**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BỘ MÔN TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

---------------o0o---------------

****

**ĐỒ ÁN MÔN HỌC**

**NHẬP MÔN AN TOÀN THÔNG TIN**

**Đề tài:**

**Đăng nhập một lần với các giao thức OpenID, OAuth**

**GVHD: PGS.TS. Nguyễn Linh Giang**

**Nhóm sinh viên: Nhóm 26**

**Đặng Quang Anh 20172942**

**Mai Thị Minh Anh 20172954**

**Nguyễn Đức Bình 20172970**

**Lê Thị Huyền Thanh 20173371**

**TP. HÀ NỘI, THÁNG 5 NĂM 2020**

Mục lục

[CHƯƠNG 1: TỔNG QUAN VỀ OPENID 1](#_Toc42021659)

[1.1. Khái niệm 1](#_Toc42021660)

[1.2. Các thành phần của một hệ thống OpenID 2](#_Toc42021661)

[1.3. Lợi ích của OpenID 4](#_Toc42021662)

[1.3.1. Đăng kí thành viên nhanh và dễ dàng 4](#_Toc42021663)

[1.3.2. Tiếp cận đến khái niệm "web identity" 4](#_Toc42021664)

[1.4. Quy trình hoạt động chính của hệ thống quản lý định danh 5](#_Toc42021665)

[CHƯƠNG 2: PHƯƠNG THỨC HOẠT ĐỘNG CỦA OPENID 5](#_Toc42021666)

[2.1. Cơ chế hoạt động của OpenID 5](#_Toc42021667)

[2.2. Cơ chế Smart mode 7](#_Toc42021668)

[2.3. Quy trình xác định thành phần Identity Provider 8](#_Toc42021669)

[2.4. Quy trình gửi thuộc tính định danh 9](#_Toc42021670)

[2.5. Quy trình kiểm tra thuộc tính định danh 11](#_Toc42021671)

[2.6. Cơ chế Dumb mode 11](#_Toc42021672)

[2.7. Cơ chế xác thực của OpenID 12](#_Toc42021673)

[2.8. Xác thực SASL 14](#_Toc42021674)

[2.9. Mô hình trao đổi khóa Diffie-Hellman 15](#_Toc42021675)

[2.10. Sơ đồ quy trình giao tiếp giữa Client – Server 16](#_Toc42021676)

[2.11. Quy trình xử lý các bước giao tiếp 17](#_Toc42021677)

[CHƯƠNG 3: SO SÁNH OPENID VÀ OAUTH 17](#_Toc42021678)

[3.1. Mục đích 17](#_Toc42021679)

[3.2. Tính năng 18](#_Toc42021680)

[3.3. Triển khai kỹ thuật 19](#_Toc42021681)

[CHƯƠNG 4: DEMO 19](#_Toc42021682)

[4.1. Giới thiệu 19](#_Toc42021683)

[4.2. Giao diện demo 19](#_Toc42021684)

[Danh mục tài liệu tham khảo 24](#_Toc42021685)

***Danh mục hình ảnh***

[Hình 1. Các thành phần của một hệ thống OpenID. 2](#_Toc42331332)

[Hình 2. Trang web sử dụng cơ chế đăng nhập người dùng để định danh 3](#_Toc42331333)

[Hình 3. Quy trình hoạt động chính của hệ thống quản lý định danh 5](#_Toc42331334)

[Hình 4. Giao tiếp giữa các thành phần trong hệ thống OpenID URI là địa chỉ của Identity Provider 6](#_Toc42331335)

[Hình 5. Giao tiếp giữa các thành phần trong hệ thống OpenID với URI không phải là địa chỉ của Identity 7](#_Toc42331336)

[Hình 6. Quy trình xác định thành phần Identity Provider 8](#_Toc42331337)

[Hình 7. Quy trình gửi thuộc tính định danh 9](#_Toc42331338)

[Hình 8. Quy trình kiểm tra thuộc tính định danh 11](#_Toc42331339)

[Hình 9. Quá trình kiểm tra thuộc tính định danh ở chế độ Dumb mode 12](#_Toc42331340)

[Hình 10. Cơ chế xác thực của OpenID 13](#_Toc42331341)

[Hình 11. Xác thực SASL 14](file:///C:\Users\thanh\Documents\baocaoNMANTT_Nhom26.docx#_Toc42331342)

[Hình 12. Mô hình trao đổi khóa Diffie-Hellman 14](#_Toc42331343)

[Hình 13. Sơ đồ quy trình giao tiếp giữa Client – Server 16](#_Toc42331344)

[Hình 14. Giao diện demo 20](#_Toc42331345)

[Hình 15. authorization server 21](#_Toc42331346)

[Hình 16. Giao diên đăng nhập thành công 22](#_Toc42331347)

[Hình 17. ID token và Access token được OpenID provider trả về cho client. 23](#_Toc42331348)

# CHƯƠNG 1: TỔNG QUAN VỀ OPENID

## 1.1. Khái niệm

OpenID là một dịch vụ định danh (Identify) chia sẻ, là một hệ thống đăng nhập một lần không có tính tập trung, cho phép người sử dụng đăng nhập nhiều website khác nhau chỉ bằng 1 định danh số, tránh việc sử dụng các tài khoản và mật khẩu khác nhau cho mỗi website. OpenID là định chuẩn mở, miễn phí và phân quyền cho phép người dùng điều khiển được các thông tin cá nhân của mình công khai trên Internet.

Một OpenID là dạng liên kết URL, URL này có thể là tên miền của website hoặc URL của nhà cung cấp định danh OpenID. Khi đăng nhập với tài khoản OpenID, bạn phải đăng nhập vào Nhà cung cấp dịch vụ định danh để kiểm tra tính hợp lệ của tài khoản. OpenID là một phương thức giúp bạn xác thực tài khoản đăng ký tại một provider duy nhất mà bạn tin tưởng và cho phép người dùng thực hiện việc đăng nhập vào các lần sau.[1]

Được phát triển bởi tổ chức phi lợi nhuận OpenID Foundation, OpenID cho phép user có thể được authen bởi rất nhiều website (Relying Parties hoặc RP) sử dụng service của bên thứ 3. Nó giảm được việc phái thiết lập riêng logic sign up/login cho mỗi website, cho phép các user có thể login tới nhiều webstie ko hề liên quan tới nhau mà ko cần phải có những định danh và password riêng cho mỗi site. [3]

