

**Employee Name :** PravinKumar RajaSahayam  
**Employee Id :** 11946

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

**Screenshots :**

**Test Case Successfully Passed :**

```
Jasmine 4.6.0 Options
.....

64 specs, 0 failures, randomized with seed 79547 finished in 0.908s

Add Course form Component Test
  • update existing Course to the database
  • Insert new Course to the database
  • Add Course form Component GUI count for button
  • should have as title 'Add Course Component'
  • Add Course form by Invalid data
  • Add Course form by Valid data
  • To test the Add Course component created
  • Add Course form Component GUI count for input
  • Add Course form initial value

AppComponent
  • should have as title 'captcha'
  • should create the app

View Course List Component Test
  • To test the View Course List component created

Add Chapter Component Test
  • To test the Add Chapter component created
  • Add Chapter form by Valid data
  • Add Chapter form Component GUI count for input
  • Add Chapter form initial value
  • Add Chapter form Component GUI count for button
  • Insert new Chapter to the database
  • should have as title 'Add Chapter Component'
  • update existing Chapter to the database
  • Add Chapter form by Invalid data
```

```
Add Teacher Component Test
  • Add Teacher form Component GUI count for input
  • To test the Add Teacher component created
  • Add Teacher form Component GUI count for button
  • Insert new Teacher to the database
  • Add Teacher form initial value
  • should have as title 'Add Teacher Component'
  • Add Teacher form by Invalid data
  • Add Teacher form by Valid data

Registration Teacher form Component Test
  • Registration Teacher form valid by invalid data
  • Registration Teacher form Component GUI count for button
  • Registration Teacher form Component GUI count for input
  • Registration Teacher form initial value
  • Registration teacher form valid by Valid data
  • should have as title 'Registration Teacher Component'
  • To test the Teacher Registration component created

Login Teacher form Component Test
  • should have as title 'Teacher Login Component'
  • To test the Login Teacher component created
  • Teacher Login form valid by Valid data
  • Teacher Login form Component GUI count for input
  • Teacher Login form initial value
  • Teacher Login form Component GUI count for button
  • Teacher Login form valid by invalid data

Login Admin form Component Test
  • Admin Login form Component GUI count for button
  • should have as title 'Admin Login Component'
  • Admin Login form valid by invalid data
  • Admin Login form valid by Valid data
  • To test the Login Admin component created
  • Admin Login form initial value
  • Admin Login form Component GUI count for input
```

```
Login User form Component Test
  • User Login form valid by Valid data
  • User Login form valid by invalid data
  • should have as title 'User Login Component'
  • User Login form Component GUI count for button
  • User Login form initial value
  • User Login form Component GUI count for input
  • To test the Login user component created

Registration User form Component Test
  • To test the user Registration component created
  • Registration User form initial value
  • should have as title 'Registration User Component'
  • Registration user form valid by invalid data
  • Registration user form Component GUI count for button
  • Registration User form Component GUI count for input
  • Registration User form valid by Valid data
```

## Spec.TS Files :

- Add Chapter Component :

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AddchapterComponent } from '../addchapter.component';
import { HttpClientModule } from '@angular/common/http';
import { HttpClientTestingModule, HttpTestingController } from
 '@angular/common/http/testing';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AppComponent } from 'src/app/app.component';
import { RegistrationComponent } from
 '../registration/registration.component';
import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
 'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';

describe('Add Chapter Component Test', () => {
  let component: AddchapterComponent;
  let fixture: ComponentFixture<AddchapterComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [AddchapterComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModuleModule,
        FormsModule,
        ReactiveFormsModule,
        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
```

```

    ],
    providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
    ]
  }).compileComponents();
});
it('To test the Add Chapter component created', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  expect(app).toBeTruthy();
});

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

it('Add Cahpter form Component GUI count for button', ()=>{
  const fixture = TestBed.createComponent(AddchapterComponent);
  const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
  const inputElements = formElement.querySelectorAll("button");
  expect(inputElements.length).toEqual(1);
});

it('Add Chapter form Component GUI count for input', ()=>{
  const fixture = TestBed.createComponent(AddchapterComponent);
  const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
  const inputElements = formElement.querySelectorAll("input");
  expect(inputElements.length).toEqual(17);
});

it('Add Chapter form initial value', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  const myFormGroup = app.addchapterform

```

