

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ім. Ю.М. Потебні  
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

**КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

## **Кваліфікаційна робота**

**Перший (бакалаврський)**

(рівень вищої освіти)

на тему **Використання Django REST фреймворку для розробки  
автоматичного аналізатора лінії подій та результатів букмекерських  
контор**

Виконав: студент 4 курсу, групи 6.1219-пзс  
спеціальності 121 Інженерія програмного  
забезпечення

(код і назва спеціальності)

освітньої програми Програмне  
забезпечення систем

(код і назва освітньої програми)

О. І. Степаненко

(ініціали та прізвище)

Керівник доцент, к.т.н., доцент О. М. Михайлуца  
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя  
2023

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ**  
**ім. Ю.М. Потебні**  
**ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Спеціальність 121 Інженерія програмного забезпечення  
(код та назва)

Освітня програма Програмне забезпечення систем  
(код та назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_ Т.В. Критська  
“ 01 ” \_\_\_\_\_ березня 2023 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

\_\_\_\_\_  
Степаненку Олегу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Використання Django REST фреймворку для розробки автоматичного аналізатора лінії подій та результатів букмекерських контор

керівник роботи Міхайлуца Олена Миколаївна, доцент, к.т.н.  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від \_\_\_\_\_ від 29.12.2022 1893-с

2. Строк подання студентом кваліфікаційної роботи \_\_\_\_\_ 14.06.2023

3. Вихідні дані бакалаврської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми збору та аналізу даних з веб-сторінок
- створення програмного продукту та його опис;
- дослідження поставленої проблеми та розробка висновків.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
слайдів презентації

## 6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.03.2023

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1	Аналіз предметної області	24.04.2023	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	30.04.2023	виконано
3	Аналіз існуючих методів рішення	05.05.2023	виконано
4	Аналіз методів розробки аналізаторів та веб скраперів	11.05.2023	виконано
5	Аналіз сучасних веб фреймворків для роботи з парсерами даних	15.05.2023	виконано
6	Узгодження подальших дій з науковим керівником	16.05.2023	виконано
7	Програмна реалізація парсера даних	20.05.2023	виконано
8	Програмна реалізація телеграм бота	28.05.2023	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	29.05.2023	виконано
10	Реалізація користувачького інтерфейсу	04.06.2023	виконано
11	Перевірка роботоздатності проекту	08.06.2023	виконано
12	Оформлення звіту	10.06.2023	виконано
13	Оформлення презентації. Отримання рецензій від опонентів.	14.06.2023	виконано

Студент \_\_\_\_\_ Степаненко О.І.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Міхайлуца О.М.  
( підпис ) (прізвище та ініціали)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_ Скрипник І.А.  
( підпис ) (прізвище та ініціал)

## АНОТАЦІЯ

Сторінок: 70

Рисунків: 15

Таблиць: 2

Джерел: 16

Степаненко О.І. Використання Django REST фреймворку для розробки автоматичного аналізатора лінії подій та результатів букмекерських контор : кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. Керівник О.М. Михайлуца. Запоріжжя : ЗНУ, 2023.

Мета і завдання роботи полягає у дослідженні теми букмекерської справи та її тісного відношення до сфери інформаційних технологій у сучасному світі. На основі цих даних необхідно створити додаток який буде спрямований на широкий ринок користувачів та буде висвітлювати проблематику та нормативний погляд на цю тему.

У процесі вивчення теми були порушені такі проблеми як: залежність від грального спрямування, недостатня регуляція, вплив третіх сторін як на спортивні матчі так і на інформаційні системи корегування сторін, не цілеспрямованість витрат. Як результат був створений додаток на базі Django фреймворку для аналізу та порівняння лінії спортивних подій та результатів букмекерських контор. Даний додаток зчитує коефіцієнти з лінії подій різноманітних букмекерських контор, показуючи їх користувачу, в той же час проводить аналіз зчитаних подій на момент помилки або спроби шахрайства. Після ідентифікації порушення користувач повідомляється через заздалегідь підписаного телеграм бота.

Ключові слова: *букмекерська контора, лінія подій, коефіцієнт, телеграм бот, Django фреймворк, зчитування.*

## SUMMARY

Pages: 70

Drawings: 15

Tables: 2

Source: 16

Stepanenko O.I. Use of the Django REST framework for the development of an automatic analyzer of the line of events and results of bookmakers: qualification work of the bachelor of specialty 121 "Software engineering" / Science. Head O.M. Mihailutsa Zaporizhzhia: ZNU, 2023.

The purpose and task of the work consists in researching the subject of bookmaking and its close relation to the field of information technologies in the modern world. On the basis of this data, it is necessary to create an application that will be aimed at a wide market of users and will highlight the issues and the normative view on this topic.

In the process of studying the topic, such problems were raised as: dependence on the playing direction, insufficient regulation, influence of third parties both on sports matches and on the information systems of adjusting the parties, lack of purposefulness of expenses. As a result, an application was created based on the Django framework for analyzing and comparing the line of sports events and the results of bookmakers. This application reads the coefficients from the event line of various bookmakers, showing them to the user, at the same time it analyzes the read events at the time of an error or attempted fraud. After identifying a violation, the user is notified via a pre-signed Telegram bot.

Keywords: *bookmaker, event line, coefficient, telegram bot, Django framework, reading.*

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ АНАЛІЗУ ЛІНІЇ ПОДІЙ БУКМЕКЕРСЬКИХ КОНТОР .....	11
1.1 Огляд проблеми аналізу та збору інформації з букмекерських контор .....	11
1.2 Огляд проблематики зчитування даних з веб-джерел, з точки зору права та законодавства .....	12
1.3 Огляд літературних джерел .....	13
1.4 Види технологій для зчитування та парсингу даних з веб сторінок .....	14
1.5 Сфери використання аналізатора лінії подій букмекерських контор .....	16
1.6 Приклади застосунків аналізаторів .....	17
1.7 Аналіз існуючих програмних систем парсингу .....	18
2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ .....	26
2.1 Аналіз сучасних фреймворків для створення веб застосунків.....	26
2.1.1 Огляд фреймворків для створення веб застосунків .....	30
2.2 Аналіз сучасних технологій для створення телеграм ботів .....	35
2.3 Аналіз бібліотек для веб скрапінгу .....	39
3 РОЗРОБКА АНАЛІЗАТОРА ЛІНІЇ ПОДІЙ БУКМЕКЕРСЬКОЇ КОНТОРИ .....	45
3.1 Опис предметної області .....	45
3.2 Архітектура системи.....	46
3.3 Функціональні вимоги системи.....	46
3.4 Проектування системи проекту .....	49
3.5 Програмна реалізація застосунку .....	54
3.5.1 Програмна реалізація парсингової частини .....	54
3.5.2 Програмна реалізація інтерфейсу користувача .....	63

3.6 Демонстрація веб-інтерфейсу користувача.....	63
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68

## ВСТУП

### **Актуальність теми**

Популярність грального бізнесу, зокрема ставки на спорт, мають значний попит серед людей, які цікавляться спортом і хочуть відчутти додатковий адреналін, спробувати свою удачу або виявити свої аналітичні навички. Букмекерські контори надають можливість зробити ставки на різноманітні спортивні події, що привертає увагу широкої аудиторії.

Створення аналізатора лінії подій букмекерських контор може бути вигідним та актуальним як для простих гравців так і для самої букмекерської контори яка вирішить придбати аналізатор для свого попиту. Аналізатор може допомогти конторі отримати конкурентну перевагу, надаючи клієнтам більш детальну та об'єктивну інформацію про спортивні події, що дозволяє зробити кращі розрахунки та ставки.

Також продукт надає звичайним гравцям можливість отримати доступ до статистики, оцінок команд або гравців, прогнозів та інших факторів, які впливають на результати спортивних подій. Це допомагає клієнтам зробити більш обґрунтовані та успішні ставки.

Продукт допоможе конторі виявляти потенційні ризики та знижувати вплив невизначеності на її фінансові результати. Шляхом систематичного аналізу та прогнозування подій, контора може зменшити втрати і збільшити свою прибутковість. Аналізатор подій може доповнити стандартний набір послуг букмекерської контори, роблячи її більш привабливою для клієнтів.

