

demo
username-kb@farmx.com
password-123456789
grant_type-password
client_id-5x7EuN09HAeBn2pYJnvq7szgJaULh14

live admin
tech@efarmexchange.com / Tech25@EFAD

```
{"username": "tech@efarmexchange.com", "password": "Tech25@EFAD", "grant_type": "password",  
"client_id": "5x7EuN09HAeBn2pYJnvq7szgJaULh14"}
```

ssh -i "farmx.pem" ubuntu@ec2-13-127-234-85.ap-south-1.compute.amazonaws.com

B-Dv3'&;s26HxPq[

AIzaSyAHbKWGMwqvOGJhaVj1XKMXFZeLdzzXK

Holding_2324
C6sJWyA}#kC[

1. What is the difference between ViewState and SessionState?

'ViewState' is specific to a page in a session.

'SessionState' is specific to user specific data that can be accessed across all pages in the web application.

Explain how can you submit a form using JavaScript?

To submit a form using JavaScript use document.form[0].submit();
document.form[0].submit();

What is the use of Void(0)?

Void(0) is used to prevent the page from refreshing and parameter "zero" is passed while calling.

Void(0) is used to call another method without refreshing the page.

7. What is difference between Array.splice() and Array.slice() method in JavaScript?

12. Explain what is Javascript? List some data types supported by Javascript?

3. How to encode and decode a URL in JavaScript?

How to find if two arrays contain any common item in Javascript?

Input: array1 = ['a', 'b', 'c', 'd', 'e']

array2 = ['f', 'g', 'c']

Output: true

var rowSize = 100; // => container height / number of items

```
var container = document.querySelector(".container");
var listItems = Array.from(document.querySelectorAll(".list-item")); // Array of elements
var sortables = listItems.map(Sortable); // Array of sortables
var total    = sortables.length;
```

```
TweenLite.to(container, 0.5, { autoAlpha: 1 });
```

```
function changeIndex(item, to) {
```

```
    // Change position in array
    arrayMove(sortables, item.index, to);
```

```
    // Change element's position in DOM. Not always necessary. Just showing how.
```

```
    if (to === total - 1) {
        container.appendChild(item.element);
    } else {
        var i = item.index > to ? to : to + 1;
        container.insertBefore(item.element, container.children[i]);
    }
```

```
    // Set index for each sortable
```

```
    sortables.forEach((sortable, index) => sortable.setIndex(index));
}
```

```
function Sortable(element, index) {
```

```
    var content = element.querySelector(".item-content");
    var order   = element.querySelector(".order");
```

```
    var animation = TweenLite.to(content, 0.3, {
        boxShadow: "rgba(0,0,0,0.2) 0px 16px 32px 0px",
        force3D: true,
        scale: 1.1,
        paused: true
    });
```

```
    var dragger = new Draggable(element, {
        onDragStart: downAction,
        onRelease: upAction,
        onDrag: dragAction,
        cursor: "inherit",
        type: "y"
    });
```

```
    // Public properties and methods
```

```
    var sortable = {
        dragger: dragger,
        element: element,
```

```
    index: index,  
    setIndex: setIndex  
};
```

```
TweenLite.set(element, { y: index * rowSize });
```

```
function setIndex(index) {  
  
    sortable.index = index;  
    order.textContent = index + 1;  
  
    // Don't layout if you're dragging  
    if (!dragger.isDragging) layout();  
}
```

```
function downAction() {  
    animation.play();  
    this.update();  
}
```

```
function dragAction() {  
  
    // Calculate the current index based on element's position  
    var index = clamp(Math.round(this.y / rowSize), 0, total - 1);  
  
    if (index !== sortable.index) {  
        changeIndex(sortable, index);  
    }  
}
```

```
function upAction() {  
    animation.reverse();  
    layout();  
}
```

```
function layout() {  
    TweenLite.to(element, 0.3, { y: sortable.index * rowSize });  
}
```

```
return sortable;  
}
```

```
// Changes an elements's position in array  
function arrayMove(array, from, to) {  
    array.splice(to, 0, array.splice(from, 1)[0]);  
}
```

```
// Clamps a value to a min/max
```

```
function clamp(value, a, b) {  
  return value < a ? a : (value > b ? b : value);  
}
```

R@#ul@123

\$P\$B3Y5AyEz11iIm1kQs3dy/vuxu2apOJ1

admin@123

Lots of things improved and so many to improve like timely completion of tasks and synchronize with team. Understanding the concept is one of the major thing to improve.