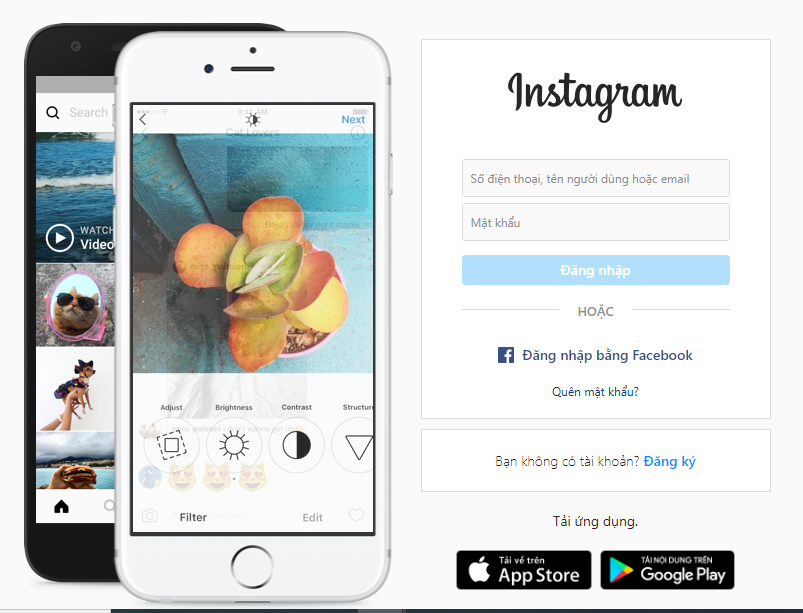
## 1.2. Các thành phần của một hệ thống OpenID

A picture containing clock

Description automatically generated

Hình . Các thành phần của một hệ thống OpenID.

Relying Party: là dịch vụ sử dụng cơ chế định danh để chứng thực. Ví dụ, một số trang web sử dụng cơ chế đăng nhập người dùng để định danh như trang Zingplay, trang ... Hiện nay đã có rất nhiều thành phần Relying Party trên mạng. Phần lớn trong số đó đã hỗ trợ định danh bằng hệ thống khác như tài khoản email của Facebook hay Gmail.



Hình . Trang web sử dụng cơ chế đăng nhập người dùng để định danh

Identity Provider: là thành phần có nhiệm vụ quản lý các thuộc tính định danh của người dùng hệ thống. IdP có chức năng truyền những thông tin cần thiết để thực hiện chứng thực đến Relying Party sau khi xác định đúng là người dùng đang sử dụng dịch vụ. Hiện nay đã có rất nhiều hệ thống nổi tiếng đã xây dựng thành phần Identity Provider cho riêng mình dựa trên cơ chế của hệ thống OpenID như Google, Yahoo…

Identity Selector (Browser): là thành phần trung gian của hệ thống, là cầu nối giữa người dùng, Relying Party, Identity Provider. Mọi hoạt động của thành phần này được điều khiển trực tiếp bởi người dùng.

## 1.3. Lợi ích của OpenID

### 1.3.1. Đăng kí thành viên nhanh và dễ dàng

OpenID còn rất dễ dàng để login. Với OpenID bạn chỉ cần nhớ 1 username và 1 password. Bởi vì bạn login vào website với OpenID của bạn, cho nên bạn chỉ cần bảo mật với OpenID là đủ rồi. Password này của bạn thực tế là chỉ được quản lý bởi OpenID provider của bạn, ko phải là các website bạn đang login, khi bạn login vào OpenID provider sẽ bảo với các website đó là bạn đang login đấy và bạn là ai. Ko có bất kì một website nào nhìn được password của bạn, cho nên bạn ko cần phải quá bận tâm về vấn đề bảo mật khi sử dụng OpenID. [1].

### 1.3.2. Tiếp cận đến khái niệm "web identity"

Bởi vì OpenID định danh bạn là duy nhất trên Internet, một khi bạn thiết lập mình như là người sử dụng OpenID, bất cứ khi nào ai đó nhìn thấy OpenID của bạn được sử dụng, bất cứ nơi nào trên Internet, họ sẽ biết rằng đó chính là bạn. Tương tự như vậy, nếu bạn truy cập vào một trang web mới và thấy rằng một người nào đó với OpenID của bạn bạn, bạn có thể gần như chắc chắn rằng đó là bạn của bạn.  
Điều đó có thể khiến bạn lo lắng rằng OpenID là sẽ làm cho tất cả các hoạt động trực tuyến của bạn trở nên quá rõ ràng và dễ dàng bị phát hiện. Hãy yên tâm, mặc dù OpenID thống nhất thông tin về bạn, nhưng nó chỉ hợp nhất thông tin mà bạn đã cho phép public. Bạn có thể chọn, sử dụng OpenID, thông tin nào? và thông tin đó ai sẽ thấy được.[3]

## 1.4. Quy trình hoạt động chính của hệ thống quản lý định danh

A picture containing clock, drawing

Description automatically generated

Hình . Quy trình hoạt động chính của hệ thống quản lý định danh

# CHƯƠNG 2: PHƯƠNG THỨC HOẠT ĐỘNG CỦA OPENID

## 2.1. **Giao tiếp giữa các thành phần trong hệ thống OpenID**

A picture containing clock

Description automatically generated

Hình . Giao tiếp giữa các thành phần trong hệ thống OpenID URI là địa chỉ của Identity Provider

A screenshot of a video game

Description automatically generated

Hình . Giao tiếp giữa các thành phần trong hệ thống OpenID với URI không phải là địa chỉ của Identity

## 2.2. Cơ chế hoạt động của OpenID

OpenID có hai cơ chế hoạt động chính:

-Smart mode

-Dumb mode

Hai cơ chế này được dựa trên khả năng của Relying Party.

Trong chế độ Smart mode, Relying Party có khả năng lưu lại khóa chia sẻ bí mật cho việc chứng thực sau đó. Ngược lại, ở chế độ Dumb mode, Relying Party không có khả năng lưu trữ thông tin nên phải thực hiện thêm một số bước để hoàn tất quá trình chứng thực.

## 2.3. Cơ chế Smart mode

Cơ chế hoạt đông Smart mode có thể chia làm 3 quy trình con sau:

-Quy trình xác định thành phần Identity Provider

-Quy trình gửi thuộc tính định danh

-Quy trình kiểm tra thuộc tính định danh

## 2.4. Quy trình xác định thành phần Identity Provider

A close up of a clock

Description automatically generated

Hình . Quy trình xác định thành phần Identity Provider

Bước 1.1: Người dùng sẽ nhập địa chỉ URL của Relying Party vào Browser.

