Overview

Read the complete assignment before you start. Understand clearly what is required so that your work will be appropriate and easier. When you do that let us know how many days you are going to need to complete assignment.

Preconditions

- You should work on your local machine.
- You may use any IDE or editor for developing the application.
- You may use .Net Framework or .Net Core (Core will be considered as a plus).
- You may use MySQL or MS SQL as a database.
- You should strive to use as few as possible third-party libraries
- Keep in mind we want to see a good implementation of the S.O.L.I.D principles.
- Try to make clean, reusable and extendable code that is easy to support and understand.
- Try and cover the most critical areas with unit or integration tests.

Requirements:

Objective

Create a simple REST API to manage traders.

Functional Specifications

Create a simple REST API that can be used to create, list, update and delete traders. It should be possible to replace the DB engine without any controller changes. Apply input validations and constraints wherever necessary to create a stable application.

- Create/update Trader (all fields are required)
    - ✓ First Name - text
    - ✓ Last Name - text
    - ✓ Birthdate – date (ISO format)
    - ✓ Cryptocurrencies – list
        - ➢ Currency - text
        - ➢ Symbol - text
    - ✓ Avatar - image
    - ✓ Login
        - ➢ Email - text
        - ➢ Password - text
- Delete Trader - It should be soft delete
- List Traders – following columns should be displayed
    - ✓ Name (First Name and Last Name)
    - ✓ Age
    - ✓ Email

  Implement pagination and sorting for Age and Email columns.

Deliverables

Readme

Create a txt (or markdown formatted) file with the following information:

- Steps to create and initialize the database.

- Steps to prepare the source code to build and run properly.

Design diagrams

Create a doc file containing the following information and diagrams:

- List of technologies and design patterns used.
- An overall activity or sequence diagram.
- An overall layer or component interaction diagram.

To be evaluated

- The quality of the output (functionality).
- Code quality and completeness.
- Technologies applied.
- Extra validations and assumptions which are not described.
- Add missing requirements to the implementation, according to your experience.

Delivery

Delivery for this assignment should consist of an archive named **<your_name>-.NET.zip** containing the following:

- Source code
- Readme.txt (or markdown formatted) file containing the instructions to configure and run the application
- Design.doc with needed diagrams
- Any feedback you may wish to give about improving the assignment. You may use Readme file for that purpose.