**Backend Tasks Project Requirements:**

*Task 1*

* Develop an application that reads the file “Input.txt”. this class. Loop through each object of “MovieStar” and display the properties to the console. Calculate the age of each actor as shown in the example picture bellow. Use “Newtonsoft.Json” framework.

Graphical user interface, text

Description automatically generated

*Task 2*

Create a console application that would calculate and output the net salary given the gross value as input. The taxation rules in the country of Imaginaria as of date are as follows:

1.) There is no taxation for any amount lower or equal to 1000 Imagiaria Dolars (IDR).

2.) Income tax of 10% is incurred to the excess (amount above 1000).

3.) Social contributions of 15% are expected to be made as well. As for the previous case, the taxable income is whatever is above 1000 IDR but social contributions never apply to amounts higher than 3000.

Example 1 : George has a salary of 980 IDR. He would pay no taxes since this is below the taxation threshold and his net income is also 980.

Example 2 : Irina has a salary of 3400 IDR. She owns income tax : 10% out of 2400 => 240. Her Social contributions are 15% out of 2000 => 300. In total her tax is 540 and she gets to bring home 2860 IDR

In order task 1

and task 2 to be accepted you should fulfill the following requirements:

**In order task 1 and task 2 to be accepted you should fulfill the following requirements:**

1. Get the exam project
2. You should finish exam project as follow:
   * Create menu from which user should be able to choose from task 1, task2 or Exit

Text

Description automatically generated

* + You should use folder structure in the project to place correct files in them.
  + You should use ***Dependency Injection*** (DI)

(in the project is installed Autofac, but you can use another one if you feel more comfortable with it).

* + You should use logger to log errors in appropriate places in the code (in the project is installed Log4Net, but you can use another one if you feel more comfortable with it).
  + You should use error handling in appropriate places in the code.
  + You should try to name appropriately all classes/methods.
  + Try to follow all clean code practices.

1. You should consider your implementation to allow easily adding of new taxes.
2. After completing the task, please leave some comments explaining how new taxes could be added to the system.

**Frontend Tasks Project Requirements:**

Use vanilla Javascript

Your task is to read the xml file (Study.xml) located at https://test.ce2s.net/Study.xml, create function to generate table (similar to the picture below) and **get the values using XPath**. In addition, you have to export the table to CSV format. We are giving you the JS script for export.

Feel free to style a little the table (not necessary)

Fields for the table:

Study.xml

<alias> -- <field>

sys\_System -- name

sm\_Study -- name, state

sm\_Task -- keyed\_name

ar\_SimulationResult -- ar\_resultname, ar\_resultvalue, ar\_resultunit, ar\_target\_met

re\_Requirement -- ar\_conditionexp

The result should be a script that reads XML files and generates table that can be exported in CSV from button.

Graphical user interface

Description automatically generated with medium confidence