

Since the back-end logic and overall design is of dynpro is based on object-oriented approach.

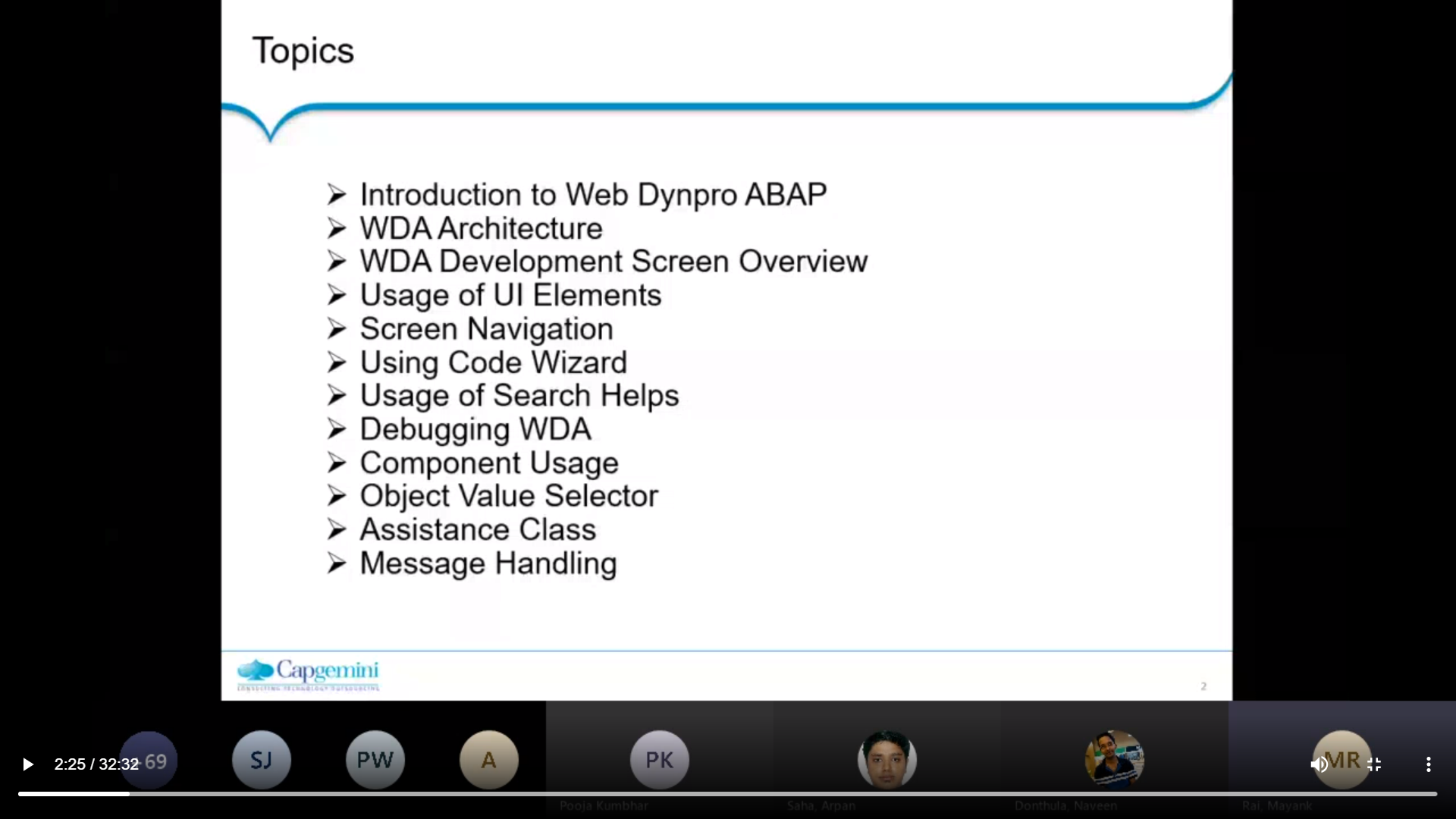
In methods, different types of methods and all so on that so basic knowledge of object-oriented ABAP and knowledge of co abap is requested for these sessions.

