TẠO SITEMAP ĐỘNG TRONG ASP.NET MVC C#

B1: Tạo project ASP.Net MVC hoặc sử dụng project có sẵn.

B2: Tạo folder Helper và các lớp tiện ích sau:

```
using System.Text.RegularExpressions;

    public static class RegexUtils
    {
        /// <summary>
        /// A regular expression for validating slugs.
        /// Does not allow leading or trailing hypens or whitespace
        /// </summary>
        public static readonly Regex SlugRegex = new Regex(@"(^[a-z0-9])([a-z0-
9_-]+)*([a-z0-9])$");
    }
```

```
using System.Text;
using System.Text.RegularExpressions;

    public static class StringExtensions
    {
        /// <summary>
        /// A nicer way of calling <see
cref="System.String.IsNullOrEmpty(string)"/>
        /// </summary>
        /// <param name="value">The string to test.</param>
        /// <returns>true if the value parameter is null or an empty string ("");
otherwise, false.</returns>
        public static bool IsNullOrEmpty(this string value)
        {
            return string.IsNullOrEmpty(value);
        }

        /// <summary>
        /// A nicer way of calling the inverse of <see
cref="System.String.IsNullOrEmpty(string)"/>
        /// </summary>
        /// <param name="value">The string to test.</param>
        /// <returns>true if the value parameter is not null or an empty string
(""); otherwise, false.</returns>
        public static bool IsNotNullOrEmpty(this string value)
        {
            return !value.IsNullOrEmpty();
        }

        /// <summary>
        /// Slugifies a string
        /// </summary>
        /// <param name="value">The string value to slugify</param>
        /// <param name="maxLength">An optional maximum length of the generated
slug</param>
        /// <returns>A URL safe slug representation of the input <paramref
name="value"/>.</returns>
```

```
        public static string ToSlug(this string value, int? maxLength = null)
        {
            // Ensure.Argument.NotNull(value, "value");

            // if it's already a valid slug, return it
            if (RegexUtils.SlugRegex.IsMatch(value))
                return value;

            return GenerateSlug(value, maxLength);
        }


        /// <summary>
        /// Credit for this method goes to
http://stackoverflow.com/questions/2920744/url-slugify-alrogithm-in-cs
        /// </summary>
        private static string GenerateSlug(string value, int? maxLength = null)
        {
            // prepare string, remove accents, lower case and convert hyphens to
whitespace
            var result = RemoveAccent(value).Replace("-", "
").ToLowerInvariant();

            result = Regex.Replace(result, @"[^a-z0-9\s-]", string.Empty); //
remove invalid characters
            result = Regex.Replace(result, @"\s+", " ").Trim(); // convert
multiple spaces into one space

            if (maxLength.HasValue) // cut and trim
                result = result.Substring(0, result.Length <= maxLength ?
result.Length : maxLength.Value).Trim();

            return Regex.Replace(result, @"\s", "-"); // replace all spaces with
hyphens
        }



        private static string RemoveAccent(string value)
        {
            var bytes = Encoding.GetEncoding("Cyrillic").GetBytes(value);
            return Encoding.ASCII.GetString(bytes);
        }

    }
```

```
using System.IO;

    /// <summary>
    /// Utility methods for working with resource paths
    /// </summary>
    public static class PathUtils
    {
        /// <summary>
        /// Makes a filename safe for use within a URL
        /// </summary>
        public static string MakeFileNameSafeForUrls(string fileName)
        {

            var extension = Path.GetExtension(fileName);
            var safeFileName =
Path.GetFileNameWithoutExtension(fileName).ToSlug();
            return Path.Combine(Path.GetDirectoryName(fileName), safeFileName +
extension);
        }

        /// <summary>
        /// Combines two URL paths
        /// </summary>
        public static string CombinePaths(string path1, string path2)
        {


            if (path2.IsNullOrEmpty())
            {
                return path1;
            }

            if (path1.IsNullOrEmpty())
                return path2;

            if (path2.StartsWith("http://") || path2.StartsWith("https://"))
                return path2;

            var ch = path1[path1.Length - 1];

            if (ch != '/')
                return (path1.TrimEnd('/') + '/' + path2.TrimStart('/'));

            return (path1 + path2);
        }

    }
```

