp, thông qua quá trình này tạo ra các đơn vị tiền điện tử. Mỗi một giao dịch sẽ được kiểm tra thông qua các thuật toán: Proof of Work, Proof of Stake, Proof of Authority. Hai cơ chế xác thực phổ biến được sử dụng nhiều cho phần lớn các loại tiền điện tử là Proof of Work và Proof of Stake.

Proof of Work: Cơ chế này đưa ra các bài toán cho các thợ đào tiền ảo cùng cạnh tranh nhau giải mã. Hệ thống nào có lời giải sớm hơn sẽ nhận được loại tiền ảo của hệ thống tham gia. Tuy vậy, quá trình này tiêu tốn rất nhiều tài nguyên và năng lượng máy tính.

Proof of Stake: Đây là cơ chế giúp tiết kiệm năng lượng hơn so với cơ chế Proof of Work. Số lượng tham gia xác minh được giới hạn thông qua cơ chế đặt cược (Staking). Các bên tham gia giao dịch có thể đặt cược tiền điện tử để xác minh. Nhóm xác thực sẽ được xếp vào nhóm giao dịch khi đã thu thập đủ cổ phần.

2.3.3 Phân loại Crypto

Crypto được chia làm hai loại:

- Coin: Là một loại tiền điện tử được phát hành, phát triển trên Blockchain riêng biệt. Coin được phát hành với mục đích giải quyết các vấn đề thanh toán, tài chính, bảo mật, phát triển ứng dụng, Mỗi mạng lưới Blockchain sẽ chỉ có 1 coin duy nhất. Ví dụ: BTC là đồng coin của Blockchain Bitcoin, ETH là đồng coin của Blockchain Ethereum, ...
- Token: Là loại tiền điện tử được phát hành trên Blockchain có sẵn, token không có Blockchain riêng biệt như coin. Ví dụ: Chainlink (LINK) là token được lưu trữ, giao dịch trên Blockchain Ethereum; Serum (SRM) là token được lưu trữ, giao dịch trên Blockchain Solana, Token có thể dùng làm phương thức thanh toán trong một dự án, do token là tài sản kỹ thuật số được phát hành bởi dự án.

2.3.4 Một số đồng Crypto phổ biến

Hiện nay, có đến hàng ngàn loại tiền điện tử. Một số đồng phổ biến hiện nay như:

Bitcoin (BTC): Được phát minh bởi một nhóm nhà phát triển ẩn danh gọi là Satoshi Nakamoto và được cho ra mắt vào năm 2009, Bitcoin là loại tiền điện tử đầu tiên trên thế giới. Do là đồng tiền ra đời đầu tiên và giữ vị trí đầu trong danh sách các loại tiền điện tử, Bitcoin hiện tại đang đạt mức cao nhất và được sử dụng phổ biến nhất.

Ethereum (ETH): Được ra đời vào năm 2013 – ra đời thứ 2 sau Bitcoin. Ethereum cũng là một nền tảng Blockchain có tiền điện tử riêng, được gọi là Ether (ETH) hoặc Ethereum. Đây là loại tiền điện tử phổ biến nhất chỉ sau Bitcoin. Ethereum có nhiều ứng dụng tiềm năng, đáng chú ý là “hợp đồng thông minh” (Smart Contract).

Tether (USDT): Là loại tiền điện tử được ra mắt vào năm 2014, có giá trị phản ánh giá trị của đồng đô la Mỹ (USD). Được biết đến với vai trò là “Stablecoin”, do tính ổn định và ít biến động, được sử dụng như đồng tiền trung gian.

Litecoin (LTH): Đây là loại tiền điện tử gần giống với Bitcoin nhưng được phát triển trên nền tảng khác với Bitcoin, được cải tiến về tốc độ tạo Blockchain nhanh hơn, quy trình thanh toán nhanh hơn để cho phép nhiều giao dịch hơn, ít hao tốn tài nguyên hơn.

Ripple (XRP): Được phát triển vào năm 2012 bởi công ty Ripple. Ripple cũng sử dụng công nghệ Blockchain tương tự như Bitcoin, nhưng khác nhau ở mục đích hoạt động. Khác với mục tiêu trở thành tiền điện tử có quy mô sử dụng toàn cầu của Bitcoin, Ripple hướng đến mục tiêu trở thành hệ thống thanh toán của các ngân hàng hay tổ chức tài chính thế giới, tạo ra các giải pháp thanh toán và chuyển đổi tiền tệ nhanh hơn cho các tổ chức này.

Các loại tiền điện tử không phải Bitcoin được gọi chung là “Altcoin”.

2.3.5 Ưu điểm và nhược điểm của Crypto

Ưu điểm của tiền điện tử:

- Tiền điện tử giúp trao đổi trực tiếp giữa hai bên mà không cần thông qua bên thứ ba nào.
- Tiền điện tử không thể làm giả vì nó không tồn tại dưới dạng vật chất, chúng được mã hóa và lưu trữ trong ví điện tử.
- Tiền điện tử có thể được giao dịch ở bất cứ thời điểm nào, bất cứ nơi đâu, giảm thiểu các chi phí giao dịch.
- Tiền điện tử có tính bảo mật cao và an toàn.

Nhược điểm của tiền điện tử:

- Ở một số quốc gia, tiền điện tử vẫn chưa được chấp nhận về mặt pháp lý và được sử dụng rộng rãi.
- Để sử dụng được tiền điện tử, đòi hỏi người dùng phải có kiến thức, hiểu biết về công nghệ.
- Tiền điện tử luôn có những biến động giá lớn.
- Tội phạm có thể thông qua giao dịch tiền điện tử để rửa tiền.

2.4 Một số nền tảng hỗ trợ

2.4.1 Solidity

2.4.1.1 Giới thiệu về Solidity

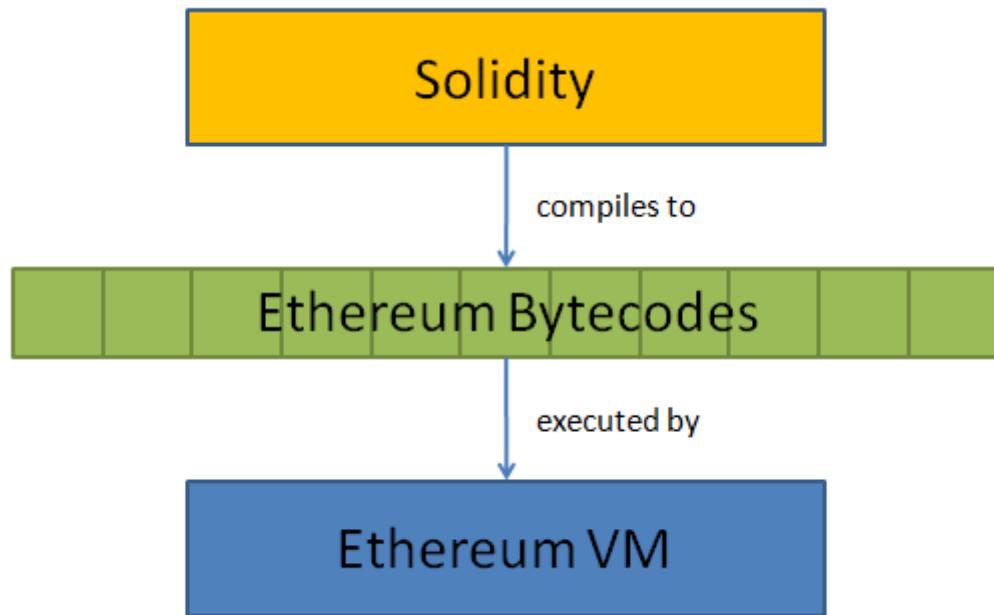
Solidity là ngôn ngữ lập trình hướng đối tượng được tạo riêng bởi nhóm Ethereum Network để xây dựng và thiết kế các hợp đồng thông minh trên nền tảng Blockchain.

- Solidity được sử dụng để tạo các hợp đồng thông minh triển khai logic kinh doanh và tạo ra chuỗi hồ sơ giao dịch trong hệ thống blockchain.
- Solidity hoạt động như một công cụ để tạo mã cấp máy và biên dịch nó trên Máy ảo Ethereum (EVM).
- Solidity có nhiều điểm tương đồng với C và C++ và khá đơn giản để học và hiểu.

Tương tự với các ngôn ngữ lập trình khác, lập trình Solidity cũng có các biến, hàm, lớp, phép toán số học, thao tác chuỗi và nhiều khái niệm khác.

2.4.1.2 Máy ảo Ethereum

Máy ảo Ethereum (EVM) cung cấp môi trường thời gian chạy cho các hợp đồng thông minh Ethereum. EVM chủ yếu liên quan đến việc đảm bảo an ninh và thực thi các chương trình không đáng tin cậy thông qua việc sử dụng mạng lưới các nút công cộng quốc tế. EVM chuyên ngăn chặn các cuộc tấn công từ chối dịch vụ và chứng nhận rằng các chương trình không có quyền truy cập vào trạng thái của nhau cũng như thiết lập liên lạc mà không có khả năng bị can thiệp.



Hình 2.1: Máy ảo Ethereum

(Nguồn: “What Is Solidity Programming in Ethereum | Simplilearn,” n.d.)

2.4.1.3 Bắt đầu với Solidity

Version Pragma

Pragma là các chỉ thị cho trình biên dịch về cách kiểm tra phiên bản hiện tại của Solidity có tương thích hay không. Trình biên dịch sẽ báo lỗi nếu phiên bản không tương thích. Mỗi dòng mã nguồn Solidity phải bắt đầu bằng "version pragma" để chỉ định phiên bản nào của trình biên dịch Solidity được sử dụng.

```
pragma solidity ^0.7.0;
```

Hình 2.2: Version Pragma

Điều này có nghĩa mã nguồn sẽ biên dịch với phiên bản Solidity lớn hơn 0.7.0 và nhỏ hơn 0.8.0. Nếu dùng phiên bản 0.8.0 để biên dịch, trình biên dịch sẽ báo lỗi.

Tùy khóa “Contract”

```
contract Test {
    //Functions and Data
}
```

Hình 2.3: Khai báo Contract

Khai báo một “Contract” đóng gói đoạn code.

Khai báo biến

```
uint public a;
uint public b;
uint public c;
```

Hình 2.4: Khai báo biến

Các biến trạng thái được viết trên Blockchain Ethereum và được duy trì vĩnh viễn trong kho lưu trữ hợp đồng.

Dòng “`uint public a`” khai báo một biến trạng thái kiểu uint có tên là a (kiểu số nguyên 256 bit), từ khóa `public` có ý nghĩa biến có thể được truy cập bởi các contract khác.

Khai báo hàm

```
function set(uint a, uint b) public
```

Hình 2.5: Khai báo hàm

Đơn vị thực thi mã nguồn được gọi là hàm. Ở đây là một hàm “set” của kiểu modifier. Hàm lấy một biến a và một biến b thuộc kiểu dữ liệu “`uint`” làm tham số, hàm có thể được sử dụng bởi các contract khác.

2.4.1.4 Thực thi code

Chế độ Offline

- Điều kiện:
 - Tải và cài đặt node.js.

- Cài đặt Truffle.
- Cài đặt ganache-cli.
- Cách chạy:
 - Tạo một dự án truffle và thiết lập một development network.
 - Develop, deploy một smart contract.
 - Từ Truffle console, tương tác với smart contract.
 - Tạo bài test để đánh giá các tính năng chính của Solidity.

Chế độ Online

Ở chế độ online, Remix IDE thường được sử dụng để biên dịch và chạy các Solidity Smart Contract.

Một số IDE platforms hỗ trợ Solidity:

- Truffle
- Hardhat
- Microsoft Visual Studio
- Microsoft Visual Studio Code
- Tendermint on Microsoft Azure

2.4.2 ReactJS

2.4.2.1 Giới thiệu về ReactJS

React là thư viện phát triển giao diện người dùng dựa trên JavaScript. Facebook và một cộng đồng các nhà phát triển mã nguồn mở điều hành nó. React không phải là ngôn ngữ mà là một thư viện và được sử dụng rộng rãi trong xây dựng các trang web. React lần đầu xuất hiện vào tháng 5 năm 2013 và đang là một trong những thư viện giao diện người dùng được sử dụng phổ biến nhất để phát triển web.

Ngoài giao diện người dùng đơn thuần, React còn cung cấp nhiều tiện ích mở rộng khác nhau, chẳng hạn như Flux và React Native để hỗ trợ toàn bộ kiến trúc ứng dụng.

2.4.2.2 Một số tính năng nổi bật của ReactJS

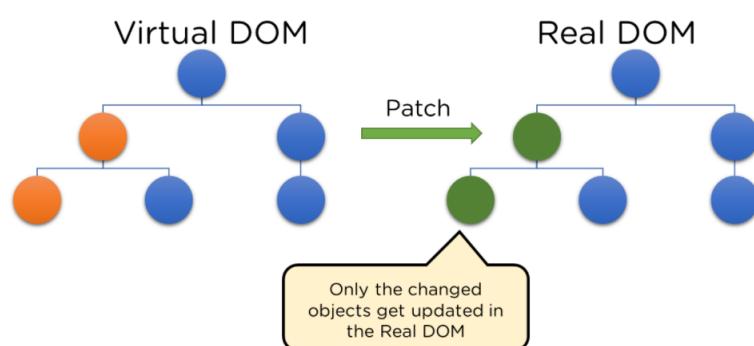
JSX (JavaScript Extension): là một phần mở rộng cú pháp JavaScript. Đó là một thuật ngữ được sử dụng trong React để mô tả giao diện người dùng trông như thế nào. Bạn có thể viết cấu trúc HTML trong cùng một tệp với mã JavaScript bằng cách sử dụng JSX.



Hình 2.6: JSX

(Nguồn: “The Best Guide to Know What Is React [Updated],” n.d.)

Virtual Document Object Model (DOM): Virtual DOM là phiên bản nhẹ hơn của Real DOM của React. Thao tác Virtual DOM nhanh hơn đáng kể so với thao tác Real DOM. Khi xảy ra sự thay đổi trạng thái của đối tượng, Virtual DOM chỉ cập nhật đối tượng đó trong Real DOM chứ không cập nhật tất cả. DOM (Document Object Model) xử lý tài liệu XML hoặc HTML dưới dạng cấu trúc cây trong đó mỗi nút là một đối tượng đại diện cho một phần của tài liệu.



Hình 2.7: Virtual DOM

(Nguồn: “The Best Guide to Know What Is React [Updated],” n.d.)

Architecture: React chịu trách nhiệm về giao diện của ứng dụng trong kiến trúc Model View Controller (MVC). MVC là một mẫu kiến trúc chia lớp ứng dụng thành Model, View và Controller.

Extension: React còn chứa các phần mở rộng bao trùm kiến trúc ứng dụng. React giúp xây dựng các ứng dụng di động và cung cấp khả năng hiển thị phía máy chủ. Flux và Redux, là những extensions tiêu biểu của React.

Data Binding: React sử dụng liên kết dữ liệu một chiều nên mọi hoạt động đều theo mô-đun và nhanh chóng. Hơn nữa, luồng dữ liệu một chiều có nghĩa là việc lồng các thành phần con bên trong các thành phần cha mẹ khi phát triển dự án React là điều phổ biến.

Debugging: Các ứng dụng React có thể được kiểm tra rất đơn giản và dễ dàng vì React được sử dụng phổ biến và có một cộng đồng các nhà phát triển rộng lớn. Các tiện ích mở rộng trình duyệt giúp đơn giản hóa và đẩy nhanh quá trình gỡ lỗi React cũng được cung cấp bởi Facebook.

