Creating a web application for analyzing the investment attractiveness of medical drugs

Taras Dubyniak[1,*,†], Aleksandra Kłos-Witkowska[2,†], Oleksandra Manziy[3,†] , Stanislav Andreychuk[3,†] and Tetiana Kukharchuk[3,†]

[1] *Ternopil Ivan Puluj National Technical University, Rus'ka str. 56, Ternopil, 46001, Ukraine*
[2] *University of Bielsko-Biala, Willowa St. 2, Bielsko-Biala, 43-300, Poland*
[3] *Lviv Polytechnic National University, S.Bandera str. 12, Lviv, 79013, Ukraine*

Abstract
The main development tools and functionality of the web application designed for analyzing the investment attractiveness of medical drugs are described. Technical indicators of charts, including theoretical pricing foundations and their signals for forecasting investor behavior, are analyzed. The role of informational factors and their impact on asset price formation is explored. Detailed descriptions are provided of the main price indicators and technical analysis tools used as the basis for market condition analysis when making investment decisions regarding the trading of medical drugs under informational uncertainty.

Keywords
Information system, medical drugs, financial markets, online platform, data analysis tools

1. Introduction

The application of information technology significantly enhances the accessibility of information for investors and ensures its rapid dissemination, facilitating more informed decision-making. Analysis of recent research highlights the importance of the informational component and its analytical methods for effective decision-making regarding investments or product analysis under unforeseen circumstances [1]. Classical methods of fundamental and technical analysis [2], which form the basis for asset price forecasting and investment decisions, need to be updated and supplemented due to the increasing role of informational components and the expanded technological capabilities for analyzing large data sets, particularly in the context of informational uncertainty. More

BAIT'2024: The 1st International Workshop on "Bioinformatics and applied information technologies", October 02-04, 2024, Zboriv, Ukraine
* Corresponding author.
† These authors contributed equally.
d_taras@ukr.net (T.Dubynyak); awitkowska@ath.bielsko.pl (A. Kłos-Witkowska); oleksandra.s.manzii@lpnu.ua (O.Manziy); stanislav.k.andreichuk@lpnu.ua (S.Andreychuk); tetiana.kukharchuk.pm.2020@lpnu.ua (T.Kukharchuk)
0000-0003-1529-6951 (T.Dubynyak); 0000-0003-2319-5974 (A. Kłos-Witkowska); 0000-0002-6480-2307 (O.Manziy); 0000-0003-3226-7300 (S.Andreychuk); 0000-0002-9421-8566 (T.Kukharchuk)

studies confirm the relevance of using artificial intelligence and neural networks in investment analysis [3-5].

Important sources of financial information that provide up-to-date data for further analysis and use in applications include APIs (Application Programming Interfaces) such as those provided by Financial Modeling Prep (FMP) and Yahoo Finance. Overall, using these tools and financial data sources can significantly ease the process of analyzing medical drug prices and making financial decisions regarding investments in the medical sector. However, it is important to understand their limitations and risks, particularly concerning the accuracy and availability of data.

Today, there are many professional information systems and software products for analyzing and managing investments effectively. However, most of these products are complex multifunctional platforms that require specialized training for use. Meanwhile, presenting reliable information to consumers in a visually understandable format, along with investment tools that enable non-professional users to assess the investment attractiveness of products in a selected sector, is crucial. Information systems and web applications described in [6-7] were developed to properly inform end-users about the characteristics of offered financial products. Accessible presentation of product information allows consumers to evaluate whether the selected product meets their needs and whether they are prepared to accept the associated investment risk.

Modern software products offer libraries that are powerful tools for data visualization and processing, with most of the information encountered by investors presented in tabular or graphical formats. According to the methodology of scientific knowledge, people better perceive visualized methods of information presentation. This underscores the relevance of developing a custom software product that will provide the capability to analyze the investment attractiveness of selected medical sector products, forecast price movements based on historical data, and take into account technical analysis indicators.

2. Development tools

After analyzing existing tools for web application development, it was decided to focus on the most commonly used web resources for financial asset data available freely, and modern programming languages whose libraries provide the necessary tools for visualizing asset price movement analysis results.

2.1. Using APIs for data retrieval

An API (Application Programming Interface) is a set of rules and mechanisms that allows programs to interact with each other. In general terms, an API is a way for programs to communicate, opening up numerous opportunities for integration, development, and automation of software.

Using APIs for data retrieval typically involves making HTTP requests. The web server that provides the API processes these requests and returns a response in various formats, such as JSON, XML, or others. Developers can use this data in their applications for purposes such as analysis, visualization, processing, or displaying to the user. [9] API usage allows programmers to quickly and conveniently access up-to-date information for their projects without the need to develop their own data collection mechanisms.

In the developed web application, APIs are used to provide access to the functions or data of the web server. This allows different programs and services to exchange information and perform various actions over the Internet.

Currently, the most popular sources of freely available data for financial assets are considered to be Yahoo Finance and Financial Modeling Prep.

YahooFinance (https://finance.yahoo.com)

Yahoo Finance is an online platform that provides a wide range of financial information, including data on stocks, financial news, charts, analyses, portfolios, and other tools for investors and finance professionals. Key features of Yahoo Finance include access to real-time stock prices and other financial instruments, analytical reports, data visualization charts, portfolio management functionality, and financial news updates.

