Shape, square

Description automatically generated

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**

|  |  |
| --- | --- |
|  | |
|  |  |

A picture containing text

Description automatically generatedLogo

Description automatically generated

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**BẢO MẬT WEB**

**XÂY DỰNG WEBSITE KINH DOANH SÁCH**

**METIS’S BOOK**

**GVHD: Ths. Lê Thị Minh Châu**

**Nhóm sinh viên thực hiện :**

Hà Vĩ Khang 20110657

Nguyễn Đức Khải 20110655

Nguyễn Đức Thịnh 20110256

Lê Nguyễn Tuấn Kiệt 20110286

**Tp. Hồ Chí Minh, tháng 5 năm 2023**

***Nhận xét của giảng viên***

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

*Tháng 5 năm 2023*

**Giáo viên chấm điểm**

# LỜI CẢM ƠN

Lời nói đầu tiên, nhóm thực hiện xin được gửi đến cô **Lê Thị Minh Châu** – giảng viên bộ môn lời cảm ơn chân thành và sâu sắc nhất.

Nhóm thực hiện xin cảm ơn sự quan tâm và giúp đỡ tận tình của cô trong suốt quá trình giảng dạy. Cảm ơn cô đã luôn giải đáp những thắc mắc cũng như đưa ra những nhận xét, góp ý giúp nhóm thực hiện cải thiện chất lượng công việc của nhóm.

Vì khả năng còn hạn chế nên trong quá trình thực hiện báo cáo không tránh khỏi sai sót, kính mong nhận được những ý kiến đóng góp từ thầy để nhóm có thể cải thiện hơn sau này.

Nhóm thực hiện xin chân thành cảm ơn.

**MỤC LỤC**

# PHÂN CHIA CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | MSSV | Phân công | Mức độ hoàn thành |
| Hà Vĩ Khang | 20110657 | UC-4, UC-6, UC-8, UC-17 | 100% |
| Nguyễn Đức Khải | 20110655 | UC-1, UC-2, UC-3, UC-5, UC-7 | 100% |
| Lê Nguyễn Tuấn Kiệt | 20110286 | UC-9, UC-12, UC-14, UC-16 | 100% |
| Nguyễn Đức Thịnh | 20110256 | UC-10, UC-11, UC-13, UC-15 | 100% |

# Phần 1: Giới thiệu đề tài

## 1. Giới thiệu

### *1.1. Đối tượng của đề tài*

- Khách hàng

- Quản lý

- Nhân viên

### 1.*2.Chức năng*

#### 1.2.1. Khách hàng

* Đăng ký tài khoản: Người dùng tiến hành đăng ký tài khoản, thông tin cần cung cấp để đăng ký bao gồm:
  + - Thông tin bắt buộc: tên đăng nhập, mật khẩu, tên người dùng, gmail, số điện thoại.
    - Thông tin không bắt buộc: Địa chỉ, giới tính, ngày tháng năm sinh.
* Đăng nhập tài khoản: Người dùng nhập tên tài khoản và mật khẩu để đăng nhập vào tài khoản.
* Tìm kiếm sản phẩm: Người dùng có thể tìm kiếm các loại sách dựa vào tên sách, tên tác giả, thể loại bằng khung tìm kiếm và lọc sách theo các tiêu chí.
* Cập nhập giỏ hàng: Người dùng có thể thêm, xóa, sửa các sản phẩm trong giỏ hàng.
* Mua hàng: Người dùng có thể chọn các sản phẩm trong giỏ hàng và hoàn thành các thông tin trong đơn hàng để mua hàng.
* Lựa chọn phương thức thanh toán: Người dùng có thể thanh toán đơn hàng qua các phương thức thanh toán: Card(ATM, MoMo, Zalopay,…), thanh toán bằng tiền mặt.
* Lựa chọn phương thức vận chuyển: Người dùng có thể lựa chọn các phương thức vận chuyển như là vận chuyển hỏa tốc, vận chuyển nhanh, vận chuyển thường.
* Đánh giá sản phẩm: Người dùng sau khi trải nghiệm sản phẩm có thể để lại đánh giá đóng góp ý kiến về sản phẩm mình đã mua.
* chia sẻ sản phẩm: Người dùng có thể review giới thiệu sản phẩm thông qua việc chia sẻ đường dẫn liên kết của sản phẩm qua các nền tảng xã hội như là: Facebook, Zalo, Twitter.
* Gửi yêu cầu sản phẩm: Người dùng có thể gửi yêu cầu về loại sách chưa có thông qua tên sách và tên tác giả.
* Thay đổi thông tin tài khoản: Người dùng có thể thay đổi thông tin tài khoản cá nhân.

#### 1.2.2. Nhân viên

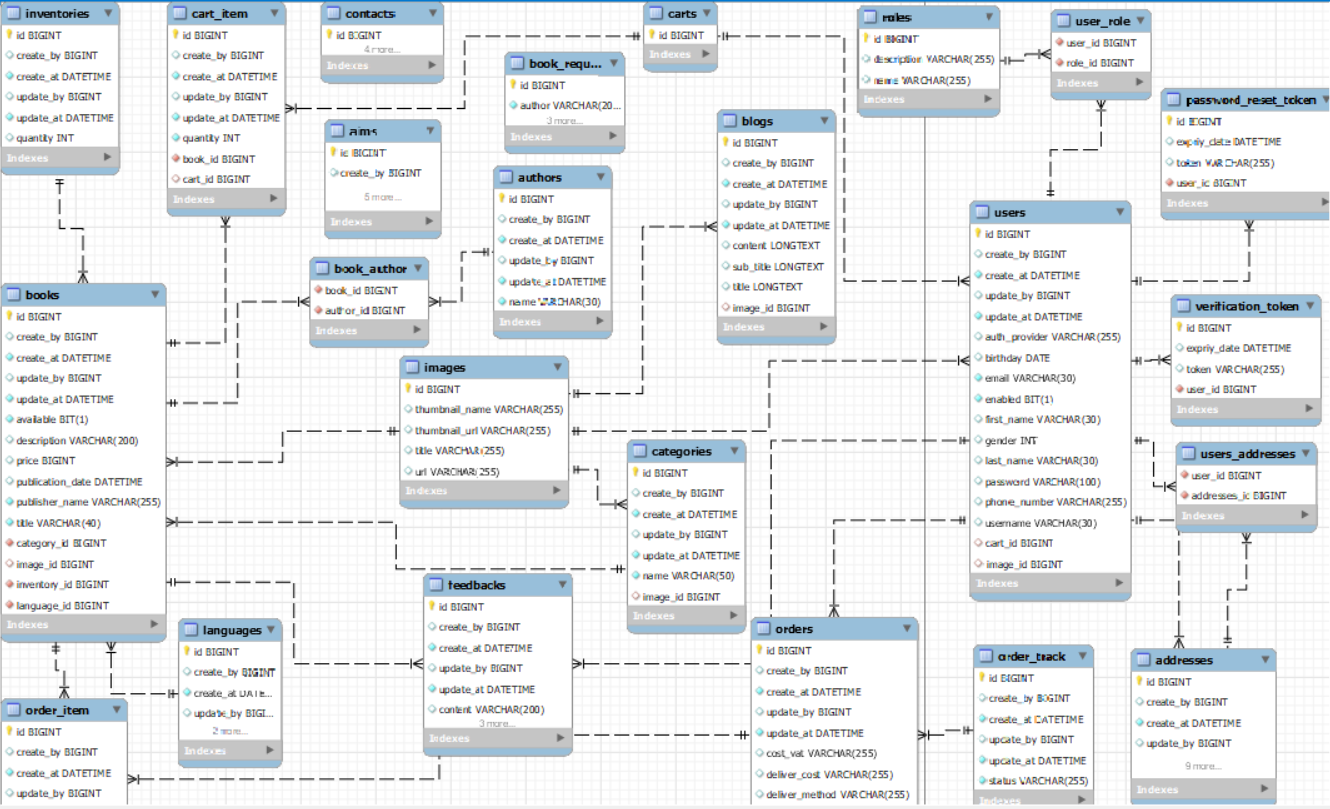
* Quản lý đơn hàng: Nhân viên có thể tiếp nhận đơn hàng, sửa đổi, xóa đơn hàng, chuẩn bị hàng để gửi.
* Quản lý sản phẩm: Nhân viên có thể quản lý các thông tin sản phẩm theo yêu cầu của người quản lý, thêm số lượng sản phẩm, sửa đổi thông tin, xóa sản phẩm.
* Quản lý bài đăng: Nhân viên có thể viết bài, sửa bài, xóa bài và đăng bài lên các mục quảng bá của trang web cũng như các trang thông tin của cửa hàng.
* Phản hồi đánh giá: Nhân viên có thể thêm, xóa, sửa và phản hồi đánh giá của khách hàng.

