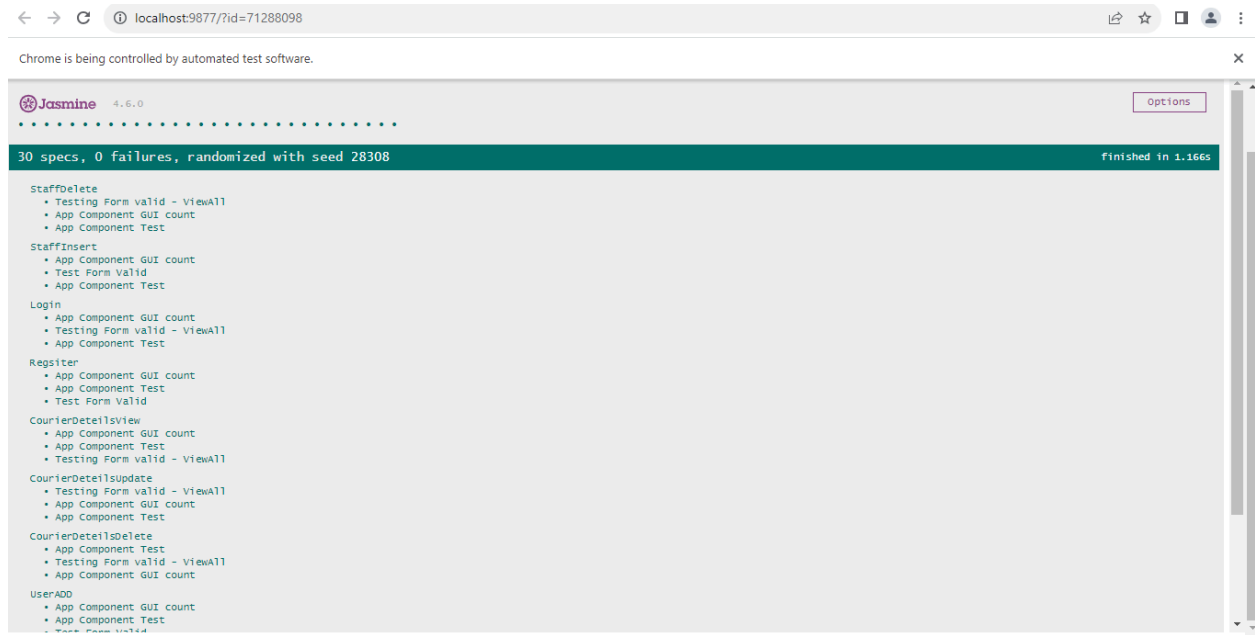
