Name : Asha Mol Dharmaraj

Employee id : 11997

White Box Testing (Unit testing) :

Write the Unit Test cases by using NUnit for your Back-End [CSharp File].

NUnit test for Create WebApi :

```
using CreateWebapi.Model;
using CreateWebapi.Controllers;
using CreateWebapi.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Options;
using Microsoft.AspNetCore.Mvc;
using System.ComponentModel.DataAnnotations;

namespace CreateWebapiTest
{
    public class Tests
    {
        dynamic optionsbuilder;
        AppDbContext applicationdbContext;

        [SetUp]
        public void Setup()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=Password@1234
5;database=RelevantzMovieCenter", new MySqlServerVersion(new Version())); ;
            applicationdbContext = new AppDbContext(optionsbuilder.Options);
        }
```

TestCase : 1

```
        [Test]
        public void ApplicationDbContext_should_connect_to_mysql()
        {

            bool expected = true;
            // act
            bool result = applicationdbContext.Database.CanConnect();

            // assert
            Assert.AreEqual(expected, result);

        }
```

TestCase : 2

```
        [Test]
        public void ApplicationDbContext_with_wrong_password_should_not_connect_to_mysql()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=wrongpassword;d
atabase=RelevantzMovieCenter", new MySqlServerVersion(new Version()));

            applicationdbContext = new AppDbContext(optionsbuilder.Options);

            bool expected = false;

            // act
            bool result = applicationdbContext.Database.CanConnect();

            // assert
            Assert.AreEqual(expected, result);
        }
```

TestCase : 3

```
        [Test]
        public void Post_with_correct_type_returns_correctResult()
        {
            AddMovieController addMovieController = new
AddMovieController(applicationdbContext);

            MovieDetails movieDetails = new MovieDetails()
            {
              MovieId =0,
              MovieName="Kiren1",
              MovieType="Romance",
              MovieLanguage="Tamil",
              MovieDurations="2.40 hour"
            };
            //var oldmovie = applicationdbContext.MovieDetails.Count();

            var result = addMovieController.Post(movieDetails).Result;

            //var newmovie = applicationdbContext.MovieDetails.Count();
            Assert.IsInstanceOf<OkResult>(result);
        }
```

