OWASP

Yêu cầu tổng thể của đồ án:

* Xây dựng web → kịch bản tấn công → giải pháp cho cuộc tấn công(fix lại web đó)
* Xây dựng web để demo cho cuộc tấn công → tấn công → fix lại web

OWASP A05 (Security Misconfiguration)

* Đây là loại lỗ hổng:
  + **Missing Security Hardening**: Đây là việc thiếu các biện pháp bảo mật cần thiết trên các thành phần khác nhau của ứng dụng, bao gồm máy chủ, cơ sở dữ liệu, v.v.
    - Ví dụ dùng http thay vì https
  + **Unnecessary Features Enabled**: Việc có các cổng, dịch vụ, hoặc tính năng không cần thiết kích hoạt có thể tạo ra điểm yếu trong bảo mật ứng dụng
  + **Default Accounts and Passwords**:
  + **Error Handling Revealing Sensitive Information**:
    - Ví dụ thực tế: thông báo lỗi chi tiết thay vì chỉ hiển thị một thông báo lỗi tổng quát (như "Có lỗi xảy ra, vui lòng thử lại sau"), ứng dụng trả về thông báo chi tiết
  + **Insecure Security Features in Upgraded Systems**: Trong quá trình nâng cấp ứng dụng, các tính năng bảo mật có thể bị tắt hoặc không được cấu hình đúng cách
  + **Insecure Security Settings in Components**: Các hệ thống thành phần không được thiết lập bảo mật đúng cách, mở ra lỗ hổng cho kẻ tấn công khai thác
    - Ví dụ: Cơ sở dữ liệu không thiết lập quyền truy cập, Cơ sở dữ liệu như MySQL hoặc PostgreSQL được cài đặt nhưng không CVE-2015-9251 cấu hình đúng quyền truy cập, cho phép bất kỳ người dùng nào truy cập toàn bộ dữ liệu
  + **Missing or Insecure Security Headers**: Các tiêu đề bảo mật (như X-Frame-Options, Content-Security-Policy) không được cấu hình chính xác, giúp kẻ tấn công khai thác qua các phương thức như clickjacking.
  + **Outdated or Vulnerable Software**: Các thành phần phần mềm đã biết có lỗ hổng do không được cập nhật hoặc vá lỗi thường tạo ra rủi ro bảo mật cao
* Ý tưởng cho demo A05: **Missing or Insecure Security Headers**
  + Kịch bản cho cuộc tấn công: thực hiện bằng phương pháp Clickjacking
    - Tạo ra trang độc hại và nhúng vào trang web vào:
      * 1 website của hacker có link same same trang của mình
      * Nhúng iframe vào trang web lừa đảo
      * Tạo ra một nút bấm giả mạo như **nhận thưởng** → Người dùng đang thao tác trên trang thật nhưng không hề biết.
  + Giải pháp: Biện pháp: X-Frame-Options: DENY ngăn không cho bất kỳ trang nào nhúng trang web(Dùng node.js)

OWASP A06 (Vulnerable and Outdated Components)

* Đây là loại lỗ hổng:Lỗi này xảy ra khi bạn sử dụng các phần mềm, thư viện hoặc module đã lỗi thời và chứa lỗ hổng bảo mật mà chưa được vá lỗi.
* Ý tưởng cho demo A06:
  + Kịch bản cho cuộc tấn công:
    - JavaScript cũ, jQuery phiên bản 1.12.4,lỗ hổng CVE-2015-9251
    - Chèn một đoạn mã javascript vào đoạn bình luận
    - Khi người dùng login vào và đọc vô phần bình luận, đoạn mã javascript này sẽ được chạy và kết quả là cookies của người dùng sẽ được gửi đến máy chủ của kẻ tấn công
  + Giải pháp: Cờ HttpOnly là một thuộc tính của cookie cho phép bảo vệ cookie khỏi việc bị truy cập thông qua JavaScript. Điều này giúp ngăn chặn việc đánh cắp cookie từ các cuộc tấn công XSS.

