Razor Page is similar to the HTML page but it loads data easily. A Razor Page is almost the same as ASP.NET MVC’s view component. It has basically the syntax and functionality same as MVC.

The basic difference between Razor pages and MVC is that the model and controller code is also added within the Razor Page itself

**#1 — Razor Pages is Better Organized**

If you’ve ever used MVC for any kind of web development, then you probably know that how much time it takes to code an entire app. Creating dynamic routes, naming things properly, and hundred other stuff consumes a lot of time.

Razor Pages, on the other hand, is more organized compared to MVC.

In Razor Pages, the files are basically more organized. You have a Razor View and the entire code behind a file, same way the old ASP.NET WebForms did.

#### 2 — Single Responsibility

Again, if you have ever used an MVC framework before, you have probably seen that there are some huge controller classes which are generally filled with a lot of various actions. These classes are like a virus which grows bigger and bigger as new things are added.

But, in Razor Pages, each app page is self-contained with its own view and code organized together, which as a results, is less complex than the MVC.

### A Quick Comparison of How Requests Are Handled in Both

We explained in above points that building a web application using ASP.NET Core Razor Pages is less complex than the MVC. Here, we will demonstrate that with action.

#### Let’s start with MVC

Here’s a quick overview of how MVC handles the requests.

Diagram

Description automatically generated

As you can see, routing is the key to how MVC decides to handle requests. The default configuration for routing is the combination of action and controller names.

So if you request for /staff/index, then it will route you the action named Index on the StaffController class.

But, it can be customized or configured to route any request to any controller with a block of code.

#### Now Compare the Same with Razor Pages

Here’s a quick overview of how Razor Pages handle the requests.

Diagram

Description automatically generated

The difference between the two is that in Razor Pages, when you make a request, the default routing configuration will find a Razor Page for that specific request in the Pages folder.

Suppose you make a request for /contact/, then ASP.NET Core will look for a page having same name that you used in request and will route you directly to it.

That means, a request to /contact/ will route you to Contact.cshtml

And for any .cshtml file to be considered as a Razor Page, it must be placed in the Pages folder and also contain @Page in its markup.

This way, the Razor Page will then act as a controller action. Now comparing this to MVC, the configuring the custom route will be less complex as there will no extra coding involved.