What is fiori?

SAP Fiori is the new client experience (user experience) for SAP programming. It is a set of cross device applications. Fiori provides a set of applications that are used in regular business functions like work approvals, financial apps, calculation apps and various self-service apps. Fiori is basically a design principle. SAP is developing its Fiori apps on the latest interface framework, SAPUI5.

What SAPUI5?

Sapui5 is development toolkit for HTML5.SAPUI5 is SAPs own user interface to develop web applications. sapui5 is JavaScript framework which is made up of html5, JavaScript, css and less.
.it is responsive for all devices and browsers.
Sapui5 has become a huge library with more than 150 controls which results in a high efficiency when it comes to creating business applications.

Fiori design principles?

1) Role based
2) Adaptive
3) Coherent
4) Simple
5) Delightful

Sap fiori apps?

1) Transactional apps
2) Fact sheet apps
3) Analytical apps

Advantages of fiori apps

- 1)Responsive Design
- 2)Stateless Fiori Apps
- 3)Fast performance compare to ABAP Apps
- 4)Apps runs on all types browsers
- 5)Contains rich UI elements

about Launchpad, tile, group, CatLog etc.

The SAP Fiori Launchpad is the access point to apps on mobile or desktop devices. The apps are displayed via tiles. These tiles provide access to the single apps. Also provides a feature like theming, search integrity, you can add your company's logo etc.

The apps are organized by the following entities:

- **CatLog**
  A catalog is a set of apps you want to make available for one role. Depending on the role and the catalogs assigned to the role, users can browse through the catalogs and choose the apps that they want to display on the entry page of the SAP Fiori Launchpad.

  Group
  A group is a subset of apps from one or more catalogs. Which tiles are displayed on a user's entry page depends on the groups assigned to the user's role. In addition, the user can personalize the entry page by adding or removing apps to pre-delivered groups or self-defined groups.
  Roles (PFCG): Provides access to the assigned groups and catalogs.

1) **Different ways of binding in sapui5**
   Property binding, aggr binding, element binding

2) **Different types of models**
   Json, OData, resource model (i18n), xml
3) **Different types of views**
   Xml, html, js, json

Lifecycle of UI5 Controls

6) **Onint**
7) **onExit**
8) **onAfterRendering**
9) **onBeforeRendering**

4) **Different types of tiles**

1) **Static tile**
   -A standard tile that displays static content. You basically configure a title, subtitle, and an icon.
2) **Dynamic tile -A tile that displays data that is updated at regular intervals. The data is retrieved from a backend system using OData services.**

3) **Custom tiles-If the standard app launchers are not what you need, try configuring a tile with a different look and feel by selecting the** Custom App Launcher **tile type. In this case you can customize your tile by selecting different background colors, or choosing from a variety of icons, placing them in different areas of the tile, and so on**

4) **News-**
   Displays a news feed alternating between messages according to how you configure the tile. Only used for the News **app type.**
5) **Kpi -** A **KPI tile** is the graphical representation of the evaluation of the KPI, which is vis- ible to the user at runtime.
6) **No tile**

What are the different types of deployment options?

1) **Hub deployment Embedded deployment**
2) **Embedded Deployment Option**

Difference between Embedded and Central hub deployment

Central Hub Deployment Option

With this kind of deployment, the SAP Fiori UI add-ons (both central and product specific) and SAP Gateway components are deployed in the ABAP front-end server. The back-end data and the business logic are deployed on the ABAP back-end server

Embedded Hub Deployment Option

Embedded deployment means that SAP Fiori UI add-ons (both central and product specific) and SAP Gateway are deployed along with the back-end SAP Business Suite

Use of component.js file

-Includes libraries and set global model which will be applicable for all views.

-The component file is the first to be executed for an UI5 application.

-This file is called from index.html.

-Initially the routes and targets were mentioned here for navigation which is later done in manifest.Json now.

-It also defines the root view, models to be instantiated for the application.