#### 1.2.3. Quản lý

* Quản lý tài khoản khách hàng: Quản lý có thể xem, thêm, sửa, xóa các thông tin liên quan đến tài khoản khách hàng.
* Quản lý sách: Quản lý có thể xem và chỉnh sửa thông tin sách, xem yêu cầu sách.
* Quản lý tài khoản nhân viên: Quản lý có thể xem, thêm, sửa, xóa các thông tin liên quan đến tài khoản nhân viên.
* Phân tích đánh giá lịch sử mua hàng: Quản lý có thể xem các thống kê về lịch sử mua hàng của một khách hàng cụ thể, lịch sử mua hàng của tổng thể, thu nhập, lịch sử hoạt động của nhân viên.

## Diagram Description automatically generated2. Lược đồ usecase

## 3. Lược đồ quan hệ ERD



## 4. Hướng dẫn cài đặt và chạy

### 4.1. Sơ lược về project

- Link project: <https://github.com/NoahSavant/bookstore_MetisBook.git>

- Ngôn ngữ lập trình: Java

- Framework: Spring boot

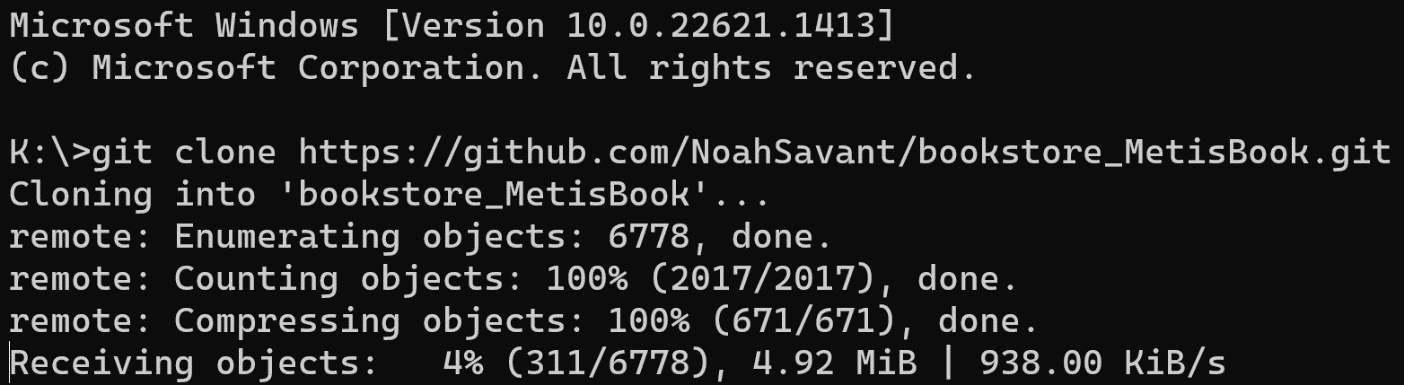
- Cơ sở dữ liệu: MySQL

- Version control: GitHub

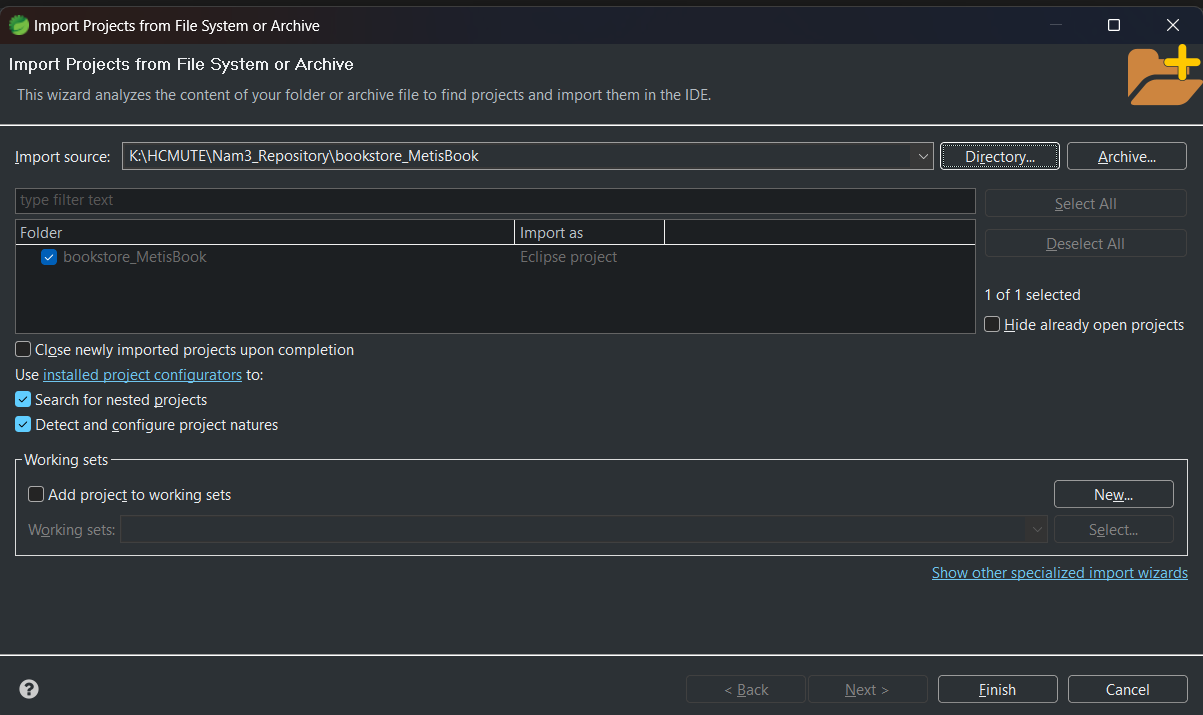
- IDE: Spring tool suite 4.0 for Eclipse, Eclipse, Intelliji

### 4.2. Hướng dẫn cài đặt

**Bước 1:** Tải project từ github về bằng câu lệnh:

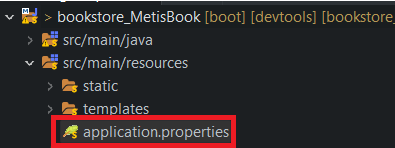
git clone <https://github.com/NoahSavant/bookstore_MetisBook.git>

**Bước 2:** Import project vào IDE

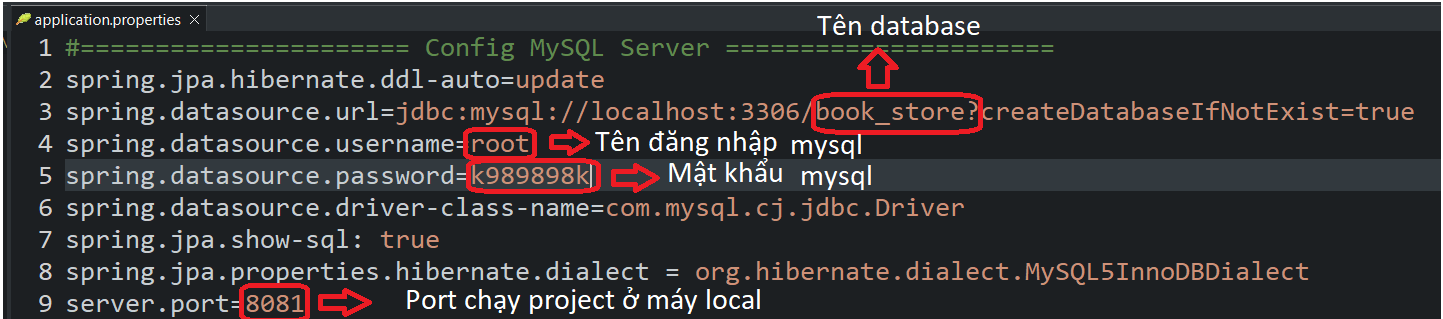
Mở 1 IDE bất kỳ hỗ trợ java ( Nên dùng STS cho eclipse ). Chọn file -> Open Project From file System -> Directory -> Chọn project -> Finish

### 4.3. Hướng dẫn chạy

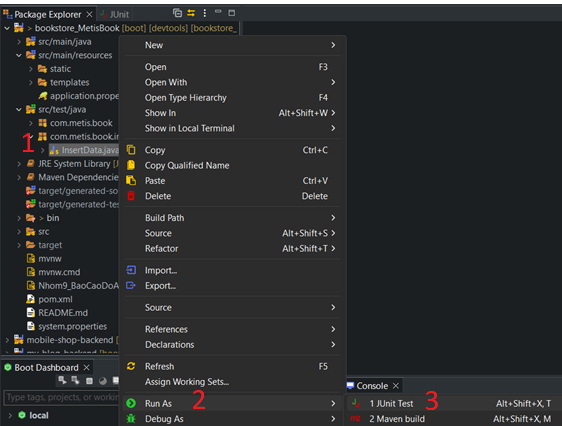
**Bước 1**: Sau khi tải project, mở file application.properties trong đường dẫn bookstore\_MetisBook/src/main/resources/application.properties



