**File Upload in ASP.NET Core MVC**

In most of the applications we have an option to upload profile photos which will be displayed to identify the user. For example, Facebook, WhatsApp, Twitter and many more. This is a very basic feature which is to have an option to upload the photo which will be saved and displayed in an application

ASP.NET core MVC provides multiple ways to upload files, process and save them. It supports uploading single and multiple files.

ASP.NET core MVC provides two general ways to upload files.

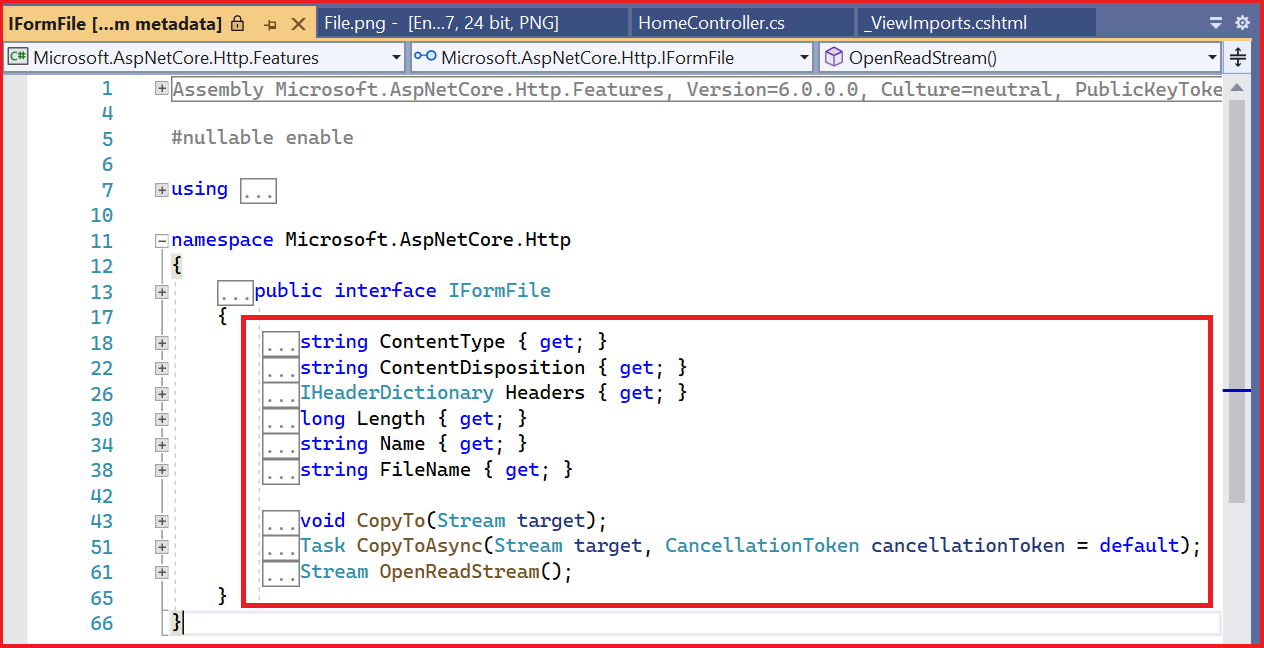
1. Buffering
2. Streaming

**Buffering**

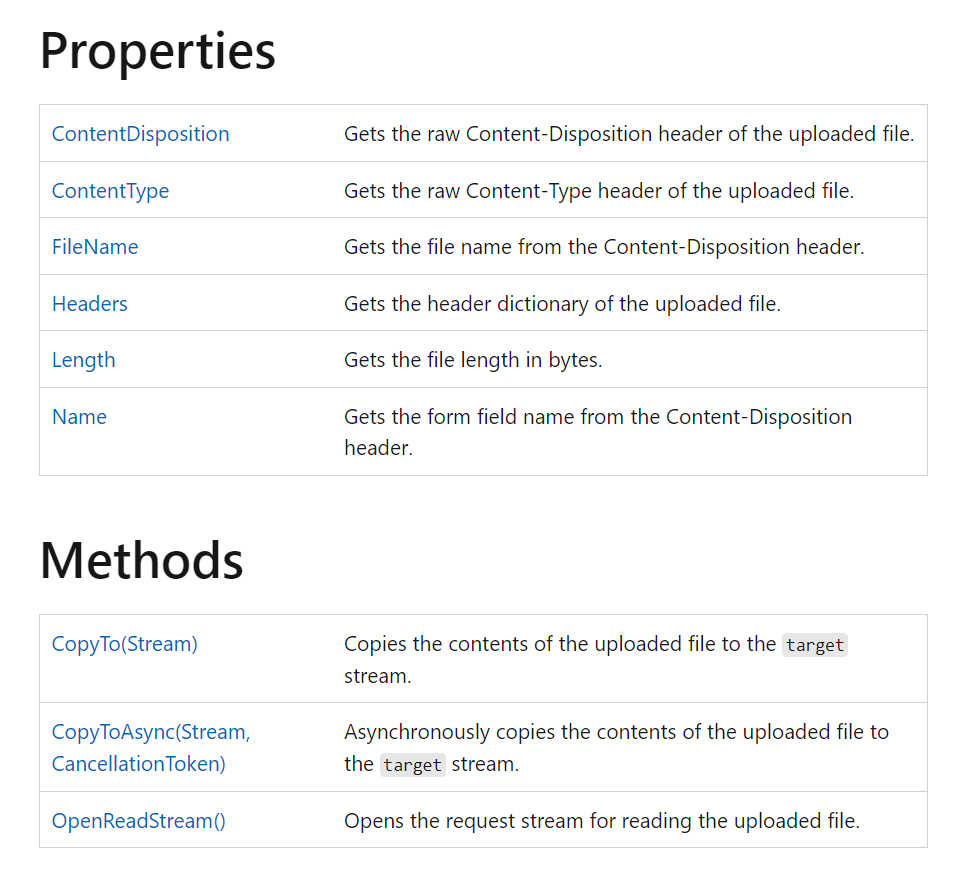
In this approach the entire file is read into an “**IFormFile**”. IFormFile is a C# representation of the file used to process or save the file. “IFormFile” represents a file sent with the HTTP Request.

