Digital Cloud Technologies

.NET Developer Test Assignment

It is necessary to implement a program that will display various information related to cryptocurrencies.

To implement the task, you **must** use the **C# language, the .NET platform, and UWP or WPF technologies**. Only «core» libraries are allowed, such as Newtonsoft.JSON. All functionality of the program **must** be written by the candidate.

It implies the use of one (or several) of the open APIs:

* CryptingUp: <https://www.cryptingup.com/apidoc/>
* CoinCap: <https://docs.coincap.io/>
* CoinGecko: <https://www.coingecko.com/en/api/documentation>

The application must support the following functionality:

* It must be a multi-page application with navigation.
* The main page displays the top N currencies by popularity on some market (or top 10 currencies that were returned by API).
* Page with the ability to view detailed information about the currency: price, volume, price change, in which markets it can be purchased and at what price (the ability to go to the currency page on the market **is a plus**).
* Possibility of searching for currency by name or code.
* Usage of MVVM (frameworks are also welcome).

**Any** application UI design is allowed, but it must be aesthetic.

Appropriate use of patterns **is encouraged**.

The architecture and application code should be **as clean as possible**.

**It will be a plus** if the program implements additional functions (the more, the better):

* Displaying quote charts for currencies (Japanese candlestick chart or some other).
* Page in which you can convert one currency to another (we neglect the method and possible commission).
* Light / dark theme support.
* Support for multiple localizations.

It is recommended that you initially determine the amount of work that will be performed and select a “key” API for project, due to the fact that not all APIs support functionality for all tasks. But, combining the functionality of several APIs is welcome.

The source code of the program **must** be stored in any public Git version control system (Github, Bitbucket, Gitlab, ...). Having a commit history **is a** **must**, the use of branching is optional.

The result of the work done is a **link to the repository** with a working project. The presence of the README.md file in the root of the repository with a list of implemented functionality is **not necessary**, but **is a plus**.