POC BackGroundProcess

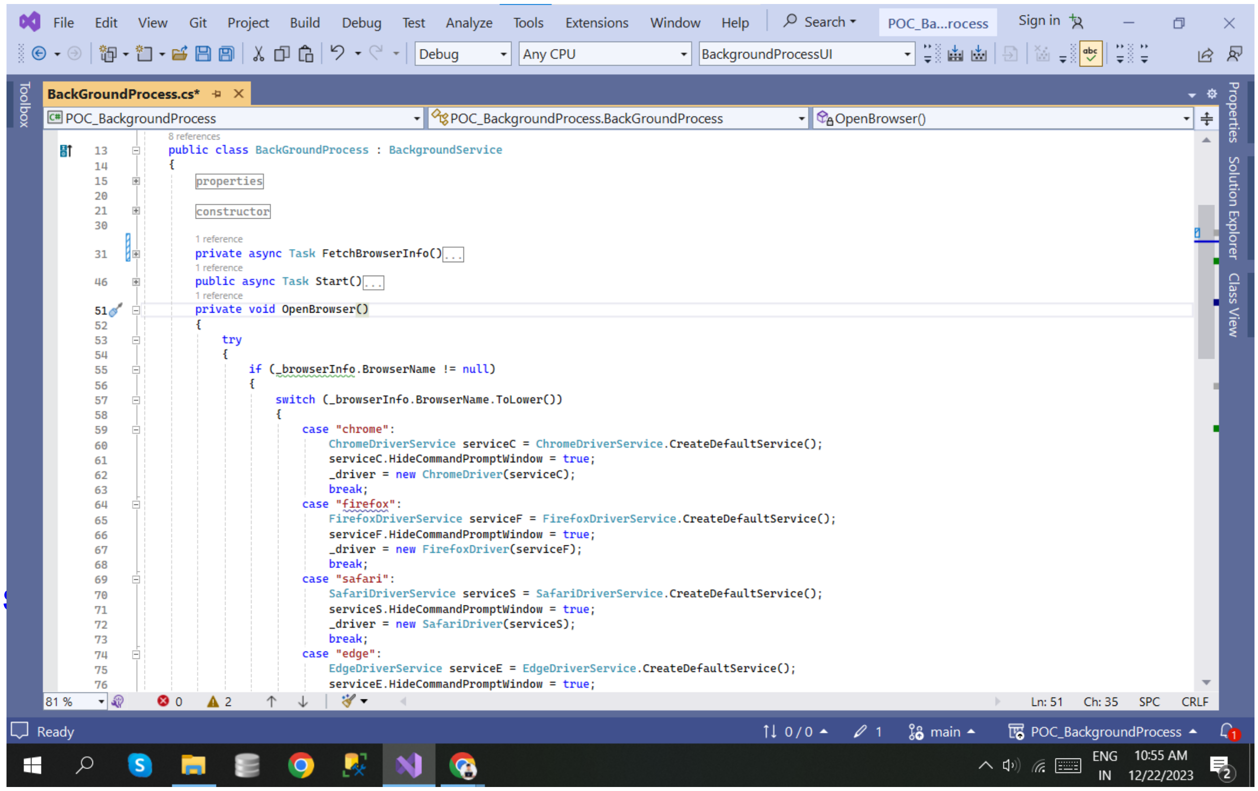
Different ways to create a background process in .net core.

1. Hosted Services
    - A built in feature.
    - The IHostedService interface is used for the implementation of Hosted services.
    - Hosted services can be registered in startup/program.cs file.
2. Quartz.NET
    - a powerful and feature-rich open-source scheduling library.
    - It allows us to schedule background tasks with cron expressions.
    - Quartz.NET can be used in combination with ASP.NET Core to create sophisticated job scheduling.
    - To use this, we need to install the Quartz.AspNetCore NuGet package and set up the scheduler in your Startup/Program.cs file.
3. Hangfire
    - Hangfire is a library that simplifies background processing in .NET applications.
    - It provides a dashboard for monitoring background tasks.
    - To use this, we need to install the Hangfire.AspNetCore NuGet package and configure it in your Startup/Program.cs file