So, the topics that we would be covering in this session would be the Web dynpro, road map, introduction to the web app like how the.

App was evolved into web dynpro app and now.

We have evolved much and now we have you UI-5, Fiori, those concepts, and web.

Quite large applications and many are still dependent or built into the dynpro app. But the future is U F5 and since we still need to support urban publication and, in some cases, where they have not migrated to Hana and those things so we still have a web development getting so knowledge is must as of now.

So, the next topic would be the web dynpro architecture.

So overall web dynpro component, how it is defined in the app GUI and how it interacts with the web thing.

So that will cover then the web Dynpro app development screen like how we do the Web dynpro app coding.

Again, there are UI element concept in Web app, so we would be.

Uh, giving an overview on few of the mostly used UI elements so.

So, we would be giving knowledge of first a few of the elements, rest you can try and there are also some existing web control components standard web dynpro component that provides.

So, you can refer to those as well like how it behaves

That then, since it is a.

Web application there.

Still obviously we screen navigation from one screen to other,

how to write the code

code result in web app.

Uh, looking into that as well then since there are fields, we would be requiring some search helps like we have in the SDP screens.

So, there are some different approaches for search helps in web app we would have that then since many of you would be in.

Support project or maybe even if you are in development projects you might require.

To debug a particular scenario.

So how we can debug the web app even that we will be?

Covering then there's a concept of re usability in web component wherein we can use reuse an existing web component into another one.

So, we'll be covering that in component usage.

Uh, again, object value selector is a concept of search help, so we will be.

Uh checking on a basic component or application of that then we have assistance class wherein we can use a particular class.

Uh, which would have some specific logic which can be reused in multiple web dynpro components, so we'll be working on that.

Then there's the concept of message handling in web dynpro, how we can display or control the messages in a browser or.

Vibrant pro component.

So, these are our topics that we would be focusing on.

OK, so starting as I mentioned that from today, for today and tomorrow himanshi would be conducting the session.

So, I would be stopping from here and rest of the session would be continued by himanshi.

So, I'll stop the sharing.

And I would be giving hand over to himanshi so.

Thank you for joining.

Hey man, she he can continue from here.

Thank you, my uncle, I will share my screen.

Please let me know when you can see my screen.

Yeah, it's visible.

Yes, we can see.

So as Mayank has already given us a broad picture, what topics we will be covering in this set.

Uh so as he said that now as we have moved quite forward now the new technologies has come in like UI-5 fury which are playing a key role for a web-based designing, but there are still some existing distils and projects.

Which are relying on web dynpro and still some support projects where the knowledge of web dynpro is required. So here we are to learn the web how to use the web dynpro tool given by ASAP to design a web-based application.

Options. So why there was a need of web dynpro? Because when initially when there was no such tool given by SP to design a web-based application so.

Uh, there was a need came into picture that that there should be some technology which should support a browser-based application.

So earlier we were focusing, we were relying on the module pool programming to design views and controlling those view.

Uh. Which was?

Is a quite a code specific?

Uh, lots of programming it was required even to design a simple view layout and to handle the programming behind the view.

So, with the when the web.

Pro came into picture.

It quite nicely separated the data presentation and data processing.

So how?

How it was separated by web dynpro, how web dynpro?

Which on which architecture?

Web dental was uh relying into it.

So, it was mostly for web dynpro was designed based on MVC architecture.

Which is which being a very well-known market.

Uh, so what is an MVC architecture?

Web dynpro has a different it.

It has given us different nodes.

The one node we where we can design the view layouts, and the other node we can handle the logic behind that view layout.

And there's.

One node which will come in, which will make a communication between views and the logic behind those views.

That is the data migration between the view and the SFP back end.

So that's how.

Web in Pro was developed by SAP and using MVC architecture it was designed.

So, web dynpro for a baptism, SAP standard UI technology for developing a web application.