Bước 1.2: Dựa vào URL người dùng nhập vào, Browser sẽ giao tiếp với thành phần Relying Party.

Bước 1.3: Relying Party sẽ trả về Browser trang đăng nhập có hỗ trợ OpenID trong đó có textbox yêu cầu người dùng nhập vào URI của Identity Provider.

Bước 1.4: Browser hiển thị trang đăng nhập cho người dùng.

Bước 1.5: Người dùng sẽ điền URI của Identity Provider vào Browser. Sau khi điền vào URI, người dùng nhấn nút “Đăng nhập”.

Bước 1.6: Browser sẽ chuyển thông tin về URI người dùng nhập vào đến Relying Party. Relying Party sẽ lấy thông tin về URI người dùng nhập vào để xác định được thành phần Identity Provider tương ứng. URI người dùng nhập vào sẽ có hai loại:

-Loại 1: URI đó chính là địa chỉ của Identity Provider. Trong trường hợp này, Relying Party đã có được địa chỉ của Identity Provider chính là URI người dùng nhập cung cấp. ---Loại 2: URI này không phải là địa chỉ của Identity Provider. Trong trường hợp này, thành phần Relying Party phải dùng Yadis để lấy địa chỉ của Identity Provider. Dịch vụ Yadis có vai trò nhận vào một URI và sẽ trả về địa chỉ và thông tin về Identity Provider tương ứng.

## 2.5. Quy trình gửi thuộc tính định danh

A screenshot of a video game

Description automatically generated

Hình . Quy trình gửi thuộc tính định danh

Bước 2.1: Relying Party sau khi xác định được thành phần Identity Provider ở quy trình xác định thành phần Identity Provider (xem phần trước). Bước 2.1 là bước tùy chọn bao gồm hai trường hợp xảy ra như sau:

- Trường hợp 1: Relying Party và Identity Provider chưa có khóa chia sẻ bí mật ở những lần định danh trước đây, hoặc khóa chia sẻ bí mật đã hết thời gian sử dụng. Trong trường hợp này, Relying Party sẽ kết nối bằng một kênh truyền an toàn với Identity Provider để chia sẻ khóa bí mật. Khóa bí mật sẽ được sử dụng để kiểm tra các thuộc tính định danh ở quy trình kiểm tra thuộc tính định danh sau này ở bước 3.1 hay những lần định danh sau đó.

-Trường hợp 2: Nếu thành phần Relying Party đã có được khóa bí mật chưa hết thời gian sử dụng ở các lần thực hiện định danh trước đây thì không cần phải thực hiện bước này. Vì vậy bước 2.1 là bước tùy chọn.

Bước 2.2: Relying Party gửi danh sách tên các thuộc tính yêu cầu Identity Provider cung cấp để chứng thực.

Bước 2.3: Identity Provider sẽ yêu cầu người dùng đăng nhập bằng cách trả về Browser trang đăng nhập.

Bước 2.4: Browser sẽ hiển thị trang đăng nhập đến người dùng.

Bước 2.5: Người dùng sẽ đăng nhập vào Identity Provider (ví dụ, người dùng sẽ nhập vào username và password để đăng nhập). Sau đó, người dùng sẽ nhấn nút “đăng nhập”.

Bước 2.6: Browser sẽ chuyển thông tin đăng nhập người dùng đến Identity Provider để kiểm tra

Bước 2.7: Identity Provider sẽ kiểm tra thông tin đăng nhập. Sau đó, Identity Provider sẽ dựa trên danh sách tên các thuộc tính yêu cầu từ Relying Party; Identity Provider sẽ tạo một thông điệp có chứa các thuộc tính tương ứng. Cuối cùng, Identity Provider sẽ ký trên danh sách các thuộc tính định danh và trả về Browser.

Bước 2.8: Browser sẽ hiện lên tất cả thuộc tính định danh nhận được từ Identity Provider cho người dùng.

Bước 2.9: Người dùng sẽ kiểm tra các thuộc tính định danh có hợp lệ. Sau đó, người dùng sẽ xác nhận truyền các thuộc tính định danh.

Bước 2.10: Browser sẽ truyền các thông tin định danh của người dùng đến Relying Party.

## 2.6. Quy trình kiểm tra thuộc tính định danh

A picture containing clock, drawing

Description automatically generated

Hình . Quy trình kiểm tra thuộc tính định danh

Bước 3.1: Dựa trên thuộc tính định danh nhận được từ thành phần Identity Provider ở quy trình gởi thuộc tính định danh (xem phần trước) cùng với khóa chia sẻ bí mật được tạo ra ở bước 2.1. Relying Party sẽ kiểm tra xem thuộc tính định danh có hợp lệ hay không.

Bước 3.2: Relying Party trả về kết quả định danh về Browser.

Bước 3.3: Browser sẽ hiển thị kết quả định danh đến người dùng

## 2.7. Cơ chế Dumb mode

Chế độ Dumb mode cũng tương tự như chế độ Smart mode. Nhưng ở chế độ Dumb mode, Relying Party không có khả năng lưu trữ các thông tin trước đó. Do đó thành phần Identity Provider và Relying Party sẽ chưa có khóa chia sẻ để kiểm tra thuộc tính định danh. Vì vậy, ở bước 3.1 trong quy trình kiểm tra thuộc tính định danh (xem phần trước) trong chế độ Dumb mode, Relying Party cần phải tạo kết nối an toàn với Identity Provider để kiểm tra thuộc tính định danh. Các bước khác ở chế độ Dumb mode hoàn toàn giống với chế độ Smart mode.

Quá trình kiểm tra thuộc tính định danh ở chế độ Dumb mode khác so với chế độ Smart mode ở bước 3.1.

A picture containing clock

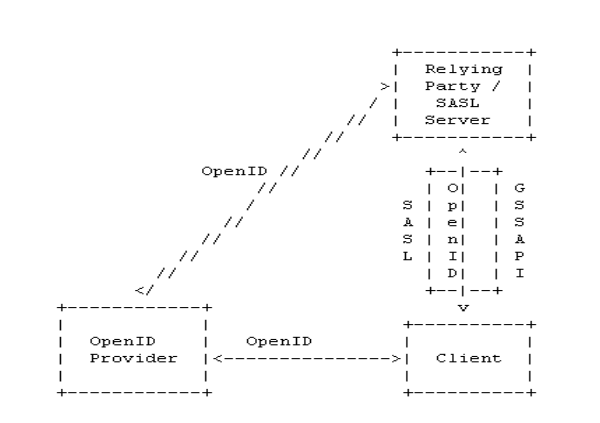
Description automatically generated

Hình . Quá trình kiểm tra thuộc tính định danh ở chế độ Dumb mode

## 2.8. Cơ chế xác thực của OpenID

OpenID sử dụng cơ chế xác thực SASL (Simple Authentication and Security Layer), sử dụng các giao thức lớp ứng dụng như IMAP, POP, XMPP với mục tiêu modules hóa và bảo mật lớp.