2.4.3 NodeJS

2.4.3.1 Giới thiệu NodeJS

Node.js là thư viện và môi trường runtime chạy JavaScript đa nền tảng, mã nguồn mở để chạy các ứng dụng web bên ngoài trình duyệt của máy khách. Các nhà phát triển sử dụng Node.js để tạo các ứng dụng web phía máy chủ và nó phù hợp cho các ứng dụng sử dụng nhiều dữ liệu vì nó sử dụng mô hình hướng sự kiện, không đồng bộ.

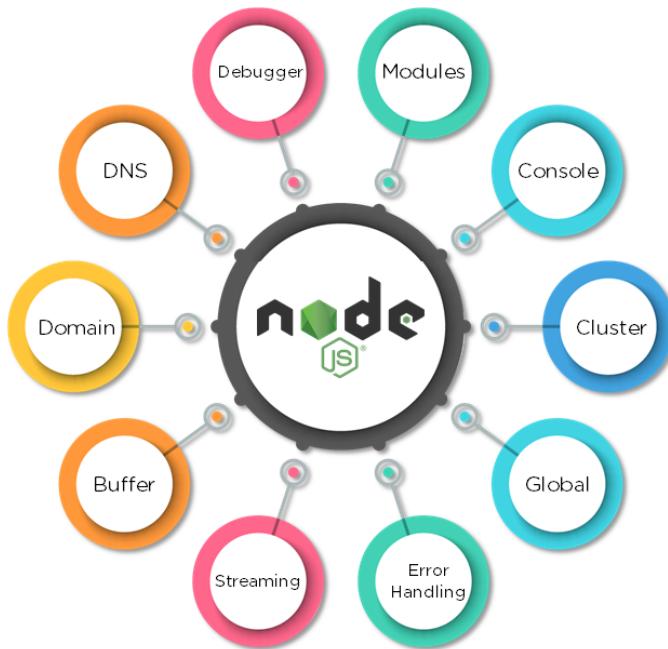
2.4.3.2 Tính năng và kiến trúc NodeJS

Tính năng:

- Lập trình hướng sự kiện và không đồng bộ
- Thời gian thực thi code nhanh
- Đơn luồng với khả năng mở rộng cao
- Không có buffer (vùng nhớ tạm thời)
- Truyền dữ liệu nhanh

- Khả năng tương thích trên nhiều nền tảng

Kiến trúc: Modules, Console, Cluster, Global, Error Handling, Streaming, Buffer, Domain, DNS, Debugger.

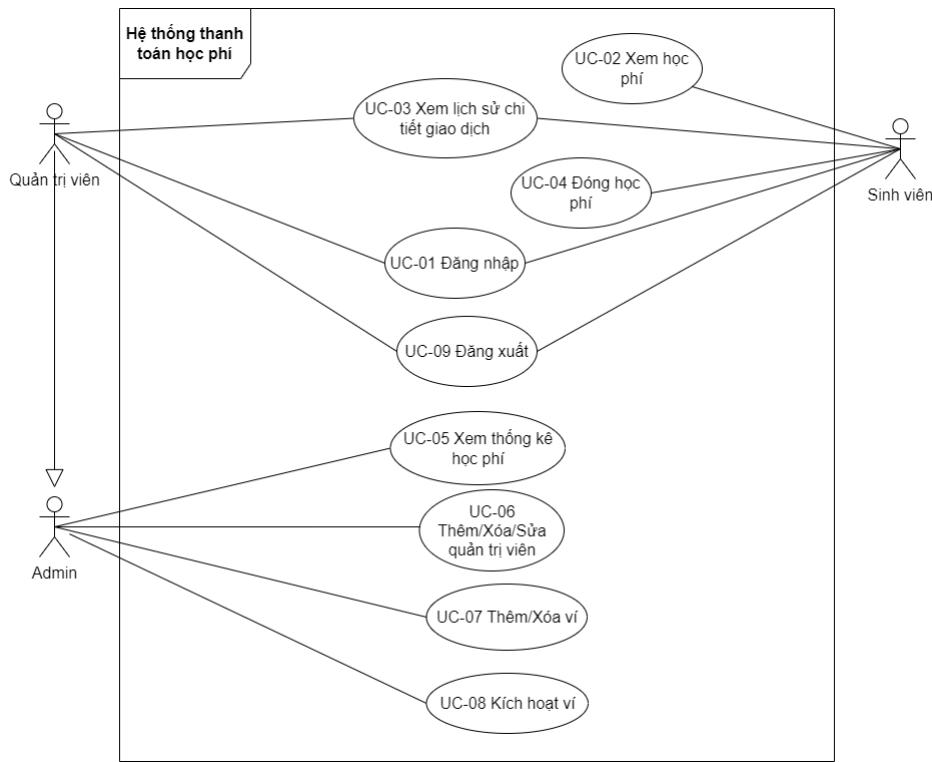


Hình 2.8: Kiến trúc NodeJS

(Nguồn: “What Is Node.Js,” n.d.)

CHƯƠNG 3. PHƯƠNG PHÁP NGHIÊN CỨU

3.1 Sơ đồ Use case tổng quát



Hình 3.1: Sơ đồ Use Case

Sau đây là mô tả chi tiết về các tác nhân và các Use Case trong hệ thống:

Bảng 3.1: Tác nhân của hệ thống

Tác nhân	Mô tả
Sinh viên	<ul style="list-style-type: none"> Là viên trường đại học Tôn Đức Thắng, đăng nhập vào hệ thống bằng tài khoản sinh viên đã được cung cấp. Sinh viên có thể chọn học kì xem học phí, đóng học phí bằng USDT, xem thông tin chi tiết giao dịch, đăng xuất khỏi hệ thống.
Quản trị viên	<ul style="list-style-type: none"> Là những người quản lý theo dõi các giao dịch, đăng nhập vào hệ thống bằng tài khoản quản trị viên.

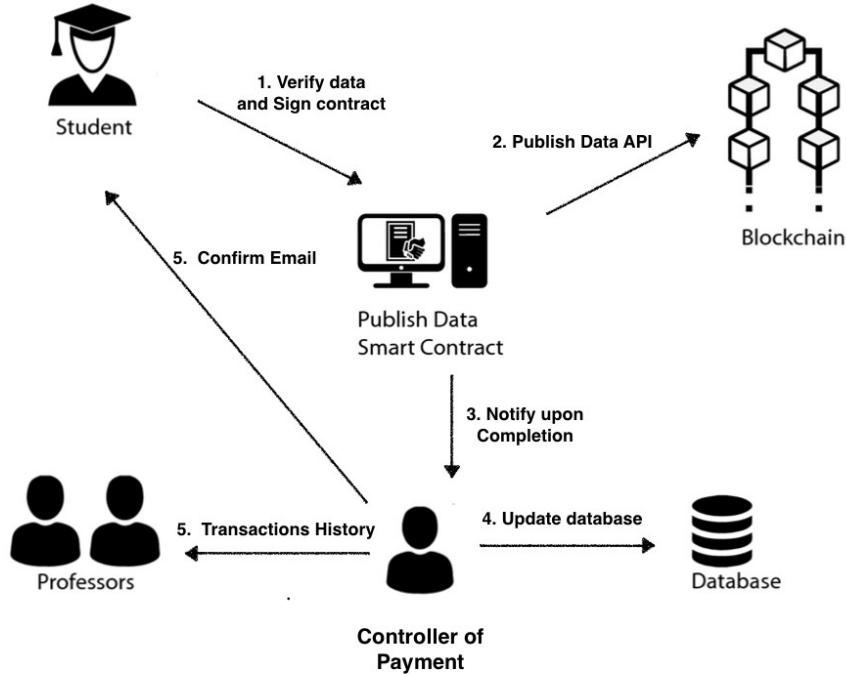
	<ul style="list-style-type: none"> Quản trị viên có quyền xem sinh viên nào chưa đóng hoặc đã đóng học phí rồi, xem thông tin chi tiết tất cả các giao dịch, theo dõi các sự cố, các lỗi phát sinh để báo cáo cho Admin, đăng xuất khỏi hệ thống.
Admin	<ul style="list-style-type: none"> Là người có quyền cao nhất trong hệ thống, đăng nhập vào hệ thống bằng tài khoản riêng của Admin. Admin quản lý các quản trị viên, do đó Admin có hết tất cả các quyền mà quản trị viên có. Ngoài ra, Admin còn có thể xem thống kê học phí, thêm/xóa/sửa quản trị viên, thêm/xóa ví, kích hoạt một ví khác.

Bảng 3.2: Use Case của hệ thống

ID	Tên Use Case	Mô tả	Tác nhân tương ứng
UC-01	Đăng nhập	Sinh viên, Quản trị viên, Admin đều có tài khoản phân quyền riêng để đăng nhập vào hệ thống. Cần phải đăng nhập để sử dụng các chức năng trong hệ thống.	Sinh viên, Quản trị viên, Admin
UC-02	Xem học phí	Sinh viên có thể xem chi tiết học phí theo từng học kì.	Sinh viên
UC-03	Xem lịch sử chi tiết giao dịch	Sau khi sinh viên đóng học phí thành công, sinh viên có thể xem chi tiết giao dịch như ngày/giờ đóng, số tiền đã đóng, học kì,... Quản trị viên, Admin	Sinh viên, Quản trị viên, Admin

		quản lý các giao dịch nên cũng có thể xem thông tin chi tiết của giao dịch.	
UC-04	Đóng học phí	Sau khi xem học phí, nếu trạng thái là chưa đóng, sinh viên có thể tiến hành thanh toán học phí bằng USDT.	Sinh viên
UC-05	Xem thống kê học phí	Admin có thể xem thống kê học phí theo từng học kì, từng năm, tổng số tiền sinh viên đã đóng, tổng số tiền chưa đóng.	Admin
UC-06	Thêm/Xóa/Sửa quản trị viên	Admin có quyền thêm/xóa/sửa quản trị viên nếu có thay đổi về nhân sự quản lý giao dịch.	Admin
UC-07	Thêm/Xóa ví	Admin có thể thêm hoặc xóa ví điện tử dùng để giao dịch trên thẻ thông.	Admin
UC-08	Kích hoạt ví	Admin có thể kích hoạt một ví điện tử khác để nhận tiền từ sinh viên.	Admin
UC-09	Đăng xuất	Sau khi hoàn tất các giao dịch, nếu người dùng không cần sử dụng đến hệ thống nữa thì có thể đăng xuất khỏi hệ thống để đảm bảo mọi thông tin được bảo mật.	Sinh viên, Quản trị viên, Admin

3.2 Tổng quan về mô hình ứng dụng công nghệ blockchain vào hệ thống



Hình 3.2: Sơ đồ tổng quan ứng dụng Blockchain vào hệ thống
Xác nhận số tiền và kí hợp đồng:

- Sau khi sinh viên xác nhận số tiền cần thanh toán ở đơn vị VND hệ thống sẽ gọi API sang api.coingecko.com để nhận tỉ lệ chuyển đổi từ VND sang USDT. Vì smart contract sẽ được thanh toán bởi USDT.
- Sinh viên chấp nhận với tỉ lệ chuyển đổi hiện tại. Thì hệ thống sẽ gọi sang cho smart contract của USDT token với tham số là lượng USDT cần để thanh toán học phí để smart contract của token approve smart contract của TDTU Payment có thể sử dụng lượng token đó.
- Sau khi sinh viên ấn chọn approve lượng token thì sinh viên cần kí xác nhận để thực hiện giao dịch đóng tiền học phí vào smart contract của TDTU Payment.

Gửi giao dịch lên mạng blockchain:

- Sau khi giao dịch được sinh viên xác thực, giao dịch sẽ được hệ thống gửi nó lên mạng blockchain thông qua một node.
- Node và dịch vụ giao dịch sẽ chịu trách nhiệm xác minh và đưa giao dịch của sinh viên vừa tạo vào một khối (block) mới trên blockchain.
- Khi được ghi xong hệ thống sẽ vào hàm đợi để mạng blockchain xác nhận và trả về transaction mới.
- Vì transaction này là thông tin quan trọng và cần minh bạch nên nhóm chúng tôi sẽ quyết định lưu lịch sử giao dịch này lên mạng blockchain.

Hoàn thành, xác nhận giao dịch:

- Sau khi đã xác nhận được giao dịch đã được nhận, xác thực và ghi vào mạng blockchain thì hệ thống sẽ chuyển thông tin về backend.
- Ở đây backend sẽ call một lần nữa lên hệ thống blockchain để xác nhận giao dịch đó có thật sự tồn tại và trạng thái đã hoàn thành chưa.

Update thông tin vào database:

- Sau khi backend đã xác nhận được giao dịch thì sẽ tiến hành lưu trữ thông tin vào database.
- Chuyển trạng thái của học phí sinh viên từ unpaid to paid.

Gửi email xác nhận giao dịch cho sinh viên:

- Thông tin thanh toán đã ghi nhận xuống database thành công thì hệ thống sẽ gửi một email xác nhận giao dịch thành công vào email của sinh viên.

Xem lịch sử giao dịch:

- Vì transaction đã hoàn thành và đã được ghi mạng blockchain nên admin có thể call để tra cứu bất cứ lúc nào.
- Vì thế mà admin hay quản trị viên có thể hỗ trợ và xác nhận giao dịch cho sinh viên trong tình huống xảy ra lỗi bất kì ở phía sinh viên như: mất mạng giữa lúc giao dịch, ...

3.3 Quy trình thực hiện

3.3.1 Tạo smart contract với Remix IDE:

Vì nhóm chúng tôi quyết định sử dụng token để thực hiện giao dịch là stable-coin USDT. Stablecoin là một loại tiền điện tử mã hóa, vì thế mà nó thừa hưởng đầy đủ các tính năng đặc trưng như tính phi tập trung, bảo mật cao và được bảo trợ, kiểm soát nghiêm ngặt. Với USDT là một loại stablecoin được đảm bảo bởi crypto thường được sử dụng cho các hợp đồng thông minh để quản lý việc đúc, đốt tiền, từ đó tạo niềm tin vì người đầu tư có thể kiểm tra các hợp đồng. USDT là đồng tiền mã hóa neo giá với đồng Đô la Mỹ. Mỗi đơn vị USDT thì sẽ có giá trị tương đương 1 USD. Vì USDT sẽ bảo vệ người dùng khỏi sự biến động của thị trường tiền mã hóa nên đây cũng là lý do chính nhóm chúng tôi chọn đồng token này để thực hiện giao dịch học phí cho đê tài của mình.