Use this approach to upload files of relatively smaller sizes.

All the properties and the methods supported by the “IFormFile” interface are highlighted in the below screenshot



The details about the properties and the methods of the “IFormFile” interface as given by Microsoft are shown below



**Streaming**

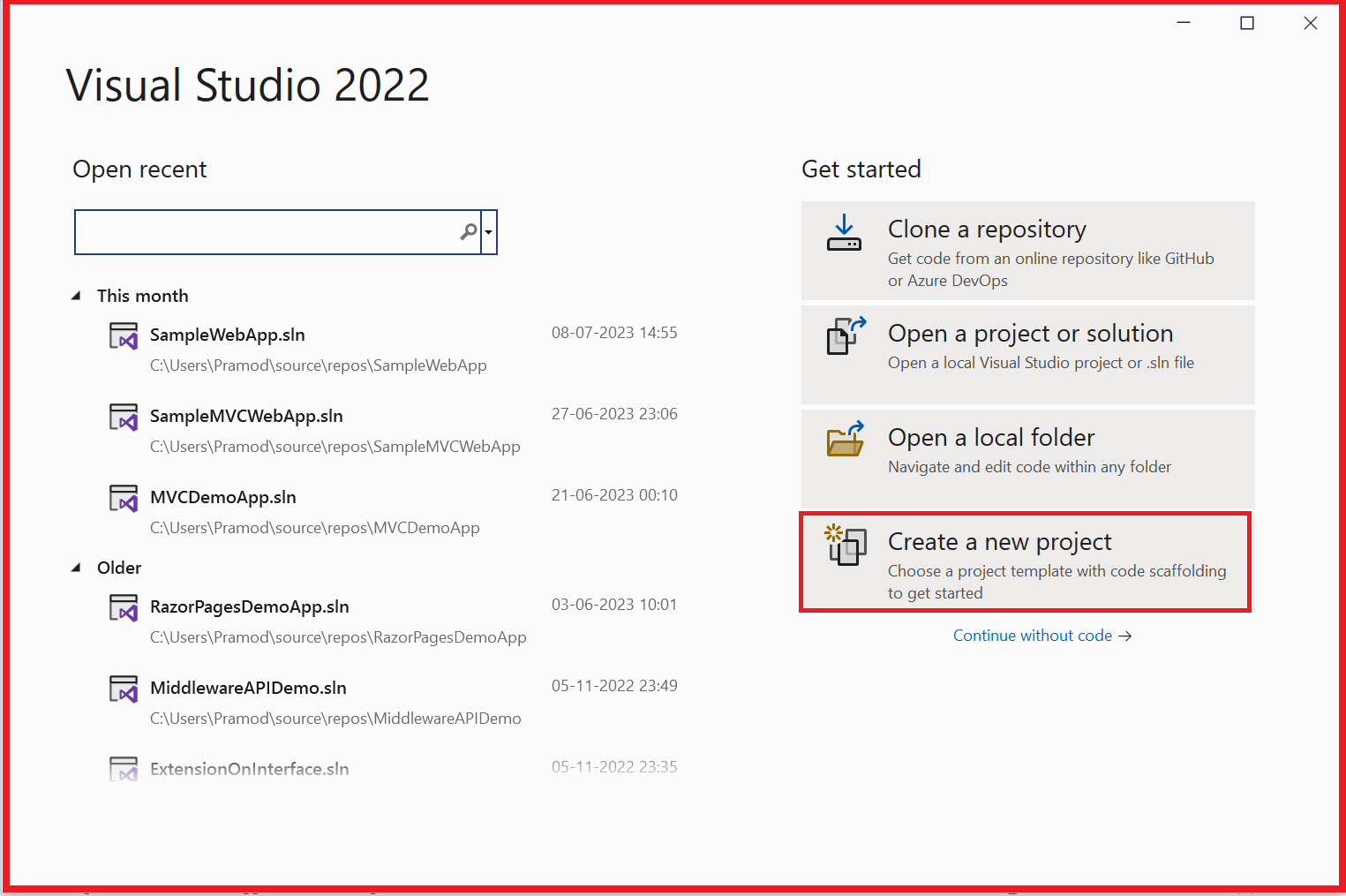
This approach is useful when you want to handle large files efficiently without loading the entire file into memory. Instead of using the “IFormFile” interface, you can directly read the uploaded file as a stream and process it accordingly.

In this article we will see how to implement the file upload feature to upload a single file and how to process and save the file using buffering approach in ASP.NET 6 MVC application.

