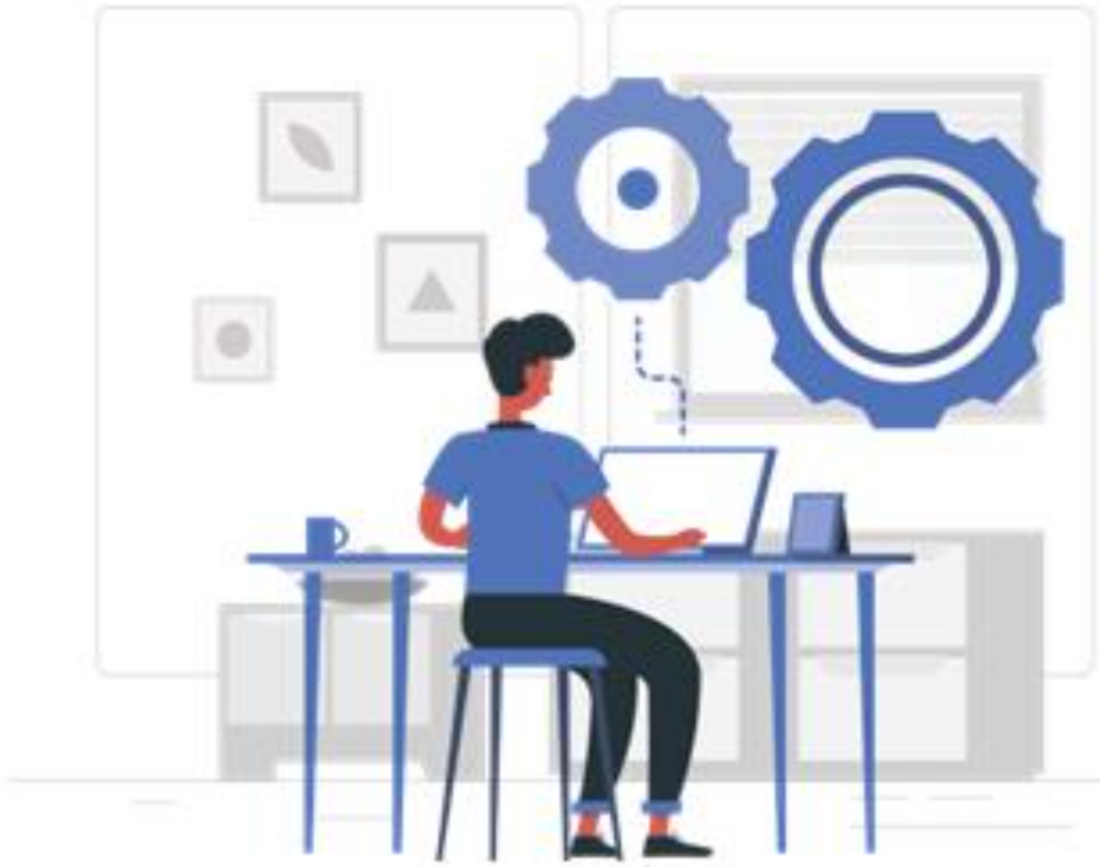

Practice

Implement Unit Testing For Angular Components and Services

Practice Exercises

- Enable Testing for Box Office App

Points to Remember

- The default settings provided in the configuration files – karma.conf.js, angular.json, test.ts for execution of test code should not be modified.
- The Angular CLI command "ng test" should be used to execute the test code.
- Write and execute one test case at a time.
    - Proceed with the next case only if the currently written test code succeeds.
- In addition to the specified test requirements, if required, more test cases can be added, but it should be ensured that they pass before the solution is submitted.

Instructions for the Practice

- Click here for the boilerplate.
- Please read the README.md file provided in the boilerplate for further instructions about the practice exercise.
- Fork the boilerplate into your own workspace.
- Clone the boilerplate into your local system.
- Open the folder containing the boilerplate code in VS Code.
- Provide the test specifications in the files specified with the task details.
- Open the command terminal and run the command **ng test** to run the test code.
- Submit the files (excluding the `node-modules` folder) for manual evaluation.

Context

Box Office is a popular and trusted source of movies and television shows. One can find ratings and reviews for the latest movies and television shows.

The Box office SPA created using Angular CLI attractively displays a list of trending movies to the end users. Movie information like movie's name, poster, genre, and ratings are visible to the user. Users can also search for a movie from the list of available titles.

The components created in the Box Office App should be tested to ensure a quality product.

PRACTICE

Enable Testing for Box Office App

Create test cases to enable testing for the solution of Box Office App.

The test code should ensure:

1. The expected title is available in the Header component
2. The Dashboard component displays movies.
3. Searching for movies returns the expected result from the Dashboard component.

Note: Steps to complete the task are given in the upcoming slide.

Tasks

Following are the steps to complete this exercise:

- Step 1: Copy the solution of Box office app created inside **p3-search-movies** of **fe-c4-s2-components-practice of sprint – Develop SPA Using Angular Components into the root folder.**
- Step 2: Open the **app.component.spec.ts** and refactor the generated test code to test whether the component is instantiated successfully
    - the title `Trending Movies` is displayed as expected.
- Step 3: Refactor **app.component.ts** code to achieve the expected result if the test cases are failing.
- Step 4: Open the **header.component.spec.ts** and refactor the generated test code to test whether
    - the component is instantiated successfully
    - the title `Box Office` is displayed as expected.
- Step 5: Refactor **header.component.ts** code to achieve the expected result if the test cases are failing.

Tasks (Cont’d.)

- Step 6: Open the **dashboard.component.spec.ts** and refactor the generated test code to test various functionality as given in the upcoming slide.
- Step 7: Refactor **dashboard.component.ts** code to achieve the expected result if the test cases are failing.

Dashboard Component Testing Requirements

Functionality	Test Condition
Display Movies	Should display 6 movie card details
	Should contain a movie with name "jurassic world" and its rating
Search Movies	Should contain a search text box
	Should contain 2 buttons: search and reset button
	Should contain a Search button with "GO" as its caption
	Should contain a Cancel button with "Clear" as its caption
	Should load the movie when searched by its full name
	Should load all the movies when no text is entered in the search box
	Should load all the matching movies when searched by starting letter(s)/partial text
	Should display no movie when starting letter(s)/ partial text entered has no matching movies

Expected Output