B3: Tạo Model và interface cho Sitemap

Ví dụ một node trong sitemap:

```
<url>
  <loc>http://www.domain.com/bai-viet</loc>
  <lastmod>2020-01-01</lastmod>
  <changefreq>weekly</changefreq>
  <priority>0.8</priority>
</url>
```

Tạo enum `SitemapChangeFrequency`

```
    /// <summary>
    /// How frequently the page is likely to change.
    /// This value provides general information to search engines and may not
correlate exactly to how often they crawl the page.
    /// </summary>
    /// <remarks>
    /// The value "always" should be used to describe documents that change each
time they are accessed. The value "never" should be used to describe archived
URLs.
    /// </remarks>
    public enum SitemapChangeFrequency
    {
        Always,
        Hourly,
        Daily,
        Weekly,
        Monthly,
        Yearly,
        Never
    }
```

Tạo các interface sau trong folder Models:

```
    /// <summary>
    /// An interface for sitemap items
    /// </summary>
    public interface ISitemapItem
    {
        /// <summary>
        /// URL of the page.
        /// </summary>
        string Url { get; }

        /// <summary>
        /// The date of last modification of the file.
        /// </summary>
        DateTime? LastModified { get; }

        /// <summary>
        /// How frequently the page is likely to change.
        /// </summary>
        SitemapChangeFrequency? ChangeFrequency { get; }

        /// <summary>
        /// The priority of this URL relative to other URLs on your site. Valid
values range from 0.0 to 1.0.
        /// </summary>
        double? Priority { get; }
    }
```

```
using System.Xml.Linq;

    public interface ISitemapGenerator
    {
        XDocument GenerateSiteMap(IEnumerable<ISitemapItem> items);
    }
```

Tạo class `SitemapItem` kế thừa từ `ISitemapItem`

```
/// <summary>
    /// Represents a sitemap item.
    /// </summary>
    public class SitemapItem : ISitemapItem
    {
        /// <summary>
        /// Creates a new instance of <see cref="SitemapItem"/>
        /// </summary>
        /// <param name="url">URL of the page. Optional.</param>
        /// <param name="lastModified">The date of last modification of the file.
Optional.</param>
        /// <param name="changeFrequency">How frequently the page is likely to
change. Optional.</param>
        /// <param name="priority">The priority of this URL relative to other
URLs on your site. Valid values range from 0.0 to 1.0. Optional.</param>
        /// <exception cref="System.ArgumentNullException">If the <paramref
name="url"/> is null or empty.</exception>
        public SitemapItem(string url, DateTime? lastModified = null,
SitemapChangeFrequency? changeFrequency = null, double? priority = null)
        {
            // Ensure.Argument.NotNullOrEmpty(url, "url");

            Url = url;
            LastModified = lastModified;
            ChangeFrequency = changeFrequency;
            Priority = priority;
        }

        /// <summary>
        /// URL of the page.
        /// </summary>
        public string Url { get; protected set; }

        /// <summary>
        /// The date of last modification of the file.
        /// </summary>
        public DateTime? LastModified { get; protected set; }

        /// <summary>
        /// How frequently the page is likely to change.
        /// </summary>
        public SitemapChangeFrequency? ChangeFrequency { get; protected set; }

        /// <summary>
        /// The priority of this URL relative to other URLs on your site. Valid
values range from 0.0 to 1.0.
        /// </summary>
        public double? Priority { get; protected set; }
    }
```