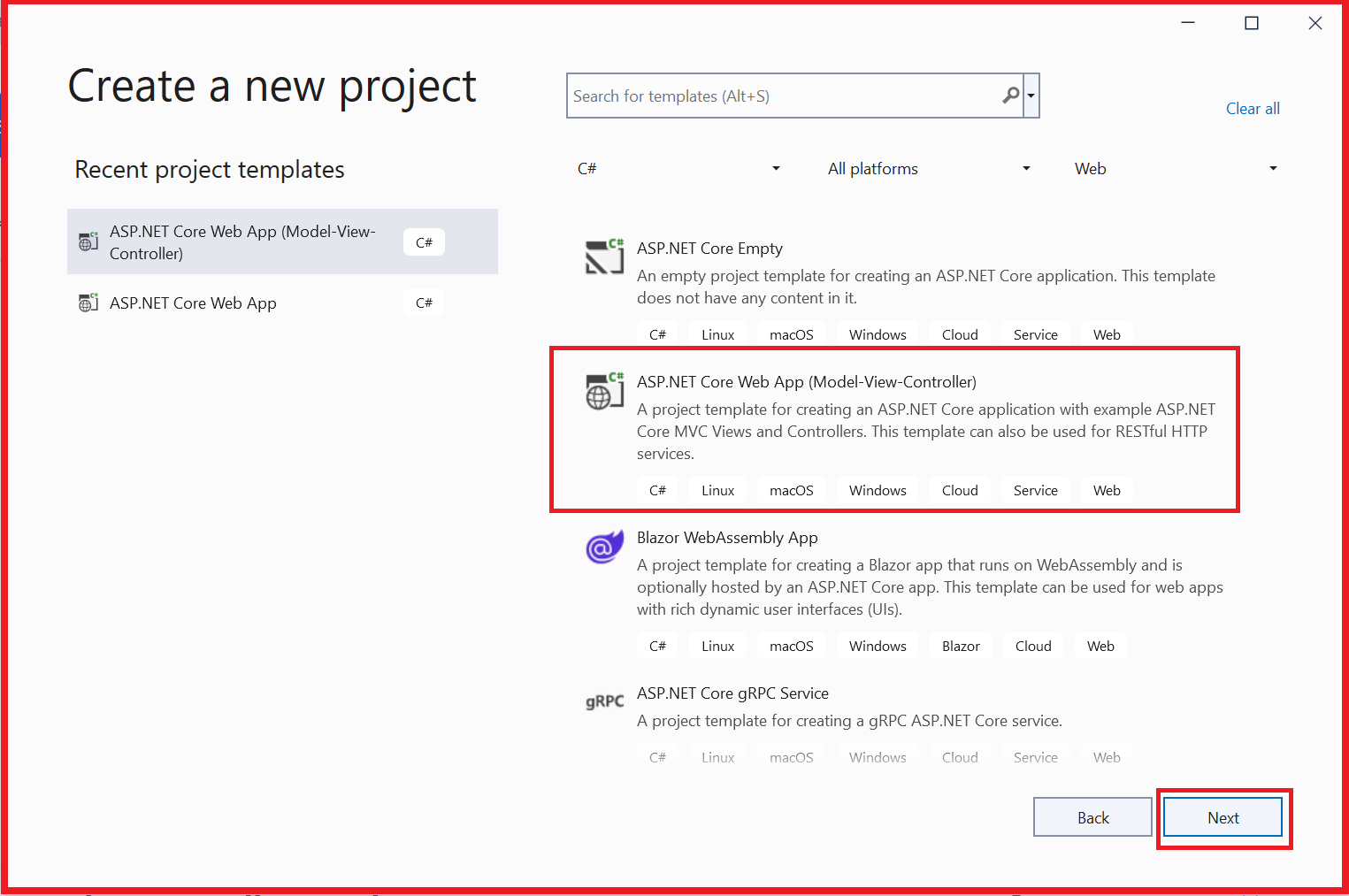
In this example we add a form control to the index.cshtml file within the Views --> Home folder. Add the file upload control and submit button within the form. In the home controller, add an action method for form submit. In the action method we save the uploaded file.