Vì smart contract này nhóm chúng tôi sẽ thực hiện trên mạng lưới Ethereum nên sẽ dùng USDT chuẩn ERC-20. Vì thế ở phần đầu smart contract sẽ khai báo một interface IERC20 để tương tác với smart contract token chuẩn ERC20. Contract này định nghĩa các hàm transfer, transferFrom và balanceOf. Vì thế khi deploy ta sẽ điền địa chỉ smart contract của USDT vào thì interface thì sẽ tương tác đến smart contract của USDT.

```
interface IERC20 {
    function transfer(address to, uint256 value) external returns (bool);
    function transferFrom(address from, address to, uint256 value) external returns (bool);
    function balanceOf(address account) external view returns (uint256);
}
```

Ở trong contract SchoolPaymentSystem sẽ có các biến và struct như:

- Biến "admin" lưu trữ địa chỉ của người quản trị hệ thống.
- Biến "receiverAddress" lưu trữ địa chỉ nhận thanh toán học phí.
- Biến "tokenAddress" lưu trữ địa chỉ của contract token.
- Struct "Transaction" định nghĩa các thuộc tính của một giao dịch bao gồm địa chỉ của sinh viên, số lượng, thời gian, mã học phí và mã sinh viên.

```

contract SchoolPaymentSystem {
    address public admin;
    address public receiverAddress;
    address public tokenAddress; // Địa chỉ của contract token

    struct Transaction {
        address student;
        uint256 amount;
        uint256 timestamp;
        string tuitionCode;
        string studentCode;
    }
}

```

Ở trong contract SchoolPaymentSystem sẽ có các Mảng và mapping như:

- Mảng "transactions" lưu trữ lịch sử giao dịch.
- Mapping "studentBalance" lưu trữ số dư token của từng sinh viên.

```

Transaction[] public transactions;
mapping(address => uint256) public studentBalance;

```

Ở trong contract SchoolPaymentSystem tạo một Constructor: constructor sẽ được gọi khi triển khai contract và khởi tạo các thông tin ban đầu như địa chỉ nhận thanh toán và địa chỉ contract token.

```

constructor(address _receiverAddress, address _tokenAddress) {
    admin = msg.sender;
    receiverAddress = _receiverAddress;
    tokenAddress = _tokenAddress;
}

```

Ở trong contract SchoolPaymentSystem tạo một Modifier: vì các smart contract rất chú trọng việc bảo mật và chính xác nên trong lập trình solidity có một phương thức là modifier được sử dụng trong một phương thức trong Smart Contract để kiểm tra các điều kiện trước khi các đoạn mã code trong phương thức đó được thực thi. Trong smart contract lần này sẽ dùng phương thức modifier "onlyAdmin" đảm bảo chỉ có admin mới có thể gọi các hàm được đánh dấu bằng modifier này.

```

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can call this function");
    _;
}

```

Ở trong contract SchoolPaymentSystem tạo một Event: Event trong solidity là một sự kiện dùng để lắng nghe khi có một sự kiện hay hành động khác được thực thi

thì Event này sẽ được kích hoạt theo. Trong smart contract này Event "TuitionPaid" được kích hoạt khi một sinh viên đã nộp học phí thành công.

```
event TuitionPaid(address indexed student, uint256 amount, string tuitionCode, string studentCode, uint256 timestamp);
```

Hàm changeReceiver: Hàm này cho phép admin thay đổi địa chỉ ví nhận tiền từ học sinh đóng học phí vào.

```
function changeReceiver(address _newReceiver) public onlyAdmin {
    receiverAddress = _newReceiver;
}
```

Hàm getTransactionHistory: Hàm này trả về tất cả các lịch sử giao dịch được thực hiện trên smart contract.

```
function getTransactionHistory() public view returns (Transaction[] memory) {
    return transactions;
}
```

Hàm depositTuition:

- Hàm này cho phép sinh viên nộp học phí. Sinh viên gửi số lượng token cần nộp từ ví của mình đến contract. Số lượng token được chuyển đến địa chỉ nhận thanh toán và số dư của sinh viên được cập nhật.
- Giao dịch được ghi lại trong mảng "transactions" và kích hoạt sự kiện "TuitionPaid".

```
function depositTuition(uint256 _amount, string memory _tuitionCode, string memory _studentCode) public {
    require(_amount > 0, "Amount must be greater than 0");

    IERC20 token = IERC20(tokenAddress);
    require(token.transferFrom(msg.sender, address(this), _amount), "TransferFrom failed");
    token.transfer(receiverAddress, _amount);

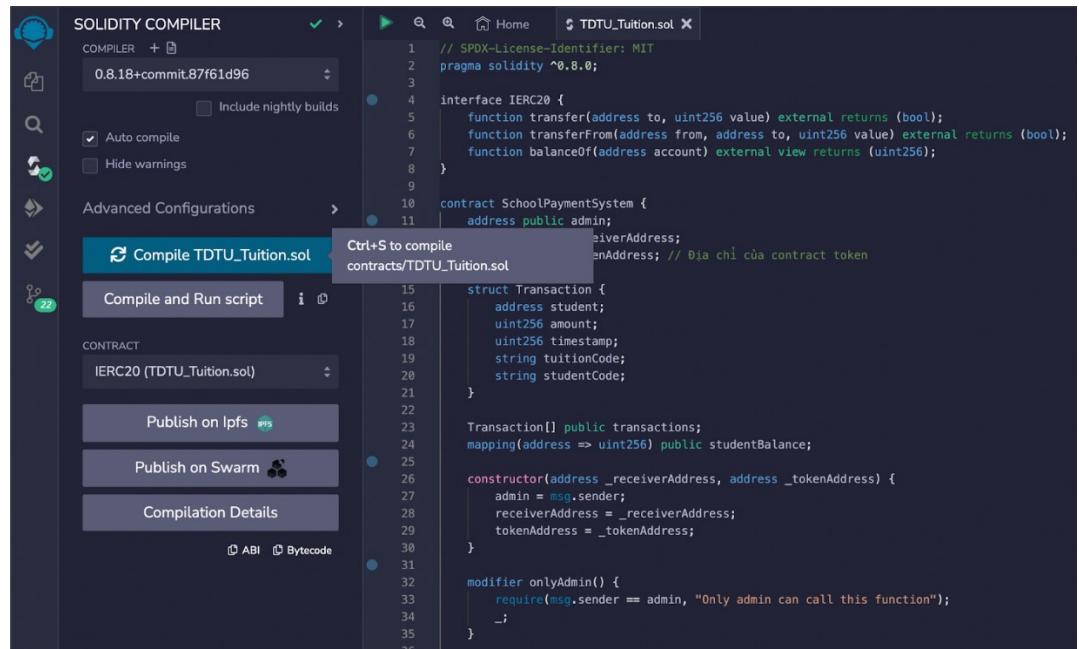
    studentBalance[msg.sender] += _amount;

    transactions.push(Transaction({
        student: msg.sender,
        amount: _amount,
        timestamp: block.timestamp,
        tuitionCode: _tuitionCode,
        studentCode: _studentCode
    }));

    emit TuitionPaid(msg.sender, _amount, _tuitionCode, _studentCode, block.timestamp);
}
```

3.3.2 Deploy smartcontract lên mạng Ethereum testnet

Sau khi hoàn thành smart contract. Tiến hành compile file TDTU_Tuition.sol. Ở tab Solidity Compiler.



```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

interface IERC20 {
    function transfer(address to, uint256 value) external returns (bool);
    function transferFrom(address from, address to, uint256 value) external returns (bool);
    function balanceOf(address account) external view returns (uint256);
}

contract SchoolPaymentSystem {
    address public admin;
    address receiverAddress;
    address tokenAddress; // Địa chỉ của token

    struct Transaction {
        address student;
        uint256 amount;
        uint256 timestamp;
        string tuitionCode;
        string studentCode;
    }

    Transaction[] public transactions;
    mapping(address => uint256) public studentBalance;

    constructor(address _receiverAddress, address _tokenAddress) {
        admin = msg.sender;
        receiverAddress = _receiverAddress;
        tokenAddress = _tokenAddress;
    }

    modifier onlyAdmin() {
        require(msg.sender == admin, "Only admin can call this function");
       _;
    }
}

```

Hình 3.3: Compile file TDTU_Tuition.sol

Compiler thành công, ta tiến hành deploy smart contract lên mạng Ethereum.

Với cái cài đặt như sau:

- ENVIRONMENT: Injected Provider - MetaMask.
- Account: chọn địa chỉ ví có token TBNB để kí transaction thực hiện deploy smart contract này.
- `_receiverAddress`: điền địa chỉ ví nhận tiền khi sinh viên đóng tiền vào. Địa chỉ ví này có thể thay đổi sau khi sử dụng hàm `changeReceiver()`.
- `_tokenAddress`: điền địa chỉ của token chuẩn ERC-20 để smart contract có thể tương tác đến.

```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

interface IERC20 {
    function transfer(address to, uint256 value) external returns (bool);
    function transferFrom(address from, address to, uint256 value) external returns (bool);
    function balanceOf(address account) external view returns (uint256);
}

contract SchoolPaymentSystem {
    address public admin;
    address public receiverAddress;
    address public tokenAddress; // Địa chỉ của contract token

    struct Transaction {
        address student;
        uint256 amount;
        uint256 timestamp;
        string tuitionCode;
        string studentCode;
    }

    Transaction[] public transactions;
    mapping(address => uint256) public studentBalance;

    constructor(address _receiverAddress, address _tokenAddress) {
        admin = msg.sender;
        receiverAddress = _receiverAddress;
        tokenAddress = _tokenAddress;
    }

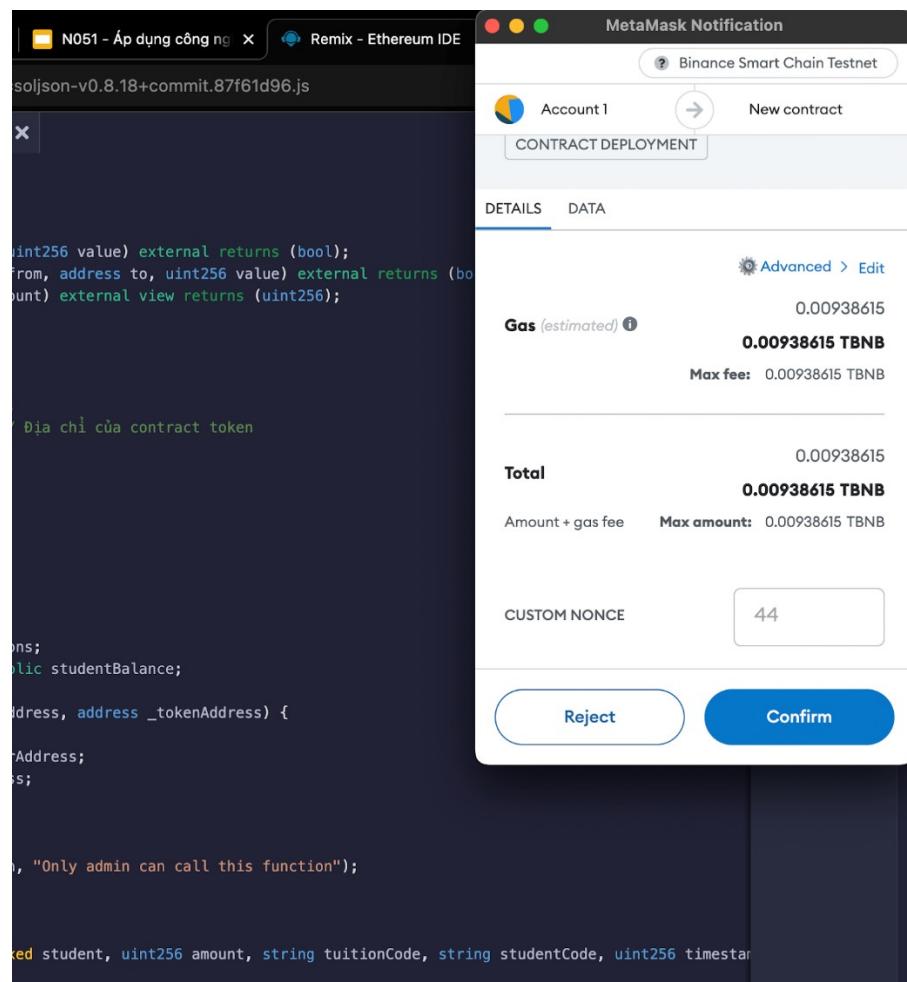
    function transfer(uint256 value) external returns (bool);
    function transferFrom(address from, address to, uint256 value) external returns (bool);
    function balanceOf(address account) external view returns (uint256);

    function addTransaction(address student, uint256 amount, string memory tuitionCode, string memory studentCode, uint256 timestamp) external {
        require(msg.sender == admin, "Only admin can call this function");
    }
}

```

Hình 3.4: Deploy Smart Contract

Ký transaction với metamask để thực hiện deploy:



Hình 3.5: Ký Transaction

Khi thì deploy thành công, ta sẽ nhận được địa chỉ của smart contract ở trên mạng Ethereum. Ta có thể xem qua smart contract trên trang BSCScan. Ta nhập địa chỉ vừa nhận được khi deploy ta sẽ thấy được giao diện của smart contract.

The screenshot shows the BscScan Testnet interface for a deployed smart contract. The contract address is 0xdFADABcaaE05aB24e4e311184535e32Db221f08b. The interface includes sections for Contract Overview, Transactions, Internal Txns, BEP-20 Token Txns, Contract (selected), and Events. Under the Contract tab, there are tabs for Code, Read Contract, and Write Contract. A search bar for source code is also present. The contract name is SchoolPaymentSystem, compiler version is v0.8.18+commit.87f61d96, and optimization is enabled with 200 runs. The contract source code (Solidity) is displayed in a code editor window, showing the SPDX license information and the pragma solidity statement.

Hình 3.6: Giao diện Smart Contract

Ở đây ta sẽ xem được phần quan trọng đó là Contract ABI. Đây là đoạn mã JSON sẽ được dùng để có thể tương tác với smart contract từ Frontend và Backend.

The screenshot shows the BscScan interface with the Contract ABI section selected. The JSON code represents the Application Binary Interface (ABI) for the smart contract. It defines various functions and their parameters, including external and internal types, state mutability, and function signatures.

```
{
  "inputs": [
    {
      "internalType": "address",
      "name": "_receiverAddress",
      "type": "address"
    }
  ],
  "internalType": "address",
  "name": "_tokenAddress",
  "type": "address"
},
{
  "stateMutability": "nonpayable",
  "type": "constructor"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "student",
      "type": "address"
    },
    {
      "indexed": false,
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "name": "depositTuition",
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "string",
      "name": "tuitionCode",
      "type": "string"
    }
  ],
  "name": "getTransactionHistory",
  "outputs": [
    {
      "internalType": "address",
      "name": "student",
      "type": "address"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "newReceiver",
      "type": "address"
    }
  ],
  "name": "changeReceiver",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "name": "withdrawTuition",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "stateMutability": "nonpayable",
  "type": "function"
}
}
```

Hình 3.7: Mã JSON

3.3.3 Xây dựng database với mySQL

Theo như chúng tôi tìm hiểu thì khi áp dụng blockchain vào dự án thì khi thực hiện các lệnh ghi dữ liệu lên blockchain thì chuỗi block sẽ phải chạy và thực hiện xác nhận việc ghi đó nên sẽ tốn nhiều phí gas khi bắt cứ thông tin nào cũng ghi lên node.

Vì thế khi thực hiện dự án trong thực tế ta cần kết hợp với việc sử dụng database và backend để hỗ trợ việc lưu trữ và truy cập thông tin. Chỉ ghi lên node nǔa

thông tin thực sự quan trọng và cần minh bạch như ở đây là: lịch sử giao dịch, địa chỉ ví nhận tiền của trường, ...

Vì ở đồ án này nhóm chúng tôi chỉ chú trọng vào việc tìm hiểu và chú trọng và việc xử lý quá trình thanh toán, kiểm tra học phí nên tôi sẽ giả định sync được học phí từ data của trường.

