

#1 ASP.NET & ASP.NET CORE

- A) ASP.NET from Online (Babu) + Extras (Aditav)
- B) ASP.NET from MyNotes (Basics only)
- C) DotNet Core / ASP.NET Core
 - ① .Net Framework vs Core
 - ② ASP.NET Core with Code
 - ③ Windows Service (a bit from Maestro)
- D) ASP.NET / ASP.NET Core Interview Questions
 - ① InterviewBit
 - ② Top 20 (DotNet Ticks)

#2 ADO.NET / Management Studio

- A) ADO.NET My Notes (Vikas first taught)
- B) Management Studio IDE (For MySQL Microsoft Server)
- C) Lecture-7: ADO.NET from Bafta

134.92A

B) ASP.NET From My Notes (Basics only)

#1 What is ASP.NET Part 1

④ In this session :-

- what is ASP.NET
- what is a web Application
- what other technologies can be used to build web applications
- What are the advantages of web application
- How ASP.NET web Applications work .

⑤ What is ASP.NET ?

⇒ ASP.NET is a web application framework developed by Microsoft to build dynamic data driven web applications and web services .

ASP.NET is a subset of .NET framework . A framework is a collection of classes .

ASP.NET is the successor to classic ASP (Active Server Pages) .

⇒ We can build different types of application using .NET framework . like console applications , web applications , windows applications , windows services etc . so ASP.NET is subset of that .NET framework to build web Applications and web services .

⇒ ADO.NET is also similarly subset of .NET framework . which helps us to pull data from databases .

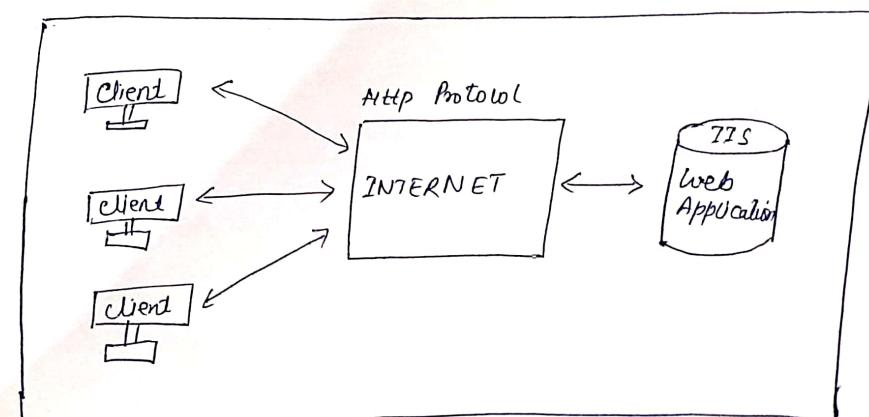
⇒ ASP.NET introduced in 2002 with ASP.NET 1.0 . Before that we had Classic ASP (Active server Pages)

③ Web Application is an application accessed by users using web Browser . e.g. chrome , mozilla , search in internet etc . Apple Safari

④ Other Technologies to build web Applications → PHP , Java , Perl , HERL .

⑥ Advantages of Web Applications :-

- ① Need to be installed only on web server , whereas desktop applications need to be installed in every computer , where you want to access them .
- ② Maintenance , support and patches are easier to provide .
- ③ Only a browser is required on the client machine to access a web application .
- ④ Accessible from anywhere , provided there is internet .
- ⑤ Cross Platform



5***

⇒ IIS is used to host Microsoft ASP.NET Web Applications.

so ASP.NET web applications runs under IIS.

⇒ When we search some URL then it sends request using HTTP protocol ('<http://120096.com>') and we get response from server (IIS).

⇒ Protocol describes how two or more items communicate over the internet.

⇒ We have different technologies but request to browser will only understand HTML. The web application has to emit HTML to be seen in browser at the end of the day.

#2 Creating ASP.NET Website - Part 2 :-

Since → it's an 2012 edition video here we are switching to learn ASP.NET from Udemy torrent downloaded course!

X X



Now Learning ASP.NET from
Udemy Torrent downloaded course

⇒ A Gentle Introduction to ASP.NET Web Forms
for Beginners

⑤ Total 8 ~~videos~~ folders in it.

#1 Course Information :- contains 4 videos

#1 Welcome :-

⇒ Here we are going to learn to create dynamic websites using ASP.NET for total beginners.

#2 Why ASP.NET .. Some Background :-

⑥ Take a look at the Big picture... The evolution of web development.

⇒ Basic HTML

⇒ HTML Forms ... introduction of tags for textboxes and buttons

⇒ server side (PHP ... Hypertext Preprocessor / ASP.NET ... stands for Active Server Pages)

⇒ Client side Programming (JavaScript)

⇒ static vs Dynamic Content

⇒ static content

→ Typically HTML files return the same thing each time requested

→ updates do content require updating the html file

⇒ Dynamic Content

→ ASP.NET Web Forms (.aspx) can contain dynamically generated content.

→ No need to update .aspx files for new content

→ all code runs on the web server the final result is sent back to the client as ordinary HTML



⑧ Learn what does ASP.NET do for us?

- ⇒ ASP.NET is the Microsoft platform for developing Web Applications. ASP.NET is part of the larger Microsoft .NET framework. C#, is another component.
- ⇒ It allows us to create dynamic websites using server web controls and applications.
- ⇒ Dynamic websites allow users to query databases in real time rather than loading static pages on each iteration.
- ⇒ Using ASP.NET you can create e-commerce (like Amazon.com) data driven portals and just about anything else you can find on the internet.

⑨ Different flavours of ASP.NET :-

- ⇒ There are different number of flavors of ASP.NET for example, Web Forms (Web sites and Web Application) Model-View-Controller (MVC) and the newest one Core.
- ⇒ This course is aimed at anyone who wants to create dynamic websites with ASP.NET web forms. Best of all you don't need to paste together a jumble of HTML and script code. Instead you can create full scale web apps using nothing but code (C#) and a design tool like visual studio.
- ⇒ MVC (Model View Controller) and Core offer different ways to build dynamic web pages. To some MVC and Core are cleaner and more suited to the web. To others it's a whole lot of extra effort with no clear payoff. Either way, you need a strong knowledge of basic ASP.NET before moving onto the newer technologies.



C) Dot Net Core / ASP.NET Core

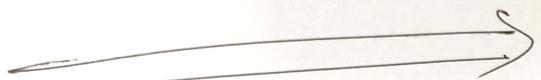
- ① .Net Framework vs Core
- ② ASP.NET Core with codes
- ③ Windows Service a bit for Maestros

Difference between .Net Framework and Dot Net Core :-

.NET Framework	.NET Core	.Net Framework	.Net Core
→ Old framework	→ New framework. Newly created by microsoft and maintained and created by microsoft & .Net community of Github.	→ We have all the <u>ASP.NET</u> components like. we have ① to ASP.NET Web Forms ② ASP.NET MVC ③ ASP.NET Web API ④ ASP.NET R ⑤ ASP.NET WPF ⑥ ASP.NET WCF ⑦ ASP.NET APIs/Services.	→ But in .Net Core (ASP.NET) we have only two components ② ASP.NET cores ③ Universal Windows Apps (UWA)
→ Not an open source. Only prebuilt libraries provided by microsoft.	→ It's an open source. Means you can download the code of .Net core and can do any changes in the code. we can also participate in the .Net community which is on Github.	→ .Net framework is <u>monolithic</u> . i.e. whenever we are going to publish an application, then whole complete framework library should be deployed on the client side.	→ In case of .Net core, it's <u>modular library</u> collection. So framework library is divided in multiple modules. So on the basis of our requirement we can use those libra to develop our applic
→ Only windows. .Net framework was only for windows. whatever app. is created in .Net framework can be published only on windows application. can be used only on windows operating system.	→ Cross Platform. Any application built in .Net core is cross platform can be used on multiple platforms. whether it be windows, linux or Macintosh. It's not platform dependent.		

ASP.NET

CORE



Learning. (Basics of ASP.NET You can click
in "ASP.NET - I" video link)

⇒ youtube source :- 'kudvenkat' Published - Dec 29 2018

↳ Asp.net Core Tutorial.

#1

Part 1 - Asp.net core Tutorial

Yt source → kudvenkat

5 In this course we will discuss

- All the basics, intermediate and advanced asp.net core concepts, that help us to build data driven web applications.
- At the end of the course we will be able to perform all the CRUD operations that is CREATE, READ, UPDATE & DELETE using SQL Server as our database.
- We will create ASP.NET core project from scratch as we build this project and progress from the course.
- We will discuss everything to know about data-driven web applications using this new asp.net core web framework from microsoft.

