

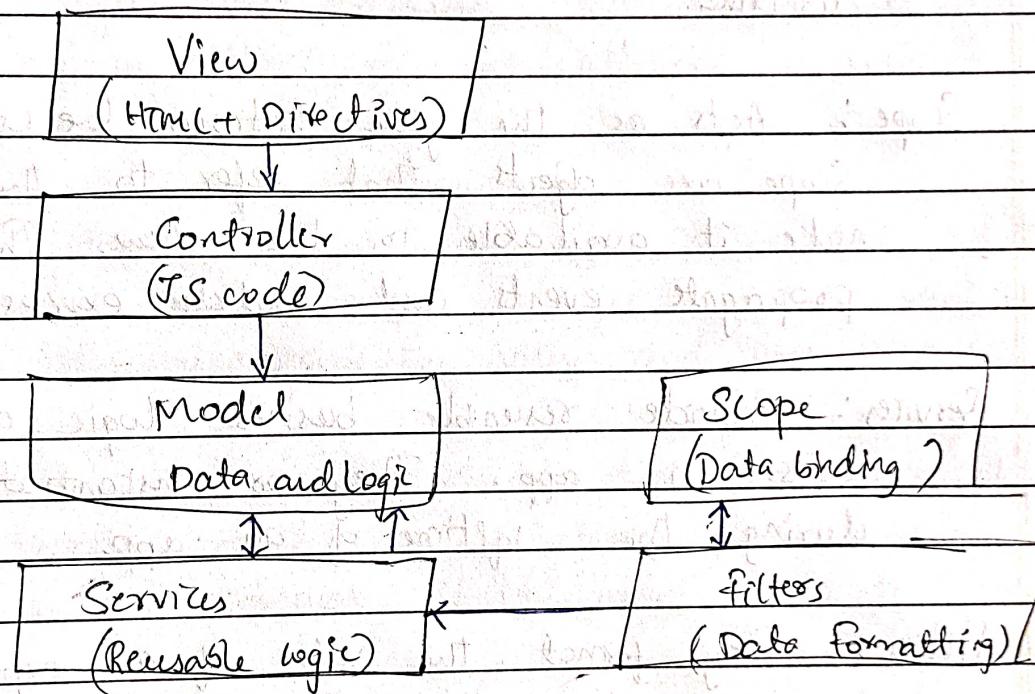
ASSIGNMENT

Date _____

Page _____

- 1) Explain the architecture of Angular.js

Soh. AngularJS is javascript framework which follows Model-View-Controller (MVC) pattern, although it's often referred to as Model-View-Whatever (MVW) due to its flexibility.



Model : The model is responsible for managing the data of the application. It can directly retrieve and manipulate data from the backend.

Models are typically plain JS objects or variables defined in controllers and services.

View : The view is made up of HTML templates that display the data from the model to the user. They are defined using directives that bind the data from model to DOM elements.

Controller: Manages the communication between the model and view. Controllers are JS functions or objects that are responsible for setting up the initial state of the scope and adding behavior to the scope. They are defined using ng-controller directive.

Scope: Acts as the glue between the controller and view. Scope are objects that refer to the model and make it available to the view. They also propagate events and watch expressions.

Services: Provide reusable business logic and data access across the app. They are instantiated only once during the lifetime of an app.

Filters: They format the value of an expression for display to the user. They can be used in view templates, controllers, or services to transform data such as currency, date, etc.

2) How does an Angular application work?

The application works by integrating its architecture and binding data seamlessly. Here is the flow:

1) Initialization: The app begins with a HTML template and includes the AngularJS library and defines the Angular JS app module. The ng-app directive is placed in the root HTML element.

- 2) Modular Declaration: AngularJS module is created using the "angular.module" method which acts as the main container for the app component.
- 3) Controller definition: Controllers are defined to manage the app logic and initialize data with the scope. The "ng-controller" directive attaches a controller to a part of the view.
- 4) Data Binding: The "\$scope" object binds data between the controller and the view. changes in the model automatically updates the view and vice versa.
- 5) Directives and Expressions: directives like "ng-model" and "ng-repeat" extend HTML functionality. AngularJS expressions "{{}}" bind data to HTML elements.
- 6) Services and dependency injection: Services provide reusable logic and data. They are injected into controllers using AngularJS's dependency injection mechanism.
- 7) Filters: Format the data to displaying it in the view.
- 8) Routing: The "ngRoute" module allows for navigation between different views without reloading the entire page.
- 9) Digest Cycle: AngularJS uses a digest cycle to monitor changes to the model and update the view accordingly.

