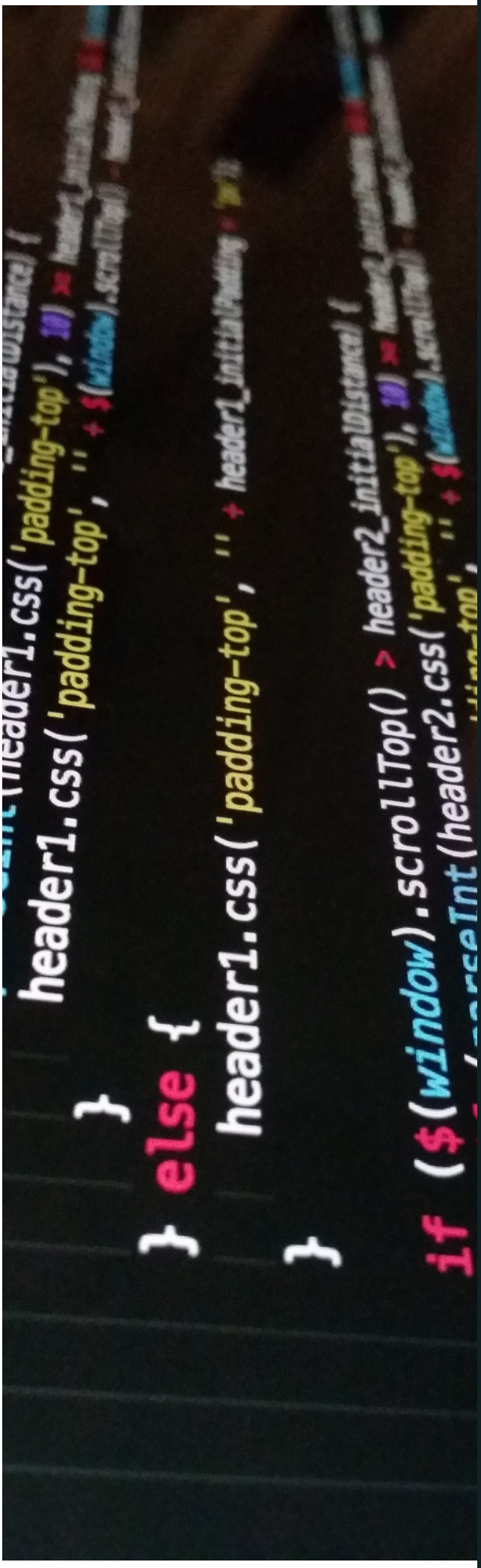


Python/SQL

Project **Report**

Prepared By :

Karolina Solarska
Zuzanna Kostecka

Title:

Stock Management Tool

2023
Faculty of Economic's Sciences,
Warsaw University

Motivation

In the conditions of international uncertainty about upcoming financial, energy and environmental crises it is easy to make impulsive investment decisions. Under the panic, investors are less resilient to market pressure, which can lead to significant losses. We decided to focus on the automotive industry, which is influenced either by the fuels crisis, energy transformation, environmental motives and semiconductor market shortage.[1]

Due to the unrest in the markets caused by various factors, we created a tool that, based on a formula, will help to decide whether a stock is best to 'buy', 'sell' or 'hold'. The user will be able to compare its profitability of relevant market positions and make a more prudent decision.

Tool limitations

The model is irrespective for the market conditions. It doesn't accounts the company's policy, investments decisions and geopolitical conditions. It brings only arbitral output of fair value for the stock. Another limitation is that model assumes the company pays dividends, thus the user should use it only for companies with dividends.

The tool is not based on webscrapping, neither connected to any just in time data, which make it static and require from the user to input the data every time they need to check the recommendation.

Recommendations

We can distinguish 3 approaches among the investors: aggressive, conservative and neutral. Users are able to interpret recommendations for all three approaches. To determine the recommendation for aggressive approach, the Fair Value price need to be at least 5% higher than Market Price for "buy" or 5% lower for "sell", otherwise, recommendation will remain "hold". Similarly for conservative and neutral, with differences correspondingly 20% and 10%.

Methodology

User's input

Dividend
Company's last paid dividend, accessible at company's financial reports.

Hurdle Rate
The reference interest rate from Central Bank + 15%. However, hurdle rate can be higher for some companies with relatively high dividend payout ratio.

Dividend Date
Available with value of last paid dividend.

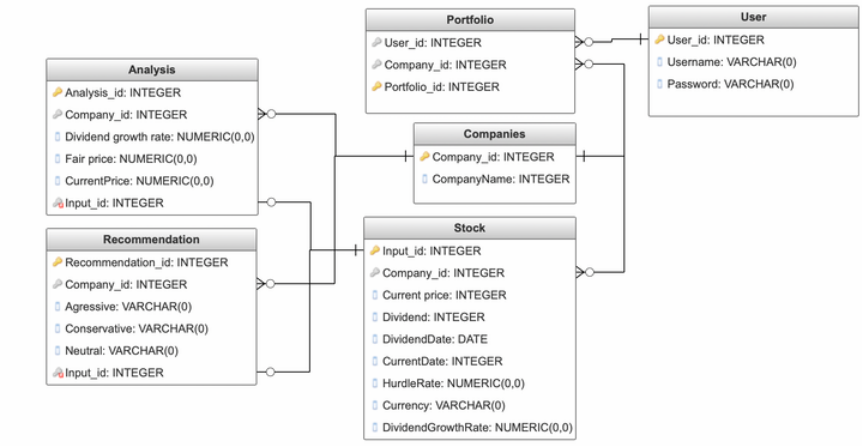
Company
The user is choosing the company, which stock they want to buy.