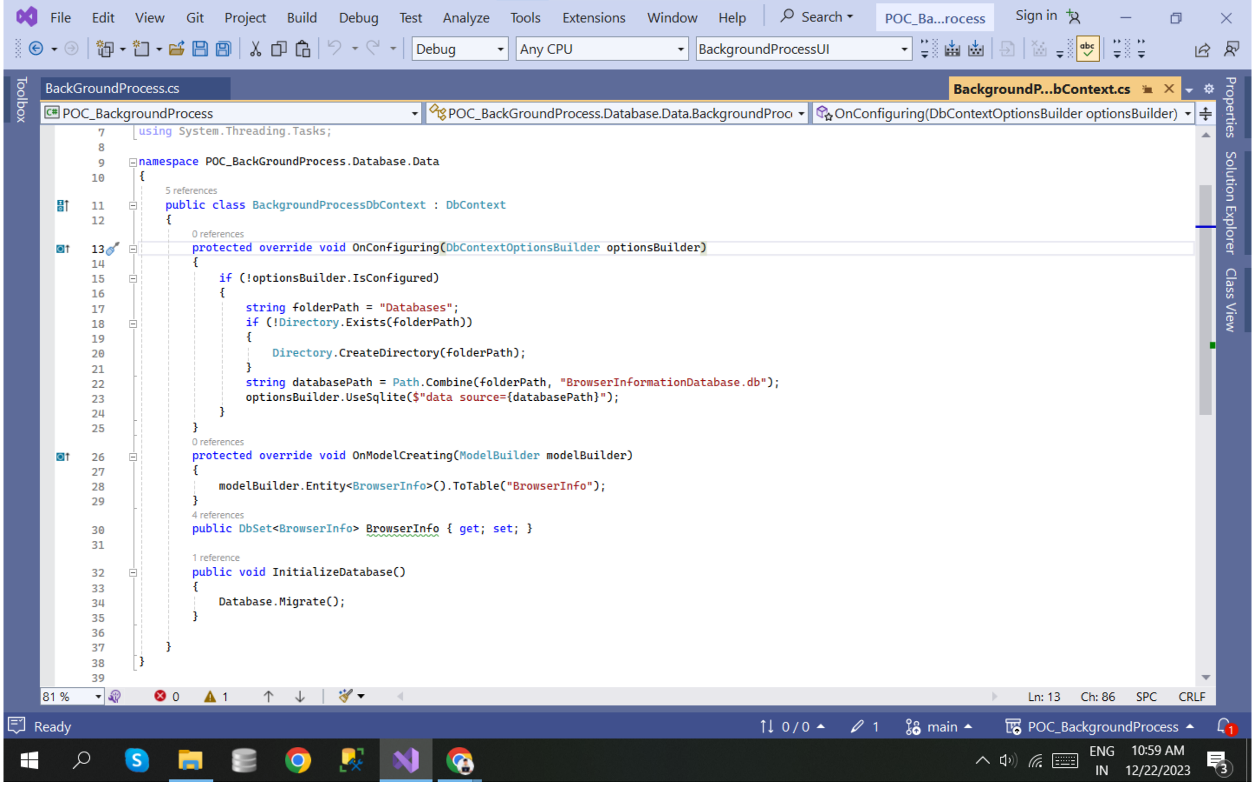
Steps that I followed to implement this task :

- Created a BackgroundProcess using Hosted Service method.
- Created SQLite database.
- Created WinForm for user interface.

