Sure, I'll guide you through creating a new .NET console application for uploading files to S3 using Visual Studio Code. Here are the detailed steps:

**Step 1: Install .NET SDK**

1. **Download and Install .NET SDK:**
   * Go to the [.NET download page](https://dotnet.microsoft.com/download) and download the latest version of the .NET SDK.
   * Follow the installation instructions for your operating system.
2. **Verify the installation:**
   * Open a terminal (Command Prompt, PowerShell, or any terminal in VS Code) and run:

dotnet –version

**Step 2: Create a New .NET Console Application**

1. **Open Visual Studio Code.**
2. **Open the Terminal in Visual Studio Code:**
   * You can open the terminal by navigating to View > Terminal or using the shortcut Ctrl+`.
3. **Create a new directory for your project:**

mkdir S3Uploader

cd S3Uploader

1. dotnet new console --name S3Uploader. cd S3Uploader

**Step 3: Install Necessary Packages**

1. dotnet add package AWSSDK.S3
2. Install Configuration Packages:
   1. dotnet add package Microsoft.Extensions.Configuration
   2. dotnet add package Microsoft.Extensions.Configuration.Binder
   3. dotnet add package Microsoft.Extensions.Configuration.Json

**Step 4: Set Up the Project**

**Update S3Uploader.csproj file:**

Open the S3Uploader.csproj file and ensure it looks like this:

<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>

    <OutputType>Exe</OutputType>

    <TargetFramework>net8.0</TargetFramework>

    <ImplicitUsings>enable</ImplicitUsings>

    <Nullable>enable</Nullable>

    <SelfContained>true</SelfContained>

    <RuntimeIdentifier>win-x64</RuntimeIdentifier>

  </PropertyGroup>

  <ItemGroup>

    <PackageReference Include="AWSSDK.S3" Version="3.7.309.10" />

    <PackageReference Include="Microsoft.Extensions.Configuration" Version="8.0.0" />

    <PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="8.0.0" />

    <PackageReference Include="Microsoft.Extensions.Configuration.Binder" Version="8.0.1" />

    <PackageReference Include="AWSSDK.Extensions.NETCore.Setup" Version="3.7.301" />

  </ItemGroup>

</Project>

**Step 5: Create the Configuration File**

1. Create appsettings.json in the project root:
2. {
3. "AWS": {
4. "Profile": "S3",
5. "Region": "us-east-2",
6. "AccessKey": "AKIAZQ3DS3XP26EOBBHB",
7. "SecretKey": "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",
8. "BucketName": "authbillpro"
9. }
10. }

aws configure --profile S3 “command line to set the above app settings to match each other.”

**Step 6: Update the Program File**

using Amazon.S3;

using Amazon.S3.Transfer;

using Microsoft.Extensions.Configuration;

class Program

{

    static async Task Main(string[] args)

    {

        if (args.Length < 1)

        {

            Console.WriteLine("Please provide the folder path to upload.");

            return;

        }

        string folderPath = args[0];

        // Load configuration

        var builder = new ConfigurationBuilder()

            .SetBasePath(Directory.GetCurrentDirectory())

            .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);

        IConfiguration config = builder.Build();

        var awsOptions = config.GetAWSOptions(); // Ensure this is recognized

        var s3Client = awsOptions.CreateServiceClient<IAmazonS3>();

        var transferUtility = new TransferUtility(s3Client);

        try

        {

            await transferUtility.UploadDirectoryAsync(folderPath, config["AWS:BucketName"]);

            Console.WriteLine("Folder uploaded successfully.");

        }

        catch (Exception e)

        {

            Console.WriteLine("Folder upload failed: " + e.Message);

        }

    }

}

**Step 7: Build and Run the Project**

dotnet build

dotnet run "C:\path\to\your\folder"