TestCase : 4

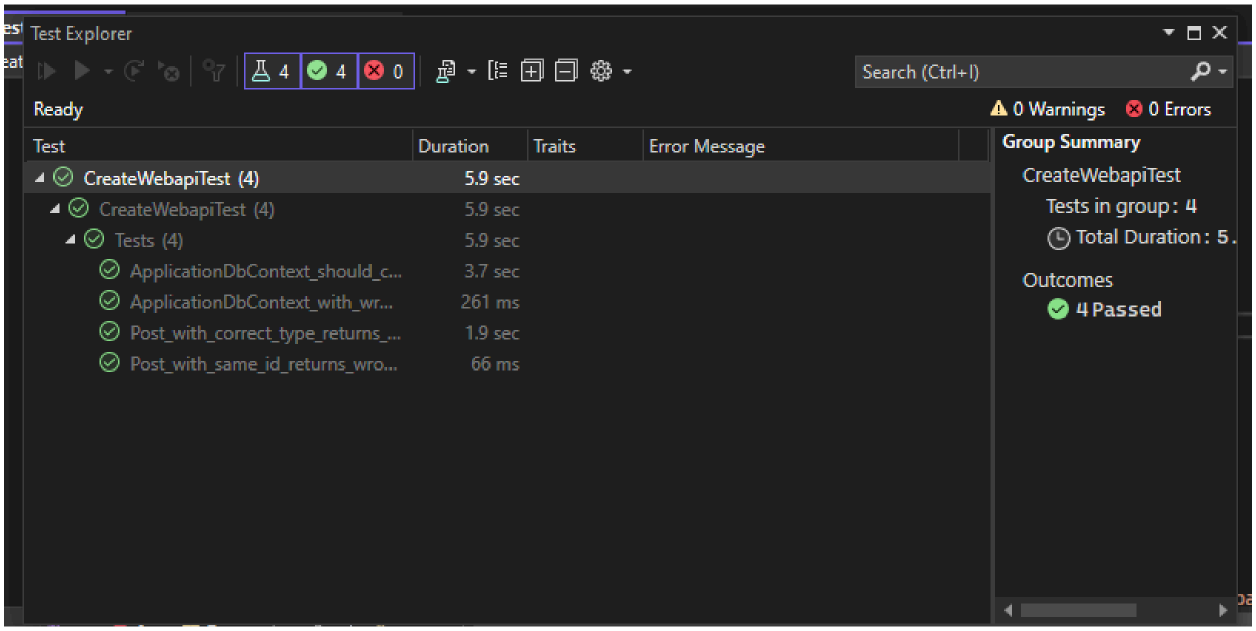
```
        [Test]
        public void Post_with_same_id_returns_wrongResult()
        {
            AddMovieController addMovieController = new
AddMovieController(applicationdbContext);

            MovieDetails movieDetails = new MovieDetails()
            {
                MovieId = 13,
                MovieName = "Kiren1",
                MovieType = "Action",
                MovieLanguage = "Tamil",
                MovieDurations = "2.40 hour"
            };
            //var oldmovie = applicationdbContext.MovieDetails.Count();

            var result = addMovieController.Post(movieDetails).Result;

            //var newmovie = applicationdbContext.MovieDetails.Count();
            Assert.IsInstanceOf<BadRequestObjectResult>(result);
        }

    }
}
```


```
Test Explorer
Ready                                    0 Warnings   0 Errors
Test                                        Duration  Traits  Error Message
CreateWebapiTest (4)                        5.9 sec
  CreateWebapiTest (4)                      5.9 sec
    Tests (4)                               5.9 sec
      ApplicationDbContext_should_c...      3.7 sec
      ApplicationDbContext_with_wr...       261 ms
      Post_with_correct_type_returns_...    1.9 sec
      Post_with_same_id_returns_wro...      66 ms

Group Summary
CreateWebapiTest
  Tests in group: 4
  Total Duration: 5.
Outcomes
  4 Passed
```

NUnit test for Fetch WebApi :

```
using FetchWebapi.Controllers;
using FetchWebapi.Data;

using Microsoft.EntityFrameworkCore;
using FetchWebapi.Model;
using System;
using Microsoft.AspNetCore.Mvc;
namespace FetchWebapiTest
{
    public class Tests

    {
        dynamic optionsbuilder;
        Appdbcontext applicationdbContext;

        [SetUp]
        public void Setup()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=Password@1234
5;database=RelevantzMovieCenter", new MySqlServerVersion(new Version()));

            applicationdbContext = new Appdbcontext(optionsbuilder.Options);
        }
```

TestCase : 1

```
   [Test]
    public void ApplicationDbContext_should_connect_to_mysql()
    {

        bool expected = true;

        // act
        bool result = applicationdbContext.Database.CanConnect();

        // assert
        Assert.AreEqual(expected, result);
    }
```

TestCase : 2

```
    [Test]
    public void ApplicationDbContext_with_wrong_password_should_not_connect_to_mysql()
    {
        optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=wrongpassword;d
atabase=RelevantzMovieCenter", new MySqlServerVersion(new Version()));

        applicationdbContext = new Appdbcontext(optionsbuilder.Options);

        bool expected = false;

        // act
        bool result = applicationdbContext.Database.CanConnect();

        // assert
        Assert.AreEqual(expected, result);
    }
```

TestCase : 3

```
    [Test]
    public void Get_by_id_returns_NotFound_for_invalid_Id()
    {
        FetchMovieController fetchMovieController = new
FetchMovieController(applicationdbContext);
```

```
          var result = fetchMovieController.GetIndividual(1).Result;

          Assert.IsInstanceOf<NotFoundResult>(result);
      }
```