Таким чином, актуальність в створенні аналізатора подій спортивних матчів букмекерських контор, який буде зчитувати інформацію про матчі, турніри, видів спорту та порівнювати їх, надаючи користувачу важливу інформацію у зручному для нього форматі та в актуальний для нього час є беззаперечним.

### **Мета і завдання дослідження**



Мета і завдання роботи полягає у дослідженні теми букмекерської справи та її тісного відношення до сфери інформаційних технологій у сучасному світі. На основі цих даних необхідно створити додаток який буде спрямований на широкий ринок користувачів та буде висвітлювати проблематику та нормативний погляд на цю тему.

### **Об'єкт дослідження**

Об'єктом дослідження є сайти букмекерських контор та події які вони пропонують користувачам.

### **Предмет дослідження**

Предметом дослідження є визначення можливості безперешкодно зчитувати дані з сайтів букмекерських контор для подальшого їх аналізу та обробки.

### **Методи дослідження**

Теоретичні — обробка літературних джерел, синтез дослідження та аналіз досліджуваного матеріалу.

### **Практичне значення одержаних результатів**

Практичне значення одержаних результатів дослідження полягає у тому, що розробка одного застосунку замість довгого пошуку інформації самотужки, підвищує швидкість та ефективність праці як букмекерської контори так і особистого власника додатку. Результати дослідження можуть бути використані для пришвидшення та полегшення праці співробітників букмекерської контори, або дати можливість окремим користувачам швидко та легко отримувати додаткову інформацію без необхідності в додатковому пошуку.

### **Апробація результатів**

Результати роботи було представлено на XVI університетській науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених «Молода наука-2023» [16].

## Глосарій

*Букмекерська контора* — це компанія, яка приймає ставки від користувачів на події в світі спорту або суспільного життя. Організація дає свою оцінку ймовірності різних результатів, після чого ставить свій коефіцієнт. Після виграшу ставки її сума множиться на коефіцієнт, це і буде виграш гравця.

*Лудоманія (гральна залежність, ігроманія)* — розлад поведінки, що полягає в патологічній пристрасі до азартних ігор.

*Лінія спортивних подій* — набір спортивних подій, які проаналізували експерти букмекерської контори.

*Парсер* — програма, що забезпечує парсинг, тобто синтаксичний аналіз контенту в мережі по певній математичній моделі, створеній на одній з мов програмування.

*Телеграм бот* — це програма, яка функціонує в месенджері Telegram і виконує автоматизовані завдання.

*Django* — високорівневий відкритий Python-фреймворк (програмний каркас) для розробки вебсистем.

*BotFather* — це офіційний телеграм-бот, який використовується для створення та керування іншими телеграм-ботами в Telegram. Він є основним інтерфейсом для реєстрації та налаштування ботів у Telegram Bot API.

*Вебскрапінг* — перетворення у структуровані дані інформації з вебсторінок, які призначені для перегляду людиною за допомогою браузера.

## РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ АНАЛІЗУ ЛІНІЇ ПОДІЙ БУКМЕКЕРСЬКИХ КОНТОР

### 1.1 Огляд проблеми аналізу та збору інформації з букмекерських контор

Аналіз даних букмекерських контор є складною та багатогранною задачею, яка зустрічається в галузі спортивних ставок. Букмекерські контори пропонують широкий спектр спортивних подій та ринків ставок, але обробка та аналіз великого обсягу даних можуть становити виклик для індивідів та організацій.

Одна з основних проблем аналізу даних букмекерських контор полягає в недоступності та обмеженості доступу до повних даних. Букмекерські контори збирають значну кількість даних про спортивні події, такі як статистика гравців, командні результати, історія протистоянь та багато іншого. Однак, вони не завжди надають повний доступ до цих даних, особливо непублічним користувачам. Це може обмежувати можливості аналізу та впливати на якість прогнозів.

Ще одна проблема аналізування даних букмекерських контор пов'язана зі складністю обробки та інтерпретації даних. Великий обсяг і різноманітність даних вимагають від аналітиків та дослідників високої кваліфікації та експертизи. Важливо вирішити питання збору, очищення, перетворення та візуалізації даних, а також застосувати відповідні методи аналізу для отримання значущих результатів.

Також, аналіз даних букмекерських контор може стикатися з проблемою недостатньої якості та достовірності даних. Оскільки багато спортивних подій піддаються факторам випадковості та непередбачуваності, результати можуть бути неточними або спотвореними. Також існує ризик виникнення помилок при зборі даних або побічного впливу зовнішніх факторів на результати. Це може призвести до недостовірних прогнозів та помилкових висновків.

Крім того, проблема аналізування даних букмекерських контор включає в себе і ризики пов'язані з правовими та етичними аспектами. У деяких країнах спортивні ставки та аналіз даних букмекерських контор можуть бути обмежені або незаконні. Це може вплинути на доступність даних та відкритість досліджень. Крім того, важливо етично використовувати зібрані дані та забезпечити конфіденційність особистої інформації користувачів.

Загалом, проблема аналізування даних букмекерських контор включає в себе недоступність повних даних, складність обробки та інтерпретації, недостовірність даних та правові/етичні ризики. Для успішного аналізу необхідно вирішити ці проблеми та розробити відповідні стратегії та методики для отримання точних та цінних результатів.

## **1.2 Огляд проблематики зчитування даних з веб-джерел, з точки зору права та законодавства**

Згідно з законодавством України крадіжка даних та використання їх для власного збагачення є незаконним. На особу яка скоює дане правопорушення може чекати одна з перелічених проблем:

1. **Порушення авторських прав:** Дані, які збираються та обробляються букмекерською конторою, можуть бути власністю цієї контори. Крадіжка цих даних може вважатися порушенням авторських прав, оскільки несанкціоноване отримання та використання цих даних є незаконним.

2. **Порушення закону про конфіденційність:** Букмекерські контори збирають особисту інформацію своїх користувачів, таку як особисті дані, фінансові відомості тощо. Крадіжка цих конфіденційних даних може супроводжуватися порушенням законодавства про конфіденційність та може призвести до серйозних правових наслідків.

3. **Порушення закону про кібербезпеку:** Крадіжка даних з букмекерської контори може вважатися злочином в галузі кібербезпеки. Багато країн мають закони, що криміналізують незаконне отримання доступу

до комп'ютерної системи або злочини, пов'язані з кібератаками. Такі порушення можуть призвести до суворих штрафів та навіть покарання у вигляді ув'язнення.

4. Грабіж, шахрайство та пошкодження майна: Крадіжка даних може також спричинити втрату майна або нанесення шкоди букмекерській конторі. Якщо відбувається несанкціонований доступ до їх систем або порушення їх баз даних, це може вважатися злочином, який може призвести до правових наслідків, таких як цивільна справа або кримінальне переслідування.

5. Порушення угод та договорів: Часто букмекерські контори мають угоди та договори з різними сторонами, включаючи спортивні організації, постачальників послуг та інші компанії. Крадіжка даних може порушити умови цих угод та договорів і призвести до юридичних проблем, включаючи можливі позови та відшкодування збитків.

### **1.3 Огляд літературних джерел**

Питанню теми збору та аналізу даних присвячено чимало робіт та книжок, вони ілюструють проблематику теми, обхідні шляхи для простого вирішення питання та підводні камені з якими може зіткнутись виконавець або звичайний користувач який зацікавився у цій темі [1-4]. Наприклад автором Томасом А. Рунклером в книзі “Data Analytics: Models and Algorithms for Intelligent Data Analysis” [1] описані методології збору інформації з джерел методом застосування та включення спеціально створеного та навченого штучного інтелекту, який при мінімальному втручанні виконавця зможе самостійно пристосовуватись до структури ресурсу та зчитувати всі дані з нього. Загалом на сьогодні існує чимало технологій та методик для зчитування даних як з веб застосунків так і з інших джерел. Вони можуть застосовуватись для збору даних як для бізнес потреб, маркетингових справ, особистих потреб та навіть для потреб держави та влади. Наприклад в своїй книзі D.Grune описав створення аналіз проблематики, переваги та недоліки

при створенні програми аналізатора інформації з веб-ресурсів для бізнес аналітики [4].

