

## JUNIT TEST CODE FOR LOGIN PAGE

```
package com.example.spllabportal;
import static org.junit.Assert.*;
import static org.mockito.Mockito.*;
import android.content.Intent;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;
import org.robolectric.RobolectricTestRunner;
```

```
@RunWith(RobolectricTestRunner.class)
public class MainActivityTest {
```

```
    @Mock
    FirebaseAuth mockAuth;
```

```
    @Mock
    FirebaseUser mockUser;
```

```
    MainActivity mainActivity;
```

```
    @Before
    public void setUp() {
        MockitoAnnotations.initMocks(this);
        mainActivity = new MainActivity();
    }
```

```
    @Test
```

```
public void testValidStudentEmail() {  
    assertTrue(mainActivity.isStudentEmail("geethika.ec21@bitsathy.ac.in"));  
}
```

```
@Test  
public void testInvalidEmail() {  
    assertFalse(mainActivity.isStudentEmail("randomuser@gmail.com"));  
}
```

```
@Test  
public void testValidFacultyEmail() {  
    assertTrue(mainActivity.isFacultyEmail("haripriyar@bitsathy.ac.in"));  
}
```

```
@Test  
public void testRedirectToUserMode() {  
    when(mockAuth.getCurrentUser()).thenReturn(mockUser);  
    when(mockUser.getEmail()).thenReturn("geethika.ec21@bitsathy.ac.in");  
  
    Intent expectedIntent = new Intent(mainActivity, UserModeActivity.class);  
    mainActivity.startActivity(expectedIntent);  
  
    verify(mockAuth).getCurrentUser();  
}
```

```
@Test  
public void testRedirectToAdminMode() {  
    when(mockAuth.getCurrentUser()).thenReturn(mockUser);  
    when(mockUser.getEmail()).thenReturn("haripriyar@bitsathy.ac.in");  
  
    Intent expectedIntent = new Intent(mainActivity, AdminModeActivity.class);  
    mainActivity.startActivity(expectedIntent);  
  
    verify(mockAuth).getCurrentUser();  
}
```

```

@Test
public void testInvalidEmailNoRedirect() {
    when(mockAuth.getCurrentUser()).thenReturn(mockUser);
    when(mockUser.getEmail()).thenReturn("randomuser@gmail.com");

    assertFalse(mainActivity.isStudentEmail("randomuser@gmail.com"));
    assertFalse(mainActivity.isFacultyEmail("randomuser@gmail.com"));

    // Simulate no redirection for invalid email
    Intent unexpectedIntent = new Intent(mainActivity,
RegistrationActivity.class);
    assertNotEquals(unexpectedIntent.getComponent(),
mainActivity.getComponentName());
}
}

```

## **JUNIT TEST CODE FOR PROJECT ENTRY FRAGMENT**

```

import static org.junit.Assert.*;
import static org.mockito.Mockito.*;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TableLayout;

import androidx.fragment.app.testing.FragmentScenario;
import androidx.test.core.app.ApplicationProvider;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;

```

```

import com.google.firebase.database.FirebaseDatabase;

import org.junit.Before;
import org.junit.Test;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;

public class ProjectEntryFragmentTest {

    @Mock
    private FirebaseAuth mockAuth;
    @Mock
    private FirebaseUser mockUser;
    @Mock
    private FirebaseDatabase mockDatabase;
    @Mock
    private DatabaseReference mockDatabaseReference;

    private FragmentScenario<ProjectEntryFragment> scenario;

    @Before
    public void setUp() {
        MockitoAnnotations.initMocks(this);

        when(mockAuth.getCurrentUser()).thenReturn(mockUser);
        when(mockUser.getDisplayName()).thenReturn("Test User");

        when(mockDatabase.getReference("Projects")).thenReturn(mockDatabaseReference);

        FirebaseAuth.setInstance(mockAuth);
        FirebaseDatabase.setInstance(mockDatabase);

        scenario = FragmentScenario.launchInContainer(ProjectEntryFragment.class);
    }

```

@Test

```
public void testProjectEntryFragmentInitialization() {  
    scenario.onFragment(fragment -> {  
        assertNotNull(fragment);  
        assertNotNull(fragment.getView());  
    });  
}
```

@Test

```
public void testLoadProjects_CreatesTableRows() {  
    scenario.onFragment(fragment -> {  
        TableLayout tableLayout =  
fragment.getView().findViewById(R.id.projectTable);  
        assertNotNull(tableLayout);  
        assertTrue(tableLayout.getChildCount() > 0);  
    });  
}
```

@Test

```
public void testFirebaseReferenceNotNull() {  
    scenario.onFragment(fragment -> {  
        assertNotNull(fragment.databaseReference);  
    });  
}  
}
```