

## Bug-Tracking

Jasmine karma

app.component.spec.ts:

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';
import { MatToolbarModule } from '@angular/material/toolbar';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from './app-routing.module';

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

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

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

```
});
```

## register.component.spec.ts:

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

import { RegisterComponent } from './register.component';
import { RouterTestingModule } from '@angular/router/testing';
import { ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from '../app-routing.module';
import { MatCardModule } from '@angular/material/card';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      imports: [
        RouterTestingModule,
        MatToolbarModule,
        HttpClientModule,
        ReactiveFormsModule,
        RouterTestingModule,
        BrowserModule,
        AppRoutingModuleModule,
        MatCardModule,

        ReactiveFormsModule,],
      declarations: [RegisterComponent]
    });
    fixture = TestBed.createComponent(RegisterComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create Register component', () => {
    expect(component).toBeTruthy();
  });
});
```

```

});

it(`should have as title 'Register-component'`, () => {
  const fixture = TestBed.createComponent(RegisterComponent);

  expect(component.title).toEqual('Register-Component');
});

it('Register Form GUI "Input" count', () => {
  const fixture = TestBed.createComponent(RegisterComponent);
  fixture.detectChanges();
  const formElement =
fixture.debugElement.nativeElement.querySelector("#myform");
  const inputElements = formElement.querySelectorAll("input");
  expect(inputElements.length).toEqual(3);
});

it('Testing register Form valid', () => {
  const fixture = TestBed.createComponent(RegisterComponent);
  const Register = fixture.componentInstance;
  Register.myform.controls?.['userName'].setValue("gowtham");
  Register.myform.controls?.['password'].setValue("password");

  expect(Register.myform.valid).toBeTruthy();
});

it('Testing Register Form invalid name and password ', () => {
  const fixture = TestBed.createComponent(RegisterComponent);
  const Register = fixture.componentInstance;
  Register.myform.controls?.['userName'].setValue("");
  Register.myform.controls?.['password'].setValue("");

  expect(Register.myform.valid).toBeFalsy();
});

it('Testing Register Form invalid name valid password', () => {
  const fixture = TestBed.createComponent(RegisterComponent);
  const Register = fixture.componentInstance;
  Register.myform.controls?.['userName'].setValue("");
  Register.myform.controls?.['password'].setValue("54784");

```

```

    expect(Register.myform.valid).toBeFalsy();
  });

  it('Testing Register Form valid name invalid password ', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const Register = fixture.componentInstance;
    Register.myform.controls?.['userName'].setValue("gowtham");
    Register.myform.controls?.['password'].setValue("*.djhds");

    expect(Register.myform.valid).toBeFalsy();
  });

  it('Testing Form initial values', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const app = fixture.componentInstance;
    const registerFormGroup = app.myform;
    const registerFormValues = {
      userName: '',
      password: '',
    }
    expect(registerFormGroup.value).toEqual( registerFormValues);
  });
});

```

### login.component.spec.ts:

```

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

import { LoginComponent } from './login.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule } from '@angular/forms';
import { MatCardModule } from '@angular/material/card';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { RouterTestingModule } from '@angular/router/testing';
import { AppRoutingModuleModule } from '../app-routing.module';

```

```

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

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

      imports: [
        RouterTestingModule,
        MatToolbarModule,
        HttpClientModule,
        ReactiveFormsModule,
        RouterTestingModule,
        BrowserModule,
        AppRoutingModule,
        MatCardModule,

        ReactiveFormsModule,],
      declarations: [LoginComponent]
    });
    fixture = TestBed.createComponent(LoginComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create Login component', () => {
    expect(component).toBeTruthy();
  });

  it(`should have as title 'Login-component'`, () => {
    const fixture = TestBed.createComponent(LoginComponent);

    expect(component.title).toEqual('Login-component');
  });

  it('Login Form GUI "Input" count', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    fixture.detectChanges();
    const formElement =
    fixture.debugElement.nativeElement.querySelector("#myform");
  });

```

