



Session 6

Introduction to ASP.NET MVC

Session Overview

- Describe ASP.NET MVC
- Explain ASP.NET MVC life cycle
- Define ASP.NET MVC pattern
- List the advantages of MVC-based Web applications
- Explain the Models in ASP.NET
- Explain how model binding and validation is performed
- Describe Views in ASP.NET
- Explain Controllers in ASP.NET

Introduction to ASP.NET MVC

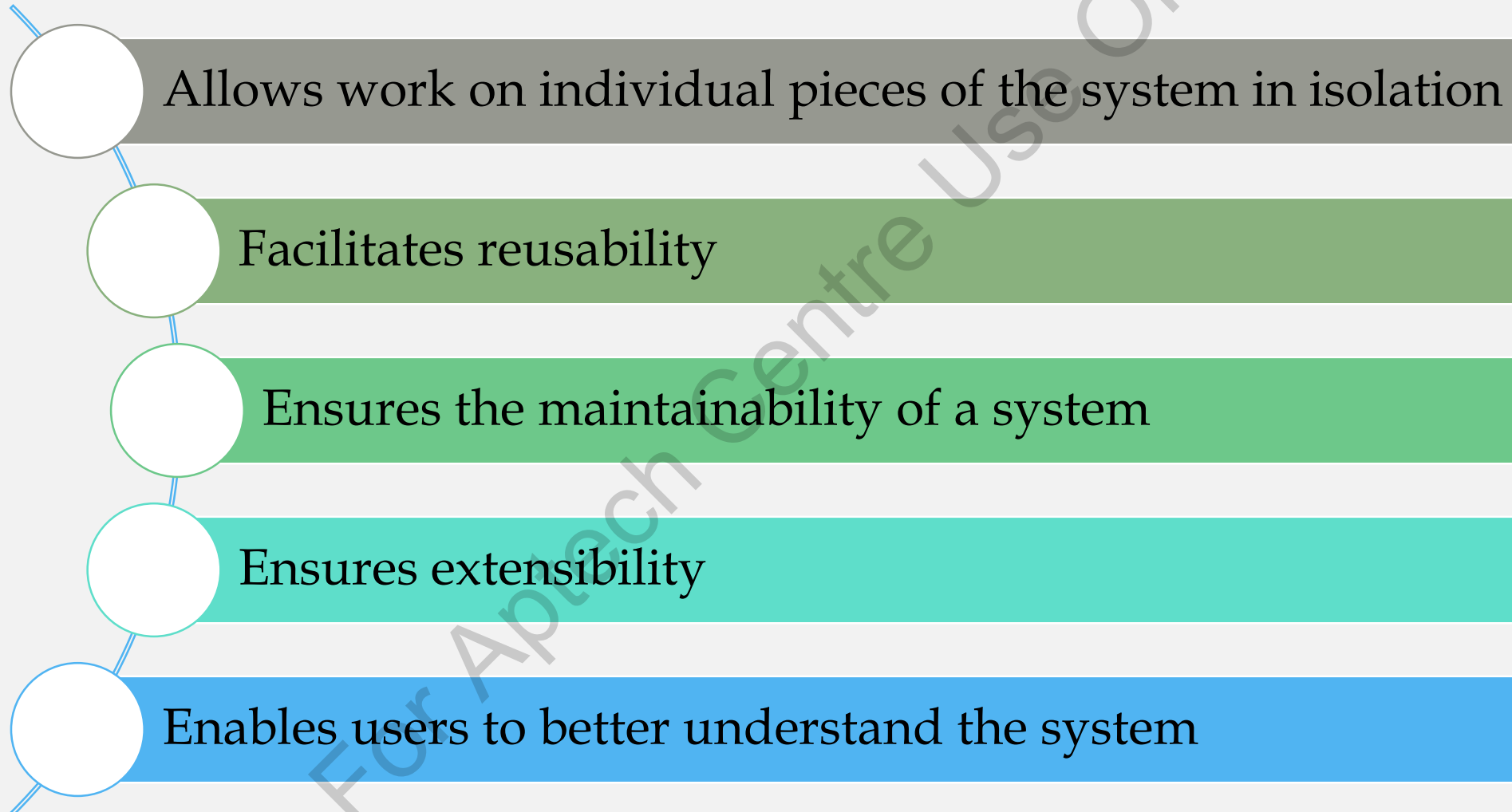
Three basic components of ASP.NET MVC:

Model (Business Entities)

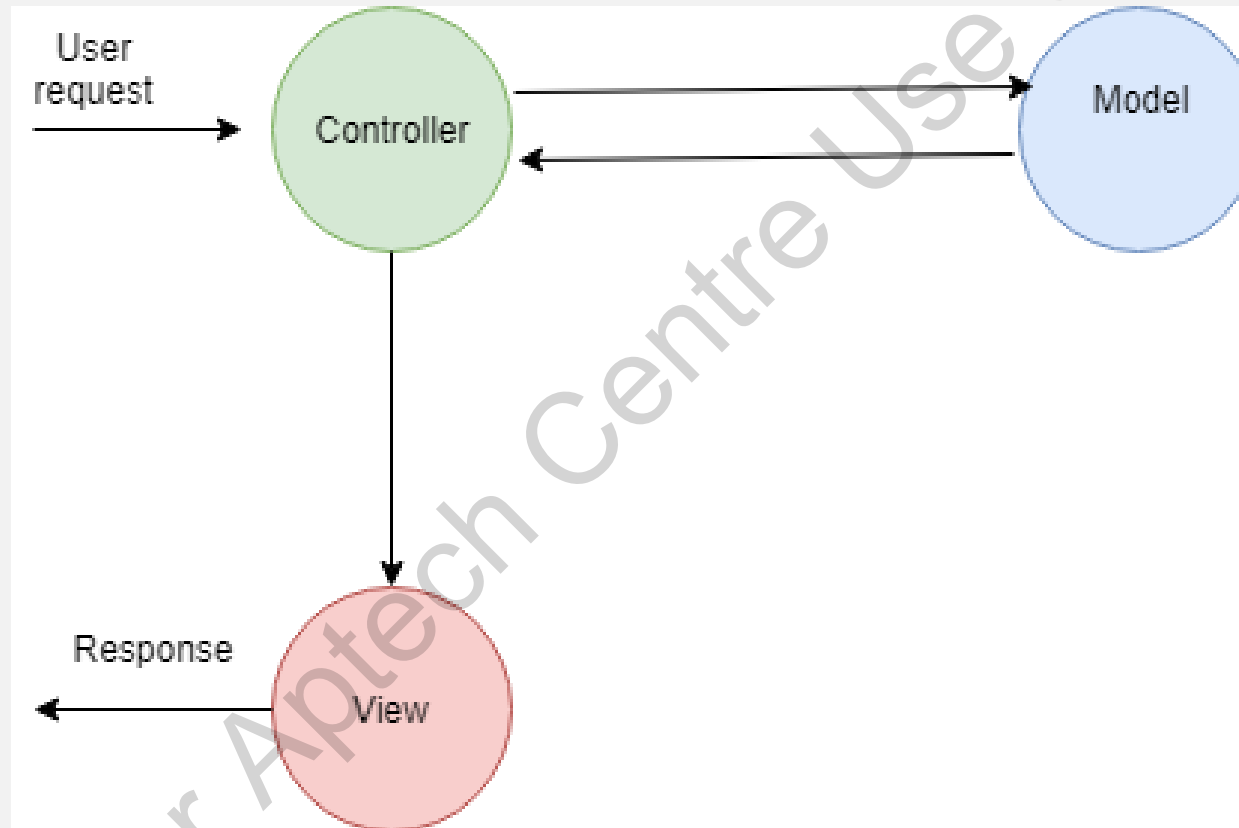
View (Presentation Logic)

Controller (Business Logic)

Advantages of Separation of Concerns (SoC)

