00:00:01 Hello viewers, how are you all? So from today we are going to start a brand new series of asp.net core six, in which we will discuss all the things related to asp.net core six, whatever is being used here in our market, so as you all know asp.net core is being used a lot to build asp.net web applications because we have this latest version and it also has long term support, so in today's lecture we will see the introduction of asp.net course 6 and then in the coming

00:00:34 In the lectures, there are some things in asp.net course 6 , we will discuss that and then later we will start its practical, so brother, before starting the video, a small request, if you have not yet subscribed to my channel, then I request you from my heart, please subscribe to this channel, so we will have the part number of speed.net core, so let us start our lecture now [Music] So first of all, we will discuss the introduction of speed.net course, okay, so what are the things in the introduction

00:01:07 I will tell you about it, go here, okay, so look here, first point nine days, everyone is talking about open source, cross platform development, I mean, nowadays, the way you know, everyone either talks about open source or talks about cross platform development, so brother, first of all, if I tell you what is open source, brother, look, open source are such programs which are created by someone else, but what has he done to make it open source, what has he implemented, and what is the purpose of open source, brother

00:01:39 The software that you are making, what can you do in it according to your needs, you can modify it and then what can you do, you can launch a new version of it and then look here, another term has been used for it, cross platform brother, look in cross platform first of all understand the meaning of platform, we call the operating system the way we have Windows, Linux, Mac, so what does cross platform development mean here brother, look, such software or such a program which can run on multiple operating systems, only that

00:02:10 We say cross platform, okay, so nowadays everyone talks about these two things. So look here, Microsoft has always been non for its Windows based product but not yet a new front of development. Look, whenever we talk about the Microsoft company, we know that Microsoft makes such products which are Windows based products. What does Windows based products mean, whenever they make any software, any application, then they run only on Windows, meaning they run only on any other platform.

00:02:39 Operating systems do not run on any other platform because you know whose product we have is Windows, it belongs to Microsoft, so whatever products Microsoft makes, whatever application software it makes, it keeps them Windows specific only, meaning those products run only on Windows platform, but brother, but today we have a concept of cross platform, it has become very popular in our market and Microsoft realized this thing and it made it a big success

00:03:10 I thought brother now what do we have to do, we have to move towards cross platform and open source, okay then look here the third point for this a new revolutionary product has come in the market which is called microsoft.net course, okay so what is it saying here that what did Microsoft do in this regard, a revolutionary product, I mean it launched such a product in the market which brought a revolution, okay and what was that, that revolution that we had was also a product which completely changed the future development and that

00:03:40 What was microsoft.net core? Ok, so when microsoft.net core came, what was there in it that it created a revolution in development? So go here, ok, now see what is it saying here, what is dot net core? Basically, dot net core is called a free, open source and cross platform framework. Now see, the same thing happened that when this dot net core came, the concept of open source also got implemented in it and the concept of cross platform framework got implemented, created for modern cloud based applications and every.net

00:04:14 Developers are proud of it, I mean that we are talking about open source and cross platform development, earlier Microsoft did not implement that thing, but when dot net core came, Microsoft made it open source as well as cross platform, meaning by using dot net core we can make such applications which are modern and which are cloud based, meaning which we can deploy on our cloud, so because of this the dot net developer started feeling very proud of it, after the launch of dot net core, then see

00:04:46 Second point nine there are no boundaries for the platform, see brother, it was said that as soon as dot net core came, now we do not have any boundaries for any platform, platform means operating system, so what used to happen earlier, whatever products Microsoft used to launch, they used to be Windows specific, but now there is no such scene, and every .net developer can say yes I am platform independent and I am using open source, means the developer can proudly say that brother I am platform independent and I am also an open source platform

00:05:20 Independent means that I am not dependent on any one platform or any one operating system. This is called platform independent. And I am also open source. Which means that whatever application I make, whatever product I make, it will be open source. This means that what will we have for the public, it will be freely open source. What will happen with this is that anyone can do with it, by making modifications in the product, in the software, he can launch a new version of it. Then look at the next point here, see what is in D saying.

00:05:49 Revolution of software development Microsoft launched its first dot net framework in the year 2000 with its first daughter framework 1.0, that means look here, he is talking about a very old thing, that means here he is saying that when the software development started, then Microsoft launched the dot net framework, when was it launched in the year 2000 and then which version of the partner framework came, 1.0 version came, okay, that means here I am not talking about dot net core framework, I am talking about the same old wale.net framework

00:06:20 I am asking when was it launched, 2000, listen what happened in 2000, it was launched and which was its first version, 1.0, okay, but after that, look here, what is it saying, this framework plays a major role in the field of software development, people love Microsoft technology products because this products are very easy to learn, that means, look here, it is saying that brother, when this dot net framework was launched, it created a stir in the software development and the people who are as knowledgeable about Microsoft technology,

00:06:48 There are products, I started liking it a lot, why did I like it because it was very easy to learn and use it, but brother, look, when the dot net framework was introduced, it was neither open source nor was it cross platform, so you remember this date, 2000, that means the dot net framework was introduced in the year 2000 , because now I will go ahead and tell you when the dot net core member was introduced, okay, go ahead and see here, in the year 2016, Microsoft has a new revolution , microsoft.net core 1.0, people always love asp.net because it's working all over the world

00:07:24 Wide web means look here what it is saying is that after at least 15 to 16 years we got microsoft.net core means after 15 to 16 years when this microsoft.net core came, the concepts of open source and cross platform were implemented in it and look what it is saying here, people always love aspi.net means the developers like speed.net from the very beginning, why do they like it because by using asp.net the working of the website that we make is very efficient and good as per our view then look at the next point now

00:07:58 Look, what is it saying here, asp.net core is a new open source cross platform framework to create a modern web based cloud based system, I mean, look what is it saying about dot net core, firstly it is open source and secondly it is a cross platform and then what can we do by using this, we can make modern applications and also create a cloud based system, do you know brother what advantage do we have with a cloud based system, can we upload it on the server of the internet, can publish it so that for systems

00:08:26 We should not lose our data inside which means you are not only working for Windows you can run in Linux Mac data is wherever you want it means you use sp.net core when we create a web application it does not mean that it should run on Windows it can also run on Linux it can also run on Mac so this is the biggest advantage of sp.net core that firstly it is open source and secondly we have cross platform then go ahead now look here brother look there are many students who are our

00:08:57 There are developers who don't know the meaning of open source, so I want to clear the concept of open source by telling you a few points here, because brother, look, first you have open source and then there is closed source, that means, what was there before Microsoft open source, it was closed source, that means, they did not provide us the coding of our software, due to which we could not modify it, but now that it has become open source, what can we do with its code, the code of the products

00:09:25 You can modify it and launch your own version, so look here, what is it saying, in general open source refers to any program whose source code is made available for it and modifications are made further by the users and other development facilities, meaning here it is saying that this open source refers to such programs and software whose source code is freely available to the users, so that they can do whatever they want with it, and if they want to modify it according to their retirement, then they can do that in it

00:09:58 Modifications can also be made and then look what it is saying here open source software is usually developed for public liberation and made freely available means look here it is saying that the open source softwares are made for public collaboration okay and are freely available so that anyone means any user publicly can modify it and can do whatever he wants, can set it up brother look the way we use Windows, Windows is an operating system and we also use Linux operating system so if I compare these two

00:10:27 Let me tell you the difference, brother, look, when Windows came to us, whenever we had to install Windows in our PC, what we would do was, we would buy it, but some of you viewers here must be thinking that brother, we could install Windows for free, but brother, the versions that you install for free are pirated versions, they are not complete versions, because if you want to install Windows with full features, then you will have to buy it, but it was not like this with Linux, you could pick up any version of Linux

00:10:57 So what do you have, it's free of cost. What you can do is you can freely download it from the internet and install it on your system. So this was the difference earlier between Windows and Linux. When you installed Linux, its code was sent to you and you could make your own operating system by making modifications to it. But this didn't happen in Windows because it was a closed source operating system and Linux is an open source operating system. So remember, it's open source.

00:11:25 What is such software, such applications which are freely available to you and in which you can make modifications, okay then go ahead now see what it is saying here 10 Software being distributed must be read distributed by any means without any restrictions, means what it is saying here is that brother the open source softwares, when they are distributed, they are distributed in such a way that the users can make modifications in it and read and distribute it again

00:11:51 Can, I mean for example, I got an open source software and I required some changes in it, so what I did was I made some changes in it and then I distributed it again for other people without any restrictions, so this is the concept of open source, then go ahead and now look here, one more point related to the source, the source code must be made available, so that the receiving party will be able to improve and modify it, I mean, look, in open source, this is what I am telling you again and again that brother, what happens in open source which

00:12:21 Your code of the product is available freely so that the receiving party who is using the product can improve it, modify it, then go ahead, now let us talk about the features of asp.net core. Alright, see, first of all we have discussed this earlier as well, the same point is being repeated again, it is cross platform, it is open source, Linux, Windows, Mac, wherever you want, alright, this point was discussed earlier, it is being repeated again, look here.

00:12:48 Second point fast development fast work over the browser means that whatever application you make on SP Important Core will be very fast on your browser, very efficient, so this is also one of its features then look work in your editor now you can work not only in Visual Studio you can also choose Visual Studio Code means here it is saying that ac.net core if you want to do it on Windows operating system then you can do it in Visual Studio but if you want to do it on Linux or Mac etc

00:13:20 But if you want to do that then you can use the traditional visual studio code that we have there and do your development inside asp.net core. So this is an important point that you have to remember. Then go ahead. Now look here, one more point is very important, it is based on model view controller MVC architecture. This means that by using asp.net core, the web applications that we will create will have the texture of MVC implemented in it. So, brother, if there are any SP.NET developers who are watching my video and if they are using asp.net

00:13:51 MBC has been implemented like MBC used to be five, MBC used to be four , so it will be very easy to reduce them with spotted core because in this also the same pattern is happening which used to happen earlier and what is that pattern, the MBC model view controller one, so this is also a very big feature of it. We will also implement MBC practically in the upcoming lectures. Then go ahead, now see here what is asp.net core. So what is asp.net core, now let's understand it. Look here, sp dot net.

00:14:21 Core is a new version of asp.net developed by Microsoft. Meaning, what is it saying here is that asp.net core is the new version of asp.net which was created by Microsoft. It means that you were using earlier version of asp.net like web forms, sp.10 mbc five, now all these things have become outdated. So now the latest version that we have is speed.net core. Fine, then look at the next point, it is an open source framework for developing web.

00:14:51 The application and it can be run on Windows, Mac and next, okay, the same point is being repeated, then go ahead, it has been completely written from scratch. Now look, this is a very important point brother, look, we had a dot net framework and now what we have is called dot net core framework, so remember this dot net core, whether it is attached with spot net or attached with dot net, it has been completely rewritten, it means that we have this dot net core, dot net core framework, it has been rewritten again

00:15:22 Why has it been rewritten so that the open source and cross platform functionality can be implemented in it, because as I told you, before dot net core, the dot net framework we had was Windows specific, which means that it neither had the concept of open source nor the concept of cross platform, that's why when asp.net core came to us, it has been rewritten from scratch, which means it has been rewritten in such a way that it also has things from the old dot net framework, but things have been kept in such a way that it looks like an open source framework.

00:15:53 It should be a source as well as cross platform then see, it is saying this here and it was initially lost sp.net five but let's rename it sp.net core 1.0. It means brother look when speed.net core was launched, it was named speed.net 5 but later it was renamed as speed.net core 1.0 so that there is no confusion and we get a completely separate framework, a framework that was rewritten from scratch, from the very beginning, okay then see the next point

00:16:24 Now look what it is saying here that asp.net core is a cross platform high performance open source framework for building modern cloud based internet connected applications. This is repeating the same point again. So look what it is saying here that there is also high performance and what can you do by using it in modern scenarios, you can create a web application and it can be cloud based and it is internet connected applications, i.e. cloud based means we can deploy it on the cloud and internet means our web

00:16:51 The applications are connected with the internet as well, then look at the next point it consists of modular components with minimal overhead now look this is a very important point brother look here it is saying that whatever you will develop using asp.net core, we have modular components in it, what does modularity mean, separation, you can also call it separation of concerns, it means every name will be separately divided, just like I told you that brother the concept of MBC is there in it

00:17:20 And whatever we do within this, we do it in the wave of modularity, which means there is separation, which means every code you have is separated, it is implemented in modularity so that we know in which file will this code be found, in which file will we find this code, so we call this thing model authority. So what are you saying here, with minimal overhead, which means by doing it to the minimum, with minimum effort, within minimum cost we can implement our modularity, so you retain flexibility.

00:17:49 Constructing your solutions and what happens with it what is the advantage of modelitis that we get a lot of flexibility a lot of ease when we are developing our software constructing it okay then go ahead now look here it is telling some more points see what it is saying here before asp dot net core means before asp dot net core Microsoft already has other similar frameworks that are asp.net web forms popular not just asp.net and asp.net mbc it means brother

00:18:17 Look what it is saying here, before speed.net core, on which two things did our asp.net developers work, either they worked on web forms or they worked on asp.net mvc. So these were the two things on which we, speed.net developers worked. Alright then look here what it is saying, the latest version of asp.net core is asp.net course 7.0. Now look this is a very important point that today, when you are watching our video, at this date, what is the latest version of sp.net core that we have, it is 7 , but right now we

00:18:49 We are also hearing mention of hp.net core that is also going to come and look this dot net course 7.0 that we also have when was it released what happened in February 2023 it was released means February 24 means what happened just two days before my birthday which version asp.net course 7.0 was released but let me tell you one important point we are not doing asp.net course 7.0 whatever we will do here we will have hp.net and 6 so see what it is saying here but in this tutorial series we are going to work with

00:19:25 asp.net course 6 because it is LTS, LTS means long term support, meaning brother look, I am not covering asp.net course 7 in this lecture series because it is not long term support right now. The asp.net course 6 that we have is long term support, till when will it be long term supporter? November 2024, which means right now we are in February 2023 , so the support that we have for asp.net core, which Microsoft supports and maintains, when will that end? At the end of next year, that means its support will end in November 2024, that's why we

00:20:03 What will we do in this, we will do dot net core in this lecture series, okay so you have to remember this point, okay let me tell you one thing brother, look, we have asp.net core which is a different thing and dot net core is a different thing, brother, look why do we use sp.net core for making web applications and asp.net core is this, it depends on which dot net core six and what is dot net core six, it is our framework, that means look here, we have this asp.net course which we will use for web applications, it depends, it bases

00:20:36 It is based on dot net core six and what is this dot net core six, basically we have a framework it means all the functionalities of asp.net core, all the codes, all the libraries, from where will we get them from dot net core six, so remember this on the right side we have a framework and this is a framework for building web applications, okay if I explain it a little more and tell you here, see when we had asp.net core 3 , on what was it based, dot net core

00:21:06 What was the framework three based on, similarly when the speed dotted core four came, it used to depend on what it was based on and this whatever we will do it, which is asp.net core 6 , on what it will be based .. on net six, that means this dot net core framework which we are calling six version, it is based on what, with this dot net core six it means the versions of asp.net core are also increasing and along with that the versions of dot net core framework are also increasing, so you have to remember this point

00:21:44 Well one more point I want to tell you here, very important, see what is being said here, from.net 5 3D world core is dropped from its name, it means when dot net five came to us, dot net five means when dot net core five came, then we dropped the word core in it, it means five and after that in all the dot net core frameworks that we have, the word core will not be there, so how will you pronounce it, you will do it like this, so all frameworks from nine will be named today dot netsex.net 7 and so on, meaning

00:22:17 yah.net six is also dot net core, dot net seven is also dot net core but the word core that was there in it will not be used now so if someone is talking about dot net five, dotted six or dotted 7 in front of you then you should understand that they are talking about dot net core, okay so this point is very important then go ahead now go to pre request, that means brother before coming to dot net core six, you have to learn some things and you should have knowledge of some things, what are they, look here, HTML is done brother for web developers

00:22:49 So it is natural to know it, okay, HTML is used, JavaScript is used along with HTML, which means you should definitely know it, and then along with it, Bootstrap, which is a framework, is optional, see, it is not necessary that you know it or not because apart from Bootstrap, we also have many other front end frameworks, okay, then we have C Sharp programming language, see, this is very important, you should definitely know programming because when you do a sp.tet course, what will be the education and education in you

00:23:20 All the coding will be implemented, whatever your object and inter programming concepts are, you should also know them. Then look, you should also have knowledge about MBC and MBC because as I told you, using sp.net core, whatever web application we will make, MBC and its texture will also be implemented in it. Okay, it's done. Then look, Visual Studio 2017 and higher version, means to implement asp.net core, you should either have Visual Studio 2017 or any higher version of it, but brother, here we are going to tell you that

00:23:50 In this tutorial series, which one will it be, Visual Studio 2022 Community version, why would we use community version, because firstly it is free of cost and inside it we will get all the features that will be there in our dot net core application, so if someone has not installed Business Studio 2022 , then download it and install it because inside this we will implement our practical in the upcoming lectures, so these were some of its pre requests, so if you liked this video of mine here, then subscribe to my video

00:24:18 Do subscribe to the channel and also share it and press the bell icon as there are more videos to come in it so if you want to get the notification of those videos first then press the bell icon in the comment box so if you have any confusion with the wall of this lecture then do ask me in the comment box so keep watching the program and now we will meet in the next video

00:00:01 Hello viewers, so as you know we have a lecture series going on on sp.net core 6 and today's lecture will be our chapter number you and in this we will talk about the difference between dotnet framework and dotnet core framework it is a very important concept okay, even in interviews it is asked a lot that what are the differences between dotnet framework which was traditional.net framework and dotnet core framework, there are a lot of differences okay we will discuss it completely in the lecture so brother if someone has

00:00:33 You did not see our part number van in which I told you the complete introduction of this speed.net course 6 so first you watch the full lecture after that come to this lecture so before starting the video a small request as you know this is my brand new channel it is a fresh channel so if you have not subscribed to it yet then please subscribe to it so this will be our part number 2 chapter number you so now we start the video [Music] difference between dot net and dot net core framework okay means both these frameworks and

00:01:06 There is dot net framework and dot net core framework is also a framework, so now let's talk about its difference. Well before coming to the difference, our traditional means our old dotted framework, using which we used to make all our applications till now, the Windows applications, Web Forms applications, MBC applications. So if I tell you a little about dot net framework, then go here, so look here, remember this is dot net framework, it is talking about the old traditional framework, okay so

00:01:35 Look here, the first point is dot net is a programming framework created by Microsoft. It means who created the dot net framework, it was created by Microsoft company and what is it, it is a framework. Now I will tell you what a framework is. Then see, developers can create applications more easily using dot net framework. What can developers do, they can create applications in an easier way. Okay, then look here, I am telling you about the framework. Now framework is just a bunch of code that's it.

00:02:05 Programmer can call it without having to write it explicitly brother see what does explicitly mean, manually means what are you saying here that what is a framework basically, it is a bunch of codes, it means that there are a lot of pre-made codes present in it, there are ready-made libraries present, who uses them, whose use is your program, means the programmer uses the code of the dot net framework library in his application, means the one who

00:02:34 A programmer does not need to write his own code manually because whatever that code is doing, whatever functions it is calling, all of them are built within the dot net framework, then look at the second point it is basically a collection of libraries, look brother I have talked about what a dot framework is, it is a bunch of code, it is a collection of libraries in which a lot of code is present, there are a lot of readymade classes, there are interfaces, there are a bunch of libraries with which what does a programmer do, if he is making an application then he has to compile the application

00:03:05 It works with dot net framework, so when you build your application without dot net framework, all the functionalities of the framework get included inside your application, just like if I am telling you that I am making an application using that or vv.net, then look, it is also called shishaf.net, we also call it vb.net, dot net is used for this reason because what does Shishap V do in combination with dot net framework, it develops the application, okay, so this point should be clear, then see next

00:03:35 Point is it has a number of pre-coded solutions that manage the execution of programs written specifically for D-framework. I mean, look what he is saying here is that there are a lot of pre-coded solutions within the daughter framework. That means, there is ready-made code inside it. That means, there are pre-coded solutions that manage the execution of your program only and only for the framework with which you are working. And which framework are we working with? We are working with dot net framework. Now, let us discuss the difference between these.

00:04:05 Let's do it, okay so go here, okay so we are seeing the difference between two things, one is dot net framework and secondly we have dot net core framework, so brother look, the first difference of dot net framework is that this is the old framework that we have, means now we do not have a new framework and neither will you get to see the new versions of dot net framework, means it has become traditional, it has become old and the new, modern framework of dot net of today's time is dot net core framework, means dot net core framework

00:04:33 It is our new framework and it is modern and the future. Then look at the next point note open source .net flamer is not our open source meaning whatever product you make, whatever application you make by using the daughter framework, its source code was not available to you for free. This means you cannot even make any modifications in it. As per your retirement, so this is a very big drawback because in today's time if any technology, any language is not open source for us then it is considered a drawback but if I use dot net core

00:05:02 If I talk about the framework, then it is open source, okay then the next point is only for Windows platform, it means that this daughter framework used to work only for the Windows platform, means that whatever application you develop using the daughter framework, it would run only on the Windows platform and inside the Windows platform also your dot net framework should be installed, but if I run in jaaun.net core framework, then it is cross platform, cross means cross means in any operating system, okay, cross platform means operating system and cross

00:05:34 Platform means on any operating system like Windows, Linux, Mac, you can run applications made on the dot net framework. Okay, look, these are the three major differences. There are many other differences too. You can discuss them here and go ahead. Okay, so now look here there are some more points. Okay, the first point is that develop in dot net framework is dependent of dot net framework installed on Windows. What does this mean brother? Look, whatever application we make, how do we run that application?

00:06:02 We make it by following any doc framework, which means there is a version of it, what we do by following that is we make an application and then when we run that application on our Windows, well then our dot net framework should be installed in the Windows naturally because the application which I had made by following the version of dot net framework, the same version or the version above that should be installed on our Windows because often you must have seen that whenever you are installing any application setup, then

00:06:33 Often a dot net framework error comes up there and what is that error is either your dot net framework is very old meaning the version is very low or you dot net framework is not installed inside Windows meaning dot net framework is not installed inside your Windows operating system that is why your app is not getting installed in your system okay so this used to happen inside dot net framework but it is not like this in dot net core framework because there is no such dependency inside dotted core framework meaning whatever

00:07:03 The dependency is implemented within the dot net core project itself, meaning that it is not required in dot net core applications that any dot net core framework should be installed on your system like this. So this is an important difference between the two. Then look at the next point. Now look at what it is saying here, dotted framework DOES N'T support dependency injection directly. Meaning, if you are making any application using the daughter framework, then there is Bill 10 dependency injection which

00:07:30 The support for dependency injection is not present there, which means if you want to implement dependency injection there, then you have to do it yourself, you have to code the dependency injection from scratch and whatever class or service related stuff is there in it, you have to do it all yourself, which means dependency injection inside the daughter framework, which is a very important concept in the modern era, implementing it used to be a headache with the .net framework but this is not the case with the dot net core framework.

00:07:59 The support for dependency injection is built in there. So when we go ahead and do it practically, we will see it by implementing it practically. So this is one more difference. We have two more differences. Now let's move ahead and talk about some more differences. Go here. Now look here. Now I have some factors. What do I do on the basis of those factors? I will explain to you the difference between the two. So there are factors. So I will tell you the difference between the Dot Net framework and the Dot Net Core framework.

00:08:26 I will tell you the difference from that. So the first factor is the application model. Okay, so we have our own application model. It means that by using that model, what kind of applications can we make with that framework. For example, if I talk about dot net framework, then look here, the dot net framework application model includes Windows Forms, which are desktop applications, like asp.net .

00:08:58 It became Web Forms, then it became MBC Applications, then it became Windows Presentation Foundation, which we use for desktop applications as well as for web, so we can make that as well. So, the application model of dot Net framework was for desktop applications and for web apps, but if I talk about dot Net Core framework, then look here in control desktop application development, it means that at present, we do not have the support for desktop application development in dot Net Core.

00:09:30 But it is possible that in future, support for desktop applications will also be added in the dot net core framework but as of now, this dot net core 6 does not have the support for desktop application development. Then see what it is saying here, focusing on instant web, Windows mobile and Windows store with asp.tet and Windows universal apps. It means that as of now, what things is the dot net core framework focusing on? It is focusing on the web and on the Windows mobile apps.

00:09:58 Then there is Windows Store, there is ASP.NET applications, and Windows Universal apps, which means those which can run on any platform. So, this is the thing that we are focusing on right now, the core framework, that is, dot net framework, which was the old framework, all the things that were in it, are not in the dot net core framework now, because look, this traditional.net framework is also called full flash framework, but we cannot call dot net core framework a full flash framework at today's time because it is still evolving, okay?

00:10:28 Version by version more things are being implemented within it. Okay then go to the next point. Okay now look what it is saying here, installation. Okay, it is a factor in installation. So look here, dot Net Framework is Single Package Installation and Run Time Environment for Windows. Meaning, look here what it is saying, how does the installation of dot Net Framework happen. That means dot Net Framework comes to you in a single package and what you have to do with that entire package is you have to add it to your application, bind it.

00:10:57 That is, brother, if you have an application based on dot net framework and if you do not want to use some libraries in it but still they are present inside your application, that means, I told you that there are a lot of libraries inside the daughter framework, there is a lot of bunch of code, so you can use some code in your application but some code will not be of any use to you but still it remains inside your app, why is it there because the daughter framework comes to you in a single package, a single complete package inside the application and then see this

00:11:27 It is saying that it provides you a run time environment for which only and only for Windows then look at the next point dot net core is not cross platform then look brother the thing is dot net framework was suitable only for Windows environment but what is it saying here dot net core is cross platform because it needs to be installed in dependent now what does independent mean brother look here if you want to install dot net framework on Linux, on Mac or on Windows then look at dot net framework

00:11:53 We will install it in a different way in Windows, we will install it in a different way in Linux, we will install it in a different way in Mac. What does it mean to install it in a different way? Brother, if you want to install dot net framework in Linux, then the criteria of the dot net framework in it will be different. Its criteria will be different in Mac, its criteria will be different for Windows, but you get the sem.net core framework on every operating system. Okay, so that's why it is being said here that it is installed independently and the operating system

00:12:21 On system then go ahead to the next point, okay now look next there is micro services support here, so look here what it is saying dot net framework has no support for micro services, this means you cannot implement micro services by using daughter framework because its support is not available but if I talk about dot net core framework then dot net core has support for micro services, this means if the concept of micro services has to be implemented inside the application inside your project, then for that

00:12:49 You will use dotted core framework, you will not use dot net framework, then go to the next difference now see what it is saying here rest services support. So see what it is saying here when it comes to d windows communication services d dot net framework, it is an fantastic choice it also works with RESTful services, meaning brother look at the concept of d, what do we call it Windows Communication Foundation, which was called d services, so its support is present inside .net framework, meaning which

00:13:19 It's an old framework but look here, dot net core has no support for those services, you would always need to create a REST API. This means that here within the framework you do not have the support for those services, what would you have to do for that, REST API, means that API would have to be created here. So this is another very important difference. Then go ahead and now look here, the next factor is performance and scalability, brother, look, this is a very important point, brother, look, we use a lot of technologies for this reason.

00:13:50 That is why we switch to it because its performance is good and we leave behind the one which does not have good performance. So look here, what is it saying, dot Net Framework is less scalable and provides low performance compared to dot Net Core, which means the capability of dot Net Core, the performance is low, but if I go here, look what it is saying, dot Net Core provides highest capability and performance compared to dot Net Framework because of its architecture, which means here it is saying that dot Net Core

00:14:22 The architecture of the framework is designed in such a way that the scalability and performance increases a lot, it becomes very high, so what does scalability mean? Here scalability means that the framework you have is fine, it becomes very difficult to change it and scale it, to extend it, to increase it, that means the scalability is very low in it, but if I talk about dot net core, then the scalability becomes very high in it.

00:14:52 Meaning we can extend it in a very good way going forward because its architecture is built in such a way that its scalability and performance is very high but dot net framework did not have such an architecture, that's why what is it being called here, less scalable and performance, okay then go ahead now look here, mobile development, now look this also was a very big revolutionary change in dot net core framework because look here what is it being said in dotted framework, d daughter framework currently not ten

00:15:22 Support Dear development, all ends here is a problem, that means the support for mobile development was not available at all in dot net framework, so if you want to do mobile development by using any framework, then what is the best one, dot net core framework, so see what it is saying here, dot net core has the same support for mobile apps, it's compatible with marine and other open source platforms for mobile applications, that means what it is saying here, the support for mobile development is available in the dot net framework, because in this

00:15:49 The concept of Zebrain is compatible with dot net core framework and other open source platforms which are used for mobile development, for mobile applications, their support is also available with dot net core framework. Okay then go ahead and see what it is saying here, CLI tools, that means CLI tools, look at dot net framework, what used to happen during the time of dot net framework was that there was not much importance of CLI there, that means whatever application we made using dot net framework,

00:16:19 We do a lot of ED, that means we do a lot of Visual Studio, we do a lot of Graphical Interface, but we don't do it as much as we want, so what is it saying here, dot Net Framework is too heavy, rather on ID, that means here it is saying that the bud that was inside the dot Net Framework was very heavy and you know that it is much easier to reduce some developments with bud, rather on ID, that means the way we are doing Visual Studio 2022 , inside that we are using ID, that means Graphical User Interface.

00:16:54 By doing that we can reduce the workload as much as possible but what is easy for developers nowadays is that they have more commands, they create command centers and controllers are created, models are created, migration is done, okay and there are a lot of other things, okay but if I talk about dot net core framework here, then see what it is saying, for all platforms, dot net core provides a very lightweight C, I told you, command line interface, there is always the option of switching to and id, I mean here it is saying that by doing that also they can reduce the workload

00:17:24 You can do this or else you can also do what on the ID, you can switch, but the code that has come here is very lightweight, after this there is a daughter framework in which there used to be a lot of heavy, then go to the next point, now see here, what is it saying packaging and shipping, now look, this is a very important point, understand it carefully, see what is it saying here, in dot net, all the dd libraries of dd dot net framework are packed and shipped together, it means what is it saying here dot net

00:17:53 All the libraries that were in the framework are packed together in one place and implemented in the application, which means that brother, for example, I am making such an application in which I will need some libraries, some will not be needed, so the libraries that I do not need are also implemented in our application, which means they are present there but they are not being used, so because of this, the dot net framework becomes a bit heavy, which means when I run the dot net framework,

00:18:22 By using the framework, for example, I create a web application and upload it to my server, then the libraries that I do not need, those libraries also get uploaded there, then look at the dotted core framework, now see what it is saying here, dot net core is a shift today on a collection of new get packages, means here it is saying that this is the complete opposite of dot net framework, because whatever application you make inside the dot net core framework, the libraries, the packages that are inside it

00:18:48 The new get packages that are inside different new get packages means they are not packed together at one place and shipped means they are not sent inside the application, so this is a very big advantage of dot net core, so these are some of its differences but brother look, now let us come to the point and what is that, look here dot net core vs dot net framework between this, I mean look at today's time, at this time where you are standing, at this time how will you do things that brother do you want to work with dot net core or dot net

00:19:19 We have to do with the framework, which of these two frameworks is better, so look here, I want to tell you one thing, look what is he saying, the answer to this question depends on the project requirement, meaning if you have to choose whether to use .net core or dot net framework, then the answer to this is that it depends on the type of application that you are making for your project, whether you will use dot net core or dot net framework, so what is he saying here and what are the project demands, so here are the same points that you have

00:19:49 Consider whether you choose best for a project out of .Net Core and .Net Core means here it is being said that whenever you have to do one of the two things, then first of all you have to see what is the retirement of your project because when you understand the retirement of the project, only then you will also understand whether you have to use .Net Core or .Net Framework in your application because in today's time applications are being made using .Net Core framework and applications are also being made using .Net Framework

00:20:17 is being created, okay so remember this point, so now I will tell you in which scenario you will use dot net core and in which scenario you will use dot net framework, okay so go here, see what it is saying here and chuj.net core, if it means you have to choose dot net core there, you have to prefer it there, look here, the first point d project demands cross platform integration, means if your project, your client says that brother, this is such a project which will be implemented on cross platform, means

00:20:46 It will run on different operating systems, so in that case what you have to do is you have to use dot net core. Then the second point is the project development of micro services, that means if there is a requirement to implement micro services in your application, then also you have to use dot net core because look, you will not find these two points in the daughter framework. Then look, the third point is the project release haveli on kali loti line interface s dot net core is suitable for ceiling.

00:21:18 Where you feel that our project that we are building, it relies a lot on CLI, which means that we will have to build our project by typing commands through CLI, so there also you have to use dot net core, you have to use it, okay, so this is also an important point because you know that dot net core is very suitable for CLI as compared to dot net framework, so this is it, where you have to use dot net core, okay then go ahead now look here

00:21:46 Prepare and chews.net framework if you mean where would you prefer dot net framework then look here the first point is applications are already running and daughter framework means such application which is already running on dot net framework so if you need to do any modification in it or if any project has come to you for maintenance then in that case also what do you need to do is you need to do it with dot net framework because brother look if you take the dot net framework application to dot net core framework then the entire architecture will change so when

00:22:17 If the architecture changes then there is a high chance of your application crashing, okay and we do not do this less, I mean the application which is built on dot net framework is sealed according to dot net framework, okay, second point d application requires technology like work flow web forms and d data not presented .net core, means if your application is redesigned and you have to implement work flow web forms or d services in it, then in that case you have to

00:22:47 Dotted Framework will have to be used because all these things are not yet present in dot net core. Okay then look at the last point, applications are built to run on Windows alone, meaning if the client gives you this requirement that brother, the project you are building should run only on Windows operating system because there are often many organizations where only Windows operating system is used, they do not use any other operating system, so they do not need the cross platform feature, so if such a

00:23:16 This is a scenario, so in this case also what you have to do is you have to use the dot net framework with your application. So whatever differences are there here, they have been completed. So if you have understood this lecture of mine, then I request you from my heart that please do subscribe to this new channel program of mine, subscribe and also press the bell icon because I am going to bring a lot more videos because this dot net core six is the latest one. If you learn it now, then it will be very beneficial for you in the future.

00:23:44 Because in future it is going to be less on this so if you have any confusion regarding this lecture then do ask me in the comment box. See you insha Allah in the next lecture till then keep watching my channel program [Music]

00:00:01 Hello viewers, as you know we have a lecture series on sp.net code 6 going on here, so today the part number of this lecture series is three in which we are going to talk about asp.net versus asp.net core, which means brother look at the traditional asp.net Which we have been doing in our earlier time in which we used to do web forms, MVC applications, using speed.net and the latest asp.net that we have today is called asp.net core, which is modern, which we have

00:00:33 The future is what is asp.net core so here we will discuss speed.net versus HP dot net core because this question is often asked in interviews I mean all the asp.net developers are asked that what is the difference between these two, so you definitely have to tell that if you want to crack your interview so before starting the video a small request as you know this is program end and this is my new channel so from the heart request please subscribe to it

00:01:01 So this will be our part number three so now we start our lecture [music] okay so here I want to discuss few points with you after that we will come to the differences of this all right so go ahead so look here its first point is asp.net watch the first version of web adapted dot net framework means look what is it saying here that when asp.net came and remember this traditional old one that we have, this speed.net is being talked about means Microsoft launched our first version of asp.net with dot net framework

00:01:42 It was introduced and we basically call it asp.net because look here with ASP, dot net is used because we use speed.net with dot net framework and when asp.net was introduced, then only Microsoft had a product, speed.net, using that we could do web development. After that look at speed.net core, so we mean asp.net was the first version and what is dot net core of today's time, it is an improved version, so look what is being said here, dot net core, it is an improved version with Richard.

00:02:12 Functionality, more comfortable interface, new libraries and other distinction means what is being said here is that the old asp.net has been improved a lot in asp.net core, a lot of functionality has been added to it, its interface is very comfortable and a lot of new libraries, whatever libraries are there according to today's modern and updated world, have been implemented in asp.net core, okay then see the next point asp.net core is the continuation of asp.net and improve it

00:02:44 That means look, you should understand that asp.net has not ended, but who is continuing asp.net, speed.net core, so what is speed.net core, it is the continuation of asp.net and asp.net core is much improved than asp.net, okay then see the next point, what is speed.net core, is asp.net available, net core is an open source tool because asp.net is an open source leader but what do we have for asp.net , it is open source. About open source, I had told you that any such tool whose source code is freely available to you.

00:03:20 It happens, you can make modifications in it and recreate it according to your requirement and then you can launch it, okay then go ahead now look here, the next point asp.net core, look at it, it is un cross platform, brother, this platform means operating system and cross platform means that it can run on multiple operating systems, so by using asp.net core, whatever web applications you make, you can make them on multiple operating systems, just like you can make them on Mac, you can make them on Linux, on Windows, it happens the same way like asp.net date

00:03:51 Works only on Windows means that the old traditional speed.net that we had, used to work only on Windows but this sp.net core is cross platform, means we can build applications on different operating systems like Mac, Linux, Windows, then look at the next point just like asp.net asp.net is based on 3d model view controller framework like most popular development frameworks, means you must remember when we studied traditional asp.net , there we had read BSB along with sp.net and sp.net in mvc5 is also a

00:04:24 There was a version using which we could build MBC applications with asp.net, so the same thing is available here inside sp.net core as well, that means inside sp.net core also we have MBC framework, using which we can build our web applications because brother, look, in today's time the pattern of MBC is very important and look, you do not get it only in sp.net asp.net core, but if I go to the PHP site, then there you must have heard the name of the framework like CodeIgniter or Laravel or Python's if

00:04:55 Go to framework, okay, there also you get to see the MBC framework, so MBC is the demand of today's modern era, that's why the concept of MBC in speed.net is also continuing, so look, this is a similarity, there is no fundamental difference, what is the similarity between speed.net and speed internet core, then go ahead and see now what is it saying, ac.net core has robust cloud support, now look this is a huge advantage for sp.net core that cloud support is present in it, that means

00:05:24 Whatever your applications are, you can deploy them on the cloud and it supports modular architecture. ASP.NET means see, when you make applications through ASP.NET, there too there was modularity, but it was less, but here in SP.NET core, there is a lot of modularity. See, I had told you earlier also about modularity, separation, separation of concerns, means for whatever I want to reduce, I will create a separate folder, I will reduce it inside a separate file, means what will I do by creating models in separate ones, I will do my development, so

00:06:00 This is what we call modular architecture, so the modular architecture of spotted core is very good, its speed.net , then go ahead now see, now here we will talk about asp.net , okay, first of all there are some points of speed.net, after that we will come to its differences, okay so look here, what is it saying, asp.net is a web development platform, which means it is a platform for web development which works with the dot net framework which provides a programming model and a comprehensive software infrastructure and

00:06:29 Various services required to build up robust web applications for PCs and now mobile devices. What does it mean? Look, what is ASP.NET ? It is a web development platform and it provides you a programming model, a method, using which you can build a web application. You can also build many comprehensive software, that is, software that runs on the web because its infrastructure is so amazing that by using it, what can you do? I just mean a web development with very good efficiency.

00:07:00 You can develop applications for PC as well as mobile devices, it is also good for this, then look at the next point, the first version of the platform blue and based on the daughter framework, meaning what is it saying here, when did the first version of asp.net come out? In 2002, at some places on the internet, you will also find the date 2000 , but here it is saying that the first version of asp.net that was launched was launched with the .net framework, meaning our asp.net is based on the dot net framework because as I told you, the dotted framework is very

00:07:31 It works in conjunction with all the languages, like sheeshab.net , fsab.net , vb.net , means you can say that there are above 50 multiple languages with which the dot net framework works, means that the libraries in the dot net framework, the code of that language, do that. Then look, the next point d platform is based on HTTP protocol for request and communication, means here it is being said that whatever asp.net does with us, it is based on the HTTP protocol, means brother, look, whenever

00:08:02 You request a website of sp.net and when that website communicates with the server and you see the result out, then the protocol that works in it is the HTTP protocol, which means that you requested a website and in return you got a response from the server, so to do all this in a secure manner we have the HTTP protocol and here it is being told that this sp.net platform is based on the HTTP protocol to perform requests and communication, okay then go ahead and look here

00:08:34 asp.net has three major development notes, means whenever we do asp.net , it is talking about traditional sp.net , means whenever we use asp.net, then what we do there is in three ways, that is, we create a wrap application, like the first one is web forms, in which there are forms, what we do inside it, by using web forms we create event driven application, means there we work through events, like when a button was clicked, then we had an event handler for the button, all the code was there, which got executed

00:09:02 Okay, so using Web Forms we would make web applications. Then the second was MBC SP dotted MVC. A glass version of this was MBC 5. So using that also we could make such web applications which were based on the model view controller architecture. Then thirdly we have web pages, that is, if we want to make dynamic web pages, if we want to make server side web pages, then what do we do for that, we would use HP.net. So these three we basically have development modes inside hp.net. Okay, then go ahead. Okay, so now we will talk about

00:09:33 Let's talk about asp.net core. Ok, there are some points regarding this, so look here, what is it saying, asp.net core is a new and standard version of asp.net. It means that this sp.net core is our new and standard version of asp.net at present, means that this is the standard now, now we have to use this in future and this is our future. Of course asp.net is also being used even now, but asp.net core is our future and this will be the standard in future because a lot of developers are slowly switching from sp.net to sp

00:10:04 On dot net core, here it is saying that dot net core becomes available open source and can execute not only on Windows but also on Linux and Mac because I see you know that in today's time the concept of cross platform is being implemented everywhere means developers and market want that we have to reduce at one place and implement it on cross platform so that's why speed.net core has been declared as the new and standard version because along with being cross platform it is also open source

00:10:32 OK then look at the next point with this update ASP.NET BKM is increasing more available to teams who usually work outside of the Microsoft ecosystem. This means here it is being said that when the cross platform and open source functionality was implemented in ASP.NET core , then those teams also started working with ASP.NET which earlier used to work outside the Microsoft environment , but as ASP.NET became the open source of cross platform, then those people who worked outside Microsoft, started working on it

00:11:05 Those who work outside the environment, they have also now started trying Spot Net. So this is a very important thing that was implemented when ASP.NET was updated. It was done. Then go ahead. Now look here, there are some more points, which I will show you here in the form of a diagram. Look, what is ASP.NET Core, it is a web framework which is open source and cross platform as well. Along with this, there is Run on.NET and Dot Net Core framework, I mean, brother, look, SP.NET , which was the traditional SP.NET, was only and only

00:11:36 It used to work on dot net framework but this asp.net core works on traditional.net framework as well as dot net core framework because as you know the applications that we used to make earlier used to run on daughter framework but now if you make a web application using this dotted core, you can run it on traditional.net framework as well as on latest.net core framework then see what it is saying here, modular, see the same thing has been mentioned again that its architecture is modular

00:12:06 It means separation of concerns, means for every work you have to create its directory separately, you have to create its file separately, which means you have to work separately, so when I explain its practical aspect then you will understand it better, means you have to keep the dependency in a separate file, okay, you have to keep the code in a separate file, you have to keep the user interface code in a separate file, you have to keep the back end code in a separate file, all these works and then look at cloud optimize, the same thing that cloud support is also available to asp.net core web applications, okay then further a

00:12:35 Let's come back now and let's discuss the differences between it. These were some of the things that I wanted to tell you before bringing up the differences. So look here, I will tell you some factors and according to those factors, I will tell you the difference between these two things. So the first factor is that we have platform support. So look here, SP works with Windows. But if I talk about this SP.net core, then it works with Windows as well as Linux and I can also work on it.