We will learn in detail about following:

- ASP.NET Core
- ASP.NET Core MVC
- ASP.NET Core Identity (for security)
- Entity Framework Core (for data access)

What is ASP.NET Core:-

- ASP.NET Core is a cross-platform, high-performance, open source framework for building modern, cloud-based, Internet-connected applications.
- It is the redesign of previous version of ASP.NET 4.x.
- So for this reason ASP.NET Core initially was called ASP.NET 5 but later renamed to ASP.NET Core 1.0. This offers several benefits and features.

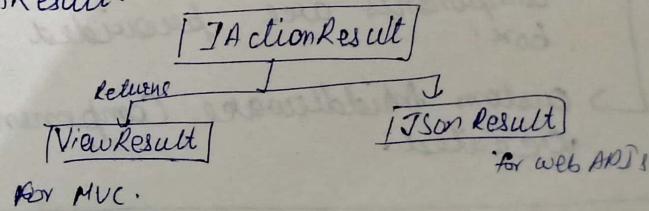
ASP.NET Core Best Benefits and Features :-

- ① Cross Platform → can be run across different platforms like
 - windows
 - Macos
 - Linux.
- Also .NET framework can be only hosted in IIS whereas ASP.NET Core can be hosted on
 - IIS
 - Apache
 - Docker
 - self-host in your own process

- From a development standpoint we can either use Visual Studio or Visual Studio Code for .NET Core applications. You can also use third party editors like Sublime.

One Programming Model for MVC & Web API

- One Unified Programming Model for MVC and Web API
 - Both the MVC Controller class and the ASP.NET Web API Controller class inherit from the same Controller base class and returns IActionResult.



Q code:-

asp.net core

③ Dependency Injection:-

→ asp.net core has built-in support for dependency injection.

④ Testability:-

→ since we've unified model for building web apps. and web APIs, unit testing of core applications is easy.

⑤ Open source:-

→ collaborated by vast community of open source developers.

⑥ Modular:-

→ asp.net core provides modularity with middleware components.

→ we use these components to compose both the request and response pipelines.

→ rich set of built-in middleware components are provided out of the box.

→ custom middleware components can also be created.



#3 Part 3 - Creating ASP.NET Core Web Application

Creating a new ASP.NET core Projects :-

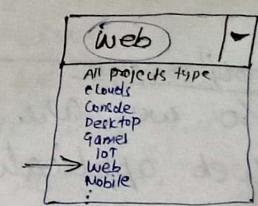
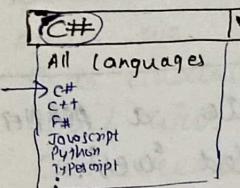
- Here we will discuss about:-
- ① Creating a new ASP.NET Core Project in Visual Studio.
- ② We will also discuss about different project templates that are available and what they do.

To create a New ASP.NET Project in Visual studio

→ Open Visual studio 2019

→ Go there you will see option to create a new project. Click on it.

→ Now choose as:-



→ So on selecting above mentioned categories. In 'Web' the ASP.NET provides us following Web Templates

- ① ASP.NET Core Empty
- ② ASP.NET Core Web API
- ③ ASP.NET Core Web App
- ④ ASP.NET Core Web App (Model-View-Controller)
- ⑤ ASP.NET Core gRPC service
- ⑥ Blazor Server App
- ⑦ Blazor Web Assembly App
- ⑧ Razor Class Library
- ⑨ A) ASP.NET Core with Angular
- B) ASP.NET Core with React.js

Supports C#, Linux, macOS, Windows, Cloud, service, web These are written in the field only along with the template.



⑪ ASP.NET core with React.js and Redux

⑫ NUnit 3 Test Project

⑬ some .Net framework in it / Just given but don't choose this

⑭ ASP.NET Web Application (.NET Framework)

⑮ WebDriver Test for Edge (.NET Core)

⑯ WebDriver Test for Edge (.NET Framework)

⑰ Container Application for Kubernetes.

at, windows, web, test
supports

⑱ "Empty" template

So we are going to create a ASP.NET core web application, so we've selected 'web'.

After selecting web we got ~~all~~ 'Web Templates' present in ASP.NET Core.

So in 'empty' template, it doesn't contain much template content.

It contains the bare minimum amount of content required to display Hello World in the browser.

Right now we will use this 'empty' template only to clearly understand how different pieces fit together in asp.net core web project.

⑲ Web Application (Model-View-Controller) Template :-

This generates a lot of content.

It creates models, views & controllers folder.

It also adds web specific things like CSS files, JavaScript files, Layout files and other resources needed for a website.

Because this template has many things in it when we create this MVC project.

So to understand simply first we need to start with empty project only so that we can create everything from scratch.

⑳ API Template :-

As the name implies, this template includes everything we need to create ASP.NET Core RESTful HTTP service.

An API does not have a user interface so all the website specific things like JavaScript files, CSS files, view files, layout files etc are not required for an API.

The data that Web API exposes is usually consumed by other applications.

So the API template just creates the 'controller' folder, and a sample controller within that folder.



cont--

- It does not create web user specific things like views, as they are not required for an API.

⑤ ASP.NET Core Web Application (Web App) Template :-

- This is 'web Application' template
- This template uses the new razor pages framework for building web Application, with razor pages coding page focused scenarios.
- It is easier and more productive.
- we can use this approach when we do not want the full complexity of asp.net MVC .
- we can think of it as a slimmer version of MVC framework.

⑥ Razor Class Library Template :-

- As the name implies, we use this template to create a reusable razor class library project.
- This project contains all the reusable user interfaces pieces, like data models, page model controllers, pages, razor views and view components.
- This razor class library project then can be used in multiple application.
- Any application that uses the Razor Class library can override the views and pages it contains.

⑦ Angular, Reactjs, Reactjs and Redux Template

- Finally we've these three templates, we use these three templates to Create ASP.NET Core web application in combination with Angular, or Reactjs, or Reactjs and Redux .

- ⑧ For now we will be using empty template only.



Q) Asp.net Core Empty Template :-

→ So select asp.net core Empty . and then click Next.

→ In the next step It will come to 'Configure your new project'

→ After this 'name your project'
let's say "EmployeeManagement"

→ choose your location folder

→ And then click Next.

→ It will come to 'Additional Information'

Let's select .NET Core 3.1 (long-term support)
in Target framework.

Configure for HTTPS }
 Enable Docker } → uncheck both of them

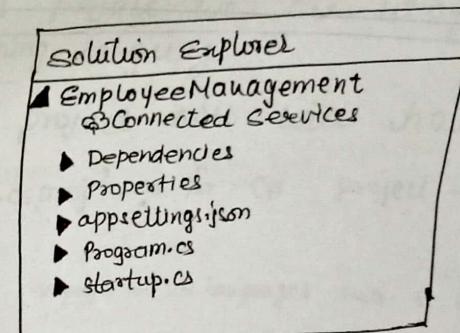
→ Now click on Create

→ Your project start being creating .

S) Project created :-

→ Now when you see in Solution explorer.

→ It will be as :-



→ Here we can see we've min^m amount of code generated by empty project.

→ All this project right now can do at the moment is display 'Hello world' in browser.

→ So let's run the project .

→ To run the project .

↳ Go in IDE 'Debug' menu

↳ Click on 'Start without debugging'

→ You see your IE browser opens up with url as :- http://localhost:8209/

and Hello world inside browser .

For turn
Ctrl F5
case.



③

→ In the next video we will explore and understand asp.net core project file.

#4 Part 4 - ASP.NET Core Project File :-

continued
from previous

same source

⑤ Our Solution Explorer from last part :-

→ In this project we are using C# as our programming language.

→ So the project will have extension as '.csproj' → for C# project → cs → csharp proj → project.

Similarly .vbproj for vb languages and so on.

→ The format of this file has been significantly changed than it was in framework.
The code has also been changed in .Net Core.

⑥ → In previous version in ASP.NET framework we used to 'Unload Project' before editing the project file. right click in today there is option for unload.

→ In core no need to unload the project to edit the project file.

→ To edit in .Net core, right click on 'EmployeeManagement' in solution explorer and then click on "Edit EmployeeManagement.csproj".
"Edit Project".

→ On doing that our 'EmployeeManagement.csproj' file comes out.



Scanned with OKEN Scanner

Q Initial code in :-

→ EmployeeManagement.csproj

Code as :-

In My Project created in VS 2019.

```
<Project Sdk="Microsoft.NET.Sdk.Web">  
  <PropertyGroup>  
    <TargetFrameworks> netcoreapp3.1 </TargetFrameworks>  
  </PropertyGroup>  
</Project>
```

file or folder references is no longer included in the project file.

