



Session 8

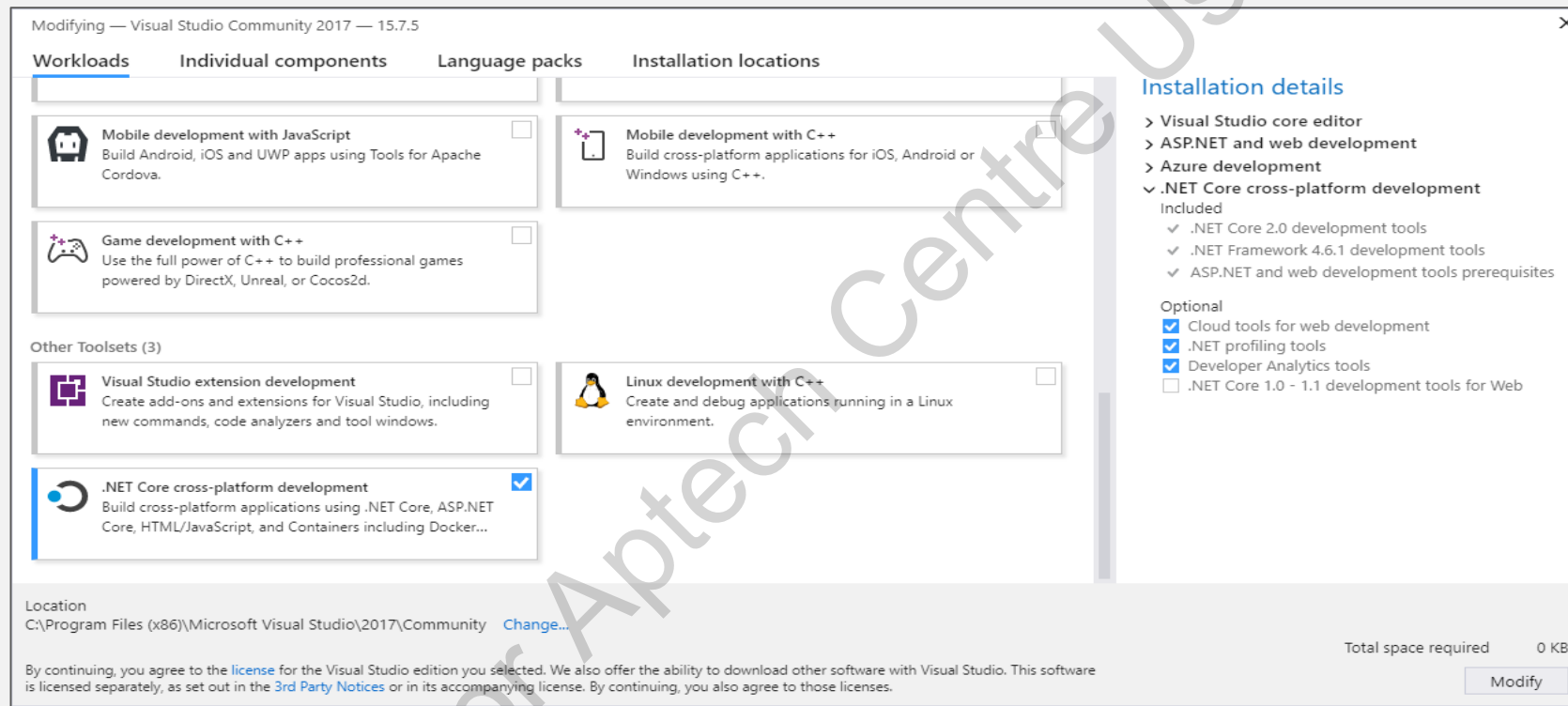
*Client-side
Development
Using ASP.NET
Core MVC*

Session Overview

- Define static files in ASP.NET
- Define ADO.NET Core MVC
- Describe client-side validation
- Describe server-side validation
- Explain how to implement a Core MVC application