-Component.js is a file where you can define a component, which acts as global controller to the application. From which you can control views and view controllers, models of the application.

Use of Manifest.json file
Also called descriptor file. it provides the space for storing the metadata associated with the application. All data is stored in json format in the manifest. json file.it contains app id, app version, data sources, along with required component and libraries. Also we write routes and targets in manifest.json file.

Use of neo-app.json file
The neo-app.json file contains all project settings for SAP Web IDE and is created in the root folder of your project. It is a JSON format file consisting of multiple configuration keys. The most important setting for you to configure is the path where the SAPUI5 runtime is located when starting the app.

Use of component preload.js

SAPUI5 applications are build with the MVC pattern and therefore consist of multiple files, like views or controllers, With fragments, it's possible to split up parts of views even further into smaller logical units.

Although having a lot of difference files based on their functionality is good during the development, it's bad when running on a production server. Every necessary file fires a separate network request. You have to wait until all network requests are completed before the app is usable. you may not notice this delay in your corporate network , but you'd notice in on a mobile phone with a moderate connection speed.

To solve this issue, you can package different files into a single file, a process called ***bundling.*** through bundling, your app only has to load a single file instead of all the separate files, which speeds up the loading time of your application because your app has to wait only for a single file to load.
In SAPUI5, this single file is called ***Component-preload.js*** . this file contains all the code of your application in a minified version. **Minification** means removing all unnecessary white-space and comments, Which also decreases the file size. In addition, the javascript source code is also uglified. ***Ugification*** means that your source code will not be as readable as before.

Difference between sap web ide and eclipse

Web IDE	**Eclipse**
1) **Application structure is sutaible for sapui5 application which contains different folders which represents mvc (model , controller ,view , i18n different-deffrent folders created)** **Also application structure contains manifest.json file, component.js file, neo-app.json file**	Eclipse folders structure is not sutaible for sapui5 app and does not contain manifest.json file, component.js file, neo-app.json

2) **No need to do any extra installation to start development**	You need to install sapui5 plugins to start sapui5 development in eclipse.
3) **Extension of standard fiori app is easy with sap web ide (bcouse of extensibility pane)**	Difficult with eclipse ide (no extensibility pane) And need to install extra fiori toolkit.
4) **When you deploye application from web ide component-preload file gets automatically created**	When you deploy app from eclipse there is no any component preload file
5) **Web ide provides different templates for application development to reduce time**	In eclipse there is no any template gets provided.
6) **Offline development not possible**	Offline development possible

Bootstraping in SAPUI5
Bootstrapping**: Loading and Initializing. To use** SAPUI5 **features in your HTML page, you have to load and initialize the** SAPUI5 **library. You can use the** SAPUI5 bootstrap **script in your page to initialize** SAPUI5 **runtime automatically as soon as the script is loaded and executed by the browser.**
Different types of binding modes in sapui5

1) One way
2) Two way
3) One time

Difference between one time binding and one way binding
One way binding – unidirectional data flow. model to view binding- text , lable which should not accept user inputs and update info.

One time binding- one time data flow. When the binding object is evaluated,its corresponding model data are read and written to the element properties once and never again.
It all comes down to the number of change listeners. Fewer listeners means less memory allocation and less stuff to update when something changes during the runtime. The binding mode OneTime has thus the potential, compared to other binding modes, to optimize performance and memory consumption if applied appropriately.
When to use: For static, non-mutating data
Example – resource model binding

what is package?

package is simply a container for the development objects.

what is roles?

roles provide authorizations for the users

what is semantic object?

semantic object is a unique name provided for that application to run on a launchpad.

intent- Semantic object + action

BSP- Business Server Pages.

What is MVC?
MVC stands for model view controller; it's an architecture to design UI based applications to achieve maximum **reusability and extensibility** of the application for future changes. Model is the representation of data in form of object, View defines the UI and Controller contains all the logic to drive as well connects view with model.

