

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА **Пояснювальна записка**

Другий (магістерський)
(рівень вищої освіти)

Розроблення системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств
(тема)

Виконав:
студент 2 курсу, групи АУПТМ-21-1

Пономаренко Д. С.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизоване управління технологічними процесами
(повна назва освітньої програми)

Керівник доц. Хрустальова С.В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАМ

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2022 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

19.12.2022

Пономаренко Дмитро Сергійович

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
 Кафедра _____ КІТАМ _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології _____
 Тип програми _____ Освітньо-професійна _____
 Освітня програма АУТП (Автоматизоване управління технологічними процесами) _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАМ _____
(підпис)

«__» _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Пономаренку Дмитру Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств

Затверджена наказом по університету від 07.11.2022 № 1463Ст

2. Термін подання студентом роботи до екзаменаційної комісії до 21.12.2022

3. Вхідні дані до роботи Таблиця відповідностей значень індикаторів і статусів; формули для розрахування коефіцієнтів стратегії вартісного інвестування.

4. Перелік питань, що потрібно опрацювати в роботі Аналіз існуючих систем підтримки прийняття рішень; розроблення структурної схеми та алгоритму роботи системи; розроблення системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстративний матеріал у вигляді презентації (*.pptx) – 20 с.

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	8.11.22	виконано
2	Аналіз сучасних систем-провайдерів фінансової інформації підприємств	12.11.22	виконано
3	Розроблення структурної схеми та алгоритму роботи системи	15.11.22	виконано
4	Програмна реалізація та експериментальні дослідження	03.12.22	виконано
5	Охорона праці	06.12.22	виконано
6	Оформлення пояснювальної записки	07.12.22	виконано
7	Перевірка керівником роботи	07.12.22	виконано
8	Нормоконтроль	16.12.22	виконано
9	Подання роботи на перевірку Інтернет-сервісом Unichек	17.12.22	виконано
10	Подання роботи на рецензію	18.12.22	виконано
11	Подання роботи на підпис зав. кафедри	19.12.22	виконано
12	Подання атестаційної роботи в ЕК	21.12.22	виконано

Дата видачі завдання 7.11.2022

Студент _____
(підпис)

Пономаренко Д. С.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

доц. Хрустальова С. В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 79 с., 1 табл., 39 рис., 1 дод., 20 джерел.

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ВАРТІСНЕ ІНВЕСТУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес прийняття рішень для управління фінансовим капіталом виробничих підприємств.

Предмет дослідження – програмний засіб для управління фінансовим капіталом виробничих підприємств.

Мета дослідження – підвищення ефективності прийняття рішень при управлінні фінансовим капіталом виробничих підприємств шляхом використання специфічних індикаторів для розрахунку відповідності підприємства стратегії вартісного інвестування.

Актуальність роботи полягає в тому, що наразі інвестиції є одним з найпопулярніших і найефективніших видів збереження та збільшення фінансового капіталу і отримання пасивного доходу. Першим кроком для аналізування потенційної інвестиції є фундаментальний аналіз підприємства, який дозволяє зробити первинні висновки щодо ведення бізнесу підприємства, що і є ґрунтом для прийняття рішення при капіталовкладенні. Автоматизація даного процесу є необхідністю, адже пошук даних підприємства для фундаментального аналізу є процесом з великими затратами часу, а людський фактор може грати ключову роль при аналізі великої кількості даних.

Упровадження системи підтримки прийняття рішень у вигляді програмного засобу, що автоматизує процес фундаментального аналізу підприємства на відповідність стратегії вартісного інвестування, допоможе приймати інвестиційні рішення не тільки власникам великого капіталу або фінансовим менеджерам виробничих підприємств, але й самостійним інвесторам, зменшить наявність людського фактору при розрахуванні необхідних для фундаментального аналізу індикаторів, а також спростить до мінімуму фундаментальний аналіз підприємства, що більше популяризує інвестування серед українців.

ABSTRACT

The explanatory note: 79 pages, 1 table, 39 figures, 1 addition, 20 sources.

DECISION SUPPORT SYSTEM, VALUE INVESTMENT, SOFTWARE.

The object of the study is the decision-making process for managing the financial capital of manufacturing enterprises.

The subject of the study is a software tool for managing the financial capital of manufacturing enterprises.

The purpose of the study is to increase the efficiency of decision-making in the management of the financial capital of manufacturing enterprises by using specific indicators to calculate the company's compliance with the value investment strategy.

The relevance of the work lies in the fact that currently investments are one of the most popular and effective ways to preserve and increase financial capital and receive passive income. The first step in analyzing a potential investment is a fundamental analysis of the company, which allows you to make initial conclusions about the business of the company, which is the basis for making a decision on capital investment. Automation of this process is a necessity, because searching for enterprise data for fundamental analysis is a time-consuming process, and the human factor can play a key role in the analysis of large amounts of data.

Implementation of a decision-making support system in the form of a software tool that automates the process of fundamental analysis of the enterprise for compliance with the value investment strategy will help to make investment decisions not only to owners of large capital or financial managers of manufacturing enterprises, but also to independent investors, will reduce the presence of the human factor when calculating the necessary for fundamental analysis of indicators, and will also simplify the fundamental analysis of the company to a minimum, which will make investing more popular among Ukrainians.

ЗМІСТ

Вступ.....	9
1 Аналіз існуючих систем підтримки прийняття рішень та методів управління фінансовим капіталом виробничих підприємств.....	10
1.1 Класифікація систем підтримки прийняття рішень, типи архітектури.....	10
1.1.1 Класифікація систем підтримки прийняття рішень.....	12
1.2 Аналіз сучасних систем-провайдерів фінансової інформації підприємств.....	14
1.3 Аналіз існуючих автоматизованих систем розрахунку показників підприємств на основі стратегії коштовного інвестування.....	24
1.4 Висновки до розділу.....	28
2 Розроблення структурної схеми та алгоритму роботи системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств.....	29
2.1 Розроблення структурної схеми системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств.....	29
2.2 Розроблення алгоритму роботи системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств.....	30
2.3 Технічна схема роботи СППР.....	32
2.4 Розрахунок коефіцієнтів.....	33
2.5 Вартісне інвестування.....	38
2.6 Висновки до розділу.....	39
3 Розроблення системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств у вигляді програмного засобу.....	40
3.1 Вибір середовища програмування.....	40
3.2 Етапи розробки програмного веб-додатку.....	41
3.3 Вибір мови програмування для розробки програмного засобу.....	46
3.4 Вибір frontend-фреймворку для розробки програмного засобу.....	48

3.5	Додаткові засоби для розробки програмного додатку.....	51
3.6	Серверна частина програмного засобу.....	53
3.7	Клієнтська частина програмного засобу.....	54
3.8	Інтерфейс програмного засобу.....	65
3.9	Висновки до розділу.....	70
4.	Охорона праці.....	71
4.1	Аналіз умов праці в лабораторії.....	71
4.2	Промислова безпека у лабораторії.....	71
	Висновки.....	74
	Список використаних джерел.....	76
	Додаток А Демонстраційний матеріал у вигляді презентації.....	79

ВСТУП

Збереження фінансового капіталу підприємства є дуже важливою частиною будь-якого бізнесу. Одним із способів зберігання, і навіть збільшення, капіталу є інвестування. На сьогоднішній день існує безліч інструментів інвестування коштів: депозити, пенсійні фонди, дорогоцінні метали, нерухоме майно, державні облігації, криптовалюти тощо. Але з усіх інструментів виділяються акції великих підприємств. Інвестування в акції може бути дуже прибутковим, однак це не означає що купівля будь-яких цінних паперів компаній є надійним вкладенням [3]. Важливою частиною інвестування є фундаментальний аналіз цінних паперів підприємств, які потенційно можуть бути придбані в інвестиційний портфель. Автоматизація цього процесу підвищить якість аналізу, виключить людський фактор з прорахунків, а також зекономить багато часу, що також є корисним при аналізуванні сотень потенційних елементів інвестиційного портфелю.

При проектуванні систем такого класу потрібні актуальні, і в той же час, перевірені роками методи оцінки фінансових показників підприємств. Також, оскільки керуючі мають великі зобов'язання перед працівниками підприємства, необхідно бути впевненими в надійності критеріїв, які аналізує система. Для розроблення програмного засобу, за основу було обрано стратегію – стратегію вартісного інвестування, яку винайшов відомий економіст, відомий інвестор, автор та викладач Колумбійського університету Бенджамін Грем, якого самий відомий інвестор всіх часів Воррен Баффет вважає своїм вчителем, а книгу Розумний інвестор – найкращою з усіх, яка б коли-небудь була написана про інвестиції. З тих часів матеріал регулярно перевидається, а до кожної з глав книги додаються коментарі, які є актуальними для сьогодення. Українською мовою перекладена та опублікована у 2019 році видавництвом Наш формат [4].

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ТА МЕТОДІВ УПРАВЛІННЯ ФІНАНСОВИМ КАПІТАЛОМ ВИРОБНИЧИХ ПІДПРИЄМСТВ

1.1 Класифікація систем підтримки прийняття рішень, типи архітектури

Система підтримки прийняття рішень (СППР) – інтерактивна комп'ютерна автоматизована система (програмний комплекс), що призначена для допомоги та підтримки різних видів діяльності людини при прийнятті рішень стосовно розв'язання структурованих або неструктурованих проблем [5]. Застосування СППР забезпечує виконання ґрунтовного та об'єктивного аналізу предметної області при прийнятті рішень в складних умовах.

Прийняття рішення – це процес вибору найбільш преференційного рішення з множини допустимих рішень або упорядкування множини рішень [6]. Прийняття рішень можливе на підставі знань про об'єкт управління, процеси, що в ньому відбуваються і можуть відбутися з перебігом часу, а також за наявності множини показників, що характеризують ефективність та якість прийнятого рішення. Тобто необхідні адекватна модель об'єкту і модель прийняття та оцінювання прийнятого рішення. Під моделлю прийняття рішень мається на увазі формальне подання поставленої задачі та процесу прийняття рішень.

Архітектура СППР містить в собі три основні підсистеми [7]:

– інтерфейс користувача, який дає змогу особі, що приймає рішення (ОПР), взаємодіяти з системою, використовуючи різні компоненти вводу інформації, такі як пошукові поля;

– підсистеми, призначеної для зберігання, керування, вибору, відображення та аналізу даних;

– підсистеми, яка містить набір моделей для забезпечення відповідей на запити користувача та інших аналітичних задач.

Комп'ютерні системи підтримки прийняття рішень признанні розв'язувати задачі для користувачів, а тому невіддільною складовою їхньої роботи має бути точне використання параметрів, добутих від користувача, і повернення йому результатів роботи системи. У розробленому програмному додатку для отримання даних від користувача було додане пошукове поле, в яке користувач вводить ключове слово для пошуку бажаного підприємства. При цьому якщо система працює коректно, але подає результати у спосіб, який є незручним для користувача, то роботу такої системи не можна вважати задовільною. Загальне побажання користувачів полягає в тому, щоб зі складними інформаційними системами можна було працювати легко і інтуїтивно-зрозуміло, оминаючи довгочасний і дорогий етап навчання. Усе це зумовлює ряд вимог та особливостей побудови інтерфейсу користувача СППР.

Архітектура СППР визначається характером взаємодії основних її складових – інтерфейсу користувача; бази та сховища даних, документів і правил; моделей і аналітичних інструментів; інфраструктури комунікацій і мереж, а також елементів цих частин. Серед факторів, які впливають на вибір конкретної архітектури СППР, можна виокремити такі [8]:

- необхідність подальшого розвитку комп'ютерної системи та адаптації, зокрема, за рахунок включення в неї наявних програмних засобів;

- застосування еволюційного підходу до розвитку СППР.

Мережа є важливим елементом інфраструктури. У сучасних СППР широко застосовуються такі головні мережеві технології:

- Інтернет (Internet), який уможливорює з'єднання окремих індивідів у планетарному масштабі;

- Екстранет (Extranet) – зв'язок окремих компаній між собою;

- Інтранет (Intranet) – з'єднання індивідів усередині компаній.

Можна поєднати головні компоненти СППР – інтерфейс користувача, базу даних, моделі й аналітичні інструментальні засоби, а також мережеву структуру СППР у загальну архітектуру СППР. Головним компонентом у проекті СППР є

інтерфейс користувача. До інструментальних засобів для побудови інтерфейсу користувача належать:

- симулятори інтерфейсу;
- СППР-генератори;
- інструментальні засоби запиту і звітів;
- пакет розроблення кінцевого користувача (front-end).

База даних СППР є сукупністю даних, які організуються для легкого доступу до них і аналізу. Великі бази даних у СППР масштабу підприємств (корпоративних СППР) часто називають сховищами даних або вітринами даних.

1.1.1 Класифікація систем підтримки прийняття рішень

Єдиної класифікації СППР зараз не існує. Існує велика кількість СППР, що відрізняються одна від одної цілями, призначенням, предметними галузями. Тому для уніфікації розуміння поняття СППР, підходів до розробки СППР та їхнього використання проводяться класифікації СППР.

На рівні користувача виділяють такі види СППР:

- активна – може зробити пропозицію, яке рішення варто вибрати;
- пасивна – допомагає у процесі ухвалення рішення, але не може внести пропозицію, яке рішення прийняти;
- кооперативна – дозволяє ОПР змінювати, поповнювати або поліпшувати рішення, пропоновані системою, надсилаючи потім ці зміни в систему для перевірки.

На технічному рівні виділяють такі види СППР:

- СППР рівня підприємства – підключена до великих сховищ даних і обслуговує багатьох менеджерів підприємства;
- персональна настільна СППР – мала система, що обслуговує лише один комп'ютер користувача.

На концептуальному рівні відрізняють такі типи СППР:

- керована повідомленнями (Communication-Driven DSS) – підтримує групу користувачів, що працюють над виконанням загальної задачі;

- керована даними (Data-Driven DSS, Data-oriented DSS) – в основному орієнтується на доступ і маніпуляції з даними;
- керована документами (Document-Driven DSS) – здійснює пошук і маніпулювання неструктурованою інформацією, заданої в різних форматах;
- керована знаннями (Knowledge-Driven DSS) – забезпечує рішення задач у виді фактів, правил, процедур;
- керована моделями (Model-Driven DSS) – забезпечує доступ і маніпуляції з математичними моделями (статистичними, фінансовими, оптимізаційними, імітаційними).

Необхідно відзначити, що деякі OLAP-системи, які дозволяють здійснювати складний аналіз даних, можуть бути віднесені до так званих гібридних СППР, що забезпечують моделювання, пошук і оброблення даних та відповідають властивостям декількох видів СППР. Залежно від даних, з якими ці системи працюють, СППР умовно можна поділити на оперативні та стратегічні.

Оперативні СППР призначені для негайного реагування на зміни поточної ситуації у керуванні фінансово-господарськими процесами компанії, об'єднання, галузі чи держави. Такі СППР називають Виконавчі Інформаційні Системи (Executive Information Systems). Зазвичай системи надають кінцеві множини звітів, побудовані за даними із транзакційної інформаційної системи оперативного обліку підприємства. Вони забезпечують адекватне відображення в режимі реального часу основних аспектів виробничої і фінансової діяльності підприємства.

Для оперативних СППР характерними є такі риси [9]:

- звіти ґрунтуються на стандартних, для організації, запитах, кількість яких відносно невелика;
- СППР подає звіти у максимально зручному вигляді, що включає надання, окрім таблиць, ділової графіки, мультимедійної інформації тощо;
- СППР зазвичай орієнтовані на конкретну сферу, наприклад, фінанси, маркетинг, керування ресурсами.

Стратегічні СППР орієнтовані на аналіз значних обсягів різномірної інформації, що збирають з різних джерел. Найважливішою метою цих СППР є пошук найбільш раціональних варіантів розвитку бізнесу компанії із урахуванням впливу різних факторів, зокрема кон'юнктури цільових, для компанії, ринків, зміни фінансових ринків і ринків капіталів, зміни у законодавстві тощо.

Такі СППР припускають достатньо глибоке спеціальне перетворення даних для їх зручного використання у процесі прийняття рішень. Невід'ємним компонентом цих СППР є правила прийняття рішень, які на основі агрегованих даних надають можливість менеджерам компанії обґрунтовувати свої рішення, використовувати фактори стійкого росту бізнесу компанії і знижувати ризики. Стратегічні СППР будують на принципах багатовимірного подання та аналізу даних у системах OLAP (Online Analytical Processing).

1.2 Аналіз сучасних систем-провайдерів фінансової інформації підприємств

Існує велика кількість систем, які надають користувачам актуальну фінансову інформацію підприємств. Такі СППР пропонують велику кількість даних, графіків та історичної інформації не лише підприємств, а й державних облігацій, криптовалют, коштовних металів тощо. Більш того, гіганти-лідери у цій сфері пропонують також новини, зв'язані з бізнесом, фінансами та економікою. Довідкова інформація включає в себе котирування і рейтинги цінних паперів, прес-релізи і фінансові звіти компаній. Деякі СППР містять в собі тематичні форуми, і навіть пропонують ряд послуг для управління персональною фінансовою інформацією.

Однією з найбільших систем для надавання фінансової інформації є finance.yahoo.com [10] який є яскравим представником-гігантом у своїй сфері. Окрім програмного засобу у браузері, система працює на мобільних платформах iOS та Android. На порталі (рисунок 1.1), можливо знайти всю необхідну

інформацію, яка може знадобитись не тільки для фундаментального аналізу підприємства, але й для більш комплексних та вузько направлених розрахунків.

