TESTING

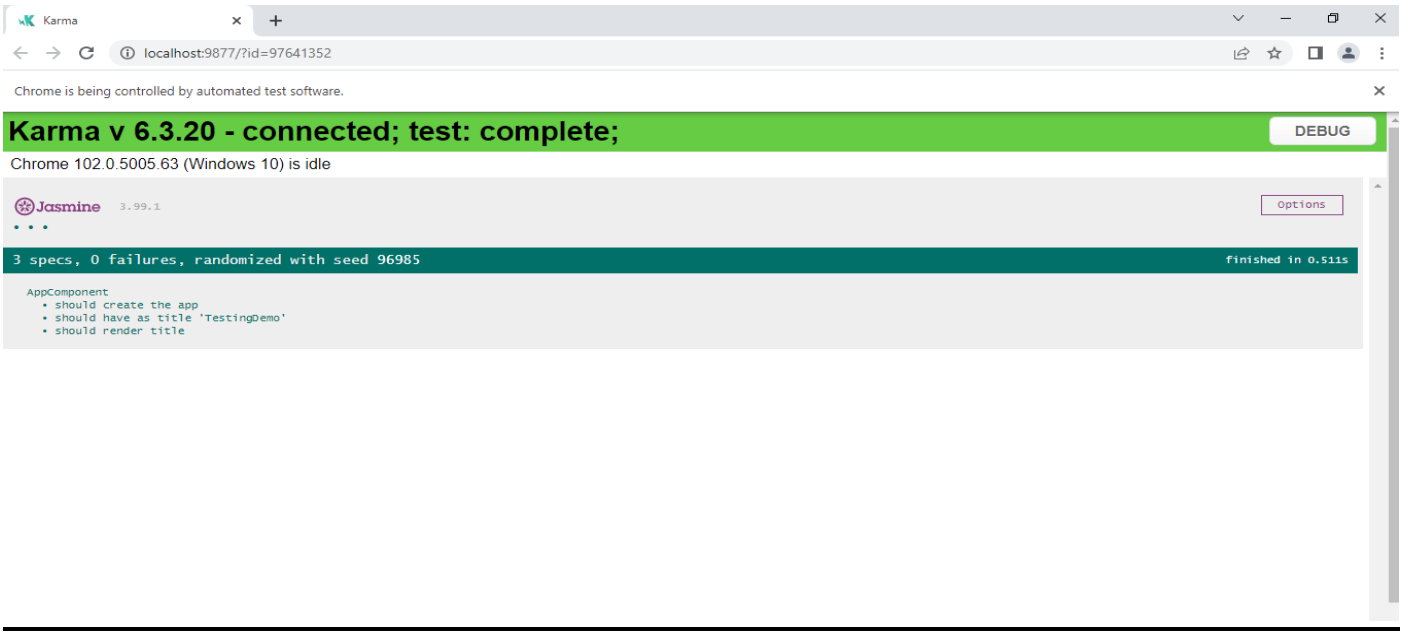
UNIT TESTING

DEFAULT TEST CASE:

```
import { TestBed } from
'@angular/core/testing';
import { RouterTestingModule } from
'@angular/router/testing';
import { AppComponent } from './app.component';
import{HttpClientTestingModule}from
'@angular/common/http/testing';
import { MasterService } from
'./service/master.service';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule,
        HttpClientTestingModule
      ],
      declarations: [
        AppComponent
      ],
      providers:[MasterService]
    }).compileComponents();
```