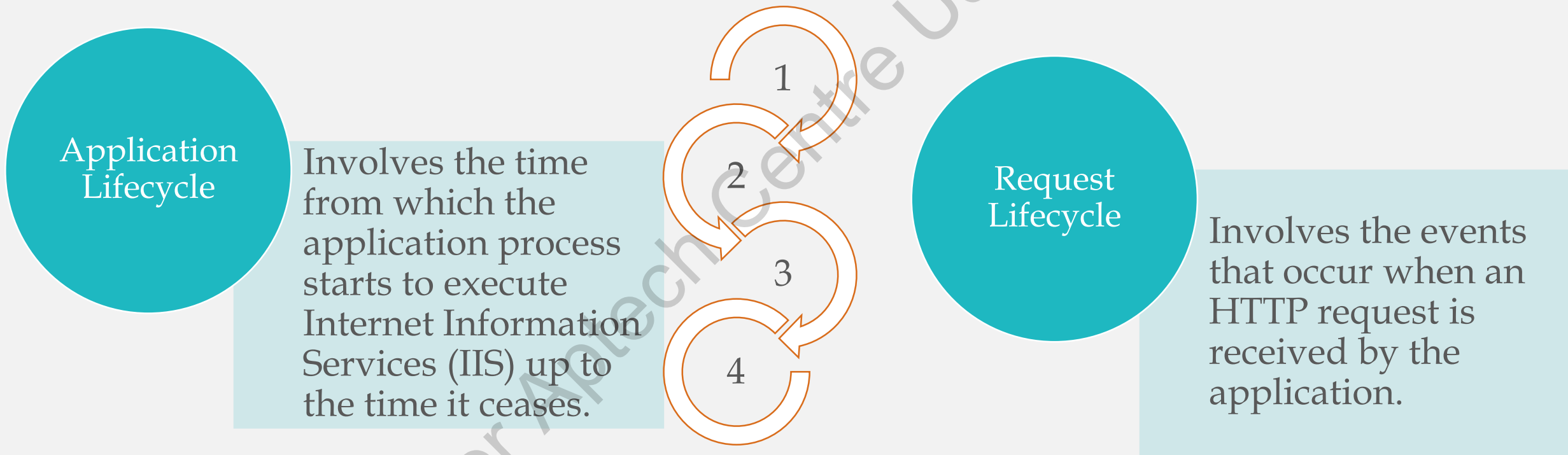
- 
- Allows work on individual pieces of the system in isolation
 - Facilitates reusability
 - Ensures the maintainability of a system
 - Ensures extensibility
 - Enables users to better understand the system