Advantages of Yahoo Finance:

- **Wide Range of Financial Information**: The platform provides access to a vast amount of financial data, news, and analytical information, enabling investors and finance professionals to obtain the necessary details for making informed decisions.
- **Free Access**: Most of Yahoo Finance’s functionality is available for free, making it accessible to a broad audience of users.
- **Flexible Tools**: The platform offers a variety of tools for data analysis, including charts, analytical reports, and portfolio management features.

Disadvantages of Yahoo Finance:

- **Limited Depth of Analysis**: While Yahoo Finance provides a substantial amount of financial information, some aspects of analysis may be limited compared to other platforms.
- **Dependence on Third-Party Sources**: Some information provided on the platform may be based on data from third-party providers, which can lead to potential inaccuracies or delays in data updates.
- **Lack of Personalized Approach**: The platform may not account for individual user needs and preferences in financial analysis.

Financial Modeling Prep (**https://site.financialmodelingprep.com*****)***

Financial Modeling Prep (FMP) is an online platform that provides tools and resources for financial modeling and analysis. Key features of FMP include access to financial reports, indicators, historical data, and APIs for retrieving financial information.

Advantages of FMP:

- **Wide Access to Financial Data**: FMP provides access to a broad range of financial data about companies, allowing for analysis of their financial stability and performance.

- **Ease of Use**: The platform offers a user-friendly interface and documentation, making it easy for users to access and utilize the information for financial modeling.
- **APIs for Developers**: FMP offers APIs for developers, enabling the automation of financial data retrieval and integration with other applications and services.

Disadvantages of FMP:

- **Limited Free Version**: While FMP offers a free plan, it is limited in functionality and data access compared to the paid plans.
- **Dependence on Third-Party Data Sources**: Some of the information provided by FMP may be based on data from third-party providers, which could lead to potential inaccuracies or delays in data updates.
- **Lack of Personalized Approach**: Although FMP provides a wide range of financial data, it may not cater to the individual needs and specific requirements of users in financial analysis.

2.2. Analysis of the programming language for the application.

Python is a high-level, dynamically typed, interpreted, and object-oriented programming language. It is used for web development, scientific computing, artificial intelligence, data analysis, task automation, and more. Its open-source nature and extensive libraries make it especially popular among developers.

Advantages:

- **Simple syntax**: Python has a clean and readable syntax, which makes learning and understanding the language easier for beginners.
- **Large number of libraries**: Python has a wide range of libraries for various tasks, allowing developers to solve problems quickly and work efficiently.
- **Wide applicability**: Python is used in various fields, including web development, scientific computing, artificial intelligence, data analysis, and much more.

Disadvantages:

- **Moderate speed**: Python, being an interpreted language, can be slower compared to compiled programming languages, especially for large computations or resource-intensive applications.
- **Resource limitations**: In some cases, Python may have limitations in resources such as memory or parallel task processing, which can make working with large data sets or resource-intensive tasks more challenging.

Given these advantages and disadvantages, the authors consider using Python effective for implementing the functionality of a web application designed to analyze the investment attractiveness of products in the medical sector.

3. Overview of the mathematical model of the application

In this section, we will review the key indicators and price metrics used in analyzing the investment attractiveness of assets in financial markets. These tools help assess the state of the medical products market and their potential for investment.

The mathematical model is based on technical indicators that help understand market dynamics and price metrics, determine asset values, and compare their performance in the current unpredictable financial environment. The model also includes volatility, which indicates the level of risk and potential price changes, and trading volume, which reflects market participants' activity.

3.1. Technical Indicators

Moving Average (MA) is a widely used indicator in technical analysis. It is calculated to help smooth out an asset's price data by creating a continually updated average price. This allows traders and analysts to better identify trends and reduce the noise of short-term price fluctuations.

The calculation of moving averages softens the impact of random short-term fluctuations on the price over a certain period of time. Simple Moving Averages (SMA) use the arithmetic mean of prices over a specified time interval, whereas Exponential Moving Averages (EMA) give more weight to recent prices over a certain period of time. [10].

Simple Moving Average (SMA) is a technical indicator used for market analysis and forecasting its future movement. The Simple Moving Average is calculated by taking the arithmetic mean of a set of data over a specified period.

The formula for calculating the Simple Moving Average (SMA) is [11]:

$$SMA = \frac{P_1 + P_2 + \ldots + P_n}{n},$$

where $P_i, i = \overline{1,n}$ – average closing prices over the selected period;
n – number of periods.

In technical analysis, when the SMA crosses the asset's price, it can indicate a potential trend change. Therefore, by analyzing the interaction between closing prices and the SMA, one can identify market trends and make appropriate trading decisions.

The **Exponential Moving Average** (EMA) gives more weight to recent prices, making it more responsive to new information. As a result, EMA is considered a weighted calculation. EMA is designed to improve on the idea of SMA by assigning greater weight to the most recent price data, which is deemed more relevant than older data. Because recent data has more weight, EMA reacts more quickly to price changes compared to SMA [10].

When calculating the EMA, an initial Simple Moving Average (SMA) for the specified period is first computed and used as the starting value for the EMA. Then, a weighting multiplier, known as the "smoothing factor," is calculated:

$$\alpha = \frac{2}{window + 1}$$

where *window* – selected period.

Formula for calculating the exponential moving average (EMA) of a security [11]:

$$EMA = (close_t \times \alpha) + EMA_p \times (1 - \alpha)$$

where $close_t$ – closing price on the last day of the selected period;

EMA_p – EMA for the previous period.

Analyzing the interaction between closing prices and EMA allows for a better understanding of market trends and helps make appropriate trading decisions, making EMA a useful tool for analysis.

Moving Average Convergence Divergence.

