

BÁO CÁO THỰC HÀNH LAB 4

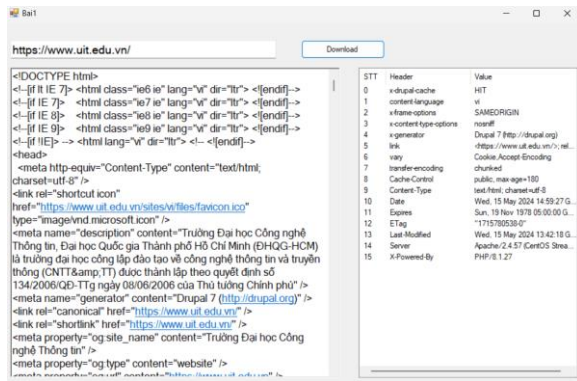
Đinh Lê Thành Công – 22520167

Mục lục

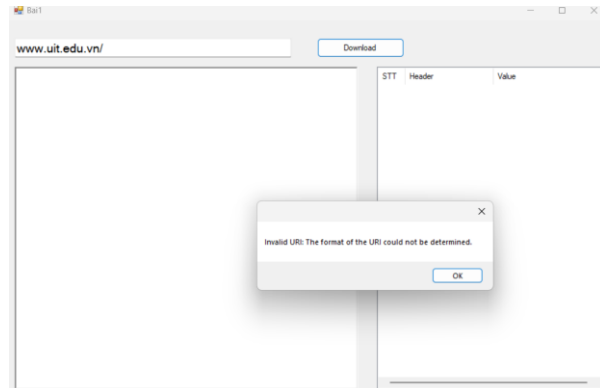
Bài 1: Chương trình tải, hiển thị nội dung header	2
1.1 Chạy chương trình.....	2
1.2 Xử lý chương trình.....	2
Bài 1.1 (Nâng cao): Cấu hình IIS	3
1.1.1 Cấu hình IIS	3
1.1.2 Xử lý chương trình.....	4
Bài 2: Chương trình xem web kèm tiện ích giả lập.....	4
2.1 Chạy chương trình.....	4
2.2 Xử lý chương trình.....	5
Bài 3: Chương trình tương tác với API	6
3.1 Chạy chương trình.....	6
3.2 Xử lý chương trình.....	6
Bài 3.1 (Nâng cao): Tự viết và vận hành API.....	7
3.1.1 Chạy chương trình.....	7
3.1.2 Xử lý chương trình.....	8
Bài 4: Web Browser cơ bản.....	9
4.1 Chạy chương trình.....	9
4.2 Xử lý chương trình.....	10

Bài 1: Chương trình tải, hiển thị nội dung header

1.1 Chạy chương trình



Hiển thị nội dung HTML và header response



Lỗi khi không nhập đúng định dạng https

1.2 Xử lý chương trình

- Ở bài này, em đã dùng WebRequest và dùng StreamReader để đọc HTML

```
WebRequest myWebRequest = WebRequest.Create(txturl.Text.Trim());  
WebResponse myWebResponse = myWebRequest.GetResponse();  
DisplayHTML(myWebResponse);  
DisplayHeader(myWebResponse);  
myWebResponse.Close();
```

Hình 1

```
private void DisplayHeader(WebResponse myWebResponse)  
{  
    for (int i = 0; i < myWebResponse.Headers.Count; ++i)  
    {  
        ListViewItem item = new ListViewItem(i.ToString());  
        item.SubItems.Add(myWebResponse.Headers.Keys[i]);  
        item.SubItems.Add(myWebResponse.Headers[i]);  
        listView1.Items.Add(item);  
    }  
}  
  
1 reference  
private void DisplayHTML(WebResponse myWebResponse)  
{  
    Stream dataStream = myWebResponse.GetResponseStream();  
    StreamReader reader = new StreamReader(dataStream);  
    string responseFromServer = reader.ReadToEnd();  
    txtcontent.Text = responseFromServer;  
}
```