→ TargetFramework

- specifies the target framework for the application
- To specify a target framework we use Target Framework Moniker (TFM).
- So value "netcoreapp3.1" is TFM of this project.

→ Naming conventions for Target Framework Monikers.

- .Net Framework → net → net451 (TFM for version 4.5.1)
- net → net472 (" " " 4.7.2")
- .Net Core → netcoreapp → netcoreapp3.0 (TFM for version 3.0)

5 cont.-

→ However in my VS 2019 there are no more than TargetFramework.

→ But in lecture video in VS 2017 there are other attributes, let's just slightly go through them.

→ AspNetCoreHostingModel :-

- Specifies how the application should be hosted
- InProcess or OutOfProcess
- At the moment we're 'InProcess' in lectures project in VS 2017.
- 'InProcess' hosting model we are using, means, host our ASP.NET core web app inside of the IIS worker process (w3wp.exe)
- 'OutOfProcess' hosting model forward web requests to a backend ASP.NET core app running in test server.
- The default is OutOfProcess hosting.
- We will learn in detail about this 'InProcess', 'OutOfProcess' & 'Kestrel' server in detail in upcoming videos.



Scanned with OKEN Scanner

S Code in lectures project for

Inside 'EmployeeManagement.csproj' is as:-

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework> netcoreapp2.2 </TargetFramework>
    <AspNetCoreHostingModel> InProcess </AspNetCoreHostingModel>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.App" />
    <PackageReference Include="Microsoft.AspNetCore.Razor.Design"
      Version="2.2.0" PrivateAssets="True" />
  </ItemGroup>
</Project>
```

S cont:-

→ Next we have 'PackageReference'

→ To PackageReference:

↳ Used to include a reference to the NuGet package that is installed for the application.

→ At the moment in our project file we're two package reference.

↳ Metapackage - (Microsoft.AspNetCore.App)

↳ A metapackage has no content of its own. It just contains a list of dependencies (other

S Metapackage continues

→ So go to solution explorer.
open 'Dependencies'

► Dependencies
 ► Analyzers
 ► Frameworks
 ► Microsoft.AspNetCore.App
 ► Microsoft.NETCore.App.

(My Project) [VS 2019]

► Dependencies
 ► Analyzers
 ► NuGet
 ► Microsoft.AspNetCore.App(2.2.0)
 ► Microsoft.NETCore.App(1.1.0)
 ► SDK

Lecture 1 [VS 2019]

This is that package.

→ When we expand this we will see the list of other dependencies or packages.

→ So this metapackage contains most of the things we need to for developing .NET Core Application.

→ In Metapackage we do not include version number in the code.

→ When the version is not specified, an implicit version is specified by the SDK.

→ We have also another package reference in our meta file. "Microsoft.AspNetCore.Razor.Design".

→ This package supports MSIX support for Razor and it is referenced by our meta package. Microsoft.AspNetCore.App.

→ Later we will see about other files in our solution explorer.



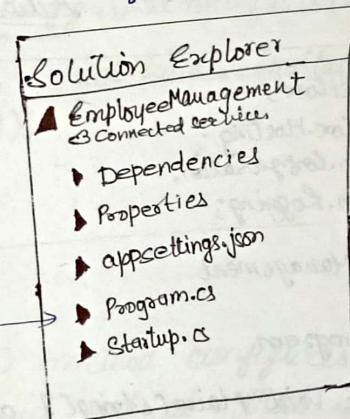
Scanned with OKEN Scanner

5 Part-5 :- Main Method in ASP.NET core

same source.

⑤ Solution Explorer:-

- We've already created a project "EmployeeManagement" previously of Asp.net core Empty web project.
- Solution explorer of our project looks like this.



⑥ In this file project.

- We've file named as "Program.cs"
- Within this class we have public static void main method.
- So we know that in any console application Program class is there with MainMethod in it as the entry.
- But here we are building Asp.net core application.



Q

→ So why do we have a Main() method in ASP.NET core web application?

→ It is bcoz, an ASP.NET core application initially starts as a console application.

→ code in 'Program.cs' is as:- *Lecture Project code*

```
using Microsoft.AspNetCore;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Logging;
namespace EmployeeManagement
{
    public class Program
    {
        public static void Main(string[] args)
        {
            CreateWebHostBuilder(args).Build().Run();
            //CreateHostBuilder (In 2019) (Rest same up to here)
        }
        public static IWebHostBuilder CreateWebHostBuilder(
            string[] args) =>
        {
            webHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>();
        }
    }
}
```

VR3065255

Q The code is similar in my VS2019 project also.

→ Just changes as below :-

In 'Program.cs' file..

```
public static IHostBuilder CreateHostBuilder(string[] args)
{
    Host.CreateDefaultBuilder(args)
        .ConfigureWebHostDefaults(webBuilder =>
    {
        webBuilder.UseStartup<Startup>();
    });
}
```

Q This Main() method configures ASP.NET core and starts it and at that point it becomes an ASP.NET core web application.

→ So in main method we are creating and running 'CreateWebHostBuilder' which returns an object 'IWebHostBuilder'. And this method called inside main method and after that we've applied 'Build()' methods in it, which builds the web host, that host this asp.net web application. And that web host is again calling the run method which runs the web applications and start listening for incoming http request.



Scanned with OKEN Scanner

#6 Part 6: ASP.NET Core InProcess Hosting

- Also we've there st 'UseStartup<Startup>()' in the fn. definition code. This startup is just that 'Startup.cs' file you can see in solution explorer which gets generated when you create this project.
- In 'Startup.cs' file there is Startup class which has 2 important methods viz 'ConfigureServices' and 'Configure'.
- "ConfigureServices" as the name implies configures the services required for our application.
- 'Configure' this method configure our application request, processing pipeline.

(S) As we saw previously, our application starts at main method for our .NET Core web application.

⇒ 'CreateHostBuilder' method is called in our main method which in turn called 'CreateDefaultBuilder' which create web host with Default builders.

(S) Some of the Tasks that CreateDefaultBuilder() perform

- ① Setting up the web server
- ② Loading the host and application configuration from various configuration sources and
- ③ Configuring logging

(S) An ASP.NET core application can be hosted :-
→ InP InProcess or
→ OutOfProcess.



InProcess Hosting

→ To configure InProcess Hosting.

↳ In your ASP.NET Core Project - fill implement this following element:

```
<aspNetCoreHostingModels> InProcess </aspNetCoreHostingModel>
```

↳ we see this in our core project file lecture in previous lectures.

→ So 'CreateDefaultBuilder()' method calls 'UseIIS()' method and host the app inside of the IIS worker process (w3wp.exe or iisexpress.exe)

→ InProcess hosting delivers significantly higher request throughput than OutOfProcess hosting.

→ Let's now print the name of our server instead of 'Hello World'.

we will make use of below code :-

```
System.Diagnostics.Process.GetCurrentProcess().ProcessName
```

→ The 'Hello world' method comes from 'Startup.cs' file.
↳ so you can see there.

↳ Initialize code in Startup.cs (In lectures Project vs 2013)

↳ code as in Startup.cs file

```
Using System;  
Using System.Collections.Generic;  
Using System.Linq;
```

namespace EmployeeManagement

```
{ public class Startup
```

```
    { public void ConfigureServices(IServiceCollection services)
```

```
        {
```

```
            public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
```

```
            {
```

```
                if (env.IsDevelopment())
```

```
                { app.UseDeveloperExceptionPage();
```

```
                }
```

```
                app.Run(async (context) =>
```

```
                { await context.Response.WriteAsync("Hello world!");
```

```
                };
```

```
            }
```

```
        }
```

```
    }
```

```
}
```



However there is slight change in my project vs (2019)

→ changes code in my 'Startup.cs' file (vs 2019)

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseRouting();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapGet("/", async context =>
        {
            await context.Response.WriteAsync("HelloWorld.");
        });
    });
}
```

→ Rest of the code remains same.

So instead of showing hello world if you want to show your server then just write that code there in appname.

→ It will show in browser as iisexpress.

→ Because we are running our project from visual studio. By default visual studio uses iisexpress to host & run our application.

→ iisexpress is customized & lightweight version of IIS. We are using iisexpress during development only & not for production. For production we use IIS.

→ If we've used IIS then in browser we would have got (w3wp)

With OutOfProcess hosting :-

→ It has two web servers

↳ Internal & External web servers.

→ The internal web server is Kestrel

→ The external web server can be IIS, Nginx or Apache depending upon the operating system you're.



Q) What is Kestrel?

- Cross-Platform Web Server for ASP.NET Core supports.
- Kestrel can be used, by itself as an edge server. Directly processes incoming http request from client.
- The process that used to host the app is dotnet.exe.
- ⇒ Right now we are running our code in Visual Studio. Visual studio by default uses iisexpress process.