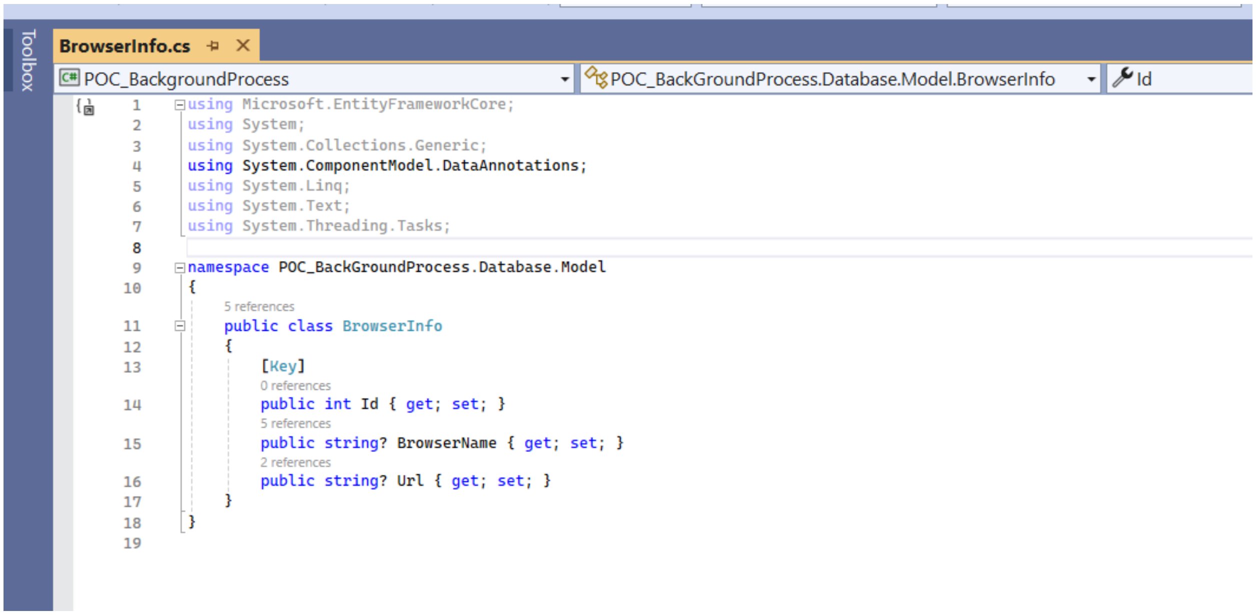
BackgroundProcess :



```
8 references
public class BackGroundProcess : BackgroundService
{
    properties

    constructor

    1 reference
    private async Task FetchBrowserInfo()...
    1 reference
    public async Task Start()...
    1 reference
    private void OpenBrowser()
    {
        try
        {
            if (_browserInfo.BrowserName != null)
            {
                switch (_browserInfo.BrowserName.ToLower())
                {
                    case "chrome":
                        ChromeDriverService serviceC = ChromeDriverService.CreateDefaultService();
                        serviceC.HideCommandPromptWindow = true;
                        _driver = new ChromeDriver(serviceC);
                        break;
                    case "firefox":
                        FirefoxDriverService serviceF = FirefoxDriverService.CreateDefaultService();
                        serviceF.HideCommandPromptWindow = true;
                        _driver = new FirefoxDriver(serviceF);
                        break;
                    case "safari":
                        SafariDriverService serviceS = SafariDriverService.CreateDefaultService();
                        serviceS.HideCommandPromptWindow = true;
                        _driver = new SafariDriver(serviceS);
                        break;
                    case "edge":
                        EdgeDriverService serviceE = EdgeDriverService.CreateDefaultService();
                        serviceE.HideCommandPromptWindow = true;
```