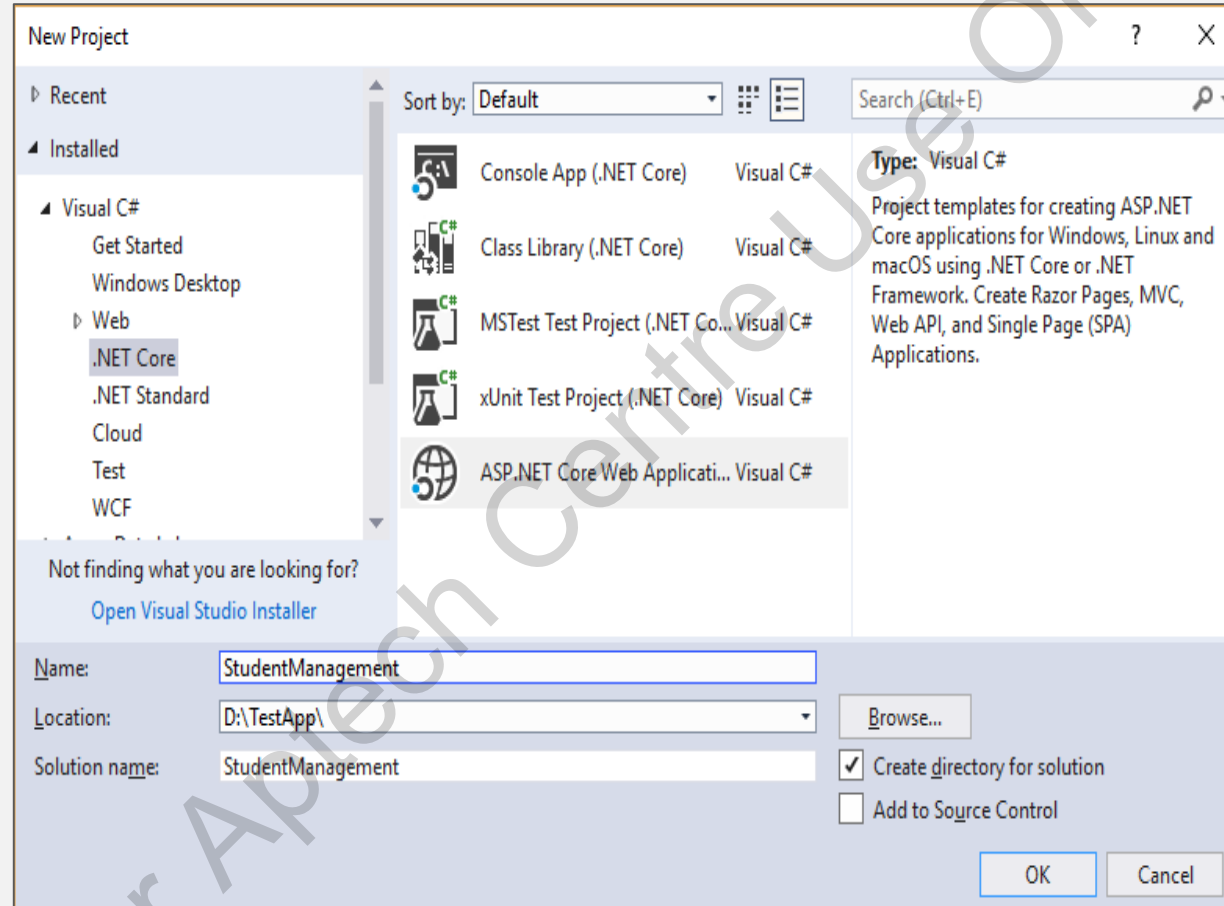
Overview of ASP.NET Core (1-2)

- ASP.NET Core is a part of Visual Studio optional installation
- It is not installed by default