00:13:01 That S means operating system then look at the second factor in format so if I talk about asp.net then the performance of even a sp.net application used to be very high performing because sp.net is a very high performing framework but if I talk about dot net core and asp.net gives better performance it means the performance of the application that you will make through sp.net core will be much better means it will be even better than sp.net okay then go ahead now see here what is it saying

00:13:33 And the factor is that we have architecture, okay see what the architecture is saying, based on dot net framework only, means the architecture that we have for asp.net is based only and only on dot net framework but if I talk about the architecture of sp.net core then it is based on dot net as well as dot net core framework, so this is also a big advantage of using speed.net core then look next point is components, okay see which components do we use inside asp.net

00:14:02 We can do Web Forms, MBC, Web API, which means that using the traditional SP.net that we had, we can create Web Forms applications, create websites, create MBC web applications, and even create Web APIs, but if I go to sp.net core, look here, even in asp.net core we can create MBC applications, create Web APIs, and create web pages, but cannot create Web Forms, which means that even today we do not have the support for Web Forms in dot net core.

00:14:30 Okay, so there should be this difference, and look, this question about Web Forms is often asked in interviews, whether support for Web Forms is available in Netcore, so what should be your answer? No, okay, then go ahead, look here, the next factor is dependency, look here, what is meant by dependency, brother, look, it often happens in our application that we have to add a lot of libraries, okay, dependency, we have to add this, some dependencies are available by default, a dot net

00:14:58 Inside the core application, this is inside the speed.net application, but we also have to add some things ourselves, so that is what I am talking about here, see what he is saying, more dependency less monitoring options, means that you used to have a lot of dependencies inside speed and net and there was no monitoring of them, means at least the options for monitoring them, the dependency, what am I talking about, I am talking about the libraries that are inside your application, so whichever dependency you add, the options for monitoring it, its tracking are very less

00:15:28 But if I talk about HP dot net core then look here what is it saying strict control over the number of dependencies means that you have very strict control over the dependencies that you have in your speed.net core application control means you can monitor it and control everything according to your requirement then look next point supported files okay if I talk about supported files then look here inside a speed.net the CFB is supported and the FWP V webconfig and ideal but if I talk about a dot net core

00:16:00 Let me talk about it, look what it is saying here, no support for global dot assets and web config files, means look, inside dot net core, you will not find the global.s file, you will not find the web config file but here you will find a file named app settings.json and what we do there is that whatever settings there are of our sp.net core application, whatever configuration there is, where do we set all of them, inside the app settings.gs file, okay so this is also a difference you should know, then further a

00:16:29 Now go and look here, one more factor is being mentioned, look here isolation models and containers brother, look isolation means separation, separating and models also brother means containers also, you have the same meaning, okay so look what is it saying here, low level of isolation in A speed and net not highly suitable for micro services and container development, means look here it is saying that brother in A speed dot net the isolation was less, the separation was less, the modularity was less, its level was very low, that's why asp.net

00:16:59 It is not considered suitable for microservices and container development where we need separation of concerns but if I talk about dot net core then look what it is saying Improved modular support Integration with Docker Powerful isolation algorithms means if I talk about dot net core then its modular architecture is very much improved and very strong that is why it can be integrated with Docker and its isolation algorithms are also very powerful then see next

00:17:29 Point Visual Studio Support Visual Studio means we have an ID using which we do our development of speed.net so look what it is saying here supports all versions means all the versions that we have support asp.net means whether it is old or new you can use speed.net everywhere but if I talk about sp.net core then what is it saying about sp.net core, look here only the latest versions are supported starting from 2015 means 2015 and the versions after that like 2017

00:18:03 Be it 2019 or 2022 , you can use HP dot net core in it, but the older versions than 2015 , like you have 2013, 2012 , 2010, do not have the support for sp.net core in them and I had told you that we will do our practical implementation of this per.net course 6 on Visual Studio 2022 because it is the latest in the market right now, so here all the differences that we have have been completed, so brother, I request you from my heart that if you watch this video of mine

00:18:34 If you liked it then please subscribe to my new channel program and share it as much as possible. So if you have any confusion regarding this video then please ask me in the comment box. See you in the next lecture, till then keep watching my channel program [Music]

00:00:01 Hello viewers how are you all so as you know we are having a lecture series of sp.net course 6 so in today's lecture we are going to talk about a very important concept of asp.net course which we call MC so brother this is going to be a very important lecture whenever you make your web application using asp.net course then the concept of MBC is also implemented in it so before starting the video a small request as you know this is my new channel program

00:00:30 And talk so if you haven't subscribed yet then please subscribe and also press welcome because there are many more videos coming in sp.net course six so this will be our part number four and now let's start the video [Music] Look at you MBC, okay it's a very important concept whether you talk about isp.net course or any other framework like HP has Laravel, Coding Nitter and all the other modern frameworks we have using which you can make a web application inside it

00:01:06 The concept of MBC is very important and it is implemented naturally. So what is the concept of MBC? So look here, first of all if I tell you the three basic components of MC, I mean what do we have in MC, what are the three basic components, which ones come first, model means here in MBC, what is the M, what does M mean, model and what does the model represent, it represents business entities, business entities, I mean brother, for example, I created a model and with that

00:01:36 I have connected my database so now you know that we create a lot of entities inside the database like if I am making an application for a school, so what business entities would be there in it, there would be employees, there would be teachers, there would be students, there would be admin, so to represent that, what do we do, we create a model, and what is a model basically, you know there are classes, so this is the basic component first, then after that there is our view, whose view am I creating, which

00:02:05 We have HTML pages, right? What is it basically called? View, inside which what do we do? Our presentation logic, I mean whatever work is there related to the user interface, where do we do it? We do it inside our view, because look, view means this view is displayed to the user, he has seen this view. So, whatever work we have related to UI, user interface, we do it in our view. So, what is it called? We have to write our presentation logic inside it. Then, look, next is the controller of the controller.

00:02:32 What we do inside is we have to write our business logic, that means whatever logic is there in our application, where do we write it, it has to be written inside the controller, that means if the controller has to do communication between the model and the view, then who will do that, the controller will do it, that means whatever important logic you have, like how to get the data from the model, how to pass it to the view, what do we do all this, we do it through the controller, that means you can also create the controller in such a way that it is a bridge between the model and the view and

00:03:01 Brother, look, inside MBC, what is called the backbone of MBC? It is called the controller because if I remove just the controller from within MBC, then my entire MBC will become useless because the concept of controller is that it creates communication between the model and the view and whenever you request your application, whenever you request the URL on the browser, then first of all the request goes to the controller. Okay, so this

00:03:31 You also have to remember the point, it means look here, you must be understanding one thing that here we have separation of concern, means whatever work is related to database, we do it through model and after that whatever work is related to our UI, user interface, we do it inside the view and whatever logic is ours, we do it inside the controller. So what is happening here, separation of concern, means we are implementing every work inside a separate model. So then further, now look here, the MBC stance.

00:04:00 But Model View Control, okay, we know that there is Model at M stance, View at C stance and Controller at C stance. So then the next point it is an R-key textural design pattern, today in fact this design pattern is used at the architecture level of an application. It means brother, look here what is MC being called, it is being called R-key textural design pattern. It means whenever we will build our application, whenever we will create the architecture of the application, whenever we will build up its structure, then by using the concept of MC we will reduce our head.

00:04:31 Okay then look here next point so d point date you are need to remember means there is one point which you have to remember very well and what is that this MBC this note n programming language means look many students consider MBC as programming language it is not a programming language it is only and only a design pattern and then look what is MBC doing this note n framework means some students some developers also consider MBC as framework remember MBC is not framework asp.net we have framework and by doing asp.net

00:05:02 Inside we have a technique, a design pattern which is called MBC so look here, it means that neither BC is a programming language nor is it a framework, what is it simply, we have a design pattern, design pattern means that by using this design pattern we build up our application, then further, now look here, when we design and application, first week create the architecture of the application and MBC plays an important role in the architecture of data, particularly

00:05:32 Application means look what is being done here, whenever we design an application then first of all we create its architecture and how will we create the architecture of our application by doing MC, that is, for example if I create an empty web application using SP.net then inside it I will have to create folders of models and controllers and inside the model I will create database entries, inside the view I will create my user interface and inside the controller I will implement my business logic.

00:06:02 I will do this and the work of how to do the interaction between the model and the view, I will do that work also inside the controller, so that's why MBC is very important for us. So what I told you is that by using MBC, we implement separation of concern in our application, means we do every work separately inside the model. So if I explain this thing a little more, look here, the advantage of SOC means separation of concern, so look here, the first point is on individual piece of the system in isolation.

00:06:31 Isolation means isolation means separate isolation means separation means whatever work you are doing, whatever application you are making, you divide it into different pieces, like I told you why do we do model work, for interaction with the database, then why do we do U work, for user interface and where do we do logic work inside the controller and look this is a very good thing for MBC to do it because you must have often heard in software houses that brother database developer

00:07:02 They are separate, okay we have separate front end and developer, there are drivers on backend and development, so for all of them we have a separate model in which they work separately, then see the next point is facilitate reusability, it means what can you do by using MBC, you can also implement reusability, what does reusability mean that brother you have declared a thing and then what can you do with this thing, you can use it again and again in your application, you can use it again and again, so this reusability

00:07:32 Where will it be implemented inside MBC, we will also see that, then which also shows the maintainability of a system, brother, like I told you sir, it is happening inside separate models, inside separate components, so brother, if I know that some issue is happening in my front end, then where will I go and maintain it, inside the view, that means inside you, I will manage and maintain that thing, similarly, if I know that some model classes are related to database, then I will go and implement them, I will maintain them, okay, so that's the point

00:08:00 Here I am telling you the next point and views is extensibility. Extensibility means that brother, for example, I have prepared my complete application, but in future if I need to extend some things within that application, I mean, I need to increase my application, then I can do that also very easily because I know that if I need to insert something in the view, I mean, if I need to insert something in my user interface, then I will work with the view, if I need business entities, I mean, the model

00:08:32 If I want to add some more NTT inside it then I will do it with the model so it means we have its extensibility that too is very easy and efficient then look next point enables user to better understand means when you do MBC and when MBC provides you inspiration of concern then through this the user of your system has to understand your system and your application in a very good way so this is the advantage of MBC's inspiration of concern okay so brother look now I will tell you

00:09:03 I understand the flow of its MBC, so to understand the flow of MBC, I have a diagram here, okay, you have to understand this very well, so look here, what do we have here, there is a model, a view, a controller and our user, user means who am I talking about, the one who visits our application, who views our web application, so brother, see, what any user does for the first time is that he enters the URL of our application in the browser, that means our application

00:09:30 The user sends the request. Request means he is trying to access our application. So whenever he requests from the URL, where does the request go first? Remember, it goes to your controller. That means, whatever request the user makes, who will receive that request first? The controller. And then what does the controller do? The controller checks whether it has any data present inside the model. If any data is present inside the model, it means if the model is taking data from the database, because as I told you, inside the model, we have such classes which are used by our

00:10:00 Database entities represent the entities of the database. I mean, as I mentioned, employees, students, teachers, so if there is any data in the model, then that data gets updated and rendered inside our view. And who is responsible for taking the data from the model and rendering it in the view? It is the controller's job. So, when the controller takes the data from the model and updates it and renders it in the view. And then the same view is visible to our user, because, brother.

00:10:30 See, the user can neither see the controller nor the model. The user will only see the view. So, whatever data of the model is there, where do we get it inside? We get it rendered in the view so that the user can see it. And who is responsible for displaying the data of the model in the view? It is the controller's. Okay, so this is one situation. Now, in the second situation, whenever the user accesses the URL of your application, I mean, whenever he sends a request to your application, then that request goes to the controller. But if we have a model

00:11:03 It does not exist means if the model is not present with us because often it happens that we do not have a model in our application, which means mostly we are doing static work there, so if we do not have a model, then what does the controller do, it directly displays its view to the user, okay, I mean I am understanding that whenever the user sends a request to the controller and you have seen that the controller says that the model does not have any data, so look here, what is it doing, if there is no data, then what does the controller do directly, it displays our view.

00:11:30 It gives this to our user. Okay, so you have to keep this flow in your mind very well because if you keep this in mind, then it will be very easy for you to create a dot net based ambitious application. Okay, so look, I told you this in the chapter about MBC, but if I compare this example with a real life example, then when you understand that real life example very well, then you will never forget the concept of MBC. And what is that real life example, look here, we have the example of a restaurant.

00:11:59 Brother, look what happens in a restaurant is, some customer or the other comes to us, so this one that you are seeing, what is this, basically it is a customer, and look here, what is this that you are seeing, it is a waiter, and then what is this that you are seeing on the right side, it is our chef, so brother, look, this customer, what is this, basically you have a view and this waiter, what am I calling this waiter, controller, and this chef that we have, what am I calling him, model, so brother, look what happens inside the restaurant, whatever customer is there,

00:12:27 Who does he get his order written by, he gets the order written by the chef and then the waiter takes that order to the chef and what does the chef do, he prepares the order, okay and when the chef has prepared the order, then the chef gives that order to whom, he gives it to the waiter and then the waiter goes and gives that order to whom, he gives it to our customer, he gives it to the client, it means look here who is in between the customer and the chef, we have a waiter because brother, look, our customer gives the order to the waiter, right?

00:13:00 It doesn't happen that the customer will directly go to the kitchen and tell the chef, yes brother please make this thing, this doesn't happen in any restaurant, I mean you have a customer and the chef, who is the one who facilitates the communication between them, the waiter, similarly we have a view, the view and the model cannot directly communicate with each other, who facilitates the communication between these two, the controller does it, it means if the view needs any data from the model, I mean if it needs any data from the database, then who makes the request to the controller

00:13:31 What does a controller do? It gets data from the model, i.e. database, and then gives it to the view. So think for yourself, if I remove this waiter, this controller, from here, then will this example be correct? No, the example will not be correct at all. So this is a real life example of MBC. If you understand from this, then you will never forget the concept of MBC. So, what is its conclusion, who gets the communication done between the view and the model, the controller, because if the view is

00:14:02 If data is needed from the model then the view will send the request to whom, it will send it to the controller and the controller will take the data from the model and display it where in the view, but if the model needs data from the view then the model will also send the request to whom, it will send it to the controller and the controller will take the data from the view and display it where in the model, just like we create a form in the view, and whenever we submit the form, then the request will go to whom, it will go to the controller and that data of the form will get stored in our database through the model

00:14:32 It will be stored. Okay, so this is a real life example. Well then if I elaborate this example a little more, okay, look here, there is another diagram, I mean, look here, client means client machine, you can use it as a browser, which our user uses, so understand that this is a browser, a client machine, so from here, the user is sending a request, then that request will go to the controller and the controller will send the response back to our client, to our browser, to our

00:14:59 It is okay for the user but look here, when the client sends a request to the controller, then look here the controller is getting the data from the model and look here from where is the model getting the data, it is getting the data from the database, okay and then what does the model do, it displays that data in the view and then where does the view controller send it in the response, it sends it in the response to the client, okay, that means look here what is it doing, whatever client sends the request, what will the controller do, it will call the view and from where will the data for the view come from the model and

00:15:27 From where is the data of the model coming? From the database. Okay, so when the data from the database is brought into our model, then where will that data go from the model, it will go to the view and then that view controller will get it displayed to our client, it will get it displayed to the user, so this is also a very good flow of MBC, it is also very important to understand this, so this was our concept of MBC, so brother, see here I have explained it to you with diagram and theory in the next lecture when we will create our first project because till now

00:15:56 We did not create the first project by doing speed.ed course 6 , so when we create our first project, what will we do after that, this is the concept of MBC, using this, what will we do, we will build our web application, so you have to understand this and the shop very well, so here we have it, lecture is complete, so if you liked my video, then definitely subscribe to my channel, share it and press the bell icon because many videos are going to come in future, so if you like this lecture

00:16:25 If you have any confusion related to this then do ask me in the comment box, Insha Allah we will meet in the next lecture, until then keep watching my channel program [Music]

00:00:02 [ Music] Hello viewers, how are you? As you know we are having a lecture series of sp.net course, so in today's lecture, we will create our very first project using asp.net course six brother, see, in the previous lectures from part one to part 4, whatever were its theoretical concepts, in that way MBC happened, difference between us has HP dot net, sp.net kar, then similarly difference between us has dot net kar or dot net, then introduction, all these things, we have seen in the previous

00:00:36 You have done it in the lecturers, so from today we are basically going to start its practical. Okay, that means we will create the first project here today. So brother, look, in today's lecture I will teach you to create two types of projects. One is an empty project in which the folders of MBC are not present. That means, if you create an empty template, or an empty project of speed.net course 6 , you will not find any folders of MBC inside it and you will not find any files inside it. So

00:01:03 That means we will have to do all the work from scratch, okay, that means we will have to add the model view controllers folder ourselves and we will have to do the code ourselves in it, okay, after that I will tell you about another project which we will create with the concept of MBC, that means if I create such an asp.net course 6 application in which we will have all the folders of MBC, inside which we have controllers, action methods, views, models are also created inside it.

00:01:32 Okay, so these are two types of projects, in today's lecture, I will tell you how we create them and I told you that the ID I am using will be Visual Studio 2022, so brother, if someone has not installed Visual Studio 2022 yet, then I would comment that you simply write Visual Studio 2022 on Google and go there and download the community version and install it, so after installing it, you have to continue from here, okay, so I go to my Visual Studio 2022

00:02:01 So look here, I have opened my Visual Studio 2022 , so look here, here you will see its Friends Project on the left side, what we have to do is look here, you have to click on Create a New Project, okay, so as soon as you click on Create a New Project, then look here you will get some files, okay, that means look here it is telling you which project you have to create, you see, you must be seeing a lot here, but I will first show you by creating a project, inside which MBC folders are also present

00:02:26 Okay, so you can search here, but for example, if I write asp.net here , okay, then look here, you will be seeing asp.net wap model view controller, if you create a project by using this template thing, then what will happen is, inside this, all the folders of MBC, all the files, etc., you will find them all here, so first of all, I will show you this, after that, I will create an empty project and show you, okay, so you have to select this and look here, you will see the next button.

00:02:55 You have to click on this and then look here, here you must be seeing the name of the project, okay, you can name the project as per your choice, I will name it, but for example Asper Web Up MBC, okay, if I put C then it will know that brother, we will also have the folders of MBC inside it, okay, so look, I will click on Next and before clicking on Next, if you look here, you want to change the location, look, what is the location I have here, C drive, Users is inside it, then here is my folder

00:03:21 Inside that there will be a source folder and inside the repos all your projects will be created here and look here if you want to change the solution name then you can change that as well, I will name it Aaj ITIS, okay then click on next, here A which is okay so look now you have to change the framework here, okay so this is a drop down, you click on it, look here it is telling you that the version of dot net code 3.1 is out of support, then dot net 5.0 is also out of support

00:03:47 It has been done but what are we talking about here, we are talking about the future dot net course 6, so brother see, our HP dot net course 6 is based on what, it is based on dot net 6 and look here, what do you see, the long term support, but brother, let me tell you one thing, this dot net course framework six is long term support but till when will it have long term support, so I will also tell you that its long term support is available till 2024, that means till November 2024 , because see, our

00:04:17 The dot net course seven has also been released but at this time it is not under long term support so we are not using the dot net course seven. Then look here, you have to select the authentication type. Here you can select individual account or you can select any other account. I am keeping this here for now. Then look here, configure HTTPS, that means if you are using HTTPS for your application which is done through certificate, then if you can configure it with HTTPS

00:04:44 If you want to run it then you can put a check on it or else if you don't want to run it over HTTPS then you can uncheck it as well. Well let me tell you one thing, some developers have systems which are PCs, in that it supports HTTPS but we also have some such systems in which HTTPS is not supported. So if you put a check here and create an application and run it, then if your application is not running then what should you do? Then you create a second project and run it with that.

00:05:13 Inside this config, you should remove the check mark on HTTPS. OK. When you remove it, your application will run but it is suitable that you keep a check mark on it. OK. And if you want to enable Docker on this application, then you can put a check mark on it as well, but I want to keep it disabled for now. OK. Then if you click on Create here, look here, our application has been created. Look on the right side, you must be seeing in the solution explorer, look at ASP and WebMBC and

00:05:41 Look here, you can see a completely direct structure and notice one thing, inside this we have created this project with MBC, so look here you can find the models folder as well, inside it there are some files by default, then look there is a controllers folder as well, inside that a default home controller is created, then here below there are views, look inside this we have created views related to the home controller, okay and then there is a shared folder, inside which there are such views which are present in your entire application.

00:06:05 That means what are across simple pages, they are shares and here look at the program dot cs settings here, here you must not be seeing the startup file startup dot and secondly global.sx which was there in the previous versions of our asp.net but that is not present here. So this is a little difference inside sp.net course 6 so this is its complete directory structure, brother what are the things, what is the use of root and what is the use of these files, we will discuss all these things later, okay, today's

00:06:40 What I told you in the lecture was how can you create your own asp.net application with MC. Now I will tell you that if I want to create an empty application, meaning in which I neither need a folder of controllers, models, nor views, meaning if I am doing everything from scratch, then how will I create an application for that. So here, the file which you must be seeing, you have to click on the file, click on new, and look here, project, you have to click on it, okay then look here.

00:07:07 You are seeing the templates, you can search here again for asp.net, but now here I have to make an emty application, so look here, look at this, you must be seeing asp.net on MBC , okay, so now if I select it, then look here brother, whenever you select any application here, whether it is emty or MBC based, then it is telling you below which languages will be there, like it is telling that brother this is education, your language will be there in it and then look, the lyrics are done Mac

00:07:34 This is Windows because as I told you we also have cross platform cloud, and then look you will see cloud here, okay you can deploy it on Azure, then you can create services, there is web and many other things, okay so I will select ac.net and click on next, okay so look here you can see the web application, I will name it here Asper Web App, okay look inside this I did not put MBC because it is empty

00:08:01 Let me write it in MTV, okay so that it knows that this is an empty project. Then I come here and click on Next. The rest of the things, I have to leave it in front of me, I go to Next and then see, it is asking which framework are you going to use, so I am leaving it here, dot net sex, and look here, what am I doing for HTTPS, I am checking it here and enable Docker is disabled, okay and now what I do here, I click on Create, so now look here, you will see

00:08:25 What is happening here is that the folders of MBC that we had, Model View Controller, that folder is not present here, meaning that it has provided us the basic project, the basic functionalities here, rest of the things we will have to do from scratch ourselves, meaning we will have to add controllers, model views here ourselves and we will have to do all the work from scratch here, so it means whatever work we will do in the coming lectures, we will do that work from scratch, okay, meaning whatever things we will require here, according to this

00:08:53 We will go on creating the folder structure directly and do our work inside it. Okay, but look, I will close it from here for now. Now, what should I tell you, look here, I had created another project in which the MBC folders were present, so let me open that. Look here, let's go to file and look here, let's go to recent projects on solutions. And look here, we have this speed.net, you must be seeing it from B, I will open it by clicking here, and inside it

00:09:18 This project which we had, if I show it to you after running it, I mean the project which we had made with MBC, how does it look after running, then look here, how to run the application, look here, you can see the play button in green and the name of your application is written along with it, so brother, look here, if I click on its drop down, then look here, if you want to run your application on any specific browser, then look here, you can see the web

00:09:44 Which browser do I have selected right now? Google Chrome. But it depends on you, brother, on which browser you want to run it. So, you will select that browser from here and then when you run it, your application will run on this browser. Apart from that, look here, if you want to disable debugging here, there is an option for that as well. If you want to run it, then you can do that as well. What I am just doing here right now is that I have selected Google Chrome here and now I am going to open this

00:10:10 Let me run the application and show it to you. Look here, you can see that there is a Never by default, inside it, the name of our application is Home and Privacy, here you can see the link buttons and look here, you can see a text Welcome, okay, and look here, what is visible, Learn about building weather apps with sp.net. So, if I tell you that this page that is getting displayed with us, basically, from where is this page getting displayed, okay, so look here, I close it, okay, I stop my application.

00:10:37 And look here, you must be seeing the controllers folder. Inside this, you must be seeing the file homecontroller.com. Open it. Inside this, you must be seeing the home controller. And look, what is home controller? It is basically a class. Here, if you go a little down, you will see an action method here, what is its name? Index. Okay, what are action methods? What are controllers? We will discuss them in detail later. So, now look here. Whenever I run my application, which controller do we have?

00:11:03 This controller has been run and inside this controller there is a method whose name is index, you know what is index, right? It means it is like this, just like you have seen it inside MBC, so look here, when you run this action method, it will return a view and where is that view created, look here, look here, you can see the folder of views, expand the apps, inside it you will see a folder of home and look here, index.cssml, you have to click it, so look at the one which you were seeing on the browser.

00:11:32 The display was basically coming from here, you can see Welcome was written in big letters, look here, you can see Welcome and above it, there is a class of display four and then look here, you can see a link, so look here, this is a link, which means it is an anchor and look here, what was written inside it, Building Web with AS SP.NET, so it was basically coming from here, okay, so now look here, you must have understood how we run your application and which file did we see for the first time, okay.

00:11:58 So you see, the rest of the things which is the complete directory structure and its complete flow is that when this application is run, which file runs first, then what happens, I will tell you all these things in the next class, I will tell you in the next lecture in detail, okay so here we start the lecture, so if you liked my video, then definitely subscribe to my channel, share it and also press the bell icon at the end because many more lectures are yet to come, so in this

00:12:22 If you have any confusion regarding the wallet of the lecture then do ask me in the comment box. See you insha Allah in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are having this lecture series of asp.net course 6, so in today's lecture we are going to talk about directory structure and flow of asp.net course six application. Brother, see, if you have seen our previous lecture, then in that I taught you how to create your first project, so if anyone has not seen our previous lecture, then watch that lecture, after that come to this lecture, okay, so to start the video

00:00:27 Before starting, a small request, if you have not yet subscribed to my channel Program Matter, then I request you from my heart, please pause the video right here and first subscribe and also press the bell icon because many more videos are going to come related to this lecture series, so this will be our part number six and now we start our lecture [Music] Alright, so brother, see, in the previous lecture, which project did we create, asp.net kar wap with MC, that means with MC, which we created

00:01:00 In the project that we created, whatever our direct structure was, I will explain that to you in today's lecture and what is the flow of the application that we created in dot net course 6 , I mean which file is run first, then which thing is run after that, I will tell you all the things in today's lecture, okay so I go to the video, so look here, you must be seeing project SP kar WAP MBC, we had created this by using MBC template and what is inside the MBC template

00:01:26 What happens is that those folders of models, views and controllers are already provided to us, so brother look here, you can directly see the structure, okay, so inside this I will tell you, first of all we have a connected services, okay, now look, till now we have not connected any service here, so what is written, no service dependency discovered, means no dependency for the service is present here yet, so when we go ahead and create the dependency of the service here, then we will add it at this place, okay, then this which

00:01:51 Dependency Dependency means whatever libraries you are going to use in your application, some libraries are already present in it, means some dependencies are already present in it, some dependencies we will add in the future so look here inside dependency we have two folders one analyzer and one framework, if I show you the frameworks, look here asp.net framework is present and dot net framework is present, so whichever framework you use, you will see it here, in the same way in future we will fix NTT

00:02:20 It means we will add the core of Microsoft Anti Framework, inside it we will also see that here. Okay, then look at the top here, the analysis means the dependencies have been divided into two categories, one is framework because our application is also dependent on the framework, then the analysis means the rest of the dependencies here you will see them, like look here there is Razor, there is Shiksha, dot net login and there are many other things, okay, so you will see all these dependencies here.

00:02:44 You will get it, okay, I will band it from here, then look, you have the next section of properties, okay, look property, inside this you will find a file named loan settings.gm, okay, if I open it, then look here, here you can see some of its settings, okay, then below, here you can see the settings of our Castrol web server, okay, then look here below also we have some environment variables, its settings are done, that means, look here, here

00:03:09 Basically what is it to run our app, I mean our local development machine, I mean my laptop in which I am doing my development, what settings are required on it, look here, you must be noticing all those things about the server port, if it's done properly, then look here, if I explain some points to you in detail about launch settings.gm, okay, if I explain it from some theoretical point of view, then you will understand it better, so look here, what is it telling me here, what is launch settings Jason

00:03:35 In hp.net, look here, in the properties folder, I mean, where do we find this file which contains the launch settings? Inside our solution explorer, there is a properties folder, we find it inside that, but this is what is happening inside the properties folder, you can find the launch settings.json file which contains settings that control how your web app is started on your development machine, I mean, what is happening here, what settings are present inside the launch settings.gn file? There are such settings which control the thing that

00:04:03 How will our web application start and run on our development machine, which is my laptop, because as you know to run we need a URL, which means we need a local host, then we need a port number, so all these settings are available with us in launch settings.gm, which means the server that will run on our development machine, which is my laptop, whatever will be its port number, whatever will be its local host, all these things are limited only to our PC.

00:04:33 Because what we are doing right now on our PC is development. Okay, then look next point d settings with in this file are going to be that one d dot net kar application i.e. from visual studio and by using dot net karkali, that means look here it is saying that brother, all these settings that we have, they help us in running our application on our development machine. And look, whenever we run our application, we can run it in two ways, one is we can run it through Visual Studio ID.

00:05:02 We can do that and secondly, the dot net kar kaise kare which is a command line interface, what can we do through it, we can run our application. Okay, so you have to remember this point that whatever settings are there on the development machine through which our application has to run, where do we keep all those things, launch settings dot jaisalmer, okay then further, now see what the next point is doing here, the most important point that you need to know in mind is this launch settings dot gn file is only that.

00:05:30 With in the local development machine, I mean look here there is a very important point that you need to remember and that is that this file, which launch settings.json, is only and only available on our local development machine, I mean we keep this only inside this project of ours, which is kept on our local development machine, I mean look here, what is it doing, that date in fact this file is not required when we are publishing our asp.net application on the production server, I mean what is it doing here

00:06:00 Brother, as long as you are developing your application in a local environment, local environment means as long as you are running it on your PC, laptop, developing it, then you will definitely need these launch settings, but when you take your website and put it on the internet server, that is, when you publish it live, then you will not need this file there, so this is also a very important question which can be asked in interviews, so you have to remember it, okay then next, now look here, here

00:06:27 Look, this is showing the code of launch settings.gm. Okay, so look here, you can see this express, which means the settings that we have, these are related to what, look here, Ayat Express is related to the server and look below here, you must be seeing where you can see the application URL, okay, the port number of the local load is visible, okay, so what are these settings that we have, look here, cast double web server, which means by default our application runs

00:06:53 It is happening on which one, it is happening on the Castrol web server, so its settings are present with us inside the launch settings, but if you want, you can do it inside this express server as well and look here, what is the other setting that we have for this express, so what are we doing here by default, which settings should we do, we are doing the settings of the Castrol web server, we will go ahead and discuss it in detail that what is Castrol web server, what is this, okay, so if I do the same thing

00:07:19 Let me show you here inside my file. Okay, so look here, this portion is here. The settings of this express server are here and look, if I take it up, what you can see is that we have the settings of Castrol web server. So look here, here in the application URL, you can see two portions, local, along with it, a seven one five three and a five one five three, so if I run my application now, then we see that our local

00:07:47 Which port will be displayed along with the host. So, coming here, if I run my application, then look here, our application has been run and look here, it is showing you, meaning a host is coming and look here, there is a welcome message and the link is visible to you and look here, local last and which port number is it, look here, 7153. Okay, it means where were these settings defined, here, which look here, what you can see, it will use two ports, either 7153 will be displayed on it.

00:08:14 or it will do it to 5153 , so basically the settings that we have are the settings of the Castrol web server, this means that we have the proof of the fact that our application that is running here is running on the Castrol web server, but if we want, we can switch it, we can go ahead and switch on this as well, you can see its settings also here, okay, so look here, whatever settings are there, whatever settings are there that we are doing on our development machine, whatever settings are related to that,

00:08:40 Here we have the display, it's okay, then I stop it, let's move ahead, okay, after that look here, after properties, what is next, you root folder, brother, look, you root, we have such a folder inside which we store our static files, static file means, if we want to display an image inside our application, then after creating a folder of that image, where will I keep it, I will keep it in the view root folder, again similarly, if I want to do some tight front and framework,

00:09:08 This way once Bootstrap is done, I will keep the Bootstrap files also here, I mean where will I keep all the static files in which there is no changing, inside www.wood , like if I open it and show it to you, look here, the global code file of our site, look here, side dot code is also present here, the JavaScript file is also present here, and look here, there is a library inside it, look here, you must be seeing Bootstrap, you must be seeing that there is core validation and here

00:09:34 Look, you must be seeing the icon, so if I expand Bootstrap here in the library, okay, look inside it, inside the distance, you must be seeing what, that means, look, the Bootstrap files are visible here, and the GAS file of Bootstrap is visible here, so brother, look here, if I open the Bootstrap file and show it to you, then look here, which version of Bootstrap is it running on by default, 5.1.0 , it means Bootstrap

00:10:03 Version five is being used here which is the latest version at this time, so you must have understood one thing here that brother, in asp.net course 6 , whatever application you make in it, the default support for bootstrap is of which version, it supports bootstrap 5 , so look, all these static files are of bootstrap, look here, you must be seeing Jquery as well, look if I open it and show you, look here, you can see the file of Jquery as well.

00:10:31 Let me open it and show you, look here sir, you must be seeing the code here and see, the latest version of Jack Core is also there, it is 3.5.1. So, that means if you are using the Bootstrap query in the application, then you do not need to install it separately. That means if you are making such an application which has BBC support, then you get all these things automatically, but in the empty template, there are the MBC folders and all these things, we have to create them ourselves, okay?

00:11:00 It got cleared till here and look here, there is also Jquery validation, that means we have a plugin for core validation, if I open it and show it to you, look here, you might be noticing, if I open it, then look, this is the validation plugin 1.73.0, so if you want to use the core validation plugin for form validation, then you can do that work here as well, so you can see all these static files, we have them here, where are they present, look, they are present here inside the root folder, okay then look here,

00:11:29 The folder of controllers must be there, the folder of models must be there, the folder of views must be there. Well, before understanding these folders, look here, I add all these files here and I run my application once again, so brother, look here, you are seeing a page, but where is this page basically coming from, okay, so now I will explain it to you, okay, so look, I open my application from here, brother, look, it is already created here and inside this, look here, there is a class of Home Controller.

00:11:56 It is already present inside it. Okay, look, you can see the class of Home Controller. Now look inside it, there are some action methods. So brother, remember, whenever our application runs for the first time, which action method of home controller is run? This one here, whose name is index. And inside this, inside the MBC application, the concept of routing is also implemented inside the sp.net application. And the concept of routing tells us that brother, whenever you run your application, first of all the index area method of home controller will run.

00:12:25 OK, this is already defined in routing, brother, I will show you where it is defined, then see, when this action method runs, what will it return, it will return a view, so where is that view present, look here, under model, you can see a folder of views, inside this you will find the folder of home and inside home, look here index dot csssdml, okay, that means look here, this home controller was doing this action method yesterday and the view which this action method was returning, which view is this one returning

00:12:52 Okay, look here it is written Welcome and whatever link was there, look here, you must have seen it in which Building Code was written, so sir this is basically coming from here, okay, okay and look inside this there is another file also privacy.cssml, which means for privacy policy, okay, then look inside this there is also a shared folder, that means inside the shared folder you must remember there is a file like layout.cssml which is a file of the master page, so that also is already created with us here and then you must have seen that brother, whenever we

00:13:18 We used to do exception handling, which means if an exception occurs in our application, then this error.cssml file gets displayed to the user. Then look, there is view importance, this is also the same concept which we have seen in MBC, okay and then look here, look at the model here, by default a model is being created for which error view model, okay, so it means this becomes the models folder, because what the model basically does is that the connectivity with the database entities, we do that inside the model classes, we will also see this later, okay, so this is basically its

00:13:46 Directly the structure is okay, I close the views from here and then look, then we have the up settings note inside json up settings dot json, look here some configuration has already been done here but this is very less, going forward there will be more configuration inside it, okay, so I come back to the up settings again, okay, so brother, look, the file with up settings dot gn is basically what is it, brother, look, in the previous asp.net versions, we used to use the web config file there, inside the web config file, we

00:14:12 What we used to do was whatever our database connection strings were or any other work related to configuration, we used to do it inside the file but look here, there is no WAP config file, there is no global dot A6 file either, so where do we do that configuration work, basically what is the name of the .gov file here, it's settings.gov. So if I will tell you in detail here, the next one which is okay, then look here, what is asp.net doing, it's settings.gov. So look here, first point if you have a

00:14:39 With the previous version of asp.net application then you may know the importance of web dot config file, meaning here what I am saying is that if you have worked with the previous version of asp.net and if you have created a dot MBC application on this then you must be very well aware of the importance of web config file there, okay then see what I am saying here, in previous versions of asp.net application they generally used to store application configuration settings, database connection string, other applications, it was used as global variable and more with in web

00:15:12 Configure means what is being done here, in the previous versions of asp.net application, whatever configuration settings we used to do there, like connection string, okay, or global variables, where did we do that work, we used to do it in the web config file, but the web config file is not present here in the asp.net core 6 application, so where will we do that work, inside the settings.gsn file, okay then further, now look here, there are some more points, what is being done when we create an asp.net application.

00:15:48 Studio automatically creates up settings, means you see here it is telling that the up settings.gn file is created automatically with which templates by Visual Studio, see what it is doing here, empty project template means if you create an empty project using asp.net then inside that also you will find the up settings file, if you create a racer page, if you create a project of it, if you create an Ambith template, if you create a Web Api template, then along with all these things, the up settings and json file are there

00:16:16 Your Visual Studio automatically creates it inside your project. So look what it is doing here. The d.up settings and gm file is an application configuration file, which means the file of up settings, basically it is the configuration file of your application, just like there used to be a web.config file, so look what it is doing here. You store the configuration settings, database connection string, other applications, this is called global variables. Extra, which means whatever database connection string of your application is the global scope.

00:16:46 Are there variables or any other configuration settings, you will do all that inside the up settings file. So this is its important point. Then again directly on the structure, so look here, now under up settings, we have a file called program.co. Now look brother, this program.co is a very important file because brother, if I tell you that whenever you run your application, the first file that gets run is program.co. So how does it run, look here, look here, you must be noticing here.

00:17:16 Look, create a web application builder, create builder means you can understand from its name itself that what is happening is we are talking about creating our web application, we are talking about building it, so look here, I have created my web application and look here, I have sent it inside a variable of the builder and then look what I am doing, builder dot services dot add controllers with views, means look here I am creating a service here, means I am adding this service, this container here inside my application.

00:17:45 What will happen is that all our controllers and views will be executed in the correct way and then look here, builder dot build means this is what I had added here under services, I built it and after that I put it inside the variable of this app and after that see what is happening here, look here, I am checking that brother, if your environment is suitable for development, then it's fine, but if it's not suitable, then look here, what is installed, note, if it's not suitable, then look here

00:18:12 First you have to send the error and in the error the same error view which we had will be displayed, okay and then look here, now you must be noticing this, look at the direction, those static files, those routing, what are all these basically you have, what are middlewares, I will tell you this in detail later but here just understand that brother whenever we request our application, that means whenever we run our application, the web application, then after running, first of all it comes on this file and look here above this file, all the

00:18:45 Only those things which have been registered get displayed, others do not get displayed, like look here, I have placed an image inside the u route here, so if I want to display that image from here on my website, then that image will not get displayed until I do not add this middleware here, similarly look here, that routing will not work in our project until I do not add this middleware, again look in the same way, there is authorization and look here, whose pattern is that, which pattern do we have

00:19:18 The pattern of routing is that first the home controller will run, the index area method of the home controller will run, so what are all these middlewares that are defined here and after this our application runs, meaning you can understand that whenever you access your application, run it, then all these middlewares will be checked here because where does the request come first, it has come inside the program.com file, meaning you have a pipeline and whatever request we have, it will go through this pipeline.