Також в своїх книжках та роботах автори спрямовують увагу на обрані типу спрямування пошуку, основної цілі пошуку для відсіювання непотрібних даних при подальшому аналізі, обрані як методу аналізу даних так і методів для реалізації як пошуку, так і для розробки програми реалізатора в подальшому. В своїй роботі Зігмунд Брандт “Data Analysis Statistical and Computational Methods for Scientists and Engineers” [2] зібрав усю інформацію з прикладами створення алгоритмів для аналізу даних. Також в книзі розкрита тема важливості аналізу даних для науковців в різних напрямках досліджень та їх спрямувань. В пошуці рішень для створення програм аналізаторів слід звернути увагу на технології для створення. Так як кожна мова та додаткові модулі до неї має свої переваги та недоліки. Слід зазначити що обирати варто технології з високим рівнем оновлюваності та великою спільнотою розробників. Опираючись на записи з книги “Python for data analysis” Веса Мак Кіннея [3] більшість розробників обирає саме мову Python для розробки та створення програм аналізаторів, через її пристосованість до збору інформації та не тривалого періоду навчання до стабільного оволодіння. Продовжуючи тему обрання технологій створення фахівці рекомендують використовувати фреймворки так як вони спростять та пришвидшать процес, та якість кінцевого продукту. При обрані фреймворку слід опиратися на метод пошуку та бібліотеки які для цього пропонує шуканий фреймворк.

#### **1.4 Види технологій для зчитування та парсингу даних з веб сторінок**

На сьогоднішній день існує безліч методів для парсингу даних з веб сторінок або додатків. Але усі вони поділяються на такі види:

1. Веб-скрапінг (Web scraping): ця технологія використовується для автоматичного зчитування даних з веб-сторінок. Вона базується на програмному забезпеченні, яке автоматично аналізує HTML-код сторінки, виділяє необхідну інформацію та збирає її в структурованому форматі.

Інструменти для веб-скрапінгу, такі як BeautifulSoup та Scrapy, дозволяють легко отримувати дані з різних джерел, включаючи веб-сайти букмекерських контор.

2. API (Application Programming Interface): API є інтерфейсом, який дозволяє програмам взаємодіяти з іншими програмними додатками або сервісами. Багато букмекерських контор надають API, які дозволяють розробникам отримувати доступ до даних про спортивні події, коефіцієнтів та інших ринків ставок. Використання API дозволяє отримувати актуальні та структуровані дані без необхідності скрапінгу веб-сторінок.

3. XML та JSON парсери: XML (eXtensible Markup Language) та JSON (JavaScript Object Notation) є популярними форматами для обміну даними. XML парсери, такі як lxml і xml.etree.ElementTree в Python, дозволяють розбирати XML-документи та отримувати доступ до їх елементів та атрибутів. Аналогічно, JSON парсери, такі як json в Python, дозволяють розбирати JSON-структури та отримувати доступ до їх ключів та значень.

4. OCR (Optical Character Recognition): OCR-технологія використовується для розпізнавання тексту зображень або сканованих документів. Вона може бути корисною, коли потрібно зчитати дані з неструктурованих джерел, таких як фотографії або скріншоти. Існують багато бібліотек та сервісів OCR, наприклад, Tesseract, Google Cloud Vision API або Microsoft Azure Cognitive Services.

5. Natural Language Processing (NLP): NLP-технології використовуються для обробки та аналізу текстової інформації. Вони можуть застосовуватись для витягування ключових слів, класифікації тексту, виявлення іменованих сутностей та багатьох інших задач. При роботі з даними з букмекерських контор, NLP може допомогти в розумінні та аналізі статистики, коментарів користувачів або новинних повідомлень.

Як можна побачити усі види мають свої особливості з перевагами та недоліками. Для виконання завдання проекту були використані методи Веб-скрапінгу для того щоб отримувати текст сторінок з усіма ключовими

елементами, JSON парсинг для того щоб зчитувати дані які приховані від звичайного скрапінгу під тегами HTML або Java-Script кодом. Метод NLP використовувався для пошуку ключових слів з які необхідно було зачепитися та від яких відштовхуватись при редагуванні вихідного потоку даних та їх коректного відображення.

### **1.5 Сфери використання аналізатора лінії подій букмекерських контор**

Аналізатор лінії подій може допомогти в аналізі спортивних подій і прогнозуванні результатів. Він може аналізувати різні фактори, такі як статистика команд, форма гравців, умови гри та інші фактори, що впливають на результати спортивних подій. Це може допомогти гравцям зробити кращі рішення при розміщенні ставок.

Він може допомогти аналізувати зміни в цінах акцій, валютних курсах, сировинних товарах та інших фінансових інструментах. Це може допомогти виявити тренди, встановити точки входу та виходу з позицій та зробити кращі торговельні рішення. Аналізатор лінії подій може бути використаний для оцінки ризиків в різних галузях. Наприклад, в страхуванні він може допомогти визначити ймовірність настання подій, що призводять до страхових виплат.

В бізнесі він може допомогти аналізувати ризики, пов'язані зі змінами в ринковому середовищі або конкурентними факторами. Аналізатор може допомогти ідентифікувати та оцінювати ризики, що дозволяє приймати обґрунтовані рішення з мінімізації ризиків.

Він може допомогти зрозуміти попит на певні товари або послуги, ідентифікувати тенденції та прогнозувати зміни в споживчому попиті. Це може допомогти компаніям впроваджувати ефективні маркетингові стратегії та покращувати свої продукти або послуги.

Аналізатор лінії подій може бути корисним в гральній індустрії, такій як покер, казино або інші азартні ігри. Він може аналізувати імовірності виграшу,



статистику гравців, стратегії гри та інші фактори, що впливають на шанси на успіх. Це може допомогти гравцям приймати розсудливі рішення та покращувати їх стратегії гри.

## 1.6 Приклади застосунків аналізаторів

На маркетплейсах та в загальному доступі є багато аналогів аналізаторів подій, які виконують схожий за стилем та поведженням функціонал.

### *Betfair API*

Betfair є однією з найбільших букмекерських контор і надає API для отримання доступу до своїх даних. Це дозволяє розробникам отримувати дані про спортивні події, коефіцієнти, ринки ставок та іншу інформацію. Betfair API дозволяє зробити автоматичні запити та отримати дані в реальному часі.

### *OddsPortal*

OddsPortal є відомим сайтом, що надає дані про спортивні коефіцієнти та лінії подій. Вони також мають платформу з API, яке дозволяє отримувати доступ до їхніх даних. За допомогою OddsPortal API можна отримати інформацію про коефіцієнти, результати матчів, статистику та інші дані.

### *Sportradar*

Sportradar є провідною компанією у сфері спортивних даних і надає послуги парсингу даних для букмекерських контор. Вони мають широкий спектр API, які дозволяють отримувати дані про різні види спорту, включаючи футбол, баскетбол, теніс та інші. Sportradar API надає доступ до детальної інформації про матчі, статистику, коефіцієнти та багато іншого.

### *Bet365*

Є однією з найпопулярніших букмекерських контор, і вони надають API для розробників. Це API дозволяє отримати дані про спортивні події, коефіцієнти, результати та іншу інформацію. За допомогою Bet365 API можна отримати дані в реальному часі та виконувати різні запити для отримання необхідної інформації.

З цього переліку хотілося б виділити сервіс Oddsportal, так як для виконання кваліфікаційного завдання саме цей сервіс був обраний як ресурс для отримання даних, він має широкий список як турнірів так і матчів всередині них, при цьому не потребує реєстрації для отримання даних, і виконавець кваліфікаційної роботи не може бути притягнутий до відповідальності так як сервіс сам збирає дані з букмекерських контор.

### **1.7 Аналіз існуючих програмних систем парсингу**

#### *Oddsportal*

“Oddsportal” — це онлайн-сервіс, який спеціалізується на зібранні та аналізі статистичної інформації про спортивні змагання з різних країн світу. Сервіс пропонує інформацію про широкий спектр видів спорту, включаючи футбол, баскетбол, хокей, теніс, бейсбол, гандбол, водне поло та багато іншого.

