

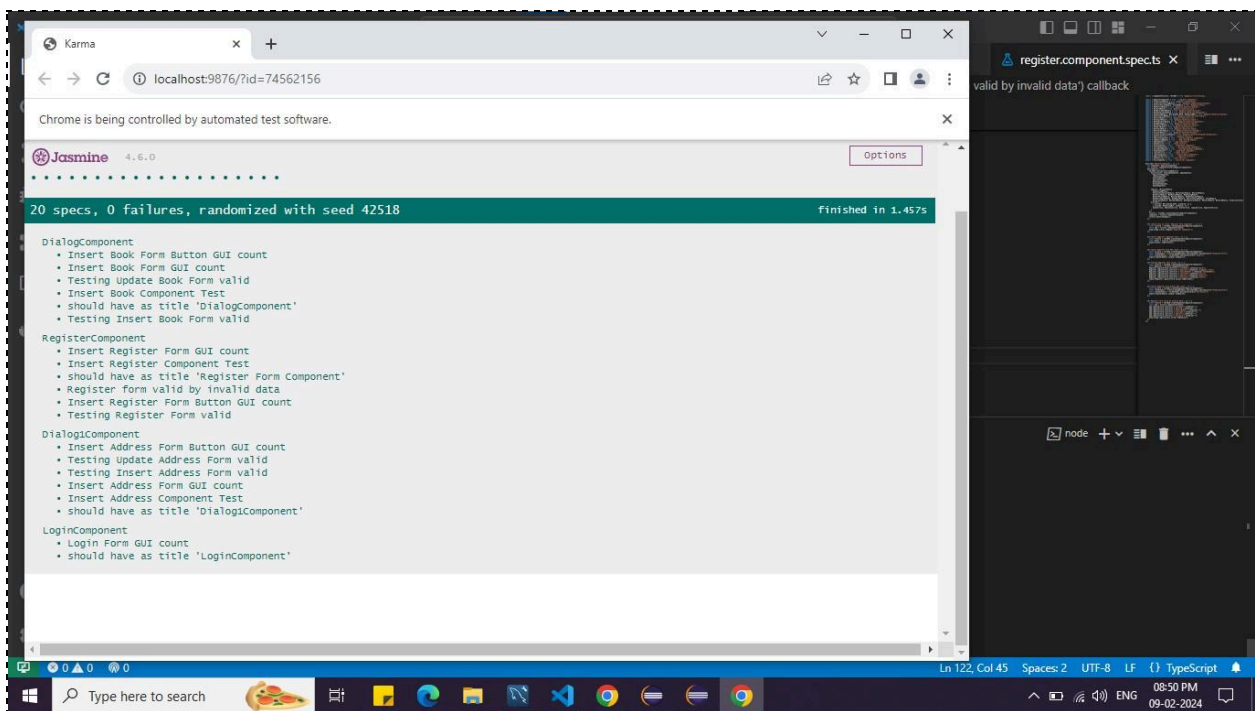
## E - Library Management System

### Jasmine & Karma

#### Console output of VS Code

```
09 02 2024 22:10:44.399:INFO [Chrome 121.0.0.0 (Windows 10)]: Connected on socket fXcN8IgcB2kpVS-RAAAN with id 74562156
Chrome 121.0.0.0 (Windows 10): Executed 20 of 20 SUCCESS (0.36 secs / 0.34 secs)
TOTAL: 20 SUCCESS
```

#### Karma Generated the Report:



#### Dialog.component.spec.ts

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

import { DialogComponent } from './dialog.component';
import { HttpClientModule } from '@angular/common/http';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
```

```
import { MatNativeDateModule } from '@angular/material/core';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatDialogModule, MAT_DIALOG_DATA, MatDialogRef } from
 '@angular/material/dialog';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatInputModule } from '@angular/material/input';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatRadioModule } from '@angular/material/radio';
import { MatSelectModule } from '@angular/material/select';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from
 '@angular/platform-browser/animations';
import { AddressService } from '../address.service';
import { AddressComponent } from '../address/address.component';
import { AppRoutingModuleModule } from '../app-routing.module';
import { AppComponent } from '../app.component';
import { Dialog1Component } from '../dialog1/dialog1.component';
import { HomeComponent } from '../home/home.component';
import { LoginService } from '../login.service';
import { LoginComponent } from '../login/login.component';
import { RegisterService } from '../register.service';
import { RegisterComponent } from '../register/register.component';
import { UserComponent } from '../user/user.component';
import { BookComponent } from '../book/book.component';
import { BookService } from '../book.service';
import { UserService } from '../user.service';
import { HttpClientTestingModule } from '@angular/common/http/testing';

describe('DialogComponent', () => {
  let component: DialogComponent;
  let fixture: ComponentFixture<DialogComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [DialogComponent, AppComponent,
        RegisterComponent,
```