00:19:50 It goes towards the application and runs it. Okay, so we have the concept of middleware which is declared where, it is declared in program.key, so if I explain a little about middleware in detail here, so look what it is doing here. Now, middleware is nothing but a component class, meaning what is the middle one doing here that it is nothing, just what is this, it is a class which is executed on every request in asp.net application. It means that whenever you make any request, whenever you get a page,

00:20:18 You request your asp.net and application, whenever you come on it for the first time, that means, whenever you request it, then what are middlewares that get executed and where is the middleware present, look here, the middleware is present in our program.com file and whose middleware am I, see what all these things are, they are middlewares, that means if you will not add these middlewares, then neither will you be able to add the image, nor will you be able to do routing, nor redirection, nor authentication, authorization because all your

00:20:48 The request is passed from this file, through this middle ware, it means all your requests of the application, they are passed from this file, through this middle ware, okay, if you understood, then our lecture is completed here, so if you liked the video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series, so if you have any confusion with this lecture, then do ask me in the comment box, so see you in the next lecture, until then see

00:21:15 Stay tuned my channel program [music]

00:00:01 Hello viewers, how are you all? So as you know we are having a lecture series on asp.net core 6 , so in today's lecture we are going to cover a very important concept called middleware. So as you know in the previous lectures we saw how we create an asp.net core 6 application and whatever is its directory structure, whatever are the folders, what is the function of the files, we saw all those things in our previous lectures. So I

00:00:28 I would recommend that if you have not seen our lectures, then first watch those lectures and then come to this lecture, so before starting the video, a small request, as you know this is my new channel Program Mentor, so if you have not subscribed to it yet, then I request you from my heart that please subscribe to it, share it and also press the bell icon because many more videos are going to come related to asp.net core 6 and I have posted advanced JavaScript i.e. modern JavaScript on this channel

00:00:55 And I have started a playlist of concepts of es6 as well, you will find that too on this channel, so this will be our part number seven and now we start our lecture [music] okay, so brother look here, what is today's topic, middlewares in asp.net core 6, okay, so let's discuss it, come forward now look here, I will explain to you with a diagram, the concept of middleware, okay, so look here brother, the way you know, we have a client server architecture, I mean, we have a client, I mean, browser, which

00:01:28 It is installed on our machine from where we request our web application, so there is a client and what happens along with this is that we have a web application which is present on the server, so what happens is that brother, what do we do from our client, that is, from the client machine, from our client browser, we request our web app, so where does that request go to our web application, that is, which web application in our case has the asp.net core 6 web application, okay and this web

00:01:53 Where is the application located? It is located on the server. So, whenever we send a request from the client to our web application, what does the web application do with that request? It processes it and then sends a response to our client. So, suppose I accessed a page here. So, where will that request go? It will go to our web app. What will the web app do? It will send us a response. So, for example, suppose I requested the home page from my client, requested the about page, or I logged in.

00:02:21 So whatever request you make like this, it goes to your web application, gets processed, and after that you get the response on your client. So now I will expand this diagram further and explain you the concept of middle ware in it. Come here. So, as I told you, we have a client and we have a web application. So, look, in between these two, we have a concept of HTTP request pipeline. Pipeline means that whatever request you send from the client, it goes to the HTTP request pipeline.

00:02:47 After passing through this pipeline, it goes to our web application, to our server. And what we do inside this HTTP request pipeline is that we define our middlewares, whether it is a single middleware or multi-triple middlewares, it is according to your requirement. So, whenever you send a request from your client, that request after passing through this pipeline, that is, after passing through these middlewares, goes to our web application, to our server, and then after that, whatever our web application sends us the response.

00:03:18 That too comes to you after passing through your middle ware, that is, it comes to your client after passing through this pipeline and see, I had told you in the previous lecture that where do we define the middle ware, in the program.cs file inside our application, our built-in middle ware are also defined in it and whatever custom middle ware we will create, we also have it inside that program.cs file because I had told you inside asp.net core 6 you don't even get the startup file global.com we write the logic and when that logic

00:04:02 It gets executed whenever we send a request from our client on our web app to our server, then that request is processed within this request pipeline and what middlewares also do is they handle that request and if we have to apply any logic on it, then we can also apply that logic within our middlewares like brother if I give you an example, for example, you requested a page from the client for which you would require authorization, then that is the work of authorization, the authorization which checks

00:04:33 For this, we can create a middle ware and that middle ware will check whether the user who is sending the request is an authorized user. If he is an authorized user, then he will be able to access the resource, otherwise an error should occur. So we can apply this kind of logic in our middle ware as well. You just have to remember one point here that whatever request you make from your client, that request will pass through your middle ware, okay, that means even when you make a request, it will pass through the middle ware and go to the web application, and when the web application

00:05:04 It sends you a response and that too goes through your middleware and reaches your client. Okay, just like if I show you that web application that we created in our previous lecture, in which we implemented the concept of MVC as well, that means we created a project in which we implemented the MVC template as well, that means we also got the folders of MVC, so if I show you that application first, then come here, okay, so look here.

00:05:30 We had created this application, App Core Web App MBC, let me open it, okay, so look here, we have this project opened and look, inside it there is a file program.cs, okay, let's go inside it, okay, so look, this is the template in which we had used MBC, that means we had created the template in collaboration with MBC, so look here, you can see the folders of models, views, controllers, so look here, I told you, first of all our application, look here, what happens, it gets built with the help of create builder and then whatever is inside it

00:05:57 The services are added here inside the builder container and then after building this application, it is stored inside the variable of this app, it is stored inside the object and then after that look here we have the environment of the app, we are checking it that if this environment is not working properly or any error comes, then this exception code will run here, which means the error page will be displayed, so what is this, we have these middlewares and look, all these that you can see below

00:06:23 You can see we have built in middlewares, that means whenever we send a request, first of all we get the control in this file and after that all these middlewares here, all the middlewares that you have that are registered, they are checked here, okay, the way you see here, a middleware is defined for redirection, okay, static files, that means if you want to access images etc. or any bootstrap files, then for that you have to put this middleware here.

00:06:48 Look, there is routing, there is authorization and look here we have all the things to decide which controller action method should be run, that means you can understand here that we have a request pipeline and inside it all these middlewares are registered with us, that means if I go to the home page, about page, login page, then every time all these middlewares that you have will be executed, that means all these middlewares work on each request because all these

00:07:15 Where are they declared? They are declared inside the request pipeline. So, these are built-in middlewares, right? How do we create a custom one? I will tell you about that as well. So, then come back to your slides. So, now you must be able to understand this figure. I will also show you practically with custom middleware. So, come ahead. Now, look here, there are some points. It is important to understand them first. So, look, what is it saying here? ASP.NET Core introduced a new concept called middleware.

00:07:39 asp.net core has introduced a new type of concept called middle ware. So what is middle ware? Look here. Middle ware is nothing but a component class which is executed on every request in asp.net core application. Meaning, what is it saying here that middle ware is nothing, it is just a class. What is its job? Its job is that whenever you send any request to your asp.net core application, then for every request, that is, for every request, we have this middle

00:08:07 What is a ware, it gets executed. OK, I mean it often happens that there is some task, some logic that I want to get executed on every request. So what can I do for that task? I can create my own middle ware. Then see the next point middle ware in asp.net core controls how our application responds to HTTP requests. It means that this middle ware inside asp.net core controls what all HTTP requests are coming to your application.

00:08:38 Whether it is a get request or a post request or any other request, how do we handle those requests? I mean, by filtering it or by applying some logic in it, how to handle that request and send its response. Then look at the third point, middle ware are software components that are assembled in an application pipeline to handle requests and responses. I mean, look what is being said here, middle ware is also being called a software component that is assembled and inserted.

00:09:08 Where in your application pipeline, and what am I calling application pipeline here which is our program.cs file because what I told you in the previous lecture that before the controller, we have the program.cs file which gets executed and all the middlewares inside it get executed, so look here it is saying that your middlewares get assembled in your application pipeline and their job is to handle whatever request is coming and whatever response it gives

00:09:37 It has to be managed. So this is the job of middle ware. Okay, then come forward. Now see here, it can also control how our application looks when there is an error. I mean, brother, look, if any error occurs inside our application, any exception occurs, then in that case, what action should be performed, which page should be displayed or what action should be performed, we can do all that work with the help of middle ware. Like, look here, the application that I showed you, had some code already in it, okay.

00:10:06 Look here, let's go to the top, look here, you can see that brother, if this environment does not work properly for our application, meaning any error comes, exception comes, then look here, I have used a middle ware, use exception handler, which basically displays a page for us, displays the error page, okay, so that is what I am talking about here, then look, the next point it is a key piece in how we authenticate and authorize the user to perform specific actions, meaning, look here I am talking about the authentication and authorization work.

00:10:33 No, after logging in we often check whether this user can access this particular resource or this particular page. So the authentication and authorization work can also be done in the middle ware, because as I told you, whatever request you send from your client, that request goes forward after passing through our middle ware. So, only then we do the authentication and authorization work within it. Then come to the next point, each piece of middle ware in asp.net core is an object and each piece has a very specific

00:11:05 Focus and a limited role brother, see what I told you about what middle ware is, it is simply a methods, meaning you have built-in middle ware as well as custom ones, same way as we see built-in methods and we create custom methods ourselves, okay we can do that work here with middle ware as well, so what am I saying here, what is middle ware, it is an object, meaning it is a method which has its own specific task, meaning it has its own limited role, as I told you here, here

00:11:32 Look, okay, look here, the middlewares that are registered with us, you must be seeing, as I told you, these are for exceptions, for error handling, these are for redirection, these static files, just like images, bootstrap files, CSS, JavaScript are used to handle them, then there is routing, then look there is authorization, then look here, there are things here to define the route, meaning every middleware has its own specific task, specific role, okay, so that's the point

00:11:57 It is telling here, okay, so now I come and I will show you by doing it practically, that means I will create an application for you and inside that I will teach you how to create a custom middleware, okay, so let's come here, okay, see here our application is already made, for middleware, I will create a separate application, go to new, go to project here, okay and look here, I get it searched, in the template, I write simple asp.net core here and see, the application that I create

00:12:20 I will create it, it will be a MT application, okay, it will be an MT application of asp.net core, which means I don't have to do the model view control work here, okay, I select simple MT application from here and click on next, okay, so look here, it's asking for the name, here I say SP SP Core and here I say middle ware, okay, I named this application App Core Middle Wares, let's do next, okay, look here, I am using .Net 6, long term support, I click on create, okay, so here

00:12:47 Look, our application has been created and look, if I run my application here and show it to you, then let's see its result, what result do we get, okay, so look here, our application has been run and look here, what message are we getting, Hello World, now how is this Hello World coming because look here, I have created an MT application, here I do not have any MBC folders, okay, so if I show you its result, okay, I come here and stop it, okay, so I have shown you

00:13:11 Did I tell you that our request pipeline, whatever request we handle, where is it located, look here inside the program.cs file, if I open it, brother, look here, there is very little code written inside the program.cs file, okay, let me show you what is in the code, look here, our web application is being built, okay, with the help of create builder, okay, and then look here, after building this builder, where is its reference being passed into the app, and now

00:13:35 Look here, a middle ware is being used with the app, what is its name, map gate, okay, what is happening in map gate, look here, whenever this URL is typed, because you know, when our application runs for the first time, then we have a simple forward sl route, so as soon as this route will run, look here, a call back function is running, inside which what result is being displayed to us, hello world, okay, so brother, look, right now it is handling our request, okay, so what do I do, I will put it here

00:14:00 For now I am commenting it because I will create my custom middle ware here because look we haven't talked about routing yet, so that's why I have commented it here, so look after commenting it, now our application does not know which code it has to execute, so now if I come here and run it, then you can see the result, look here, what message is coming to you, this local hose page cannot be found, means here it is not able to know which code

00:14:25 I want to access the resource, who wants it to be displayed. So look here, I will stop it here. So look what I do here is I will create my own custom middleware. So for that, what will I do here, look here, app, and with it I will use a method run. So what do we use here to define middlewares, we use the run method. So coming here, I tap on it. And look inside it, inside it, I pass a callback function here.

00:14:49 I will do it, okay and I will define its parameter here then look here I am passing a function here, callback function, okay, which you can also call arrow function here, okay, look here its body has come here and now look, I have used a sync here, so with this I will also have to use await, okay so look here, I say await and after that look here, this context brother, see what will be inside this parameter, means whatever HTTP request we have, to handle it, we have used it here

00:15:21 I have created a parameter and what is its name? Context. You can also call it HttpContext because it is derived from HttpContext. Okay, look, you must be seeing it below. Okay, so now I say that brother, whatever request is coming here, it will go to the contact. Okay, what do you do? Look, I say contact here and near it, now I say the response that you want to send and what do you want to write in the response and send it. Look here, write a sink and here I pass a message inside it. Welcome to asp.net core 6. Okay, meaning now

00:15:52 What will happen is now the request that we have will come here to app.run and look this is what I have created, okay, this is basically what we have, our middle ware and what is this middle ware doing, brother whatever request comes, okay, whatever request comes, what do we have to send in response to that, in response to that we have to display this message, that means this middle ware that you have, okay, what is middle ware basically, this callback function, what am I calling this, middle ware and how have I created this middle ware of the run method

00:16:21 By help it means that whatever request comes to our application, okay, against that this middleware that we have will be executed, okay, but notice one thing, even after this I have written app.run here , inside which there is no callback function, means there is no parameter, so now understand the concept of what is happening on this line, brother, look, on this line our server starts and our application is hosted on the server, it starts, okay, so that's why writing this line is also too much.

00:16:50 It is necessary. Okay, so now if I come here and run it, then you want to see its result. Look here, you can see the result. That means, against the request that we made here, look here, our middle ware is getting executed. Okay, and what did I tell you that brother, for every request, for every request, your middle ware gets executed. That means, look here, now look in the URL, what is simple forward slash, so if I write five slash and press enter here, then we have

00:17:14 See, we have this running for this request as well and if look here, I do slash home, okay, the way we access our home page, okay, here I write in short home like this, okay, so if I come here and enter, then see, even now we will have this middle ware executed, you see, okay and if I come here and do about, okay, then for that also we will have this middle ware executed, means for each and every request we will have this middle ware executed because what did I tell you

00:17:38 Often we have to do this, brother, to check every request, we put some logic here, just like authentication or authorization work is often done, if it's done okay, then one point you have to remember is this, okay, good then if you come here, I come here and stop my application, brother, look what I have created here now, now I have just created one middleware here and I have created it with the help of this, we have created it with the help of our run method, brother, we can create multiple middlewares as well.

00:18:05 Inside your request pipeline, whether it's one, two, three or four, it's up to you how many custom middlewares you want to create, and whatever built-in middleware you want to use, you have to add that here as well, as I told you, use static files, use authorization, routing, but you have to take care of one thing brother, look, the order of the middlewares we have plays a very important role, it means the middleware that you have at the top, in the first position, will run first.

00:18:37 The one that is on the second time will run at the second time, the one that is on the third time will run at the third time, so it depends on you brother, what order do you set for it because it takes the order very seriously, so look, if I copy this and come down and paste it here, okay, and here for example, I write, here I write the name of my channel, program mentor, okay, program ment, and here I have written the name of my channel, so brother, now let us check that we have both these middle wires.

00:19:04 Does it run or not? Okay, so I run my application here. So look here, what is happening here is that only the first middleware is running, the second middleware is not running. What is the reason for this? Look, the reason for this is that now see how I am defining this middleware by using the run method. So you have to remember one thing that the run method never executes your next middlewares, means it does not execute the subsequent middlewares that you will have. Which run method? Why?

00:19:35 The reason for that is brother, see what happens is that inside run we have a concept of next, that concept of next is not present with us inside run, we also have another method for creating a middle ware whose name is use, so brother if you want multiple middle ware to be executed inside your request pipeline, that means you want your request to pass through multiple middle ware, then for that you will not use run here, you will use the use function here.

00:20:04 Okay, look, I will comment out this code from here and also copy it, okay and I will paste it below and now I will use it here, there is a function whose name is use, this use function, inside this we get the facility to call the next middlewares and how do we get it, look here there is context, after context here I create a parameter next, okay, that means also remember that inside use we have two parameters use but the run method

00:20:32 Inside this we use only one parameter and look this next that we have is basically a request delegate whose job is to execute the middlewares that we have ahead but what will I do for that, I mean how will I use this next, look here I put await again on the next line and then look here I say tap tap and look next has come and inside next I pass my context here, I mean whatever

00:20:59 This is a request, look, I am transferring that request further as well so that the next middle ware that we have can also have access to this HTTP contact. So now look, when I have put next here, it means that this second middle ware that I have here will also be executed. So look, if I come here and run it, you can see the result. So look here, look, this is your first middle ware and look here, this one stuck with it is our second middle ware.

00:21:26 Here I will do one thing, here I will give it a little line break, okay so look here, where we have written as.net code 6 , after this I will put a ban here so that a line break comes on it, okay so now let us run it again, look here now you can see that means we have the first middle ware also running and the second middle ware is also running, okay but remember one thing, if your concept is that you want to execute multiple middle ware, then do not use run here.

00:21:54 What do you use here? You should use use function only here because brother, suppose if we have any middle ware after run then it will not be executed because run does not execute the next middle ware, what is the reason for that because look here, in the run method that we have, only the context parameter comes, we cannot give the parameter of next in it, okay, so there is also a difference of this that brother, inside run we have only context, means there is one parameter for use

00:22:23 Inside we have multiple parameters and inside run we don't have next, there is no delegate of next that is why this run cannot call the next middleware but if you want to call multiple middlewares then what will you do here, you will use use here, okay, that means look here I will use it, okay and here after the context I will put it here, next, okay I put next here and look here, after this I will put this line here

00:22:50 I'll apply the break, okay and after this, now look here there is await and after that I put next here, okay and inside this I pass the same context here, okay, so now what will happen with this is that if we have a third middle ware as well, then the third middle ware will also get executed here, okay, so you should understand this difference about run and about use here, okay, so now if I here on my slides, okay, look, I have told you this, now come ahead, okay, look here, next

00:23:15 What the point is saying is that ultimately we need many pieces of middle ware for an application to behave appropriately whether it is a custom middle ware or a built in middle ware. So, if we have to take our request forward by passing it through multiple middle ware, then in that case, what did we tell you that run method is also used, use method is also used, so here the same point is being told. Okay, then see the next point, the middle ware has access to all the requests and responses, means, here he is talking about the same thing that your middle ware should be accessed.

00:23:52 Access to all requests and responses is available, which means that all the middlewares that you create inside the pipeline, those middlewares are executed for all requests and responses. Then look at the next point, the order of middleware is very important. You are seeing the same thing that I told you that the order of middleware is very important. How important is it? Look here, like look here, this one which I have created with the program mentor middleware, okay, the logic can be anything.

00:24:19 It depends on you, if I place it on the top, okay, that means look, I changed its order, so it will execute in the order in which you place it, okay, so look, I brought it here, changed its order, and now look what I do here above the third, this is the one that I had commented, okay, I bring it down, okay, I come here and paste it, okay, and coming here, I uncomment it and here, for example, I change its name to any, okay, I write its name here.

00:24:46 Hello Hello viewers, let me write the text here, okay, because look here I am saying that brother I want to go to the next, so you will have to specify next here necessarily otherwise an error will come here, okay, and after this, look this is my last, it is the last middleware, so look here, I did not use use, rather I used run here, okay, correct, so now if I come here and look, if I run it, then now you will see its order, means it will execute in the order which I have set here.

00:25:11 Okay, let's come here and run it, so look here, now you can see the result clearly, see, the order that I had placed at the top of the program end is coming, then look, welcome to sp.net and then look, hello viewers, means, in this way, you definitely have to set its order here, okay, then coming here, I stop it, then we move ahead on the slides, okay, so you must have understood these points, now look, here I also have a diagram, okay, understand the diagram here, okay, look here, I

00:25:35 What did I tell you that you can also create multiple middle ware, so look here, whatever our first request comes, it will go to middle ware one and whatever logic is written inside it will be executed and after that logic, you have to give the next method here as well, that means you have to use the use function so that our request goes to the second middle ware as well and then look, if you want to execute the third one, then look here, here also we have the next method which will take this request further in the middle ware.

00:26:02 It will take us to this middle ware and we will also have logic written inside it and look brother, there is no next method inside middle ware three because I don't want to take you to any other middle ware beyond this, so in middle ware three, I used the run method, I did not use the use method, okay and after that whatever code of our application will be there, that means it will be a web app, it will go there and after that whatever response we have, okay, those responses will also go to us after passing through these middle ware, to our client.

00:26:28 We will go and also on the response, if you want to put any logic, you can write that here as well, okay and brother, remember one thing, look here, where I have called the next method, you can write code even after this because what is happening is, look here, who is calling this next middle ware, it is calling our first middle ware, means what is the first middle ware doing, it is calling our second middle ware, it means the second middle ware is being called here and this line after this, okay

00:26:57 You can write some code here also, it means when we make this call, then the request again comes above our first middle ware, so it means if you write some code even after next, then that will also be executed, but when will it be executed? When you have this middle ware called, after that, okay, so now you must have understood this figure very well that whatever request you get, it passes through all the middle ware and whatever is your response, it also passes through all these middle ware, okay then

00:27:25 Come here, I will also tell you the definition of these methods. See here, what is it saying, middle ware defined using app.run method will never call subordinate middle ware. Subsequent middle ware means that brother, the middle ware that you will create using the run method, right? Then, all the middle ware that comes after it will not be executed. Subsequent means if you created the first middle ware with the help of run method, then it means the second middle ware, the third middle ware will never be executed. So, if you create multiple middle ware

00:27:54 If you want to execute the middlewares, then for that you will use the use method. So see what it is saying in the use method. The use method places a middleware in the pipeline and allows that middleware to pass control to the next item in the pipeline. Next item means the further middlewares in our pipeline, within the request pipeline, we also want to get them executed. So for that we will use the use method. Meaning, what we will do through the use method is we will create a middleware so that our further middlewares

00:28:22 So that they can also be executed successfully and I also told you about their parameters that we have two parameters in the use method, contact and next, but in the method that runs, only one parameter of http contact comes. Ok, it's clear till here, our lecture is over here, we are winding up. So if you liked my video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series, so subscribe and share it as well.

00:28:48 So if you have any confusion regarding this video then do ask me in the comment box. See you insha Allah in the next lecture, until then keep watching my channel Program Eater [Music]

00:00:01 Hello viewers, how are you all? As you know we have a lecture series going on on ASP.NET course, so in today's lecture, we are going to discuss a very important concept called routing. So brother, before coming to this lecture, if you have not yet seen our previous lecture in which we talked about middles, in which we saw how to create an MBC project, we also saw how to create an empty project in speed.net course, so first you should watch those lectures.

00:00:30 After that if you come to this lecture then this will be our chapter number 8 part number 8 so now we start our lecture [music] so look here, today's lecture is about routing, okay it is a very important concept, how do we do it inside speed.net code 6 , we will talk about it, we will also talk about what is this, so brother, if I take you to the definition of routing, before that I want to explain a simple concept of it to you, brother, look, every person has some destination or the other

00:01:04 But if he wants to reach there then to reach the destination he has to prepare a path, he has to take a path, what does he do by walking on that path, he reaches his destination, okay, that means he has to reach his destination, so he takes the path accordingly, the path is long, so the concept of routing is also something like this, okay, see what routing has been taken out from the path and see what concept of routing can be there inside a web application, brother, but for example, I send a request to my web application from my browser.

00:01:34 Meaning that I have to write some URL on my browser and that URL will be of some web application and for example, I say that I want to access the home page from that web application, so I will write something similar in the URL, so whatever URL I will create and send it through the browser, means this concept is called sending request, whatever URL you want to write on the browser and as you enter, the request has come to your web application which is present on your server, so there we have a

00:02:06 There is a concept of routing. It means, the URL that you are requesting, does any resource related to that URL exist, means is there any data present which will be shown to you in the response. So routing checks this thing. It means, there is a routing mechanism present in dot net code 6 , which receives your addresses, checks whether any data related to this URL is present, then it sends your request to the controllers, action methods inside your web application, which do some or the other thing.

00:02:40 When we return the response, it takes it from there and returns it to you on your browser, from where you sent the request, you will receive the response at the same place, so this is basically the concept of routing, so if I explain this to you with a definition here, look here, what is routing happening in asp.net kar MBC application is this a mechanism, what is it being called, it is being called a mechanism, it is being called in a way in which it bill inspector d incoming request date is ears, means this is the mechanism of routing

00:03:10 What does this inspector do? What does it do? What does it do? What does it do? What does it do? Inspector means to inspect. It means that it investigates your incoming request. Whatever arguments you are entering, okay? Whenever we enter a URL, it means that we are sending a request to a web application, then whatever arguments you have, the routing mechanism inspects them, that is, it checks them. And then after that, it maps the request to the controllers and the direction method. And what does it do after that, when it selects your URL, it lets you do that.

00:03:43 After that, it maps your URL and connects it with a controller and with an action method of the controller, meaning whatever request you are coming from, whatever URL you are trying to access here, have you seen the routing, brother, related to that, do we have a controller or any action method of it, so if that action method is present, control is present, then what does that thing get in the response, it gets returned, so this is the concept of routing, basically, you have, okay, so now I will talk about this thing

00:04:13 If I explain it to you with a little diagram, look here brother, you know we have a client, client means our browser which is installed on our machine and then there is our web application, okay, web application means the one which is present on our server because whenever the client makes a request, basically it requests the server where our web application is present, so what does the client have to write here in the browser, he has to write a URL and when he enters the URL, this is called making a request, it means what is the client doing

00:04:40 Some or the other page is mixing and asking me to bring that page, that request is made, then in our web application where we have our routing mechanism, okay, and the routing mechanism is appropriate, the controllers, the action methods, the data is taken from there and back in the response, it gets displayed to you on your client, on your browser, okay, so this concept is going on, now I will elaborate it further, okay, here it is, okay, so look here, suppose we have this user, okay, whatever the user is,

00:05:10 Whoever will access the application and make a request, from where will he do it? Browser means what will the user write on our browser, he will write the URL, and the URL can be something like this, see, you might be seeing local log in, then port number, and then slash home, so it means that he is trying to access this home page, and by writing this in the URL, he is sending the request, so that request first goes to us, where is the routing, that means our routing mechanism inside sp.net, this URL goes to it, well

00:05:42 So here you should remember one thing that whenever you request any URL, remember that it does not go directly to the controller, your URL before the controller goes to our routing mechanism because you will remember that when we studied MBC, I told you there that whenever a user sends a request, it first goes to the controller, but what actually happens is that first of all our URL, which is our request, goes to our routing mechanism.

00:06:12 And then after the routing mechanism, it goes to our controller and it goes to the action method. This means that whatever URL you write, it will first go to the routing and what will routing do? Then whatever will be the appropriate controller's action method, what will it do inside your speed.net application, it will get the data from there and in the response, get that returned to your browser. This means like look here, I am accessing home, so by default you know what happens, the index field method of the home controller is run.

00:06:44 So when I request this URL, then routing will check whether a controller and its action method related to this URL exist in our application. If it does, then whatever result that action method returns to you, that result will be visible to you as a response in your browser. So this is basically the concept of routing. So you have to understand this figure very well, only then you will understand the concept of routing well. Now look here, I am explaining it to you with another figure.

00:07:14 OK brother, look here we have browsers, you can use any browser to send request to your Radha application, so what will you write here, you will write the URL and whenever you write your URL and enter it, then first of all see here to whom that request is going, it is going to routing, okay, it means who will receive your request first, routing and look let me tell you one thing that where do you implement routing, you implement it in your program file brother

00:07:44 See, in the previous lectures we had created an ASP.NET application using the MBC template and what we had done was we had created an ASP.NET application without the MC template, so there I had shown you the program.key file and I had also told you that brother, whatever middleware is related to routing, whatever things are there, where do you write them, inside the program.key class and look, that is the program.key class, that is the entry point of our ASP.NET application, okay.

00:08:13 You also have to remember this, so brother, look, whenever you send a request from the browser, first of all, your request URL will receive the routing, the routing will check whether any controller etc. related to it exists, then where will it send your request, it will send it to the controller and inside the controller itself, you know, we have action methods which return the response to us, and look, then I told you, if the model is present with the controller, it means the controller can also take data from the model and you know the model can do such

00:08:42 There are classes which interact with the database, so it means the controller model means the data lake from the database is accessed by its controller and then it sends it through the action method where in our view, okay, when it is sent in the view, then the same view will be visible to our user as a response on the browser, okay, so this whole floor is going on, you have to remember this because what I told you, who receives your request before the controller, routing receives it, so this is one of the points, you have understood this well

00:09:15 You have to remember this. Okay, then go ahead and now look here, one more very important thing, keep this in your mind, brother, this routing is made up of two things. Okay, look, one is the URL; another is the HTTP method. URL means, just like I showed you the URLs, whether it is local, scholar, that port number, and then whatever is followed by slash home, that is the URL. And every URL has its own HTTP method as well, brother, you must have seen the method on many STDs, the way it is get, the way it is post.

00:09:46 The way it is put and the way it is deleted means that all these four methods that you are seeing are accessed only with our URL, they are done with the URL only, brother you know that if we have to fetch data, but for example if you are making a Kiran application in which the operations of insert, update, delete, select are happening, then whenever we get the data, we do a get request, that means we do the Get HTTP method, whenever we insert data, we do the post method in static and whenever we update the data

00:10:15 Then we use put request and also use positive method and whenever we have to delete data, then we use delete HTTP method. Okay, that means we use all these four methods with our different aliases, but like an example, look here, you can see the URL, slash home means local URL, then port number, office slash home, okay, this is the URL and then the second URL, look, local URL, similarly port number and then update, look, slash one, what is this slash one, this is the ID, brother, look, you must have often seen that whenever we

00:10:47 If we update or delete data, then we update or delete through a particular unique ID. So brother, look here, there are two URLs. The first one has slash home written in it. So you can understand that this URL can be used for get request as well as post request. Similarly, this second URL that we have, this update slash one, can be used for port, meaning it can be used for update as well as delete. So, remember, whenever you implement routing,

00:11:16 So along with the URL, its HTTP method is also present which means that HTTP method also plays a very important role from which we come to know that our request is going to be got, is going to be posted, is going to be put, is going to be deleted. All right, going forward when we will study the concept of apps and work on crores of applications, then there also I will get this thing implemented practically again. All right, so you keep this slide also in front of you, now look here, now look at this from some theoretical point of view. All right, see what this is doing.

00:11:48 Mapping is done by the routing rules which are defined on the application. Brother, see whose mapping I am doing. First understand this. Mapping means whichever URL is being requested, that URL is mapped with any action method of a controller. Mapping means connecting it. In the same way, our url is there, if I am accessing slash home in it, then where will that URL go and connect, where will it get mapped with the index area method of the home controller, so this thing is coming here, mapping, so who defines this mapping for our

00:12:19 We have to write rules inside the routing mechanism, I define this thing and what I told you, where will we write the routing rules, the middleware, inside the program dot key, okay then look here, the next point, we can do this by adding routing middleware to the request processing pipeline, brother, see what I have been doing in the request processing pipeline, when we read the middleware, I told you, inside our program dot key file, we register all our middleware, okay

00:12:46 And all the middlewares are present in the program.key file which is the entry point of our application so the middlewares related to routing, I mean whatever work is related to routing, we do that also in our program.key file which is called request processing pipeline here and remember I told you that middlewares can be of many triangles, I mean there are middlewares for different varieties of tasks which are for routing, for authorization, for authentication, okay for static files

00:13:16 There are many more for that, we'll discuss that later as well. Then look here, the next point is that the speed.net framework maps and connects the incoming requests data to the alerts and controllers, action methods based on the routes configured in your application. That means, here I'm saying that the routing mechanism has to be configured in your application. So, when you configure it, and set up the routing in the program.cs file, then accordingly, all your alerts, all the incoming requests, they get mapped.

00:13:50 With the action method of your controllers, okay so this is also a very big point, okay then move forward now look here, what is it doing, what are the different types of routing supported by asp.net corembc, means look, we have asp.net core 6 mbc, how many types of routing does it support, okay so look what is it doing here, in hp.net core mc application, you can define route in which you can send r s follows, means we have two types of routing, by using which we can take our further

00:14:20 We implement routing within a pay.net application, first of all we have convention based routing and secondly we have attribute rounding, that means convention based routing is also done, attribute based routing is also done, but brother see, a trend has been set in this pay.net, as per the developer, this convention based routing, we do it in the web application that we create, but we do attribute based routing where we have to route the API along with our sp.net application

00:14:51 Whichever app needs to be implemented, there we use attribute based routing but it is not necessary, it depends on you that where you are going to use convention based routing and where you are going to use attribute based routing, so now in the next lecture, we will first talk about convention based routing and after that we will talk about attribute based routing, so here our lecture is wrapped up, so if you liked my video, then definitely subscribe to my channel and also share it and also press the bell icon.

00:15:17 Also press this because there are more videos coming in future so if you have any confusion in this lecture then do ask me in the comment box. Keep remembering me in your distance. See you Inshallah in next lecture. Till then keep watching my channel Program Matter [Music]

00:00:01 Hello viewers, how are you all? As you know we are running sp.net code 6 and in this we started the concept of routing in our previous lecture, so in today's lecture we are going to talk about convention based routing, so brother, if you have not seen our previous lecture, that is, you have not seen part number 8 in which we discussed routing, then first see that lecture, after that come to this lecture because see, in the previous lecture I had told you that we have convention based routing in this

00:00:30 There are two types of routing in pay.net one is convention based routing and the other is attribute based routing so brother look, today's concept is convention based routing, I will explain it to you in two ways, you remember we had created a project with MBC template and we had created a project without MC template so I will explain the concept of convention based routing to you in the concept of both, means I will create such a project in which Attribute will be present i.e. Model Views Controller which

00:01:01 There are folders, they will be present inside that as well, I will explain to you the concept of convention based routing and we had created an empty application, that means we had created an asp.net application in which there are no folders for models, views, controllers, that means it is a completely empty application, so how will you implement convention based routing inside that empty application, I will tell you both those things in today's lecture, so before starting the video, I have a small request, if you join my channel

00:01:30 If you are new and you haven't subscribed to my channel yet, then I request you from my heart to please subscribe to this channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our part number nine and now we start our lecture [Music] So look here, we have the first type of routing which is called convention based routing. Now, what does this convention mean in this? Okay, first I will explain it to you and after that we will come to its diagram and theory etc.

00:02:04 First of all, I will show you by creating an application with MC template and inside it I am going to show you what convention I am using. So let me come to my Visual Studio. So look here, I have opened Visual Studio 2022 here. So here I create a new project. Click on it. And look here, the application that I will take here will be asp.net web model view controller. So let's take this application from here and here I click on next paper.

00:02:31 And look here, it is asking for the name of the application. Here, I am going to name my application - Routing with MBC. Okay, what I am doing here is Routing with MBC. So, coming here, I click on Next. And which framework are we using here, dot net 6.0, which is long term support. So, coming here, I click on Create. So, look here, our application has been created. And look here, while creating this application, I had also added the MBC template. So, look here, we have Model Views.

00:02:57 Followers of controllers are also coming and look here, you must be seeing the program.key file and see, I had told you in the previous lectures that the program.key file is the entry point file of our application, that means, whenever we run our application, first of all you should understand that this program.key file is the first one to run and inside this, I had told you that our HTTP request pipeline is configured inside this as well and all the middlewares that we add, all of them

00:03:25 We register the middleware here inside this file and look here you must have seen a middleware, okay, look at this, inside which our route is defined and look here, you can see the pattern, this is completely in curly brackets, basically this is what I am calling convention and this is called convention based routing, what is convention basically, I mean, whenever we enter a URL, then the pattern of ours, means by following which pattern our request, our URL, is

00:03:56 It will be generated and that pattern of our URL, we define it here by creating a convention like this so that our application knows that we are following this pattern, so look here now, when I created my application using MBC tablet, then look here, here it was already created, meaning it has already created the convention and given it to us, what we just need to do is we have to mention the controller here, okay and whatever action method is there, we have to mention it here and if you have an ID, then we have to mention it here, look here along with the ID

00:04:27 There is a question mark. What does the question mark refer to? This parameter is optional, which means that in your URL, the controller and action method are compulsory, but the id that we will have is optional because look where this id is located. In a way, if I want to get the data of any employee, that is, the data of any particular employee, then I can use the ID there. If I want to get the data through the ID of any particular employee, then I can use this parameter if I am in an application.

00:04:57 If I am updating any data, then I can use this ID there. Even if I am deleting the data of any employee, then also I do it there, but this ID parameter is not there everywhere, that's why we have what is the ID here, it is optional and what is the symbol of optional, here there is this question mark. So this is basically the convention, by following this, all our URLs, wherever we say, are accepted. So look here, what is it telling us that which control we have first by default?

00:05:27 Home will run and inside home we will have the action method of index, so if you look here inside the controllers folder, look here, home is already created and look here inside the home controller, look here, you must be seeing the index area method, why am I calling it an action method, look here, what is its type, this action result, I will tell you about it later, Majid, when we will do the action method completely separately in the lecture, then I will tell you what the action result is, okay, so you understand what is this index of the home controller, this action

00:05:56 This is a method and see what will it return when it is executed, it will return a view, so this view is also already created with us in the MBC template, so look here, let us go to the views folder, look here, you must be seeing the home folder and look here, index.cssmtml, if I open it, look here, welcome is written inside it and look here, you must be seeing a link in which this message is written, so meaning now if I come here and run my application, then you want to see its result, look here, you will see

00:06:24 The result is visible, that means the index area method of the home controller is running, see, it said Welcome here, and here you were seeing a link, okay, so this is basically happening because look here, in the MBC template here, we have this routing which is already implemented here, you see, that means look here, it has been activated by routing and then look here, this is a middle ware of map controller out, through it, look here, its pattern, that is, the convention is defined through here.

00:06:53 It has been told that brother, see which controller will run first and after that which area method will run here, you can see, you can also change it here, okay, meaning if you want any other controller or any other index area method or any other view to be displayed, then you can provide that here as well, okay, so whatever controller and action method you put here, according to this your application will be generated here, but brother, look here, all these things have already been created with us, but if we create an empty application and run all the things from scratch

00:07:23 When we build things, how do we work? So look brother, I am going to tell you here that brother, if you are going to make an empty application using dot net code 6 , then how do we add routing in it, how do we add MBC, that is, how do we add model view controllers, because when you learn this, then you will understand very well how to run your application in future. So look, I will create a new project here. I go and click on new here.

00:07:51 I am in the project and look here now I will take an empty application, okay look here, you must be seeing that don't do asp.net, okay, I will select it, I will click on next, okay and you see, I have to name it here, Routing without MC, okay, what am I keeping, Routing without MC means it will be a simple template, it will be an empty template, okay then click on next, okay here we have to select dot net 6 , again I will click on create here, so look here, here this application has already been created, so here

00:08:18 Look, here right now you might not be seeing the MBC folder, you might not be seeing the root folder, that means there are not many things inside it, so now we will do all those things here, okay so if I show you the program dot file, then look here, you might not be seeing any routing here as well, but look here, here a map get method has been done which is telling that brother, whenever your application runs, because whenever our application runs, by default we have this slash power

00:08:46 Slash so when it runs then we have hello world, that means a call back function is running and in that function we will have hello world displayed, okay and if you look here in the first line here, our application has been created through create builder, that means through the instance of create builder, an instance has been created here and after that look here, I am building this very instance here, so mean you can understand what is happening on this line, our application is being built, is being created and after that we here

00:09:12 What are you doing, you are performing routing, which means you can routing this as well, but this is not the correct way to implement routing, I will tell you how to implement it, so now see, if I run the application and show you, then look here, what result are you getting here, hello world, but brother, we don't want to display the message in this way, we have to create the controller, call its action method, then show the view, so what do I do for that, look here, first of all I do it here

00:09:37 I come here and stop it so first of all I right click on it here inside my application and go to add and after coming here I will create a new folder, okay and inside this new folder I mean I will name it here Controller, okay and inside this I will add my home controller here, okay so right click on it, go to add and then look here you must be seeing the controller, I click on it and look here I will not put MBC controller here

00:10:04 I am late and let's add it, so look here, the MBC control is not selected and below you have to name it, I will keep its name as this, Home Controller and click on add, so look here, inside the controllers folder, Home Controller has been added and now look inside Home Controller here you can see an action method called Index but now you see it is returning a view but till now we have not created any view and neither do you see any folder of views here, so how do we create a view

00:10:30 Look here, you can see the index, right click on it and after right clicking, look here, there is an ad view, click on the ad view and then look here, you have to select the razor view from here, okay, select the razor view and click on the ad below, okay, so look here, it is asking for the name of the view, you can change its name as well, but we will keep its name the same which was the name of our action method because this review of ours will happen with this action method, that means if the name of the action method

00:10:54 It is an index so what should you name your view here? Index is fine, the rest what is this template, what is the model class, I will tell you this later, now what you have to do is just click on add, so look here, this view has been added to you and look what is its name here, index.cssdmlcs what is .csml, is the extension of our view, so you have to remember this also and then look here as soon as I added this view, look here in the solution explorer, look here a folder of views has been created and look inside this there is a home folder and after home

00:11:24 Look inside the folder, this is our view, so brother look here why is this the home folder, because brother look, what is this view doing to the home controller, it is referring to it, so when I created the view for the home controller, what it did was it created a separate home folder inside the views folder and inside that, I added our index dot com female file here, meaning, if I create more views inside home, then all those views will come inside this home folder, okay, so if you look here, I write here

00:11:52 hm the index view is okay from home controller okay so that you can understand from where it is coming okay I have given a heading inside it let's do it from this so now look here if I run my application now okay so brother look here in program dot key here I have not even told the story yet that the home controller should be executed and this index area method of home controller should run because look if this index area method will run then this view will return means which view will be returned this view will return but now look if I run my

00:12:24 If I run the application and show you, then right now our view will not be visible to us, so look here, what do you still see here, Hello World is visible, but brother, now I don't want this. Now what I want here is the action method of my home controller should happen tomorrow. So for that, what do we have to do first, first of all we have to comment out this line. So, by pressing control C, I will comment out this line here. And brother, now look, our application is getting built here on this line.

00:12:53 Okay, before this I need to register a service here, inside this builder, how do I look here, here I say builder, okay, and after builder, look here dot, and then here I say services, and after services, look here dot, and here I say look at this, add controllers with views, okay, look, you can see this, here you have to register it like this, that means, until you register it like this, you will not be able to do anything about your routing for controllers and views.

00:13:23 You will not be able to implement that. So, you have to add this service from here. You have to restart. After that, what you have to do here is build your application. So, first of all, you have to add this service here and after that, build your application here. So, you have to add this line at the end. After that, what do you have to do now? Now look here, this is up because, look, where did the reference of whatever application we have built go? And now look here, I have added a middleware.