**Let's create the ASP.NET 6 MVC application using Visual Studio 2022**

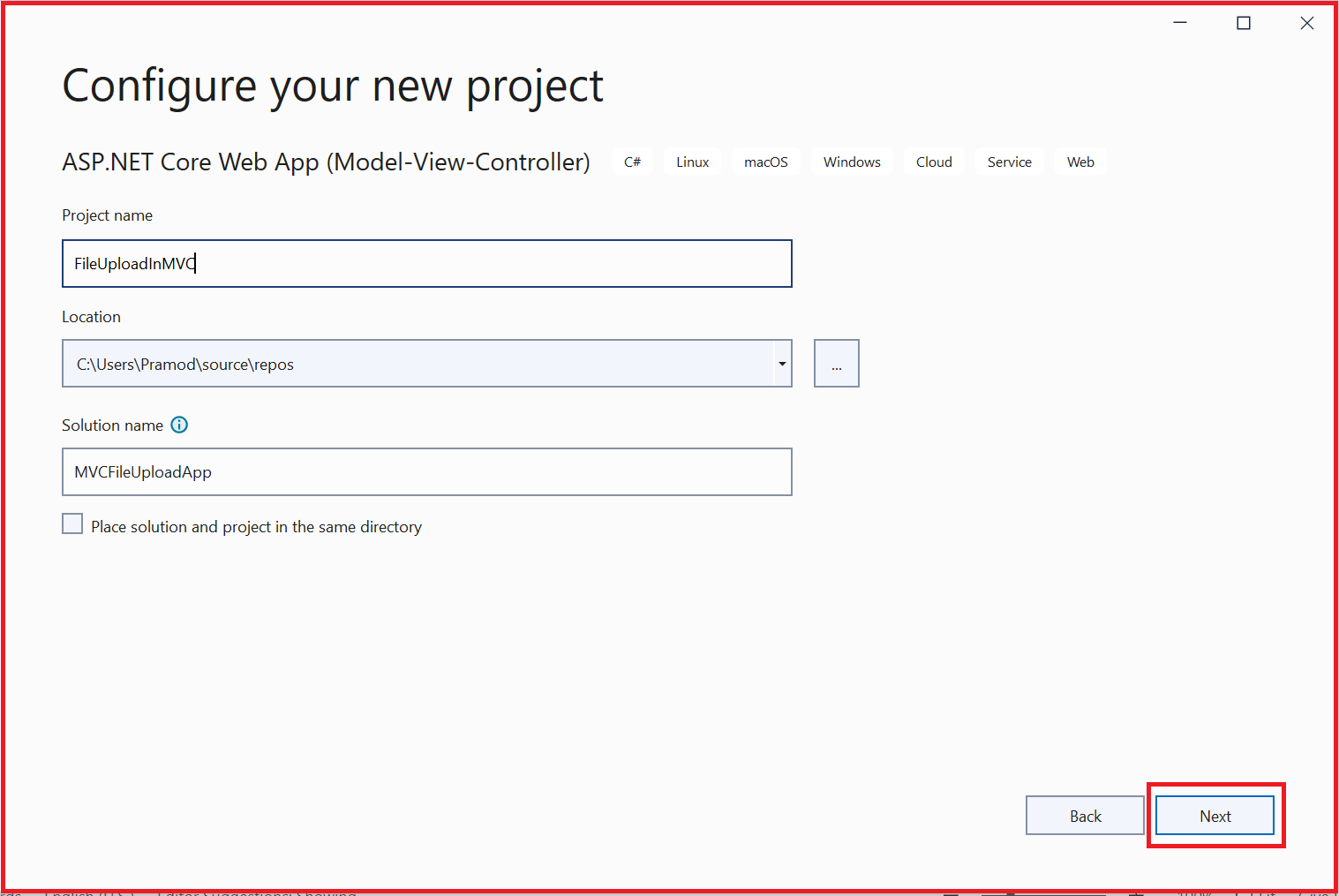
1. Launch Visual Studio 2022 and select “Create a new project”



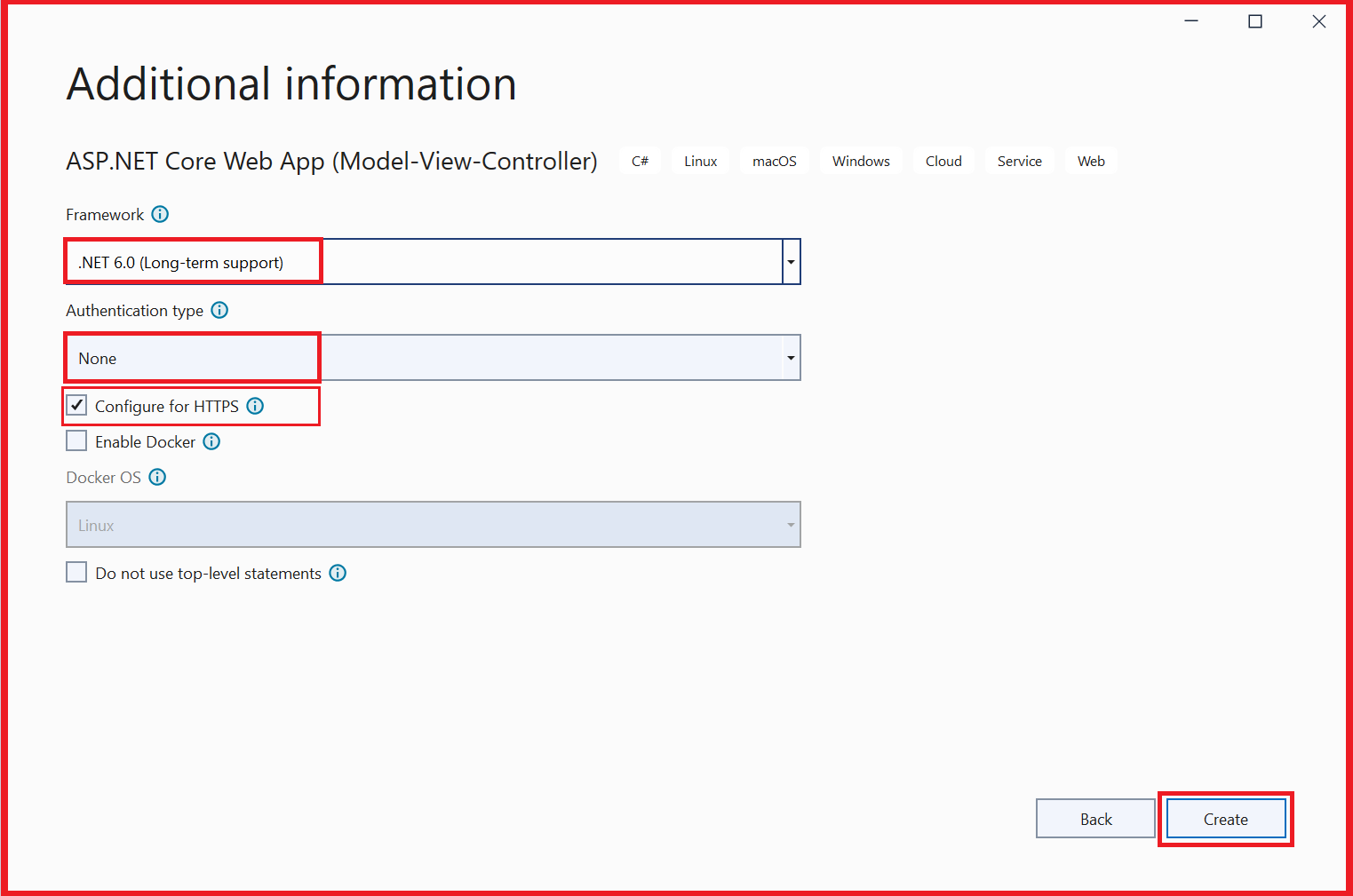
1. Select “ASP.NET Core Web App (Model-View-Controller)” project template and click on Next