```

    LoginComponent,
    UserComponent,
    AddressComponent,
    BookComponent,
    Dialog1Component,
    HomeComponent],

imports:[BrowserModule,
    AppRoutingModule,
    BrowserModule, MatToolbarModule, MatIconModule,
    MatSelectModule, MatButtonModule, MatDialogModule,
    MatFormFieldModule, MatInputModule, MatDatepickerModule,
    MatNativeDateModule, MatRadioModule, ReactiveFormsModule,
FormsModule,
    HttpClientModule, MatTableModule, MatPaginatorModule,
MatSortModule, MatCardModule, HttpClientTestingModule],
providers:[
    { provide: MAT_DIALOG_DATA, useValue: {} },
    { provide: MatDialogRef, useValue: {} },

BookService,AddressService,UserService,LoginService,RegisterService
    ]
});
fixture = TestBed.createComponent(DialogComponent);
component = fixture.componentInstance;
fixture.detectChanges();
});

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

```

```

it('Insert Book Component Test', () => {
  const fixture = TestBed.createComponent(DialogComponent);
  const Book = fixture.componentInstance;
  expect(Book).toBeTruthy();
});

it('Insert Book Form GUI count', () => {
  const fixture = TestBed.createComponent(DialogComponent);
  const formElement =
fixture.debugElement.nativeElement.querySelector("#bookForm");
  const inputElements = formElement.querySelectorAll("input");
  expect(inputElements.length).toEqual(5);
});

it('Testing Insert Book Form valid', () => {
  const fixture = TestBed.createComponent(DialogComponent);
  const Book = fixture.componentInstance;
  Book.bookForm.controls?['bookId'].setValue("204");
  Book.bookForm.controls?['bookTitle'].setValue("Maths Intro");
  Book.bookForm.controls?['authorName'].setValue("Mani");
  Book.bookForm.controls?['edition'].setValue("1999");
  Book.bookForm.controls?['bookType'].setValue("Maths");
  Book.bookForm.controls?['bookDetails'].setValue("Mathematics");

  expect(Book.bookForm.valid).toBeTruthy();
});

it('Testing Update Book Form valid', () => {
  const fixture = TestBed.createComponent(DialogComponent);
  const Book = fixture.componentInstance;
  Book.bookForm.controls?['bookId'].setValue("204");

```

```

    Book.bookForm.controls?.['bookTitle'].setValue("Maths Intro");
    Book.bookForm.controls?.['authorName'].setValue("Mani");
    Book.bookForm.controls?.['edition'].setValue("1999");
    Book.bookForm.controls?.['bookType'].setValue("Maths");
    Book.bookForm.controls?.['bookDetails'].setValue("Mathematics");

    expect(Book.bookForm.valid).toBeTruthy();
  });

  it('Insert Book Form Button GUI count', () => {
    const fixture = TestBed.createComponent(DialogComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#bookForm");
    const inputElements = formElement.querySelectorAll("mat-option");
    expect(inputElements.length).toEqual(0);
  });
});

```

## Dialog1.component.spec.ts

```

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

import { Dialog1Component } from './dialog1.component';
import { HttpClientModule } from '@angular/common/http';
import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatNativeDateModule } from '@angular/material/core';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatDialogModule, MAT_DIALOG_DATA, MatDialogRef } from
 '@angular/material/dialog';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatInputModule } from '@angular/material/input';

```

```

import { MatPaginatorModule } from '@angular/material/paginator';
import { MatRadioModule } from '@angular/material/radio';
import { MatSelectModule } from '@angular/material/select';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from
 '@angular/platform-browser/animations';
import { AddressService } from '../address.service';
import { AddressComponent } from '../address/address.component';
import { AppRoutingModuleModule } from '../app-routing.module';
import { AppComponent } from '../app.component';
import { BookService } from '../book.service';
import { BookComponent } from '../book/book.component';
import { DialogComponent } from '../dialog/dialog.component';
import { HomeComponent } from '../home/home.component';
import { LoginService } from '../login.service';
import { LoginComponent } from '../login/login.component';
import { RegisterService } from '../register.service';
import { RegisterComponent } from '../register/register.component';
import { UserService } from '../user.service';
import { UserComponent } from '../user/user.component';

describe('Dialog1Component', () => {
  let component: Dialog1Component;
  let fixture: ComponentFixture<Dialog1Component>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [DialogComponent, AppComponent,
        RegisterComponent,
        LoginComponent,
        UserComponent,
        AddressComponent,
        BookComponent,
        Dialog1Component,
        HomeComponent],

      imports: [BrowserModule,

```

```

        AppRoutingModule,
        BrowserAnimationsModule, MatToolbarModule, MatIconModule,
        MatSelectModule, MatButtonModule, MatDialogModule,
        MatFormFieldModule, MatInputModule, MatDatepickerModule,
        MatNativeDateModule, MatRadioModule, ReactiveFormsModule,
FormsModule,
        HttpClientModule, MatTableModule, MatPaginatorModule,
MatSortModule, MatCardModule, HttpClientTestingModule],
    providers:[
        { provide: MAT_DIALOG_DATA, useValue: {} },
        { provide: MatDialogRef, useValue: {} },