00:13:52 I add it here and what is that middleware, look here, I say, look here, you must be seeing Map Default Controller Route, okay, I add it here, okay, then let's do it and here it will be completed here, okay, so look here, what am I doing, which middleware have I added, Map Default Controller Route, if you take the mouse over it, then look here, look below here, it is telling you which controller will be used by default, the home one, and look inside home, it is telling you which action

00:14:19 The index will be there tomorrow, that means look, here you are seeing the same convention that I showed you a little earlier, that means when you call this middleware here, then what will happen is that by default the index field method of the home controller will be there tomorrow. So it means that you do not need to write the entire convention here, just what you do here is call this method like this and your home controller and index will be there tomorrow. So brother, look here, I run my application and its result

00:14:49 Let me show you, so look here, now the result is being shown to you in the correct way, meaning now our same controller and the same action method and view is getting displayed here, okay, but brother, look here, if you are not able to do that because brother, look here, this middleware will only and only do the home controller and the index area methodology of the home controller, but brother, for example, if I don't want to do the home controller for the first time, for example, I have to do any other controller or any other action method, then for that I

00:15:19 Here, I will not do this middle ware, what will I do for that here, see, I will comment out this also from here and look here, now I say up dot, meaning now I am going to define my convention here and through that I am going to tell which controller and which action method will be there tomorrow, so look here, I say up dot and look here, I will do that here, Map Controller Route, okay, look here you can see the Map Controller Route, okay, coming here, I will add it like this

00:15:45 And look inside this we will pass two parameters here okay the first parameter will be name here means I am setting the name of my route first and I pass it to you in the name default okay you can name it anything I am naming it default because this will be the default route okay after that common and then here I say pattern pattern means the same conventions which I showed you when I created my asp.net application with MC template so the conventions which I showed you there are the same

00:16:14 What do I do here in the convention, I get it displayed. Okay, how do I set it, look here, first of all the curly bracket will come and after that, look here, I say controller, after controller, I am saying which control by default and tomorrow it will be home. Okay, after that, look here, after its live bracket, I put a slash here because, look, I am making a pattern of the URL, means I am setting it here in the convention of the URL, so here I am telling that brother, the home controller will come after that.

00:16:39 The slash will come and then after that here, I again turn on the black bracket here and the action here and which action will be by default, here the index will be the black bracket and look here, its index has high capital and look here, home has yes capital, okay and after that I told you that we also have ID and what is ID, it is optional, so look here, here I fort slash and after that again curly bracket and inside this I pass the ID, but I am making the ID optional and to make it optional, I am passing here

00:17:07 What should I keep this question mark because brother look what I told you here that we have a controller and action method, I want it mandatory in the URL, that means it's compulsory but the ID is optional because whether we will pass the ID inside some other URL or we will not pass it inside any other URL, that's why I have kept it optional, that means you have to provide both these things mandatory in every URL, okay so look here, I have defined its convention here, okay so now if I do this then you will see that we still have it in the correct way

00:17:36 Our application will be executed. So if I run this here, you can see the result. Look here, now your application is executing perfectly fine. But look, when you define your convention here like this, then you can also do its value chain here. But for example, if I say here that I want to create another controller, look here, I right click on the controls folder, go to add, and add another controller here.

00:18:02 Let me do that and take the MBC controller from here and here you want to name this controller as User Controller, what should I name it as User Controller and look, this one that is there in me, I will remove it, okay, what am I naming it, User Controller brother, see, it happens like this, the controller of our website is different and the controller of the dashboard is different, okay, in the same way, all the pages of our website from index, about, contact, they will all go in the home controller, but for example, from the dashboard

00:18:29 If I need any related page, then I will create a separate controller for that and I will add all the pages inside it. So, for example, I am naming it User Controller. If you want, you can name it Dashboard Control. So, if I come here and add it, look here, here also by default we have an index area method added, but this is of the User Controller, so if I create a view of this as well. Look here, I right click and here I click on Add View and look from here, I am creating the Razor View, okay and

00:18:56 I will let its name remain Index here and click on Add. Okay, so look here, our view has been created and now I will call it Index View here. Okay, let's change it a little bit, Index View from User Controller, so that we can know which controller this view belongs to. Okay, but now look here, now what do I do here, this is my program dot key file, look here, now inside the controller, earlier I was home, I removed home from here and I pass it here to User. Okay, so this means now

00:19:25 Look here, this index field method of the user will be there. So, if I come here and run my application, then you can see its result. Look, what is the message that you are seeing, index view from user control. It means that whatever I had defined here inside my convention routing, according to that, it is getting executed for us. And brother, look here, if I home it again, okay, I home it again and now I say that brother, I want

00:19:52 I don't want to create the index area method of the home controller, but look here, what I do is I create another action method here and look here, I copy this from here, okay, I copy this here, pass it and change its name here and keep it About and generate a view of About, okay, right click on View, okay, click on Add View and look from here, I am creating a Razor View and I will keep its name here as well, click on About Add, okay

00:20:19 So look here, the about view has been created and what is written inside it, simple about is written, okay so look here, now if I change it here, okay look inside the action earlier there was index, now what do I give here, now out, okay, I have given about here, now let's do it and coming here I run it, then you can see its result, okay so look here, what result is being shown, the about view is being shown here, that means look here, if I make some changes in the URL, okay look here, I will change it here

00:20:43 I tell me brother, what do you want from here, home, you want the index, so look here, how have I generated the URL slash home slash index, so look here, as soon as I enter, the index view of the home controller will be displayed to us, similarly if I come here and say that brother I want the index of the user controller, okay, so look from here, I am changing the URL and as I enter here, then look here now what message is coming, index view from user controller, okay, means here you should understand

00:21:09 I might be thinking that according to the convention that I had set, the result is being shown here, but look here, I have not passed the ID here yet because the ID was optional, if I want I can give the ID, if I don't want then I cannot give the ID, okay, so now see, what do I do to work with the ID, here I create an action method, look here, I copy this one from here, okay and come down and paste it here and here I want to name it, but the example details, okay, this area method

00:21:37 I am naming it details and its return type is, from here, I will remove the direction result and here I will say wait and look here, I pass a parameter inside it and I will say integer id. If I return this id here, okay, you should understand the concept of what will happen here, means whatever ID I send in the URL, that ID will come and be stored here and the same ID will happen to us here, it will be returned, it will be displayed. So look, this is the action method of details.

00:22:05 Where have I created it? I have created it in my home control, meaning how will the URL be created, home and then details, and after that whatever ID I pass here, it will be stored here and then it will come here and get my turn. So now if I run it, you can see its result. So look here, what is being shown to us now, about home, because that is what I had set inside my convention. Look, now I say here that brother, which action method of home do I want to do here, home and then details, okay, of details.

00:22:34 The capital, okay, here how do you set it above the lower, okay and after that look, I pass the ID here, but for example, I mean what will happen now is that the request will go to the home controller and in the home controller, to the details action method and where will this go and get stored, look here, this you, I have created an ID parameter in details, will go inside it and get stored and that will be returned to us here, okay, so look here, now if I enter this here, you will see the result.

00:23:01 So look, if the result is coming here, let me zoom in a little, look at the home details and what is here that we have and the same is getting returned here, meaning, whichever ID I pass here, if I give five and press enter, then five will be displayed here, but look, this is not compulsory, meaning even if I don't do it, it's okay, but for example, if I remove the details here and give the about here, okay, so look, there is no ID in this case, so even if I enter it, I will still get the result here, okay, so this is the concept

00:23:29 What we have is the ID parameter which is the pattern of our URL, we set it here like this, okay see, you can see that this is basically what we have is conventional routing, okay so now let's come to its theory here, which is okay, so look here, its point is telling that in conventional base routing, the route is determined based on the ventures, define these route templates which will map the incoming requests from this URL to the controllers and the action method, that is, look what is here

00:23:57 I am telling you what happens in convention based routing that brother, whatever is your route, whatever is its convention, you define that in your route template, in the route template, I have created the middleware here, this one which I created the middleware and inside that, I have defined my route template here, defined the convention and then what is happening through this is that whatever request is coming, whichever URL is being requested here, we are sending it to their appropriate controllers and action methods, okay

00:24:25 So it is less that in convention based outing, okay so if I explain this with a diagram, look here, here we have generated this kind of URL, locals, then port number, and then look controller, and then look index, so look here, it is telling that in the first segment, what I have written home, it is referring to the home controller, and look in the second segment, what I have written index, that index is referring to the action sixth of our index, and then what the index area method does is that the view

00:24:52 It returns, okay then here the A which is next, okay now look what is happening here, in this asp.net course 6 application, these conventions based out are defined within the program dot key class file, that means, where do we define our convention routes inside the asp.net course 6 application, we define them inside the program dot key file, in this, the dotted course embassy application, it is this controller action method that is going to handle this incoming request, if this is you, then it means, brother, look what are you doing

00:25:21 What is happening is that whatever request has come to us, whatever our URLs are generated here, who is basically handling that request? Even the action method inside the controller is handling it because brother, you can understand that controller is a container, the actual thing that is working inside it is basically doing our action method because that is what is returning our view. Okay, so this is an important point. Then look here, what is it doing, but for example, if they issue a request, then we perform / give the index URL, it will be done.

00:25:51 The index field is the method of the home controller class which is going to handle the request. This means that if we generate this URL, then basically what is happening is who is handling this URL? Our index action method inside the home controller is basically doing what it is doing, it is handling the CR. Okay, then look here, there is another diagram. Look in this diagram, in the first segment there is the controller, in the second segment there is the action method and in the third segment there is our id optional.

00:26:20 The parameter is there, okay, this is what I just showed you by doing that I had created a separate field method for the details and whatever the ID was, I was receiving it inside it here and I was getting it returned, okay, so we will do this also practically later when we will make a credit application and will work with AP, okay, so look here now in the bigger picture of this, see the first point, what is it doing, similarly if you are issuing a request for home details, then you are giving it the details action method of the Home Controller class.

00:26:46 While it is going to process the request, if this kind of URL is being generated, then basically who is handling that URL, the action method with details, means the role of action method is very important in the controller, okay see what it is doing here, here the parameter value, you will automatically map it to the id parameter of the details action method, means here inside the third segment, whatever you passed, it will automatically get mapped with the id parameter that we had created for our details action method.

00:27:18 Inside, that means automatically this will get connected with the ID parameter, that means whatever value you enter from here will get automatically stored in the ID parameter in the third segment. So this is the concept of conventional base routing, so here our lecture has been wound up. So if you liked my video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So if you have any confusion regarding this lecture, then please do subscribe to my channel and also press the bell icon.

00:27:45 Do ask me, Inshallah we will meet in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know, and using sp.net we are doing the concept of routing, so brother, in the previous lecture, I told you the first type of routing which was named Convention Based Routing, so today we are going to do the second type of routing whose name is Attribute Base Routing, so brother, if you have not seen our previous lecture in which we talked about the first type of routing, then first watch that lecture, that is, watch part number nine, after that come to this lecture because I have told you

00:00:34 I had told you that convention based routing is also a type of it and attribute based routing is also a type of it; it depends on the developer which routing he finds easier, so I would comment that you should also see our part number 8 in which we discussed routing and you should also see part number nine in which we discussed the first type and brother see, in the last lecture we had worked on the project, in this now I will show you by implementing attribute based routing, so when you understand this lecture

00:01:04 This will be explained to you in detail so what is the difference between convention based routing and attribute based routing, you will also understand it very well. So before starting the video, a small request so that you know that this is my program and this is my channel and if you are new on this channel then please subscribe to this channel and also press the bell icon because many more videos are going to come in this playlist. So this will be our part number 10 and now we start our lecture. [Music] What is attribute based routing, okay?

00:01:37 First of all, I will tell you its definition, then let's come to its practical. So here, look, what is its first point? This attribute base routing, route attribute is used to define the route, means, we have a route which we use to define our route. It means, what we do in attribute base routing, is that we define this route attribute on our controllers and its action methods, means, we apply this route attribute and this route attribute

00:02:09 With the help of this we define whatever our routes are. Okay, when we studied convention based routing, there was no such concept there, meaning there we were defining our routes by using a middleware inside our program.tk file, we were creating conventions, but here I will have to define my routes by using the route attribute. Okay, so I go to my video studio and first of all do its practical, then come to the rest of its theory. So look here, this is the same application which we use in our

00:02:37 In the previous lecture, we were doing, see what is its name, routing without MC, without BC means because in the project that I had created, I had not used the MBC template, so the folders of controllers and views that you can see, I had created them myself, I had created them manually, so in the last class, I had told you that look here, we had created some controllers, home controller, user controller, then similarly, inside the home controller, we had created the about view, we had created the index view, similarly the user controller

00:03:04 inside this also we had created an index view. So look here, inside convention routing we had done this thing. But you do not want to implement convention routing but rather I want to implement attributes based routing. So this is the code in which I had created this convention. So here I am commenting out this code control c and look, you will remember when I taught you convention based routing, then before writing this code, I had added a service here, that means I had registered a service. Look here.

00:03:33 You must be seeing add controller with views so when we do attribute based routing, this line is very important for that too so you should not remove this line from here okay and now I need a middleware here for attribute based routing okay so for that we will use a middleware here okay with you you will do middleware and its name is map controllers here okay you are seeing these map controllers, we have a middleware, means what will we do with the help of this middleware is attribute based routing

00:04:01 We will implement it because if you don't use this middle ware then you won't be able to implement attribute routing. That means first of all this line is important, you have to add this line and after that you have to add this line here. That's it. And then what I told you that we have to use the route attribute in attribute base routing, so where will we put that route attribute, here, which is in our home controller, so look here, in the last class here, we had the action method of index. Okay, about the can method, we had created it and details.

00:04:31 I had created a method in which, look here, the ID was being accepted, so what I do here is I implement the attribute base routing, so what did I tell you that we have to do that with the route attribute, so brother, first of all, we have the index region method, on top of this, how do I add the root attribute, look here, first the square bracket and after that you will write root here and then what you will do here, you will open the parenthesis and inside this you will define your route, so brother, first of all I will define an empty route here

00:05:00 I am thinking, which is an empty route which gets executed for the first time, means whenever our application runs, then what do we have, empty route means the URL is empty, there is nothing in it, so what should we have in that, this index area method should be executed with us, so means when I applied this route on it, then it means what I am saying is that brother, whenever our URL is empty, then this action method will run in that route and when the index area method will run, then this index

00:05:29 It will return the same view. Which view will it return? Look, I will show you here. Look, inside this I have written Index View from Home Controller. Okay. And brother, look here, I have put the route attribute here. Now it is not necessary that you put a single attribute here. You can also duplicate the route attribute here, but with different parameters. Look here, I am saying that this induction will work for empty as well. And then I am saying here that if we have this home type in the URL.

00:05:55 It means home, if the controller is typed as home controller, then this action method should run in that case also and then look, I am saying here that if in the URL we have home slash index typed like this also then in that case also we should have this action method. So it means that if these three types of activities will be performed in the URL, then we will have this index area method running. So, for the first time when I run it, then this first route will be executed.

00:06:24 So in this case we will have this index field method, okay, so let's do that and here I run my application, so look here, now our URL is exactly like this, that means only the local support number at the time, so look in that case our index is also running, okay, now look here, if I do the second scenario, that means here if I just give the name of my controller, okay slash home, then come here and enter, then look in that case also we have the same index view showing and if you look here, the third one which I have defined the route

00:06:51 In which I had put homes/index so if I enter this URL here then look here, how can I also have the same index view, you can see so this is the concept of basically what you have the root attribute, okay, that means look here I have defined the root attribute like this and then look here, here we not only have the index area method, there is also about and details, so what do I do for that here, look I will copy this and put it below

00:07:18 Look here, above the about, I paste it here like this and now look from here, I have deleted the index and I pass about here, okay, again in the same way below and here, above these details, I paste this route attribute here and look here, I pass it details, okay, but now you notice one thing here that the details that we have here, it accepts an ID parameter, okay, so for that, how will I define my route here, here

00:07:45 Look, I will apply force less and after that I will define curly brackets here, okay and inside this, what do I pass here, the ID. It means whenever I give the ID after my home slash details, it will come and be stored here and from here it will come and be stored in this parameter and then what will I make this parameter return here, okay so now let's do it, I will run it and show it to you, so look here, look here, what is our URL now, so the index view is being shown on this.

00:08:11 Okay, so now look here, now I am here slash slash about, okay slash home slash about, as I enter here, then look here, the about view is getting displayed, about was written inside it, okay, then look here, now I say details, okay, what do I pass here, details, but brother, look, what did I tell you, along with the details, we also have to pass an ID, but now if you look here, if I do not pass the ID and enter, then look here, what message is coming, dislocated page

00:08:38 ca n't be found okay but if I give any ID here, done, okay but for example, I give you here and as I come here and enter, okay, then look here, the result is being shown to you, that means whatever I had passed, it went into the id and got stored in the id and brother, it is getting returned here, but brother, look, I had told you that the id parameter, we can make it optional as well, how can we make it optional, look here, okay, coming here, I stop it and I had told you what you put before the id to make it optional

00:09:04 Put a question mark here. Okay, so brother, look what will happen in this case, I mean, if I just write this here, home slash details, I mean, if I don't pass the ID, then what will happen in this case, brother, look here, what will happen in this case, that the default value of wait, what is zero, so zero will be returned to us here, I mean, if I just write this URL, homes slash details, and I don't pass the ID because the ID here is optional, so if I don't pass the ID, then this parameter here, the ID

00:09:33 Inside this, the default value of the ledger will go to zero and zero will be returned to us here. So look, I run A here and show the result. So come here, here I say home. Okay, after home I say details here. And look here, if I am not passing this ID. Okay, simple what am I writing homes slash details, if you enter it here, then look what is the default value. Here it is zero. Okay, I understood, but brother, look here, we have another concept that this

00:10:00 The parameter that we have is the id, okay. Now we can make this as well a level or a label, look here, I am putting a question mark in front of this teacher, it means that if I do not pass any id here, then what will go here, the id parameter will go null and look here, now I am putting a condition here, okay, look here, I am saying that brother, if this id is null, then what do you do with it, for example, you display one here, look, what I did here, this double question

00:10:28 Mark, what do we call this? Null operator. What I am saying here is that, brother, if we start receiving any ID here, then return that ID, but if the ID here is null, then what should happen to it? The value that I have written on the right side of the null operator should be displayed. So, this is the concept of null operator. Meaning, if there is a value here, means if there is a value here in the ID, then that value will be printed, but if the ID is null, then what?

00:10:57 One will be printed here, okay, so let's do that, I will show you the result, but look here, it has already been run and now look here, I say home and after that there is slash details here, okay, so look here, I am not giving the ID and I enter it, so look at this, okay, I did not pass the ID here, it means null went into the id parameter and when null went in, then look here, what is the null operator displaying here, one, okay, which I had defined the value of null with the black software, but look, if I enter any null here

00:11:25 I will enter the value, but for example, if I say 5 here, then now the id parameter will not go null instead some value will go inside it and the same value will be returned to us here. OK, look, if I come here and press enter, then see, the value 5 that I had sent, brother, is being returned to us here. OK, so if you want, you can also do here in this way, you can create a level, okay, so what can you do in this way also, you can define the route, so what do you also have, this is attribute routing, okay then here

00:11:52 Let me tell you one more thing, look brother, look here, I have defined the URL and this route, this URL, I have placed it on top of the index area method, so brother, look here, when you define the route here like this, then it does not matter what the name of the action method is, that means, if I look here for example, this index area method that we have, I name it data and look, I change the name of this controller here as well, I give it here for example, hello

00:12:21 Controller but brother look here, what I am doing here on the route, homeslash index, so if the URL is of this type, then this action method will get executed in it because the action method doesn't make any difference here, but brother look, when this data based action method will be executed, then it will find the data based view only, meaning, when this data based action method will be executed here, then it will look for the data dot csssdml, will find it and will try to execute it, but look here, data dot csssdml

00:12:54 We do not have any page by the name of, so what will we do here, how will we do it, look here, I will define the text of index.html myself , how will I look here, first of all, I will give the sesame symbol here, see, the sesame symbol refers to what we have, it refers to our application, which is our root, and then look here inside our application, there is a views folder, inside that there is a home folder and then look here, there is index.html , so here I will say, brother, look, this is our root, okay.

00:13:24 After root, there is a views folder inside it. Then inside that, there is a home folder and then inside that, we have this index.cssdml file. So, if you look here, the focus will be only on our URL. That means, if this URL is typed, then this method will be executed even if its name is different. So, when this one is executed, then here we will have this view returned. So, let's do it with this and I will show you the result here. So, look here, in exactly the right way.

00:13:54 The result is this, okay and if I write it here like this, homeslash index, okay and I come here and enter it, then you can see its result, look here, now our view is executing absolutely correctly and this is happening because look here, this route which you define above this action method, it focuses only on this route, it focuses on the URL, it does not focus on the name of the action method, then its name can be anything, the name of your controller can also be anything, okay

00:14:21 So this is one advantage of it. So, I set the control and then set it back. So, I remove the position of the view that I had given. And the name of the home controller that we had, I set that back as well. And what do I do back to the data as well? I index it. So, the next thing here, I'll tell you brother, look here, this route attribute that we have is not only applied on the action method, it can be applied on the controller as well. That means, look here, here I have put it like this.

00:14:48 I can also place it there. Okay, look here, I copy this home route from here and my class, Home Controller, what do I do here on top like this, I make it face, so it means, now look here, this route that we have placed on our home controller and this route that is placed on the action method, these will execute together because brother, look here, I have to type home again and again. Okay, look, I have to type home here as well. Okay, I have to type home here as well.

00:15:15 There is home with about as well. Okay, now there is home with details as well, but what I did, I picked up this home and placed it on top of my controller, so what is the benefit of this? Remove this home from here as well. Okay, that means I removed the route that I had placed on top of the action method. And look from here, the home slash that is there, I will remove it from here. Okay, and this home that was here with details, I will also remove it from here. That means that which was getting duplicated again and again, that too is gone here.

00:15:47 But the result will come to you in a completely correct way. Look, I run this. We run our application and coming here I say home slash index. Okay, let's enter it. So look, as soon as I wrote home slash index here, look, our index view is visible absolutely correctly. But if you notice one thing, what I did here was that we had home with index. I took it up. I placed it above my controller. So now what will happen is that this will increase

00:16:13 Also, you have this route that you have placed on the action method, what will this be, basically, the route that is placed on our home controller, it should get contacted with it and merge with it and its result will be displayed to us, that means if you are accessing the index here, then what will be the text, homeslash index, because both of these are getting displayed by merging together, okay, I am understanding, in this way we can also define this route attribute here on our controller, okay, but look here, now there is a spice here, that spice

00:16:44 What is it here, look here, if I empty this URL completely, it means look here, I remove the home as well and after this I remove the slash index as well and if I come here and write an empty URL and enter it here, then see, it is getting cached, this local page can not be found, it means the result is not showing here right now, what is the reason for that, look here, if you see here, the first route that I have defined here is home, so it means home should be mandatory here, but brother, when our

00:17:12 When the application runs for the first time, then this route here is empty, so how do I do that, look, this action method that we have here is not getting executed, so what do you need to do here to resolve this thing, look, add another route here on top of it, okay and inside this, you have this oil symbol and after the oil symbol, this forward slash, means, look, this is the route that we have, isn't it the route of our application, that is, when our application runs for the first time.

00:17:39 So its URL is empty, so you can understand that it is representing our root URL, that is an empty URL, because in this case, look here, this one is not reducing, so that's why I have placed it here like this, okay, so now let's do it with this and if I run it again, then you will see its result, okay, so look here our URL is completely empty, but still you can see the result is being displayed to us correctly, okay, so what can you do here in this way also, you can reduce it, okay, good then

00:18:06 Let me tell you one more thing brother, look here, here I am defining the route, here I am giving the name of my controller and look here, here I am giving the index area method, means what I am doing here also, I am giving the name of my action method, in this way here also I am naming it, we have the concept of token here, means these are the names of the controller and these are the names of the action method, we can also pass them in place of these, so what is it brother, look here, this is home, right?

00:18:34 Right here I define a token in its place and what is the token, look here controller, I have written it in square brackets, so what is this, controller's token, means the name of the controller here can be anything, means you can give any name here, but who will represent it, we have a controller's token, similarly here this index is, I am deleting it, what do I set here instead, token for action, okay, I am understanding and now look here, now this action, ticket, okay, I am here

00:19:05 I pass it over every action method because all we have is action method. I copy it from here and also pass it in the details here like this. It means the advantage of this stroken is that you don't need to give it a name. I mean the route here, you don't need to name it. I mean understand that this has become a kind of place holder. Meaning, whatever will be the name of your controller here, it will get replaced by the name. Similarly, see here if its name is index or data or anything.

00:19:33 So that name will be replaced with stockin, okay, in the same way see here everywhere, I have specified it like this here, okay, so now let me do this, I will see its result, no one will be shown in the result, okay, let's run it, okay, so look here, there will be no zero in the result, if I also put home here, okay home slash, I also put index here, come here and enter, see, no one is showing, if I change the action method here, okay, I will do it here

00:19:59 I will give you about it and if we come here and enter it, then look here, there is no issue here as well, we are getting the result in a perfect way, you can see, that means you can also specify the token here, you can see, okay, so one thing you have to remember here that whatever will be the name of your action method, it will get replaced with this token, similarly whatever will be the name of the controller, it will get replaced with this token, that means you don't even need to give the name specifically here, okay, now see a scene, look here.

00:20:25 This action, I am giving it here again and again. Okay, this action token, I am giving it above the index as well, I am giving it above the about as well and I am also giving it above the details here, so here I can do one thing, okay, this action token that we have, I copy it like this and I have put it on my controller, look here, I specify it like this, that means controller and after that its action method, because see, whenever we make a call, whenever we enter the URL

00:20:50 If we make it, mostly it happens like this, so this action that I was giving again and again, now I don't need to give it again and again, okay, look from here, look, I remove the action token from here as well, okay, and then below, I remove it from here as well, above about, okay, let's remove it, and below, the details that are there, I remove it from here as well, okay, look at this action token, I have removed it from today, okay, so now let's run it and I will show you its result, the result will be shown to us in a completely correct way, okay

00:21:15 Let's run it. So look here, here I give 'key' slash home and here I give the index. So look, even now the result is being displayed correctly because look in the URL, there is a controller and what is the action here, so look in that, we are reducing the token and it is happening correctly. Okay, that means even if you go to about here, even then you will see the data will be displayed correctly here, but brother, look, now there is a masala here, okay, that masala.

00:21:40 What is it, look here, here I have specified two things inside the route, that is, above the controller class, controller and action, but if I specify only the controller here, that means look here, I give the slash home but I am not giving the name of the action method further, so if I come here and press enter, then look, this local space card can be found, that means the issue is happening, that means when I am giving only the name of the controller, then the issue is happening here, so what do you do in this, I come here and stop it and then look what I do here

00:22:08 I do the same thing, I had written the root URL here, so I am saying that brother, if we type home here, okay, in this way, if it is typed here, then how do I call this action method, okay, so if I run it here, then what result do you see, look here, the result is coming, okay and now if I come here and just look here, I am giving the controller, I am seeing, now I am not giving the action method here, I just write the controller and enter it here, so see how I also do that in our

00:22:33 Here, its view is being returned and this one is giving us which route, which I have defined here, which one is the last, look tile slash home, okay, so what can you do in this way, you can implement your routing by using attribute base routing, okay, now let's go over the rest of its theory, okay, so now look here, you must have understood the first point, look at the second point, in attribute base routing, this route is determined based on the attributes which are configured here at the controller level and at the action method level.

00:23:04 What I am saying here is that the attribute based routing, the routes that we have are defined and determined where they are located, either we have them above the controller level, which means you can place routes above the controller as well and you can also place routes above the action method, which means by placing these routes at two places, you can define what you can do with your routes, that's okay then look here at the third point, we can do both convention based writing and attribute based routing in a single application, brother, look, this is too much

00:23:33 Important point, this is often asked in interviews also to confuse students, what is being said here that we can do both convention based routing and attribute based routing within a single application, how can we do it, brother, look here, look here, I have gone inside the program dot file, look here, what did I tell you that we have middle ware for attribute based routing and look, we have this, which I had commented, I will comment on it for control.

00:23:59 Control you, we had reduced this for convention based routing, so brother, look, it can often happen that there is a route for which I do not have attribute based routing, then what will happen in that, this convention based routing that we have will be checked, okay, it means if we have an action method, or a controller for which I have not installed attribute based routing, then what will it do in that, it will give priority to our convention routing, it will check it, so that's why it is being said here that we have this

00:24:28 Both are allowed to be present within a single application, then what happens next is now look here, if changing the controller and action names does not require the route method to be changed, what I am saying here is that if you change the name of your controller or the name of the action method, then it does not make any difference to the route that you have defined, like this, look here, what I did, look here, this is my home controller, inside it, look here, this index area method, I named it

00:24:56 What did I place instead of index, data, and instead of my home controller here, I named it Hello Controller, but still the result was getting displayed correctly, so this point is telling us that by changing the name of the controller and action method, there is no effect on the route that you have defined. Then see here, what is the controller doing on the token, the token that we had put for the controller was like this, controller was written in small font inside the brackets.

00:25:23 Similarly for the action method, because what I did with it, action was written inside the brackets, okay then look here, the last point, we can use both the tokens together, which means we can do both the tokens together, like this, where did I place the controller slash action, look here, I placed it above this controller of ours, I placed it here like this, so you can do it here also like this, means you can place these tokens together on top of your home controller, that is, on top of the class controller, okay, so you should understand these points here.

00:25:53 So here our lecture is wound up. So if you liked my video, then do subscribe to my channel and also press the bell icon as many more videos are going to come in this lecture series. So if you have any confusion regarding this video, then do ask me in the comment box. So keep me in your prayers. Insha Allah, see you in the next lecture. Till then keep watching my channel program [Music]

00:00:01 Hello viewers how are you all so as you know we are having a lecture series of asp.net 6 so in today's lecture we are going to talk about we have five methods for routing here whose name is map map get map post map port and map delete so we use all these methods to implement our routing so brother before starting this video I would like to recommend one thing to you that if till now you have not seen our previous lectures on routing in which we have discussed

00:00:30 I talked about types of routing, I talked about attribute routing, I talked about conventional routing, so if you haven't seen our previous lectures, then watch them first, after that if you come to this lecture, then you will understand it better, so before starting the video, a small request, if you are new on my channel and you have not subscribed to my channel yet, then I request you from my heart that please subscribe to this channel and along with that press the bell icon as well because many more videos are going to come in this lecture series.

00:00:56 So this will be our part number 11 and now we start our lecture [Music] So look here, here we have methods, okay, inside this, if you look at these methods, inside which there is map get, map post, MPT and map delete, brother, look, what you are seeing, get post, put delete, what are basically HTTP methods because brother, you know, whenever we work with routing, then we also have these HTTP methods, which are get, post, put, delete and

00:01:34 Brother, if you have ever made a credit application by using AP, then credit application means the concepts that we use here for the operations of insert, update, delete and read, the same concepts are basically implemented in these methods as well, that means these methods are provided to us for routing and if we have to work on the application by using these methods, then we can do that also successfully, right now I am not talking about AP, okay, here we will talk about the methods with reference to routing, but once you understand them here, then

00:02:08 Further answer, if you read the concept of UPS then you will understand it better because brother, look, I had told you one thing about routing in my previous lecture, what was that thing, I will repeat it here, look, I had told you that our routing is divided into two concepts, and what is that, first of all we have the URL and along with it we have some HTTP method, like brother, if you have made an application, then along with that we have some HTTP methods performed.

00:02:36 Similarly, if we want to search data from our table or database then we use get request there. If I want to insert or create some data then for that we use post method. Similarly, if I want to update something then we use put method or patch method and then if I want to delete a record then for that we use delete method. So, look, whenever you specify a URL, whenever you access the URL, then along with that we also have to tell the HTTP method that which one is this.

00:03:06 It will reduce your URL for the method, as by default you know that there is a get method, which means, whichever URL you type, by default, it reduces to the get HTTP method for a get request, but if you want to reduce with the other methods, then you have to specify that yourself, so what did I tell you that brother, whenever you have to write a URL, there is some HTTP method with it, as there is a get method, there is post, we have put, there is delete, okay, so with the URL

00:03:35 No, which HTTP method will be performed, this is what we implement with the help of routing. Okay, so this point should be remembered, so look here, for example, look here, this is a URL, local last, there is a port number, the port number changes and then look slash home, then look here, there is another URL like this, you must be seeing it below, look here, for example, it is written update and after that there is a route parameter, meaning there is an ID, because you know that we can use the ID while updating or deleting, so

00:04:03 I mean, look here, this first URL which has slash home in it, so we can do this with get request as well, which means we can do this with get method as well as post method, which means if you do this URL with get method then its output will be different and if you do this URL with post method then you will get a different output. Okay, similarly, look here, you can see a URL which has route parameter, ID, so I can also do this by mapping it and

00:04:32 I can do this in the delete method also because you know whenever we update or delete a record, we do it with the help of its ID. So look, to work with these requests, we have this concept. Which concept is this? I mean, we have these five methods. What do we do with these requests? We work on your routing. So now I come to my Visual Studio and will show you this method practically one by one. So, okay?

00:04:57 A here which is okay, so look, I am going to create a new project in Visual Studio, okay, and to understand this concept here, I am going to do an empty application in sp.net, from here I will select it, click next, and then look, here I have to name my project, the name of the project is, I have to keep it here, map method is okay, routing is okay, and remember I am doing this in an empty template, which means we will not find the MBC folders inside it, okay, then here A

00:05:22 Those who click on Next, okay, here I will do which dot net framework, I am doing dot net course 6 , okay, then come and click on Create, so look here, this project has been created, you must be seeing the map method routing, so now first of all I have run it inside its program dot file, okay, so look here, here our application is getting built through create builder and after that look here, we have a method for a route which is map get, and look here

00:05:46 What is it doing, brother, if the URL is like this, means for the first time when you run your application, then look here, what result will you get, here you will get the result hello world, so what is this basically, in the language of JavaScript, we can use this as a function, but in the language of dot net, what do we have, Delhi Gate, what is it doing here, it is returning a string, okay, so look, if I run this, I will show you the result, so look here, what result are you getting, hello here.

00:06:10 World okay and from where is it coming, it is coming from here, whatever I showed you, let us stop it, look, this line which is written, it is coming from here, but brother see, before map get, we have another method whose name is just map and look here now I do one more thing, look here the URL was just slash but now I want to write a URL here home okay so what I am doing is when this URL is typed then this function should run with us, okay means tomorrow the back function will run, delicate will run okay

00:06:37 But look here, now here I am only using the map method, so let me tell you one thing, for the map method, the map method works with every request, meaning whether it is a get request, post, put or delete, in all of these, we will have this map method, meaning it will only check the URL here, not the HTTP method, any method, be it get, post or delete, in that method, we will have this result which will be executed and what did I tell you, by default, it saves our battery, okay, so

00:07:08 Look here, if I run my application here, then look here, what message is coming here, this local space can be found, why is it coming like this because look what is there in the URL, there is only a forward slash in the URL and I had changed the URL of the route related to it because what did I keep the URL up, look here, slash home, okay, this is the URL, so look now if I come here and enter it, then look what result we are getting, hello world, but brother, look this is by default.

00:07:33 It is reducing for the get request, but what did I tell you, right now I am doing the map method, so it means that here I can do any method, port, pose, delete, for all of this, this URL that we have, it will reduce, okay, so for testing this thing, I will use a tool whose name is Postman, so brother Postman, you all must have used it, what do we do for AP testing, we use Postman, but right now we are not reducing on AP, right now what we will do is simple, this is the URL

00:08:02 Not only will we test this URL, okay so I will copy this URL from here, okay our application is running, you must remember one thing that you should not stop your application and you will go directly to where, you will go directly here on your postman, okay so look here, from here I had created a new file, I went to file, went to new file and then this kind of interface came up, so look here, here you must be seeing a text box inside which what is written, enter request

00:08:27 You are here, so I pass this URL, okay, and see, here for the first time, which request, which STD method do I make, get, okay, so I select get, and as soon as I come here and click on send, that means I will send the request on the image URL, then you see the result here, see what message are you getting below, look here in the output window, hello world, but look here, whatever method I am using here, which is map, which is a general method, which will be reduced with every request, so look here, what am I doing

00:08:56 Here now I do a get request, that is instead of the get method, I use the post method here and what I do is I hit the URL, so look here, if I click on the send button again, then again we are getting the result, hello world. So here now if I do the post request here, select it, and send it again, then again we will get this result, see, it is visible and similarly if I delete it here, then I will select delete.

00:09:22 If I click on send, then here we will still get the result. So if you want to do any work for different requests, then for that you can use this map method here. So coming here, I stop it. So here you must have understood this map method. So now what do I do? I copy this line like this and I also comment it out. Because now I am going to do that here, what will I do, I will do map gate. Look, map gate. So look, as the name map gate clearly states that this is just a gate.

00:09:51 Whose URL will you map with the request, that means get this URL connected with this method of yours, that is, with this call back function, but only for the get request, okay, so look here, I run it and look here, before running it, I have also written in front of it, okay, this is the get method, so that it becomes easier for you to differentiate, okay, so coming here, I run it again, so you can see quickly, okay, so here I have to write slash home, okay, I have to write slash home, I enter it here, so look, hello world and

00:10:22 What is this, get is this and look, this is the URL which we have, I copied it, I will come here on my postman, okay, so now look here, now if I select get from here, okay, I will select get, I will click on send, then we will get the correct output, you see brother, the output is this which I just showed you, hello world and get is written in front of it, but now if I select post request here, okay, I will select post request here and you can see this same URL

00:10:47 Let's hit send, okay, let's click on the send button, look here, no output was found and now if you look here, see what message is being shown to you, 405 status code is being shown to you, which represents what, method not allowed, okay, so whatever information is there, it is being shown to you below here, means there is no post method with this URL, similarly, if I select the post and click on send, then here also the same error will come, you see, similarly, if I click on delete, okay.

00:11:15 And if I come here and click on send again, then you will still get the output here, as you can see, so it means that if you want to generate, access your URL for any specific method or send a request, then for that you can use different methods, okay, how do I stop it here, so see, now I have done it with map get, I will copy it from here and paste it below, okay because we have many methods and okay, look here, I will do it, post it, okay here.

00:11:41 I call this put and here I call this delete. Okay, look this specifically mostly we do this when we have to implement some logic for every URL and its method, then we use this kind of method because what it basically is, this is the middle verse method. Okay, so you have to remember it. Look here, here instead of map get, I do post, okay, and here I do put, okay, and in the last I delete the map here. Okay, so look here, here I have done

00:12:11 It is kept with the URL, okay so that you know that when I use any specific HTTP method, then we will get specific output as per that. So if I come here and run it, then you can see its result, okay, so look here, I have to write slash, okay, let's enter, okay, so see, now this is the result and now let's go to our postman here, okay, so look, what do we have here now, it is the same URL and here, if I click on it with get and click on send, then

00:12:39 Look here, what message will come, Hello World Gate and now if I come here and click on Post and now hit this URL, okay, click on Send, then see what message is coming now, Hello World Post, that means it is working only for pose, similarly here, if I click on Pat, click on Send button, then look here, Hello World Pat and similarly, if I click on Delete and now coming here, I click on Send, then see what result we have now, Hello

00:13:04 The world delete is done okay so it means whatever your logic is according to the URL and the HTTP method, you can do that here, whatever method we have here for that, you can do it here, okay, these four are specific and this map is general because what I told you, whenever you request a URL, it passes through the request pipeline, that means it passes through this file show of the program dot key, so whatever your URL is, it comes here and matches with any one of your routing methods, but brother

00:13:41 Here now you see that whatever URL I am hitting here, we have this cancel back function running related to that, so brother, if you need to implement multiple lines or multiple logic here on a single URL, then the better way for it is a little different here, meaning we will not do it like this because we do it this way only when we need to return a simple string here, then we can do it here, but if there is any other logic, if there is logic of multiple lines, then

00:14:09 The way of writing it, that is, this map get map post, the way of writing it is a little bit different. So to implement it, look here, what do I do in this code here? I comment out the control C of the control and look here, first of all, I activate my routing here, okay? And for that, I do the routing here, okay, this middleware, I do it here, okay? And after this, what I do here is I define my more points, okay, that means if I am doing that with n points,

00:14:37 Because and inside points also what we do by using these methods is we define our routing, so for that first of all you have to activate your routing here through this middleware and after that look here now I say up dot here and here I do that and points, okay what do I do that, look here and point here in this way I do it and look here it is showing you completely what is its syntax, you tap on the button here, okay look here and

00:15:05 Points is a parameter and look here, I band it here like this and here I define its body, okay like this, okay, coming here, I will create a space by pressing enter and and here I will go a semicolon, okay, so these and points, you understand what this is, we have a parameter, okay now I will do this from here, okay, how, look here, here I say and points, okay, then I do it and then here I do it, first of all this map gate, okay, map gate is here, you see

00:15:30 And first of all what will I define inside this, I will define my URL here, okay so here I will use slash home and after that here we will get request delegate, that means that function which was the yesterday back function, I will pass that here, look like this, I will pass it here and looking at the curly brackets, I will make its body here and look inside this first of all I will pass a parameter here, what am I naming it, contact, what is this contact, basically we have this HTTP

00:15:56 Contact is HTTP contact, which means that brother, if I hit the URL, all its information, whatever its method, all that thing will come and be stored here in contacts. Like, here I am trying to access home here, and which method am I using, I am using get method, so whatever its contact, that head of HTTP contacts, will come and be stored here. And now here I am saying that brother, in response to this contact, okay, what do you do in response to this context, you display a message to the user.

00:16:26 Get it done, okay, so for that I am doing the method here, write a and look here, inside it, I pass a message, this is the home page, what have I passed here, this is the home page and look here, this code that you can see is basically called request delegate, which means that on this URL, we have this delegate function, it will be executed and look, it works asynchronously, so here before this line I will put a wait, okay, and this contact, okay, before that I will put it here

00:16:56 In this scene, okay, this error is also coming with you for this reason. When you apply this, the error that you were getting will go away. So look here, what have I done, I defined my point like this. So, if I come here and run this code, you will get the result like this. But I have just slightly different the way of writing here, because now here you can write multiple lines inside this body and apart from the string, if you want to do any other thing, then you can do that here as well. So, let me run it.

00:17:23 Okay, so look here, inside the URL, I say slash home here and as soon as I come here and enter, then see what is the result, this is the home page and now I test this URL here on Postman, okay, so look here, from there you are, here what method will I do, get, okay, let's select get and now here I send the request, so look what message are we getting here, this is the home page, okay, I am understanding the concept, means if I come here and use any other method

00:17:49 I post it and you send the request for it. Okay, so now here you see, the error that was coming, the method not allowed for four zero five, will come here. Okay, I am understanding it. So in the same way, if I stop it here and look here, now I copy it from here. Okay, this code was there. And staying inside this point, okay, and I paste it here. Okay, I also control it. Okay, so look here, now what do I say, I post. Okay, here I say map post URL.