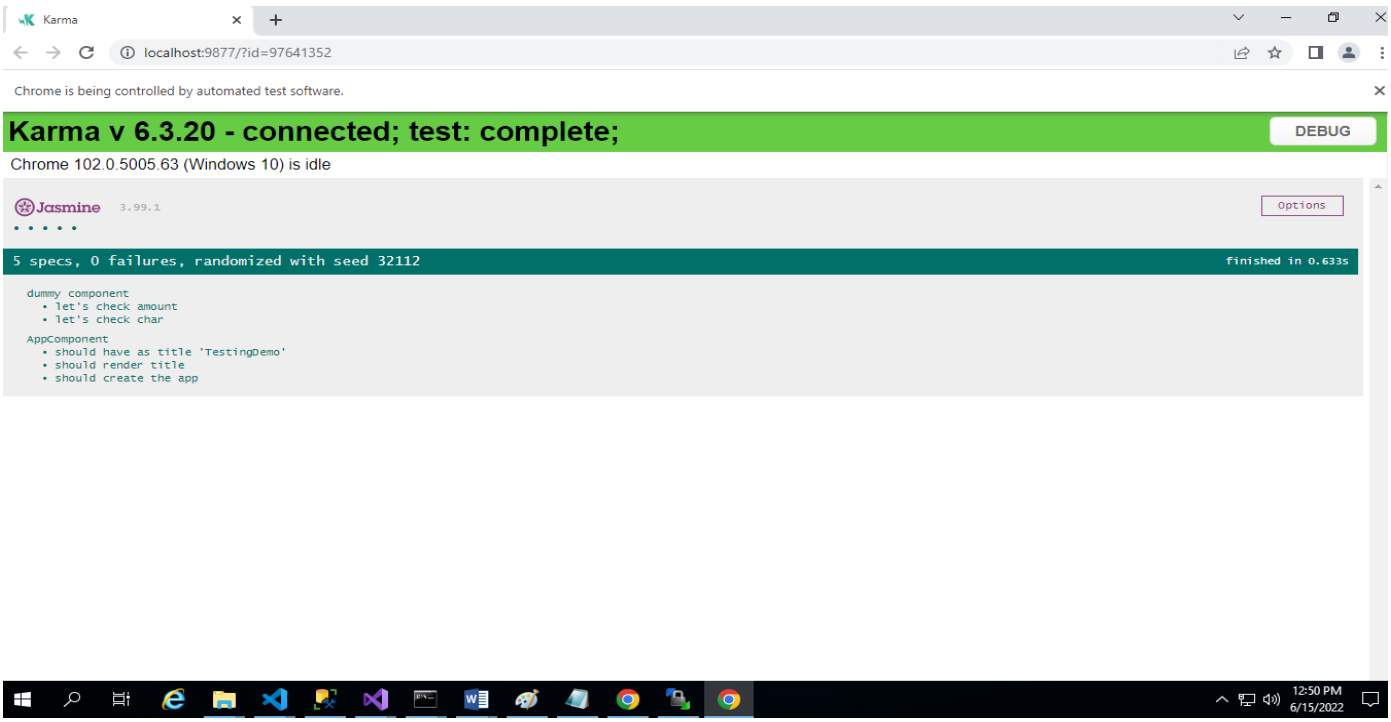
```
  });

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

  it(`should have as title 'TestingDemo'`, ()
=> {
    const fixture =
TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('TestingDemo');
  });

  it('should render title', () => {
    const fixture =
TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as
HTMLElement;
    expect(compiled.querySelector('.content
span')?.textContent).toContain('TestingDemo app
is running!');
  });
});
```

```
ngcc
CREATE TestingDemo/src/polyfills.ts (2338 bytes)
CREATE TestingDemo/src/styles.css (80 bytes)
CREATE TestingDemo/src/test.ts (745 bytes)
CREATE TestingDemo/src/assets/.gitkeep (0 bytes)
CREATE TestingDemo/src/environments/environment.prod.ts (51 bytes)
CREATE TestingDemo/src/environments/environment.ts (658 bytes)
CREATE TestingDemo/src/app/app-routing.module.ts (245 bytes)
CREATE TestingDemo/src/app/app.module.ts (393 bytes)
CREATE TestingDemo/src/app/app.component.html (23364 bytes)
CREATE TestingDemo/src/app/app.component.spec.ts (1088 bytes)
CREATE TestingDemo/src/app/app.component.ts (215 bytes)
CREATE TestingDemo/src/app/app.component.css (0 bytes)
√ Packages installed successfully.
'git' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\admin12\AngularExamples>CD TestingDemo

C:\Users\admin12\AngularExamples\TestingDemo>code ..

C:\Users\admin12\AngularExamples\TestingDemo>ng test
√ Browser application bundle generation complete.
15 06 2022 12:40:55.538:WARN [karma]: No captured browser, open http://localhost:9877/
15 06 2022 12:40:55.633:INFO [karma-server]: Karma v6.3.20 server started at http://localhost:9877/
15 06 2022 12:40:55.637:INFO [launcher]: Launching browsers Chrome with concurrency unlimited
15 06 2022 12:40:55.660:INFO [launcher]: Starting browser Chrome
15 06 2022 12:40:58.431:INFO [Chrome 102.0.5005.63 (Windows 10)]: Connected on socket rBXBr17_2Ts7sRLjAAAB with id 97641352
Chrome 102.0.5005.63 (Windows 10): Executed 3 of 3 SUCCESS (0.509 secs / 0.459 secs)
TOTAL: 3 SUCCESS
```



