**Assignment 1**

1. Write brief description about unit testing and functional testing and its benefits as developer’s perspective?

**Unit testing -** A Unit test is the procedure to check the proper functioning of a specific part of a software or a portion of a program. This is sometimes also called Isolated testing. Unit tests can be used to test the services by invoking the functionality directly. Unit testing can be used to invoke and test the behavior of a piece of code in isolation. The reusable logic written in services requires this kind of testing, as unit testing provides ways to test all possible scenarios by sending different types of data to the service methods. The most crucial tools used for Angular unit testing—Karma and Jasmine.

**Benefits of Unit Testing -**

* Debugging is easy. When a test fails, only the latest changes need to be debugged.
* Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
* If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change.
* The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.

**Functional Testing** -As it's name defines ,it is the testing of the complete functionality of an application. This is also called End to End or E2E testing. E2E test put his goal to verify that all features and functionalities work correctly together.

End to End testing is a methodology used to test an application from a user's perspective. The tests ensure the application performs as expected from start to finish. As the tests run, you will see the browser interaction just as a user would use your application. Angular end to end tests are powered by a framework called Protractor.

**Benefits of Functional Testing -**

* Functional testing ensures that the system will work according to the requirements. Basically, it tells us that what is the problem in the functionality.

2.Where and why you need Unit testing in your project, give me 10 example and code snap?

**When and Why:** Unit tests enable programmers to change code more easily, with more confidence and less side effects. It promotes / enables refactoring of code which without unit tests can be dangerous. With all this in mind I find unit testing will make our code better.

**Code Snap:**

**1. it('should create', () => {**

**expect(component).toBeTruthy();**

**});**

**2. it('should create the app', () => {**

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

**const app = fixture.componentInstance;**

**expect(app).toBeTruthy();**

**});**

**3.it(`should have as title 'demo-app'`, () => {**

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

**const app = fixture.componentInstance;**

**expect(app.title).toEqual('demo-app');**

**});**

**4. it('should render title', () => {**

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

**fixture.detectChanges();**

**const compiled = fixture.nativeElement;**

**expect(compiled.querySelector('.content span').textContent).toContain('demo-app app is running!');**

**});**