Q) So CLI in dotnet core is used for running project from command line.

- So open command prompt. Navigate to that project using cd. e.g. if your project name is 'EmployeeManagement' then navigate as such.

```
c:\Projects\EmployeeManagement\EmployeeManagement>dotnet run
```

To run the Project.

⇒ It will give output as:-

```
Hosting environment: Development  
Content root path: c:\Projects\EmployeeManagement\EmployeeManagement  
Now listening on: http://localhost:5000  
Application started. Press Ctrl+C to shut down
```

So here important thing to note is, when you use ~~Kestrel~~ dotnet core application using dotnet core CLI, Kestrel is used as web server.

localhost:5000 → type in browser

Gives output in browser as:-

dotnet

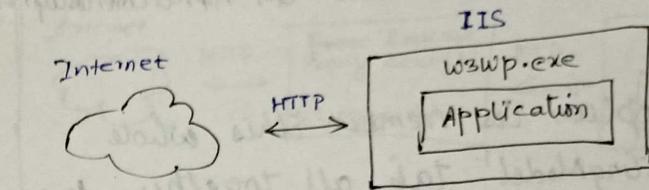
So now this server (Kestrel) using dotnet (dotnet.exe)



#7 Part 7 - ASP.NET Core Out of Process Hosting

Same Source

(5) Summary :-



(6) With Inprocess Hosting :-

- Application is hosted inside the IIS Worker process
- There is only one web server
- From a performance standpoint, Inprocess hosting is better than OutofProcess hosting.

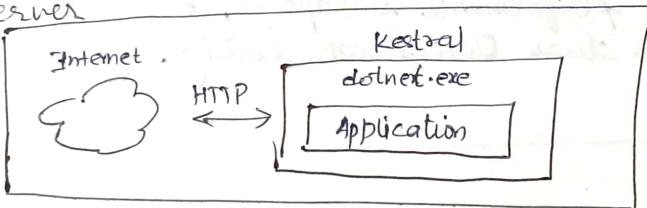


Q OutOfProcess hosting :- configuration

- To configure OutOfProcess hosting, inside 'EmployeeManagement.csproj' instead of 'InProcess'
Give value as 'OutOfProcess' in AspNetCoreHostingModel tab.
- The other option is remove this whole 'AspNetCoreHostingModel' tab all together from project file, because the default is 'OutOfProcess' hosting only.

Q OutOfProcess hosting : (servers) :- → servers (see previous lectures)

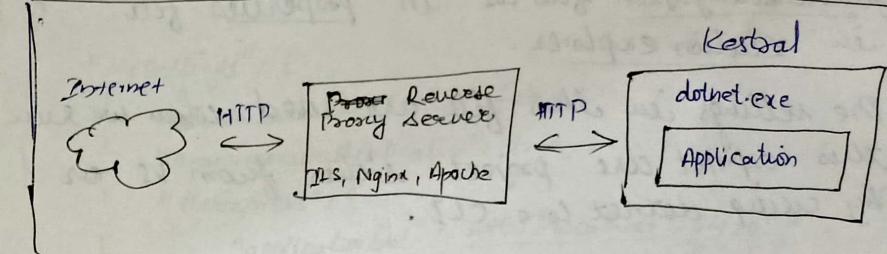
Kestrel can be used as the internet facing web server



Now after configuring from OutOfProcess hosting, run the project from CLI, now our browser shows is [dotnet] since using 'OutOfProcess'.

S cont:-

→ Kestrel can be used in combination with a reverse proxy.



Q In Process v/s Out of Process Hosting

In Process

→ Process name is w3wp.exe or iisexpress.exe

→ Only one web server

→ Better for performance

Out of Process

→ Process name is dotnet.exe

→ Two web servers

→ Penalty of proxying requests between internal & external web servers.



#8 Part 8 - Asp.net Core launchsettings.json file

same source

Launchsettings.json file is in 'properties' file in solution explorer.

→ The settings in this file are used when we run this asp.net core project either from VS or by using dotnet core CLI.

→ This file is only required on our local development machine. We do not need it for publishing our asp.net application.

If there are certain settings that you want your asp.net core application to use when you publish or deploy your app; store those settings in this appsettings.json file.

→ We usually store application configuration setting in this file (appsettings.json).

→ We also have environment specific appsettings.json file.

e.g. appsetting.staging.json file for staging environment.

Similarly for development environment we've appsetting.development.json.

Initial code in 'launchsettings.json' file C:\vs2017\lectures

Code as :-

```
{  
  "iisSettings": {  
    "windowsAuthentication": false,  
    "anonymousAuthentication": true,  
    "iisExpress": {  
      "applicationUrl": "http://localhost:15410",  
      "sslPort": 0  
    }  
  },  
  "profiles": {  
    "IIS Express": {  
      "commandName": "IISExpress",  
      "launchBrowser": true,  
      "environmentVariables": {  
        "ASPNETCORE_ENVIRONMENT": "Development"  
      }  
    },  
    "EmployeeManagement": {  
      "commandName": "Project",  
      "launchBrowser": true,  
      "applicationUrl": "http://localhost:5000",  
      "environmentVariables": {  
        "ASPNETCORE_ENVIRONMENT": "Development"  
      }  
    }  
  }  
}
```

code is also almost same in my vs 2019 just changed

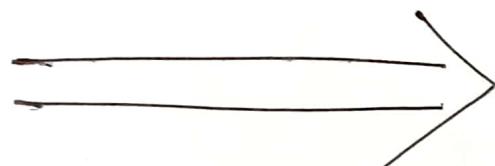


Scanned with OKEN Scanner

In launchsettings.json file :-

- In the code you can see we've got two profiles
 - ① IIS Express and (By default, ~~run if we run using VS only~~ ^{Launch} ~~IISExpress, not in web.~~)
 - ② EmployeeManagement (the name of our project) (~~when runned using CLT~~ ^{Port 5000} ~~in mode.~~)
- So when we run our visual studio for vs itself, then this 'IIS Express' profile is used.
You will see port number in url as '15410' which is in the 'iisSettings' code only for IIS Express profile.
- We can change it the url if we want to.
- Also in 'IIS Express' profile we've used 'Development' we could also have used staging or production depending on which environment, we are trying to run this project.
- Similarly if we run from dotnet core CLT 'EmployeeManagement' Profile is runned. This time Port no is 5000 you can see in browser. ^{ProcessName} ~~dotnet~~.

Windows
Service in
.NET Core



#① Windows Service in .NET Core :-

There are 3 ways in which you can create a windows service in .NET core.

- ① The "Microsoft" way
- ② The "Powershell" way
- ③ The ".NET Core Worker" way.

In our Bafa Project 'Dg-Haestos'

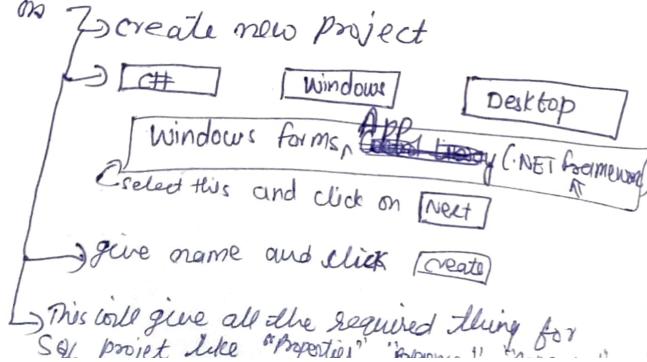
we have used "The ".NET Core worker" way,
so we will learn more detail about this.

ADO.NET

Some first Basic
know thing.

How to make ADO.net Project in .NET Framework and not in .NET Core.

While making ADO.net Projects using form in visual studio 2019 (since now .NET Core has come) we got confused which webform to choose. so just click on



If you choose Windows Forms App by default it's windows form core 2 not windowform .NET framework.

How to make connection with the database !-

② We can make connection outside visual studio also, using SSMS (Microsoft SQL Server) by app adding the server source name and its login password & user name, and write queries there only if you want to.

③ But here we will see how to make connection inside visual studio 2013.

→ But before that you require

- ① Server name (Data Source)
- ② User name
- ③ Password

Most often Microsoft SQL Server (SQL Server) as Data Source if SQL

If the server is authenticated then.

Example:- Kardode Vikas, has sent me the following details which are required for making the connection with visual studio.

Initial Catalog = NDF; Data Source = wsdne64zeot.sdi.corp.bankofamerica.15001; User ID = mdfuser; Password = M@d6f&tir;

→ This is also called as connection string.
Here it is

Data Source → Server name
(Initial Catalog - NDF) is database name.