```
const myFormValues = {
  id : '',
  chapter1name : '',
  chapter1id : '',
}

expect(myFormGroup.valid).toBeTruthy();
});

it('Add Chapter form by Valid data', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  app.addchapterform.controls?['id'].setValue("100");
  app.addchapterform.controls?['chapter1name'].setValue("Java");
  app.addchapterform.controls?['chapter1id'].setValue("101");
  expect(app.addchapterform.valid).toBeTruthy();
});

it('Add Chapter form by Invalid data', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  app.addchapterform.controls?['id'].setValue("");
  app.addchapterform.controls?['chapter1name'].setValue("");
  app.addchapterform.controls?['chapter1id'].setValue("");
  expect(app.addchapterform.valid).toBeTruthy();
});

it('Insert new Chapter to the database', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  app.addchapterform.controls?['id'].setValue("101");
  app.addchapterform.controls?['chapter1name'].setValue("Spring");
  app.addchapterform.controls?['chapter1id'].setValue("109");
  expect(app.addchapterform.valid).toBeTruthy();
});

it('Update existing Chapter to the database', () => {
  const fixture = TestBed.createComponent(AddchapterComponent);
  const app = fixture.componentInstance;
  app.addchapterform.controls?['id'].setValue("101");
```

```

        app.addchapterform.controls?['chapter1name'].setValue("Spring
Boot");
        app.addchapterform.controls?['chapter1id'].setValue("109");
        expect(app.addchapterform.valid).toBeTruthy();
    });
});

```

- **Add Course Component :**

```

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

import { AddcourseComponent } from './addcourse.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { RegistrationComponent } from
'../registration/registration.component';
import { AppComponent } from 'src/app/app.component';

describe('Add Course form Component Test', () => {
    let component: AddcourseComponent;
    let fixture: ComponentFixture<AddcourseComponent>;
    beforeEach(async () => {
        await TestBed.configureTestingModule({
            declarations: [AddcourseComponent, AppComponent],
            imports:[HttpClientModule,
                ReactiveFormsModule,
                BrowserModule,
                AppRoutingModuleModule,
                FormsModule,
                ReactiveFormsModule,

```

```

        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
    ],
    providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
    ]
    }).compileComponents();
});
it('To test the Add Course component created', () => {
    const fixture = TestBed.createComponent(AddcourseComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
});

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

it('Add Course form Component GUI count for button', ()=>{
    const fixture = TestBed.createComponent(AddcourseComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");
    expect(inputElements.length).toEqual(1);
});

it('Add Course form Component GUI count for input', ()=>{
    const fixture = TestBed.createComponent(AddcourseComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(5);
});

```

```

it('Add Course form initial value', () => {
  const fixture = TestBed.createComponent(AddcourseComponent);
  const app = fixture.componentInstance;
  const myFormGroup = app.addcourseform
  const myFormValues = {
    id : '',
    courseid : '',
    coursename : '',
    coursetype : '',
    description : '',
    enrolledcount : '',
    enrolled date : '',
    instructorinstitution : '',
    language : '',
    skilllevel : '',
  }
  expect(myFormGroup.valid).toBeTruthy();
});

it('Add Course form by Valid data', () => {
  const fixture = TestBed.createComponent(AddcourseComponent);
  const app = fixture.componentInstance;
  app.add course form.controls?.['id'].setValue("100");
  app.add course form.controls?.['courseid'].setValue("190");
  app.add course form.controls?.['coursename'].setValue("Java");
  app.add course form.controls?.['coursetype'].setValue("Website");
  app.add course form.controls?.['description'].setValue("Java with
full tutorials");
  app.add course form.controls?.['enrolledcount'].setValue("1");
  app.add course form.controls?.['enrolled
date'].setValue("01/01/2024");

  app.addcourseform.controls?.['instructorinstitution'].setValue("VVE");
  app.addcourseform.controls?.['language'].setValue("English");
  app.addcourseform.controls?.['skilllevel'].setValue("Basic");
  expect(app.addcourseform.valid).toBeTruthy();
});