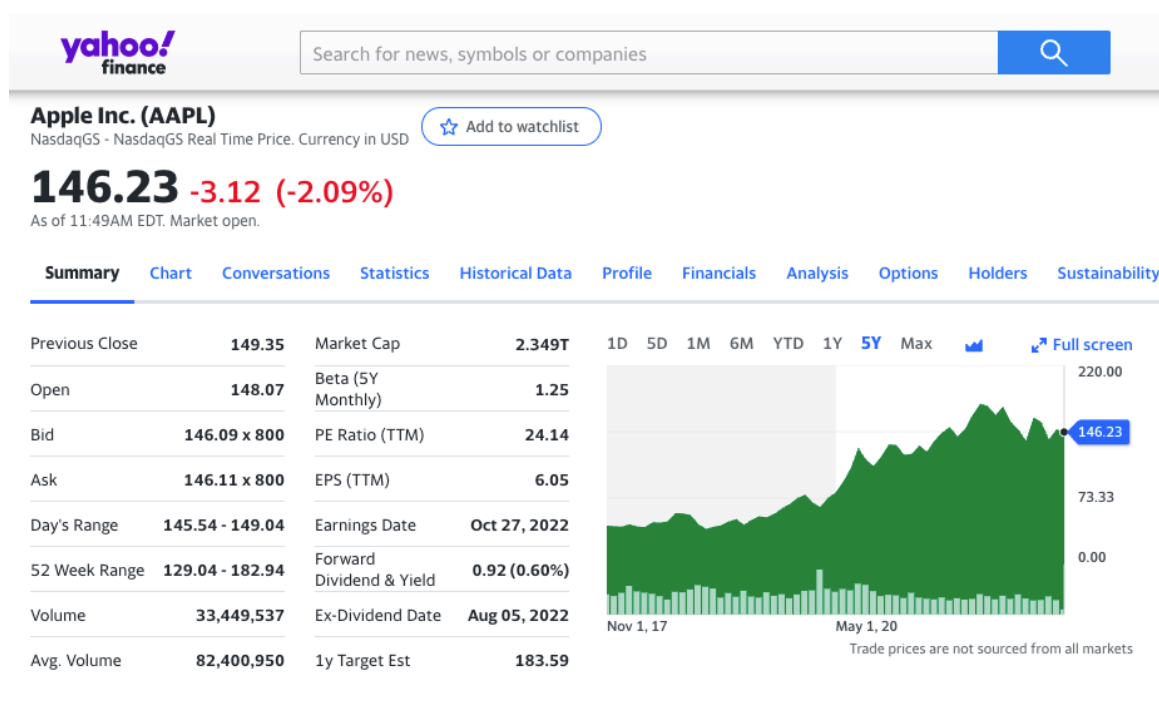


Рисунок 1.1 – Первинні дані системи finance.yahoo.com по запиту Apple Inc. за 27.10.2022

Використовуючи пошукове поле, можемо обрати необхідне підприємство. За відповідним запитом ресурс відобразить велику кількість фінансових даних підприємства, основні з них:

- **Summary** (резюме), містить в собі дані про актуальну ціну акції, діапазон цін за останній рік, ринкову капіталізацію підприємства, графік коливання ціни за вказаний період тощо;

- **Statistics** (статистика), містить в собі інформацію щодо фінансових показників компанії за фіскальний рік, ефективність менеджменту, довідку про доходи підприємства, дивіденди тощо;

- **Historical Data** (історичні дані), містить в собі інформацію щодо того як ціна однієї акції змінювалась на протяжності періоду, який можна завдати власноруч;

– Profile (профіль), містить в собі інформацію щодо головного офісу компанії (країна, конкретна адреса, поштовий індекс та контактний телефон, веб-сайт), в якому секторі економіки підприємство займає місце, індустрію, кількість співробітників. Також відображені імена, а також деяка фінансова інформація топ-менеджерів підприємства;

– Analysis (розбір), містить в собі інформацію щодо потенційних фінансових показників підприємства на основі поточних даних;

– Holders (власники), містить в собі інформацію щодо поточних власників цінних паперів підприємства, а також список найбільших інституційних власників (кількість акцій, відсоток від глобальної кількості, грошовий еквівалент та дату останнього оновлення інформації).

Також в системі finance.yahoo.com користувач може отримати інформацію щодо актуальних економічних новин, список фінансових інструментів, які зазнали найбільшого зросту або падіння за останні 24 години та отримати деякі послуги для автоматизації особистих фінансів: створити особисте портфоліо, яке може відстежувати актуальні дані щодо позицій портфелю, а також отримати інформацію щодо податків та виходу на пенсію.

Дана СППР, хоч і надає необхідні змінні для прорахунків всіх показників стратегії коштовного інвестування, але не має конкретно виділених індикаторів стратегії, які є фундаментом для пошуку недооцінених акцій.

Інший ресурс, який займає велику частину ніші фінансових ресурсів, це Investing.com [11], який є фінансовою платформою, що надає інформацію щодо акцій підприємств, ф'ючерсів, опціонів, аналітики, а також економічних новин (рисунок 1.2). Окрім програмного засобу у браузері, система працює на мобільних платформах iOS та Android. Продукт наразі не має української локалізації.

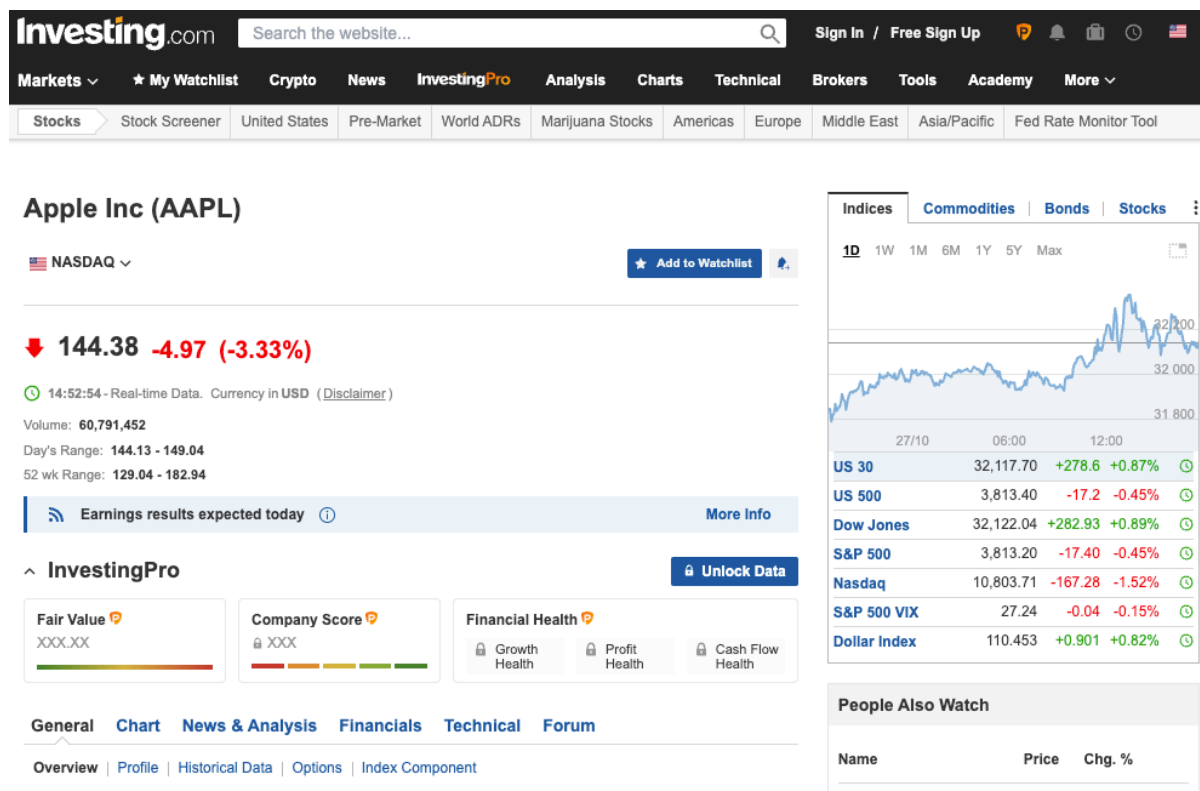


Рисунок 1.2 – Первинні дані системи investing.com по запиту Apple Inc. за 27.10.2022

За аналогією з попередньо описаним ресурсом Yahoo! Finance, на першій сторінці системи маємо пошукове поле, за допомогою якого можемо знайти інформацію про бажане підприємство. Після необхідного запиту, бачимо на екрані відображення знайдених даних про підприємство Apple Inc., а також показники найбільших фондових індексів, інформацію за схожими по запиту тікерами, а також навігаційну зону, за допомогою якою і є можливість відобразити інформацію щодо знайденого підприємства (рисунок 1.3):

The screenshot shows the Yahoo Finance page for Apple Inc. (AAPL). The current stock price is 146.18, down 3.18 (-2.13%) as of 11:47AM EDT. The page is divided into several sections: Valuation Measures, Financial Highlights, Trading Information, and Share Statistics. The Valuation Measures section includes Market Cap (2.40T), Enterprise Value (2.47T), Trailing P/E (24.69), Forward P/E (23.70), PEG Ratio (2.69), Price/Sales (6.34), Price/Book (41.31), Enterprise Value/Revenue (6.38), and Enterprise Value/EBITDA (18.77). The Financial Highlights section shows the Fiscal Year Ends on Sep 24, 2021, and the Most Recent Quarter (mrq) on Jun 24, 2022. The Trading Information section includes Stock Price History with Beta (1.25), 52-Week Change (-2.11%), S&P500 52-Week Change (-16.66%), 52 Week High (182.94), 52 Week Low (129.04), 50-Day Moving Average (152.76), and 200-Day Moving Average (156.99). The Share Statistics section shows Avg Vol (3 month) (82.4M), Avg Vol (10 day) (83.78M), Shares Outstanding (16.07B), Implied Shares Outstanding (N/A), and Float (16.05B).

Valuation Measures ⁴		Trading Information	
Market Cap (intraday)	2.40T	Stock Price History	
Enterprise Value	2.47T	Beta (5Y Monthly)	1.25
Trailing P/E	24.69	52-Week Change ³	-2.11%
Forward P/E	23.70	S&P500 52-Week Change ³	-16.66%
PEG Ratio (5 yr expected)	2.69	52 Week High ³	182.94
Price/Sales (ttm)	6.34	52 Week Low ³	129.04
Price/Book (mrq)	41.31	50-Day Moving Average ³	152.76
Enterprise Value/Revenue	6.38	200-Day Moving Average ³	156.99
Enterprise Value/EBITDA	18.77	Share Statistics	
Financial Highlights		Avg Vol (3 month) ³	82.4M
Fiscal Year		Avg Vol (10 day) ³	83.78M
Fiscal Year Ends	Sep 24, 2021	Shares Outstanding ⁵	16.07B
Most Recent Quarter (mrq)	Jun 24, 2022	Implied Shares Outstanding ⁶	N/A
		Float ⁸	16.05B

Рисунок 1.3 – Фундаментальні характеристики та індикатори підприємства Apple Inc., надані СППР finance.yahoo.com 27.10.2022

– General (головна), містить інформацію щодо поточної ціни акції, діапазон ціни акції за останній рік, ринкову капіталізацію підприємства, кількість акцій в обороті тощо;

– Chart (графік), на якому відображена ринкова ціна акції (вісь Y) та дата (вісь X) за умовчанням, а також індикатор виплати дивідендів, сплітів акції, а також додаткові динамічні можливості графіку (зміна способу відображення графіку, інструменти для технічного аналізу графіку, можливість додавання інших індикаторів);

– News & Analysis (новини і розбір), містить інформацію щодо останніх новин, які пов'язані зі знайденим підприємством, а також статті-розбори та

аналітика експертів, які за допомогою технічного аналізу та інших факторів намагаються спрогнозувати ціну акцій підприємства;

– Financials (фінанси), містить фундаментальну фінансову інформацію підприємства, таку як довідка про прибуток, бухгалтерський звіт, рух грошових активів.

На платформі [investing.com](https://www.investing.com) (рисунок 1.4), окрім фінансової інформації підприємства, користувач може знайти інформацію щодо державних біржових інвестиційних фондів та облігацій, криптовалют, статті загального аналізу ринку або аналізу по конкретному інвестиційному інструменту від запрошених експертів, графіки, інформацію щодо брокерів, інформацію щодо інвестиційних конференцій та вебінарів, а також додатковий платний контент з розширеними можливостями для менеджменту особистого інвестиційного портфелю, який містить в собі поради щодо потенційних інвестицій, інструменти для технічного аналізу з власною базою даних для збереження напрацьованого, а також графіки з додатковими індикаторами для розширеного аналізу фінансового звіту підприємства.

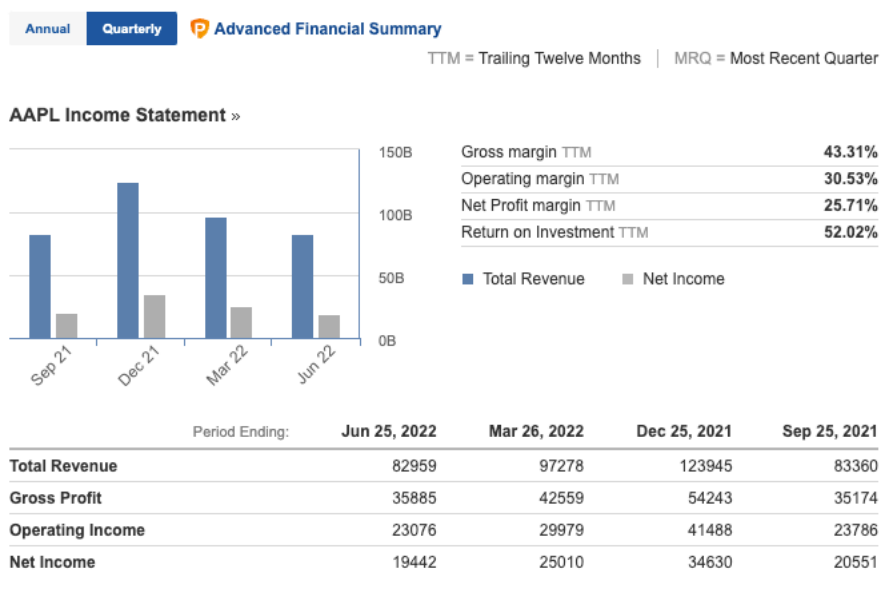


Рисунок 1.4 – Довідка про прибуток підприємства Apple Inc., надані системою [investing.com](https://www.investing.com) 27.10.2022

Необхідно зазначити, що даний розширений інструмент (рисунок 1.5) дозволяє прорахувати всі показники, які розраховуються у розробленому програмному засобі.

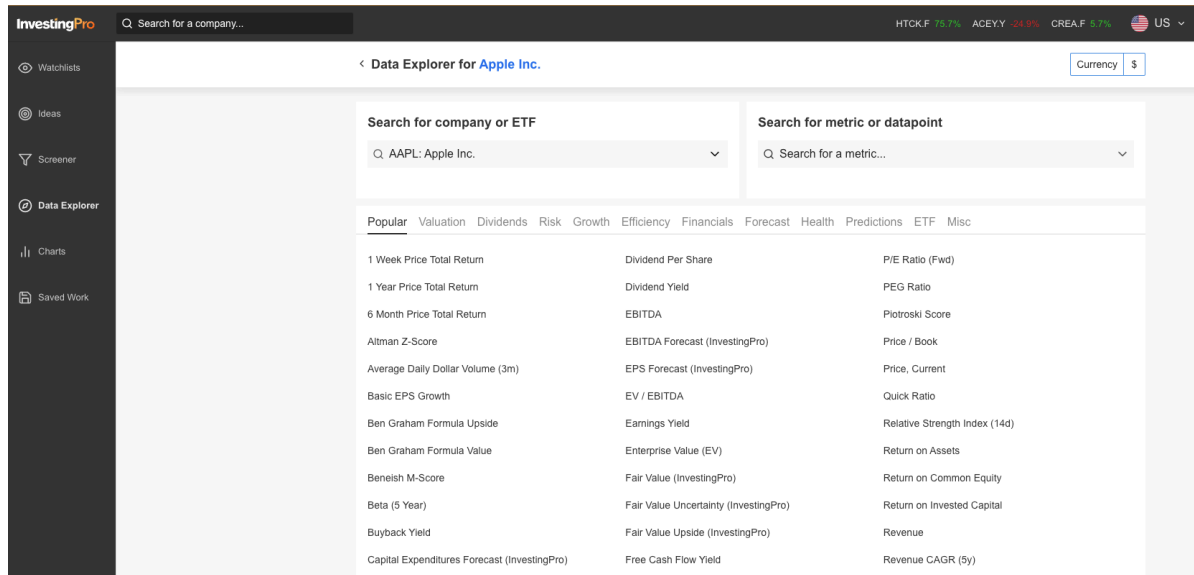


Рисунок 1.5 – Розширені можливості для фінансового аналізу підприємства системи investing.com

Проте, існує ряд недоліків, які є суттєвими для розроблення унікальної системи підтримки прийняття рішень, адаптованої під реалію сучасних українських підприємців і інвесторів:

- прорахунок статусу кожного індикатора. Оскільки основною сферою діяльності потенційних користувачів не є інвестування та економічна галузь взагалі, були розроблені статуси для кожного індикатора (рекомендований, задовільний, незадовільний), які допомагають прийняти рішення користувачу на основі таблиці відповідностей значень індикаторів і статусів, в якій рекомендовані значення були взяті з нормативів самого Бенджаміна Грема, а задовільні значення є мінімально дозволеними у прорахунках;

- відображення необхідної інформації. Система investing.com надає велику кількість комплексної інформації, в тому числі графіки та прорахунки складних фінансових індикаторів, які в рамках фундаментального аналізу не розглядаються. Розроблений програмний засіб дозволяє проаналізувати

підприємство фундаментально, тим самим відфільтрувати велику частку фінансово не здорових підприємств;

- інтуїтивно-зрозумілий інтерфейс. На відміну оглянутого investing.com, розроблена система має зручний і мінімізований інтерфейс виключно для аналізу підприємства з точки зору стратегії кошовного інвестування. Кожна функція системи зрозуміла для користувача будь-якого програмного засобу або додатку;
- українська локалізація. Оскільки програмний додаток розроблявся для використання українськими підприємцями та інвесторами, важливо щоб розроблена система надавала послуги державною мовою, зрозумілою усіма незалежно від володіння іноземною мовою.

Наступний ресурс, який користується популярністю серед аналітиків, це finviz.com [12]. Ця система надає інформацію щодо фінансової аналітики підприємств, ф'ючерсів, криптовалют, а також новин, які можуть впливати на фінансові ринки. Найбільшої популярності набув через відображення стану фондового ринку через інтерактивну мапу (рисунком 1.6), яка адаптує розмір кожного підприємства в залежності від сектору економіки та долі капіталізації на ринку.

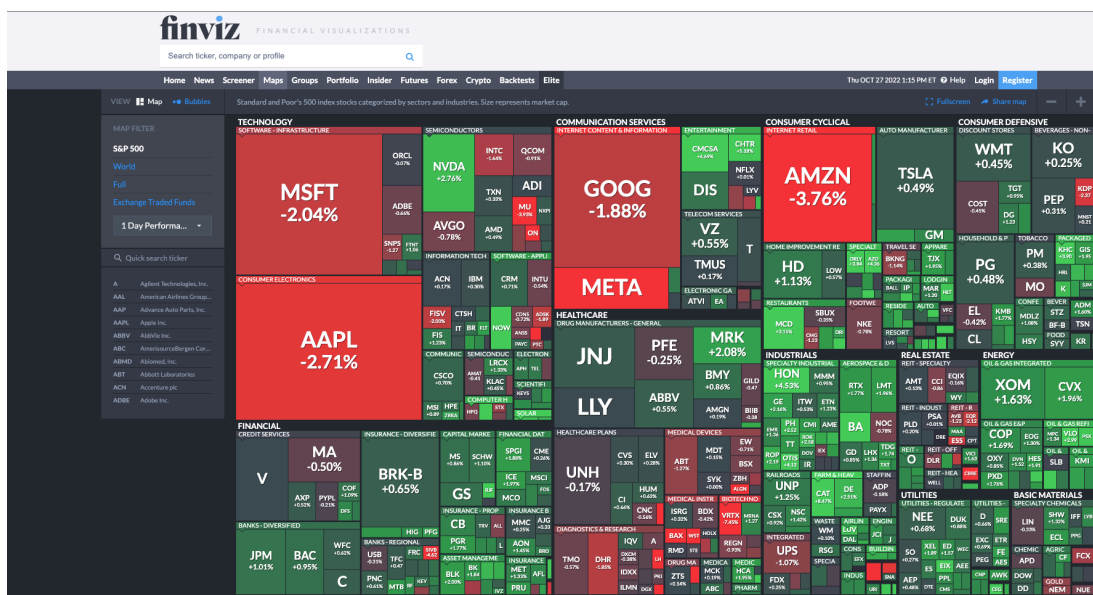


Рисунок 1.6 – Мапа фондового ринку, представлена ресурсом finviz.com

27.10.2022

Також програмний засіб надає графік, який будується на основі актуальних даних, динамічно підставляючи значення осей тими даними, які були обрані користувачем (рисунок 1.7). Є можливість відфільтрувати дані за деякими параметрами, такими як: сектор, індустрія, ринкова капіталізація та середній об'єм. Кожне підприємство відображається на графіку у вигляді кульки, розмір якої є відносним і залежить від, за замовчанням, ринкової капіталізації підприємства. Колір кульки відповідає за сектор економіки, в якій представлено підприємство. Також є можливість звужувати та розширяти осі, регулюючи діапазон відповідно вибраного значення осі. Наводячи на конкретну кульку, отримуємо графік руху ціни акції та значення, які були раніше вибрані для осі Y та розміру кульки.

