



4

Lab

Truyền thông với Web Server trong C#

Working with Web Server in C#

Thực hành Lập trình mạng căn bản
GVHD: Nguyễn Văn Bảo

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Nắm được cách giao tiếp với Web Server trong C# và vận dụng để viết các ứng dụng gửi, nhận dữ liệu, truyền thông với Web Server.
- Làm quen và tìm hiểu cách tự xây dựng một website đơn giản với HTML.

2. Môi trường

- IDE Microsoft Visual Studio 2010 trở lên.

3. Liên quan

- Sinh viên cần nắm được các kiến thức nền tảng về lập trình. Các kiến thức này đã được giới thiệu trong các môn học trước và trong nội dung lý thuyết đã học do đó sẽ không được trình bày lại trong nội dung thực hành này.
- Tham khảo tài liệu (Mục E).

B. THỰC HÀNH

1. Giới thiệu về xây dựng các ứng dụng tương tác với Internet Server

Microsoft .NET Framework hỗ trợ khả năng phân lớp, mở rộng và hỗ trợ tích hợp các dịch vụ Internet nhanh chóng và dễ dàng vào các ứng dụng. Các ứng dụng mạng có thể xây dựng trên các giao thức được xây dựng sẵn để tương tác với các server.

Các kiến thức liên quan đến nội dung thực hành trong Lab 4, sinh viên cũng có thể tham khảo thêm chi tiết tại <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/>, tập trung vào nội dung **Requesting Data** - <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/requesting-data>

Ngoài ra, sinh viên có thể tham khảo từ kiến thức từ Nội dung Lý thuyết Chương 4 (Web Server).

2. Cách tạo một Website đơn giản chạy trên localhost

Bước 1: Mở một IDE hỗ trợ soạn thảo, viết mã nguồn HTML (Notepad, Visual Studio,...), và tạo một file với đuôi là .html. Ví dụ: index.html

Bước 2: Viết mã nguồn để tạo một Website đơn giản. Tham khảo file bên dưới

Lưu ý: Sinh viên có thể tìm hiểu thêm về HTML tại: <https://www.w3schools.com/> hoặc tự tạo một trang web khác tương đương bằng HTML, CSS.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>UIT-NT106.J21.ANTT Lab 4 - Working with Web Server</title>
  <meta name="description" content="Lab4" />
  <meta name="keywords" content="Lab4" />
</head>
<body class='pushmenu-push' id="page">
  <div class="dialog" style="width:60%;margin:10% auto;text-align:center;">
    <div class="dialogBox">
      <h4>Chào bạn!</h4>
      <h3>NT106 Lab 4 - Working with Web Server</h3>
      