it('Add Course form by Invalid data', () => {

```

```

    const fixture = TestBed.createComponent(AddcourseComponent);
    const app = fixture.componentInstance;
    app.addcourseform.controls?['id'].setValue("");
    app.addcourseform.controls?['courseid'].setValue("");
    app.addcourseform.controls?['coursename'].setValue("");
    app.addcourseform.controls?['coursetype'].setValue("");
    app.addcourseform.controls?['description'].setValue("");
    app.addcourseform.controls?['enrolledcount'].setValue("");
    app.add course form.controls?['enrolled date'].setValue("");
    app.add course
form.controls?['instructorinstitution'].setValue("");
    app.add course form.controls?['language'].setValue("");
    app.add course form.controls?['skilllevel'].setValue("");
    expect(app.addcourseform.valid).toBeTruthy();
  });

  it('Insert new Course to the database', () => {
    const fixture = TestBed.createComponent(AddcourseComponent);
    const app = fixture.componentInstance;
    app.add course form.controls?['id'].setValue("91");
    app.add course form.controls?['courseid'].setValue("80");
    app.add course form.controls?['coursename'].setValue("Spring");
    app.add course form.controls?['coursetype'].setValue("website");
    app.add course form.controls?['description'].setValue("Spring");
    app.add course form.controls?['enrolledcount'].setValue("0");
    app.add course form.controls?['enrolled
date'].setValue("01/01/2024");
    app.add course
form.controls?['instructorinstitution'].setValue("VVE");
    app.add course form.controls?['language'].setValue("English");
    app.add course form.controls?['skilllevel'].setValue("Basic");
    expect(app.addcourseform.valid).toBeTruthy();
  });

  it('Update existing Course to the database', () => {
    const fixture = TestBed.createComponent(AddcourseComponent);
    const app = fixture.componentInstance;
    app.add course form.controls?['id'].setValue("91");
    app.add course form.controls?['courseid'].setValue("80");

```



```

        app.add course form.controls?.['coursename'].setValue("Spring
Boot");
        app.add course form.controls?.['coursetype'].setValue("website");
        app.add course form.controls?.['description'].setValue("Spring
Boot");
        app.add course form.controls?.['enrolledcount'].setValue("1");
        app.add course form.controls?.['enrolled
date'].setValue("01/01/2024");
        app.add course
form.controls?.['instructorinstitution'].setValue("VVE");
        app.add course form.controls?.['language'].setValue("English");
        app.add course form.controls?.['skilllevel'].setValue("Basic");
        expect(app.addcourseform.valid).toBeTruthy();
    });
});

```

- **Add Teacher Component :**

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AddteacherComponent } from '../addteacher.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { AddchapterComponent } from
'../addchapter/addchapter.component';
describe('Add Teacher Component Test', () => {
    let component: AddteacherComponent;
    let fixture: ComponentFixture<AddteacherComponent>;
    beforeEach(async () => {
        await TestBed.configureTestingModule({
            declarations: [AddteacherComponent],

```

```

    imports:[HttpClientModule,
      ReactiveFormsModule,
      BrowserModule,
      AppRoutingModule,
      FormsModule,
      ReactiveFormsModule,
      HttpClientModule,
      ReactiveFormsModule,
      FormsModule,
      HttpClientModule
    ],
    providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
    ]
  }).compileComponents();
});

it('To test the Add Teacher component created', () => {
  const fixture = TestBed.createComponent(AddteacherComponent);
  const app = fixture.componentInstance;
  expect(app).toBeTruthy();
});

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

it('Add Teacher form Component GUI count for button', ()=>{
  const fixture = TestBed.createComponent(AddteacherComponent);
  const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
  const inputElements = formElement.querySelectorAll("button");
  expect(inputElements.length).toEqual(1);
});

it('Add Teacher form Component GUI count for input', ()=>{
  const fixture = TestBed.createComponent(AddteacherComponent);