In the Web app environment, it is integrated with the SAP workbench.

So, in sea transaction you will find your web dynpro tool.

It's integrated in the chat.

A programming model for user interface finds a standard structure for user interface application.

It is derived from MVC that is model, view and controller design pattern.

So, these are the three nodes of this design pattern, that is model, view and controller.

So, when we.

Then we will start building the application.

You will clearly understand how the model, view, and controller work in case of Web Pro.

So here it gives us a picture to design our view layouts.

So, all the view screen designing is handled by view.

All the data declarations and the logic to fetch from the back end is controlled by the model and when the data is fetched from the back end and feed into the variables, they are binded with the view using a controller.

So, controller will communicate between the view.

And the model.

So, this is how the MVC works for web dynpro.

This set of tools for user interface design so.

Uh, as we covered in the as we said the web dynpro depends on most of the logic or most of the code is generated in web in through using code wizard.

So, we don't have to write large piece of logic for a certain.

Most of the things are generated by web dynpro itself.

If we write if we design A view, the code behind the view is controlled or the code is generated by web dynpro itself.

We don't have to manually write a piece of code to design an input field or a button, so the code is already created.

We just must place our UI elements in the screen, and they'll start.

Uh, we don't have the border about bother about their code.

All we must think about is how we're going to interact.

With that screen how the button click is going to work.

So, we have.

There is custom logic that on the button click I must show some popup or show some messages or must navigate to certain screen.

So that custom logic we must implement on our own.

But the basic screen, screen logics are given by Revlon through itself it is generated.

I built self.

So, code is generated using metamodel declaration.

Now our web application is.

Declarative based on the declarative programming language.

So, what is a declarative programming technique?

So before starting with the web dynpro, the first thing we must.

The first requisite we must do is have to do the declarations. For example, if like we write any C38 program, the first thing we do is to declare our variables internal tables, then we start writing out the actual logic.

Similarly, in web dynpro, the first thing which we create is the declaration of all the variables.

That's why we say it's a declarative programming technique, so everything is handled by the word declarations you have made in your context area.

Now what is context area?

We will see when we will move ahead, and we'll see the web dynpro screens.

We will understand what context area is.

So, for now assume that in the context area we.

Made we should we do all our declarations are variable table structure declarations and the code wizard or are all your web dynpro logic is going to work around all or the declaration is going to?

Process based on what declarations you have made.

That's why we called it as a declarative programming technique.

Now what are the benefit benefits?

Why should we go for a web dynpro?

So, as it said, there's a minimum coding and a maximum screen designing, so very less coding is required. All we must do is do the screen designing's go to the methods associated to the screens and the code wizard.

Which is quite good to be given by web improve?

It will generate a code for us automatically.

Just we must click on the code wizard and generate a code and few minimum coding or manual custom logic we have to put like a select queries of putting the data into those.

Tables so.

Lots of coding is being minimized here and we are mainly focusing on the design area.

We have separated the layout and logic as as in module.

The layout and logic was all together.

There was not.

There was no separate area to design the layout and rewrite the logic of the layout.

Here we have a separate screen to design the layout, and there's a separate controller to handle the logic of that layer.

So it it.

It is integrated to our back end system so we can we can communicate with the database easily and can process the screen.

We can process the data on the screen easily.

So earlier when there was no web dynpro into picture how the data used to get displayed in the browsers, there was no such.

Technology which which used to help to display data in browsers.

So what SCP used to do is to to write a program to generate a.

It used to write a simple program and the files or outputs were generated.

It was converted into some XML files.

It was sent to a UI developers, they used to convert it into, used to process those files.

Then in the form of using HTML they used to design the screens and process.

Our data which is given in the XML and we put it into a.

So there's two separate team required, an SFP team and a different UI development team.

Both used to play a role to process.

One is to process the data from SAP and the other used to get the data from SFP and displayed into the browser, but both.

By a web pronoun both this.

Tools have been integrated together and there's no separate UI team required to process or display the data in the browser.