“Oddsportal” надає користувачам можливість порівнювати коефіцієнти на різних букмекерських сайтах, щоб знайти найкращі умови для ставок. Також сервіс пропонує результати матчів та таблиці турнірів, що дозволяє користувачам відстежувати стан своїх улюблених команд та гравців (див. Рис. 1).

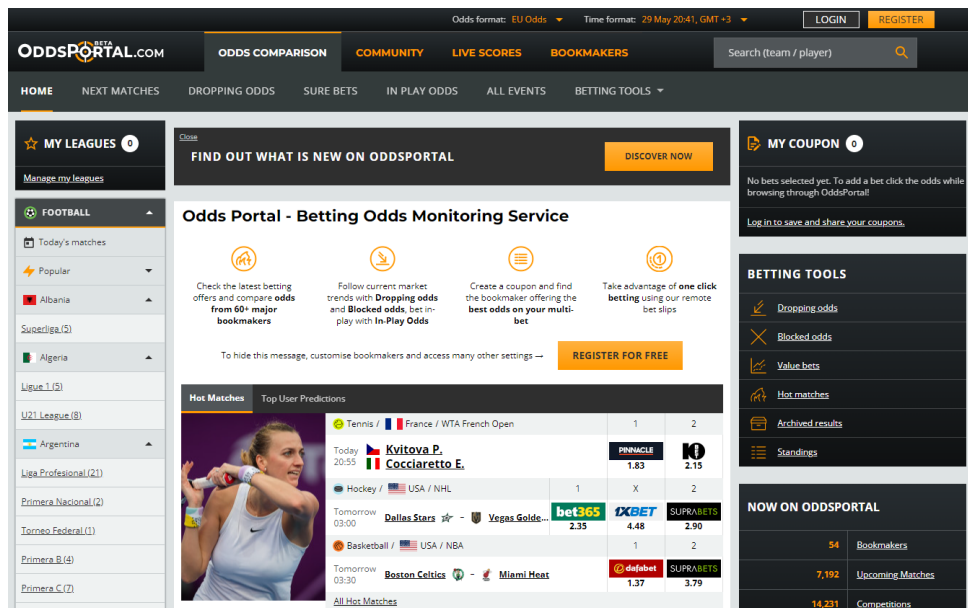


Рис. 1 Зображення головного меню сервісу Oddsportal

Сервіс також пропонує аналізи та прогнози на майбутні матчі від експертів та професійних каперів. Крім того, сервіс надає доступ до інформації про погодні умови на стадіонах та інші фактори, які можуть вплинути на результат матчу.

Для користування “Oddsportal” не потрібна реєстрація. Сервіс є безкоштовним та доступним для всіх користувачів. Однак, для користування деякими функціями, такими як розширений аналіз та прогнозування, можлива платна підписка.

У загальному, “Oddsportal” є корисним інструментом для будь-якого любителя спорту, або букмекерської контори.

#### *Переваги сервісу “Oddsportal”:*

1. Широкий вибір видів спорту та турнірів з різних країн світу.
2. Доступність інформації про розклади матчів, результати гри.
3. Можливість порівняння коефіцієнтів на різних букмекерських сайтах.
4. Аналіз та прогнозування від експертів.
5. Інформація про фактори, які можуть вплинути на результат матчу.
6. Безкоштовний доступ до основних функцій сервісу.

#### *Недоліки сервісу “Oddsportal”:*

1. Не завжди можлива 100% точність інформації.

2. Розширений аналіз та прогнозування, можуть бути доступні тільки за платну підписку.
3. Можливість перенесення чи скасування матчів, що може вплинути на точність інформації та статистику.
4. Можливе перевантаження сайту в час великих змагань, що може призвести до затримок у завантаженні інформації.

### *Bet365*

“Bet365” — це один з найбільших букмекерських операторів в інтернеті зі штаб-квартирою у Великій Британії. “Bet365” пропонує різноманітні види ставок на спорт, політику, розваги та інші події (див. Рис. 2).

“Bet365” була заснована у 2000 році у Великій Британії Денізом Коутсом та її чоловіком, Пітером Коутсом. З тих пір компанія стала однією з найбільших букмекерських компаній в світі з більш ніж 45 мільйонами зареєстрованих користувачів у більш ніж 200 країнах світу. Компанія працює з ліцензіями від урядів різних країн, таких як Великобританія, Мальта, Гібралтар та інших.

Основні можливості та функціонал платформи “Bet365” можна описати наступними пунктами:

Живі трансляції подій: “Bet365” надає користувачам можливість дивитися прями трансляції спортивних подій, на які вони ставлять. Це дозволяє користувачам бути в курсі ситуації на полі та змінювати свої ставки в режимі реального часу.

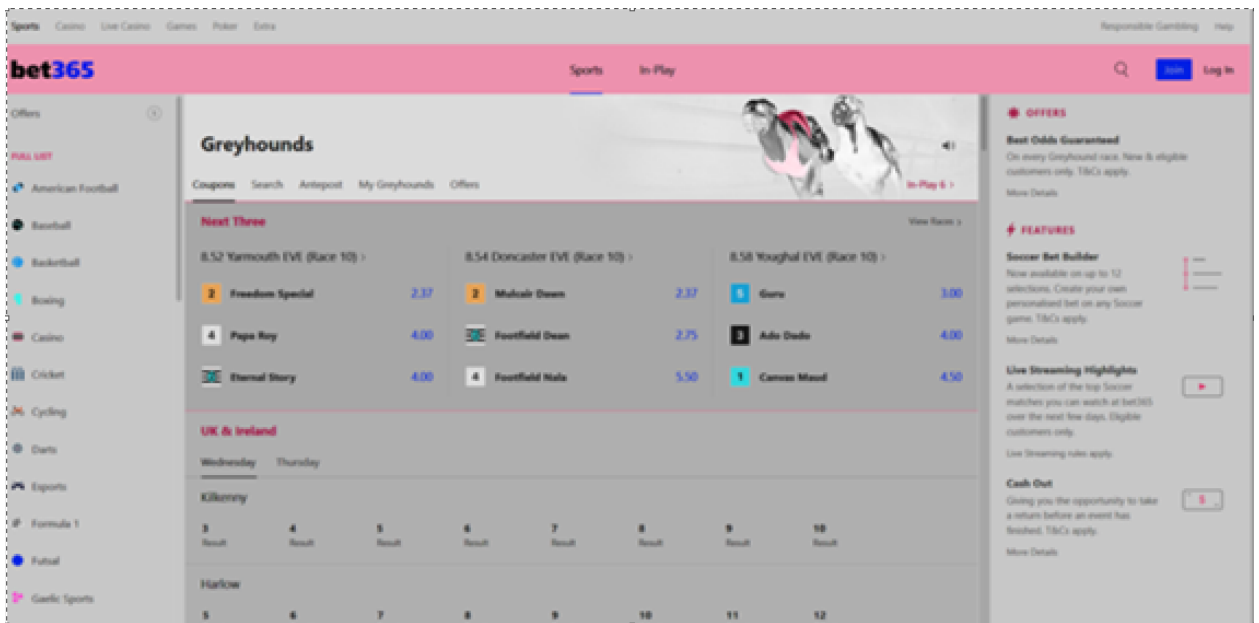


Рис. 2 Зображення головного меню платформи Bet365

Статистика та аналіз: “Bet365” надає користувачам різні інструменти для аналізу та статистики, які можуть допомогти їм зробити кращі ставки. Користувачі можуть переглядати історію результатів команд та гравців, статистику забитих голів, жовтих та червоних карток, кількість кутових та штрафних ударів та інше.

Аналіз ставок і ставкових ринків: “Bet365” надає користувачам можливість аналізувати ставки та ставкові ринки, що дозволяє їм отримувати додаткову інформацію про ринок та зробити кращі ставки.

Система сповіщень: “Bet365” пропонує користувачам систему сповіщень, яка дозволяє отримувати повідомлення про результати своїх ставок, а також про зміни в ставкових ринках та інші важливі події.

Моніторинг та контроль ставок: “Bet365” пропонує користувачам можливість моніторити та контролювати свої ставки, в тому числі змінювати або скасовувати ставки.

“Bet365” пропонує зручний та добре розроблений інтерфейс для користувачів, можливість слідкування не тільки за класичними видами спорту а й за кіберспортом.