.NET Core Installation

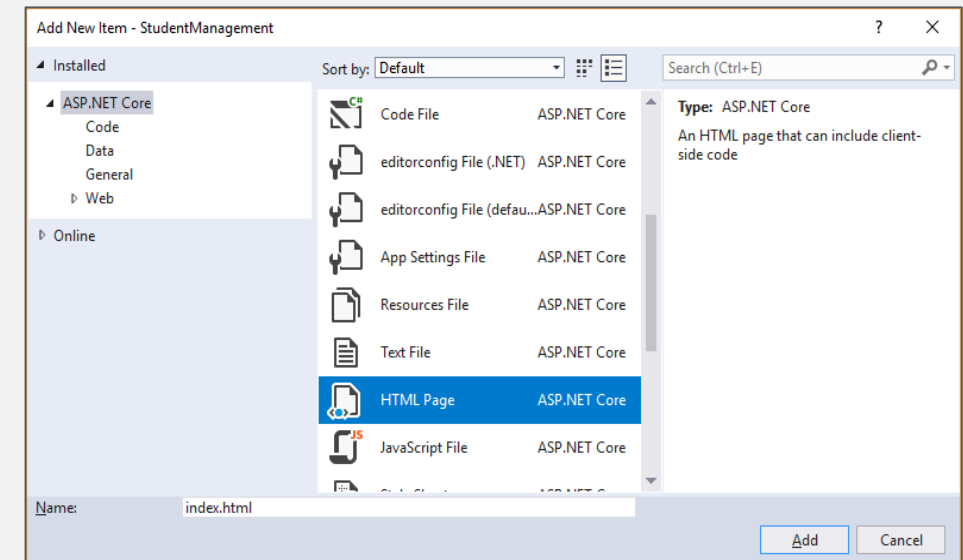
Overview of ASP.NET Core (2-2)



.NET Core Project

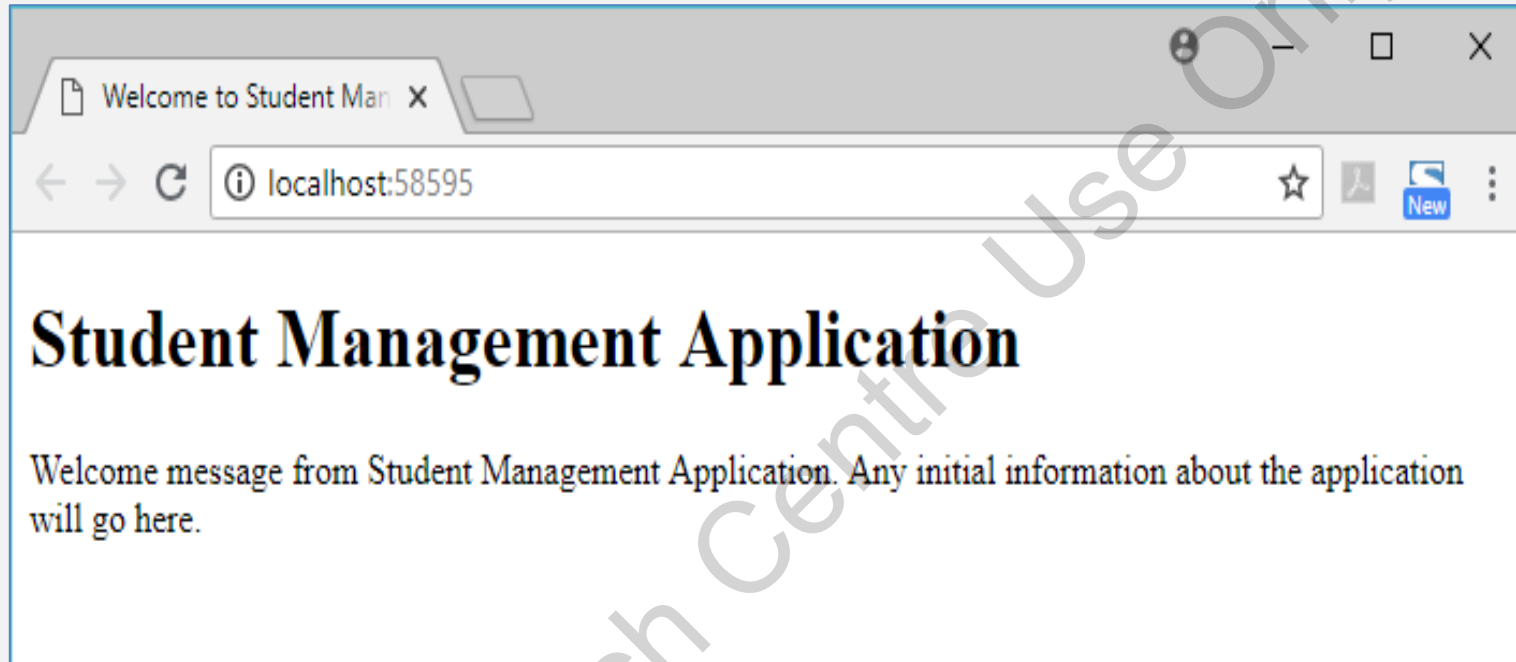
Static Files in ASP.NET (1-2)

- ASP.NET Core applications can serve static files, such as HTML, JavaScript, images, and Cascading Style Sheet (CSS) files.
- Web root (`wwwroot`) folder stores these static files.