Web in Pro is capable enough to communicate with the browser.

Support reuse of components.

Now let's say we design 1 component and functionality in the in web control and the same kind of functionality I need to.

Put it in three separate web pro components.

So why?

Why should I design the same view again and again in all my three different components.

So here it comes up.

The picture of reusability where I can design one view which needs to be reused in three different components.

So same view can be reused by these three different components which is our component usage which we called as an opponent usage in Web in pro.

Those are the advanced parts, so we will be covering it in the later session, maybe in day three or day four.

So there you'll find how how to reuse the components in web control.

Then configuration of UI patterns support web services and data binding as I said.

A web in pro can be is capable enough to run in multiple platforms like IE, Chrome.

So it is capable to run on different browsers.

So that's the high fidelity web info UI.

Oh, sorry.

That's like a main advantage of update through that is a browser based and 0 footprints and So what is 0 footprints when whatever user will, let's say our web application has been deployed and you can see it can look it into.

The browser.

So whatever user will interact in that browser, whichever buttons he is going to click, whatever data is going to enter, it is not going to be recorded somewhere.

Like there's no footprints going to be created for that?

Uh, for what actions user has taken?

So no trace as such created on on the system for what user has done.

Neither file is created nor or any.

Any repository is created for all the actions taken by the user.

So that's called 00 footprint.

Even web dynpro gives you 100% accessibility support even a blind deaf person can handle.

In application.

Uh, uh, Web Pro application. So if anybody is working on a UI, there was been knowing that even a new UI-5 also supports the accessibility features. So even in your mobile phone you can find your mobile phone supports accessibility so.

The same web Pro has also given us a feature of accessibility.

So you don't have to worry about how 20 or 30% of the users who are not able to see or listen to the system. So how they're gonna interact with it, it's because it's giving us a facility to it.

It's giving support to accessibility as well.

So these are most of the benefits of a web dynpro.

Let's, uh, move ahead.

Application scenario with a web dynpro.

So this is a broader view of how web dynpro is going to communicate, how where we gonna see the web dynpro application and how it will be, how it is going to interact with your database so.

This is your Web app server above database server.

This is your application where the web dynpro application has been built and this is our SAP net Weaver portal where we're going to integrate our web dynpro and this is going to cost.

The application in the browser.

So here are webbing for application lies from Sieti transaction. We will be designing our web in pro application. It will communicate with your Web app server about back end server. So the web-based ones are action.

Which we perform on the screen.

Our, uh, data will be loaded from the back end.

And this.

The data will be displayed on portal screen where we have integrated our web dynpro application.

So sappy let Weaver Portal is a place where we're going to configure our web improve application.

So when we whenever we create a web application.

Uh you uh URL is generated and that URL is uh.

Integrated with the portal, so when you log into the portal, you'll be able to access your web application.

156912122.

Uh, so the place where we integrate the weblink replication in the portal is called an eye view.

So basically we generate some eye view in the portal.

So this is the.

This is basically a job of a portal.

Consultant who will ask you what is your web link for application?

Name and under which node you have to create that application.

So the portal consultant will create a node in the portal where he's going to add your application.

Basically add your application name so once the configuration is maintained at the portal level, you will be able to access your web application directly from Portal.

Web in web dynpro development.

So development completely integrates into ABAP Workbench.

So on the right hand side the screen which you are able to see is your web application screen.

This is our sieti transaction from the drop down.

We have selected a web dynpro component.

Node and this is one of the web dynpro component which is opened here.

So you can see from the left hand side you will see all the elements which are created in the web dynpro when we create a new application so.

In this web dynpro controller web interface views window.

So these are all.

All elements which are created by the system itself and you design when you create a new web in for application.

On click of these elements, you'll find those elements have been opened in the work area on the right hand side of this navigation area.

If I click on the view, the view layout will be visible in this work area.

If I click on the window, I can see the window configuration in this work area.

So this is how we interact with the web link through screen.

Can everybody please go on mute?

So so.