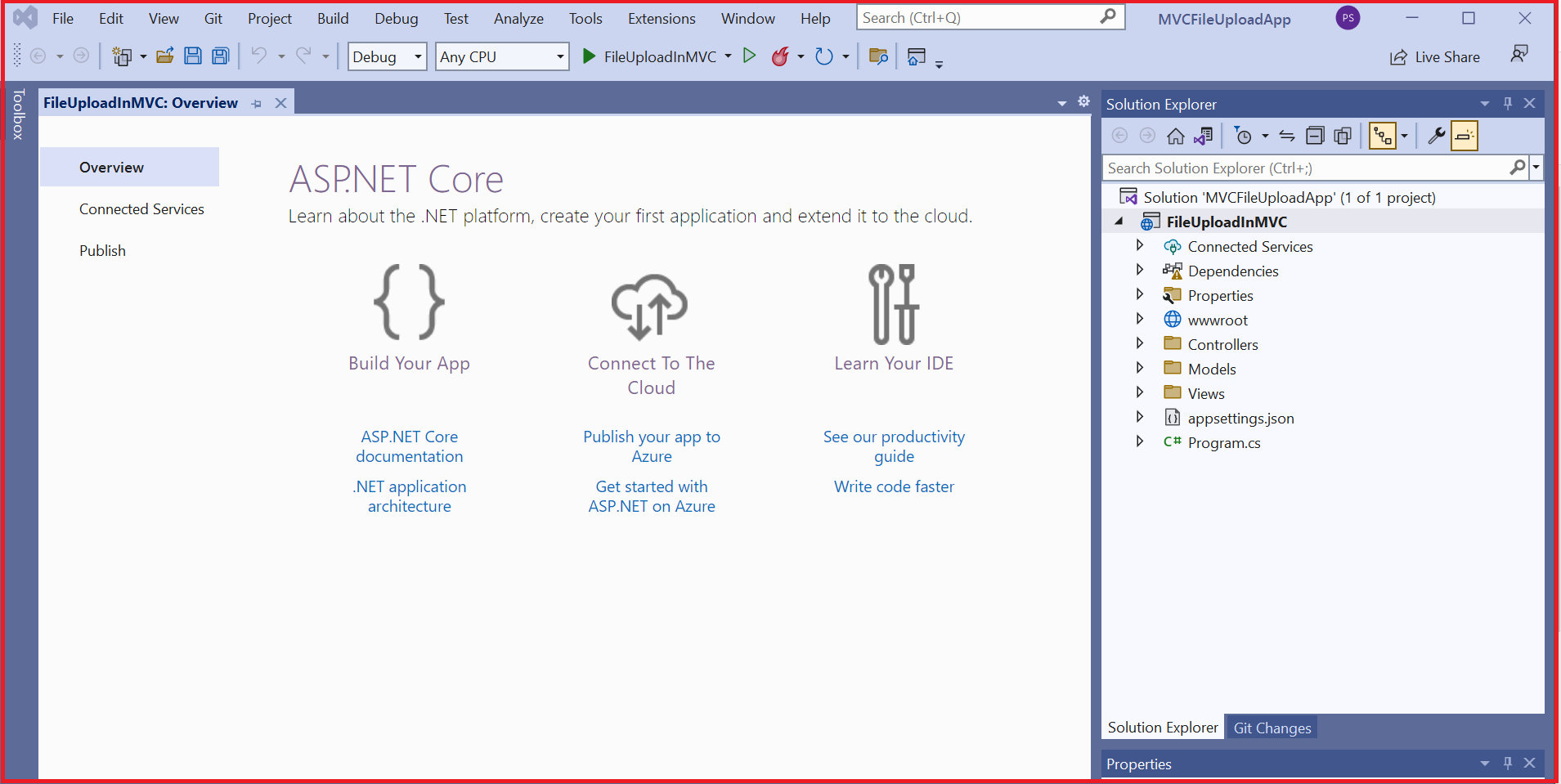
1. Give meaningful name to the Project, select the location or the path where the project needs to be created and give a meaningful name to the solution and click on Next



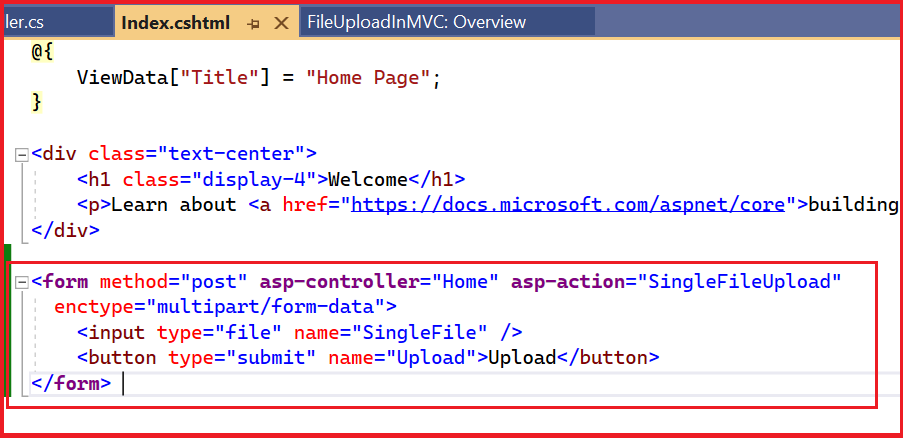
1. Make sure Framework selected is “.NET 6.0 (Long-term support)”, Authentication type is “None”, “Configure for HTTPS” is checked. Click on Create