Adding HTML File to the Application

Static Files in ASP.NET (2-2)



Index.Html File Loaded

Introduction to ASP.NET Core MVC (1-2)

- Lightweight
- Open source
- Highly testable presentation framework
- Provides a patterns-based way to build dynamic Websites
- Enables clean separation of concerns and full control over markup
- Supports Test Driven Development (TDD)-friendly development
- Uses latest Web standards

Introduction to ASP.NET Core MVC (2-2)

Web APIs

**Razor View
Engine**

**Strongly
Typed
Views**

**View
Components**

Routing

**Model
Binding**

**Model
Validation**

**Dependency
Injection**

Differences between ASP.NET Core MVC and ASP.NET MVC 5

ASP.NET Core MVC

A new approach to server-side and client-side dependency management of packages.

Apps do not use Internet Information Services (IIS) for hosting.

Single aligned Web stack for ASP.NET Core MVC and Web APIs.

Inbuilt Dependency Injection (DI) support for ASP.NET Core.

ASP.NET MVC 5

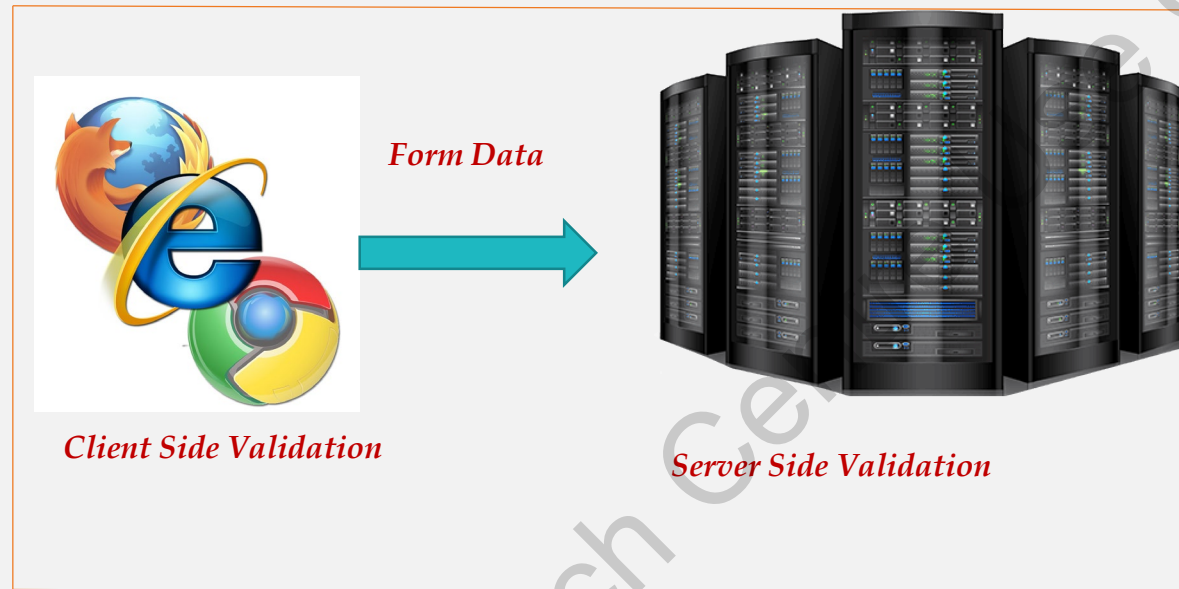
Reference to assemblies, library, framework, or any third-party packages is added using NuGet package manager.

Apps use IIS for hosting.

Provides an option to choose between MVC or Web API, or both, because Web stack for MVC 5 and Web API are different.

Separate DI containers such as Unity, Autofac, StructureMap, and so on.