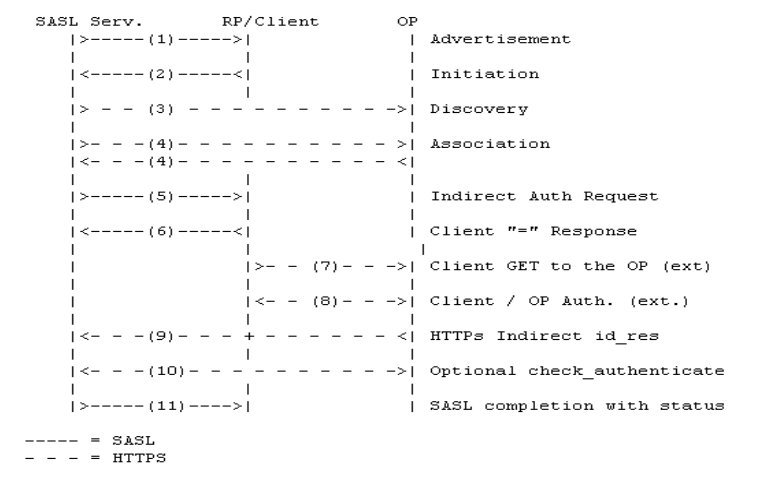
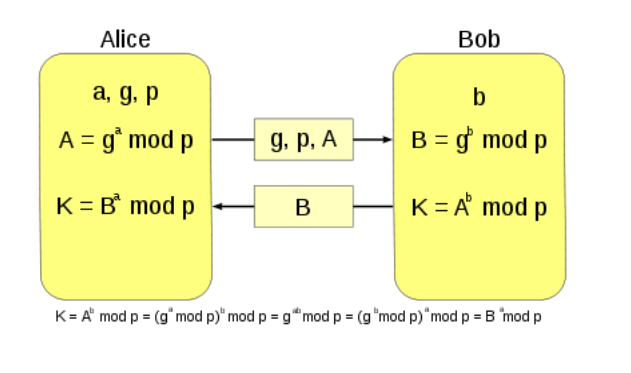
OpenID ban đầu được hình dung cho HTTP và HTML, người dùng sẽ chuyển hướng của Relying Party vào một nhà cung cấp danh tính xác thực người dùng và sau đó gửi thông tin nhận dạng về thuộc tính khác (hoặc trực tiếp hoặc gián tiếp) với Relying Party.



Hình . Cơ chế xác thực của OpenID

Khi xem xét lưu lượng trong SASL, Relying Party và người dùng đều phải thay đổi mã để thi hành cơ chế SASL.

## 2.8. Xác thực SASL

2.9. Mô hình trao đổi khóa Diffie-Hellman

Hình . Xác thực SASL

Hình 12. Mô hình trao đổi khóa Diffie-Hellman

Diffie-Hellman là một thuật toán dùng để trao đổi khóa chứ không dùng để bảo vệ tính bí mật của dữ liệu. Tuy nhiên, Diffie-Hellman lại có ích trong giai đoạn trao đổi khóa bí mật của các thuật toán mật mã đối xứng.

Thuật toán trao đổi khóa Diffie-Hellman dựa trên phép logarit rời rạc. Cho trước một số g và x=gk, để tìm k ta thực hiện phép logarit: k= logg (x). Tuy nhiên, nếu cho trước g, n và (gk mod n), thì quá trình xác định k được thực hiện theo cách khác với cách ở trên và được gọi là logarit rời rạc.

Gọi n là một số nguyên tố lớn và g là một cơ số sinh (generator, số nguyên nhỏ) thỏa điều kiện: với mọi x ∈ {1, 2, …, n-1}, ta luôn tìm được số y sao cho x=gy mod n.

Giá trị n và g được phổ biến công khai giữa các thực thể trao đổi khóa. Sau đó user A tạo ra một số riêng Xa < n, tính giá trị Ya = (gxa mod n) và gởi cho B. Tương tự, user B cũng tạo ra một số riêng Xb < n tính gá trị Yb = (gxb mod n) và gởi lại cho A. Xa và Xb tương đương khóa private, Ya và Yb tương đương khóa public. • User B xác định được khóa bí mật dùng cho phiên làm việc bằng cách tính giá trị (gxa mod n) xb = (gxaxb mod n). Bằng cách tương tự, user A cũng xác định được cùng khóa bí mật này bằng cách tính giá trị (gxb mod n) xa = (gxaxb mod n).

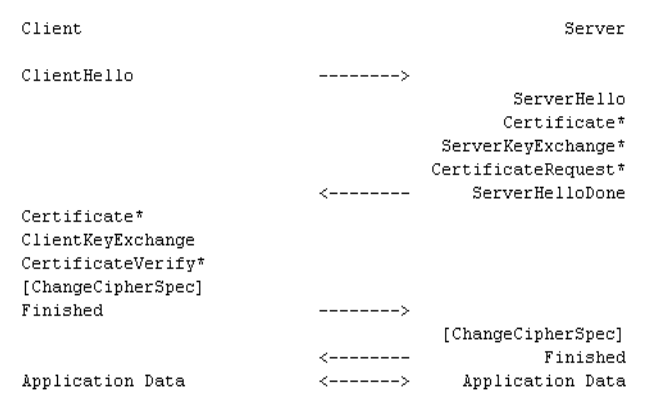
Giả sử trong quá trình trao đổi các giá trị, phía tấn công bắt được (gxa mod n) và (gxb mod n), họ rất khó xác định được Xa và Xb vì độ phức tạp của phép toán logarit rời rạc là rất cao.

