TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

XÂY DỰNG HỆ THỐNG CHỐNG GIẢ MẠO TÀI LIỆU VÀ VĂN BẢN

MÔN HỌC: AN TOÀN VÀ AN NINH THÔNG TIN

Sinh viên thực hiện: Huỳnh Tuấn Anh - 21004266
Lê Thị Ngọc Hân - 21004092
Phan Phương Mỹ Huyền - 21004267
Khóa: 46
Người hướng dẫn: ThS. Trần Thái Bảo

Vĩnh Long, năm 2024

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

XÂY DỰNG HỆ THỐNG CHỐNG GIẢ MẠO TÀI LIỆU VÀ VĂN BẢN

MÔN HỌC: AN TOÀN VÀ AN NINH THÔNG TIN

Sinh viên thực hiện: Huỳnh Tuấn Anh – 21004266

Lê Thị Ngọc Hân – 21004092

Phan Phương Mỹ Huyền – 21004267

Khóa: 46

Người hướng dẫn: ThS Trần Thái Bảo

Vĩnh Long, năm 2024

NHẬN XÉT VÀ ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN

- Ý thức thực hiện: .....

.....

.....

- Nội dung thực hiện: .....

.....

.....

- Hình thức trình bày: .....

.....

.....

- Tổng hợp kết quả:

□ Tổ chức báo cáo trước hội đồng

□ Tổ chức chấm thuyết minh

Vĩnh Long, ngày ..... tháng ….. năm ……

Người hướng dẫn

(Ký và ghi rõ họ, tên)

LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn chân thành thầy Trần Thái Bảo đã tận tình hướng dẫn, truyền đạt kiến thức, kinh nghiệm cho chúng em trong suốt quá trình thực hiện bài báo cáo này.

Xin gửi lời cảm ơn chân thành đến toàn thể quý Thầy cô Khoa Công nghệ Thông tin, Trường Đại học Sư Phạm Kỹ Thuật Vĩnh Long. Chúng em vô cùng biết ơn về những kiến thức quý báu mà quý Thầy cô đã tận tình truyền dạy cho chúng em trong suốt thời gian qua. Những kiến thức và sự hướng dẫn tận tình của quý Thầy cô đã giúp em hoàn thành tốt quá trình nghiên cứu và thực hiện báo cáo.

Dù đã cố gắng hết sức, nhưng do thời gian thực hiện có hạn và kinh nghiệm còn chưa nhiều, báo cáo của em vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự chia sẻ, cảm thông và góp ý quý báu từ quý Thầy cô, cũng như các bạn, để em có thể bổ sung, nâng cao kiến thức và kỹ năng cho việc học tập, nghiên cứu và công việc trong tương lai.

Cuối cùng, em kính chúc quý Thầy cô luôn dồi dào sức khỏe, nhiệt huyết và niềm tin để tiếp tục sứ mệnh cao quý của mình - truyền đạt kiến thức cho thế hệ mai sau.
Một lần nữa, em xin chân thành cảm ơn quý Thầy cô!
Trân trọng

LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan rằng đề tài này đã được thực hiện một cách trung thực và không sao chép hoặc sử dụng kết quả từ bất kỳ đề tài nghiên cứu nào tương tự. Mọi hỗ trợ trong quá trình thực hiện bài tiểu luận này đã được ghi rõ nguồn gốc và được phép công bố.

Chúng em cam kết chịu trách nhiệm hoàn toàn nếu phát hiện bất kỳ hành vi sao chép kết quả nghiên cứu từ bất kỳ đề tài khác nào. Chúng em xin tuân thủ nguyên tắc trung thực và tôn trọng công sức nghiên cứu của người khác.

MỤC LỤC

DANH MỤC HÌNH

MỞ ĐẦU

Trong thời đại chuyển đổi số, việc bảo vệ và quản lý tài liệu số trở thành một thách thức lớn đối với các tổ chức và cá nhân. Các hành vi giả mạo tài liệu không chỉ gây tổn thất về tài chính mà còn làm suy giảm niềm tin vào các hệ thống quản lý thông tin. Với sự phát triển của công nghệ blockchain – một giải pháp đột phá trong việc lưu trữ dữ liệu minh bạch, bảo mật và không thể thay đổi – việc ứng dụng công nghệ này để xây dựng hệ thống chống giả mạo tài liệu đã mở ra nhiều triển vọng mới.

Hệ thống sử dụng blockchain không chỉ ngăn chặn các hành vi gian lận mà còn cung cấp một môi trường minh bạch, phi tập trung, nơi các bên liên quan có thể dễ dàng xác thực nguồn gốc và tính toàn vẹn của tài liệu. Điều này đặc biệt quan trọng trong các lĩnh vực yêu cầu mức độ tin cậy cao như giáo dục, y tế, tài chính, và quản lý nhà nước.

Bài báo cáo này sẽ trình bày tổng quan về công nghệ blockchain, phân tích các lợi ích khi áp dụng blockchain trong xây dựng hệ thống chống giả mạo tài liệu, đồng thời đề xuất một mô hình thực tiễn để triển khai. Qua đó, chúng em hy vọng mang lại cái nhìn rõ ràng về tiềm năng của công nghệ blockchain trong việc giải quyết các vấn đề liên quan đến an ninh và bảo mật dữ liệu trong thời đại số hóa.

Chương 1 TỔNG QUAN ĐỀ TÀI

1.1 Giới thiệu:

Hệ thống chống giả mạo tài liệu và văn bản là một giải pháp công nghệ hiện đại nhằm đảm bảo tính xác thực và toàn vẹn của thông tin trong các loại tài liệu, văn bản. Trước sự gia tăng của các hành vi giả mạo và thao túng dữ liệu, đặc biệt trong các lĩnh vực quan trọng như tài chính, pháp lý, giáo dục và y tế, việc xây dựng một hệ thống có khả năng phát hiện và ngăn chặn các hành vi này trở thành một yêu cầu cấp thiết. Thông qua việc áp dụng các công nghệ tiên tiến như chữ ký số, mã hóa, trí tuệ nhân tạo và blockchain, hệ thống không chỉ bảo vệ quyền lợi của người sử dụng mà còn góp phần xây dựng một môi trường làm việc và giao dịch an toàn, minh bạch, đáng tin cậy.

1.1.1 Lịch sử

Làm giả tài liệu và văn bản đã là một tội ác được biết đến từ khi con người biết đến chữ viết và bảng chữ cái. Theo " Forensic Document Examination ", luật chống làm giả có từ năm 80 trước Công nguyên, khi người La Mã cấm làm giả các tài liệu được sử dụng để chuyển nhượng quyền sở hữu bất hợp pháp từ chủ đất này sang chủ đất khác.

Ngày nay, làm giả giấy tờ vẫn là một tội phạm kinh tế phổ biến, thường liên quan đến việc thay đổi một giấy tờ thật và tạo ra một giấy tờ giả. Ví dụ, lừa dối ai đó bằng cách hát tên của người khác. Khi nói đến giấy tờ tùy thân , đây cũng là một hành vi phổ biến nhằm sử dụng danh tính của người khác để đạt được lợi ích cá nhân hoặc thực hiện các tội ác khác. Ví dụ, vay tiền bằng giấy tờ tùy thân của người khác.

Các loại giấy tờ giả phổ biến nhất cho đến ngày nay là hộ chiếu giả và giấy phép lái xe. Đôi khi, trẻ vị thành niên sử dụng giấy tờ tùy thân giả cho các dịch vụ hạn chế độ tuổi , như các nền tảng thương mại điện tử bán rượu. Đây là lý do tại sao mọi loại hình doanh nghiệp, cả hoạt động trực tuyến và tại các địa điểm thực tế, cần xây dựng các hệ thống xác minh phù hợp có thể chống lại tình trạng làm giả giấy tờ theo cách không làm gián đoạn trải nghiệm của khách hàng .

1.1.2 Tầm quan trọng của việc chống giả mạo tài liệu và văn bản

Việc chống giả mạo tài liệu và văn bản giữ vai trò đặc biệt quan trọng trong việc bảo vệ tính toàn vẹn, đáng tin cậy của dữ liệu và đáp ứng yêu cầu pháp lý trong nhiều lĩnh vực như tài chính, giáo dục, y tế và pháp luật. Các tài liệu, bao gồm bằng

cấp, giấy chứng nhận, hợp đồng, hoặc thông tin kinh doanh, cần được thiết kế theo tiêu chuẩn bất biến (immutable) để đảm bảo không thể bị sửa đổi sau khi phát hành và chống giả mạo (tamper-proof) nhằm ngăn chặn các hành vi xâm phạm trái phép.

Blockchain đã nổi lên như một công nghệ lý tưởng giúp mã hóa và lưu trữ dữ liệu minh bạch, an toàn, tạo ra hệ thống không thể thay đổi, đồng thời cho phép xác minh nguồn gốc tài liệu. Điều này không chỉ giảm thiểu nguy cơ gian lận mà còn tăng cường lòng tin giữa các bên liên quan, đặc biệt trong quản lý bằng cấp, giấy phép và các loại chứng từ quan trọng. Bằng cách tích hợp các giải pháp chống giả mạo, doanh nghiệp và tổ chức không chỉ đáp ứng yêu cầu tuân thủ quy định mà còn đảm bảo các tài liệu mang tính xác thực cao, hỗ trợ xây dựng danh tiếng trong môi trường kỹ thuật số.

Chương 2 CƠ SỞ LÝ THUYẾT

2.1 Giả mạo tài liệu và văn bản

2.1.1 Định nghĩa về giả mạo tài liệu và văn bản

Giả mạo tài liệu và văn bản là hành vi tạo ra hoặc chỉnh sửa các tài liệu, văn bản để đánh lừa hoặc đạt được lợi ích bất hợp pháp. Các loại tài liệu thường bị giả mạo bao gồm bằng cấp, giấy chứng nhận, hợp đồng, hóa đơn, và các tài liệu nhận dạng cá nhân như căn cước công dân, hộ chiếu. Hậu quả của việc giả mạo không chỉ gây thiệt hại về tài chính mà còn ảnh hưởng đến uy tín của các tổ chức, cá nhân và phá vỡ niềm tin trong giao dịch.

2.1.2 Các loại giả mạo tài liệu và văn bảng

Giả mạo vật lý: Chỉnh sửa hoặc tạo tài liệu giả trực tiếp trên bản giấy, ví dụ như tẩy xóa, thêm chi tiết, hoặc làm giả con dấu và chữ ký.

Giả mạo kỹ thuật số: Thay đổi nội dung thông qua chỉnh sửa file điện tử bằng phần mềm như Photoshop hoặc tạo tài liệu giả hoàn toàn.

Sử dụng tài liệu giả: Cố tình cung cấp hoặc sử dụng các tài liệu bị làm giả để lừa đảo hoặc thu lợi bất hợp pháp, ví dụ: bằng cấp giả, giấy phép giả.

Chèn nội dung không hợp lệ: Thay đổi, thêm hoặc bớt dữ liệu trong văn bản gốc mà không có sự cho phép của bên liên quan.

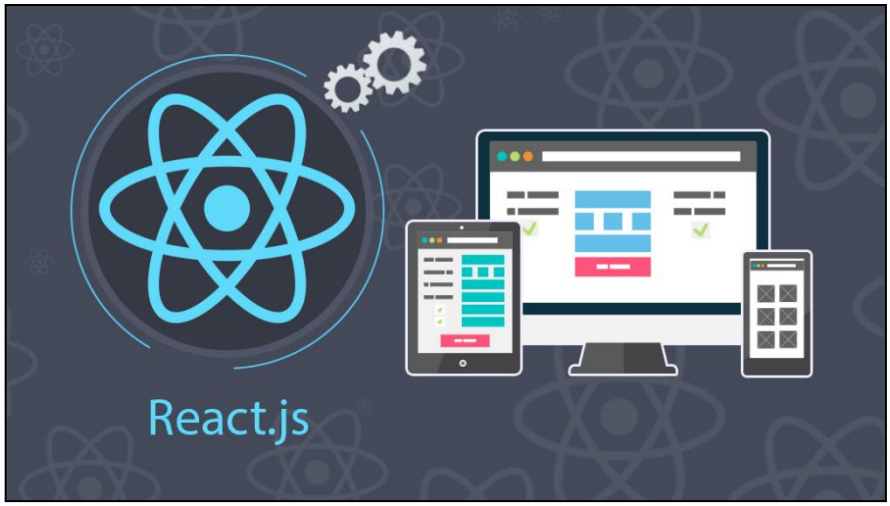
2.2 Giới thiệu các công cụ dùng để xây dựng FE

2.2.1 ReactJS

React (Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dung nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dung các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.

Nó cũng sử dụng khái niệm là Virtual DOM (DOM ảo). Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là Reactjs chỉ render những component hay subcomponent nào thực sự thay đổi.

Mỗi framework hay thư viện luôn có những ưu và nhược điểm riêng. Sau đây tôi xin trình bày về ưu điểm và nhược điểm của nó.



Hình 2.1 Ảnh minh họa ReactJS

Ưu điểm:

ReactJS phù hợp với đa thể loại website: ReactJS hỗ trợ đáng kể trong việc khởi tạo website đơn giản hơn vì bạn chỉ cần sử dụng JavaScript và HTML, các công cụ và tính năng khác đã được ReactJS cung cấp để áp dụng cho nhiều trường hợp khác nhau.

Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.

Dễ dàng tái sử dụng các Component: Bạn chỉ cần tốn thời gian vào việc xây dựng ban đầu và có thể sử dụng lại trong các dự án sau này nếu bạn xây dựng các component đủ linh hoạt và tốt, có thể đáp ứng các yêu cầu của nhiều dự án khác nhau. Ngoài ra, không chỉ có ReactJS mà các framework hiện nay như Flutter cũng cho phép chúng ta thực hiện điều này.

Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.

Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Nhược điểm:

Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax

Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

React khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh

Khó tiếp cận cho người mới học web

2.2.2 TailwindCSS

Tailwind là một framework CSS mạnh mẽ theo phong cách utility-first, giúp tối ưu hóa quá trình xây dựng giao diện người dùng tùy chỉnh. Với một loạt các lớp tiện ích mở rộng, Tailwind cho phép các nhà phát triển nhanh chóng thiết kế và bố trí trang web một cách hiệu quả. Điểm nổi bật của Tailwind chính là tính linh hoạt, cho phép tùy chỉnh và tinh chỉnh thiết kế dễ dàng. Sử dụng Tailwind giúp tiết kiệm thời gian và công sức, đồng thời tạo ra các ứng dụng web đẹp mắt và đáp ứng tốt trên nhiều thiết bị.