Dividend Growth Rate
Expressed as "g", the financial data provided by most of stock exchange portals. The data are provided i.a. on the gurufocus[3] website.

Current Price
The price of stock will be crucial in deicision, whether is "buy", "hold" or "sell", comparing to the Fair Value form the model. It can be substracted from xtb brokerage house.

SQL database

Database below is created for user, which is interested in full overview of own and other investors object of potential investment. They are able to provide data of potential company (Stock) and obtain FairPrice (Analysis Table) for further recommendation, depending on their approach - Aggresive, Conservative or Neutral (Recommendation table). Their personal data - Username, User_id and Password are stored in User Table, data about Company are in Companies Table, Table Portfolio is for integrating this two groups. Data are connected between tables using Foreign keys: Portfolio Table - User_id to User Table, Company_id to Companies Table, Analysis Table: Company_id to Companies Table, Input_id to Stock Table, Recommendation table: Company_id to Companies Table, Input_id to Stock Table, Stock Table - Company_id to Companies Table. Intuitively, corresponding Tables contains Primary Keys, which are both Unique and not null.



Each time the user is creating the input for the new calculation, the data is saved to tables: Analysis, Companies and Stock.

Gordon Growth Model [2]

$$P_0 = \frac{D_0(1+g)}{(1+r)-(1+g)}$$

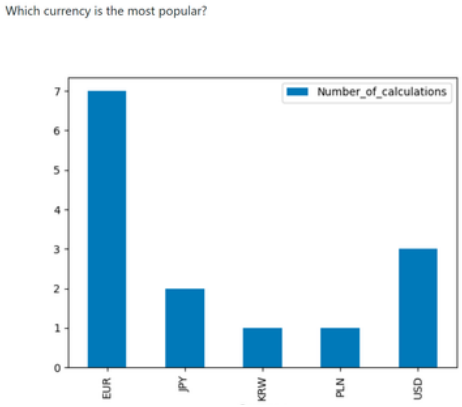
The Fair Value is calculated from the Gordon Growth Model , using the user's input data: dividend, dividend growth rate and hurdle rate. The calculations are provided using following code:

```
fairprice = round((float(dividend) * (1 + float(growth_rate))) / (float(discount_rate) -
                                  float(growth_rate)),4)
```

Which currency is the most popular? - Graph.

```
graph_data = pd.read_sql_query('SELECT Currency, count(*) as Number_of_calculations FROM
             Analysis where currency is not null group by currency', conn)
```

We have created the graph, which shows number of users' analysis using defined currency. Each new input with specific currency is recorded in the database and influence the graph. The intuition of the query is as follows: firstly we use pandas for reading the SQL query, then selecting Currency column and counting the rows (eventually grouped by the currency name). We also drop the null values, which would damage the graph.



Which currencies creating the biggest gap? - Statistic.

```
c.execute("SELECT Currency, ROUND(SUM(ABS(CurrentPrice-FairPrice)),2) FROM Analysis
          WHERE Currency IS NOT Null GROUP BY Currency ORDER BY Currency")
```

The query starts with 'c.execute' which is necessary for executing SQL queries in our editor. We selecting the Currency column and rounded, absolute difference between Current Price and Fair Price from the Analysis table, including condition that Currency column is not null - it would only bias the statistics with Null values. In the end we needed to group it by the currency, as we wanted the summed up comparison.

Differences between Fair Price and Current Price by currency

Currency	Difference
EUR	41.28
JPY	5.83
KRW	9.83
PLN	10.7
USD	24.38

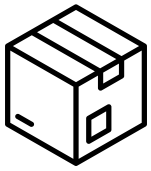
Packages and modules

flask

We used the flask package to create the web application. We used functions Blueprint, render_template, request, flash, redirect, session, url_for, jsonify.

sqlite3

SQLite3 package is our key tool for connection to the SQL database.



pandas [3]

For the model calculations the Pandas package has been used. The main purpose was to process out database to convenient form for the graph.

datetime

In order to transform pandas' objects to the date format, we used datetime packag

werkzeug.security

This module is a part of this library that provides various security-related functions. We use "generate_password_hash" to generate a secure hash of a given password. It uses the bcrypt algorithm to generate a password hash, which is a one-way function that can be used to store a password in a database.
Then we use "check_password_hash" to check if a given password matches the hash generated by generate_password_hash. It takes in the password and the hash and returns a Boolean indicating whether or not the password is valid.

functools

Functools is a built-in Python module that provides various tools for working with functions. wraps is a decorator function from the functools module that can be used to modify the behavior of a function. The wraps decorator is typically used to "preserve" the metadata of the decorated function, such as its name, docstring, and argument signature.

os

The os package was used to check for the existence of a directory and create it if it does not exist.

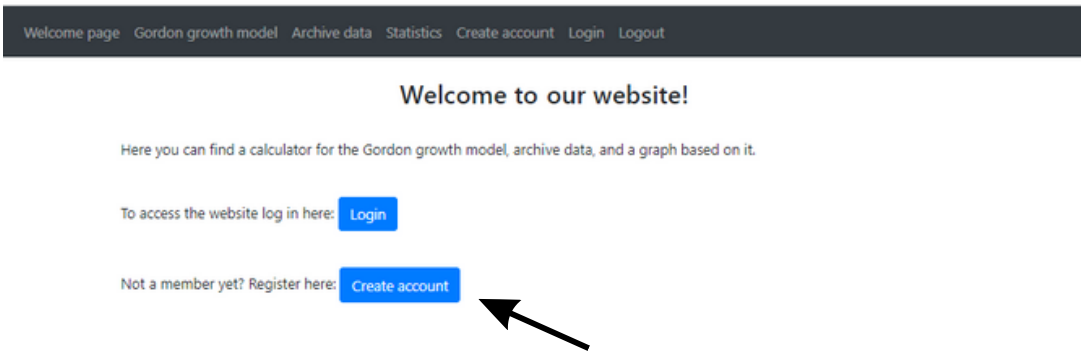
website

Create_app function from the website module creates an app and starts the app by running the app.run(debug=True) command.