Client-side Validation (1-3)



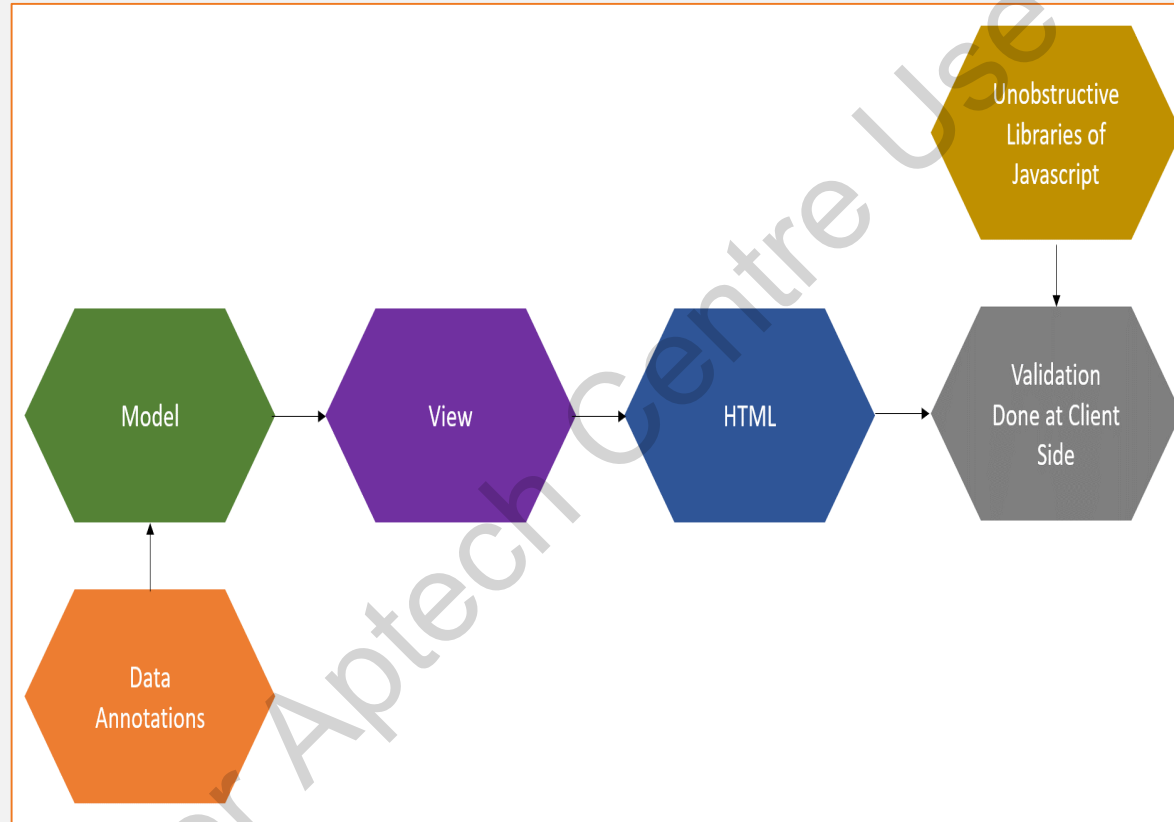
Client-side Validation (2-3)

CreateStudent

Student Name:	<input type="text"/> Student Name is required.
Student Email:	<input type="text"/> Student Email is required.
Date of Join:	<input type="text" value="01-01-0001 00:00"/>
<input type="button" value="Submit"/>	

Client-side Validation Using JavaScript

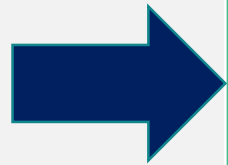
Client-side Validation (3-3)



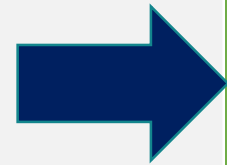
Client-side Validation

Server-side Validation (1-4)

Add data annotation attributes to ViewModel model class.



Update the view method. Display validation error message using span tag helper with asp-validation-for attribute.



Verify ModelState by updating controller action method. Add data to database if ModelState is valid. Else, update ViewModel and render view method again with validation error message.

Server-side Validation (2-4)

Required:

An indication that the property is required.

Range:

The minimum and maximum constraints are defined.

MinLength:

The minimum length that a property must have for the validation to succeed is defined.