Hình 2.2 Ảnh minh họa TailwindCSS

Ưu điểm:

Một trong những ưu điểm của Tailwind CSS là khả năng mở rộng. Bằng cách sử dụng các lớp CSS có sẵn và kết hợp chúng, chúng ta có thể tạo ra những lớp CSS mới phù hợp với yêu cầu cụ thể của dự án. Điều này cho phép chúng ta linh hoạt trong việc tạo ra giao diện độc đáo và tùy chỉnh theo ý muốn.

Với Tailwind CSS, chúng ta không cần phải viết CSS tùy chỉnh nhiều như trước đây. Thay vào đó, chúng ta sử dụng các lớp CSS có sẵn để áp dụng các quy tắc và kiểu dáng cho các phần tử trong giao diện. Điều này giúp tăng hiệu suất phát triển, giảm thời gian viết CSS từ đầu và tạo ra giao diện một cách nhanh chóng.

Tiết kiệm thời gian:

Với các lớp tiện ích được định nghĩa sẵn, bạn có thể xây dựng và tùy chỉnh giao diện người dùng một cách nhanh chóng mà không cần viết lại CSS từ đầu.

Tính linh hoạt cao:

Dễ dàng kết hợp các lớp tiện ích để tạo giao diện độc đáo.

Hỗ trợ cấu hình qua file tailwind.config.js để tùy chỉnh màu sắc, font chữ, kích thước, v.v.

Không cần đặt tên lớp phức tạp:

Bạn không cần phải suy nghĩ về việc đặt tên lớp CSS phức tạp, thay vào đó, chỉ cần sử dụng các lớp tiện ích được cung cấp bởi Tailwind CSS.

Khả năng mở rộng và bảo trì:

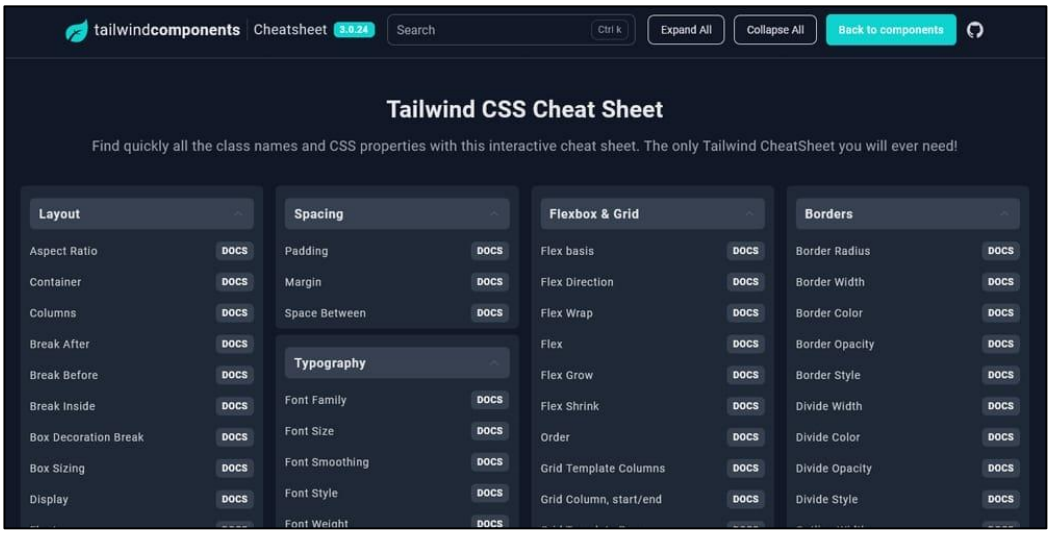
Tailwind CSS giúp mã nguồn dễ đọc, dễ hiểu và dễ bảo trì hơn nhờ vào cấu trúc rõ ràng và các lớp tiện ích có tên gọi tự giải thích.

Hỗ trợ responsive tốt:

Tailwind CSS cung cấp các lớp tiện ích responsive giúp tối ưu giao diện trên mọi kích thước màn hình.

Chế độ Just-In-Time (JIT mode):

Với chế độ Just-In-Time (JIT), Tailwind CSS chỉ tạo ra các lớp CSS cần thiết trong quá trình phát triển, giúp giảm kích thước tệp CSS và tăng tốc độ tải trang.



Hình 2.3 Các thành phần của Tailwind bảng cheat sheet CSS của Tailwind

2.3 Giới thiệu về các công cụ dùng để xây dựng BE

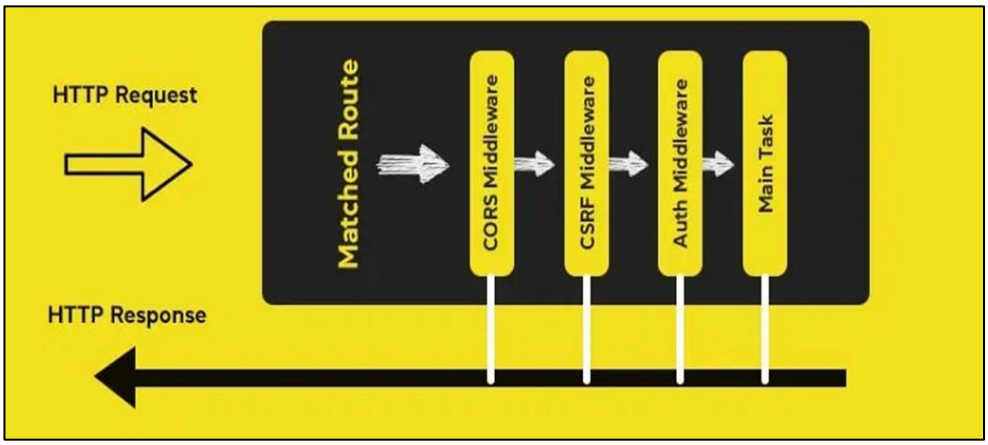
2.3.1 ExpressJS

ExpressJS là một framework miễn phí và mã nguồn mở dành cho việc xây dựng ứng dụng web trên nền tảng Node.js. Với ExpressJS, người dùng có thể nhanh chóng thiết kế và phát triển các ứng dụng web và API một cách dễ dàng và tiện lợi. Điều này đặc biệt hữu ích cho các nhà phát triển và lập trình viên đã quen thuộc với JavaScript.

Vì ExpressJS là một framework phát triển ứng dụng web trên Nodejs, các lập trình viên có thể sử dụng mã đã có sẵn để xây dựng các ứng dụng web đơn trang (SPA), đa trang hoặc kết hợp cả hai. Ngoài ra, ExpressJS còn cung cấp một kiến trúc MVC (Model-View-Controller) hữu ích để tổ chức các ứng dụng web phía máy chủ.

Với các tính năng hỗ trợ nâng cao của Nodejs, ExpressJS giúp giảm bớt sự phức tạp của việc xây dựng API hiệu quả. Nếu không sử dụng ExpressJS, lập trình viên sẽ phải viết rất nhiều mã để xử lý các yêu cầu khác nhau.

Nhưng ExpressJS giúp cho việc này trở nên đơn giản hơn đáng kể. Với sự hỗ trợ từ cộng đồng và các thư viện bên thứ ba, ExpressJS là một công cụ hữu ích cho việc xây dựng các ứng dụng web và API trên Nodejs.

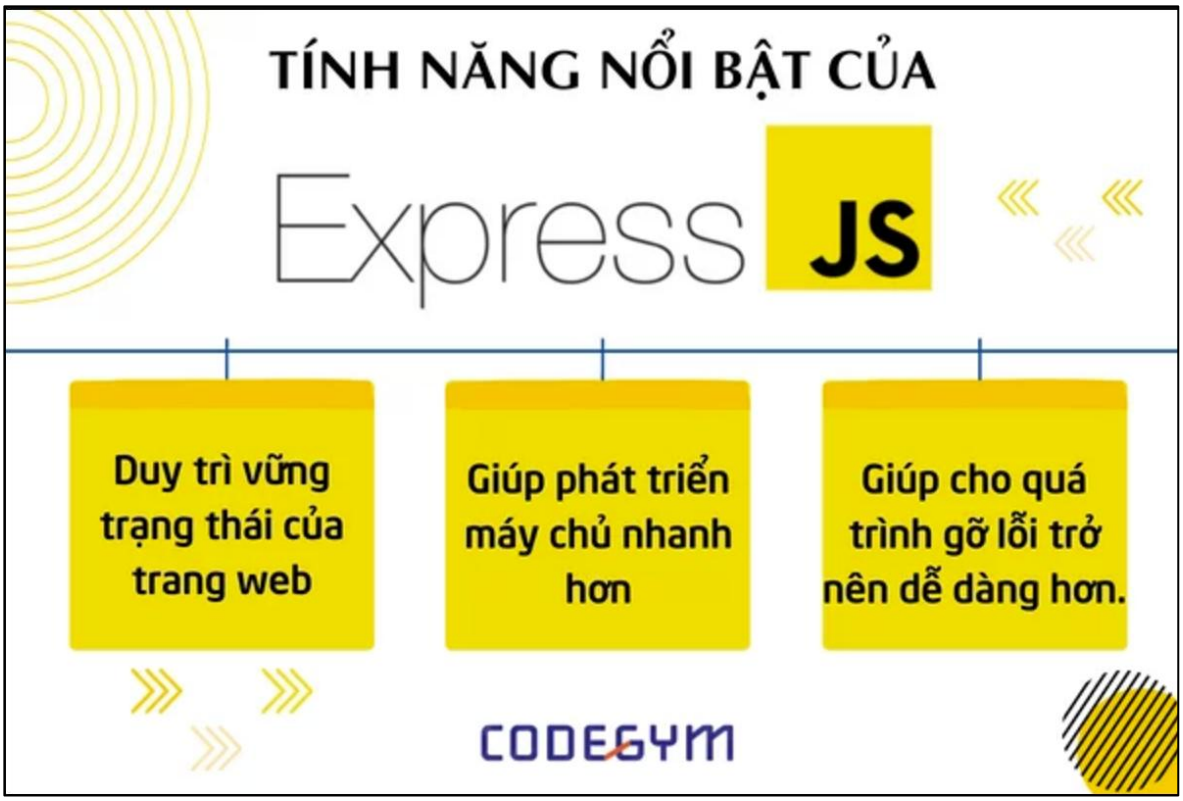


Hình 2.4 Quy trình xử lý yêu cầu HTTP trong một ứng dụng web

ExpressJS có thể giúp phát triển máy chủ nhanh hơn bằng việc cung cấp các tính năng phổ biến của Nodejs dưới dạng hàm có thể tái sử dụng. Ngoài ra, ExpressJS cũng đóng vai trò là phần mềm trung gian, giúp tổ chức các chức năng khác nhau của ứng dụng.

ExpressJS cũng cung cấp một cơ chế định tuyến nâng cao có thể giúp duy trì vững trạng thái của trang web. Nó cũng cung cấp các công cụ tạo khuôn mẫu cho phép các nhà phát triển tạo được nội dung động trên các trang web này bằng việc bắt đầu xây dựng các mẫu HTML ở phía máy chủ.

Gỡ lỗi là yếu tố quan trọng để phát triển các ứng dụng web và ExpressJS cung cấp cơ chế giúp xác định chính xác phần ứng dụng web có lỗi, giúp cho quá trình gỡ lỗi trở nên dễ dàng hơn.



Hình 2.5 Tính năng nổi bật của ExpressJS

Các tính năng chính của ExpressJS :

+ Templating Engines

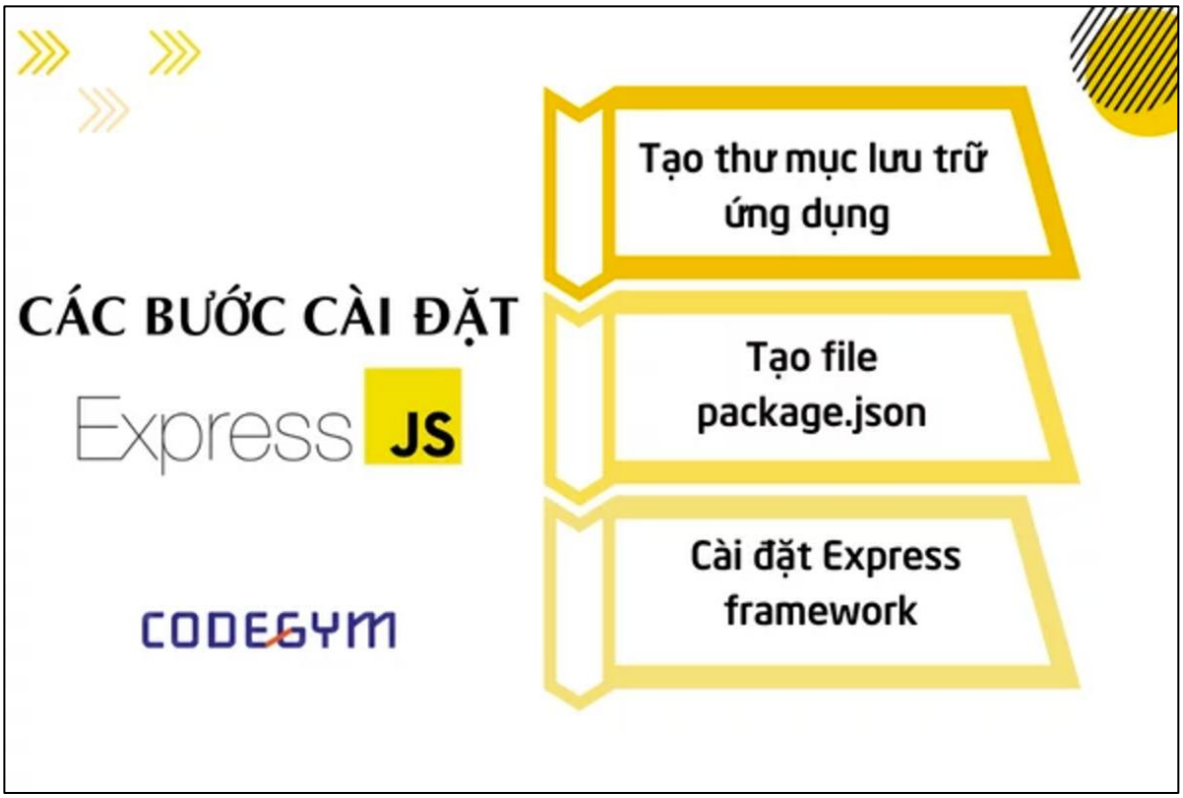
+ Phục vụ các tệp tĩnh

+ Định tuyến

+ Middleware

+ Hỗ trợ xây dựng RESTful API

Các cài đặt ExpressJS



Hình 2.6 Cài đặt ExpressJS

Để bắt đầu xây dựng ứng dụng của bạn bằng ExpressJS, bạn có thể thực hiện các bước sau:

Bước 1: Tạo thư mục lưu trữ ứng dụng của bạn, ví dụ như E:\express\myapp, sau đó mở Command Prompt hoặc Terminal và nhập lệnh cd E:\express\myapp để truy cập vào thư mục này.