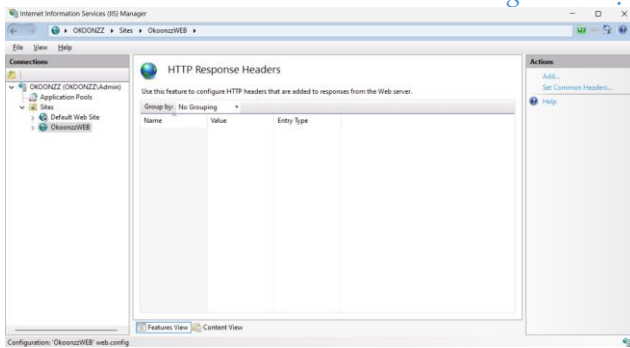
Hình 2

Bài 1.1 (Nâng cao): Cấu hình IIS

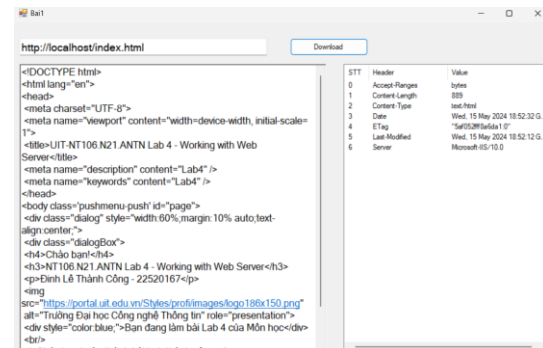
1.1.1 Cấu hình IIS



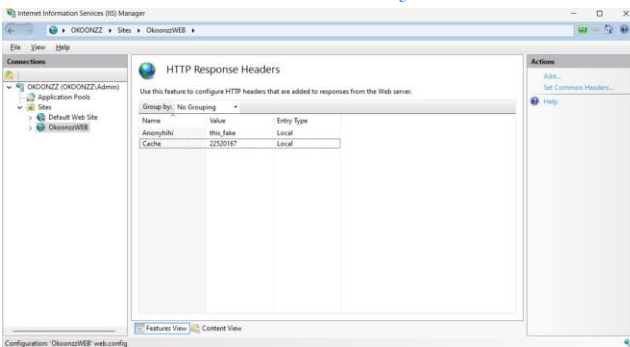
Trang web được host bởi iis



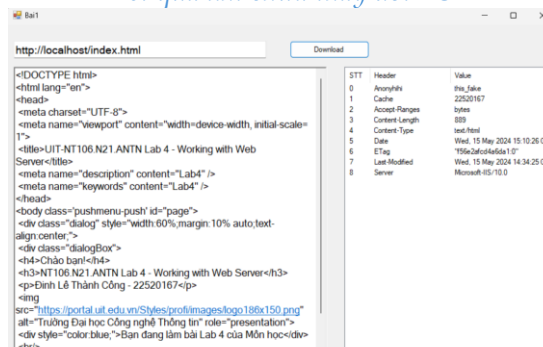
IIS chưa thực hiện thay đổi nào



Kết quả khi chưa thay đổi IIS



Thực hiện thay đổi header



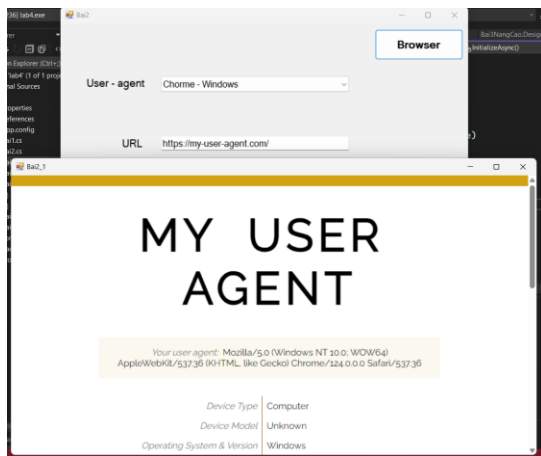
Kết quả khi thay đổi header

1.1.2 Xử lý chương trình

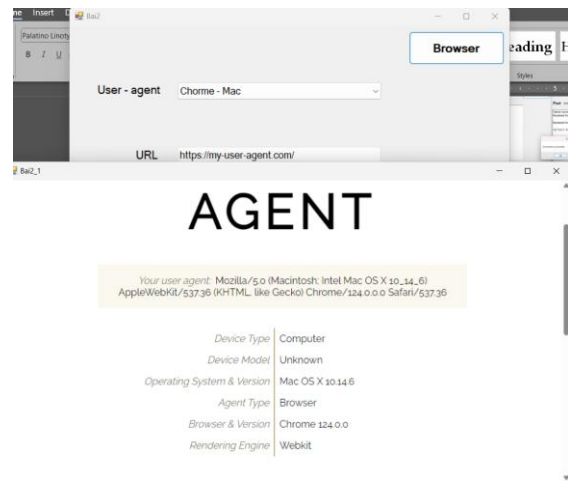
- Ở đây em đã chuẩn bị sẵn một file html (được đính kèm trong mã nguồn). Sau đó thiết lập một vài thông tin HTTP Response Header để làm giả thông tin ở header response