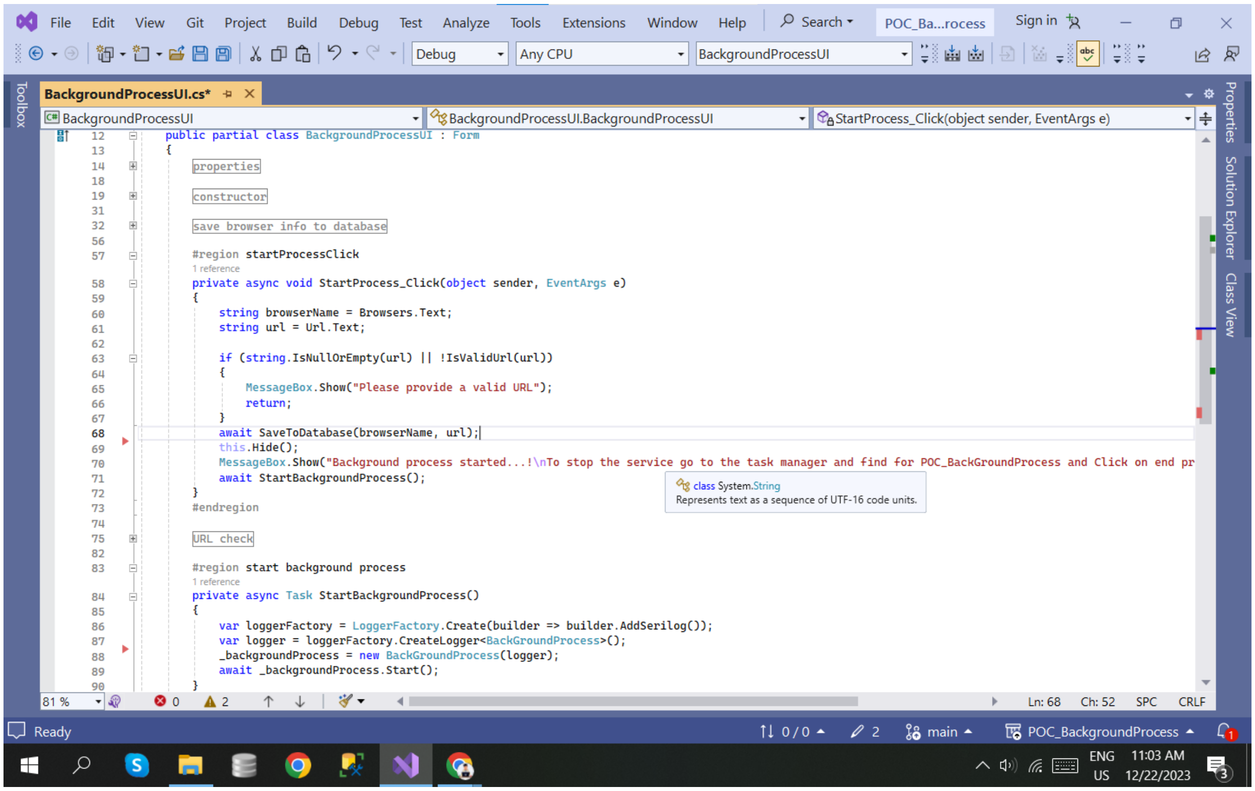


```
using System.Threading.Tasks;

namespace POC_BackGroundProcess.Database.Data
{
    5 references
    public class BackgroundProcessDbContext : DbContext
    {
        0 references
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            if (!optionsBuilder.IsConfigured)
            {
                string folderPath = "Databases";
                if (!Directory.Exists(folderPath))
                {
                    Directory.CreateDirectory(folderPath);
                }
                string databasePath = Path.Combine(folderPath, "BrowserInformationDatabase.db");
                optionsBuilder.UseSqlite($"data source={databasePath}");
            }
        }
        0 references
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<BrowserInfo>().ToTable("BrowserInfo");
        }
        4 references
        public DbSet<BrowserInfo> BrowserInfo { get; set; }

        1 reference
        public void InitializeDatabase()
        {
            Database.Migrate();
        }

    }
}
```



```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace POC_BackGroundProcess.Database.Model
{
    5 references
    public class BrowserInfo
    {
        [Key]
        0 references
        public int Id { get; set; }
        5 references
        public string? BrowserName { get; set; }
        2 references
        public string? Url { get; set; }
    }
}
```

