**Steps to add and use configuration data in appsettings.json**

1. Add the settings in appsettings.json

"CompanyDetails": {

"Title": "Wai IT.",

"Slogan": "We can do it!!"

}

1. Create a settings Model class matching the entry in appsettings.json

public class CompanyDetails

{

public string Title { get; set; }

public string Slogan { get; set; }

}

1. Configure this setting for DI, by adding following line in Program.cs

builder.Services.Configure<CompanyDetails>(builder.Configuration.GetSection("CompanyDetails"));

1. Get the settings in any controller or service class using DI

private readonly CompanyDetails \_companyDetails;

public HomeController(

ILogger<HomeController> logger,

IOptions<CompanyDetails> companyDetailsOptions)

{

\_logger = logger;

\_companyDetails = companyDetailsOptions.Value;

}

1. Pass the model to any View that needs it

public IActionResult Index()

{

return View(\_companyDetails);

}

1. Use this in the View as follows

@model CompanyDetails

<h3><span class="text-secondary">@Model.Title</span>@Model.Slogan</h3>

**What is a ViewModel**

1. MVC View requires a Model to display or use dynamic data
2. This Model should ideally not be the same as the database model. Why?
   1. View may need data from multiple models.
   2. However, MVC View only take single model as parameter
3. What is the solution?
   1. Create a separate Model called the ViewModel
   2. Add properties which include all data required by the view
   3. In the controller create object of ViewModel and populate it with required data
   4. Pass the ViewMsodel to the View
   5. Use this ViewModel in the View cshtml
4. Graphical user interface, text, application, chat or text message

   Description automatically generated