So there are there.

There are various components inside.

There are various elements inside the web link road komponent.

So when we create a new web dynpro application component, it creates a.

View and window and view window component controller component interface.

So there will be always one window created and one blank view generated by an application.

So in a web token it will be incomplete without a window and a view.

So there's no such application without a window and our view, so.

Now what is window and waterviews?

Window is nothing but a entire screen is.

If you see, is nothing but a window and the different elements inside the screen are views.

So window is nothing but a collection of different views.

If I say this toolbar up, there is one view and the navigation area at the left hand side is other view and the work area which I can see at the middle is third view and my window is a collection of all such views.

If I if I click on some button and if I went into another view.

So that's mean I have more than one views in my window and I'm navigating from one view to another view.

So that's why we have a window node and view view node.

So once the view was created we have to integrate it with the window.

And window will decide how we will be navigating from one view to another view.

So when we will move towards creating our basic application and we will see how to design A view and window and how we will be navigating from one view to other.

So it's a declarative viewer development.

As I said, the first thing we have to do is declaration of elements.

If you see in this picture, there's a no.

There's a column called context.

So when we click on the context, we will see where there's a place where we do all our declaration.

In this screen you can see there's our table is some fields, so all the fields which are associated to our table if you design any input field so whatever variables required for those fields have been declared in the context area.

Ah, but deterred with forward navigation so.

If if you want to perform any action on the button click, so whatever logic we gonna write is all handled by the methods which are in the method section and there's uh above uh, we focus on the above uh.

Programming is also using a programming technique.

We can write a logic to control the view UI elements.

We will see how to do debugging in the future sessions.

Move ahead.

So now as we start designing an application a few things few nodes which are created as I said earlier.

So we'll see in detail what are the purpose of each and every node.

So let's say if you when you when you will start designing our application.

We will design as application.

You'll find a component controller is generated.

There's one node for component interface, there's one custom controller, view windows and component usage.

So these are a few four to five nodes which are generated as soon as you create your custom application.

Now what is a component controller?

No. Uh.

If we if we understand the concept of oops uh, we have a place called attributes.

Attributes is the location where we declare most of our global internal table structures and variables.

Why we declare them in in attributes?

Because these are those variables and.

Structures which we're gonna reuse in various methods, so that's why they are declared in the attributes section so.

So which means that attribute has a global visibility across the across different method, same as the functionality of component controller, component controller has a global visibility across the application and the instance of a component controller is active.

Throughout the application until it is the application is finished.

So whichever.

Methods which uh, we have to access in different views can be defined in the component controller all.

All the variables or declarations which are necessary for multiple views that can be defined in a component controller.

So so basically this is a place where we do all our global declaration and define all the methods which we need to reuse in different views so that it should get a global picture or it should be active throughout the web dynpro.

Next is a component interface, now component.

What is component interface?

So as I said we can reuse our most of the components.

Uh, the we have given one one of the example like there's a view which we needs to.

Similar view we need to design in three separate components and why we should go and keep on designing the same kind of view again and again in different components.

We can reuse that that view from a component 1 into component 323 and four so here.

Now comes the component interface.

Uh, here it comes the component interface into picture where we can reuse the components.

Uh, different.

Uh, from of other components.

A custom controller.

Custom controller is similar to a component controller.

Uh, it's.

It has same properties as that of a component controller, just it can say one more.

Uh, node available so if you don't want to.

To load your component controller, you can go for a custom designing a custom controller.

Then views.

So views is a place where we will be designing all our UI screen using various UI elements.

And then window is where we will be embedding all our views together.

So in this particular screenshot you can see there are two views, so a window.

Uh. When? Uh.

We could be navigating from the input view to result based on some button click.

So the navigation from input 2 result is defined at the window level.

So window will we need to define in the window that yeah there is going to be 2 views.

And you have to navigate from input view to result view so.

A window will be a collection of your views and it will set the properties saying from navigation property defining from where we have to navigate to from which we do what other view.

And there's a component usage, so.