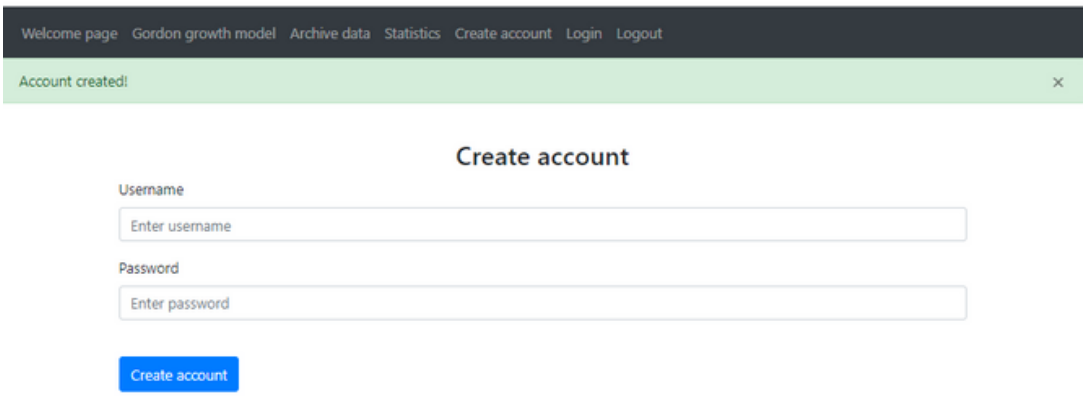
All the queries (included in the presentation and the remaining) are avaliable in the Appendix.

User manual

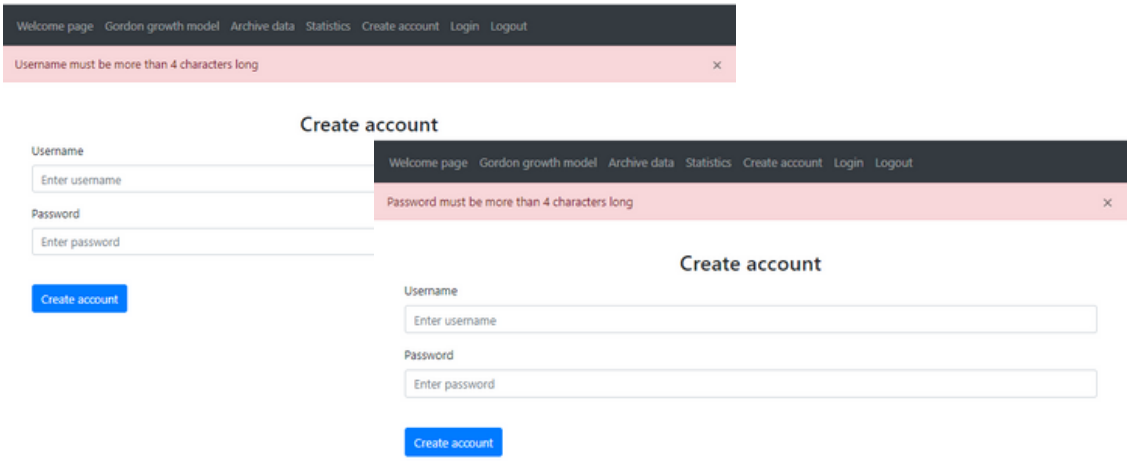
Our application starts with a welcome page, where you can choose whether you are already registered and want to log in or whether you are a new user and want to create an account.



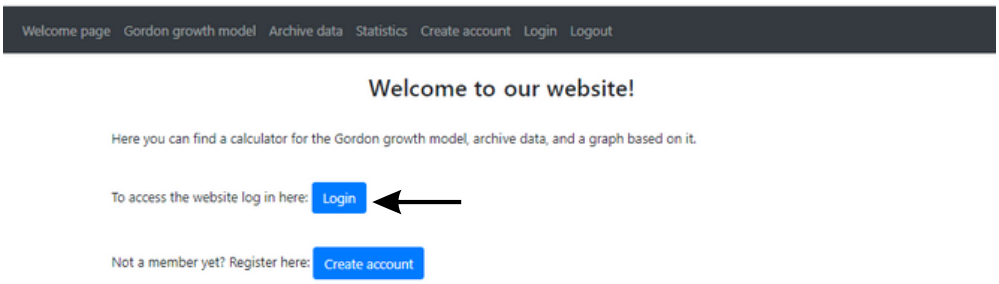
If you select create account you should enter username and password then it will say "Account created!"