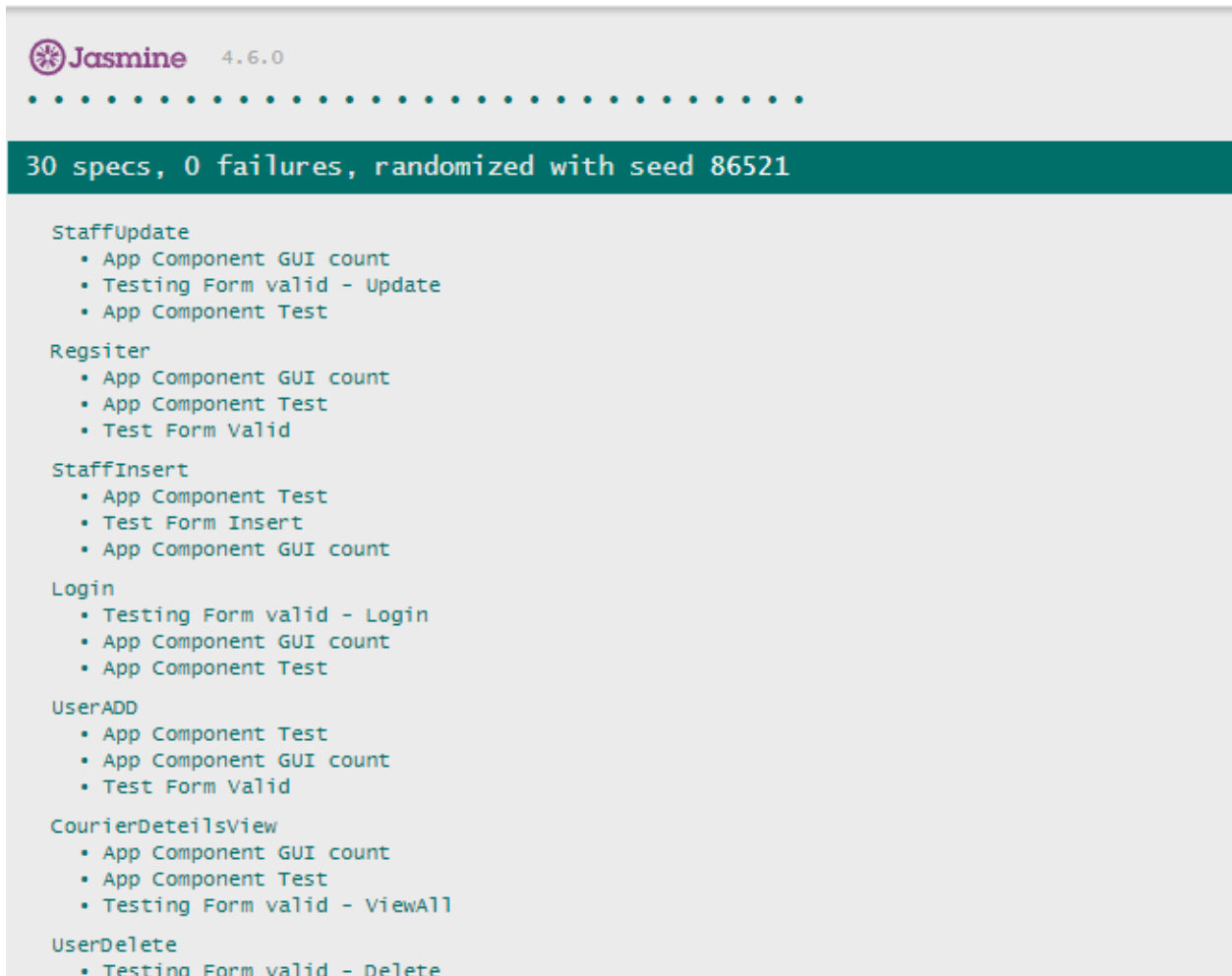


