FORTRESS
CAREER

ANGULAR JS

What Angular JS is

- Definition and overview
- History and evolution
- Features of AngularJS
- MVC architecture in AngularJS
- Advantages and disadvantages

What are directives in AngularJS?

- Introduction to directives
- Types of built-in directives
- Role of directives in DOM manipulation
- Using directives in templates

Creating custom directives

- Directive definition object (DDO)

- Restrict options (Element, Attribute, etc.)
- Template and templateUrl
- Scope (Shared, Isolated, Inherited)
- Compile and link functions
- Transclusion
- Best practices for custom directive

Controllers

- Purpose of controllers
- Creating controllers
- $scope object usage
- Controller as syntax
- Nested controllers
- Best practices

Two-Way Data Binding

- Concept and overview
- How it works in AngularJS
- Binding expressions
- $digest cycle and $apply
- Performance considerations

Understanding two-way data binding

- Scope object and model synchronization
- ng-model directive
- Change detection mechanism
- Dirty checking
- Watchers and listener functions

Services

- What services are in AngularJS
- Built-in services (e.g., $http, $location)
- Singleton nature of services
- Using services in controllers and directives
- Custom services

Dependency Injection (DI)

- What is DI
- How DI works in AngularJS
- Injector and provider
- Dependency annotation
- Minification safe DI
- Lazy instantiation **Modules**

- Defining AngularJS modules

- Dependency management in modules
- Registering components (controllers, services, etc.)
- Config and run blocks
- Best practices for module structure **Filters**

- Purpose of filters
- Built-in filters (filter, currency, date, etc.)
- Chaining filters
- Creating custom filters
- Using filters in controllers and services

Routing (ngRoute)

- What is routing in AngularJS
- Configuring routes with $routeProvider
- TemplateUrl and controllers in routes
- Route parameters
- Nested views and partials
- Route guards and resolvers

Promises and Asynchronous Programming

- What are promises
- $q service in AngularJS
- Creating and resolving promises
- Chaining promises
- Handling errors with promises

- Practical use cases (e.g., API calls with

$http) **Forms in AngularJS**

- Creating forms
- ng-model and form controls
- Form validation (built-in and custom)
- ngForm directive
- Error handling and displaying messages
- Submitting forms

Custom Events and Broadcasting

- $emit, $broadcast, and $on
- Event propagation in AngularJS
- Parent-child scope communication
- Use cases of custom events
- Performance considerations

Testing in AngularJS

- Importance of testing
- Unit testing controllers, services, and directives
- Using Jasmine framework
- End-to-end testing with Protractor
- Mocking dependencies and services
- Test-driven development (TDD) with AngularJS

Animations in AngularJS

- ngAnimate module overview
- Applying animations to directives (ngShow, ngHide, ngRepeat)
- CSS and JavaScript-based animations
- Animation events (beforeAddClass, afterAddClass, etc.)
- Best practices for smooth animations
- Performance considerations