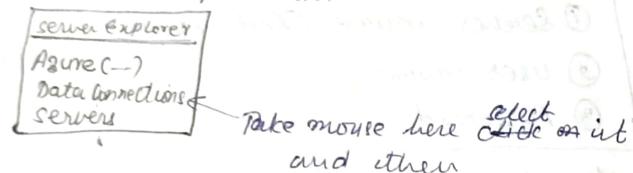
So just open the Visual Studio 2013
2015

Make any project in which SQL server's connection is required. mostly ADO.NET projects.

⑧ click on **VIEW** menu

⑨ click on "server explorer"

⑩ A side sub menu opens up as "server explorer".
Here you will see options as:-



⑪ Right click on mouse, and then click on "Add Connection."

⑫ 'Add Connection' small window appears up.

→ Data Source → mostly Microsoft SQL Server Database only we use (SQL Client)

→ Data Server Name → Add your server name which Vikas or people has given to you to use → upto 1500 → See previous page copy full upto 1500.

→ Use SQL Authentication → For adding username & Password

After that → You can select or enter database name if you want to select specific database name.

⑬

→ However here we are using 'MDP' database which Vikas has sent.

→ And then Test Conn.

→ And then click **OK**.

⑭ Your database will get connected with your project.

⑮ Now you can expand the 'Data connection' and see in it all your table name, Index name, stored procedures in it etc.

⑯ To open SQL Query writing board on visual studio, just go in one of the 'Data base' inside 'Data connection' right click on it. and then click on 'New Query'. A new page 'SQLQuery1.sql' opens up, there you can write SQL Query and check.

→ So this is all about How you can connect.

107 A simple ADO.NET Project

(Y) The Concept Academy
By Ideas Solution
For Understanding
Introduction to ADO.NET || Data Providers || Connection
Command || Data Reader.

⑤ We will deal with many concepts with ADO.NET.

→ Here we will understand in a very basic way: A very basic project overview to understand.

→ We've seen before, ^{pages} only how to create connection with database and creating a simple Windows Forms Application in visual studio 2013 or visual studio 2019.
(No core then)

core now so to reduce confusion
see back pages to clear
Net framework one and
not core forms)

⑥ This here we are following 4t lecture series from The Concept Academy By Ideas Solutions.
Video 1/7 (1st out 87). Introduction to ADO.net || Data Provider || Connection || Command || Data Reader.



② Let's start with our very basic project.

→ Once you create Form window Form Project & made DB connection.

→ So before we start let's do some brush up with ADO.NET.

→ ADO.NET is part of .NET framework which helps us to establish a connection w/in between an Application and the Data Source. Data source can be SQL Server, Access file, XML, Azure etc.

→ ADO.NET consists of classes that can be used to connect, retrieve, insert and delete data.

→ All the ADO.NET classes are located into 'System.Data.dll' (reference) or in 'System.Data'. If XML is your data source then you've to use 'System.Xml.dll' (reference) or 'System.Xml'.

→ In simple words:- Bridge between application and database is called ADO.NET.

③ We've two separate topics to understand ADO.NET.

- ④ Data Providers (in this project)
- ④ DataSet

⑤ Here we will be discussing and making projects based on 'Data Providers'.

→ Following are the core object of Data Provider

Object	Description
① Connection	It is used to establish a connection to a specific data source. (This is first step in ADO.NET i.e. to make connection b/w application and database).
② Command	(After making connection you obviously would like to run some command or perform some query). It is used to execute queries to perform database operations.
③ DataReader	(After this comes third step, we can either use DataReader or DataAdapter, we will see in our project). It is used to read data from data source. The DbDataReader is a base class for all DataReader objects.
④ DataAdapter	It populates a dataset and resolves updates with the data source. The base class for all DataAdapter object is the DbDataAdapter class.

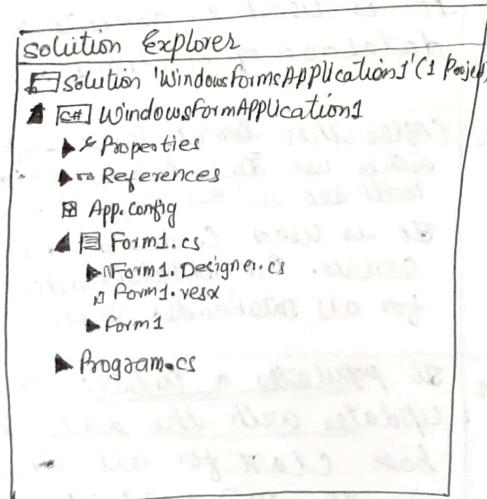


(S) However here in this project we will be using DataReader only along with connection and command. DataAdapter we will use that in another next project.

→ DataReader → is used to read the data line by line and stores those data in the object of DataReader.

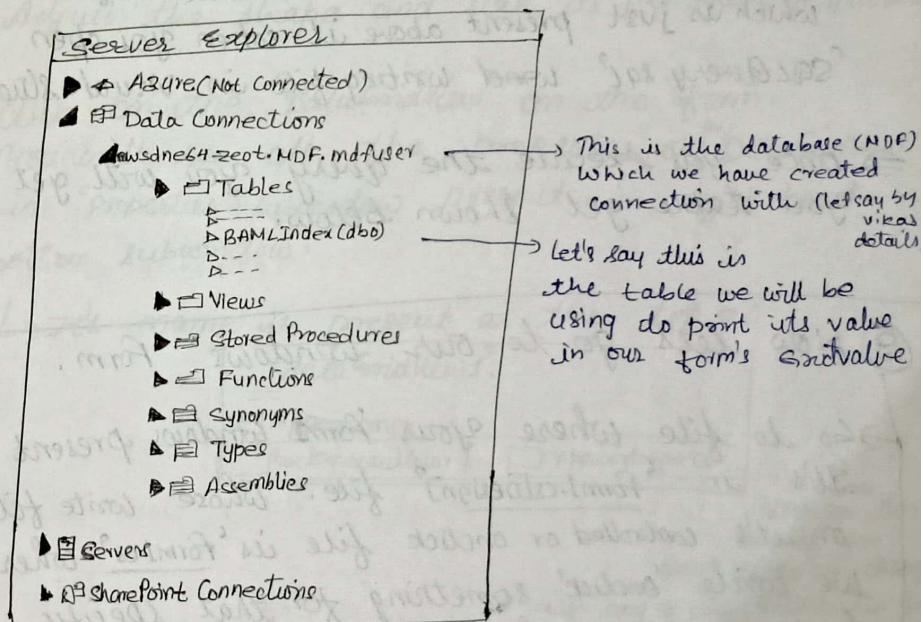
So now let's understand this by our practical project.

→ After you create WindowsFormApplication project. Set project name as "WindowsFormApplications". Once your project is created your Solution Explorer will look like this (present in right hand side sub window in visual studio IDE)



(S) On the other hand inside 'Server Explorer' subwindow present at the right left hand side of the visual studio where our database is present.

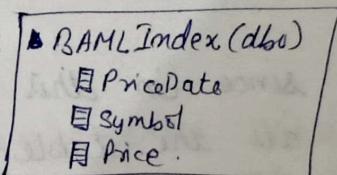
Let we've database called NDF (from Vikas Project) it will look like this.



This is the database (NDF) which we have created connection with (lets say by vikas details)

Let's say this is the table we will be using so point its value in our form's GridValue

→ If you expand the Table



→ we will be using this columns to be mapped in our application and show this table in our Gridview in forms.

⑦ So now we've got Table and its data and we know its attributes name also. However you can check its all data items also by writing query as

"Select * from RAMIndex" inside "SQLQuery.sql" file, and then click  → Execute(Ctrl+Shift+E) which is just present above it when you open "SQLQuery.sql" and writing file in visual studio.

⇒ Once you execute the query you will get your table get shown below.

⑧ Now let's go to our windows Form.

→ Go to file where your Form windows present. It's in "Form1.cs[Design]" file. Whose write file or its controlled or onclick file is "Form1.cs" where we write "onclick" something for that specific toolbar or the whole form.

→ So since in this project we want to get all the table datas to be shown in our Form so we need a Gridview tools.

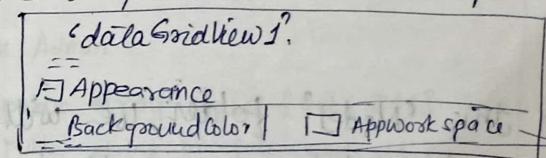
Also click on 'Toolbox' (or go to  menu and then click on 'Toolbox'). You will see a whole list of Toolbox appears on your left sub-window.

→ Make sure you are on "Form1.cs[Design]" file opened only. Otherwise 'Toolbox' won't show any tools in it.

