

**Vendor Management System**  
**Jeban Ignesh J-11963**

**White Box Testing**

**1. Write the Unit Test cases by using Jasmine & Karma for your Front-End [Angular].**

**App.component.spect.ts:**

**Code:**

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

import { RouterTestingModule } from '@angular/router/testing';

import { AppComponent } from './app.component';

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();

  });

});
```

### Vendor.component.spect.ts

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

import { VendorComponent } from './vendor.component';

import { AppComponent } from '../app.component';

import { RouterTestingModule } from '@angular/router/testing';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModuleModule } from '../app-routing.module';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';
```

```
describe('Vendor Component Test', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [VendorComponent, AppComponent],

      imports: [RouterTestingModule, BrowserModule, AppRoutingModuleModule,

        HttpClientModule, FormsModule, ReactiveFormsModule],

    }).compileComponents();

  });

  it('app component test', () => {

    const fixture = TestBed.createComponent(VendorComponent);

    const app = fixture.componentInstance;

    fixture.detectChanges();
```

```
    expect(app).toBeTruthy();
  });

  it('Vendor Insert Form GUI count', () => {

    const fixture = TestBed.createComponent(VendorComponent);

    const formElement = fixture.debugElement.nativeElement.querySelector("#Form");

    const inputElements = formElement.querySelectorAll("input");

    const buttonElement = formElement.querySelectorAll("button");

    expect(inputElements.length).toEqual(8);

    expect(buttonElement.length).toEqual(4);

  });
```

```
  it('Testing Form valid', () => {

    const fixture = TestBed.createComponent(VendorComponent);

    const Vendor = fixture.componentInstance;

    Vendor.myForm.controls?['id'].setValue("11967");

    Vendor.myForm.controls?['name'].setValue("Jeyavel");

    Vendor.myForm.controls?['country'].setValue("India");

    Vendor.myForm.controls?['company'].setValue("RtZ11902");

    Vendor.myForm.controls?['ssymbol'].setValue("#789");

    Vendor.myForm.controls?['realationship'].setValue("1999");

    Vendor.myForm.controls?['email'].setValue("j@gmail.com");

    Vendor.myForm.controls?['mobile'].setValue("9078563456");
```

```
    expect(Vendor.myForm.valid).toBeTruthy();  
  });
```

```
it('Testing Insert Form valid Falsy', () => {
```

```
    const fixture = TestBed.createComponent(VendorComponent);  
    const Vendor = fixture.componentInstance;  
    Vendor.myForm.controls?['id'].setValue(" ");  
    Vendor.myForm.controls?['name'].setValue(" ");  
    Vendor.myForm.controls?['country'].setValue(" ");  
    Vendor.myForm.controls?['company'].setValue(" ");  
    Vendor.myForm.controls?['ssymbol'].setValue(" ");  
    Vendor.myForm.controls?['realationship'].setValue(" ");  
    Vendor.myForm.controls?['email'].setValue(" ");  
    Vendor.myForm.controls?['mobile'].setValue(" ");  
    expect(Vendor.myForm.valid).toBeFalsy();  
  });
```

```
it('Vendor Update Form GUI count', () => {
```

```
    const fixture = TestBed.createComponent(VendorComponent);  
    const formElement = fixture.debugElement.nativeElement.querySelector("#Form");
```

```

const inputElements = formElement.querySelectorAll("input");
const buttonElement = formElement.querySelectorAll("button");
expect(inputElements.length).toEqual(8);
expect(buttonElement.length).toEqual(4);
});

it('Testing Form valid', () => {
  const fixture = TestBed.createComponent(VendorComponent);
  const Vendor = fixture.componentInstance;

  Vendor.myForm.controls?['id'].setValue("11967");
  Vendor.myForm.controls?['name'].setValue("Jeyavel");
  Vendor.myForm.controls?['country'].setValue("England");
  Vendor.myForm.controls?['company'].setValue("RtZ11902");
  Vendor.myForm.controls?['ssymbol'].setValue("#789");
  Vendor.myForm.controls?['realationship'].setValue("1999");
  Vendor.myForm.controls?['email'].setValue("j@gmail.com");
  Vendor.myForm.controls?['mobile'].setValue("9078563456");
  expect(Vendor.myForm.valid).toBeTruthy();
});

it('Testing Form Update valid Falsy', () => {
  const fixture = TestBed.createComponent(VendorComponent);
  const Vendor = fixture.componentInstance;

  Vendor.myForm.controls?['id'].setValue("");
  Vendor.myForm.controls?['name'].setValue("");

```

```
Vendor.myForm.controls?.['country'].setValue(" ");  
Vendor.myForm.controls?.['company'].setValue(" ");  
Vendor.myForm.controls?.['ssymbol'].setValue(" ");  
Vendor.myForm.controls?.['realationship'].setValue(" ");  
Vendor.myForm.controls?.['email'].setValue(" ");  
Vendor.myForm.controls?.['mobile'].setValue(" ");  
expect(Vendor.myForm.valid).toBeFalsy();  
});
```