The Moving Average Convergence Divergence (MACD) is a technical indicator that helps investors identify price trends, measure trend momentum, and determine entry points for buying or selling. The MACD shows the relationship between two exponential moving averages (EMAs) of an asset's price [12].

The MACD line is calculated by subtracting the 26-period EMA from the 12-period EMA. The 9-period EMA line, known as the signal line, is plotted over the MACD line and can serve as a trigger for buy or sell signals. Investors may buy a security when the MACD line is above the signal line and sell when the MACD line is below the signal line.

Formula for calculating MACD[13]:

$$MACD = 12PeridEMA - 25PeridEMA$$

Thus, buying or selling an asset can be recommended based on the position of the MACD line relative to the signal line. The MACD is a useful tool for analyzing price changes.

Relative Strength Index

The Relative Strength Index (RSI) is a momentum indicator used in technical analysis. RSI measures the speed and magnitude of recent price changes to evaluate overbought or oversold conditions of a stock's price.

RSI is displayed as a line graph on a scale from 0 to 100. If it reaches very high or very low values, it may indicate that the current trend could change. RSI is used to identify potential entry and exit points in the market. For example, if the RSI shows a value above 70, it may signal a potential price decline. Conversely, if the RSI shows a value below 30, it may suggest a potential price increase [11].

The formula for calculating the RSI indicator consists of two equations. The first equation is used to obtain the initial Relative Strength (RS), which represents the ratio of the average closing prices during upward movements ("Up") to the average closing prices during downward movements ("Down") over a period of NNN. The formula is as follows:

$$RS = \frac{gain}{loss}$$

where *gain* - average closing price Up over period N;

loss - average closing price Down over period N.

The actual RSI value is calculated by normalizing the indicator to a range from 0 to 100 using the following formula:

$$RSI = 100 - \frac{100}{1 + RS}$$

RSI helps assess the degree of overbought or oversold conditions of an asset, which can indicate to the investor a potential trend reversal.

Bollinger Bands.

Bollinger Bands are a technical analysis tool used to determine where prices are high or low relative to each other. The bands expand when a stock's price becomes more volatile and contract when it is more stable. Stocks are considered overbought when their price approaches the upper band and oversold when it approaches the lower band, signaling potential trading opportunities. The three lines that make up the Bollinger Bands are based on price changes of securities. The central line represents the intermediate-term trend and is typically a 20-day SMA of the closing prices. The upper and lower bands are plotted a certain number of standard deviations, usually two, above and below the central line [14].

Formula for calculating Bollinger Bands:

$$\text{UpperBand} = SMA + (k \times SD)$$
$$\text{LowerBand} = SMA - (k \times SD)$$

where $SD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - SMA)^2}$; k – number of standard deviations (usually 2).

Bollinger Bands are a useful technical analysis tool that helps assess market volatility and identify potentially overbought or oversold conditions.

3.2. Price Indicators

Price indicators are numerical values used to analyze asset price movements in financial markets. These indicators help investors understand and forecast market dynamics. Here are some of the most common price indicators:

- **Close price** (closing price): The price at which an asset closed at the end of the trading day. It is often used to analyze trends and determine support and resistance levels.
- **Open price** (opening price): The price at which an asset opened at the beginning of the trading day. It can indicate the start and direction of price movements throughout the day.
- **High price**: The highest price an asset reached during the trading day. This indicator can be used to determine the maximum activity of buyers.
- **Low price**: The lowest price an asset reached during the trading day.

It can indicate minimal seller activity. These price indicators are used for various purposes, including technical analysis, fundamental analysis, and forecasting market trends.

Volatility is a measure that indicates how much the prices of a financial asset fluctuate over a specific period [14]. If the volatility is high, it means that the price is experiencing significant fluctuations; if it is low, the price is more stable. High volatility presents the potential for greater profits, but it also increases risks, as the price of a security can change dramatically in a short period in either direction. Conversely, when volatility is low, the price of the security does not fluctuate sharply and tends to be more stable, reducing risks but also potentially limiting significant profit opportunities [15, 16].

Trading volume is a measure of how much of a particular asset has been sold over a specified period. For stocks, volume is measured by the number of shares traded. Volume determines liquidity and, in combination with technical indicators, is a crucial metric for making investment decisions. Volume can indicate market strength, as rising markets with increasing volume are typically considered strong and healthy. When prices fall along with increasing volume, the downtrend gains strength, indicating that selling pressure dominates over buying pressure, leading to a decrease in prices. Thus, trading volume is crucial for market analysis and trading decisions. It helps assess the strength of price movements, confirm trading signals, and evaluate market liquidity.

For the investment attractiveness of medicinal products, it is proposed to use the results of qualitative neural network research, including discrete and distributed time delays [17]. A method of calculating the exponential decay rate for a neural network model based on differential equations with a discrete delay was developed and applied [18], [19].

When developing the investment attractiveness of medicinal products, the direction of using sensors [20], [21], in particular for monitoring important indicators of a person and assessing his functional state, is promising. An important characteristic [22] of various types of biosensors is stability [23], [24]. Scientific studies [25], [26] give examples of sensor response modeling. Numerical modeling in cyber-physical biosensor systems [27], [28] is important at the stage of their design.

4. Overview the web application functionality

This section provides an overview of the web application developed by the authors for in-depth analysis of data from companies in the medical sector and comparison of their financial indicators with other market participants. This tool, combined with its built-in analysis and visualization features, enables users to gain a better understanding of price trends to make informed investment decisions.