The Model View Controller (MVC) concept is used in SAPUI5 to separate the representation of information from the user interaction. This separation facilitates development and the changing of parts independently.
The **view** is responsible for defining and rendering the UI.
The **model** manages the application data.
The **controller** reacts to view events and user interaction by modifying the view and model.

Use of aliases in sap?
Aliases are names that system administrators define for each system they create so that portal components can reference these systems. They are a means of retrieving the information stored on database servers without having to know the name of the server. System alias contains a combination of a lot more information/config like Local GW, Local App, RFC destination and Software Version. Based on the details provided in the system alias, the gateway decides where to search for the DPC and MPC classes. This also signifies the Box which we are referring to like is it CRM or ECC or something else.

If we have to use RFC destination alone then how we will be able to add those extra parameters with RFC. Thus we have a provision of System alias.

What is a client?

A client is defined as a self-contained technical unit within an SAP System. This means that all business data within a client is protected from other clients. Each client has its own customer data

Sandbox Client - It has all the settings as it is production, all the development activities are first done here for practice.

Development Client - All the development related activities related to customization or new programs are done here.

Quality Client - This is often called as Testing Client, mostly used for testing purposes of all the developments/ enhancements done before transporting to production.

Production Client - All the real time entries are done in this Client, this is user interface client, where online data is posted.

Map method in javascript

Factory function in sapui5 ?

Callback function
A callback is a function that is to be executed after another function has finished executing—hence the name 'call back'.
Any function that is passed as an argument is called a callback function.
Example

```
function doHomework(subject, callback) {
  alert(`Starting my ${subject} homework.`);
  callback();
}

doHomework('math', function() {
  alert('Finished my homework');
});
```

Json Parse-
The JSON.parse() method parses a JSON string,

/UI2/FLP for directly navigate to *SAP* ***Fiori Launchpad***
/UI2/FLPD_CUST for directly navigate to ***Fiori Launchpad Designer***

Difference between for and foreach?

```
foodArray.forEach((food, index) => {
  console.log(`i value: ${index} | Food Name:`, food);
});
```

1. forEach keeps the variable’s scope to the block

```
const num = 4;
const arr = [0, 1, 2];

arr.forEach(num => {
  console.log(num);
});

// Result:
// 0
// 1
// 2
console.log(num);

// Result:
// 4
```

2. Lower chance of accidental errors with forEach

```
for (let i = 0; i <= foodArray.length; i++) {
  console.log(`i value: ${i} | Food Name:`, foodArray[i]);
}

// Result:
// i value: 0 | Food Name: { name: 'Burrito' }
// i value: 1 | Food Name: { name: 'Pizza' }
// i value: 2 | Food Name: { name: 'Burger' }
// i value: 3 | Food Name: { name: 'Pasta' }
// i value: 4 | Food Name: undefined
```

4. You can break out of a for loop earlier

```
for (let i = 0; i < foodArray.length; i++) {
  if (foodArray[i].name === 'Pizza') {
```

```
    console.log('I LOVE PIZZA');
    break;
  }
}
```

What is closure in js?

A closure is a function defined inside another function (called the parent function), and has access to variables that are declared and defined in the parent function scope.

The closure has access to variables in three scopes:

- Variables declared in their own scope
- Variables declared in a parent function scope
- Variables declared in the global namespace

What is callback function?
A callback is a function that is to be executed** after **another function has finished executing — hence the name 'call back'.

```
function first(){
  console.log(1);
}function second(){
  console.log(2);
}
first();
second();

// 1
// 2
```

Case 2

```
function first(){
  // Simulate a code delay
  setTimeout( function(){
    console.log(1);
  }, 500 );
}function second(){
  console.log(2);
}
first();
second()
// 2
// 1
```

Now callback

```
function doHomework(subject, callback) {
alert(`Starting my ${subject} homework.`);
callback();
}

doHomework('math', function() {
alert('Finished my homework');
});
```

```
function doHomework(subject, callback) {
  alert(`Starting my ${subject} homework.`);
  callback();
}

function alertFinished(){
  alert('Finished my homework');
}
```

doHomework('math', alertFinished);

Path to activate front end bsp app?
Go to SICF
***Path** default host -> sap/bc/ui5_ui5/sap/appln name activate it*
***Path** default host -> sap/bc/bsp/appln name activate it*

Path to activate backend services?