→ Now just drag and drop "DataGridView" toolbox from toolbox to the "Form1". Adjust the shape and size as per your requirement.

→ Click on the "DataGridView" on the form. You will see all the properties of "DataGridView" in "Properties" subwindow present in Right hand bottom subwindow.

→ Its name is present at the top as

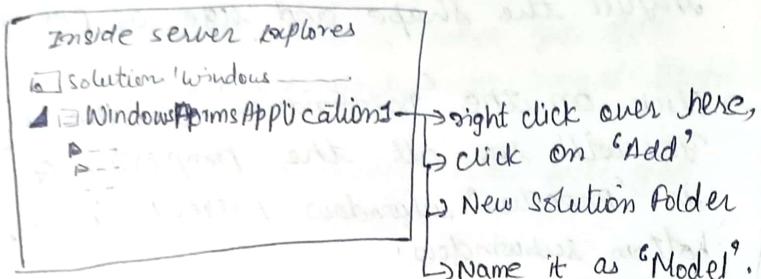


→ So now as we speak earlier that table we want to show in our DataGridView, but first we want some POJO class mapper that will map the database contents data or attributes and get them.

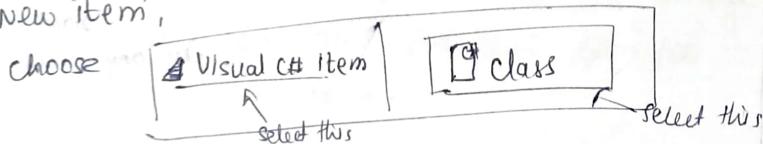


→ So for that we let's create or Add a new folder named as 'Model' where we will put all our classes.

→ So right click on



→ So in this 'Model' folder we will again right click on it, click on 'Add', then click on 'New item'.



and after selecting class give name as 'Admin.cs' and then click on Add.

→ A new class file will get added in your 'Model' folder.

→ Similarly add one more class name ist as 'AdminOrderLayer.cs'.

③ So now in this 'Admin.cs' file we will map the column names of our table.

→ We know our table has following columns

	PriceDate	Symbol	Price	
1	2014-09-10	.MSFT-NL	113.7000	→ sample data

→ So inside 'Admin.cs' our code will be as:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApplication1.Model
{
    class Admin
    {
        public string symbol { get; set; }
        public string priceDate { get; set; }
        public int price { get; set; }
    }
}
```

→ So here 'Admin' class will work as entity ('Table') and its properties data will act as its columns.

Now let's come to other class inside 'Model' folder only i.e. 'AdminLogicLayer.cs' file. This will work as our business logic.

In this file we will write our code where we want to fetch our data.

So our initial code in 'AdminLogicLayer.cs' file will be as:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient; This two are just added manually for our Ado Project.
```

```
namespace WindowsFormsApplication.Model
```

```
{
```

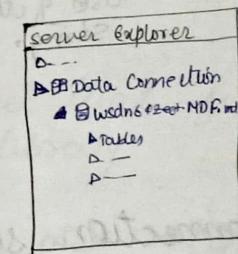
```
    class AdminLogicLayer
```

```
{
```

```
}
```

Now our next step in this file is to create a connection.

As we've already connected with our database just click on that database which you are using inside server explorer sub window only inside 'Data Connection'



Just select this database

After selecting this you will find in right Hand side bottom sub window will give properties of this database so it will give data such as

Connection	Connection String	Data Source
'Provider'	—	—
'State'	—	—
'Type'	—	—
'Version'	—	—

Just double click on here to select the full text or copy it;

so now you will get your connection string, which is required to set up connection in our AdminLogicLayer



Scanned with OKEN Scanner

The Connection String we got as:-

→ Data Source = wsdns4zeot.8di.corp.bankofamerica.com, 15001; Initial Catalog = NDF; UserID = mdfuser; Password = -----
you can edit here your password while making use of this connection string.

Now as we got our connection string so now just go to 'App.config' file and put the tag for connection string in it as follows:- code in 'App.config' file

<?xml version="1.0" encoding="utf-8"?>
<configuration>
 <startup>
 < startup>
 <connectionString>
 <add name="db" connectionString="Data Source = wsdns4zeot.8di.corp.bankofamerica.com, 15001; Initial Catalog = NDF; Persist Security Info = True; UserID = mdfuser; Password = M@d6fsvir" providerName="System.Data.SqlClient"/>
 </connectionString>
 </ startup>
 </startup>
</configuration>

Now after this is again in back in 'AdminLogicLayer.cs' file we also need to add one more library called 'System.Configuration', but we won't get that library here because we are using desktop application.

So to get that first we need to add that library.

To add that 'Configuration' library just right click on 'Reference' file present in solution explorer.

And then after goto 'Add Reference' click on it. We will see a window pop up named as 'Reference Manager'.

Over there click select framework or assemblies on left hand side. And then search for 'Configuration' on right hand top side.

You will get two search results as

System.Configuration
 System.Configuration.Install

mark tick or select both of them and then click on [OK].



Now you can see in your reference' on expanding it. There is that system present now 'configuration' one.

So now add these two system in our AdminLogicLayer.cs file.

So code as:-

```
using System.Configuration;
```

Now after this we will make connection string in our 'AdminLogicLayer.cs' file.

So code as:-

```
class AdminLogicLayer
{
    private string connectionstring = ConfigurationManager.ConnectionStrings["db"].ConnectionString;
}
```

we are using this name from that App.config file.

we do this instead of writing the whole connection string here because in future when we shift to other PCs or when we change the server name again, then we've to do copy/paste from all ones.

So by using 'config' way we just only have to do changes in string connection string in only 'App.config' file. So it will be time saving.

So now we will method for GetData. Also since we want to get data in the form of list. Then lets make list of Admin's Pojo class only.

So inside AdminLogicLayer class after connection string code as

```
private string connectionstring;
```

```
public List<Admin> getAllAdmins()
{
    List<Admin> li = new List<Admin>();
    try
    {
        catch (Exception)
    }
    return li;
}
```

Since we can make or open connection in try only in case connection related problem occurs then



Scanned with OKEN Scanner

④ So now other steps will be done inside try block only.

→ Step 1 :- Make connection and then open the connection so code as :-

```
try {
```

```
//connection.. Step 1
```

```
SqlConnection conn = new SqlConnection(connectionString);  
conn.Open();
```

// Step 2 : Make SQL Command Object and store the query.

```
SqlCommand cmd = new SqlCommand("Select * from BANLIndex",  
                                conn);
```

// Step 3 : DataReader which reads the data line by line

```
SqlDataReader rdr = cmd.ExecuteReader();
```

- Now after this we will make while loop which will run until our rdr has got data records in it.

```
while (rdr.Read())
```

```
{  
    Admin ad = new Admin();  
    ad.price = Convert.ToInt32(rdr.GetValue(0));  
    ad.symbol = rdr.GetValue(1).ToString();  
    ad.priceDate = rdr.GetValue(0).ToString();
```

```
    li.Add(ad);
```

My
numbers
specifying

⑤ Column position in table as :-

Price	Date	Symbol	Price

1st position

1st position

2nd position.

→ So we have written according

as :-
ad.price = rdr.GetValue(2);
symbol = rdr.GetValue(1);
priceDate = rdr.GetValue(0);

0, 1 & 2

in the while loop
while filling the
list in previous page

→ So these are according to indices.

⑥ Now just save the file upto now whatever you've made.



Scanned with OKEN Scanner

⑧ Now to call this logic.

→ Just go to 'Form1.cs[Design]'.

→ Click anywhere on the form, but not on the DataGridView, it will navigate you to the file 'Form1.cs'. and also a function will be build up as 'private void Form1_Load_1(object sender, EventArgs e)'.

→ So in this function we have to write our logic of adding list in the DataGridView..

→ But first we need to import 'LogicLayer' 'AdminLogicLayer' namespace in Form1.cs file so Code as:-

```
using WindowsFormsApplication1.Model;
```

And code inside function as:-

```
private void Form1_Load_1(object sender, EventArgs e)
{
    AdminLogicLayer obj = new AdminLogicLayer();
    dataGridView1.DataSource = obj.getAllAdmin();
    assigning list to dataGridView
}
```

⑨ So now just run the code!

→ Everything will run fine and there will be Table list shown in DataGridView in Form1.

→ So this is all about ADO.NET way of writing code.