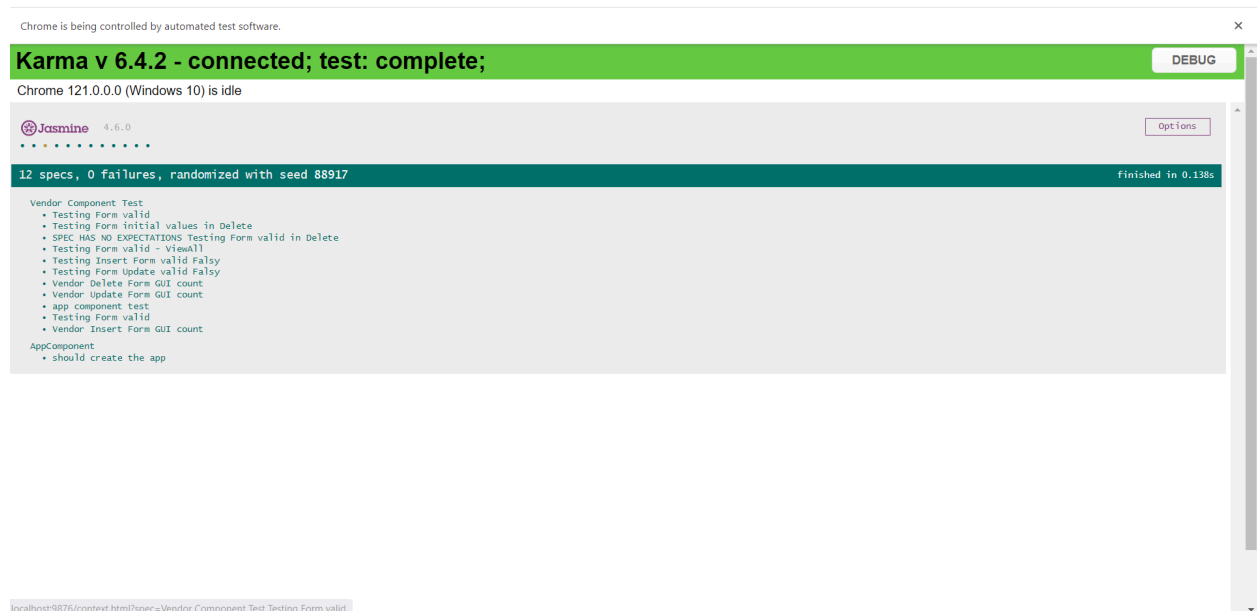
```
it('Vendor Delete Form GUI count', () => {  
  const fixture = TestBed.createComponent(VendorComponent);  
  const formElement = fixture.debugElement.nativeElement.querySelector("#Form");  
  const inputElements = formElement.querySelectorAll("input");  
  const buttonElement = formElement.querySelectorAll("button");  
  expect(inputElements.length).toEqual(8);  
  expect(buttonElement.length).toEqual(4);  
});
```

```
it('Testing Form initial values in Delete', () => {  
  const fixture = TestBed.createComponent(VendorComponent);  
  const app = fixture.componentInstance;  
  const loginFormGroup = app.myForm;  
  const loginFormValues = {
```

```
    id: ",  
    name: ",  
    country:",  
    company:",  
    ssymbol:",  
    realationship:",  
    email: ",  
    mobile:",  
  }  
  expect(loginFormGroup.value).toEqual(loginFormValues);  
});
```

```
it('Testing Form valid in Delete', () => {  
  const fixture = TestBed.createComponent(VendorComponent);  
  const app = fixture.componentInstance;  
  app.myForm.controls?['id'].setValue("11902");  
  expect(app.myForm.valid).toBeTruthy;  
});  
});
```

## Jasmine & Karma Output:



### Vendor.component.spect.ts:

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { VendorviewComponent } from './vendorview.component';
```

