UNIT TESTING AND TOOLS IN ANGULARJS

JavaScript is a clientside language which makes the web page dynamically changed, but we cannot compile the JavaScript code. For this reason, the JavaScript code should be tested with strong set of tests. But in angularjs, we have built many features for testing a angularjs application. In AngularJS, there is no chance of making an application without testing.

Separation of Concerns:

- Unit testing is about testing individual unit of code.
- Unit tests try to answer the following questions
  - ➢ Did I think about the logic correctly?

    (or)

  - ➢ Does the sort function order the list in the right order?
- The AngularJS isolate the unit of code under test for answer the above question.
- So, we don't want to be forced to creating a related part such as the DOM elements or XHR calls to fetch the data to sort.
- The AngularJS provide the dependency injection for XHR request, so our request can be simulated.
- In AngularJS, we can test our model without having to manipulate the DOM directly. So, our tests have been sorted without having to create or look at the state of the DOM or to wait for any XHR request to return data.
- In AngularJS, the individual sort function can be tested in isolation.

Additional tools for testing AngularJS Applications:

- For testing AngularJS applications, there are certain tools that will make testing much easier to set up and run. They are Karma and Jasmine.

Karma

- Karma is a JavaScript command line tool that can be used to offspring a web server which loads our application's source code and executes our tests.
- This tool give the confident that our application runs on all the browsers we need to support.
- This tool is executed on the command line and will displays the result of our testes on the command line once they have run in the browsers.
- Karma is a NodeJS application, and should be installed through npm / yarn.

Jasmine

- Jasmine is a behavior driven development framework for JavaScript.
- Jasmine is one of the most popular choice for testing AngularJS applications.
- This tool provides functions to help with structuring our testes and also making assertions.
- Jasmine helps achieve our tests grow, keeping them well-structured and documented is vital.
- In jasmine, the describe function is used to group our tests together:

```
describe (“sorting the users”, function () {
          // individual testes go here
});
```

- And then each individual test is defined within a call to the it function

```
describe ('sorting the users', function () {
      it ( 'sorts in descending order by default', function() {
         // our test assertion goes here
});
});
```

- The **describe** block and **it** call is used to keep our tests self-documenting. Here the grouping related testes within **describe** block and describing each individual test within an **it** call.
- Finally, Jasmine provides matchers which let you make assertions:

```
describe ('sorting the users', function () {
      it ( 'sorts in descending order by default', function() {
         var user=[ 'adam' , 'leo', 'angel'];
         var sort= sortUsers(user);
         epect(sort).toEqual ( [ 'leo', 'angel', 'adam' ] );
});
});
```

- Jasmine comes with a number of matchers that help you make a variety of assertions.
- To use jasmine with karma, use the karma-jasmine test runner.

Angular-mocks

- In AngularJs the ngMock module is used to provides mocking for our tests.
- The ngMock is used to inject and mock AngularJS service within unit tests and also it is able to extends other modules so they are synchronous.
- If the testes having synchronous keeps the testing much cleaner and easier to work with.

- One of the most valuable parts of ngMock is $httpBackend, which is useful for our mock XHR requests in tests, and return sample data instead.