00:18:16 I have kept it here and look here, this map gate was okay, here I write gate and coming here I write post, okay so that it can be differentiated that this will work for gate and this will work for post, similarly I come down and paste it again and here I do it, pat, okay, pat means for whatever work we do for update etc., we do it here, okay and here I put pat and then it remains with us, so here I do it, map delete, okay

00:18:44 It's done and coming here also I write this and here delete it, it's okay, see the URLs can be different, the URLs you have can be completely different, I have just kept it here to understand the concept, okay, so now if I come here and run it, then you can see its result, okay, so I come here, I have to write slash home inside the URL, okay, let's enter it, okay, so see the result here is absolutely correct, let's go to Postman to test the rest of the method, so what have I written here, post, okay, so now if I

00:19:10 I hit the URL and click on send here, then you see what the status is here, 200 OK, which means the request is absolutely OK, and look here, what result are we getting, this is the home page and what is placed next to it, post. Similarly, if I write pat here and click on send, then pat will be written here and similarly, if I select delete here and click on send, then we will see delete written here, okay, so in this way it happens here and one more thing I want to tell you

00:19:35 Let me tell you what I told you, brother, these middlewares that we have, we use them when we have to put some specific logic on a specific URL. Otherwise, what did I tell you in the previous lecture? In the previous lecture, I told you about such routing which we used to do to some controller or the other, we used to do to some action method inside the controller, but this thing we do is that whenever we have to put our custom logic, then for that, we use these middlewares here.

00:20:05 And notice one thing here, inside this I have not even used the next method. Why do we do the next method? I had told you when we studied the concept of middles, I had told you that brother, if I do the next method here inside this function, look here, after the line with a weight, then this method will also run and the method after it will also run, but what is happening here now is that whatever I have done here, I have defined more points in it, in which we have at a time one Rawat here.

00:20:37 It is executing. So you have to remember this concept. Then one more thing I told you is that by default we have only slashes. That means when we run your application for the first time, by default there is only slashes. So in that case, we had an error here. So to resolve that error, look here, what do I do? Look here, I say up dot run. I do that here. Okay and inside this, look here, I create a parameter. Okay, and that parameter will be the HTTP contact.

00:21:05 Type means object of which class of HTTP Context, okay, right, and then look here, here I create this array like this and here its body is added, okay, and now here I am saying, I copy this line from here, okay, but for example, I am saying here, I paste it here, and here I pass a simple message here, but for example, page not found, okay, you can write anything, just to understand, what am I writing here, page not found, and look here, I have applied a weight.

00:21:28 So here, the parameter before this is the object, before this, what do I put here, a sync, so what does it mean here that when our application runs for the first time, then in the URL, we have only a slash, for that, we have this middleware executed here and one thing you have to remember brother, look, the home that I am giving here, flash home, it goes and appends with our local host, so look here, what have I put in the beginning of it, a forward slash, but if you want then

00:21:56 You can also remove the slash here, but for my convenience I have put this four slash here. So now look, if I run this here for the first time, then I will get a message like this, meaning page not found. Okay, if I run it and see it, look, you can see that here you have a simple forward slice, so this is the result here, but if I type home here, okay and the get request will pass here, so look here, what message is coming, this is home page get. So look here, here I

00:22:22 I am telling you its definition here, look here, I have done a and point, I am going to tell you a little bit about it, so look here, first of all the point d map get method is used to define an n point, means this was the map get method in which we were passing our URL and passing the request delegate, what does this method do, we use it to create an n point, now what is an n point, look here, what is it doing, an and point is something that can be selected by matching the d URL and

00:22:51 HTTP method means look here, what is it doing, what had we created inside this point, we had created a map get, we had created a map post, we had created a map delete, but what is it doing here, brother, there are multiple points inside this point, but the method that I provide with the URL and the method that I get done, according to this we have an end point because I had created more points, one was for map get, one was for post, one was for put, one was for delete, so what was the other point doing, it was matching.

00:23:22 Which are those two things, the URL and what is its HTTP method, according to that it was calling our endpoint, okay, that means I had created those important things, so it was calling it with the help of URL and SDP method and then see here, what is it doing, executed by running de delegate, means it used to execute the point to me, so what it used to do, basically it used to execute one of our delegates, it used to run and what did I tell you, whose delegate am I, look here, look this one portion, okay, this one portion to you

00:23:53 You can see what I have highlighted, this is basically what I am calling request delegate, so whatever your URL is and whatever method is, according to that what you have, this request delegate gets executed because look here, here I have created multiple request delegates, that means I have created call back functions, so our lecture is over here, think about it, so if you liked my video, then do subscribe to my channel and also press the bell icon because there are many more videos in this

00:24:19 This is going to be a part of a lecture series so if you have any confusion about this lecture then do ask me in the comment box. See you in the next lecture, Inshallah. Till then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you? Do you know we have a lecture series of 6 on sp.net going on, so in today's lecture, we are going to discuss a very important concept called controller. So brother, if you have not yet seen our previous lectures in which we discussed routing, then first watch those lectures and then come to this lecture. So before starting the video, a small request, if you are on my channel for the first time and you have not yet subscribed to my channel, then please subscribe to my channel.

00:00:29 Subscribe to the channel and also press the bell icon because there are many more videos coming in this lecture series so this will be our part number 12 and now we start our lecture [Music] You see, today's lecture is about Controller in HP.net, brother, see this is a very important concept because I told you that whatever application you make using asp.net, it is based on MBC, okay, that means MBC framework is the framework of Model View Controller, by using that we can make

00:01:03 If we build an application on our computer, then the most important concept of MC that we have is that we have the controller. So far we have not done the model, we have not done the views. Here we are looking at the first thing, the controller, because in MBC, the controller is called the backbone of MC, that is, the controller is the backbone, without it you cannot build an MBC application. So look, first of all, I am going to tell you a few things about it. After that we will also see its practical. So here

00:01:33 So brother, look here, I am trying to make you understand one thing through a diagram. Look, inside MBC, you know that we have a model, a view, a controller. So, look here, there is a diagram. Look, I had explained this diagram to you earlier also, when I told you about the model, so here is the user. User means the one who is accessing your website, so whenever he requests a URL, means whenever he enters a URL on his browser, then the request first goes to the controller, means this model.

00:02:00 There are views and controllers, inside them first of all your request from the user, who receives it, the controller receives it and after that the work of the controller is to get the data from the model, model means those classes which interact with the database, so what the controller does is it gets the data from the model, if the model is present then it's fine and then after getting the data from the model, it updates it in our view, view means the user interface on HTML and then that same view is visible to the user, okay this is done.

00:02:30 What happens in one scenario and the other is that when a user sends a request, I mean sends a URL, then first of all it goes to the controller, so the controller sees that if the model is not present, because where is the data coming from, it is coming from the model, but suppose we don't have the model right now, then what does the controller do, it directly calls our view, that means if there is no data, if the model is not present with us, then what does the controller do, it directly calls our view and that view is visible to our user, it means that there is less data here

00:03:01 The controller is doing it because the first thing that is receiving the request is done by the controller and the controller is doing the work of managing the model and the view, which means that the model and the view cannot directly communicate with each other, which means that data is sent from the control and the model and the controller itself will update the model's data in the view, which means that there is no direct communication between the view and the model, who gets their communication done, the controller gets it done, so you have to remember this one point, okay, you should remember this diagram very well.

00:03:32 Then here I am going to tell you through the next diagram, look what I told you, what is there inside MBC, there is a model, there is a view, and there is a controller. So look, what I just told you, that this controller that we have, it works like a traffic policeman between the view and the model. It means that brother, for example, if the view needs any data, then the request will go to whom first, it will go to the controller, and what will the controller do, it will insert the data from the model, and then after taking the data from the model, where is that controller?

00:04:01 It will transfer the data to our view, this means that there is no direct communication between these wave models that we have, who gets this communication done by our controller, brother, that is why the controller is called the backbone of MBC because if I remove the controller, then what will happen, our views and models will never be able to communicate with each other, okay, one thing is this and the second thing is that brother, whatever request has come from our view, who receives that request first, the controller receives it and then inside the controller itself we

00:04:34 We apply the logic, means the request that is coming from the view, how to handle that request, which code is to be executed, the controller gives it to us. OK, this means the view becomes our user interface. Inside the controller, we apply the logic in our program and the model, we have such classes that interact with the database. OK, so here you understand what a controller does; it provides communication between the view and the model. OK, so this point is again here, now look here.

00:05:02 I am trying to make you understand the flow of your asp.net kar MC application, what happens in this flow, brother, look, we have a browser, what happens on the client, a browser is installed, what do we do on the browser, we request any URL, that means we enter any URL, that URL is of any asp.net kar web application, so what happens, see here, whenever we request any URL on the browser, where does that request go, our asp.net kar

00:05:29 That application has a URL. So inside our asp.net application, the URL that we are requesting first goes to the routing mechanism. That means, whatever URL you are entering here, that means, whatever URL you are requesting on your browser, then first of all that request goes to the routing mechanism. That means, whatever routing mechanism we have defined inside our asp.net application, the routing mechanism first checks your URL.

00:05:58 It does that, okay and then if your URL is valid, then what does routing do? It sends the request to the controller. Okay and then inside the controller, there are action methods. So what does a controller do? What does it do with an action method? And when an action method is called, basically what happens is that we get a view returned. Okay, so this means what I told you in the previous figure that whenever a user makes a request, then first of all the request goes to the controller. But brother, look.

00:06:29 Before the controller, we have received the request from routing, so you have to remember this point very well because whatever our URL is, whatever you enter, who does the work of checking it, who does the work of testing it, our routing mechanism and what is routing mechanism, we have already discussed this in detail very well, so when the routing of your URL is checked, after that where does that request go, from the controller to the action method and the action method returns some view or the other to you, okay, so see, this is the flow, this is the flow

00:06:58 You have to keep this in mind because if this flow remains in your mind then it will be very easy for you to create an application. Okay then next, now look here, now I am giving you one more information about the controller brother, look here, what do we basically have in a controller? A controller is your class. Okay, that means there is a container which we call your class. Okay, that means don't consider controller as a different thing. What is a controller basically? A controller is your class. What do we do inside it? We define any method.

00:07:27 Let's do it in the same way, like index, about, details and there can be many more methods inside your controller class, so the methods that I create in my control class, these methods are basically called action methods, which means, index, about, details, what are these three, we have action methods because as I told you, whatever request has come to the controller, the controller performs some action method or the other according to that, it means, brother, for example, if I have entered some such URL in the URL that brother, I want to take the home page

00:07:57 If we want to access it, then this URL that we have, after being checked by routing, where does it go in the controller and what will the controller do, it will call whatever action method is related to it and the action method will run, so what did I tell you, some view will be returned to us, again similarly, if I access a URL like this that brother slash about, then the action method with about will be returned to me, okay, similarly, if I want to write details here, okay, inside my URL, then the action method with details that we have inside our controller, will return that to us

00:08:24 You will get it tomorrow. Okay, so what is this, what is a controller, it is a class and the methods inside it are basically called action method in the language of MBC like sp.net, okay, so in this lecture, I am telling you about controller, in the next lecture we will see action method, okay, so now let us come to some of its theoretical points, after that we will come to its practical, okay, here first of all the point controller manage, you are seeing it the flow of the application, I mean your which

00:08:51 Who manages the entire flow of the application? The controller manages it because as I told you, after the routing is checked, the first request that comes in goes where it goes to our controller. Okay then look here, a controller is used to define it and a group a set of actions, that means why do we create a controller because what do we do inside a controller, set of actions, actions means we define the action method inside which controller, because brother look what is a controller, we have only an empty controller

00:09:21 It is a container but the actual work that is being done inside it, is being done by you through your action method. So, you have to remember this point. Then, see, the third point controller is responsible for intercepting incoming requests and executing the appropriate application code. That means, it is being told here that brother, what is the controller responsible for, that whatever request has come from you, that is, whatever you enter, it accepts those requests, receives them and whatever appropriate code is related to it.

00:09:52 Whatever is the appropriate action method, the view executes it. So this reduces the header and gives you your controller. Then the next point, look here, what is the controller doing? This is the backbone of MBC. I had told you this point in the beginning also that if you remove the controller from within the MBC application, then what will happen to your entire MBC application? It will be ruined. Then the next point, now look here, what is the controller doing? Communication with the model of the application and select the request view and then render it.

00:10:22 But look at the request, what I told you was that whenever we send a request, whenever we request a URL, we see some view or the other in the output, some view or the other is displayed to the user, but the data that we have inside that view is basically coming from the model, did I tell you that the model is also one such class that interacts with the database, so it means what the controller does is it takes data from the model and then renders it inside the view and displays it to our user.

00:10:52 Okay, so this point is also very important. Then look at the next point, allow separating the business logic of the application from the presentation logic. So look here, what it is doing is that the controller separates your business logic and presentation logic. Separate means whatever is the presentation logic, it means whatever user interface work is there, where do we do it, we do it inside our views and where do we put our logic, business logic, inside our controller

00:11:19 If we put it inside then what is the same thing doing here, which controller is doing the separation because brother look, the pattern of MBC promotes the separation of concerns in the same way, means brother if you have a front end developer, he will work with the views, okay, you have a backend developer, he will work with the controllers and you have data based developers, they will work with the models, because they have to interact with the database, so that's why here it is being said that brother business logic and presentation logic.

00:11:47 is kept separate, okay then look here, the next point incoming requests are mapped to actions through routing, now look brother, I am telling you the thing here that brother whatever incoming request happens, I mean whichever URL you request, it is mapped and connected with an action method and this work is done through routing, we have done this work very well in our previous lectures, so if you have not seen the lectures on routing, then you should also see those lectures, okay, then go ahead

00:12:14 So now look here, controllers help in managing the complete flow of the application including accepting input and rendering appropriate output, meaning look here what it is saying is that controllers help you in managing the complete flow of your application, like this brother, look here it is saying accepting input, what does accepting input mean, brother, we create forms, what does the user do through the forms, he inserts some data, means where do we create the forms, we create them in the view, it means inside the form

00:12:42 Through whichever user sends his data, who receives that data, who handles it, your controller and then after the form is submitted, that means the controller receives the data of your form, after that which appropriate code, which appropriate output is to be displayed, that work is also done by your controller then look at the next point controllers are basically C sharp classes, look brother, the thing which I told earlier that what are controllers, from C your classes are data inherit from microsoft.aspi netco.nbc.controller this means brother look

00:13:16 Whatever custom controllers we will create, it will be a child class, it will be a child class, which controller of the parent class, okay and that controller-parent class, by which name space will you get it, look microsoft.asp.net dot mc, okay, so I will show you this practically, so you will understand it better, so I come to my video, okay, look here in Visual Studio, I will come here and create a new project, okay and let's come here, from here I will choose the application, okay, absolutely.

00:13:43 This is a scratch, I will explain things to you, let's select it, click on next, okay and look here, this is the name of my application, what do I give here, controllers and actions, okay, controllers and actions, okay, this is what I named my application and now I click on next and after that look here, we are doing the six version of dot net here, okay, so we have selected it, now we click on create, so look here, our application has been created, you must be seeing it, okay, inside this now

00:14:08 What I do first is I create a controller. So to create a controller, the first thing I need to do is add a folder named controllers to my project. So look, I'll right-click on my application. From here, we'll go to add. And after clicking on add, I'll take a new folder from here. And what will I name it? Look, you have to pay attention to the name. Okay, this is the name we keep. Basically, this is the convention. And if you were to create a project using the MBC template, then the folder that's already there would also be there.

00:14:34 There are folders created, its name is controllers, so now I add a controller inside it, okay, so right click on my controller, okay, go to add and here I am controller, okay, you must be noticing, you have to select the control from here and see here you have multiple varieties of controllers, you can see the MBC controller is empty, there is MBC controller with read write actions, then see what is being done here, MBC controller with views using NTT Framework, brother, see, till now we have not worked on NTT Framework

00:14:59 Aditya Framework: When will we use it? When we connect our application with the database through dot net, that means, when we will create a cred application, then we will create login sign up in future, then for that, when we work, we will use this MBC controller and then see here it is saying MBC controller with read write actions, it means that we have methods related to cred operation, there are action methods, in the way it has been created, updated, deleted, details have been done, if we need that, then for that we can do it here

00:15:28 You can use this controller, which means that inside it, all the action methods related to the card application are already readymade, but I will not take MBC controller here, which means it is completely empty, so I double click on it, double click and see here, MBC controller is selected here and look here below, I have to name it, so look here, by default its name is selected, Home Controller, so I will keep its name as it is, but you remembered one thing.

00:15:54 You have to keep it. Okay, that means whenever you are setting the name of the controller, you have to remember one thing that you always have to change its prefix. That means whatever name you want to give, you can give it here like dog, I have written home, instead of home, if I write here admin controller. Okay, then this admin is the name of our controller. But look here, this controller that we have, you should not delete it. That means whatever name you want to keep, you have to keep it at this place. Okay, this one which I have highlighted, that means I kept the name admin and controller.

00:16:24 It should be placed here by default, you should not remove it, okay, so you have to remember this, so here, if I make it home again here, then what will be the name of our controller, Home Control, and look, the dot key is placed, it means that we already have a class of sh, okay, so now if I click on add here, then look here our home controller has been created, and look, this home controller is basically our child class, and who is the parent of this child class, look here, this is our inheritance

00:16:51 It is the parent and look from where is this class coming, this class is coming from this name space, look, you are seeing it as Microsoft asp.netco.mbc, why do you see, if I remove this name space from here, okay, if I remove this name space, then look, we are getting an error below the parent class, it is coming because this parent class is defined in which name space, it is defined inside this name space, okay, so this can also become an interview question for you, so you have to remember this as well, so look here, this is the point

00:17:20 This last point was being told here, I told you that whatever our controllers are, whatever custom controllers we create, they are basically inheriting from a parent class, and which class is that controller, and by which name this controller class is coming from this space, from this name space, okay then here, the next point is now see here what is the controller doing along with its action method accept incoming browser request model information and provide suitable response, means see here what is it doing that

00:17:49 Controller along with its action method means the action methods inside the controller, what do these two do together, they accept your incoming request coming from the browser, that is, your URL, and then according to that, whatever model information is there, that is, the database information, they retrieve it, that is, they get it and whatever suitable response is there, they generate it and give it, that means this controller and its action methods are of great importance to us.

00:18:18 Let's give the name of our asp.net application. Okay then look here the next point it's recommended data class name of a controller ends with six controllers. Look brother, the thing that I just told you that brother, whatever name you are keeping for your controller class, you need to add controller in its name. Just like I told you that if you are naming your controller as admin controller, then in the beginning there will be admin, so along with admin, you need to add your second world controller which is called six. Okay, so this is a very important point and it is recommended.

00:18:51 Also, you keep its naming convention like this then look here, the next point controllers are located in the root level controllers folder, meaning what I am doing here is that where do you create all your controllers, inside the controllers folder, and where are the controllers located inside your root folder, which is the root folder, look here, this application that we have, controllers and actions, this is our root level, meaning it is the root folder, inside it we have created this controllers folder and whatever number of controllers I create, they will

00:19:19 Basically, inside this folder, I will create one, right now, I have created only one home controller, for example, if I go ahead and create a complete management system, then I can also create an admin controller for that, inside which I will create the admin dashboard work. So, this is an important point. Now, look here, what is it doing in this asp.net MBC application controller, is it responsible for you? I mean, for what things is the controller responsible inside the MBC application, the first point here is

00:19:47 Look, lock the appropriate method to call foreign incoming request, means what does the controller do, it calls the appropriate method and which method does it call related to which we have received a URL, as I told you, if we have local slash home in the URL, then it means we have to call the action method of index, similarly if slash about comes, then we have to call the action method of about, so the same point is telling here, then look here, valid date of data of incoming request before incoming requested method, means it is doing here

00:20:20 Brother, whatever your incoming request is, if any data is being sent with it, then data means, just like I told you, brother, we can pass any data inside the route parameter, we can also send data through our incoming request, like inside the URL, often you must have noticed that whenever we do any searching, like for example, I searched for asp.net tutorial, then the text that we have is passed in our URL, which means that our data is contrasted with the URL in our application.

00:20:47 So inside the application, who does the work of validating and checking the data that is coming with the incoming request, the controller does it and then after that, whatever method is the appropriate method, it executes it, then see the next point, retrieve the request data and passing it to the request it method, today arguments brother, see what it is doing here, brother, whatever data is being passed through the URL, but for example, I told you that we also pass an ID inside the route parameter in our URL, in this way

00:21:15 If I am updating or deleting any data, then we pass the ID in the URL, but for example if I am updating or deleting the data of 1 2 3 , then that ID is also passed to our action method, our controller also passes such an argument, it means if I send update slash one in the URL, that means send the ID, then what the controller does is that it transfers the ID that I have specified inside the route parameter, it passes that ID inside the arguments of the action method, so that means this

00:21:45 The controller also does important work, then the next point is now look here, what is it doing, handle other exceptions data of the request method, brother, look, we have action methods inside the controller, and you know that inside the method we also have the power to generate exceptions, so who does the work of handling those exceptions, the controller does it, means the action methods that are inside the controller, what will you do inside this, you will handle the exceptions that are getting generated inside your controller, okay, this is also an important point

00:22:14 Then see help in rendering the view based on the result of the requested method. It means the method that has been requested in the URL, the action method that has been requested, according to this, whose responsibility is it to render and display your view, it is the controller's responsibility. Okay, did you understand? So our lecture is over here. After the wind up controller, the next lecture will be our action method. Okay, and inside that, we will do the proper work with the controls and action method. So if you liked my video.

00:22:43 So do subscribe to my channel and also press welcome because many more videos are going to come in this lecture series, so if you have any confusion regarding this video, then do ask me in the comment box, Insha Allah, we will meet in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are having a lecture series on hp.net code 6 , so in today's lecture we are going to talk about action method. Look, in the previous lecture we talked about controllers, what are controllers, how to use them, how to create them, what is their purpose in our MBC application, so today we are going to talk about action method. Look, it is a very important concept because, without action method, you cannot do anything.

00:00:30 You can never create your own application by using speed.net and brother, before the action method, you should also know the concept of controllers very well, so I will comment that if you have not seen our previous lecture yet, then watch the previous lecture and after that come to this lecture, so before starting the video, a small request that if you are new on this channel, if you have come for the first time, then please do subscribe to my channel and along with it press the bell icon also because many more videos are going to come in this lecture series, so this

00:00:59 This will be our part number 13 and now we start our lecture [Music] so look here, action method, okay brother, look what is action method, I had given you an overview of it in the lecture on controllers, that is, in the previous lecture, so today we will discuss it in detail, okay, you will remember that in the previous lecture, I had explained to you one floor of our MBC application of dot net, okay, what did I tell you in that flow, look here, look here, what did I tell you that brother

00:01:31 We have a browser and what we do in the browser is we access any URL of our MBC application. It means if I want to access my asp.net kar ambishi application then what will I do, I will write the URL in the browser and when I enter that URL, where will the request go? The request will go to our ac.net kar application and inside our asp.net application, first of all our request, who checks our URL, it checks it by routing, after it is checked by routing, our request, which is the URL,

00:02:01 Where does it go to the controller and what have we defined inside the controller, action methods are defined, even then your URL is, whatever is your request, according to that some action method of the controller is called and what does the action method do, by default it returns a view to you, but brother, it is not necessary that the action method returns only a view to you, the action method can return many types of data to you, similarly the action method can also return a view, J7 can also return a partial view.

00:02:31 It can also return and there are many other things that can also return, okay so this is its normal flow, you have to remember this, I mean the flow of our ac.net kar mc application completes in this way, then next, look here, I told you what our controller is, it is an empty container, I mean what is a controller basically, I told you that it is a class inside which we create methods, like look here, the index is done, about is done, details are done, what is this, this is the method and in these

00:02:59 What do I call the method? Action method. And brother, you must remember one thing, look, we don't have anything in the controller, it is just an empty container, meaning it is an empty space, but whichever URL request we make, that request is received by some action method inside the controller, and when the action method is received, it means that the action method itself generates the response and gives it, so suppose brother, I want to write a URL here, and what is that URL, look here it is local slash home, so when I type this one

00:03:29 If I enter the URL then the request will first go to the routing and after being checked by the routing it will go to the controller and inside the controller we will have an action method which will get executed just like in the slash home key you know the index method will run again similarly if I call the about method here in the same way if I want to write about in the URL then the action method of about will run again similarly if I want to write details in my URL then we will have the action method of details which will run so when you call this action method

00:03:59 You have these action methods, so these can return a view as well, these can return JSON, these can return a parcel view as well and there are many other things, I will tell you about them now. So now look here, here I am going to tell you some points regarding action method, first of all look at the point, what is it saying, a controller class can contain one or more action methods also non-controller actions, means here it is saying that you can create one or more action methods inside a controller class, so that

00:04:27 The action methods are also known as controller actions. Okay, so this is one point. Now look here, the second point is actions and method in controller class which are responsible for returning view and GM data. That means, look what is being said here. What are the actions being called, what is that, simply the methods are inside our controller class which are responsible, that means, their work is that they either return the view or return J7. So, look here, right now, there are only two things.

00:04:55 So I am talking about it means the action method is talking about returning the view or returning the gestion data but other than this there are many other data types that the action method can return to you and then here action methods are public methods in controller classes, I mean look here what it is saying is that action methods are public methods, so this is also an important point that you have to remember, I mean we don't make the action method private or protected because whatever request is there, it is a result of the controller's action

00:05:26 It has come to method, so if I keep the action method private, then what response will it return, so what are action methods, we have public methods, okay, so this is one point to be remembered, so brother, before understanding the rest of the theory, let me come to our business studio, in the previous lecture, we had created an application which we named Controllers and Actions, so inside this application, now I will make you understand the concept of action method, okay, so here, look here, here you must be seeing Controllers and Actions

00:05:53 In our previous lecture we had created an application by the name of, inside which, look here, we had created a folder of controllers and inside that we had created a controller by the name of Home Controller, whose code, look here, you can see here, so brother, this is our Home Controller class and inside it, look, this which I have highlighted, I am calling it action method, so look, when you create a controller, then inside the controller, by default you get this action method because I told you that I have set the action method as empty.

00:06:20 If you have created a controller, then inside an empty controller, by default, you have an index field method. So now notice one thing here, look, what is the return type of this action method? I action result. And look, we have I in the beginning, so this is an indication of what this is, it is an interface. And look here, what is it returning? It is returning a view. Brother, look, if anyone has read the concept of SP.net MBC 5 before the speed.net course , then you will remember that there we have this interface.

00:06:50 It does not mean that there was no action result, rather we had an abstract class which is just an action result, so the class which does the action result here, the same class which does the action result, in the MBC 5 application we had only the action result, so brother, this means that the class which used to do the action result in MBC 5 , the same class will interface with the action result here, so brother, now I am telling you a very important point, you have to understand this carefully, brother, look, this interface of the action result has been created by many classes

00:07:25 It has been implemented, okay, I mean which classes have been implemented, but for example, I will tell you some classes here, like look here, there is View Result, okay, after View Result we have Parcel View Result, okay, and then here we have Json Result, okay, we have Json Result, right, and Acceptor, that means look here, all these classes, which interfaces have been implemented by these classes, which action result interface, okay, I mean, look, there are not just these three classes, there are many more

00:07:55 There are all the classes which have implemented action result. That is why look here, this index area method of ours, what is its return type? I action result. I action result means that brother, this index area method can return many types of data. That means it can also return a view. View means it can return view result. That means if you look here, if I take the mouse and go there, then look here, it is telling you what it returns. Look, create sub view result object, view result object, means our

00:08:26 Our view which is passed, returns it to us and then look here, it can also return any partial view as result, meaning you do not need to change the return type here, you can use the same return type in the region method and then this same action method can also return the view result for you, it can also return parcel view and see it and Json result and apart from this, there are many other data types whose data can be returned by this action result interface because brother, look, this

00:08:57 All these classes are child classes of which we have the action result interface because as I told you all these classes are implementing this interface and the reason why all these classes are implementing our interface is that this action method can return different types of data, meaning you can return partial view, JSON, empty result and many more types of data which this action can return

00:09:29 It can be done by default, look here, by default it is returning a view, meaning it is returning view result type of data to us, so something similar used to happen in MBC five as well, but as I told you, there was no interface there, there was abstract class, this one, meaning if you want, you can do that concept here as well, but what is the recommended way here, that you should do the action result that comes here, it means if you feel that your action method returns different type of data, then

00:09:59 What do you do here, I do the action result, but brother, if you know that this action method of yours will only return the view, then you know what to do, you remove the action here and put it here, view result here, see there is no error, that means what am I returning here, I am returning the view, and what type of data is the view, the view result type of data, means now that I have changed its return type here, it means that now this action method will only and only specifically

00:10:30 It will return the view only but if there was an action result here then it could return a view, it could also return a partial view, it could return a JSON result and many other types of data, so this is the concept of action result that you get, you have to remember this very well, so brother now see what I do here, now if I run my application here and show it because now you see if I go to program dot key, if I run my application now then what will I see here, hello world because I have just written the story also

00:10:59 Routing has not been implemented means I haven't even told the story yet as to which controller will be run, which action method will be run, which view will be run, I haven't even told the story, so if I run 'a' here now, we will see 'Hello World'. Okay, look here, if I run this, then look here, you can see 'Hello World', but brother, I have to run a controller, view, or an action method here, so what do I do for that here, here, when I stop this, so what do I do first of all?

00:11:25 First of all look here, I am commenting out this line tickets that we have in the control of control C and after that now I am going to write routing here to get my controllers and views aligned and I have explained this concept to you in routing so if you have not seen our previous lecture on routing then watch that lecture then you will understand it better so the code of that routing is fine, I am copying and pasting it here so that our time is not wasted so look here first

00:11:50 First what I have to do is look here, the line that you can see is builder dot but before that I have to add a line here, okay and look here what is that line, I told you builder means this one which we have a variable, okay, look with this what I am doing, builder dot services and with this look here what I am doing, fat controller with views, means look here here I am adding a service which will provide us routing for the controller and the views because with the help of this what we will do is our controller and

00:12:18 You will be able to add views and display them inside your application. So, first you have to write this line, after that the second thing you have to do is, look here, this line is builder dot build, after this I come here and pass the code, look here, I had explained this code to you when we discussed routing, that means, look here, I am using the map controller route here, which is a middleware, and inside it, look here, I have defined a pattern, I explained this pattern to you in detail.

00:12:46 When I explained routing to you, then look here, it is telling me that brother, when I run my application, then which controller will run, which home controller action method will run, which index will run, okay, so here the code that I have written, this one and this line which I have added here, so you have to write it in the same sequence, means this line which is there, you have to write it here before builder dot build, okay and when you write this, then what will happen then your home controller will be there and its action method will be there, but

00:13:14 Brother, now look here, inside the home controller, this index area method, what is it returning, it is returning a view, but now notice one thing, look here inside our application, we do not have any view, neither do we have a folder of views, so what you have to do to create a view, look here, right click on this index, okay and after that click on add view here, and look here, from here I am taking razor view, okay, you have to select it, add

00:13:39 You have to click on it so look here it is asking you the name of your view, okay and whatever else is there, leave it for now, I will tell you about it later, okay and now if I click on add, then look here our view has been added and if you look in our solution explorer, then look here a folder of views has been added, okay and look here if I expand it, then look since I have just created a view through the action method of the home controller, so look here inside views create a folder with the name home and

00:14:05 Look inside this, this is our view file, what is its name, look here , index.css, means this is the same file which is being displayed to you here, look at this, it means now if I come here and run my application, then this index area method of home controller will run and what will it do, it will return a view result to us and what is the view result, means this will return it in the view and its routing, its path, where have I defined it, look it is defined here, okay, so now I come here and

00:14:33 When I run my application, you want to see its result. Look, what result is you seeing? The index means our view is running successfully and if you look here, if I stop it and go to my view, then see what was written inside the index view, the index inside h1, okay, so this is the concept of whose action method you have because as I told you, what is a controller, it is just an empty class, but inside this, what do we do for the controller, we do its action method and brother

00:15:00 Look, it is not necessary that your action method should only return the view. What I told you is that it can also return all these classes and it can also return the normal data types that we have, meaning it can return a string, it can also return a wait, it can also return a boolean, how, look here, look here, I will copy this code from here, okay, I will come down and paste it here, okay and I will remove this line from here.

00:15:25 hmm, okay and then look here, I will change the name of this action method otherwise an error will come, okay but for example here, what do I give, display, okay, what did I name this action method, display, and in its return type, I should keep it here, string, okay, what did I keep here, string, and look here, I make a string return and inside this, I write here, welcome you program matter, okay, okay and then look here, here, I say copy it, come down and try it, and look here, I

00:15:50 I call it display ID, okay and I am saying that brother, it will accept a parameter which we will have, ID, and look here, what do I do with this ID, I make it return, okay and after checking the ID, we have to wait, so what do I do here, I remove the string and put an block here, okay, it's done, I am understanding, okay, so now look here, here I call both these action methods, how do I do it, look here, when I run the application, so here

00:16:16 Look, what's happening right now by default is that the index field method of the home controller is running. Look here, I say, brother, I want to do this display action method of the home controller here, which was returning a string. So look here, as soon as I enter, look what string is it returning, welcome to the program. After this, look here, I had created another action method, display ID, which accepted an ID parameter. So, I put one of my route parameters in the URL.

00:16:43 Example: I give it the value three, so where will this three go and be stored? Look here, this three will come inside this ID parameter and it will be returned to us and it will be returned successfully because look here, what have I kept as its return type, integer. So what do I do now after coming here, I enter it, so you see the result, look here, our three result is getting shown absolutely successfully, meaning whatever data I give here, if I give five here and enter it,

00:17:08 So if I do that, look here what do we see? Five. So, it means that this concept of action method that we have, along with these data types, can also return data of the type string, integer from the normal data type. So, then coming here, I'll stop it. Let's move ahead. So, we have these three points. Then, now look here, the next point related to the action method. Then, look what is it doing here. The action method is responsible for processing the request.

00:17:34 Date R Sent To D Controller, see brother, the thing is that whatever requests come to the controller, who basically handles those requests, who receives them, your action method means, whatever URL, whatever request is coming to the controller, I have to accept it, receive it and who does the work of generating the response according to that, the action method, then see the next point action method can return multiple types of data, see brother, the thing is that what did I tell you that action method can also return a view to you, partial view J7 string has become integer

00:18:07 Done, it means it can return data of multiple types. Then look here, the next point, by default it generates our response in the form of ActionResult interface and ActionResult abstract class. Look brother, the thing that I just told you to understand is that when you create a controller, by default, what is the return type of the action method that comes with it, I action result, means what does this interface do in its form, it generates a response for you, means you can do it as ActionResult or you can also do it as ActionResult abstract class.

00:18:36 Because I told you that all our classes, like view result or parcel, all those classes implement the action result interface that we have and action result also implements class and what did I tell you that this is the action result and this action result that I am telling you, you can understand that these two are the parents of all the classes, so this is the reason that when we have the return type of the action method, it is action result, so this is the reason that the different types of response are returned

00:19:08 It has the power to do so. Okay, so this point is very important, you have to remember it well. Then what you see here, what is it doing, the rules that you need to consider will be creating and the action methods that you follow. Meaning, whenever you create an action method, there are some rules that should be in your mind very well. Like, look here, what is it doing, the masses should be declared as public. Meaning, here, you see, these action methods that I have created, all of these are public, meaning, you cannot keep them private or protected because you cannot use them.

00:19:35 If you keep it private or protected, then when it calls the action method, it will not get the action method, so it can have an error, an issue, that is why we do not reduce this, that means we keep all the methods public, then look at the next point, action method is also public, it cannot be private and protected, that means the same thing which I just told you that all the action methods should also be public, then look at the next point, it cannot be declared static, that means see what is it doing here, look, this education is your concepts, that means you have this

00:20:03 What is public private protected access modifiers means access specifiers which are your concepts like static method, instance method, we also have the concept of these, so here it is saying that your action method cannot be static, right, it means you can say that it can be instance, okay then further, now look what it is doing here, which is the public method in side a controller which responds to the URL and non as action method, means here it is telling the second definition of an action method.

00:20:31 Brother, those public methods which are defined inside your controller which generate the response to your URL are basically called our action methods. Then look at the next point in and asp.net mbc application, you can create multiple action methods in a controller. The same thing which I just demonstrated practically. Can you also create multiple action methods inside your dot net mbc application? Then look at the next point, what is it doing, you can invoke an action method by specifying an URL in the web browser containing the name of the controller.

00:21:03 And you invoke the action method, that means look what I am doing here is that if you want to invoke an action method, then what you have to write in your browser is the URL and inside the URL you mention the controller and after that after the controller there is the name of your action method, so whichever controller you specify, after that whatever action method you specify in the URL, your result will be generated according to that, then further, now look here, this action result, okay, look in the big section of this action result, I have given you

00:21:29 I told you that there is a Cake interface which is being implemented by many classes. That is why your action method of the action result type can generate responses of multiple types of multiple data types. So now look here, what is it doing, DI action result when multiple action result return types are possible in an action. That means, look here, it is saying that the return type of the action method, you keep the action result when you know that this action method can generate responses of multiple types.

00:22:03 It is possible because as I told you, through action result you can return view result as well, you can return partial view as well in the result, you can return Json data as well in the result and many more. We have classes, there are data types, so what I am saying here is that if you feel that your action method will return multiple types of data, it will return data of multiple data types, then you can use this, but if you know that your action method should only return view, then you can reduce the size of your action method

00:22:33 Instead of the return type of action result, you can also keep view result, meaning you can say that i action result is general which works with multiple results and multiple data types, then look at the next point i action result and action result as a container has another action result interface and action result which we have an extra class, both of these basically work as a container and both of these are containers of other action results, what are the other action results, as I told you, it becomes view result, partial view result, Json data.

00:23:07 The result is done, so all these action results are inside which container, action result or action result okay, so you should know this concept very well because what I told you, all the action results are children of direction result and action result then look at the next point in date i action result is an interface and action result class date other action result in every it from which you are seeing, did you understand, it means all the other action results, as I told you the view result has become parcel

00:23:36 The view result is done, all of them are children of the direction result interface and the action result abstract class. So this point should be clear. Then next, look, now I will tell you here that brother, the return type of the action method, which we keep, action result, data of which classes can it return. So look here, here I have a lot of what we have action results, I mean look two, so I told you reservation result and action result, apart from that, look which ones

00:24:06 Results means what are the data types, what are the classes that the action method can return. Look here, there is content result, empty result, Jason result, parcel view result, view result and view component result, meaning from here, from content result to lake view component result, whatever you are seeing, all these types of data can be returned by your action method. Okay, meaning if you look here, if I talk about content result, then look under the context result, what is being represented, tax result, meaning if you see any tax result

00:24:37 If you want to display the string result through the action method, then you can use this action result here, meaning you can do this instead of result in the return type, but even if you use mention result, you can still return the contact result because who is the parent of the content result, you have action result, okay then look, what is the result, see what is it representing and action result date when executing bill do nothing, meaning look at what is there here, brother, if you

00:25:07 If you are thinking that your action result should not perform any less then for that you can do an empty result, brother. Now you must be thinking that such a scenario never happens, it definitely happens, such a situation can come to you, brother, when you are thinking that this action method of mine should generate an empty result, no result means nothing, I do not want any result to be displayed, then for that you can do this, then look at the Jason result, okay, what is it doing here, what does it do, you mean see what is it doing here.

00:25:37 Does it return a Json object to you, which means whatever result is there, it formats it in Json and returns the result to you, then look at the partial view parcel as well, that is a complete lecture, it is a separate step, we will see it later, here you just understand that we can also return the partial view through our action method, then look at the view result, view result, I told you, which is our simple view, and then look at the view component result, this is also a separate lecture, we will see it later, so here you have understood the concept

00:26:04 You have to understand that this type of result can be returned through the action method and one more thing I told you is that all these action results that we have, see in front of which it is blue, all of these are children of the direction result and action result means you can say that all these action results that we have, we can use both these action results or action result that means if I have to return all these results then instead of writing them separately I can just write

00:26:36 Can I do that? I have an action result or an action result because all the classes that you are getting, whether it is content result or result, whether it is Json result, all these are derived from these interfaces and action results. Like, if you look here, look here, you can see the index area method, brother, see, what is it returning, it is returning a view result, but if you look here, what am I returning, I am not returning the view result, I am returning the action result, so that is why the return

00:27:05 I am making you understand that through this action result we can also return our view result, we can also return partial view result, J7 as well and acceptor. So, I am explaining the same concept here. Well, if I show you code snippets of some of these examples, so look here. First of all, I will explain about content result. Look, what is content result? Represent tax result. Look here, we will write its code like this. Meaning, we have created an action method, what is its name? Content result. And look at that.

00:27:32 The return type is we have not kept i action result, what have we kept is content result, which look here, is it displaying the content, I mean a string, a tax return, what is written inside it, I am content result, it means when you know that brother, this action method of yours should only and only display specific tax content, then you may not use i action result, you can use content result, so the code for it, you will write something like this, you can practice it and see, okay, then next, what is next, then see Jason

00:28:01 The result is okay what is it doing is Jason result format d live object next to Jason Jason result is used to return GM formatted data means here if your action method is to return data result in the form of Jason then for that you can do any kind of result and you will write it like this you can see the action method its return type is result and look here I have taken a name Farhan Ahmed and look here what am I doing by converting it to Jason in this way I am returning it because you know

00:28:29 When we have to pass data from our application to the server, then what we usually do is we wrap it in Json and send it, then there we have a lot of it in it, then next now see what is happening here, partial view result, now brother, look partial view result is a completely separate concept, I will give you an overview of it here, brother, look, just like we have a complete review, there are small partial views inside our entire view, you must have often seen, if someone has worked with Edge, then in Ajax

00:28:57 What used to happen is that suppose I have several portions in a page and what I am doing is just with the first portion, I am reducing it, I am updating it, so what happens is that only that first portion, the first partial view, gets updated but all the other partial views, the portions, do not get updated, so partial view means there is a view inside the view, it is a small view with which we reduce, I am matching like that brother I have a complete view, inside that I have several portions

00:29:25 If I want to display a view then I can definitely do this, that means you can make a partial view an inner view as well as a nested view, so look here, what is the partial view doing, it represents the result and the action result data renders. Parcel view means what does a partial view result do, it performs an action result and gives the result, that means this is not a normal view, it is also a parcel. What is a partial view which is rendered inside a view, then look what is the partial views doing, they are accentuated bed net which comes to loading a part of the page through AJAX.