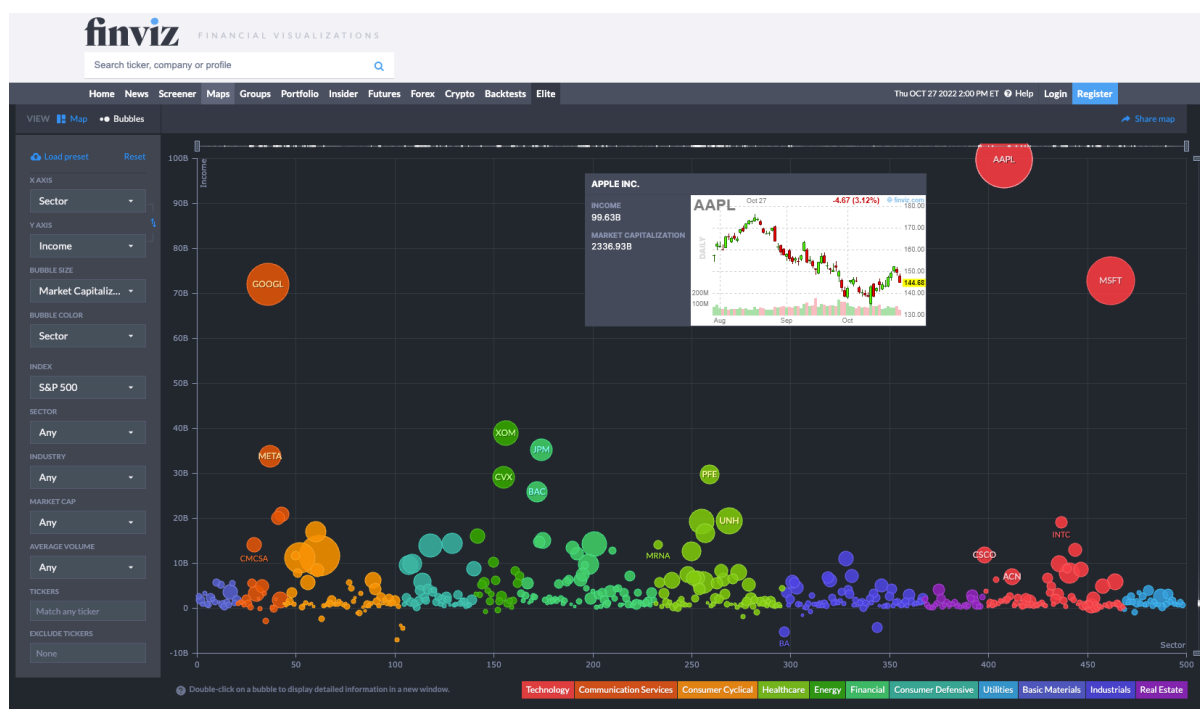


Рисунок 1.7 – Графік фондового ринку, представлений ресурсом finviz.com

27.10.2022

Ресурс, звісно, надає розширену інформацію по конкретному підприємству. Для цього необхідно двічі натиснути на відповідну геометричну фігуру на графіку. На екрані (рисунок 1.8) зображений графік ціни акції, який можна змінити візуально, а також змінити період часу, який відображає графік.

Нижче відображена таблиця з додатковими фінансовими показниками та прорахованими індикаторами, більш того, деякі з них мають статус (відображаються зеленим або червоним кольором), який вказує на фінансове здоров'я цього параметру.

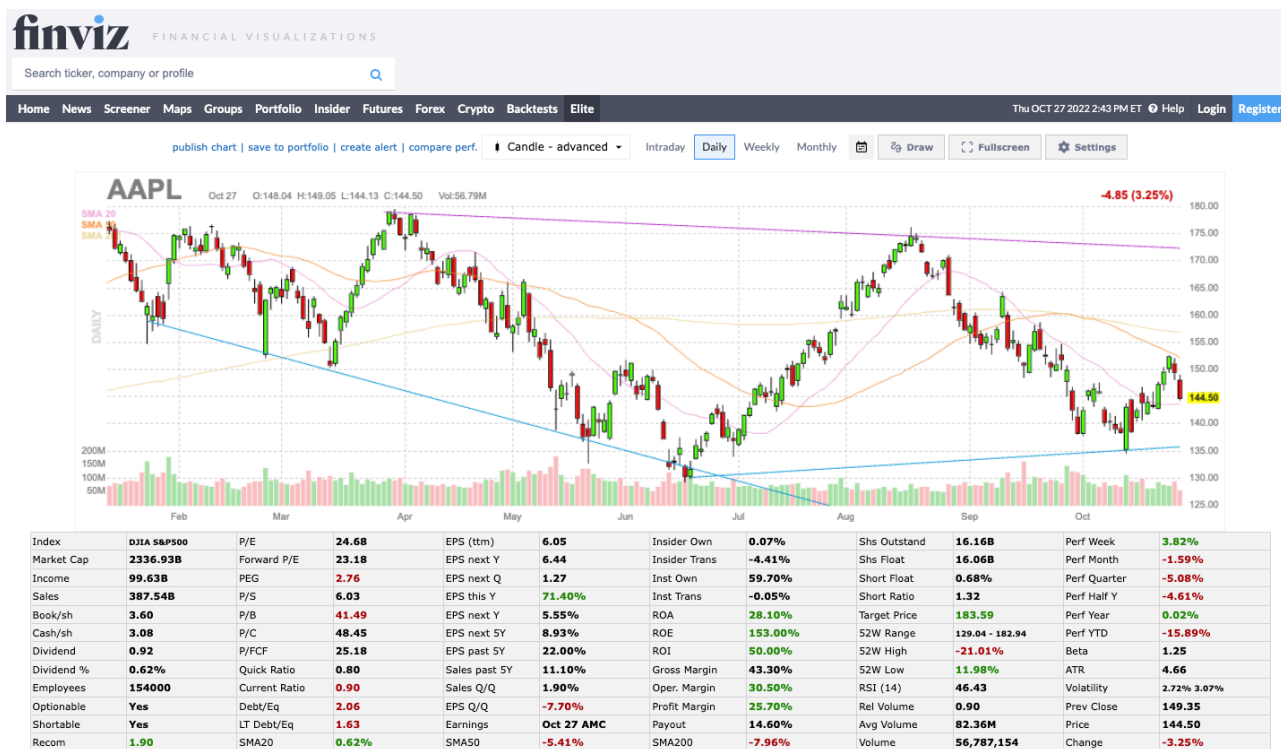


Рисунок 1.8 – Розширені дані системи finviz.com по запиту Apple Inc. за 27.10.2022

Програмний засіб finviz.com є зручною системою для перегляду фундаментальної фінансової інформації, а статуси деяких показників якісно виділяє систему серед інших конкурентів, які були оглянуті вище, хоча ті і пропонують більшу кількість даних підприємств. finviz.com має більш мінімізований інтерфейс, що є перевагою для поставлених задач розробленого додатку. Проте, недостатність деяких фінансових та історичних даних, які необхідні для прорахунку відповідності підприємства до стратегії коштовного інвестування, робить неможливим використання системи для цієї конкретної задачі. Відсутність української локалізації є спільним недоліком усіх оглянутих СППР.

1.3 Аналіз існуючих автоматизованих систем розрахунку показників підприємств на основі стратегії коштовного інвестування

На сьогоднішній день, у широкому доступі немає систем, які б аналізували підприємство комплексно і виключно за стратегією коштовного інвестування. На даному етапі були розглянуті системи, які частково або опосередковано розраховували той чи інший показник стратегії окремо. Це і є головним недоліком нижчеописаних систем, бо кожний індикатор не вказує на надійність інвестування в підприємство, проте комплексність і рекомендованість усіх показників є надійністю підприємства для довгострокового інвестування у нього.

Програмний засіб dqydj.com [13] пропонує своїм користувачам велику кількість калькуляторів, які прораховують індикатори в таких напрямках як теорія оцінки, банківська ліквідність, прибутку, чистої коштовності на ін. Також система пропонує калькулятори оцінки підприємств у деяких показниках, один з них це Graham Rule Calculator (мається на увазі число Грема). Для того, щоб прорахувати число Грема, використовуючи даний калькулятор (рисунок 1.9), необхідно власноруч ввести поточну ціну однієї акції підприємства, балансову вартість однієї акції, а також прибуток на акцію. Мультиплікатори балансової вартості та прибутку на акцію встановлюються за умовчанням на рекомендовані Гремом значення автоматично, але є можливість змінити значення власноруч.

Результатом розрахунків даного калькулятора є значення справедливою ціни акції (Graham Number / Fair Value) згідно змінних, які були введені користувачем, а також потенціал зросту акції з поточної ціни до справедливої ціни однієї акції підприємства.

Graham Number Calculator for Stock Value Screening

Graham Number Calculator

Stock Price

Book Value Per Share

Earnings Per Share

Optional: Change the Calculation Weightings

Book Value Multiplier EPS Multiplier

Graham Number / Fair Value

Over / Under Value Percentage

Рисунок 1.9 – Калькулятор числа Грема у системі dqudj.com

Оглянутий калькулятор, окрім розрахованого значення, також відображає кольоровий статус результату, що також позитивно впливає на досвід використання користувача і допомагає швидше прийняти рішення.

Система gurufocus.com [14] дозволяє прорахувати деякі показники зі стратегії кошовного інвестування, такі як число Грема (рисунок 1.10), коефіцієнт Грема (рисунок 1.11), балансову вартість (рисунок 1.12) тощо.

Apple's Graham Number for the **quarter** that ended in **Jun. 2022** is calculated as

```

Graham Number
= sqrt of (22.5 * Tangible Book per Share * EPS without NRI (TTM))
= sqrt of (22.5 * 3.61 * 6.06)
= 22.19

```

Рисунок 1.10 – Прорахунок числа Грема на платформі gurufocus.com підприємства Apple Inc. за 28.10.2022

Платформа пропонує розрахунок коефіцієнтів та числа Грема, отримуючи дані за звіти, які були надані за кінцевий фінансовий рік та кінцевий квартал.

Apple Net Current Asset Value Calculation

Apple's Net Current Asset Value (NCAV) per share for the fiscal year that ended in Sep. 2021 is calculated as

$$\begin{aligned}
 &\text{Net Current Asset Value Per Share (A: Sep. 2021)} \\
 &= (\text{Total Current Assets} - \text{Total Liabilities} - \text{Minority Interest} - \text{Preferred Stock}) / \text{Shares Outstanding (EOP)} \\
 &= (134836 - 287912 - 0 - 0) / 16426.8 \\
 &= -9.32
 \end{aligned}$$

Рисунок 1.11 – Прорахунок коефіцієнту Грема на платформі gurufocus.com підприємства Apple Inc. за 28.10.2022

Система також надає інформацію щодо балансової вартості підприємства, що так саме є важливим показником для розрахунку відповідності підприємства до стратегії коштовного інвестування.

Apple Current Ratio Calculation

The current ratio is mainly used to give an idea of the company's ability to pay back its short-term liabilities with its short-term assets.

Apple's Current Ratio for the fiscal year that ended in Sep. 2021

$$\begin{aligned}
 \text{Current Ratio (A: Sep. 2021)} &= \text{Total Current Assets (A: Sep. 2021)} / \text{Total Current Liabilities (A: Sep. 2021)} \\
 &= 134836 / 125481 \\
 &= 1.07
 \end{aligned}$$

Apple's Current Ratio for the quarter that ended in Jun. 2022 is calculated as

$$\begin{aligned}
 \text{Current Ratio (Q: Jun. 2022)} &= \text{Total Current Assets (Q: Jun. 2022)} / \text{Total Current Liabilities (Q: Jun. 2022)} \\
 &= 112292 / 129873 \\
 &= 0.86
 \end{aligned}$$

Рисунок 1.12 – Прорахунок балансової вартості на платформі gurufocus.com підприємства Apple Inc. за 28.10.2022

На першому екрані по запиті на бажане підприємство та число Грема (рисунок 1.13), бачимо опис самого коефіцієнта, ціну акції та прораховане число Грема на кінець останнього кварталу. Відображена базова статистика щодо прорахунків обраного коефіцієнту за останні 10 років, а також найвищий

показник за історичний період, до якого є доступ у системи. Окрім цього, система надає доступ до аналізу багатьох інших коефіцієнтів, які згруповані по основним групам, таким як: експлуатаційні дані, фундаментальні дані, коштовні коефіцієнти, прибутковість, ціна, дивіденди, довідка про прибуток, бухгалтерський баланс, рух грошей та технічні індикатори. На відміну від системи dqudj.com, gurifocus.com не потребує від користувача власноручне введення змінних для прорахунків коефіцієнту, а вже має ці дані і відображає автоматично прорахований коефіцієнт. Також програмний засіб відображає формулу, за якою розраховується результат, а також значення всіх змінних, які використовуються для розрахунків.

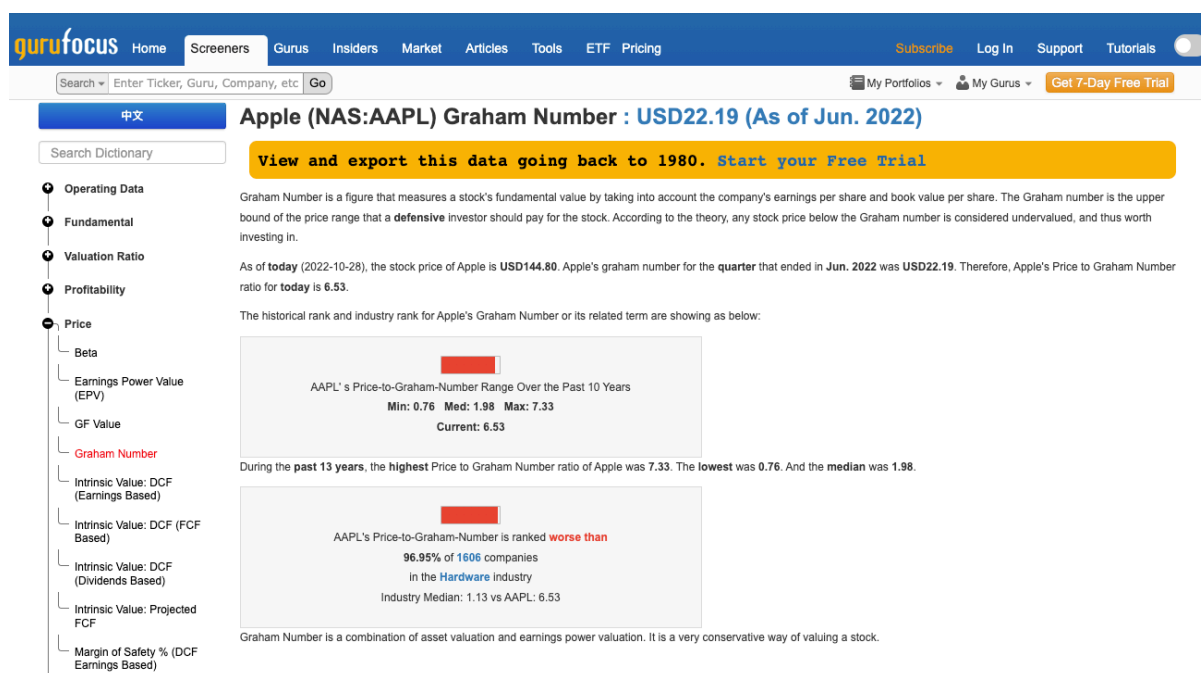


Рисунок 1.13 – Первинне вікно СППР gurifocus.com за запитом Apple Inc. за 28.10.2022

Система так само, як і попередні коефіцієнти, прораховує коефіцієнти в залежності від даних кінцевого фінансового року та кінцевого кварталу, що може бути корисно для аналізу, якщо різниця суттєво відрізняється.

gurifocus.com надає інформацію також і про зріст прибутку на акцію, що є одним з показників стратегії Грема (рисунок 1.14). Окрім графіку зросту,

відображається інформація щодо прибутків підприємства, рух грошей, дивіденди, балансова вартість тощо.

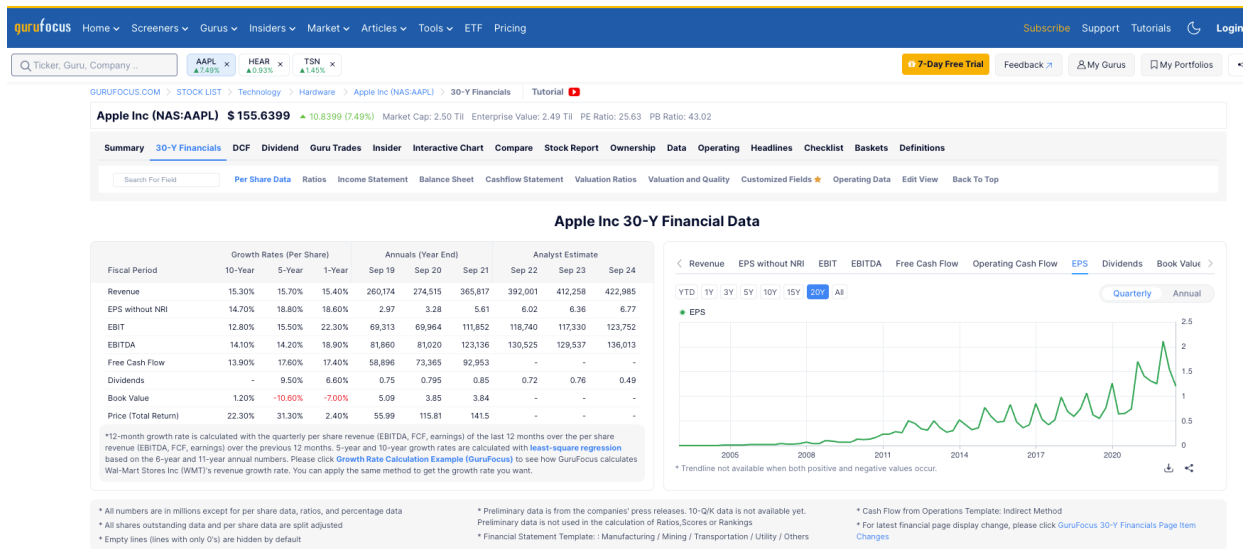


Рисунок 1.14 – Базова фінансова інформація на платформі gurufocus.com підприємства Apple Inc. за 28.10.2022

1.4 Висновки до розділу

У даному розділі було проведено аналіз схожих систем підтримки прийняття рішень, в яких було виділено ряд недоліків, зокрема: громіздкий не інтуїтивно-зрозумілий інтерфейс; відображення всього лише фактичних значень коефіцієнтів, але без якісних статусів відповідно рекомендаційних значень; відсутність української розкладки. Так само розглянуто функціональність систем, які частково виконують поставлену задачу для кваліфікаційної роботи, тим самим проаналізовано недоліки даних частин, а саме необхідність введення великої кількості змінних для розрахунку кожного з коефіцієнтів. Вищеописані недоліки були проаналізовані та взяті до уваги при розробці програмного засобу.

