# Q1. Which testing tool is good for your project and why (write full description and give some example)

Ans. There are some tools used for unit testing in Angular applications. These tools are categorised into three groups. They are:

# Mocha:

This tool runs on node.js and is used with browsers for asynchronous testing. A feature-rich JS framework, Mocha test runs serially mapping uncaught exceptions to correct test cases. It has flexible and accurate reporting.

# Siesta:

Siesta is a unit testing and UI testing tool, written as a generic tool in javaScript. It’s able to test both Node.js processes and also web pages. Siesta supports all major browsers.

# Jasmine:

Jasmine is a Behaviour Driven Development (BDD) testing framework. Among its advantages, the main ones are that it doesn’t rely on other javaScript frameworks. It also doesn’t require DOM. One of the main goals of Jasmine is to provide an easy syntax, allowing you to write tests more easily. Jasmine is the default test framework used with Angular. It ships with Angular CLI by default.

# 4.Jasmine:

An open source framework developed and maintained by the GitHub community. This is the most compatible test runner for Angular known for testing on real devices. Karma provides continuous integration with various browsers with easy debugging directly from IDE.

### 5.Angular Mock:

A unit test expert which is Namespace from angular.mocks.js test related code. It is important to isolate the unit code which is under test for Angular 5. The inject method of Angular mock enables us to inject named Angular dependencies.

# Examples:

1. describe(‘Good Morning’, () => { (1) it(‘says hello’,() =>{ (1)

expect(helloWorld()).toEqual(‘Good Morning’);

});

});

1. describe(“simple test”, function()

{

beforeEach(function() {

});

afterEach(function() {

});

it (“a is a string”, function(){

})

})

1. it(‘should have defined component’, () =>{ expect(component).toBeDefined();

});

# Q2 . Write five sample unit test cases in Karma or Jest. Ans. Example 1:

beforeEach(async(() => { TestBed.configureTestingModule({ declarations: [ AppComponent ],

}).compileComponents();

}));

**Example 2:**

it('should create the app', async(() => {

const fixture = TestBed.createComponent(AppComponent); const app = fixture.debugElement.componentInstance; expect(app).toBeTruthy();

}));

**Example 3:**

it(`should have as title 'angular-unit-test'`, async(() => {

const fixture = TestBed.createComponent(AppComponent); const app = fixture.debugElement.componentInstance; expect(app.title).toEqual('angular-unit-test');

}));

**Example 4:**

it('should render title in a h1 tag', async(() => {

const fixture = TestBed.createComponent(AppComponent); fixture.detectChanges();

const compiled = fixture.debugElement.nativeElement; expect(compiled.querySelector('h1').textContent).toContain('Welcome to angular-unit-test!');

}));

**Example 5:**

it("should create Quote component", () => {

expect(component).toBeTruthy();

});