00:29:55 Return under HTML means, look here I am saying that partial views can be very useful under the concept of Ajax, so if someone knows the concept of Ajax, then he must have understood the concept of partial view a little bit, but if someone is coming after studying the concept of MBC 5 , then he will know 100% about parcel view, so throughout this lecture series, we will be doing parcel view completely separately, so if I have to write the code of parcel view here, then how will I write it, look here, the return type

00:30:22 Look what is partial view result and look here it is returning a parcel view and look here what is the name of the parcel view okay then next look here now look here view result view result means the same which is our normal view which I have just demonstrated so look here this is what view result represents and action result date of view you de respond means what does view result do it represents such an action result to you which renders a view and gives you in response means when you call your region method then

00:30:49 In the result, you will get a view result. View result means a view will be rendered in the response. Okay, then look, it is used to render of you. They give it when you want to render a simple dot cas tml view. Brother, look, the index page that I had created, the index view that I had created, what was its extension dot cas tml, it means, if you want to render such views, then you can use the view result here. Like, look here, there is an action method whose return type is view result. So

00:31:17 Look, what is it doing? It's returning a view to us. And what this view is basically is a view result. But in our case, which I just showed you the code of, in our case, our view was being handled by someone. It was returning an action result. But if you know that your action method will only return a view, then instead of direction result here, what can you write specifically here? View result. Then look here, empty result. I told you that brother, if you

00:31:45 Are you seeing that your action result does not display any result? For that you can do an empty result. See what the empty result represents and the action result date when they are executed, do nothing, they can use it when they want to return an empty result. Meaning, often such a situation arises that we are seeing that our action method should not do any work, meaning it should not display any result, meaning it should display an empty result. So for that we do this action result of empty result and its code

00:32:10 How to write it, look here, like this, brother don't keep the diesel return type and don't write the result here, what I did was I passed an object, okay, so we had some examples, what our action method can do, it can make a return, and what I told you, instead of all these classes, even if you write the action result here, even then your work will be done, so this is the special thing about the action result, but if you want to do it specifically, then you can do it separately, okay, so our lecture is over here.

00:32:38 This is the wind up, so if you liked any video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series, so if you have any confusion regarding this video, then do ask me in the comment box, see you Insha Allah in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we have a lecture series on asp.net codes going on, so in today's lecture we are going to talk about very important concepts and that is we have View Razor Intex and Razor View Engine in speed.net course 6 because brother you know that whatever application we make after using asp.net course, it is based on MBC, so in the previous lecture we talked about controller, okay, inside the controller there are action methods, we talked about them and in today's lecture

00:00:31 In the lecture, we are going to talk about View which is an important part of MC and along with View, we also discuss concepts of Razor Syntax and Razor View Engine. We will discuss all these three things in this lecture today. So, brother, I would like to comment that if you haven't seen our previous lectures yet, then see them first and then come to this lecture because there, I had created a project in Visual Studio in which we talked about Action Method Controllers. In this, today I will talk about View, Razor Syntax and Razor View Engine.

00:00:59 I will tell you the concept of this video. So before starting the video, a small request that if you are new on my channel and you haven't subscribed to my channel yet, then please subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our chapter number 14. And now we start our lecture. [Music] Okay, so look here, today we are going to discuss three things - View Razor Syntax and Reserve Engine. So first of all, let's talk about View Razor Syntax and Reserve Engine.

00:01:31 This is view, okay, understand what is view, means the simplest concept in MBC is this and what is view, so look here, if I explain to you in simple words what is view, brother, see, whatever is the HTML page of our application, basically it is called view, okay, so now look here, it has some points, see, first of all what is the point, now you provide the user interface, UI of the application, to the user, means what is view, view gives you a user interface which means your complete

00:01:59 The website has a user interface which includes a header, a navigation bar, a footer, it provides all these things to you and these are the things that are visible to the user of our website because as I told you, whether it is models or controllers, our user does not see them, the user only sees the user interface. Then look at the next point, now view is used to display content of an application and also you accept user importance, that means look at the same thing that why do we use view to provide the content of our application to the user

00:02:28 We use view to display it and brother you know what view we have is an HTML page and there we also create our forms and what is inside the forms, we get the input from the user, that means what we do through UI, we take user input from the user, that means whoever submits the form, what do we do, we get its data through view then see the third point view uses model data to create this UI, that means what is it doing here that view uses whose data, model data, that means

00:02:57 I am saying the same thing in model here, which we are going to study in MBC, model means what is a model, in model we have such classes that interact with the database, it means taking data from the database and displaying it in the view, and through this data, through this model data, what we do is we also generate our UI, that means we generate views also, so the same point is being told here, then see here, the next point view contains both HTML markup and C Sharp code, run on the web server, brother, see, now this can be a little bit unique concept for you.

00:03:29 And what is that, look here, what is it saying is that the view contains two things, one is HTML markup, that is, the HTML code, and the other is your code, brother, look, inside a view, we know that there is HTML code, but here it is saying that you can also insert C Sharp code inside your view, which runs on your web server and gets executed, okay, now I will understand this in more detail, okay then look here, the view has a file extension of dot csssml, that is, look, I told you this in the previous lecture

00:03:59 I also told you what is the extension of whatever view we create, dot CSSDML. Now look, what does it mean? If I go ahead and clear it to you, then you will understand it better. Till now, you should understand this much that whatever our view file is, its extension is dot CSSDML. Similarly, the extension of an Mp3 song is dot MP3. Okay, like this, good now, this which I told you is the concept of view, the concept of Razor Syntax, the concept of reserve engine, I will explain it to you with a diagram, then we will come to the rest of its theory, okay

00:04:29 So you will understand better from the diagram, so look here, here we have a view, okay, suppose what is this, we have a view and as I told you, the most common thing by which we refer to a view is that it is our HTML page, okay, it is an HTML page, it means inside the view we will have some HTML content, like look here, there is a heading and a paragraph and this view, what is its extension inside the dot net car application, dot csssdml, so now look here, its extension is this

00:04:58 It is dot CSSDML, right? It also has a meaning and what is that meaning, look here, C Sharp plus HTML means C Sharp on stance and HTML on HTML stance. It means that in this view file, we can write the code of both your education and HTML in our view file. It means, look here, right now you can see only the HTML code in the view, so what can I do inside it, I can write my C Sharp code as well, but I cannot write C code directly, means, like the way we write HTML

00:05:27 You can write it directly in your view, right? But, I cannot write your code directly in my view. For that, we have to use a syntax, a concept, which is called region, that means, by using Razor syntax, what do we do, we search your code in our view? It means, brother, look at Razor, by using it, I can insert your variables, loops, condition, switch, accept, that code here inside my view. By using Razor syntax, I can

00:05:59 Shiksha I cannot insert your code directly inside my view, if I want to insert your code, but for example, I want to create variables, loops inside my view, create conditions, etc., so for that, I will have to use the concept of Razor syntax and I will have to write my code inside the Razor syntax, so this is the concept, basically, what is the razor of your razor, it is basically a syntax and what we do inside that syntax is, we have to write the code of our own Shiksha, it means it is not a new programming language, but brother

00:06:28 The education that you have studied, we will do it. Okay then let me understand this concept a little more. See, what I told you is that we have a view and inside the view we can keep HTML code as well; we can also keep your education code. Okay, that means look here, this is my view, inside it I can keep HTML code as well and inside it, look here, I have written your education code as well. Okay, that means by using wire, I have created a variable and inside it, I have kept a string value, program.

00:06:54 I mean the name of your channel, so brother see, this is our view, in which our HTML code as well is there, your code is also there, but brother, when this view will be compiled on our browser, that means when this view will be rendered and sent to our browser, then to reduce this, we have a concept inside sp.net MC application which we call Reserve View Engine, because brother see, our view, in it, there is not only HTML code, but your C code is also there, that means your code is made from HTML and C

00:07:24 But brother you know that our browser has to understand only one language, it has to understand HTML, but inside our view, your code is also present, so brother it means that our view will be compiled, it will be converted into HTML and who does this work, the Razor View Engine means inside sp.net MC application there is a concept that there is a mechanism called Razor Engine whose work is that it compiles your view and converts it into HTML

00:07:56 inside and then it renders on the browser means this C Sharp code that you are seeing will basically be converted into HTML and then it will be rendered on our browser and this conversion is done by your C Sharp view engine. So this concept should be clear well. Now, I will tell you some of its theories in theory. Look here, in the words of Razor, it is telling what is Razor syntax. What is Razor syntax? It means it is a way of writing code based on DSP dotted frame.

00:08:24 That also means creating views means what we do by using Razor is we create our views and inside Razor we create our views by writing your code then look next point Razor is used to simplify the process of creating views, see brother this means we use this Razor so that our process of creating views becomes easy because look brother when I implement my your coding through razor, I will create variables, I will apply conditions, I will create loops, arrays

00:08:55 If I make it, it means that my process of creating views there will become very easy. So this is one point, then look at the next point, Razor is simple and easy, you can understand if I users are familiar with shishaf.net programming language. So look here, what is it saying that it is very easy to understand Razor. How easy is it that brother, if you have learnt C Sharp language, be it C Sharp or vv.net , if you have learnt this language, then what can you do, you can easily use the Razor syntax because what code do we have to write inside Razor?

00:09:28 Do you want to write your own code or write VB.NET code, but we have been teaching you here, so we will teach you here that, we will not do that, then look at the next point ahead, now see here, what is DMC framework using view engine, you can convert the code of a view into HTML markup data of the browser, see brother, the thing which I just explained to you in the diagram above that inside DMC framework, we have a view engine which I am calling here, Razor view engine, what is its use

00:09:56 What happens is that it converts the code of our view into HTML markup because inside our view there is your code as well, so the responsibility of converting that lies with our razor engine because as I told you our browser only understands HTML only, which means that your HTML page which is running on the browser, you will only see the HTML code in it, neither your code will be seen in it nor our code will be seen nor the Python code will be seen because when

00:10:27 If you right click and go to inspector element or view its page source or look at the coding, then you will see only HTML code there. Okay, then look here, the next point Razor engine is used today as the default view engine by the Ambassador Framework, I mean, look here, it is being said that the Razor engine is there within the MBC framework, that is, a dot net C++ is the default view engine within the MBC framework, that means compiling the views, rendering them into HTML, rendering them inside the browser, this one is less

00:10:53 What does it do? Razor engine does it, I mean the conversion of views into HTML, the razor engine does it. Okay, then go ahead and now see what it is doing here, it compiles a view of your application, when a view is requested for the first time, that means when a view is requested in the URL, when you specify a URL for it, then when you get the view in return, then what does the razor engine do, it compiles your view. Look here, why the term compile is used for it because inside the view there is also C Sharp code, that's why

00:11:23 Here I am talking about compiling, then look at the next point, deliver compile view on subsequent requests until you make changes to the view. This means here that whenever you make any changes in your view, okay, then your view gets compiled and after that, the view gets displayed to us against the place from where the request has come. okay then next, the next point, now look here, 10 Note introduces a new set of programming language but provides template markup syntax to differentiate

00:11:52 HTML markup and programming code in our view, I mean, look what I am saying here is that the concept of Razor does not provide us any new programming language, rather it provides us a syntax, a template within which we can write our programming code in the view as well as HTML code, I mean by combining both we can generate our view, okay then look at the next point, first it requires identifying the server side code from the markup code to interpret the server side

00:12:20 Code embedded inside a view file brother, look what is there here, the server side code, do you know whose code is it, your code and whose markup code is it which is your HTML code, it means that when this razor engine compiles your view, converts it into HTML, then first it checks which is the server side code and which is the markup code, that means which is your code and which is the HTML code, so when it identifies that what is the server side code which is inserted inside our view

00:12:51 We have to do it, we have to compile it, okay, so who does this work, our Razor view engine, then look, the next point uses the rate symbol to separate the server side code from the markup code, now look, this is a very important point, I mean, if we have to write C Sharp code in our view, then we have to use the @ symbol here, where will I use the @ symbol, wherever I have to write C Sharp code, wherever I have to write server side code, because brother, look, in HTML, you know that we use angular brackets there

00:13:19 But where you have to write your code in C, you have to specify @ there , then further, now look here, what is it doing, wild creating a razor view, you should consider following Russia, that means our MBC view, you can also call it a razor view because it has the razor syntax in it, so look here, first point start in line expressions with a single line, that means look here, if you want to give a single line, that means if you want to write a single line code, then you can code that.

00:13:49 You will start, you will start with the @ symbol, meaning for a single line, for one line, if you want to write your code, meaning one line of code, then before that you will put @ , then see here, what is it doing, close code block between @ curly bracket and curly bracket close, meaning here it is saying that brother, if you want to write code of multiple lines, then what will you do, then you will use this syntax, meaning @ and after that curly bracket and then curly bracket off, meaning all the multiple lines that are there, you will put them in between the brackets.

00:14:20 If you insert it then this is the syntax, I will now show it to you practically also, so that you can understand it better, see, variables are declared with the vir keyword, meaning, if you want to create variables, then you will create variables by using the word vir in your razor syntax, then look at the next point, now look here, what is it doing, enclose strings in quotation marks, meaning, if you want to specify a string, then what will you do with it, quotation marks, quotation marks, meaning, I am talking about the double codes that we use, okay, so if

00:14:48 Let me show you its practical till here. Okay, so let me come to my visual studio. Okay, look, I have come to my visual studio and look here, this is the same application that we had worked in in the previous lecture. What is its name? Look here, controllers and actions. So look here, here I had created a controller and inside that, look here, you must be seeing the action method of index, which returns a view of the index. So I have gone to my view because whatever the reduction of razor is, sir, we

00:15:11 If we do it inside our view, then look here, right now you are seeing the index. So if I run the application till here and show it to you, because brother, look, where have I defined its routing, look here, inside the program dot file, look at this, I have told you that by default the home controller will run here, its index area method will run and if the index area method is run, then we will get this view returned. So till here, if I run it and show it to you on the browser, then here, look here, you will see the result.

00:15:37 The index is running, meaning the application is running perfectly, so now here it is okay, I stop it and now what I do here, for example, I will give you a demo of our Razor syntax, so look here, if I put an h1 tag here, okay, I took the h1 tag here and inside it, if you look here, I say current date and time, okay, so look here, we have a class from C whose name is DateTime and it has a property which we call no, meaning look, what is this date-time-dot?

00:16:07 This is a class in data, this is your class from C and dot no what is it, this is its property and what it does is that it gets the current date and time in our system and brings it here, but now look here, this is being tweeted as a normal plain HTML, meaning we don't have any server side code here, so if I run it till here and show you, then you can see the result, so look here, what did it do next, it displayed what I had written here inside the text, but now what do I say here

00:16:37 Hmm, look, I come here and look here, here I put the @ symbol at the beginning because what I told you that this @ symbol represents our Razor syntax in which we have to write our code, so look here, as I have put @ , its color also see, what has changed here, and if you take the mouse over it, then look here it is telling that brother, what is this, we have a class, means a structure, so brother, look, right now my application is running, I have done my

00:17:02 The application was not stopped, so brother, you know that we have the concept of hot reload here. Hot reload means if you have made any changes or modifications in your view, then you can directly do it from here and after doing it, when you come to your browser, look here, here you cannot see the result changed yet, so what do you do, there is a hot reload option, look here, you have to click on it and look here, you can see hot reload, you simply have to click on it, okay.

00:17:33 So what will happen by clicking on this is that you will get the hot reload option here and look here, if there is a paragraph here, then you have to put a check on it, see, I have already checked it earlier, okay, so now if I do it from here and come to the browser, then look here, you are seeing this result, see what is it showing, current date, that means it is showing today's date and look here, it is showing the time, that's correct, you can see here below, even inside my system, it is exactly the same time, okay, so

00:17:58 This is the concept of its result syntax, well one thing you have noticed that we have in line expression, meaning single line, so whenever you want to give a single line, then you do not need to put this semi collar here, that means you can directly write it here like this, okay, so in the same way, look here, I copy this line from here, I paste it below and here I say, but for example current air, okay, current air, look here, I say date- time.nav and I do it here by putting a dot, it has a property.

00:18:30 Which we call the property of air, okay, so now let's do it with this and come here on the browser, brother, look, this hot reload feature that we have, if you have a good system, then the result is immediately sent to you, but it also takes a little delay sometimes, so to avoid the delay, what do you do, refresh it, if you refresh, then see here, what is the result that you are getting, current air is 2023, okay, did you understand the concept, so look here, I will stop it, look here, we have this, which I have now

00:18:58 I am reducing it, right? This is an in-line expression, which I just told you here, look here, what am I doing here, start in-line expression with at least the rate, but brother, if I have to give multiple lines, then for that, look here, how will I do this code block, look here, look, I come below this, here I start the code block @ and then the curly bracket, look, this is the code block of Razor syntax and now inside this, look here, what I had told you was that whenever we will create a var, we will use the keyword of var, so look

00:19:26 I am going to store the name of my channel here. Okay, look here, I am doing the program and then see what I told you that if you want to write a string, then you will write it inside double quotes. Okay, and look here, inside this, I will copy it and put another line here. Okay, and inside this, for example, what do I give here? Okay, next I will do it here. Next we will have to wait, so I will do it here. 30, so now I will put both these variables here, for example h2.

00:19:51 I get it here inside s2, okay , I get it here and here I say name, okay, after name what do I do, I will put @ here and this variable that I had created, name, I am calling it here like this, okay, and then I copy this same line from here and paste it below and here I say, I am calling the next variable that I had created above with @ and coming here I tell it, next, so now come here, let's do this and run it, so see its result, you can see it here, right.

00:20:19 The result is this in the program and what is next here 30 okay so come here, okay so look here, I have done all these four things and shown you, okay now next thing which is now look here and a razor code statement with semi colon brother, see, if you are writing a single line then you do not need to put a semi colon before it but if you are writing multiple lines, like look here, look here, I was doing it here with multiple code blocks, so after this you have to put a semi colon here because inside this we

00:20:48 If you are writing multiple lines, then it is mandatory to use semicolons here, but if you are writing a single line, then look here, here you don't need to use semicolons in and, okay, then look here, next point, the HTML extension of D is the store D racer view file data, C sharp dot net and D programming language, meaning what is it doing here, if you are trying to do it from C inside the view, then what would be the extension of your view, dot cshtml, but look here, what is it doing, what is it doing, vb.net she has to do it next to you

00:21:17 Programming language So what would be the extension of your view file? VHTML because in both C sharp and C++ we can create our own sp.net and ambic application. Okay then next, now look here in this lecture we will learn. Okay, what all things are we going to do here, in line expression statement, I have just told you how to write your education in one line and then declare variable using wire keyword, I have just shown you by doing this, then third point multi statement code block this also

00:21:46 I just showed you by doing it. Okay, mean look here, this has become an inline expression, mean it has become a single line. Okay, look here, this has become a code block and the variable that I had created here, we also have this reduced inside it. Then further, look at the next point, mean you can do IF cells etc. here, you can also do switch, mean whatever concept you have come up with for creating conditions, you can write that inside the razor syntax, similarly there is ARs, similarly there are loops, we can write all these things inside it now. Okay, if I

00:22:17 Let me show you a demo. Okay, so here, I want to tell you one thing brother, look, I am writing C code inside my view, this is not a new thing, brother, if any student has studied PSP, then the same thing used to happen there also, what we used to do in our HTML page, we used to open a code block and write HP code inside it, similarly it happens in Laravel as well, which is the framework of PSP, similarly it used to happen in asp.net web forms as well, but what we used to do there, there we used to use these angle brackets and angle

00:22:46 This percentage sign with brackets, we used to do it like this and what we used to do inside this was we used to reduce our code, that means inside this we had to write our code of programming language, that's it, okay, now what do I do here, look here, I will show you by creating a condition here, okay, look here, I open the code block here again and here, for example, I will create a variable, rather let's name it mi name, okay, mi name, I named this variable and inside this I can store it here

00:23:12 I will give it to you Adil, okay and now I will do it within an IF condition, okay, in the case of IF condition everyone will know, okay and look here, here I tell it that brother, this variable named me, if it is equal, okay, for this I am doing equals here, I say if this is equal to whose Adil, okay, so what should I do here, here you are putting the tag of S3 here, okay and inside this I am doing, but the example is Adil, okay, meaning if this condition matches then move, but if

00:23:41 I have done the else part here, okay, that means if this condition is burnt in S3 , okay, I do it here in S3 and inside it I write 'ka' here, bye, okay, if you want to write it in a simple way, okay, so look, I do this and come to the browser and after coming here I refresh it, hot reload is not working, so see here, what message is coming to you, this is heart, okay, but if I run the else part and here for example I write 'ka' Kumar, okay, then now this condition will be burnt with you, so which

00:24:09 This part will work, so let's do it from here, let's come to the browser, if I press f5 here , then see what result you are getting, bye. Okay, so this is the concept of which conditions you have, if you want to give it here, I mean you can just give it here, you can also give a switch here, video, anything, you can do that, mean if it loops, that is what it loops or if it loops, okay, yes, I will give you a demo of this as well, okay, look here, for this, I am creating a separate code block here, let's reserve and inside this, look here, I

00:24:39 I will do it twice after writing four, okay, like you will press tap on two-two, see, it will open its entire body and give it, okay, so look here it is telling what is inside i, that means it will start from zero and till where will it go, I will call it till 10, that means till the time our this ai (pi), this will keep on starting from 10 , then you would do an increment inside i, which is okay and now I will get this done here through h2, I will get it done tomorrow, so i is variable, so what will I put in the beginning @ and after that ai (pi), it means look here, this our loop will run from zero to 9 and the

00:25:11 Its number is also there, it will be displayed inside h2 . So I do this once, then I hot reload it here, okay. And coming to the browser, I come here and make it five. So see here, the loop started from zero and where is it going on, it is going on till nine. So, you can implement a loop inside this and then if you want, you can also write a foreach loop here, but for a foreach loop, you know, if you need a list, you need a collection, then for example, look here, here

00:25:37 I am creating an array, okay, what type of array is it? I am creating a string type array and here I call it name and inside it I pass nothing like this, but for example, I pass the array here, Adil, okay, and then I pass the array here, but for example, Kumar, okay, and then I pass the array here, but for example, Prem, okay, so what I have is an array of string type and look here, now I write forest and do it twice here, then okay, so look here, we have this forest look body, okay.

00:26:04 Here the name of this array is name, right? I put an s in front of it because I have already created a variable named name above, okay, names, and look here, inside the collection, I pass this variable here, which name is our one array, okay, so what will happen now, in the one by one names array, where will the data go, it will go to the item and I will get the item displayed here, for example, inside H3 , okay, look here @ item, okay, the square bracket will not come here, okay, it will come here, curly bracket, okay, so here

00:26:33 I'll enclose it by removing the brackets, put it inside the curly brackets, okay now let's do this, okay so now I'll show you the result in the browser, here, okay, let's do f5 , scroll down, so look below, you can see Adil Kumar and Prem, meaning any loop, be it par ville or do ville or par i, you can write that too in your reserved syntax and look, whatever code it is, okay, if I go here, it's ticked in my browser, look, I right click on it and

00:26:59 Here I have gone to view page source, look it is visible, you can see what happened inside this code in HTML, brother it has been converted and this code of ours was a server side code, who is doing the work of converting it into HTML, our Razor view engine. So you have to remember this point very well because this is also asked in interviews. So here we have all these things and here our lecture has been wound up. So if you liked my video, then definitely subscribe to my channel.

00:27:28 Go and also press the bell icon because many more videos are going to come in this lecture series, so if you have any confusion regarding this video then do ask me in the comment box, see you insha Allah in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are having a lecture series on asp.net course 6, so in today's lecture we are going to talk about layout view. Layout view whose other name is also master page, means this is the same concept of layout view which is the concept of master page in other programming languages. It means brother, whenever we talk about web development, then whatever language is related to it, the concept of master page is definitely implemented in it because what is this

00:00:32 What it does is it reduces our time and our effort. Whenever we are developing any website design etc. then look, inside SP.net, we have the layout view file, its name is score layout.cssdml and this is called the master page file. Brother, if someone already has the concept of master page, like web forms or any other language, if it has the concept of master page, then this lecture will be of great harm to him. So brother.

00:01:02 If you haven't watched our previous lectures yet, then I would recommend that you watch the previous lectures first and then come to this because in the previous lectures we had created an application in which we had implemented controllers, action methods, views and razor syntax. Now in this application, I will implement the concept of layout view. So brother, before starting the video, I have a small request that if you are visiting my channel for the first time and if you haven't subscribed to my channel yet, then please do so with all your heart.

00:01:27 I request you to subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our part number 15. And now we start our lecture. [Music] You see, this concept of layout view, which is also called master page, first I will give you a scenario of it with the help of a diagram. This will make it easier for you to understand the layout view. After that, we will come to its practical. So here, brother, you know, we use a web page.

00:01:59 Let's create it inside our application. Okay, suppose I have created an index page inside my asp.net course application. Index means home page and you know brother, in all the pages that we have, we create a header and in that we also create a footer. Inside the header portion, there is a heading, the name of the website, there is navigation bar, etc., okay, there is a header portion and then there is a footer portion and what is in the center, the specific details of the index page, we create that in the header and

00:02:28 Let's insert it in the middle of the footer. Again, in the same way, if I create another page, the about page, then it also has a header and footer. That means, the header and footer that we had in the index page is the same in the about page as well, but the center data is different in the about page. Similarly, if we take a contact page, then the portion of the header and footer in it is also common, but the content in the middle is different.

00:02:56 So what it means is that brother, whatever pages we have that we create inside our web applications, what mostly happens in it is that the header and footer portions are present repeatedly on top of every page, which means that the header and footer portions are common on every page, so brother, now here a scenario often comes which makes our work heavy, makes our work large, what happens is that brother, the code of the header and footer, we have to repeat it again and again on every page, but brother, I am thinking that I have such a

00:03:29 There should be a concept by doing that, this header and footer, I mean this common code which I have to copy-paste again and again on every page, I mean there should be such a concept through which I don't have to write this header and footer code again and again on every page, so for that concept, we do it, how to see our layout view, I come here, okay now suppose that I have created a layout view here, what am I doing, master page, so brother, now that data which is getting repeated on multiple pages, means

00:04:00 That data which is common on other pages is from and what is that code, like the header and like the footer, means this is such a code which is getting repeated again and again on every page, so what I did, that which was common data, I defined that data where, inside your layout view inside your masterpiece, okay and now what will I do, now I will create my other pages, okay like the index page, index page, we basically have a view and what is the extension of the view, I told you dotted line, in the same way I did

00:04:31 Here I created another page, the About page, and similarly here I created a third page, the Contact page. So now you know what I will do, the data in the index page will be related to the index only, which means in the index page I will write only that data which is unique. Similarly in About also I will write only that data which is related to About and then in Contact also I will write only that data which is related to the Contact page which will be unique. This means that I will not create headers and footers in these three pages. Where in header footer

00:05:00 I will take it from my master page, I will take it from the layout view, it means that you should understand that this layout view of ours, this master page will be the parent and the rest of the pages, index, about, contact, will be our children, these three will be the children of this layout view of our master page, it means that now this master page, in which the code of header and footer is specified, I will get this page connected with the index as well, I will get this page connected with the about page as well and I will get this same page connected with my contact page as well, so see, when I

00:05:35 My master page which is my parent page, when I connect it with all the child pages, then what will happen is that this code of the header and footer, this portion of the header and footer will automatically get transferred to all our pages, it will be shared, this means that I don't need to write the code of header and footer in the index, there is no need to write it in about, there is no need to write it in contact, as soon as I connect my master page with my index page, with about, with contact, then

00:06:04 What will happen is that the code of the header and footer will automatically get transferred to all these pages and will be shared, that's it. And this data, that is, the index data and the about data will be placed at the center of the header and footer. If I show you this practically, you will understand it better. And see, what I am saying is that I will connect my layout view with my page in the same way, the index, about, contact, so how will I get this connected?

00:06:33 See we have a property inside spot net car whose name is layout so what I will do by using the layout property is I will connect my master page with my other page that is with the child page so remember on the master page I am making it the parent and all the pages I am connecting with it, views I am connecting them, I am making them the child and remember where will I do this layout property, I will do it inside my child pages as well and after doing this inside the child page

00:07:04 Whose reference will I give, of my layout view, of my master page. Alright, so brother, now I will come inside my visual studio and I will demonstrate it by doing it practically and then we will do the rest of the theory after this. So here, which is in my visual studio, alright, so look here, I have opened the same project. You must be seeing here the controllers and actions which we had reduced in our previous lecture. And look, in the previous lecture, we had talked about Razor syntax, we had talked about Razor View, so look.

00:07:29 All the code that I had written for this, from here to here, okay, look, all the code that is after the index, I am selecting it from here and I am commenting on it, how will I comment, control C, okay, this is a shortcut, see through this, sir, I selected the code, what have I done, I commented out and you notice one thing, look, there was razor syntax also inside this code, so look at its comments, look at the way it comes here, there is a d rate star and then at the last there is a, and while doing this, first there is a star and then there is @, so meaning now if

00:07:57 If I run then what will I see only, this is index, so look, inside this, I do some changes here, okay, but for example, I call it index page here and inside this I also create a paragraph here, okay, I create a paragraph and inside this here I call it this is my home page, okay, this is my homepage, right, so look, right now I have created only the view of the index here, okay, if you look here, inside the home controller, right now there is only the action view of the index, so here I

00:08:23 Let me create two more action methods. This is for the index, right? And I create one here, for about, called about and then after that I create a third one here, for contact. Here, I say contact. And if you look here, inside our views folder, inside home, look here, right now there's only the index.c html page. So, I'm going to create views for both of them. Right now, I'm going to right click on out and go to add here. Look, here you must be seeing a razor.

00:08:48 You also have to select this and add it. Okay, look here, let the name View remain as it is, do not change anything else and click on Add. Okay, so look here, the about page has come and inside it, I will copy paste this code from here and bring h1 and paragraph here. Okay and coming here, I paste it like this and inside it, what do I give here, about page, okay and here below, I will write this, about space, okay, about space has come inside it, right now here

00:09:14 Let's go to the third one, I will create an action method, look here the view is related to the contact, so right click on the ad view above it, here I select the razor view from here and its name is Contact, okay and here if we click on the ad of A, look here this contact page has also been added, I will remove this from here as well and copy paste the same code here, okay and here I call it Contact Page and here I call it this Contacts Page, okay, I have given the contact space here, now let's do it from here, so

00:09:40 Look here, what have I done now? I have created child pages here. In this diagram, I told you that the index page is done, about page is done, contact page is done. I have created these and here, notice one thing, here I have not given the data for header and footer. Even in the contact page, there is no data for Hydra Footer. Look here, it is not there in about page and look here, it is not there in index page either. So, now what do I do here, I create my own layout view. How will I create the layout view? Look, first of all, I have gone to my

00:10:03 Inside Views and inside the Views folder I'll create another folder. So look, I'll right click on Views and go to Add and here I'll select New Folder and look, what will I name the folder that I'm creating inside this? I'll name it Shared and what will I do inside this shared folder is I'll create my layout view, my master page, because as I told you, this master page, this layout view will be shared within multiple views, so I created a folder for it.

00:10:30 What is its name? Shared and the recommendation is also that you should name it Share. So now what I do in this is add my layout view master page. Look here, right click on the ad above it and after that from here, above the new item and look here, you must be seeing the search. Here you have to search for Razor. If you want to search for Razor, enter it. Look here, you must be seeing Razor Layout. You have to click on it and look, as soon as you select it, look here, the file

00:10:56 What is the name of the score layout dot cssml showing you inside, so I will name it as it is here, if you want you can change its name, okay, so I am leaving its name as it is because through this name what is it called, layout view or masterpiece, okay, so now look here, if I click on add, then look here, like I added, then look here, inside the shared folder, this has been added in layout dot css which is our master page file and now if you look here, look here you will see

00:11:22 Some code is visible, HTML code is visible, look at its body, there is dev inside its body and inside it, look here, here you can see a kid of method, what is its name, render body brother, now understand the concept, you know what is this render body referring to, what is it pointing out to, this render body is pointing out to the data that we have, which is our child page present inside it, like this, look here, this code which is inside the index, this two lines of code, where will it go and get replaced by this render body

00:11:50 Again in the same way inside about, this two line code which is written here, will go and get replaced, look here, from inside this cylinder body, similarly whatever data of the contact will be displayed inside the render body here, so now inside the layout view, that is, inside this view, I will write my header and footer code, look here, this div which must have been enclosed inside this, I am removing this div from here, okay and now what I do here, look, I create a separate div here, I create a div and inside this I

00:12:16 I am creating an h1 here brother see, I am not making a proper header and footer, a lot of time will be wasted in that, you can try it yourself here I am creating a lamp and inside it look here I am creating a ha one and inside it here I give what is header portion, okay, what am I giving header portion, and brother see, I am not even creating a lamp here, rather we have the tag, which is the header tag for which header, okay look here I am specifying it here so that you can understand it better here

00:12:45 I will keep it in the middle and after rendering the body, look here, I am putting the footer tag, okay, the HTML footer tag here, and inside this, look here, I am putting another h1 tag and inside this, I am putting footer portion, and brother, look here, I will also change its background color a bit, okay, like here, we have this header, okay, inside this, I open the style attribute, how do we apply it, okay, here I am right now doing it in line, okay, look here, I call it background color and

00:13:16 Here, I will give its background color, here it is, okay and then look, I copy this same line from here and take it here and paste it in the footer and here I call it, for example, aqua, this is a blue color, light blue color, okay, I have given it here, so that means now you notice one thing, the common data, common data means that which will be there on all the pages, the way there is a header, a footer, I have created it inside this layout view of mine, that means I have created it inside my master page and look what is there in the middle of it, in the middle of it and inside

00:13:42 Body and what did he say, which data will the render body point out to, which is our child p inside it, okay, which is here, this is it, okay, but now you see, if I run it, then here in my index page, this about page, I will not be able to see the header and footer, because because till now, this layout of ours, this master page, I have not connected it with my other pages, and what did I tell you, what will I do to get it connected, see here in the first place

00:14:12 Look at the layout property and what I told you in the layout property, where will I do that in my child pages and inside that property, what will I give the reference of my major paste, how will I give it here, which is okay, so look here where I am present right now in my index.cssml , so now first of all I will connect my layout view which is the master page with my index, so look what will I do here, look, this is what you can see is the razor syntax, which we had in our previous lecture

00:14:36 We had discussed it inside, so look here, in Razor Syntax, this first line which is written about view data, I come here and open my without property here, okay, I will give the text of my layout view here in my master page, how will I give the text, look here, okay, first of all I will do that here, above the oil symbol tap, you will find a button, you can press shift and see, we have this sesame symbol, what does it refer to, it refers to our root folder, root folder means what is the name of our project

00:15:09 Controllers and Actions and inside this look here we have a folder called Views so look here I say Views okay and inside views I have created a folder called Shared and inside shared look here we have a file called underscore layout dot csshtml which is our master page file okay so look inside shared I again have forward slash and look here underscore layout dot csssdml okay so look here which was my index page I have placed

00:15:36 By using the out property I gave the address of my layout view, what happened was that my layout view got connected with my index page here, okay, so now if I run this here and show it, okay, before that look here, let's check the routing, look here, in program dot key, the home controller will run and its index area method will run, okay, so let's run it further and see its result, so look here, you are seeing your index page now and you can see these two lines, I have added them in my

00:16:04 It was written inside the index view but look, this header portion and footer portion that you are seeing, from where is it coming, it is coming from our layout view because I have connected my index page with my layout view but brother, look, I have not yet connected the about and contact page with the layout view, so if I access it here, look, I say here that brother slash home and inside home, I have to call the action method of about here, so if I come here and press enter, then

00:16:32 Look here, we don't have header and footer yet because I haven't set the layout property in it yet. Similarly, if I come here and do this in contact and after coming here, then this contact is not showing. So, what will I do for it here? Let's stop it here. Look, this line is there in which index, I had done layout proper. I copy it from here and I paste it here in about. Look here, I do Control S and then

00:17:01 Let's go to contact and look at this here as well, I will scroll down and do this, okay, so you see, now I have connected my layout view with each of my child views, so now look, when I run this here, you can see its result, see this, now this is becoming the index page, okay, I go here, I put home slash about here, okay, I enter one here, so see, now about is showing and inside that also header and footer are coming here, similarly if I give contact here, okay

00:17:29 If I come here and write contact and enter it, look here, now header and footer are being shown in the contact page as well. So, did you understand? See, how beneficial this is and how useful it is for our projects? So, you should be very clear on the concept of this. There are some very important points in the theory, look here. You must have understood the diagram well. Whatever points are there, look here, what is happening? The layout view provides a consistent look on the D views.

00:17:55 In a web application consistent look means because brother you know that your header and footer are dependent on what is inside all the views, so what I did was I defined its header and footer in the layout view and I shared this layout view in all my child views, so what happened was that the look of my website became consistent then look next point layout view, ahead of this asp.net form master page brother see if anyone read the concept of web forms in this dotted

00:18:24 So the concept of master page there, the same concept of layout view is there in speed.net, then look next point extension of layout view is dot cs html which is a number of our normal view, the same is the extension of our layout view but what is the difference in this that we have a master page brother, see in a way, if you have ever gone to a five star hotel, then for example in a five star hotel you have 500 rooms and each room has a separate key, but the manager of the hotel

00:18:53 It has a master key. The master key is such that it is placed at the bottom of every room, means if you put it in room number 299 , even then that room will open and if you want to open room number 300 by using this master key, even then it will open, so the same concept has been used here, after doing that on the master page, see next point the default name of layout view is underscore layout dot c scheme, means when you add layout view for the first time in your application, then its default name is this.

00:19:26 If you want, you can also change its name, there is no show in it, then look at the next point ahead, now look here, what is it doing, the layout view file is usually located in view slash brother, look, this layout view, I had created it inside the shared folder and where was the shared folder created inside the shared folder, so I mean, there are as many files of layout view, that means you can create multiples as well, it is not necessary that you create a main page because brother, look, in the way you know our web pages are, in the way it is indexed, about, contact is done, it

00:19:56 The layout is different but if you go to the admin dashboard, there are multiple views inside it as well but its layout is different so it means I can define two layouts there one web page for this and other for my admin dashboard but all the layout views that I will create, I will create master pages, where will they be inside the views in the shared folder then look next point and application can have multiple layout views the same thing which I just told you that you can also create multiple layout views for your single

00:20:23 Inside the application, then look at the next point layout properties, you connect layout view with child view, brother I am talking about lay out properties, you use this property to connect your layout page, your master page with any view, because look, if you remove this layout property from your child view, then what will happen is that the connection that was happening with the master page will end, okay, so here our lecture is over, so if you liked my video, then definitely subscribe to my channel and also like it.

00:20:52 Also press the bell icon because many more lecturers are going to come in this lecture series, so if you have any confusion regarding this lecture then do ask me in the comment box, then Inshallah we will meet in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are running hp.net code 6 , so in today's lecture we are going to talk about view start file, view start file, the name of this file we keep with you, what is its name, look here, inside cshtml, it means it is a view file, but what do we call it, view start file, so what is it, for what purpose do we use it, we will see this in today's lecture, but brother, before starting this lecture, if you have not seen our chapter number 15

00:00:35 In which I talked about layout view, layout view means master page, which was a very important concept which we discussed in part number 15, so if you have not seen our part number 15 , then first watch that lecture on layout view means master page, after that come to this lecture because if you do not understand layout view, then you will not be able to understand view start file also, so to understand this lecture it is compulsory that you have to watch part number 15, if you want to watch the lecture on layout view, then before starting the video click on the small

00:01:05 I request you that if you are visiting my channel for the first time and you haven't subscribed to my channel yet, then please subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our part number 16 and now we start our lecture. [Music] You see, view start file means when do we use the view start file in our project. So, first understand this scenario and then we will come to it.

00:01:37 Okay, so here, look at the practical, first of all there is a question and what is that question, do we need way start dot cssml file in asp.net kar mbc application, means the question is that why do we need this view start file in our sp.net kar application, first understand this, next of all, no one has used d layout property to associate a view with d layout view s, below brother, see, in the previous lecture, we had studied the concept of layout view, layout view means the concept of master page

00:02:08 I had read it in which I had told you that such data which you want to repeat again and again in all the pages, where do you keep that code in your layout view in the master page and then how do you connect all your views with your layout view, but for example inside the layout view, inside the master page file, I had created a header and footer and had connected the layout view to the master page with all my views, so when I had connected my layout view with all the views, then the benefit from it

00:02:39 What happened was that I didn't have to write the header and footer code again and again in all the other views, but the header and footer code was being inherited from where, it was being inherited from our layout view, okay, so which one is this, now we have seen this in the previous lecture, right? And how did we define the layout property in our view, look here, like this, that means inside you, we used to use the layout property in the razor syntax, and then inside this we used to use our layout dot cssml

00:03:08 How did we define the text of the file, as I told you, this layout view file used to be in the folder inside the shirt and the shared folder was in the views folder and what was this oil symbol doing to our root project for days, okay, so brother, now it is okay to understand the concept here, look here, it is a very important point, suppose they have 100 news in the application and if you want 100 views, then that is the layout file, now brother, look here, its purpose is being told about the view start, brother, look, if you suppose that our

00:03:38 There are 100 views in the application, whether it is 100 views or 50 views or 20 views, and if all of those views need to use the same layout file, that layout file means the master page file, if we need to do that, then what do we need to do, look here, we need to set the layout property in which you are seeing 100 views, that means what is happening here is that if you have 100 views in the application and in all those 100 views you need to use the same layout file, the same master page file, then what do you need to do that

00:04:11 The layout property is there, you will have to write it 100 times inside your views, that means you can see how much longer our work will become, that is, how much longer the copy paste work will become, that brother, I will have to copy paste that layout property again and again within 100 views, because I want to know the full form of that layout property, don't repeat you sir, it means that brother, if I want only one view file in 100 views, if I want a layout file, then what will I have to do? I have a layout property, inside which our layout view's address was there, that one had the text.