      <div style="color:blue;">Bạn đang làm bài Lab 4 của Môn học</div>
      <br/>
      <i>Chúc bạn hoàn thành bài Lab thành công.</i>
    </div>
  </div>
</body>
```

Đoạn mã nguồn trên, khi chạy trên trình duyệt sẽ hiển thị dưới dạng như sau:

Chào bạn!

NT106 Lab 4 - Working with Web Server

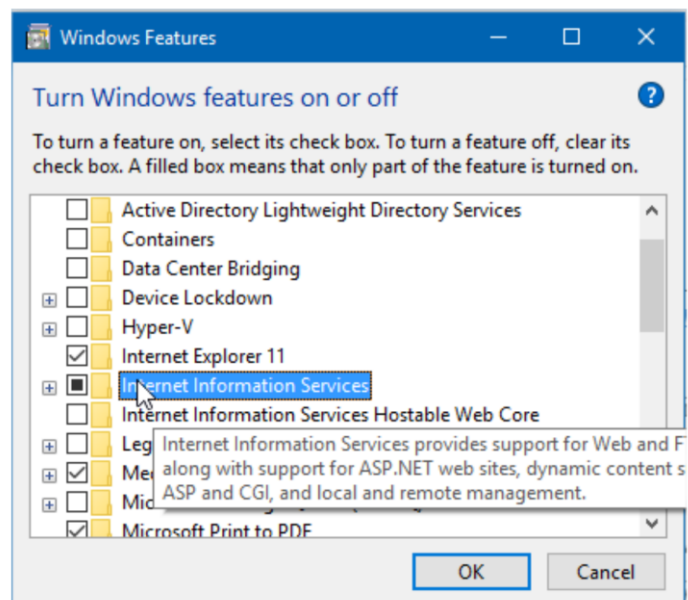


UIT-HCM

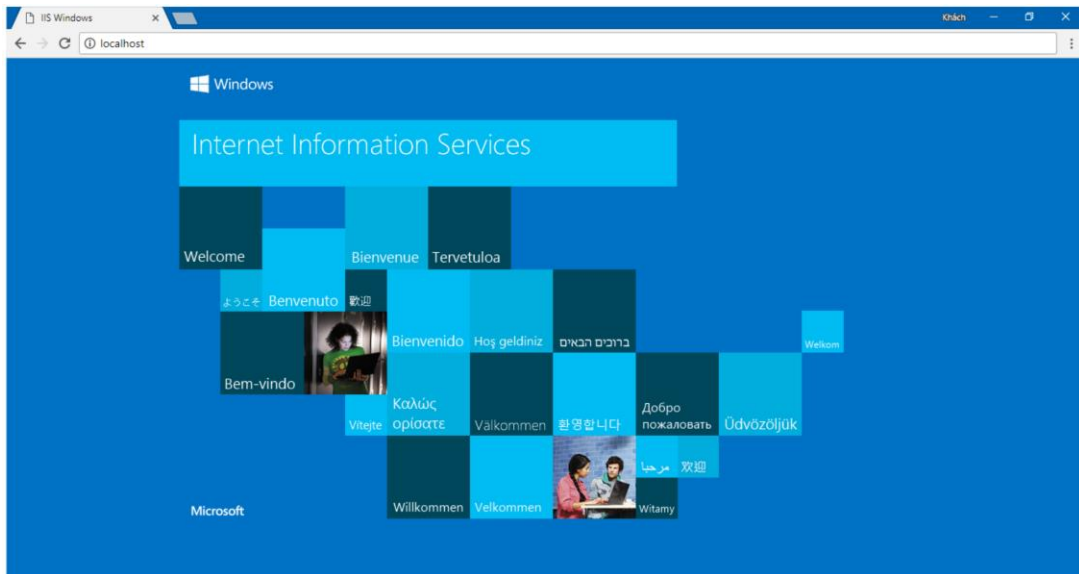
Bạn đang làm bài Lab 4 của Môn học

Chúc bạn hoàn thành bài Lab thành công.

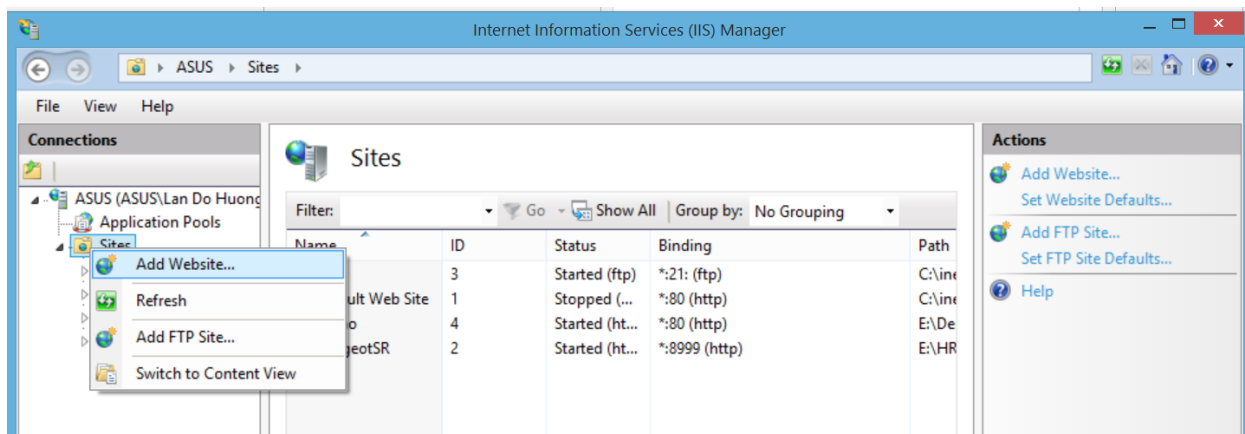
Bước 3: Bật dịch vụ **IIS** (dịch vụ để tạo máy chủ Web/FTP,...) trên **Windows 7/8/10** với các bước như sau: **Control Panel** → **Programs and features** → **Turn Windows Features on or off** → Chọn **Internet Information Service** → **OK**.