Employee Name:Muthuraj Periyasamy
Employee Id:11949

Jasmine/Karma





```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';
import { RegisterComponent } from './register/register.component';
import { AppRoutingModule } from './app-routing.module';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { LoginComponent } from './login/login.component';
import { UserComponent } from './user/user.component';
```

```

import { ViewComponent } from '../view/view.component';
import { StaffComponent } from '../staff/staff.component';
import { Parcel } from '../model/Parcel';

describe('AppComponent', () => {
  beforeEach(() => TestBed.configureTestingModule({
    imports: [RouterTestingModule],
    declarations: [AppComponent]
  }));

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

describe('Register', () => {
  let component: RegisterComponent;
  let fixture: ComponentFixture<RegisterComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        RegisterComponent
      ],
      imports: [
        BrowserModule,
        AppRoutingModule,
        ReactiveFormsModule,
        HttpClientModule,
      ]
    }).compileComponents();
  });

  it('Regsiter Component Test', () =>{
    const fixture = TestBed.createComponent(RegisterComponent);
    const app = fixture.nativeElement as HTMLElement;
    expect(app).toBeTruthy();
  })

  it('Register Component GUI count', () =>{

```

```

const fixture = TestBed.createComponent(RegisterComponent);
const app = fixture.nativeElement as HTMLElement;
const inputElement = app.querySelectorAll("input");
const buttonElement = app.querySelectorAll("button");
expect(inputElement.length).toEqual(6);
expect(buttonElement.length).toEqual(1);
})

it('Test Form Valid', () =>{
  const fixture = TestBed.createComponent(RegisterComponent);
  const User = fixture.componentInstance;
  User.myForm.controls?.['id'].setValue("1");
  User.myForm.controls?.['username'].setValue("Raja");
  User.myForm.controls?.['email'].setValue("raj@gmail.com");
  User.myForm.controls?.['address'].setValue("tenkasi");
  User.myForm.controls?.['phoneNumber'].setValue("6789976");
  User.myForm.controls?.['password'].setValue("raja");

  expect(User['user']).toBeTruthy();
})

describe('UpdateComponet', () => {
  let component: RegisterComponent;
  let fixture: ComponentFixture<RegisterComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        RegisterComponent
      ],
      imports: [
        BrowserModule,
        AppRoutingModule,
        ReactiveFormsModule,
        HttpClientModule,
      ]
    }).compileComponents();
  });
})

it('Regsiter Component Test', () =>{
  const fixture = TestBed.createComponent(RegisterComponent);
  const app = fixture.nativeElement as HTMLElement;

```

```

    expect(app).toBeTruthy();
  })
  it('Register Component GUI count', () =>{
    const fixture = TestBed.createComponent(RegisterComponent);
    const app = fixture.nativeElement as HTMLElement;
    const inputElement = app.querySelectorAll("input");
    const buttonElement = app.querySelectorAll("button");
    expect(inputElement.length).toEqual(6);
    expect(buttonElement.length).toEqual(1);
  })
  it('Test Form Valid', () =>{
    const fixture = TestBed.createComponent(RegisterComponent);
    const User = fixture.componentInstance;
    User.myForm.controls?.['id'].setValue("1");
    User.myForm.controls?.['username'].setValue("Raja");
    User.myForm.controls?.['email'].setValue("raj@gmail.com");
    User.myForm.controls?.['address'].setValue("tenkasi");
    User.myForm.controls?.['phoneNumber'].setValue("6789976");
    User.myForm.controls?.['password'].setValue("raja");

    expect(User['user']).toBeTruthy();
  })

  //login:

  describe('LoginComponet', () => {
    let component: LoginComponent;
    let fixture: ComponentFixture<LoginComponent>;

    beforeEach(async () => {
      await TestBed.configureTestingModule({
        declarations: [
          LoginComponent
        ],
        imports: [
          BrowserModule,
          AppRoutingModule,
          ReactiveFormsModule,
          HttpClientModule,

```

```

    ]
    }).compileComponents();
  });
})

it('Login Component Test', () =>{
  const fixture = TestBed.createComponent(LoginComponent);
  const app = fixture.nativeElement as HTMLElement;
  expect(app).toBeTruthy();
})

it('Login Component GUI count', () =>{
  const fixture = TestBed.createComponent(LoginComponent);
  const app = fixture.nativeElement as HTMLElement;
  const inputElement = app.querySelectorAll("input");
  const buttonElement = app.querySelectorAll("button");
  expect(inputElement.length).toEqual(1);
  expect(buttonElement.length).toEqual(1);
})

it('Test Form Valid', () =>{
  const fixture = TestBed.createComponent(LoginComponent);
  const User = fixture.componentInstance;
  User.myForm.controls['username'].setValue("Raja");
  User.myForm.controls['password'].setValue("raja");

  expect(User['user']).toBeTruthy();
})

///Parcel

describe('UserComponent', () => {
  let component: UserComponent;
  let fixture: ComponentFixture<UserComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        UserComponent
      ],
      imports: [
        BrowserModule,
        AppRoutingModule,