Tạo class SitemapGenerator kế thừa từ ISitemapGenerator

```
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Xml.Linq;

    /// <summary>
    /// A class for creating XML Sitemaps (see
http://www.sitemaps.org/protocol.html)
    /// </summary>
    public class SitemapGenerator : ISitemapGenerator
    {
        private static readonly XNamespace xmlns =
"http://www.sitemaps.org/schemas/sitemap/0.9";
        private static readonly XNamespace xsi =
"http://www.w3.org/2001/XMLSchema-instance";

        public XDocument GenerateSiteMap(IEnumerable<ISitemapItem> items)
        {
            //Ensure.Argument.NotNull(items, "items");

            var sitemap = new XDocument(
                new XDeclaration("1.0", "utf-8", "yes"),
                    new XElement(xmlns + "urlset",
                      new XAttribute("xmlns", xmlns),
                      new XAttribute(XNamespace.Xmlns + "xsi", xsi),
                      new XAttribute(xsi + "schemaLocation",
"http://www.sitemaps.org/schemas/sitemap/0.9
http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"),
                      from item in items
                      select CreateItemElement(item)
                      )
                 );

            return sitemap;
        }

        public XElement CreateItemElement(ISitemapItem item)
        {
            var itemElement = new XElement(xmlns + "url", new XElement(xmlns +
"loc", item.Url.ToLowerInvariant()));

            // all other elements are optional

            if (item.LastModified.HasValue)
                itemElement.Add(new XElement(xmlns + "lastmod",
item.LastModified.Value.ToString("yyyy-MM-dd")));

            if (item.ChangeFrequency.HasValue)
                itemElement.Add(new XElement(xmlns + "changefreq",
item.ChangeFrequency.Value.ToString().ToLower()));

            if (item.Priority.HasValue)
                itemElement.Add(new XElement(xmlns + "priority",
item.Priority.Value.ToString("F1", CultureInfo.InvariantCulture)));

            return itemElement;
        }


    }
```

B4: Tạo class `SitemapResult` kế thừa `ActionResult` trong folder Controllers

```
using System.Text;
using System.Web.Mvc;
using System.Xml;
using <project-name>.Models;

    public class SitemapResult : ActionResult
    {
        private readonly IEnumerable<ISitemapItem> items;
        private readonly ISitemapGenerator generator;

        public SitemapResult(IEnumerable<ISitemapItem> items) : this(items, new
SitemapGenerator())
        {
        }

        public SitemapResult(IEnumerable<ISitemapItem> items, ISitemapGenerator
generator)
        {

            this.items = items;
            this.generator = generator;
        }

        public override void ExecuteResult(ControllerContext context)
        {
            var response = context.HttpContext.Response;

            response.ContentType = "text/xml";
            response.ContentEncoding = Encoding.UTF8;

            using (var writer = new XmlTextWriter(response.Output))
            {
                writer.Formatting = Formatting.Indented;
                var sitemap = generator.GenerateSiteMap(items);

                sitemap.WriteTo(writer);
            }
        }

    }
```

Tạo controller `SitemapController`

Tạo ActionResult Index để có được sitemap tĩnh:

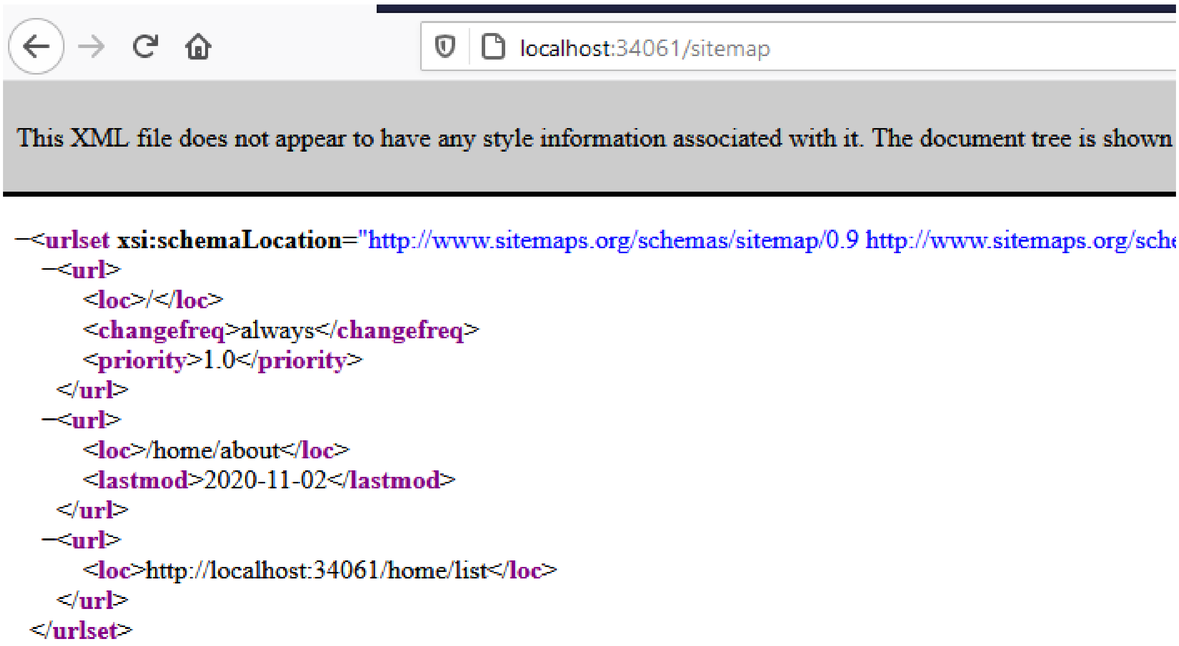
```
using <project-name>.Helper;
using <project-name>.Models;

    public class SitemapController : Controller
    {
        // GET: Sitemap
        public ActionResult Index()
        {
            var sitemapItems = new List<SitemapItem> {
            new SitemapItem(Url.Action("index", "home"), changeFrequency:
SitemapChangeFrequency.Always, priority: 1.0),
            new SitemapItem(Url.Action("about", "home"), lastModified:
DateTime.Now),
            new
SitemapItem(PathUtils.CombinePaths(Request.Url.GetLeftPart(UriPartial.Authority),
"/home/list"))
            };

            return new SitemapResult(sitemapItems);
        }

    }
```

Kết quả sau khi chạy Action này:



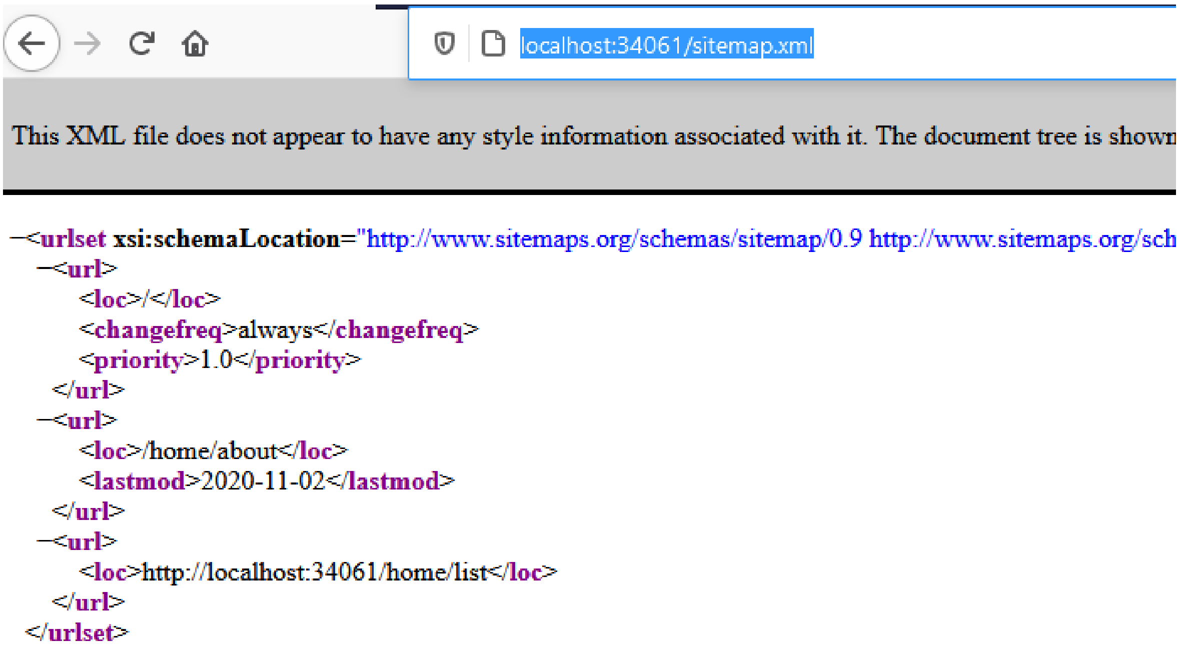
This XML file does not appear to have any style information associated with it. The document tree is shown

```
—<urlset xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9 http://www.sitemaps.org/sche
  —<url>
      <loc>/</loc>
      <changefreq>always</changefreq>
      <priority>1.0</priority>
    </url>
  —<url>
      <loc>/home/about</loc>
      <lastmod>2020-11-02</lastmod>
    </url>
  —<url>
      <loc>http://localhost:34061/home/list</loc>
    </url>
  </urlset>
```

