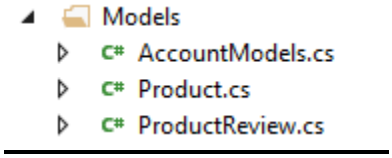
Playbook for Code First Approach MVC

1. Define the models



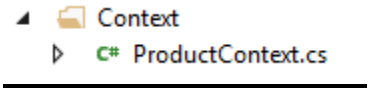
2. Define the Attributes in the Model (Keys)

```
public class Product
{
    [Key]
    3 references
    public int ProductId { get; set; }
    2 references
    public string ProductName { get; set; }
    2 references
    public string ProductType { get; set; }

    2 references
    public int ProductPrice { get; set; }

    2 references
    public string ProductDesc { get; set; }
```

3. Define Navigation Property

```
public class ProductReview
{
    [Key]
    3 references
    public int ReviewId { get; set; }
    2 references
    public int Rating { get; set; }

    2 references
    public string comment { get; set; }


    // foreign key
    2 references
    public int ProductId { get; set; }

    //Navigation property

    0 references
    public Product Product { get; set; }

}
```

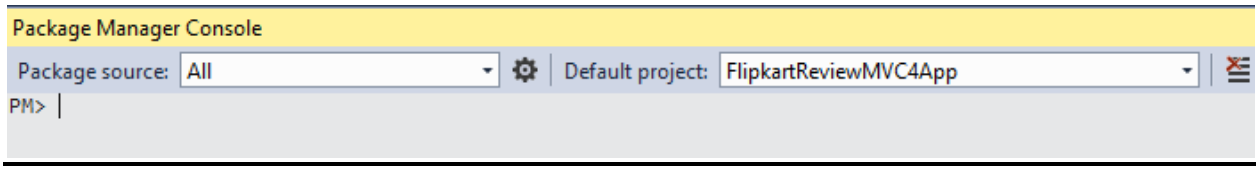
4. Create Corresponding Context for the application



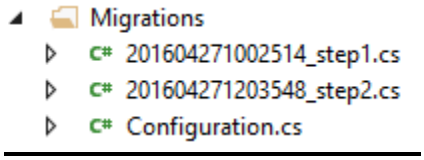
5. Register the Context in Web.Config by copying and replacing the connection string and replacing the default connection string to the context name define in step 4.