TestCase : 4

```
      [Test]
      public void GetById_returns_correctResult()
      {
          FetchMovieController fetchMovieController = new
FetchMovieController(applicationdbContext);
          var result = (OkObjectResult)fetchMovieController.GetIndividual(4).Result;
          var value = (MovieDetails)result.Value;
          Assert.AreEqual(4, value.MovieId);

      }
```

TestCase : 5

```
      [Test]
      public void GetById_returns_correctResultType()
      {
          FetchMovieController fetchMovieController = new
FetchMovieController(applicationdbContext);
          var result = (OkObjectResult)fetchMovieController.GetIndividual(4).Result;
          var value = (MovieDetails)result.Value;
          Assert.IsInstanceOf<MovieDetails>(value);

      }
```

TestCase : 6

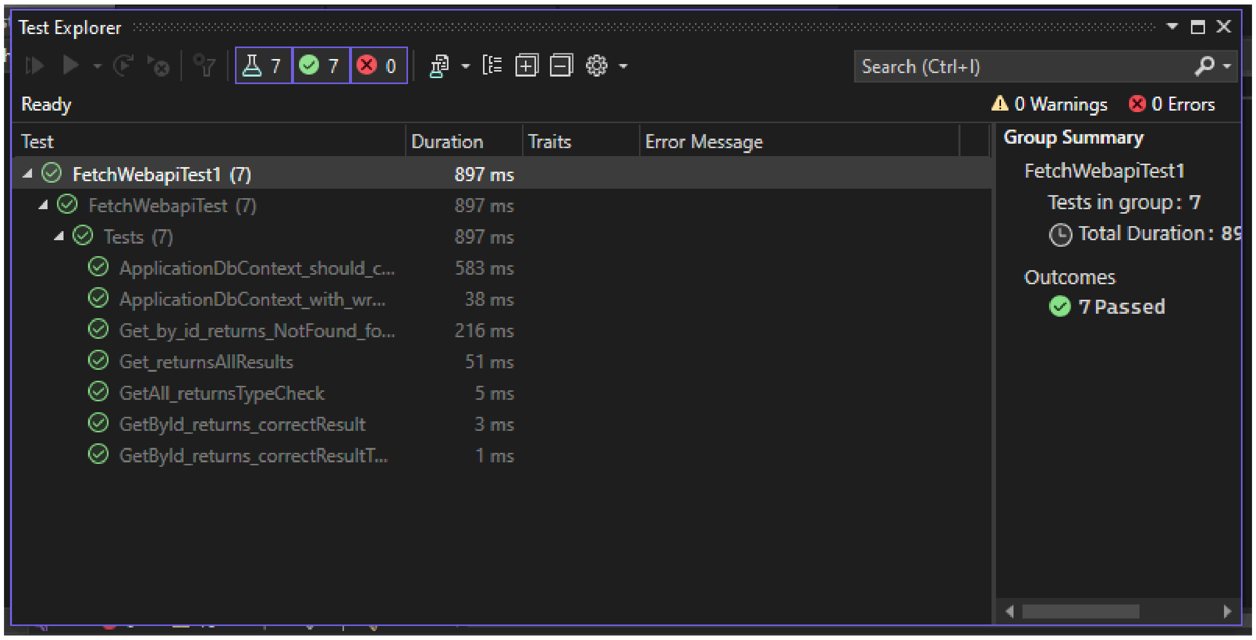
```
      [Test]
      public void Get_returnsAllResults()
      {
          FetchMovieController fetchMovieController = new
FetchMovieController(applicationdbContext);
          var result = fetchMovieController.Get().Result as List<MovieDetails>;
          List<MovieDetails> value = (List<MovieDetails>)result;
          var countresult = (OkObjectResult)fetchMovieController.TotalCount().Result;
          int countvalue = (int)countresult.Value;
          //assert
          Assert.AreEqual(countvalue, value.Count);
```

```
        }
```