OWASP A07 (Identification and Authentication Failures)

* Đây là loại lỗ hổng: Lỗi này liên quan đến việc **xác thực người dùng** không an toàn, dẫn đến việc người dùng trái phép có thể truy cập vào hệ thống.
  + **Permits automated attacks such as credential stuffing:** Credential stuffing là một kỹ thuật tấn công mà kẻ tấn công sử dụng danh sách tên người dùng và mật khẩu hợp lệ để truy cập vào tài khoản của người dùng. Các tài khoản này thường bị tấn công khi người dùng sử dụng cùng một mật khẩu trên nhiều dịch vụ khác nhau
  + **Permits brute force or other automated attacks:** Brute force attack là một kỹ thuật mà kẻ tấn công cố gắng tất cả các khả năng mật khẩu cho đến khi tìm thấy mật khẩu đúng. Điều này có thể diễn ra một cách tự động bằng các công cụ.
  + **Permits default, weak, or well-known passwords:** Việc cho phép sử dụng các mật khẩu mặc định, yếu hoặc phổ biến (như "Password1", "123456", hoặc "admin/admin") có thể dẫn đến nguy cơ cao về bảo mật.
  + **Uses weak or ineffective credential recovery and forgot-password processes:** Quy trình khôi phục mật khẩu kém hoặc không hiệu quả, như câu hỏi bảo mật (ví dụ: "Bạn bao nhiêu tuổi?") có thể bị đoán hoặc dễ dàng tìm thấy.
  + **Uses plain text, encrypted, or weakly hashed passwords data stores:** Lưu trữ mật khẩu dưới dạng văn bản thuần (plain text), hoặc sử dụng các phương pháp mã hóa yếu, hoặc hash không an toàn có thể khiến mật khẩu dễ dàng bị lộ.
  + **Has missing or ineffective multi-factor authentication:** Multi-factor authentication (MFA) yêu cầu người dùng cung cấp nhiều hình thức xác thực trước khi được truy cập vào tài khoản. Nếu không có hoặc thực hiện không hiệu quả, tài khoản sẽ dễ bị tổn thương.
  + **Exposes session identifier in the URL:** Nếu một ứng dụng bao gồm thông tin phiên (session identifier) trong URL, điều này có thể dẫn đến việc lộ thông tin phiên cho người khác thông qua lịch sử trình duyệt hoặc các bản ghi máy chủ (theo tôi hiểu thì nó như kiểu truy cập vào lịch sử tìm kiếm của người dùng)
  + **Reuse session identifier after successful login:** Nếu một phiên làm việc (session) không được làm mới sau khi người dùng đăng nhập thành công, điều này có thể khiến phiên đó dễ bị đánh cắp
  + **Does not correctly invalidate Session IDs:** Sau khi người dùng đăng xuất hoặc không hoạt động trong một khoảng thời gian nhất định, phiên làm việc của họ cần được hủy hiệu lực (invalidate)
* Ý tưởng cho demo A07 (có nhiều nhưng mình đề xuất cái này Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin" **vs** Has missing or ineffective multi-factor authentication) :
  + Kịch bản cuộc tấn công: Trang web cho phép người dùng thực hiện đăng ký tài khoản với mật khẩu đơn giản
  + Giải pháp:
    - Yêu cầu mật khẩu phức tạp (chữ in hoa, kí tự đặc biệt, số, ký tự đặc biệt)
    - Sử dụng Multi-factor authentication(xác thực nhiều yếu tố):
      * Xác thực qua sms
      * Cài đặt hệ thống:(này nhờ mn vote)

**Node.js**: Sử dụng thư viện speakeasy để tạo mã xác thực TOTP.

**Django**: Sử dụng django-otp

**Twilio**: để gửi tin nhắn đến người dùng thông mã totp