Tell me about your resume?

**Tell me about yourself?**

Well I am currently working with T-Mobile. It’s a mobile phone service provider company and here I am working as Front-end Developer. In which I am worked with HTML5, CSS3, JavaScript, jQuery and also working with angularJS in which I made custom directive, custom filter and also get the JSON data by using AJAX call. And also use the services to share JSON data in different controller. Also mad the page responsive using bootstrap and media queries.

Current client, what is it doing?? , Project: internal web application, custom directives

**Tell me about your last project you worked on?**

My recent project was for AbbVie. Coming to the project details it’s an internal web application **used for customer’s orders, sales tracking and marketing management**. The application will allow the authorize customer to login and place an order, **sales and distributed department keep track of it** and marketing management team also login via authorize login and track the sales and confirm that order is properly reach to the customer or not.

To make this portal we used latest technology like AngularJS and for the responsive designing part we used inbuilt framework which they made on based of bootstrap 3 and we also make it responsive which open in desktop, tablet ad mobile tool. To get the data from the server we used Restful API in which we get the data in JSON format.

My role as UI developer.

* I was involved in working with system analysts to analyze all the necessary requirements by involving in discussion session.
* Later I was involved in designing and developing the GUI for the user interface with various controls.
* I implemented business logic and rules by writing code for classes and functional methods in core JavaScript and jQuery.
* As a part of development I had to write a code for jQuery AJAX to fetch and manipulate the data in database accordingly.
* Also when working with AngularJS I made custom directives, custom filters and to get the data from the JSON file we used an AJAX call. Also worked with services to share the data.
* The development lifecycle was complete Agile process where we used daily scrum to resolve any blockers.
* I also work with QA engineers in their testing efforts and resolved the bugs reported by the testers.

Sometimes if I stuck with an issue than I exhaust instantly

**What is your most challenging task you faced? How you solved it?**

The software development itself was the most challenging task. We have to create the project within a specific timeslot and we have to choose the correct solution for the problem like choosing the design patterns for the problem etc. The last project was challenging to understand the requirement from the different level of customers and solving the problem by creating the project within time was more interesting.

What your strong point and week point?

**AngularJS**

**Angular Advantage disadvantage??**

Advantage:

* It provide the **capability to create a single page application** in a very clean and maintainable way.
* It has also capability to provide data binding to HTML. So, user can get rich and responsive design.
* It’s unit testable.
* AngularJS provide reusable components.
* In AngularJS, developer achieve more functionality with less code.
* Two way data binding.
* Follow the MVC structure.

Disadvantage:

**1. Not secure**: Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.

**2. Not degradable**: If the user of your application disables JavaScript, then nothing would be visible, except the basic page.

**What is RESTful web services?**

* RESTful Web Services are REST architecture based web services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web based applications.

**Controller:** Controller is defined by a JavaScript **constructor function.** AngularJS controllers **control the data** of AngularJS applications

The **ng-controller** directive defines the application controller

AngularJS will **invoke the controller** with a **$scope** object.

Use controllers to:

* Set up the initial state of the **$scope object**.
* **Add behavior to the $scope** object.

Do not use controllers to:

* Manipulate DOM — Controllers should contain only business logic. Putting any presentation logic into Controllers significantly affects its testability. Angular has [databinding](https://docs.angularjs.org/guide/databinding) for most cases and [directives](https://docs.angularjs.org/guide/directive) to encapsulate manual DOM manipulation.
* Format input — Use [angular form controls](https://docs.angularjs.org/guide/forms) instead..
* Filter output — Use [angular filters](https://docs.angularjs.org/guide/filter) instead.
* Share code or state across controllers — Use [angular services](https://docs.angularjs.org/guide/services) instead.
* Manage the life-cycle of other components (for example, to create service instances).

**Directives:** Directives are the core feature of AngularJS. Directives allow us to extend the web through reusable HTML elements, attributes, and classes.