**Bước 2**: Thay đổi cấu hình tùy chỉnh trong application.properties



**Bước 3:** Nháy chuột phải vào file Insert Data nằm trong thư mục src/test/java trong package com.metis.book.insertData. Sau đó chọn Run as và chọn Junit test

****

Sau khi chạy file InsertData, database và dữ liệu mẫu sẽ được tự động thêm vào database được cài đặt trong file application.properties

# Phần 2: Bảo mật website – Kiểm tra thủ công

## Hidden Field Manipulation

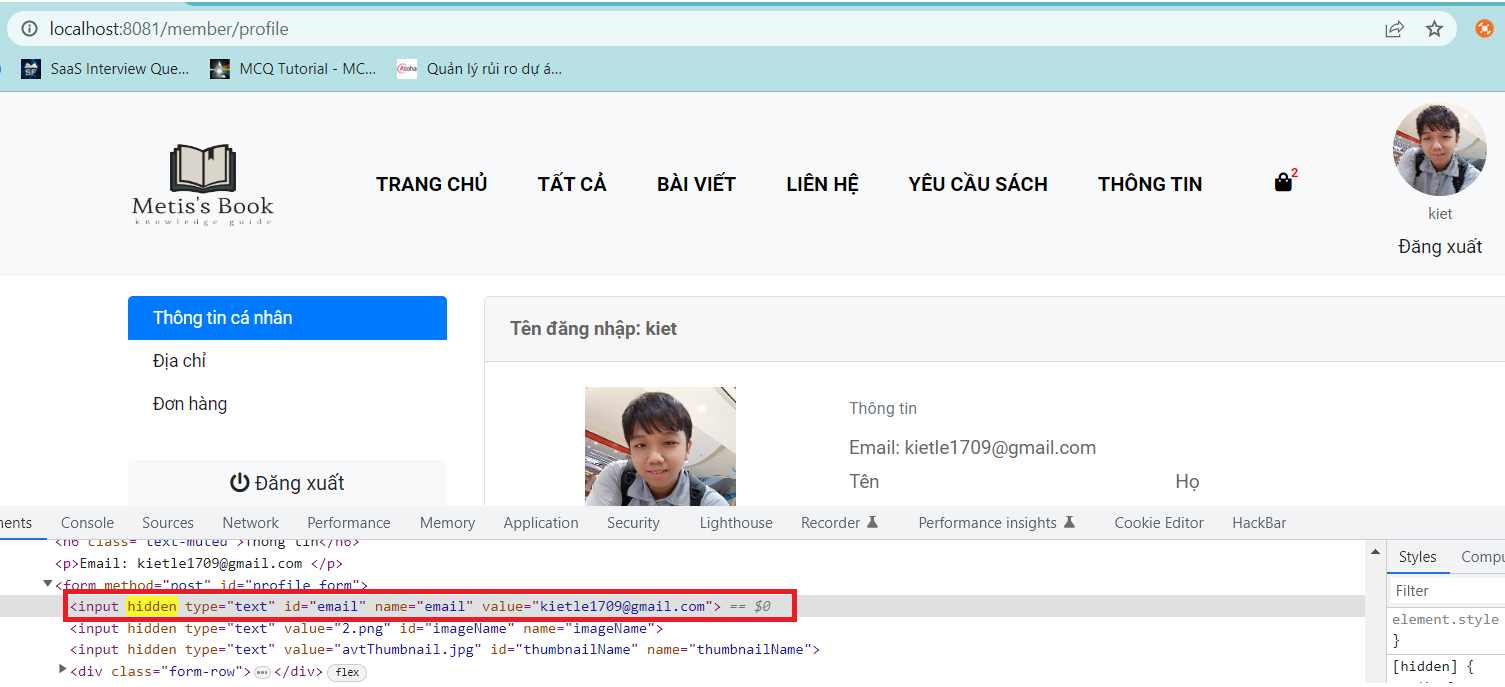
### Khái niệm

Là thể loại tấn công web trong đó lợi dụng các ô input được ẩn đi bởi các lập trình viên. Các ô input này thường chứa giá trị như: id, session id,... Hacker sẽ sử thay đổi giá trị các ô input này để vượt qua cơ chế bảo mật của website và gửi yêu cầu đến phía server. Nếu server kiểm tra dữ liệu lỏng lẻo, không có cơ chế xác thực người dùng. Hacker có thể thay đổi được các dữ liệu quan trọng

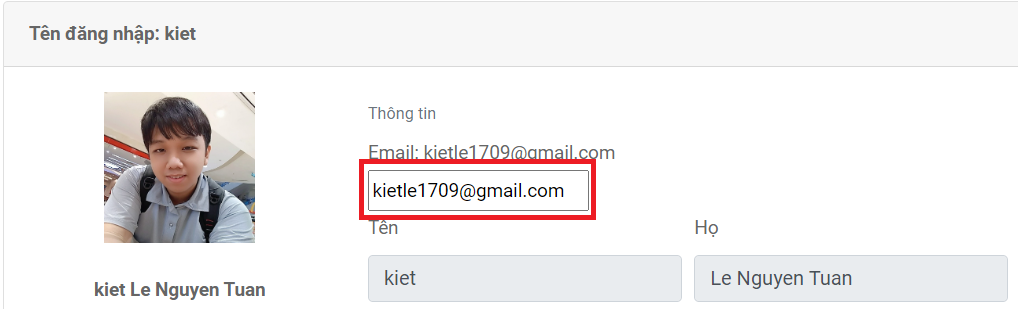
### Tấn công project

Trong project, tại trang quản lý thông tin cá nhân, người dùng có thể tùy ý chỉnh sửa thông tin tài khoản của mình tại form chỉnh sửa.

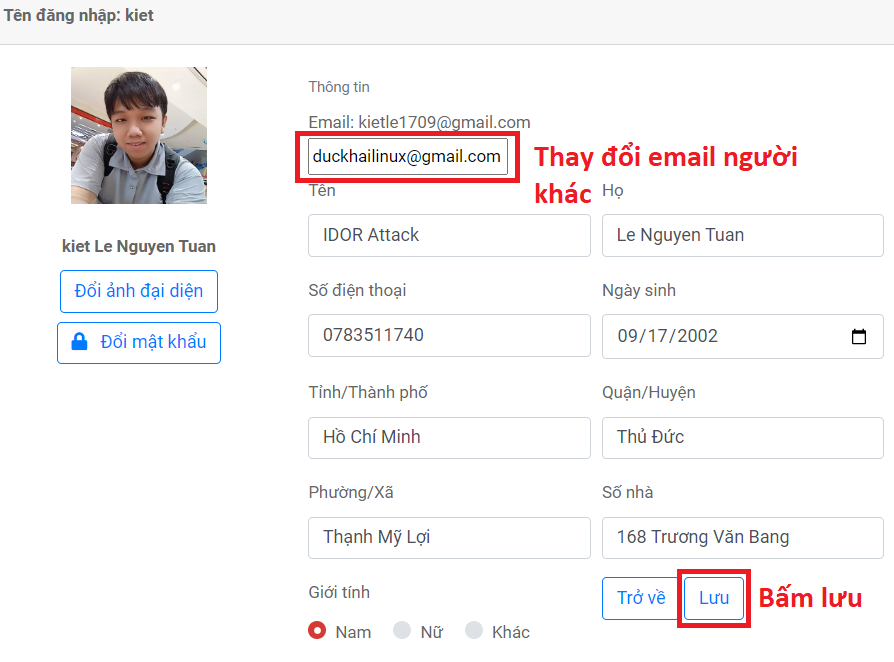
Tuy nhiên khi kiểm tra bằng F12, ta có thể nhận ra tồn tại thẻ input với thuộc tính là hidden, thẻ input này cho phép thay đổi email của yêu cầu chỉnh sửa



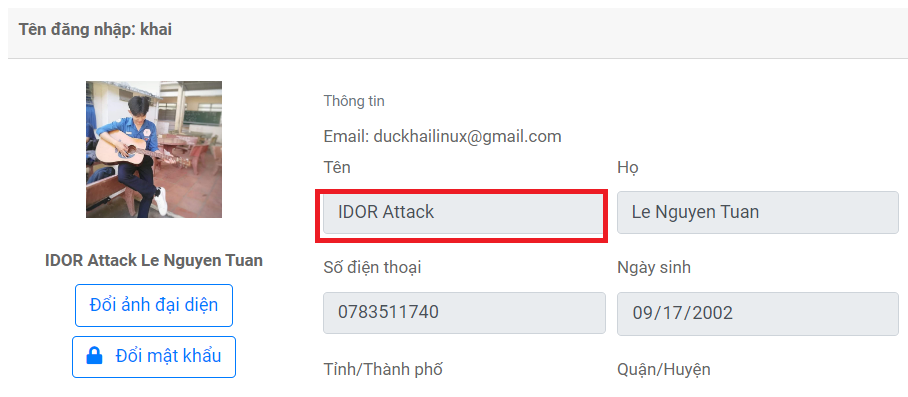
Khi ta loại bỏ thuộc tính “hidden”, 1 ô input hiện ra cho phép ta chỉnh sửa email:



Sau đó, thay đổi email ở ô hidden thành email của người dùng khác, sau đó chỉnh sửa thông tin mình mong muốn ở form chỉnh và bấm nút “Lưu”



Ta kiểm tra bằng cách đăng nhập vào tài khoản nạn nhân ( [duckhailinux@gmail.com](mailto:duckhailinux@gmail.com)). Và thông tin của nạn nhân đã bị chỉnh sửa trái phép bởi người dùng khác



### Nguyên nhân

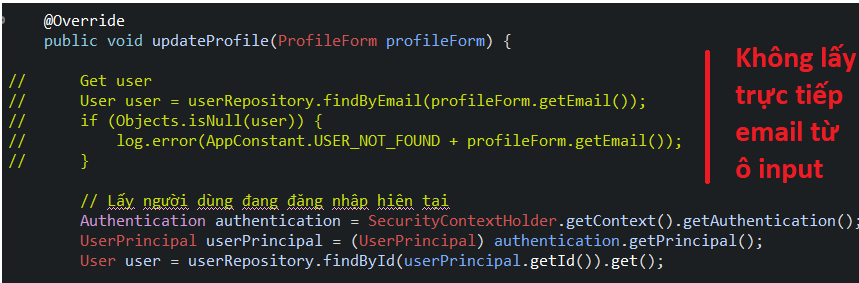
Khi xem code project ở phần chỉnh sửa thông tin cá nhân, ta thấy server không hề xác thực người dùng mà chỉ đơn thuần là nhận trục tiếp email trong ô input có thuộc tính “hidden” ở phía client làm đối tượng thay đổi thông tin.



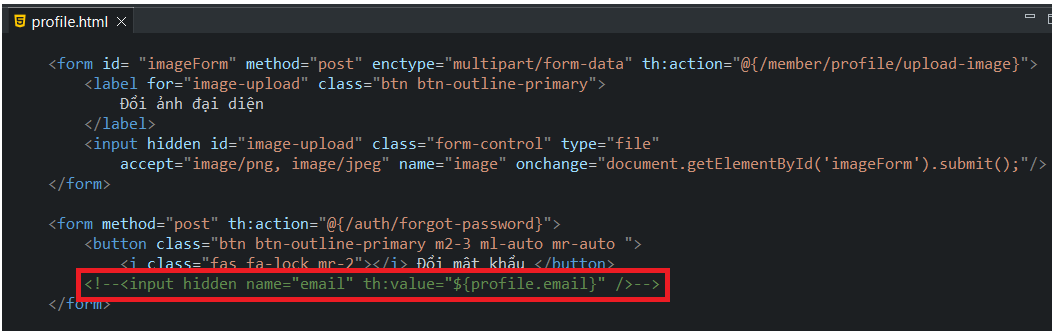
Vì vây, khi người dùng thay đổi email ở ô input, sẽ làm thay đổi đối tượng được yêu cầu chỉnh sửa thông tin.

### Khắc phục

Để khắc phục lỗ hổng này, ta sẽ tiến hành xác thực người dùng ở phía server và không sử dụng trực tiếp giá trị “email” từ ô input hidden nữa:



Đồng thời loại bỏ đi ô input hidden:



Nguyên nhân nằm ở cơ chế phân quyền, vì vậy việc phát triển ứng dụng tuân thủ chặt chẽ việc phân quyền và kiểm tra quyền truy cập tới dữ liệu là điều quan trọng nhất. Luôn kiểm tra quyền truy cập của mỗi người dùng ở tất cả các request. Việc kiểm tra phân quyền luôn luôn cần thực hiện trên phía server, không kiểm tra ở phía client. Đồng thời không nên sử dụng trực tiếp các giá trị quan trọng từ phía client mà phải qua cơ chế xác thực

## Insecure Direct Object Reference ( IDOR )

### Khái niệm

IDOR là cụm từ viết tắt của Insecure Direct Object Reference (Tham chiếu đối tượng trực tiếp không an toàn). Lỗ hổng này nằm trong TOP 10 lỗ hổng nguy hiểm mà OWASP đã đưa ra IDOR. Lỗ hổng này xảy ra khi chương trình cho phép người dùng truy cập tài nguyên (dữ liệu, file, thư mục, database..) một cách trực tiếp thông qua dữ liệu do người dùng cung cấp nhưng kém an toàn.

Nguyên nhân sâu xa của lỗ hổng IDOR nằm ở cơ chế phân quyền và kiểm soát truy cập người dùng trên website. Việc website không phân quyền đúng cũng như kiểm soát chặt chẽ việc truy cập tài nguyên của user trên hệ thống dẫn đến việc truy cập trái phép từ kẻ tấn công.

IDOR không chỉ cho phép truy cập thông tin trái phép của người dùng khác mà còn có thể cho phép kẻ tấn công thực hiện thay đổi hoặc xóa dữ liệu của người dùng khác. Vì vậy lỗ hổng này có mức độ nguy hiểm rất cao, gây hậu quả nghiêm trọng tới người dùng. Đặc biệt là các hệ thống có dữ liệu quan trọng của người dùng như: mạng xã hội, tài chính - ngân hàng, y tế..

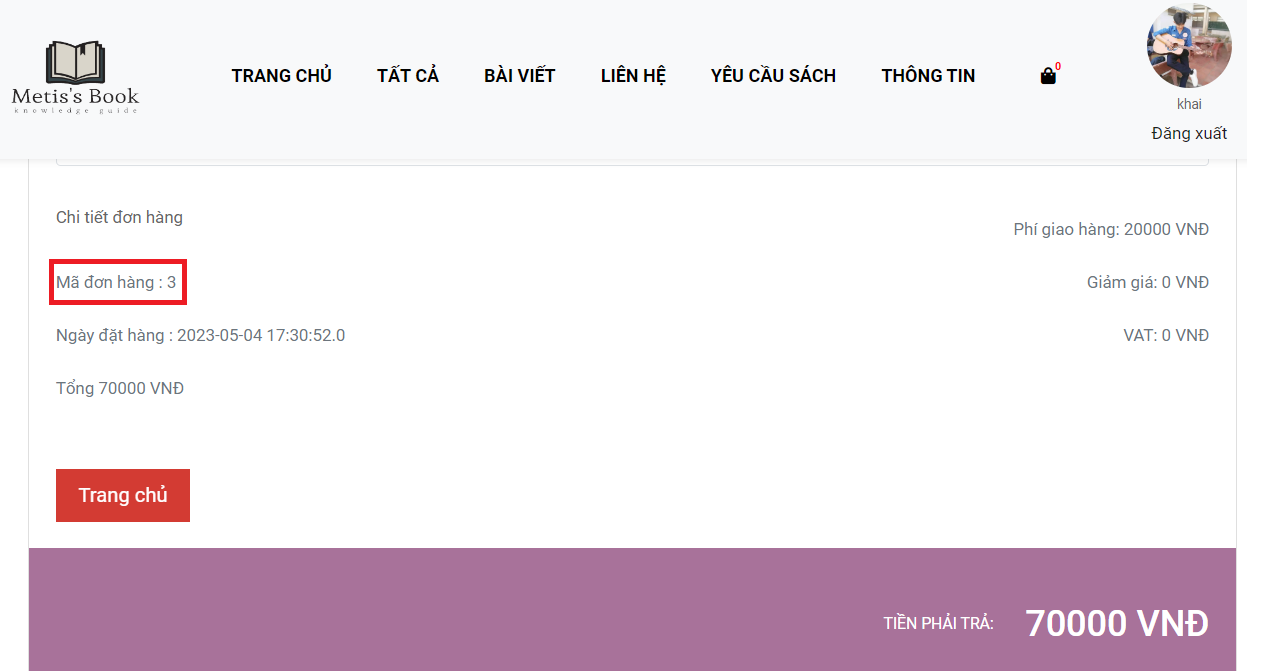
### Tấn công project

Trong project, tại trang xem thông tin hóa đơn, để ý trên url người dùng sẽ thấy có thể tìm kiếm hóa đơn theo id bằng cách thay đổi trực tiếp giá trị “orderId” :