```
import { AppComponent } from '../app.component';
```

```
import { RouterTestingModule } from '@angular/router/testing';
```

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
```

```
import { BrowserModule } from '@angular/platform-browser';
```

```
import { AppRoutingModuleModule } from '../app-routing.module';
```

```
import { HttpClientModule } from '@angular/common/http';
```

```
import { VendorComponent } from '../vendor/vendor.component';
```



```
describe('VendorviewComponent', () => {  
  beforeEach(async () => {  
    await TestBed.configureTestingModule({  
      declarations: [VendorviewComponent, AppComponent, VendorComponent],  
      imports: [RouterTestingModule, BrowserModule, AppRoutingModule,  
        HttpClientModule, FormsModule, ReactiveFormsModule],  
    }).compileComponents();  
  });  
  
  it('Vendor component test', () => {  
    const fixture = TestBed.createComponent(VendorviewComponent);  
    const app = fixture.componentInstance;  
    fixture.detectChanges();  
    expect(app).toBeTruthy();  
  });  
  
  it('Vendor Form GUI count', () => {  
    const fixture = TestBed.createComponent(VendorviewComponent);  
    const formElement = fixture.debugElement.nativeElement.querySelector("#Form");  
    const buttonElement = formElement.querySelectorAll("button");  
    expect(buttonElement.length).toEqual(1);  
  });  
  
  it('Vendor Update Form GUI count', () => {  
    const fixture = TestBed.createComponent(VendorviewComponent);  
    const formElement = fixture.debugElement.nativeElement.querySelector("#Form");
```

```

    const buttonElement = formElement.querySelectorAll("button");

    expect(buttonElement.length).toEqual(1);

  });

  it('Testing Form valid', () => {

    const fixture = TestBed.createComponent(VendorviewComponent);

    const Vendor = fixture.componentInstance;

    Vendor.myForm.controls?['id'].setValue("11967");

    Vendor.myForm.controls?['name'].setValue("Jeyavel");

    Vendor.myForm.controls?['country'].setValue("England");

    Vendor.myForm.controls?['company'].setValue("RtZ11902");

    Vendor.myForm.controls?['ssymbol'].setValue("#789");

    Vendor.myForm.controls?['realationship'].setValue("1999");

    Vendor.myForm.controls?['email'].setValue("j@gmail.com");

    Vendor.myForm.controls?['mobile'].setValue("9078563456");

    expect(Vendor.myForm.valid).toBeTruthy();

  });

  it('Testing Form Update valid Falsy', () => {

    const fixture = TestBed.createComponent(VendorviewComponent);

    const Vendor = fixture.componentInstance;

    Vendor.myForm.controls?['id'].setValue("");

    Vendor.myForm.controls?['name'].setValue("");

    Vendor.myForm.controls?['country'].setValue("");

```

```
Vendor.myForm.controls?.['company'].setValue(" ");  
Vendor.myForm.controls?.['ssymbol'].setValue(" ");  
Vendor.myForm.controls?.['realationship'].setValue(" ");  
Vendor.myForm.controls?.['email'].setValue(" ");  
Vendor.myForm.controls?.['mobile'].setValue(" ");  
expect(Vendor.myForm.valid).toBeFalsy();  
});
```

```
it('Vendor Delete Form GUI count', () => {  
  const fixture = TestBed.createComponent(VendorviewComponent);  
  const formElement = fixture.debugElement.nativeElement.querySelector("#Form");  
  const buttonElement = formElement.querySelectorAll("button");  
  expect(buttonElement.length).toEqual(1);  
});
```

```
it('Testing Form initial values in Delete', () => {  
  const fixture = TestBed.createComponent(VendorviewComponent);  
  const app = fixture.componentInstance;  
  const loginFormGroup = app.myForm;  
  const loginFormValues = {  
    id: "",  
    name: "",  
    country: "",
```

```
    company: ",  
    ssymbol: ",  
    realationship: ",  
    email: ",  
    mobile: ",  
  }  
  expect(loginFormGroup.value).toEqual(loginFormValues);  
});
```

```
it('Testing Form valid in Delete', () => {  
  const fixture = TestBed.createComponent(VendorviewComponent);  
  const app = fixture.componentInstance;  
  app.myForm.controls?['id'].setValue("11901");  
  expect(app.myForm.valid).toBeTruthy;  
});  
});
```

Chrome is being controlled by automated test software.

**Karma v 6.4.2 - connected; test: complete;**

DEBUG

Chrome 121.0.0.0 (Windows 10) is idle

 **Jasmine** 4.6.0

Options

21 specs, 0 failures, randomized with seed 51819

finished in 0.173s

```
Vendor Component Test
  • Testing Form valid - ViewAll
  • app component test
  • SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  • Testing Form valid
  • Testing Form Update valid Falsy
  • Vendor Update Form GUI count
  • Testing Insert Form valid Falsy
  • Testing Form initial values in Delete
  • Vendor Delete Form GUI count
  • Vendor Insert Form GUI count
  • Testing Form valid

AppComponent
  • should create the app

VendorViewComponent
  • Vendor Delete Form GUI count
  • Testing Form valid
  • Testing Form Update valid Falsy
  • SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  • Vendor Update Form GUI count
  • Testing Form initial values in Delete
  • Testing Form valid - ViewAll
  • Vendor component test
  • Vendor Form GUI count
```

localhost:9876/context.html?spec=Vendor Component Test Vendor Update Form GUI count