Username: Y11cPMPTsq
Password: ED:K|&=GX}RB

residentmanager@auramahvalley.com
admin@123

Password	email	Username
cgV4E#>CnJ	- aman@digisails.com	- aman
Qp7c\$F=eX#	- manoj@imperialholding.in	- imperial
9Rv/xT&b=]	- valorie5761266795422@mailinator.com	- valorie57
y_5R!k}WUm	- samayra@uniqueretreats.com	- marc
%R2E=bs>'J	- jeresmash@gmail.com	- jere
WL^ka(-9qc	- admin@villasamayra.com	- administrator
87]R^+QVp{	- gaynell1069994880621@mailinator.com	- gaynell10

sakshi - ^vvZ#3F_P9v,QKsD

aman - 7&/TUGv:}N&3Cy9h

shyam - BjsKG6Y!\$!]ZZdSN

Company - Dell
Model - Inspiron 13 5000
Screen size - 13.3' Full HD display
Processor Brand - Intel
Processor Name - Core i5

Processor Generation - 8th Gen
SSD Capacity - 256 GB
RAM - 8 GB (DDR4) 2400 MHz
Processor Variant - 8250U Clock Speed 1.6 GHz with Turbo Boost Upto 3.4 GHz
Cache - 6 MB
Graphic Processor - Intel Integrated UHD 620
Operating System - Windows 10 Home

<https://morioh.com/topic/javascript>
<https://morioh.com/p/0d01f0dd56d0>
https://morioh.com/p/b2d2c75c24b5?fbclid=IwAR1fDMvumAWq7xjOQJgdVWb05XP2JAYx0clEYQ8t_2lE5ZXan-vMarKA5WY
<https://morioh.com/p/b2d2c75c24b5-video>

<https://www.lambdatest.com/blog/best-javascript-framework-2020/>
<https://medium.com/analytics-vidhya/frontend-developer-interview-questions-reallife-dont-miss-it-92a9dc081a8f>
<https://www.freecodecamp.org/news/javascript-closures/>
<https://www.freecodecamp.org/news/the-react-cheatsheet-for-2020/>
<https://blog.greenroots.info/understanding-javascript-execution-context-like-never-before-ckb8x246k00f56hs1nefzpysq>
<https://blog.greenroots.info/javascript-interpreted-or-compiled-the-debate-is-over-ckb092cv302mtl6s17t14hq1j>
<https://raygun.com/blog/improve-javascript-performance/>
<https://morioh.com/p/57e32b48e935>
JavaScript localStorage Crash Course
📺 <https://morioh.com/p/57e32b48e935>
https://www.youtube.com/watch?v=KxtPHZcjlBI&feature=emb_rel_pause
<https://morioh.com/p/7e3f247c5b91>
<https://morioh.com/p/dd13d2fe8870---deployment-nginx-pm2-server>

<https://levelup.gitconnected.com/extremely-useful-lodash-methods-b38f121fea7e>
<https://www.youtube.com/watch?v=1f4J-XOkXf4&list=PLURGhPJ9RxpOKGoObOCWKApIU6T0GASlm&index=39>

<https://www.monkwhocode.com/2020/05/javascript-throttling.html> -----imp
<https://morioh.com/p/ec7cb2b7aed0> -----var const let

<https://morioh.com/p/94ba191d6a96> -----send email from js

<https://morioh.com/p/4a700cfa4ad4> -----array all

<https://morioh.com/p/015df2c6da33> -----all array method

https://morioh.com/p/b2d2c75c24b5?fbclid=IwAR1fDMvumAWq7xjOQJgdVWb05XP2JAYx0clEYQ8t_2lE5ZXan-vMarKA5WY ====all cource javascript
JavaScript Programming Tutorial - Full JavaScript Course for Beginners

🔗 <https://morioh.com/p/b2d2c75c24b5>

<https://raygun.com/blog/debug-javascript-google-chrome/>

<https://jscomplete.com/learn/javascript-for-react>

<https://blog.logrocket.com/7-ways-to-improve-node-js-performance-at-scale/>

[https://mariusschulz.com/blog/removing-elements-from-javascript-arrays#:~:text=Approach%20%231%3A%20indexOf\(\)%20and%20splice\(\)&text=The%20remove\(\)%20function%20removes,it%20sees%20a%20negative%20index.](https://mariusschulz.com/blog/removing-elements-from-javascript-arrays#:~:text=Approach%20%231%3A%20indexOf()%20and%20splice()&text=The%20remove()%20function%20removes,it%20sees%20a%20negative%20index.)

