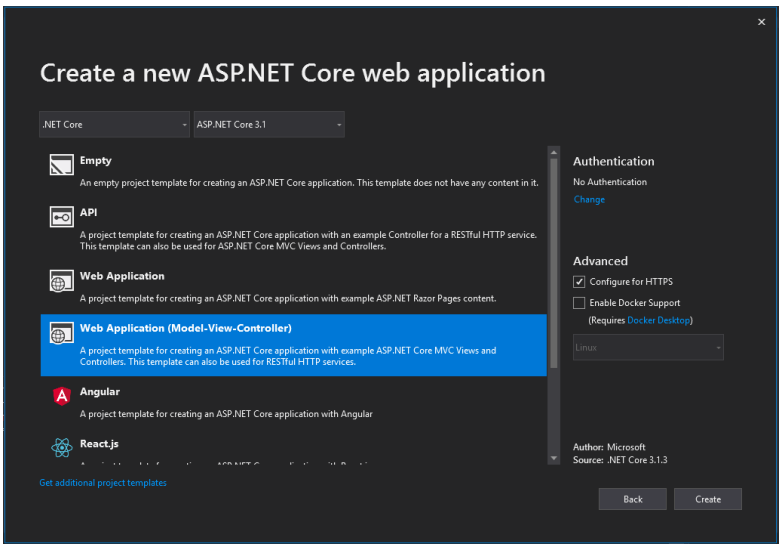
Create a Bank Account Web App.

Step 1. Create the Application

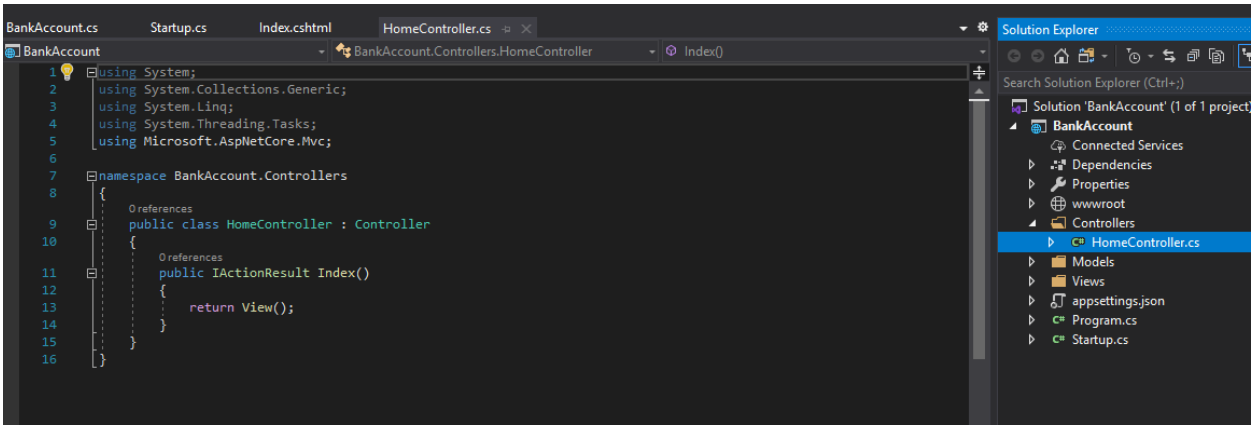
Select ASP.NET Core Web Application, name it BankAccount.

Select Web Application (Model-View-Controller) and deleted files as outlined Video Step 1.



