MenuItems.cs

```
namespace JoePizza.Models
{
    public class MenuItem
    {
        public int Id { get; set; }
        public string ItemName { get; set; }
        public string Toppings { get; set; }
        public int Price { get; set; }
    }
}
```

MenuItemsController.cs

```
using JoePizza.Models;
using Microsoft.AspNetCore.Mvc;

namespace JoePizza.Controllers
{
    public class MenuItemsController : Controller
    {
        public static List<MenuItem> menu=new List<MenuItem>()
        {
            new MenuItem(){Id=1,ItemName="Margerits Pizza",Toppings="Cheese &
olives",Price=99},
            new MenuItem(){Id=2,ItemName="Corn & Cheese",Toppings="Cheese,Corn",Price=109}

        };

        public IActionResult Index()
        {
            return View(menu);
        }
        [HttpGet]
        public IActionResult Create()
        {
            return View(new MenuItem());
        }
        [HttpPost]
        public IActionResult Create(MenuItem item)
        {
            if(ModelState.IsValid)
            {
                menu.Add(item);
                return RedirectToAction("Index");
            }
            else
            {
                return View(menu);
            }
        }
    }
}
```

Index.cshtml

```
@model IEnumerable<JoePizza.Models.MenuItem>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
```

```
            <th>
                @Html.DisplayNameFor(model => model.ItemName)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Toppings)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Price)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.ItemName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Toppings)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Price)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
                <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
                <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
            </td>
        </tr>
}
    </tbody>
</table>
```

Create.cshtml

```
@model JoePizza.Models.MenuItem

@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>MenuItem</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="ItemName" class="control-label"></label>
                <input asp-for="ItemName" class="form-control" />
                <span asp-validation-for="ItemName" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Toppings" class="control-label"></label>
                <input asp-for="Toppings" class="form-control" />
                <span asp-validation-for="Toppings" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Price" class="control-label"></label>
                <input asp-for="Price" class="form-control" />
                <span asp-validation-for="Price" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```