TestCase : 7

```
        [Test]
        public void GetAll_returnsTypeCheck()
        {
            FetchMovieController fetchMovieController = new
FetchMovieController(applicationdbContext);
            var result = fetchMovieController.Get().Result as List<MovieDetails>;
            List<MovieDetails> value = (List<MovieDetails>)result;
            Assert.IsInstanceOf<MovieDetails>(value.First());

        }
    }
}
```



```
Ready                                                   0 Warnings   0 Errors
Test                                   Duration  Traits  Error Message     Group Summary
FetchWebapiTest1 (7)                   897 ms                              FetchWebapiTest1
  FetchWebapiTest (7)                  897 ms                                Tests in group: 7
    Tests (7)                          897 ms                                Total Duration: 89
      ApplicationDbContext_should_c...  583 ms                             Outcomes
      ApplicationDbContext_with_wr...    38 ms                               7 Passed
      Get_by_id_returns_NotFound_fo...  216 ms
      Get_returnsAllResults              51 ms
      GetAll_returnsTypeCheck             5 ms
      GetById_returns_correctResult       3 ms
      GetById_returns_correctResultT...   1 ms
```

NUnit test for Update WebApi :

```
using UpdateWebapi.Model;
using UpdateWebapi.Controllers;
using UpdateWebapi.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Options;
using Microsoft.AspNetCore.Mvc;

namespace UpdateWebapiTest
{
    public class Tests
    {
        dynamic optionsbuilder;
        AppDbContext appdbContext;

        [SetUp]
        public void Setup()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySQL("server=localhost;user=root;password=Password@123
45;database=RelevantzMovieCenter");
            appdbContext = new AppDbContext(optionsbuilder.Options);

        }
```

TestCase : 1

```
        [Test]
        public void ApplicationDbContext_should_connect_to_mysql()
        {
            var expected = true;

            // act
            var result = appdbContext.Database.CanConnect();
            Assert.That(result, Is.EqualTo(expected));
            // assert

        }
```

TestCase : 2

```
        [Test]
        public void ApplicationDbContext_with_wrong_password_should_not_connect_to_mysql()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySQL("server=localhost;user=root;password=wrongpassword;
database=RelevantzMovieCenter");

            appdbContext = new AppDbContext(optionsbuilder.Options);
            bool expected = false;

            // act
            bool result = appdbContext.Database.CanConnect();

            // assert
            Assert.That(result, Is.EqualTo(expected));

        }
```

TestCase : 3

```
        [Test]
        public void Post_with_correct_type_returns_correctResult()
        {

            UpdateMovieController updateMovieController = new
UpdateMovieController(appdbContext);

            MovieDetails movieDetails = new MovieDetails()
            {
                MovieId =13,
                MovieName = "Kiren1",
                MovieType = "Romance",
                MovieLanguage = "Tamil",
                MovieDurations = "2.40 hour"
            };
            //var oldmovie = applicationdbContext.MovieDetails.Count();

            var result = updateMovieController.UpdateMovie(movieDetails).Result;

            //var newmovie = applicationdbContext.MovieDetails.Count();
            Assert.IsInstanceOf<OkResult>(result);

        }
```