The main features of the web application include:

- **Volatility Chart -** Displays the price fluctuations of the selected company's stock. Users can compare this chart with the volatility of other companies, allowing them to assess risks and identify more stable investment opportunities.
- **Trading Volume Chart** - Shows the trading activity of the company's stock. The ability to compare trading volume with other companies helps to understand the popularity of the stock among investors and its liquidity.
- **Closing Price Chart -** Displays the closing prices of the stock. Users can overlay various indicators, enabling technical analysis and forecasting of future price movements.

The developed web application can be a valuable tool for analysts, investors, and other stakeholders, providing them with powerful means for analyzing financial data. The ability to compare key metrics across different companies will aid in making more informed and balanced investment decisions, contributing to the achievement of financial goals.

4.1. Main page

On the main page, the user must enter the abbreviation of the desired company (e.g., "JNJ") into a form (see Figure 1):

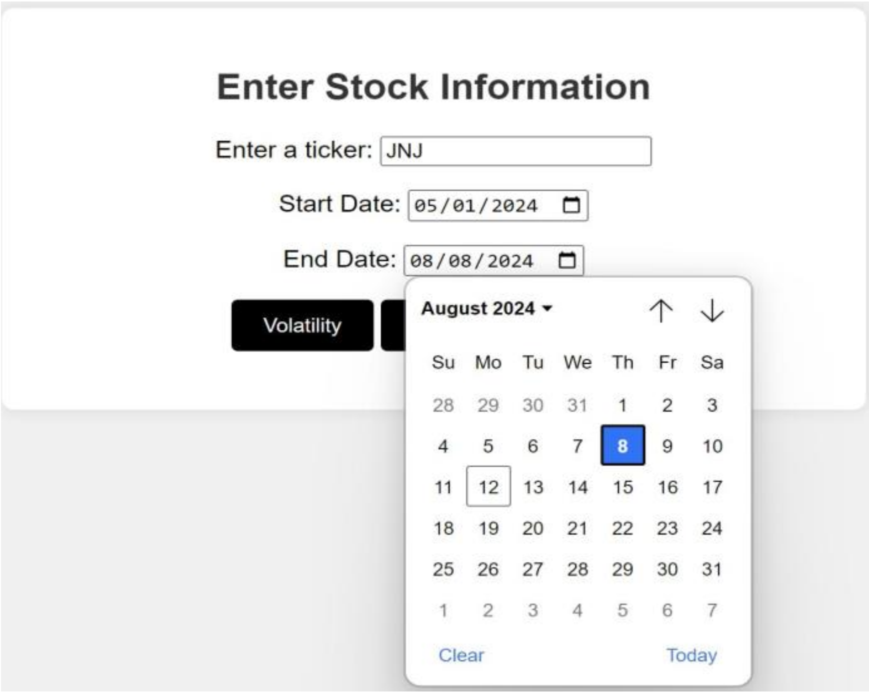

Figure 1: Company and Period Input

The user must also specify the analysis period, which can be entered manually or selected using a dropdown calendar. After entering these parameters, the user has the option to choose one of three analysis options (see Figure 2):

Figure 2: Analysis Options and Buttons

In the section of the code in Figure 3, the data retrieved from the form is processed. If the form is valid, the data is read: the company abbreviation and the date range (start and end dates). Then, depending on which button the user clicks, the request is redirected to the corresponding function for further processing and result output.

```
def home(request):
    if request.method == 'POST':
        form = TickerForm(request.POST)
        if form.is_valid():
            tickers = form.cleaned_data['tickers'].split()[:10]
            start_date = form.cleaned_data['start_date']
            end_date = form.cleaned_data['end_date']
            ...
            if 'volatility' in request.POST:
                return vol_tickers(request)
            if 'volume' in request.POST:
                return volume_tickers(request)
            if 'indicators' in request.POST:
                return price_tickers(request)
    else:
        form = TickerForm()
    return render(request, template_name: 'main/about.html', context: {'form': form})
```

Figure 3: Retrieving Data from the Form

If the user chooses to view the volatility chart, a page with the corresponding chart will appear on the screen. The user can compare the volatility of this company with that of others by entering the abbreviations of the other companies, separated by spaces, into the form located in the upper left corner, and then clicking the "Compare" button. After making this selection, the user can add additional companies if needed or click the "Reset" button (see Figure 4) to return to the initial volatility chart.

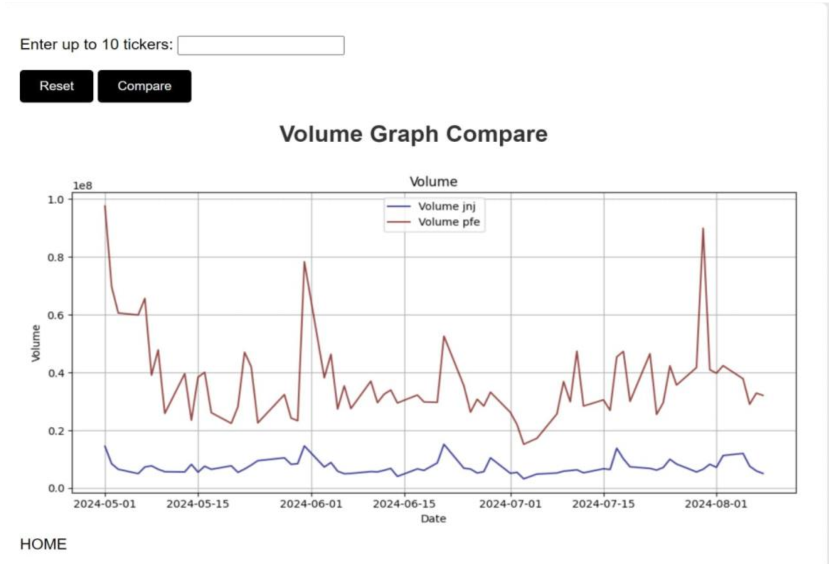

Figure 4: Comparative Volatility Chart for Selected Companies

The plot_volatility function (see Figure 5) is responsible for retrieving historical data for each company in the list of tickers for the specified period using a financial API.