```

```

        const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
        const inputElements = formElement.querySelectorAll("input");
        expect(inputElements.length).toEqual(10);
    });

    it('Add Teacher form initial value', () => {
        const fixture = TestBed.createComponent(AddteacherComponent);
        const app = fixture.componentInstance;
        const myFormGroup = app.addteacherform
        const myFormValues = {
            teacherid : '',
            teachername : '',
            email : '',
            degreecompleted : '',
            department : '',
            experience : '',
            gender : '',
            institutionname : '',
            mobile : '',
            password : ''
        }
        expect(myFormGroup.valid).toBeTruthy();
    });

    it('Add Teacher form by Valid data', () => {
        const fixture = TestBed.createComponent(AddteacherComponent);
        const app = fixture.componentInstance;
        app.addteacherform.controls?['teacherid'].setValue("100");
app.addteacherform.controls?['teachername'].setValue("PravinKumar");
app.addteacherform.controls?['email'].setValue("rjpravin123@gmail.com")
;
        app.addteacherform.controls?['degreecompleted'].setValue("PHD");
        app.addteacherform.controls?['department'].setValue("ECE");
        app.addteacherform.controls?['experience'].setValue("3");
        app.addteacherform.controls?['gender'].setValue("Male");
        app.addteacherform.controls?['institutionname'].setValue("VVE");
    });

```

```

        app.addteacherform.controls?.['mobile'].setValue("9875234160");

app.addteacherform.controls?.['password'].setValue("PravinKumar@11946");
        expect(app.addteacherform.valid).toBeTruthy();
    });

    it('Add Teacher form by Invalid data', () => {
        const fixture = TestBed.createComponent(AddteacherComponent);
        const app = fixture.componentInstance;
        app.addteacherform.controls?.['teacherid'].setValue("");
        app.addteacherform.controls?.['teachername'].setValue("");
        app.addteacherform.controls?.['email'].setValue("");
        app.addteacherform.controls?.['degreecompleted'].setValue("");
        app.addteacherform.controls?.['department'].setValue("");
        app.addteacherform.controls?.['experience'].setValue("");
        app.addteacherform.controls?.['gender'].setValue("");
        app.addteacherform.controls?.['institutionname'].setValue("");
        app.addteacherform.controls?.['mobile'].setValue("");
        app.addteacherform.controls?.['password'].setValue("");
        expect(app.addteacherform.valid).toBeTruthy();
    });

    it('Insert new Teacher to the database', () => {
        const fixture = TestBed.createComponent(AddteacherComponent);
        const app = fixture.componentInstance;
        app.addteacherform.controls?.['teacherid'].setValue("101");

app.addteacherform.controls?.['teachername'].setValue("PravinKumar");

app.addteacherform.controls?.['email'].setValue("rjpravin123@gmail.com")
;

        app.addteacherform.controls?.['degreecompleted'].setValue("PHD");
        app.addteacherform.controls?.['department'].setValue("ECE");
        app.addteacherform.controls?.['experience'].setValue("3");
        app.addteacherform.controls?.['gender'].setValue("Male");
        app.addteacherform.controls?.['institutionname'].setValue("VVE");
        app.addteacherform.controls?.['mobile'].setValue("9875234160");

app.addteacherform.controls?.['password'].setValue("PravinKumar@11946");

```

```

    expect(app.addteacherform.valid).toBeTruthy();
  });

});

```

- **View Course List Component :**

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CourselistComponent } from '../courselist.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { AddchapterComponent } from
'../addchapter/addchapter.component';

describe('View Course List Component Test', () => {
  let component: CourselistComponent;
  let fixture: ComponentFixture<CourselistComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [CourselistComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModuleModule,
        FormsModule,
        ReactiveFormsModule,
        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule

```

```

    ],
    providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
    ]
    }).compileComponents();
  });
  it('To test the View Course List component created', () => {
    const fixture = TestBed.createComponent(CourselistComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  // it(`should have as title 'View Course List Component'`, () => {
  //   const fixture = TestBed.createComponent(AddchapterComponent);
  //   const app = fixture.componentInstance;
  //   expect(app.title).toEqual('View Course List Component');
  // });

  // it('Add Cahpter form Component GUI count for button', ()=>{
  //   const fixture = TestBed.createComponent(AddchapterComponent);
  //   const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
  //   const inputElements = formElement.querySelectorAll("a");
  //   expect(inputElements.length).toEqual(1);
  // });
  });

```

- **Login Component for Admin :**

```

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

import { LoginComponent } from '../login.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';

```

```

import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { RegistrationUserComponent } from
'../registrationuser/registrationuser.component';

describe('Login Admin form Component Test', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [LoginComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModule,
        FormsModule,
        ReactiveFormsModule,
        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
      ],
      providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
        ]
    }).compileComponents();
  });
  it('To test the Login Admin component created', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