ASP.NET MVC Pattern

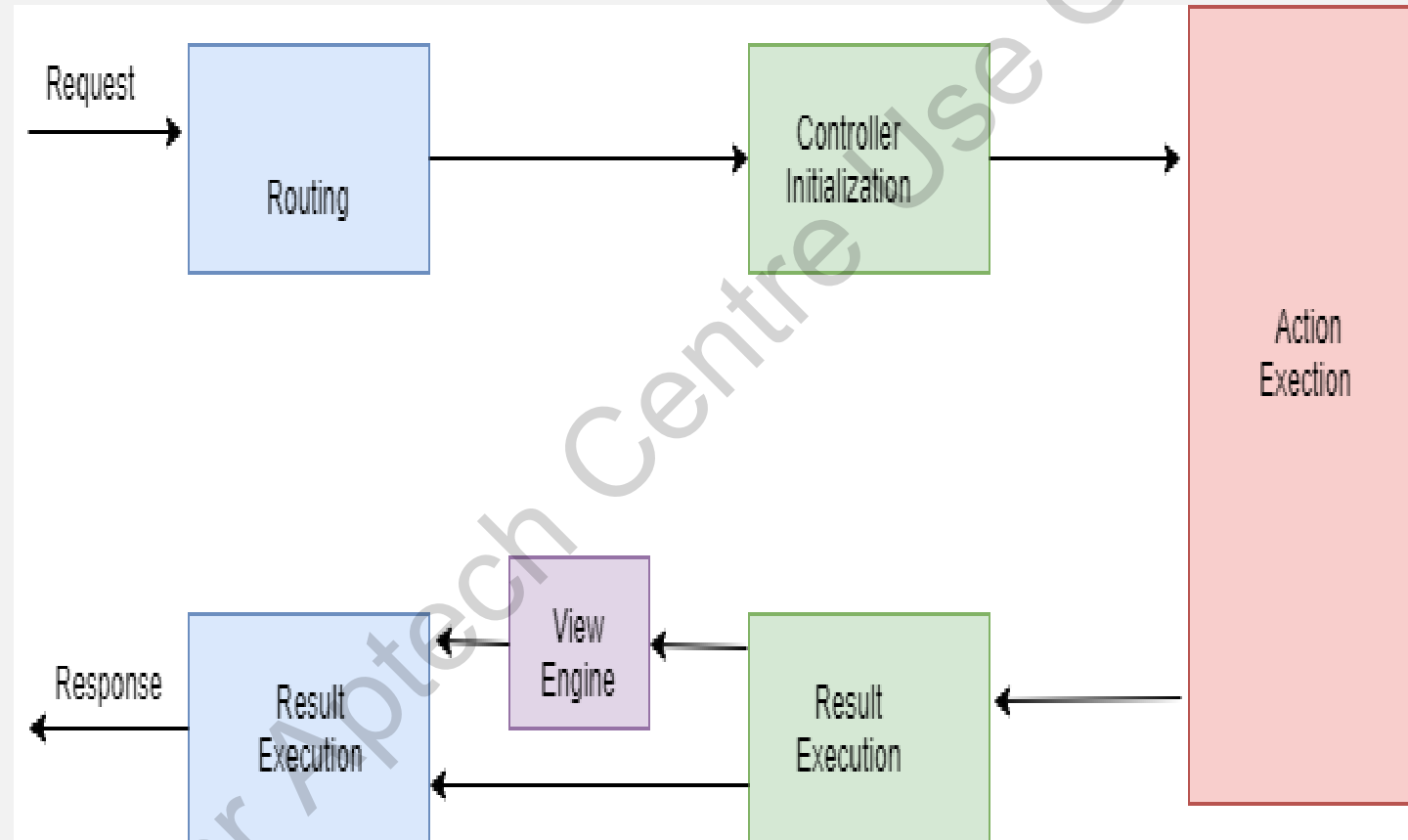


ASP.NET MVC Pattern

ASP.NET MVC Lifecycle



ASP.NET MVC Request Lifecycle



Request Life Cycle

Advantages of ASP.NET MVC

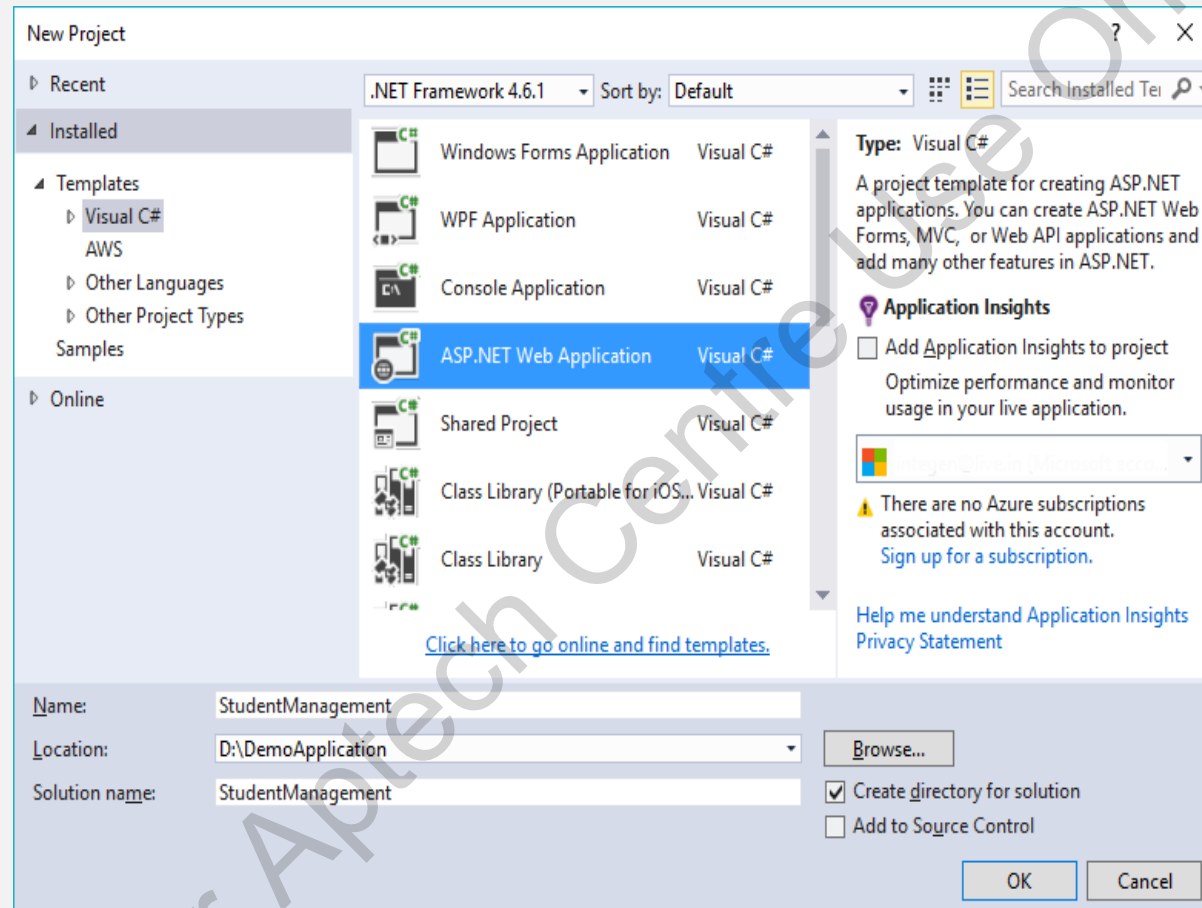