Bước 2: Tạo file package.json cho ứng dụng bằng lệnh npm init. Lệnh này sẽ hướng dẫn bạn nhập tên, mô tả và phiên bản của ứng dụng. Nếu bạn muốn sử dụng tên file index.js làm điểm vào của ứng dụng, hãy nhấn Enter để chấp nhận giá trị mặc định cho trường entry point.

Bước 3: Cài đặt Express framework bằng lệnh npm install express –save. Điều này sẽ cài đặt Express và các module quan trọng khác như body-parser, cookie-parser và multer.

Bước 4: (Tuỳ chọn) Nếu bạn cần sử dụng các module trung gian khác, bạn có thể cài đặt chúng bằng lệnh npm install <module-name> –save.

2.4 Cơ sở dữ liệu (Database)

2.4.1 Cơ sở dữ liệu tập trung MongoDB

MongoDB là một cơ sở dữ liệu NoSQL, một dạng database hướng tài liệu. Chúng thường được sử dụng để lưu trữ dữ liệu khối lượng lớn. MongoDB không sử

dụng cấu trúc dạng bảng như relational database. Thay vào đó, MongoDB sẽ lưu trữ dữ liệu dưới dạng Document JSON. Vì vậy, mỗi một collection sẽ các các kích cỡ và các document khác nhau. Bên cạnh đó, việc các dữ liệu được lưu trữ trong document kiểu JSON dẫn đến chúng được truy vấn rất nhanh.

Hình 2.7 Cơ sở dữ liệu MongoDB

2.4.2 Cơ sở dữ liệu phân tán phi tập trung (Blockchain Network)

Trong blockchain, phi tập trung đề cập đến việc chuyển quyền kiểm soát và quyền ra quyết định từ một thực thể tập trung (cá nhân, tổ chức hoặc nhóm) sang một mạng phân tán. Các mạng phi tập trung cố gắng giảm mức độ tin tưởng mà những người tham gia phải đặt lên nhau và ngăn chặn khả năng họ thể hiện quyền lực hoặc kiểm soát lên nhau theo những cách làm suy giảm chức năng của mạng.

Phi tập trung không phải là một khái niệm mới. Khi xây dựng một giải pháp công nghệ, chúng ta sẽ thường xem xét ba kiến trúc mạng chính: tập trung, phân tán và phi tập trung. Mặc dù các công nghệ blockchain thường sử dụng các mạng phi tập trung, một ứng dụng blockchain không thể được phân loại một cách đơn giản là phi tập trung hay không. Thay vào đó, phi tập trung là một thang trượt và cần được áp dụng cho tất cả các khía cạnh của một ứng dụng blockchain. Bằng cách phân cấp phi tập trung quy trình quản lý và quyền truy cập vào tài nguyên trong một ứng dụng, bạn có thể đạt được dịch vụ tốt hơn và công bằng hơn. Phi tập trung thường có một vài sự đánh đổi như thông lượng giao dịch thấp hơn, tuy nhiên, theo cách lý tưởng thì sự đánh đổi là xứng đáng vì chúng mang lại mức độ ổn định và dịch vụ được cải thiện.

Các đặc điểm khi so sánh phi tập trung:

Phi tập trung nên được áp dụng khi hợp lý. Chỉ vì một ứng dụng sử dụng công nghệ blockchain không có nghĩa là nó phải hoàn toàn phi tập trung 100%. Mục tiêu của bất kỳ giải pháp blockchain nào là cung cấp những gì cần thiết cho người dùng giải pháp đó và điều này có thể bao gồm hoặc không bao gồm các mức độ phi tập trung nhất định. Để hiểu rõ hơn về các mạng phi tập trung, bảng dưới đây trình bày đặc điểm khi so sánh mạng phi tập trung với các mạng tập trung và phân tán phổ biến hơn.

	Tập Trung	Phân Tán	Phi Tập Trung
Tài nguyên mạng/phần cứng	Được duy trì và kiểm soát bởi một thực thể duy nhất tại một địa điểm tập trung	Trải rộng trên nhiều trung tâm dữ liệu và khu vực địa lý; thuộc sở hữu của nhà cung cấp mạng	Tài nguyên thuộc sở hữu và được chia sẻ bởi các thành viên trong mạng; khó khăn khi duy trì vì không ai sở hữu nó
Thành phần giải pháp	Được duy trì và kiểm soát bởi thực thể trung tâm	Được duy trì và kiểm soát bởi nhà cung cấp giải pháp	Mỗi thành viên đều có cùng bản sao chính xác của sổ cái phân tán
Dữ liệu	Được duy trì và kiểm soát bởi thực thể trung tâm	Thường được sở hữu và quản lý bởi khách hàng	Chỉ được thêm vào thông qua sự đồng thuận của nhóm
Kiểm soát	Được kiểm soát bởi thực thể trung tâm	Thông thường, trách nhiệm được chia sẻ giữa nhà cung cấp mạng, nhà cung cấp giải pháp và khách hàng	Không ai sở hữu dữ liệu và mọi người đều sở hữu dữ liệu
Điểm lỗi chí mạng đơn lẻ	Có	Không	Không
Khả năng chịu lỗi	Thấp	Cao	Cực cao
Bảo mật	Được duy trì và kiểm soát bởi thực thể trung tâm	Thông thường, trách nhiệm được chia sẻ giữa nhà cung cấp mạng, nhà cung cấp giải pháp và khách hàng	Tăng khi số lượng thành viên trong mạng tăng lên

Hiệu năng	Được duy trì và kiểm soát bởi thực thể trung tâm	Tăng khi tài nguyên mạng/phần cứng tăng quy mô theo tài nguyên và theo phiên bản	Giảm khi số lượng thành viên trong mạng tăng lên
Ví dụ	Hệ thống ERP	Điện toán đám mây	Blockchain

2.5 Ether (Thư viện dùng để kết nối ExpressJS và Blockchain Network)

Hình 2.8 Hình ảnh minh họa Ehters.js

Nói một cách đơn giản, ethers.js là một thư viện được viết bằng Javascript giúp Dapp tương tác với mạng Ethereum Blockchain.

Các tính năng nổi bật ethers.js gồm có:

- Giữ private key ở client một cách an toàn
- Import và export ***JSON wallets***
- Import và export ví theo chuẩn BIP 39
- Hỗ trợ ABI, ABIv2 và Human-Readable ABI
- Kết nối với Ethereum nodes thông qua nhiều provider như JSON-RPC, INFURA, Etherscan, Alchemy, Cloudflare, MetaMask ...
- Hỗ trợ ***ENS***
- Nhẹ (88kb khi nén và 284kb khi không nén)
- Hỗ trợ ***TypeScript***

2.6 HardHat

HardHat là một môi trường phát triển (development environment) cho các hợp đồng thông minh Ethereum. Nó giúp các nhà phát triển xây dựng, kiểm tra, triển khai và gỡ lỗi các hợp đồng thông minh.

Một số điểm tính năng nổi bật của Hardhat:

• Tích hợp mạng local hardhat, dễ dàng chạy và debug code ngay trên local.

• Debug dễ dàng hơn: Với Hardhat, chúng ta có thể debug code Solidity dễ dàng hơn khi có thể console.log ra các biến (Solidity vốn ko hỗ trợ console.log)

• Hệ thống plugin: Giúp developer có thể bổ sung chức năng, tùy vào từng dự án cụ thể

• Hỗ trợ TypeScript trên mạng Ethereum.

Hình 2.9 Hình ảnh minh họa Hardhat

Deploy smart contract giúp gì cho mạng blockchain?

Tạo ra các ứng dụng phi tập trung (dApps): Smart contract là nền tảng để xây dựng các ứng dụng phi tập trung.

Tự động hóa các quy trình: Smart contract có thể tự động hóa nhiều quy trình, ví dụ như chuyển tiền, quản lý tài sản, thực thi hợp đồng...

Tạo ra các tài sản kỹ thuật số: Smart contract có thể tạo ra các loại tài sản kỹ thuật số mới, ví dụ như token, NFT.

2.7 Ngôn ngữ Solidity

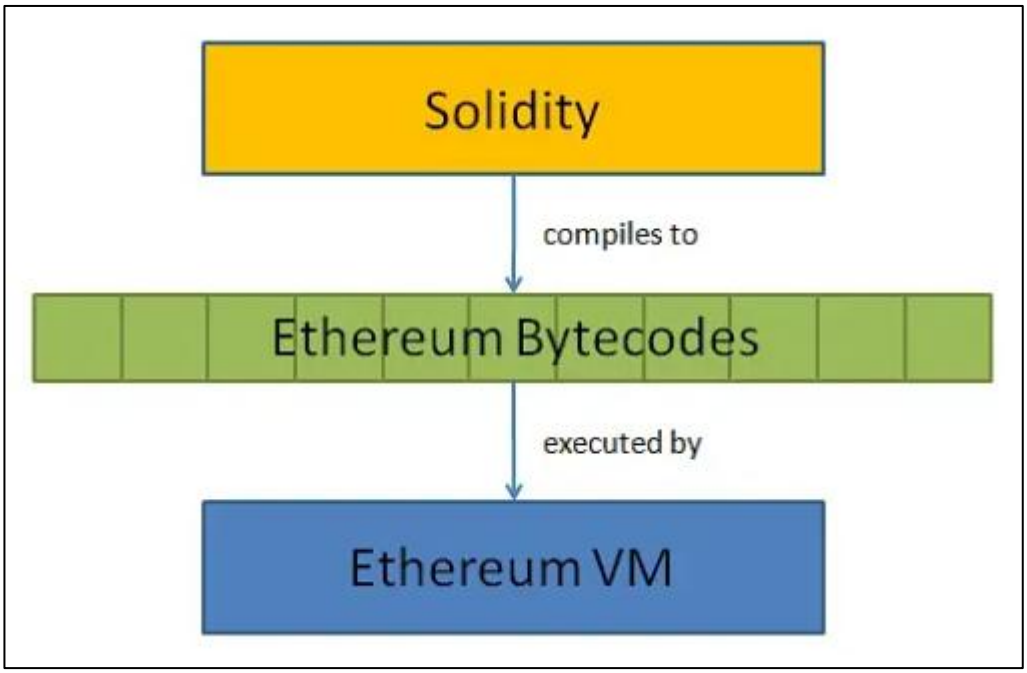
Hình 2.10 Ngôn ngữ Solidity

Solidity là một ngôn ngữ lập trình hướng đối tượng, high-level, curly-bracket được phát triển bởi team Ethereum Network.

Ngôn ngữ này ra đời nhằm để xây dựng và thiết kế các Smart Contracts (hợp đồng thông minh) trên các nền tảng của blockchain.

Nó có rất nhiều điểm tương đồng với C và C ++. Solidity khá đơn giản để học và dễ hiểu. Ví dụ: main trong C tương đương với contract trong Solidity.

Giống như các ngôn ngữ lập trình khác, Solidity cũng có biến, hàm, classes, Toán tử số học, thao tác chuỗi và nhiều khái niệm khác.



Hình 2.11 Cách hoạt động của Solidity

Các đoạn code solidity sẽ được compile sang Ethereum Bytecodes và được EVM thực thi thành các ứng dụng chạy trên Ethereum.

EVM là từ viết tắt của Ethereum Virtual Machine. Nó cung cấp một môi trường runtime cho Ethereum smart contracts.

Smart contracts cho phép bạn tiến hành các giao dịch đáng tin cậy mà không cần có sự tham gia của bên thứ ba. Các giao dịch này là có thể dễ dàng truy vết và không thể đảo ngược được.

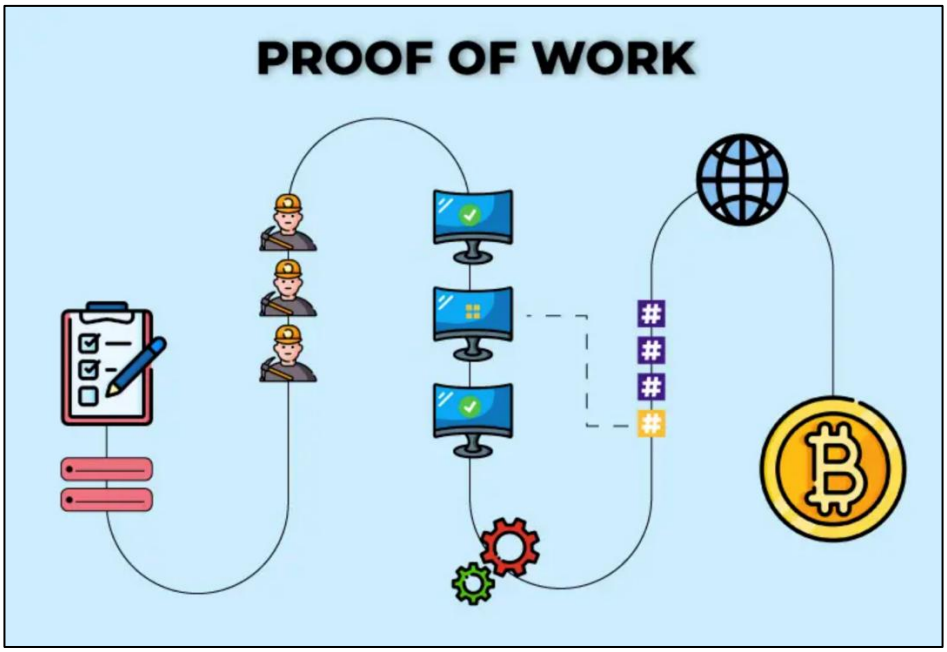
Ngôn ngữ lập trình thường được sử dụng để tạo và viết các hợp đồng thông minh là Serpent, Solidity, Mutan và LLL.