*Переваги платформи “Bet365”:*

1. Широкий вибір спортивних подій: Bet365 пропонує ставки на більше ніж 40 різних видів спорту, що дозволяє користувачам вибирати з більшої кількості подій для ставок.
2. Легкий доступ: Bet365 пропонує зручний та легкий веб-інтерфейс, що дозволяє легко знаходити необхідну інформацію.
3. Привітні умови для нових користувачів: Bet365 пропонує різні бонусні пропозиції та акції для нових користувачів, що робить сервіс більш привабливим для нових клієнтів.

#### *Недоліки платформи “Bet365”:*

1. Відсутність деяких дружніх турнірів на класичні та кіберспортивні види спорту.
2. Часта невідповідність з часовим поясом користувача.
3. Заблокований у багатьох країнах.

#### *Betfair*

Betfair — це програмний інтерфейс (API), який надає розробникам доступ до різноманітних функцій та даних платформи Betfair, одного з найбільших і найпопулярніших букмекерських операторів у світі. Betfair API дозволяє створювати програми, які взаємодіють зі спортивними ставками, отримувати дані про ринки, розміщувати ставки та отримувати оновлення живих коефіцієнтів (див. Рис. 3).

Використання Betfair відкриває перед користувачем такі можливості:

Розміщення ставок: Betfair API дозволяє автоматично розміщувати ставки на ринках. Ви можете виконати ставку на певні події з певними коефіцієнтами, використовуючи API-запити.

Отримання оновлень живих коефіцієнтів: Ви можете підписатися на оновлення живих коефіцієнтів для вибраних ринків. Це дозволить отримувати регулярні оновлення коефіцієнтів для ставок у реальному часі.

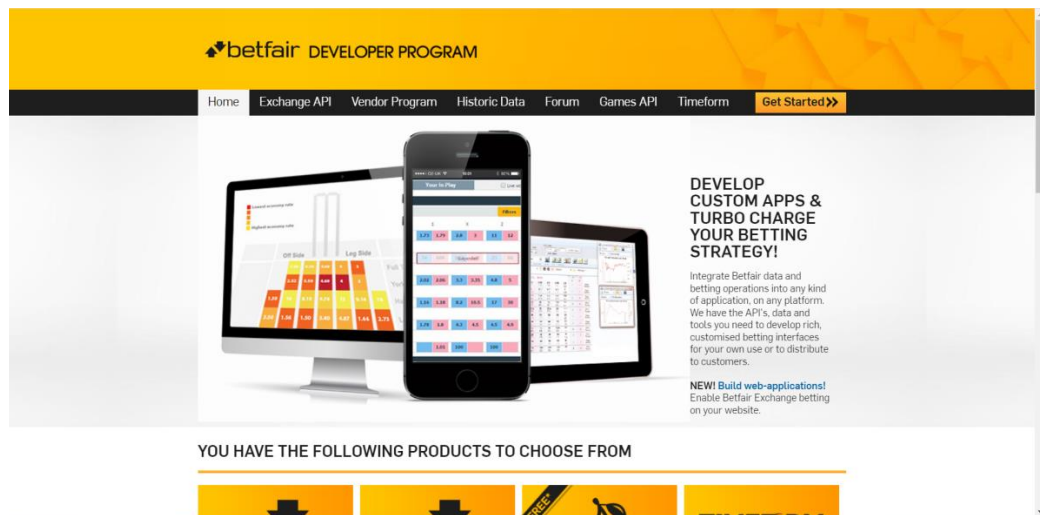


Рис. 3 Зображення головного меню сервісу Betfair

Управління замовленнями: Ви можете отримувати інформацію про відкриті та виконані замовлення, а також скасовувати відкриті замовлення за допомогою API.

Betfair API використовує REST-архітектуру, що означає, що взаємодія з API відбувається через HTTP-запити, такі як GET, POST та DELETE. Дані повертаються у форматі JSON, який легко обробляти іншими програмами.

Для доступу до Betfair API потрібно мати активний обліковий запис Betfair, а також отримати ключ розробника API. Цей ключ використовується для автентифікації запитів до API і забезпечення безпеки.

#### *Переваги сервісу Betfair:*

1. Автоматизовані стратегії.
2. Доступ до ринкових даних.
3. Висока швидкість та ефективність.
4. Гнучкість та налаштування.

#### *Недоліки сервісу Betfair:*

1. Складність використання.
2. Вимоги до інтеграції.
3. Обмеження доступу.
4. Зміна API.

5. Ризик виконання ставок.

6. Вартість.

### *Sportradar*

Sportradar це компанія яка спеціалізується в галузі спортивних матчів та аналітики. Також компанія пропонує для своїх користувачів можливості в зборі, обробці, аналітиці спортивних даних та їх автоматичного моніторингу (див. Рис. 4).

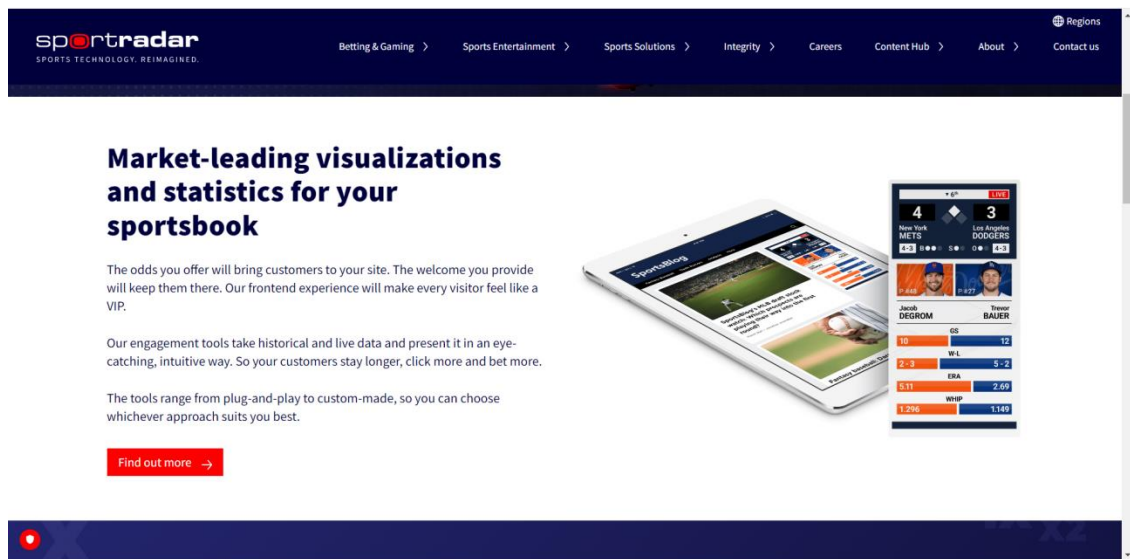


Рис. 4 Зображення головного меню сервісу Sportradar

Основні можливості компанії:

Sportradar пропонує одну з найширших покриттів спортивних дисциплін у галузі. Вони співпрацюють з численними професійними спортивними лігами та організаціями, що дозволяє їм збирати дані про різні види спорту в реальному часі.

Компанія покладає особливий наголос на якість та точність своїх спортивних даних. Вони використовують високотехнологічні системи збору та обробки даних, а також мають широку мережу спортивних експертів для перевірки та підтвердження інформації.



Sportradar використовує штучний інтелект, машинне навчання та аналітику даних, щоб забезпечити розуміння спортивних подій і надати клієнтам цінну інформацію.

Компанія володіє розширеними системами моніторингу, які виявляють незвичайні та підозрілі активності в спортивних подіях, що можуть бути пов'язані з шахрайством.

*Переваги компанії Sportradar:*

1. Широке охоплення спорту.
2. Висока якість та точність даних.
3. Технологічні інновації.
4. Протидія шахрайству.
5. Різноманітність послуг.

*Недоліки компанії Sportradar:*

1. Висока вартість.
2. Обмежена доступність даних.
3. Залежність від сторонніх джерел.
4. Відсутність доступу до ряду деяких спортивних подій.