## Contract Component

### Contract.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { ContractComponent } from './contract.component';
```

```
import { AppComponent } from '../app.component';
```

```
import { RouterTestingModule } from '@angular/router/testing';
```

```
import { BrowserModule } from '@angular/platform-browser';
```

```
import { AppRoutingModuleModule } from '../app-routing.module';
```

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
```

```
import { HttpClientModule } from '@angular/common/http';
```

```
describe('Contract Component Test', () => {
```

```
beforeEach(async () => {

  await TestBed.configureTestingModule({

    declarations: [ContractComponent, AppComponent],

    imports: [RouterTestingModule, BrowserModule, AppRoutingModule,

      HttpClientModule, FormsModule, ReactiveFormsModule],

  }).compileComponents();

});

it('Vendor component test', () => {

  const fixture = TestBed.createComponent(ContractComponent);

  const app = fixture.componentInstance;

  fixture.detectChanges();

  expect(app).toBeTruthy();

});

it('Vendor Insert Form GUI count', () => {

  const fixture = TestBed.createComponent(ContractComponent);

  const formElement = fixture.debugElement.nativeElement.querySelector("#Form");

  const inputElements = formElement.querySelectorAll("input");

  const buttonElement = formElement.querySelectorAll("button");

  expect(inputElements.length).toEqual(6);

  expect(buttonElement.length).toEqual(4);

});

it('Testing Form valid', () => {
```

```
const fixture = TestBed.createComponent(ContractComponent);

const Contract = fixture.componentInstance;

Contract.myForm.controls?['id'].setValue("11908");

Contract.myForm.controls?['name'].setValue("Jeyavel");

Contract.myForm.controls?['cno'].setValue("12345");

Contract.myForm.controls?['catogory'].setValue("WholeSale");

Contract.myForm.controls?['products'].setValue("Shoes");

Contract.myForm.controls?['status'].setValue("onboarding");

Contract.myForm.controls?['description'].setValue("Nothing");

expect(Contract.myForm.valid).toBeTruthy();

});
```

```
it('Testing Insert Form valid Falsy', () => {
```

```
const fixture = TestBed.createComponent(ContractComponent);

const Contract = fixture.componentInstance;

Contract.myForm.controls?['id'].setValue(" ");

Contract.myForm.controls?['name'].setValue(" ");

Contract.myForm.controls?['cno'].setValue(" ");

Contract.myForm.controls?['catogory'].setValue(" ");

Contract.myForm.controls?['products'].setValue(" ");

Contract.myForm.controls?['status'].setValue(" ");
```

```
Contract.myForm.controls?.['description'].setValue(" ");

expect(Contract.myForm.valid).toBeFalsy();

});

});
```

Chrome is being controlled by automated test software.

# Karma v 6.4.2 - connected; test: complete;

Chrome 121.0.0.0 (Windows 10) is idle

 **Jasmine** 4.6.0

.....

25 specs, 0 failures, randomized with seed 58094

Contract Component Test

- Contract component test
- Testing Insert Form valid Falsy
- Testing Insert Form valid
- Contract Insert Form GUI count

Chrome is being controlled by automated test software.

## Karma v 6.4.2 - connected; test: complete;

DEBUG

Chrome 121.0.0.0 (Windows 10) is idle

 **Jasmine** 4.6.0

.....

25 specs, 0 failures, randomized with seed 58094

Finished in 0.24s

Contract Component Test

- Contract component test
- Testing Insert Form valid Falsy
- Testing Insert Form valid
- Contract Insert Form GUI count

AppComponent

- should create the app

VendorViewComponent

- Vendor component test
- Vendor Form GUI count
- Vendor Update Form GUI count
- Vendor Delete Form GUI count
- Testing Form initial values in Delete
- SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
- Testing Form valid - ViewAll
- Testing Form Update valid Falsy
- Testing Form valid

Vendor Component Test

- Testing Form valid
- Vendor Insert Form GUI count
- app component test
- Vendor Update Form GUI count
- Testing Form update valid Falsy
- Testing Insert Form valid Falsy
- Vendor Delete Form GUI count
- Testing Form initial values in Delete
- SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
- Testing Form valid
- Testing Form valid - ViewAll



## **ContractView Component:**

**contract view.component.spec.ts**

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ContractviewComponent } from './contractview.component';
import { AppComponent } from '../app.component';
import { RouterTestingModule } from '@angular/router/testing';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModuleModule } from '../app-routing.module';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';

describe('ContractviewComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ContractviewComponent, AppComponent],
      imports: [RouterTestingModule, BrowserModule, AppRoutingModuleModule,
        HttpClientModule, FormsModule, ReactiveFormsModule],
    }).compileComponents();
  });

  it('ContractView component test', () => {
    const fixture = TestBed.createComponent(ContractviewComponent);
```

```
const app = fixture.componentInstance;

fixture.detectChanges();

expect(app).toBeTruthy();

});
```

```
it('ContractView Form GUI count', () => {

  const fixture = TestBed.createComponent(ContractviewComponent);

  const formElement = fixture.debugElement.nativeElement.querySelector("#Form");

  const buttonElement = formElement.querySelectorAll("button");

  expect(buttonElement.length).toEqual(1);