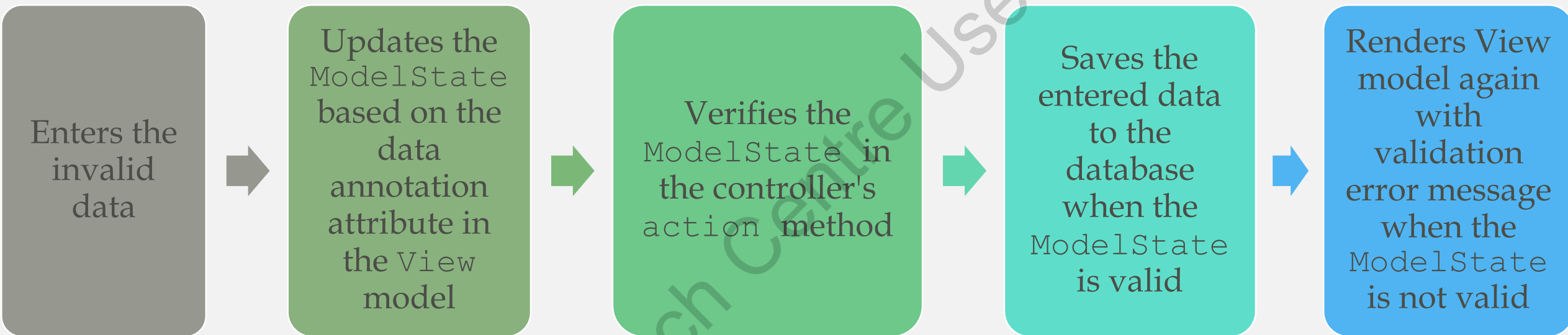
MaxLength:

The maximum length of the property is defined. The validation fails if the length of the property value exceeds.