1. ASP.NET 6 MVC project will be created



Let’s update the “Index.cshtml” file which is under “Views” --> “Home” folder to add a form control. The form control encloses an input control of type file and followed by a button control of type submit as shown below



“

<form method="post" asp-controller="Home" asp-action="**SingleFileUpload**"

enctype="multipart/form-data">

<input type="file" name="**SingleFile**" />

<button type="submit" name="Upload">Upload</button>

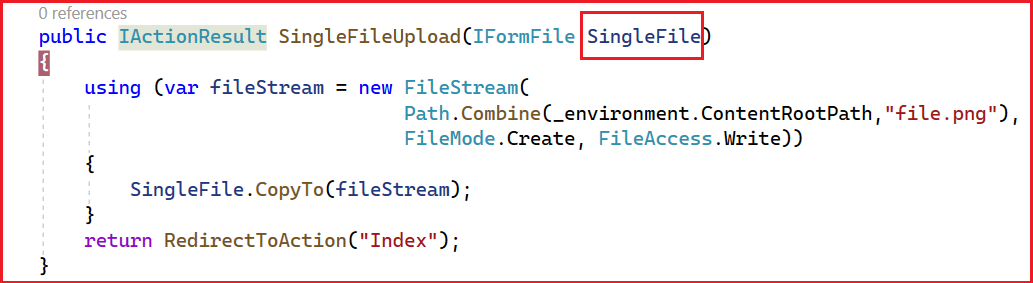
</form>

”

Notice that

1. form control has method = post, asp-controller="**Home**" asp-action="**SingleFileUpload**" attributes.  
   On submitting the form, the “**SingleFileUpload**” action method in the “**Home**” controller will be executed.
2. The name attribute of the file type input is **SingleFile**

Let’s update the “Home” controller to add a new action method “**SingleFileUpload**”, In this action method we save the file to the application root directory using the **CopyTo(Stream)** method of **IFormFile** interface as shown below



“

public IActionResult **SingleFileUpload**(**IFormFile** **SingleFile**)

{

using (var fileStream = new FileStream(

Path.Combine(**\_environment.ContentRootPath**,"file.png"),

FileMode.Create, FileAccess.Write))

{

SingleFile.CopyTo(fileStream);

}

return RedirectToAction("Index");

}

”

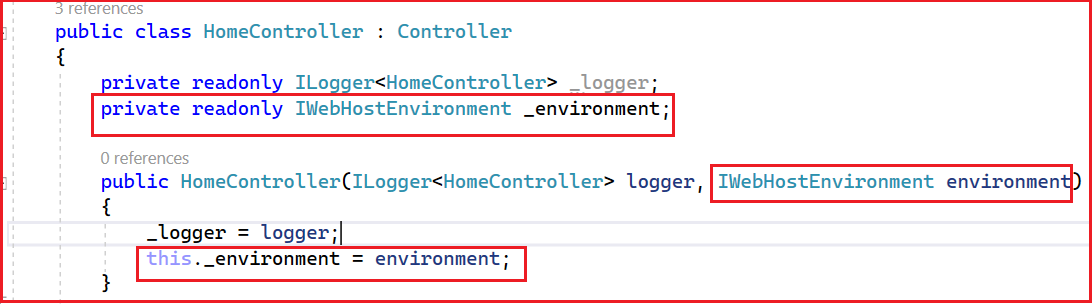
Notice that

1. We have an input parameter of type “**IFormFile**” for the action method.
2. The name of the input parameter is “**SingleFile**”. The name of the input parameter must match with the name of the file type input control that we have added



1. We are creating a file stream which takes in file path, file mode and file access parameters.
   1. File path – In the content root directory, we are saving the uploaded file as “file.png”. Here we get the content root directory using the **ContentRootPath** property of IWebHostEnvironment
   2. File mode – Create
   3. File Access – Write
2. Using the “CopyTo” method of IFormFile method we are saving the file by passing the file stream

To get the content root path we are using “IWebHostEnvironment” and we are getting the instance of it using dependency injection as shown below



“

private readonly IWebHostEnvironment \_environment;

public HomeController(ILogger<HomeController> logger, IWebHostEnvironment environment)

{

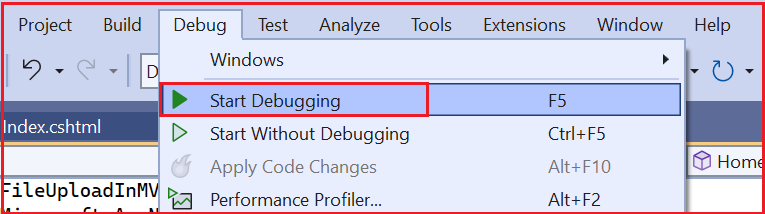
\_logger = logger;

this.\_environment = environment;