B5: Thêm MapRoute vào RouteConfig để mở /sitemap.xml

```
routes.MapRoute(
"Sitemap",
"sitemap.xml",
new { controller = "Sitemap", action = "Index" }
);
```

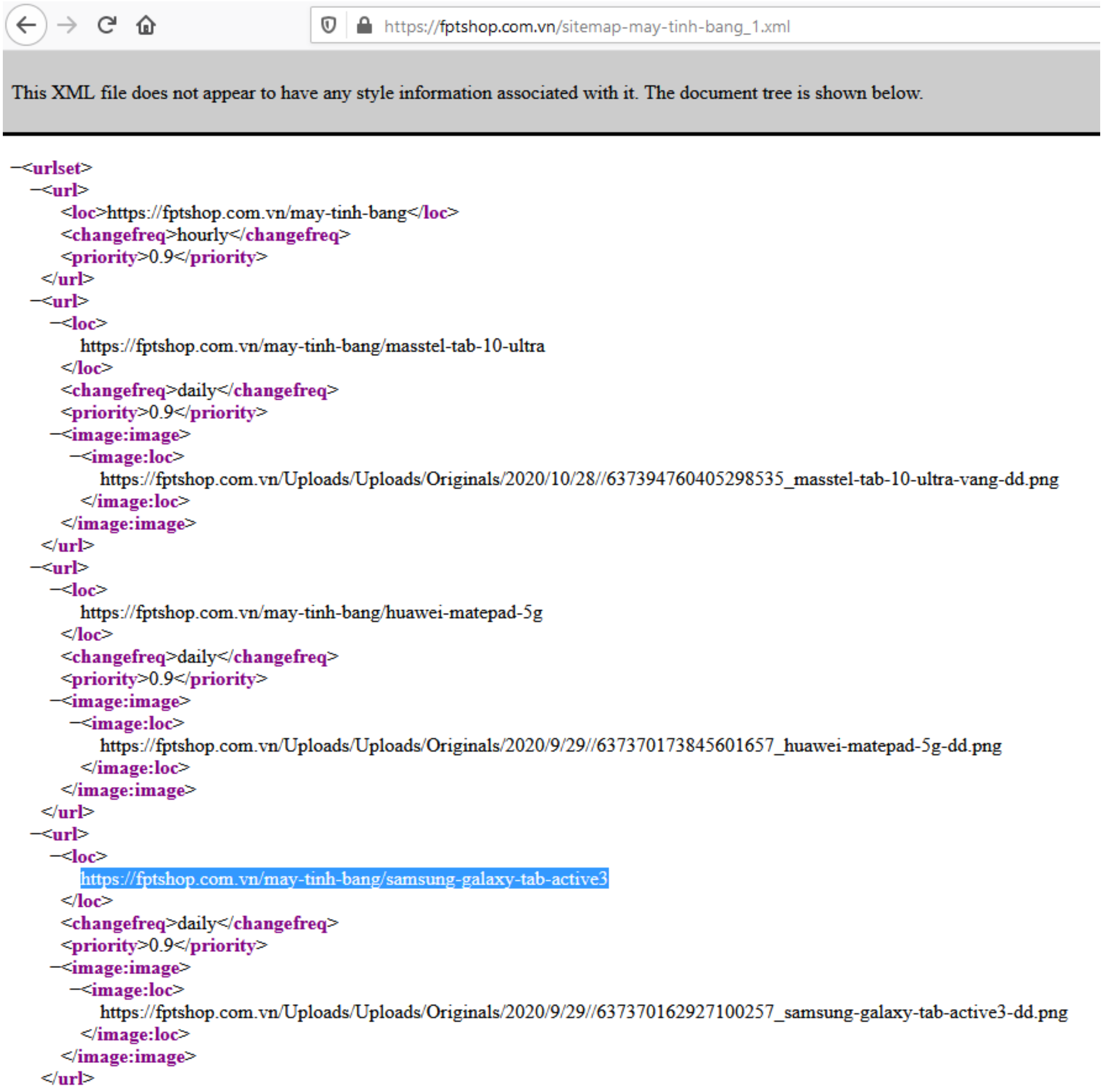
Kết quả:



This XML file does not appear to have any style information associated with it. The document tree is shown

```
—<urlset xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9 http://www.sitemaps.org/sch
  —<url>
      <loc>/</loc>
      <changefreq>always</changefreq>
      <priority>1.0</priority>
    </url>
  —<url>
      <loc>/home/about</loc>
      <lastmod>2020-11-02</lastmod>
    </url>
  —<url>
      <loc>http://localhost:34061/home/list</loc>
    </url>
  </urlset>
```

YÊU CẦU: Tạo sitemap với dữ liệu được lấy từ CSDL tương tự như hình



https://fptshop.com.vn/sitemap-may-tinh-bang_1.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
–<urlset>
  –<url>
      <loc>https://fptshop.com.vn/may-tinh-bang</loc>
      <changefreq>hourly</changefreq>
      <priority>0.9</priority>
    </url>
  –<url>
    –<loc>
        https://fptshop.com.vn/may-tinh-bang/masstel-tab-10-ultra
      </loc>
      <changefreq>daily</changefreq>
      <priority>0.9</priority>
    –<image:image>
      –<image:loc>
          https://fptshop.com.vn/Uploads/Uploads/Originals/2020/10/28//637394760405298535_masstel-tab-10-ultra-vang-dd.png
        </image:loc>
      </image:image>
    </url>
  –<url>
    –<loc>
        https://fptshop.com.vn/may-tinh-bang/huawei-matepad-5g
      </loc>
      <changefreq>daily</changefreq>
      <priority>0.9</priority>
    –<image:image>
      –<image:loc>
          https://fptshop.com.vn/Uploads/Uploads/Originals/2020/9/29//637370173845601657_huawei-matepad-5g-dd.png
        </image:loc>
      </image:image>
    </url>
  –<url>
    –<loc>
        https://fptshop.com.vn/may-tinh-bang/samsung-galaxy-tab-active3
      </loc>
      <changefreq>daily</changefreq>
      <priority>0.9</priority>
    –<image:image>
      –<image:loc>
          https://fptshop.com.vn/Uploads/Uploads/Originals/2020/9/29//637370162927100257_samsung-galaxy-tab-active3-dd.png
        </image:loc>
      </image:image>
    </url>
```