***t-code** ->Iwfnd/maint_service*
select your odata service
click on icf node button
and activate it.

SAP Netweaver Gateway-
SAP Gateway is a technology that provides a simple way to connect devices, environments, and platforms to SAP applications, using any programming language or model without the need for SAP knowledge, by using REST services and OData protocols.

SAP Gatway offers:-

1) Supports any device and any platform

2) CRUD operations
3) Doesn't require coding; well-suited for non-ABAP developers
4) No need for SAP knowledge

5) **Android debugging**

Click Settings icon in main screen
Developer options/ About phone
Scroll down to " Build number " menu item in " About phone " menu list.
You need totally tap seven times on " Build number " menu item,

Then go back to Settings menu list, you can see " Developer options " menu item in it.

Step 2: Enable USB Debugging Mode

Type on" Developer options" menu item in Settings menu list.
Check" USB debugging" checkbox to enable it.
It will prompt you to ask whether" Allow USB debugging" or not, click OK button.
Now the enable USB debugging mode operation is completed successfully.

Step 3: Check Whether Device Is Connected with PC Properly

2. Open dos window.
6) **Execute command" adb devices ". Then you can see connected android devices list as below**

7) **Launchpad configuration**

For pckge creation- se80
Step 1**: Register your application**
use T Code "LPD_CUST"
Click on "New Launchpad"/create role
Write details:
Role ,instance, desc,namespace (you can write anything)
Press enter and save it.
Now Launchpad has been created.

Add target appln-
Click on your created Launchpad
Click on "New Application"
Link Test=write free text related to your application.
Select URL option from Application Type.

Click on edit button icon in the Application Parameter section.
Write full path to access the application in the text area.
Ex-/sap/bc/ui5_ui5/sap/" your upload name of BSP application".
Write component name in Additional Information.
SAPUI5.Component=component name of your application
Save it.
you will get the list in left hand side
Now Application has been added to launchpad.

Step 2:-
Create customizing TR
Step 3:
Login to Launchpad designer url
http://ehp7prd.sap.in:8049/sap/bc/ui5_ui5/sap/arsrvc_upb_admn/main.html
host name will be varied and path will be same
enter credentials and click on setting icons
select your created Customizing TR and click on OK.
Step 4:
Create Catelog
Click on + Icon
Provide title and ID (you can write anything)
Then save it.
Now catalog has been created
Step 5: tile creation
Select app launcher#static tile
Click on create tile and provide details- you can write anything
Title , keyword , and icon
Step 6
Create semantic obj
T Code (/n/ui2/semobj)
Click on edit→ click on new entries
Semantic Objcet:"ZtestMaster"
Semantic Object Name:"test Mock Master"
Semantic Object description:"test mock semantic object"

Save it , provide TR and enter..your semantic object is created.

Go to your browser where you opened Launchpad designer and refresh it.
Select Semantic Object from F4 that has been already created.

Action:"testMaster" (Write action related to your application).
Click on save button.You will get below screen.
Click on Ok button.You will get below screen.
Now Tile has been added to catalog successfully.

Step7

Target mapping
Select Target Mapping tab.You will get below screen
Click on Create Target Mapping button
Select your created semantic object.
Provide same Action as previous.

For Launchpad Role, Launchpad Instance , Application Alias
T code(LPD_CUST). Press enter.
Select your created Launchpad
Launchpad Role:"ZVIK_TEST"
Launchpad Instance:"TRANSGEN"
Application alias:"MasterApp"
Click on Save button.and ok
Now Target Mapping done successfully

Step 8: Create Group
Select groups tab then click on plus icon.
Provide details.