2 РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ ТА АЛГОРИТМУ РОБОТИ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ УПРАВЛІННЯ ФІНАНСОВИМ КАПІТАЛОМ ВИРОБНИЧИХ ПІДПРИЄМСТВ

2.1 Розроблення структурної схеми системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств

На сьогоднішній день існує безліч систем підтримки прийняття рішень, в тому числі економічних. Такі системи реалізовані у вигляді програмних засобів (найчастіше – веб-застосунків або десктопних програм), наприклад, таких як yahoo.finance або finviz. Користуючись цими системами, можна знайти велику кількість історичної інформації, графіків, новин, експертних статей та аналітики. Однак, вони не пропонують користувачу жодної стратегії для того, яким чином йому використовувати цю інформацію. Тому, пропонується система підтримки прийняття рішень, яка дозволяє користувачу, використовуючи пошукове поле, аналізувати цінні папери підприємств і перевіряти відповідність підприємства під стратегію вартісного інвестування.

Структурна схема – схема, яка визначає основні функціональні частини виробу, їх взаємозв'язки та призначення. Структурна схема була побудована для того, щоб відобразити загальну структуру системи підтримки прийняття рішень, тобто його основних компонентів, вузлів та зв'язків між ними. Вимоги щодо структурної схеми потребують, що візуалізація потрібна передавати яким чином працює система і як взаємодіють його частини.

Вартісне інвестування – стратегія інвестування, яка заснована на виявленні недооцінених цінних паперів за допомогою фундаментального аналізу, запропонована Бенджаміном Гремом та Девідом Доддом. Цінні папери, що відповідають стратегії, повинні мати низку критеріїв, за допомогою яких інвестор може оцінити відповідність стратегії: цінний папір має торгуватися нижче балансової вартості, повинен мати гарну дивідендну історію, а також

низькі показники ціна/прибуток та ціна/балансова вартість. Конкретні рекомендаційні значення вказані в таблиці відповідностей значень індикаторів і статусів.

На рисунку (рисунок 2.1) зображена структурна схема роботи програмного засобу. Початок роботи програма бере з того, що отримує дані від користувача за допомогою пошукового поля. Отримуючи дані, клієнт (frontend) програмного засобу оброблює їх до того формату, який розуміє сервер, і відправляє відповідні запити. Сервер, в свою чергу, робить відповідні запити для бази даних, для того щоб також обробити їх і відправити клієнту необхідну відповідь. Після того, як клієнт отримує відповідь від серверу, дані проходять два рівні обробки – через сервіс індикаторів та сервіс статусів, в яких вже відбувається бізнес-логіка програмного засобу – прорахунок індикаторів та співставлення кожного індикатора з таблицею відповідностей для того, щоб сформувати статус індикатора і віддати вже сформовані дані для рендеру у HTML.

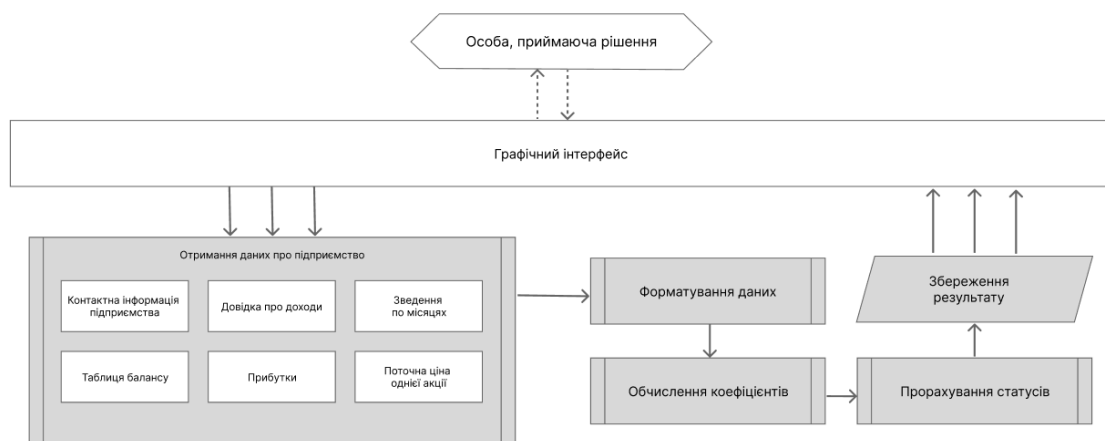


Рисунок 2.1 – Структурна схема роботи СППР

2.2 Розроблення алгоритму роботи системи підтримки прийняття рішень управління фінансовим капіталом виробничих підприємств

Після того, як користувач ввів ключове слово для пошуку підприємства і натиснув на будь-яке підприємство у наданому листі збігів, відправляється запит

на API для того, щоб отримати відповідь з серверу з відповідною інформацією. Програмний код перевіряє, чи прийшов 200 статус у відповіді, і тільки в цьому разі програма продовжує розрахування з даними з серверу. В будь-якому іншому разі, користувач побачить на інтерфейсі повідомлення про те, що сервер не має змоги повернути коректні дані. В такому разі користувач може спробувати повторити свої дії або спробувати обрати іншу компанію у листі, що відображається після введення ключового слова для пошуку (рисунок 2.2).

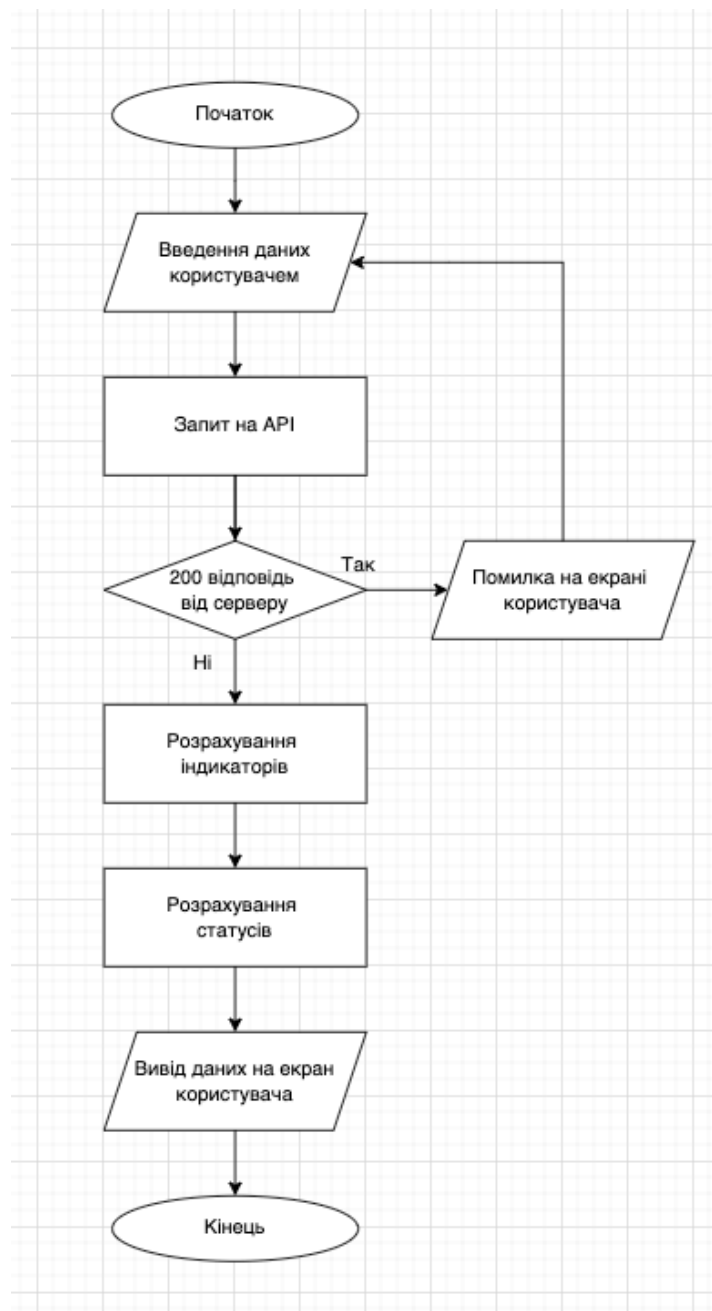


Рисунок 2.2 – Алгоритм роботи розробленого програмного засобу

2.3 Технічна схема роботи СППР

Схематично зображено (рисунок 2.3), яким чином технічно працює розроблена система підтримки прийняття рішень. Вхідні дані необхідно отримати від користувача, як частіше за все через елементи форми – пошукове поле для введення ключових слів для пошуку підприємства для аналізу. Після того, як користувач ввів дані генерується список знайдених підприємств, в якому користувач має змогу вибрати бажане підприємство.

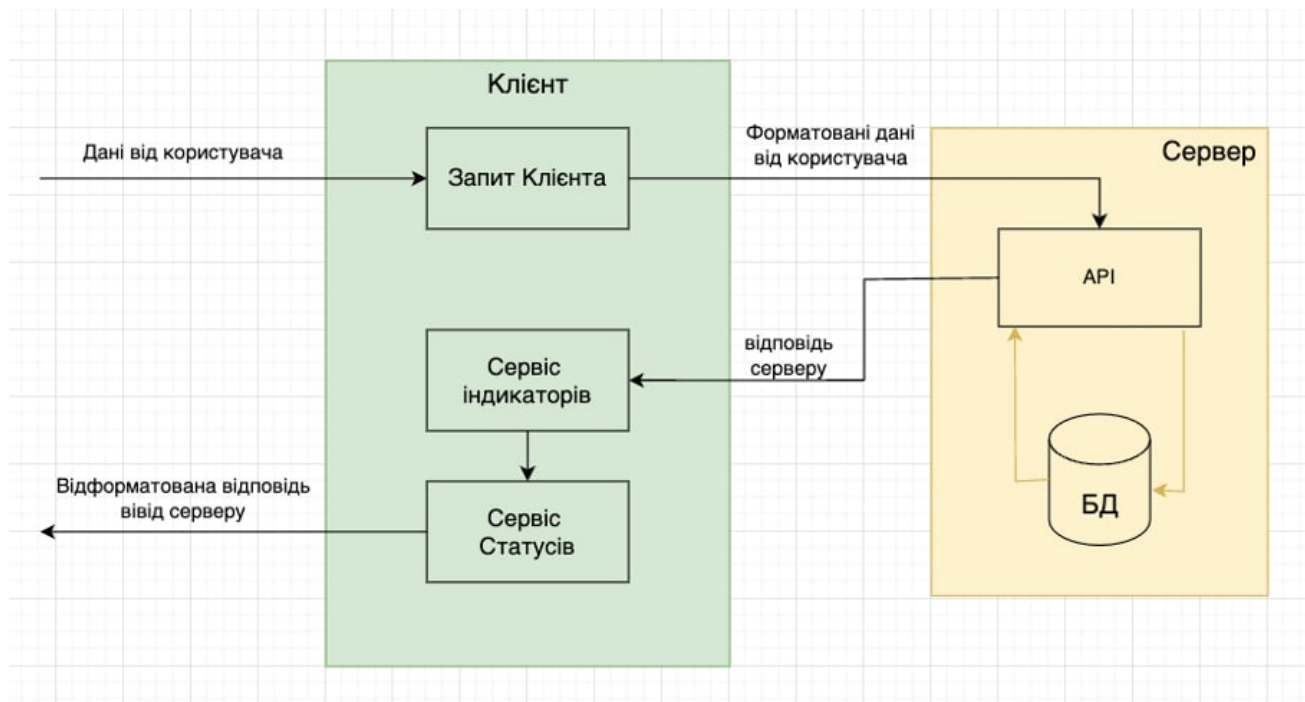


Рисунок 2.3 – Технічна схема роботи СППР

Після відповідного вибору, клієнт отримує інформацію про обране підприємство у зрозумілому форматі, таким чином маючи змогу надіслати коректний запит на сервер для отримання всієї інформації про підприємство. API, отримуючи запити від клієнту, повертає на клієнт відповідні дані через HTTPs протокол, використовуючи унікальний ключ безпеки, без якого повернути дані на клієнт технічно неможливо зі сторони серверу. У разі відсутності ключа, сервер повертає на клієнт 403 помилку – Forbidden, який зазвичай надсилається сервером через внутрішні обмеження. Після того, як дані коректно відправились клієнту, вони проходять два рівні обробки – через сервіс

індикаторів і сервіс статусів, щоб здійснилось необхідне форматування для того, щоб система змогла коректно відобразити відформатовані дані на екрані користувача. Після успішного форматування, користувач бачить перед собою інтерфейс системи – підприємство додалось у список проаналізованих підприємств, а також праворуч на екрані відображається розширена інформація про підприємство, графіки та прораховані коефіцієнти.

2.4 Розрахунок коефіцієнтів

Показники, які Бенджамін Грем аналізував у своїх дослідженнях, є актуальними і зараз, тому що вони є фундаментальними для будь-якої компанії і показують, наскільки бізнес підприємства є стабільним, зростаючим та є бізнесом, який піддається фінансовим світовим кризам. Показники представлені нижче.

– Коефіцієнт поточної ліквідності.

Даний показник характеризує те, наскільки підприємство є платоспроможним. Вказує на здатність підприємства погашати свої поточні зобов'язання, таким чином, що дане підприємство конвертує свої оборотні активи в гроші, а ці гроші відповідно використовуються для покриття зобов'язань. Рекомендованим показником є значення показнику більше ніж 2. Якщо даний показник розраховується нижчий, ніж рекомендований, це свідчить про те, що підприємство має проблеми з платоспроможністю, адже має недостатньо оборотних активів для відповідання за поточними зобов'язаннями.

$$\text{Коефіцієнт поточної ліквідності} = \frac{\text{Поточні активи підприємства}}{\text{Поточні борги підприємства}} \quad (2.1)$$

Очевидно, що для потенційного інвестора чим більше значення даного показника, тим краще. Але з іншого боку, якщо показник є занадто високим, це свідчить про неефективну структуру активів, тобто неефективності менеджерів і власників підприємства. Однак варто зазначити, що при прорахунку необхідно

проаналізувати поточну ліквідність підприємства з конкурентами у галузі, бо у багатьох сферах рекомендоване значення коефіцієнту може відрізнятись. З часом поточна ліквідність великих успішних підприємств знижується. Це пов'язано з покращенням планування бюджету та пришвидшенням обороту активів.

– Стабільний прибуток компанії.

Загально цього показнику, компанія не повинна мати збиткових кварталів або фіскальних років на протязі останніх десяти років.

Стабільний прибуток є важливою складовою будь-якого бізнесу. Прибуток підприємства є основним джерелом фінансування розвитку підприємства, забезпечення усіх форм інвестування, виплати дивідендів, удосконалення матеріально-технічної бази. Вся діяльність підприємства спрямована на стабілізації прибутку на певному рівні, або навіть на забезпечення зростання прибутку. Стабільний прибуток є показником успішного менеджменту, актуальності продукції, що виготовляється або послуг, які надаються підприємством.

– Дивідендна історія. Компанія на протязі останніх двадцяти років повинна стабільно платити дивіденди власникам цінних паперів. Дивіденди – частина чистого прибутку, розподілена між учасниками (власниками) відповідно до частки їх участі у статутному капіталі підприємства. Дивідендна історія є важливою складовою стратегії акціонерних товариств. Цей показник впливає на привабливість підприємства для інвесторів, вартість бізнесу, імідж підприємства. Окрім фактичної виплати дивідендів, відмінним показником є зріст виплати дивідендів на кожну акцію. Це свідчить про те, що прибуток підприємства зростає. Стабільний рівень дивідендних виплат сигналізує про те, що вкладений інвесторами капітал працює ефективно.

– Зріст прибутку.

Найбільш комплексний показник, який розраховується як різниця прибутку на кожен цінний папір останніх і перших три роки декади. Середні дані за перші і останні 3 роки декади беруться для того, щоб згладити часті коливання ринку, які були викликані економічним циклом. З точки зору оцінки

потенційною прибутковості компанії, середній показник є більш надійним і точним, якщо зрівнювати показник лише за останній рік. Більш того, перевага усереднення в тому, що з допомогою нього можна вирішити питання усіх додаткових витрат і відстрочок виплати податків, що також є частиною фінансової історії підприємства.

– Коефіцієнт Грема розраховується по формулі:

$$\frac{\text{Активи компанії} - \text{борги компанії}}{\text{кількість звичайних акцій}} \quad (2.2)$$

Бенджамін Грем вперше обговорив чисту поточну вартість однієї акції (NCAV) у виданні Аналіз Безпеки [15], у співавторстві з Девідом Доддом. У книзі чиста вартість поточних активів визначається як тільки поточні активи мінус усі зобов'язання та претензії до випуску. Бенджамін Грем шукав підприємства, ринкова вартість яких становила менше двох третин їх чистого оборотного капіталу. Цей показник є одним з ключових для стратегії вартісного інвестування, оскільки якби підприємство припинило свою діяльність і продала все свої фізичні активи, вартість цих активів була б ліквідною вартістю підприємства.

Ліквідна вартість підприємства – загальна вартість усіх її фізичних активів, таких як інвентар, обладнання, нерухомість. Він виключає нематеріальні активи, такі як інтелектуальна власність та пізнаваність бренду.

Таким чином, цінний папір, який торгується нижче коефіцієнту Грема, дозволяє інвестору купити долю підприємства за нижчою ціною, ніж вартість її поточних активів. І поки підприємство має розумні перспективи, інвестори, ймовірно, отримають значно більше, ніж вони платять. За словами Грема, інвестори отримають значну вигоду, якщо інвестуватимуть у підприємства, ціна акцій яких не перевищує 67 % від їх коефіцієнта Грема на акцію. Однак необхідно чітко зрозуміти, що не всі акції, вибрані за допомогою формули NCAVPS, матимуть високий прибуток, і що інвестори також повинні диверсифікувати свої активи, використовуючи цю стратегію. Грем рекомендував мати 10-30 цінних паперів у портфелі.