});
```

```
it('Testing Update Form valid', () => {

  const fixture = TestBed.createComponent(ContractviewComponent);

  const Contract = fixture.componentInstance;

  Contract.myForm.controls?.['id'].setValue("11908");

  Contract.myForm.controls?.['name'].setValue("Jeyavel");

  Contract.myForm.controls?.['cno'].setValue("12345");

  Contract.myForm.controls?.['catogory'].setValue("WholeSale");

  Contract.myForm.controls?.['products'].setValue("Shoes");
```

```
Contract.myForm.controls?.['status'].setValue("onboarding");  
Contract.myForm.controls?.['description'].setValue("Nothing");  
expect(Contract.myForm.valid).toBeTruthy();  
});
```

```
it('Testing Update Form valid Falsy', () => {
```

```
    const fixture = TestBed.createComponent(ContractviewComponent);  
    const Contract = fixture.componentInstance;  
    Contract.myForm.controls?.['id'].setValue(" ");  
    Contract.myForm.controls?.['name'].setValue(" ");  
    Contract.myForm.controls?.['cno'].setValue(" ");  
    Contract.myForm.controls?.['catogory'].setValue(" ");  
    Contract.myForm.controls?.['products'].setValue(" ");  
    Contract.myForm.controls?.['status'].setValue(" ");  
    Contract.myForm.controls?.['description'].setValue(" ");  
    expect(Contract.myForm.valid).toBeFalsy();  
});
```

```
it('Testing Form initial values in Delete', () => {
```

```
    const fixture = TestBed.createComponent(ContractviewComponent);  
    const app = fixture.componentInstance;  
    const loginFormGroup = app.myForm;
```

```
const loginFormValues = {  
  id: "",  
  name: "",  
  cno: "",  
  category: "",  
  products: "",  
  status: "",  
  description: "",  
}  
expect(loginFormGroup.value).toEqual(loginFormValues);  
});
```

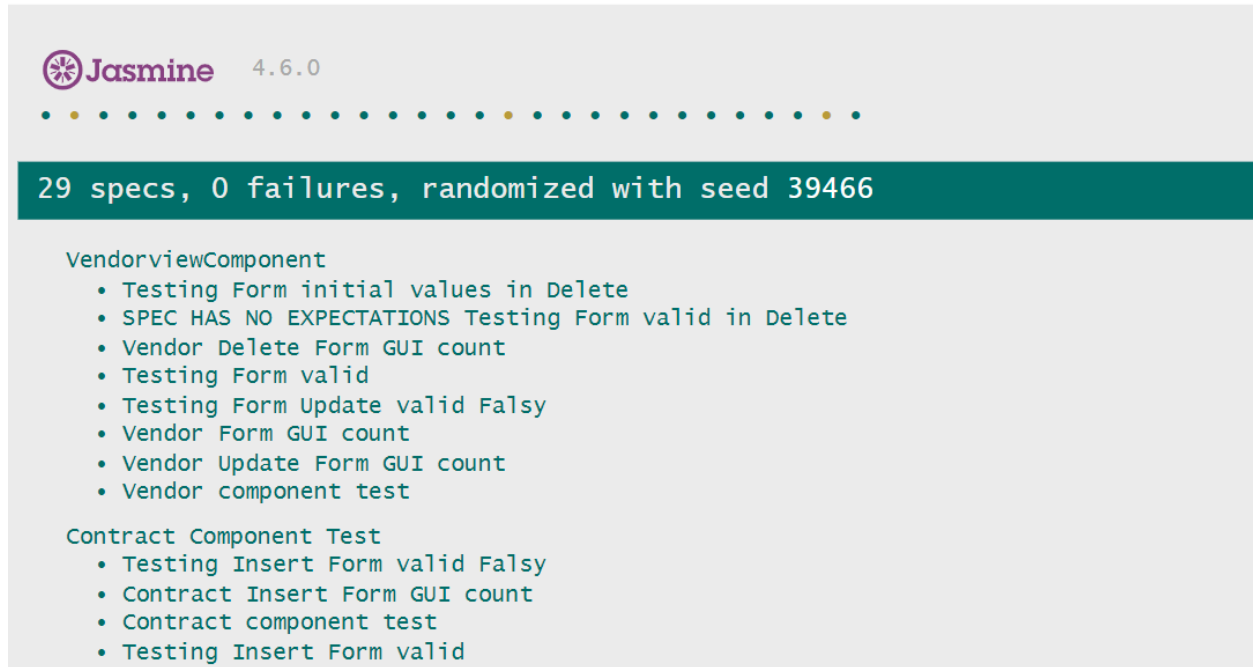
```
it('Testing Form valid in Delete', () => {  
  const fixture = TestBed.createComponent(ContractviewComponent);  
  const app = fixture.componentInstance;  
  app.myForm.controls?['id'].setValue("11901");  
  expect(app.myForm.valid).toBeTruthy;  
});
```