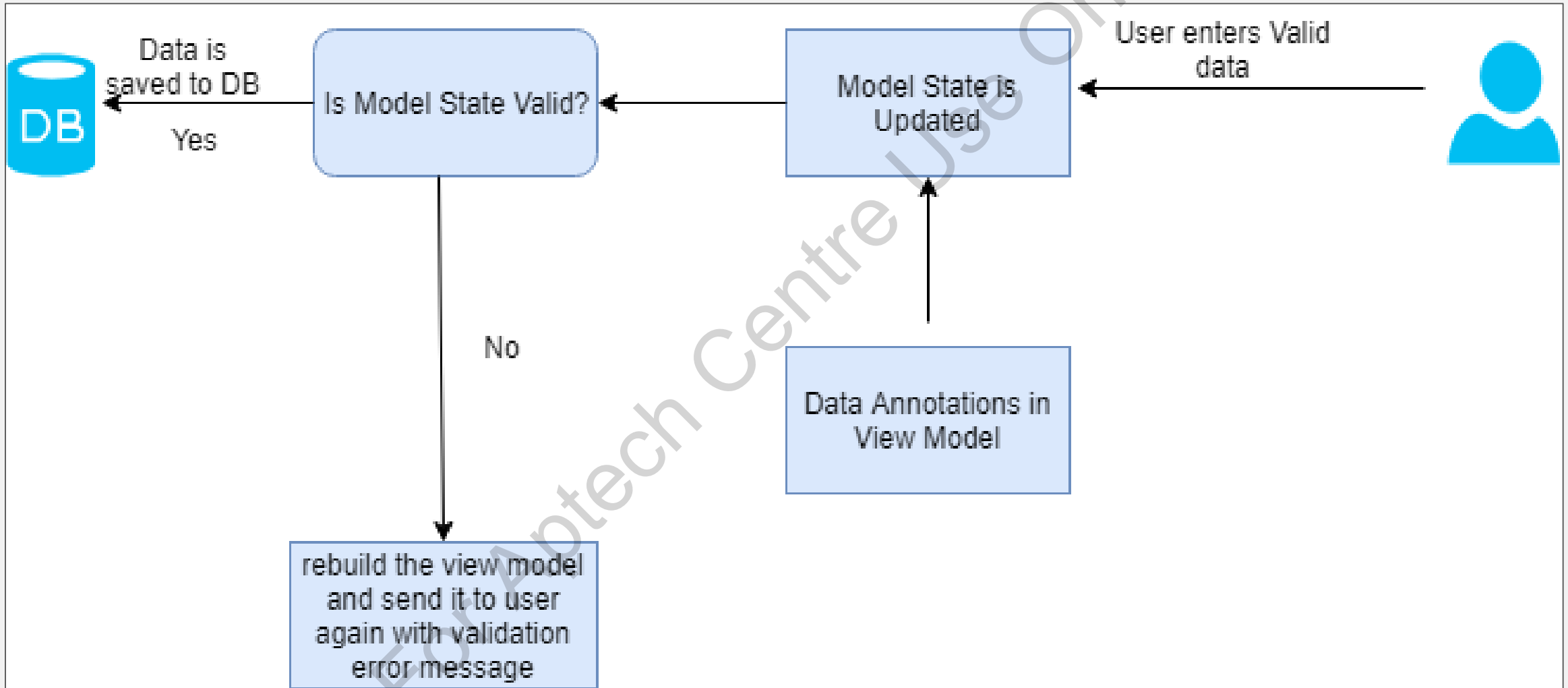
RegularExpression

: A regular expression for data validation is used by using this attribute.

Server-side Validation (3-4)

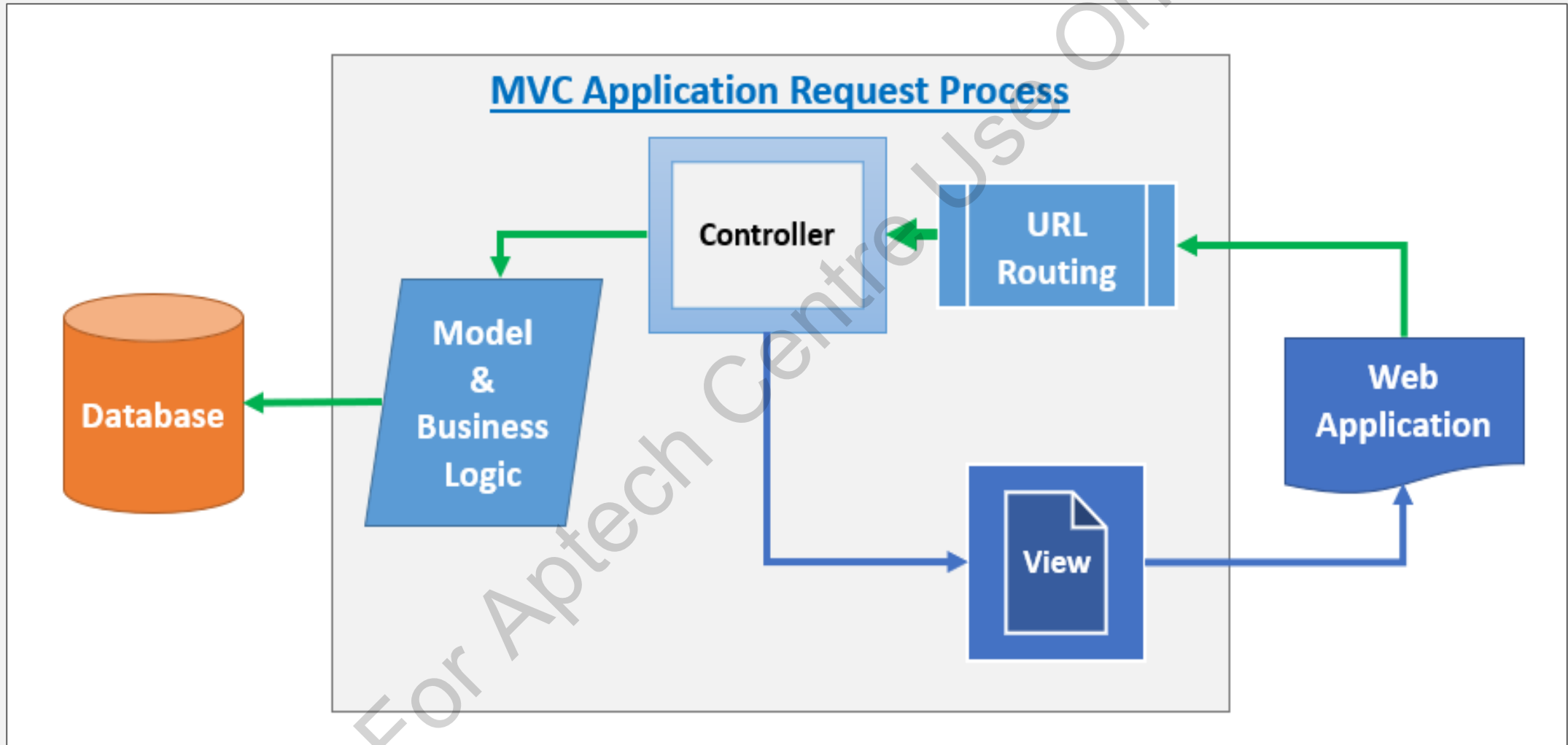


Server-side Validation (4-4)



