**Hướng dẫn sử dụng Screenshot Tool**

### ****Công nghệ & Công cụ sử dụng****

#### ****Frontend****

#### **- Framework:** Angular 19

#### -**Ngôn ngữ:** TypeScript

#### **- Style & UI:**

#### **+** Tailwind CSS

#### + notyf (thông báo popup)

#### - **Công cụ Build/Test:** Angular CLI

#### ****Backend****

#### **- Framework:** .NET 8

#### - ORM: Entity Framework Core

#### - **Ngôn ngữ:** C#

#### - So sánh hình ảnh**:** Emgu CV (wrapper của OpenCV cho .NET)

#### - **Database:** SQLite

#### - Chụp ảnh màn hình trang web**:** Microsoft Playwright

#### - Polly (retry policy, resilience)

#### - **API Documentation:** Swashbuckle (Swagger UI)

1. **Installation & Setup**

**a. Backend (.NET)**

- Yêu cầu:

+ Cài đặt **.NET 8**

**+** Hệ điều hành: **Windows** (sử dụng net8.0-windows)

- Cài đặt các gói NuGet:

+ Entity Framework Core (kết nối cơ sở dữ liệu):

dotnet add package Microsoft.EntityFrameworkCore.Sqlite

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Tools

+ Emgu CV (xử lý ảnh):

dotnet add package Emgu.CV

dotnet add package Emgu.CV.runtime.windows

**+ Polly (Retry, Resilience):**

**dotnet add package Polly**

**dotnet add package Polly.Core**

**dotnet add package Polly.Extensions**

**+ Swagger (OpenAPI / API Docs):**

**dotnet add package Swashbuckle.AspNetCore**

**- Cài đặt Playwright**

dotnet tool install --global Microsoft.Playwright.CLI

playwright install

- Chạy ứng dụng:

cd [ScreenShotTool](https://github.com/diepdau/ScreenShotTool/tree/main)/[ScreenShotDotnetWebApi](https://github.com/diepdau/ScreenShotTool/tree/main/ScreenShotDotnetWebApi)

dotnet build

dotnet run

**b. Frontend (Angular)**

- Yêu cầu cài đặt: NodeJS, NPM, Angular

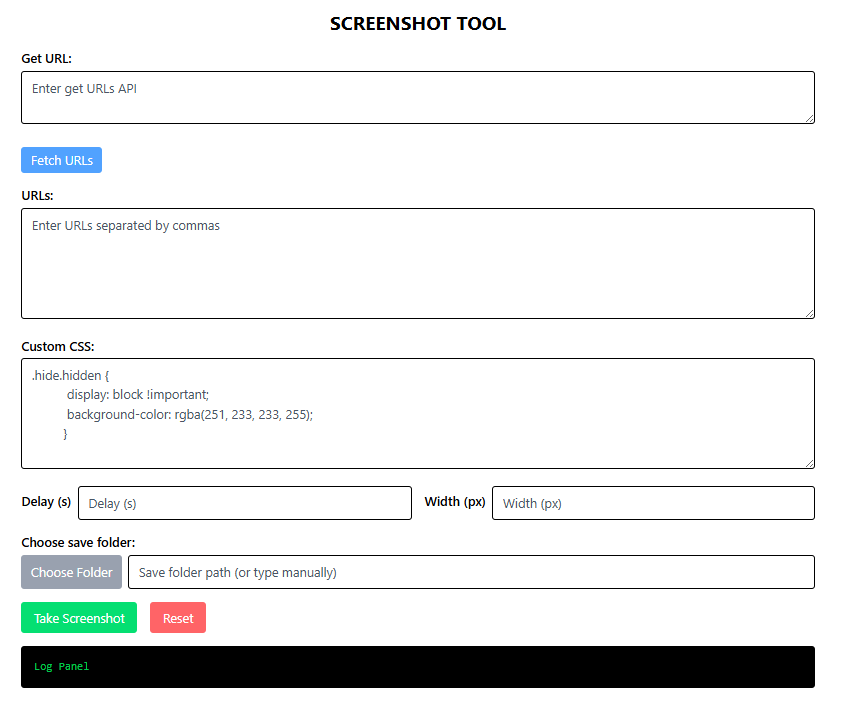
- Cài đặt và chạy:

cd [ScreenShotTool](https://github.com/diepdau/ScreenShotTool/tree/main)/ScreenShotToolWeb

npm install

ng serve

Mở trình duyệt: <http://localhost:4200>



### **3. Cách sử dụng**

#### **URLs (bắt buộc)**

#### **Cách 1 – Lấy URLs từ API**

Sử dụng API sau để lấy danh sách URLs

*https://tcma-api-public.dev.usms.impartner.io/showcase/urls?ProjectSlug=rv-portal-template&LanguageId=1&AccountId=147347*

#### **Cách 2 – Nhập URLs thủ công**

* Nhập các URLs **cách nhau bằng dấu phẩy (,)**  
  → Ví dụ: *https://abc.com/page1,https://abc.com/page2*
* **URLs là bắt buộc** để thực hiện chức năng chụp hình

### ****Tùy chỉnh thêm (không bắt buộc):****

#### ****Custom CSS****

#### **-** Mặc định:

*.hide.hidden {*

*display: block !important;*

*background-color: rgba(251, 233, 233, 255);*

*}*

*-* Nhập 0 để **bỏ CSS mặc định**

* ****Delays:****

Chờ bao nhiêu ***milliseconds*** trước khi chụp  
 (ví dụ: 1000 để đợi 1 giây)

#### ****Width****

Nhập **chiều rộng màn hình khi chụp, tối thiểu 301px**

### ****Choose save folder (bắt buộc)****

#### Cách 1: Bấm **Choose save folder** để chọn thư mục lưu

#### Cách 2: Nhập đường dẫn thư mục trực tiếp vào ô input

### ****Take Screenshot****

* Chỉ thực hiện khi đã nhập **URLs** và chọn thư mục lưu file (Choose save folder)

- Khi chụp hình thành công sẽ log trong LogPanel có đếm thứ tự URLs - > chụp hoàn tất thông báo "Photo processing complete"

- Khi chụp không thành công sẽ log trong LogPanel có đếm thứ tự URLs - > thông báo "Error when taking photos"

- Khi chụp không thành công do timeout sẽ log trong LogPanel có đếm thứ tự URLs - > thông báo "Timeout image - Cannot be captured"

- Khi chụp có hình giống nhau về kích thước ,màu, pixel sẽ log trong Duplicate for ${url} có đếm thứ tự URLs - > thông báo "Duplicate image"

* Đường dẫn thư mục lưu ảnh:

<Thư mục đã chọn>/screenshot/year/month/day/<ảnh chụp>

### ****Reset****

Xóa toàn bộ **nội dung đã nhập** và **LogPanel**

### Ghi chú:

LogPanel hiển thị kết quả từng URL theo thứ tự và trạng thái thực hiện yêu cầu từ người dùng

Link github:<https://github.com/diepdau/ScreenShotTool>