```

    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(3);
  });

  it('Testing LogIn Form valid', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const Login = fixture.componentInstance;
    Login.myform.controls?['userName'].setValue("siva");
    Login.myform.controls?['password'].setValue("siva@1234");
    expect(Login.myform.valid).toBeTruthy();
  });

  it('Testing LogIn Form Invalid', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const Login = fixture.componentInstance;
    Login.myform.controls?['userName'].setValue("");
    Login.myform.controls?['password'].setValue("");
    expect(Login.myform.valid).toBeFalsy();
  });

  it('Testing LogIn Form Invalid name valid password', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const Login = fixture.componentInstance;
    Login.myform.controls?['userName'].setValue("");
    Login.myform.controls?['password'].setValue("74847g");
    expect(Login.myform.valid).toBeFalsy();
  });

  it('Testing LogIn Form valid name invalid password', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const Login = fixture.componentInstance;
    Login.myform.controls?['userName'].setValue("valan");
    Login.myform.controls?['password'].setValue("*.djhdg");
    expect(Login.myform.valid).toBeFalsy();
  });

  it('Testing Login Form initial values', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const app = fixture.componentInstance;
    const LoginFormGroup = app.myform;
  });

```

```

    const loginFormValues = {
      userName: '',
      password: '',
    }
    expect(LoginFormGroup.value).toEqual( loginFormValues);
  });
});

```

### admin.component.spec.ts:

```

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

import { AdminHomepageComponent } from './admin-homepage.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule } from '@angular/forms';
import { MatCardModule } from '@angular/material/card';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { RouterTestingModule } from '@angular/router/testing';
import { AppRoutingModuleModule } from '../app-routing.module';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      imports: [
        RouterTestingModule,
        MatToolbarModule,
        HttpClientModule,
        ReactiveFormsModule,
        RouterTestingModule,
        BrowserModule,
        AppRoutingModuleModule,
        MatCardModule,

```

```

    ReactiveFormsModule,],
    declarations: [AdminHomepageComponent]
  });

  fixture = TestBed.createComponent(AdminHomepageComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create Admin home component', () => {
  expect(component).toBeTruthy();
});

it(`should have as title 'Admin-Home'`, () => {
  const fixture = TestBed.createComponent(AdminHomepageComponent);

  expect(component.title).toEqual('Admin-Home');
});

it('Testing Resolve bug Form valid ', () => {
  const fixture = TestBed.createComponent(AdminHomepageComponent);
  const Resolvebug = fixture.componentInstance;

  Resolvebug.myform.controls?.['bugId'].setValue(3);
  Resolvebug.myform.controls?.['bugName'].setValue("system error");
  Resolvebug.myform.controls?.['userName'].setValue("aravind");
  Resolvebug.myform.controls?.['description'].setValue("error is occurred
in system");
  Resolvebug.myform.controls?.['status'].setValue("close");
  Resolvebug.myform.controls?.['solution'].setValue("give valid
decription");

  expect(Resolvebug.myform.valid).toBeTruthy();
});

it('Testing Resolve bug Form invalid id ', () => {
  const fixture = TestBed.createComponent(AdminHomepageComponent);
  const Resolvebug = fixture.componentInstance;

```



```

    Resolvebug.myform.controls?.['bugId'].setValue("");
    Resolvebug.myform.controls?.['bugName'].setValue("system error");
    Resolvebug.myform.controls?.['userName'].setValue("aravind");
    Resolvebug.myform.controls?.['description'].setValue("error is occurred
in system");
    Resolvebug.myform.controls?.['status'].setValue("close");
    Resolvebug.myform.controls?.['solution'].setValue("give valid
decription");
    expect(Resolvebug.myform.valid).toBeFalsy();

  });

it('Testing Resolve bug Form   invalid user name ', () => {
  const fixture = TestBed.createComponent(AdminHomepageComponent);
  const Resolvebug = fixture.componentInstance;

  Resolvebug.myform.controls?.['bugId'].setValue(3);
  Resolvebug.myform.controls?.['bugName'].setValue("system error");
  Resolvebug.myform.controls?.['userName'].setValue("");
  Resolvebug.myform.controls?.['description'].setValue("error is occurred
in system");
  Resolvebug.myform.controls?.['status'].setValue("close");
  Resolvebug.myform.controls?.['solution'].setValue("give valid
decription");
  expect(Resolvebug.myform.valid).toBeFalsy();

});

it('Testing Resolve bug Form   invalid data ', () => {
  const fixture = TestBed.createComponent(AdminHomepageComponent);
  const Resolvebug = fixture.componentInstance;

