**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**🙞☸🙜**

**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP HÀ NỘI**

****

**ĐỒ ÁN MÔN HỌC**

**AN TOÀN VÀ BẢO MẬT THÔNG TIN**

**Giáo viên hướng dẫn : Mai Hà An**

**Sinh viên thực hiện :**

1. **2174801040866 – Nguyễn Hữu Thắng**
2. **2174801040909 – Phạm Minh Đức**

**Chuyên ngành : Hệ thống thông tin**

**TÓM TẮT ĐỒ ÁN**

|  |  |
| --- | --- |
| Họ và tên | Nhiệm vụ |
| **Nguyễn Hữu Thắng** | Tìm kiếm thông tin, làm chỉnh sửa word, powerpoint báo cáo |
| **Phạm Minh Đức** | Triển khai đồ án, viết chạy code, tìm thông tin |

- Lớp: K66 HTTT

- Giảng viên hướng dẫn: Mai Hà An

- Tên đề tài :

BẢO MẬT THÔNG TIN MỘT TRANG WEB BẤT KỲ BẰNG OTP SMS VÀ RECAPTCHA

- Tóm tắt :

Những nội dung chính được nghiên cứu trong đồ án và kết quả đạt được :

+ Nghiên cứu về cách bảo mật một trang web bất kì

+ Nghiên cứu và triển khai bảo mật bằng Recaptcha và mã OTP

**MỤC LỤC**

LỜI MỞ ĐẦU

I. Mô tả đồ án ……………………………………………….1

1. Đồ án được tạo bởi

2. Mục tiêu của đồ án

3. Những khó khăn khi làm đồ án

II. Cách thực hoạt động của đồ án…………………………….2

III. Cách bước thực hiện đồ án………………………………..3

1.Lấy id và key trong firebase

2. Tích hợp vào đồ án

IV. Cách chạy đồ án…………………………………………..8

TÀI LIỆU THAM KHẢO

**LỜI MỞ ĐẦU**

Ngày nay, công nghệ thông tin đang trên đà phát triển

mạnh mẽ. Thế nhưng đi kèm với nó là việc mất an toàn về thông

tin, dữ liệu của một cá nhân hay tổ chức. Do đó việc bảo mật

thông tin an toàn là một điều vô cùng quan trọng. Đây cũng là ý

do nhóm chúng em lựa chọn đề tài là bảo mật thông tin cho một

trang web bằng việc xác thực hai lớp. Điều này có thể giúp cho

thông tin trong trang web được bảo mật một cách kĩ càng, ngăn

chặn được sự xâm nhập trái phép từ bên ngoài.

Nhóm em đã cố gắng hoàn thành đề tài trong phạm vi và

khả năng có thể. Tuy nhiên sẽ không thể tránh khỏi những thiếu

xót. Chúng em rất mong nhận được sự cảm thông và tận tình chỉ

bảo của thầy. Xin chân thành cảm ơn!

I. Mô tả đồ án

1. Dự án được tạo bởi :

- ReactJS

- Bootstrap

- Firebase

2. Mục tiêu của đồ án

- Recaptcha

Sử dụng mã Recaptcha để tránh những cuộc tấn công DDOS và chặn những truy cập tự động không mong muốn vào trang web và các tác nhân độc hại khác.

Ngăn chặn spam

- OTP SMS

Sử dụng để đăng nhập vào một trang web nhằm tăng tính bảo mật cho người dùng, mã xác thực SMS cung cấp một số lớp bảo mật bổ sung bằng cách yêu cầu người dùng nhập một mã duy nhất được gửi đến điện thoại di động của họ. Điều này giúp ngăn chặn truy cập trái phép từ việc xâm nhập vào tài khoản của người dùng.

3.Những khó khăn gặp phải khi làm đồ án

- Khó tìm kiếm những nguồn tài liệu, truy xuất thông tin.

1

- Gặp nhiều trở ngại trong việc viết code, thường xuyên xảy ra lỗi trong vấn đề lập trình.