– Число Грема розраховується по формулі:

$$\sqrt{22.5 * \frac{\text{ринкова капіталізація}}{\text{річний прибуток компанії}} * \frac{\text{ринкова капіталізація}}{\text{балансова вартість компанії}}} \quad (2.3)$$

Число Грема вимірює фундаментальну вартість акцій, враховуючи прибуток компанії на акцію (EPS) і балансову вартість на акцію (BVPS). Число Грема – це верхня межа цінового діапазону, яку захисний інвестор повинен заплатити за акції. Згідно з теорією, будь-яка ціна акцій, нижча за число Грема, вважається недооціненою, і тому варто інвестувати. Коефіцієнт використовується як загальний тест, коли намагаються визначити акції, які зараз продаються за хорошою ціною. Значення 22,5 включено в розрахунок, щоб врахувати переконання Грема, що співвідношення ціна-прибуток (P/E) не має перевищувати 15x, а BVPS не має перевищувати 1,5x (отже, $15 \times 1,5 = 22,5$). Число Грема також є ключовим індикатором в розрахунку відповідності цінного паперу до стратегії вартісного інвестування, хоча і не враховує багато фундаментальних характеристик, які вважаються надійними для пошуку потенційної інвестиції, наприклад якість управління, основні акціонери, характеристики галузі та конкурентне середовище.

Для розробки бізнес-логіки програмного додатку, необхідно було проаналізувати недоліки існуючих систем для вирішення конкретного завдання: аналіз підприємства на відповідність стратегії коштовного інвестування. Основними недоліками є:

- недостатність даних для прорахунку необхідних індикаторів;
- відсутність статусів на прорахованих індикаторах;
- інтуїтивно-зрозумілий інтерфейс;
- українська локалізація.

Після аналізу основних конкурентів у даній галузі, необхідно було спроектувати таблицю відповідностей, завдяки якій будуть прораховані статуси кожного індикатора (рекомендований, задовільний, незадовільний). Значення в таблиці для кожного індикатора були взяті з рекомендацій Бенджаміна Грема,

які він описував у своїй книзі Розумний інвестор, а також коефіцієнту та числу Грема, які Бенджамін розрахував для того, щоб знаходити недооцінені та потенційно економічно успішні підприємства.

Таблиця 2.1 – Відповідності значень індикаторів і статусів, які відображає розроблений програмний засіб

Назва коефіцієнта	Рекомендований статус	Задовільний статус	Незадовільний статус
Коефіцієнт поточної ліквідності	≥ 2	< 2	< 1
Стабільний прибуток	Не виявлено збиткових років	Виявлено збитковий квартал, але не рік	Виявлено збитковий рік
Дивідендна історія	Підприємство виплачує дивіденди останні 20 років	Підприємство наразі виплачує дивіденди	Підприємство не виплачує дивіденди
Зріст прибутку	Середній прибуток на акцію за останні 3 роки вище мінімум на 33%, ніж середній прибуток за перші 3 роки останньої декади	Середній прибуток за останні 3 роки вище, ніж середній прибуток за перші 3 роки останньої декади	Середній прибуток за останні 3 роки нижче, ніж середній прибуток за перші 3 роки останньої декади
Зріст прибутку			За десятиріччя зафіксований збиток
Коефіцієнт Грема	Поточна ціна акції в межах 50-70% від коефіцієнту	Поточна ціна акції в межах 40-80% від коефіцієнту	Поточна ціна акції не в межах 40-80% від коефіцієнту
Коефіцієнт Грема			Негативний показник
Число Грема	Число Грема більше, ніж поточна ціна акції	—	Число Грема менше, ніж поточна ціна акції
Число Грема			Множина коефіцієнтів P/E та P/B більше, ніж 22,5

2.5 Вартісне інвестування

Вартісне інвестування – стратегія інвестування, заснована на виявленні недооцінених цінних паперів шляхом фундаментального аналізу. Основи стратегії запропонували в 1928 році Бен Грем і Девід Додд – викладачі бізнес-школи Колумбійського університету. Це процес вибору цінних паперів, що продаються нижче їх внутрішньої вартості. Важливо, що вартісне інвестування пов'язано не з тим, наскільки акція впала чи виросла в ціні, а з тим, яка внутрішня ціна акції підприємства і чи торгується вона нижче своєї ринкової вартості. Відомий американський інвестор Воррен Баффет вказує, що суть вартісного інвестування полягає в покупці акцій за ціною нижчою за їхню внутрішню вартість. За висловом Баффета, вартісне інвестування полягає в умінні знайти видатну компанію за розумною ціною.

Основна концепція вартісного інвестування полягає в тому, щоб купити щось коштовне за ціною, яка не переважає фактичну коштовність цієї речі. Справа вартісних інвесторів в тому, щоб шукати цінні папери, які, на їх думку, є недооціненими ринком і купувати їх. Бенджамін Грем виступав за підхід, який був заснований на конкретних правилах. Автор стратегії вартісного інвестування був зосереджений на побудові інвестиційного портфеля на основі набору об'єктивних коефіцієнтів, які прораховуються у розробленому додатку.

Акції можуть бути недооціненими з декількох причин:

- негативний новинний фон;
- тимчасові кризові явища у галузі;
- падіння сезонного попиту на товари підприємства;
- судові розгляди тощо.

У двадцять першому столітті інформація може зіграти злий жарт не тільки з відомими людьми, але й з підприємствами. Відомі випадки, коли після несприятливої статті у відомій газеті або посту у Twitter ціни акцій змінюються більше, ніж свіжий бухгалтерський звіт. В такому разі необхідно провезти фундаментальний аналіз підприємства, і якщо ціна є нижче за свою

справедливу, це може бути відмінним шансом для інвестування у дане підприємство.

Фінансова грамотність є дуже важливим фактором у сучасному світі, а вартісне інвестування є однією з стратегій не тільки збереження коштів підприємця або інвестора, але й забезпечення пасивного доходу.

2.6. Висновки до розділу

В другому розділі кваліфікаційної роботи було розроблено структурну схему системи підтримки прийняття рішень фінансового капіталу виробничих підприємств, а також алгоритм роботи та технічну схему роботи програмного засобу, що буде розроблений. Також в розділі представлена інформація щодо даних, які необхідні для прорахунку кожного коефіцієнту, а також теоретична інформація щодо стратегії вартісного інвестування.

3 РОЗРОБЛЕННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ УПРАВЛІННЯ ФІНАНСОВИМ КАПІТАЛОМ ВИРОБНИЧИХ ПІДПРИЄМСТВ У ВИГЛЯДІ ПРОГРАМНОГО ЗАСОБУ

3.1 Вибір середовища програмування

На даний момент існує велика кількість IDE (англ. – Integrated Development Environment) для розробників програмного забезпечення. Це інтегровані єдині середовища розробки, які використовуються розробниками для створення різноманітного програмного забезпечення. IDE є комплексом з декількох інструментів, а саме: текстового редактора, компілятора або інтерпретатора, засобів автоматизації збірки коду та відладження. Більш того, середовища програмування можуть містити інтерфейс для інтеграції таких функцій як системи контролю версій або тестування. Більшість сучасних середовищ програмування налаштовані на розробку з декількома мовами програмування, хоча існують і ті, які працюють лише з однією.

Для розроблення даної системи підтримки прийняття рішень, використовувалось середовище програмування Visual Studio Code. Це безкоштовний засіб для створення сучасних програмних засобів. Позиціонується як легкий редактор коду для кроссплатформової розробки програмних додатків.

Для зручності розробника, середовище Visual Studio Code має досить широкі можливості для кастомізації, такі як зміна теми середовища, зміна гарячих клавіш для виконання різноманітних команд, конфігураційні файли. Більш того, має великих набір розширень, які встановлюються через окремо вкладку у меню та розширюють функціональність програмного середовища. Для деяких мов має функцію рефакторинга, за допомогою якої розробники можуть отримувати поради щодо покращення написаного коду. Найчастіше використовується розробниками програмного забезпечення на платформі браузерів, бо дане середовище програмування має повну інтеграцію з такими

мовами як HTML, CSS та JavaScript, що є базовими для цього типу розробки програмного забезпечення.

3.2 Етапи розробки програмного засобу

Розробка програмних застосунків на платформі браузера, зазвичай поділяється на front-end (розробка клієнтської частини програмного засобу) та back-end (розробка серверної частини програмного засобу), хоча це розділення є досить поверхневим. Насправді, розробка програмного засобу у складається з декількох великих етапів окрім програмування.



Рисунок 3.1 – Етапи розробки програмного застосунку

Програмний застосунок у сучасному світі є дуже ефективним інструментом для ведення успішного бізнесу, інформування цільової аудиторії або автоматизації процесів на виробництві, та часто не обмежується лише етапами (рисунок 3.1). Найчастіше це залежить від задачі, яку має виконувати застосунок для підприємства, організації культурного заходу, інформування населення щодо важливих подій тощо.

Визначення тематики та основної мети проекту. Перш за все, необхідно зрозуміти навіщо потрібен програмний засіб та яку користь він приносить користувачу або власнику бізнесу. Від цього залежить основні вимоги до програмного засобу, портрет цільової аудиторії, тип ресурсу тощо.

Розробка технічного завдання. ТЗ є фундаментом для подальшої роботи, в якому прописуються всі деталі: структура програмного засобу, вимоги щодо дизайну, маркетингове наповнення (чи його відсутність), функціональні вимоги тощо. ТЗ має бути детальним, чітким та лаконічним, адже від цього залежить

розуміння технічним менеджментом побажань клієнта та зменшить кількість запитань до власника бізнеса, що зекономить час саме власника та команди розробників. Якщо технічне завдання складено правильно, то це дозволяє зменшити або навіть мінімізувати трудовитрати між сторонами договору і уникнути більшості проблем, пов'язаних з розбіжностями у критерії оцінки або зміни технічних параметрів. Чим більше конкретики в технічному завданні, тим краще для обох сторін. Щоб зрозуміти що технічне завдання складене правильно, після прочитання документу ні у замовника, ні у команди розробників не повинно залишатись будь-яких питань перед початком розробки застосунку або мобільного додатку.

Прототипування та дизайн. На цьому етапі створюється прототип програмного засобу (рисунок 3.2), який відображає базову структуру застосунку. Зазвичай це чорно-білий макет, на якому структурно позначено розташування основних блоків. До прототипу не додають стилі (кольори, зображення, типографіку тощо) задля фокусування на UX-складової. Після затвердження прототипу дизайнеру легше додавати візуальну (UI) складову, адже після затвердження прототипу зазвичай не змінюють розташування блоків. Наступним кроком додають такі візуальні складові, як основна палітра кольорів, зображення, градієнти, анімації, типографіку тощо та окремо узгоджують з замовником.



Рисунок 3.2 – Прототип розробленої СППР

UI (User Interface) – це користувацький інтерфейс. При використанні цього терміну, розробники програмного засобу мають на увазі те, як продукт виглядає: кольори, анімації, форми, шрифт тощо. Будь-який візуальний елемент або взаємодія з ним (анімація, зміну кольору або розміну шрифту, додавання тіні або підсвічення) повинні бути розроблені в процесі UI-дизайну. Дизайнери користувацького інтерфейсу створюють зовнішній вигляд інтерфейсу програми. Саме ці дизайнери щільно пов'язані з естетикою, адже мають бути впевненими в тому що користувацький інтерфейс є привабливим, відповідає тематиці застосунку, а також має бути впевненим в тому що кожен візуальний компонент застосунку використовує єдині візуальні стилі.

UX (User Experience) – це користувацький досвід. При використанні цього терміну, розробники застосунку мають на увазі те, яким чином користувач буде використовувати розроблений додаток, наскільки йому буде зручно знайти необхідну функцію (zareєstrуватись, додати товар у козину, знайти необхідне місце на карті або здійснити фундаментальний аналіз підприємства на відповідність стратегії вартісного інвестування). UX-дизайнери аналізують дії користувачів, і на основі отриманих досліджень розуміють, яким чином розмістити основні і другорядні функції застосунку. Дизайнери цієї галузі досліджують різні підходи щодо виконання окремої дії в застосунку, аналізують цільову аудиторію застосунку, тестують створений прототип на основі отриманих даних [16]. Саме UX-дизайнери забезпечують те, наскільки інтерфейс є інтуїтивно-зрозумілим для користувача.

Оскільки принципи застосування інтерфейсу користувача мають узагальнений характер, то при виконанні магістерської роботи зі створення СППР необхідно ураховувати також і потреби потенційних користувачів. Типи користувачів, задач і ситуацій, пов'язаних з прийняттям рішення, мають визначити специфічні особливості всього процесу розробки інтерфейсу користувача. Для керівників вищої ланки управління конче потрібна і є більш доречною зовсім інша, ніж для керівників середньої ланки, техніка інтерфейсу

користувача СППР, орієнтовані на швидкість реакції або на кризові ситуації, мають зовсім інші вимоги до інтерфейсу, ніж системи підтримки довгострокового планування; альтернативні контексти задач (наприклад, чи передбачається використання СППР для підтримки структурування задачі або для одержання прогнозів) також потрібно урахувати у процесі синтезу або добору формальних засобів та інструментів для компонування мов дій і мов відображень людино-машинного інтерфейсу. Дослідження і роботи зі створення інтерфейсу користувача проводяться в багатьох країнах, і тому можна очікувати появу досконаліших механізмів взаємодії користувача і системи. На даний час найбільшої уваги заслуговують чотири конкретні альтернативні варіанти інтерфейсу користувача: інтерфейс, що ґрунтується на меню; адаптивний інтерфейс; інтерфейс із застосуванням природної мови; графічні засоби для удосконалення діалогу користувач–система.

Верстка та програмування. Верстка є одним з найважливішим етапів розробки програмного засобу, в результаті якого спроектований дизайнером макет перетворюється в HTML і CSS код. Для того, щоб якісно зверстати макет, потрібні знання особливостей роботи браузерів, принципи позиціонування елементів, а також семантики сторінок у браузері. Даний етап є відтворенням дизайну у програмний продукт, який буде інтерактивно працювати у браузері. Часто верстка передбачає відтворення дизайну не тільки в те розширення, яке є на макеті дизайнера, але й адаптування дизайну на інші розширення екрану. Найчастіше, для мобільних версій додатків використовують найпопулярніші моделі телефонів, ноутбуків та планшетів як базу для верстки. Частіше за все, верстальник отримує макет у розширенні 1920 на 1080 пікселів. Програміст, використовуючи різноманітні інструменти розробки, збирає файли у класичний HTML, CSS та JavaScript, які браузер зчитує, проводить рендеринг та відображає користувачу. На першому етапі верстальнику необхідно дослідити структуру макета на рахунок можливих елементів, що повторюються, винесення стилів, що використовуються декілька разів у проекті для того щоб не дублювати їх у коді. Але навіть після того, як весь код написаний і верстка макету готова, необхідно

ще переконатися в тому, що код працює коректно в різних браузерах і незалежно від цього фактору верстка відображається правильно. Тільки після того, як всі вимоги замовника були виконані, можна приступати до наступного етапу розробки програмного засобу.

Замовники досить часто вимагають від верстальників верстку, яка називається Pixel-perfect. Це означає що макет, який буде відтворений верстальником за допомогою HTML та CSS у браузері не повинен нічим відрізнитись від спроектованого дизайнером макету. Перевіряється це таким чином: робиться скріншот дизайну і за допомогою розширення у браузері накладається на верстку. Макет, зверстаний по pixel-perfect моделі повинен накладатися рівномірно на верстку і не мати будь-яких відмінностей від вищенакладеного скріншоту.

Наповнення контентом та SEO-оптимізація. Маркетингова складова проекту, яка передбачає додавання у готову верстку заголовки, статті, графічні матеріали тощо. Для того, щоб програмний застосунок успішно просувався у пошукових системах, контент має відповідати стандартам оптимізації, таким як відповідність ключових слів та опису контенту на сторінці, семантика верстки тощо.

Тестування. Включає в себе проведення різноманітних перевірок на предмет помилок, як з даними, так і з можливими діями користувача. Частіше за все, тестування проводить особа, який безпосередньо не розроблювала додаток і може проробити неочевидні випадки, які не змогли знайти розробники, які також мають проводити самотестування.

Здача готового проекту. Після передачі проекту замовнику, розробники або менеджмент проекту можуть провезти навчання роботи з застосунком. Це можуть бути дії щодо оновлення контенту в додатку, зміни розташування основних блоків, додавання пунктів меню для навігації тощо.

3.3 Вибір мови програмування для розробки програмного засобу

Мова програмування є найважливішим інструментом для будь-якого розробника. Вона використовується як інструмент для створення програм, які контролюють поведінку машин. Мова програмування, фактично, схожа на іноземну мову, але з її допомогою можна донести інформацію не до людини, а до комп'ютера. Продовжуючи проводити паралелі між іноземною мовою та мовою програмування, можна сказати що мета програміста полягає в тому, щоб використовувати мову так, щоб комп'ютер розумів розробника.

Розроблений додаток був створений за допомогою frontend-фреймворку Vue.js, який був написаний на мові програмування JavaScript. JavaScript є динамічною, об'єктно-орієнтовною скриптовою прототипною мовою програмування. Найчастіше використовується на веб-сторінках, для того щоб зробити інтерфейс динамічним, але також може використовуватись для розробки десктоп-додатків, backend'у, а також мобільних додатків.

Окремо необхідно додати, що розробка клієнтських додатків у браузері неможлива без використання таких мов як HTML (HyperText Markup Language) і CSS (Cascading Style Sheets). HTML є мовою розмітки гіпертексту, яку розуміє абсолютно кожен браузер, далі інтерпретує код в інтерфейс, який відображається на екрані монітора. CSS в свою чергу є мовою стилів, що використовуються для опису зовнішнього вигляду HTML-документів. За допомогою цих мов неможливо виразити конкретний алгоритм дій, які буде виконувати комп'ютер, проте ці мови є дуже важливими для веб-розробки, адже саме завдяки ним користувач бачить перед собою зручний і зрозумілий інтерфейс, який вже динамічно працює за допомогою мови програмування JavaScript. Всі ці три мови є основними технологіями всесвітнього павутиння World Wide Web.

Якщо описувати окремо лише frontend розробку, то програми на JavaScript називають сценаріями, які вбудовуються в HTML-код сторінки та виконуються браузером. Сценарії виконуються як простий текст, тому для них не є необхідним компілятор або інша додаткова обробка. Мова JavaScript не надає

низькорівневого доступу до пам'яті або процесора, оскільки початково була створена як мова, яка може додати динаміку у програмні додатки у браузері. На сьогоднішній день, JavaScript є безальтернативною мовою програмування для клієнтських додатків, оскільки має два суттєвих плюси: цілковита інтеграція з HTML/CSS та підтримка всіма сучасними браузерами.

3.4 Вибір frontend-фреймворку для розробки програмного засобу

Якщо JavaScript є безальтернативною мовою програмування для розробки клієнтських додатків, то для вибору фреймворку, написаного для основи JavaScript, необхідний більш детальний аналіз. На даний момент існує велика кількість frontend-фреймворків, які програміст може вибрати для розробки клієнтських додатків. З безлічі з них якісно виділяється три: React, Angular, Vue.

React – це декларативна, ефективна і гнучка JavaScript-бібліотека для значного спрощення побудови й маніпулювання DOM-елементами (Document Object Model). Розроблена компанією Facebook. React.js є дуже популярною бібліотекою, хоча і не є фреймворком, тому що не має повної відповідності моделі MVC (Model-View-Controller), а відповідає лише за частину View. Із плюсів даної бібліотеки необхідно виділити популярність, розвинуту екосистему, високу швидкість роботи, а також зручні інструменти для налагодження коду такі як React Developer Tools у браузері.