<mongodb://landx:PRu2VI1eO7eM0#wg@staging.landx.co.in:27017/?authSource=landx-staging&readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false>

[mongodb://farmx:Dzv#f3o&Rm\(x4Mn@farmx.co.in:27017/?authSource=farmx-dev&readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false](mongodb://farmx:Dzv#f3o&Rm(x4Mn@farmx.co.in:27017/?authSource=farmx-dev&readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false)

mongodb://farmx-new:kfAA3GW4Lc2_XF1@dev.farmx.co.in:27017/?authSource=farmx-dev&readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false

db name : farmx-dev

username: farmx-new

password: kfAA3GW4Lc2_XF1

What is Dependency Injection?

What are web packs?

What is Prototype Object?

What is the difference between onload and onloadstart?

How do you optimize a website's assets?

What are three ways to reduce page load time?

What are the limitations when serving XHTML pages?

What is the difference between linking to an image, a website, and an email address?

What are web packs?

What's the difference between standards mode and quirks mode?

What kind of things must you be wary of when the design or developing for multilingual sites

reactive from ,services, view ,child, component and coding

<https://www.indeed.com/career-advice/interviewing/angular-interview-questions>

What is Angular?

How familiar are you with Angular?

What is Angular used for?

What's the difference between Angular and AngularJS?

Can you describe what scope is in Angular?

What does data binding refer to?

Can you tell me what directives are in Angular?

What is NgModule?

Can you tell me what filters are used for in Angular?

How would you optimize an Angular application?

What are decorators in Angular?

What types of compilations are there in Angular?

Describe the types of directives in Angular.

Can you describe some of the advantages of using Angular?

What are expressions?

What does DOM stand for?

What does BOM stand for?

How would you create a service within Angular?

What's the difference between automatic and manual bootstrapping?

What's the difference between one-way and two-way data binding in Angular?

liuv in nodejs

nodejs

which language are liuv =c++

help execute synchronus task

streams = read and write

buffer

thread pull

role of thread pull and barger pull

Program Statement

Enter two strings. The task is to find the length of the longest common substring.

Rajanish Mishra4:18 PM

Let S1 = "ABCDGH", S2 = "ACDGH"

output :4

The diamond star pattern is a combination of the pyramid

Pyramid will be generate as per the input number.

```
*  
***  
*****
```

Question no: 1 :- Whats the use of Angular ?

Question no: 2 :- What are directives in Angular ?

Question no: 3 :- Explain the different types of Angular directives ?

Question no: 4 :- Explain the importance of NPM and Node_Modules folder ?

Question no: 5 :- Explain the importance of Package.json file in Angular ?

Question no: 6 :- What is typescript and why do we need it ?

Question no: 7 :- Explain importance of Angular CLI ?

Question no: 8 :- Explain the importance of Component and Modules ?

Question no: 9 :- What is a decorator in Angular ?

Question no: 10 :- What are Annotation or MetaData ?

Question no: 11 :- What is a template ?

Question no: 12 :- Explain the four types of Data bindings in Angular ?

Question no: 13 :- Explain architecture of Angular ?

Question no: 14 :- What is SPA in Angular ?

Question no: 15 :- How to implement SPA in Angular ?

Question no: 16 :- How to implement routing in Angular ?

Question no: 17 :- Explain Lazy Loading ?

Question no: 18 :- How to implement Lazy Loading in Angular ?

Question no: 19 :- Define Services ?

Question no: 20 :- What is Dependency Injection ?

Question no: 21 :- How to implement Dependency Injection ?

Question no: 23 :- Whats the benefit of Dependency Injection ?

Question no: 24 :- Differentiate between ng serve and ng build ?

Question no: 25 :- Explain the --prod parameter in ng build ?

Question no: 26 :- Explain ViewChild and ViewChildren ?

Question no: 27 :- Why do we need Template reference variables ?