Một số nền tảng blockchain có hỗ trợ Solidity:

- Ethereum
- Binance Smart Chain
- Ethereum Classic
- Tron
- Hedera Hashgraph
- Avalanche

2.8 Các cơ chế đồng thuận trong Blockchain Network

2.8.1 POW (Proof of work)



Hình 2.12 Proof of work

Proof of Work hay bằng chứng công việc là thuật toán đồng thuận thường thấy ở các blockchain, được sử dụng trong việc xác nhận các giao dịch và tạo ra các khối mới trên blockchain (chuỗi khối) đó.

Cụ thể thuật toán PoW yêu cầu những người được phép thêm dữ liệu hay xác nhận giao dịch trên một blockchain phải thực hiện một khối lượng công việc. Khối lượng công việc đó có thể là một bài toán đố. Từ đó các giao dịch trên blockchain trở nên đáng tin cậy hơn và có thể được diễn ra một cách ngang hàng với nhau (Peer to peer) mà không cần phải qua một bên thứ ba nào như Paypal hay Momo,...

2.8.2 PoS (Proof of Stake)

Hình 2.13 Proof of Stake

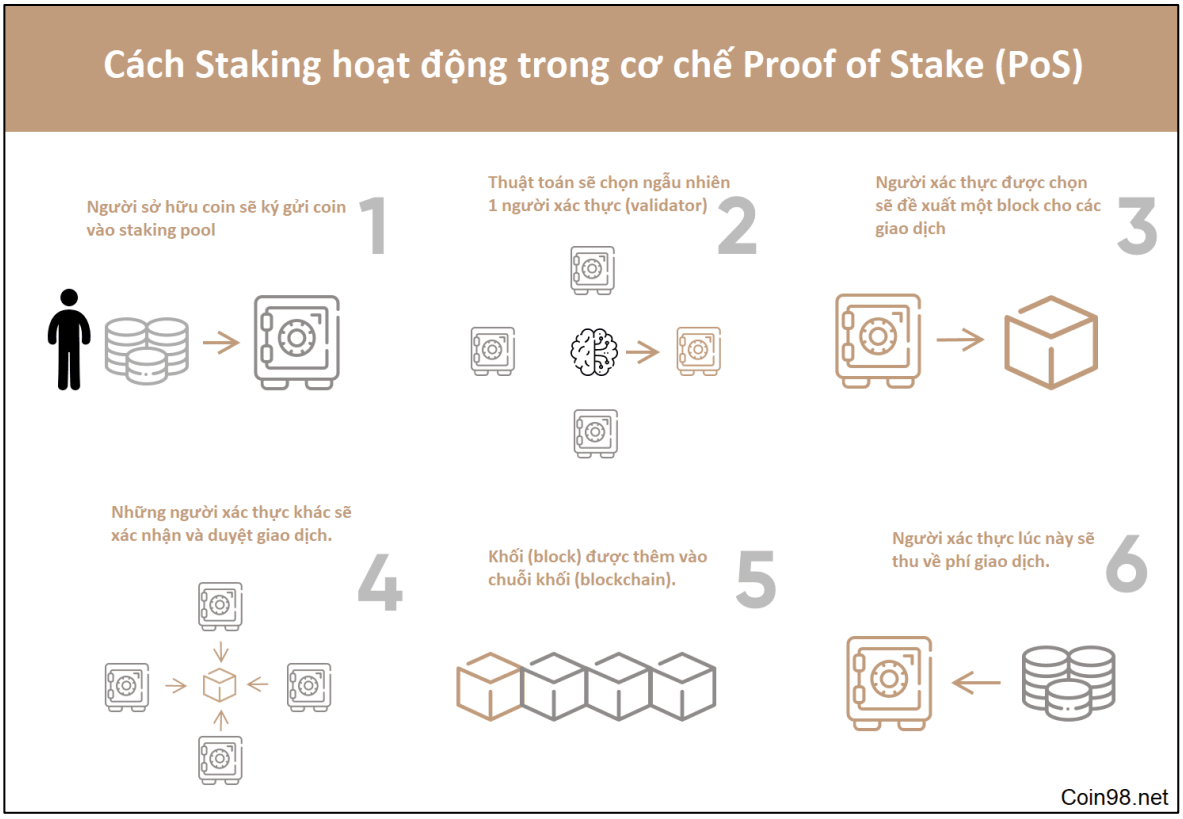
Proof of Stake (bằng chứng cổ phần) là một thuật toán làm việc của Blockchain. PoS cho phép người dùng kiếm được phần thưởng cho việc xác thực các khối trên blockchain.

Có thể hiểu nôm na là người dùng sẽ ký gửi một lượng tài sản nhất định để trở thành Validator (người xác thực) của Blockchain.

Không giống như Proof of Work (được sử dụng bởi Bitcoin), người dùng không cần phần cứng mining đắt tiền hoặc lượng điện lớn. Thay vào đó, mạng lưới lựa chọn các cá nhân để validate các block dựa trên lượng coin mà họ sở hữu. Lượng coin sở hữu càng cao, người dùng càng có nhiều khả năng được chọn để validate.

Các Validator này sẽ xác minh các giao dịch trên mạng lưới, gửi bằng chứng vào khối. Nếu đúng, các Validator sẽ được nhận thưởng là lạm phát của Blockchain,

hoặc phí giao dịch thu về. Nếu sai, họ sẽ chịu phạt là mất đi tất cả, hoặc một lượng tài sản đã ký gửi.



Hình 2.14 Cách Proof of Stake hoạt động

Proof of Stake (PoS) hoạt động theo các bước dễ hiểu như sau:

- **Stake token:** Người dùng cần sở hữu đủ tối thiểu số lượng token mà blockchain đó yêu cầu và stake token vào mạng lưới để trở thành một node trong quá trình đồng thuận. Hành động này đảm bảo tính trung thực và tạo sự cổ phần hóa cho các node trong hệ thống.

- **Chọn node:** Một số node được chọn ngẫu nhiên để tham gia vào quá trình đồng thuận. Các node được chọn phải đáp ứng yêu cầu về việc sở hữu số lượng token và tuân thủ các quy định về tính toán và bảo mật.

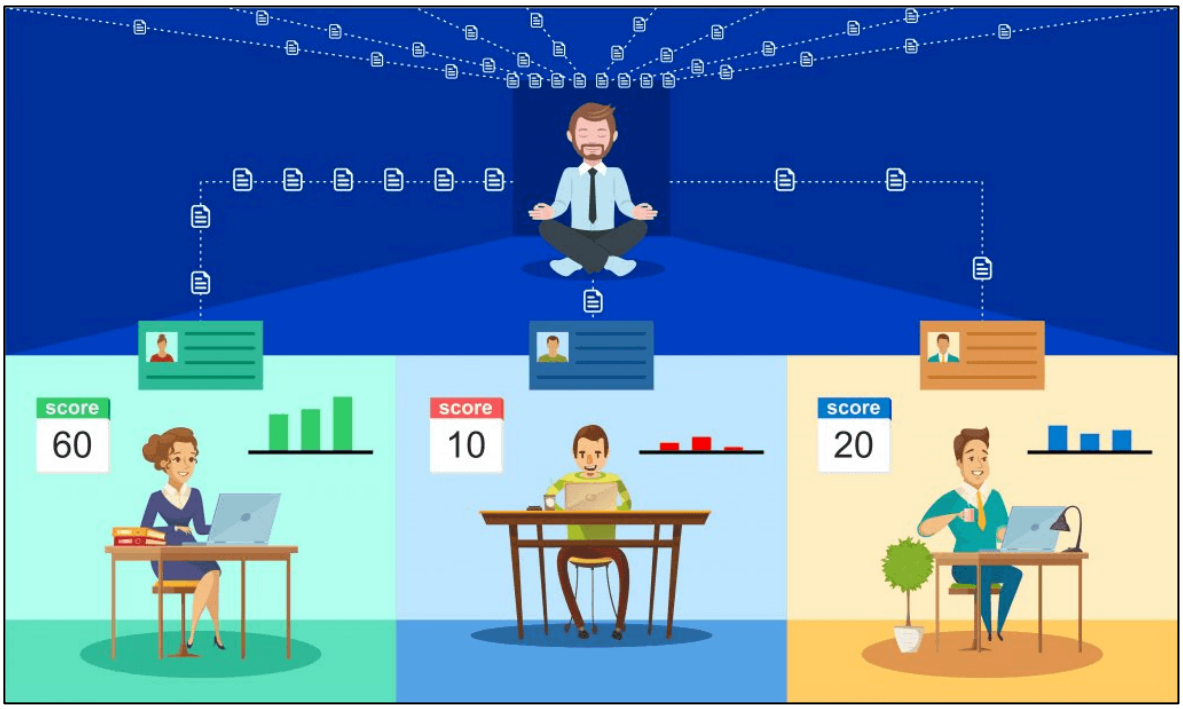
- **Xác minh giao dịch:** Các node xác minh giao dịch mới trên mạng. Khi giao dịch được xác minh chính xác, các node thêm giao dịch này vào block mới được tạo ra.

- **Tạo block mới:** Sau khi các node đã đồng thuận về một giao dịch, họ cùng nhau tạo block mới trên mạng. Token mà họ đã stake được sử dụng để tính tỷ lệ chia sẻ phần thưởng cho quá trình đồng thuận và tạo block mới này.

- Cập nhật blockchain: Block mới được tạo ra sẽ được cập nhật vào blockchain và thông báo đến tất cả các node trong hệ thống.

- Phần thưởng: Các node sẽ nhận được phần thưởng tương ứng với số lượng token mà họ đã stake để thực hiện quá trình đồng thuận và tạo block mới.

2.8.3 PoA (Proof of Authority)



Hình 2.15 Proof of Authority

PoA là viết tắt của Proof of Authority, tức Bằng chứng ủy quyền, là một thuật toán đồng thuận dựa trên danh tiếng, mang lại một giải pháp thực tế và hiệu quả cho các blockchain. Thuật ngữ này do nhà đồng sáng lập và cựu CTO của Ethereum, Gavin Wood, đề xuất vào năm 2017.

Proof of Authority là một biến thể của cơ chế đồng thuận Proof of Stake, trong đó thuật toán đề cao giá trị của danh tính và danh tiếng của những người tham gia, chứ không dựa trên giá trị token mà họ nắm giữ.

Mô hình Proof of Authority dựa trên số lượng validator có giới hạn, và điều này khiến nó trở thành một mô hình có khả năng mở rộng dễ dàng. Các khối và giao dịch được xác thực bởi những người tham gia đã được phê duyệt, họ đóng vai trò như là những người điều tiết của hệ thống.

Cơ chế hoạt động của Proof of Authority:

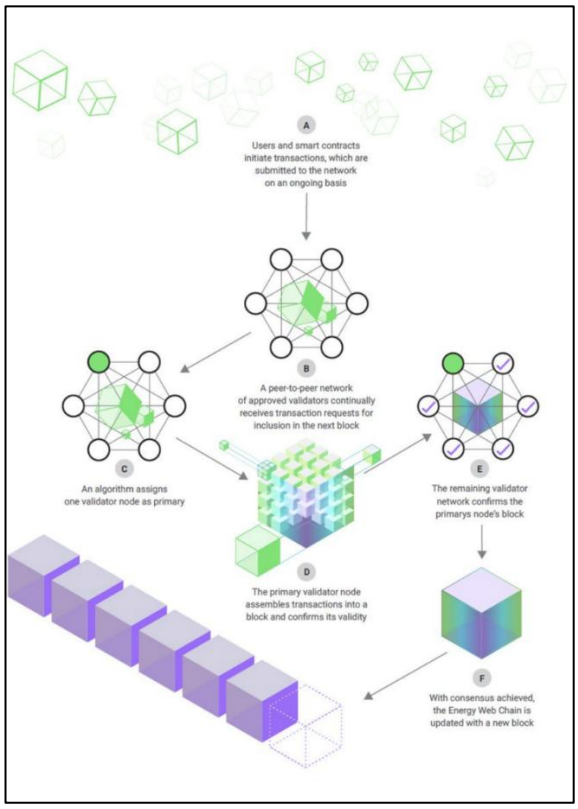
Proof of Authority có số lượng validator (trình xác thực) giới hạn, vì vậy thuật toán này giúp cho các blockchain có khả năng mở rộng cao hơn. Các block và giao

dịch được kiểm duyệt bởi các validator đáng tin cậy hơn vì PoA sở hữu các node có danh tích đã được xác thực.

Nhiệm vụ của các validator là khởi chạy ứng dụng để tiếp nhận yêu cầu giao dịch vào block. Nhưng vì mô hình PoA tự động hoàn toàn, nên các validator không cần phải liên tục theo dõi máy tính để cập nhật. Tuy nhiên, máy tính và trang web quản trị luôn phải được duy trì trong trạng thái hoạt động.