⑩ So overall file used are:-

① In References → Just add Configuration

② Model → Admin.cs → (our pojo class/entity class to map table)
→ AdminLogicLayer.cs → (where we write business logic
i.e. make connection, command & data reader)

③ Form1.cs[Design] → Our form where we drag & drop datagridview tools. (Preview)

Form1.cs → Same as above (But this writes code when you double click on above form it points here with its respective tool or form function) (You can do it manually also)

④ App.Config → Added connection string tag to save time and efficient

⑤ SQLQuery.sql → Write queries here and execute them to check dates and other stuffs checking



ADO.NET

- From youtube video Playlist 'kudrenkat' ADO.NET for beginners.
(2012 edition)
we've total 21 videos

Some other important learning in Basic Visual Studio (Some must know thing about VS 2013 & ADO.NET and connection with server database How to?)

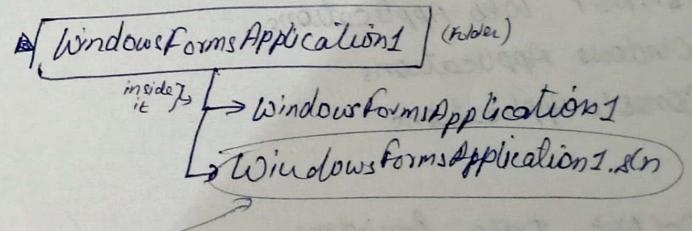
① Use **F10** → debug in visual studio 2013 for debugging purpose to go to next step.

→ You can add the break points just as you do in Java side. And then run **Start** the program, it will reach to the breakpoints and rest you know how to debug.

→ **F11** → In case you are just putting the breakpoint and after that you want to check each line one by one without applying breakpoints to all of them. And then after just click on **Continue** to finish debugging and run the project.

Q While 'opening' any existing saved project from your system into visual studio, Click on **File** and then click on **Open** and then click on 'Project/Solution' windows folder opens up Just go to the project. You should click on that project. After that select that project **.sln** file and open it. The normal file project folder won't open in visual studio.

e.g. your project name



Select this and then click on **Open**.

Then only your project will open in visual studio.

Q 1 What is ADO.NET - Part 1

→ Here first we've created a simple ASP.NET applicat

S → What is ADO.NET

↳ What are .NET Data Providers.

S → What is ADO.NET ?

⇒ ADO.NET is not a different technology. In simple terms you can think of ADO.NET, as a set of classes (framework), that can be used to interact with data sources like Databases and XML files.

This data can be then be consumed in any .NET application. ADO stands for Microsoft ActiveX Data Objects.

⇒ The following are, a few of the different types of .NET applications that use ADO.NET to connect to database, execute commands, and retrieve data.

ASP.NET Web Applications

Windows Applications

Console Applications

⇒ Dot Net Data Providers :-

Data Provider for SQL Server - System.Data.SqlClient

" " " Oracle - System.Data.OracleClient

" " " OLEDB - ". ". OleDb (Access/MS)

" " " ODBC - ". ". Odbc



Q#2 How to fill combo box in C# Windows application from database // Dataset // SqlDataAdapter.

Lecture 2/7 videos
Some Yt Syra
Continue

As we studied in our last class project, that we can target either dataset or DataProvider.

⇒ We started DataProvider in last lecture.

⇒ In DataProvider we have 4 core objects

- ① Connection
- ② Command
- ③ DataReader
- ④ DataAdapter.

⇒ In the last project we've used Connection, Command and DataReader.

⇒ Today we shall move towards 'DataAdapter'

⇒ There is some differences in 'DataReader' and 'DataProvider'

⇒ DataReader → directly reads from the data source (database)

whereas

⇒ DataAdapter → It reads the data and populates the data in 'Dataset'. And it works in 'Disconnected' Architecture. I.e. even if your data source gets disconnected from your application there will be still this data remain present in 'Dataset', which are loaded in memory already.

But for DataReader, connection must be there between your application and data source. If there is no connection then it won't be able to read the data.

Now let's go to our project work.

→ We will continue from previous project only.

→ This time we will use DataAdapter.

→ So just go to 'AdminLogiclayer.cs' file, now this time we will get data by using Adapter.

So just write this code in 'AdminLogiclayer.cs'

as

```
using ___.  
namespace ___.  
| class ___.  
| {  
| | private string connectionString = ___.  
| | public List<Admin> getAllAdmin() { ... } // DataReader code.  
| |}  
| }  
}
```



(S) So let's make function as GetAUData() :- just after getAUAdmin() fn:

→ Code as:-

→ Here we will be returning dataset and not list of Admin for mapping.

in AdminLogicLayer

→ Our code

```
// DataAdapter to get all data
public Dataset GetAUData()
{
    Dataset ds = new Dataset(); // creating object of dataset to store our
                               // data and we will return it.

    // Make connection:
    SqlConnection conn = new SqlConnection(connectionstring);

    // Open connection and other command & filling of dataset in try catch
    // block only do.
    // block only do.
    // Adapter stored Command here
    // DataAdapter from the query.

    try
    {
        conn.Open(); // Open the connection
        // Make SQL query in data adapter, using query & connection name.
        // Adapter stored Command here
        // DataAdapter from the query.
        SqlDataAdapter sda = new SqlDataAdapter("select
                                                * from BMIndex", conn);

        // Fill the data in dataset
        sda.Fill(ds);
    }
    catch (Exception)
    {
    }

    return ds;
}
```

(S) Now this code in 'AdminLogicLayer.cs' is fully correct.

→ So if you run this code in Asp.net Web Form then it will run fully fine.

If we have below code like in DataGridView

→ i.e. Code in Form 'Form1.cs'

(Also drag and drop DataGridView to show this data)

```
private void Form1_Load_1(object sender, EventArgs e)
```

```
{
    AdminLogicLayer obj = new AdminLogicLayer();
    // code to populate from datareader from last project.
    dataGridView1.DataSource = obj.getAUAdmin();

    // code to populate from DataAdapter on dataset in gridview.
    dataGridView2.DataSource = obj.GetAUData(); // This code won't run
                                                // here in windows
                                                // but can run in
                                                // ASP.NET webform

    // dataGridView2.DataMember = "BMIndex";
    // For ASP.NET we can use this also
    // just leave this.
}
```

→ so when you run the project you will see your DataGridView in the form is giving Blank.

→ This is because whatever data are being fetched from our Dataset, we need to specify, which table you want to show.



→ 'Dataset' is basically the collection of tables.

→ So the data we're in Dataset is of 1st Table. So we need to specify the index.

→ So we will write codes as inside Form1.cs inside Form1_Load_1 method as:-

```
dataGridview2.DataSource = obj.GetAllData().Tables[0];
```

⑧ Now we will see what is the benefit of using Dataset.

→ For example we've multiple tables in our database and we want to show other tables as well.

→ Let's say we've query as in DataAdapter as:-

⑤ For showing multiple table queries it changes code as follows:-

inside 'AdminLogicLayer.cs' file inside GetAllData method

```
SqlDataAdapter sda = new SqlDataAdapter("select * from Table1  
select * from Table2", conn);
```

The rest of the code remains same

→ Add one more Grid DataGridView3 to just show the data of this second table in Dataset.

so code in 'Form1.cs' file inside Form1_Load_1 method as:-

```
AdminLogicLayer obj = new AdminLogicLayer();  
dataGridview1.DataSource = obj.GetAllAdmin();  
  
dataGridview2.DataSource = obj.GetAllAdmin();  
dataGridview2.DataSource = obj.GetAllData().Tables[0];  
dataGridview3.DataSource = obj.GetAllData().Tables[1];
```



⑧ You can also make this form UI look more better by using 'Combo Box' tool to show multiple tables by selecting from the dropdown.

My own Assignment Given By Vikas :- ADO.NET.

[many concepts in it]

The question is :-

→ You've been given a stored Procedure as :-

pGetRedisStreamServerSymbols 'wva616wmdfso11v', 'DEV'
↑
stored Procedure name
↑
These are its two parameter.

→ We know stored Procedures are like function in SQL which does some query related task.

→ So by using this stored Procedure, you've to show the tables in the DataGridview.

→ So when I run this stored procedure query in SQL Server in visual studio It returned 2 tables.

→ To run stored procedure query :- just write as below in SQLQuery.sql in VS.

storedProcedureName, 'parameter1', 'parameter2';

and execute the query you will get the tables returned.

→ So since, it returns 2 tables, we need to make use of DataAdapter and store in Dataset. Also we've to make Parameters structure.