00:04:51 I will have to repeat this line again and again in 100 views, so brother, look, in programming there is a principle which has to be taken care of very much by a programmer, a developer, and what is that principle? Dry dry means repeat yourself, so right now we are violating this dry principle, aren't we violating it, how are we going against it because we are repeating that layout property again and again, so brother, look, when you violate the try principle, when you go against it, then there are some decided advantages, those distinctions

00:05:23 Code redundant code means duplicate means you will have to work with duplicate code again and again, first of all this is its disadvantage and then secondly maintenance overhead maintenance overhead means that brother when you repeat one thing again and again, copy paste it, then it will become very difficult to maintain it, I will give you an example of this brother suppose you have used layout property in 100 views , I mean if you have added layout file here, then suppose that brother tomorrow we get such a scenario in which I am

00:05:55 I am saying that brother, in the 10 views out of 100, I have to give a different layout file, that means I have to give a file with a different master page, so what will you do then, how will it become difficult to maintain it for you guys, so only then we get the view star.cssml file, that means our view star.cssml file, this one makes it very easy, I will show you this also practically, then you will understand, then look here, now look, it is telling you where to place the view start file, so look here, what is it

00:06:24 Generally the view start file is located in Piyush folder, that means this view start file, you said that you have to create it inside your views folder, now I will do it practically and explain it to you in a better way, okay then look here, now there are some tasks, okay, we perform these tasks one by one, but first of all, I will tell you that brother, where is this view start file created, how to do it, okay so come here, look here on the browser, I will show you the output of that application

00:06:53 I am going through this in which we had done the layout view and we had done this in our previous lecture in part number 15 so look here, as soon as I entered the local host and port number here, that means when I ran my application for the first time, then look here, only the index page is visible to us but in this index, if you see, the header and footer are also being shown, so brother, look here, this header and footer portion is coming from where, it is coming from your layout view file, that means the master page one.

00:07:21 By file I mean this index page, I have written only these two lines of code in it, but because I have connected this index view with my layout view with the master page, so the header and footer that I had defined inside the master page, you can see it here and in the same way I had also shown the about here, that means if I come here, I see the home contact first, so look, this is the contact page, there is a header and footer inside it as well because inside this also I have shown my

00:07:50 What have you done with the layout file? It has been connected and again in the same way if you look here, I click on home, home controller and inside it, click on about. Then here also you will get to see the header and footer portion. How will you see it? Come here. After coming here, if I stop my application, then look, right now we have the index view open here, inside which I have written these two lines. So look here, there is another property, which I have put inside my razor syntax. And look here, I have entered my layout dot HTML.

00:08:17 The text of the file is given here, okay, and I had created it, look here, under views, I had created a folder of sets and inside it I had created my layout dot html file, okay, okay, and inside this, if I show you, look here, you must be seeing the code of hydro footer and this render body, I had told you which code will come here, which would be of our child views, like index is done, about is done, contact is done, okay, but you see what is the problem, now the problem is that brother, I have three

00:08:45 There are views, right? And within all the three views, I need the layout file from here. So look here, this line, I have copied it here as well. Look here, I have copied it in About as well. Look here, I have copied it in Contact as well. So it means that if I have 100 views in my project and I need the layout file from all the 100 views, then what will I have to do? I will have to copy-paste this line 100 times , which is not a good best practice at all. So, I

00:09:15 What I'll do is I'll just create a file called viewstart and inside that I'll just define this and define my layout file, so the viewstart file will automatically apply that layout file to all your views. Okay, so what I'll do now is look here, I told you that where will we create that viewstart file? Inside our views folder, so look here, on the views folder, I'll right click and go to add and here I'll go to new item, okay.

00:09:42 Look here, you are seeing search, in your search you have to write it simply as region, okay, look here, like you wrote razor, look here, when we created the layout view file, then we named it Razor Layout, okay, you must have seen Razor Layout, but now I want to create a view start file, so look, right below this we have this file, what is its name, Razor View Start, look, when I selected it, look below, the same file is there which I told you, what is its name, inside score view

00:10:07 star.cssml Okay, so from here you have to select it and if you want to change its name by coming here then you can do that but I'll leave its name as it is and click on add so see here this file has been added and if you look here directly in structure then inside views you will find this file view start and inside this you can see Razor Syntax is already opened and look here the text of that which was our layout file is already given here but you notice one thing that the text of this

00:10:36 This is not the complete text and you don't even need to write the complete text here, it means if you look here in the index file, you should have given the complete part here, but what do you do here, here you just have to write the name of your layout file here, because look what is the name of our layout view file here, look in underscore layout dot com team, so here you don't even need to put the CSSDML meaning extension, it means what is the use of this view start file, is it for all the views that you have

00:11:06 Also, what will I do to this layout file, it will attack, it will connect, it means now whatever views you have here, now you do not need to write this line, so I will comment out it, Control C, okay, so I have commented it here, now on this about, okay, let us select the about from here, by doing Control C, what do I do here, I will comment out it and then on this contact also, what do I do to this line here, Control C, it means that the

00:11:33 The dry principle was being violated now what did we do here, how did we finish it, we created a view start file here and inside that the text that we had of our layout view file, I defined it here with the help of layout property like this, meaning you will get the result from here but what is different here is that look here, inside contact, inside about, I was copying and pasting the layout property again and again, now I don't need to give it, okay, so now let's start our application

00:12:00 Let me run it and I will show you the result here, look, this is running on the index and look inside this, the header and footer portion that you have from our layout view file, its result is showing perfectly here and if I go here, inside home, I go to about here, okay, look here, I have changed the URL, if I enter it, then look at this, even now our view file, the layout view file, is connected perfectly with our view, then this, which is the last one here

00:12:26 We had the contact, so if I change the URL and enter the contact here, look here also you are getting the result correctly, you can see, so this is the purpose, this is the aim of the view start file, but brother look, the layout view file and this view start file, by using this, you can also achieve multiple tasks, I have told you that task here, here, I will tell you that task one by one, so look here, first of all, what is the first point saying, see if they have multiple layout views.

00:12:54 Yes, we can have multiple layout views with view start file now look what I am doing here, suppose I have multiple layout views, multiple layout views means brother it is possible that this file on the major is multiple, I told you in the previous lecture, so brother if I have multiple layout views, then how will I do my view start file with it because brother now look here, come here and I stop it, now look here, these three views are index, about and contact

00:13:22 Along with that, we also have a layout attack. What is its name? Layout inside Store Layout. So, now I want to understand the concept. See, what I am saying is that brother, with these about and contact views, what do I need to do is, I need to connect this layout file. But this index file, okay, I am saying that with this, I need to connect a different layout view file. So, what I do for that is, for a second, I will create a view file here. Look here, under shared, you will see this first layout view.

00:13:52 You must be seeing the file here. So what I do here is I create another layout file here. So what you have to do is right click on the lion, go to add and here you are on new item, okay and what do you have to write here, region. Okay, in search you have to write region and after that look here, you must be seeing the razor layout, you have to select it. So look here, now look at the name of this file here, this layout one, I will make it a layout, okay, so that means the first file.

00:14:16 That's layout one and we have this layout, okay so let's add it, okay look it's added here, so what I do is I simply do a decrease here, I put a h one inside it, okay random body, I'm leaving it here and here for you to understand, I will write second layout view file, okay, I just put an h1 here so that you can differentiate it a little bit, so what I am doing now, I am saying that brother, this layout view file that we have, add it to

00:14:43 I will connect it here in the index, so what I need to do here is simple, look here, I will apply the layout here, okay, the layout property is okay and inside this, look here, this text is there, okay, I will copy it from here, this which I had commented out, I will copy it completely from here, okay and inside this, look here, which layout file will I give, layout file, you are right, so now let me understand the concept, so what will happen now, look here, inside the index, this layout file which I have given, this will be given priority

00:15:11 It will give meaning, it will give importance to this and in the index view, this layout file that we have will be connected, that means when you use the layout property here, then what will our application do to this layout file of yours, it will get priority and this file, right, inside the view start, which you had defined out, what will it do to it, it will ignore it, okay, it means look here, inside the index, this index page, in it, the layout file will be there and these other two files

00:15:42 These are about and contact because I did not specify the layout properties in it, so which layout file will be connected in it, this one which I have specified here inside view start, okay so look here, if I run this file, I will see its result, so look here, look at this because our index page is connected with our second layout file, so what did I write inside the second layout file, look at this inside h1, second layout view file, okay, so its data is already connected here.

00:16:09 With which, with index, but if I show you here, this was our home context, if we enter it, then look, this is connected with G view, this is connected with G layout view, there we had created these header and footer portions, okay, did you understand? Similarly, if you go to about, then inside about, you will also see the same header and footer, you see, in this way our first task was achieved, which I had told you here, okay, now you must have understood it, so look here, this is what is happening if they specify the layout.

00:16:39 property in a view and then well, in view star what happens, I mean, look, I have just explained this scenario to you, what will it do to the layout property, it will give priority, I mean, it will look at the layout property, the layout that is defined in it, it will give priority to it and the layout that you had defined inside the view start, it will ignore it, okay, so this is also an important point, then look at the third point, if you don't want layout view in a particular view, look here, it is saying that, if we have a layout in our view

00:17:08 I don't want any layout view inside it, that means I am saying that brother, I don't want any layout view connected with my one view, I don't want any masterpiece connected with it, so what will I do with it, so look here, I stop it here, so brother, look here, now what we have in index.csc, what is it doing to the layout file, it is inheriting it, that means it is connected with it, but I am saying that brother, I don't want any layout inside this index file of mine, that means

00:17:35 I don't want any master page here so what will I do for that, look here, so what will I do for that here, look, I will first comment out this line from here, okay and then I will paste it here below and look here, what will I do simply here, I will pass my layout property here, null, so look what will happen by passing null is that brother, this index file will not have any layout view attack with it, I mean what was happening earlier, this view start file is there, inside this our

00:18:02 Bring it closer, it is defined, the same was there in our index, but now see what I have done, I have made the layout null, by making it null, will this happen? Now the index view that we have will not be attached to any of our masterpieces, or any layout file, so let me run it, you should see its result, look here, here neither the first layout view is showing to us, nor the second layout view is connected here, because what have I done to my layout property, I have made it null, okay, whenever you work with layout views,

00:18:34 Then this is the last one, now see what he is doing in the last point, he can select an layout conditionally in viewstart file, means, look here he is saying that we can put a condition inside our viewstart file and what we can do according to the condition, we can connect our layout view with the views, this means, I will create an IF condition inside the viewstart file and inside the IF condition, I will say that brother, if this condition is true, then this layout view should get connected, but if that condition is true, then this layout view will connect

00:19:06 Let's do it, I mean look here, I'll explain you a scenario brother, suppose I'm making a student management system, here I'm making a school management system, I mean I'm making an application for the student or the school, I'm making an application, so here's an example application, students will also access it and here's an example admin will also access it, so brother, look, I'm saying that brother, for the students, I've created a separate layout view file, it's necessary, and for the admin, I've created a separate layout view file, so I say that brother, first I'll check

00:19:32 I ask which user has logged in, that means what I am saying is that if a student is logged in then the student's layout view should be shown and if an admin is logged in with you then the admin's layout view should be shown, so how will I put the condition for this concept, look here, okay, look, I have gone here inside my view start file, okay and look, this line which is there, I will comment it out from here and inside it, look here, how do I put the condition, look here, okay, look, I will put if

00:19:59 I am applying the condition, I told you that within the Razor syntax you can also apply IF condition, so look here, but for example, I am saying, user is in role. Okay, here we have a method user is in role, and I am saying, brother, if our user is an admin, then what should we do, so which is this layout, the one that I created first, in which you are not written, that means it is underscore layout dot CSSML, right, that means I am saying, brother, if you are an admin, then connect this layout with all our views, and here I am saying

00:20:28 hmm, when will the lesson of else run when the student logs in, then for that what do I give here that brother, this is my layout, which underscore layout, this one that I had created here, you must be seeing it in the shared folder, okay, so in this way also you can do, you can apply the condition, but brother see, right now this condition will not work for us here because till now I have not managed the role here, means right now I am not making a big application in which there are users, in which we login, so

00:20:54 We tell there whether the brother is an admin or a student, so that's why he won't do it now, I'm just explaining to you here how you can define the layout through the condition, you can apply it, okay, so this was the concept here, which I was telling you that you can select a layout view conditionally in the view start file, okay, it's clear, till here our lecture has been wound up, so if you liked this video, then do subscribe to my channel and along with it, press the bell icon as well because there are more

00:21:20 Many videos are going to come in this lecture series, so if you have any confusion regarding this lecture then do ask me in the comment box. See you insha Allah in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are having a lecture series of speed.net course 6, so in today's lecture we are going to talk about passing data from controller to view, brother this is a very important concept because if you are working with asp.net 6 application and making any application, then this concept is implemented in every application, means the data can be any, we need to pass it from controller to view, and this task can be achieved by multiple triangles, so

00:00:37 We will discuss all those methods in our lecture series, so today we will see the first method of how we can pass data from our controller to our view because look, whatever output or result the user of our website gets, where does it appear? It appears inside the view and what I told you, where do we put that logic? We put it in the controller. So, whatever will be the result of that logic, whatever will be the output, we have to pass it and get it displayed inside our view. That's why

00:01:08 Here we will discuss the concept of passing data from controller to view. Okay, so before starting the video, a small request, if you are visiting my channel for the first time and you have not subscribed to my channel yet, then please subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our chapter number 17 and now we start our lecture. [Music] You see, passing data from controller to view, you have to make sir in this video.

00:01:41 You have to watch it further because this is a very important concept which is done in every ASP.NET application. So brother, here I will tell you some points about it, after that we come to its practical aspect. So look here, in this dotted ambition application controller, typically the performance of the business logic of the application and the needs of the user, you return the result and the user through a view. Look brother, the thing which I just told you is what does the controller do in our MBC application?

00:02:10 Any logic that is implemented within our application means the logic of our application, we have to write it in the controller but whatever is the output of that logic, but for example, I have used ILS to create a logic, so now you know either this part executes or the other part executes, so what I am saying is that brother, if this part executes, then some different data should be passed to our view and if the other part executes, then some different result should be passed to our view, so for this concept, we

00:02:38 We do that for passing data from controller views, there are some techniques which we do that brother, if I give you a real life example like if you are making an application, a credit application means an application in which we can insert data in our database, can also update it, can also delete it, so often you must have seen that whenever you delete data, then you get a pop message in which it is written that your data has not been deleted or your data has not been inserted or updated, so how is this happening

00:03:07 What happens is, look brother, we put its logic in the controller but after applying the logic, its pop message which we have to display in our view, we pass it from our controller to our view and then we access it inside our view and get it displayed, okay, so from this thing you can guess how important a concept it is, okay, so now look further, what is being done here, you can do it by following objects to pass data between controller and view, means, as I told you, there are some techniques through which

00:03:36 We can pass our data from control to view, so here there are some objects, using which we can pass our data from controller to view, okay, what are those objects, I mean, what are the techniques, so look here, first of all there is view data, okay, after that there is view back and then after that there is team data and then in the last we have strongly typed view, it means we have these concepts, using which we can do what, we can pass data from our controller, where inside our view because brother, look, I

00:04:05 I told you that who manages the head of the controller inside the controller? Action methods manage it. This means that we will generate some data inside the action method and then we will pass it inside the view. So this is basically a concept, but action methods are inside the controller. So what is the name of the concept here? Passing data from controller to view. So in today's lecture, we will first talk about view data and after that we will discuss these other three concepts. So look here, first of all, we will discuss view data.

00:04:33 So related to that, I will tell you one first point here, look at view data, it passes data from a controller to a view, brother, the thing that I just told you about the view data term, we do that, if we have to pass our data from the controller to the view, then we use the concept of view data, so brother, before telling you the rest of its theory, I will tell you its practical and before the practical, I will explain its syntax to you, okay, it is very important to understand the syntax, so see what D is doing here

00:04:58 General syntax of view data is as follows: So look here, first of all you have to write the view data like this and after that you have to open the square brackets and inside the double codes you have to define its key, which means you can also make it a variable, because the data that is stored in the view data is stored in the form of key value, so whenever you write view data here, you will have to define its key inside the double quotes here, in this way we name the variables, in this way you have to define the key.

00:05:26 It will be there and then you will store any value inside it, so after storing the value, when we access that view data, then how will we access it with the help of that, then see here what is it doing, so look here, there are two things, and the value is there, so what is it saying in the key, see that this is a string value to identify the object present in the view data, means what does it do, brother, whatever data is stored inside your view data, to identify it, we can use a unique key here.

00:05:54 Let's define it and what we do through this is we access our data, so this is done and what is it, look it is a string value, you have to remember it, it means it is natural that you have to keep your screen inside double codes, okay then look at the next point value, what is it doing here, value and what is it doing in the larger form of value, look at the value, this is a d object present in view data, this object has a b string object list, and different types, integer, character, flow, double, date, time, except here.

00:06:21 What I am saying is that you can store data inside this view data, it can be of all kinds of data types, look here it can be string as well, means there can be string type data inside the value, it can be an object, list, means there can be a collection, oh it can be any single value as well, in the same way I am waiting for care floor double or date time etc, means the complex types, you can also get them stored inside your view data, in the complex type there are classes like this because

00:06:48 Date time you know is a class and if you create any custom classes, you create your own classes, then you can also store it by creating its object in this view data, okay so now let's come to its practical here, which so look here I have opened in my Visual Studio 2022 and look here I have opened the same project which we had worked on in our previous lecture, look here what was its name Controllers and Actions, inside which we talked about controllers, we talked about views inside it, okay

00:07:12 Today we are talking about where we have to pass our data from the controller inside our view, so brother, look here, I had created a controller, whose name was Home Controller and inside it, I had created some action methods, so look here, there is an action method named Index, what is it returning, which view is it returning, look here, I have opened it, look, we had done some work inside this view in our previous lecture, okay, so now what should I do here, look here,

00:07:37 But let me create some data. Okay, I will create some data here and pass it where in the view of my index. Okay, so how will you pass it? What did I tell you, first of all the technique is view data, so view data, first of all I do it here, so look here, I say view data, okay, its capital and then the capital is like this, you enter the data here and what did I tell you its syntax, you have to open the square packet, you have to name it, look, I have to name it here, data one, you something

00:08:04 You can also name it like this, what you want to keep for the variable depends on you, okay, then after that inside this, for example, first of all I store some string data, okay, what do I give here in the string, but for example, program matter, I have given the name of my channel here, program, okay, then see, again in the same way, I copy it, paste it below and I want to name the second view data with this, data you, and here, for example, I pass any wait value, but example 25, okay and again in the same way I

00:08:30 I come here and push it down again and inside this, I pass example data three, okay and inside this, see what I do here, okay, I say date time here, okay, date time class, I am doing that here which is related to date and time, we do that, okay and then date time dot I do that here, no because this property gives the current date and time by getting it, no and then I am converting it, okay, what am I converting it into, see here, what am I doing here, look

00:08:55 Long date string means here I am saying that brother I need the current date and time but in which format, in long format, what happens in the long format is that the name of the month is also displayed to us, so look here, here I have created three view dates, okay and look inside every view data, I have stored data of different data types and brother look, when you have stored data inside the view data, then what is special about it, can you access your view data, where

00:09:23 Inside our index view because we are understanding this concept that you are passing data from controller to view so look, I have created this data now you could have used this data in the logic of any A4L also, means you could have done it in this way that brother if the IF is executed, that means its body is executed, then this view data should be run and if the else part is run, then this view data should get executed with us, okay, so right now I have not applied its logic, I have just written it here directly like this, so now in

00:09:52 I access all the view data here. Okay, how do I access it here, below this paragraph? Okay, look, I copy it because the data is stored inside it and I will access it through this, but now look, here it is being displayed to us in a simple tax form, so what do I do? I will put @ at the beginning of it. @ represents the Razor syntax, so here you will have to access it by using the Razor syntax. Okay, again in the same way, look here, I will copy it.

00:10:23 Let me do it, I will paste it below, I will paste it here as well and you see, data you are given here, I will give data three, okay and brother, look, in between these three, I will also put a B here, okay so that we can have line bricks, okay, so look here, so now what do I do, I will run this and I will show you its result on the browser, so let's come here and run it, so see, our application has been run and look here, what had I stored inside data one, inside program data you, the value 25 was integer.

00:10:50 and then look inside data three, look here, I had stored the long date format after converting it to string, so look here it is telling you the name of the date as well and here it is also telling you the name of the month i.e. April, okay, so what can you do in this way, look here, okay, I will stop this and here this is in the controller, okay, look here, I have stored data of different data types in my different view dates, well, notice one thing, what I am doing now, look here, this is my action method

00:11:15 There is also an index and right now I am getting its data inside the index view but going forward we also have some such concepts that brother if we have to pass our data of our controller through the index area method to any other view i.e. inside about or contact, then for that concept also we use a concept, I will tell you about it later, so till now you should understand that here we can store different types of data and brother look, not just single data type values, we can also store errors or collections inside it

00:11:45 How can we pass it, look here, I create another view data here, okay, I create another video data, look, I create another view data here, it should be like this, okay, look and I keep it here, but on the example data, okay, and inside this, I pass an array here, okay, or you can also pass it directly here, or else you can make a separate arbi, how can we see here, I make a separate curry, okay, here I create an array and I name it AR here, okay and for this

00:12:09 Look inside here, I put some values. Okay, I put the values and I put them in the values like this. Okay, here I put the name of the student, for example, Kumar, okay. And then what do I put here, Prem, okay. And then what do I put here? These are the three elements that we have, which I have stored inside Eric. So now look, now this is what I pass inside my view data, okay, right? Look here, what did I do? I created this here and named it and

00:12:36 Then I stored it here inside this video of mine. Now how will we access it brother. Look, whether it is a collection, what do we do to access it? We do a loop of H for that. So look here, I open the manager syntax for multiple line block and inside this, look here, I write forest and press tap twice. Then what is it? Look, the entire syntax gets opened and now look here, what is the name of our collection? Look, collection means what we had stored in our view data.

00:13:02 I will insert it here and its name was data 4. Okay, but now look, what is happening here? An error is happening. Do you know why it is happening because, I see, whenever you store data inside view data and try to access it like this, then you have to do type casting here. It means, brother, look, inside this also there is an array of string type, so the error is coming here because it is not able to recognize its data type, it is not recognizing the data type, so that's why

00:13:29 View Data Whenever you do that and type collection or complex inside it, that means if you do that with any class created by you, then what do you have to do here, type casting is mandatory, okay so how will you do the type casting, look here, I have created a string type, oh so look at this, I opened the bracket and look here at the beginning of it, what have I done, inside the parents thesis, here I have told that brother what is this, there is an error of string type, okay, it's done, it's correct, so mean look here, this view

00:13:55 This is the data, which one will return an error? Hey, look at this which I have created here, okay and where will I send it one by one inside this item variable of mine and now I am getting it here, okay, look, I create the h2 tag here and inside it I first put @ reserve syntax and after that this item variable, this means how many times this foreach loop will run, it will run thrice because inside our view data there are three values stored and they will be displayed one by one inside s2 here, okay, so now I am going to send it here as a

00:14:23 If I run it then you want to see the result. Look, this result is visible. Kumar Prem and Jain, from where is this coming? This is from our string array. So this is one concept. Then look, whatever class you use to create the generality of the channel list, okay, so we can also store list type data inside our view data. How, look here, I create another video data here. Okay, I create another video data here and I want to name it data five here. And look inside this, directly I have the data.

00:14:52 I am storing it, okay, how, by using the list, and which list is this, it is a generic list, by using it and what I am doing here also, I am using the string only, okay, string, here I am creating a list of string type, okay, and look here, its body is given here and inside it I pass some data here, but for example, I pass the name of India here, I say cricket, put a comma and give the second value here, but for example, football, okay, and I give the third value here

00:15:19 Hockey So what I did with these three values was I stored it inside my list object and passed it inside my view data. Okay, so now let me access it like this, in this way I opened my razor syntax. Okay, but let's open it here. Okay, it's here. Okay and what I do here is I copy it in the view data like this and the value that was inside it, I pass it here, that was data five, which means what we had, okay, see, there is an error here as well. Why is there an error here as well?

00:15:50 Because the data here also is of complex type, that means I am creating a collection using the List class and I have stored its data here which is not getting recognized, so for that what do I need to do here, type casting, okay, how, here what will I do first, I will open the parents and after that see here is the list, okay and inside it like this I say that brother, what do we have of this string type, it is an object, okay, so what have I done, I have done its type casting, see,

00:16:15 The error was there below, it has gone away, you can ignore the warnings and brother, look what is the concept of type casting, brother, look here, I am converting one data type into another data type, but what is happening here is that I am converting this view data object, inside this, inside my list, of which type is the spring type, so this also will go one by one inside the item, okay, so now I am placing the tag of h3 here, okay and inside h3 here, look like this

00:16:40 Item okay, it is necessary to put @ otherwise you will not be able to access this data. So now let me come here and run it, then you can see its result. Look here, a message is coming, which means an error is coming. And why is there an error, I will tell you here, look, this is the closing curly bracket for it and you have to put a semicolon here. Okay, simple, let us do it again. So look, you can see the list which I had created as a string type, its result is being displayed here. So what does it mean?

00:17:04 Inside view data we can store multiple types of data whether it is array, string, date time, integer, string or any other data type, you can store it in the view data object. Now the rest of it is up to you. So look here, in the first point I had already explained to you that this is what we do to pass the data from the controller to the view. Then look at the next point, what is view data? This is a dictionary of objects. It is derived from the view data dictionary class, meaning this object

00:17:31 From which class is this derived? View Data is derived from the Dictionary class and is this a dictionary of objects, which means you can also call it a collection of objects, which means we can store data of different data types in it, which means we can reduce the view data with different data types by using it and the reason for using view data is that we have to pass the controller's data to the view, then see the next point some of the characteristics of your data. Follow now here some information about view data

00:17:59 I am telling you the characteristics, so look, first point the life of a view data object exists only during the current request, brother, look this is a very important point, understand this point carefully, that means here we are talking about the life of the view data, that means how long will its life be, brother, look what we were doing right now, we were passing data from the index area method of our controller to our index view, so till here I was doing it successfully, but brother, look, if I had to pass the view data which I had created in my index view, then

00:18:29 If I access this view data in any other view, that means if I access or contact it in the view, then I cannot do this work with the view data. That means the action method is from the view and the view is also from the view and you are passing the data of your action method to the view, then it will reduce till there, but if after accessing it in the index view, if you try to access it in any other view, that means from the index, if you go to the about view, the read happens directly from your index view to the about view

00:19:01 So it will not be there in the about view, so you cannot access the view data in any other view. So look here, this point is telling that the value of view data becomes null if the request is redirected, which means the data inside this data becomes null if you read directly, which means if you are accessing the view data inside the index, if you access it in your about view, then you will not get any result there because the view data becomes null there whenever you read directly.

00:19:31 You do it, okay so this is the point, then look at the last point view data requires type casting, here it is complex data type, you can avoid error, look brother, what I am talking about, brother, whenever you store complex data type inside the view data, complex means it is listed or you have a custom class, then if you store data of complex data type inside it, then when you access it in the view, then you will have to do its type casting, that means you will have to do data type conversion, okay then this, the last and important one is, look here, what

00:20:00 What is view data doing do these not provide compile time error checking means view data does not provide you compile time error checking, compile time checking means when you run your application before that whatever errors etc. get resolved, where do they happen, they happen at compile time, so see what is it doing here, for example if you miss a spell which you would get other compile time errors, you get you know about the errors only at run time bhaiya see, what does it mean here is that when we create view data, inside that what do we do

00:20:31 So if you spell any key in your view data, please misspell it, meaning if you make any spelling mistake there, then you will not get any error at compile time, but where will you get the error at run time, meaning when you run your application and then you will get the error, like this look here, okay, so look here, I have entered my view and look here, here I had created a list, which I had accessed inside my foreach loop, so look here, now I have given the name of the key absolutely correctly, data five

00:21:00 But if I misspell it here, how, look here, I have spelled it three times, look here, and what have I put, I have put 'A', so brother, we do not have any key with this name, so look here, when I made a spelling mistake here, then what happened, there is no error coming at compile time, when will the error come, it will come at run time, okay, so look, when I run my application, then what result do you want to see, so look here, what did it give me, it gave me an exception, and what is the exception, look, null reference

00:21:27 Exception means what we have inside this object is null. Okay, if you click on view details, then here I will tell you the information about it, that null reference exception is coming and see, if I continue it from here, then the same error will be shown on the browser also, I will show it to you, see, it is telling you that brother you are getting null reference exception and look here, it is also telling you that brother where is it coming, you see where you made the spelling mistake, so it means the error is not coming at compile time.

00:21:53 But it is coming at run time but we also have some such concepts in which we can do compile time checking as well. Okay, we will also see that in the upcoming lectures. Okay, so here our lecture is winded up. So if you liked my video, then do subscribe to my channel and along with it press welcome as well because many more videos are going to come in this lecture series. So if you have any confusion regarding this lecture, then do ask me in the comment box. So see you Inshallah.

00:22:17 Till then in next lecture keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we are having a lecture series on asp.net code 6 and in it we are talking about how to pass data from controller to view. So in chapter number 17, I had told you that the concept of how to pass data from controller to view is very important and it is implemented in almost every asp.net code MC application. So there I had also told you about the concept of passing data from controller.

00:00:31 You view, there are multiple techniques in it, means there are multiple objects by using which we achieve this task. So, in the previous lecture, I had made you do its first technique which was named View Data, but in today's lecture, we are going to do its second technique which is called View Back. So brother, before starting the video, I would recommend that if you have not seen our previous lectures, then first watch those lectures, after that when you will come to this lecture, then this lecture will seem easy to you, so in the video

00:01:00 Before going further, a small request, if you are visiting my channel for the first time and you have not yet subscribed to my channel, then please subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our chapter number 18 and now we start our lecture [Music] So look here, this object which is called viewback, we use this, for passing the data of our controller to the view, we use this

00:01:33 In this way, just like we saw the concept of view data in the previous lecture, so brother, here, as I told you, look here, we have these things, that means you can use following objects to pass data between controller and view, that means we have following objects, these are basically these concepts with the help of which we can pass the data of our controller to the view, okay, one is we have view data, then after that comes view back, then comes tame data and then comes strongly typed views, okay, that means there are these techniques

00:02:02 To pass the controller's data to the view, look, we have already discussed the view data in the previous lecture, but in today's lecture, we will talk only and only about its view back. So, you have to make sure that you watch this video till further notice so that you can understand its complete concept. So, look, in the context of view back, I will tell you one point and what is that point, look here, view back is also used to transfer data from a control converted to view, that means, the way we passed the view data to it, right?

00:02:28 In this way we also use view bag to transfer data of controller to view, but you will understand the difference between view bag and view data yourself when I end this lecture. So first let us come to its practical, I will tell you about its syntax in detail, then let us come to the rest of its theory, so here, which is okay, so look here, what is its syntax, what will we do with it in practical, look here in the syntax, first of all you have to write here

00:02:53 View back, which is an object, after that you have to write dot here and then after that what is the name of the property, brother, see, the name of this property is like this, just like we had taken the concept of key in view data, means whatever property you create with view back, you will specify its name yourself, means it depends on you what you name this property, in this way, in view data we had seen that we could name our key anything, similarly we can name our property anything here, but the key that we used to specify there was

00:03:25 We used to do it inside the square packet and inside double quotes but look here what is the syntax of view bag, first you have to write view bag then put a dot after the dot then name any of your property and then what you have to do inside this property is you have to store any type of data, that means you have to store values, so now this value can be of any data type, okay, so look here where the property is doing this, this property name is telling in big what is the property doing

00:03:51 String value data represents a property of view back means what is this property name, it is a string value but you will not write it in double codes, what will you do directly, first view back dot and then directly the property name, for example here I say view back dot next, so whatever happens next, you should not write it in double quotes, rather you have to specify it directly here after the dot and then whatever value you are wanting to give inside next, you can give that, it's okay and this property name, if you are using multiple view bags

00:04:19 So you have to keep the property names different, you have to keep them unique. Okay, then look at this, this property and then look at the value, I am storing this value in it, what is it doing in its value, this d value of d property of view back value b string object list, oh and n different types are true, wait character flow double the time extra, mean look at what this property is doing, the data that you will store inside it, the value that you will store can be of any data type, it can be string as well.

00:04:47 It can be an object as well, it can be an array or a collection, or any Scala data type like integer, character, flow, double, or date, time, etc. You can also store this inside this value in your view back property. Okay, so now first of all, I will come to its practical, then let's discuss its theory. So, I will come to my Visual Studio, look here, this is the application that we had created in our previous lecture, what is its name? Look here, controllers and actions and see.

00:05:13 In the previous lecture, we have our index area method in our home controller, inside it, look, I had reduced the view data. Okay, that means, by creating this data, I passed the data of my controller in the index view in this video, like this. So now what do I do, I am commenting out this star data from here. Look, now what do I do, I have written this star data user inside the index area method, after doing that, I am commenting out this star data from here. Okay, look, from here, take it here.

00:05:38 By doing control c I comment it out because now what I will do inside this index area method is I will do the concept of view back and look and look here, this Sir view data, we have data, I will comment it out now, let's select control c control c, okay, this Sir data, what happened from here, the comment is done, well notice one thing that the comment that I have made here, look what are the symbols here, see the comment is starting from here, okay and below here at A

00:06:06 Look, the comment is getting closed here, okay, so remember this, inside these views, you comment out like this and if you go to the education file, then there you can see, you comment out like this, okay, so this is the concept, now here, now let us talk about view back, so what did I tell you the syntax is that first of all you will write here, view back like this, okay and after that, I create a property here and I want to name it here, but for example, data one, brother, look, it is like this, in the same way as I told you about view data.

00:06:32 You can name the key anything, you can keep it unique, that means you name the variable in the same way, like this, what have I named my view back here, Data One, and inside it, look here, I store the name of my channel, but for example, if it's okay, then look here, now I do this and I access this view back in my index view and what is the name of its property, Data One, so I will access it by this name, so here, look, this h1 paragraph is at the beginning of it.

00:06:59 Here I say view back okay and after putting view back dot here what is the name of our property data one but now look it is tweeting it in a simple HTML simple text form what do you have to put here initially before view back @ which represents what represents our razor syntax so now whatever name of my channel that I had stored inside this view back, you will see it here okay let me do it with SMS I will show you its output in browser

00:07:27 If we run it, look here, our application has been run and look here, you can see the program and this is the same place where I had accessed my view back dot data one property here. So brother, look here, you can store data of multiple data types inside the view back, in the same way as we were doing in view data. So look here, I will create another property here. Okay, I will do it from here. First, let me copy it, come down and paste it. Okay and now here

00:07:54 I want to name it Data Tu, okay, I want to name it Data Tu and inside it, for example, I can store any number 24, okay, then come here below, I will paste it and here inside Data Three, for example, I will store any date time, okay, so here I will say date time dot no and after that what do I give here, short date, okay, means what am I doing here in short date time string, I am doing it, short, means our short of date here

00:08:24 I am going to tell you the format, what is the short format, like today 28, what do we have, date and which format is going on, fourth is going on, so the date that is shown in numbering, we call it short date, so I am converting this into string and putting it inside my view bag data three, okay, then here I create another view bag and inside this what do I do now, I pass an array, okay and I had created it in the last class, okay, so from here I am going to copy it from below, see this you

00:08:51 The error is visible, it is a string A, okay, I copy it from here and come here and paste it on top, and what is its name, Erica, look, A R R, so I store A R R directly inside this here, like this, that means, inside our view back, we can store Arabi, any collection, any list, that too, we can store in the object of this view back, okay, then I create another view bag here, okay, and I call it data four, data four has been done, I will create this

00:09:18 Let me say data five, okay and now what I do inside it is I put a list collection inside it here, okay so below, okay, here I had also created a list collection in the last class, see, you can see this, I copy it completely from here and I come and paste it here, okay, let's come here and paste it and I uncomment this in the code, control key control you, okay, mean look here, this data four, I had put a string inside it, but now look here, I here

00:09:43 I am entering a list and you can see what type of list it is, it is of string type and this one which was an error, it had names of the day before yesterday and look here, it has names of India, so look here I have created at least five view bags and inside every view bag, you see, you must be seeing different type of data here, if it is done then now I access all these view bags one by one here, okay, first of all let's do this, let's go to our view, so look here, I copy it, I put a line break below here.

00:10:10 Okay, let me put one on the line, let me do this here, let me access the data you, then after that again what I do here is I put one and now here we access the view back three here, that means we access the data three here, okay so this will be three, okay this three in which there is single type of data, but now look, now the view bag that we had here on the data, there is yer inside it and how do we retrieve yer with the help of a foreach loop, okay so here a which here now I have to write my foreach loop, okay

00:10:39 So first of all I will open the razor syntax here with curly brackets and here I say write forish twice and then do it, the whole syntax will appear and you see, where is the error stored, it is stored, look here inside the view back data, but brother, now you notice one thing, brother, look, when I did the view data here in the last class, then there was an error here, but look here, this time we are not getting any error, what is the reason for that, brother, look here, the reason for that is that this view back

00:11:10 After this, I am writing by putting a dot, data one, data two, data three. First of all, what am I calling it, it is a property, but it is a dynamic property. Dynamic property means that brother, I can put data of any data type in this dynamic property, and where will this check be done as to what type of data is there, this will be checked at run time, that means look here, here, as soon as I look here, wrote this view back dot data fold, look here, there is no error, that means we have to type in the view back key.

00:11:42 Casting is not required means we do not need to change the data type because the type of data inside it will be decided at run time means it will be checked at run time that what type of data is inside this view bag, only then we do not need to do type casting here. OK, so this is a difference between view data and view bag, you also have to remember this, okay so now here, I open the h2 tag, okay and inside it I access this item variable, and here

00:12:14 Coming to the beginning of this, I will put a rate, which means what is inside this view back dot data, oh, right? So, the one by one element in our error will go where inside the item and one by one it will get displayed inside h2 here , so this one loop will run three times because the error inside our view back has three elements. So, from now on, in the same way, I will copy this entire thing from here in the reserved syntax and paste it down and now what will I access here, look, data five, okay.

00:12:43 Let me show you what was inside data five. Look here, inside data five, we have our list. I mean, the concept that we have in the list collection, I have stored it here inside our data five. So, brother, look at the collection, we also have a list of multiple data. So, how will we access this also with the help of this for loop. So, look here, I have not done anything. What I have done is simple. I have applied a forish loop. I have copied that one and inside this, I have made this from data four to data five. So, look here.

00:13:10 Also you don't need to do type casting because with view back here you are using dynamic property, dynamic means run time means its data will be checked at run time, it would not have been checked at compile time, so this is also a very important point of this, you have to remember, okay, so now I will run it and show you its result so that you can understand it better, when I run it, look this, you are seeing 24 , see from where this 24 is coming, look here, let's go under your control, see where did I put 24

00:13:39 You had stored it inside view back data two, so here 24 is coming from here and then look here in the data three that was there, I had got my date time dot not here, that means current date and time, I had got it here, but in short format, okay, so look here what is the short format, look here it is the 4th month , 28 is today's date and which year is running, 2023 and then after that look what was inside data four data four, look here we had this are, inside which there were three values, so look, these three values are getting printed here, then this which

00:14:08 We have data five, inside data five, look here there were names of India. So I am accessing it here like this. So what can you do in this way, you can implement this view back. So first of all, what did I tell you the difference between view back and view data? Look here, I stop it. And here, inside this index, what did I tell you that in view back, you do not need to type cast it, meaning you do not need to change its data type, but in view data, we have to do this.

00:14:35 I had to reduce it means there is no need to do type casting in the view bag, but it is there in the view data, well but let me tell you one more interesting thing brother, look, this view and this view data, we can do it interchangeably as well, now what is the concept of this brother, look here, look here, what do I do, look here, I create a view data, okay, I create a view data here, okay, what am I creating, I am creating a view data, so understand what I am saying carefully, okay, how do we create the view data, look here and

00:15:03 Then after that its key comes, okay and inside it I store it here, its name is called here, mean, okay and inside it I store it here, but for example Adil, okay so see what have I created, I have created view data, not view back, view data has been created, but how will I access it inside the view, with the help of view back, okay so notice here what have I named my key, me name, so now see how will I access it, okay so come here, look, I have gone to the top, okay here, a jo

00:15:30 Okay, come here, I will call it view back, view back and what did I name my key here? I will define it as my property here, see this way, okay, did you understand the concept, mean look here, I had defined view data and what did I name its key, my name, and what value was stored inside it, Adil, but now how am I accessing this view data in my view with the help of view back and the key that I had kept there, I defined it as a property here, then

00:15:59 Look here, this is my name, View Back. After this, I will also put a B here, so that we get a line break. Let's do this. And now coming here, I will run it again. So, look what you are seeing on the top, Adil. And how is this heart looking? I had created the view data in the controller, but in my view, I am accessing it here with the help of view back, so I am getting the result absolutely correctly. And in the same way, if I create a bag here, that means here, where I had created the view data, here

00:16:28 Now what do I do, let's create a view bag, okay so here, let's create a view bag, okay and what do I give it here, data 6, okay, I give it data six and inside it I put some value, but for example, here I say welcome to my channel, okay, but look what have I created, I have created a view bag, but now how will I access it inside my view, see what do I do here, I create view data, okay, I create view data and what is the name of this, what should I keep here

00:16:54 Hmm, look, this way the data 6 is correct, that means look, I had created a property here inside the view bag, okay, now I am passing it in the view data like this, you see, and here also I put a line break, okay, so now I run it here, you see its result, see, you are getting absolutely correct result, welcome to my channel, that means I had created a view bag in the controller there, but accessed it with the help of view data, okay, so this is also an important point with reference to view bag and view data, okay

00:17:22 Ok then there will be a small practical on top of the rest of it and inside the theory, so stay with us, okay here, okay, you must have understood this syntax, now let's come to the rest of its theory, okay, so look here, what is our next point doing, view back, this is a dynamic data type property of the base class of the controllers which is a controller based class, I mean, look what is our doing here, view back is basically this is a property for a dynamic data type, dynamic data type means

00:17:50 Whose data type is decided at run time, it is checked at run time, it is not checked at compile time and where is this view back property defined, it is defined in our Controller Base class which is the parent of all our Controller classes, all our Controllers, whatever custom controllers we make, which is their parent class, Controller Base and inside Controller Base this view back property is defined, that means look here, let us stop it and I come back to my

00:18:19 Okay, in the controller, that means look, this our home controller is a child class and I am calling it, this controller which is the parent, I am calling it the base controller class, what am I calling it, you are seeing the base controller class, you must be seeing it below in the definition, now the base class Foreign MBC Controller with View Support, that means inside this we have that view bag, only then we get the support of view bag view data justice inside our controller, okay then further, the next point, now look here, what am I doing, view bag is this

00:18:46 Dynamic data type which internally uses view data to store value, brother. Now you must have understood the concept very well that view back internally uses view data only. That is why we can make interchangeability of the data of view bag and view data. That means we can access the data of view bag through view data and we can access the data of view data with the help of view bag. So this is also an important point to remember. Then what is the next point doing here, see, view back exists only on current request and becomes null if it is not.