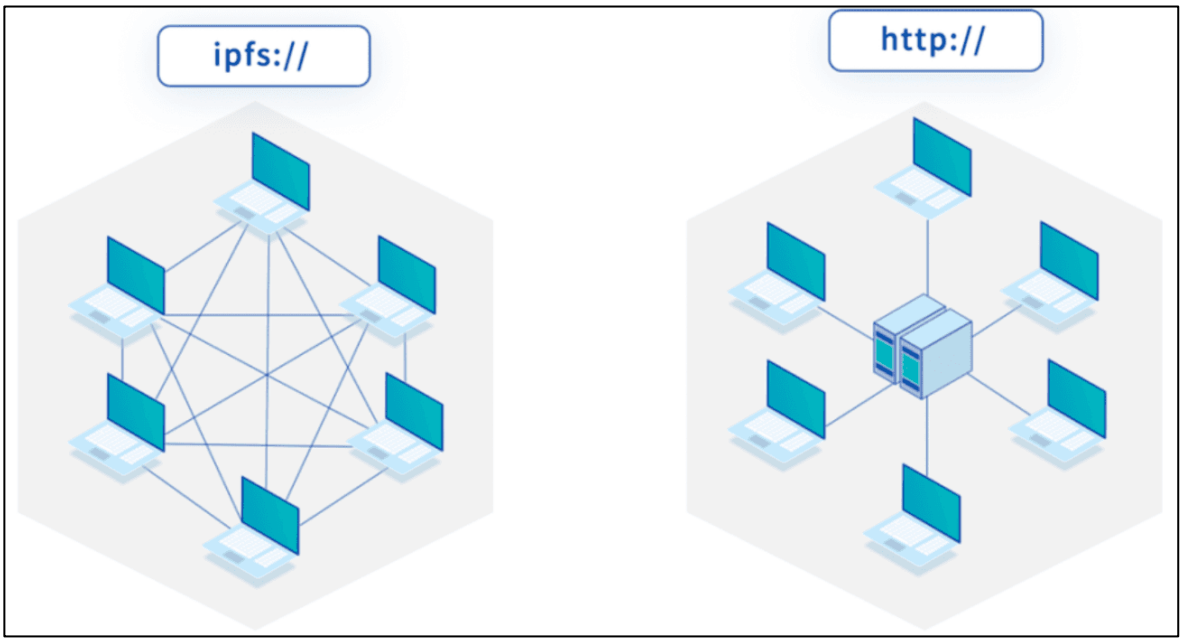
Những điều kiện và quyền hạn cho mỗi validator là giống nhau. Có nghĩa là họ có cơ hội tạo block mới và nhận số phần thưởng tương tự nhau. Chính vì vậy, PoA sẽ sử dụng ít năng lượng hơn các thuật toán đồng thuận khác, ví dụ như PoW. Cách thức hoạt động của PoA như sau:

- Đầu tiên, hệ thống sẽ chọn ngẫu nhiên một validator để xác thực giao dịch và tạo khối mới cho nền tảng blockchain. Validator này sẽ phụ thuộc vào hệ thống bỏ phiếu của validator được ủy quyền trước đó.
- Sau đó các validator sẽ xác thực các giao dịch diễn ra trong blockchain, sau khi xác thực thành công họ sẽ nhận phần thưởng được trích từ phí giao dịch.
- Mặt khác, nếu validator không thể đảm bảo các giao dịch trong hệ thống được diễn ra suôn sẻ hoặc gây hại cho mạng lưới thì danh tiếng của họ sẽ bị đánh giá thấp. Đồng thời, hệ thống sẽ loại bỏ vĩnh viễn quyền xác thực của họ.



Hình 2.16 Cách hoạt động của Proof of Authority (PoA)

2.9 IPFS (Interplanetary File System)



Hình 2.17 Hình ảnh minh họa IPFS

IPFS (viết tắt của từ Interplanetary File System) là một hệ thống tập tin phân tán ngang hàng kết nối tất cả các thiết bị máy tính với nhau. Cụ thể hơn, nó sẽ phân phối dữ liệu được lưu trữ theo hình thức P2P, hay còn gọi là mạng ngang hàng (mạng đồng đẳng).

Trong đó, các hoạt động của IPFS chủ yếu dựa vào khả năng tính toán băng thông của tất cả các máy tham gia chứ không tập trung vào một phần nhỏ các máy chủ trung tâm như giao thức HTTP.

Nói cách khác, IPFS là mạng lưới chuyển phát nội dung hoàn toàn phi tập trung cho phép quản lý và lưu trữ dữ liệu một cách linh hoạt. Mỗi máy tính tham gia trong mạng lưới đảm nhận nhiệm vụ download và upload dữ liệu mà không cần sự can thiệp của máy chủ trung tâm.

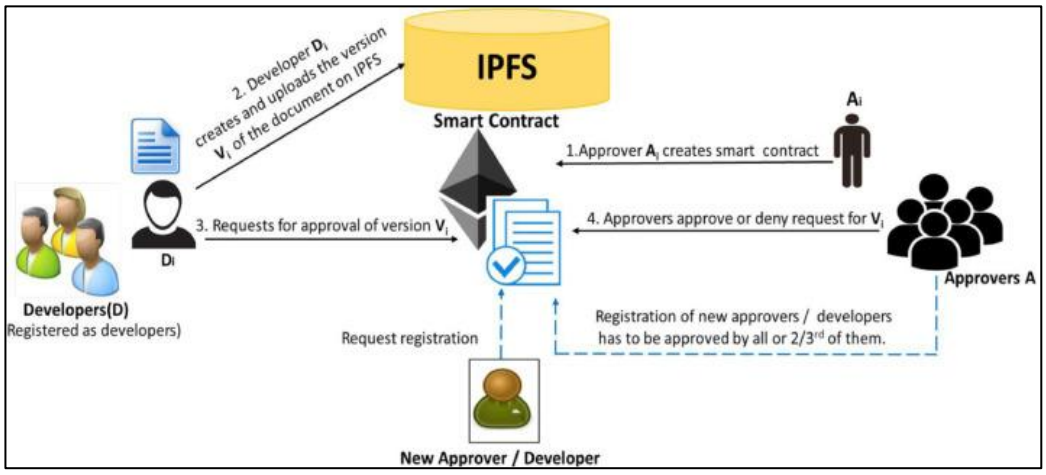
Cách thức hoạt động của IPFS:

Đầu tiên mọi dữ liệu sẽ được mã hoá và được lưu dưới dạng mã hash (còn gọi là đối tượng IPFS). Ý tưởng chủ đạo là nếu trình duyệt của bạn muốn truy cập một trang nào đó trên IPFS thì chỉ cần đưa ra mã hash rồi mạng sẽ tìm máy có lưu trữ dữ liệu khớp với mã hash và sau đó tải dữ liệu, trang đó về từ máy tính đấy về cho bạn.

Hình 2.18 Các thức hoạt động của IPFS

Cách thức hoạt động của IPFS sẽ tương tự như BitTorrent, đồng nghĩa với mỗi máy tính tham gia trong mạng lưới của nó sẽ đảm nhận cả việc download lẫn upload dữ liệu mà không cần có sự có mặt của một máy chủ trung tâm. Tổng quan, cách hoạt động của IPFS sẽ có 2 phần chính:

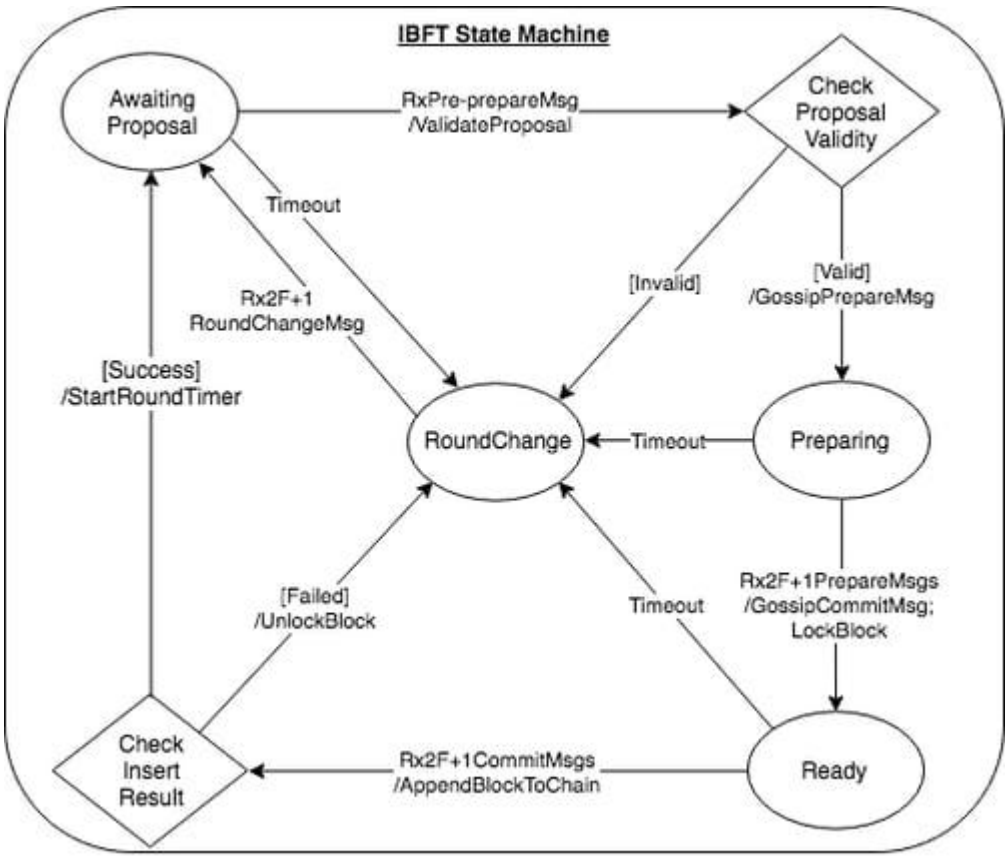
- Xác định tệp có địa chỉ nội dung (giá trị hash của tệp đó).
- Tìm dữ liệu được lưu trữ và tải xuống: khi bạn có đoạn hash của file hay trang cần tải, mạng sẽ tìm và connect tới máy tốt nhất để tải dữ liệu xuống cho bạn.



Hình 2.19 Quy trình quản lý phiên bản tài liệu bằng blockchain và IPFS

2.9.1 IBFT (Istanbul Byzantine Fault Tolerance)

Istanbul Byzantine Fault Tolerance (IBFT) là một phương pháp đồng thuận được phát triển để giúp các blockchain đối phó với tình trạng lỗi Byzantine (khi một hoặc nhiều nút trong mạng có thể hành động sai lệch hoặc gian lận). IBFT được thiết kế đặc biệt cho các mạng blockchain có sự quản lý, nơi các nút tham gia được xác định trước và có thể tin cậy vào một số lượng nhất định các validators (người xác nhận).



Hình 2.20 Quy trình hoạt động của IBFT

Các blockchain sử dụng IBFT

Một số blockchain nổi tiếng sử dụng IBFT hoặc các biến thể của nó bao gồm:

- **Ethereum Classic:** Một trong những blockchain đầu tiên sử dụng IBFT.
- **Binance Smart Chain:** Một blockchain tương thích với Ethereum, được biết đến với tốc độ giao dịch nhanh và phí giao dịch thấp.
- **Một số blockchain riêng:** Nhiều doanh nghiệp và tổ chức đã xây dựng các blockchain riêng sử dụng IBFT để phục vụ các mục đích cụ thể.

IBFT là một thuật toán đồng thuận quan trọng trong lĩnh vực blockchain, giúp các mạng lưới đạt được sự cân bằng giữa hiệu suất, khả năng mở rộng và bảo mật. Nếu bạn muốn tìm hiểu sâu hơn về IBFT và các thuật toán đồng thuận khác, bạn có thể tham khảo thêm các tài liệu chuyên sâu về blockchain.

2.9 Tìm hiểu Hyperledger Besu

Hyperledger là một dự án blockchain doanh nghiệp toàn cầu cung cấp các khung làm việc, tiêu chuẩn, hướng dẫn và công cụ cần thiết để xây dựng các blockchain mã nguồn mở và các ứng dụng liên quan, phục vụ cho nhiều ngành công nghiệp khác nhau. Các dự án của Hyperledger bao gồm nhiều nền tảng blockchain cấp doanh nghiệp được cấp quyền (permissioned blockchain) và các giải pháp plug-and-play.

Sử dụng các thành phần có sẵn trong hệ sinh thái Hyperledger, doanh nghiệp có thể áp dụng các giải pháp và dịch vụ blockchain mô-đun để cải thiện đáng kể hiệu suất hoạt động và tối ưu hóa quy trình kinh doanh của mình.

Hình 2.21 Ảnh minh họa Hyperledger Besu

Điểm chính cần lưu ý:

- Hyperledger là một cộng đồng mã nguồn mở tập trung vào phát triển bộ khung làm việc, công cụ và thư viện ổn định cho các triển khai blockchain được cấp quyền và đạt chuẩn doanh nghiệp.
- Đây là một sự hợp tác toàn cầu, bao gồm các tổ chức thành viên là những đơn vị hàng đầu trong các lĩnh vực tài chính, ngân hàng, Internet vạn vật (IoT), chuỗi cung ứng, sản xuất và công nghệ.
- Có nhiều dự án con nổi bật, bao gồm Hyperledger Fabric, Cello, Besu, và Caliper.

2.9.1 Cách Hyperledger hoạt động

Hãy hình dung Hyperledger như một hệ điều hành chạy trên máy tính, laptop, thiết bị khác hoặc một mạng lưới, tương tự như hệ điều hành mã nguồn mở Linux.

Linux là nền tảng lập trình cơ bản được sử dụng để tạo ra nhiều hệ điều hành khác nhau. Trên thực tế, Linux được sử dụng phổ biến hơn các hệ điều hành độc quyền khác trong các ứng dụng doanh nghiệp. Các hệ điều hành này có thể được thiết kế khác biệt, giao diện người dùng tùy chỉnh và hoàn toàn có thể cá nhân hóa. Sau khi cài đặt hệ điều hành Linux (hoặc thiết bị của bạn đã có sẵn), bạn có thể chọn chương trình mình muốn sử dụng, ví dụ như thay đổi trình duyệt web hoặc bộ ứng dụng văn phòng mặc định.

Hyperledger cũng tương tự như vậy, đó là một hệ thống cho phép người dùng quyết định họ cần gì dựa trên nhu cầu của mình. Một hệ thống có thể được xây dựng bằng cách sử dụng một trong các dự án Hyperledger hiện có (coi các dự án này như những hệ điều hành được thiết kế cho các mục đích cụ thể) với các module khác nhau để đáp ứng từng nhu cầu cụ thể.

Các dự án nổi bật trong hệ sinh thái Hyperledger:

Hyperledger Fabric:

Là nền tảng phát triển các sản phẩm, giải pháp và ứng dụng dựa trên công nghệ blockchain và sổ cái phân tán (distributed ledger) dành cho doanh nghiệp.

Hyperledger Cello:

Cung cấp mô hình triển khai Blockchain-as-a-Service (BaaS), cho phép doanh nghiệp sử dụng blockchain theo yêu cầu như một dịch vụ.

Hyperledger Besu:

Là một client Ethereum, hỗ trợ các nhà phát triển xây dựng ứng dụng sử dụng Hyperledger và kết nối với mạng blockchain Ethereum.

Hyperledger Caliper:

Là công cụ đo lường hiệu suất blockchain (blockchain benchmarking tool), giúp đánh giá khả năng hoạt động của một nền tảng blockchain cụ thể.