Bài 2: Chương trình xem web kèm tiện ích giả lập

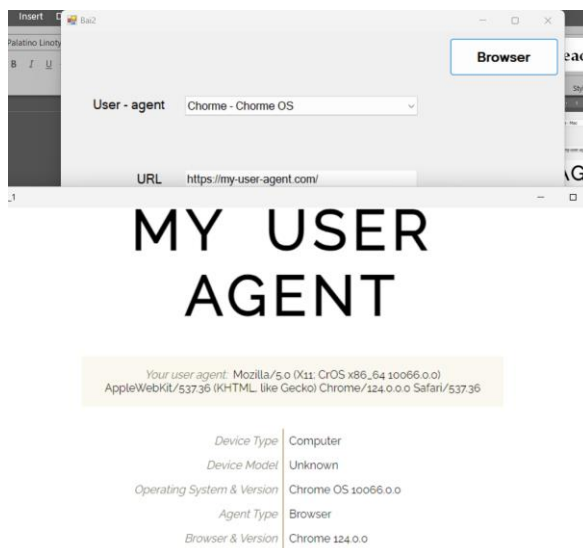
2.1 Chạy chương trình



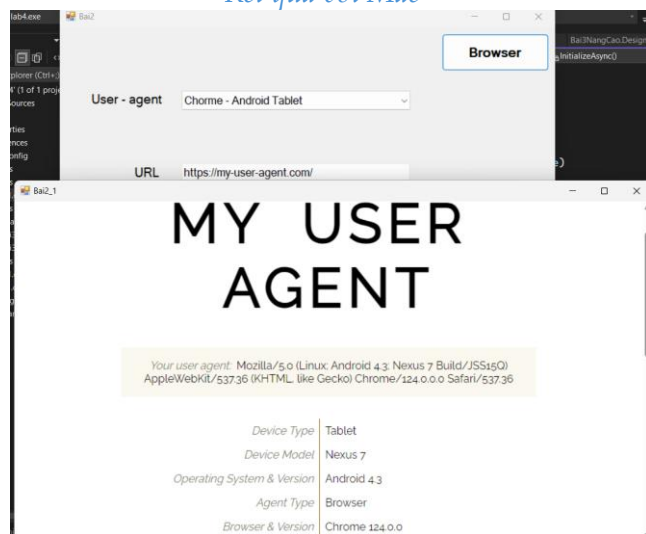
Kết quả với Windows



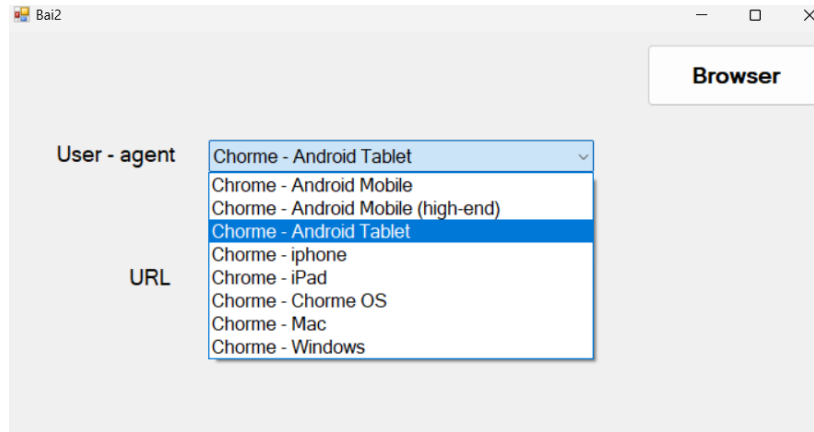
Kết quả với Mac



Kết quả với chrome OS



Kết quả với android tablet



Các UA được hỗ trợ

2.2 Xử lý chương trình

- Ở bài này, em đã sử dụng webview2 được tải từ nuget. Bởi vì webview2 sẽ hỗ trợ tốt hơn webbrowser nhất là những trang như youtube,...
- Em đã sử dụng 8 loại UA, và lưu chúng vào dictionary theo cặp key – value. Người dùng chỉ cần chọn key và chương trình sẽ tự động xử lý value tương ứng

```
Dictionary<string, string> userAgents = new Dictionary<string, string>
{
    { "Chrome - Android Mobile", "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Mobile Safari/537.36" },
    { "Chrome - Android Mobile (high-end)", "Mozilla/5.0 (Linux; Android 10; Pixel 4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3989.92 Mobile Safari/537.36" },
    { "Chrome - Android Tablet", "Mozilla/5.0 (Linux; Android 4.3; Nexus 7 Build/JSS15Q) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Tablet Safari/537.36" },
    { "Chrome - iphone", "Mozilla/5.0 (iPhone; CPU iPhone OS 13_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0 Mobile/15E148 Safari/604.1" },
    { "Chrome - iPad", "Mozilla/5.0 (iPad; CPU OS 13_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0 Mobile/15E148 Safari/604.1" },
    { "Chrome - Chrome OS", "Mozilla/5.0 (X11; CrOS x86_64 10066.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3989.92 Safari/537.36" },
    { "Chrome - Mac", "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3989.92 Safari/537.36" },
    { "Chrome - Windows", "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3989.92 Safari/537.36" }