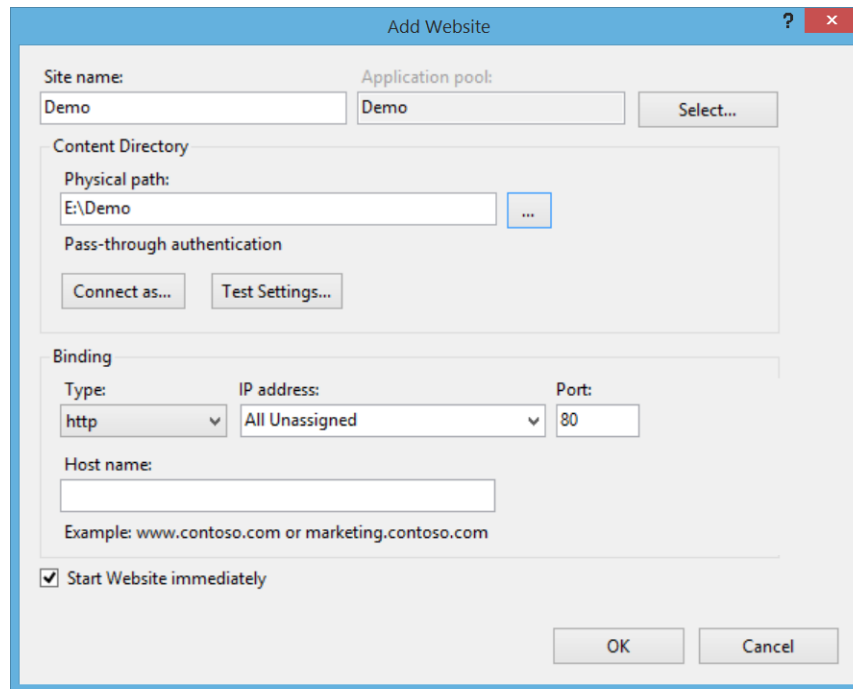


Bước 3: Sau khi bật dịch vụ IIS, kiểm tra dịch vụ đã hoạt động hay chưa bằng cách sử dụng một trình duyệt bất kỳ, và gõ **localhost**, nếu một trang Web mặc định của IIS hiện ra là thành công.

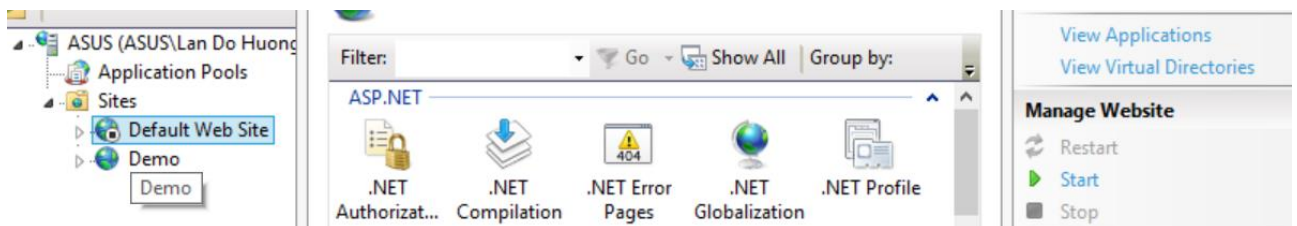


Bước 4: Lúc này ta đã có một máy chủ web đang hoạt động ở địa chỉ localhost. Mở IIS, để chạy một website trên địa chỉ này, nhấp chuột phải vào **Site** và **Add Website**. Với **Physical path** là tên folder chứa file html đã tạo ở Bước 1,2.





Lưu ý: Dừng chạy (Stop) trang mặc định của IIS và Chạy (Start) trang vừa khởi tạo để địa chỉ localhost chạy web site mới.



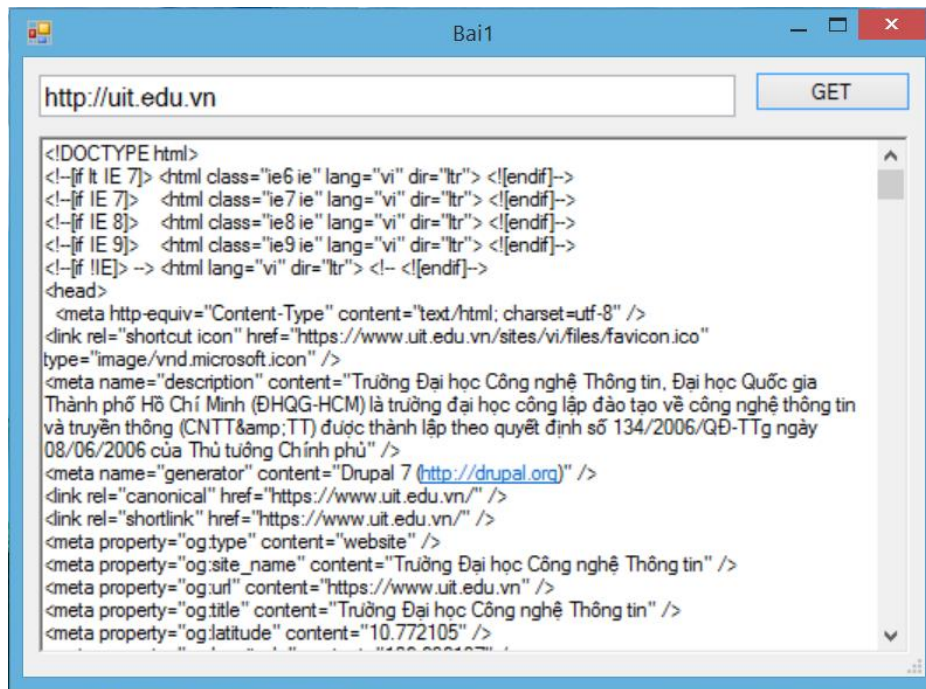
Bước 5: Truy cập vào trang web với đường dẫn <http://localhost/index.html> hoặc <http://localhost>. Truy cập thử trang web của sinh viên khác từ trình duyệt bằng các gõ URL như sau: <http://A.B.C.D/index.html> với A.B.C.D là địa chỉ IP của máy tính mà bạn mình sử dụng.

Lưu ý: sinh viên và bạn mình phải sử dụng chung mạng Local (LAN).

3. Nội dung thực hành

Bài 1: Viết chương trình hiển thị nội dung HTML của một trang web bất kỳ:

Giao diện tham khảo:



Gợi ý: Sử dụng WebRequest và WebResponse:

- Khởi tạo WebRequest với đường link web bất kỳ.
- Dùng Stream để đón nhận dữ liệu từ WebResponse trả về.
- Tải nội dung trang web lưu vào biến String.
- Đọc dữ liệu và in ra màn hình.

Lưu ý: Khai báo sử dụng các thư viện cần thiết.

Code tham khảo: Hàm lấy dữ liệu sử dụng WebRequest:

```
private string getHTML(string szURL)
{
    // Create a request for the URL.
    WebRequest request = WebRequest.Create(szURL);
    // Get the response.
    WebResponse response = request.GetResponse();
    // Get the stream containing content returned by the server.
    Stream dataStream = response.GetResponseStream();
    // Open the stream using a StreamReader for easy access.
    StreamReader reader = new StreamReader(dataStream);
    // Read the content.
    string responseFromServer = reader.ReadToEnd();
    // Close the response.
    response.Close();
    return responseFromServer;
}
```


Sinh viên tự viết hàm bắt sự kiện nút **GET** và gọi hàm **getHTML**, hiển thị thông điệp từ hàm **getHTML** vào vùng hiển thị (giả sử dùng RichTextBox).

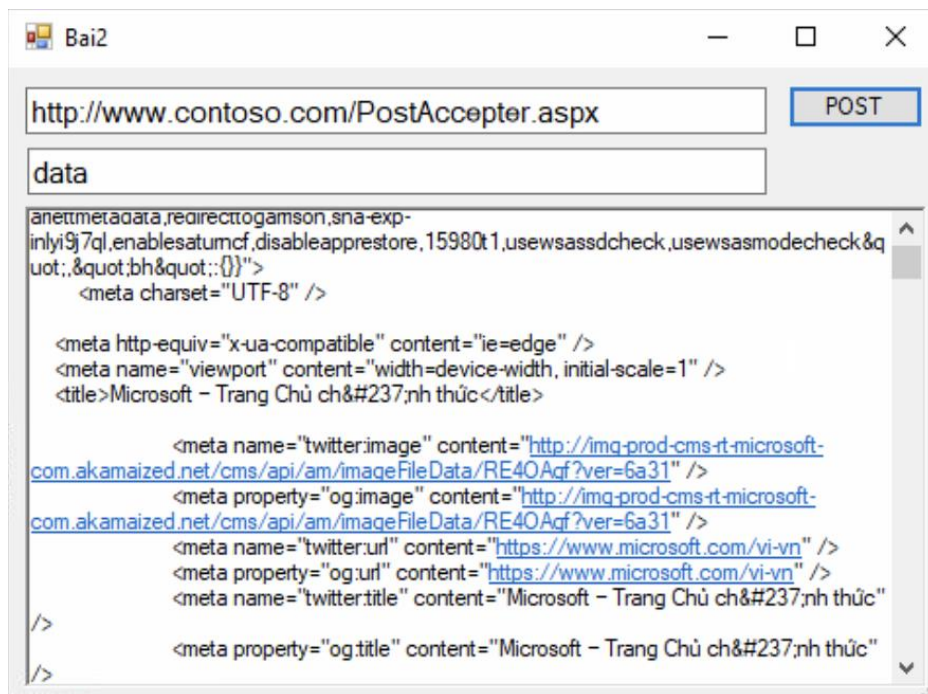
Tham khảo thêm tại:

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-request-data-using-the-webrequest-class>

Bài 2: Viết chương trình gọi dữ liệu bất kỳ lên địa chỉ:

<http://www.contoso.com/PostAcceptor.aspx> bằng phương thức POST.

Giao diện tham khảo:



Gợi ý:

- Khởi tạo request với địa chỉ URL trên sử dụng phương thức POST.
- Lấy dữ liệu POST từ form, sau đó convert sang mảng byte.
- Thiết lập độ dài của nội dung trả về (Content Length) bởi WebRequest.
- Thiết lập thuộc tính Content Type của WebRequest.
- Tạo request stream bằng cách get request stream từ WebRequest.
- Ghi dữ liệu vào request stream.
- Get response từ WebRequest gán vào response stream.

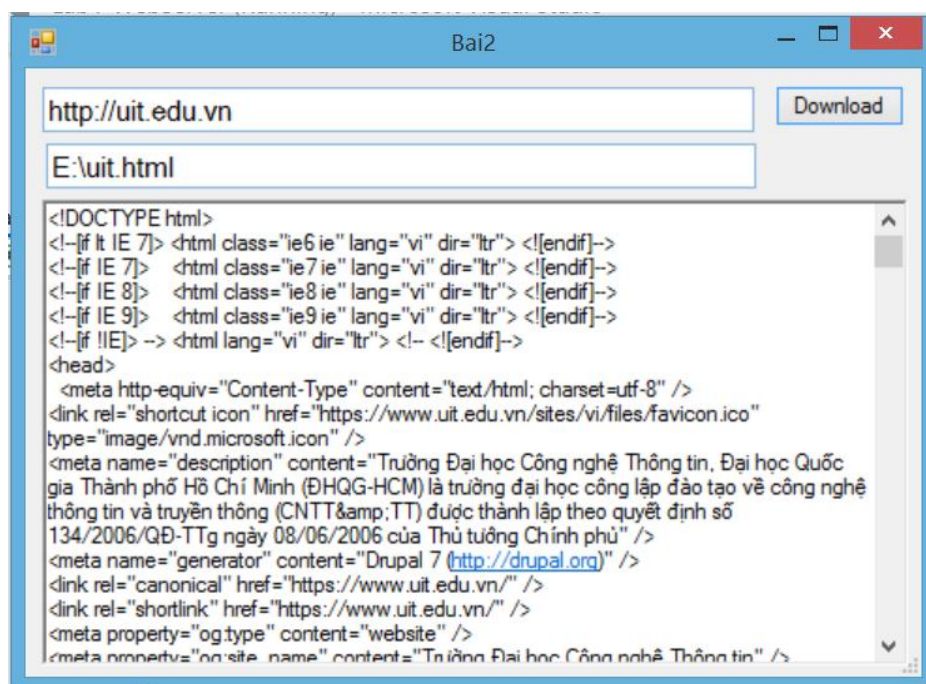
- Đọc dữ liệu từ response stream ghi vào.

Tham khảo thêm tại

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-send-data-using-the-webrequest-class>

Bài 3: Viết chương trình download nội dung trang web bất kỳ từ một địa chỉ URL bất kỳ và ghi thành file HTML, sau đó hiển thị nội dung trang web lên form.

Giao diện tham khảo:



Gợi ý: Sử dụng WebClient, và dùng phương thức DownloadFile của WebClient để lấy file về:

- Khởi tạo 1 WebClient.
- Sử dụng phương thức OpenRead để đọc nội dung trang web vào một Stream
- Viết code xử lý sự kiện khi nhấn nút download, với 1 textbox chứa địa chỉ URL cần get nội dung, 1 textbox chứa đường dẫn lưu file chứa nội dung cần get về. Sử dụng thuộc tính DownloadFile của WebClient để get nội dung trang web về lưu vào file đã chỉ định ở trên.