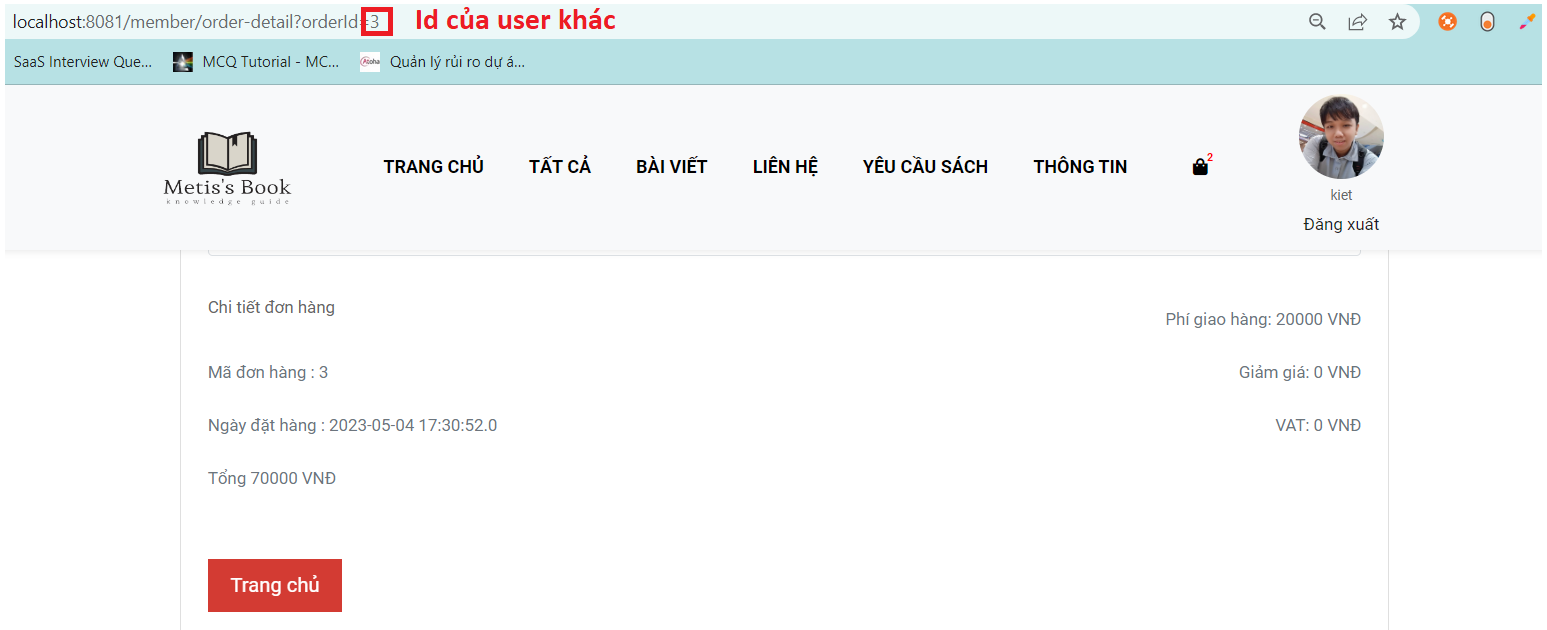


Tuy nhiên vấn đề xảy ra khi người dùng thay đổi giá trị “orderId” không phải của mình.

Lấy ví dụ tài khoản của người dùng với email: [duckhailinux@gmail.com](mailto:duckhailinux@gmail.com). Có 1 hóa đơn với id là 3:



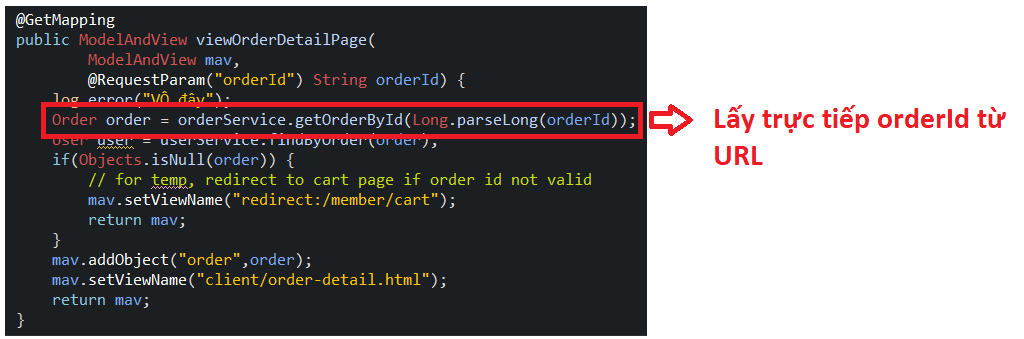
Người dùng với email [kietle1709@gmail.com](mailto:kietle1709@gmail.com). Khi thay đổi giá trị “orderId” trên url có thể truy cập và xem thông tin hóa đơn với id là 3:



Như vậy, các thông tin hóa đơn nhạy cảm có thể bị đọc bởi người dùng khác.

### Nguyên nhân

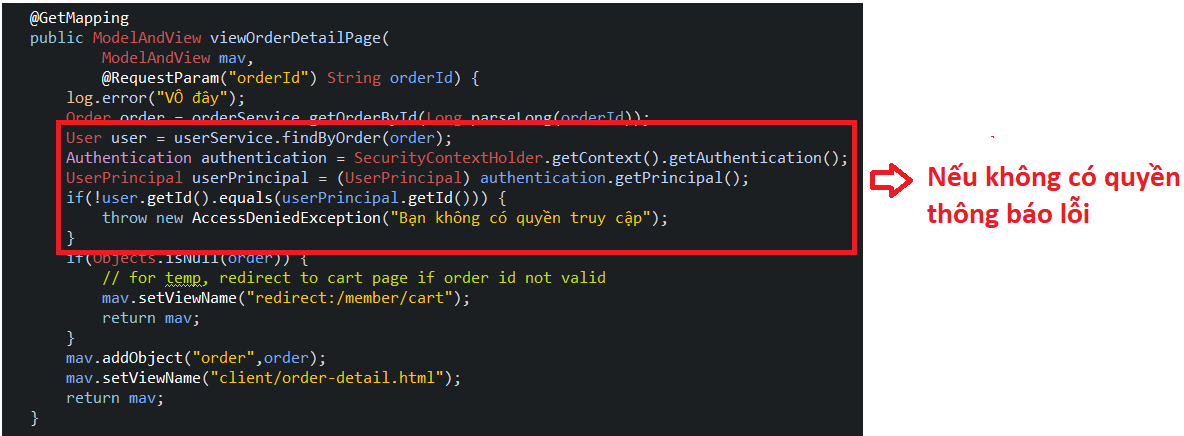
Nguyên nhân đến từ phía server đã không tiến hành xác thực người dùng trước khi cho phép truy cập vào thông tin hóa đơn. Xem lại code, ta thấy:



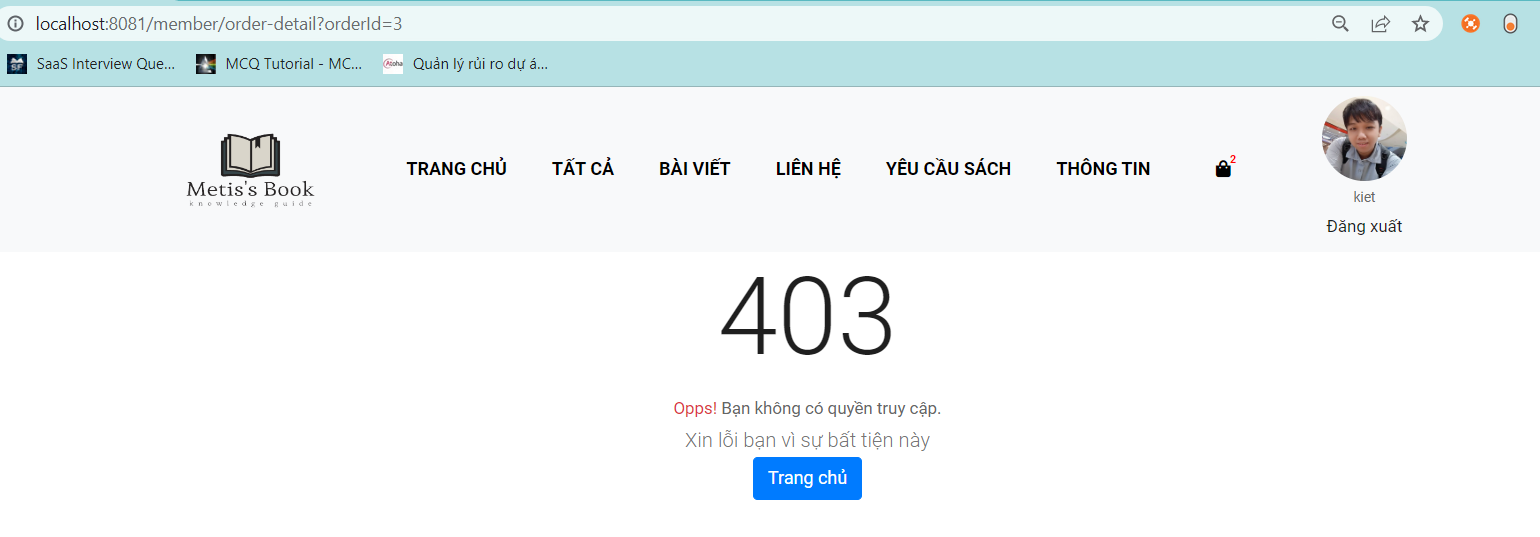
Việc giá trị orderId được lấy trực tiếp từ đường dẫn mà không qua bất kỳ cơ chế xác thực nào, đã khiến cho thông tin hóa đơn có thể dễ dàng bị truy cập bởi người dùng trái phép