```
});
```

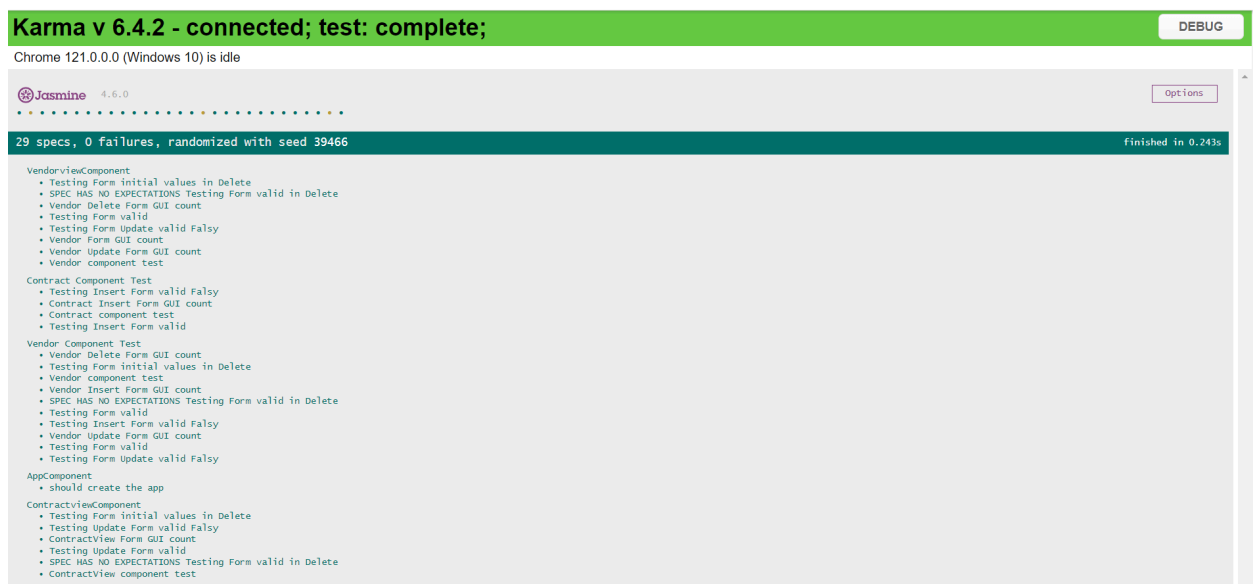
Chrome is being controlled by automated test software.

**Karma v 6.4.2 - connected; test: complete;**

Chrome 121.0.0.0 (Windows 10) is idle



Chrome is being controlled by automated test software.



**Product Component:**

**Product.component.spec.ts**

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ProductComponent } from './product.component';
import { AppComponent } from '../app.component';
import { RouterTestingModule } from '@angular/router/testing';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from '../app-routing.module';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

describe('ProductComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ProductComponent, AppComponent],
      imports: [RouterTestingModule, BrowserModule, AppRoutingModuleModule,
        HttpClientModule, FormsModule, ReactiveFormsModule],
    }).compileComponents();
  });

  it('Product component test', () => {
    const fixture = TestBed.createComponent(ProductComponent);
    const app = fixture.componentInstance;
```

```
    fixture.detectChanges();

    expect(app).toBeTruthy();

  });

  it('Product Insert Form GUI count', () => {

    const fixture = TestBed.createComponent(ProductComponent);

    const formElement = fixture.debugElement.nativeElement.querySelector("#Form");

    const inputElements = formElement.querySelectorAll("input");

    const buttonElement = formElement.querySelectorAll("button");

    expect(inputElements.length).toEqual(5);

    expect(buttonElement.length).toEqual(4);

  });
```

```
  it('Testing Insert Form valid', () => {

    const fixture = TestBed.createComponent(ProductComponent);

    const Product = fixture.componentInstance;

    Product.myForm.controls?.['id'].setValue("102");

    Product.myForm.controls?.['cno'].setValue("3456");

    Product.myForm.controls?.['pname'].setValue("Shoes");

    Product.myForm.controls?.['price'].setValue("10000");

    Product.myForm.controls?.['brand'].setValue("Puma");

    expect(Product.myForm.valid).toBeTruthy();

  });
```

```

it('Testing Insert Form valid Falsy', () => {

  const fixture = TestBed.createComponent(ProductComponent);

  const Product = fixture.componentInstance;

  Product.myForm.controls?.['id'].setValue(" ");

  Product.myForm.controls?.['cno'].setValue(" ");

  Product.myForm.controls?.['pname'].setValue(" ");

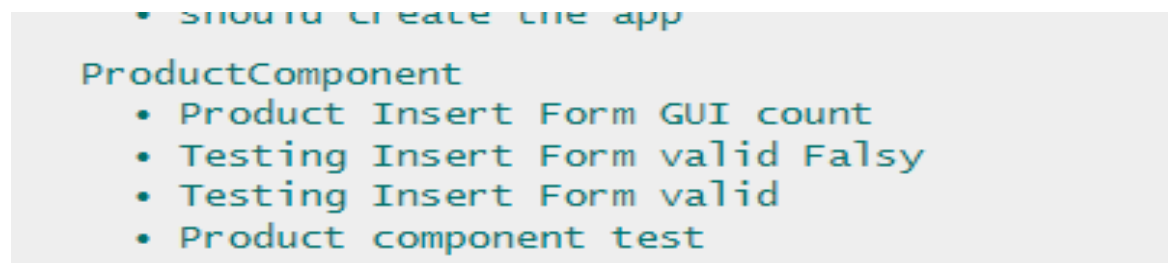
  Product.myForm.controls?.['price'].setValue(" ");

  Product.myForm.controls?.['brand'].setValue(" ");

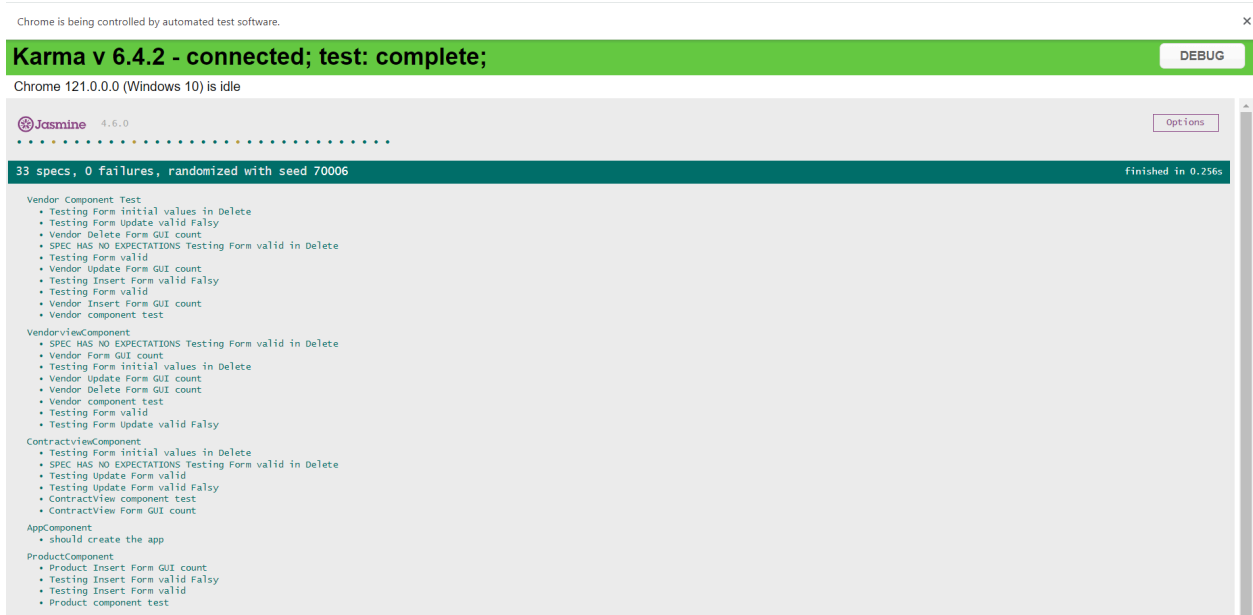
  expect(Product.myForm.valid).toBeFalsy();

});
});