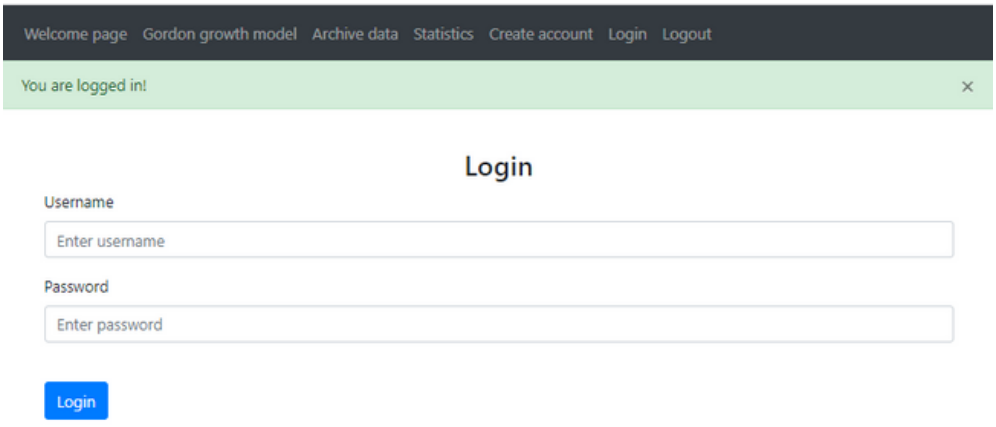
However, if the password or username is too short the message "Password must be more than 4 characters long" or "Username must be more than 4 characters long" will appear.



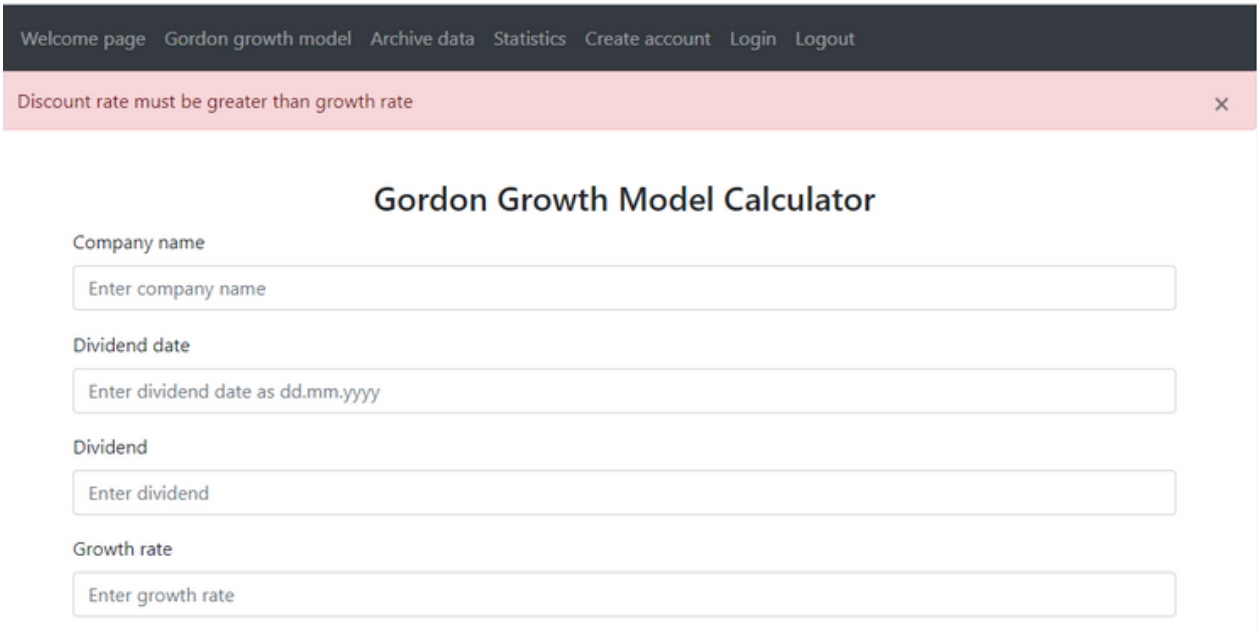
If you select "Login" on the welcome page the button will take you to a subpage where you should enter your username and password. Once you have logged in correctly then it will say "You are logged in!".

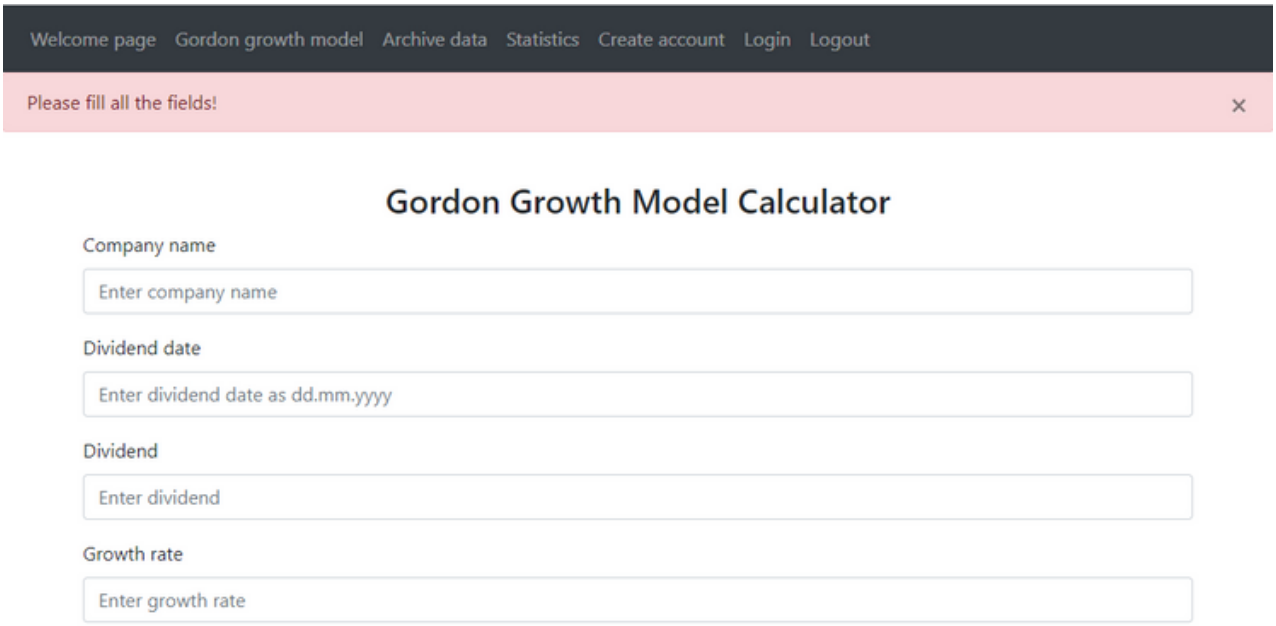


Once logged in, you can use the website and all its functions.