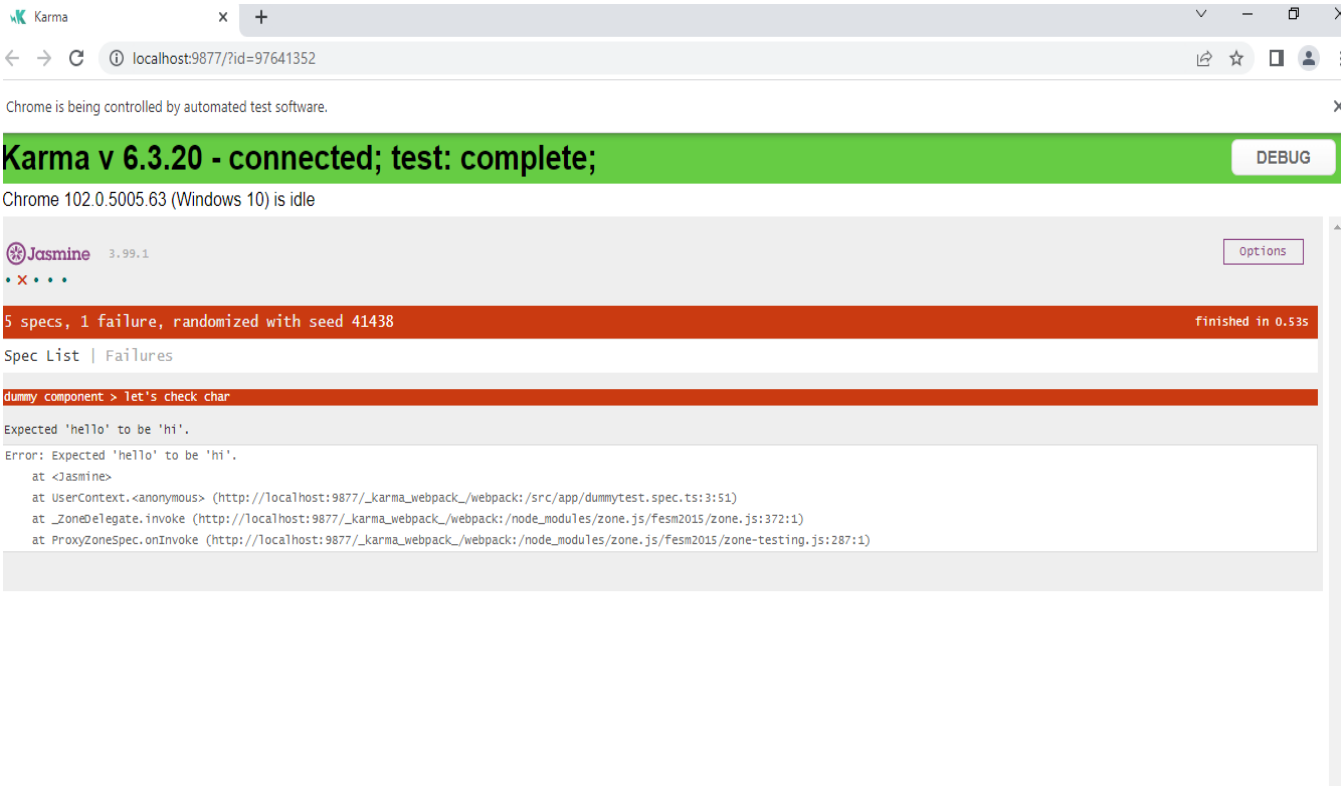
Dummytest.spec.ts (case 1:Success)

```
describe("dummy component",()=>{
        it("let's check
amount",()=>expect(100).toBe(100))
        it("let's check
char",()=>expect("hello").toBe("hello"))
    })
```



Dummytest.spec.ts (case2:Fail)

```
describe("dummy component",()=>{
        it("let's check
amount",()=>expect(100).toBe(100))
        it("let's check
char",()=>expect("hello").toBe("hi"))
    })
```