Question no: 28 :- What is ContentProjection ?

Question no: 29 :- Explain Content projection Slot ?

Question no: 30 :- What is ContentChild and ContentChildren?

Question no: 31 :- ViewChild vs ViewChildren vs ContentChild vs ContentChildren ?

Question no: 32 :- Explain the importance of Component life cycle ?

Question no: 33 :- Explain events and sequence of component life cycle ?

Question no: 34 :- constructor vs ngOnInit() ?

Question no: 35 :- How to make HTTP calls using Angular ?

Question no: 36 :- What is the need of Subscribe function ?

Question no: 37 :- How to handle errors when HTTP fails ?

Question no: 38 :- How to pass data between components ?

Question no: 39 :- What are input , output and event emitters ?

Question no: 40 :- How to pass data during routing ?

Question no: 41 :- Is it a good practice to pass data using services ?

Question no: 42 :- What is the need of Angular Pipes ?

Question no: 43 :- Can you name some built-in Angular pipes ?

Question no: 44 :- How to create Custom pipes in Angular ?

Question no: 45 :- What is the fullform of RxJS ?

Question no: 46 :- What is the purpose of RxJS ?

Question no: 47 :- What are Observables and Observers ?

Question no: 48 :- Explain the use of Subscribe with sample code ?

Question no: 49 :- How to unsubscribe in RxJS ?

Question no: 50 :- Explain the concept of operators with sample code?

Question no: 51 :- How to install RxJS ?

Question no: 52 :- Differentiate between promise and RxJS ?

Question no: 53 :- In Angular where have you used RxJS?

Question no: 54 :- Which operators have used from RxJS ?

Question no: 55 :- What is the Push/Reactive vs Pull/Imperative?

Question no: 56 :- What are Interceptors in Angular?

Question no: 57 :- How to implement Interceptors?

Question no: 58 :- Give some use of Interceptors?

Question no: 59 :- Can we provide multi-Interceptors

<https://www.stackchief.com/tutorials/JavaScript%20ES6%20Loops>
<https://www.sololearn.com/Discuss/2542855/javascript-question>
<https://www.sitepoint.com/whats-the-difference-between-function-call-and-function-apply/>
<https://www.fullstack.cafe/blog/javascript-code-interview-questions>
<https://www.fullstack.cafe/interview-questions/angular>
<https://dev.to/maxpou/typical-javascript-interview-exercises-explained>
<https://karthiktechblog.com/angular/how-to-implement-time-based-caching-in-angular-using-http-interceptor>

<https://www.programiz.com/c-programming/examples/prime-number-intervals>
<https://www.w3resource.com/javascript-exercises/javascript-array-exercises.php>
<https://www.programiz.com/javascript/array>
<https://www.codingame.com/playgrounds/6181/javascript-arrays---tips-tricks-and-examples>
<https://github.com/jwill9999/JavaScript-Array-Interview-Practice>
<https://www.javatpoint.com/how-to-run-a-c-program-in-visual-studio-code>
<https://www.interviewbit.com/javascript-interview-questions/#recursion>
<https://www.javatpoint.com/javascript-interview-questions>
<https://www.javatpoint.com/javascript-data-types>
<https://www.guru99.com/javascript-interview-questions-answers.html>
<https://trio.dev/blog/javascript-developer-interview-questions>
<https://dev.to/maxpou/typical-javascript-interview-exercises-explained>
<https://www.freecodecamp.org/news/javascript-interview-prep-cheatsheet/>

Currently i am working as Sr. software Developer in efarm exchange pvt ltd this is the ecommerce agriculture product based company. total 5+ years of experience in angular technologies and specializing in front-end and basic exposure in back-end web technologies & aws ec2 hosting part. Interested in devising a better problem-solving method for challenging tasks, and learning new technologies and tools if the need arises.