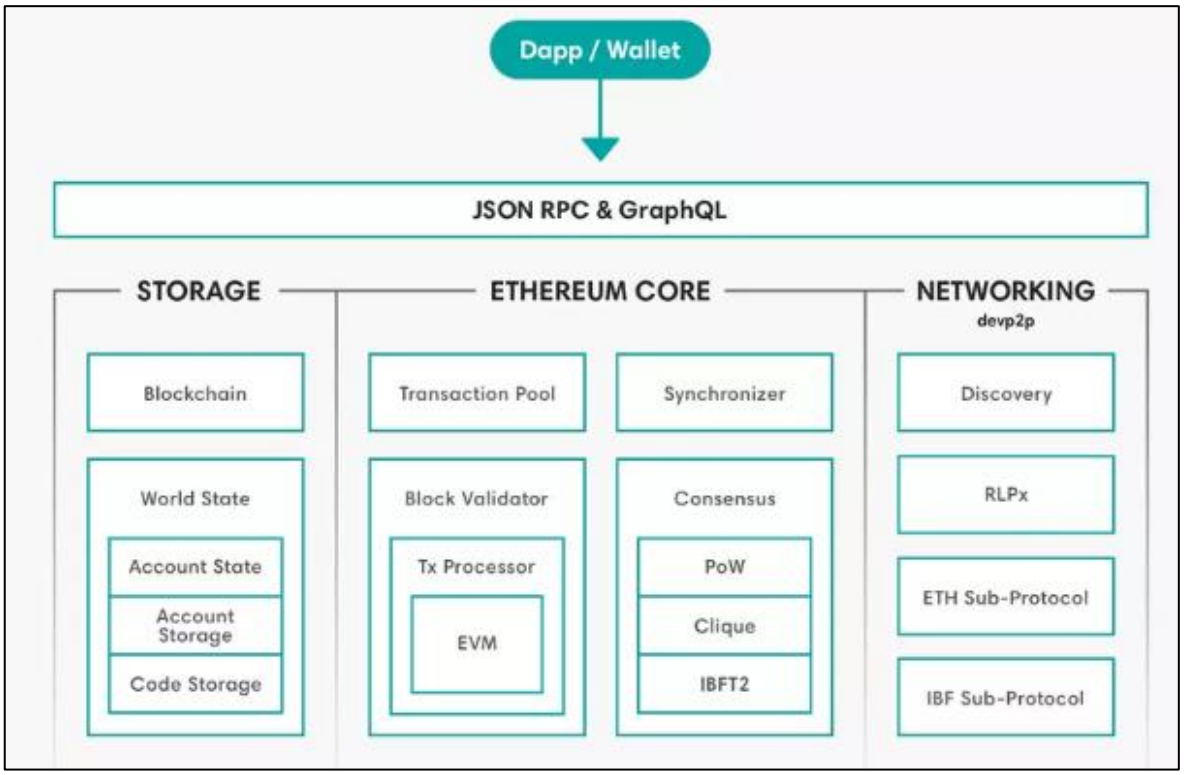
2.9.2 Phương pháp thiết kế của Hyperledger

Tất cả các dự án trong hệ sinh thái Hyperledger đều tuân thủ các nguyên tắc thiết kế sau:

- Kiến trúc mô-đun và khả năng mở rộng linh hoạt.
- Khả năng tương tác giữa các nền tảng và hệ thống.
- Tích hợp các tính năng bảo mật cấp doanh nghiệp.

Mặc dù Hyperledger không tập trung vào việc phát triển token hoặc tiền mã hóa, các nhà phát triển vẫn có thể tạo ra chúng nếu cần thiết.

2.9.3 Kiến trúc



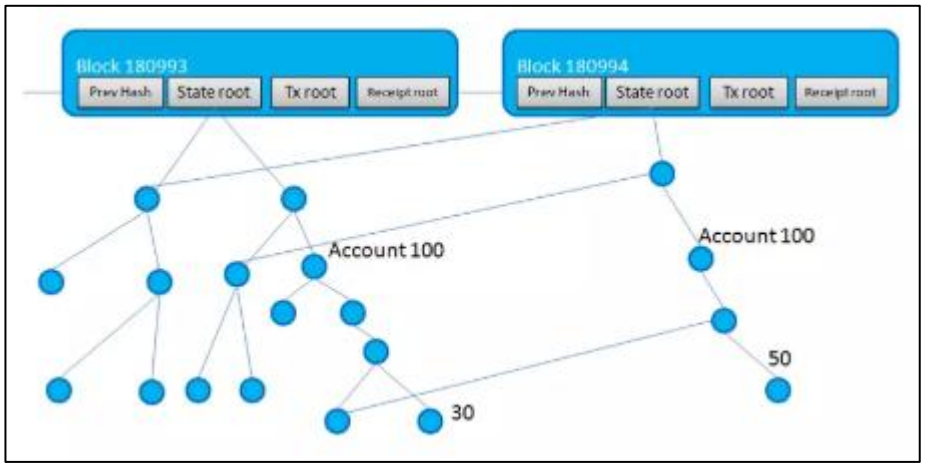
Hình 2.22 Kiến trúc của Besu

Kiến trúc của Besu chia ra làm 4 phần chính là **Storage, Ethereum Core , Networking** và **User-facing APIs**

Storage: Besu sử dụng RocksDB để lưu trữ dữ liệu dạng key-value. Data lưu trong storage chia thành 2 phần.

Blockchain: Lưu chuỗi các block (gồm header và body block).

World State: Lưu StateTree (mapping address account với state của account (vd như số dư)).



Hình 2.1 Ảnh minh họa

Ethereum Core:

EVM: Biên dịch, xử lý các smart contracts cũng như các giao dịch.

Thuật toán đồng thuận (Consensus Algorithms): Hỗ trợ 3 thuật toán là **Proof of Work (Ethash)**, **Clique** và **IBFT 2.0**.

Networking

Discovery: Giao thức dựa trên UDP để tìm các peers trên network

RLPx: Giao thức dựa trên TCP tương tác với các peers khác. Tùy vào thuật toán đồng thuận mà node sẽ sử dụng **ETH Sub-protocol** hay là **IBF Sub-protocol**.

User-facing APIs

Phía ứng dụng có thể thông qua **HTTP JSON-RPC**, **WebSocket JSON-RPC** hay **GrapgQL** để tương tác với mạng blockchain. Khi phát triển dapp với Besu chúng ta có thể dùng các công cụ quen thuộc như **Truffle, Remix** hay **web3**.

Chương 3 NỘI DUNG NGHIÊN CỨU

3.1 Đặc tả hệ thống

Hệ thống chống giả mạo tài liệu văn bản giúp quản lý và xác thực tài liệu, văn bằng hoặc chứng chỉ một cách bảo mật, minh bạch và hiệu quả. Nó kết hợp nhiều công nghệ hiện đại như blockchain, IPFS, Express.js và MongoDB để tạo ra một quy trình vận hành mượt mà và an toàn.

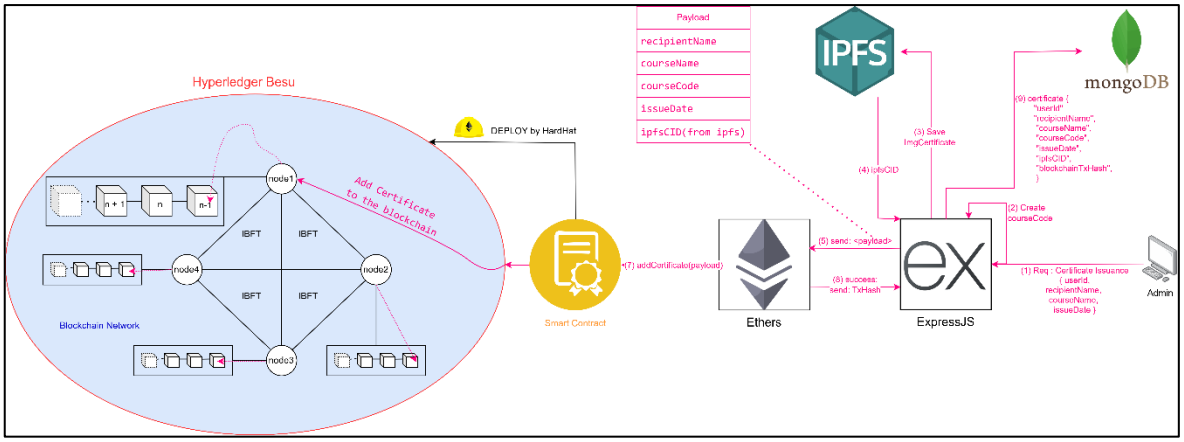
Khi một người dùng muốn lưu trữ tài liệu, họ sẽ tải tệp đó lên qua giao diện web. Hệ thống sẽ xử lý tài liệu này bằng cách gửi nó tới IPFS – một nền tảng lưu trữ phi tập trung. IPFS sẽ tạo ra một mã định danh duy nhất, gọi là CID, cho tệp đó. CID này giống như một "chữ ký" đại diện cho nội dung của tài liệu. Từ đó, CID sẽ được lưu trên blockchain thông qua một Smart contract (hợp đồng thông minh), đảm bảo rằng không ai có thể thay đổi hoặc giả mạo tài liệu. Đồng thời, thông tin như tên tài liệu, người tạo, hoặc ngày đăng ký sẽ được lưu vào cơ sở dữ liệu MongoDB để dễ dàng quản lý.

Khi người dùng hoặc bên thứ ba cần xác minh một tài liệu, họ chỉ cần tải tệp đó lên hệ thống. Hệ thống sẽ kiểm tra mã hash của tệp và so sánh với CID được lưu trên blockchain. Nếu mã hash này trùng khớp, nghĩa là tài liệu hợp lệ và chưa bị chỉnh sửa. Ngược lại, nếu không trùng, hệ thống sẽ thông báo rằng tài liệu có vấn đề.

Phần backend của hệ thống được xây dựng trên Express.js, đảm nhiệm vai trò giao tiếp giữa frontend (giao diện người dùng), blockchain, IPFS và MongoDB. Blockchain ở đây đóng vai trò lưu trữ CID, còn IPFS là nơi lưu trữ tài liệu thực tế. MongoDB sẽ giúp lưu thêm các thông tin liên quan để việc truy vấn dữ liệu trở nên nhanh chóng và dễ dàng hơn.

3.1.1 Chức năng cấp chứng chỉ của admin

Khi admin thực hiện chức năng cấp chứng chỉ thì khi đó Admin sẽ điền các thông tin của người dùng như: "UserId, Tên người nhận, Tên khóa học, Ngày cấp CC", sau đó gửi đến phía backend, sau đó sẽ tạo 1 Mã khóa học (đóng vai trò là trường duy nhất không trùng) và tạo hình ảnh chứng chỉ (file mềm), tiếp tục là lưu hình ảnh đó vào ipfs và được trả mã hóa và trả về ipfsCID, sau đó mới gửi toàn bộ thông tin lên Smart Contract và được Smart Contract thực hiện thành công và lưu vào blockchain thì sẽ tạo được 1 mã Transaction Hash (TxHash) mã này như là 1 mã sau khi thực hiện giao dịch thành công (nó đóng vai trò là duy nhất không bị sao chép hay làm giả được). Cuối cùng sẽ gửi mã TxHash kèm thông tin của chứng chỉ mới cấp lưu vào MongoDB.

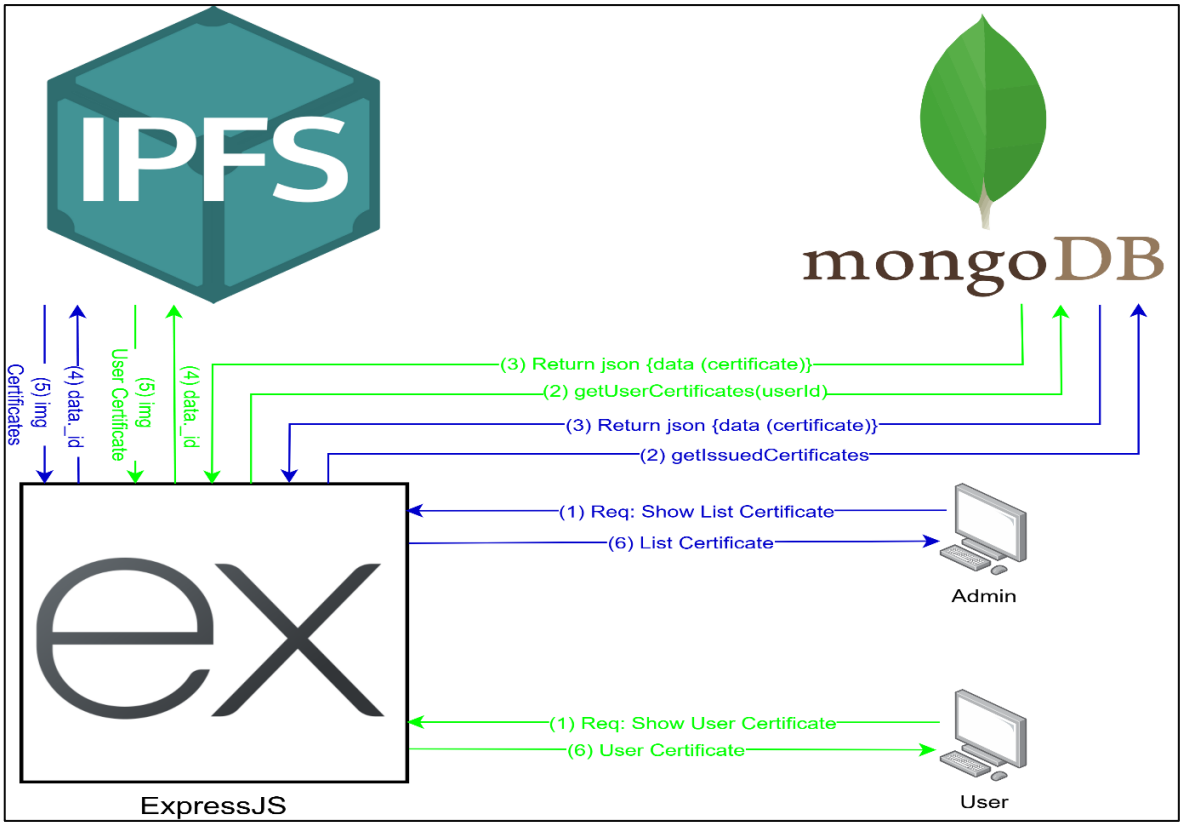


Hình 3.1 Chức năng cấp chứng chỉ của Admin

3.1.2 Chức năng xem danh sách chứng chỉ của Admin và xem chứng chỉ cá nhân của người dùng

Admin gửi yêu cầu xem danh sách chứng chỉ được cấp đến phía BackEnd thì khi đó sẽ hệ thống sẽ truy vấn vào cơ sở dữ liệu đến MongoDB lấy ra tất cả các dữ liệu của chứng chỉ, sau đó sẽ tiến hành lấy id của chứng chỉ đưa vào ipfs để lấy ra hình ảnh (file mềm) của chứng chỉ đó, cuối cùng sẽ trả về danh sách các chứng chỉ kèm với hình ảnh tương ứng của chứng chỉ đó