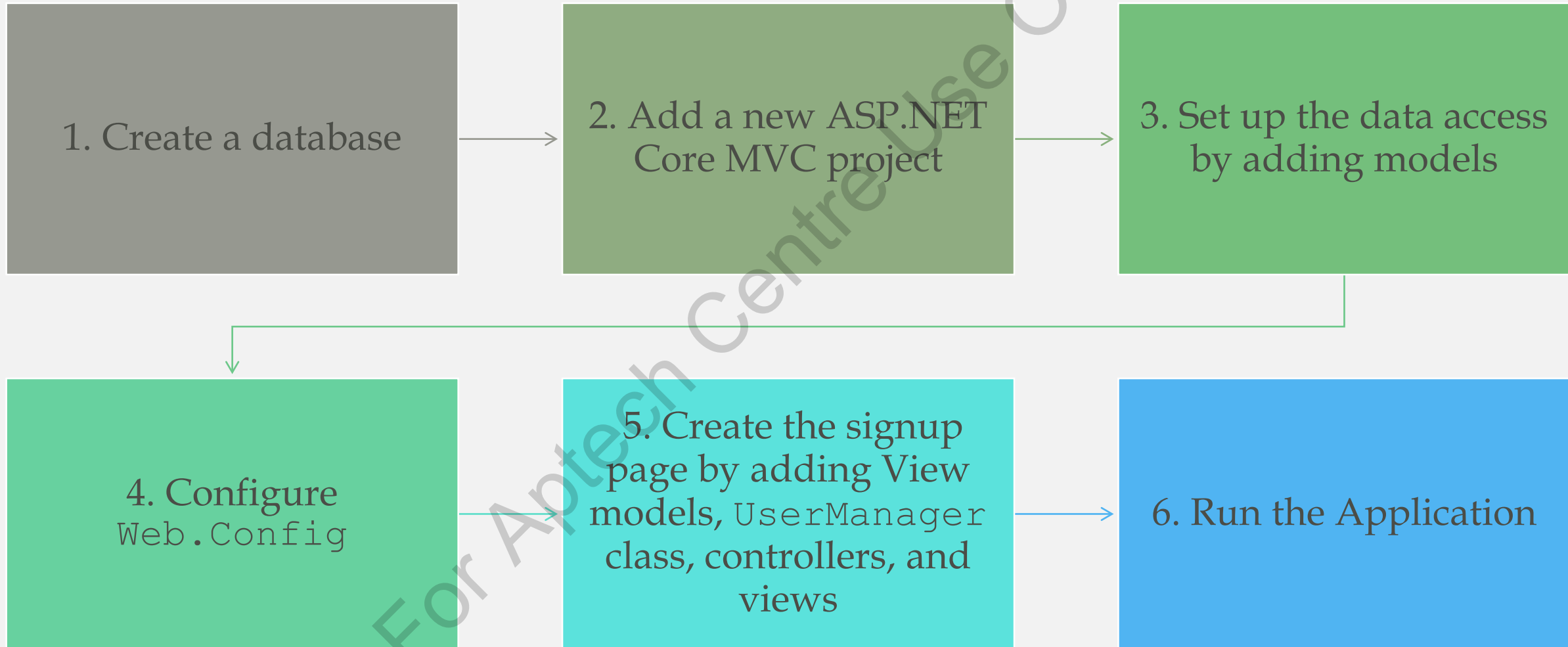
Server-side Validation

Implementing a Core MVC Application (1-2)



Core MVC Web Application

Implementing a Core MVC Application (2-2)



Summary

- ASP.NET Core uses ASP.NET Core MVC framework. It is lightweight, open source, and a highly testable presentation framework.
- In server-side validation, the server validates the input submitted by the user. Post validation, a dynamically generated new Web page sends the feedback back to the client.
- In client-side validation, all user inputs are validated in the user's browser itself.
- Data annotation attributes define the validation rules for properties of `model/ViewModel`.
- Commonly used data annotation attributes are `Required`, `Range`, `MinLength`, `MaxLength`, and `RegularExpression`.