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

```

```

    const app = fixture.componentInstance;
    expect(app.title).toEqual('Admin login Component');
  });

  it('Admin Login form Component GUI count for button', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");
    expect(inputElements.length).toEqual(1);
  });

  it('Admin Login form Component GUI count for input', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(2);
  });

  it('Admin Login form initial value', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const app = fixture.componentInstance;
    const myFormGroup = app.adminform
    const myFormValues = {
      email : '',
      password : ''
    }
    expect(myFormGroup.valid).toBeTruthy();
  });

  it('Admin Login form valid by Valid data', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const app = fixture.componentInstance;
    app.adminform.controls?['email'].setValue("admin@gmail.com");
    app.adminform.controls?['password'].setValue("admin123");
    expect(app.adminform.valid).toBeTruthy();
  });

```



```

    it('Admin Login form valid by invalid data', () => {
      const fixture = TestBed.createComponent(LoginComponent);
      const app = fixture.componentInstance;
      app.adminform.controls?['email'].setValue("");
      app.adminform.controls?['password'].setValue("");
      expect(app.adminform.valid).toBeTruthy();
    });
  });
});

```

- **Login Component for User(Teacher) :**

```

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

import { LoginTeacherComponent } from '../loginTeacher.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { LoginComponent } from '../login/login.component';

describe('Login Teacher form Component Test', () => {
  let component: LoginTeacherComponent;
  let fixture: ComponentFixture<LoginTeacherComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [LoginTeacherComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModuleModule,
        FormsModule,
        ReactiveFormsModule,

```

```

        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
    ],
    providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
    ]
    }).compileComponents();
});
it('To test the Login Teacher component created', () => {
    const fixture = TestBed.createComponent(LoginTeacherComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
});

it(`should have as title 'Teacher Login Component'`, () => {
    const fixture = TestBed.createComponent(LoginTeacherComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('Teacher login Component');
});

it('Teacher Login form Component GUI count for button',()=>{
    const fixture = TestBed.createComponent(LoginTeacherComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");
    expect(inputElements.length).toEqual(1);
});

it('Teacher Login form Component GUI count for input',()=>{
    const fixture = TestBed.createComponent(LoginTeacherComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(3);
});

```

```

it('Teacher Login form initial value', () => {
  const fixture = TestBed.createComponent(LoginTeacherComponent);
  const app = fixture.componentInstance;
  const myFormGroup = app.teacherform
  const myFormValues = {
    email : '',
    password : ''
  }
  expect(myFormGroup.valid).toBeTruthy();
});

it('Teacher Login form valid by Valid data', () => {
  const fixture = TestBed.createComponent(LoginTeacherComponent);
  const app = fixture.componentInstance;

app.teacherform.controls?['email'].setValue("anitha123@gmail.com");
  app.teacherform.controls?['password'].setValue("Anitha%23");
  expect(app.teacherform.valid).toBeTruthy();
});

it('Teacher Login form valid by invalid data', () => {
  const fixture = TestBed.createComponent(LoginTeacherComponent);
  const app = fixture.componentInstance;
  app.teacherform.controls?['email'].setValue("");
  app.teacherform.controls?['password'].setValue("");
  expect(app.teacherform.valid).toBeTruthy();
});
});

```

- **Login Component for User(Student):**

```

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

import { LoginUserComponent } from './loginUser.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';

```

```

import { AdminService } from 'src/app/services/admin.service';
import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { LoginTeacherComponent } from
'../loginTeacher/loginTeacher.component';

describe('Login User form Component Test', () => {
  let component: LoginUserComponent;
  let fixture: ComponentFixture<LoginUserComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [LoginUserComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModule,
        FormsModule,
        ReactiveFormsModule,
        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
      ],
      providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
      ]
    }).compileComponents();
  });
  it('To test the Login user component created', () => {
    const fixture = TestBed.createComponent(LoginUserComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'User Login Component'`, () => {

```