### Khắc phục

Vì nguyên nhân bắt nguồn từ việc không xác thực người dùng ở phía server. Nên để khắc phục, ta sẽ tiễn hành xác thực người dùng:



Khi người dùng cố ý thay đổi orderId để truy cập vào thông tin hóa đơn người khác, website sẽ dẫn đến màn hình lỗi:



IDOR là một lỗ hổng nguy hiểm có thể đe dọa quyền riêng tư của người dùng và tính toàn vẹn của ứng dụng của bạn. Việc tuân thủ các cơ chế phân quyền chặt chẽ và kiểm tra ứng dụng kĩ sẽ giúp các bạn hạn chế được nguy cơ về lỗ hổng bảo mật IDOR.

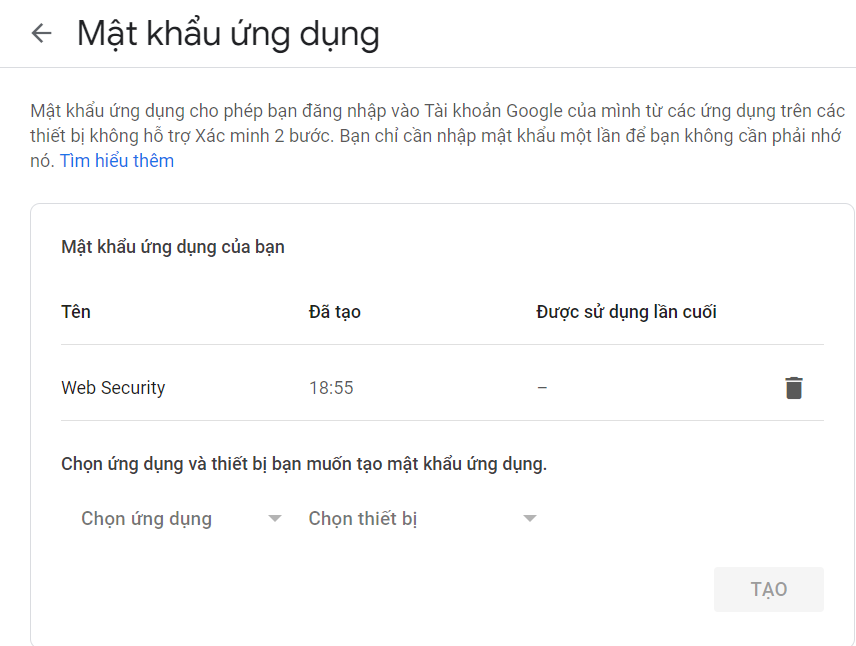
## SMTP credentials exposed

### Khái niệm

SMTP credentials exposed (hay còn gọi là "lộ thông tin đăng nhập SMTP") là khi thông tin đăng nhập cho giao thức SMTP (Simple Mail Transfer Protocol) bị tiết lộ hoặc bị phát hiện bởi người không được ủy quyền hoặc bởi một bên thứ ba không đáng tin cậy. Thông tin đăng nhập này bao gồm Id và secretId, được sử dụng để gửi email từ địa chỉ email đó.

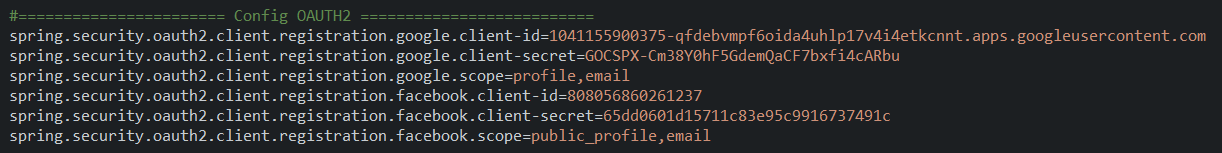
Khi SMTP credentials exposed, kẻ tấn công có thể thực hiện các hành động gửi email giả mạo, lừa đảo hoặc tấn công khác để gây hại cho người dùng hoặc các liên hệ của họ. Vì vậy, việc bảo vệ thông tin đăng nhập SMTP là rất quan trọng để bảo vệ tài khoản email của người dùng.

Truy cập vào google account, ta có thể tạo một cặp mật khẩu cho SMTP, cặp mật khẩu này dùng để tự động gửi mail trong cái ứng dụng, ví dụ như thư viện Java Mail Sender.