The screenshot shows the MySQL Workbench interface. On the left, a tree view displays the database schema with several tables under the 'tuition-crypto' database. On the right, a table provides detailed statistics for these tables:

Table	Action	Rows	Type	Collation	Size	Overhead
admin_managers	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	64.0 Kib	-
network_types	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 Kib	-
roles	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 Kib	-
school_wallets	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	48.0 Kib	-
semester	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
student_wallets	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 Kib	-
tuition	Browse Structure Search Insert Empty Drop	102	InnoDB	utf8mb4_general_ci	32.0 Kib	-
7 tables	Sum				118 InnoDB utf8mb4_general_ci 256.0 Kib	0 B

Hình 3.8: Database

Database sẽ tạo một bảng tuition chứa thông tin học phí với các cột sau:

- TuitionID: Kiểu int(11), không được null, là khóa chính của bảng.
- Contents: Kiểu varchar(255), có thể null, lưu trữ nội dung của học phí.
- PreviousPendingCharges: Kiểu decimal(10,2), có thể null, lưu trữ số tiền chờ thanh toán trước đó.
- SemesterTuition: Kiểu decimal(10,2), có thể null, lưu trữ học phí của học kỳ.
- Reduction: Kiểu decimal(10,2), có thể null, lưu trữ số tiền giảm giá, học bổng.
- TotalTuitionUnpaid: Kiểu decimal(10,2), có thể null, lưu trữ tổng số học phí chưa thanh toán.
- TotalTuitionPaid: Kiểu decimal(10,2), có thể null, lưu trữ tổng số học phí đã thanh toán.
- RemainingUnpaidTuition: Kiểu decimal(10,2), có thể null, lưu trữ số học phí còn lại chưa thanh toán.
- Note: Kiểu text, có thể null, lưu trữ ghi chú.

- LatestUpdatedTime: Kiểu timestamp, không được null, mặc định là thời gian hiện tại khi thêm dữ liệu.
- SemesterID: Kiểu int(11), có thể null, là khóa ngoại tham chiếu đến cột SemesterID của bảng semester.
- studentId: Kiểu int(11), có thể null, lưu trữ ID của sinh viên.
- IsPaid: Kiểu tinyint(1), không được null, mặc định là 0, lưu trữ trạng thái thanh toán (1: Đã thanh toán, 0: Chưa thanh toán).

Bảng semester có các cột sau:

- SemesterID: Kiểu int(11), không được null, là khóa chính của bảng.
- SemesterName: Kiểu varchar(50), có thể null, lưu trữ tên học kỳ.
- due_day: Kiểu date, có thể null, lưu trữ ngày đáo hạn.

Bảng school_wallets: dùng để quản lý các loại ví mà trường sẽ sử dụng. Và chỉ có một ví được active trong một thời điểm để nhận tiền. Có các cột sau:

- id: Kiểu int(11), không được null, là khóa chính của bảng.
- address: Kiểu varchar(255), không được null, lưu trữ địa chỉ ví của trường học.
- network_type_id: Kiểu int(11), có thể null, lưu trữ ID của loại mạng.
- created_at: Kiểu timestamp, không được null, mặc định là thời gian hiện tại khi thêm dữ liệu.
- updated_at: Kiểu timestamp, không được null, mặc định là thời gian hiện tại khi thêm dữ liệu và được cập nhật khi có sự thay đổi.
- is_active: Kiểu tinyint(1), mặc định là 0, lưu trữ trạng thái hoạt động của ví trường học (1: hoạt động, 0: không hoạt động).

Bảng network_types: dùng để quản lý và khoá ngoại cho bảng student_wallet và shool wallet trong trường hợp trường sẽ mở rộng thanh toán qua nhiều mạng sau này. Nhưng với demo hiện tại thì nhóm chỉ sử dụng mạng Ethereum. Có các cột sau:

- id: Kiểu int(11), không được null, là khóa chính của bảng.
- name: Kiểu varchar(50), không được null, lưu trữ tên loại mạng.

- created_at: Kiểu timestamp, không được null, mặc định là thời gian hiện tại khi thêm dữ liệu.
- updated_at: Kiểu timestamp, không được null, mặc định là thời gian hiện tại khi thêm dữ liệu và được cập nhật khi có sự thay đổi.
- symbol: Kiểu varchar(10), có thể null, lưu trữ biểu tượng hay ký hiệu của loại mạng.

Bảng admin_managers: ngoài admin ra thì nhóm có tạo thêm một role nữa là giáo viên (teacher) có thể đăng nhập vào hệ thống và xem các lịch sử giao diện nhằm có thể hỗ trợ sinh viên khi có lỗi trong quá trình thực hiện giao dịch hay vấn đề nào đó mà không cần đến admin. Có các cột sau:

- id: Kiểu int(11), không được null, là khóa chính của bảng.
- fullName: Kiểu varchar(255), không được null, lưu trữ tên đầy đủ của quản trị viên.
- email: Kiểu varchar(255), không được null, lưu trữ địa chỉ email của quản trị viên.
- username: Kiểu varchar(50), không được null, lưu trữ tên đăng nhập của quản trị viên.
- password: Kiểu varchar(255), không được null, lưu trữ mật khẩu của quản trị viên.
- role_id: Kiểu int(11), có thể null, lưu trữ ID của vai trò (quyền hạn) của quản trị viên.

Bảng roles có các cột sau:

- id: Kiểu int(11), không được null, là khóa chính của bảng.
- name: Kiểu varchar(50), không được null, lưu trữ tên vai trò (quyền hạn).

3.3.4 Sử dụng NodeJs để tương tác với database và smart contract

Ở phần Backend của dự án này nhóm chúng tôi sử dụng công nghệ là NodeJs và framework là Express JS để xây dựng các API tương tác với smart contract và Frontend.

Về tổ chức thư mục thì nhóm chúng tôi sẽ tổ chức với 4 thư mục chính:

- controller: chức các file js thực hiện việc xử lý thông tin từ request, tương tác với database và smartcontract.
- helper: các file js thực hiện các chức năng hỗ trợ cho controller.
- routers: thực hiện tạo các đường dẫn cho từng method và call validator và controller để thực hiện các yêu cầu.
- validations: các file js thực hiện các chức năng xác thực các trường thông tin nhận từ request trước khi truyền các trường dữ liệu sang controller để thực hiện lưu, thay đổi thông tin xuống database.

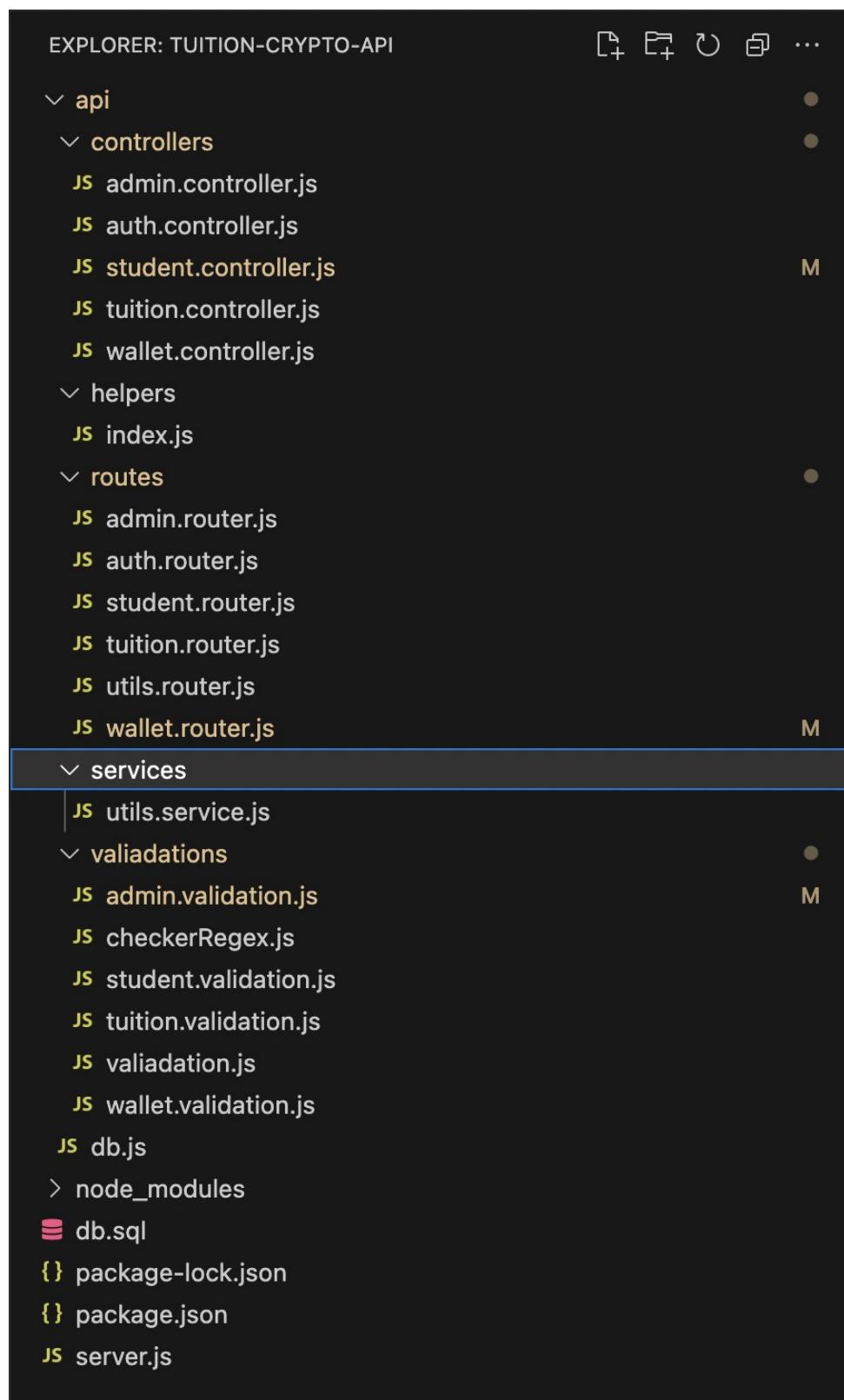
Dự án sẽ có các router chính như:

```

30  app.use(cors(corsOpts));
31
32 // for parsing application/json
33 app.use(express.json());
34
35 // for parsing application/x-www-form-urlencoded
36 app.use(express.urlencoded({ extended: true }));
37
38 // for parsing multipart/form-data
39 app.use(upload.array());
40 app.use(express.static('public'));
41
42
43 // Router
44 app.use('/auth', authRouter);
45
46 app.use('/admin', adminRouter);
47
48 app.use('/tuition', tuitionRouter);    You, 3 weeks ago • update ...
49
50 app.use('/wallet', walletRouter)
51
52 app.use('/student', studentRouter)
53
54
55 app.use(function(req, res) {
56   | | res.status(404).send({url: req.originalUrl + ' not found'});
57 })
58
59 // listen on environment
60 app.listen(port, () => console.log('port listen hereeeeeee', port))
61

```

Hình 3.9: Router



Hình 3.10 API

auth: dùng cho phần đăng nhập của Admin.

admin: với các thao tác như:

- thêm/sửa/xoá tài khoản của quản trị viên,
- lấy thông tin thống kê học phí từ database.

```

61   // GET
62
63   router.get('/user-list', adminController.GetAllUser);
64
65   router.get('/wallet-list', adminController.GetWallets);
66
67   router.get('/network-types', adminController.GetNetworkTypes);
68
69   // POST
70
71 > router.post('/create', ...
108 )
109
110 > router.post('/login', ...
130 )
131
132 > router.post('/edit', ...
158 )
159
160 > router.post('/delete', ...
194 )
195
196 > router.post('/tuition-analytics', ...
215 )
216
217 > router.post('/wallet-create', ...
251 )
252
253 > router.post('/wallet-delete', ...
287 )
288
289 > router.post('/wallet-active', ...
323 )
324
325 > router.post('/wallet-edit', ...
359 )
360
361   module.exports = router;

```

Hình 3.11: Admin

tuition:

- lấy thông tin số tiền học phí cần đóng cho học sinh.
- danh sách học kì
- gửi mail xác nhận đóng tiền khi giao dịch của sinh viên thành công.

```

You, 3 weeks ago | 1 author (You)
1 'use strict';          You, 3 weeks ago • update ...
2 const express = require('express');
3
4 const { validationResult } = require('express-validator');
5
6 const controller = require('../controllers/tuition.controller')
7
8 const validation = require('../validations/tuition.validation')
9
10 const db = require('../db')
11 const router = express.Router();
12
13 router.get('/semester-list', controller.GetSemester);
14
15 router.post('/filter',
16
17   validation.filter,
18
19   async (req,res) => {
20     const errors = await validationResult(req);
21
22     if (!errors.isEmpty()) {
23
24       return res.status(422).json({
25         status: false,
26         msg: errors?.errors[0]?.msg,
27         position: errors?.errors[0]?.param,
28       })
29     }
30     else{
31       controller.Filter(req,res)
32     }
33   }
34 )
35
36 module.exports = router;

```

Hình 3.12: Tuition

wallet:

- thêm/sửa/xoá các địa chỉ ví, tiến hành active địa chỉ ví nhận tiền.

student:

- lấy thông tin số tiền học phí cần đóng cho học sinh.
- thực hiện xác nhận giao dịch khi sinh viên đóng tiền

```

You, 2 weeks ago | 1 author (You)
1  'use strict';
2  const express = require('express');
3
4  const { validationResult } = require('express-validator');
5
6  const controller = require('../controllers/wallet.controller')
7  const validation = require('../validations/wallet.validation')
8
9  const db = require('../db');
10 const router = express.Router();
11
12 async function isExists(value, rowName, tableName) {
13   return new Promise((resolve, reject) => {
14     const sql = `SELECT COUNT(*) AS count FROM ${tableName} WHERE ${rowName} = ?`;
15     db.query(sql, [value], (err, result) => {
16       if (err) {
17         reject(err);
18       } else {
19         resolve(result[0].count > 0);
20       }
21     });
22   });
23 }
24
25 router.get('/list', controller.GetWallets)
26
27 > router.post('/add', ...
28 )
29
30 > router.post('/delete', ...
31 )
32
33 module.exports = router;| You, 2 weeks ago • add : handle student wallet

```

Hình 3.13: Wallet

Với các với method cần verify thông tin thì nhóm em sử dụng express-validator để tiếp hành validation cho các trường thông tin trước khi cập nhập vào database.

```
You, 6 days ago | 1 author (You)
1  'use strict';
2  const express = require('express');
3
4  const { validationResult } = require('express-validator');
5
6  const controller = require('../controllers/student.controller')
7  const validation = require('../validations/student.validation')
8
9  const db = require('../db');
10 const router = express.Router();
11
12 async function isExists(value, rowName, tableName) {
13     return new Promise((resolve, reject) => {
14         const sql = `SELECT COUNT(*) AS count FROM ${tableName} WHERE ${rowName} = ?`;
15         db.query(sql, [value], (err, result) => {
16             if (err) {
17                 reject(err);
18             } else {
19                 resolve(result[0].count > 0);
20             }
21         });
22     });
23 }
24
25 router.get('/test-email', controller.SendEmail);
26
27 > router.post('/tuition', ...
52 )
53
54 > router.post('/infor', ...
71 )
72
73 > router.post('/confirm-payment', ...
90 )
91
92 module.exports = router;      You, 2 weeks ago • update
```