Title:"TestMas-Det"
Id: "ZTest_mastergrp

Click on Save button.
Now group has been created successfully.

Step 9: **Add tile to group**

Step 10 create Role
T-Code PFCG is use for creating Role
Write role Z:S_test_mas
Click on single role
Write description- Description: "single role for Master Application".
And save it
Role has been created
Step 11
Assigning catalog to created role
Select Menu Tab
Click on circle area of transaction
Select sap fiori catalog from options
Provide Catalog ID that already has been created.
Catalog ID:" ZTest_master"
And press enter
Now Catalog has been assigned successfully to role.
Step 12
Assign Group to created role.

Select "SAP Fiori Tile group" from options.
Provide Group ID that already has been created.
Group ID:" ZTest_mastergrp"
Press enter. Enter free text.
Now Group has been assigned successfully to role.

Step 13
Assign user to role.
Select User tab
Enter user name to whom you want to display catalog and group on Launchpad.
Press enter and save it
Now user has been assigned successfully to role.
Now this user can see assigned catalog and group on Launchpad.

Career objective

I want to contribute to the growth and the success of the organization by undertaking challenging assignments and delivering timely results using my professional knowledge, skills and expertise. I look forward to work in a professional environment to sharpen my skills.

Project information

Project Name: Tata Motors Finance Application
Brief Description: An application developed for complete finance process for all types of Tata Vehicles.
Our team is handling sourcing part of this application. It is a bunch of 18-19 applications which contains different functionalities from customer come to TATA motor for inquiry to customer is ready or eligible to buy a TATA Vehicle by completing all finance process.
Application start from my leads in which we add a new customer in our database with all required data of customer.
Then we move to need analysis in which we understand all needs of customers i.e. how many vehicles he want, is customer type is individual, group or organization. Then what is payment mode then which model of which vehicle in which quantity he wants etc. etc. Also we add co-applicants and guarantors for that customer.
Then we go for quotation of this app in which we find a quotation for customer's vehicles.
And there are so many stages. After particular bunch of processes, we move to next stage for ex. When you create a lead means u have added a new customer then application comes to first stage lead created. Then when you create a quotation then it moves to pre quote generated and after it moves to quotation generated.
Last stage release to ctf…in which we release all documents.
there are 4 logins for complete process of this application n u have to go through all these 4 logins for complete process.

Roles and responsibilities in this application: - everything about UI.
Handling development related issues and added a new functionality in application. And worked on many existing functionalities and changes the logic of existing UI.

New functionalities and complex logics
Recent functionalities

1) **Co-Applicant and guarantor screen changes**
2) **Co-Applicant and Guarantor Dropdown**
3) **ID Type of Co-applicant and Guarantors**
4) **App navigation using session variables**

Naroda Outbound Process
An application used for outbound process which include all functionality about logistics outbound. i.e. picking, shipping, create invoice etc.

Roles and Responsibilities: -
Getting requirements from client and creating application mockup from the requirements.
Complete Application development from scratch and handled development related issues.
Remove redundant code from application and provide quality code in application.
Giving support to application till it works fine on production server.

New functionality and complex logic: -
Barcode Scanning functionality
Showing count and quantity of selected line items in which items get selected by scanning, manual selection, all selection and entering number in input box for selection.
Use of Amcharts.

Weight Confirmation Application
A retail application is developed for product weight confirmation.

New functionality and complex logic: -
Make application compatible on all devices and browsers.

Work Order Management Notification: -

Purchase Order Creation App

Terminal Blending Application
This application is developed for blending different oil materials.
There are different types of oils and this application is used to blending means mixing different types of oils in particular oil in proper quantity so that blending will be perfect.

Material Downgrade App:-
This application is developed for material transport Source and destination.