## 2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

### 2.1 Аналіз сучасних фреймворків для створення веб застосунків

На сьогоднішній день майже жоден веб застосунок або веб сервіс не обходиться без використання фреймворку. Фреймворк — це набір загальних інструментів, бібліотек і правил, які допомагають розробникам швидше і ефективніше створювати програмні продукти. В основному, фреймворк надає структуру та архітектуру для розробки додатків, спрощуючи процес програмування. Своїм початком фреймворки забов'язані компанії “Microsoft” яка у 80-х роках створила набір інструментів “Smalltalk”. Він забезпечував механізми для створення графічних інтерфейсів користувача. 1990-ті роки були переломними для розвитку фреймворків. З'явилися нові мови програмування та інструменти, такі як Java, Python і Ruby, які надали потужність та гнучкість для створення фреймворків. У цей час стали популярними фреймворки, такі як Java EE (Enterprise Edition), .NET Framework та Ruby on Rails. У період двохтисячних, фреймворки стали ширше використовуватися в розробці веб-додатків. З'явилися фреймворки, такі як Django (Python), Ruby on Rails і Laravel (PHP), які спрощували розробку веб-додатків шляхом надання готової структури та інструментів для роботи з базами даних, аутентифікації, маршрутизації тощо. Згодом з'являються фреймворки для розробки мобільних додатків, штучного інтелекту, аналітики даних та інших сфер. Також виникають фреймворки на базі JavaScript, такі як Angular, React та Vue.js, які стають популярними для розробки фронтенду. На сьогоднішній день фреймворки стали необхідним інструментом для багатьох розробників. Вони пропонують широкий спектр функцій, розширюються і покращуються за допомогою спільноти розробників та забезпечують швидку розробку програмного забезпечення.

Основною метою фреймворку є забезпечення готового набору інструментів які значно спростять роботу розробників.

Основні функції фреймворку:

*Прискорення розробки.* Фреймворки надають готові компоненти, шаблони та інструменти, які допомагають розробникам прискорити процес розробки програмного продукту. Вони забезпечують заздалегідь визначені правила та структуру, що полегшує написання коду.

*Забезпечення структури.* Фреймворками надаються готові архітектурні шаблони та структуру для розробки програмних продуктів. Це дозволяє розробникам організувати свій код у логічну структуру та забезпечити його розширюваність та обслуговування у майбутньому.

*Використання готових компонентів.* В середині фреймворки зазвичай мають багато готових компонентів, які можна використовувати для реалізації різних функціональностей. Це дозволяє розробникам економити час та зусилля, використовуючи перевірені рішення замість написання коду з нуля.

*Спрощення підтримки та розширення.* Надання інструментарію для легкої підтримки та розширення програмного продукту. Завдяки цьому розробники можуть швидко вносити зміни, додавати нові функції та розширювати функціональність без значного переписування коду.

*Стандартизація.* Використання фреймворка допомагає розробникам використовувати спільні стандарти та практики. Це сприяє зрозумінню коду і полегшує співпрацю в команді розробників.

За видами додатку для якого застосовується фреймворк можна виділити такі типи:

*Веб-розробка:* Ruby on Rails, Django, ASP.NET, Laravel. Багато фреймворків для веб-розробки використовують модель MVC або подібні архітектурні підходи. Це дозволяє розділити додаток на компоненти: модель (дані), представлення (вигляд) та контролер (логіка). Це сприяє організації коду і полегшує розуміння, підтримку та розширення додатку. Цей тип фреймворків зазвичай мають механізм маршрутизації, який визначає, які URL-адреси пов'язані з якими функціями або контролерами. Це дозволяє легко визначати поведінку додатку на основі URL-адреси та обробляти запити від

користувачів. Часто ці фреймворки надають інструменти для роботи з базами даних. Вони допомагають спростити створення, зчитування, оновлення та видалення даних з бази даних, а також забезпечують безпеку й управління підключенням до бази даних. Також передбачають систему шаблонів, яка дозволяє вам створювати вигляди вашого веб-додатку. Вони надають готові шаблони та можливість створювати власні шаблони, що дозволяє швидко та легко створювати сторінки з необхідним виглядом. Також бажано виділити інструменти та практики для захисту веб-додатку від загроз безпеки, таких як атаки хакерів, введення шкідливого коду тощо.

*Мобільна розробка:* React Native, Flutter, Xamarin. Одні з основних особливостей фреймворків для мобільної розробки полягають у можливості створювати кросплатформові додатки. Це дає можливість розробляти додатки один раз і запускати їх на різних платформах, таких як iOS та Android. Це забезпечує значну економію часу та зусиль, оскільки не потрібно писати окремий код для кожної платформи. Не менш важливою особливістю даного типу фреймворків є реактивний інтерфейс. Це означає, що зміни в стані додатку автоматично відображаються на його інтерфейсі без необхідності вручну оновлювати його. Це дозволяє створювати більш динамічні та взаємодіючі додатки. Фреймворки для мобільної розробки зазвичай надають доступ до нативних функцій та API платформи. За рахунок цього відкриваються специфічні можливості пристрою, такі як камера, геолокація, сповіщення тощо, безпосередньо з вашого додатка.

*Ігрова розробка:* Unity, Unreal Engine. Фреймворки для ігрової розробки надають потужні інструменти для графічного рендерингу, що дозволяє створювати вражаючі візуальні ефекти, реалістичну графіку та спеціальні ефекти в іграх. Вони підтримують шейдери, освітлення, текстурки та інші графічні можливості. Важливим фактором в графічному рендерингу є фізичне моделювання поведінки персонажів в іграх. Воно дозволяє створювати реалістичну фізику, симулювати колізії, рухи та інші фізичні взаємодії між об'єктами в грі. Фреймворки для ігрової розробки надають можливості для

інтеграції звукових ефектів та музики в ігри. Вони підтримують різні формати аудіо, можливість управління звуком, створення 3D-звукових ефектів та інші аудіофункції. Ще однією цікавою особливістю цього типу фреймворків є анімація та керування персонажами. Вони надають можливість створювати складні рухи, анімаційні переходи, поведінку персонажів та інші анімаційні ефекти. Також для зручності виконавця та самого користувача в таких фреймворках передбачений функціонал управління введення. Він дозволяє обробляти різні типи введення, реагувати на жести та комбінації клавіш, імітувати рухи геймпада та інші функції управління.

*Штучний інтелект та аналітика даних:* TensorFlow, PyTorch. Фреймворки для штучного інтелекту та аналітики даних надають багато вбудованих бібліотек та інструментів для машинного навчання. Ці бібліотеки містять реалізації різних алгоритмів машинного навчання, таких як навчання з учителем, не навчання з учителем, кластеризація, рекомендації та інші. Фреймворки для AI та аналітики даних зазвичай підтримують розподілене обчислення та можливість роботи з великими обсягами даних. Вони дозволяють розподіляти обчислювальні завдання на кластері комп'ютерів або використовувати графічні процесори (GPU) для прискорення обчислень. Також цей тип фреймворку надає інструменти для візуалізації та інтерактивного аналізу даних. Це дозволяє відображати графіки, діаграми, хмари слів та інші візуальні елементи для більш зрозумілого представлення даних та виявлення закономірностей. Останньою але не менш важливою особливістю даного типу фреймворків є оптимізація та розширення. Це інструменти для оптимізації алгоритмів, покращення продуктивності та розширення функціональності. В собі можуть містити оптимізовані реалізації алгоритмів, можливості паралельного обчислення, підтримку розподіленої обробки даних та інші інструменти для ефективної роботи з великими наборами даних.

### 2.1.1 Огляд фреймворків для створення веб застосунків

#### *Django*

Django — це високорівневий веб-фреймворк, написаний на мові програмування Python, який дозволяє розробляти швидкі та масштабовані веб-додатки [5]. Фреймворк Django був створений для полегшення процесу розробки веб-додатків шляхом надання готових компонентів і структури проекту. Історія створення Django починається в 2003 році, коли журналістів з видання Lawrence Journal-World потребували новий інструмент для швидкого створення веб-додатків. В результаті, вони розробили прототип фреймворку, який був названий "Django" на честь джазового гітариста Джанго Рейнхардта. У 2005 році була випущена перша публічна версія Django, і з того часу фреймворк отримав значну популярність та визнання веб-розробників по всьому світу. Протягом років Django пройшов значну кількість оновлень та випусків нових версій. Команда розробників Django активно працює над вдосконаленням фреймворка, виправленням помилок та впровадженням нових функцій. Кожна нова версія Django пропонує поліпшення продуктивності, безпеки та зручності в розробці веб-додатків.