};
```

- Tại đây em có sử dụng thêm bất đồng bộ để có thể mở nhiều form với nhiều UA khác nhau

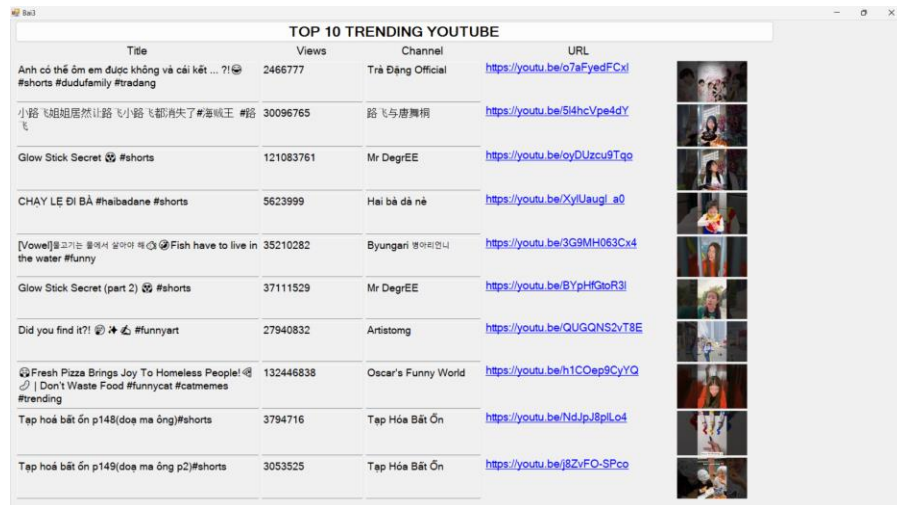
```
Thread thread = new Thread(() =>
{
    Bai2_1 bai2_1 = new Bai2_1(this);
    bai2_1.ShowDialog();
});
thread.SetApartmentState(ApartmentState.STA);
thread.Start();
```

- Thay đổi UA với webview2 với phương thức đã được định nghĩa sẵn

```
string userAgent = bai2.SelectedUserAgent;
string URL = bai2.SelectedWebsite;
webView = new WebView2
{
    Dock = DockStyle.Fill,
};
Controls.Add(webView);
await webView.EnsureCoreWebView2Async(null);
webView.CoreWebView2.Settings.UserAgent = userAgent;
webView.CoreWebView2.Navigate(URL);
```

Bài 3: Chương trình tương tác với API

3.1 Chạy chương trình



Tiêu đề	Views	Channel	URL
Anh có thể ốm em được không và cái kết ... ?! #shorts #dudufamily #tradang	2466777	Trà Đăng Official	https://youtu.be/o7aFyedFCxI
小路 姐姐居然让路 飞小路 飞都消失了 #海贼王 #路飞	30096765	路飞与唐舞桐	https://youtu.be/5i4hcVpe4dY
Glow Stick Secret #shorts	121083761	Mr DegrEE	https://youtu.be/oyDUzcu9Tqo
CHẠY LỆ ĐI BÀ #haibadane #shorts	5623999	Hai bá dâ nè	https://youtu.be/XylUaugl_a0
[Vowel] 물고기는 물에서 살아야 해 #Fish have to live in the water #funny	35210282	Byungeri 영어리만니	https://youtu.be/3G9MH063Cx4
Glow Stick Secret (part 2) #shorts	37111529	Mr DegrEE	https://youtu.be/BYpHfGaoR3I
Did you find it?! #funnyart	27940832	Artistomg	https://youtu.be/QUGQNS2vT8E
Fresh Pizza Brings Joy To Homeless People! Don't Waste Food #funnycat #catmemes #trending	132446838	Oscar's Funny World	https://youtu.be/h1COep9CyYQ
Tạp hoá bất ổn p148(dọa ma ông)#shorts	3794716	Tạp Hóa Bất Ổn	https://youtu.be/NsJpJ8pL04
Tạp hoá bất ổn p149(dọa ma ông p2)#shorts	3053525	Tạp Hóa Bất Ổn	https://youtu.be/j8ZvFO-SPco