Ví dụ: với n=11 và g=5

– A: x=7 =>y=57 mod 11 =3 k = 47 mod 11 = 5

– B: x=3 =>y=53 mod 11 = 4 k = 33 mod 11 = 5

## 2.10. Sơ đồ quy trình giao tiếp giữa Client – Server



Hình 13. Sơ đồ quy trình giao tiếp giữa Client – Server

## 2.11. Quy trình xử lý các bước giao tiếp

Các thông số mật mã của trạng thái session được tạo ra bởi giao thức bắt tay TLS, hoạt động trên đầu trang của TLS Layer. Khi một khách hàng TLS và máy chủ đầu tiên bắt đầu giao tiếp, họ đồng ý trên một phiên bản giao thức, chọn thuật toán mã hóa, tùy chọn xác thực lẫn nhau, và sử dụng kỹ thuật mã hóa khóa công khai để tạo ra bí mật được chia sẻ. Các giao thức bắt tay TLS bao gồm các bước sau đây:

Trao đổi thông điệp bắt tay thỏa thuận đồng ý giữa các bên về các thuật toán, trao đổi ngẫu nhiên giá trị, và kiểm tra phiên giao dịch.

Trao đổi các thông số mật mã cần thiết để cho phép người dùng cuối và máy chủ đồng ý trên một khóa bí mật premaster.

Giấy chứng nhận exchange và thông tin mật mã để cho phép máy khách và máy chủ xác thực bản thân.

Tạo ra một bí mật tổng thể từ bí mật premaster và trao đổi ngẫu nhiên giá trị.

Cung cấp các thông số an ninh lớp ghi.

Cho phép các máy người dùng cuối và máy chủ xác minh, tính toán các thông số an ninh và những cái bắt tay xảy ra mà không làm bị can thiệp từ bên thứ ba tấn công.

# CHƯƠNG 3: SO SÁNH OPENID VÀ OAUTH

## 3.1. Mục đích

OpenID được tạo để xác thực liên kết, nghĩa là cho phép bên thứ ba xác thực người dùng của bạn cho bạn, bằng cách sử dụng các tài khoản họ đã có. Thuật ngữ được liên kết ở đây rất quan trọng vì toàn bộ quan điểm của OpenID là bất kỳ nhà cung cấp nào cũng có thể được sử dụng (ngoại trừ danh sách trắng). Bạn không cần phải chọn trước hoặc đàm phán thỏa thuận với các nhà cung cấp để cho phép người dùng sử dụng bất kỳ tài khoản nào khác mà họ có.

OAuth được tạo ra để loại bỏ nhu cầu người dùng chia sẻ mật khẩu của họ với các ứng dụng của bên thứ ba. Nó thực sự bắt đầu như một cách giải quyết vấn đề OpenID: nếu bạn hỗ trợ OpenID trên trang web của mình, bạn không thể sử dụng thông tin xác thực HTTP (tên người dùng và mật khẩu) để cung cấp API vì người dùng không có mật khẩu trên trang web của bạn.

Vấn đề xảy ra với sự phân tách OpenID này để xác thực và OAuth cho ủy quyền là cả hai giao thức có thể thực hiện nhiều điều giống nhau. Mỗi cái cung cấp một tập hợp các tính năng khác nhau được mong muốn bởi các triển khai khác nhau nhưng về cơ bản, chúng có thể thay thế cho nhau. Về cốt lõi, cả hai giao thức đều là phương thức xác minh xác nhận (OpenID bị giới hạn ở ‘ở đây tôi là ai’, trong khi OAuth cung cấp ‘access token’ có thể được trao đổi cho bất kỳ xác nhận được hỗ trợ nào thông qua API).

## 3.2. Tính năng

Cả hai giao thức cung cấp một cách để một trang web chuyển hướng người dùng ở một nơi khác và quay lại với một xác nhận có thể kiểm chứng. OpenID cung cấp xác nhận danh tính trong khi OAuth chung chung hơn dưới dạng mã thông báo truy cập, sau đó có thể được sử dụng để "đặt câu hỏi cho nhà cung cấp OAuth". Tuy nhiên, mỗi loại đều hỗ trợ các tính năng khác nhau:

OpenID - tính năng quan trọng nhất của OpenID là quá trình khám phá của nó. OpenID không yêu cầu mã hóa cứng mỗi nhà cung cấp bạn muốn sử dụng trước thời hạn. Sử dụng khám phá, người dùng có thể chọn bất kỳ nhà cung cấp bên thứ ba nào họ muốn xác thực. Tính năng khám phá này cũng đã gây ra hầu hết các sự cố của OpenID bởi vì cách nó được triển khai là bằng cách sử dụng HTTP URI làm định danh mà hầu hết người dùng web không nhận được. Các tính năng khác OpenID có hỗ trợ đăng ký khách hàng đặc biệt bằng cách sử dụng trao đổi DH, chế độ ngay lập tức để trải nghiệm người dùng cuối được tối ưu hóa và cách xác minh các xác nhận mà không cần thực hiện một chuyến đi khứ hồi nào cho nhà cung cấp.

OAuth - tính năng quan trọng nhất của OAuth là mã thông báo truy cập cung cấp phương thức lâu dài để thực hiện các yêu cầu bổ sung. Không giống như OpenID, OAuth không kết thúc bằng xác thực nhưng cung cấp mã thông báo truy cập để có quyền truy cập vào các tài nguyên bổ sung được cung cấp bởi cùng dịch vụ của bên thứ ba. Tuy nhiên, vì OAuth không hỗ trợ khám phá, nó yêu cầu lựa chọn trước và mã hóa cứng các nhà cung cấp mà bạn quyết định sử dụng. Người dùng truy cập trang web của bạn không thể sử dụng bất kỳ số nhận dạng nào, chỉ những người được bạn chọn trước. Ngoài ra, OAuth không có khái niệm về danh tính nên sử dụng nó để đăng nhập có nghĩa là thêm một tham số tùy chỉnh (như được thực hiện bởi Twitter) hoặc thực hiện một lệnh gọi API khác để có được người dùng hiện đang "đăng nhập".

## 3.3. Triển khai kỹ thuật

Hai giao thức chia sẻ một kiến ​​trúc phổ biến trong việc sử dụng chuyển hướng để có được sự cho phép của người dùng. Trong OAuth, người dùng cho phép truy cập vào các tài nguyên được bảo vệ của họ và trong OpenID, đối với danh tính của họ. Nhưng đó là tất cả những gì họ chia sẻ.