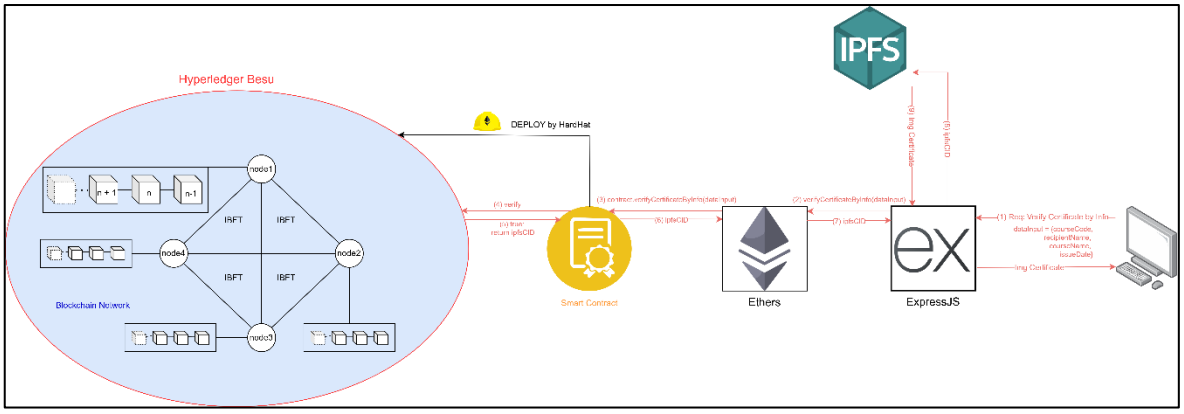
User gửi yêu cầu xem chứng chỉ cá nhân đến phía BackEnd, khi đó hệ thống sẽ truy vấn đến cơ sở dữ liệu đến MongoDB lấy ra chứng chỉ theo (userId) và cũng tiến hành lấy id của chứng chỉ đưa vào ipfs để lấy ra hình ảnh (file mềm) của chứng chỉ đó, cuối cùng là sẽ trả về các chứng chỉ tương ứng với người dùng đó.



Hình 3.2 Chức năng xem danh sách CC của Admin và xem chứng chỉ cá nhân của User

3.1.3 Chức năng xác thực chứng chỉ bằng thông tin

Khi người dùng thực hiện nhập các thông tin của chứng chỉ vào để xác minh thì hệ thống sẽ gửi đến SmartContract đưa vào blockchain để xác thực nếu trong blockchain có chứng chỉ hợp lệ thì sẽ tiến hành lấy ipfsCID từ blockchain ra và hệ thống sẽ tiến hành đem ipfsCID đưa vào ipfs để nhận được ảnh tương ứng của chứng chỉ đó, Cuối cùng sẽ trả về thông báo có chứng chỉ hợp lệ và ảnh tương ứng của chứng chỉ đó cho phía người dùng.



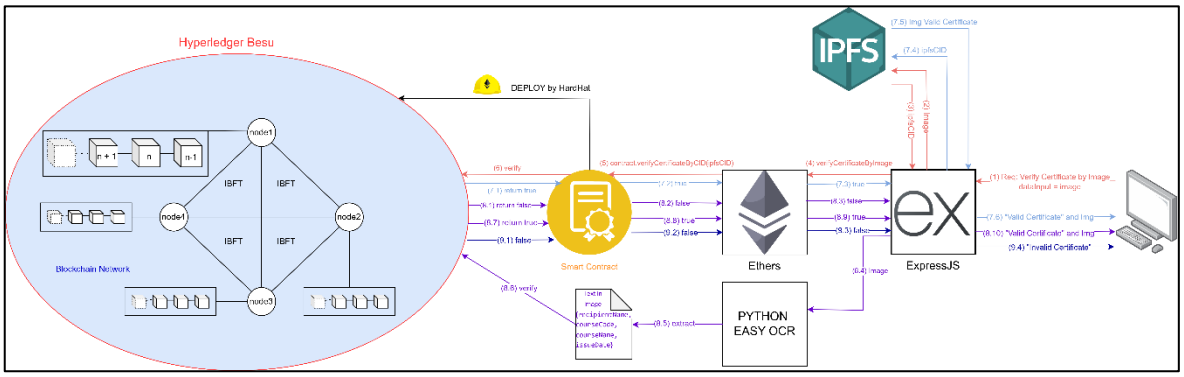
Hình 3.3 Chức năng xác thực bằng thông tin

3.1.4 Chức năng xác thực chứng chỉ bằng hình ảnh

Khi người dùng upload ảnh lên thì hệ thống sẽ tiến hành kiểm tra xem ảnh đó là ảnh từ file mềm hay là ảnh được người dùng chụp hình lại (như là người dùng sẽ photo ra và chụp lại để xác minh). Lúc này sẽ có 2 trường hợp sau:

Trường hợp 1: nếu ảnh đó là file mềm thì hệ thống sẽ tiến hành đưa ảnh đó vào ipfs hash ra và nhận về ipfsCID và sẽ đem ipfsCID này gửi cho SmartContract để xác thực với blockchain xem nó đã có chứng chỉ nào hợp lệ với ipfsCID này hay không, nếu trong blockchain xác thực có thì sẽ gửi về thông báo chứng chỉ hợp lệ, còn nếu ipfsCID không khớp trong blockchain thì sẽ tiếp đến trường hợp 2.

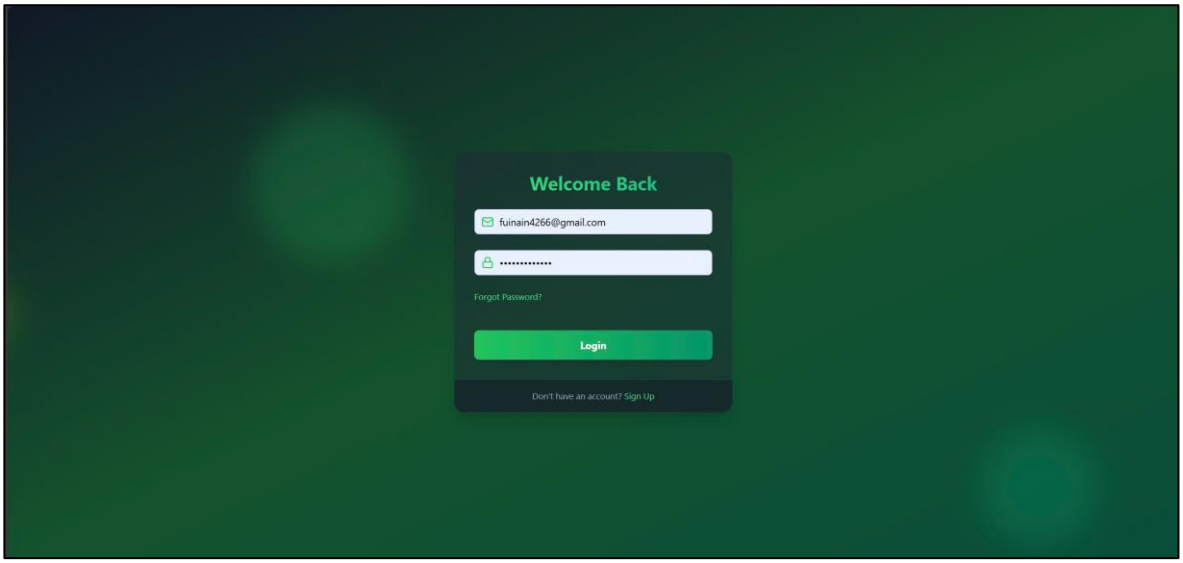
Trường hợp 2: nếu xác thực không có ipfsCID trong blockchain thì tiếp tục hệ thống sẽ chạy một script python lấy toàn bộ text trong hình sau đó tiếp tục lấy đoạn text đó duyệt qua từng khối lưu trữ trong blockchain nếu khớp 4/4 trường dữ liệu là (mã khóa học, tên người nhận, ngày cấp, tên khóa học) thì sẽ gửi thông báo về cho người dùng là chứng chỉ hợp lệ.