Hinh 3.14: Student

```

1  You, 2 weeks ago | 1 author (You)
2  const { check, body } = require('express-validator');
3
4  const {
5    onlyLettersAndNumbers,
6    onlyLettersAndSpaces,
7    checkerPassword
8  }
9  = require('../checkerRegex')
10
11
12 exports.login = [
13   check('username', 'User ID is requied').notEmpty(),
14   // check('username', 'Invalid user id').isInt({ min: 1 }),
15   check('password')
16     .notEmpty().trim().withMessage('Password is requied')
17     .trim().isLength({ min: 6 }).withMessage('Password at least 6 characters long')
18 ]
19
20 > exports.create = [...]
21
22
23 > exports.edit = [...]
24
25
26 > exports.delete = [...]
27
28
29 > exports.tuitionAnalytics = [
30   check('semesterId', 'Semester ID is requied').notEmpty(),
31 ]
32
33
34 > exports.walletDelete = [
35   check('addressId', 'Address ID is requied').notEmpty(),
36 ]
37
38
39 > exports.walletActive = [
40   check('id', 'ID is requied').notEmpty(),
41 ]
42
43
44 > exports.walletCreate = [
45   check('address')
46     .notEmpty().trim().withMessage('Address is requied')
47     .trim().isLength({ min: 6 }).withMessage('Invalid address'),
48 ]
49
50
51 > exports.walletEdit = [
52   check('address')
53     .notEmpty().trim().withMessage('Address is requied')
54 ]
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

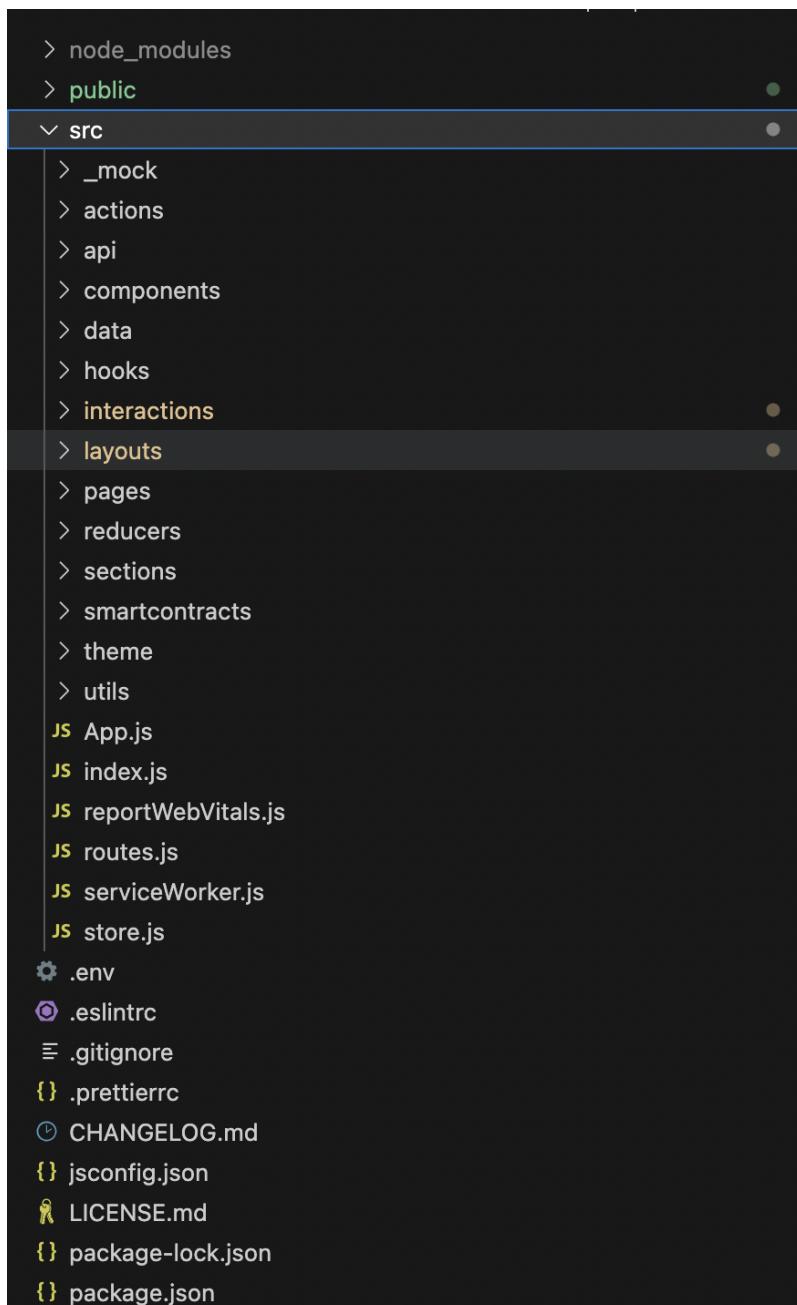
Hình 3.15: Validation

3.3.5 Sử dụng ReactJS và MUI để xây dựng giao diện

Nhóm chúng tôi tổ chức project react với các thư mục chính như:

- action, reducers và file store.js: nhằm sử dụng redux để quản lý các state và việc rerender và truyền dữ liệu của các component. Nâng cao hiệu năng cho trang web.
- api: chứa các hàm dùng để thực hiện các method xuống backend.

- component: chứa các giao diện được sử dụng lại trong các trang như bảng, button, ...
- interactions: chứa các hàm dùng để tương tác lên smart contract
- smartcontracts: chứa thông tin như ABI và contract address của PaymentTuition và USDT contract.
- utils: chứa các hàm dùng chung như: format giá tiền, format ngày,



Hình 3.16: Interactions

Tiến hành kết nối/huỷ kết nối ví vào trang student-tuition trước khi thực hiện giao dịch:

```

4  export async function connectWallet() {
5    if(window.ethereum){
6      try {
7        await window.ethereum.request({ method: 'eth_requestAccounts' })
8        if (window.ethereum.selectedAddress) {
9          return window.ethereum.selectedAddress;
10        }
11      } catch (error) {
12        return error
13      }
14    }else{
15      alert("install metamask extension!!!")
16    }
17
18    return false;
19  }
20
21 export async function disconnectWallet() {
22  if(window.ethereum){
23    try {
24      await window.ethereum.request({
25        method: "wallet_requestPermissions",
26        params: [
27          {
28            eth_accounts: []
29          }
30        ]
31      });
32    } catch (error) {
33      return error
34    }
35  }else{
36    alert("install metamask extension!!!")
37  }
38
39  return false
40}

```

Hình 3.17: Kết nối ví

Hàm connectWallet():

- Kiểm tra xem trình duyệt có hỗ trợ Web3 Provider (Metamask) thông qua biến window.ethereum.
- Nếu trình duyệt hỗ trợ, thực hiện một yêu cầu gọi đến Web3 Provider để yêu cầu quyền truy cập tài khoản ví số học (eth_requestAccounts).
- Nếu người dùng chọn một địa chỉ ví từ Metamask, hàm sẽ trả về địa chỉ ví đã chọn. Nếu có lỗi xảy ra trong quá trình yêu cầu, hàm sẽ trả về lỗi đó.

- Nếu trình duyệt không hỗ trợ Web3 Provider, hiển thị thông báo yêu cầu người dùng cài đặt tiện ích Metamask. Cuối cùng, hàm trả về giá trị false để cho biết kết nối ví không thành công.

Hàm disconnectWallet(): Kiểm tra xem trình duyệt có hỗ trợ Web3 Provider (Metamask) thông qua biến window.ethereum.

- Nếu trình duyệt hỗ trợ, thực hiện một yêu cầu gọi đến Web3 Provider để yêu cầu ngắt kết nối tài khoản ví (wallet_requestPermissions).
- Nếu có lỗi xảy ra trong quá trình yêu cầu, hàm sẽ trả về lỗi đó.
- Nếu trình duyệt không hỗ trợ Web3 Provider, hiển thị thông báo yêu cầu người dùng cài đặt tiện ích Metamask.
- Cuối cùng, hàm trả về giá trị false để cho biết ngắt kết nối ví không thành công.

Hàm approve token đến contract của USDT:

```

1 import Web3 from 'web3';      You, last week * update ...
2
3 import SMART_CONTRACT from '../smartcontracts';
4
5 export const approveToken = async (amount) => {
6   try {
7     const web3 = new Web3(window.ethereum);
8
9     const contract = new web3.eth.Contract(SMART_CONTRACT.TOKEN.ABI, SMART_CONTRACT.TOKEN.ADDRESS);
10
11    const data_ = await contract.methods.approve(
12      SMART_CONTRACT.TDTU_TUITIОН.ADDRESS,
13      web3.utils.toWei(amount.toString(), 'ether')
14    ).encodeABI();
15
16    const transactionParameters = {
17      to: SMART_CONTRACT.TOKEN.ADDRESS,
18      from: window.ethereum.selectedAddress,
19      data: data_,
20      chainId: 97,
21    };
22
23    const txHash = await window.ethereum.request({
24      method: 'eth_sendTransaction',
25      params: [transactionParameters],
26    });
27      console.log('Hash of the transaction:', txHash);
28      console.log(true)
29      return true;
30    } catch (error) {
31      console.error('An error occurred:', error);
32      return null;
33    }
34  }

```

Hình 3.18: Token Contract

- Hàm nhận đầu vào là amount (số tiền) cần được chấp thuận.
- Hàm bắt đầu bằng việc tạo một đối tượng Web3 mới sử dụng window.ethereum làm nhà cung cấp.
- Sau đó, hàm tạo một đối tượng hợp đồng (contract) sử dụng địa chỉ hợp đồng và ABI của token.
- Tiếp theo, hàm mã hóa yêu cầu chấp thuận số tiền bằng cách gọi phương thức encodeABI() trên đối tượng hợp đồng. Yêu cầu này bao gồm địa chỉ hợp đồng học phí của TDTU và số tiền được chuyển đổi thành đơn vị wei.
- Xây dựng thông tin giao dịch bao gồm địa chỉ hợp đồng token, địa chỉ ví được chọn, dữ liệu mã hóa và chainId.
- Gửi yêu cầu giao dịch tới Web3 Provider thông qua eth_sendTransaction và nhận lại mã giao dịch (txHash). In ra mã giao dịch và trả về true để cho biết yêu cầu chấp thuận đã thành công.
- Nếu có lỗi xảy ra trong quá trình thực hiện, hàm sẽ in ra lỗi và trả về null.

Tiến hành thanh toán học phí:

- Hàm nhận đầu vào là amount (số tiền), tuitionCode (mã học phí), studentCode (mã sinh viên), onComplete (hàm hoàn thành), và data (dữ liệu khác cần thiết).
- Hàm bắt đầu bằng việc tạo một đối tượng Web3 mới sử dụng window.ethereum làm nhà cung cấp.
- Sau đó, hàm gọi hàm approveToken(amount) để xác nhận việc chấp thuận số tiền hợp lệ.
- Nếu số tiền được chấp thuận, hàm tiếp tục thực hiện các bước sau: Gọi hàm TDTUContract() để nhận đối tượng hợp đồng thông minh (smart contract).

```

68  export const despositTuition = async (amount, tuitionCode, studentCode, onComplete, data) => {
69    try {
70      const web3 = new Web3(window.ethereum);
71
72      if(approveToken(amount)){
73        try{
74          const contract = await TDTUContract();
75
76          const data_ = await contract.methods.depositTuition(
77            web3.utils.toWei(amount.toString(), 'ether'),
78            tuitionCode,
79            studentCode
80          ).encodeABI();
81
82          const transactionParameters = {
83            to: SMART_CONTRI.any )TU TUITION.ADDRESS,
84            from: window.ethereum.selectedAddress,
85            data: data_,
86            chainId: 97,
87          };
88
89          const txHash = await window.ethereum.request({
90            method: 'eth_sendTransaction',
91            params: [transactionParameters],
92          });
93
94          await waitForTransactionConfirmation(txHash);
95
96          const result = await tuitionApi.confirmPaymentTuition(
97            {
98              studentId : studentCode,
99              paidAmount : data.TotalInVND,
101             rate : data.rate,
102             USDTAmount : amount,
103             tuitionId : tuitionCode,
104             tx : txHash,
105             semesterName : data.semesterName
106           }
107         )
108
109         if(result.status === 200){
110           await onComplete()
111         }
112       } catch (error){
113         console.error(error);
114       }
115     }
116   } catch (error) {
117     console.error(error);
118   }
119 }
120 }

```

Hình 3.19: Tiến hành thanh toán

- Sử dụng đối tượng hợp đồng để mã hóa thông tin đặt cọc học phí bao gồm số tiền (được chuyển đổi thành đơn vị wei), mã học phí và mã sinh viên.
- Xây dựng thông tin giao dịch bao gồm địa chỉ hợp đồng thông minh, địa chỉ ví được chọn, dữ liệu mã hóa và chainId.

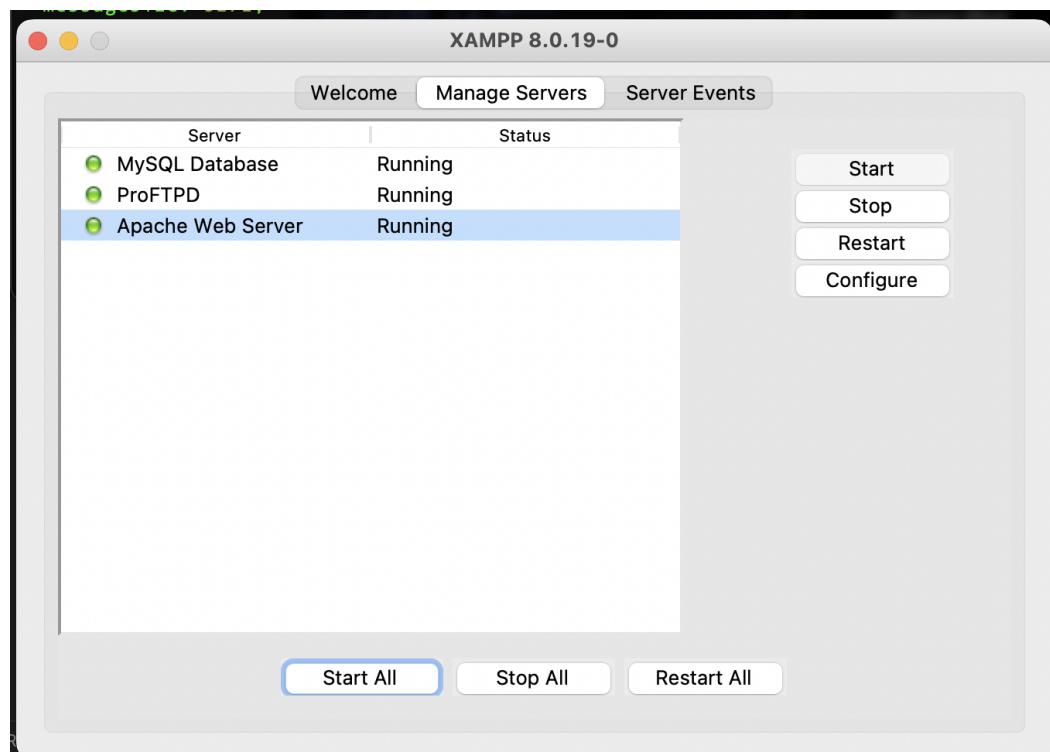
- Gửi yêu cầu giao dịch tới Web3 Provider thông qua eth_sendTransaction và nhận lại mã giao dịch (txHash).
- Chờ xác nhận giao dịch bằng cách gọi hàm waitForTransactionConfirmation(txHash). Gửi yêu cầu xác nhận thanh toán học phí đến API thông qua hàm tuitionApi.confirmPaymentTuition(), truyền các thông tin cần thiết như mã sinh viên, số tiền đã thanh toán, mã giao dịch và các thông tin khác.
- Nếu yêu cầu xác nhận thanh toán thành công (status === 200), thực thi hàm onComplete(). Nếu số tiền không được chấp thuận hoặc có lỗi xảy ra trong quá trình thực hiện, hàm sẽ in ra lỗi tương ứng.

CHƯƠNG 4. THỰC NGHIỆM

4.1 Cài đặt thực nghiệm

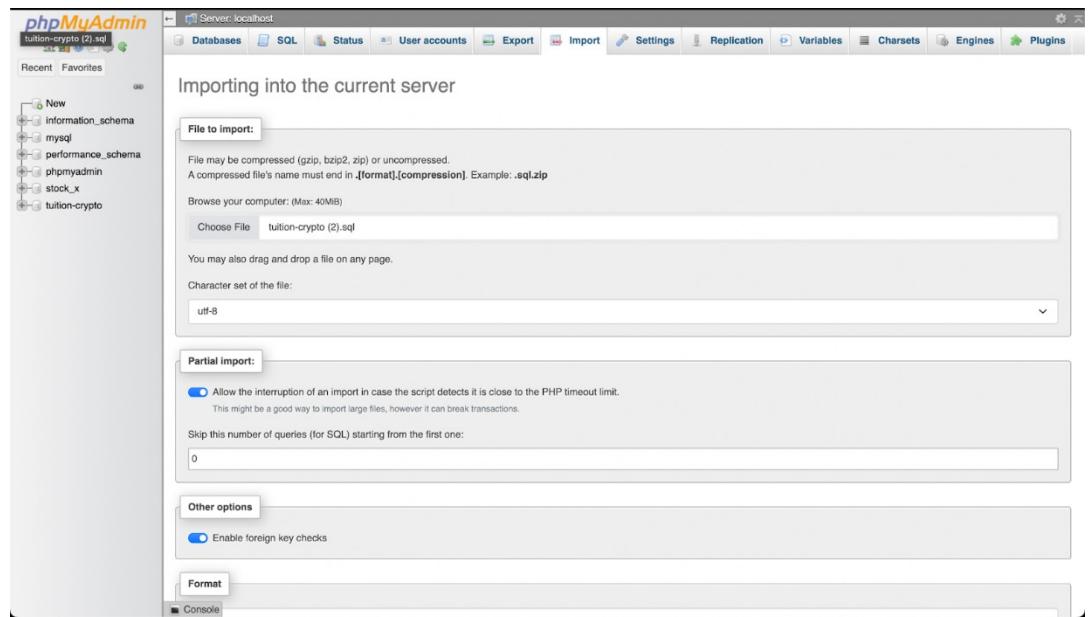
Thiết bị sử dụng để tiến hành cài đặt thực nghiệm nhóm chúng tôi sử dụng là: MacBook Pro (13-inch, M1, 2020) - 16GB RAM - macOS Monterey 12.4.

Database: mở Manager-osx và chọn Start All.



Hình 4.1: Cài đặt thực nghiệm

- Truy cập trang: <http://localhost/phpmyadmin/> để xem giao diện của phpMyAdmin.
- Import file database:



Hình 4.2: Import Database

Xem được tất cả bảng:

Table	Action	Rows	Type	Collation	Size	Overhead
admin_managers	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	64.0 Kib	-
network_types	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 Kib	-
roles	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 Kib	-
school_wallets	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	48.0 Kib	-
semester	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
student_wallets	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 Kib	-
tuition	Browse Structure Search Insert Empty Drop	102	InnoDB	utf8mb4_general_ci	32.0 Kib	-
7 tables	Sum				118 InnoDB utf8mb4_general_ci 256.0 Kib	0 B

Hình 4.3: Xem tất cả các bảng

Backend:

- Mở terminal mới ở thư mục tuition-crypto-api.
- Chạy lệnh: npm i -f để tiến hành cài đặt cái module cần thiết.
- Chạy lệnh: npm start