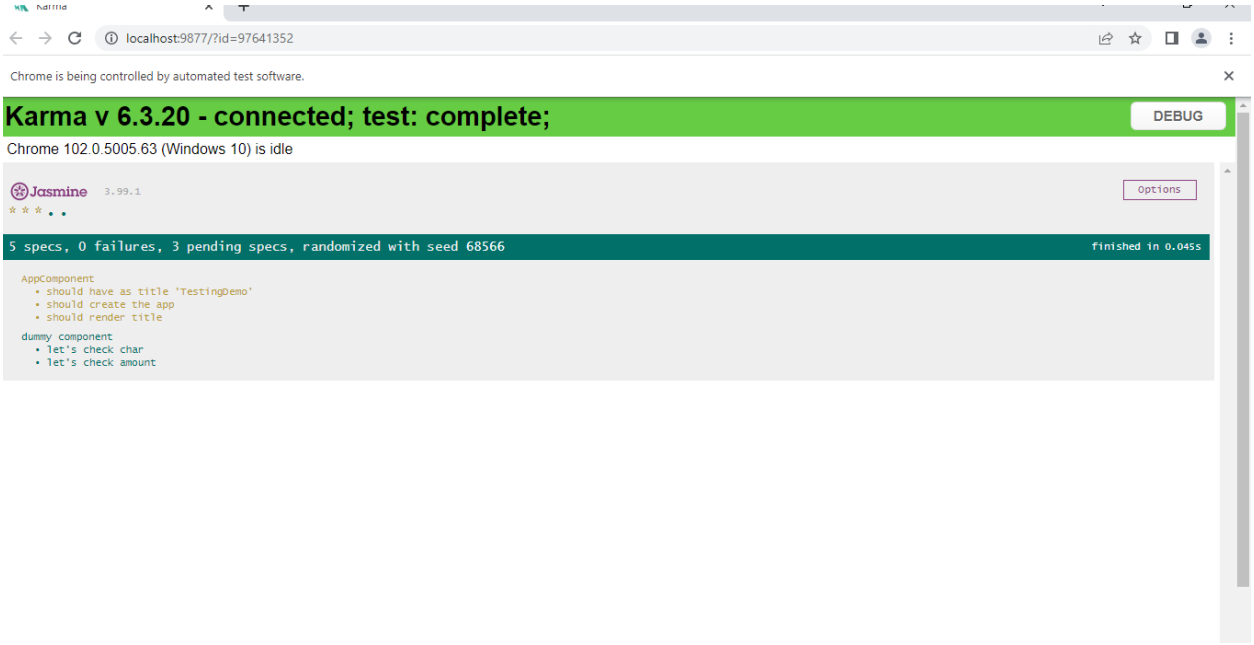


To exclude a particular test case:

```
import { TestBed } from
'@angular/core/testing';
import { RouterTestingModule } from
'@angular/router/testing';
import { AppComponent } from
'./app.component';
import{HttpClientTestingModule}from
'@angular/common/http/testing';
import { MasterService } from
'./service/master.service';

xdescribe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule,
        HttpClientTestingModule
      ],
      declarations: [
        AppComponent
      ],
      providers:[MasterService]
    }).compileComponents();
  });
```

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

  it(`should have as title 'TestingDemo'`,
() => {
    const fixture =
TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('TestingDemo
');
  });

  it('should render title', () => {
    const fixture =
TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement
as HTMLElement;
    expect(compiled.querySelector('.conten
t
span')?.textContent).toContain('TestingDem
o app is running!');
```