TestCase : 4

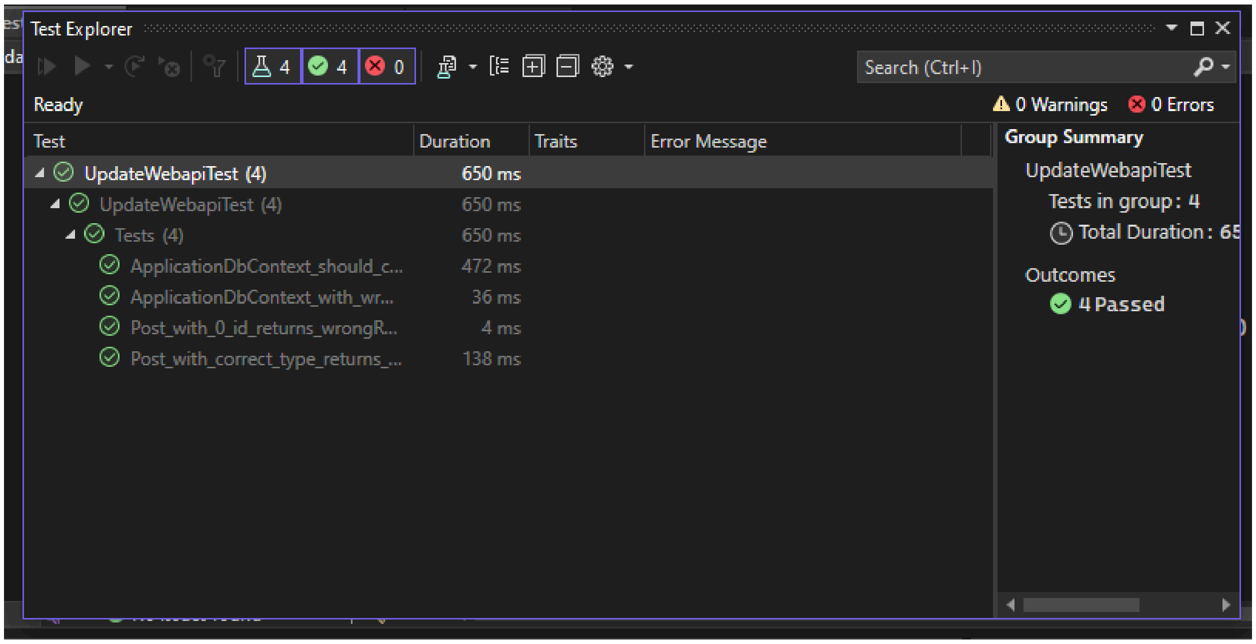
```
        [Test]
        public void Post_with_0_id_returns_wrongResult()
        {
            UpdateMovieController updateMovieController = new
UpdateMovieController(appdbContext);
```

```
            MovieDetails movieDetails = new MovieDetails()
            {
                MovieId = 0,
                MovieName = "Kiren1",
                MovieType = "Action",
                MovieLanguage = "Tamil",
                MovieDurations = "2.40 hour"
            };
            //var oldmovie = applicationdbContext.MovieDetails.Count();

            var result = updateMovieController.UpdateMovie(movieDetails).Result;

            //var newmovie = applicationdbContext.MovieDetails.Count();
            Assert.IsInstanceOf<BadRequestResult>(result);
        }

    }
}
```



```
Test Explorer
4  4  0
Search (Ctrl+I)
Ready                                                   0 Warnings   0 Errors
Test                                  Duration   Traits   Error Message        Group Summary
UpdateWebapiTest (4)                  650 ms                                   UpdateWebapiTest
  UpdateWebapiTest (4)                650 ms                                     Tests in group: 4
    Tests (4)                         650 ms                                     Total Duration: 65
      ApplicationDbContext_should_c... 472 ms
      ApplicationDbContext_with_wr...   36 ms                                  Outcomes
      Post_with_0_id_returns_wrongR...   4 ms                                    4 Passed
      Post_with_correct_type_returns_... 138 ms
```

NUnit test for Delete WebApi :

```
using System;
using DeleteWebapi.Controllers;
using DeleteWebapi.Data;
using Microsoft.EntityFrameworkCore;
using System;
using Microsoft.AspNetCore.Mvc;

namespace DeleteWebapiTest
{
    public class Tests

    {
        dynamic optionsbuilder;
        AppDbContext applicationdbContext;

        [SetUp]
        public void Setup()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=Password@1234
5;database=RelevantzMovieCenter", new MySqlServerVersion(new Version()));

            applicationdbContext = new AppDbContext(optionsbuilder.Options);
        }
```

TestCase : 1

```
        [Test]
        public void ApplicationDbContext_should_connect_to_mysql()
        {

            bool expected = true;

            // act
            bool result = applicationdbContext.Database.CanConnect();

            // assert
            Assert.AreEqual(expected, result);
        }
```