Main role and responsibility.

i handling as web team lead and using technologies in angular 8, 10, 12 version running on 3 ecommerce based project.

tree shaking

create pure and impure pipe

view encapsulation

@output

what is host listener

event binding..

what is ngModel

what is ngModule

what are the object in ngModule

where we define service

form types

spa

angular life cycle

explain one by one

when destroy is calling.
when we use promise
and when we use observable
arrow function
what is subject behaviour
create custom directive
create service..
write post method. in service..
observable & promise difference
scope
many ways to declare variable.
event emitter
unit test case run jasmine karma..
ivy features
latest angular features and updates
jit & aot
cookies session localstorage
how to define cookies manually.
interceptor how to write it
what is eager loading.
lazy loading.
scope
var & const difference
var let const difference
how to inject any javascript library
how to control component using reactive form.
<https://swatijaiwal03.medium.com/reusable-custom-components-in-angular-7359e1894d25>
name age

agGrid implement..
route event.
wlidcard..
interface & model

=====acs=====

```
var a={},  
b={key:'b'},  
c={key:'c'};
```

```
a[b]=123;  
a[c]=456;
```

```
console.log(a[b]);  
=====
```

```
var x = 21;  
var test = function () {  
  console.log(x);  
  var x = 20;  
};  
test();
```

```
array.splice()  
array.slice();
```

how to unsubscribe method in ngOnDestroy()

hoisting..

closure

call and apply

router Event

wild card router

form types

pipes

routing guards

shr

cookies and storage difference

==<https://medium.datadriveninvestor.com/cookies-vs-local-storage-2f3732c7d977>

clitus=====

how does the browser actually render a website

When a web page is loaded, the browser first reads the HTML text and constructs DOM Tree from it. Then it processes the CSS whether that is inline, embedded, or external CSS and constructs the CSSOM Tree from it. After these trees are constructed, then it constructs the Render-Tree from it.

hoisting

closure

pure & impure pipe difference

routing guard

@ViewChild

@Component({}) create=====

viewbox in css

what is ajax

html5 storage

faujiti company..

what is momentjs and why we use it.

what are the es6 features.

what is destructuring.. with example.

what is components how to display value. in html.

hoisting.

what is spread oprator.

what is difference between promise and obserable

how to validate hh.mm.ss.SS

```
let flattened = [[0, 1], [2, 3], [4, 5]]
```

```
var officers = [
```

```
{ id: 20, name: 'Captain' },
```

```
{ id: 24, name: 'General Veers' },
```

```
{ id: 56, name: 'Admiral Ozzel' },
```

```
{ id: 88, name: 'Commander Jerjerrod' }
```

```
];
```

newbuy

```
company=====
=====
```

Create Component Dynamically using ComponentFactoryResolver and ViewContainerRef in Angular.

Create Component Dynamically.

ViewContainerRef in Angular

```
import {pipe, PipeTransform} from '@angular/core';
```

```
@pipe({
  name: 'reverse'
})
```

```
export class upperCasePipe implements PipeTransform{
  transform(value:unknown,...args:unknown[]):unknown{
    return null;
  }
}
```

<https://www.freecodecamp.org/news/when-and-why-you-should-use-es6-arrow-functions-and-when-you-shouldnt-3d851d7f0b26/>

<https://dev.to/mathlete/anonymous-functions-vs-named-functions-vs-arrow-functions-57pm#:~:text=Arrow%20functions%20are%20just%20shorter,human%2Dreadable%20than%20anonymous%20functions.>

<https://www.tektutorialshub.com/angular/angular-router-events/>

₹50
₹5,000
₹50,000
₹5,00,000
₹5,00,00,000

```
{{ 12345.23 | currency: 'INR': 'symbol': '1.0-0' }}
```

<https://www.bestinterviewquestion.com/es6-interview-questions>
<https://www.interviewbit.com/es6-interview-questions/#what-is-webpack>

<https://www.techiediaries.com/angular-local-json-files/>
<https://www.javatpoint.com/display-data-from-json-file-in-angular>
<https://www.tektutorialshub.com/angular/angular-httpclient-http-interceptor/>
https://www.simplilearn.com/tutorials/angular-tutorial/angular-interview-questions?source=frs_recommended_resource_clicked

https://www.simplilearn.com/coding-interview-questions-article?source=frs_category

persanal tab.....