Основні можливості Django включають:

*Модель-представлення-контролер* (Model-View-Controller, MVC) архітектура: Django використовує MVC-підхід, що дозволяє розділити логіку додатку на компоненти для покращення модульності та повторного використання коду.

*Об'єктно-реляційне відображення* (Object-Relational Mapping, ORM): Django надає потужний ORM, який дозволяє взаємодіяти з базами даних з використанням об'єктно-орієнтованого підходу, замість написання складних SQL-запитів.

*Адміністративний інтерфейс*: Django має вбудований адміністративний інтерфейс, який автоматично генерує адміністративну панель для веб-додатків, що базуються на моделях Django. Це значно спрощує створення та управління адміністративними функціями.

*Роутинг та обробка URL-запитів:* Django має потужний механізм роутингу, який дозволяє легко визначати URL-шаблони та відповідні до них функції-обробники.

*Шаблонний двигун:* Django постачається з шаблонним двигуном, що дозволяє розробникам відокремлювати логіку від представлення та створювати багаторівневі шаблони для веб-сторінок.

*Переваги Django:*

1. **Продуктивність:** Django пропонує готові компоненти та структуру проекту, що значно прискорює розробку веб-додатків.
2. **Стабільність та масштабованість:** Django має велику спільноту розробників, що забезпечує підтримку та безпеку фреймворка. Він також масштабується добре, що дозволяє обробляти великі навантаження.
3. **Безпека:** Django має вбудовані заходи безпеки, такі як захист від хакерських атак, захист від внесення коду та обробка введення користувача.
4. **Розширюваність:** Django дозволяє легко розширювати функціональність за допомогою сторонніх пакетів та розширень.

*Недоліки Django:*

1. **Великий об'єм коду:** через свою потужність та функціональність, Django може мати значний об'єм коду, що може бути складним для розуміння для початківців.
2. **Відносно висока складність:** для повного розуміння та використання Django може знадобитися певний час та зусилля.
3. **Обмежена гнучкість:** через свою структуру та конвенції, Django може бути менш гнучким в порівнянні з іншими фреймворками, що призводить до деяких обмежень у виборі архітектурних рішень.

*ASP.NET*

ASP.NET — це веб-фреймворк, розроблений компанією Microsoft для побудови сучасних веб-додатків та служб. ASP.NET базується на платформі

.NET і надає розробникам потужні інструменти для швидкої та ефективної розробки веб-додатків [6]. Історія створення ASP.NET починається з появи платформи .NET в 2002 році. Вона була розроблена компанією Microsoft для побудови різноманітних програмних продуктів. ASP.NET був випущений разом з першою версією .NET Framework і став ключовим компонентом для веб-розробки. З того часу ASP.NET пройшов кілька оновлень та розширень, що покращили його продуктивність, функціональність та забезпечили сумісність з останніми технологіями веб-розробки.

Основні можливості ASP.NET:

*Мови програмування:* ASP.NET підтримує багато мов програмування, включаючи C#, Visual Basic.NET і F#. Це дає розробникам можливість використовувати ту мову, з якою вони найбільш знайомі.

*Модель програмування:* ASP.NET пропонує модель програмування на основі подій (event-driven programming), де код виконується у відповідь на події, такі як клік миші або натискання кнопок. Це дозволяє розробникам створювати динамічні веб-додатки.

*Розширюваність:* ASP.NET підтримує модульну архітектуру, що дозволяє розробникам додавати та використовувати розширення, компоненти та сторонні бібліотеки для покращення функціональності своїх додатків.

*Велика спільнота розробників:* Через підтримку Microsoft та популярність платформи .NET, ASP.NET має велику спільноту розробників, яка надає підтримку, документацію, плагіни та інші корисні ресурси для розробки веб-додатків.

*Переваги ASP.NET:*

1. Висока продуктивність: ASP.NET використовує компіляцію коду на етапі розгортання, що дозволяє досягти високої швидкодії веб-додатків.
2. Масштабованість: ASP.NET має вбудовану підтримку для масштабування веб-додатків, що дозволяє розробникам легко розширювати свої додатки, щоб вони могли обробляти великі навантаження та використовувати кластеризацію.



3. Безпека: ASP.NET надає вбудовані механізми безпеки, такі як автентифікація та авторизація, захист від атак на міжсайтове виконання скриптів (XSS) та інші загрози безпеки.
4. *Недоліки ASP.NET:*
5. Вимоги до серверної інфраструктури: Розгортання та налаштування ASP.NET додатків можуть бути складними через вимоги до серверної інфраструктури, такі як встановлення та налаштування Internet Information Services (IIS).
6. Вартість: Деякі інструменти та розширення для ASP.NET можуть бути платними, що може вплинути на вартість розробки та підтримки веб-додатків.

#### *Laravel*

Laravel — це веб-фреймворк з відкритим вихідним кодом, розроблений Тейлором Отвеллом (Taylor Otwell). Він був випущений в 2011 році і швидко став одним з найпопулярніших фреймворків для розробки веб-додатків на мові програмування PHP. Laravel має чистий та елегантний синтаксис, а також надає розробникам потужні інструменти для швидкого створення як простих, так і складних веб-додатків [7]. Історія створення Laravel починається з особистого проекту Тейлора Отвелла, який пізніше перетворився на фреймворк. Отвелл хотів створити фреймворк, який би забезпечував простоту використання та прискорював процес розробки веб-додатків. В результаті він створив Laravel, заснований на найкращих практиках з інших фреймворків та власних ідеях. Оновлення Laravel відбуваються регулярно. Команда розробників Laravel прагне покращувати фреймворк, додавати нові функції та виправляти помилки. Кожне оновлення має свою версію, наприклад, Laravel 5, Laravel 6, Laravel 7 тощо. Кожна нова версія зазвичай пропонує покращену продуктивність, нові функції та зміни в структурі фреймворку.

Основні можливості Laravel:

*Маршрутизація та обробка запитів:* Laravel має потужну систему маршрутизації, яка дозволяє визначати URL-шаблони та відповідні контролери для обробки запитів.

*ORM (Object-Relational Mapping):* Laravel використовує Eloquent ORM, що дозволяє розробникам легко працювати з базами даних, використовуючи об'єктно-орієнтований підхід.

*Шаблонізація та система макетів:* Laravel має вбудовану систему шаблонів Blade, яка дозволяє розробникам створювати чистий і зрозумілий HTML-код з динамічними елементами.

*Аутентифікація та авторизація:* Laravel надає гнучкі інструменти для реалізації аутентифікації користувачів та контролю доступу до різних частин веб-додатку.

*Переваги Laravel над іншими фреймворками:*

1. Простота використання: Laravel має зрозумілий та елегантний синтаксис, що робить його легким для вивчення та розуміння.
2. Розширюваність: Laravel підтримує велику кількість сторонніх пакетів та бібліотек, які дозволяють розширювати функціональність фреймворку безпосередньо.
3. Зручна документація: Laravel має детальну та зрозумілу документацію, що полегшує розробку та вирішення проблем.
4. Активна спільнота: Laravel має велику та активну спільноту розробників, яка надає підтримку, допомогу та створює різноманітні розширення для фреймворку.

*Недоліки Laravel:*

1. Швидкість: Деякі розробники вказують на те, що Laravel може мати трохи меншу швидкодію порівняно з іншими фреймворками.
2. Складність при масштабуванні: За ростом веб-додатка та збільшенням обсягу коду, масштабування Laravel може стати викликом.

## 2.2 Аналіз сучасних технологій для створення телеграм ботів

Існує декілька технологій для створення телеграм боту, будь якої складності та масштабності. Кожна з технологій має свої переваги та недоліки, але у всіх них є спільна риса, вони усі функціонують с платформою та месенджером Telegram. Telegram — месенджер, що надає наскрізні зашифровані чати, відеодзвінки, VoIP та можливість обміну файлами.