Angular – повноцінний фреймворк, який керується принципами MVVM (Model-View-ViewModel) побудови застосунків. Розроблений компанією Google. Для зрівняння з React.js, Angular має багато можливостей зі старту роботи з фреймворком і без додатково завантажених бібліотек. Наприклад, Angular містить в собі CLI для генерування структури файлів, засоби лінтування, роутінг, засоби для роботи з анімаціями та формами тощо. Із плюсів необхідно виділити модульність (можливість чітко групувати елементи), відсутність необхідності додавання інших бібліотек для повноцінної розробки, використання MVVM принципів для побудови додатків.

Vue – так само, як і Angular, є повноцінним фреймворком для створення клієнтських додатків. Розроблений програмістом Еваном Ю та його командою. Працює на базі звичайного HTML, CSS та JavaScript, ж можливістю декларативно програмувати користувацькі інтерфейси будь-якої складності. Vue є сумішшю двох попередньо описаних представників, адже може використовуватись як бібліотека, яку можна впровадити на окремо взяту веб-сторінку на будь-якому сайті для швидкого і простого додавання динаміки на програмний засіб у браузері, а також може використовуватись як повноцінний фреймворк, адже за допомогою впровадженого у екосистему Vue-CLI має всі можливості для розробки проєктів будь-якого розміру самостійно без сторонніх бібліотек, такі як менеджмент стану даних, роутінг, підтримка від інтегрованих середовищ розробки IDE, а також додатків для тестування та налагодження коду. Vue є найпопулярнішим JavaScript-фреймворком на Github (рисунок 3.3).

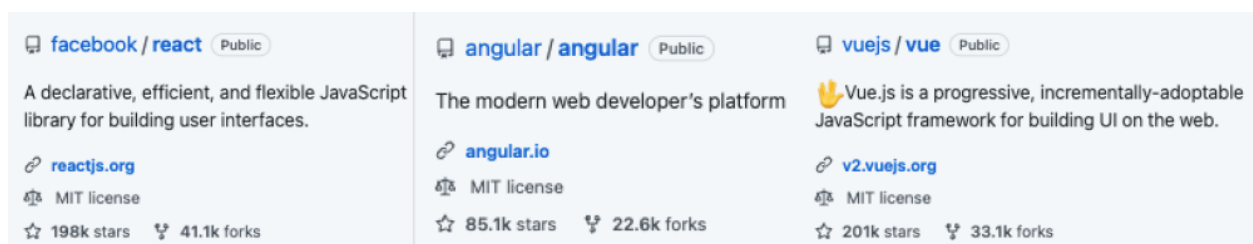


Рисунок 3.3 – Порівняння популярності React, Angular та Vue на платформі Github.com за 21.11.2022

Для розробки програмного додатку для перевірки відповідності підприємства до стратегія вартісного інвестування був обраний фреймворк Vue, адже він має низку плюсів для розробки, а саме:

- швидкий старт розробки – для того, щоб почати розробляти додаток, необхідно лише запустити відповідну команду у командному рядку для створення базової файлової структури та обрати необхідну конфігурацію у, яка містить в собі кореневий компонент для розробки логіки та інтерфейсу додатку, файл з базової конфігурацією додаткових модулів додатку та файл для

підключення базових розширень додатку, таких як бібліотека компонентів Vuetify та менеджменту станів Vuex (рисунок 3.4);

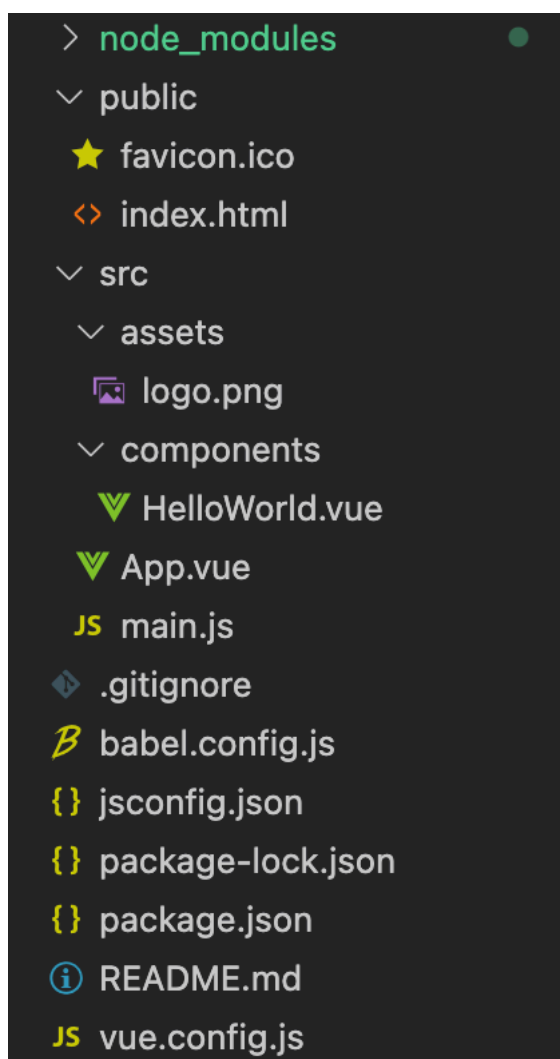


Рисунок 3.4 – Базова файлова структура у фреймворку Vue

– зручна структура файлу компонента – концепція Vue пропонує запис усіх трьох базових технологій всесвітнього павутиння HTML, CSS та JavaScript у одному файлі, хоча і існує можливість розбиття на декілька окремих файлів. Це дає можливість зручно контролювати всі стани компоненту у одному місці, не перемикаючись на окремі файли (рисунок 3.5);



```
1 <template>
2   <h1 class="hello-world__title">{{ greetings }}</h1>
3 </template>
4
5 <script>
6   export default {
7     name: 'HelloWorldComponentName',
8
9     data() {
10      return {
11        greetings: 'Hello, World!'
12      }
13    }
14  }
15 </script>
16
17 <style lang="scss" scoped>
18   .hello-world {
19     &__title {
20       font-size: 24px;
21     }
22   }
23 </style>
24
```

Рисунок 3.5 – Базова структура компонента у фреймворку Vue

– легкість фреймворка – на відміну від іншого фреймворка Angular, вага мінімізованої базової файлової структури проекту, написаного за допомогою Vue є 86 КБ, при 250 КБ у Angular. Різниця у три рази є суттєвою, більш того якщо брати до уваги те, що розроблений програмний додаток є односторінковим додатком для відображення відносно невеликої кількості інформації.

3.5 Додаткові засоби для розробки програмного додатку.

Існує велика кількість додаткових інструментів, за допомогою яких розробляються клієнтські програмні засоби. Це можуть бути бібліотеки з готовими методами, які можуть зменшити написання коду, набір компонентів зі заздалегідь написаними стилями та функціональністю, метамови з розширеною функціональністю для HTML/CSS тощо. Звісно, програмний засіб, розроблений на основі Vue, може працювати і без додаткових інструментів, але вони якісно покращують усі аспекти розробки та готового проекту. Використовувались для розробки:

- Axios.js. http-клієнт, який був оснований на об'єкті мови JavaScript – Promise, який частіше за все використовується для відкладних та асинхронних запитів до серверу. Маючи зручний API, маємо легкий доступ до HTTP-методів GET, POST, PUT, DELETE тощо, які приймають в собі посилання на ресурс, на який необхідно зробити запит, а також структуровані параметри запиту, такі як headers, params тощо. Використовуючи дану бібліотеку, розробник спрощує написання коду при використуванні методів для асинхронних запитів до серверу (рисунок 3.6);

- Vuetify – бібліотека готових компонентів, написана на базі фреймворку Vue. Використовуючи Material Design Specification [17], дизайн Vuetify є акуратним, лаконічним та інтуїтивно-зрозумілим для користувача. Бібліотека містить в собі більшість компонентів, які можуть знадобитись розробнику будь-якого додатку, а також має досить широку функціональність, яка допомагає уникнути написання зайвого коду;

- Chart.js. бібліотека для створення інтерактивних графіків. У розробленому додатку використовується для побудови таких графіків, як графік дивідендів на акцію підприємства, графік прибутку на акцію підприємства, а також графік прибутку за квартал підприємства. Бібліотека дозволяє побудувати декартову систему координат, X та Y в якій можуть бути тими параметрами, які поставить розробник при побудові графіку. Наприклад, у графіку прибутку за

квартал підприємства, за вісь X відповідає дата закінчення кварталу, а за вісь Y – значення прибутку, задане у мільйонах. Також у функціональність графіку закладено прийняття кольору значень, заголовку, спливаючих вікон тощо. Таким чином на основі поставлених даних будується графік, який відображається у користувача на екрані;

```

src > api > settings > JS http.js > Http

You, a minute ago | 2 authors (Dmytro Ponomarenko and others)
1 import axios from 'axios';
2 |
3 const url = process.env.VUE_APP_HOST;
4 const apikey = process.env.VUE_APP_API_KEY;
5
Dmytro Ponomarenko, 2 months ago | 1 author (Dmytro Ponomarenko)
6 export default class Http {
7   constructor() {
8     this.headers = axios.defaults.headers;
9   }
10
11   static async get(params) {
12     try {
13       const response = await axios.get(url, {
14         headers: this.headers,
15         params: {
16           ...params,
17           apikey
18         }
19       });
20     } catch (e) {
21       return response.data;
22     }
23   }
24 }
25
26 }
Dmytro Ponomarenko, 2 months ago • Master

```

Рисунок 3.6 – Використовування бібліотеки axios в тілі метода get класу Http для виклику асинхронних запитів на сервер

– `Lodash.debounce`. `Lodash` – популярна бібліотека, яка містить в собі велику кількість корисних методів для JavaScript розробників, які ще не були додані у офіційну документацію мови. Для зменшення розміру проекту, бібліотека має окремі підбібліотеки, які містять в собі конкретно необхідний для проекту метод, такий як `debounce`. Метод дозволяє блокувати визивання функції, якщо вона була викликана деякий час тому. Час задається у мілісекундах як параметр методу `debounce`. Даний метод є оболонкою над іншим методом пошуку підприємства у пошуковій строці і блокує визивання пошуку, коли користувач швидко набирає ключове слово у пошуковому полі. Після заданого

часу (500 мілісекунд або 0,5 секунд) і останнього набраного користувачем символу пошук визивається і користувач бачить перед собою список відповідностей своєму запиту;

– Material design icons font. легковісна бібліотека, яка дозволяє бібліотеці компонентів Vuetify, за допомогою компоненту v-icon відображати користувачу іконки для таких функцій, як додати в улюблене, видалити або встановити активним підприємство;

– SCSS. розширена мова каскадних стилів, написана на основі CSS. SCSS має більше функціональних можливостей, ніж класичний CSS, такі як повноцінне використання змінних, можливість будувати файлову структуру стилів через винесення файлів в окремі директорії, а також повна підтримка бібліотек компонентів, таких як Vuetify.

3.6 Серверна частина програмного засобу

Для розробки програмного засобу було прийнято рішення використовувати сторонній API, який надає інформацію великої кількості підприємств, а також якісні дані які необхідні конкретно під стратегію вартісного інвестування, які запропонував Бенджамін Грем. При вивченні APIs, які надають достатньо інформації для стратегії, розглядалися такі системи як Yahoo Finance, Twelve Data та Alpha Vantage. При детальному аналізі ендпоінтів, які пропонує кожен API, стало зрозуміло що достатньо інформації пропонує лише Alpha Vantage, оскільки Yahoo finance не пропонує достатньо історичної інформації для дивідендного аналізу та зросту прибутку на акцію, а Twelve Data не пропонує інформацію щодо прибутків підприємства.

Ендпоінти, які використовуються у програмного додатку:

– OVERVIEW – повертає базову інформацію про підприємство, таку як значення коефіцієнтів P/E та P/B, прибуток на акцію, балансову вартість підприємства, кількість звичайних акцій тощо;

– BALANCE_SHEET – повертає фінансову інформацію про підприємство, таку як кількість поточних активів та боргів, значення яких використовуються при прорахунку коефіцієнту Грема, а також коефіцієнту поточної ліквідності;

– INCOME_STATEMENT – історична інформація підприємства щодо прибутків. Дані використовуються при побудові графіку квартальних прибутків, а також значення стабільного прибутку;

– TIME_SERIES_MONTHLY_ADJUSTED – використовується для побудови графіку дивідендних виплат, а також для значення дивідендої історія;

– EARNINGS – повертає дані про прибутки компанії, які використовуються для побудови графіку зросту прибутку та коефіцієнту зросту прибутку;

– TIME_SERIES_INTRADAY – історична інформація щодо ціни однієї акції, використовується для визначення останньої актуальної ціни акції, яка в свою чергу використовується для прорахунку коефіцієнту та числа Грема;

– SYMBOL_SEARCH – повертає список підприємств, які були знайдені за ключовим словом, яке користувач ввів у пошуковому полі.

3.7 Клієнтська частина програмного засобу

Для розробки клієнтської частини програмного засобу використовувався фреймворк Vue, який дозволяє планувати файлову структуру проекту за власним бажанням, хоч і пропонує деякі рекомендації. Таким чином сформувалась певна зручна структура, яку використовують багато команд у своїх проектах (рисунок 3.7).

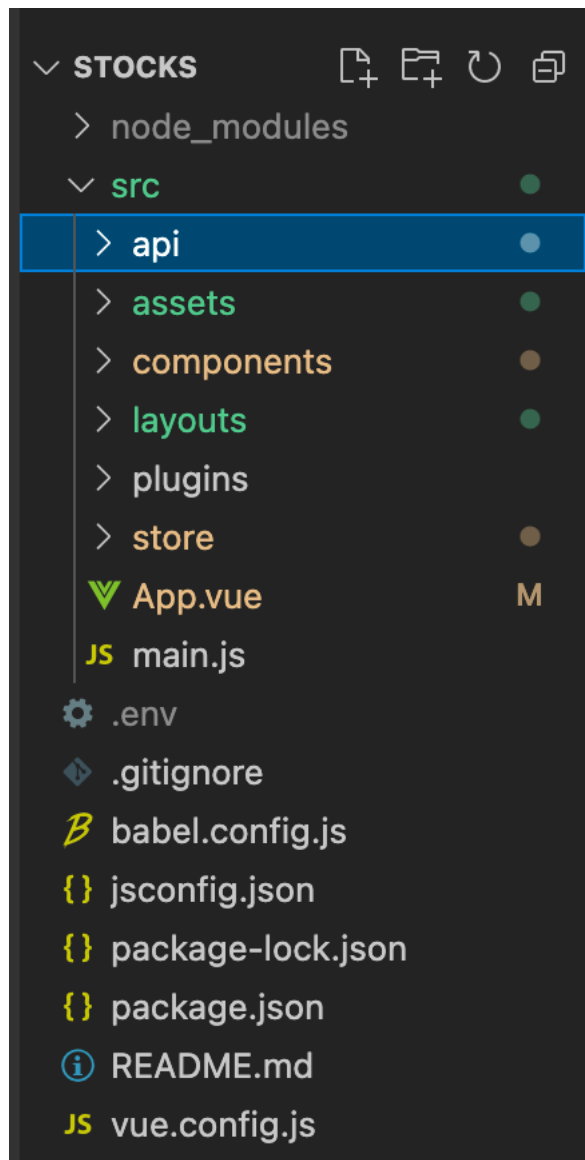


Рисунок 3.7 – Файлова структура програмного засобу

В першій директорії `node_modules` зберігаються всі встановлені бібліотеки і залежності, за допомогою яких функціонує проект. Версії відповідних пакетів зберігаються у файлі `package.json` у корені проекту (рисунок 3.8). Також в цьому файлі прописані скрипти – в даному контексті це ключові слова для команд, які можна виконувати із терміналу інтегрованої середовища розробки, залежності версій необхідних для роботи додатку пакетів, а також список браузерів які мають підтримувати написаний код.

```

{} package.json > {} dependencies
1  {
2    "name": "stocks",
3    "version": "0.1.0",
4    "private": true,
5    > Debug
6    "scripts": {
7      "dev": "vue-cli-service serve",
8      "build": "vue-cli-service build"
9    },
10   "dependencies": {
11     "@mdi/font": "^7.0.96",
12     "axios": "^0.27.2",
13     "chart.js": "^3.9.1",
14     "core-js": "^3.8.3",
15     "lodash.debounce": "^4.0.8",
16     "material-design-icons-iconfont": "^6.7.0",
17     "vue": "^2.6.14",
18     "vue-chartjs": "^4.1.1",
19     "vue-cli-plugin-vuetify": "^2.5.8",
20     "vuetify": "^2.6.6",
21     "vuex": "^3.6.2"
22   },
23   "devDependencies": {
24     "@vue/cli-plugin-babel": "~5.0.0",
25     "@vue/cli-service": "~5.0.0",
26     "sass": "^1.32.7",
27     "sass-loader": "^12.0.0",
28     "vue-template-compiler": "^2.6.14"
29   },
30   "browserslist": [
31     "> 1%",
32     "last 2 versions",
33     "not dead"
34 ]
35 }

```

Рисунок 3.8 – Контент файлу package.json у корені програмного коду

У файлі package-lock.json записані додаткові залежності, які необхідні для стабільної роботи кожного окремого пакету. Даний файл, зазвичай, контролюється не розробником додатку, а записи додаються у файл конфігурації розробником самого пакету.

Окремо можна виділити плагін, який підключений на строчці 23 – @vue/cli-plugin-babel. Даний плагін підключається у файл babel.config.js, який лежить у корені файлової структури проекту. Бібліотека Babel виконує перезапис JavaScript коду на менш актуальну версію, тим самим дозволяючи запускати програмний засіб на більш старих версіях браузерів, які не підтримують нові специфікації JavaScript.

У файловій структурі також створений необов'язковий файл `vue.config.js` (рисунок 3.9). Даний код додає можливість використовувати аліаси, які є складовими частинами шляху до файлів. Код нижче свідчить про те, що розробник замість прописаного абсолютного шляху до файлу може напряму звертатись до директорії `src` через символ `@`, і вже після символу прописувати наступні необхідні директорії. Для розробленого етапу лише одного аліасу достатньо, але конфігурація дозволяє додати декілька аліасів і спростити використання методу `import` і звертання до інших файлів.

```
JS vue.config.js > <unknown> > chainWebpack
Dmytro Ponomarenko, 2 months ago | 1 author (Dmytro Ponomarenko)
1  const path = require('path')
2
3  module.exports = {
4    chainWebpack: config => {
5      config.resolve.alias
6        .set('@', path.resolve(__dirname, 'src'));
7    }
8  }
```

Рисунок 3.9 – Контент файлу `vue.config.js`

Перша директорія, яка знаходиться у головній для Vue-проектів директорії `src`, є `api` (`src/api` або `@/api`). В даній директорії знаходиться все, що відповідає за обробку поступаючих з сервера (або статично змодельованих) даних. Бізнес-логіка програмного додатку знаходиться в директорії `services` (`@/api/services`), яка містить в собі три класи-сервіси, кожен з яких має конкретну функцію.