AngularJS has a set of built-in directives which offers functionality to your applications. The built in directive starts with prefix “ng”. Such as ng-app, ng-init, ng-model, ng-repeat,

AngularJS also lets you define your own directives.

Directives are where you **manipulate the**

**DOM and catch DOM events**. This is why the directive's compile and link functions both receive the "element" as an argument. You can

* define a bunch of HTML (i.e., a template) to replace the directive
* bind events to this element (or its children)
* add/remove a class
* change the text() value
* Watch for changes to attributes defined in the same element (actually it is the attributes' values that are watched -- these are scope properties, hence the directive watches the "model" for changes) etc.

**What is custom directive? What are its kinds? Give example**

**Custom Directives:-**

New directives are created by using the .directive **function**.

To invoke the new directive, make an HTML element with the same tag name as the new directive.

When **naming a directive**, you must use **a camel case name**, w3TestDirective, but when invoking it, you must use - separated name, w3-test-directive:

You can invoke a directive by using:

* Element name
* Attribute
* Class
* Comment

Example: I made an

**$scope:** [Scope](https://docs.angularjs.org/api/ng/type/$rootScope.Scope) is an **object that refers to the application** **model**. It is the **binding part between the HTML (view) and the JavaScript** (controller).

The scope is an object with the available properties and methods.

The scope is **available for both the view and the controller**.

If we consider an AngularJS application to consist of:

* View, which is the HTML.
* Model, which is the data available for the current view.
* Controller, which is the JavaScript function that makes/changes/removes/controls the data.

Then the scope is the Model.

The scope is a JavaScript object with properties and methods, which are available for both the view and the controller.

**Services:-**

Angular services are substitutable objects that are wired together using [dependency injection (DI)](https://docs.angularjs.org/guide/di). You can use services to organize and share code across your app

In AngularJS you can make your own service, or use one of the many built-in services. Such as $location, $http, $timeout, $interval etc.

Suppose if we want to use data of 1 page on different page than we have to store in services and parse throughout website.

**Expression**

Angular expressions are **JavaScript-like code snippets**

AngularJS expressions can be written inside double braces: {{ expression }}.

AngularJS expressions can also be written inside a directive: ng-bind="expression".

**Module:-**

An AngularJS module **defines an application.**

The module is a container for the different parts of an application.

The module is a **container for the application controllers**.

**Controllers always belong to a module** and can be **added as ng-controller =” ”**;

**Filters:-**

A filter formats the value of an expression to display required output to the user. They can **be used in view templates, controllers or services** and it is easy to define your own file.

**What is 2-way binding?**

When you make changes in controller, it will get reflected in view and vice versa.

For example, you might have an object with a first name as one of the fields. This name would show up as the text of, say, an *input*element, and if the user changes the text in the *input*element, it will also change the data (first name field) in the model (object).

**How to achieve single binding?**

**How to do calls in AngularJS?**

**In form validation explain about**

**Ng-min length:**

**Ng-max length:**

**$error:**

**$valid:** input is valid

**$invalid:** input is invalid

**$dirty:** user changed something

**$pristine:** plain form- user has not entered anything

**How u deal with error in AngularJS?**

**How we know its Angularjs app?**

**NPM:**

**What is it?**

**What is bower?**

**What is grunt? Why we use it?**

**GIT: What if your co-worker works on same branch and push code?**

**Ans: Conflict will happen & it can be solved manually.**

**JAVASCRIPT:**

**Promise?? Why we use it?? Different methods of it??**

**Stringify??**

**Call back??**

**Closet vs paraent**

**Parent method in jQuery?**

**HTML:**

**What is new in HTML5?**

**Local storage? Why we use it? Advantage??**

**Web workers?**

**CSS:**

**Selectors in CSS?**

**Specify example**

**SASS why u used it? Features about SASS?**

**What kind of version in SASS??(gulp, node.js)**

Quirk mode and strict mode