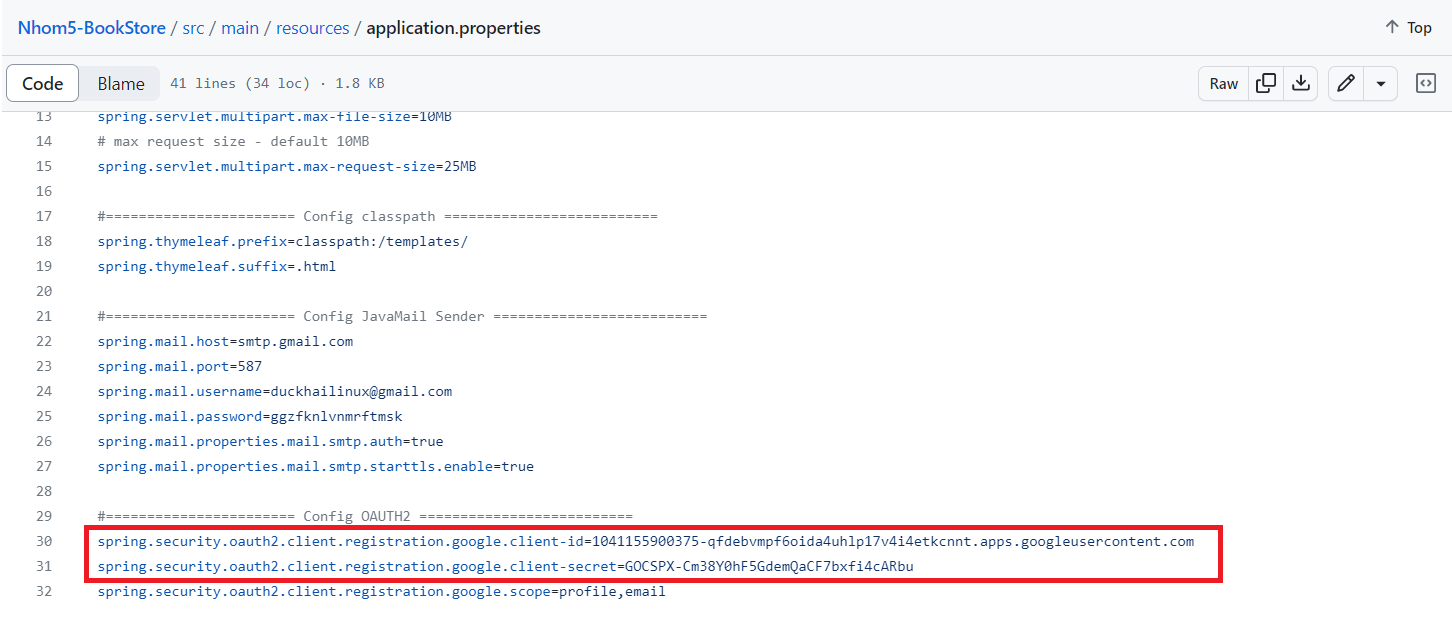


### Lỗ hổng trong project

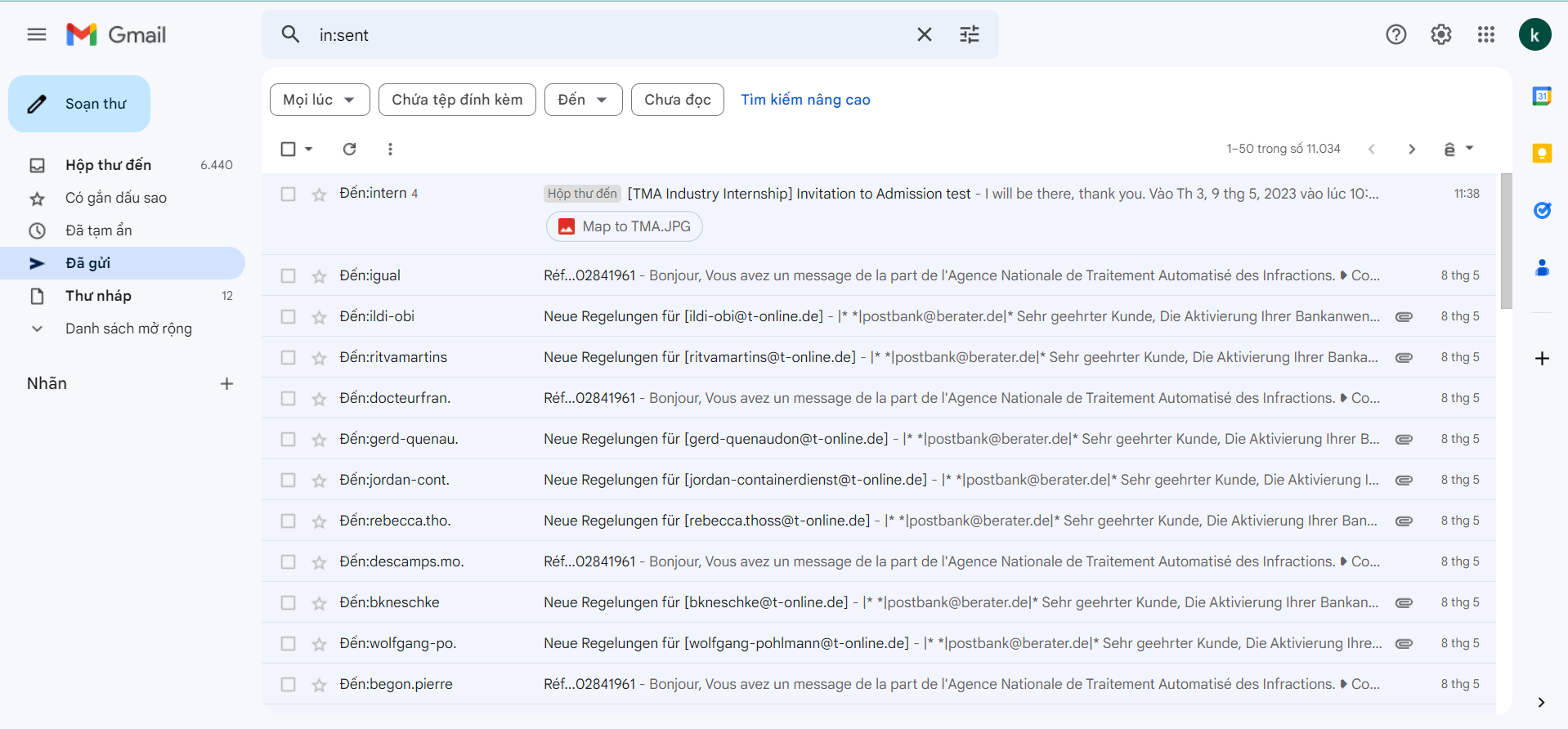
Trong project, tại file cài đặt, nhóm lập trình đã cấu hình tài khoản và mật khẩu của SMTP phục vụ cho quá trình tự động gửi mail của website ( xác nhận tài khoản, lấy lại mật khẩu ), trong file application.properties:



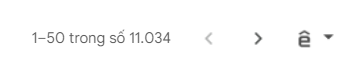
Tuy nhiên, nhóm lập trình sử dụng github làm nơi lưu trữ mã nguồn mà không che giấu đi thông tin quan trọng này. Khiến lộ thông tin trên github:



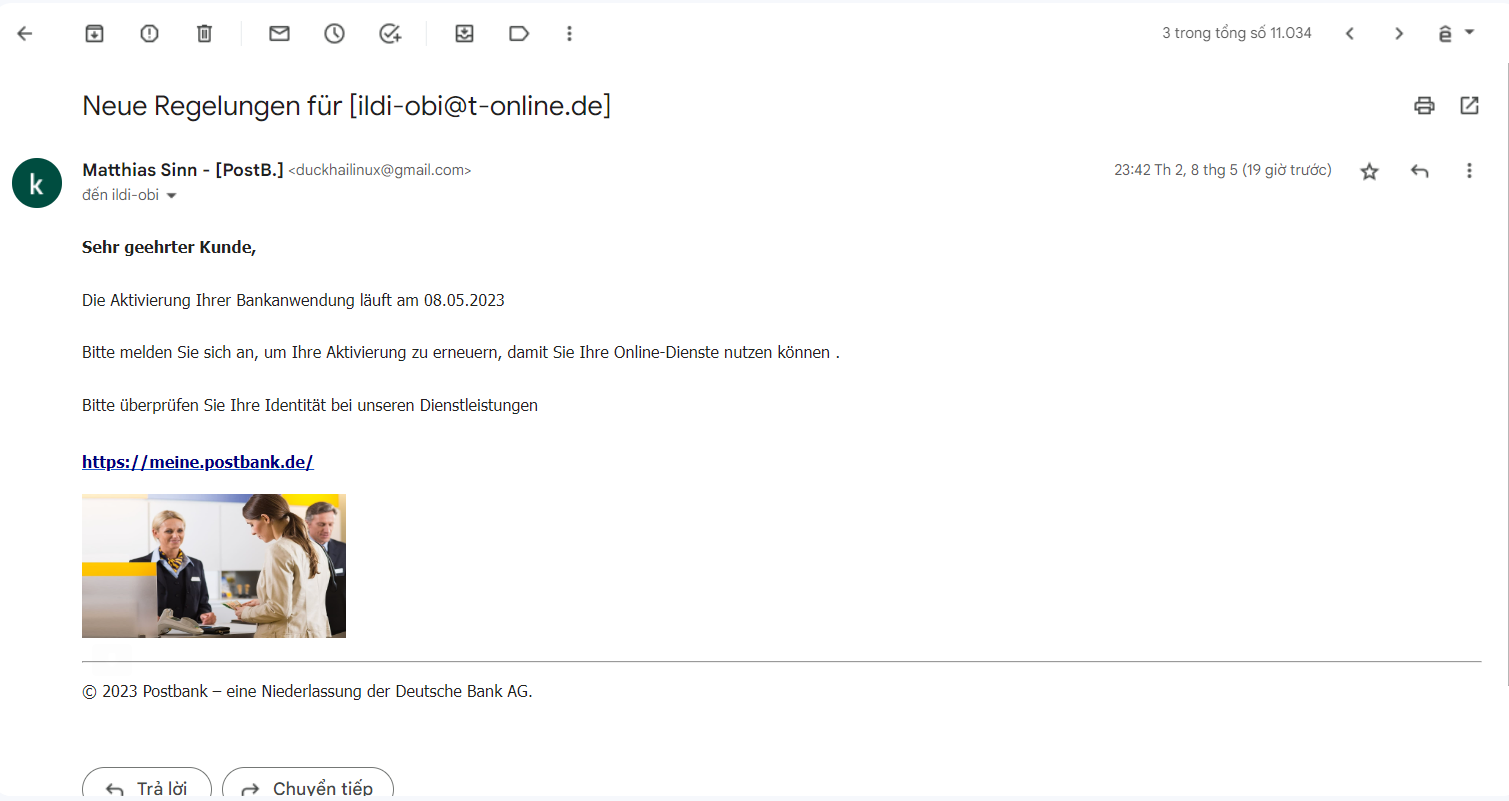
Hacker có thể lợi dụng và sử dụng SMTP để làm những hành động xấu. Điển hình là với email của bạn Khải: [duckhailinux@gmail.com](mailto:duckhailinux@gmail.com). Hacker đã sử dụng tài khoản SMTP này để gửi mail spam cho nhiều người:



Có thể thấy trong suốt quá trình bị lộ thông tin, hacker đã gửi email spam tổng cộng gần 11000 lần:



Nội dung email chủ yếu là tin nhắn spam đại diện cho 1 ngân hàng nào đó ở nước Đức:



Có thể thấy, việc làm lộ thông tin SMTP đã mang lại rất nhiều phiền toái, chưa kể, trong số email được gửi đi, có thể có email lừa đảo ( Tuy nhiên rất may mắn là việc đó không xảy ra ).