```
WebClient myClient = new WebClient();
```

```
Stream response = myClient.OpenRead(url);  
myClient.DownloadFile(url, fileurl);
```

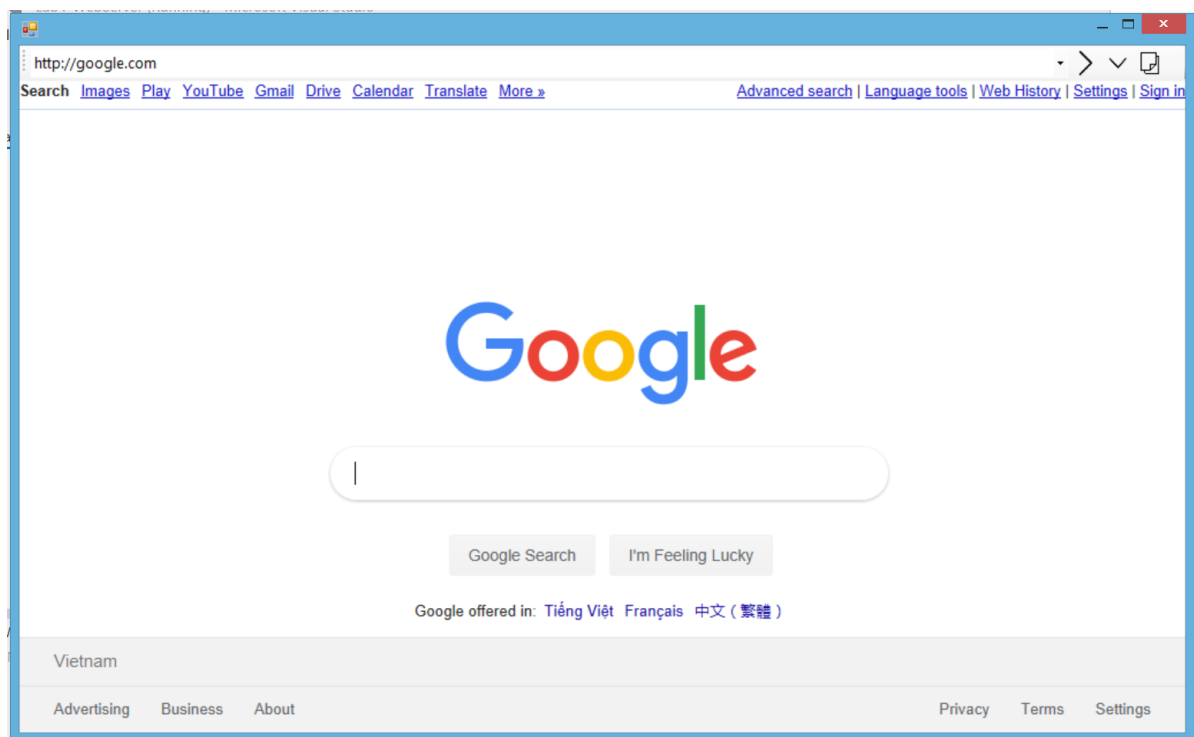
Tham khảo thêm về WebClient tại

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-request-a-web-page-and-retrieve-the-results-as-a-stream>

Bài 4: Viết chương trình hoạt động như Web Browser cơ bản cho phép thực hiện các tính năng sau:

- Xem nội dung Website
- Download File html
- Xem Source

Giao diện tham khảo:



Gợi ý:

- Sử dụng Control WebBrowser để hiển thị nội dung của Website.

- Mở rộng tính năng Download toàn bộ Source, bao gồm hình ảnh, các file liên quan bằng cách sử dụng **HTMLAgilityPack** để xử lý các file dựa trên các thẻ HTML.