Kết quả chạy với API youtube

3.2 Xử lý chương trình

- Với bài này em đã sử dụng API của youtube và key sẽ tự tạo với trang của [google](https://developers.google.com/youtube/v3).
- Em đã sử dụng HttpClient và Newtonsoft Json để request và parse json.

```
...clear();*/
client.BaseAddress = new Uri(URL);
client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
HttpResponseMessage response = client.GetAsync(URL).Result;
if (response.IsSuccessStatusCode)
{
    string dataObjects = await response.Content.ReadAsStringAsync();
    dynamic content = JObject.Parse(dataObjects);
    JArray items = content.items;

    for (int i = 0; i < 10; i++)
    {
        title[i] = items[i]["snippet"]["title"].ToString();
        channels[i] = items[i]["snippet"]["channelTitle"].ToString();
        views[i] = items[i]["statistics"]["viewCount"].ToString();
        _url[i] = items[i]["id"].ToString();
        _image[i] = items[i]["snippet"]["thumbnails"]["default"]["url"].ToString();
    }
    addImage(_image);
}
```

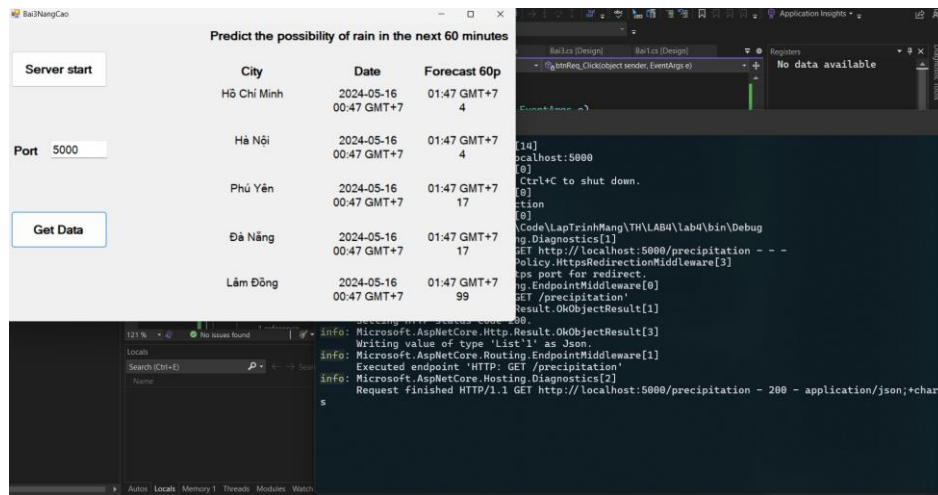
- Để hiển thị hình ảnh tiêu đề video em sử dụng thêm WebRequest và dùng bitmap để chuyển thành hình ảnh.

```
private void addImage(string[] url)
{
    for (int i = 0; i < url.Length; i++)
    {
        string tmpURL = url[i];
        WebRequest request = WebRequest.Create(tmpURL);
        WebResponse response = request.GetResponse();
        Stream responseStream = response.GetResponseStream();
        Bitmap bitmap = new Bitmap(responseStream);

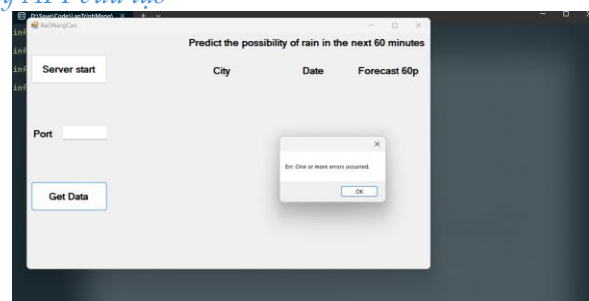
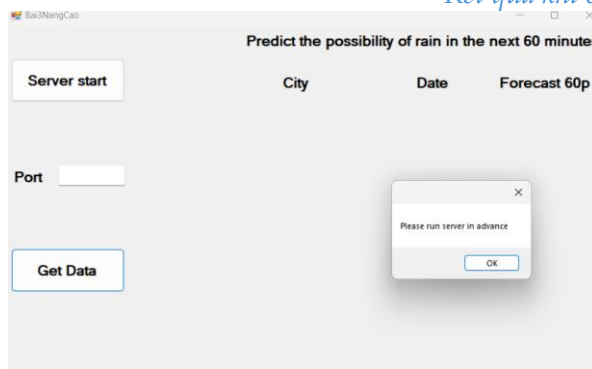
        // Thêm Bitmap vào danh sách
        bitmapList.Add(bitmap);
    }
}
```