<https://swatijaiwal03.medium.com/angular-cheat-sheet-8e81c17010cf>
<https://swatijaiwal03.medium.com/reusable-custom-components-in-angular-7359e1894d25>
<https://www.ag-grid.com/angular-data-grid/filter-text/>
<https://www.ag-grid.com/angular-data-grid/getting-started/>
<https://www.toptal.com/javascript/interview-questions>
<https://gist.github.com/andrewmkhoury/3fc03d6a71699467810f6722bfbabb96>
<https://flexiple.com/flatten-array-javascript/>
<https://www.boardinfinity.com/blog/top-10-features-of-es6/#6-destructuring-assignment>
https://www.tutorialspoint.com/momentjs/momentjs_quick_guide.htm
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map

hoisting allows only with var variable type..
its not working with let & const.

```
hcl=====
var arr=[{ name:'abhai', mark: 100, address:{phone:9377777777} },
{ name:'kumar', mark: 250, address:{phone:9377777777} },
```

```
{ name:'mishra', mark: 200, address:{phone:9377777777} },
{ name:'rahul', mark: 300, address:{phone:9377777777} }
]
```

```
var result = arr.filter((item, index, ar) => {
  if(item.mark == 200 || item.mark==300){
    arr[index].address.phone=1111111111;
    return item;
  }
})
```

console.log(result);

why angular is introduce why its need

how to it works process of angular application works

what is bootstrapModule in angular

what is platformBrowserDynamic what is the work of it

fmgg===

kamla pasand===

encapsulation: viewEncapsulation.Shadow or viewEncapsulation.Native === we don't inherit the global styles and have to define all the required styles on our component decorator.

ViewEncapsulation.Native: If we want Angular to use the shadow DOM we can set the encapsulation parameter to use ViewEncapsulation.Native

With ViewEncapsulation.Native styles we set on a component do not leak outside of the components scope.

ViewEncapsulation.None: If we don't want to have any encapsulation at all, we can use ViewEncapsulation.None.

some of any given no untill single digit response got...

```
var res = 0;
function addNum(number){
  let num= number.toString();
  var sum=0;
  for(let i=0; i<num.length;i++){
    sum= sum + parseInt(num[i]);
  }
}
```

```
if(sum.toString().length>1){
  return addNum(sum);
} else{
  return sum;
}
```

```

    }
  }
  console.log(addNum(4252524242));

```

```
[[3,1,4],[2,4,8],[7,6,9]]
```

If we don't encapsulate anything, the style we defined in the component will leak out and started affecting the other components.

ViewEncapsulation.Emulated: Angular changes our generic css class selector to one that target just a single component type by using automatically generated attributes.

Any styles we define on a component don't leak out to the rest of the application but with ViewEncapsulation.Emulated our component still inherits global styles.

1) Polymorphism in angular

2) pure and impure pipes

3) difference between filter and map etc.

lazy loading, view child , interceptor

prototyping, array new functions, lazy loading in angular some css basic questionns

Difference between service and factories?

What is closure, inheritance, prototype, Numeric array sort without inbuilt fns, Palindrome pgm like that

Angular.js Question Only

1) Why preferred Angular.js

2) What is signal page application

3)What is two way data binding.

Basics about javascript and few questions on Angular JS (advantages, services,providers,custom directives)

object oriented JavaScript

Javascript concepts

explain how event loop works

What is coding and software

How interceptors work in Angular

Basic OOPS concepts and interface

Javascript - closure, apply,call, bind

Rxjs oprators Pipe in rxjs

Change detection and how to controlleing

All in change detection

HMR = hot reloading madule.

```
let ab = [{name : 'rah'}];
```

```
let abc = [{age : 25}];
```

```
let employee = {
```

```
  ...ab[0],
```

```
  ...abc[0]
```

```
};
```

```
console.log(employee);
```

Explain angular architecture.

Execution process of angular application.

What is grunt and gulp?

Different ways to add third party API/cdn in angular app.

Version of typescript used in angular 9.

Interviewer said some properties of angular and asked in which version of angular were they first included

Routing process with example (with code)

Basics of angular architecture and javascript

The questions were basic, related to AngularJS version.

What is rootscope in AngularJS.

What are filters?

What are pipes and custom pipes?

What is the difference between AngularJS and Angular.

What are directives in AngularJS

What are arrow functions?

How to share data between two components in AngularJS

reactive from ,services, view ,child, component and coding

Observables, Dependency Injection, Decorators, Difference between Angular 11 and 13

1.data pass to different components

2.dependency injection

3.directives and it types

4.lifecycle hooks and difference between ngOnchanges and ngondestroy

5.retry Rxjs operator

6.subjects and behaviour subjects

7.create custom directives

8.css block vs block inline

9.new elements of html5

10.bootstrap and how to make page responsive ?