Provides asynchronous technique

Changes do not impact the complete model

Easy to Test, called Test Driven Development (TDD)

SEO friendly development platform

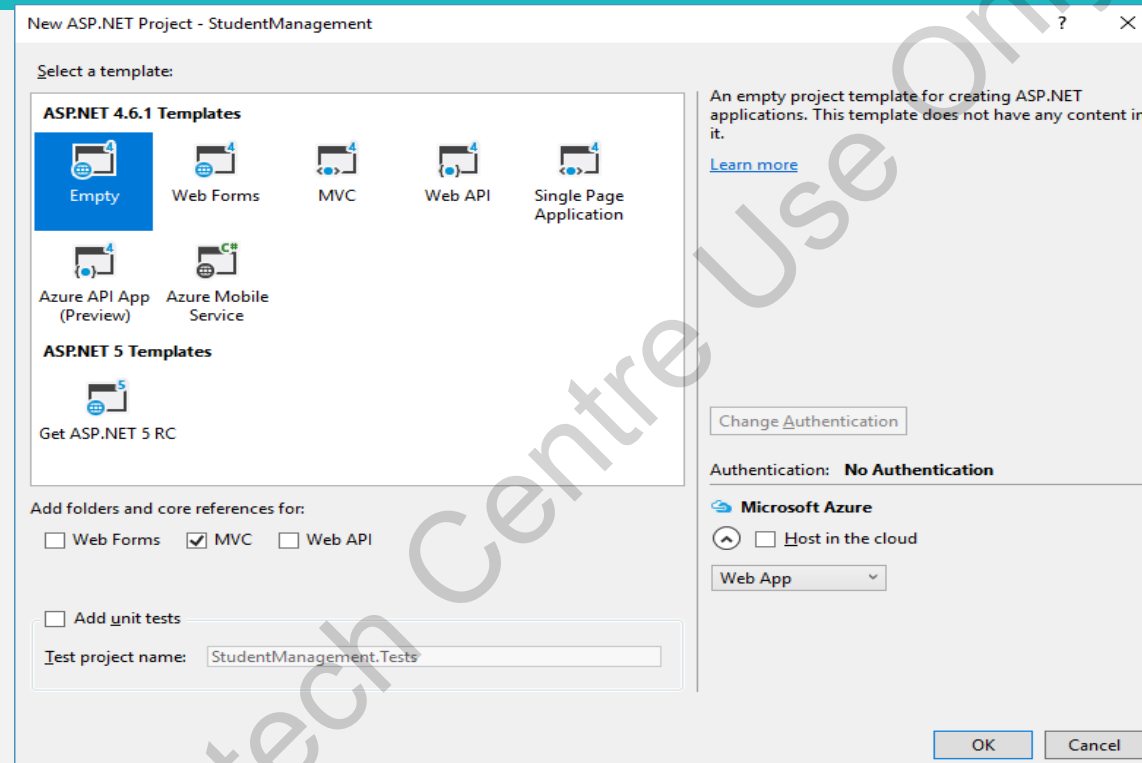
Creating a Simple MVC Application (1-2)