Bài 3.1 (Nâng cao): Tự viết và vận hành API

3.1.1 Chạy chương trình

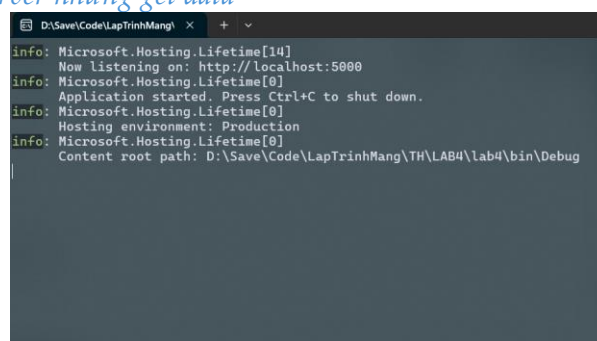


Kết quả khi chạy API vừa tạo



Lỗi chưa nhập port của server

Lỗi khi chưa chạy server nhưng get data



Thông tin khi chạy server

3.1.2 Xử lý chương trình

- Đầu tiên em sẽ dùng ASP để tạo ra một API, ở đây khi gọi API sẽ trả về json bao gồm tên thành phố, thời gian hiện tại, thời gian tiếp theo 60p, ngày hiện tại, khả năng mưa.

```

1 using Microsoft.AspNetCore;
2
3 var builder = WebApplication.CreateBuilder(args);
4
5 builder.Services.AddEndpointsApiExplorer();
6 builder.Services.AddSwaggerGen();
7 var app = builder.Build();
8 if (app.Environment.IsDevelopment())
9 {
10     app.UseSwagger();
11     app.UseSwaggerUI();
12 }
13 app.UseHttpsRedirection();
14 var random = new Random();
15 app.MapGet("/{precipitation}", () =>
16 {
17     var cities = new List<string> { "Hố Chí Minh", "Hà Nội", "Phủ Yên", "Đà Nẵng", "Lâm Đồng" };
18     var random = new Random();
19
20     //time current
21     string time_cur = DateTime.Now.ToString();
22     DateTime dateTime_cur = DateTime.Parse(time_cur);
23     TimeSpan time_cur = dateTime_cur.TimeOfDay;
24     string formattedTime_cur = time_cur.ToString("hh\\:mm") + " GMT+7";
25
26     //60p next
27     string time_for = DateTime.Now.AddMinutes(60).ToString();
28     DateTime dateTime_for = DateTime.Parse(time_for);
29     TimeSpan time_for = dateTime_for.TimeOfDay;
30 }

```

Hình 1

```

31
32
33 var precipitations = new List<Precipitation>();
34 foreach (var city in cities)
35 {
36     var precipitation = new Precipitation
37     {
38         City = city,
39         Date = DateOnly.FromDateTime(DateTime.Now),
40         Time_cur = formattedTime_cur,
41         Time_for = formattedTime_for,
42         Amount = random.Next(0, 100)
43     };
44     precipitations.Add(precipitation);
45 }
46 return Results.Ok(precipitations);
47 }
48 [WithName("GetPrecipitation")]
49 [WithOpenApi()]
50 app.Run();
51 // references
52 [internal, record] Precipitation
53 {
54     1 reference
55     public string? Time_cur { get; set; }
56     1 reference
57     public string? Time_for { get; set; }
58     1 reference
59     public string? City { get; set; }
60     1 reference

```

Hình 2

```
GET {{createAPI_HostAddress}}/precipitation/
  @createAPI_HostAddress = http://localhost:5267

  Send request | Debug
  ✓ GET {{createAPI_HostAddress}}/precipitation/
    Accept: application/json
    |
    |
    | ###
```