  Resolvebug.myform.controls?.['bugId'].setValue("");
  Resolvebug.myform.controls?.['bugName'].setValue("");
  Resolvebug.myform.controls?.['userName'].setValue("");
  Resolvebug.myform.controls?.['description'].setValue("");
  Resolvebug.myform.controls?.['status'].setValue("");
  Resolvebug.myform.controls?.['solution'].setValue("");
  expect(Resolvebug.myform.valid).toBeFalsy();

```

```

    });

    it('Testing Resolve bug Form initial values', () => {
        const fixture = TestBed.createComponent(AdminHomepageComponent);
        const app = fixture.componentInstance;
        const adminFormGroup = app.myform;
        const adminFormValues = {
            bugId: '',
            bugName: '',
            userName: '',
            description: '',
            status: '',
            solution: '',
        }

        expect(adminFormGroup.value).toEqual(adminFormValues);
    });
});

```

### user-home.component.ts:

```

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

import { UserHomepageComponent } from './user-homepage.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule } from '@angular/forms';
import { MatCardModule } from '@angular/material/card';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { RouterTestingModule } from '@angular/router/testing';
import { AppRoutingModuleModule } from '../app-routing.module';

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

```

```

beforeEach(() => {
  TestBed.configureTestingModule({
    imports: [
      RouterTestingModule,
      MatToolbarModule,
      HttpClientModule,
      ReactiveFormsModule,
      RouterTestingModule,
      BrowserModule,
      AppRoutingModule,
      MatCardModule,

      ReactiveFormsModule,],
    declarations: [UserHomepageComponent]
  });
  fixture = TestBed.createComponent(UserHomepageComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create user-home component', () => {
  expect(component).toBeTruthy();
});

it(`should have as title 'user home-component'`, () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);

  expect(component.title).toEqual('user-home');
});

it('RegisterBug bug Form GUI "Input" count', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  fixture.detectChanges();
  const formElement =
fixture.debugElement.nativeElement.querySelector("#myform");
  const inputElements = formElement.querySelectorAll("input");
  expect(inputElements.length).toEqual(4);
});

```

```

it('RegisterBug bug Form GUI "Button" count', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  fixture.detectChanges();
  const formElement =
fixture.debugElement.nativeElement.querySelector("#myForm");
  const inputElements = formElement.querySelectorAll("button");
  expect(inputElements.length).toEqual(2);
});

it('Testing registerBug Form valid', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  const registerBug = fixture.componentInstance;
  registerBug.myform.controls?['bugName'].setValue("mobile bug");
  registerBug.myform.controls?['userName'].setValue("aravind");
  registerBug.myform.controls?['description'].setValue("while using
camera");

  expect(registerBug.myform.valid).toBeTruthy();
});

it('Testing registerBug Form Invalid bug name', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  const registerBug = fixture.componentInstance;
  registerBug.myform.controls?['bugName'].setValue("");
  registerBug.myform.controls?['userName'].setValue("aravind");
  registerBug.myform.controls?['description'].setValue("while using
camera");

  expect(registerBug.myform.valid).toBeFalsy();
});

it('Testing registerBug Form Invalid userName', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  const registerBug = fixture.componentInstance;
  registerBug.myform.controls?['bugName'].setValue("mobile error");
  registerBug.myform.controls?['userName'].setValue("");
  registerBug.myform.controls?['description'].setValue("while using
camera");

  expect(registerBug.myform.valid).toBeFalsy();
});

```

```
});

it('Testing registerBug Form Invalid description', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  const registerBug = fixture.componentInstance;
  registerBug.myform.controls?.['bugName'].setValue("mobile error");
  registerBug.myform.controls?.['userName'].setValue("aravind");
  registerBug.myform.controls?.['description'].setValue("");

  expect(registerBug.myform.valid).toBeFalsy();
});

it('Testing registerBug Form Invalid data', () => {
  const fixture = TestBed.createComponent(UserHomepageComponent);
  const registerBug = fixture.componentInstance;
  registerBug.myform.controls?.['bugName'].setValue("");
  registerBug.myform.controls?.['userName'].setValue("");
  registerBug.myform.controls?.['description'].setValue("");

  expect(registerBug.myform.valid).toBeFalsy();
});
});
```