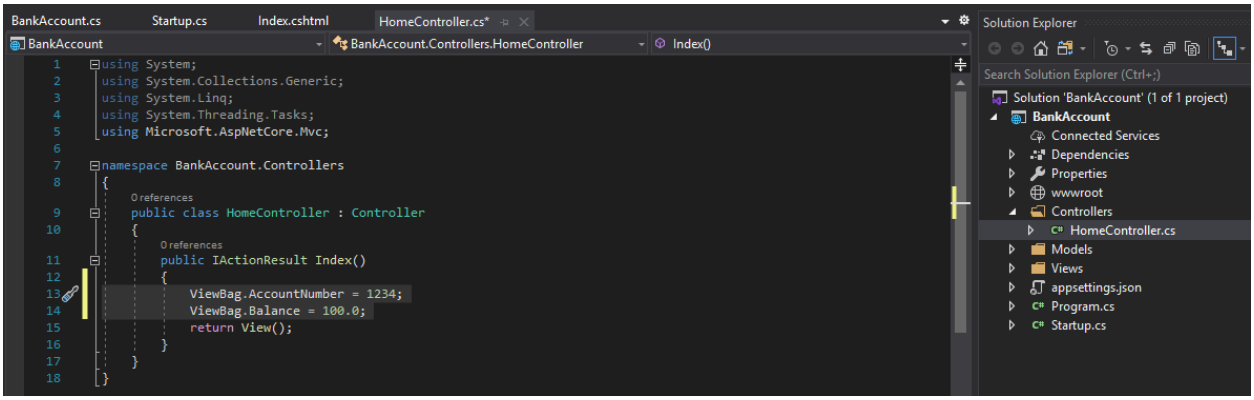
Step 2. Add a Controller

Right Click on the Controllers folder. Add Controller, MVC Controller-Empty, name it HomeController.



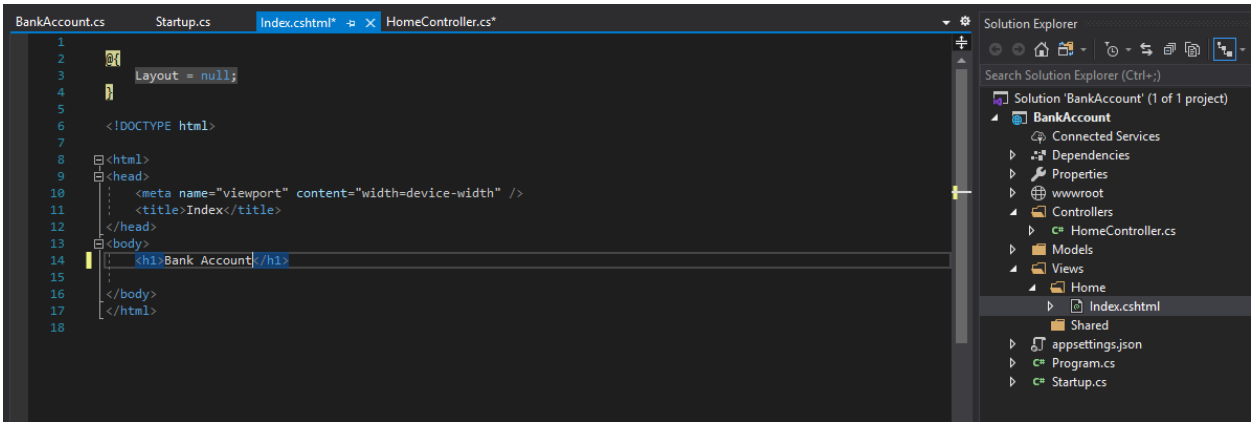
Add lines above return View and Save.

```
ViewBag.AccountNumber = 1234;
ViewBag.Balance = 100.0;
```



Step 3. Add a View

On Home sub folder in Views folder. Add View, name Index. Verify empty Template: Empty [without model] and uncheck "Use a layout page"

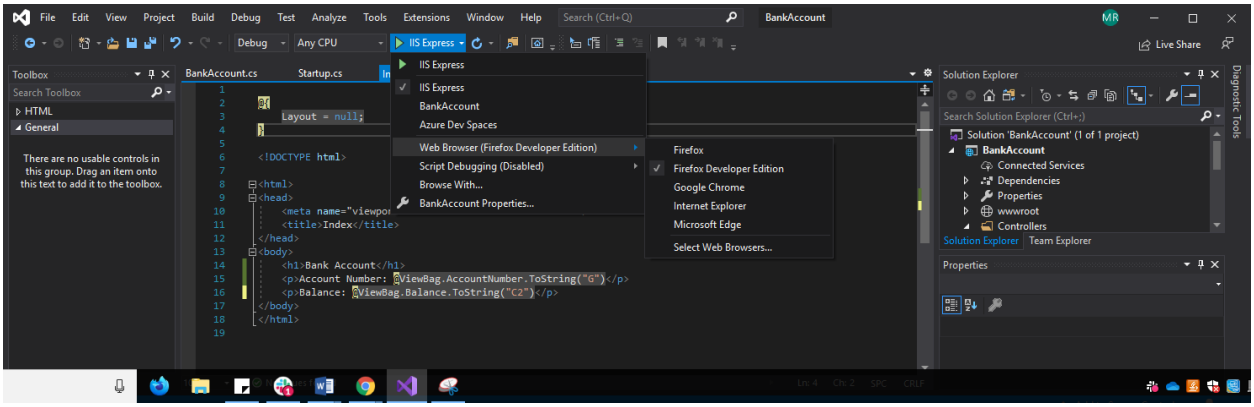


Add the following html code and save the file:

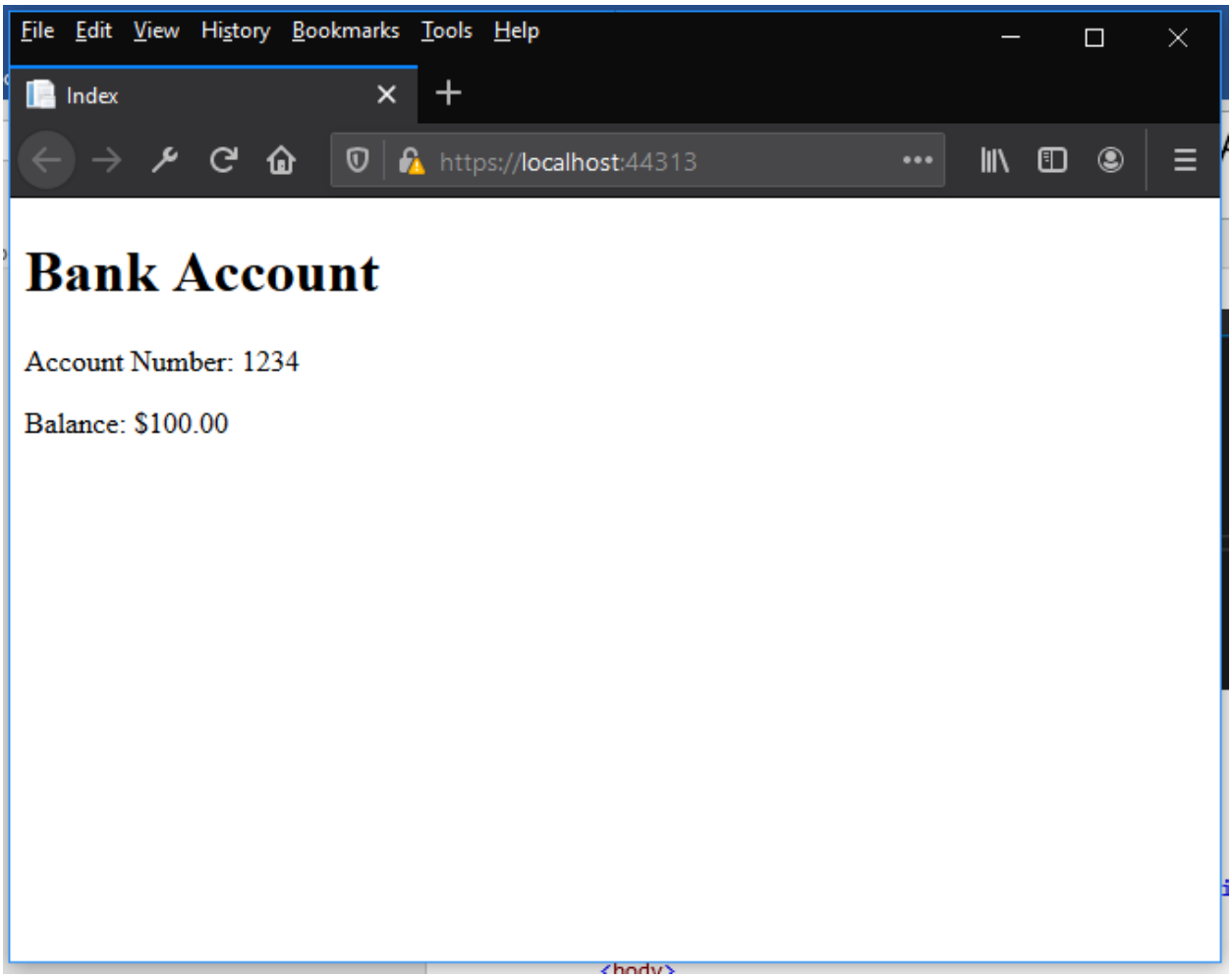
```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Bank Account Web App</title>
</head>
<body>
    <h1>Bank Account</h1>
    <p>Account Number: @ViewBag.AccountNumber.ToString("G")</p>
    <p>Balance: @ViewBag.Balance.ToString("C2")</p>
</body>
</html>
```

Step 4. Configure and Run

On the green run button “IIS Express”, use the arrow on the left to select the browser for running your code:



Run using the green button. To end the run, click, close the Web browser and click the red square.