How do you do lazy-loading? When should you use lazy-loading?

What are the risks of lazy-loading and how can you mitigate them? Hint: Multiple instances of singleton services

Explain the core-shared-feature module pattern

Explain the use and purpose of the annotations @Input, @Output and @ViewChild annotations

How would you integrate Bootstrap with Angular?

How would you use Sass in Angular?

How would you run Angular application in production? What are the options?

What is AOT compiler? How do you enable it?

What is Angular Universal? Why should we use it? How do you set it up in a project? How can you run Angular with Angular Universal in production?

What are the differences between declarations and providers in NgModule?

Only Authenticated users can access certain pages. How do you implement that?

These are all I can think of at the moment. Besides these in an Angular interview you may also be asked questions about Javascript, CSS, HTML, Sass etc.

What are components and how to create them.

What are modules.

Services and providers

Dependency Injection

Built-in and custom pipes

Built-in and custom directives

Lifecycle hooks

Routing and what is router-outlet

Lazy loading

Difference between ngOnInit and constructor.

@input and @output for component interaction.

Angular cli (command line interface)

TCS is part of biggest and trustworthy Indian brand TATA.. The name is enough to work. I have my greatest respect for this brand. They took Indian Economy several places higher in global ranking.

Their contribution to social welfare of the country is enormous. I will feel honored if i get to work for this organisation.

CanActivate: Checks route navigation before the component is loaded.

CanActivateChild: Checks route children navigation before the component is loaded.

CanDeactivate: Checks navigation from the current route eg leaving the partially filled form.

Resolve: Resolve loads/ retrieves data before the route is activated.

CanLoad: Checks to see if a user can route to a module that lazy-loaded.

tab

dynamic component

smart & dumb component

how to manage 10000 data in angular project

<https://blog.bitsrc.io/3-ways-to-render-large-lists-in-angular-9f4dcb9b65>

<https://stackoverflow.com/questions/48386458/angular-4-how-to-handle-display-10000-records-in-grid>

And for your example, in order to be able to load 100000 rows, you can use what they call "infinite-scrolling"

```

<div class="panel-body" style="overflow-y:scroll;height:20em;width:100%;"
  infiniteScroll
  [infiniteScrollDistance]="5"
  [infiniteScrollUpDistance]="2"
  [infiniteScrollThrottle]="300"
  (scrolled)="onScroll()"
  [scrollWindow]="false">
  <table class="table table-hover table-striped table-sortable">
    <thead>
      <tr>
        <th *ngFor="let column of columns" >
          {{column.display}}
        </th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let row of reportResult">
        <td *ngFor="let column of columns">
          {{row[column.variable]}}
        </td>
      </tr>
    </tbody>
  </table>
</div>

ts=====
reportResult = [];
responseResult = [];
isFinished = false;

onScroll () {
  if(!this.isFinished) {
    const condition = this.dbQuery + '.skip(' + this.reportResult.length + ').limit(100)';
    this.execute(condition);
  }
}

execute(qry){
  const that = this;
  $('#loading').show();

  this.reportService.getReportData(qry).then((data: any) => {
    this.isPageValid = true;
    $('#loading').hide();
    try{
      if(data.operation == "SUCCESS"){
        if(data.data === "")

```

```

{
  that.toastr.warning("No results found, please refine your conditions!", "Report");
}else{
  that.responseResult = JSON.parse(that.stringifyJson(data));

  if(that.responseResult.length < 100) { that.isFinished = true; }
  for(let i = 0; i < that.responseResult.length; i++) {
    that.reportResult.push(that.responseResult[i]);
  }
}
} else {
  that.toastr.warning("Data size is larger, please refine your conditions!", "Report");
}
}
catch(err){
  $('#loading').hide();
  that.toastr.warning("Data size is larger, please refine your conditions!", "Report");
}
},function(err){
  $('#loading').hide();
  that.toastr.warning("Data size is larger, please refine your conditions!", "Report");
}); }

```

=====

```

<ng-select [items]="arrayWithStrings" [virtualScroll]="true"
[formControlName]="name"></ng-select>

```

<https://www.thecodecampus.de/blog/optimize-the-performances-of-large-tables-in-your-angular-application/>

In brief:

Pipe: Used to stitch together functional operators into a chain. Before we could just do `observable.filter().map().scan()`, but since every RxJS operator is a standalone function rather than an Observable's method, we need `pipe()` to make a chain of those operators (see example above).