Cấu hình đường dẫn

- Sau khi có được API em sẽ tiếp tục sử dụng HttpClient và Newtonsoft để gửi và nhận dữ liệu, sau đó parse dữ liệu sang và hiển thị lên form.

```

    }
    URL = $"http://localhost:{txtport.Text.Trim()}/precipitation";
    using (HttpClient client = new HttpClient())
    {
        try
        {
            client.BaseAddress = new Uri(URL);
            client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
            HttpResponseMessage response = client.GetAsync(URL).Result;

            if (response.IsSuccessStatusCode)
            {
                string dataObjects = await response.Content.ReadAsStringAsync();
                dynamic data = Newtonsoft.Json.JsonConvert.DeserializeObject(dataObjects);
                for (int i = 0; i < 5; i++)
                {
                    city[i] = data[i].city.ToString();
                    date[i] = data[i].date.ToString();
                    amount[i] = data[i].amount.ToString();
                    time_current[i] = data[i].time_cur.ToString();
                    time_forward[i] = data[i].time_for.ToString();
                }
                display();
            }
        }
    }
}

```


Bài 4: Web Browser cơ bản

4.1 Chạy chương trình



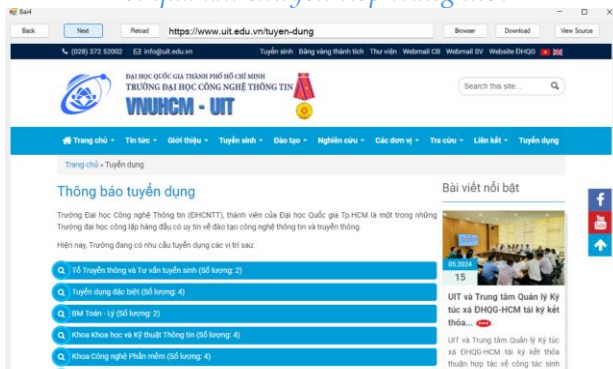
Kết quả khi render HTML



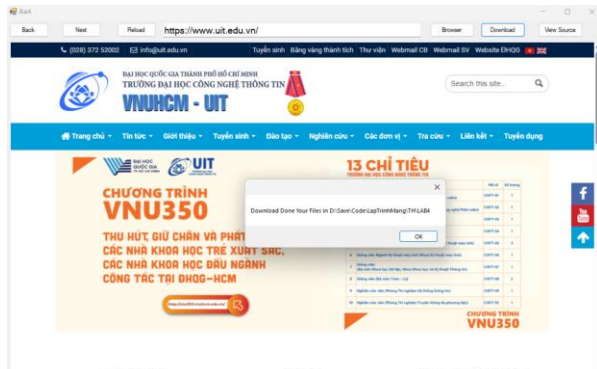
Kết quả khi chuyển tiếp trang mới



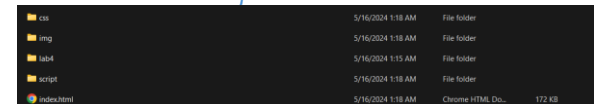
Kết quả với nút back



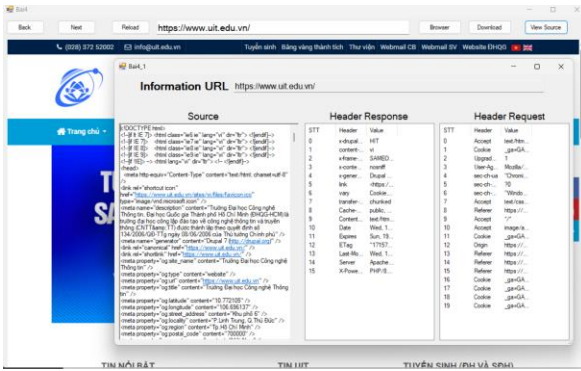
Kết quả với nút next



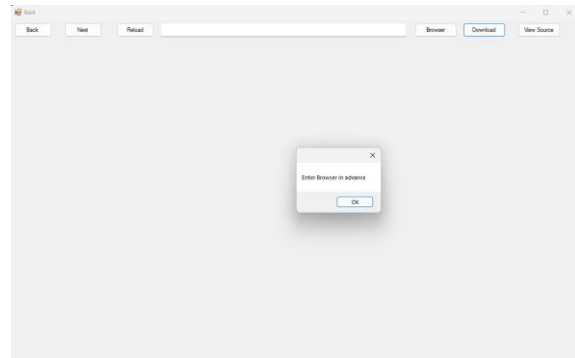
Hoàn thành lưu source code



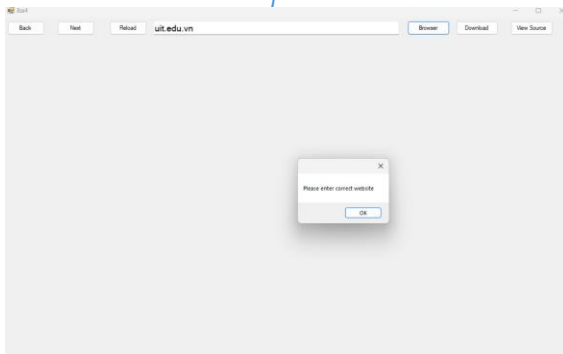
Kết quả lưu source code



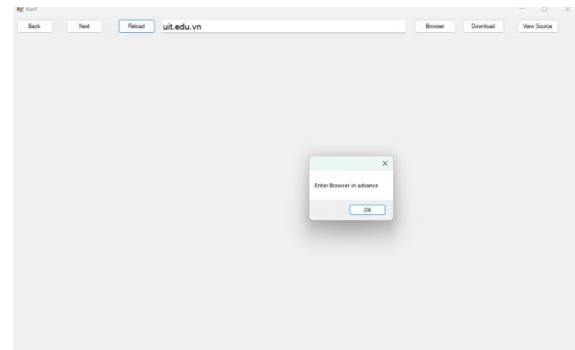
Hiện thị source code, header request, header response



Lỗi khi chưa load web nhưng tải source



Lỗi khi nhập không đúng định dạng url



Lỗi khi chưa load web nhưng reload

4.2 Xử lý chương trình

- Ở bài này em vẫn dùng webview2 và sử dụng package ở nuget với tên HtmlAgilityPack để tải toàn bộ source code (bao gồm cả js, css, image, ...)

```
DialogResult result = ofd.ShowDialog();
if (result == DialogResult.OK && !string.IsNullOrEmpty(ofd.SelectedPath))
{
    savePath = ofd.SelectedPath;
    string URL = txturl.Text.Trim();
    var web = new HtmlWeb();
    var doc = await Task.Factory.StartNew(() => web.Load(URL));
    var nodeCollection = doc.DocumentNode.DescendantsAndSelf();

    //download HTML
    using (var client = new WebClient())
    {
        var htmlContent = await client.DownloadStringTaskAsync(URL);