(5) So my code for the given project is as :- (Continues server explorer techniques & model from previous's project only)

→ Code inside 'AdminLogicLayer.cs' file as :- inside its class as -

```
class AdminLogicLayer
{
    private string connectionString = ConfigurationManager.ConnectionStrings["db"].ConnectionString;

    public DataSet GetAllData()
    {
        DataSet ds = new DataSet();
        try
        {
            SqlConnection conn = new SqlConnection(connectionString);
            conn.Open();

            SqlDataAdapter sda = new SqlDataAdapter("pGetRedisStreamServerSymbols", conn);
            // We need to specify that the query is stored procedure
            sda.SelectCommand.CommandType = CommandType.StoredProcedure;

            // Set Parameter 1 and also set its properties
            SqlParameter param1 = new SqlParameter()
            {
                ParameterName = "@MachineName", // You can check its name in while
                SqlDbType = SqlDbType.VarChar,
                Value = "wva616wmdfso11v", // Passing the argument here
                Direction = ParameterDirection.Input // Since its argument to Pd not 'out' in C#
            };

            // Set Parameter 2 and also set its properties
            SqlParameter param2 = new SqlParameter()
            {
                ParameterName = "@Environment",
                SqlDbType = SqlDbType.VarChar,
                Value = "DEV",
                Direction = ParameterDirection.Input
            };

            // Put these both parameters in our Adapter
            sda.SelectCommand.Parameters.Add(param1);
            sda.SelectCommand.Parameters.Add(param2);

            // Fill or populate our dataset
            sda.Fill(ds);

            catch (Exception) { throw; }
        }
        return ds;
    }
}
```



After this we just need two DataGridViews drag and dropped in our Form1.cs [design].

→ And then code in 'Form1.cs' file inside

Form1_Load method as :-

You will get
this method
by double clicking
on Form1.cs.

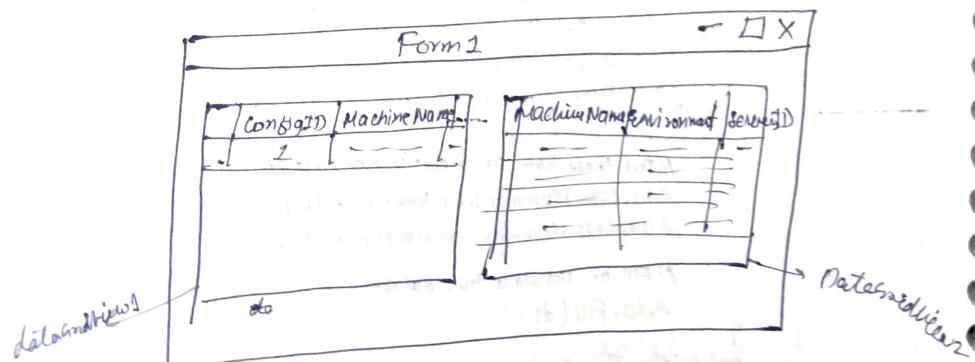
First write 'Using sys
using WindowsFormsApplication1.Model' to use
AdminLogicLayer
object here.

→ code as :-

```
private void Form1_Load(object sender, EventArgs e)
{
    AdminLogicLayer obj = new AdminLogicLayer();
    dataGridView1.DataSource = obj.GetBUDataC().Tables[0];
    dataGridView2.DataSource = obj.GetADDataC().Tables[1];
}
```

For first Table
For second Table

⇒ So now run the code it will run fine. It will give output table as :- in ~~button~~ Form1. (Windows Application)



student short + 1012
1010 = student 1010
1011 = 1011
1012 = 1012

1010 1010 1010

1010	1010	1010	1010
10	1010	10	10
10	1010	10	10
10	1010	10	10



Q# If in case we've stored procedure returning only single Table then we can make use in following way

If 1 Table only then we don't require DataAdapt and Dataset instead we could do by using SqlCommand and SqlDataReader for this single Table or called DataReader rather than Dataset.

All the files remains same as from 1st Project see in very beginning first project.

Files such as 'Admin.cs' in 'Model' folder , 'App.config'.

We've to do some code in AdminLogicLayer.cs file and 'Form1.cs' file.

Let our stored procedure name be = SPName

Name & its parameter be = ~~SPName1~~, ~~SPName2~~

And the table it return be for following stored procedure as

SPName 'Delhi', 20;			
Select * From tableName where city = 'Delhi' and age = '20';			
ID	Name	City	Age
1	A	Delhi	20
2	B	Delhi	20
3	C	Delhi	20

So code in 'AdminLogicLayer.cs' file as :-

```

class AdminLogicLayer
{
  private string connectionstring = ConfigurationManager.ConnectionStrings["db"].ConnectionString;
  public List<Admin> getAllAdmins()
  {
    List<Admin> li = new List<Admin>();
    try
    {
      //make connection
      SqlConnection conn = new SqlConnection(connectionstring);
      conn.Open();
      //Create SQL command
      SqlCommand cmd = new SqlCommand();
      cmd.CommandText = "SPName", //Stored Procedure Name
      cmd.Connection = conn,
      cmd.CommandType = CommandType.StoredProcedure;
      //Create Parameter 1 and set its properties
      SqlParameter param1 = new SqlParameter();
      param1.ParameterName = "@City",
      param1.SqlDbType = SqlDbType.VarChar,
      param1.Value = "Delhi",
      param1.Direction = ParameterDirection.Input;
      //Create Parameter 2 and set its properties
      SqlParameter param2 = new SqlParameter();
      param2.ParameterName = "@Age",
      param2.SqlDbType = SqlDbType.Int,
      param2.Value = 20,
      param2.Direction = ParameterDirection.Input;
      //Adding parameter to our command
      cmd.Parameters.Add(param1);
      cmd.Parameters.Add(param2);
      //SqlDataReader reads the record line by line
      SqlDataReader rdr = cmd.ExecuteReader();
      //Now lets retrieve data from rdr to our list
      //to show in DataGridView.
    }
  }
}

```



Mapping and storing data in the list

while(rdr.Read()) //Reads file will read DataReader line by line until null.

```
    Admin ad = new Admin();
    ad.id = Convert.ToInt32(rdr.GetValue(0));
    ad.name = rdr.GetValue(1).ToString();
    ad.city = rdr.GetValue(2).ToString();
    ad.age = Convert.ToInt32(rdr.GetValue(3));
    li.add(ad);
}
```

catch (Exception)

```
{  
    throw;
}
```

return li;

}

⑤ So now run the code the table will be returned in DataGridView.

code in Form1.cs file inside Form1_Load method as:

```
AdminLogicLayer obj = new AdminLogicLayer();
dataGridView1.DataSource = obj.getAllAdmin();
```

⑥ Also over 'Admin.cs' file class which works as Entity mapper. Code in it will be as :-

```
public class Admin
{
    public int id {get; set;}
    public string name {get; set;}
    public string city {get; set;}
    public int age {get; set;}
}
```

⑦ So this is all it.

Here we will see how we can use single DataReader only to return multiple query results from multiple tables.

we only need to make use of

GetResultSet()

reader move to next table if any present than



Scanned with OKEN Scanner

Q# Also we can make use of 'SqlDataReader' only in previous question to return multiple tables, we just need to make use of `rsdr.NextResult()` :-

- ⇒ The question remains as same as first one.
- ⇒ Here we made use of SqlDataAdapter and Dataset in case we've multiple table to be returned.
- ⇒ Here we will see how we can use `rsdr.NextResult()` only to return that stored procedure query to return multiple tables.
- ⇒ We only need to make use of `rsdr.NextResult()` to make our reader move to next table if any present then,

Q In a simply simple code we could do our datareader code for multiple tables reading as follows:-

Sample code:- sqldatareader with multiple tables.

```
String rslt = "";
using (sqldatareader dr = cmd.ExecuteReader())
{
    do
    {
        while (dr.Read()) // To check or read each row until there is row present
        {
            rslt += $"ReqID:{dr["REQ-NR"]}, Spnr: {dr["Spnr"]}";
        }
    } while (dr.NextResult()); // To check or read next table if there are other tables.
}
```

Q Since we've now two tables to be read using data reader and return them in our form.

→ So we need to make one more POJO class for another table.

→ To make 'Admin2.cs' similarly like 'Admin.cs' by first checking the column attributes present in second table and then name them as data members in 'Admin2.cs' class.

→ So code in 'Admin2.cs' will be as:-

```
using System;
using __
namespace WindowsFormsApplication2.Model
{
    class Admin2
    {
        public string machineName { get; set; }
        public string environment { get; set; }
        public string serverId { get; set; }
        public int sequenceNumber { get; set; }
    }
}
```



⑧ Now since we've to return two tables in our form.

So code in Form1.cs will be as in Form1_Load function as:-

```
private void Form1_Load(object sender, EventArgs e)
{
    AdminLogicLayer obj = new AdminLogicLayer();
    List<Admin> list1 = new List<Admin>();
    List<Admin2> list2 = new List<Admin2>();
    obj.getAllAdmin(out list1, out list2);
    dataGridView1.DataSource = list1;
    dataGridView2.DataSource = list2;
}
```

So now when you run the code both the tables can be seen now.

So this is how we can make use of datarader to read multiple tables