Afterward, volatility is computed in the rolling_volatility function (see Figure 5) for each company, which will be represented on the chart with a separate line in a corresponding color.

```
def rolling_volatility(close_prices, window=21):
    log_returns = np.log(close_prices / close_prices.shift(1))
    rolling_std = log_returns.rolling(window=window).std()
    annualized_volatility = rolling_std * np.sqrt(252)
    return annualized_volatility
2 usages
def plot_volatility(tickers, start_date, end_date):
    plt.figure(figsize=(10, 5))
    colors = plt.cm.jet(np.linspace( start: 0,  stop: 1, len(tickers)))
    for ticker, color in zip(tickers, colors):
        query_string = f'https://financialmodelingprep.com/api/v3/historical-price-full/{ticker}?from=
        df = pd.read_csv(query_string)
        df['date'] = pd.to_datetime(df['date'])
        volatility = rolling_volatility(df['close'])
        plt.plot( *args: df['date'], volatility, label=f'Volatility {ticker}', color=color, alpha=0.75)
    plt.title('Stock Volatility Comparison')
    plt.xlabel('Date')
    plt.ylabel('Volatility')
    plt.legend()
    plt.tight_layout()
    plt.grid(True)
    return toImage(plt)
```

Figure 5: Constructing the Volatility Chart

4.2. Price Chart Page

On this page, the user receives a closing price chart of the selected medical sector company. Here, they can also overlay an indicator on this chart by selecting it from a dropdown menu (see Figure 6) and then clicking the "Submit" button.

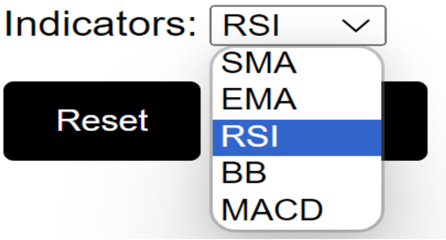

Figure 6: Dropdown Menu with Indicators

The examples of indicators overlaid on price charts you can see in the Figures 7-8 and the screenshots of the corresponding code in the Figures 9-11:

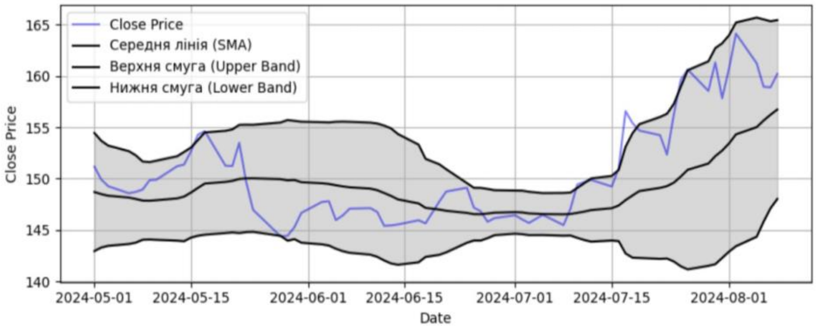

Figure 7: Price Chart with Overlaid Bollinger Bands Indicator

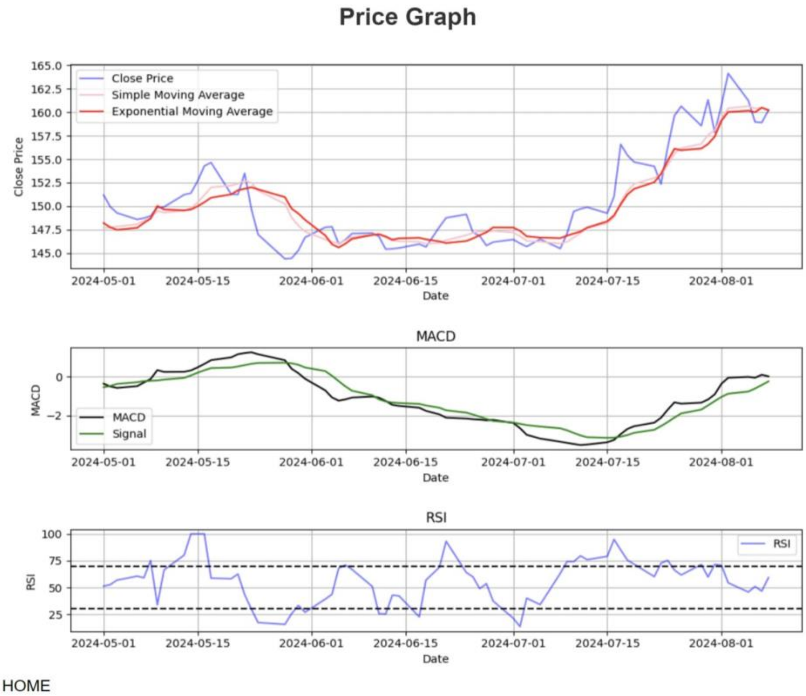

Figure 8: Price Chart with Overlaid Indicators: SMA, EMA, MACD and RSI

In the functions in Figure 9 the simple moving average (SMA) and the exponential moving average (EMA) are computed. These functions are used to visualize the movements of these indicators on the price chart:

```
def MovingAverage(df, mask):
    df['date'] = pd.to_datetime(df['date'])
    date = df['date']
    df['MA_7'] = df['close'].rolling(window=7).mean().shift(-6)
    MA = df['MA_7']
    date = date[mask]
    MA = MA[mask]
    return plt.plot( *args: date, MA, color='pink', alpha=1, label='Simple Moving Average')
1 usage
def ExponentialMovingAverage(df, mask):
    df['date'] = pd.to_datetime(df['date'])
    date = df['date']
    df['EMA_7'] = df['close'].ewm(span=7, adjust=False).mean().shift(-6)
    ema = df['EMA_7']
    date = date[mask]
    ema = ema[mask]
    plt.plot( *args: date, ema, color='red', alpha=1, label='Exponential Moving Average')
```

Figure 9: Functions for Visualizing SMA and EMA Indicators on the Price Chart

The function in Figure 10 is responsible for calculating Bollinger Bands and visualizing their movement on the price chart.

```
def BolingerBands(df, mask):
    date = df['date'] = pd.to_datetime(df['date'])
    period = 20
    SMA = df['SMA'] = df['close'].rolling(window=period).mean().shift(-20)
    # Розрахунок стандартного відхилення
    df['STD'] = df['close'].rolling(window=period).std().shift(-20)
    # Розрахунок верхньої та нижньої смуг
    multiplier = 2
    Upper_Band = df['SMA'] + (multiplier * df['STD'])
    Lower_Band = df['SMA'] - (multiplier * df['STD'])
    date = date[mask]
    SMA = SMA[mask]
    Upper_Band = Upper_Band[mask]
    Lower_Band = Lower_Band[mask]
    # Побудова графіку
    plt.plot( *args: date, SMA, label='Середня лінія (SMA)', color='black')
    plt.plot( *args: date, Upper_Band, label='Верхня смуга (Upper Band)', color='black')
    plt.plot( *args: date, Lower_Band, label='Нижня смуга (Lower Band)', color='black')
    return plt.fill_between(date,  Upper_Band, Lower_Band, color='grey', alpha=0.3)
```

Figure 10: Functions for Visualizing Bollinger Bands on the Price Chart

The RSI function (Fig. 11) computes the relative strength index (RSI) for price fluctuations. The RSI values are then visualized on the chart, which also displays overbought and oversold level lines.

```python
def RSI(df, mask):
    df['date'] = pd.to_datetime(df['date'])
    date = df['date']
    delta = df['close'].diff()
    gain = delta.where(delta > 0, 0).rolling(window=7).mean().shift(-6)
    loss = -delta.where(delta < 0, 0).rolling(window=7).mean().shift(-6)
    rsi = 100.0 - (100.0 / (1.0 + (gain / loss )))
    date = date[mask]
    rsi = rsi[mask]
    plt.plot( *args: date, rsi, color='blue', alpha=0.5, label='RSI')
    plt.axhline(y=70, color='black', linestyle='--')
    plt.axhline(y=30, color='black', linestyle='--')
    plt.title('\nRSI')
    plt.xlabel('Date')
    plt.ylabel('RSI')
    plt.tick_params(axis='y', which='both', left='off', right='off')  # Turn off y-axis ticks
    plt.tight_layout()
    plt.legend()
    return plt
```

Figure 11: Functions for Function for plotting a separate chart for the RSI indicator

In the function in Figure 12 the MACD and signal line values are calculated. First, the 9-period EMA, which serves as the signal line, is computed. Then, the MACD is calculated by subtracting the 12-period EMA from the 26-period EMA. Finally, the MACD and signal line values are visualized on the chart.

```python
def MACD(df, mask):
    exp1 = df['close'].ewm(span=12, adjust=False).mean().shift(-6)
    exp2 = df['close'].ewm(span=26, adjust=False).mean().shift(-6)
    macd = exp1 - exp2
    signal = macd.ewm(span=9, adjust=False).mean().shift(-6)
    date = df['date']
    date = date[mask]
    macd= macd[mask]
    signal = signal[mask]
    plt.plot( *args: date, macd, color='black', alpha=1, label='MACD')
    plt.plot( *args: date, signal, color='green', alpha=1, label='Signal')
    plt.title('\nMACD')
    plt.xlabel('Date')
    plt.ylabel('MACD')
    plt.tight_layout()
    plt.legend()
    plt.grid(True)
    return plt
```

Figure 12: Functions for Function for plotting a separate chart for the MACD indicator

4.3. Volume Chart Page

On a separate page, the user can analyze the trading volume graph of the selected company and also has the option to compare this company's trading volume with that of other companies (Fig. 14). The user should enter the abbreviations of these companies in the designated form, separated by spaces, and then click the "Compare" button.