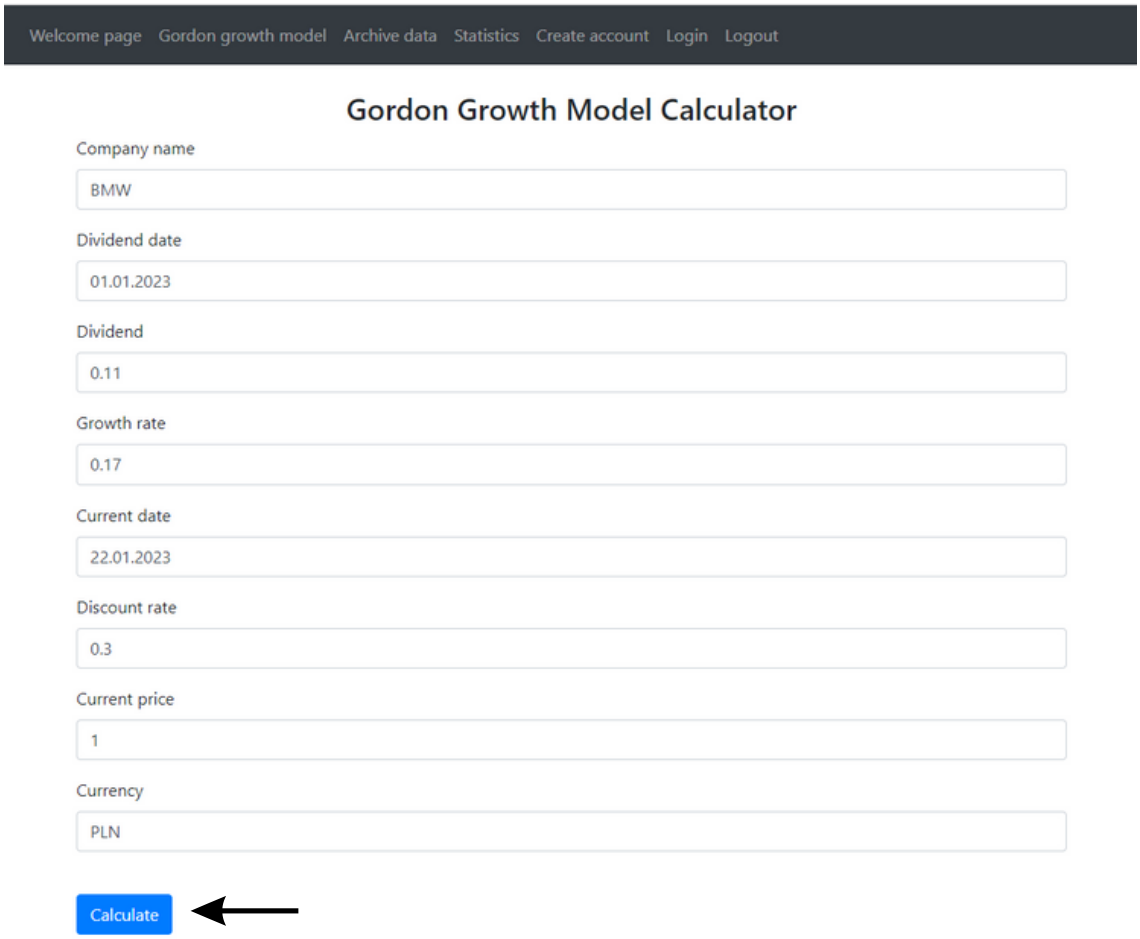


First of all, you can calculate the fair price in the Gordon Growth Model Calculator, but you must remember to fill in all the fields, enter divident date and current date as dd.mm.yyyy and that the discount rate must be greater than the growth rate.

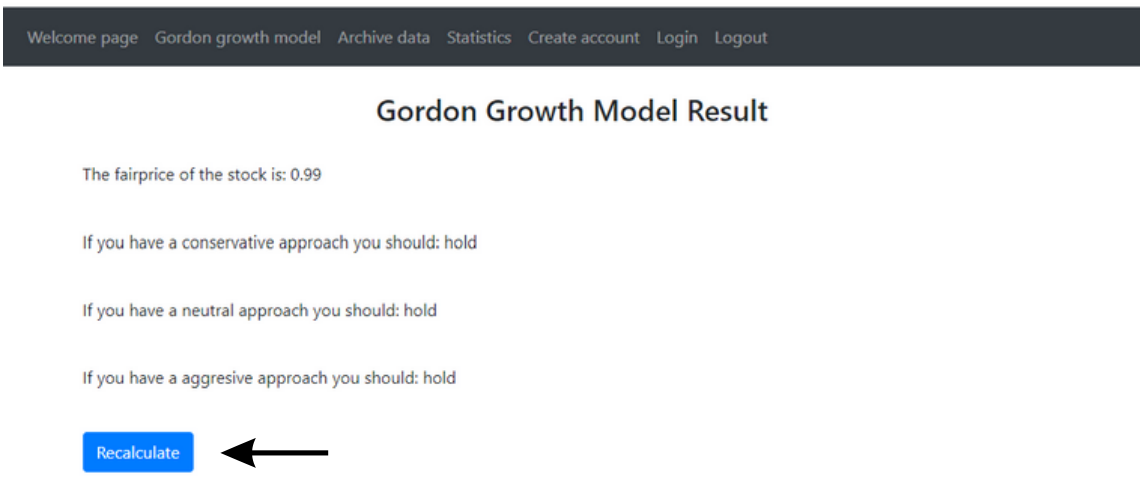




If you enter all the data correctly, click "Calculate" and it is computed to the fair price value (however, some of them are not necessary for the calculations, just for art of financial data sake, e.g. the currency).