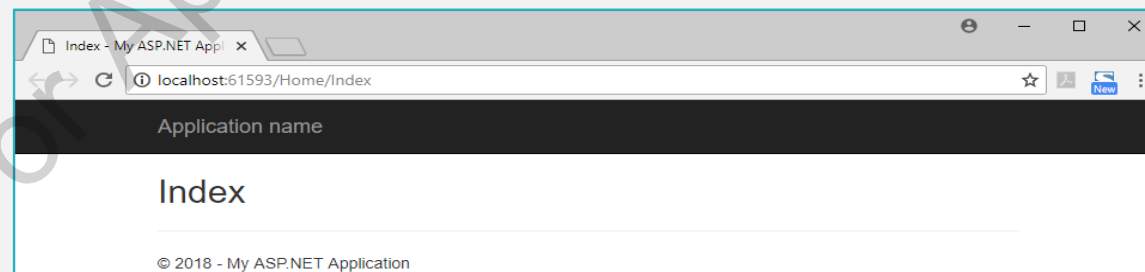
New Project Window

Creating a Simple MVC Application (2-2)

New ASP.NET Web Application Window



Index() Method Output



Models in ASP.NET MVC

Models are Containers that are specific to the business

Don't Repeat Yourself (DRY)

Model Binding

Adding Validations to Model

Controllers in ASP.NET MVC

Controllers help in managing complete flow of application including accepting input and rendering appropriate output.

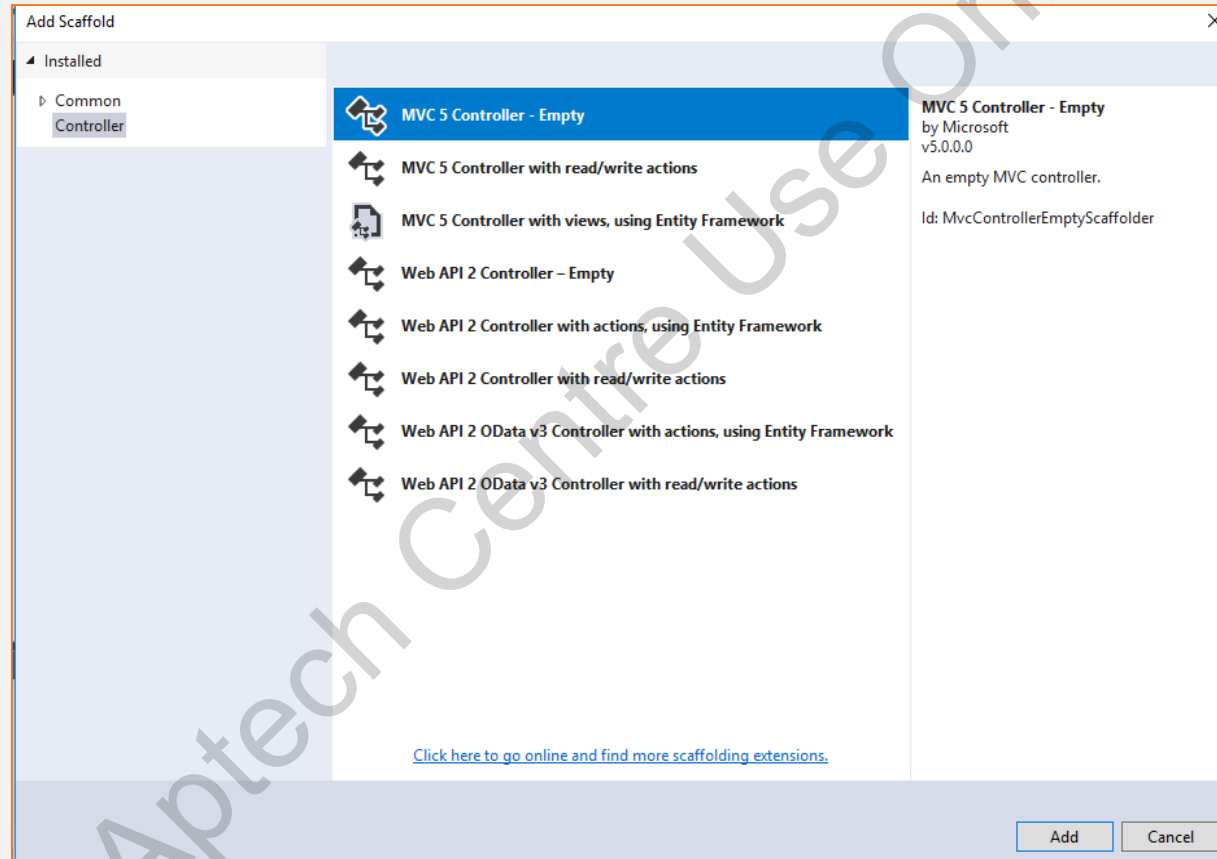
Controllers are basically C# classes that inherit from `System.Web.Mvc.Controller`.