- Việc co ngắn, kéo dài dung lượng bài viết sao cho hợp lí, chi tiết để trình bày đồ án còn gặp nhiều hạn chế.

II. Cách thức hoạt động của đồ án

Đồ án sẽ sử dụng dịch vụ của Firebase

(firebase.google.com) để tạo xác thực hai lớp :

- Recaptcha

- OTP SMS

- Khi người dùng nhấn vào “đăng nhập bằng điện thoại” thì ngay lập tức trang sẽ hiện ra form đăng nhập có một ô input để nhập số điện thoại, người dùng sẽ nhập số điện thoại của mình vào.

- Sau khi nhập số điện thoại, người dùng nhấn vào Send OTP lập tức sẽ hiện ra hộp xác thực CAPTCHA với câu "Tôi không phải là người máy”. Sau khi xác thực thành công sẽ tiến hành gửi mã OTP.

- Mã OTP sẽ được về thông qua SMS của số điện thoại người dùng vừa nhập, người dùng sẽ sử dụng mã OTP đó để đăng nhập vào trang web.

2

III. Các bước thực hiện đồ án

1. Lấy id và key trong firebase

- Đăng ký và đăng nhập tài khoản vào website firebase.google.com

- Tạo một dự án mới trong firebase

- Vào mục Authentication

- Chọn mục Sign-in method -> Add new provider

- Enable các mục Email/Password, Google, Phone

- Tiếp theo sẽ them firebase cho web app

- Vào mục project setting để lấy những key và id mà firebase cung cấp

2. Tích hợp vào đồ án

- Tạo file firebase.js để lưu trữ các key và id mà firebase cung cấp, key và id sẽ được truyền vào dưới dạng call back và xuất ra dưới dạng mô đun.

- Tạo file context để cấu hình các dịch vụ của firebase để quản lý chức năng và xác thực người dùng.

- Sử dụng các hàm và các thuộc tính được định nghĩa trong firebase như signInWithEmailAndPassword để xác thực người dùng bằng email và mật khẩu

3

createUserWithEmailAndPassword để tạo mới người dùng bằng email và mật khẩu,...

- Định nghĩa hàm setUpRecaptcha (number) để thiết lập và sử dụng RecaptchaVerifier

function setUpRecaptcha(number) {

    const recaptchaVerifier = new RecaptchaVerifier(

      "recaptcha-container",

      {},

      auth

    );

    recaptchaVerifier.render();

    return signInWithPhoneNumber(auth, number,recaptchaVerifier)

  }

- Sau khi RecaptchaVerifier được tạo, hàm gọi phương thức render() của RecaptchaVerifier để hiển thị Recaptcha trên phần tử có id là "recaptcha-container". Quá trình render này cho phép người dùng xác minh hình ảnh hoặc văn bản để chứng minh rằng họ không phải là robot.

- Tiếp theo, hàm gọi signInWithPhoneNumber(auth, number, recaptchaVerifier) để xác thực số điện thoại người dùng. Đối số auth là đối tượng xác thực của Firebase, number là số điện thoại cần xác thực và recaptchaVerifier là đối tượng RecaptchaVerifier đã được thiết lập trước đó.

- Cuối cùng sử dụng signInWithPhoneNumber từ firebase để xác thực người dùng bằng số điện thoại và recaptcha đã tạo.

4

- File App.js sẽ có nhiệm vụ nạp các thành phần được xuất ra trong component và cấu hình các đường dẫn của đồ án sau đó trả lại giao diện cho người dùng

- Trong folder components chứa các file cấu hình giao diện và các đường dẫn cho dự án được viết bằng JSX và dùng các thuộc tính có sẵn của bootrap.

- File home.js là cấu hình cho trang chủ của dự án khi người dùng đăng nhập thành công thì sẽ được điều hướng tới Home componet để xác nhận đã đăng nhập thành công.

- File Login.js là component thực hiện việc đăng nhập của Người dùng

- Component Login sẽ trả về giao diện trong đó có những ô input. Ta sẽ lấy giá trị mà người dùng nhập vào qua các state.

const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [error, setError] = useState("");

- Tiếp theo đó định nghĩa các hàm như HandleSubmit, HandleGoogleSignIn để xử lý đăng nhập và submit của người dùng.

