# ASSIGNMENT COVERSHEET

Prague College

| | |
|---|---|
| Student Name: Oleksandr Chako | |
| Class: Programming Fundamentals B | |
| Assignment: ICA Project Part 2 | |
| Lecturer: Mohamed Bettaz | Semester: 2204 |
| Due Date: 2023-01-25 | Actual Submission Date: 2023-01-13 |

| **Evidence Produced (List separate items)** | **Location (Choose one)** | |
|---|---|---|
| | X | Uploaded to the Learning Center (Moodle) |
| | | Submitted to reception |
| *Note: Email submissions to the lecturer are not valid.* | | |

| **Student Declaration:** | |
|---|---|
| **I declare that the work contained in this assignment was researched and prepared by me, except where acknowledgement of sources is made. I understand that the college can and will test any work submitted by me for plagiarism.**<br>**Note:** The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement | |
| Date: January 13, 2023 | Student Signature: Oleksandr Chako |

A separate feedback sheet will be returned to you after your work has been graded.
Refer to your Student Manual for the Appeals Procedure if you have concerns about the grading decision.

| **Student Comment (Optional)** |
|---|
| Was the task clear? If not, how could it be improved? |
| Was there sufficient time to complete the task? If not, how much time should be allowed? |
| Did you need additional assistance with the assignment? |
| Was the lecturer able to help you? |
| Were there sufficient resources available? |
| How could the assignment be improved? |
| *For further comments, please use the reverse of this page.* |

# Technical Report

Oleksandr Chako

Prague City University

Course: Programming Fundamentals

Lecturer: Mohamed Bettaz

Semester: 2204

# Table of Contents

# 1 Introduction

For the ICA part 2, I decided to make an application for screening and analyzing financial data of the currency exchange market. From my perspective one of the most demanded areas of data science - is Financial. The finance industry generates lots of data. Big data in finance refers to the petabytes of structured and unstructured data that can be used for the creation of strategies for business institutions.

In this report, we look at the design, data, and deployment process. Implementation and data cleaning process in the details you can find in the Jupyter notebook file, attached to ICA part 2

# 2 Design

There are a lot of different services for market price screening, with different designs and different sets of features. Here are some examples of real screening services:

- Finviz
- Yahoo finance
- Investing

No difference in how complicated the screener is, there are always some base components, like charts, indicators, and control panels to change data and view.

Since it's only a demo project, we will create something not complicated but with all the main features needed for a real price screening application:

- Price chart with different display options
- Indicator chart with different display options
- Ability to change price chart figure. The default figures are: candlesticks, bars, and line
- Dark and light theme for the chart.
- Slider with the ability to change the date range
- Ability to select different currency pairs
- Ability to change the timeframe for selected currency pair
- Possibility to retrieve data by year from a data frame

Also, in this project, I'll use some principles of technical analysis of the price charts and some special terminology. Anyway, I'll try to describe everything in simple language for a non-technical person. But, if you don't have any understanding of what price charts in the market are, you would find all the basic information in this article: Understanding Basic Candlestick Charts

According to the requirements our design is:


# 3 The Data

The most important thing is to find proper data. I decided to take real financial data from BITSTAMP EXCHANGE DATA. From a demo perspective, I took only a few currency pairs: BTCEUR, BTCUSD, ETHBTC, and ETHEUR. Each coin pair is represented by two data frames: a day timeframe, and an hour timeframe. In total, we have 8 data frames.

The data frame looks like this:

```
https://www.CryptoDataDownload.com
unix,date,symbol,open,high,low,close,Volume BTC,Volume EUR
1672790400,2023-01-04 00:00:00,BTC/EUR,15798,15802,15787,15788,1.65765029,26170.98277852
1672786800,2023-01-03 23:00:00,BTC/EUR,15783,15806,15783,15801,4.79777073,75809.57530473
1672783200,2023-01-03 22:00:00,BTC/EUR,15786,15789,15772,15783,3.24313136,51186.34225488
1672779600,2023-01-03 21:00:00,BTC/EUR,15775,15791,15770,15786,9.63655850,152122.712481
1672776000,2023-01-03 20:00:00,BTC/EUR,15743,15781,15743,15778,7.54115443,118984.33459654
1672772400,2023-01-03 19:00:00,BTC/EUR,15758,15758,15735,15745,5.13883025,80910.88228625
1672768800,2023-01-03 18:00:00,BTC/EUR,15759,15772,15749,15758,7.16390344,112888.79040752
1672765200,2023-01-03 17:00:00,BTC/EUR,15736,15759,15730,15759,4.82580493,76049.85989187
```

Looks like, we have a set of different prices and dates. We need to understand that every second and millisecond there is appeared a hundred transactions in the real market, with every currency

pair. It's impossible to show all of the price movement in the chart, but we don't need it, it's quite enough to show the result of these transactions in the selected time frame. For example, let's get the first row in our data frame. We are interested in the open and close values. Here we can see that the open is 15798 and the close - is 15788, the close is lower than the open which means that we have a downtrend and the candle is red. The high of the candle is the difference between opening and closing. In other words, we can say if the difference is positive - it's a green candle, up trend, if negative - a red candle, downtrend.