First our result is displayed, i.e. the fair price of the stock, and then we have 3 recommendations depending on whether our approach is conservative, neutral or aggresive. Then the 'Recalculate' button enables for the user to come back to the input page.



Stock Table

Input data

→ Calculations

Analysis Table

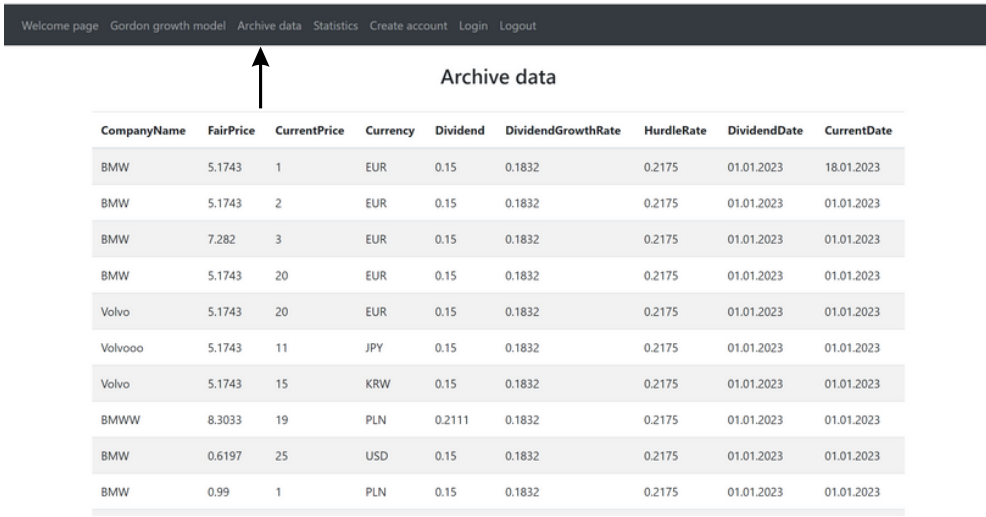
Fair Price

→ Comparison to the Market Price, approach based assignment

Recommendation Table

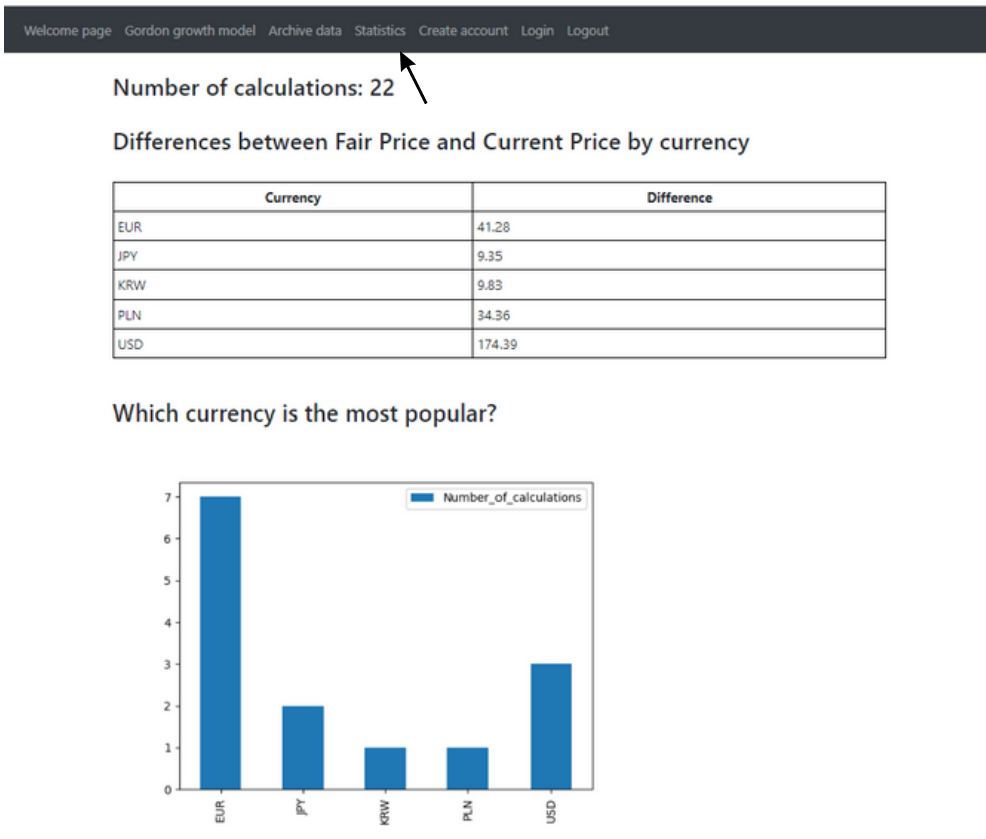
Recommendation

In the next step, we can view archived data which users have entered on the calculator.