}

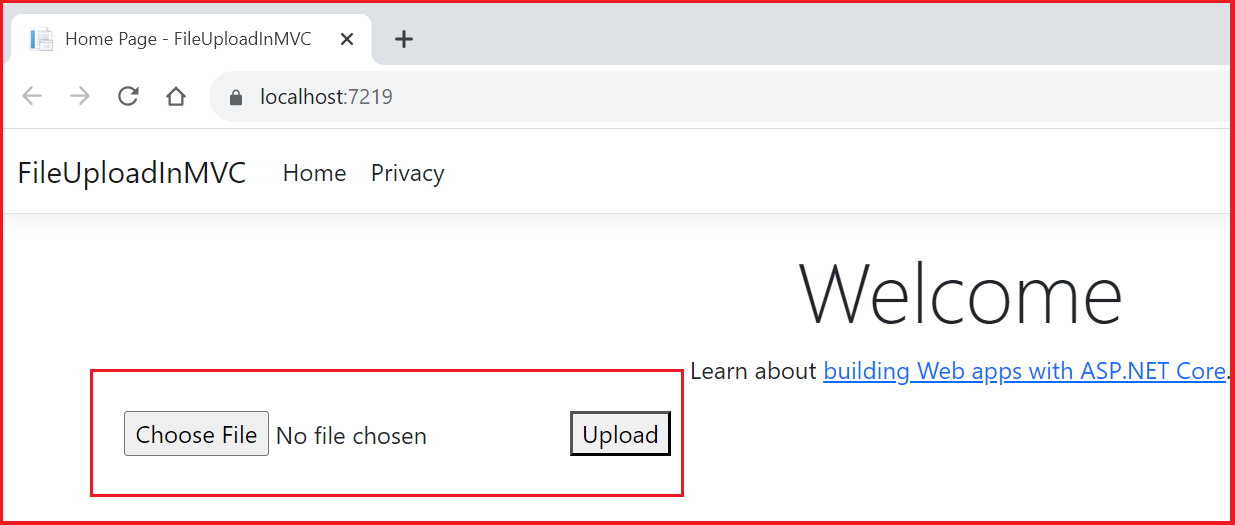
“

Re-build the application and start debugging

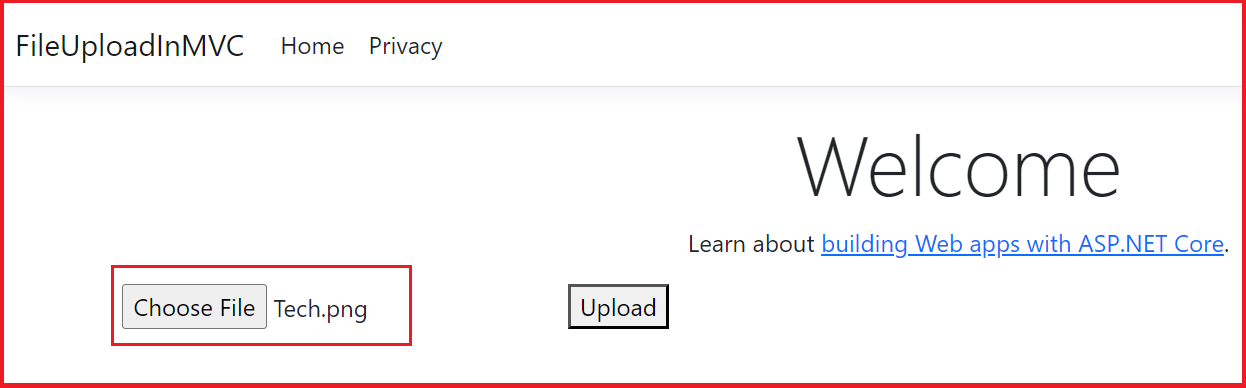


Application will be launched, and index view of the home controller will be displayed as shown below.

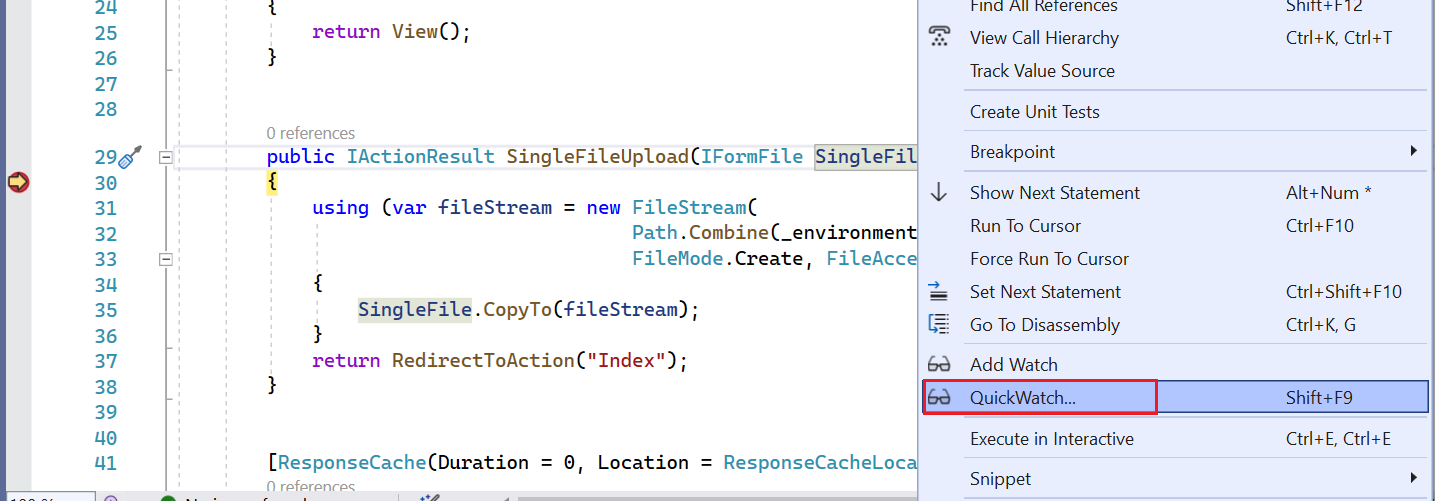
The index view has file type input control and a submit button which we added as shown below