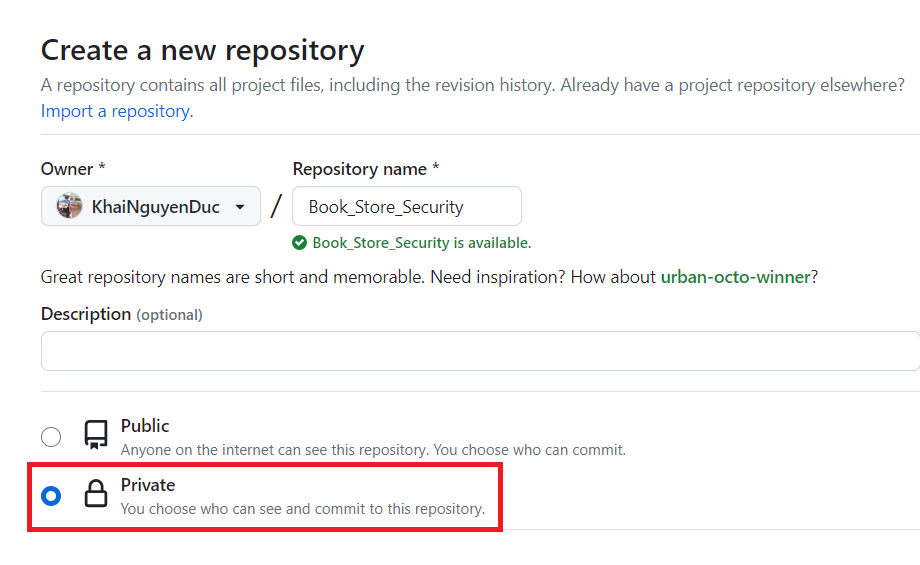
### Nguyên nhân

Nguyên nhân chủ yếu từ việc không cẩn thận che giấu thông tin nhạy cảm trong quá trình công khai mã nguồn. Github là nơi lưu trữ mã nguồn mở, và nhóm đã không cẩn thận khi upload code lên đây.

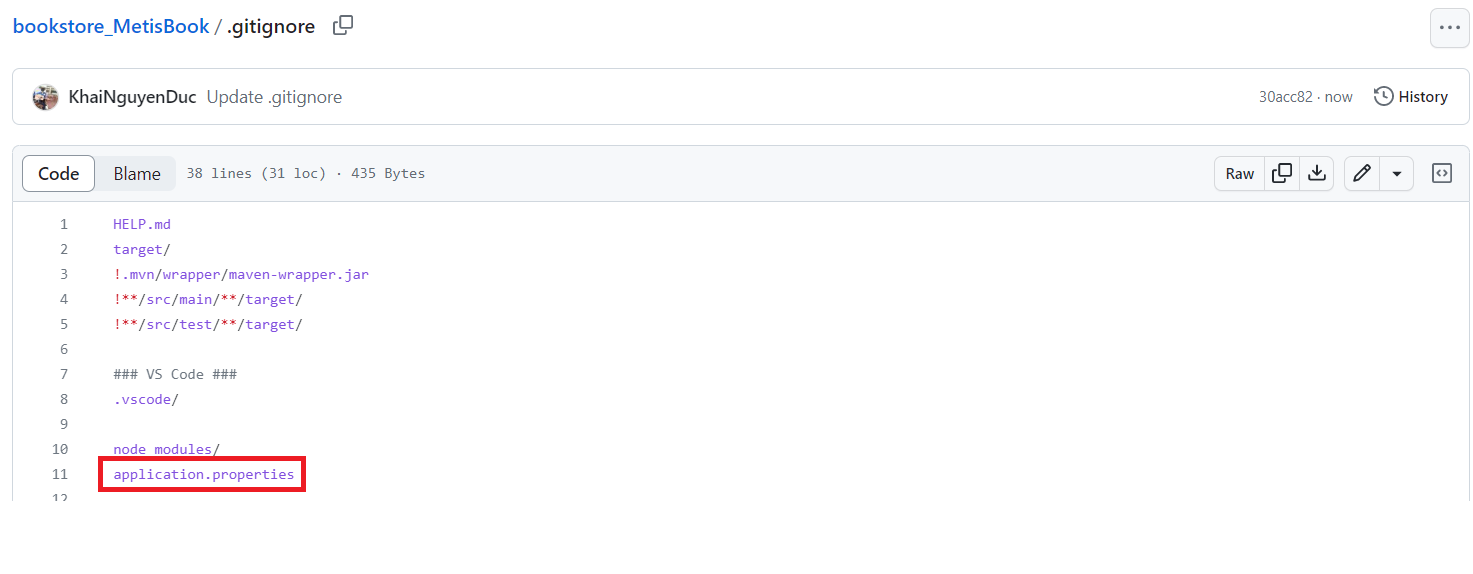
### Khắc phục

Có nhiều cách để khắc phục lỗ hổng này.

**Cách 1**: Sử dụng github repository private. Github cung cấp mỗi tài khoản ( không thuộc loại vip ) một nơi lưu trữ mã nguồn riêng tư. Khi phát triển những dự án quan trọng, ta nên đưa code lên nơi lưu trữ riêng tư để tránh người khác xem code và đánh cắp dữ liệu quan trọng.



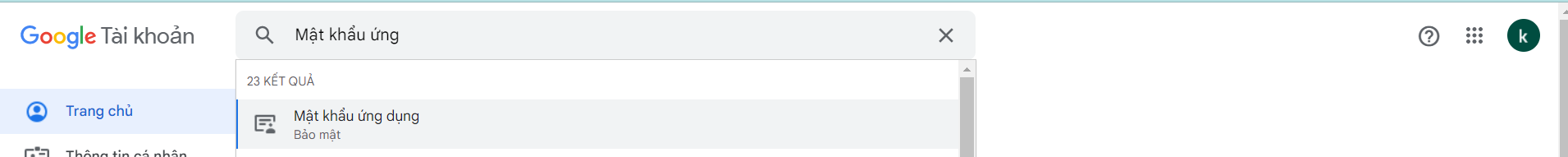
**Cách 2:** Sử dụng file .gitignore. Trong trường hợp ta cần công khai code trên github. Hãy thêm tên những file quan trọng trong file .gitignore:



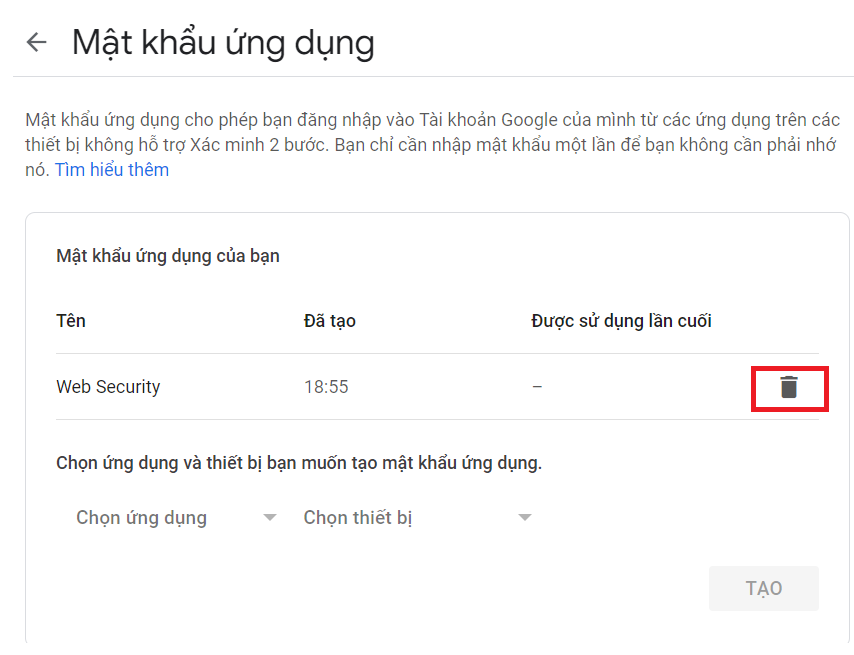
Khi đưa code lên github, những file được liệt kê trong .gitignore sẽ được ẩn đi và không được đưa lên cùng mã nguồn. Điều này khiến mã SMTP không còn được public ra nữa.

Dù sử dụng cách nào đi nữa, một khi đã bị lộ mã SMTP, ta phải lập tức xóa mật khẩu ứng dụng tại trang <https://www.google.com/account/about/>. Điều này làm cho mã SMTP hiện tại không còn sử dụng được nữa.

Tìm kiếm mật khẩu ứng dụng:



Nhấn vào biểu tương sau để xóa:



Như vậy, nhóm thực hiện đã thành công khắc phục được lỗ hổng này, và rút được kinh nghiệm quý báu về tầm quan trọng của bảo mật thông tin

## Brute Force Attack ( Admin Password )

# Phần 2: Bảo mật website – Quét bằng ZAP