Controller along with its action method accepts incoming browser requests, retrieves required model information, and provides suitable responses.

It is recommended that class name of a controller ends with 'Controller'.

Creating Controller in an Application

Add Scaffold Dialog Box

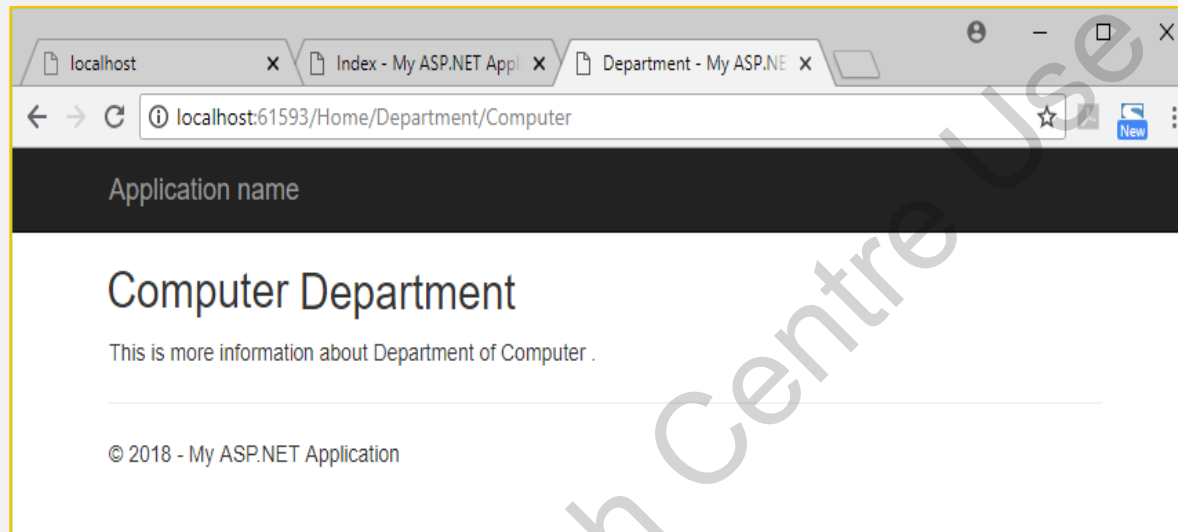


The pre-installed code generators for MVC and Web API projects provided by Visual Studio is called **Scaffolding**.

Routing and Custom Routes (1-2)

- Routing defines how the application will process and respond to an incoming HTTP request.
- When an ASP.NET MVC application is created using Visual Studio.NET, a route is automatically configured in the RouteConfig.cs file under **App_start** folder.
- The RouteConfig class contains a MapRoute() method.

Routing and Custom Routes (2-2)