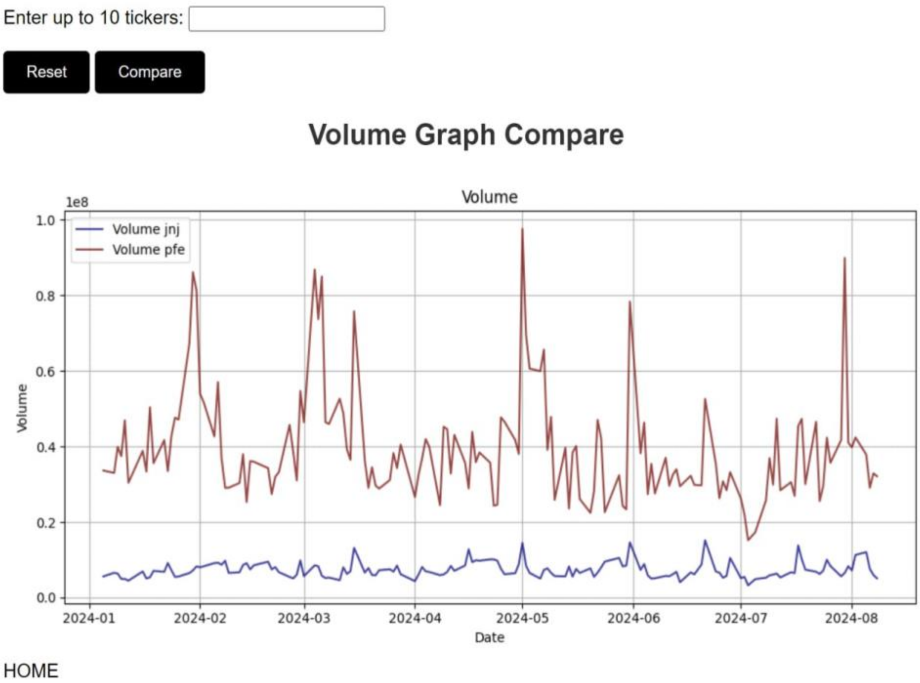

Figure 13: Comparative Trading Volume Chart of Selected Companies

After making a selection, the user can either add more companies or click the "Reset" button to return to the original trading volume chart.

The plot_volume function in the Figure 14 is responsible for retrieving historical data for each company in the list of tickers using a financial API over the specified period. Subsequently, a chart is constructed for each company with a distinct line color (Fig. 14).

```
def plot_volume(tickers, start_date, end_date):
    plt.figure(figsize=(10, 10))
    colors = plt.cm.jet(np.linspace(0, 1, len(tickers)))
    for ticker, color in zip(tickers, colors):
        query_string = f'https://financialmodelingprep.com/api/v3/historical-price-full/{ticker}?fro
        df = pd.read_csv(query_string)
        df['date'] = pd.to_datetime(df['date'])
        plt.plot(df['date'], df['volume'], label=f'Volume {ticker}', color=color, alpha=0.75)
    plt.title('Volume')
    plt.xlabel('Date')
    plt.ylabel('Volume')
    plt.legend()
    plt.tight_layout()
    plt.grid(True)
    return toImage(plt)
```

Figure 14: Volume Chart Construction

On each page, except for the home page, there is a "Home" button located at the bottom left corner. This button allows users to return to the main page with the analysis parameter input form.

The visual tools provided to users enable them to:

- Forecast the price trend of the selected medical product.
- Conduct a comparative analysis with other products in the same segment.
- Analyze the impact of informational factors on the pricing of medical products in the selected segment.
- Perform a comprehensive analysis of market information related to the selected medical product.
- Make an informed conclusion about the investment attractiveness of the chosen medical product**.**

5. Conclusion

This article explores the capabilities of modern technologies and describes the tools used to create a web application aimed at analyzing product prices in financial markets. The application includes features for constructing volatility charts, trading volume graphs, closing prices, and applying technical indicators. The factors considered in price movement analysis and the tools used for data visualization are analyzed. A detailed description of methods and techniques helps to understand how valuable information can be extracted from financial data and used for making effective investment decisions in the medical sector. The study of key indicators and the practical analysis of their application in the web application developed by the authors reveal new opportunities for enhancing the analysis and making well-informed investment decisions.

6. References

[1] Wang, Xin, Tapani Ahonen, and Jari Nurmi. "Applying CDMA technique to network-on-chip." IEEE transactions on very large scale integration (VLSI) systems 15.10 (2007): 1091-1100.

[2] P. S. Abril, R. Plant, The patent holder’s dilemma: Buy, sell, or troll?, Communications of the ACM 50 (2007) 36–44. doi:10.1145/1188913.1188915.

[3] S. Cohen, W. Nutt, Y. Sagic, Deciding equivalances among conjunctive aggregate queries, J. ACM 54 (2007). doi:10.1145/1219092.1219093.

[4] J. Cohen (Ed.), Special issue: Digital Libraries, volume 39, 1996.

[5] D. Kosiur, Understanding Policy-Based Networking, 2nd. ed., Wiley, New York, NY, 2001.

[6] D. Harel, First-Order Dynamic Logic, volume 68 of Lecture Notes in Computer Science, Springer-Verlag, New York, NY, 1979. doi:10.1007/3-540-09237-4.

[7] I. Editor (Ed.), The title of book one, volume 9 of The name of the series one, 1st. ed., University of Chicago Press, Chicago, 2007. doi:10.1007 3-540-09237-4.

[8] I. Editor (Ed.), The title of book two, The name of the series two, 2nd. ed., University of Chicago Press, Chicago, 2008. doi:10.1007/3-540-09237-4.

[9] A. Z. Spector, Achieving application requirements, in: S. Mullender (Ed.), Distributed Systems, 2nd. ed., ACM Press, New York, NY, 1990, pp. 19–33. doi:10.1145/90417.90738.