`@/api/services/StockMapper.js`. На цьому етапі відбувається ініціалізація підприємства (рисунок 3.10), яке бажає додати до списку користувач, у розробленому додатку. Саме цей клас делегує остаточну обробку інформації по підприємству, адже ініціалізація складається з запиту даних на сервер, першочергову обробку інформації при розрахунку необхідних для стратегії вартісного інвестування коефіцієнтів, і другочергову обробку інформації – отримання статусу коефіцієнту на основі рекомендованих значень, які представлені в таблиці відповідностей значень індикаторів і статусів.

```
async initialize() {  
  const [  
    overview,  
    incomeStatement,  
    monthlySummary,  
    balanceSheet,  
    earnings,  
    lastUpdatedPrice  
  ] = await Promise.all([  
    getOverview(this.symbol),  
    getIncomeStatement(this.symbol),  
    getMonthlySummary(this.symbol),  
    getBalanceSheet(this.symbol),  
    getEarnings(this.symbol),  
    getLastUpdatedPrice(this.symbol),  
  ])  
}
```

Рисунок 3.10 – Асинхронне отримання даних підприємства від серверу.

Оскільки щоб отримати кожну частину необхідною інформації потрібно зробити декілька рівноцінних запитів на сервер, був використаний метод `all` об'єкту `Promise`, завдяки якому всі запити ідуть паралельно. Таким чином зменшується кількість очікування користувачем, оскільки кожному запиту не потрібно чекати на закінчення попереднього. Наступним кроком для отриманих даних буде обробка сервісом `RatioService`, результат обробки якого відповідно буде надісланий до `StatusService` (рисунок 3.11), який остаточно сформує дані для того, щоб відобразити їх користувачу.

```

35     this.name = overview.Name;
36     this.industry = overview.Industry;
37     this.lastUpdatedPrice = lastUpdatedPrice;
38     this.overview = overview;
39     this.incomeStatement = incomeStatement;
40     this.monthlySummary = monthlySummary;
41     this.balanceSheet = balanceSheet;
42     this.earnings = earnings;
43
44     this.earningsByEachQuarter = RatioService.getEarningsByQuarter(incomeStatement.quarterlyReports);
45     this.dividendsByEachYear = RatioService.getDividendHistory(monthlySummary['Monthly Adjusted Time Series']);
46     this.earningsGrowth = RatioService.getEarningsGrowth(earnings.annualEarnings);
47     this.PERatio = +overview.PERatio;
48     this.PBRatio = +overview.PriceToBookRatio;
49
50     const currentRatio = RatioService.getCurrentRatio(balanceSheet);
51     const lastUnprofitableQuarter = RatioService.getLastUnprofitableQuarter(incomeStatement.quarterlyReports);
52     const lastUnprofitableYear = RatioService.getLastUnprofitableYear(incomeStatement.annualReports);
53     const GrahamRatio = RatioService.getGrahamRatio(balanceSheet, overview.SharesOutstanding);
54     const GrahamNumber = RatioService.getGrahamNumber(+this.overview.EPS, +this.overview.BookValue);
55
56     this.ratios = [
57         StatusService.getLastUpdatedPrice(this.lastUpdatedPrice),
58         StatusService.getCurrentRatio(currentRatio),
59         StatusService.getStableEarnings(lastUnprofitableYear, lastUnprofitableQuarter),
60         StatusService.getDividendHistory(this.dividendsByEachYear),
61         StatusService.getEarningsGrowth(this.earningsGrowth),
62         StatusService.getGrahamRatio(GrahamRatio, this.lastUpdatedPrice.price),
63         StatusService.getGrahamNumber(this.PERatio, this.PBRatio, GrahamNumber, this.lastUpdatedPrice.price),
64     ];
65
66     return this;
67

```

Рисунок 3.11 – Виклик сервісів для обробки даних підприємства у методі initialize класу StockMapper

`@/api/services/Ratio.js`. На цьому етапі відбувається прорахування коефіцієнтів, які необхідні для аналізу підприємства на відповідність стратегії вартісного інвестування. Клас складається з методів, кожен з яких прораховує конкретний коефіцієнт на основі даних, які були отримані на етапі ініціалізації підприємства. На прикладі метода отримання дивідендної історії для відповідно названого графіка показано, яким чином обробляється інформація у `RatioService`. Оскільки дані з серверу приходять в форматі, який не є зручним для клієнтського додатку, ключі у об'єкті результату перейменовуються. На першому етапі (строки 34-41) дані з серверу форматуються у зручний об'єкт, який містить в собі лише рік виплати дивідендів та їх кількість. Наступним кроком є групування і розрахування суми дивідендів за кожен рік, адже підприємство може виплачувати дивіденди декілька разів щорічно. Останнім кроком є фільтрування інформації, для того щоб результуючий масив містив лише дані про останні 20 років дивідендної історії, адже сервісу статусів цього буде достатньо щоб дати рекомендацію по пункту дивідендної історії.

`@/api/services/Status.js`. На цьому етапі обробки даних підприємства додається такий параметр, як статус коефіцієнту. Це одна з найголовніших бізнес-функцій програмного засобу, адже саме вона допомагає приймати рішення щодо покупки цінних паперів підприємства. Для користувача, особливо який не є досвідченим у питанні фундаментального аналізу цінних паперів, це взагалі найголовніше в аналізі підприємства, оскільки це допомагає йому визначитись, основуючись на рекомендованих значеннях, які присутні у таблиці відповідностей значень індикаторів і статусів. На прикладі відформатованих попереднім кроком даних про дивідендну історію показано яким чином сервіс `StatusService` оброблює дані про кожен коефіцієнт аналізу (рисунок 3.12). Для конкретно цього коефіцієнту, відповідно до таблиці відповідностей значень індикаторів і статусів, аналізуються дані про дивідендну історію, які були попередньо відформатовані сервісом `Ratio` (рисунок 3.13). Не рекомендований статус коефіцієнту надається в тому разі, якщо компанія наразі не виплачує дивіденди. Якщо компанія ніколи не виплачувала дивіденди останні 20 років, користувач при наведенні на значення коефіцієнту побачить на екрані повідомлення Компанія не платила дивіденди останні 20 років. Але якщо компанія виплачувала дивіденди за останні 20 років, чого не можна сказати про минулий рік (при аналізі приділяється увага саме попередньому року, адже поточний рік ще фактично не завершився і компанія має змогу виплатити дивіденди), користувач бачить повідомлення Компанія наразі не виплачує дивіденди. Задовільний статус надається підприємству, яке наразі виплачує дивіденди, але робить це не стабільно на протязі двадцяти останніх років. При будь-яких інших сценаріях, які є рекомендованими для стратегії вартісного інвестування, коефіцієнту надається статус рекомендовано. Бейдж становиться зеленого кольору, що свідчить про те що система наголошує на тому що даний коефіцієнт підприємства має рекомендоване значення.

```

33 static getDividendHistory(dividendHistory) {
34   const dividendsAccordingMonths = Object.entries(dividendHistory).map(data => {
35     const [date, infoByMonth] = data;
36
37     return {
38       year: date.slice(0, 4),
39       dividendAmount: +infoByMonth['7. dividend amount']
40     };
41   });
42
43   const dividendsAmountByEachYear = dividendsAccordingMonths.reduce((acc, dataForMonth) => {
44     const currentYearInAcc = acc.find(yearInfo => yearInfo.year === dataForMonth.year)
45
46     if (!currentYearInAcc) {
47       acc.push({ year: dataForMonth.year, dividendAmount: dataForMonth.dividendAmount })
48     } else {
49       currentYearInAcc.dividendAmount = +(currentYearInAcc.dividendAmount + dataForMonth.dividendAmount).toFixed(2);
50     }
51
52     return acc;
53   }, []);
54
55   const currentYear = new Date().getUTCFullYear();
56   const endDateRange = currentYear - 20;
57
58   const dividendsForLastPeriod = dividendsAmountByEachYear
59     .filter(({ year }) => year <= currentYear && year >= endDateRange)
60
61   return dividendsForLastPeriod.reverse();
62 }
63

```

Рисунок 3.12 – Форматування дивідендної історії підприємства

```

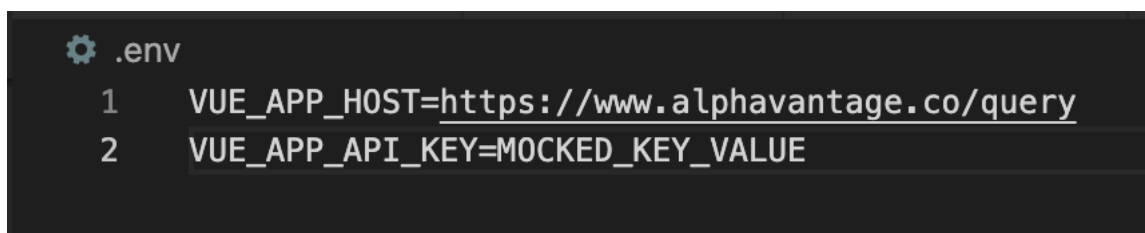
66 static getDividendHistory(dividendsByEachYear) {
67   const title = 'Дивідендна історія';
68   const key = 'dividends';
69   const yearsWithDividends = dividendsByEachYear.filter(year => year.dividendAmount);
70   let status;
71   let ratio;
72   let message;
73
74   const currentYear = new Date().getUTCFullYear();
75   const previousYear = currentYear - 1;
76   const hadLastYearPayment = yearsWithDividends.find(({ year }) => +year === currentYear || +year === previousYear);
77   const IDEAL_DIVIDENDS_PAYMENT_RANGE = 20;
78
79   if (!yearsWithDividends.length) {
80     status = danger;
81     ratio = 'N/3';
82     message = 'Компанія не платила дивіденди останні 20 років';
83   } else if (!hadLastYearPayment) {
84     status = danger;
85     ratio = 'N/3';
86     message = 'Компанія наразі не виплачує дивіденди'
87   } else if (hadLastYearPayment && yearsWithDividends.length < IDEAL_DIVIDENDS_PAYMENT_RANGE) {
88     status = warning;
89     ratio = previousYear;
90     message = 'Компанія наразі виплачує дивіденди';
91   } else if (yearsWithDividends.length === IDEAL_DIVIDENDS_PAYMENT_RANGE) {
92     status = success;
93     ratio = '✓';
94     message = 'Компанія платить дивіденди останні 20 років';
95   }
96
97   return { status, ratio, message, title, key };
98 }

```

Рисунок 3.13 – Обчислення статусу коефіцієнту дивідендної історії

Варто зазначити, що для кожного асинхронного запиту на сервер необхідний унікальний ключ, без якого API не може повернути будь-які дані, які необхідні програмному додатку. API отримує ключ як один з параметрів get-запиту від клієнту, як вказано на 17 строчці (рисунок 3.6). Даний підхід використовується з безпекових міркувань, а також для того щоб монетизувати

API. Отримуючи доступ до унікального ключа, небезпечно зберігати ключ у програмному коді додатку, адже зловмисники зможуть скопіювати його та використовувати у власних цілях, витрачаючи ліміт запитів на хвилину, тим самим перевантажуючи шлях доступу до даних. Щоб уникнути цієї проблеми, фреймворк Vue пропонує зберігати ключ у так званому `.env` файлі (рисунок 3.14), який зазвичай не завантажується у систему контролю версії Git. Це вдається за допомогою іншого конфігураційного файлу `.gitignore`, вписуючи відносний шлях файлу, який необхідно проігнорувати при завантаженні у Git. Зазвичай `.env` файл знаходиться на тому ж рівні, що і `.gitignore`, тому необхідно просто написати назву файлу на новій строчці. Для прикладу, значення ключа було змінено на `MOCKED_KEY_VALUE`, хоча насправді значення ключа генерується випадковим чином незалежно від користувача, дати отримання ключа або будь-якого іншого впливу. Також у файлі прописаний `VUE_APP_HOST`, який посилається на URL, значення якого використовується як URL при кожному запиті на сервер (рисунок 3.6).



```
1 VUE_APP_HOST=https://www.alphavantage.co/query
2 VUE_APP_API_KEY=MOCKED_KEY_VALUE
```

Рисунок 3.14 – Контент файлу конфігурації середовища `.env`

Важливою деталлю для будь-якого проекту є статичні дані, константи, які не є динамічними і можуть використовуватись у декількох місцях програми одночасно. Через цей фактор зручно зберігати такі дані згруповано в одному місці програми. Для цього у проекті існує директорія `@/api/static`, де зберігаються такі константні дані як інформація про статуси коефіцієнтів (рисунок 3.15). Таким же чином у проекті функціонує файл `@/assets/variables.scss`, в який винесено кольори та деякі стилі, адже ці змінні використовуються в різних місцях програмного коду.

Структура статусів представлена у вигляді об'єкту для зручності, адже в деяких місцях програми зручніше звертатись до статусу по конкретному ключу за назвою статусу, а в інших місцях – знаходити статус по ключу value.

```
src > api > static > JS statuses.js > [⌘] default
1  export default {      Dmytro Ponomarenko, 2 months ago • Master
2      danger: { value: 'danger', color: 'red' },
3      warning: { value: 'warning', color: 'orange' },
4      success: { value: 'success', color: 'green' },
5      secondary: { value: 'secondary', color: 'grey' },
6  }
7  |
```

Рисунок 3.15 – Інформація про статуси, винесена в окремий файл

Файли, які відповідають за відображення інформації користувачу, розподілені на дві директорії: `@/global-components` та `@/layouts`. Глобальні компоненти – компоненти, які використовуються декілька разів у різних місцях програмного коду і не прив'язані до будь-якого великого блоку інтерфейсу. Для кожного великого блоку інтерфейсу була створена окрема директорія:

- `@/layouts/header` – містить в собі верхню частину інтерфейсу, так званий header, в якому в свою чергу знаходиться пошукове поле, в яке користувач може вписувати ключові слова для пошуку необхідного підприємства;

- `@/layouts/sidebar` – містить в собі список раніше проаналізованих підприємств, які були додані у список обраних підприємств або були проаналізовані у поточній сесії браузера;

- `@/layouts/details` – містить в собі компоненти, які відображають розширену інформацію про підприємство, яке є активним на сторінці, а саме формули, графіки та мета-інформацію підприємства.

В кожній з директорій функціонує кореневий файл `index.vue`, який містить в собі інші дочірні компоненти, які відображаються на сторінці (рисунок 3.16).

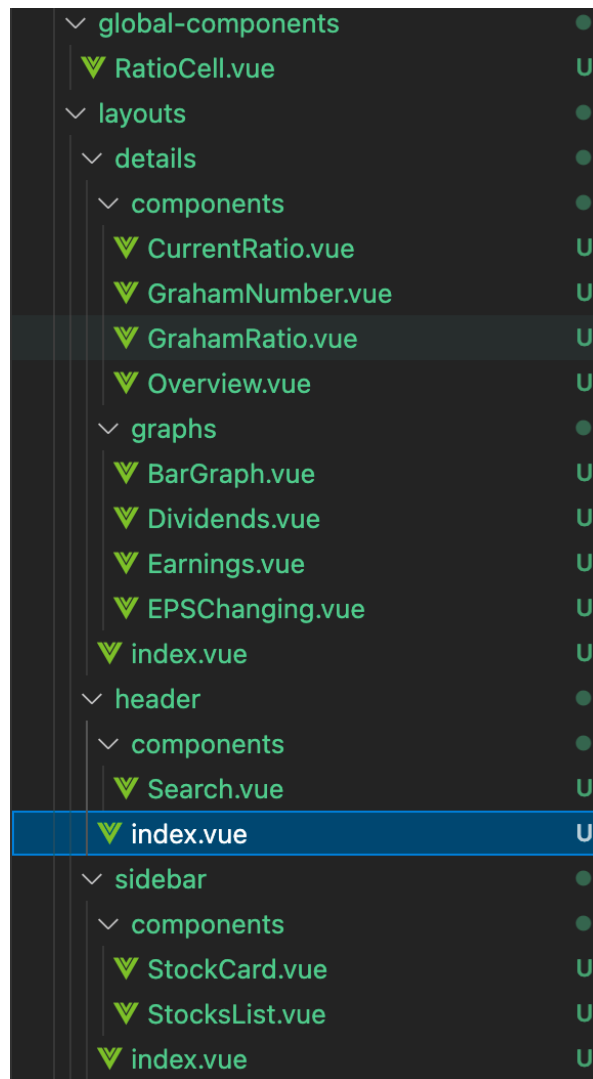


Рисунок 3.16 – Файлова структура компонентів, відповідаючих за відображення інформації на екран користувача

Однією з найголовніших переваг фреймворку Vue є менеджер станів Vuex – глобальне сховище даних, доступ до яких є у кожному компоненті проекту. Це зручно тим, що зміни можуть відбуватись з будь-якого компоненту і Vuex отримує запит на кожну зміну глобальних даних і оновлює їх, таким чином підтримуючи актуальність даних всюди. Vue, використовуючи метод мови JavaScript `Object.defineProperty`, оновлює дані реактивно, завдяки чому вони оновлюються не тільки у глобальному сховищі, але й у розмітці, тобто на екрані у користувача, автоматично. Всі методи, які викликають зміни у Vuex, винесені окремо у директорію `@/store/stock/actions` (рисунок 3.17), а саме: отримання даних про підприємство, встановлення активного підприємства на сторінці,

видалення підприємства зі списку, синхронізація списку обраних підприємств зі списком, збереженим браузером тощо. Проміжна директорія `stock` була додана для потенційного розширення глобального сховища на інші сутності.

```
src > store > stock > JS actions.js > [⌘] default
1 // import { mockedData } from '@api/static/mocked-data'; for testing
2 import {
3   getSearchResults,
4 } from '@api';
5
6 import StockMapper from '@api/services/StockMapper.js';
7 import { syncFavoritesStocksWithLocalStorage } from '@api/helpers';
8
9 export default {
10  async getStockBySymbol({ commit }, symbol) {
11    const stockMapper = new StockMapper(symbol);
12    const stock = await stockMapper.initialize();
13
14    commit('addStockDataToList', stock);
15  },
16
17  setActiveStock({ commit }, symbol) {
18    commit('setActiveStock', symbol);
19  },
20
21  removeStockFromList({ commit, dispatch }, symbol) {
22    commit('removeStockFromList', symbol);
23    dispatch('syncStocksWithFavorites', { symbol, action: 'remove' });
24  },
25 }
```

Рисунок 3.17 – Методи, які оновлюють глобальне сховище даних Vuex

3.8 Інтерфейс програмного засобу

Дизайн програмного засобу (рисунок 3.18) проектувався з огляду на те, що більшість користувачів не будуть досвідченими в питанні фундаментального аналізу підприємства, а отже інтерфейс має бути наскільки це можливо простим та інтуїтивно-зрозумілим для будь-якого користувача [19]. Кожен прорахований коефіцієнт відображається у геометричній фігурі з відповідному до статусу кольору – зеленому, жовтому або червоному. Зі шкільної програми відомо, що червоний колір означає небезпеку, а зелений навпаки сприймається спокійно і символізує безпеку. Жовтий, в контексті програмного засобу, є проміжним статусом між рекомендованим і не рекомендованим, не сприймається

користувачем радикально як негативне значення, але й змушує звернути увагу і додатково зважити своє рішення [18]. Ціна однієї акції підприємства відображена у фігурі нейтрального синього кольору, адже тільки після аналізу коефіцієнтів стратегії вартісного інвестування користувач може зрозуміти, чи є ціна паперу справедливою.