Example: index.html

```
<!DOCTYPE html>
<html> <head>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.js">
<script src="app.js" type="text/javascript">
</head>
<body ng-app="myApp">
<div ng-controller="MainCtrl">
<p>{{ message }}</p>
<input type="text" ng-model="userInput" placeholder="Type Something..."/>
<p>User Input: {{ userInput }}</p>
<p>Formatted Amount: {{ amount | currency }}</p>
</div> </body> </html>
```

app.js

```
var app = angular.module('myApp', []);
app.service('DataService', function() {
  this.getMessage = function() {
    return "Hello, AngularJS";
  };
});
```

```
app.controller('MainCtrl', function($scope, DataService) {
  $scope.message = DataService.getMessage();
  $scope.amount = 1234.56;
});
```

3.

What are directives in Angular?

Exh

They are special markers on a DOM element that tell AngularJS's HTML Compiler (`$compile`) to attach a specified behavior to that DOM element

Types of directives

- 1) Attribute directive - Apply behavior to existing elements
- 2) Element directive - Define new custom HTML elements
- 3) Class directives - Apply behavior based on CSS class name
- 4) Comment directives - Apply behavior to comments in HTML

Common built-in directives

- 1) `ng-app` → bootstraps an AngularJS app
- 2) `ng-bind` → binds `innerHTML` to model property
- 3) `ng-model` → binds the value of HTML controls to app data
- 4) `ng-repeat` → repeats an element for each item in collection
- 5) `ng-if` → removes and recreates an element based on condition
- 6) `ng-show` / `ng-hide` → shows or hides an element
- 7) `ng-class` → dynamically binds one or more CSS classes to HTML element.

4) Explain Components, Modules and Services in Angular

Components: They are a special type of directive that uses a simpler configuration which is suitable for defining a component based app. structure

They are part of AngularJS 1.5+ and provide reusable

UI

* It encapsulates view and behavior

* It is defined using "component" method

```
var app = angular.module('myApp', []);
app.component('myComponent', {
  template: '<h1>Hello, {{ctrl.name}}!</h1>',
  controller: function() {
    this.name = 'world';
  }
});
```

In HTML

```
<body ng-app="myApp">
<my-component> </my-component>
</body>
```

Modules

They serve as containers for different parts of the application, such as controllers, services, directives, filters and configurations. They help organize the application into cohesive blocks of functionality.

* Organize into logical units and manage dependencies.

* Modules are created using "angular.module" method.

```
var app = angular.module('myApp', []);
app.controller('MainCtrl', function($scope) {
  $scope.message = "Hello, AngularJS!";
});
```

In HTML

```
<body ng-app="myApp">
```

```
<div ng-controller = "MainCtrl">
  <p> {{ message }} </p>
</div>
</body>
```

Services:

They are singleton objects that are used to organize and share code across the app. They are typically used for data retrieval, business logic or other reusable functions.

Services are injected into controllers, directives, filters, and other services using Angular's dependency injection.

They are defined using the "service", "factory" or "provider".

```
var app = angular.module('myApp', []);
app.service('DataService', function() {
  this.getData = function() {
    return "Some data";
  };
});
```

```
app.controller('MainCtrl', function($scope, DataService) {
  $scope.message = DataService.getData();
});
```

HTML:

```
<body ng-app = "myApp">
  <div ng-controller = "MainCtrl">
    <p> {{ message }} </p>
  </div>
</body>
```

5 What are some of the advantages of Angular over other frameworks?

Sohi:

- 1) Two-way Data binding: Synchronizes model and view automatically
- 2) Dependency injection: Simplifies managing dependencies, promoting better code organization
- 3) Component-Based Architecture: Enhances reusability and maintainability with self-contained components.
- 4) Comprehensive Ecosystem: Provides a complete set of tools and libraries for front-end development
- 5) Strong TypeScript Support: Improves code quality and tooling with TS's static typing.
- 6) Rich Template Syntax: Allows dynamic and complex UI creation
- 7) Powerful Routing: Facilitates SPAs (Single Page Apps) with features like lazy loading and nested routes
- 8) Testing Capabilities: Built-in support for unit and end-to-end testing.