BookService, AddressService, UserService, LoginService, RegisterService
    ]
});
fixture = TestBed.createComponent(Dialog1Component);
component = fixture.componentInstance;
fixture.detectChanges();
});

it(`should have as title 'Dialog1Component'`, () => {
    const fixture = TestBed.createComponent(Dialog1Component);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('Dialog1Component');
});

it('Insert Address Component Test', () => {
    const fixture = TestBed.createComponent(Dialog1Component);
    const Address = fixture.componentInstance;
    expect(Address).toBeTruthy();
});

```

```
it('Insert Address Form GUI count', () => {
  const fixture = TestBed.createComponent(Dialog1Component);
  const formElement =
fixture.debugElement.nativeElement.querySelector("#addressForm");
  const inputElements = formElement.querySelectorAll("input");
  expect(inputElements.length).toEqual(5);
});

it('Testing Insert Address Form valid', () => {
  const fixture = TestBed.createComponent(Dialog1Component);
  const Address = fixture.componentInstance;
  Address.addressForm.controls?['doorNo'].setValue("204");
  Address.addressForm.controls?['street'].setValue("Vnr");
  Address.addressForm.controls?['city'].setValue("Madurai");
  Address.addressForm.controls?['state'].setValue("TamilNadu");
  Address.addressForm.controls?['pincode'].setValue("626789");

  expect(Address.addressForm.valid).toBeTruthy();
});

it('Testing Update Address Form valid', () => {
  const fixture = TestBed.createComponent(Dialog1Component);
  const Address = fixture.componentInstance;
  Address.addressForm.controls?['doorNo'].setValue("204");
  Address.addressForm.controls?['street'].setValue("Vnr");
  Address.addressForm.controls?['city'].setValue("Madurai");
  Address.addressForm.controls?['state'].setValue("TamilNadu");
  Address.addressForm.controls?['pincode'].setValue("626789");

  expect(Address.addressForm.valid).toBeTruthy();
});
```



```

    it('Insert Address Form Button GUI count', () => {
        const fixture = TestBed.createComponent(Dialog1Component);
        const formElement =
fixture.debugElement.nativeElement.querySelector("#addressForm");
        const inputElements = formElement.querySelectorAll("mat-option");
        expect(inputElements.length).toEqual(0);
    });
});

```

## login.component.spec.ts

```

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

import { LoginComponent } from '../login.component';
import { HttpClientModule } from '@angular/common/http';
import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatNativeDateModule } from '@angular/material/core';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatDialogModule, MAT_DIALOG_DATA, MatDialogRef } from
 '@angular/material/dialog';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatInputModule } from '@angular/material/input';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatRadioModule } from '@angular/material/radio';
import { MatSelectModule } from '@angular/material/select';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from
 '@angular/platform-browser/animations';

```

```

import { AddressService } from '../address.service';
import { AddressComponent } from '../address/address.component';
import { AppRoutingModuleModule } from '../app-routing.module';
import { AppComponent } from '../app.component';
import { BookService } from '../book.service';
import { BookComponent } from '../book/book.component';
import { DialogComponent } from '../dialog/dialog.component';
import { Dialog1Component } from '../dialog1/dialog1.component';
import { HomeComponent } from '../home/home.component';
import { LoginService } from '../login.service';
import { RegisterService } from '../register.service';
import { RegisterComponent } from '../register/register.component';
import { UserService } from '../user.service';
import { UserComponent } from '../user/user.component';

describe('LoginComponent', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [DialogComponent, AppComponent,
        RegisterComponent,
        LoginComponent,
        UserComponent,
        AddressComponent,
        BookComponent,
        Dialog1Component,
        HomeComponent],

      imports: [BrowserModule,
        AppRoutingModuleModule,
        BrowserAnimationsModule, MatToolbarModule, MatIconModule,
        MatSelectModule, MatButtonModule, MatDialogModule,
        MatFormFieldModule, MatInputModule, MatDatepickerModule,
        MatNativeDateModule, MatRadioModule, ReactiveFormsModule,
FormsModule,
        HttpClientModule, MatTableModule, MatPaginatorModule,
MatSortModule, MatCardModule, HttpClientTestingModule],
      providers: [

```

```

        { provide: MAT_DIALOG_DATA, useValue: {} },
        { provide: MatDialogRef, useValue: {} },

BookService,AddressService,UserService,LoginService,RegisterService
    ]
    });