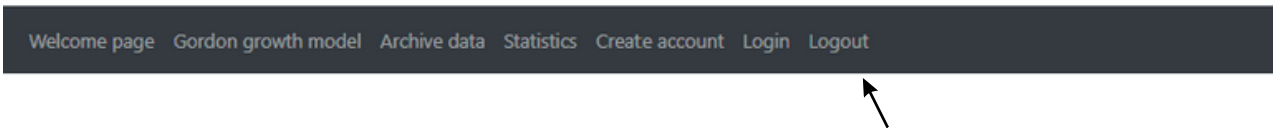
CompanyName	FairPrice	CurrentPrice	Currency	Dividend	DividendGrowthRate	HurdleRate	DividendDate	CurrentDate
BMW	5.1743	1	EUR	0.15	0.1832	0.2175	01.01.2023	18.01.2023
BMW	5.1743	2	EUR	0.15	0.1832	0.2175	01.01.2023	01.01.2023
BMW	7.282	3	EUR	0.15	0.1832	0.2175	01.01.2023	01.01.2023
BMW	5.1743	20	EUR	0.15	0.1832	0.2175	01.01.2023	01.01.2023
Volvo	5.1743	20	EUR	0.15	0.1832	0.2175	01.01.2023	01.01.2023
Volvooo	5.1743	11	JPY	0.15	0.1832	0.2175	01.01.2023	01.01.2023
Volvo	5.1743	15	KRW	0.15	0.1832	0.2175	01.01.2023	01.01.2023
BMWW	8.3033	19	PLN	0.2111	0.1832	0.2175	01.01.2023	01.01.2023
BMW	0.6197	25	USD	0.15	0.1832	0.2175	01.01.2023	01.01.2023
BMW	0.99	1	PLN	0.15	0.1832	0.2175	01.01.2023	01.01.2023

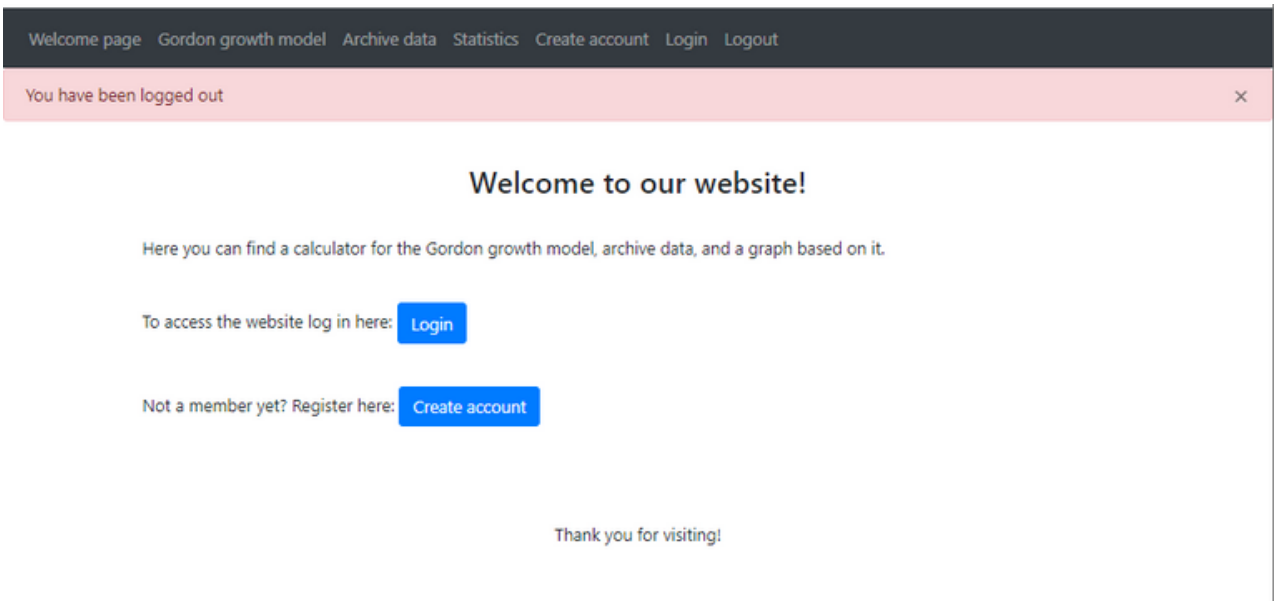
We can also view some statistics created from the data entered.



Currency	Difference
EUR	41.28
JPY	9.35
KRW	9.83
PLN	34.36
USD	174.39

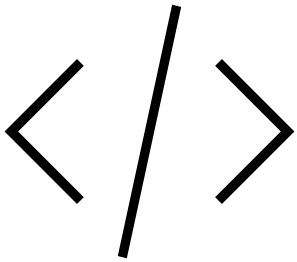
Finally, we can log out.





STOCK MANAGEMENT TOOL

SQL queries distribution



Karolina

```
INSERT INTO User (Username, Password) VALUES (?,?)", (username,
hashed_password)
```

This statement is inserting a new user with a given username and hashed password into the User table in a SQLite database.

```
SELECT Companies.CompanyName, Analysis.FairPrice, Analysis.CurrentPrice,
Analysis.Currency, Stock.Dividend, Stock.DividendGrowthRate, Stock.HurdleRate,
Stock.DividendDate, Stock.CurrentDate FROM Analysis JOIN Companies ON
Analysis.Company_id = Companies.Company_id JOIN Stock ON Analysis.Company_id =
Stock.Company_id
```

The JOIN clause is used to combine rows from two or more tables based on a related column between them. ON clause is used to specify the condition for the join. This query is selecting and joining multiple columns from three tables based on the common column, Company_id, to retrieve the information about a specific company, its fair price, current price, currency, dividend, dividend growth rate, hurdle rate, dividend date and current date. It is stored on the Archive data tab, where the user has full insight for potential portfolio (based on every users' input).

```
SELECT count(*) FROM Analysis
```