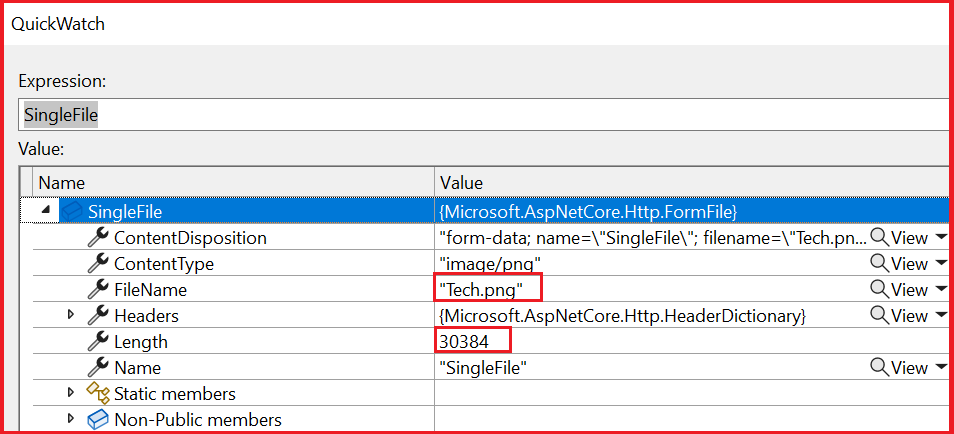
Click on Choose File and select a “\*.png” file. After selecting, the file name will be displayed as shown below



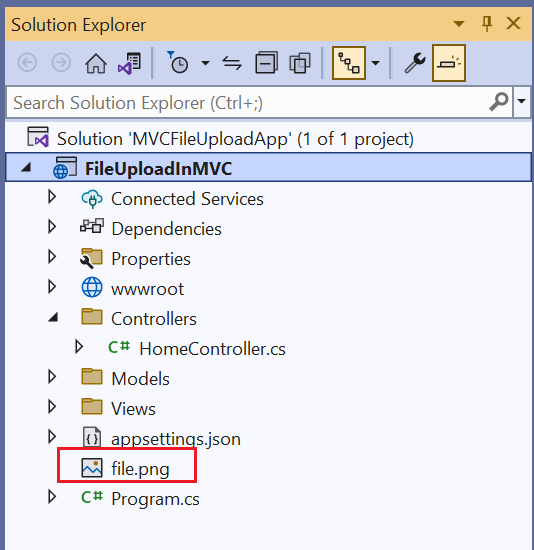
Click on the Upload button. The “SingleFileUpload” action method in the HomeController will be hit. Have a breakpoint at the entry of action method. Quick watch on the input parameter.



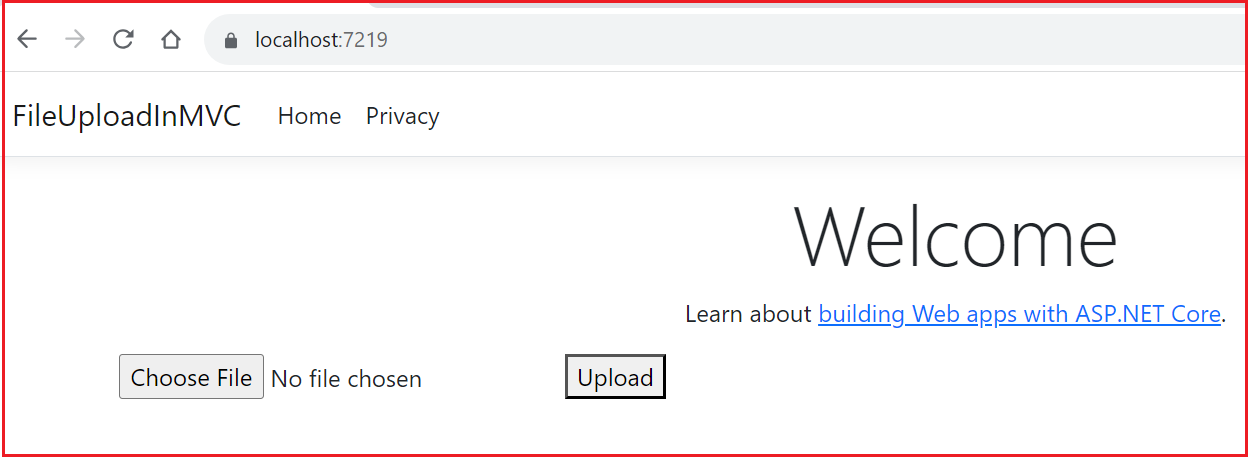
Notice it contains all the details about the file we just uploaded as shown below



Click on F5 to continue with debugging, once the file is copied to the specified path, index page will be reloaded, and we should be able to see the uploaded file in the solution explorer as shown below



After saving the file, we are re-directing the application again to “index” page.



In this article we discussed the file upload feature in asp.net core MVC. We created a sample asp.net 6 MVC application and learnt how to upload a single file, process and save it.

Let me know your thoughts on this in the comment section. In the next article we will learn about uploading multiple files in asp.net core MVC