```

● ○ ● tuition-crypto-api — node -v npm start TERM_PROGRAM=Apple_Terminal SHELL=/bin/zsh...
(base) lixxkook@Macbut-M1-cua-Linh tuition-crypto-api % pwd
/Users/lixxkook/Desktop/TDTU/DACNTT2/tuition-crypto-api
(base) lixxkook@Macbut-M1-cua-Linh tuition-crypto-api % np start
zsh: command not found: np
(base) lixxkook@Macbut-M1-cua-Linh tuition-crypto-api % npm start
> stockx-api@1.0.0 start
> nodemon server.js

[nodemon] 2.0.19
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node server.js'
port listen hereeeeeee 8080

```

Hình 4.4: Chạy lệnh backend

Frontend:

- Mở terminal mới ở thư mục tuition-crypto-ui.
- Chạy lệnh: npm i -f để tiến hành cài đặt cái module cần thiết.
- Chạy lệnh: npm start

```

Compiled successfully!

You can now view @minimal/material-kit-react in the browser.

Local:          http://localhost:3000
On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use yarn build.

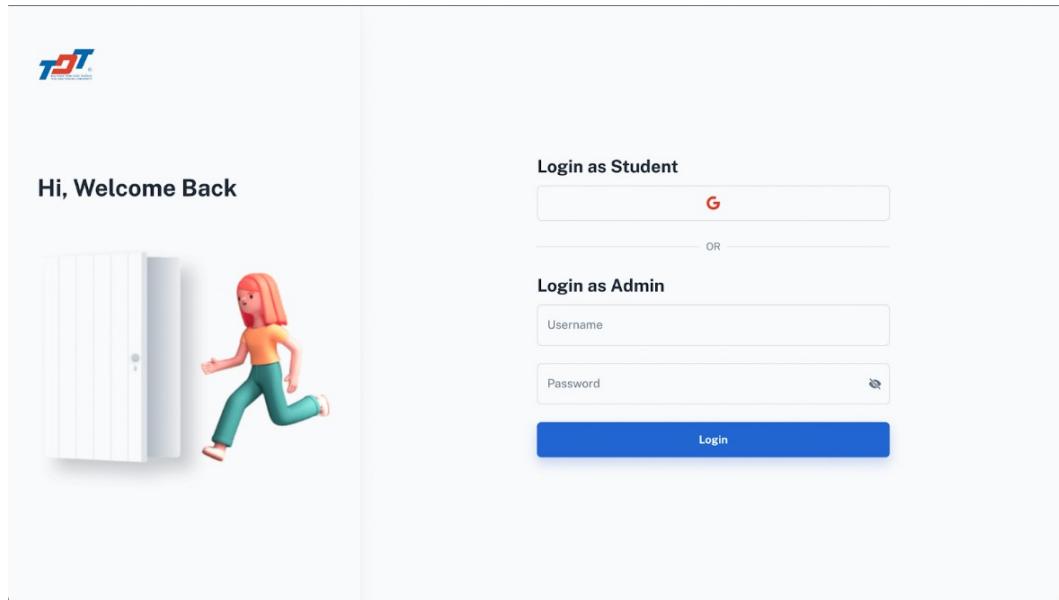
webpack compiled successfully

```

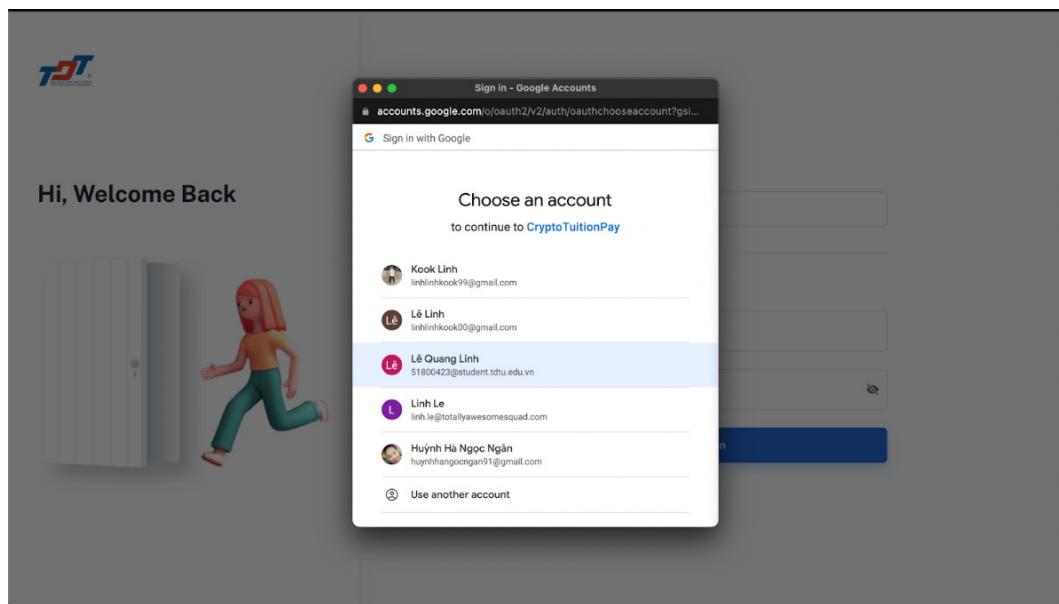
Hình 4.5: Chạy lệnh Frontend

4.2 Kết quả thực nghiệm:

Trang đăng nhập:



Hình 4.6: Trang đăng nhập



Hình 4.7: Chọn tài khoản đăng nhập

Trang xem chi phí môn học:

The screenshot shows a student dashboard for 'Lê Quang Linh' (Student). The main section is titled 'Tuition'. A table displays financial information for the current semester:

Previous Pending Charges	Semester Tuition	Reduction	Total Tuition Unpaid	Total Tuition Paid	Remaining Unpaid Tuition	Status
(1)	(2)	(3)	(4) = (1) + (2) - (3)	(5)	(6) = (4) - (5)	(7)

Courses that have included tuition fees in the semester

Hình 4.8: Trang xem học phí

The screenshot shows a student dashboard for 'Lê Quang Linh' (Student). The main section is titled 'Tuition'. A table displays financial information for the Fall 2021 semester:

Previous Pending Charges	Semester Tuition	Reduction	Total Tuition Unpaid	Total Tuition Paid	Remaining Unpaid Tuition	Status
(1)	(2)	(3)	(4) = (1) + (2) - (3)	(5)	(6) = (4) - (5)	(7)
0	17,250,000	0	17,250,000	17,250,000	0	Paid

Courses that have included tuition fees in the semester

Hình 4.9: Học phí đã thanh toán

Trang chi phí môn học chưa được thanh toán:

The screenshot shows a tuition payment interface. At the top, there is a logo for TDTU and a user profile for Lê Quang Linh, Student. A search bar and a connect wallet button are also present. The main section is titled "Tuition" and shows a table for the "Spring 2022" semester. The table has columns for Previous Pending Charges, Semester Tuition, Reduction, Total Tuition Unpaid, Total Tuition Paid, Remaining Unpaid Tuition, and Status. The data is summarized in the following table:

Previous Pending Charges	Semester Tuition	Reduction	Total Tuition Unpaid	Total Tuition Paid	Remaining Unpaid Tuition	Status
(1)	(2)	(3)	(4) = (1) + (2) - (3)	(5)	(6) = (4) - (5)	(7)
0	13,800,000	0	13,800,000	0	13,800,000	Unpaid

Below the table, a button labeled "Pay Tuition Now" is visible. The footer of the page displays the text "Courses that have included tuition fees in the semester".

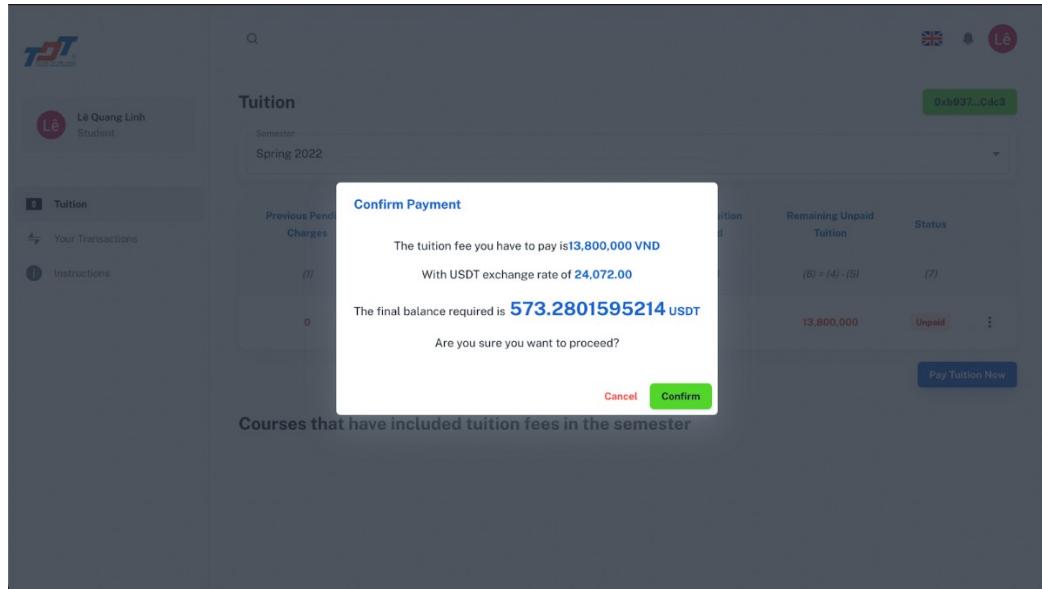
Hình 4.10: Học chưa được thanh toán

Kết nối ví để tiến hành thanh toán học phí:

The screenshot shows a tuition payment interface with a Metamask wallet connection overlay. The overlay features a fox logo and the text "Welcome back! The decentralized web awaits". It includes fields for "Password" and a large blue "Unlock" button. Below the button are links for "Forgot password?" and "Need help? Contact MetaMask support". The main tuition payment interface is partially visible in the background.

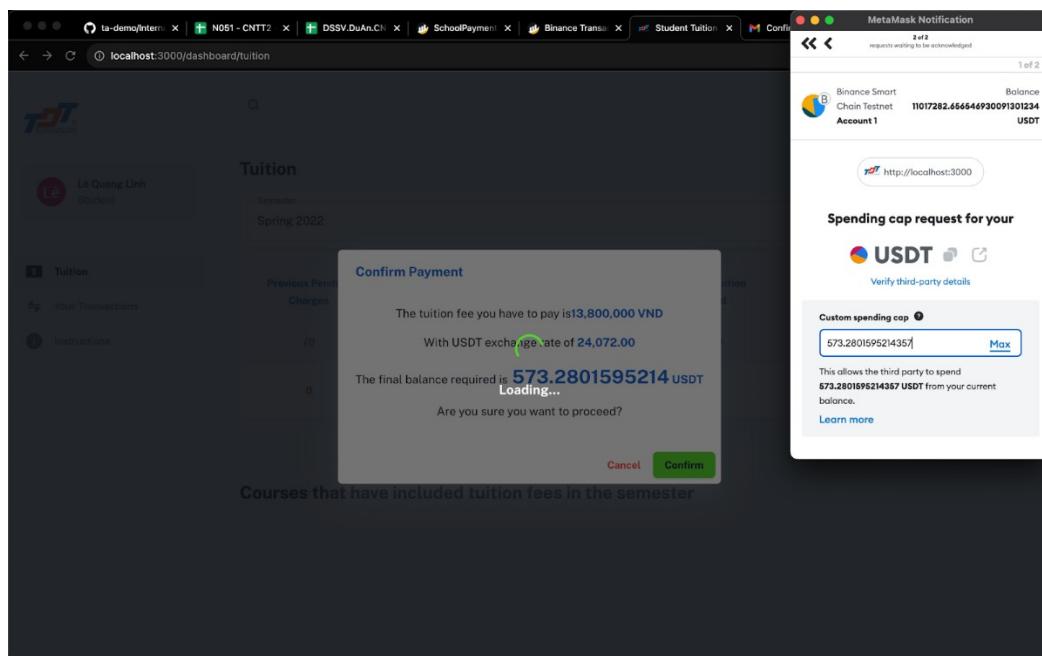
Hình 4.11: Kết nối ví thanh toán

Xem tỉ lệ chuyển đổi từ VND sang USDT trước khi xác nhận thanh toán:



Hình 4.12: Chuyển VND sang USDT

Tiến hành ký giao dịch:



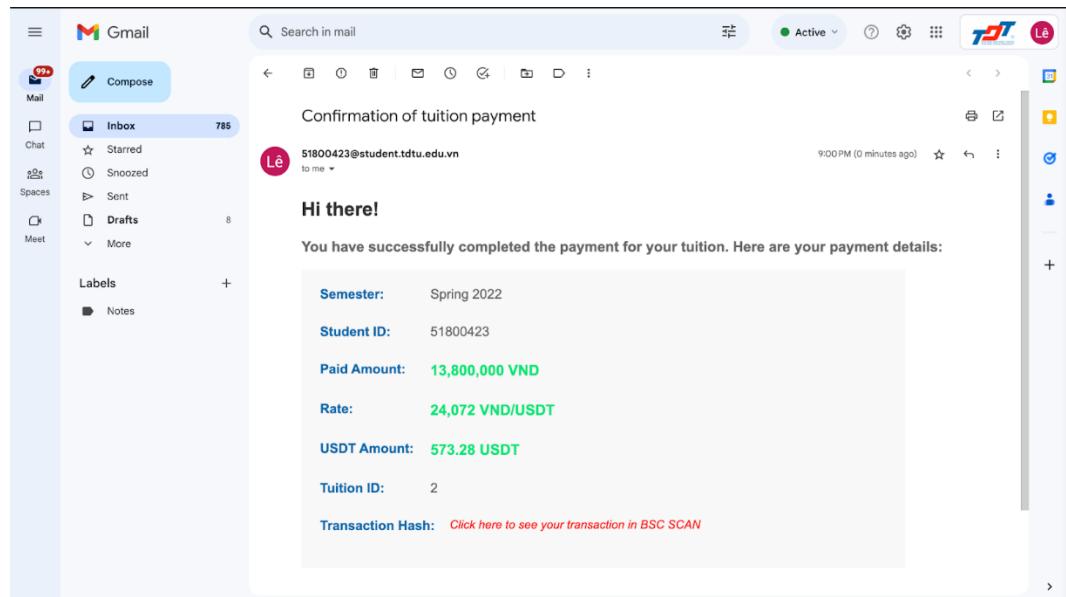
Hình 4.13: Xác nhận ký giao dịch

Xem lịch sử giao dịch:

	Wallet Address	Amount	Tuition Code	Time	Student Code	Status
	0x699...	USDT	0	0	0	0
6 Tuition	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.2801595214357	2	02/09/2023, 21:00:07	51800423	Complete
4 Your Transactions	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.208722741433	2	01/09/2023, 22:15:37	51800423	Complete