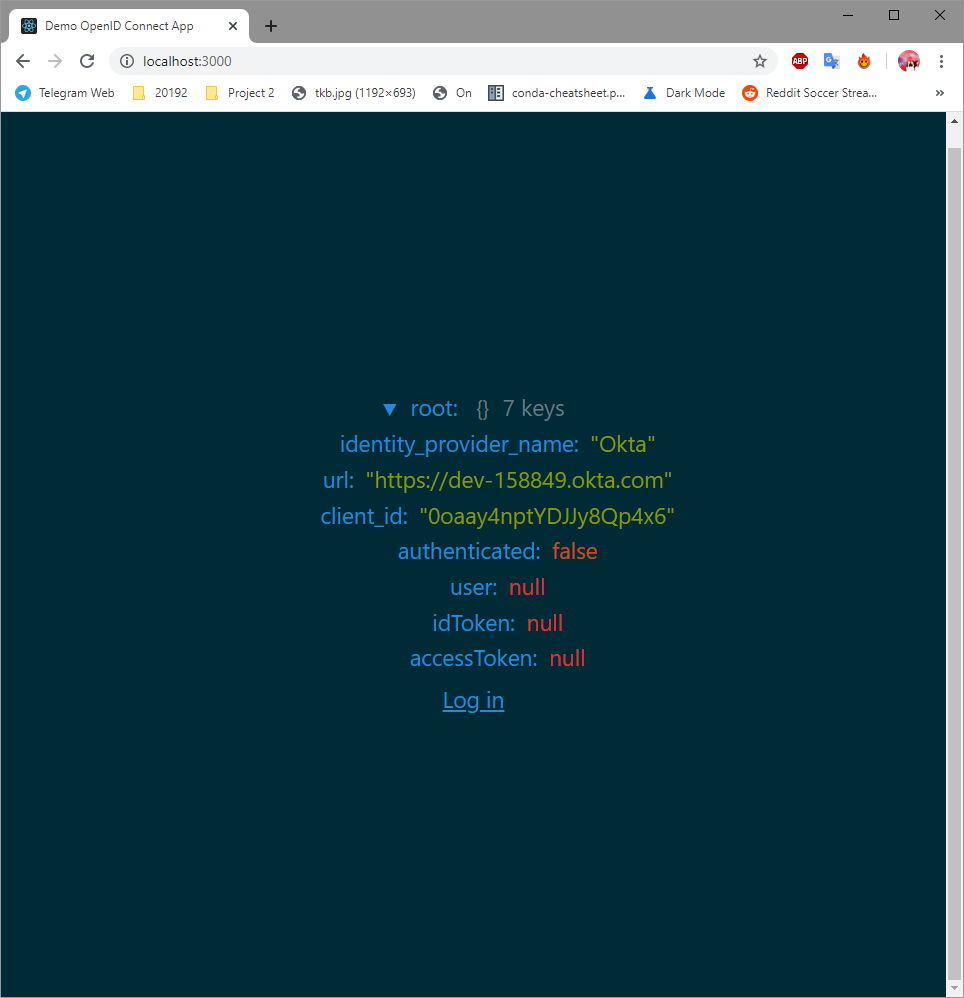
Mỗi giao thức có một cách tính chữ ký khác nhau được sử dụng để xác minh tính xác thực của yêu cầu hoặc phản hồi và mỗi giao thức có các yêu cầu đăng ký khác nhau.[2]

# CHƯƠNG 4: DEMO

## 4.1. Giới thiệu

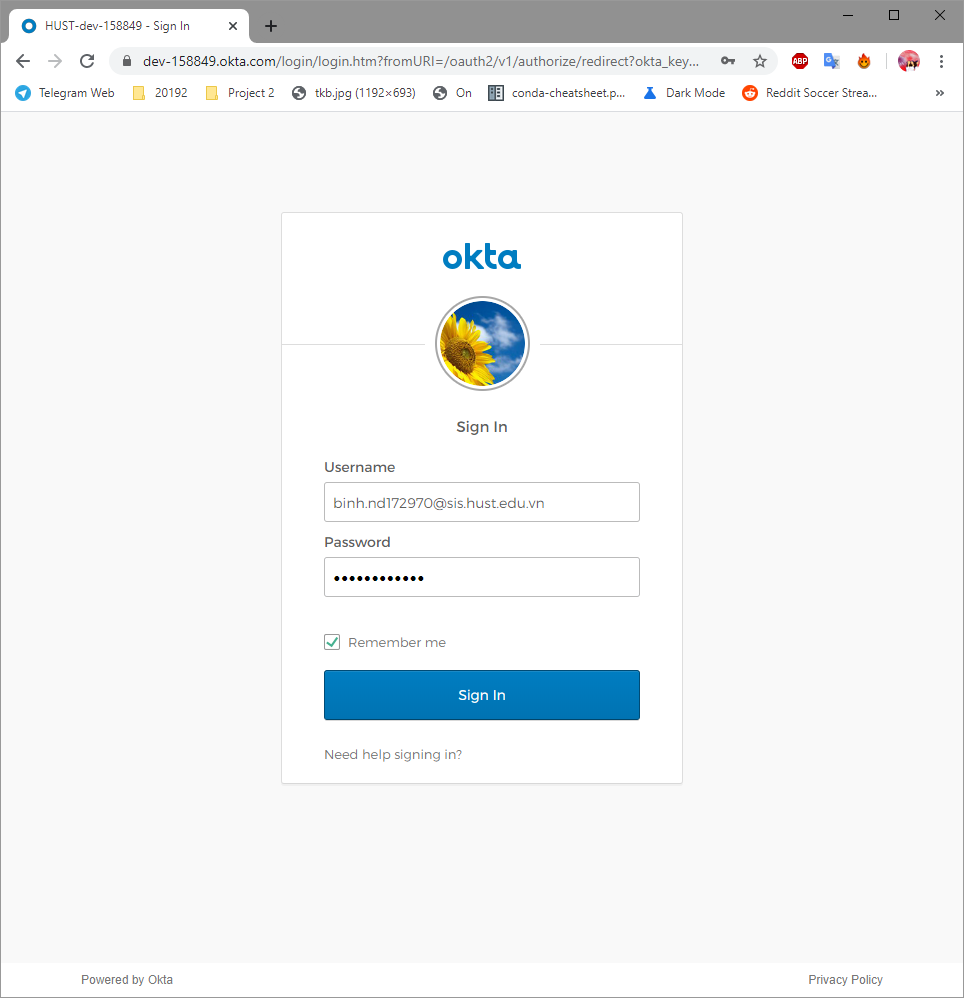
* Demo sử dụng giao thức OpenID Connect. OpenID Connect được sử dụng để xác thực người dùng bằng ứng dụng web. Ứng dụng sử dụng ID token được trả về từ authorization server để biết ngượi dùng đã được xác thực chưa và lấy các thông tin của người dùng. OAuth2.0 trong OpenID Connect được sử dụng để ủy quyền cho người dùng truy cập và sử dụng API. Một access token được sử dụng bởi resource server để xác thực mức độ tin cậy và quyền truy cập của người dùng. Khi sử dụng OpenID Connect, authorization server sẽ xác dụng người dùng và cấp phát ID token và/ hoặc access token.
* Authorization server là một công cụ để tạo ra các OpenID Connect token và để áp dụng các chính sách truy cập. Mỗi authorization server sẽ có một issuer URI duy nhất và có một ký khóa riêng để bảo mật.
* Demo sử dụng Okta Developer để xác thực và ủy quyền người dùng.