- File PhoneSignUp.js xuất ra component PhoneSignUp

5

có tác dụng xử lý khi chúng ta dùng chức năng đăng nhập

bằng điện thoại.

- Cũng như component trên PhoneSignUp dùng các state để lấy giá trị mà người dùng nhập vào gồm số điện thoại và mã OTP

- Component này sử dụng 1 số đoạn code logic để khi người dùng nhấn vào send OTP sẽ hiện ra mã captcha, người dùng sau khi xác nhận mã captcha thì mới gửi được tin nhắn.

- Sau khi OTP được gửi thì component này có nhiệm vụ trả về ô input mã xác thực để người dùng nhập mã xác thực vào.

- Component này có tác dụng để lấy thông tin người dùng hiện tại. Nếu user là falsy (không tồn tại hoặc là giá trị false, null, undefined, 0, hoặc một chuỗi rỗng), nghĩa là người dùng chưa xác thực, đoạn mã sẽ chuyển hướng người dùng về trang "/" bằng cách sử dụng thành phần Navigate từ thư viện React Router.

- Nếu user tồn tại, tức là người dùng đã xác thực, đoạn mã sẽ trả về thành phần con (được truyền vào thông qua prop children) để hiển thị nội dung của route bảo vệ.

6

- Đoạn mã trên có chức năng đảm bảo rằng chỉ có người dùng đã xác thực mới được truy cập vào các thành phần con của route bảo vệ.

- Component Signup trong file Signup.js có tác dụng thực hiện chức năng đăng ký người dùng sử dụng các state để lấy giá trị mà người dùng nhập vào

- Khi người dùng nhấn nút "Sign up", sự kiện onSubmit được kích hoạt, và hàm handleSubmit được gọi. Trong hàm này, lỗi được xóa (setError("")), sau đó hàm signUp từ useUserAuth được gọi để thực hiện quá trình đăng ký với email và password đã nhập. Nếu quá trình đăng ký thành công, người dùng sẽ được chuyển hướng về trang chủ ("/") bằng cách sử dụng navigate("/"). Nếu có lỗi xảy ra trong quá trình đăng ký, lỗi sẽ được lưu vào state error và hiển thị lên giao diện.

- Tất cả sẽ được nạp vào file App.js. File App.js sẽ xuất ra compnent App chứa tất cả những component được định nghĩa và sẽ được render qua file index.js

- Mọi số điện thoại hoặc email mà người dùng nhập vào sẽ được lưu trữ ở https://console.firebase.google.com/u/0/project/otpdemo-8ccad/authentication/users

7

IV. Cách chạy đồ án

Đồ án được chạy trên môi trường Visual Studio Code

1.Kéo đồ án về bằng câu lệnh:

git clone <https://github.com/phamduc29012003/DoAnFinal.git>

2. Cài đặt các gói, thư viện cho dự án bằng câu lệnh

npm install

3. Chạy đồ án bằng câu lệnh

npm start

8

**TÀI LIỆU THAM KHẢO**

**Chủ yếu tham khảo trên các trang mạng Internet**

<https://firebase.google.com/docs/auth/web/firebaseui?hl=vi>

<https://firebase.google.com/docs/auth/web/manage-users?hl=vi>

<https://firebase.google.com/docs/auth/web/password-auth?hl=vi>

<https://firebase.google.com/docs/auth/web/phone-auth?hl=vi>

<https://firebase.google.com/docs/auth/web/anonymous-auth?hl=vi>

<https://firebase.google.com/docs/auth/web/multi-factor?hl=vi>

<https://react.dev/learn>

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://bkhost.vn/blog/captcha/#:~:text=Captcha%20l%C3%A0%20m%E1%BB%99t%20ph%C6%B0%C6%A1ng%20ph%C3%A1p,h%C3%ACnh%20th%E1%BB%A9c%20t%E1%BA%A5n%20c%C3%B4ng%20kh%C3%A1c>