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

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

it('Login Form GUI count', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#loginForm");
    const inputElements = formElement.querySelectorAll("input");
    const inputElements1 = formElement.querySelectorAll("button");
    expect(inputElements1.length).toEqual(1);
    expect(inputElements.length).toEqual(2);
});
});

```

## register.component.spec.ts

```

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

import { RegisterComponent } from './register.component';
import { HttpClientModule } from '@angular/common/http';
import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';

```

```
import { MatCardModule } from '@angular/material/card';
import { MatNativeDateModule } from '@angular/material/core';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatDialogModule, MAT_DIALOG_DATA, MatDialogRef } from
 '@angular/material/dialog';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatInputModule } from '@angular/material/input';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatRadioModule } from '@angular/material/radio';
import { MatSelectModule } from '@angular/material/select';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from
 '@angular/platform-browser/animations';
import { AddressService } from '../address.service';
import { AddressComponent } from '../address/address.component';
import { AppRoutingModuleModule } from '../app-routing.module';
import { AppComponent } from '../app.component';
import { BookService } from '../book.service';
import { BookComponent } from '../book/book.component';
import { DialogComponent } from '../dialog/dialog.component';
import { Dialog1Component } from '../dialog1/dialog1.component';
import { HomeComponent } from '../home/home.component';
import { LoginService } from '../login.service';
import { LoginComponent } from '../login/login.component';
import { RegisterService } from '../register.service';
import { UserService } from '../user.service';
import { UserComponent } from '../user/user.component';

describe('RegisterComponent', () => {
  let component: RegisterComponent;
  let fixture: ComponentFixture<RegisterComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [DialogComponent, AppComponent,
        RegisterComponent,
```

```

        LoginComponent,
        UserComponent,
        AddressComponent,
        BookComponent,
        Dialog1Component,
        HomeComponent],

imports: [BrowserModule,
        AppRoutingModule,
        BrowserAnimationsModule, MatToolbarModule, MatIconModule,
        MatSelectModule, MatButtonModule, MatDialogModule,
        MatFormFieldModule, MatInputModule, MatDatepickerModule,
        MatNativeDateModule, MatRadioModule, ReactiveFormsModule,
FormsModule,
        HttpClientModule, MatTableModule, MatPaginatorModule,
MatSortModule, MatCardModule, HttpClientTestingModule],
        providers: [
            { provide: MAT_DIALOG_DATA, useValue: {} },
            { provide: MatDialogRef, useValue: {} },
            BookService, AddressService, UserService, LoginService,
RegisterService
        ]
    });
    fixture = TestBed.createComponent(RegisterComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
});

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

it('Insert Register Component Test', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const Event = fixture.componentInstance;

```

```

    expect(Event).toBeTruthy();
  });

  it('Insert Register Form GUI count', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#registerform");
    const inputElements = formElement.querySelectorAll("input");
    expect(inputElements.length).toEqual(7);
  });

  it('Testing Register Form valid', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const Register = fixture.componentInstance;
    Register.registerform.controls?['userName'].setValue("Raja");
Register.registerform.controls?['emailId'].setValue("raja@gmail.com");
Register.registerform.controls?['mobileNumber'].setValue("1234567890");
    Register.registerform.controls?['userType'].setValue("user");
    Register.registerform.controls?['address'].setValue("Tenkasi");
    Register.registerform.controls?['password'].setValue("1212");
    expect(Register.registerform.valid).toBeTruthy();
  });

  it('Insert Register Form Button GUI count', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const formElement =
fixture.debugElement.nativeElement.querySelector("#registerform");
    const inputElements = formElement.querySelectorAll("mat-option");
    expect(inputElements.length).toEqual(0);
  });

  it('Register form valid by invalid data', () => {
    const fixture = TestBed.createComponent(RegisterComponent);
    const app = fixture.componentInstance;

```

```

    app.registerform.controls?.['userName'].setValue("");
    app.registerform.controls?.['emailId'].setValue("");
    app.registerform.controls?.['mobileNumber'].setValue("");
    app.registerform.controls?.['userType'].setValue("");
    app.registerform.controls?.['address'].setValue("");
    app.registerform.controls?.['password'].setValue("");
    expect(app.registerform.valid).toBeFalsy();
  })
});

```

### user.component.spec.ts

```

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

import { UserComponent } from '../user.component';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [UserComponent]
    });
    fixture = TestBed.createComponent(UserComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

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

```

### address.component.spec.ts

```

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

```

```
import { AddressComponent } from './address.component';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [AddressComponent]
    });
    fixture = TestBed.createComponent(AddressComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

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

### book.component.spec.ts

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

import { BookComponent } from './book.component';

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [BookComponent]
    });
    fixture = TestBed.createComponent(BookComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
});
```



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

**It ends here...Thank you..!**