```

```

        ReactiveFormsModule,
        HttpClientModule,
    ]
  }) .compileComponents();
});
})

it(' User Component Test', () =>{
  const fixture = TestBed.createComponent(UserComponent);
  const app = fixture.nativeElement as HTMLElement;
  expect(app).toBeTruthy();
})

it('Register Component GUI count', () =>{
  const fixture = TestBed.createComponent(UserComponent);
  const app = fixture.nativeElement as HTMLElement;
  const inputElement = app.querySelectorAll("input");
  const buttonElement = app.querySelectorAll("button");
  expect(inputElement.length).toEqual(6);
  expect(buttonElement.length).toEqual(1);
})

it('Test Form Valid', () =>{
  const fixture = TestBed.createComponent(RegisterComponent);
  const User = fixture.componentInstance;
  User.myForm.controls?.['recipientName'].setValue("Jeban");
  User.myForm.controls?.['fromAddress'].setValue("Tenkasi");
  User.myForm.controls?.['recipientAddress'].setValue("Tieunelveli");
  User.myForm.controls?.['parcelName'].setValue("Books");
  User.myForm.controls?.['recipientPhoneNumber'].setValue("78776667");
  User.myForm.controls?.['Weight'].setValue("70");

  expect(User['user']).toBeTruthy();
})

//View All

describe('ViewComponent ', ()=>{
beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [
      ViewComponent
    ],

```

```

    imports: [
        BrowserModule,
        AppRoutingModule,
        ReactiveFormsModule,
        HttpClientModule,

    ]

    }).compileComponents();
});

it('App Component Test', () =>{
    const fixture = TestBed.createComponent(ViewComponent);
    const app = fixture.nativeElement as HTMLElement;
    expect(app).toBeTruthy();
})

it('App Component GUI count', () =>{
    const fixture = TestBed.createComponent(ViewComponent);
    const app = fixture.nativeElement as HTMLElement;
    const inputElement = app.querySelectorAll("input");
    const buttonElement = app.querySelectorAll("button");
    expect(inputElement.length).toEqual(6);
    expect(buttonElement.length).toEqual(1);
})

it('Testing Form valid - ViewAll', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const Laptop = fixture.componentInstance;
    expect(Laptop['myForm'].valid).toBeFalsy();
});

});

//Staff

describe('StaffComponent', () => {
    let component: StaffComponent;
    let fixture: ComponentFixture<StaffComponent>;

    beforeEach(async () => {
        await TestBed.configureTestingModule({

```



```

    declarations: [
        StaffComponent
    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        ReactiveFormsModule,
        HttpClientModule,
    ]
  }) .compileComponents();
});
})
it(' User Component Test', () =>{
    const fixture = TestBed.createComponent(StaffComponent);
    const app = fixture.nativeElement as HTMLElement;
    expect(app).toBeTruthy();
})
it('Register Component GUI count', () =>{
    const fixture = TestBed.createComponent(StaffComponent);
    const app = fixture.nativeElement as HTMLElement;
    const inputElement = app.querySelectorAll("input");
    const buttonElement = app.querySelectorAll("button");
    expect(inputElement.length).toEqual(2);
    expect(buttonElement.length).toEqual(1);
})
it('Test Form Valid', () =>{
    const fixture = TestBed.createComponent(StaffComponent);
    const Parcel = fixture.componentInstance;
    Parcel.myForm.controls?.['staffName'].setValue("Muthu");
    Parcel.myForm.controls?.['phoneNumber'].setValue("78776667");

    expect(Parcel['parcel']).toBeTruthy();
})
//View All

describe('ViewComponent ', () =>{
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [

```

```

        ViewComponent
    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule,
    ]
  }) .compileComponents();
});

it('App Component Test', () => {
  const fixture = TestBed.createComponent(StaffComponent);
  const app = fixture.nativeElement as HTMLElement;
  expect(app).toBeTruthy();
})

it('App Component GUI count', () => {
  const fixture = TestBed.createComponent(StaffComponent);
  const app = fixture.nativeElement as HTMLElement;
  const inputElement = app.querySelectorAll("input");
  const buttonElement = app.querySelectorAll("button");
  expect(inputElement.length).toEqual(1);
  expect(buttonElement.length).toEqual(1);
})

it('Testing Form valid - ViewAll', () => {
  const fixture = TestBed.createComponent(StaffComponent);
  const Staff = fixture.componentInstance;
  expect(Staff['myForm'].valid).toBeFalsy();
});
});

```