```







### Productview.component.spec.ts:

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { ProductviewComponent } from './productview.component';
```

```
import { AppComponent } from '../app.component';
```

```
import { RouterTestingModule } from '@angular/router/testing';
```

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
```

```
import { AppRoutingModuleModule } from '../app-routing.module';
```

```
import { BrowserModule } from '@angular/platform-browser';
```

```
import { HttpClientModule } from '@angular/common/http';
```

```
describe('ProductviewComponent', () => {
```

```
  beforeEach(async () => {
```

```

await TestBed.configureTestingModule({
  declarations: [ProductviewComponent, AppComponent],
  imports: [RouterTestingModule, BrowserModule, AppRoutingModule,
    HttpClientModule, FormsModule, ReactiveFormsModule],
}).compileComponents();
});

it('Productview component test', () => {
  const fixture = TestBed.createComponent(ProductviewComponent);
  const app = fixture.componentInstance;
  fixture.detectChanges();
  expect(app).toBeTruthy();
});

it('Productview Form GUI count', () => {
  const fixture = TestBed.createComponent(ProductviewComponent);
  const formElement = fixture.debugElement.nativeElement.querySelector("#Form");
  const buttonElement = formElement.querySelectorAll("button");
  expect(buttonElement.length).toEqual(1);
});

it('Testing Update Form valid', () => {
  const fixture = TestBed.createComponent(ProductviewComponent);
  const Product = fixture.componentInstance;