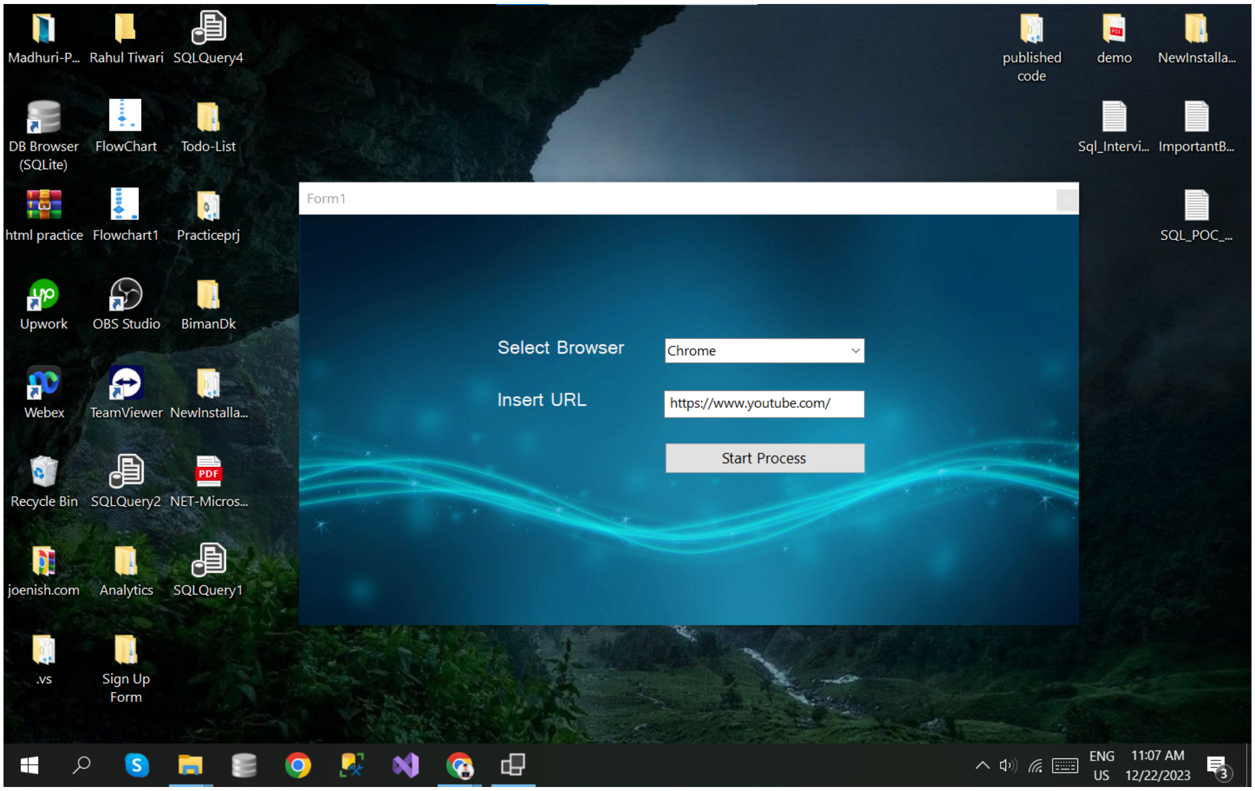
WinForm for User Interface :



```
public partial class BackgroundProcessUI : Form
{
    properties

    constructor

    save browser info to database

    #region startProcessClick
    1 reference
    private async void StartProcess_Click(object sender, EventArgs e)
    {
        string browserName = Browsers.Text;
        string url = Url.Text;

        if (string.IsNullOrEmpty(url) || !IsValidUrl(url))
        {
            MessageBox.Show("Please provide a valid URL");
            return;
        }
        await SaveToDatabase(browserName, url);
        this.Hide();
        MessageBox.Show("Background process started...!\nTo stop the service go to the task manager and find for POC_BackGroundProcess and Click on end pr
        await StartBackgroundProcess();
    }
    #endregion

    URL check

    #region start background process
    1 reference
    private async Task StartBackgroundProcess()
    {
        var loggerFactory = LoggerFactory.Create(builder => builder.AddSerilog());
        var logger = loggerFactory.CreateLogger<BackGroundProcess>();
        _backgroundProcess = new BackGroundProcess(logger);
        await _backgroundProcess.Start();
    }
```

