SOFTWARE ENGINEERING 2
PREFINAL EXAMINATION

UNIT TESTING AND TEST DOCUMENTATION (100 Points)

Scenario: "A Campus Event Tracker (Angular) and a Student Grading System (C# Console Application) were previously developed and are fully functional. However, both systems lack automated unit testing and must now be tested to ensure software quality and reliability."

Task:
You will work individually to complete three (3) parts of the examination:
1. **Frontend Unit Testing**
   – Create unit tests using Jasmine and Karma for the existing functionalities of the application.
2. **Backend Unit Testing**
3. – Create unit tests using MSTest for the existing functionalities of the application.
4. **Test Documentation**
   – Document and provide a brief explanation of all test methods created for both systems.

Grading Criteria:

- Angular Unit Testing (30)
- C# Unit Testing (30)
- Code Coverage (10)
- Test Documentation (30)

Exam Parts:

Part 1: **FRONTEND UNIT TESTING - Campus Event Tracker**
Create unit tests for Campus Event Tracker application.
Document the code coverage results and successful test executions.
The file name must be: UI-CodeCoverage-FullName.pdf

Functionalities to Test:
Create unit tests for:
1. Add Event
2. Display All Events

Testing Requirements:
You must:
- Use Jasmine and Karma
- Follow the Arrange – Act – Assert pattern
- Test the component logic

- Ensure no failing test cases and has 80% code coverage.

Example Test Scenarios:
- Should add event successfully
- Should not add event if title is empty
- Should display all added events

Part 2: **BACKEND UNIT TESTING - Student Grading System**
Create unit tests for the Student Grading System.
Document the code coverage results and successful test executions.
The file name must be: BackEnd-CodeCoverage-FullName.pdf

Functionalities to Test:
Create unit tests for:
1. Add Student
2. Add Subject Grade to Student
3. View Student Report
4. View All Students

Testing Requirements:
You must:
- Use **MSTest**
- Follow the Arrange – Act – Assert pattern
- Test functionalities
- Ensure no failing test cases and has 80% code coverage.

Example Test Scenarios:
- Should add student successfully
- Should not add student with empty name
- Should correctly add subject grade
- Should return all students

Part 3: **TEST CASE DOCUMENTATION**
Document all test methods you created.
Write a short explanation for each test case.

Required Format with Example
Use the table format below:

For Angular Unit Testing:

Test Method Name	Scenario	Expected Result
should create		
should add events	Adds a valid event with title and date	Event is added to event list

should delete existing events	Deleting events through deleteEvent methods in eventService	Event with selected id should deleted from event list
Should return all events	Returning all events through getEvents method	Return events and their details
Should return total number of events	Returning the total numbers of events via getTotalEvents() method	Returning the total number of events
Should update existing events	Mock new event data and pass the new data with the selected event id in updateEvent() method	Event must be updated with a new data encoded.

Code Coverage (Screenshot)