TestCase : 2

```
        [Test]
        public void ApplicationDbContext_with_wrong_password_should_not_connect_to_mysql()
        {
            optionsbuilder = new
DbContextOptionsBuilder().UseMySql("server=localhost;user=root;password=wrongpassword;d
atabase=RelevantzMovieCenter", new MySqlServerVersion(new Version()));

            applicationdbContext = new AppDbContext(optionsbuilder.Options);

            bool expected = false;

            // act
            bool result = applicationdbContext.Database.CanConnect();

            // assert
            Assert.AreEqual(expected, result);
        }
```

TestCase : 3

```
        [Test]
        public void Delete_returns_BadRequest_InvalidId()
        {
            DeleteMovieController deleteMovieController = new
DeleteMovieController(applicationdbContext);
            var result=deleteMovieController.DeleteMovie(-1).Result;
            Assert.IsInstanceOf<BadRequestResult>(result);

        }
```

TestCase : 4

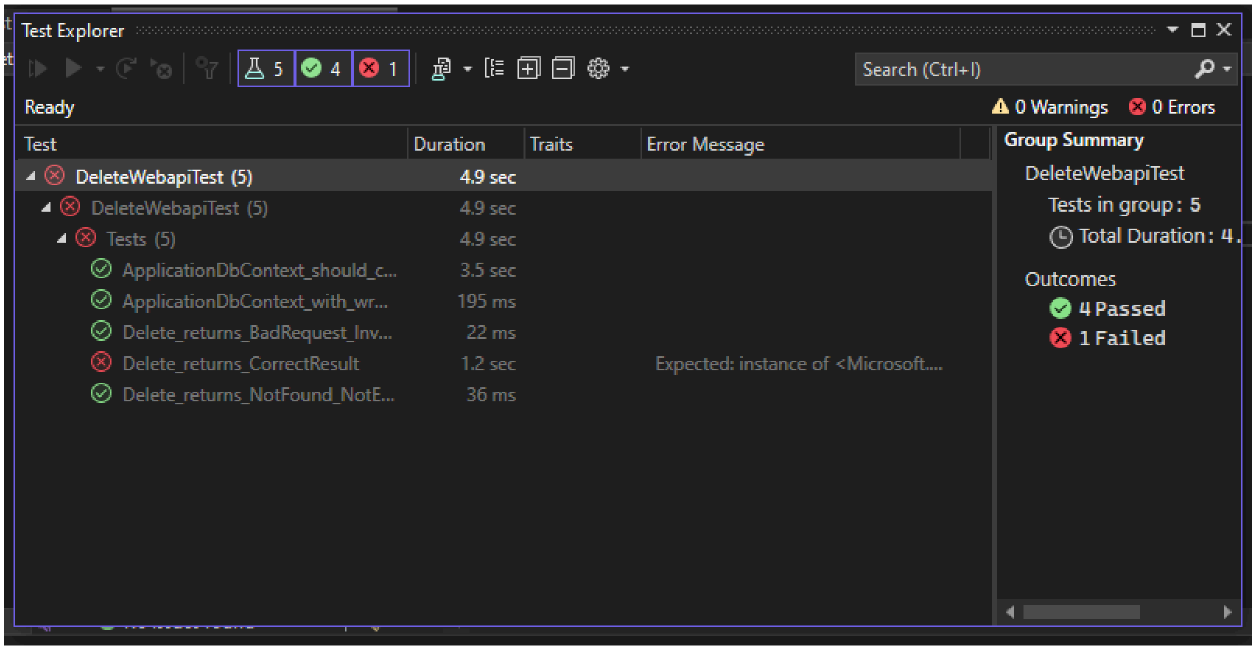
```
        [Test]
        public void Delete_returns_NotFound_NotExistsId()
        {
            DeleteMovieController deleteMovieController = new
DeleteMovieController(applicationdbContext);
            var result = deleteMovieController.DeleteMovie(1).Result;
            Assert.IsInstanceOf<NotFoundResult>(result);

        }
```