00:19:21 Request, look at it as read directed, there is similarity, I mean this point, I had also told you at the time of view data, this was the same point, it means there is no difference, this view bag and view data are one and the same thing and what is that the view bag remains alive for its lifetime only for the current request, I mean current request, I was just passing my data to the index view through the index area method, so current request means current page, index page, so till there, our view bag will remain alive, but

00:19:53 If I get redirected from index and go to about or from about I go to contact then I will not be able to access the view bag in about and contact because when you get redirected, when you go from one page to another and try to access your view bag on the other page, then your view bag becomes null there, which means the data that is inside it becomes empty, it becomes null, so I will tell you its practical in the upcoming lecture when we will fix all that data.

00:20:21 Then look, look at the next point, it is telling the same thing, blue bag is a dynamic property based on the dynamic features introduced in Shiksha 4.0, meaning, look, it is saying that in the Shiksha 4.0 version that came, some dynamic features were launched and in that we got the support of view back, that means the support of view back, this dynamic property was not there in Shiksha one, it was not there in you, it was not there in three, we got it in four, okay then the next point view back ten notes required type casting

00:20:49 When you look at that complex data type, what is the complex greater type of it, like this, we have a collection, a list, here we have some custom classes, so what is happening here is that view back does not require type casting, that means the data inside the U back is of any complex data type, then you do not need to change its data type, but in view data we did need to do type casting, okay, because what did I tell you, where does the view back get checked at run time, that means the data of the U back runs

00:21:19 If it is checked at compile time, then there is no need to type cast it at compile time anyway, now look here, there is a note, this note is the same which I had told you at the time of view data also, see what is it doing, view back ten notes provide compile time error checking, for example if you miss a spell, you would not get other compile time errors, you will get to know about the error only at run time, look this is also a similarity in view bag and view data and what is it, whether it is view bag or view data, it does not provide you compile time error checking for this

00:21:54 That means brother, see how we access the view bag with the help of its property names, but if we make any spelling mistake in its property names, then the error will not be found at compile time, rather it will be found at run time. Okay, let me show you by doing this here, okay, look here, the view bag that I had created, you must be seeing data five, okay, and I accessed it here, okay, look inside data five, this four, this loop is visible, data five, but what do I do, look here, I

00:22:20 Here I make a spelling mistake in its property name. Look, like this I have put 3 A's here. So brother, look, with this name we do not have any view bag and its property. So this means, look here, right now there is no error here. That means you will not get it at compile time, but when you run it, then you will get to see that error. So look here, if I run it, then look, what exception is coming? Look, Null Reference Exception, Null, means inside this, you have passed the property here.

00:22:50 There is no view back property with this name, right? And if you look here, if you continue, it will take you to the browser and it will show you the result on the browser, what is the error and the reason for it. Look, it is telling you that in forish, at line number 30, an error is coming that brother, we do not have any property with this name in the control, right? And what is the exception, Null Reference Exception, when does it come when your application receives a null, right? So, if I look here, right, you can see here

00:23:16 Let me stop it, okay and here, I had placed these three A's, I am removing them from here, okay and running it again, then you can see it here, see, the application has run and see, we are getting our list by accessing it correctly here, so this is the point I was telling us here, okay then look, this was the last point, view data and view back can access other data, see interchangeability, brother, the thing which I told you just now by doing it practically, if you have put view data in controller

00:23:42 If you have created a view bag then you can access that view data with the help of view data in your view and if you have created a view bag in the controller then you can access it with the help of view data in your view, that means this thing which I told you, look here, let's stop it, okay, in home controller, look, this was the view bag data 6 , I created this view bag but how did I access it here, look here, with the help of view data, so this is the thing I am saying that when you interact, you can do this, so

00:24:08 Our lecture is over here, find out. So, if you liked my video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So, if you have any confusion regarding this lecture, then do ask me in the comment box. You will get the update in the next lecture, until then keep watching my channel program [Music]

00:00:01 Hello viewers, how are you all? As you know we have speed.net service running here, so in today's lecture we are going to talk about data. Data is a technique through which we can pass the data of our controller to our view. Even before this we have used some techniques like view data, view bag, this concept is also used by us to pass the data of the controller to the view. So if you have not seen our chapter number 18 and 17, then I will tell you that

00:00:30 First watch those lectures and then come to this lecture because in today's lecture I will explain you the comparison i.e. difference between View Data, View Back and Tame Data, so that you will understand the concept of these three very well. So before starting the video, a small request, if you are visiting my channel for the first time and you have not subscribed to my channel yet, then please subscribe to my channel and also press welcome because there are many more videos of this lecture series.

00:00:57 It's going to come in so this will be our chapter number 19 and now we start our lecture [Music] okay so time data is also used a lot whenever we pass the data of the controller to the view okay so let's take this lecture forward so see what is happening here you can do the following objects to pass data between controller and view means we have some objects using which we can pass the data of our controller to the view first of all we saw was the view data okay after that there was view back and today we will lecture

00:01:29 We will see temp data and after this we have a concept strongly typed to views. Okay, in the previous lectures we have implemented the concept of view data and seen the view back. In today's lecture we will talk about temp data. Okay, so you have to make a sir or watch this video till now. Okay, so we are discussing here in the context of temp data. So look, in the context of temp data, first point is a passage data from controller. You brother, see, this definition, I had told you in view data as well.

00:01:54 I had told you about view bag as well and I am telling you this about time data as well, it means that all these three things that we have, you data, view back, time data, have one thing common in them and that is that we can reduce all the three by using them, we can pass the data of the controller to the view, but I had told you the difference between view bag and view data, in today's lecture, I will tell you about time data, now first of all let's do its practical, okay after that we will come to the rest of it, so here, look, I am in my studio

00:02:22 I have gone and look brother, I have opened the project in which we were working in the previous lecture, so look here, inside the index area method, I have some code of view back and view data which we had done in the previous lecture, so look, I am commenting it out from here, okay sir, this code which is inside the action, inside the index area, sir, I am commenting it out from here, control of control c, okay and all this code is there, I am cutting it from here, means this is our index

00:02:46 This is the area method, inside this, all the code that I have commented out for the data and the view back, I am removing all of it from here. That means, I cut from here and this is the page that we have, exactly like it, and where the class ends, I will place it here, so that if we have to copy paste something, then we can take that data from here. So, look here, now I have made my index area method completely empty. So, look here, we have three action methods.

00:03:14 There is index, there is about and there is contact. Okay, if I open the views of all these three here, let's go to home and look here, let's open index as well, open about and then open contact. Okay, so look here, inside index also, I had retrieved it. Okay, the view data, view bag etc., so sir, I will comment out this code from here as well. Okay, so that it becomes completely empty with us. Okay, we have this data with us.

00:03:42 What happened, the comment is out, okay so now here this, which is in the home controller and here we go to its index field method, okay, see what I told you that in today's lecture I will also explain to you the difference between all three, okay, that means in which scenario the 10 data is different from view bag and view data, I will also explain that to you, so look here this, which here I first create the view data, okay, I create the view data, how to create the value, okay, here I will name it Vrindavan.

00:04:07 And inside this I pass simple view data here, okay and after view back, here I want to name this property as data you and inside this I pass a simple string here, view back, okay now I create the time data brother, see, we create the time data also in this way, in the same way we learned to create view data here, okay so here, this which here I call 10 data, means here also we will pass the value inside, I want to name it data three and inside this what do I give here, tempo data and see in

00:04:39 These three, which means that we have this you data, it is a view, there is time data, where will I access these three? In my index view. Well, remember one thing, where have I created these three now? In the action method of the index and where will I access these three now? I will access these three in my index view. So, look, from here, I copy the view data and come here to my index view and here, before the A, I put @ , the syntax here and here, it is accessed like this.

00:05:06 View Data Or I am going to copy the view back from here, okay, let's copy the view back and it comes here, okay, first of all here I will @ and then view bag dot data, okay, then here on the third one we have 10 data, correct, so coming here I will copy it and coming here I will paste it, okay, I had told you what is the similarity in all three, what can we do by doing these three, we can pass the data of our controller to the view, okay, so what did I do now, these three here

00:05:34 I have created them in the action method of the index and what I am doing in the view of the index is I am accessing them, okay and also at the beginning of this, at the beginning of the time data, what do I put here, this @ reserve syntax, okay, correct, so I have got all three here in my index view and where did I declare all three here in the action method of my index, okay, so now coming here, I run it and I will show you its result, so look, this index, we have a beauty display and look here, you are getting it

00:05:59 We are getting the back as well as the data as well, okay, this means that here we have a similarity in all three, that what can we do with the data of all three, we can pass it from our controller to the view, but brother, look, we have already discussed you data and view back, today we are going to talk about 10 data, that means what is the difference in 10 data, understand this carefully brother, look, just now I said that I had created these three objects in the action method of my index and I can access all three of them

00:06:25 Where am I doing it? I am accessing it in the index view itself, but brother, if I am saying that I don't want to access it in the index view, rather where do I want to access it, but for example, I want to access it in about view, so inside the about view, I will be able to access only 10 data from the maths view, I will not be able to access the view data and the view back. So, I will show you this by doing it. Okay, here, let's stop it. Okay, so here, look, inside the index, we had accessed these three, I will cut these three from here.

00:06:52 Let me do that, okay I am removing it from here and then I will take it to my about, okay I will take it to about and after coming here I will paste it below, so brother, now understand the concept, look, these three that we have, you data view back time data, where have I declared these three in the action method of my index, but now I am accessing them, I am not accessing them in the index view, rather I am accessing them in the about view, so what did I tell you, view data and view bag are less here

00:07:18 We will not do it, here only which one will work is team data, because what happened here, I got redirected, what does this mean, I am not in my index view, rather I am in the about page, so in this view we will get the result of only time data, okay, so look here, if I run it here, then see, this is my index view which you can see right now, but where have I accessed the data, in my about, okay, so look, I say om slash about, okay, so if I come here and enter, then look at this

00:07:45 You are getting the result of temp data only and only, neither you will get the result of pay view back here nor you will get the result of pay view data here, so this is the concept of temp data means that within a single redirection, the object of temp data is accessible, okay, this is one point, now the second point is that temp data is accessible to us only one time, it means that I have accessed it in the background, it means that now I cannot access it anymore, so what did I do now

00:08:18 I got redirected from my index to about, so I accessed it there, but now if I want to access it again, then I cannot access it, it is only accessible one time, like this, look here, I understand you, see, I come here and stop it, okay, look here inside about, I had accessed these three here, right, so I copy it from here and I took it here to the index, okay, so now look, where is this view that I have created

00:08:44 Data view back time data, I created this in the action method of index so what I am doing now is I am accessing these three in the index view as well and I am accessing these three inside the about view as well but what did I tell you when I access the time data here, that means when I access my time data in the index view, then now I will not be able to access it again inside the about view because that data is accessible only one time whether you access it inside the current page or

00:09:14 Then you can access it on any second page, correct, but this will be accessible only one time, so if I come here and press run, then you will see its result, see, the you data is also being accessed, the view back is also being accessed, the time data is also being accessed, so see, I have made a forward shift of the time data, so now if I go here in home about, it is okay because I am accessing my time data above home about as well, okay, but look here, if I come here and press enter, then look, now you will get the time data

00:09:42 I am not getting it because I have already accessed the time data on the index but if I do not access my time data on the index and directly come on about then I would get my time data on the scout. So this is the concept of time data. If you practice it a bit, you will understand it better. So now let us come to its theory. I will stop it by coming here. Look, you must have understood the first point. Now look here, what are you doing here?

00:10:10 Data is used only on current and subsequent requests as it is a very short life instance, meaning look what is happening here is that where can you access this data in current request or in subsequent requests, current request means brother either you can access this data in the current page or you can access it in the inside page but once you have accessed it once, you cannot access it again in any redirection because look what is happening here, its life is very long

00:10:43 It is more short it means you can access it only once either you can access it in current request or you can access it in read direction but this can be done only once after that this instance of it becomes dead gets killed because as I told you this is a one time access then see next point redirecting it only allows one user to relay on time data means what I am saying here is that brother if you have retirement of read direction then you can access time data because

00:11:13 What did I just show you in the beginning, these view bag and view data get killed in read directing but the data is only accessible one time, brother, look, this is such a concept, like if we make an application and if we delete any data there, then the message that the data has not been deleted is shown only once, it does not get shown repeatedly, so the concept of showing it once, we can do it with the help of 10 data, okay, I would explain you one more concept here

00:11:44 I am here, so look here, right now I have accessed all these three objects in the index and also in about, so now see what I do, from here, I am cutting these three objects from the index, I am keeping it in about as well and now I want to place it in contact, okay, where am I placing it, it is in contact, right, so now if I come here and run it, then you can see its result, okay, so look, right now we are not seeing anything in the index page, I have left from here.

00:12:08 Okay in your home about, home about, if I enter here then see, the view bag and view data are not getting displayed, only the tame data is getting displayed because as I told you that tame data is only one time accessible, whether you access it on the current page or access it on another page by redirecting, so see where I come after redirecting, I came to about, see I am getting the data here but if I go from here to contact, because inside contact also I have accessed my time data, but if

00:12:35 If I go here and enter, look, you are not getting the time data in the contact page because this is only a one-time access, but now what I am doing is I am trying to access it for the second time, so here I am not getting the time data. So this is a very important concept, you have to remember it. So this is the second point. Then look, thirdly the directing current request is scaled and the new request is created on the server to serve the redirected view. Meaning, what I am doing here is that whenever we redirect

00:13:03 So what happens is that the request for the current page gets killed, becomes dead and a new request gets created on the server which does what our redirected view does, it gets it displayed, okay then look at the next point sharing data between controller actions are done through d asp.net mbc time data dictionary, means look here it is saying that if you want to pass and share the data of the controller to the view, then for that you can use this concept of 10 data, then whatever happens next

00:13:31 Now look what is being done here, Tam Data is a dictionary object derived from the Tam Data Dictionary class, what does this mean, time data, you have a dictionary object, dictionary object means the data inside which we pass that value inside, and from where has this object been derived, what has this been derived from the Tam Data Dictionary class, and this is an object of time data, you can put any type of data inside it, in the same way, as I told you in the view bag, in the view data, that you can put time data

00:13:58 You can put integer data inside the object like character, float, string or in the collection we have a list, okay, or date, time, etc., so you can store all these things inside the time data as well, okay. Then look at the next point, your data stores the value of data, just like we used to store data inside blue data, in this way we will reduce the value inside your data as well. Then look at the next point, now see what your data value is doing here, it is a great type of before, you look at it brother, this is the same point.

00:14:29 which I also explained to you inside the view data means if there is any complex type store inside your time data means if any year is present inside your time data or any list means collection is present whether it is generic or non generic collection then that type will have to be typed when you access it, okay how to see here, okay look here, I stop my application after coming here and what I do here is I create a time data, okay I create a time data here, okay and its name is I want to keep it here, that means

00:14:58 I want to place it here on data, okay and inside it, look here, I access the data, that means I have a list which we had created in the last class, I am copying it from here, look, you can see this, inside it there are names of India, I copy it from here and bring it to the top so that our time goes back, okay and I uncomment it from here, control key, okay, so look here, you can see this, what have I done here, I have created a time data, what has I named it?

00:15:25 on data and inside this look here I have put my list collection here which is of type this is train type okay so now I access it now where do I access it and for example I am going to access it here how will I access it in my index in the same way as we have accessed earlier also look here you can see this razor syntax I copy it from here and I bring it to the top here here I will copy it and look here here I will access it

00:15:48 Okay, I am doing this with data, 10 data and what have we named its key, data here, so now you see the result, see what state has given me, it has given an error, it is giving an error because at compile time it does not know what type of data is there, so therefore, type casting will have to be done here, meaning data type conversion will have to be done in a way, I told you what should you keep in the beginning, put parenthesis here in the beginning and pass the list here and what type is our list, it is of string type, so

00:16:17 Inside this, what do you pass here like this, string, okay, means here I am typing that this time data, what type of data is there in it, we have a list type, okay, and now look here, this data that we had inside about, okay, I will comment it out from here, inside the control of control C contact also, I will comment it out from here, control of control C, means here I have only and only typed this here, okay, the time data is visible and what did I tell you

00:16:42 Before using the time data, it is necessary to type it. So if we run this here and see its result, look, you can see the result. This is the result of the time data, in which I had stored my zen blacklist here. It means that you can store any type of data in the time data, in this way, as we had seen in view data and view bag. Then here, this is one point. You see, here, the next point check if null is valid to avoid runtime errors. This means

00:17:08 Brother, it is possible that the object that we have, the tamp data, which we are using, may also contain null values. So what I am saying here is that you should check the null values so that you can avoid the run time errors, run time exceptions, so what I am saying here, okay, look come here, let me stop it, look here, for example, this tamp data can be null. So, what I do for that here, look, I have to put this loop inside the IF condition.

00:17:33 Okay and how do we create the IF condition here, look here I say that brother this is the time data, if this is not equal to you then what does it mean if it is not null then after this our for this loop should execute, okay, simple means here I am checking the time data whether there is null in it or not, if it is not null then only this loop will run for us, okay so look let's do that and if we run it here then you will also see that it will run in a perfectly correct way, okay look we have to run it in a perfectly correct way

00:18:03 It is getting displayed, okay, but if I pass null here, okay in this way, the time data that we have on the data, inside it, look here, I comment it out from here, okay, control C, now look, now I paste it here on top, okay and inside it what do I pass here, I pass null, okay, look, that means I am removing it completely from here and what am I putting here, I am putting null, okay, I am putting null and now look here, since I have checked this,

00:18:28 If it is placed inside the if condition, then what will happen in this case, the else part will run and inside the else part, look what I do here, for example, I display an h1 here , okay and inside this I write no data found, correct, I pass a simple message no data found and now if we run this here, then you see its result, see what the message is, no data found, why because what was inside our time data was null, correct, so in this way you can also do null on check, okay, so here I will write this

00:18:55 Let me increment it, sorry, I will comment it out, and from here, this data is there, I will comment it out from here, okay, control of control, okay, so this is one point, then now look at the next point, your data also passes data from the current request, you subsequent request during the request read direction, this point is telling here, can you use that data within the subsequent request as well, that means, I created 10 data, I created it on the index, but where was I using it when

00:19:23 Where was I going after getting redirected, I was going to about, so single time, we had this time data accessible, okay so this is one point of it, okay then further, now look here, its syntax, I told you, is like this its syntax is on the value and inside this where the key is, what does the key mean that brother, you have a string value which you define in such a way that it is unique through which you can do your time data, you can identify what is inside your value

00:19:51 Value we have an object subject means we can store any type of data inside our time data okay and if I explain this to you with a little diagram so look here there is a diagram look what is happening here that brother you make the first request like this and inside it you see here what is happening is that here I am creating tum data I have named it here mi data inside it the value passed test but look here I am accessing it inside about means in the second request look here I am accessing this

00:20:16 Where am I accessing inside about, so it will be accessed here because I am accessing it in one time, here I am accessing it in single time, so it will be accessed here but when I go from about to contact, then see, I will not be able to access this time data inside the contact because when this read will be done directly and when accessed, then the time data becomes null there, means it gets killed, it becomes dead, okay, so I have explained it to you through a diagram, okay then see, inside the time data there is a

00:20:42 And a very important concept and what is that, I will tell you now, see what it is saying here, tama data in this asp.net MBC can be used to store temporary data which can be used in these subsequence requests, subsequence requests means that brother, when we get redirected and go to another page, then we can access our temporary data there, but only one time, that's why it is being said here that inside tama data you can store temporary data, which should be accessible only single time, if you want to use it

00:21:09 If you access the second time then you will not get the data then look next point 10 data will be cleared after the completion of a subsequent request means what I am saying here is that when you access your time data then what happens after that, it gets cleared and gets deleted, okay but brother look, let me tell you an interesting thing, like what I have told you here, which was the diagram, okay, what did I tell you that you accessed all your data here, look in the second request, but when you

00:21:37 If you are accessing it in the contact or accessing it in the third request, then this time will not reduce the data, but brother, it is possible that you may have such a request in which you can access it in the contact as well, that means you are saying that brother not in a single redirection but I need this data in multiple redirections as well, so what do you need to do for that, look, for that here we have a point, I will tell you in the third point, see what is happening to this data dotted line method which gives the value of this data in a third request

00:22:08 I mean, look what I am doing here is that if you want to maintain your time data inside the third request, then for that you will use the method and where will you use this method and how will you do it, I will tell you, okay here, look, we have time data, what did I keep for it, data three, so look here, the index here, I am commenting out this code from here, inside it I had put the if condition of and this loop, okay, the control of control C, I am commenting it out from here.

00:22:36 I am doing this and look here inside about, look here this code is there, okay, I uncomment it, control of control you and then this code is inside contact, what do I do with this from here, control of control this and I can access this data in my index view as well, okay, to understand the concept, okay, I can access it here, look, these three things, view data view back time data, I have created, where is it in the action method of index, but now look, I am going to access this in the index view

00:23:03 I am accessing it in view as well, I am accessing it in about as well, I am accessing it in contact as well, okay, but what did I tell you the concept that if you access that data a single time here in the index view, then you cannot access it in about. Okay, if I run it again and show it, okay, so look here, 10 data is being accessed here, but if I go after getting redirected, I go to about, okay, look I am going to about, if I press enter, then look here, now here

00:23:27 We are not able to access that time data but if I look here, I say that look brother, I have accessed it in the index, I have accessed it in the index, so what happens is that after accessing your time data, it gets deleted, but if you want to keep it, then what do you do, you use the method of tam data dot cup here, okay, what will happen with this is that brother, look, this time data that we have, will be maintained even after it is accessed in the index view, it will be present, so when this is here

00:23:56 So it will be maintained, so I can access it in my about action method, that is, in the about view, here. So if I come here and run it and see the result, see, right now the index page is running and look here, the temperature is one, but now if I go from here to home, go to about, okay, so look, if I go to about and press enter, then see, now the time data is accessible to us here, but now look, if I go from about to contact, okay, then we will not get it on contact, look here, I press enter.

00:24:26 If I do this, look, we did not get it in the contact because we had not clicked it inside the about action method, so if I stop this here, look, and this tam data dot cup, I pick it up from here, okay and coming here, I paste it inside about here before my return view, like this, so it means that our time data, which both the time data, okay, means whether we have this time data or this time data, okay, both of them will remain with us, just what you have to do is put it in these

00:24:56 You have to apply it on both sides. Okay, so look, in about also, I have applied this temp data dot cup here and I have to access it in contact also, so what do I do here also, I put temp data dot cup, that means, look, this is a simple concept, wherever you want your temp data, what do you apply there, dog, apply the method of your time data dot dog, okay, okay, and look, inside this, whatever time data you want to get here, you can pass its key as well, look here, how am I doing, I have to access data three here, so here

00:25:25 Look, I have put data three, okay now I will copy it here, okay, I will take it here in my about, okay, let me paste it here and I will also put it here, my time data dot cup inside it, okay, that means look here, I have specified that brother, I am only going to access this, data three, okay, so what have I done here, tum data and which one am I copying, I am copying we have data three, okay, so coming here, I will run it now, so

00:25:50 You want to see the result of this, so look here, you are getting the time data, text you are getting, okay, I have gone here on Om, inside About, okay, here also you are getting the time data, but if I go from here to my contact, okay, then look here, if we enter, then see, here also we are getting these 10 data, okay, so what can you do in this way, you can maintain the data across multiple views, but brother, look there is a problem here and what is that problem, brother, suppose that if I have 10 views

00:26:18 If you want to maintain your data inside then you will have to do this data dot app again and again which is not a best practice because brother, if you have any such data, any such state which you want to maintain across multiple views, then for that we have another best concept which we call session. Session is used in many programming languages which are used in web development, we get the concept of session everywhere, like this in sp.net course also

00:26:49 We get the concept of session, so if you have this requirement, if you need to maintain your data across multiple views, if you need to maintain your state, then you should not use session, then you should use session. We will do that also later because you can use session where you need to access data only one time, it suits there, but if you need to do it again and again, then you will not use this method again and again, so what we do in this case is avoid it, so this is an important point.

00:27:18 Correct, so this was the third point that I just explained to you. So here our lecture has been wound up. So if you liked this video, then do subscribe to my channel and also press welcome because there are many more videos that are going to come in this lecture series. [Music]

00:00:01 Hello viewers, how are you all? As you know we are currently studying isp.net course 6 , so in today's lecture we are going to talk about a very important concept of isp.net course which is called model. Look, I had told you that by doing this isp.net course 6 , all the applications that we develop are all MC based, which means MBC architecture is followed in it. So, this M in MBC stands for model, brother, look, controllers and views, okay, that means model.

00:00:32 Views and Controllers In this we have seen the concept of views and controllers in our previous lectures but in today's lecture we will talk about model which is a very important concept. That means in today's time any professional web application has models in it. Once we understand the concept of model after that we have a concept of strong typing which we use to pass the data of our controller to the view, we will also see that.

00:01:00 I would like to comment that if you have not read our previous lecture yet and then come to this lecture, then you will understand this lecture even better. So before starting the video, a small request, if you are visiting my channel for the first time and you have not subscribed to my channel yet, then please subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series. So this will be our chapter number 20 and now we start our lecture. [Music]

00:01:31 So look, first of all I will explain the concept of model to you through a diagram so that you can understand its concept very well. So here, brother, look, what are the three core concepts in our MBC application? There is the model, there is the view, and then there is the controller. And what I told you that what does the controller do? It maintains all the interaction and communication that happens between the view and the model. That means if you need any data, then it sends a request to the controller.

00:01:57 The controller will take data from the model and display it inside our view, so what was mentioned in the above about the view, this is our user interface layer, the controller is for our logic, and whatever data has to be maintained between the view and the model, whatever logic has to be maintained, we do that and then we get the model, okay, what does the model do, the model performs handling with the data of our application, I mean all the data of our application, which we get displayed in our views etc.

00:02:27 Whose responsibility is it to manage the data, it is the model's and brother, look, this application data is the same data that mostly comes from our database, okay, but it is not necessary that the model's data is always from the database, what can you do, you can also manually type your data inside the model, you can do this because look, the first purpose of the model is the application data, brother, to handle and represent the data of whatever application you have, we call it the concept of model.

00:02:57 Okay, so I will explain the model here, okay, here it is, okay, so look what I told you, what is a model, basically we have classes from C, I mean, whenever you want to create a model, what is a model basically, C are your classes and these models, which are classes, what do they interact with, they interact with our data source, data source means from where the data will come inside our application, okay, so what are the data sources, the data source is power, look here, it can be a database also or

00:03:29 Then you may have a manual list or what you also call a collection, but in a large application, mostly our data source is a database, so I mean the concept is that the model that we have is education classes and there are such education classes that communicate with a data source, now that data source, what have I mentioned, can be a database as well or you may have a manual list or collection, so it means that if you have a data source for an application

00:03:57 If you want to work with the data, whether it is coming from a database or you are writing it manually, you will definitely need to create a model there. Okay, then further, now let me explain you some theory here. Okay, look here, first point, now look at the model is an class, the same thing that I just told you, what is a model, it is a class with dot key dot key, meaning education and what is this, it is an extension which is an extension of our class, meaning model is not a separate concept, what is a model basically, what do you do to implement it, education, you

00:04:26 You create a class inside the application, okay, that means if you want to create a model inside your sp.net code application, then what you have to do is simply create a class whose extension is dot key, so look, this is what you are creating, model is a class with dot key but it has an extension having both properties and methods, that means look what is happening here, model is a class, okay, inside which we create properties and methods, but what happens mostly, mostly what do we create inside the model that we create

00:04:53 Let's create properties now, but for example, I will explain you with an example, the concept of model and property, brother, see, suppose I am making a web application for school management system, I mean I am making an application for school management system, I am making a web application, so in that you know that we have a student, so I will represent the student with which model, I mean what will I create for the student, I will create a model and what will this student be called, basically we will have a model class called student, okay, so

00:05:20 What I did here to represent this is I created a model class of Student and inside Student, you know there are many properties, that means a student has many features, properties, like in Student, you know there is also a roll number, it also has some name, it also has gender, it also has class, you can create a lot of things, so that means whatever model class you will create, you will name it Student and whatever properties you will create inside it, what will those properties be, this one, okay.

00:05:46 In this way, roll number is done, name, gender, class is done. Okay then look here, models are used to set and get the data, I mean, why do we use model that inside the model we can set the data as well as get the data. Okay, just like I told you, inside the Student model class we will create properties. So basically what can we do inside these properties, we can set any data and also get the data. If you would have read the concept of properties in your education, then you must have seen there that

00:06:13 The keywords get and set are there so it is an indication that we can also set the data within the properties of the model class and from there we can also get the data and access it as well. Okay, then there is some more theory, after that we come to its practical. Okay, so here are some more important points, look here if the application does not have data then there is no need for a model, that means look what is happening here is that if your application does not have any data then you need to create a model.

00:06:42 There is no need for even that, meaning what I am saying here is that if your application has only static data, which you write inside your web page through taxes, then you will not need a model there, but this happens in very rare cases in today's modern era because in today's modern applications, the data of our application is inevitable, that means there is also database loss, so therefore retirement of the model is inevitable, okay, so then see here, what is happening if your application is lost

00:07:10 Give data, you need a model, which means if your application has data, then you will definitely need a model. Okay, then look ahead. Now see here, what is the model doing, in a speed dot net kar mc contains a set of classes. Now look at a very important point, it is not necessary to make only one model class, you can make more than one model classes. Just like I told you, if I am making an application for a school, which means I am making a complete school management system, then you know that inside the school, your

00:07:40 There are teachers as well as students, there will be courses as well as subjects, meaning there will be a lot of entities for which what I will do is I will create separate classes within the model, meaning within the modular we will have a student class, a teacher class and a subject class as well. So this means that all the entities that are there in your web application, we represent them with the help of model classes. So you have to remember this one point. So look, this is what you are looking for - D Model in HP.net and MBC.

00:08:09 Contents and set of classes that are used to represent domain data. Domain data is the same thing. What is our domain? School Management System. Whatever entities are there within it, we will represent them with the help of set of classes. So look, what is this domain data doing here? You can also use business data, which means you can also treat your domain data as business data. Well, it also uses logic against logic to manage domain and business data model, you can also apply logic, which means you can also implement any logic so that you can use the data of your application.

00:08:42 How do you handle it, maintain it. Then look at the next point, in simple words, we can say that data model in asp.net computer applications are used to manage the data. It means it's a simple concept that if someone asks you in an interview that what are models, then what do you have to tell them. These models are there so that we can manage the data of our application, handle it. Simple, okay then, now look here, what is happening if you are working with other weather application data is based on MC design pattern.

00:09:11 In MBCS application three things are common, that is model, view and controller, we all know this. Then look at the next point. Now look here, it is explaining all three things in big words, that means it is giving the summary of all three things in small points. See what it is doing. Controllers are used to manage the flow of the application. That means what controllers do is it maintains the flow of our complete application, it manages it because what did I tell you, this is the first time whenever a user requests your application, then

00:09:38 After routing, the request that we have received is passed to the controller and then what the controller does is it takes data from the model and gets it displayed inside the view and we have to write all the logic inside the controller only. So then see what the model is doing, it is responsible for the data and these data are used on views. See brother, I am talking about what these models are responsible for, the data of your application and what did I tell you that that data can come from the database or even manually, you can create a list or collection and use it

00:10:05 Where can you display that in your views? Okay, then look at the views section, it's telling me that views are basically HTML pages that get rendered in the browser of the client. So what is it doing here? Views are simply HTML pages that are rendered and displayed on your client's browser. Then look here. Next point, now look at what it's doing here, they can perform operations like insert, update, delete by using model. So, look at this, a very important point. If

00:10:32 We have to perform credit operations like creation, update, deletion, read, these kinds of operations, if we want to perform them in our web application with the database, then we will have to implement the concept of model there also. Then, look, the next point, this is basically a note, look what is it saying, it is not mandatory, meaning it is saying here it is not mandatory, but what it is is a good practice, you store the 3D model classes with in the 3D model folder, meaning, look what is it doing here that this is not mandatory.

00:11:00 But this is a good practice, it is a good programming practice. Where will you create all your model classes? Inside a separate model folder. Just like we were creating controllers in a separate folder, we were creating views in a separate folder, similarly, where will you create all your model classes? They will be created inside the models folder. But brother, look, I will tell you one thing now that till now we have not started the concept of database here, I mean, till now we have not read NTT Framework database.

00:11:27 The concept of Slava etc. has not been attacked in it yet, so in today's lecture, I will teach you to create a simple model and tell you how to generate and display data through it, but in the upcoming lectures, we will tell you how we will work with the database by doing the model. Okay, so now I come to its practical and I go to my business studio. Okay, look, I am doing it in Business Studio 2022. Here I create a new project. Click on it.

00:11:52 Another one, look here, the application that I will make here will be asp.net, look, web app model view controller, that means I am not doing it in empty, I am making my speed.net application with Ambit template, so let's select this, let's go next, okay, look, it is asking for the name of the project, the name of the project is, let's keep it here model in asp4, okay, this is what I am naming my project, okay, let's go next, after that, see which framework have I selected, dot net view, it has been done here, okay, this is what you have to select and

00:12:22 After that we have to click on create here. So look here our application has been created and you must be seeing model in hp so look inside it we have already created folders of controllers. It is already created inside the model. Some views are also created inside it. So now what do we have to do first? Look what do I do inside the model, I will create my own model. And I will create the same thing in the model which I just told you in the slide for the student. So look we have a by default here.

00:12:47 The model has been created which is for the error view. We don't need it. So let us leave it as it is. Let's create our own model. So what you need to do on the model is right click. Go to add. And what did I tell you, what is a model, we have simple classes. So in the new item, okay, so look here, you must be seeing this at the top. If you don't see the class here, then you can search it from here. So look here, I am naming it Student Model. What should I keep?

00:13:12 I am going to name it Student Model. So let's enter A here. So look, we already have a model class A by the name of Student Model. And look, I had told you that education is our class, so what do I have to create inside it, I have to create properties, inside which we will be able to set the data of our application and we will be able to get it from there. So, what was the option for this, that we create properties. So, the shortcut to create a property is that you have to write PROP here and then click on it, that means two

00:13:39 Once you do it then it will be generated like this. So first of all we have a property called roll number. So in this way the roll number which will be of the type 'wait' but after that again I come here and say 'prop'. And then here I am creating another property of string type and the name of this property will be 'name'. This will be for the name of the student. And then after this again I will say 'accused'. And then here I will say 'string' and this will be for gender. So here I will call it 'gender'. And then

00:14:08 The last property that we will have will be for the class. So, I will write the props and then do that. Okay, we will have the class waiting for us. But brother, look, I cannot name this property in the class because you know that class is already a single word. So, the name that I am naming it here is, it is called standard, it has to be studied in the sixth standard, it has to be studied in the fifth standard. So, related to this, I have named it here. So, this Google class of ours has been created there and I have created properties inside it.

00:14:34 Look brother, look, this is the model class, to generate its data, to get that, we have the best practice that we create a class using the reporting pattern, and what we do through that is we phase the data and take it to the controller and after that we get our data displayed there, but look brother, in today's lecture, I will not go over the repository pattern because if I go over it now, then you will not understand the concept of model properly, so we will explain the concept of repository pattern

00:15:06 We will do it in the next lecture, but here how will I generate the data for my model because right now we don't even have a database, but here what I will do now is I will generate its data manually by creating a list, okay how will I do it, look here, there is our controller, come inside the controllers folder here, look I have created a controller, home controller, I will open it and inside it we will have an index area method, okay you must be seeing it here

00:15:32 And a view of this is also already created, I will show you inside our MBC application, okay, so look here, you can see the views folder, I will expand it and look here inside home, you will see a file named index, if I open it, then look inside it, you can already see some data here, okay, but what I do here, look, I have created this student model, now I generate data using this, okay, where is this action of mine

00:15:57 What is the name of the method, index? So what do I do for that here, look here I create a variable, okay, and I want to name it here, student, okay, what should I name it, student, and look inside this, I generate a list here, okay, I generate a list here, and what type will that list be? The student model that I have created, it will be of this type, look below you can see the student model here, now when you enter it here, okay, whatever its name is, space

00:16:21 It will be added at the top here, look, you can see this, its name space, this is the name of our application and inside it I am accessing the dot model folder here, inside this we have our model class, what is its name, Student Model, okay, after that look here, after that I start the curly brackets here, okay, I turn on the curly brackets here and I put the cm and now I generate the data here, okay, how will I generate the data, look here, I am creating an object here.

00:16:47 What type of object of each of this will we have, look here, student model, means what I am doing, I am creating a list of students here, but I am creating it manually, okay, means this data is not coming from any database, I am creating it manually here, but when we read the database later, then we will search the data from the database, so brother, look, inside this I had created properties, right, which one did I create first, roll number, I will specify the roll number here, but example one, after that, what was the second thing inside the name here

00:17:13 Example, what do I give here Kumar, okay, Kumar I have given it and after that here at the last what did we have, standard, standard means class, so class, what do I give here, but example 11, okay now see, I copy this entire line like this, I am about three students late here, okay so look here, I change the roll number here, 123 and I change the name here, but example, what do I give here, I write my name, Adil and here, example what do I give, Natasha, okay, Natasha, do it here

00:17:43 Let's give it a try, one thing we got and that was gender. Okay, so I'm going to put the gender here after the name, look here there's gender, but here I'll change it's gender, for example, male. Put a comma here. Okay, then I'll copy this from here. Copy it and then come here and paste it below. This Natasha, what we have, is female. Okay, so here I've made it female. Let's change the standard of all three. Okay, it's 11. Let's make it 10 here.

00:18:11 okay, let me do it here. 12 okay, so see what I have done, I have placed my model here and by using it I have generated the data of my application here. okay, but the same point which I mentioned earlier that right now I am just explaining the concept of model to you, that is why I am reducing it like this here, but the best practice is that we create a class through a repository pattern and what do we do inside that class, we access our data source, our database, data access layer, and whatever logic is there,

00:18:39 That is, brother, how to access the data, we do that work there and after that we bring the data to our controller, but here I am wanting to keep it simple, so here I am explaining the concept to you in this way, this work which I am telling you, which we will do in the next lecture, okay, so look here I have created this, student, now I pass it, how do I pass it inside my view, which we had read about view data, okay, we had read the concept of view data here, okay, so I will name it

00:19:05 I want to keep it but for example 'me student', okay I am naming its key 'me student' and inside this, I have created a list of students here, okay so this student, I pass it here like this, okay, I have passed student in my view data, now I access this view data, okay where will I access it, this is our index.css team, brother, look here you can see the razor syntax, first of all, by taking the son from this view data, I will get it inside a variable here.

00:19:33 hmm, okay student, and inside this, look here, I have my view data, what was the name of its key? I get it as 'm student' and look here, I also do its type casting, okay, and in type casting, I tell you what type of data is this, this is a list, we have the model of the Student model, okay, look like this, mean, look further, what am I doing, what am I doing, I am doing type casting, because I told you that what we have to do here, is to type our view data.

00:20:02 If it is read then I am telling you that brother what type of data is there inside this view data, it is the list of student model, so by getting the data from here, I am putting it in the variable of this student, okay, now how do I access it, look here, I create a table, okay, I create a table, an HTML table, okay and inside this, first of all I create a row, okay and inside that I create a table, that means I create a heading, okay and inside that I specified the heading.

00:20:29 Whatever properties were there, I will get them displayed in a heading like this, first there is the roll number, okay, after that we have name, okay, after that we have gender, okay, gender is here and lastly here we had standard, okay, instead of standard here I can write class, okay, there is no show and now what I do is here I start the for loop, okay, so look here, like this, by putting @ , by applying razor syntax, look here, this for loop is gone and inside this, look here, I create a variable

00:20:58 STD is okay and from where will I take the data, see, I have created a variable above, from where I am taking the data, I am taking it from the view data, so I will copy it from here and pass it inside the for loop here, so see we have three students in total, so where will they go one by one, they will go to STD and I will do this ST here, okay, look, this one which is in the for loop, I have to keep it outside this side, okay, right, this side is only for this for heading, okay, look outside here.

00:21:26 What I am doing here is I am pasting this forest look here and now inside this, look what I do is I create this layer and inside this layer first of all I open a TD and inside this TD, look here, I pass STD and STD is an object, so inside this, what am I accessing first, I am accessing the roll number here, okay, then coming here, I am copying it and pasting it below, if there are properties, then I am pasting it, okay, after STD, here

00:21:52 Here we have name, okay, after that we have gender, on the third one, okay, and on the last one we have standard, so here, in this way, I have passed it, okay, that means, now we have three students inside this object, inside this variable, so which student is the first, look here, the first student is Kumar, then Adil, then Natasha, so mean, look here, first of all, Kumar's row will be added here and from here it will go to STD, so now, first of all, look here, I am accessing Kumar's roll number, I am accessing the name, General.

00:22:23 I am accessing and doing standard access, similarly the forest look will run again, Adil's data will come here, this entire row of Adil will be updated and the rate will be increased, so here it is decreasing like this, okay and if you want to give a border here then you can do that as well, if you want to apply any kind of thing like Bootstrap etc. then you can do that as well, but I have kept it just simple here, okay, so now I will run my application here and I will show you how I have phased the data of my model and got it displayed, okay.

00:22:47 So if I come here and run it, then you want to see the result, see, you are seeing the result, that means this index has also opened and look here, everything is visible because the index and look, this data which we had generated from the model and then passed it from the controller to the view and here in the view, we have displayed it inside the table like this, so this is the concept that you basically have of model, but I told you what is the best practice that we implement this, with the help of repository pattern, okay so

00:23:15 We will see that in the next lecture, so this lecture is done here, as a reminder, so if you liked my video, then do subscribe to my channel and also press the bell icon because many more videos are going to come in this lecture series, so if you have any confusion regarding this lecture, then ask me in the comment box, Insha Allah, we will meet in the next lecture, until then keep watching my channel Program Matter [Music]