Hình 3.4 Chức năng xác thực bằng hình ảnh

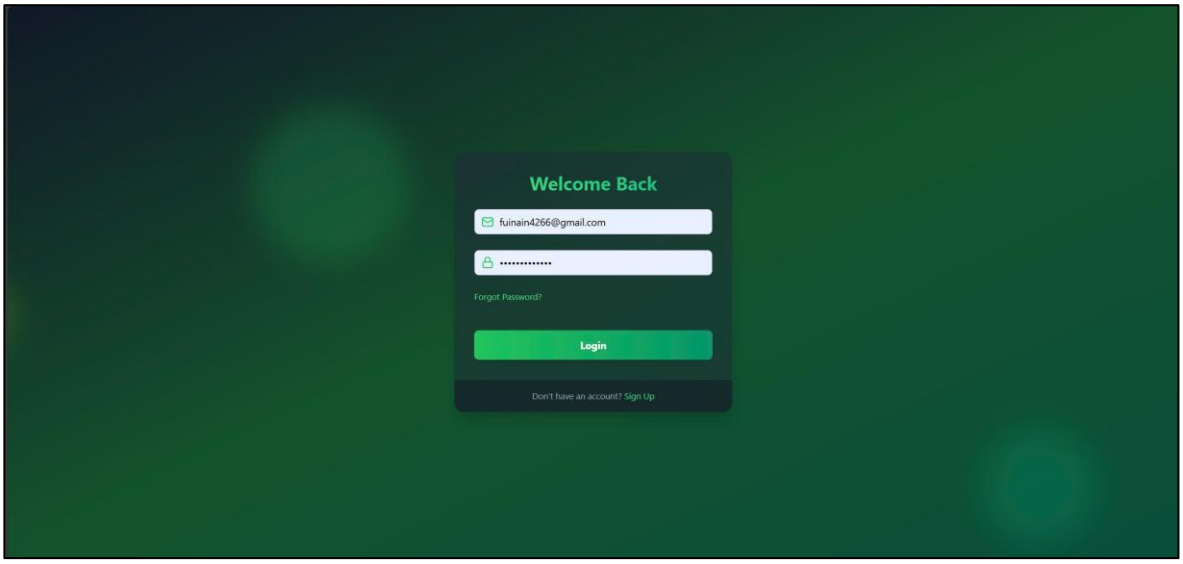
3.2 Giao diện hệ thống

3.2.1 Trang đăng nhập



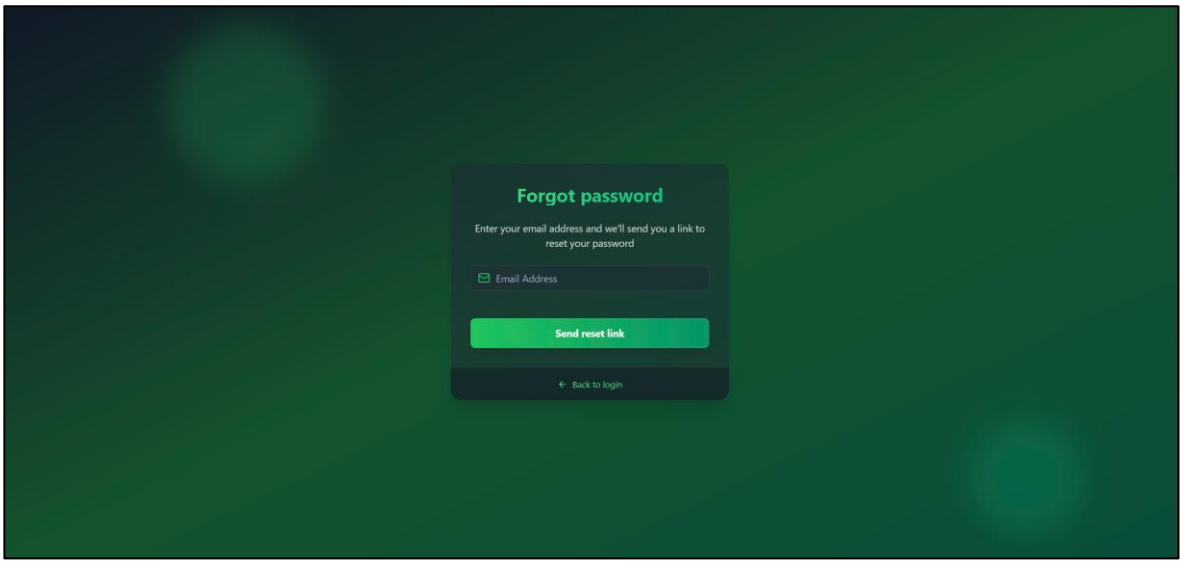
Hình 3.5 Trang đăng nhập

3.2.2 Trang đăng ký



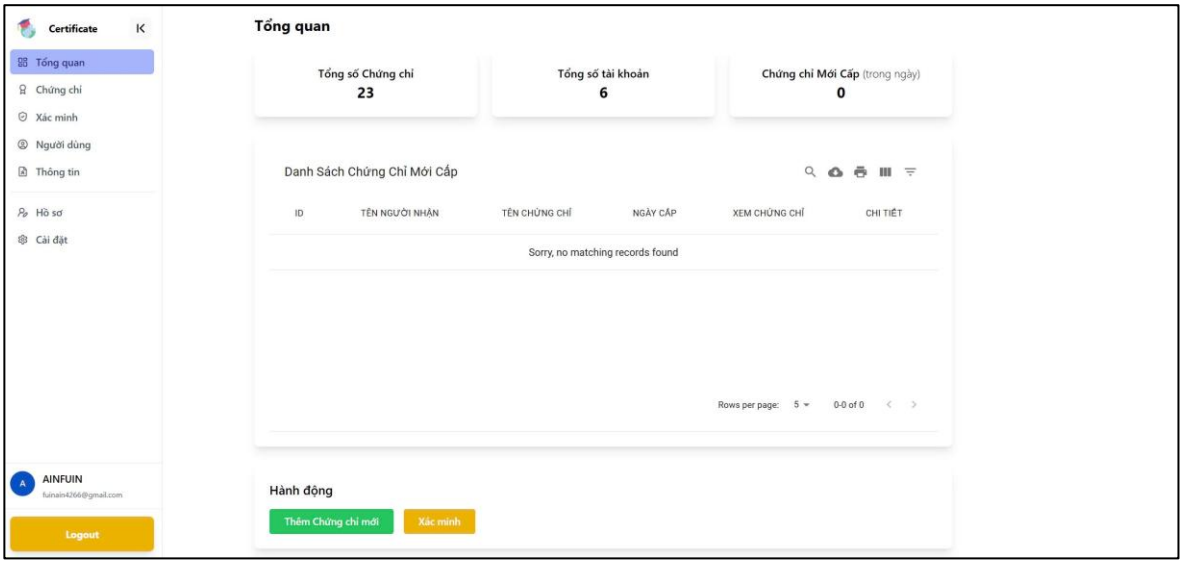
Hình 3.6 Trang đăng ký

3.2.3 Trang quên mật khẩu



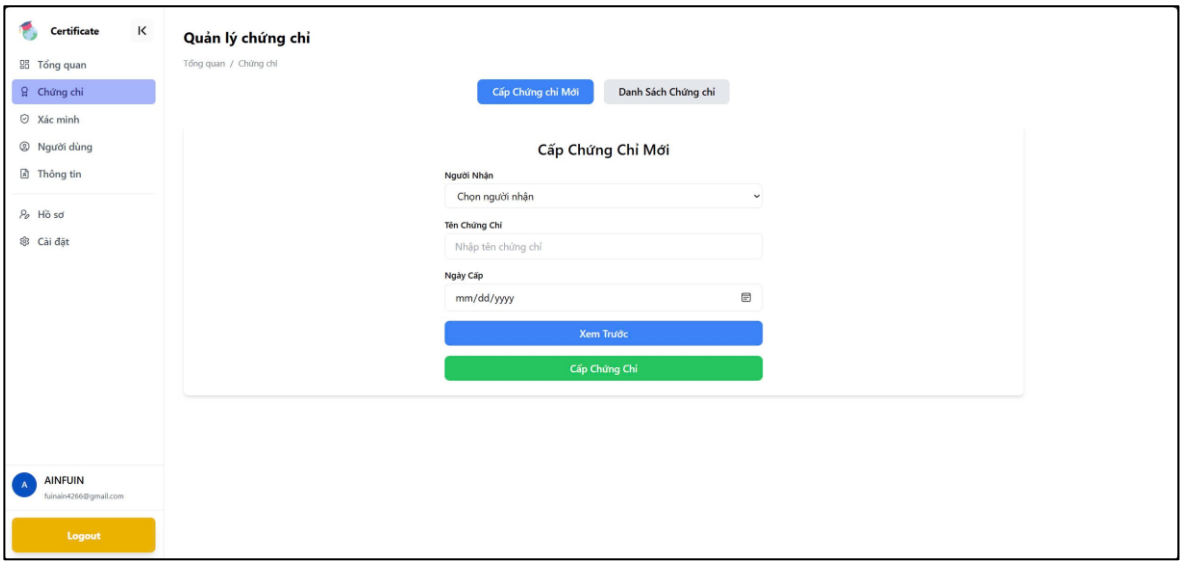
Hình 3.7 Trang quên mật khẩu

3.2.4 Trang tổng quan



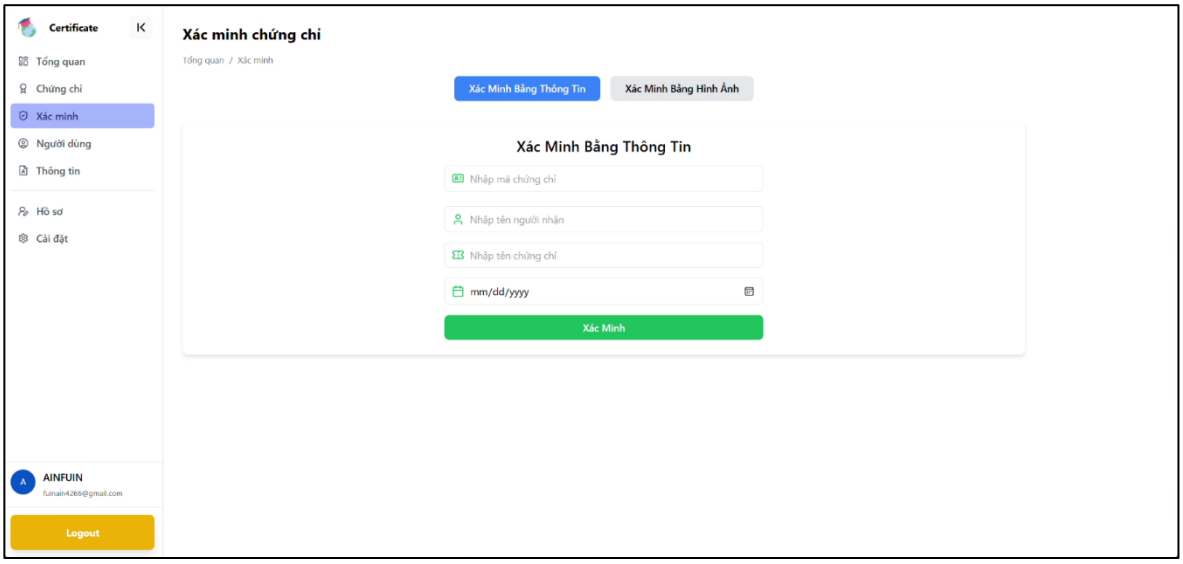
Hình 3.8 Trang tổng quan

3.2.5 Trang cấp chứng chỉ



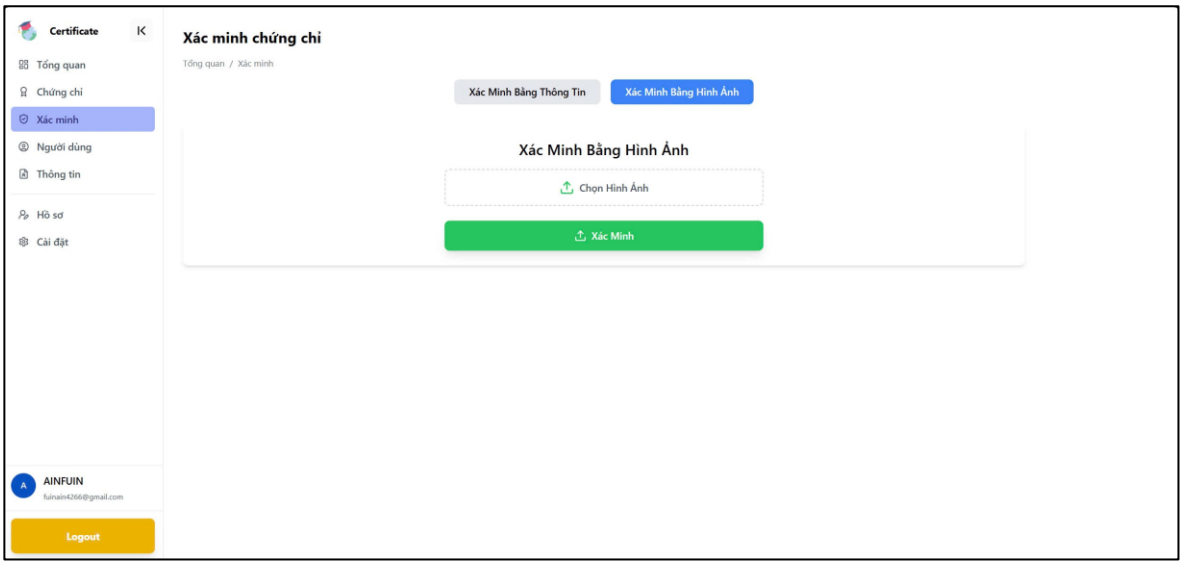
Hình 3.9 Trang cấp chứng chỉ

3.2.6 Giao diện xác minh chứng chỉ bằng thông tin



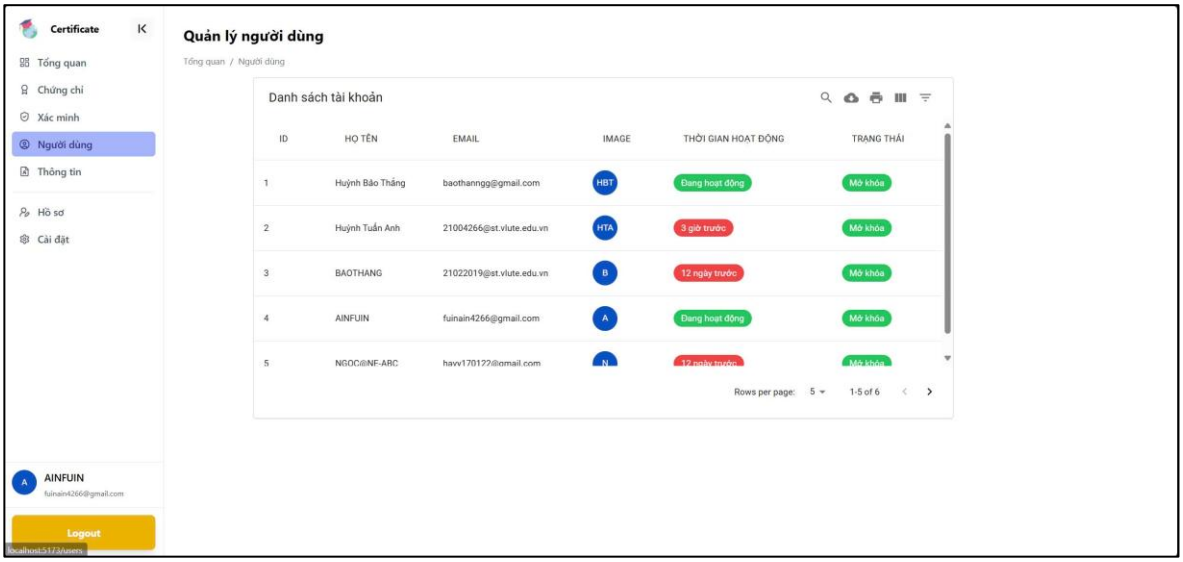
Hình 3.10 Giao diện xác minh chứng chỉ bằng thông tin

3.2.7 Trang xác minh chứng chỉ bằng hình ảnh



Hình 3.11 Trang xác minh chứng chỉ bằng hình ảnh

3.2.8 Trang quản lý người dùng



Hình 3.12 Trang quản lý người dùng

3.2.9 Trang xác thực 2 bước

Hình 3.13 Trang xác thực

Chương 4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết Luận

Hệ thống chống giả mạo tài liệu dựa trên blockchain mang lại một giải pháp hiệu quả, minh bạch và bảo mật cao trong việc quản lý, xác thực và bảo vệ tài liệu số. Bằng cách tận dụng tính phi tập trung, không thể sửa đổi và minh bạch của blockchain, hệ thống giúp ngăn chặn các hành vi giả mạo, đảm bảo rằng thông tin trên tài liệu luôn đáng tin cậy.

Hơn nữa, hệ thống này cung cấp khả năng kiểm tra nguồn gốc tài liệu một cách dễ dàng, cho phép các bên liên quan có thể xác thực thông tin một cách nhanh chóng mà không cần phụ thuộc vào bên thứ ba. Điều này không chỉ nâng cao độ tin cậy mà còn giảm chi phí vận hành, đồng thời cải thiện hiệu quả trong các ngành như giáo dục, tài chính, y tế và quản lý nhà nước.

4.2 Hướng Phát Triển

Ứng dụng blockchain để xây dựng hệ thống chống giả mạo tài liệu, văn bằng, chứng chỉ mở ra nhiều cơ hội cải tiến và mở rộng. Dưới đây là những hướng phát triển quan trọng:

Sử dụng công nghệ mã hóa nâng cao

- Bảo vệ thông tin nhạy cảm trong tài liệu, văn bằng bằng các phương pháp mã hóa hiện đại như Zero-Knowledge Proof (ZKP), giúp xác minh mà không tiết lộ dữ liệu chi tiết.
- Đảm bảo quyền riêng tư và tuân thủ các quy định như GDPR trong việc lưu trữ và chia sẻ thông tin.

Tăng cường bảo mật và phòng chống tấn công mạng

- Ứng dụng các cơ chế bảo mật mạnh mẽ hơn để chống lại các cuộc tấn công như giả mạo danh tính, chiếm quyền điều khiển hệ thống.
- Thường xuyên cập nhật và tối ưu hóa mạng lưới blockchain để tăng khả năng chống chịu trước các mối đe dọa.

Hỗ trợ tương tác giữa các blockchain khác nhau

- **Liên chuỗi (Interoperability):** Đảm bảo các hệ thống blockchain từ nhiều tổ chức có thể giao tiếp với nhau, giúp việc xác thực chứng chỉ được thực hiện nhanh chóng dù chúng được phát hành trên các nền tảng khác nhau.

TÀI LIỆU THAM KHẢO

[1] https://react.dev/ (Truy cập: Truy cập: 30/10/2024 lúc 14:17)

[2] https://fullstack.edu.vn/courses/reactjs (Truy cập: 01/11/2024 lúc 09:45)

[3] https://tailwindcss.com/ (Truy cập: 03/11/2024 lúc 18:20)

[4] https://fullstack.edu.vn/blog/tailwind-css-va-cach-cai-dat-co-ban.html (Truy cập: 3/11/2024 lúc 22:11)

[5] https://expressjs.com/ (Truy cập: 02/11/2024 lúc 10:32)

[6] https://fullstack.edu.vn/courses/nodejs (Truy cập: 28/10/2024 lúc 16:05)

[7] https://coin98.net/proof-of-authority-poa-la-gi (Truy cập: 01/11/2024 lúc 12:50))

[8] https://viblo.asia/p/nhung-goc-nhin-dau-tien-ve-ipfs-Az45b9PwlxY (Truy cập: 3 0/10/2024 lúc 19:40)