Instructions	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.0183116721339	2	01/09/2023, 16:46:19	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	573.042106137364	2	01/09/2023, 16:41:01	51800423	Complete
	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	572.970286692962	2	31/08/2023, 19:11:34	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.475898625145	2	30/08/2023, 15:08:10	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.310287751087	2	30/08/2023, 13:06:13	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.2629879538022	2	30/08/2023, 13:01:55	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.23934100505	2			Please check your student email for payment detail
	0xB937149b75C07d98192103e247026899177fcDc3	571.3339405481494	2	30/08/2023		School payment made successfully!!!

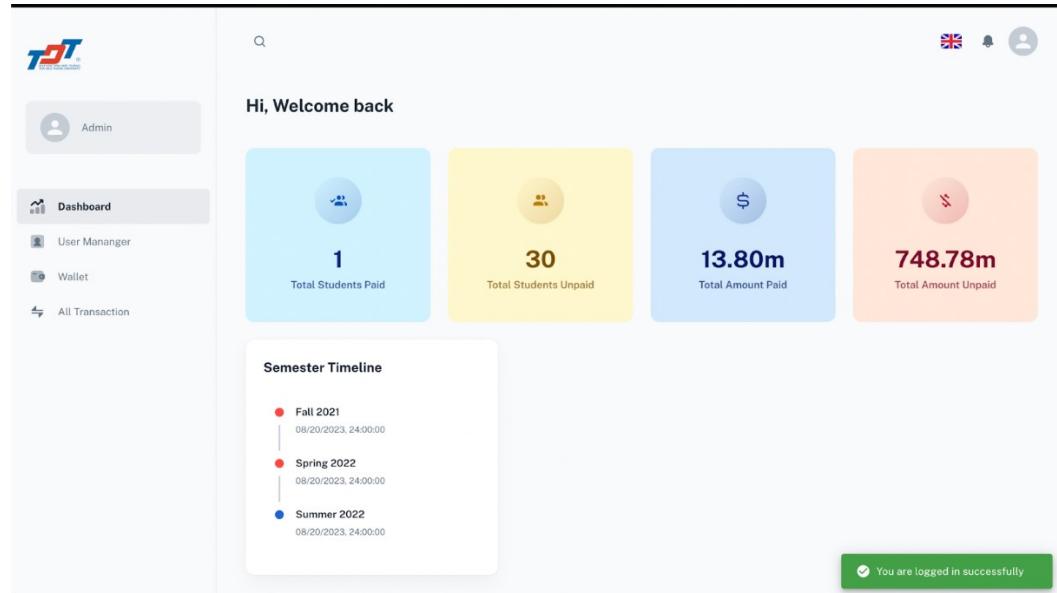
Hình 4.14: Xem lịch sử giao dịch

Email xác nhận đã thanh toán học phí thành công:



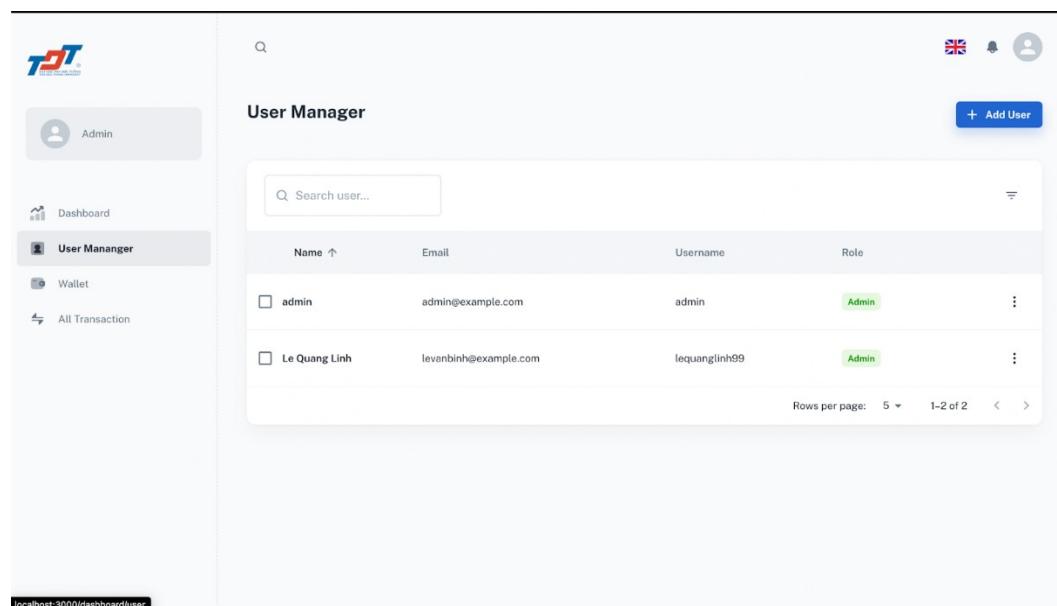
Hình 4.15: Email xác nhận

Admin xem thông kê học phí:

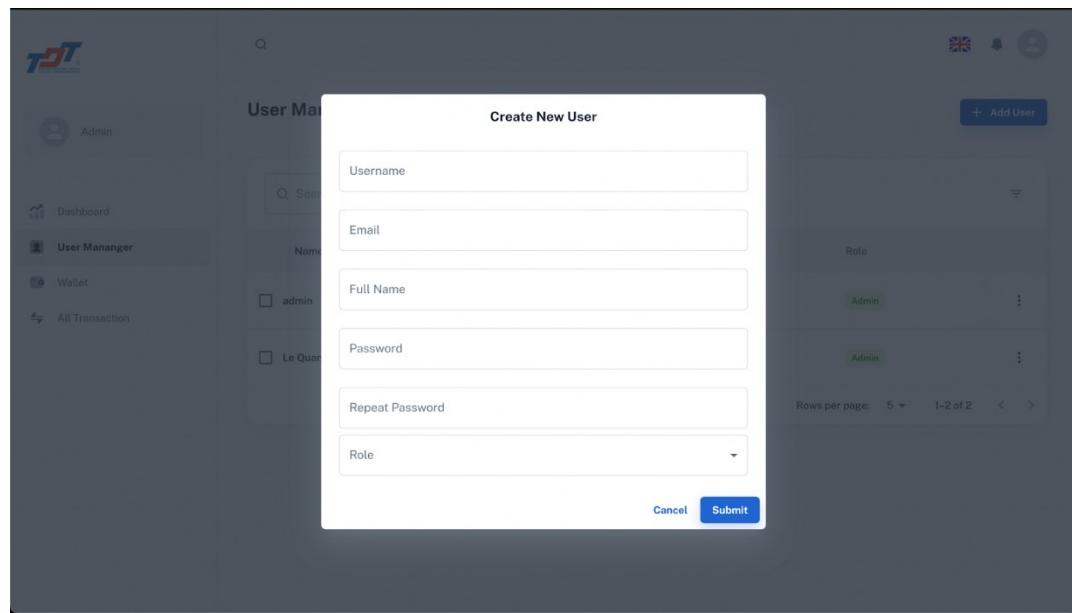


Hình 4.16: Xem thông kê học phí

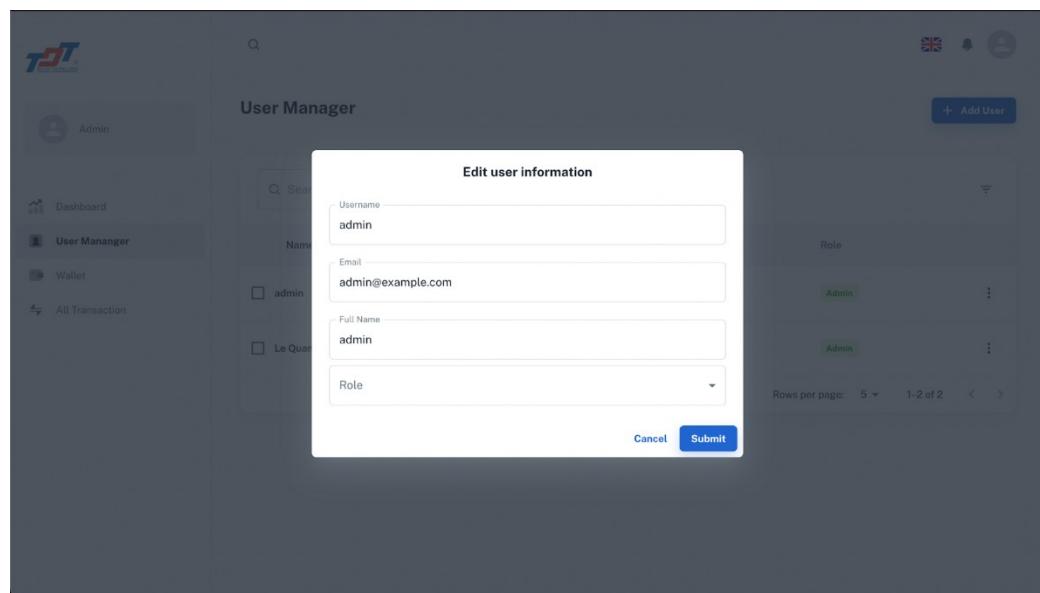
Admin quản lý tài khoản của quản trị viên:



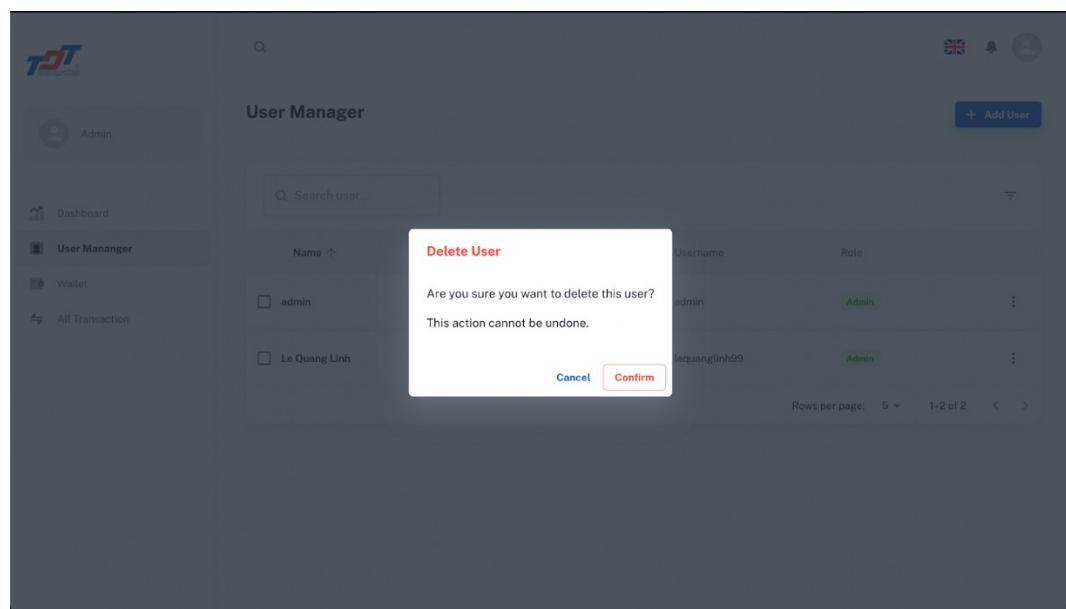
Hình 4.17: Quản lý quản trị viên



Hình 4.18: Thêm quản trị viên

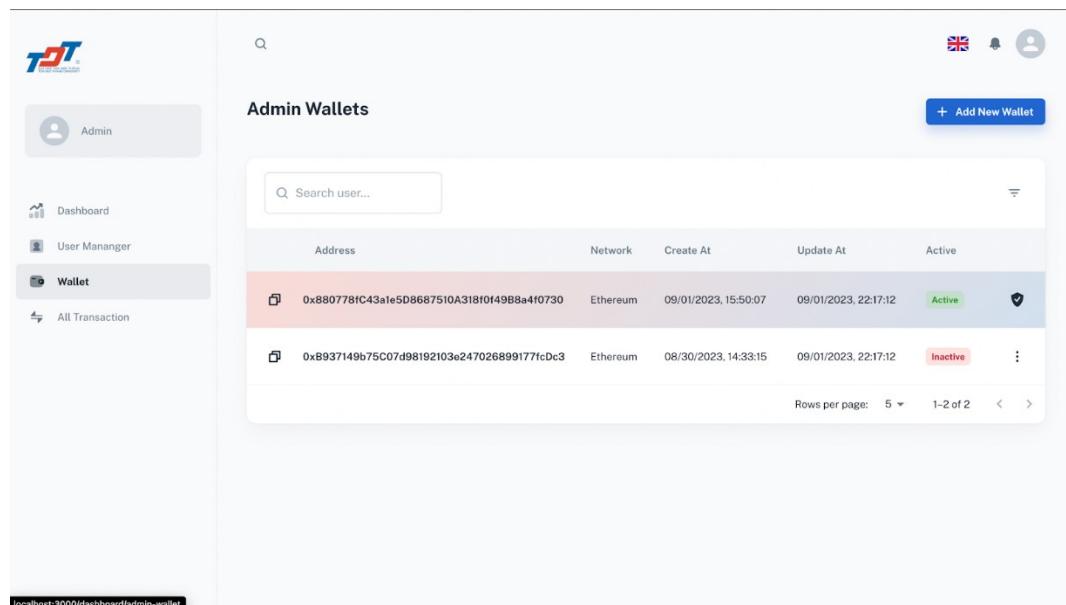


Hình 4.19: Sửa thông tin quản trị viên

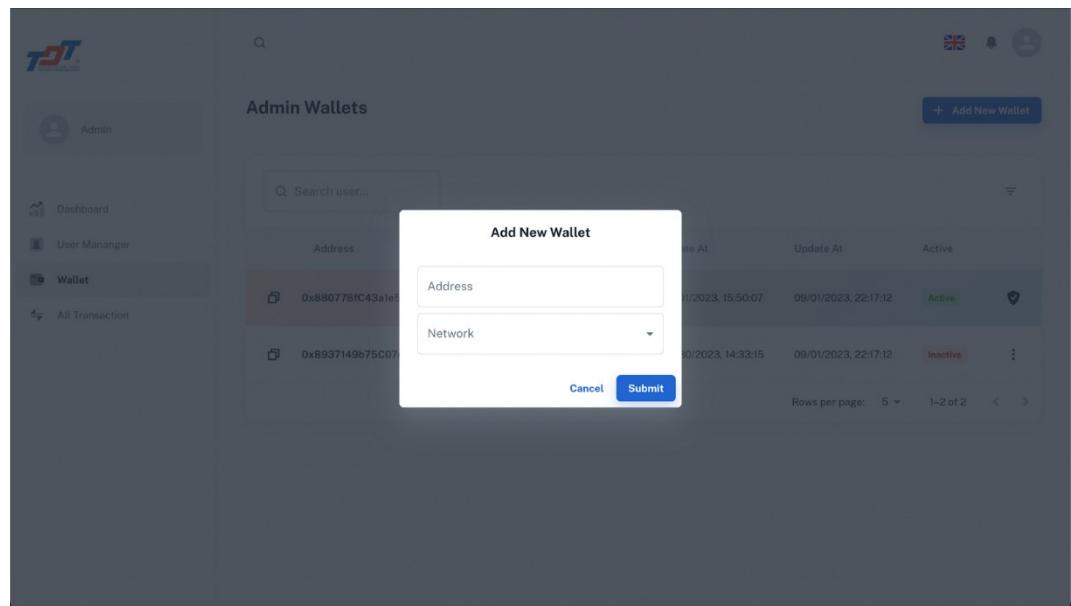


Hình 4.20: Xóa quản trị viên

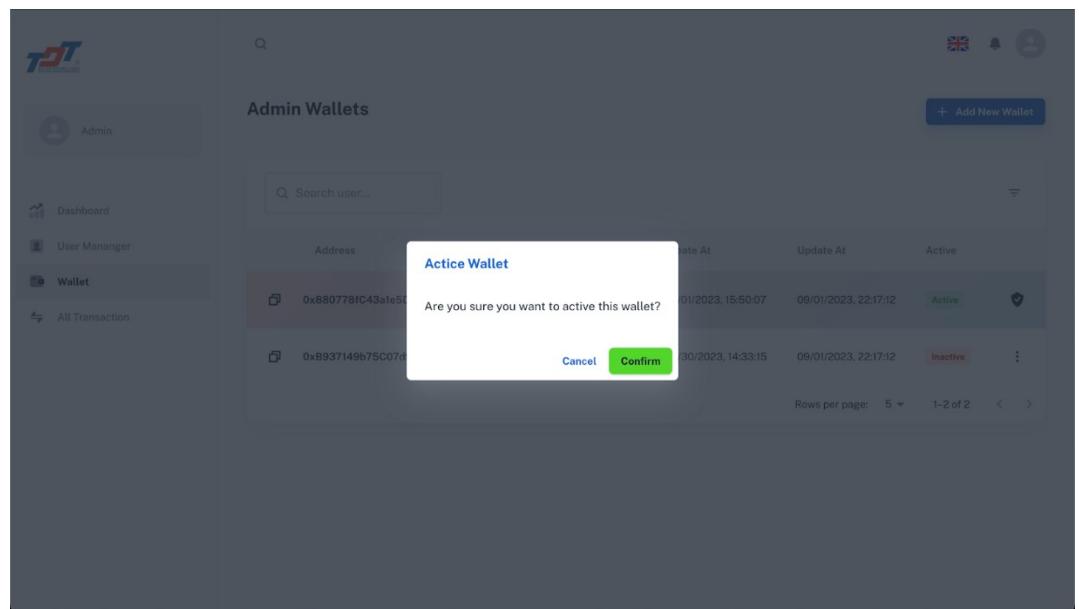
Admin quản lý ví nhện tiền:



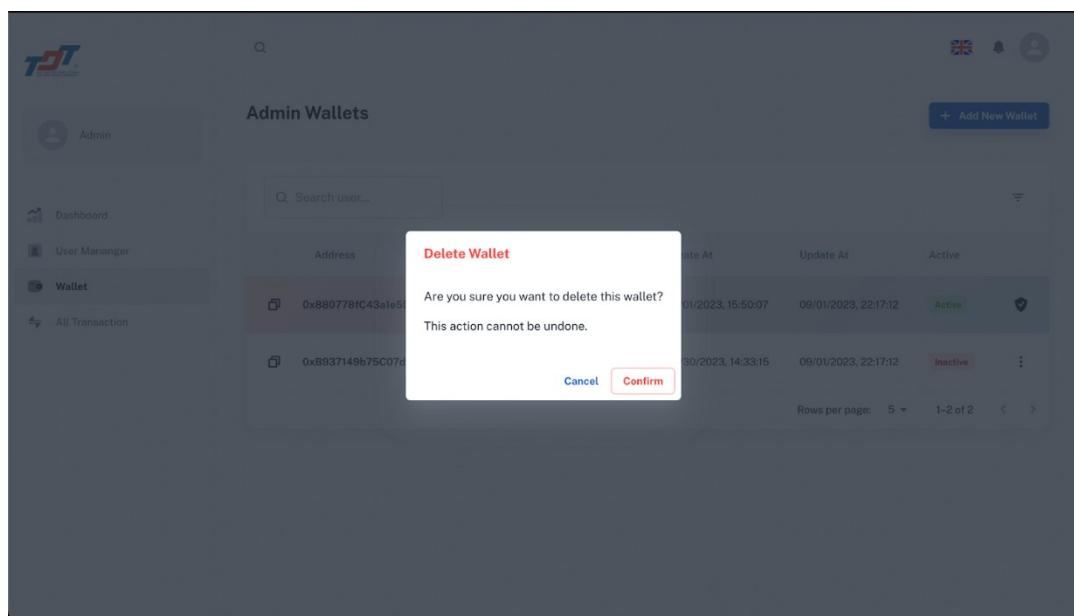
Hình 4.21: Quản lý ví nhện tiền



Hình 4.22: Thêm ví nhặt tiền



Hình 4.23: Kích hoạt ví



Hình 4.24: Xóa ví nhận tiền

Admin xem lịch sử giao dịch:

	Wallet Address	Amount	Tuition Code	Time	Student Code	Status
	0x699...	USDT	0	0	0	0
Dashboard	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.2801595214357	2	02/09/2023, 21:00:07	51800423	Complete
User Manager	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.208722741433	2	01/09/2023, 22:15:37	51800423	Complete
Wallet	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	573.0183116721339	2	01/09/2023, 16:46:19	51800423	Complete
All Transaction	0xB937149b75C07d98192103e247026899177fcDc3	573.042106137364	2	01/09/2023, 16:41:01	51800423	Complete
	0x880778fc43a1e5d8687510a318f0f49b8a4f0730	572.9707286692962	2	31/08/2023, 19:11:34	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.475898625145	2	30/08/2023, 15:08:10	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.3102877251087	2	30/08/2023, 13:06:13	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.2629879538022	2	30/08/2023, 13:01:55	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.23934100505	2	30/08/2023, 12:52:43	51800423	Complete
	0xB937149b75C07d98192103e247026899177fcDc3	571.3339405481494	2	30/08/2023, 12:50:04	51800423	Complete

Hình 4.25: Xem lịch sử giao dịch

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Với sự phát triển của công nghệ ngày nay, công nghệ Blockchain và Tiền điện tử (Crypto) đang dần trở nên phổ biến, được nhiều quốc gia phát triển và áp dụng thực tế vào một số hệ thống. Ở nhiều quốc gia trên thế giới nói chung và Việt Nam nói riêng, đã nhận thấy được tiềm năng của công nghệ Blockchain, muốn phát triển cũng như áp dụng công nghệ này vào các lĩnh vực tài chính – ngân hàng, logistic, y tế, giáo dục, ... Nhận thấy được sự phát triển của công nghệ này, chính vì thế nhóm chúng tôi đã chọn đề tài “Áp dụng công nghệ Blockchain vào việc thanh toán học phí qua Crypto” với mục tiêu tìm hiểu sâu hơn về công nghệ Blockchain và Crypto, cũng như phổ biến công nghệ này đến với nhiều người hơn, giúp cho mọi người có một cái nhìn khác về lợi ích của Blockchain, xóa bỏ định kiến về tiền ảo.

Sau quá trình nghiên cứu và thực hiện đề tài, các nghiên cứu và kết quả được thể hiện trong bài báo cáo này. Báo cáo đã trình bày tổng quan các cơ sở lý thuyết liên quan đến Blockchain, Smart Contract, Crypto, ... Giới thiệu về ngôn ngữ Solidity, các mô hình và công nghệ được áp dụng để xây dựng hệ thống. Phân tích về nhu cầu thanh toán học phí của sinh viên và những chức năng cần xây dựng cho hệ thống như xem học phí, thanh toán học phí, xem lịch sử giao dịch, xác nhận giao dịch qua email, ... Cách thức áp dụng công nghệ Blockchain vào thanh toán, tạo các ví điện tử, ký các smart contract, ... Cuối cùng là trình bày sản phẩm đạt được, giao diện của hệ thống, các chức năng đã được hoàn thiện. Công nghệ Blockchain và tiền điện tử cũng là một công nghệ mới và đang được phát triển trong những năm gần đây, đặc biệt là đối với Việt Nam. Việc phát triển công nghệ này cũng đang gặp nhiều thách thức do Việt Nam chưa có nhiều chuyên gia về Blockchain, các tài liệu nghiên cứu về chúng cũng còn bị giới hạn. Do giới hạn về khả năng nghiên cứu, kinh nghiệm và điều kiện về thời gian, dự án khó tránh khỏi nhiều thiếu sót về phương diện cơ sở lý thuyết cũng như chưa đạt được sản phẩm cuối cùng như mục tiêu đã đề ra. Rất