## 4.2. Giao diện demo

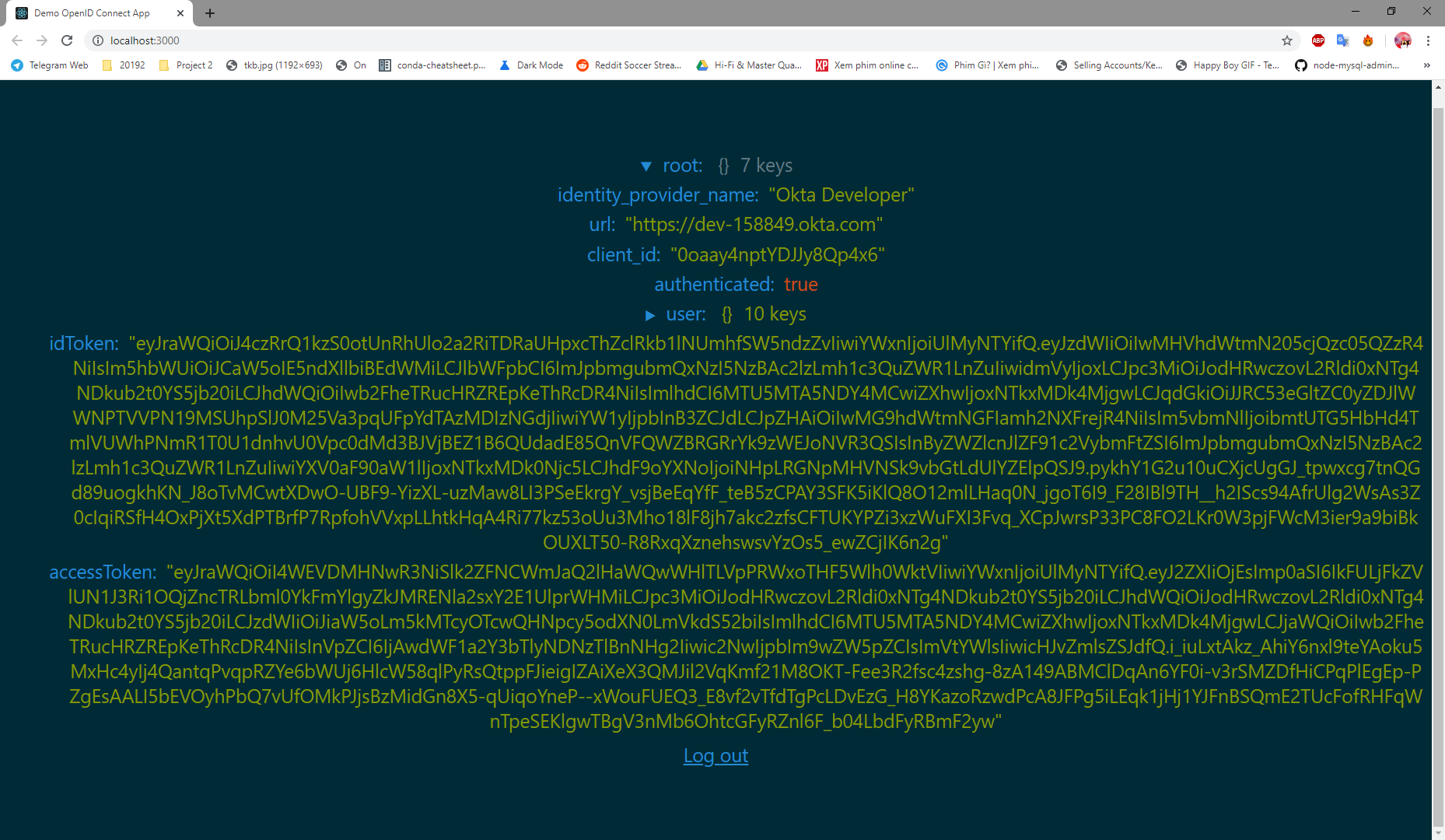


Hình 14. Giao diện demo

* Chúng ta có các thông tin sau:
  + Tên Identity Provider (identity\_provider\_name): Okta Developer
  + Org authorization server URL: <https://dev-158849.okta.com>
  + Client ID (là duy nhất và được Identity Provider cung cấp cho mỗi user): 0oaay4nptYDJJy8Qp4x6
  + Trạng thái xác thực (authenticated) là false do chúng ta chưa login
  + Khi chưa login và authenticate, chúng ta sẽ chưa có các thông tin user, idToken và accessToken.
* Khi nhấn login, ứng dụng web sẽ chuyển hướng tới authorization server.