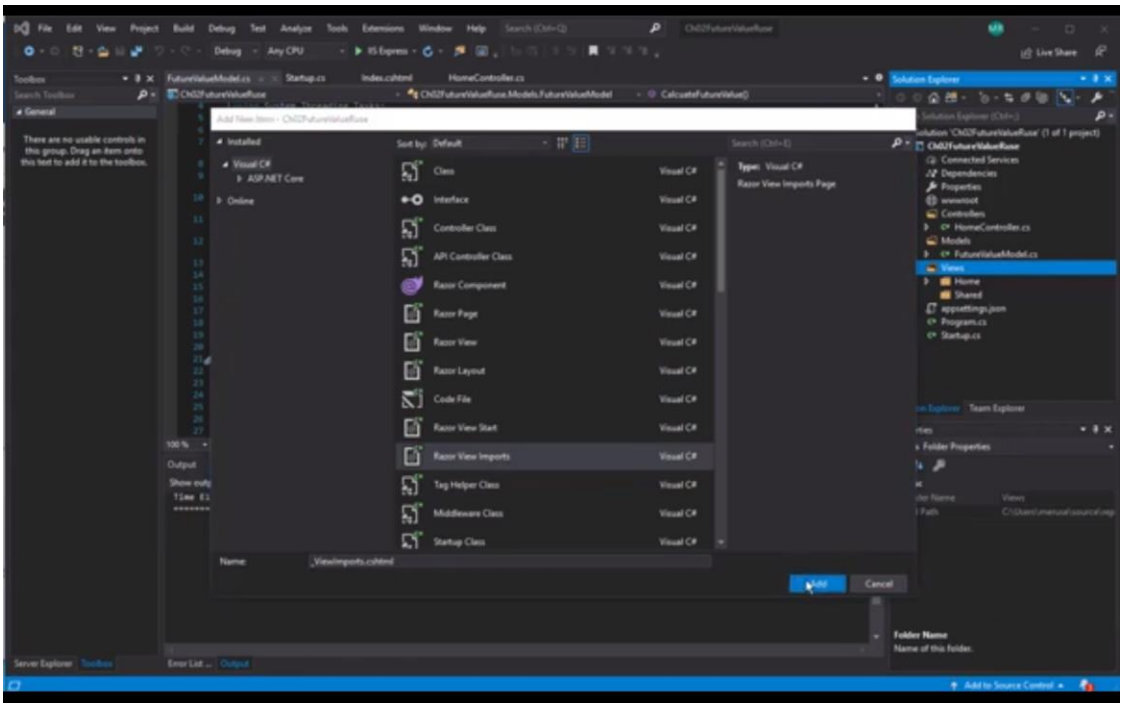


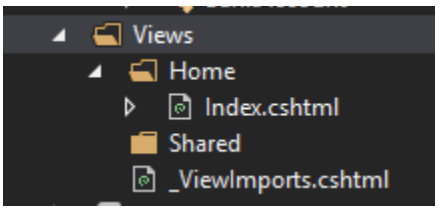
Step 5. Add a Model and View Imports

Add a Model, a C# class. Right click on Model folder. Add Class BankAccount. Add the code below and save.

```
public class BankAccount
{
    private decimal _balance;
    private decimal _amount;

    public int AccountNumber;

    public void Withdraw()
    {
        _balance = (_balance - _amount);
    }
    public void Deposit()
    {
        _balance = (_balance + _amount);
    }
}
```

Add a Razor View Imports page. Right click on Views folder. Add New Item, Razor View Imports





In _ViewImports.cshtml add the code

```
@using BankAccount.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Step 6. Handle GET and POST requests

Under Views Home, open Index.html. At the top add `@model BankAccount`

Remove the lines

```
<p>Account Number: @ViewBag.AccountNumber.ToString("G")</p>
<p>Balance: @ViewBag.Balance.ToString("C2")</p>
```

Add the following in place of the above:

```
<form asp-action="Index" method="post">
    <div>
        <label asp-for="AccountNumber">Account Number</label>
        <input asp-for="AccountNumber" />
    </div>
    <div>
        <label asp-for="Balance">Account Balance</label>
        <input asp-for="Balance" />
    </div>
    <div>
        <label asp-for="Amount"> Amount</label>
        <input asp-for="Amount">
    </div>
    <div>
        <label> Ending Balance</label>
        <input value="@ViewBag.EndingBalance.ToString("C2")" readonly />
    </div>
    <button type="submit">Withdraw</button>
    <a asp-action="Index">Clear</a>
</form>
```

Open Controllers, HomeController.cs and add to the directives `using BankAccount.Models;`

Then add

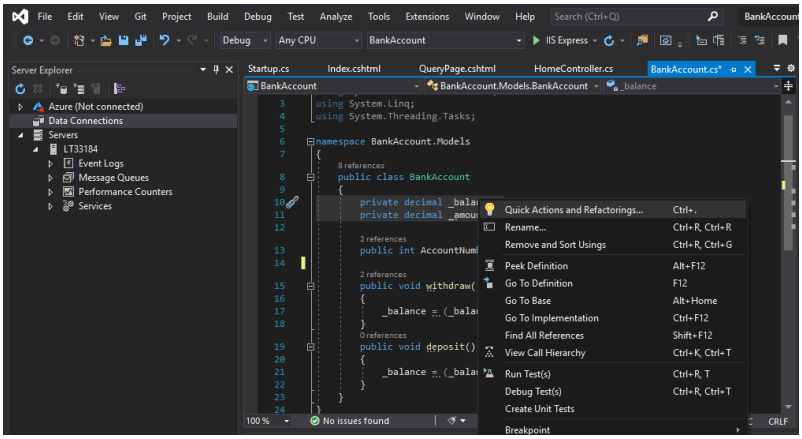
```
public class HomeController : Controller
{
    [HttpGet]
    public IActionResult Index()
    {
        ViewBag.EndingBalance = 0;
        return View();
    }
    [HttpPost]
    public IActionResult Index(BankAccount.Models.BankAccount model)
    {
        model.Withdraw();
        ViewBag.EndingBalance = model.Balance;
        return View(model);
    }
}
```

Step 7. Update Class and Manually Test

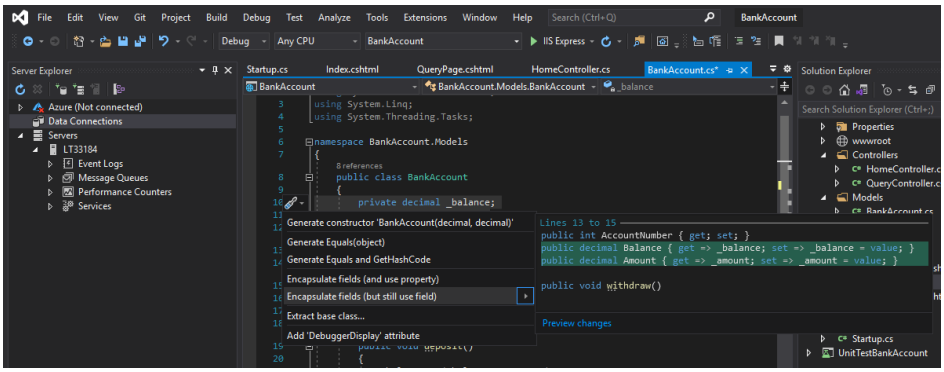
Run your code and notice it is not working (Yet!).

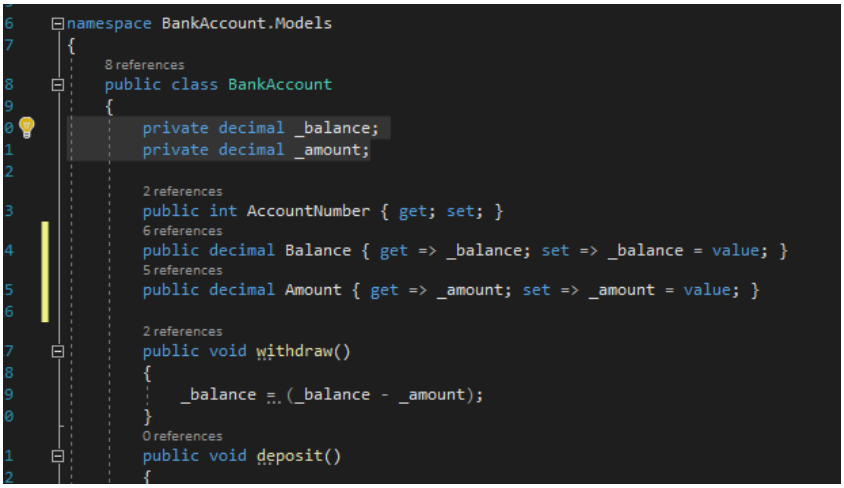
Update your Bank Account class properties (setters and getters) to return their properties and to set the value in each.

Highlight the properties, right click and use Quick Actions and Refactorings…



Select Encapsulate Fields (but still use Fields)





Ask on the discussion board if you need help!