mong nhận được sự đóng góp, chỉ bảo tận tình của quý Thầy/Cô để dự án có thể hoàn thiện hơn nữa.

5.2 Hướng phát triển

Với mục tiêu phát triển hệ thống thanh toán học phí bằng tiền điện tử, chúng tôi mong muốn phát triển hoàn thiện hơn nữa một số chức năng của hệ thống, xây dựng thêm các chức năng mới, cho phép sử dụng nhiều loại tiền điện tử hơn nữa để thanh toán học phí. Với mong muốn công nghệ Blockchain được biết đến rộng rãi hơn, không chỉ trong lĩnh vực giáo dục mà các tổ chức thuộc nhiều lĩnh vực khác cũng có thể áp dụng công nghệ Blockchain vào hệ thống. Trong tương lai, công nghệ Blockchain sẽ ngày càng phát triển, tiền điện tử ngày càng trở nên phổ biến, hi vọng các quốc gia trên thế giới cũng như Việt Nam có thể cởi mở hơn và công nhận việc giao dịch, thanh toán bằng tiền điện tử là hợp pháp.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Gsolution. (2022, June 24). Công nghệ Blockchain là gì? Ứng dụng trong cuộc sống. Retrieved September 2, 2023, from Toàn Cầu | GSolution website: <https://gsolution.vn/cong-nghe-blockchain-la-gi-ung-dung-trong-cuoc-song/>

Exness. Blockchain Là Gì? Ứng Dụng Công Nghệ Blockchain 4.0 Vào Cuộc Sống. (n.d.). Retrieved September 2, 2023, from <https://www.exness.com/blog/blockchain-la-gi>

Tạp chí ngân hàng. Blockchain và ứng dụng trong hoạt động tài chính—Ngân hàng. (n.d.). Retrieved September 2, 2023, from <https://tapchinganhang.gov.vn/blockchain-va-ung-dung-trong-hoat-dong-tai-chinh-ngan-hang.htm>

MarginATM. Crypto là gì? Kiến thức đầu tư tiền điện tử cơ bản cho người mới. (n.d.). Retrieved September 3, 2023, from <https://marginatm.com/crypto-la-gi>
 helenngn. (2022, January 24). Crypto là gì? Cách đầu tư Cryptocurrency hiệu quả trong năm 2022. Retrieved September 3, 2023, from <https://remitano.com/forum/vn/post/4223-cryptocurrency-la-gi>

TesterProVN. React là gì? Lợi ích khi sử dụng React. (2021, September 15). Retrieved September 3, 2023, from <https://testerpro.vn/react-la-gi/>

MarginATM. Smart contract là gì? Cách hoạt động của hợp đồng thông minh. (n.d.). Retrieved September 3, 2023, from <https://marginatm.com/smart-contract-la-gi-pMAqLv4d>

DNSE. Smart Contract là gì? Cách thức hoạt động và ứng dụng của Smart Contract. (n.d.). Retrieved September 3, 2023, from Entrade X by DNSE website: <https://www.dNSE.com.vn/hoc/smart-contract-la-gi>

Tiếng Anh

Simplilearn. The Best Guide to Know What Is React [Updated]. (n.d.). Retrieved September 3, 2023, from Simplilearn.com website: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>

Kaspersky. What is cryptocurrency and how does it work? (2023, June 30). Retrieved September 3, 2023, from [Www.kaspersky.com](https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency) website: <https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency>

Simplilearn. What is Node.js: A Comprehensive Guide. (n.d.). Retrieved September 3, 2023, from Simplilearn.com website: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>

Simplilearn. What Is Solidity Programming in Ethereum |Simplilearn. (n.d.). Retrieved September 3, 2023, from Simplilearn.com website: <https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-solidity-programming>

Investopedia. Blockchain Facts: What Is It, How It Works, and How It Can Be Used. (n.d.). Retrieved September 3, 2023, from Investopedia website: <https://www.investopedia.com/terms/b/blockchain.asp>

Investopedia. What Are Smart Contracts on the Blockchain and How They Work. (n.d.). Retrieved September 3, 2023, from Investopedia website: <https://www.investopedia.com/terms/s/smарт-contracts.asp>

Simplilearn. What is Node.js: A Comprehensive Guide. (n.d.). Retrieved September 3, 2023, from Simplilearn.com website: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>