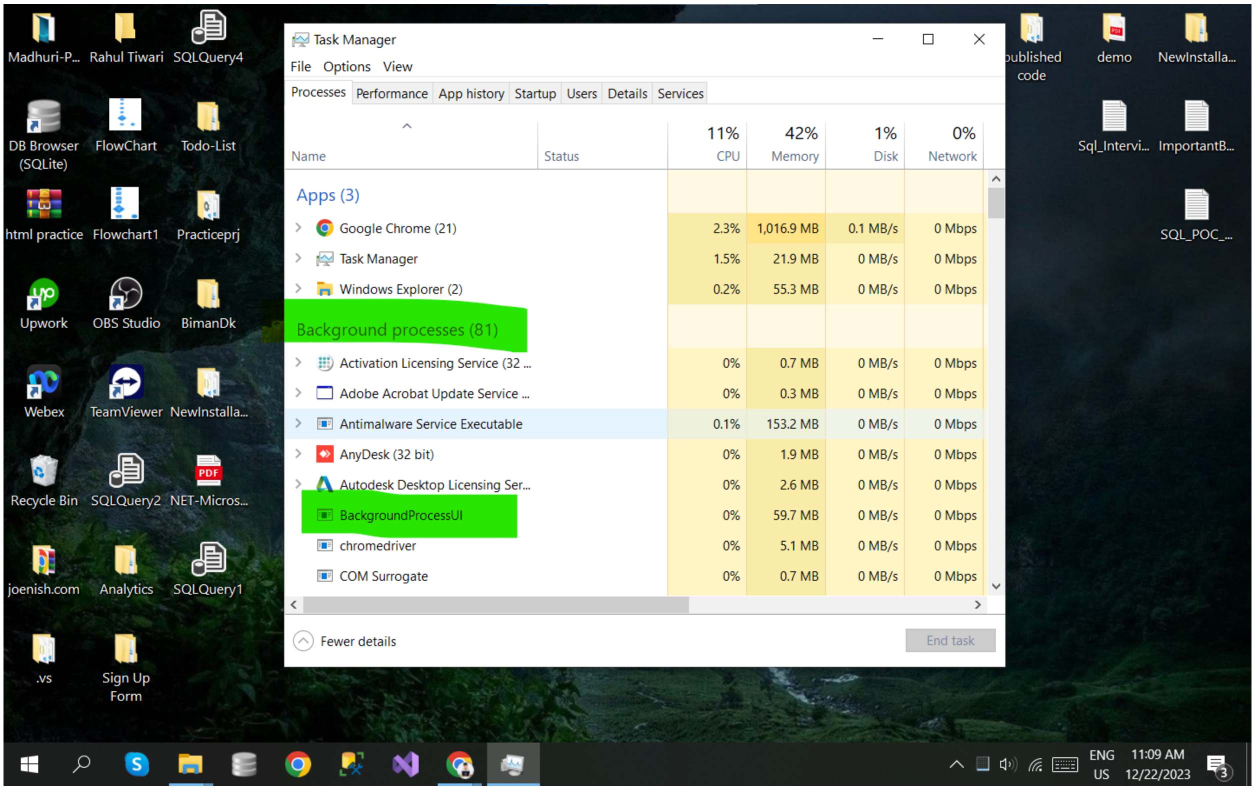
class System.String
Represents text as a sequence of UTF-16 code units.

Output Snapshots :



Select Browser: Chrome

Insert URL: https://www.youtube.com/



Name	Status	11% CPU	42% Memory	1% Disk	0% Network
Apps (3)					
Google Chrome (21)		2.3%	1,016.9 MB	0.1 MB/s	0 Mbps
Task Manager		1.5%	21.9 MB	0 MB/s	0 Mbps
Windows Explorer (2)		0.2%	55.3 MB	0 MB/s	0 Mbps
Background processes (81)					
Activation Licensing Service (32 ...		0%	0.7 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service ...		0%	0.3 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0.1%	153.2 MB	0 MB/s	0 Mbps
AnyDesk (32 bit)		0%	1.9 MB	0 MB/s	0 Mbps
Autodesk Desktop Licensing Ser...		0%	2.6 MB	0 MB/s	0 Mbps
BackgroundProcessUI		0%	59.7 MB	0 MB/s	0 Mbps
chromedriver		0%	5.1 MB	0 MB/s	0 Mbps
COM Surrogate		0%	0.7 MB	0 MB/s	0 Mbps

Sharing the Repo link : https://github.com/JoenishEmp15/POC_BackgroundProcessTask.git