- Hiển thị cửa sổ Xem Source bằng cách tạo một Form mới và parse dữ liệu sang, hiển thị Form với dữ liệu.
- Để sử dụng HTMLAgilityPack, sử dụng Nuget Packet để thêm vào Project.

Tham khảo cách thêm và sử dụng một packet tại:

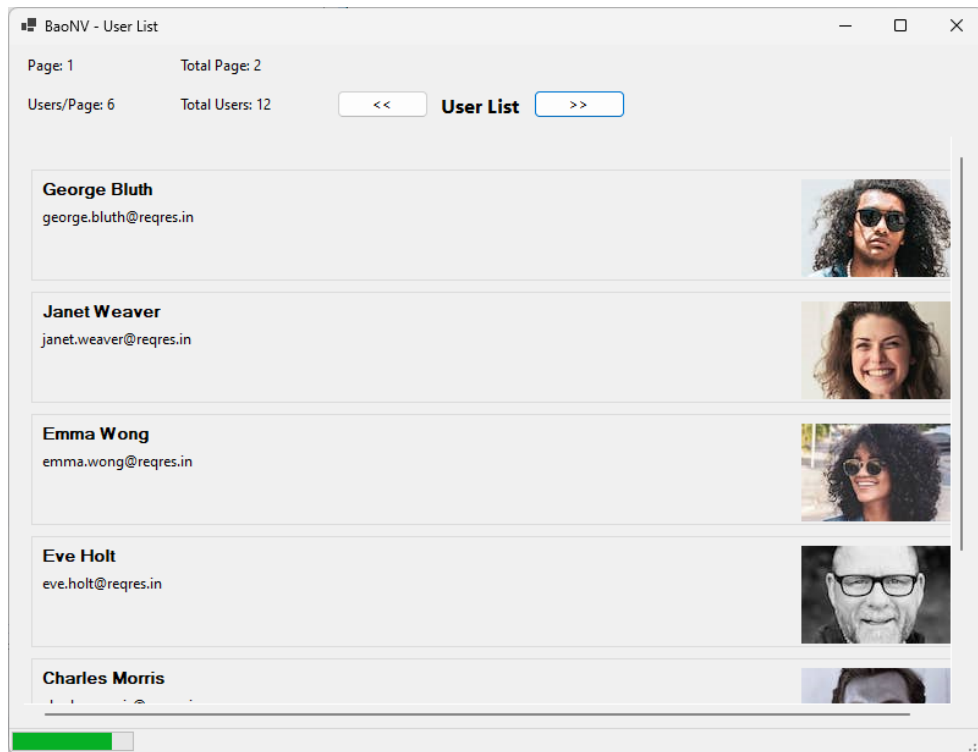
<https://docs.microsoft.com/en-us/nuget/quickstart/install-and-use-a-package-in-visual-studio>

Bài 5: Viết chương trình đọc dữ liệu từ webserver thông qua API và hiển thị thông tin lên giao diện:

- Khởi tạo một kết nối HTTP Get tới địa chỉ: <https://reqres.in/api/users?page=1>
- Nội dung phản hồi có dạng:

```
{
  page: 1,
  per_page: 6,
  total: 12,
  total_pages: 2,
  data: [
    {
      id: 1,
      email: "george.bluth@reqres.in",
      first_name: "George",
      last_name: "Bluth",
      avatar: "https://reqres.in/img/faces/1-image.jpg"
    },
    {
      id: 2,
      email: "janet.weaver@reqres.in",
      first_name: "Janet",
      last_name: "Weaver",
      avatar: "https://reqres.in/img/faces/2-image.jpg"
    },
    {
      // ...
    }
  ]
}
```

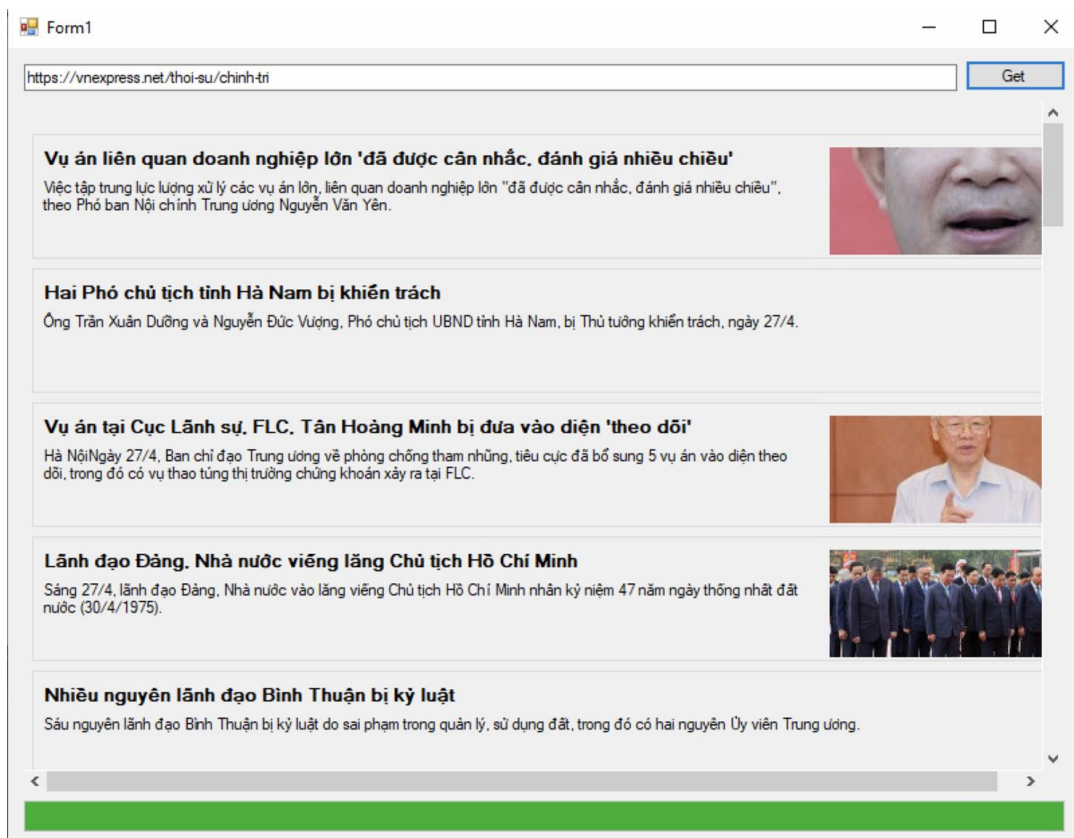
- Chuyển nội dung nhận được sang Json Object.
- Trích xuất thông tin từ Json Object để hiển thị lên giao diện.



Bài 6: Viết một ứng dụng đọc tin tức bằng cách trích xuất các thông tin của một trang Web và hiển thị theo dạng danh sách. Các tính năng cụ thể như sau:

- Xem nội dung trích xuất được dưới dạng danh sách bằng việc trích xuất các thông tin của website.
- Thành ProgressBar để hiển thị tiến trình trích xuất thông tin.
- Nếu chọn vào 1 bài tin tức cụ thể thì load website đó lên để người dùng đọc toàn văn bài báo.

Giao diện tham khảo:



C. YÊU CẦU & NỘI BÀI

1. Yêu cầu

- Các giao diện ở trên chỉ mạng tính chất minh họa, sinh viên tiến hành thiết kế giao của riêng mình đảm bảo các tiêu chí: dễ nhìn, thể hiện hết được các yêu cầu cần thực hiện, đẹp.
- Code “sạch” [2], đặt tên biến rõ ràng.
- Nội bài không đầy đủ; lỗi, không chạy được; nộp trễ; sao chép code bạn khác, nguồn có sẵn: *xử lý tùy theo mức độ*.

2. Đánh giá kết quả

- Sinh viên thực hành và nộp bài theo **Nhóm (Nhóm trưởng nộp)** tại website môn học theo thời gian quy định.
- Trình bày báo cáo gồm Ảnh chụp màn hình kèm mô tả, giải thích các bước hoạt động, thực hiện của ứng dụng đã viết trong từng bài:
- Source code nộp trên github như đã quy định.

File báo cáo đặt tên theo mẫu sau:

LabX-MSSV1-MSSV2

Ví dụ: *Lab04-20520001-20520002*

D. THAM KHẢO

- [1] Microsoft (2018). C# Guide. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- [2] Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
- [3] Network Programming in the .NET Framework: <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/>

HẾT