Tap: Can perform side effects with observed data but does not modify the stream in any way. Formerly called `do()`. You can think of it as if observable was an array over time, then `tap()` would be an equivalent to `Array.forEach()`

Today work--- 20-06-2022

Upload code on dev server. admin

Bug resolve in frontend and upload file on live server

upload frontend build in dev server

upload admin build in live server

resolve bug on input finance management -admin
upload api on live server

Thank you for giving me an opportunity to work in your esteemed organisation. I hereby formally accept the position of IT Analyst with Tata Consultancy Services. I also accept the offered annual CTC of Rs.16,20,010

How to unsubscribe from all observables in one component at once in Angular

Ans=====

I tend to store all my subscriptions in an array.

```
private subs: Subscription[] = []
```

```
ngOnInit() {  
  this.subs.push(  
    this.obs1.subscribe(),  
    this.obs2.subscribe() // etc  
  )  
}
```

```
ngOnDestroy() {  
  this.subs.forEach(s => s.unsubscribe())  
}
```

<https://blog.bitsrc.io/6-ways-to-unsubscribe-from-observables-in-angular-ab912819a78f>
<https://www.mattspaulding.org/The-Best-Way-to-Unsubscribe-From-Angular-Observables/>
<https://coryrylan.com/blog/subscribing-to-multiple-observables-in-angular-components>

=====HMR=====

<https://www.freakyjolly.com/angular-hot-module-replacement-technique-tutorial/>

=====

Delete Element=====

shift =====

The shift() method removes the first item of an array.

The shift() method changes the original array.

The shift() method returns the shifted element.

Pop()=====

The pop() method removes (pops) the last element of an array.

The pop() method changes the original array.

The pop() method returns the removed element.

Add Element=====

unshift()=====

The unshift() method adds new elements to the beginning of an array.

The unshift() method overwrites the original array.

The unshift() method returns The new length of the array.

push() =====

The push() method adds new items to the end of an array.

The push() method changes the length of the array.

The push() method returns the new length of the array.

slice()=====

The slice() method returns selected elements in an array, as a new array.

The slice() method selects from a given start, up to a (not inclusive) given end.

The slice() method does not change the original array.

splice()=====

The splice() method adds and/or removes array elements.

The splice() method overwrites the original array.

At position 2, remove 2 items:

```
const fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi"];
```

```
fruits.splice(2, 2);
```

At position 2, add 2 elements:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.splice(2, 0, "Lemon", "Kiwi");
```

```
array.splice(index, howmany, item1, ....., itemX)
```

index Required.

The position to add/remove items.

Negative value defines the position from the end of the array.

howmany Optional.

Number of items to be removed.

item1, ..., itemXOptional.

New element(s) to be added.

Return Value

An array containing the removed items (if any).

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.splice(2, 1, "Lemon", "Kiwi");
```

At position 2, add new items, and remove 1 item:

If pass howmany in 0 then no remove item only add item..

xceedence company MNC

```
for (let i = 0; i < 3; i++) {
```

```
  console.log('hello');
```

```
  setTimeout(function() {
```

```
    console.log('hi')
```

```
  }, 0)
```

```
  console.log('ok');
```

```
}
```

```
hello
```

```
ok
```

```
hello
```

```
ok
```

```
hello
```

```
ok
```

```
hi
```

```
hi
```

```
hi
```

```
let a = [1,2];
```

```
a[-1] = 10;
```

```
console.log(a[a.indexOf(1000)]);
```

```
let a = [1,2,3] + [4,5,6];
```

```
console.log(a);
```

```
const mobject = {
```

```
  mymethod(items) {
```

```
    console.log(this);
```

```
    const callback = () => {
```

```
      console.log(this);
```

```
    }
```

```
    items.forEach(callback);
```

```
  }
```

```
}
```

```
mobject.mymethod([1,2,3]);
```

```
var person = {
```

```
  name: "abhai",
```

```
  getName: function() {
```

```
    function xyz() {
```

```
      console.log(this)
```

```
    }
```

```
    xyz();
```

```
  }
```

```
}
```

```
person.getName();
```