An example of prices is represented by candles:


The thin lines above and under the candle body it's a range of prices in the different timeframes, in other words, the lowest and highest prices in the current timeframe. It's represented by high and low values in the data frame.

Another important value is date, we'll use it to manage the order of candles in the chart.

Other columns like unix, Volume BTC, Volume EUR are not necessary for the chart. But in more detail all data cleaning processes are described in the Jupyter notebook file, attached to this report.

# 4 Deployments

The submission of ICA2 required next steps:

- Python file containing Dash application
- Jupyter notebook containing a description of data cleaning and code methods
- Technical report - current document
- Link to the deployed application

In addition to mentioned attachments, I've prepared a complete python application in a module syntax according to PEP 8 standards. This application is stored in GitHub with data and Jupiter

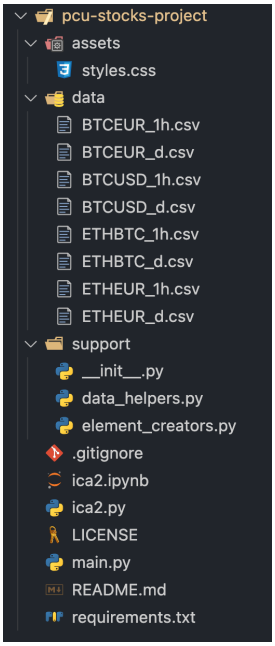
notebook. It's a complete repository with all necessary deliverables, also in the repository, you can find an exhaustive README.md file with instructions on how to set up and launch applications locally

## 4.1 Python code

The attached python file is a self-sufficient application, you just need to run it in a Python environment with all necessary libraries. In this file, we have methods to clean data first and then application methods. Also, I used inline CSS styles just in the HTML creation methods. An alternative approach is to use a module structure repository for application, I'll describe it below. Also so far I can't use data frames as additional files, I use the method to download it from the GitHub repository where I published it.

The mentioned repository with Python module used a Python local environment. It means that you don't need to install libraries on your local machine, instead, you need to create a local environment and install all libraries from the requirements.txt file from the repository. Exhaustive instructions on how to do it you can find in README.md. Also, instead, to keep all methods in one python script file, I used a module structure to separate code by a different logic. The main benefit of this approach is to make our application more extendable, than the "one file" approach. If we want to add new features in the future, it'll be much easy to do in the module-separated application than in the "one-file" application.

The architecture of the module:




Since we have a repository, we can keep our data in it, instead to download it by link, this approach is more quickly and more reliable. But in another hand, we lose the ability to download

different files dynamically. This approach is used only as an example prospect.

Another benefit of the module approach is that we can use CSS styles in the additional .css file, which makes our python code clean

### 4.2 Jupyter notebook

Jupyter notebook represented by **ica2.ipynb** file consists of two parts: **Data cleaning** and **Coding**. The data cleaning part described which problem we have with existing data, and methods that we can use to make proper cleaning and transformation. The coding part described different methods for creating HTML tags and filtering data methods

### 4.3 Links

- `https://python-ica2.onrender.com` - Link to published application. I've used render.com service. Due to the fact I used a free plan to deploy, the first navigation to this link make take a while
- `https://github.com/achako2012/pcu-stocks-project` - Link to my public repository with the application. Also here you can find: **ica2.py** - file attached to moodle as a python file with the entire application, **ica2.ipynb** - submitted in moodle Jupyter notebook

## 5 Tools

List and description of used libraries in the project:

- Gunicorn - Used for deploying applications into render.com service
- Dash - one of the main libraries used to display data, such as charts, HTML tags, etc. Also used to manage HTML elements events and make a logic layer
- Pandas - one of the main libraries to retrieve and clean data

List and description of other used tools:

- GitHub - cloud-based service for storing application
- render.com - service for deploying application
- Visual Studio Code - main IDE I used for developing my projects