

# 1. .NET Core-FootBallTournament-TDD

For TDD, unit test cases mostly will be written by developers them self. And in agile sense there is no separate testing team. All are part of one development team. May be something like this will help. "Product team has decided to add a new feature to football tournament application. The unit test cases have already been designed for you and available in **FootBallTournament.Tests/NewFeatureTests.cs**. Your task for this challenge is to use TDD approach to build the new functionality in the application. Please ensure the existing functionality is not impacted by running regression tests.

Add a feature where users can sign up as a viewer and make bookings for matches. The matches are available in the table Mappings.

Implement the following APIs:

**/viewer/login** : user should be able to login with username(name) and password(password). The API returns either JWT token on successful login or error message on invalid credentials.

**/viewer/register** : user should be able to register as a new viewer/audience with username(name) and password(password).

**/viewer/viewmatches** : This is an authenticated endpoint. Once the user logs in as a Viewer, he/she can fetch the list of matches from Mappings table.

**/viewer/book/{matchId}** : This is an authenticated endpoint. Once the user logs in as a Viewer, he/she can make a booking for a match provided by path vairable.

**/viewer/bookings** : This is an authenticated endpoint. Once the user logs in as a Viewer, he/she can fetch all the bookings made by the user.

Use the existing code and add the new feature.

- FootBallTournamentTests.cs is the testcase file for the existing code
- NewFeatureTests.cs is the testcase file for the new features to be added

In the process of adding newfeature existing code should not be disturbed, all the test cases should be passed

Steps to do Handson

1. Install the dependencies by clicking on the **project -> install** button
2. Run the application either using **dotnet run** command or click on **project->run** button. Do not make any changes to the application
3. To test the application please click on **RunTests** button and user can see the number of testcases passing
4. Evreytime user wants to test, user should restart the application by clicking on **run** button

## Software Instructions

This question requires **.NET Core 3.0**. If you don't have these installed on your local environment already, follow the instructions below to install them before you being the test.

- [Install .NET Core 3](#)

## Git Instructions

Use the following commands to work with this project

## Run

```
cd FootballTournament && fuser -k 8001/tcp && dotnet run
```

## Test

```
cd FootballTournament.Tests && rm -rf reports && dotnet build && dotnet test
```

## Install

```
cd FootballTournament && dotnet build && cd ../FootballTournament.Tests && dotnet build
```