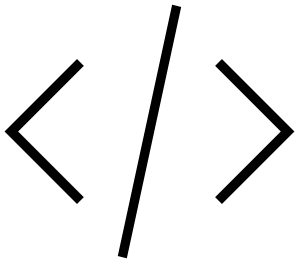
```
SELECT Currency, ROUND(SUM(ABS(CurrentPrice-FairPrice)),2) FROM Analysis WHERE
Currency IS NOT Null GROUP BY Currency ORDER BY Currency
```

```
SELECT Currency, count(*) as Number_of_calculations FROM Analysis where
currency is not null group by currency
```

We used above queries for Statistics tab. First one provides information about number of calucations the users have provided. Second, refers to the difference between Current Price and Fair Price of analyzed companies. It provides valuable information, how big is a absolute potential profit, depending on the currency. Purpose of the last query is to determine number of calculations depending on the currency, in other words the currency popularity.

STOCK MANAGEMENT TOOL

SQL queries distribution



Zuzanna

```
SELECT * FROM User WHERE Username=?", (username,)
```

This query is selecting all columns from the User table where the value of the Username column matches the specific value passed as a variable. It retrieves data from the "User" table in a database, where the value of the "Username" column matches a specific value. The query is using the SELECT, FROM and WHERE clauses. The SELECT clause is used to select all columns from the table by using the wildcard (*), this means that all columns will be retrieved. The FROM clause is specifying the table that the data should be retrieved from, which is "User" in this case. The WHERE clause is used to filter the rows based on a specific condition. In this case, it is used to filter the rows where the "Username" column is equal to the value passed as a variable (username).

```
SELECT MAX(Company_id) FROM Stock
```

```
SELECT MAX(Input_id) FROM Analysis
```

AThis is a SQL query that retrieves the maximum value of the "Input_id" column from the "Analysis" table. The query is using the SELECT and MAX() statements. The SELECT statement is used to select the data from the table and MAX() function is used to retrieve the maximum value of a specific column. The query is returning the highest value of the "Input_id" column so that in the next step we can generate a new unique one.

```
INSERT INTO Companies (CompanyName) VALUES (?)", (companyname,)
```

```
INSERT INTO Stock (Company_id, Dividend, DividendDate, DividendGrowthRate,
CurrentDate, HurdleRate, CurrentPrice, Currency) VALUES (?,?,?,?,?,?,?,?)",
(company_id, dividend, dividend_date, growth_rate, current_date, discount_rate,
current_price, currency,)
```

```
INSERT INTO Analysis (Company_id, CurrentPrice, FairPrice, Currency, Input_id)
VALUES (?,?,?,?,?)", (company_id, current_price, fairprice, currency,
input_id,)
```

The queries are important for the Gordon Growth Model tab, where each new Company Name is being added to the Companies table. Subsequently, we have the query for user's input to the Stock table from the Gordon Growth Model tab. And lastly, previously defined fair price from GGM is being added to the Analysis table, together with company id, current price, currency and input id.

This statement is inserting a new record into the Stock, Analysis and Company table, with values, which the user enters on the page in the Gordon Growth Model Calculator. This is a SQL statement that is inserting a new record into the "Stock" table in a database. The statement is using the INSERT INTO clause and the VALUES clause to specify the values to be inserted. The INSERT INTO clause is specifying the name of the table that the data should be inserted into. The VALUES clause is providing the values for each column in the table, in the order that the columns are listed.

References and work distribution

1) Gambetti, E., & Giusberti, F. (2019). Personality, decision-making styles and investments. Journal of Behavioral and Experimental Economics, 80, 14–24. https://doi.org/10.1016/j.socec.2019.03.002

2) Gordon Growth Model formula: How to calculate constant growth rate. (n.d.). https://www.paddle.com/resources/gordon-growth-model

3) Ali, A. (2022). How to insert pandas Dataframe to SQLite table in Python. AlixaProDev. https://www.alixaprodev.com/2022/04/insert-pandas-dataframe-to-sqlite-table.html

4) https://www.xtb.com/pl

5) https://www.gurufocus.com

Karolina Solarska

- Project proposal: database scheme, description SQL queries (Calculating formula), list of external Python packages
- Web application: creating in database tables Analysis, Recommendation, Companies, home page, login and logout subpages, the password hash creation, defining decorator functions "auth_required", flashes in login and logout subpages, statistics - obliczanie i wstawianie graphu i tabeli, gordon growth model - recommendations calculations, templates - base, statistics, login, datadump (tables, graphs, buttons, captions, fields design) the website's layout,
- Report: introduction, user manual description, SQL queries (relating to Karolina)
- Presentation: database scheme, screenshots (create account, gordon growth model calculator, gordon growth model result, archive data)

Zuzanna Kostecka

- Project proposal: introduction, description SQL queries (Recommendation formula), pictures describing screens of designed application
- Web application: creating in database tables Portfolio, Stock, User, create account subpage, welcome page, defining decorator functions "login_required", flashes in create account and gordon growth model subpages, datadump - wyświetlanie danych w tabeli na stronie, gordon result subpage, defining "get_max_company_id" and "get_max_input_id", gordon growth model - fair price calculation, saving data to database, templates - base, sign_up, gordon_growth_model, gordon_result (tables, buttons, captions, fields design)
- Report: database scheme, screenshots of all screens, SQL queries (relating to Zuzanna), references
- Presentation: introduction, screenshots (welcome page, login, statistics, logout), references