```

    const fixture = TestBed.createComponent(LoginUserComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('User login Component');
  });

  it('User Login form Component GUI count for button', () => {
    const fixture = TestBed.createComponent(LoginUserComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");
    expect(inputElements.length).toEqual(1);
  });

  it('User Login form Component GUI count for input', () => {
    const fixture = TestBed.createComponent(LoginUserComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(3);
  });

  it('User Login form initial value', () => {
    const fixture = TestBed.createComponent(LoginUserComponent);
    const app = fixture.componentInstance;
    const myFormGroup = app.userform
    const myFormValues = {
      email : '',
      password : ''
    }
    expect(myFormGroup.valid).toBeTruthy();
  });

  it('User Login form valid by Valid data', () => {
    const fixture = TestBed.createComponent(LoginUserComponent);
    const app = fixture.componentInstance;

app.userform.controls?.['email'].setValue("praivnkumar123@gmail.com");
    app.userform.controls?.['password'].setValue("PravinKumar@11946");
    expect(app.userform.valid).toBeTruthy();
  });

```

```

    });

    it('User Login form valid by invalid data', () => {
        const fixture = TestBed.createComponent(LoginUserComponent);
        const app = fixture.componentInstance;
        app.userform.controls?['email'].setValue("");
        app.userform.controls?['password'].setValue("");
        expect(app.userform.valid).toBeTruthy();
    });
});

```

- **Registration for User(Teacher) :**

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RegistrationComponent } from '../registration.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { RegistrationService } from
'src/app/services/registration.service';
import { LoginService } from 'src/app/services/login.service';
import { AdminService } from 'src/app/services/admin.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { TeacherprofileComponent } from
'../teacherprofile/teacherprofile.component';
import { TeacherdashboardComponent } from
'../teacherdashboard/teacherdashboard.component';
import { LoginTeacherComponent } from
'../loginTeacher/loginTeacher.component';
import { TeacherlistComponent } from
'../teacherlist/teacherlist.component';

describe('Registration Teacher form Component Test', () => {
    let component: RegistrationComponent;
    let fixture: ComponentFixture<RegistrationComponent>;

```

```

beforeEach(async () => {
    await TestBed.configureTestingModule({
        declarations:
[RegistrationComponent,TeacherprofileComponent,TeacherdashboardComponent
,LoginTeacherComponent,TeacherlistComponent],
        imports:[HttpClientModule,
            ReactiveFormsModule,
            BrowserModule,
            AppRoutingModule,
            FormsModule,
            ReactiveFormsModule,
            HttpClientModule,
            ReactiveFormsModule,
            FormsModule,
            HttpClientModule
        ],
        providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
        ]
    }).compileComponents();
});

it('To test the Teacher Registration component created', () => {
    const fixture = TestBed.createComponent(RegistrationComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
});

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

it('Registration Teacher form Component GUI count for button',()=>{
    const fixture = TestBed.createComponent(RegistrationComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");

```

```

        expect(inputElements.length).toEqual(1);
    });

    it('Registration Teacher form Component GUI count for input', () => {
        const fixture = TestBed.createComponent(RegistrationComponent);
        const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
        const inputElements = formElement.querySelectorAll("input");
        expect(inputElements.length).toEqual(11);
    });

    it('Registration Teacher form initial value', () => {
        const fixture = TestBed.createComponent(RegistrationComponent);
        const app = fixture.componentInstance;
        const myFormGroup = app.teacherform
        const myFormValues = {
            teacherid : '',
            teachername : '',
            email : '',
            degreecompleted : '',
            department : '',
            experience : '',
            gender : '',
            institutionname : '',
            mobile : '',
            password : ''
        }
        expect(myFormGroup.valid).toBeTruthy();
    });

    it('Registration teacher form valid by Valid data', () => {
        const fixture = TestBed.createComponent(RegistrationComponent);
        const app = fixture.componentInstance;
        app.teacherform.controls?['teacherid'].setValue("100");
        app.teacherform.controls?['teachername'].setValue("PravinKumar");
app.teacherform.controls?['email'].setValue("rjpravin123@gmail.com");
        app.teacherform.controls?['degreecompleted'].setValue("PHD");
        app.teacherform.controls?['department'].setValue("ECE");
    });

