Assignment-1

1)Write brief description about unit testing and functional testing and it's benefit in project, as a developer perspective?

What is Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

Unit Testing Techniques:

- Black Box Testing - Using which the user interface, input and output are tested.
- White Box Testing - used to test each one of those functions behaviour is tested.
- Gray Box Testing - Used to execute tests, risks and assessment methods.

Benefits of Unit Testing

- **Improve the design of implementations.**
  Start coding a feature without giving it a lot of thought to the design is a very common mistake among developers. Using unit testing is going to enforce you to think and re-think the design, and if you are using TDD the impact is even bigger.
- **Allows refactoring.**
  Since you already have tests that ensure that everything is working as expected, you can easily add changes to that code with the certainty that you are not adding any bugs.
- **Add new features without breaking anything.**
  When you are adding a new feature you can run the tests to ensure that you ain't breaking any other part of the application.

FUNCTIONAL TESTING is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Functional testing mainly involves black box testing and it is not concerned about the source code of the application. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test. The testing can be done either manually or using automation.

Benifits

- It ensures that the customer or end- user is satisfied.
- It produces a defect-free product/software.
- it ensures the all the requirements should be met.
- It ensures the proper working of all the functionalities of an application/software/product.
- It ensures that the software/product works as expected.
- It ensures security and safety.
- It improves the quality of the product.
- The risks and loss associated with the product/software reduced.

2)Where and Why we need the Unit Testing in our project, give me 10 example and code snap?

Unit Testing is the software testing technique where a group of software program components or modules are tested individually. This technique effectively helps in validating the accuracy of a section of code by considering stubs, mock objects, drivers, and unit testing frameworks. Since it is practiced at the initial testing phase, this testing technique assures to identify and fix the bugs at the early stage of SDLC even before they become expensive for the enterprises to fix when identified at a later stage.

Some developers may attempt to save time by performing minimal unit testing, or when unit testing is skipped, it obviously leads to higher defect fixing costs during system testing, integration testing, and even beta testing when the application is completed. Moreover, in addition to these, unit testing helps the development teams to understand the code base, validate the correctness of the developed code, reuse the code, and to make the changes faster in the code.

With a proper unit testing practice in place, the developers and testers can help to save time as bugs can be identified early in the process as it is the initial phase of testing. And, skipping or limiting the practice of unit testing can adversely increase the defects and it becomes complex to fix them at a later stage. Hence, it is essential to practice unit testing at the initial stage of the software testing process before planning for the integration testing.

```
1)
  it('should create the app', async(() => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app).toBeTruthy();
  }));

2)
  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');
  }));

3)
  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!');
  }));

4)
it('should truncate a string if its too long (>20)', () => {
    const pipe = new TroncaturePipe();
    const ret = pipe.transform('12345678901234567890123456789012345');
    expect(ret.length).toBeLessThanOrEqual(20);
  });
```

5)
```
it('should create', () => {

    const title = 'Hey there, i hope you are enjoying this article';
    const titleElement = element.querySelector('.header-title');
    component.title = title;
    fixture.detectChanges();
    expect(titleElement.textContent).toContain(title);
  });
```

6)
```
it('should have one user',()=>{
    expect(comp.users.length).toEqual(1);
});
```

7)
```
it('sorts in descending order by default', function() {
    var users = ['jack', 'igor', 'jeff'];
    var sorted = sortUsers(users);
    expect(sorted).toEqual(['jeff', 'jack', 'igor']);
  });
```

8)
```
it("should be able to lower case a string",function() {
        expect(utils.toLowerCase).toBeDefined();
        expect(utils.toLowerCase("HELLO WORLD")).toEqual("hello world");
  });
```

9)
```
it("should be able to upper case a string",function() {
    expect(utils.toUpperCase).toBeDefined();
    expect(utils.toUpperCase("hello world")).toEqual("HELLO WORLD");
  });
```

10)
```
  it("should be able to confirm if a string contains a substring",function() {
    expect(utils.contains).toBeDefined();
    expect(utils.contains("hello world","hello",0)).toBeTruthy();

  });
```