```

```
Product.myForm.controls?.['id'].setValue("103");  
Product.myForm.controls?.['cno'].setValue("7867");  
Product.myForm.controls?.['pname'].setValue("Tshirts");  
Product.myForm.controls?.['price'].setValue("10000");  
Product.myForm.controls?.['brand'].setValue("Puma");  
expect(Product.myForm.valid).toBeTruthy();  
});
```

```
it('Testing Update Form valid Falsy', () => {
```

```
    const fixture = TestBed.createComponent(ProductviewComponent);  
    const Product = fixture.componentInstance;  
    Product.myForm.controls?.['id'].setValue(" ");  
    Product.myForm.controls?.['cno'].setValue(" ");  
    Product.myForm.controls?.['pname'].setValue(" ");  
    Product.myForm.controls?.['price'].setValue(" ");  
    Product.myForm.controls?.['brand'].setValue(" ");  
    expect(Product.myForm.valid).toBeFalsy();  
});
```

```
it('Testing Form initial values in Delete', () => {
```

```
    const fixture = TestBed.createComponent(ProductviewComponent);  
    const app = fixture.componentInstance;  
    const loginFormGroup = app.myForm;
```

```

const loginFormValues = {
  id: "",
  cno: "",
  pname: "",
  price: "",
  brand: "",
}

expect(loginFormGroup.value).toEqual(loginFormValues);
});

it('Testing Form valid in Delete', () => {
  const fixture = TestBed.createComponent(ProductviewComponent);
  const app = fixture.componentInstance;
  app.myForm.controls?['id'].setValue("11901");
  expect(app.myForm.valid).toBeTruthy();
});
});

```

#### ProductviewComponent

- Testing Update Form valid Falsy
- Testing Update Form valid
- SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
- Testing Form initial values in Delete
- Productview Form GUI count
- Productview component test

DEBUG

Options

finished in 0.276s

- ContractViewComponent
  - ContractView Form GUI count
  - ContractView component test
  - Testing Update Form valid Falsy
  - Testing Update Form valid
  - SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  - Testing Form initial values in Delete
- Contract Component Test
  - Contract component test
  - Testing Insert Form valid
  - Contract Insert Form GUI count
  - Testing Insert Form valid Falsy
- Vendor Component Test
  - Testing Form initial values in Delete
  - Vendor Update Form GUI count
  - Testing Form valid
  - Testing Form valid
  - Testing Insert Form valid Falsy
  - SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  - Vendor component test
  - Vendor Delete Form GUI count
  - Vendor Insert Form GUI count
  - Testing Form Update valid Falsy
- AppComponent
  - should create the app
- VendorViewComponent
  - SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  - Vendor Form GUI count
  - Vendor Update Form GUI count
  - Vendor Delete Form GUI count
  - Vendor component test
  - Testing Form initial values in Delete
  - Testing Form Update valid Falsy
  - Testing Form valid
- ProductViewComponent
  - Testing Update Form valid Falsy
  - Testing Update Form valid
  - SPEC HAS NO EXPECTATIONS Testing Form valid in Delete
  - Testing Form initial values in Delete
  - ProductView Form GUI count
  - Productview component test
- ProductComponent
  - Product component test
  - Testing Insert Form valid
  - Product Insert Form GUI count
  - Testing Insert Form valid Falsy

```
> ▾ TERMINAL
✓ Browser application bundle generation complete.
WARN: 'Spec 'ContractviewComponent Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 13 of 37 SUCCESS (0 secs / 0.098 secs)
WARN: 'Spec 'Vendor Component Test Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 23 of 37 SUCCESS (0 secs / 0.171 secs)
WARN: 'Spec 'VendorviewComponent Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 29 of 37 SUCCESS (0 secs / 0.207 secs)
Chrome 121.0.0.0 (Windows 10): Executed 37 of 37 SUCCESS (0.27 secs / 0.231 secs)
TOTAL: 37 SUCCESS
✓ Browser application bundle generation complete.
WARN: 'Spec 'ContractviewComponent Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 4 of 39 SUCCESS (0 secs / 0.042 secs)
WARN: 'Spec 'Vendor Component Test Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 15 of 39 SUCCESS (0 secs / 0.13 secs)
WARN: 'Spec 'VendorviewComponent Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 21 of 39 SUCCESS (0 secs / 0.161 secs)
WARN: 'Spec 'ProductviewComponent Testing Form valid in Delete' has no expectations.'
Chrome 121.0.0.0 (Windows 10): Executed 31 of 39 SUCCESS (0 secs / 0.2 secs)
Chrome 121.0.0.0 (Windows 10): Executed 39 of 39 SUCCESS (0.276 secs / 0.241 secs)
TOTAL: 39 SUCCESS
□
```