TestCase : 5

```
        [Test]
        public void Delete_returns_CorrectResult()
        {
            DeleteMovieController deleteMovieController = new
DeleteMovieController(applicationdbContext);
            var result = deleteMovieController.DeleteMovie(18).Result;
            Assert.IsInstanceOf<OkResult>(result);

        }
    }
}
```



Test Explorer

5 | 4 | 1

Ready | 0 Warnings | 0 Errors

Test	Duration	Traits	Error Message
DeleteWebapiTest (5)	4.9 sec		
DeleteWebapiTest (5)	4.9 sec		
Tests (5)	4.9 sec		
ApplicationDbContext_should_c...	3.5 sec		
ApplicationDbContext_with_wr...	195 ms		
Delete_returns_BadRequest_Inv...	22 ms		
Delete_returns_CorrectResult	1.2 sec		Expected: instance of <Microsoft....
Delete_returns_NotFound_NotE...	36 ms		

Group Summary
DeleteWebapiTest
Tests in group: 5
Total Duration: 4.

Outcomes
4 Passed
1 Failed

Write the Unit Test cases by using Jest for your Front-End [React].

Snapshot TestCase :

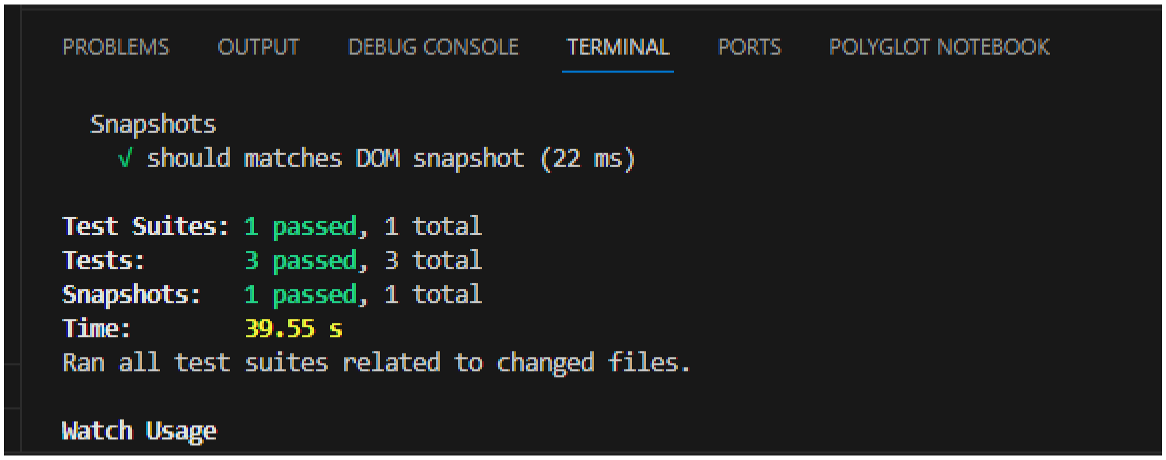
```
describe("Snapshots",()=>{
    it("should matches DOM snapshot",()=>{
      const tree = renderer.create(<App/>).toJSON();
      expect(tree).toMatchSnapshot();
    });
  });
```

TestCase to check whether the data displayed in the UI using mock data :

```
import { render, fireEvent, screen, waitFor } from
"@testing-library/react";
import renderer from 'react-test-renderer';
import axios from "axios"
import App from "./App";
jest.mock("axios")
const dummyData=[
  {
    MovieName:varathan,
    MovieDuration:2hrs
  }

]
describe("check the Ui"()=>{
  it("should display",async()=>{
    data:dummyData
  })
  render(<App/>)
  const data=await waitFor(()=>Screen.getByTestId("data"))
  expect(data).toHaveLength(1)
})
```

Final TestCase Output for Jest :



```
  Snapshots
    √ should matches DOM snapshot (22 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   1 passed, 1 total
Time:        39.55 s
Ran all test suites related to changed files.

Watch Usage
```