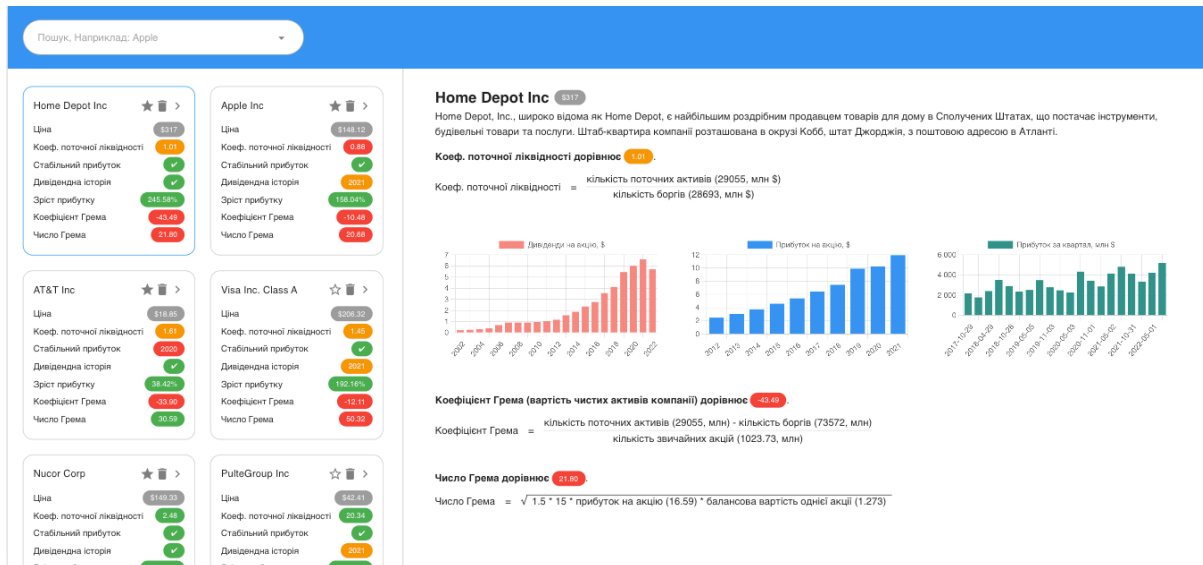


Рисунок 3.18 – Інтерфейс програмного засобу

У верхній частині екрану, яку розробники називають header, розташоване пошукове поле, завдяки якому користувач за допомогою ключових слів має змогу знайти необхідне підприємство для аналізу. Після вводу ключового слова, додаток посилає запит на сервер на отримує список підприємств, які знайшлись по відповідному запиту у базі даних Alpha Vantage (рисунок 3.19).

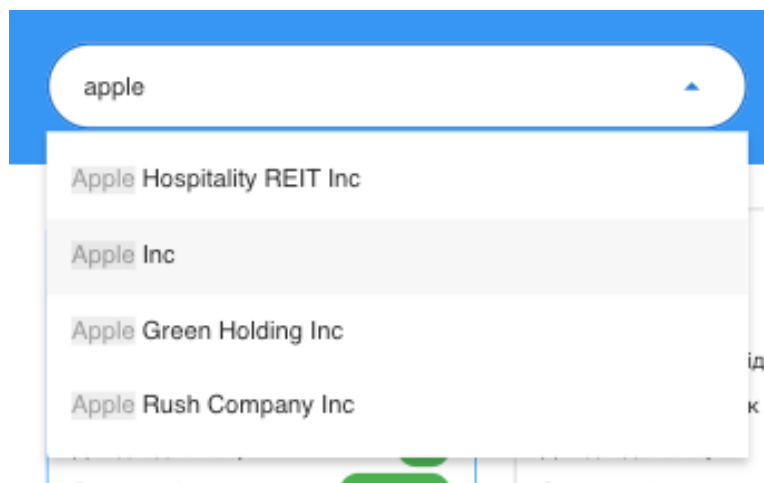


Рисунок 3.19 – Список знайдених підприємств по запиту apple

Користувач може вибрати будь-яке з них, і після цього підприємство буде додане у список проаналізованих підприємств (рисунок 3.20). Інформація про кожне підприємство відображається у вигляді картки (рисунок 3.21), яка містить в собі назву підприємства, іконки для швидких дій, таких як додати в обране, видалити зі списку та зробити активним, а також назву та значення коефіцієнта з відповідним статусом. Активним є те підприємство, яке користувач додав останнім у список або обрав самостійно через відповідну іконку. На інтерфейсі, активне підприємство виділене рамкою синього кольору, який є основним для додатку – він використовується для верхньої частини додатку, а також як колір графіку прибуток на акцію. Синій колір є більш нейтральним, ніж кольори які використовуються для статусів, а отже не буде відволікати користувача від основної функціональності додатку, але додає фірмовий стиль та відокремить інтерфейс додатку від верхньої частини інтерфейсу самого браузера.

<p>Home Depot Inc ★ 🗑️ ></p> <p>Ціна \$317</p> <p>Коеф. поточної ліквідності 1.01</p> <p>Стабільний прибуток ✓</p> <p>Дивідендна історія ✓</p> <p>Зріст прибутку 245.58%</p> <p>Коефіцієнт Грема -43.49</p> <p>Число Грема 21.80</p>	<p>Apple Inc ★ 🗑️ ></p> <p>Ціна \$148.12</p> <p>Коеф. поточної ліквідності 0.88</p> <p>Стабільний прибуток ✓</p> <p>Дивідендна історія 2021</p> <p>Зріст прибутку 158.04%</p> <p>Коефіцієнт Грема -10.48</p> <p>Число Грема 20.68</p>
<p>AT&T Inc ★ 🗑️ ></p> <p>Ціна \$18.85</p> <p>Коеф. поточної ліквідності 1.61</p> <p>Стабільний прибуток 2020</p> <p>Дивідендна історія ✓</p> <p>Зріст прибутку 38.42%</p> <p>Коефіцієнт Грема -33.90</p> <p>Число Грема 30.59</p>	<p>Visa Inc. Class A ☆ 🗑️ ></p> <p>Ціна \$206.32</p> <p>Коеф. поточної ліквідності 1.45</p> <p>Стабільний прибуток ✓</p> <p>Дивідендна історія 2021</p> <p>Зріст прибутку 192.16%</p> <p>Коефіцієнт Грема -12.11</p> <p>Число Грема 50.32</p>

Рисунок 3.20 – Список аналізованих підприємств у поточній сесії користувача

1	Apple Inc	2	3	4	
					6
	Ціна				
5	Коеф. поточної ліквідності				
	Стабільний прибуток				7
	Дивідендна історія				
	Зріст прибутку				
	Коефіцієнт Грема				
	Число Грема				

Рисунок 3.21 – Картка з загальною інформацією підприємства

де, 1 – Повна назва підприємства. Верстка спроектована таким чином, що якщо назва буде занадто довгою і не буде розміщуватись у картці, назва обрізається і останні символи заміщаються трикрапкою.

2 – Іконка для додання підприємства до списку обраних підприємств. Якщо підприємство вже є у списку, іконка є залитою сірим кольором, в іншому випадку відображається лише силует іконки, що свідчить про те що підприємство не додане у список обраного. Після перезавантаження сторінки, вся інформація про підприємства, що були додані у список обраного, знову буде завантажена з максимально актуальними даними.

3 – Іконка видалення. Видаляє підприємство не тільки з списку проаналізованих карток поточною сесією браузера, але й зі списку обраних підприємств.

4 – Іконка встановлення підприємства в активне положення. Після натискання на іконку, до картки додається рамка синього кольору, що свідчить про те що підприємство стає у активне положення. Праворуч екрану відобразиться детальна інформація про підприємство: графіки дивідендів на акцію, прибутку на акцію, прибутку за квартал, а також детальна інформація про обчислення коефіцієнтів стратегії вартісного інвестування.

5 – Назви коефіцієнтів.

6 – Ціна акції. При наведенні на ціну з'являється повідомлення з інформацією про останнє оновлення ціни однієї акції підприємства.

7 – Прораховані коефіцієнти, відповідно назвам, а також з автоматично прорахованими статусами, такими як рекомендовано, задовільно та не рекомендовано з відповідними до назви статусів кольорами.

Праворуч на екрані користувача відображається розширена інформація про розраховані коефіцієнти (рисунок 3.22), а також невеликий опис проаналізованого підприємства. Кожному коефіцієнту відповідає або формула, завдяки якій коефіцієнт був прорахований, або графік, на якому там відображені дані, за допомогою яких було сформоване значення коефіцієнту. Ці дані є дуже важливими для користувача, адже завдяки ним він, при бажанні, може

перевірити коректність розрахунків, пересвідчитись в тому які саме дані необхідні для фундаментального аналізу підприємства, а також побачити закономірність на графіках, наприклад, в якому кварталі підприємство отримує найбільший прибуток чи який рух виплати дивідендів підприємства. При аналізі декількох схожих за значеннями коефіцієнтів підприємств, це може зіграти першочергову роль у прийнятті рішення.

Apple Inc \$144.44

«Apple Inc. — це американська багатонаціональна технологічна компанія, яка спеціалізується на споживчій електроніці, комп'ютерному програмному забезпеченні та онлайн-сервісах. Apple є найбільшою технологічною компанією в світі за доходом (загалом \$274,5 млрд у 2020 році), а з січня 2021 року — найдорожчою компанією в світі. Станом на 2021 рік Apple є четвертим у світі постачальником ПК за обсягом продажів і четвертим за величиною виробником смартфонів. Вона входить до «Великої п'ятірки» американських компаній інформаційних технологій разом із Amazon, Google, Microsoft і Facebook».

Коеф. поточної ліквідності дорівнює 0.88.

Коеф. поточної ліквідності = $\frac{\text{кількість поточних активів (135405, млн \$)}}{\text{кількість боргів (153982, млн \$)}}$



Коефіцієнт Грема (вартість чистих активів компанії) дорівнює -10.48.

Коефіцієнт Грема = $\frac{\text{кількість поточних активів (135405, млн)} - \text{кількість боргів (302083, млн)}}{\text{кількість звичайних акцій (15908.1, млн)}}$

Число Грема дорівнює 20.90.

Число Грема = $\sqrt{1.5 * 15 * \text{прибуток на акцію (6.11)} * \text{балансова вартість однієї акції (3.178)}}$

Рисунок 3.22 – Розширена інформація щодо коефіцієнтів підприємства

3.9 Висновки до розділу

На даному етапі був створений програмний засіб, а саме написаний програмний код, використовуючи фреймворк Vue. Створений інтерфейс, згідно спроектованого дизайну та врахованими до цільової аудиторії потребами. Проведено тестування, згідно до найчастіших дій та запитів користувача, у зв'язку з цим проведена робота над помилками і їх виправлення.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз умов праці в лабораторії

Приміщення, де ведеться проектування, являє собою лабораторію. Площа Даного приміщення становить 60 м^2 , обсяг – 210 м^3 , 2 робочих місця. Розміри приміщення $6 \text{ м} \times 10 \text{ м} \times 3,5 \text{ м}$.

Електроживлення здійснюється від трифазної мережі з глухозаземленою нейтраллю змінною напругою $220/380 \text{ В}$, частотою 50 Гц [20].

Для забезпечення нормальних умов праці ДСанПіН 3.3.2-007-98 встановлює на одне робоче місце площу виробничого приміщення не менше 6 м^2 , висота приміщення повинна бути не менше $3,2 \text{ м}$, об'єм повітряного простору 20 м^3 . В даному випадку реальна площа на одного працюючого становить 16 м^2 , об'єм 56 м^3 , що відповідає санітарним нормам.

Розглядаючи людину в нерозривному зв'язку і в безперервній обмінній взаємодії з навколишнім середовищем, як об'єкт вивчення можна виділити систему Людина-машина-середовище (Л-М-С), а предметом вивчення безпеки і її вплив на людину в процесі її функціонування і розвитку системи Л-М-С. головним в системі Л-М-С є безпека людини.

Згідно ДСТУ 12.0.003-74 можна виділити небезпечні та шкідливі виробничі фактори в приміщенні.

4.2 Промислова безпека у лабораторії

Електричне живлення устаткування здійснюється від трифазної мережі з глухозаземленою нейтраллю напругою $220/380 \text{ В}$, частотою 50 Гц . Захист персоналу від ураження електричним струмом, необхідно здійснювати з допомогою занулення НПАОП 40.1-1.32-01. Для цього з'єднати металеві неструмопровідні частини обладнання з нульовим проводом мережі, за

допомогою алюмінієвого дроту, перетин якого дорівнює перерізу фазного проводу мережі.

При замиканні фази на занулення корпусу електроустановка автоматично відключається, якщо значення струму однофазного короткого замикання I_k , А, задовольняє умові:

$$I_k \geq kI_{ном}, \quad (4.1)$$

де $I_{ном}$ – номінальний струм плавкої вставки запобіжника або струм спрацьовування автоматичного вимикача, А;

k – коефіцієнт кратності струму. Він приймається в залежності від типу захисту електроустановки. Якщо захист здійснюється автоматичним вимикачем, що має тільки електромагнітний розчіплювач, тобто, який спрацьовує без витримки часу, то k приймається в межах 1,25. Якщо установка захищена плавкими запобіжниками, час перегорання яких залежить від величини струму, то k приймають ≥ 3 (у вибухонебезпечних приміщеннях ≥ 4).

$$I_{кз} = \frac{U_{\phi}}{\left(\frac{Z_T}{3} + Z_{\Pi}\right)}, \quad (4.2)$$

де $Z_T/3$ – повний опір обмоток трансформатора, визначається виходячи з потужності трансформатора, $Z_T/3 = 0,226$ Ом;

Z_{Π} – опір петлі фаза-нуль; $Z_{\Pi} = 0.8$ Ом.

Розрахування дійсного струму $I_{кз}$ за формулою (4.2):

$$I_{кз} = \frac{220}{\left(\frac{0,226}{3} + 0,8\right)} = 251\text{А}$$

З умови формули (4.1):

$$I_{\text{н}} \leq \frac{I_{\text{кз}}}{k}, \quad (4.3)$$

де $k = 1.25$ коефіцієнт кратності струму.

$$I_{\text{н}} \leq \frac{251}{1,25} = 200\text{А}$$

Відповідно до НПАОП 0.00-4.12-05, всі працівники проходять інструктажі з охорони праці: вступний, первинний на робочому місці, повторний і, при необхідності, позаплановий та цільовий.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи основну увагу приділено розробленню системи підтримки прийняття рішень для управління фінансовим капіталом виробничих підприємств.

У першому розділі проведено аналіз схожих систем підтримки прийняття рішень, в яких було виділено ряд недоліків, зокрема: громіздкий не інтуїтивно-зрозумілий інтерфейс; відображення всього лише фактичних значень коефіцієнтів, але без якісних статусів відповідно рекомендаційних значень; відсутність української розкладки.

У другому розділі було створено структурну схему роботи системи, алгоритм роботи програми, а також технічну схему роботи програми, які відображають яким чином працює програма та які саме кроки відбуваються між отриманням запиту від користувача до відображення на екрані повністю відформатованих даних підприємства. Також в розділі представлена інформація щодо даних, які необхідні для прорахунку кожного коефіцієнту, а також теоретична інформація щодо стратегії вартісного інвестування.

У третьому розділі представлено технічне забезпечення системи підтримки прийняття рішень, обґрунтовані технічні рішення щодо вибору мови програмування, фреймворку для клієнтської розробки, а також додаткових бібліотек для клієнтської розробки та API для отримання даних підприємства. Розглянуто кроки, які необхідні для розробки системи підтримки прийняття рішень від етапу зародження ідеї до реалізації. Розроблено простий та інтуїтивно-зрозумілий для цільової аудиторії інтерфейс.

Завершальним кроком кваліфікаційної роботи був успішний експеримент, проведений користувачем при введенні у пошукове поле даних підприємства Apple Inc., і отримання усіх очікуваних результатів при відображенні розширеної інформації на екран користувача.

Для подальшого розвитку системи планується:

- функціональність авторизації користувача для збереження даних не у локальному сховищі у браузері, а у хмарному сховищі даних;
- для того, щоб недосвідчений користувач мав змогу краще скласти свій інвестиційний портфель, планується додавання списку рекомендованих для купівлі паперів підприємств на основі коефіцієнтів стратегії вартісного інвестування;
- для того, щоб користувач мав змогу краще проводити диверсифікацію інвестиційного портфелю, планується додавання окремої сторінки, вхідними даними для якої буде список підприємств, які система зможе проаналізувати на основі даних про індустрію та сектор економіки, і відобразити рекомендацію щодо достатнього коефіцієнту диверсифікації портфелю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП «УкрНДНЦ», 2016. – 30 с.
3. Луців Б. Л. Інвестування: підручник / Б. Л. Луців, І. С. Кравчук, Б. Б. Сас. – Тернопіль: Економічна думка, 2014. – 544 с.
4. Грем Б. Розумний Інвестор. Стратегія вартісного інвестування. – Київ: Наш Формат, 2019. – 123 с.
5. Stanhope P. Get in groove: Building Tools and Peer-to-Peer Solutions with the Groove Platform, – 2019. – 34 p.
6. Cioca M. Decision Support Systems – A bibliography, 2015, – 10 p.
7. Tkachuk R. L. Logic-cognitive models of temporal reality when taking operational decisions in crisis conditions of functioning of technological systems (Part I) // Scientific Bulletin of UNFU, – 2018. – pp. 107-116.
8. Man Sing Wong. Development of an Integrated MicroEnvironmental Monitoring System for Construction Sites. Procedia Environmental Sciences. – 2016. – pp. 207-214.
9. П. І. Бідюк. Системи і методи підтримки прийняття рішень : підручник / Упоряд. П. І. Бідюк, О. Л. Тимошук, А. Є. Коваленко, Л. О. Коршевніюк. – Київ: КПІ ім. Ігоря Сікорського, 2022. – С. 43-45.
10. Yahoo.finance. URL: <https://finance.yahoo.com/> (дата звернення: 29.10.2022).

11. Investingт. URL: <https://www.investing.com/> (дата звернення: 29.10.2022).
12. Finviz. URL: <https://finviz.com/> (дата звернення: 29.10.2022).
13. dcydj.com. URL: <https://dcydj.com/> (дата звернення: 29.10.2022).
14. gurufocus.com. URL: <https://www.gurufocus.com/> (дата звернення: 29.10.2022).
15. Graham B., Dodd D. Security Analysis. Whittlesey House, McGraw-Hill Book Co., 1934. – 725 p.
16. Gadomski A., An Approach to the Intelligent Decision Advisor (IDA) for Emergency Managers, – 2015, – 12 p.
17. Material Design Specification. URL: <https://m3.material.io> (дата звернення: 02.12.2022)
18. Borges J., Computer-based tools for supporting forest management. The experience and the expertise world-wide, 2014. – 113 p.
19. Молодецька-Гринчук К. В. Метод виявлення ознак інформаційних впливів у соціальних інтернет-сервісах за змістовними ознаками / К. В. Молодецька-Гринчук – 2017. – С. 80.
20. Дзюндзюк Б. В. Охорона праці. Збірник задач: навч. посібник/ Б. В. Дзюндзюк. – Харків: ХНУРЕ, 2006. – 236 с.