Project rules

You're expected to connect to the CryptoCompare API through a program mainly written in C# .Net (or .Net Core) over the course of the semester, and display some information that you find useful for some crypto-currencies.

The CryptoCompare API is very complete: it has many endpoints, clear documentation, and easy to use/test API calls. Yet, as any big project depending on an external website, please note that it can go under maintenance, crash, or even be unavailable at times. So, starting to work on this project asap really is a good idea.

This project can be done alone, or in groups up to four. All will be marked in a *somewhat* similar fashion.

The exam itself will be done in reserved slots, at school, with me. You'll have to explain your workflow/architecture, make a small demo of your final project, and I'll look at your code. The "wow-effect" is a part of the grade. Impress me!

I can mark your project at any point during this semester, upon request. If you want to be marked before the exam date, I'll give you bonus points, and this will be done through teams (online). Once you're marked, you don't need to do any more TPs. Please note that I'll only mark you once, so don't rush your project either.

Only one computer per group will be allowed. You'll have to clean and rebuild your solution from the IDE and start the demo directly. You can bring extra material to help you with the demo (PowerPoint, notes on paper, UML diagram and so on), but this is not a requirement.

Marks will be given by tier. You'll have to complete every tier requirement to "unlock" the next tier. Advanced features are given as an example. Any feature that you'll implement on your own will award bonus points. New features can be done and will be marked regardless of tier. I will also mark the explanation you'll give me on how your code works and is structured.

1. Basic tier:
   a) The code compiles and the program runs in Release mode.
   b) A connection is done to the API and the code accepts parameters.
   c) You'll have made some calculation with the data on your side.
   d) Interesting data display is done through a graphical interface of your choice (local website, phone app, Xaml or WPF interface, or any interface you desire).

2. Expected tier:
   a) The code is clean and well organized, it compiles without warnings and the program runs in Release mode.
   b) Connections are done to the API on multiple endpoints and data is refreshed on a regular basis. The code accepts dynamic parameters given by the user.
   c) The calculations done are complex and use data from multiple endpoints or a lot of data (i.e.: all points from the last 3 months).
   d) Interesting data display is done through a graphical interface of your choice (local website, phone app, Xaml or WPF interface, or any interface you desire). Multiple graphs show data on multiple timeframes. This graphical interface should accept parameters given by a user.

3. Advance features (examples):
   a) Data can be exported to a *.csv or *.xlsx file.
   b) Your program can notify you when a special event is triggered (i.e.: price drop of some specific currencies).
   c) Your program gives insight on whether you should buy or sell some cryptos (you can find risk related calculations online).
   d) You display how much correlation you have between two given cryptos (i.e.: how much are $ETH and $SOL value correlated).
   e) You used a versioning tool during development and have multiple, useful and meaningful commits.
   f) You have MEANINGFUL unit tests that use the AAA pattern and that passes.
   g) You used interesting design patterns to improve scalability/maintainability.
   h) You have some debug specific logs that are not shown in Release mode.
   i) You used Dependency Injection and SOLID Principles

Once again, marks will be given for meaningful projects that have an added value. Just displaying data won't be enough. The "wow-effect" of the demo is crucial. Clean, well-organized code, meaningful classes, and functions names. Scalability, testability, and robustness are also key elements I'll look at during your presentation.