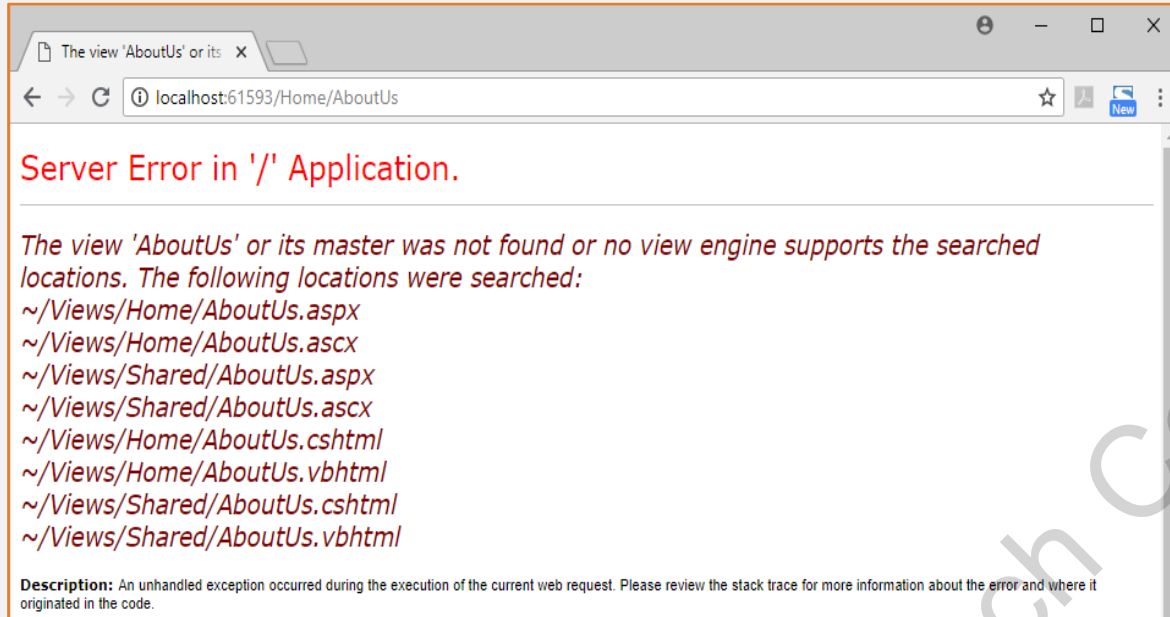


Department Output

Views in ASP.NET (1-2)

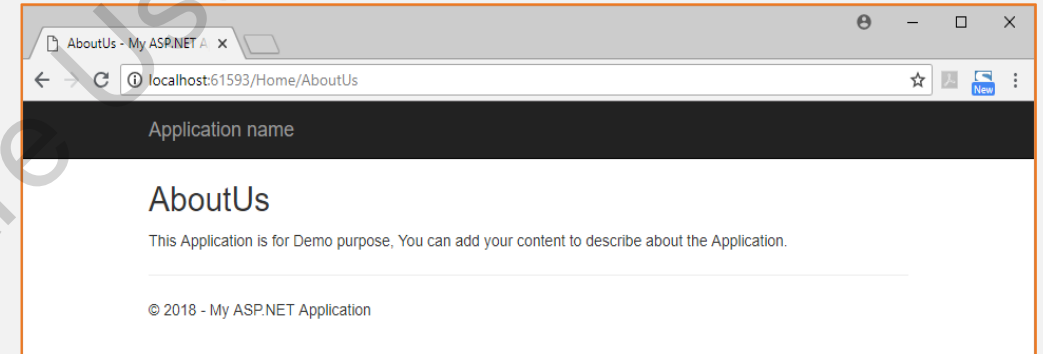
- An ASP.NET MVC application includes a view, which is similar to a page.
- A controller action performs one of the two tasks at a time, displaying a view or forwarding it to another controller action.
- It is possible to create additional views in an application.

Views in ASP.NET (2-2)



Erroneous AboutUs

Output of Code Snippet 7



AboutUs

Output of Code Snippet 8

Summary

- MVC is a framework that helps developers to create Web applications in which sections of code are organized by the functions they perform.
- Three basic components of MVC are Model, View, and Controller.
- Fundamental pattern component of an MVC application is the component called the view. It is accountable for rendering the user interface, irrespective of it being an HTML or a UI widget on a desktop application.
- A model can be defined as a collection of classes that help developers to work with data and business logic.
- Validation forms a vital aspect in ASP.NET MVC applications. It helps developers to assess if the user input is valid.
- Controllers form the central unit of the ASP.NET MVC application. It is the first unit that any incoming HTTP Request communicates with. It also has the responsibility of deciding which model to choose.
- The MVC design pattern is definitely a great approach to create software applications. Projects created using MVC model consume lesser expenditure and time too.