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;Initial C
  <add name="ProductContext" connectionString="Data Source=DESKTOP-I57JOS9\SQLEXPRESS20
</connectionStrings>
```

6. Run the Package Manager Console

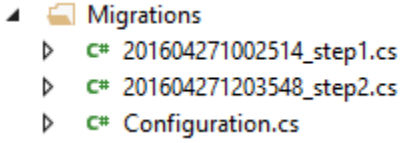


7. Enable-Migrations -ContextType ContextName(Specified in Step 4)
   a. A Migrations folder with Configuration.cs is created as shown below

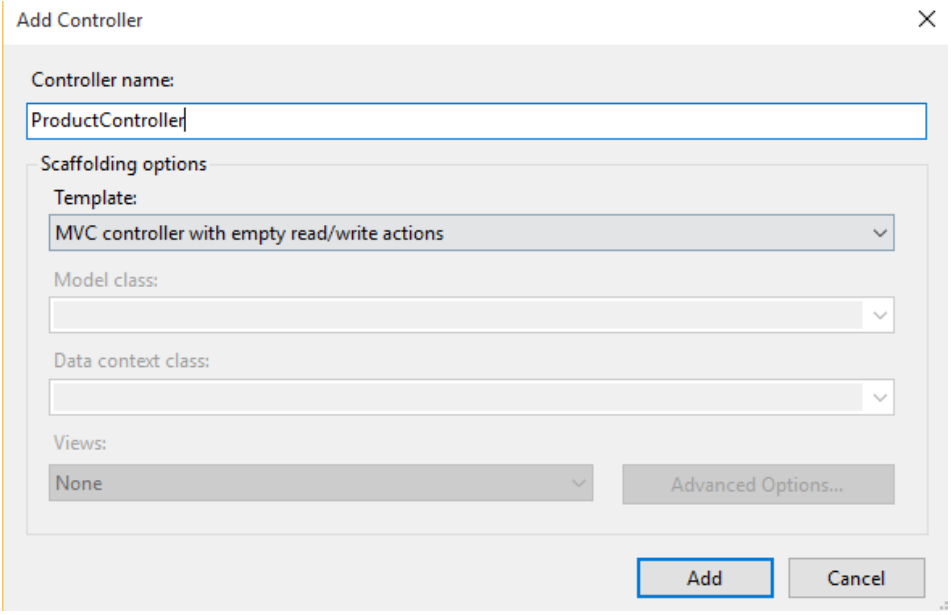


8. Update-Database

9. Add-Migration step1

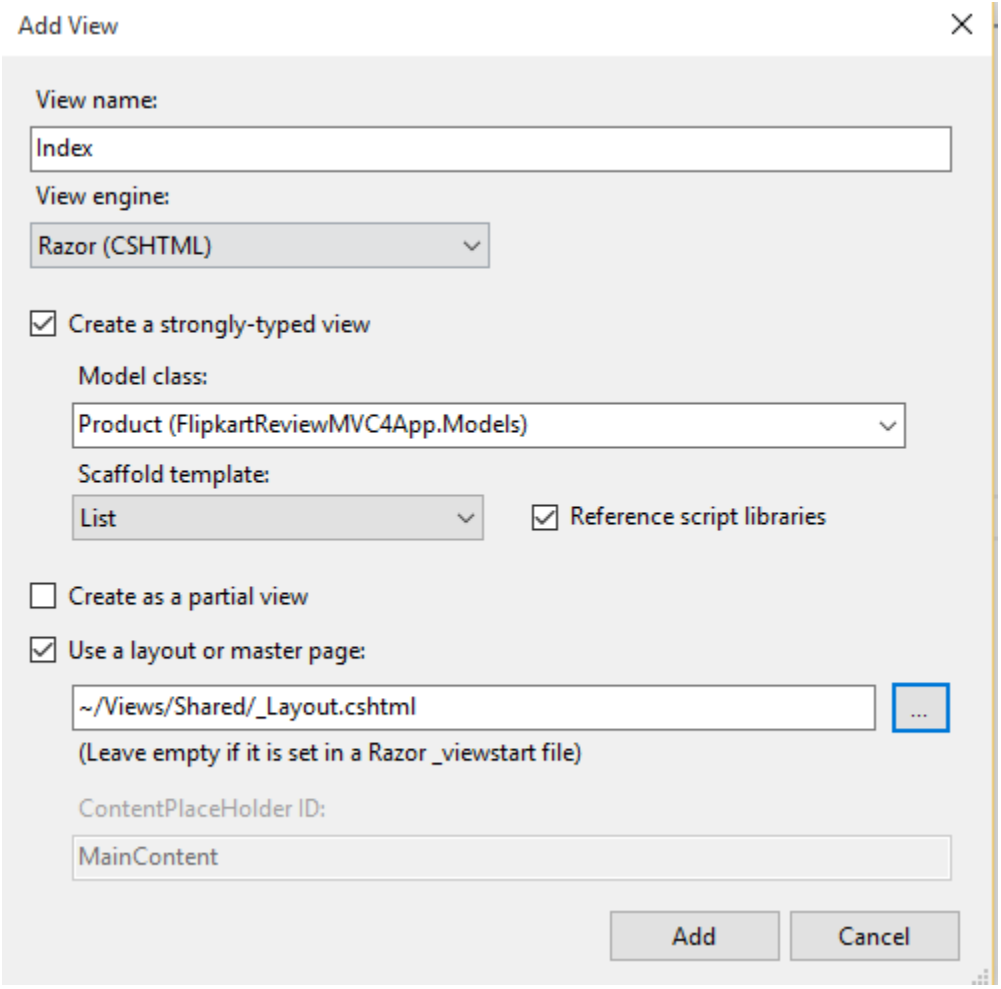


10. Define the corresponding controllers for the models to tie up the models to the controllers. Use appropriate scaffolding to suite to the requirements (preferably read/write actions)




```
0 references
public class ProductController : Controller
{
    ProductContext db = new ProductContext();
    //
    // GET: /Product/

    0 references
    public ActionResult Index()
    {
        return View(db.Products.ToList());
    }

    //
    // GET: /Product/Details/5

    0 references
    public ActionResult Details(int id)
    {
        return View();
    }
```

11. Render corresponding views by clicking on the controller action methods and choose appropriate scaffolding to suit your requirements.



12. Build your application and run the application
13. Navigate to the corresponding routes
14. http://localhost:port/product/index