[10] B. P. Douglass, D. Harel, M. B. Trakhtenbrot, Statecarts in use: structured analysis and object-orientation, in: G. Rozenberg, F. W. Vaandrager (Eds.), Lectures on Embedded Systems, volume 1494 of Lecture Notes in Computer Science, Springer-Verlag, London, 1998, pp. 368–394. doi:10.1007/3-540-65193-4_29.
[11] D. E. Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.), Addison Wesley Longman Publishing Co., Inc., 1997.
[12] S. Andler, Predicate path expressions, in: Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages, POPL '79, ACM Press, New York, NY, 1979, pp. 226–236. doi:10.1145/567752.567774.
[13] S. W. Smith, An experiment in bibliographic mark-up: Parsing metadata for xml export, in: R. N. Smythe, A. Noble (Eds.), Proceedings of the 3rd. annual workshop on Librarians and Computers, volume 3 of LAC '10, Paparazzi Press, Milan Italy, 2010, pp. 422–431. doi:99.9999/woot07-S422.
[14] M. V. Gundy, D. Balzarotti, G. Vigna, Catch me, if you can: Evading network signatures with web-based polymorphic worms, in: Proceedings of the first USENIX workshop on Offensive Technologies, WOOT '07, USENIX Association, Berkley, CA, 2007.
[15] D. Harel, LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER, MIT Research Lab Technical Report TR-200, Massachusetts Institute of Technology, Cambridge, MA, 1978.
[16] Shanaida M, Jasicka Misiak I, Makowicz E, Stanek N, Shanaida V, Wieczorek PP. Development of high-performance thin layer chromatography method for identification of phenolic compounds and quantification of rosmarinic acid content in some species of the lamiaceae family. J Pharm Bioallied Sci. 2020;12(2):139-45.
[17] V. Martsenyuk, I. Andrushchak, A. Sverstiuk, A. Klos-Witkowska. On investigation of stability and bifurcation of neural network with discrete and distributed delays.) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018. Vol. 11127. LNCS, pp. 300 – 313. DOI: 10.1007/978-3-319-99954-8_26.
[18] V. Martsenyuk, A. Sverstiuk. An exponential evaluation for recurrent neural network with discrete delays. System Research and Information Technologies, 2019. Vol. 2. pp. 83 – 93. DOI: 10.20535/SRIT.2308-8893.2019.2.07
[19] O. Nakonechnyi, V. Martsenyuk, A. Sverstiuk. On Application of Kertesz Method for Exponential Estimation of Neural Network Model with Discrete Delays. Mechanisms and Machine Science. 2020. Vol. 70. pp. 165 – 176. DOI: 10.1007/978-3-030-13321-4_14.
[20] V. Martsenyuk, A. Klos-Witkowska, S. Dzyadevych, A. Sverstiuk. Nonlinear Analytics for Electrochemical Biosensor Design Using Enzyme Aggregates and Delayed Mass Action. Sensors 2022, 22(3), 980; https://doi.org/10.3390/s22030980.
[21] O. Saiapina, K. Berketa, A. Sverstiuk, L. Fayura, A. Sibirny, S. Dzyadevych, O. Soldatkin. Adaptation of Conductometric Monoenzyme Biosensor for Rapid Quantitative Analysis of L-arginine in Dietary Supplements. In Sensors. 2024. Vol. 24, Issue 14, p. 4672. https://doi.org/10.3390/s24144672.
[22] V. Martsenyuk, O. Soldatkin, A. Klos-Witkowska, A. Sverstiuk, K. Berketa. Operational stability study of lactate biosensors: modeling, parameter identification, and stability

analysis. In Frontiers in Bioengineering and Biotechnology. 2024. Vol. 12. Frontiers Media SA. https://doi.org/10.3389/fbioe.2024.1385459.

[23] A. Sverstiuk. Research of global attractability of solutions and stability of the immunosensor model using difference equations on the hexagonal lattice. Innovative Biosystems and Bioengineering. 2019. Vol. 3 (1), pp. 17 – 26. DOI: 10.20535/ibb.2019.3.1.157644.

[24] V. Martsenyuk, A. Sverstiuk, I. Andrushchak. Approach to the study of global asymptotic stability of lattice differential equations with delay for modeling of immunosensors. Journal of Automation and Information Sciences. 2019. Vol. 51 (2), pp. 58 – 71. DOI: 10.1615/jautomatinfscien.v51.i2.70.

[25] A. Nakonechnyi, V. Martsenyuk, A. Sverstiuk, V. Arkhypova, S. Dzyadevych. Investigation of the mathematical model of the biosensor for the measurement of α-chaconine based on the impulsive differential system CEUR Workshop Proceedings. 2020. Vol. 2762, pp. 209 – 217.

[26] V. Martsenyuk, O. Bahrii-Zaiats, A. Sverstiuk, S. Dzyadevych, B. Shelestovskyi. Mathematical and Computer Simulation of the Response of a Potentiometric Biosensor for the Determination of α-chaconine. CEUR Workshop Proceedings. 2023. Vol. 3468, pp. 1 – 11.

[27] V. Martsenyuk, A. Sverstiuk, A. Klos-Witkowska, K. Nataliia, O. Bagriy-Zayats, I. Zubenko. Numerical analysis of results simulation of cyber-physical biosensor systems. CEUR Workshop Proceedings. 2019. Vol. 2516, pp. 149 – 164.

[28] V. Martsenyuk, A. Sverstiuk, O. Bahrii-Zaiats, A. Kłos-Witkowska. Qualitative and Quantitative Comparative Analysis of Results of Numerical Simulation of Cyber-Physical Biosensor Systems. CEUR Workshop Proceedings. 2022. Vol. 3309, pp. 134 – 149.