

Contents

Steps to create a Data Entry Screen in MVC.....	2
1. Create a DataModel class in the Models Folder	2
2. Create a Service Interface and Implementation in the Services Folder	2
3. Add the Service to the Asp.NET Core Services Container	2
4. Create a Controller class with Index Action for GET & Index Action for POST. Remember to add Dependency Injection in the constructor.	2
5. Create a View	3

Steps to create a Data Entry Screen in MVC

1. Create a DataModel class in the Models Folder

```
public class DepositInfo
{
    [Required]
    [Range(10000, 1000000)]
    public int Amount { get; set; }

    [Required]
    [Range(1, 10)]
    public int Years { get; set; }

    public double Interest { get; set; }
}
```

2. Create a Service Interface and Implementation in the Services Folder

```
public interface IBankInterestService
{
    public double Calculate(int principle, int numberOfYears, double rateOfInterest);
}

public class IciciSimpleInterestService : IBankInterestService
{
    public double Calculate(int principle, int numberOfYears, double rateOfInterest)
    {
        return principle * numberOfYears * rateOfInterest / 100;
    }
}
```

3. Add the Service to the Asp.NET Core Services Container

```
services.AddScoped<IBankInterestService, IciciCompoundInterestService>();
```

4. Create a Controller class with Index Action for GET & Index Action for POST.
Remember to add Dependency Injection in the constructor.

```
public class IciciBankDepositController : Controller
{
    private const double Interest = 14.5;

    private IBankInterestService _bankInterest;

    public IciciBankDepositController(IBankInterestService bankInterest)
    {
        _bankInterest = bankInterest;
    }

    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Index(DepositInfo depositInfo)
    {
        if (!ModelState.IsValid) return View(depositInfo);

        //var simpleInterestService = new IciciSimpleInterestService();

        depositInfo.Interest =
            _bankInterest.Calculate(depositInfo.Amount, depositInfo.Years, Interest);

        return View(depositInfo);
    }
}
```

5. Create a View

```
@model DepositInfo

<div class="container">
  <div class="row">
    <div class="col-12">
      <h2>ICICI Bank Deposit</h2>
    </div>
  </div>
</div>

<form method="post">
  <div asp-validation-summary="All" class="text-danger"></div>
  <div class="mb-3">
    <label asp-for="Amount"></label>
    <input asp-for="Amount" class="form-control" />
    <span asp-validation-for="Amount" class="text-danger"></span>
  </div>
  <div class="mb-3">
    <label asp-for="Years"></label>
    <input asp-for="Years" class="form-control" />
    <span asp-validation-for="Years" class="text-danger"></span>
  </div>

  <button type="submit" class="btn btn-primary form-btn-size">
    Calculate Interest</button>
  <br /><br />
  @if (Model != null)
  {
    <div class="p-2 bg-primary text-white">
      <label asp-for="Interest"></label>
      <span>@Model.Interest</span>
    </div>
  }
</form>

@section Scripts
{
  <partial name="_ValidationScriptsPartial" />
}
```