



Please click “RSVP Now” button click.

A screenshot of a web application interface showing the RSVP form. The top navigation bar is black and contains 'Home List of Guests 0:07:42'. The main content area has a blue header with the text 'RSVP'. Below the header is a form with four fields: 'Your name:' with a yellow input box, 'Your age:' with a yellow input box, 'Arrival time:' with a yellow input box containing '--:-- --' and a clock icon, and 'Request parking?' with a yellow dropdown menu showing 'Choose an option' and a downward arrow. Below the form is a blue button with the text 'Submit RSVP'. At the bottom, a black footer bar contains the text 'Copyright © 202 Halloween Party' and 'cop4020@email.com'.

Then you can see this screen.

```

0 references | 0 requests | 0 exceptions
public IActionResult Index()
{
    return View();
}

```

Controller part.

You can see public function is running. Because , All programs will run the Index function first. So, in this Home Controller class we can know that at first the IActionResult Index() function will be run.

“return View()”.

This means this method return the View file.

A part of View file.

```

<div class="">
|   <label asp-for="Name">Your name:</label>
| </div>
| <div>
|   <input asp-for="Name" style="width: 350px; background-color: #ffffcc;" />
| </div>
-

```

This is name input:

If you clicked submit button, your name is sent model via controller.

```

public ViewResult RsvpForm(GuestResponse guestResponse)
{
    if (ModelState.IsValid)
    {
        Repository.AddResponse(guestResponse);
        return View("Thanks", guestResponse);
    }
}

```

```

public static class Repository
{
    private static List<GuestResponse> responses = new List<GuestResponse>();
    1 reference | 0 exceptions
    public static IEnumerable<GuestResponse> Responses => responses;
}

```

```
public class GuestResponse
{
    [Required(ErrorMessage = "Please enter your name")]
    public string Name { get; set; }
}
```

As you can see, each input parts response according to their value. If you don't fill the username input, you will get "Please enter your name" message.

As you can see the below, all parts are the same functionalities with the username input part. Then you can know all this functionalities.

```
<div class="form-group">
    <label asp-for="guestAge">Your age:</label>
</div>
<div>
    <input asp-for="guestAge" style="width: 350px; background-color: #ffffcc;" />
</div>
```

```
<body>
<h5 class="bg-primary text-white text-center m-2 p-2">RSVP</h5>
<form asp-action="RsvpForm" method="post" class="m-2">
    <div asp-validation-summary="All"></div>
    <div class="">
        <label asp-for="Name">Your name:</label>
    </div>
    <div>
        <input asp-for="Name" style="width: 350px; background-color: #ffffcc;" />
    </div>
    <div class="form-group">
```

```
[HttpPost]
public IActionResult RsvpForm(GuestResponse guestResponse)
{
    if (ModelState.IsValid)
    {
        Repository.AddResponse(guestResponse);
        return View("Thanks", guestResponse);
    }
    else
    {
        return View();
    }
}

public IActionResult ListResponses()
{
    return View(Repository.Responses.Where(r => r.WillAttend == true));
}
```

```
foreach (PartyInvites.Models.GuestResponse r in  
{  
    <tr>  
        <td>r.Name</td>  
        <td>r.guestAge</td>  
        <td>r.Phone</td>  
        <td>r.WillAttend.ToString()</td>  
    </tr>  
}
```

And you can see the foreach in the above picture.

We first get all data from model data that we saved before, and all data is showed with the <td> tags.

That's all!

Please understand the MVC pattern.