```



```

        app.teacherform.controls?['experience'].setValue("3");
        app.teacherform.controls?['gender'].setValue("Male");
        app.teacherform.controls?['institutionname'].setValue("VVE");
        app.teacherform.controls?['mobile'].setValue("9875234160");

app.teacherform.controls?['password'].setValue("PravinKumar@11946");
        expect(app.teacherform.valid).toBeTruthy();
    });

    it('Registration Teacher form valid by invalid data', () => {
        const fixture = TestBed.createComponent(RegistrationComponent);
        const app = fixture.componentInstance;
        app.teacherform.controls?['teacherid'].setValue("");
        app.teacherform.controls?['teachername'].setValue("");
        app.teacherform.controls?['email'].setValue("");
        app.teacherform.controls?['degreecompleted'].setValue("");
        app.teacherform.controls?['department'].setValue("");
        app.teacherform.controls?['experience'].setValue("");
        app.teacherform.controls?['gender'].setValue("");
        app.teacherform.controls?['institutionname'].setValue("");
        app.teacherform.controls?['mobile'].setValue("");
        app.teacherform.controls?['password'].setValue("");
        expect(app.teacherform.valid).toBeTruthy();
    });
});

```

- **Registration for User(Student) :**

```

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

import { RegistrationUserComponent } from
'./registrationuser.component';
import { UserlistComponent } from '../userlist/userlist.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModuleModule } from 'src/app/app-routing.module';
import { AdminService } from 'src/app/services/admin.service';

```

```

import { LoginService } from 'src/app/services/login.service';
import { RegistrationService } from
'src/app/services/registration.service';
import { TeacherService } from 'src/app/services/teacher.service';
import { UserService } from 'src/app/services/user.service';
import { RegistrationComponent } from
'../registration/registration.component';

describe('Registration User form Component Test', () => {
  let component: RegistrationUserComponent;
  let fixture: ComponentFixture<RegistrationUserComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [RegistrationUserComponent],
      imports:[HttpClientModule,
        ReactiveFormsModule,
        BrowserModule,
        AppRoutingModule,
        FormsModule,
        ReactiveFormsModule,
        HttpClientModule,
        ReactiveFormsModule,
        FormsModule,
        HttpClientModule
      ],
      providers:[
RegistrationService,LoginService,AdminService,TeacherService,UserService
        ]
    }).compileComponents();
  });
  it('To test the user Registration component created', () => {
    const fixture =
TestBed.createComponent(RegistrationUserComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'Registration User Component'`, () => {

```

```

    const fixture =
TestBed.createComponent(RegistrationUserComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('Registration User Component');
  });

  it('Registration User form Component GUI count for button', () => {
    const fixture =
TestBed.createComponent(RegistrationUserComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("button");
    expect(inputElements.length).toEqual(1);
  });

  it('Registration User form Component GUI count for input', () => {
    const fixture =
TestBed.createComponent(RegistrationUserComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#ngForm");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(9);
  });

  it('Registration User form initial value', () => {
    const fixture =
TestBed.createComponent(RegistrationUserComponent);
    const app = fixture.componentInstance;
    const myFormGroup = app.userform
    const myFormValues = {
      userid : '',
      username : '',
      email : '',
      address : '',
      gender : '',
      profession : '',
      password : ''
    }
    expect(myFormGroup.valid).toBeTruthy();
  });

```

```

    });

    it('Registration User form valid by Valid data', () => {
        const fixture =
TestBed.createComponent(RegistrationUserComponent);
        const app = fixture.componentInstance;
        app.userform.controls?['userid'].setValue("100");
        app.userform.controls?['username'].setValue("PravinKumar");
app.userform.controls?['email'].setValue("rjpravin123456789@gmail.com")
;
        app.userform.controls?['address'].setValue("Bengaluru");
        app.userform.controls?['gender'].setValue("Male");
        app.userform.controls?['profession'].setValue("CSE");
        app.userform.controls?['password'].setValue("PravinKumar@11946");
        expect(app.userform.valid).toBeTruthy();
    });

    it('Registration user form valid by invalid data', () => {
        const fixture =
TestBed.createComponent(RegistrationUserComponent);
        const app = fixture.componentInstance;
        app.userform.controls?['userid'].setValue("");
        app.userform.controls?['username'].setValue("");
        app.userform.controls?['email'].setValue("");
        app.userform.controls?['address'].setValue("");
        app.userform.controls?['gender'].setValue("");
        app.userform.controls?['profession'].setValue("");
        app.userform.controls?['password'].setValue("");
        expect(app.userform.valid).toBeTruthy();
    });
});

```