```
  });
});
```

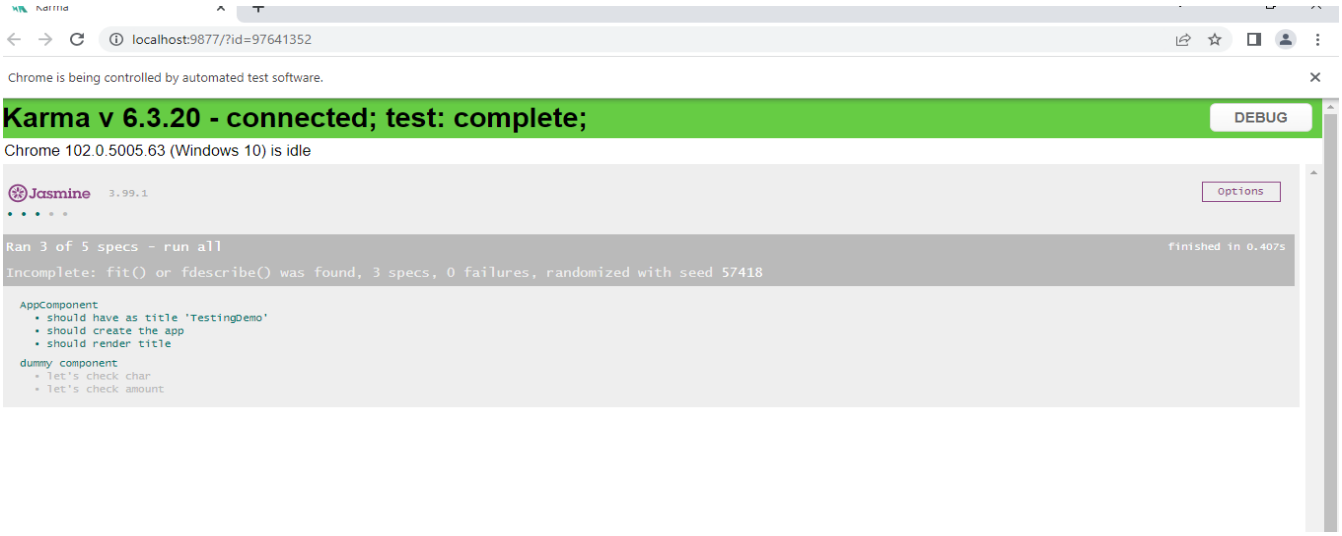


To execute a particular test case specifically :

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from
'@angular/router/testing';
import { AppComponent } from './app.component';
import{HttpClientTestingModule}from
'@angular/common/http/testing';
```

```
import { MasterService } from
'./service/master.service';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule,
        HttpClientTestingModule
      ],
      declarations: [
        AppComponent
      ],
      providers:[MasterService]
    }).compileComponents();
  });

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

  fit(`should have as title 'TestingDemo'`, ()
=> {
    const fixture =
TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('TestingDemo');
```

```
  });

  it('should render title', () => {
    const fixture =
TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as
HTMLElement;
    expect(compiled.querySelector('.content
span')?.textContent).toContain('TestingDemo app
is running!');
  });
});
```



Service Testing:

Order.service.spec.ts

```
import { TestBed } from
'@angular/core/testing';
import { products } from
'../models/product';
import { OrderService } from
'./order.service';
```

```
import { UserService } from
'./user.service';

describe('OrderService', () => {
  let service: OrderService;

  const userServiceSpy =
jasmine.createSpyObj<UserService>('UserSer
vice', ['getActiveUser']);
  beforeEach(() => {
    TestBed.configureTestingModule({});
    service =
TestBed.inject(OrderService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

Order.service.ts

```
import { Injectable } from
'@angular/core';
```

```
import { UserService } from
'./user.service';
import { order } from '../models/order';
import { products } from '../models/product';

@Injectable({
  providedIn: 'root'
})
export class OrderService {

  constructor(private readonly
userService: UserService) {}
createOrder(product:products):order {
return {
id: Date.now().toString(),
user: this.userService.getActiveUser(),
product
};
}
}
```

User.service.spec.ts

```
import { TestBed } from
'@angular/core/testing';
import { user } from '../models/user';
```

```
import { UserService } from
'./user.service';

describe('UserService', () => {
  let service: UserService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UserService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });

  it('should set the active user
correctly', () => {
    // Arrange
     const user:user = {
    id: 'test',
    name: 'test'
    };
    // Act
    service.setActiveUser(user);
    // Assert
    expect(service['activeUser'].id).toEqu
al('test');
```

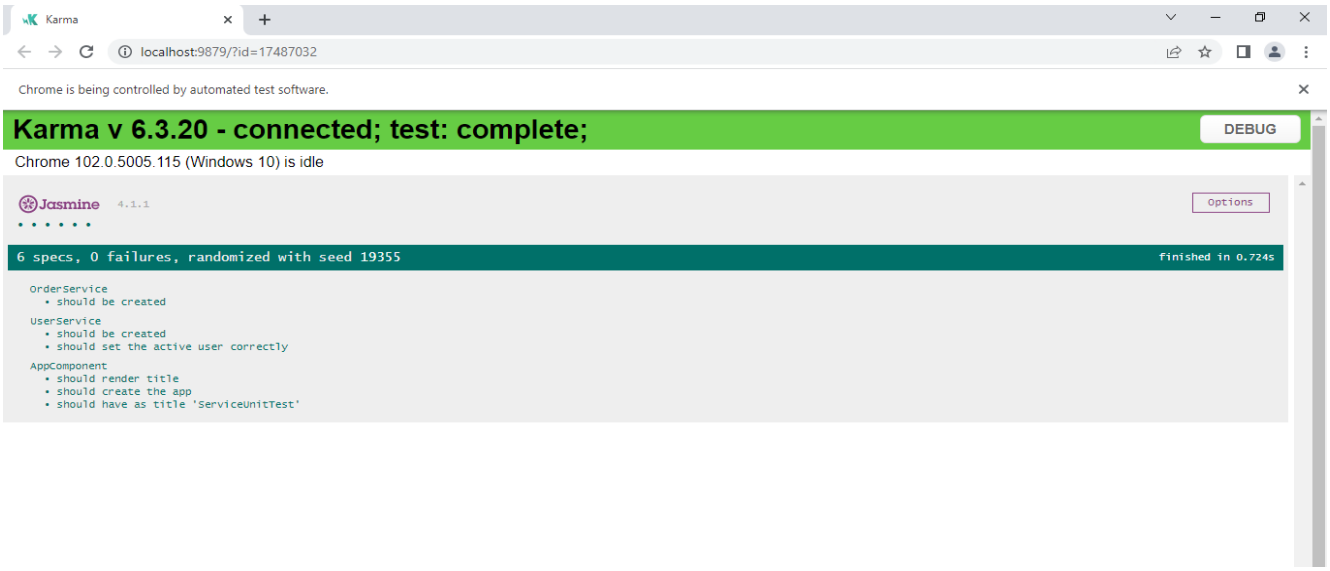
```
    expect(service['activeUser'].name).toE
qual('test');
    });

});
```

User.service.ts

```
import { Injectable } from
'@angular/core';
import { user } from '../models/user';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  private activeUser!: user;
  constructor() { }
  getActiveUser(){
  return this.activeUser as
Readonly<user>;
  }
  setActiveUser(user:user){
  this.activeUser=user;
  }
```

```
}
```

Output


Karma
localhost:9879/?id=17487032
Chrome is being controlled by automated test software.
Karma v 6.3.20 - connected; test: complete;
DEBUG
Chrome 102.0.5005.115 (Windows 10) is idle
Jasmine 4.1.1
Options
6 specs, 0 failures, randomized with seed 19355
finished in 0.724s
OrderService
• should be created
UserService
• should be created
• should set the active user correctly
AppComponent
• should render title
• should create the app
• should have as title 'ServiceUnitTest'

WHAT IS ANGULAR UNIT TESTING?

Unit testing in angular refers to the process of testing individual units of code. An Angular unit test aims to uncover issues such as incorrect logic, misbehaving functions, etc. by isolating pieces of code. This is sometimes more difficult than it sounds, especially for complex projects with poor separation of concerns. Angular is designed to help you write code in such a way that enables you to test your apps functions individually in isolation.

WHY IS UNIT TESTING NEEDED IN ANGULAR APPS?

Angular unit testing enables you to test your app based on user behavior. While testing each possible behavior would be tedious, inefficient, and ineffective, writing tests for each coupling block in your application can help demonstrate how these blocks behave. One of the easiest ways to test the strengths of these blocks is to write a test for each one. You don’t necessarily need to wait until your users complain about how the input field behaves when the button is clicked. By writing a unit test for your blocks (components, services, etc.), you can easily detect when there is a break.

FRAMEWORKS IN ANGULAR UNIT TESTING?

Karma is a JavaScript test runner that runs the unit test snippet in Angular. Karma also ensures the result of the test is printed out either in the console or in the file log.By default, Angular runs on Karma. Other test runners include Mocha and Jasmine. Karma provides tools that make it easier to call Jasmine tests while writing code in Angular