        string path = Path.Combine(savePath, "index.html");

        using (StreamWriter writer = new StreamWriter(path))
        {
            await writer.WriteAsync(htmlContent);
        }
    }
}
```

Hình 1

```
//download all file
foreach (var node in nodeCollection)
{
    try
    {
        if (node.Name == "img" || node.Name == "script" || node.Name == "link")
        {
            var attribute = node.Name == "link" ? "href" : "src";
            var path = node.GetAttributeValue(attribute, null);
            if (path != null)
            {
                // Kiểm tra xem đường dẫn có phải là đường dẫn tuyệt đối hay không
                var uri = path.StartsWith("http") ? new Uri(path) : new Uri(new Uri(URL), path);
                string folderName = node.Name == "link" ? "css" : node.Name;
                DownloadFile(uri, folderName);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Hình 2

```
private void DownloadFile(Uri uri, string folderName)
{
    if (!isBrowserButtonClicked)
    {
        MessageBox.Show("Enter Browser in advance");
        return;
    }
    using (var client = new WebClient())
    {
        try
        {
            var path = Path.Combine(savePath, folderName, Path.GetFileName(uri.LocalPath));
            Directory.CreateDirectory(Path.GetDirectoryName(path));
            client.DownloadFile(uri, path);
        }
        catch
        {
        }
    }
}
```

Hình 3

- Với phần hiển thị ra request header, em sử dụng phương thức của webview2 đó là WebResourceRequest, sau đó lưu các header vào một set với cặp key và value.

```
webView21.CoreWebView2.NavigationCompleted += CoreWebView2_NavigationCompleted;
// Thêm một bộ lọc cho tất cả các yêu cầu
webView21.CoreWebView2.AddWebResourceRequestedFilter("*", CoreWebView2WebResourceContext.All);
// Đăng ký sự kiện WebResourceRequested
webView21.CoreWebView2.WebResourceRequested += CoreWebView2_WebResourceRequested;
selectURL = txturl.Text;
webView21.CoreWebView2.Navigate(URL);
isBrowserButtonClicked = true;
```

Dùng WebResourceRequest

```
public HashSet<KeyValuePair<string, string>> headersList = new HashSet<KeyValuePair<string, string>>();
public void CoreWebView2_WebResourceRequested(object sender, CoreWebView2WebResourceRequestedEventArgs e)
{
    CoreWebView2HttpRequestHeaders headers = e.Request.Headers;
    foreach (var header in headers)
    {
        headersList.Add(new KeyValuePair<string, string>(header.Key, header.Value));
    }
}
```

Lưu các header vào set