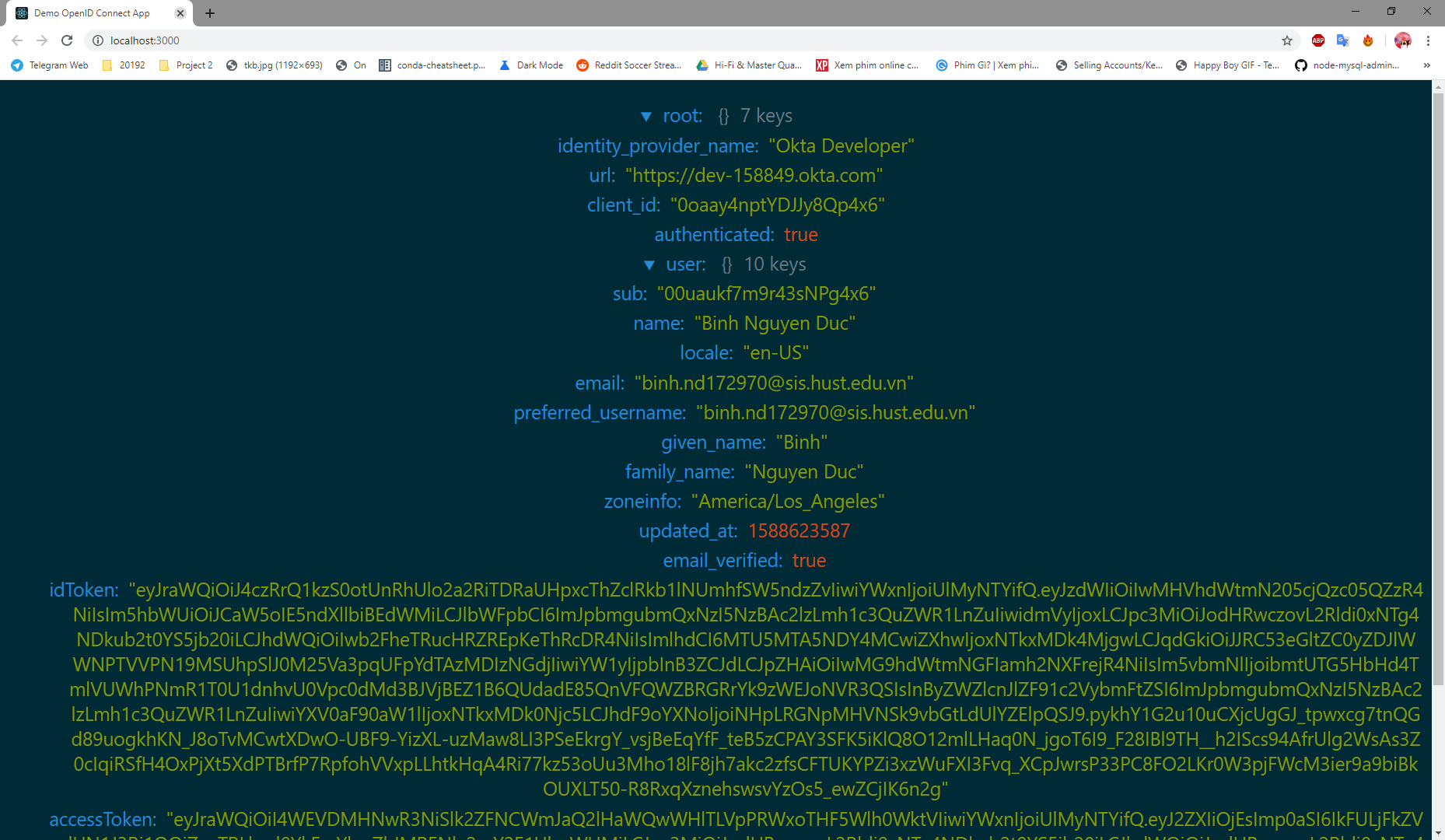


Hình 15. authorization server

* Sau khi đăng nhập thành công thì web sẽ có giao diện như sau:

Hình 16. Giao diên đăng nhập thành công

* Lúc này khi đã được xác thực, trạng thái authenticated đổi thành true, thông tin về người dùng cũng như ID token và Access token được OpenID provider trả về cho client.



Hình 17. ID token và Access token được OpenID provider trả về cho client.

* ID token: “eyJraWQiOiJ4czRrQ1kzS0otUnRhUlo2a2RiTDRaUHpxcThZclRkb1lNUmhfSW5ndzZvIiwiYWxnIjoiUlMyNTYifQ..pykhY1G2u10uCXjcUgGJ\_tpwxcg7tnQGd89uogkhKN\_J8oTvMCwtXDwO-UBF9-YizXL-uzMaw8LI3PSeEkrgY\_vsjBeEqYfF\_teB5zCPAY3SFK5iKlQ8O12mlLHaq0N\_jgoT6I9\_F28IBl9TH\_\_h2IScs94AfrUlg2WsAs3Z0cIqiRSfH4OxPjXt5XdPTBrfP7RpfohVVxpLLhtkHqA4Ri77kz53oUu3Mho18lF8jh7akc2zfsCFTUKYPZi3xzWuFXI3Fvq\_XCpJwrsP33PC8FO2LKr0W3pjFWcM3ier9a9biBkOUXLT50-R8RxqXznehswsvYzOs5\_ewZCjIK6n2g”

Access token: “eyJraWQiOiI4WEVDMHNwR3NiSlk2ZFNCWmJaQ2lHaWQwWHlTLVpPRWxoTHF5Wlh0WktVIiwiYWxnIjoiUlMyNTYifQ.eyJ2ZXIiOjEsImp0aSI6IkFULjFkZVlUN1J3Ri1OQjZncTRLbml0YkFmYlgyZkJMRENla2sxY2E1UlprWHMiLCJpc3MiOiJodHRwczovL2Rldi0xNTg4NDkub2t0YS5jb20iLCJhdWQiOiJodHRwczovL2Rldi0xNTg4NDkub2t0YS5jb20iLCJzdWIiOiJiaW5oLm5kMTcyOTcwQHNpcy5odXN0LmVkdS52biIsImlhdCI6MTU5MTA5NDY4MCwiZXhwIjoxNTkxMDk4MjgwLCJjaWQiOiIwb2FheTRucHRZREpKeThRcDR4NiIsInVpZCI6IjAwdWF1a2Y3bTlyNDNzTlBnNHg2Iiwic2NwIjpbIm9wZW5pZCIsImVtYWlsIiwicHJvZmlsZSJdfQ.i\_iuLxtAkz\_AhiY6nxl9teYAoku5MxHc4yIj4QantqPvqpRZYe6bWUj6HlcW58qlPyRsQtppFJieigIZAiXeX3QMJil2VqKmf21M8OKT-Fee3R2fsc4zshg-8zA149ABMClDqAn6YF0i-v3rSMZDfHiCPqPlEgEp-PZgEsAALI5bEVOyhPbQ7vUfOMkPJjsBzMidGn8X5-qUiqoYneP--xWouFUEQ3\_E8vf2vTfdTgPcLDvEzG\_H8YKazoRzwdPcA8JFPg5iLEqk1jHj1YJFnBSQmE2TUcFofRHFqWnTpeSEKlgwTBgV3nMb6OhtcGFyRZnl6F\_b04LbdFyRBmF2yw”

# Danh mục tài liệu tham khảo

[1] Báo cáo về OpenID, Khoa Công nghê Thông tin trường Đại học kĩ thuật công nghệ Thành phố Hồ Chí Minh

[2] <https://stackoverflow.com/questions/1087031/whats-the-difference-between-openid-and-oauth>

[3] <https://techblog.vn/tim-hieu-ve-openid>