### *Telegram Bot API*

Telegram Bot API — це інтерфейс програмування застосунків (API), наданий компанією Telegram, який дозволяє розробникам створювати та взаємодіяти з телеграм-ботами. Цей API надає доступ до різноманітних функцій та можливостей, необхідних для розробки інтерактивних ботів в Telegram [8]. Історія створення Telegram Bot API пов'язана зі створенням месенджера Telegram компанією Telegram Messenger LLP. Telegram був запущений в 2013 році Павлом Дуровим та його братом Ніколаєм Дуровим. У 2015 році Telegram оголосив про введення підтримки ботів через Telegram Bot API. З тих пір Telegram Bot API став ключовим інструментом для розробки телеграм-ботів у різних сферах, від комунікації та розваг до бізнесу та послуг.

### Основні можливості Telegram Bot API:

*Отримання та обробка повідомлень:* Telegram Bot API дозволяє ботам отримувати та обробляти текстові повідомлення, зображення, відео, аудіо, документи та інші медіафайли, що дозволяє розробникам створювати багатофункціональні боти.

*Взаємодія з клавіатурами:* API надає можливість відправляти клавіатури разом з повідомленнями, що дозволяє створювати інтерактивні меню, кнопки та швидкий доступ до функцій бота.

*Робота з командами:* Боти можуть реагувати на команди, введені користувачами, та виконувати певні дії. Це дає можливість створювати прості та зручні інтерфейси для взаємодії з користувачами.

*Отримання сповіщень:* Боти можуть отримувати сповіщення про різні події, такі як нові повідомлення, приєднання нових користувачів, зміна статусу та інші. Це дозволяє ботам реагувати на події в реальному часі та виконувати відповідні дії.

*Переваги Telegram Bot API:*

1. Простота використання: Telegram Bot API має зрозумілий та простий для розробників інтерфейс, що полегшує розробку та розгортання телеграм-ботів.
2. Багатофункціональність: API надає широкий набір можливостей для створення різноманітних ботів, включаючи роботу з медіафайлами, клавіатурами та командами.
3. Велика активна спільнота: Телеграм має широку та активну спільноту розробників, яка надає підтримку, документацію та приклади коду для розробки ботів.
4. Висока безпека: Telegram використовує криптографічні протоколи для захисту приватності та безпеки користувачів, що робить його привабливим для створення захищених ботів.

*Недоліки Telegram Bot API:*

1. Обмежена функціональність порівняно з деякими іншими платформами месенджерів. Деякі більш спеціалізовані функції або можливості можуть бути відсутніми у Telegram Bot API.
2. Обмеження на кількість повідомлень та швидкість відправки: Telegram має обмеження на кількість повідомлень, які можуть бути відправлені за певний проміжок часу, а також на швидкість відправки повідомлень, що може обмежити певні сценарії використання.
3. Залежність від доступу до Інтернету: Для роботи телеграм-бота необхідне постійне з'єднання з Інтернетом, що може становити проблему в деяких випадках.

*BotFather*

BotFather — це спеціальний бот в Telegram, який дозволяє користувачам створювати та керувати своїми власними телеграм-ботами. Він надає простий інтерфейс для створення нових ботів, налаштування їх параметрів та отримання токена доступу, який потрібно використовувати для взаємодії з Telegram Bot API [9]. Історія створення BotFather пов'язана з розвитком платформи Telegram. BotFather був створений компанією Telegram з метою полегшити процес створення та управління телеграм-ботами. Він був введений разом з появою підтримки ботів у Telegram, щоб надати зручний спосіб для створення та налаштування ботів.

Основні можливості BotFather:

*Створення нового бота:* BotFather надає можливість створити нового бота за допомогою простої команди. Користувач може вказати назву бота, його короткий опис та встановити його аватарку.

*Налаштування параметрів бота:* BotFather дозволяє користувачам налаштувати різні параметри свого бота, такі як команди, клавіатури, дозволи на доступ до повідомлень та інші.

*Отримання токена доступу:* Після створення бота BotFather надає унікальний токен доступу, який використовується для ідентифікації та взаємодії з Telegram Bot API. Цей токен використовується при написанні програмного коду для бота.

*Переваги BotFather:*

1. Простота використання: BotFather має дружній та зрозумілий інтерфейс, що полегшує створення та налаштування телеграм-ботів, навіть для користувачів без досвіду в розробці програмного забезпечення.
2. Швидкість та зручність: дякуючи BotFather процес створення та отримання токена доступу для бота займає лише кілька кроків і виконується швидко та без зайвих зусиль.
3. Інтеграція з Telegram Bot API: BotFather дозволяє легко отримати токен доступу, який потім можна використовувати для взаємодії з Telegram Bot API та розробки власного функціоналу для телеграм-бота.

#### 4. *Недоліки BotFather:*

5. Обмежені можливості: BotFather надає базовий набір функціональності для створення ботів, але він може бути обмежений в розширених випадках, коли потрібно реалізувати складніші сценарії або функції.
6. Відсутність гнучкості: BotFather надає обмежений набір параметрів для налаштування бота, іноді важко досягти бажаного функціоналу через обмеженість можливостей налаштування.

#### *Telegraf*

Telegraf — це високорівнева бібліотека на мові програмування Go, яка надає зручний і простий спосіб розробки телеграм-ботів. Вона забезпечує швидке та ефективне створення ботів з використанням Telegram Bot API [10]. Telegraf була створена командою розробників з компанії Telegram. Вона була розроблена для того, щоб спростити процес розробки телеграм-ботів та забезпечити зручні інструменти для взаємодії з Telegram Bot API. Перший реліз Telegraf був випущений у 2016 році і з того часу вона стала популярним вибором серед розробників ботів на мові програмування Go.

#### Основні можливості Telegraf:

*Зручний API:* Telegraf надає простий та зрозумілий API для створення та керування телеграм-ботами. Вона дозволяє розробникам взаємодіяти з Telegram Bot API без необхідності працювати з низькорівневими деталями протоколу.

*Обробка повідомлень:* Telegraf дозволяє створювати обробники повідомлень, які можуть відповідати на текстові повідомлення, команди, зображення, відео та інші типи повідомлень, які надсилаються користувачами.

*Розширені можливості:* Telegraf надає широкий спектр можливостей для розробки телеграм-ботів, включаючи клавіатури, inline-режим, розсилку повідомлень, роботу з базами даних та інші.

#### *Переваги Telegraf:*

1. Простота використання: Telegraf має простий і зрозумілий синтаксис, що полегшує розробку телеграм-ботів навіть для початківців.

2. Швидкість та ефективність: Telegraf написана на мові програмування Go, яка відома своєю швидкодією та ефективністю. Це дозволяє ботам, розробленим з використанням Telegraf, працювати швидко та ефективно.
3. Багата функціональність: Telegraf має багато функцій та розширених можливостей, які дозволяють розробникам створювати потужні та функціональні телеграм-боти.

#### *Недоліки Telegraf:*

1. Можливість обмеження: Telegraf надає базовий набір функціональності, і у вас можуть виникнути обмеження при реалізації складніших або специфічних сценаріїв.
2. Відсутність офіційної підтримки: Telegraf є відкритим проектом з активною спільнотою, але вона не має офіційної підтримки від компанії Telegram, що може вплинути на розробку та підтримку бота в майбутньому.

### **2.3 Аналіз бібліотек для веб скрапінгу**

При виборі технологій для створення додатку необхідно звертати увагу не тільки на фреймворк та переваги які він дає користувачу, але й на бібліотеки які сумісні з цим фреймворком або які він надає. Так як скрапінг веб сторінок це клопіткий процес а використання спеціалізованої бібліотеки зможе скоротити час та оптимізувати роботу виконавця. Ось декілька прикладів бібліотек які найчастіше використовуються для скрапінгу веб сторінок:

#### *BeautifulSoup*

BeautifulSoup є однією з найпопулярніших бібліотек для парсингу HTML і XML в мові програмування Python. BeautifulSoup була створена Леонардом Рамаліо (Leonard Richardson) у 2004 році як відкрите програмне забезпечення. Леонард хотів створити простий інструмент для парсингу HTML, який би був дружнім до початківців і мав зрозумілий API [11].

Результатом його роботи стала бібліотека BeautifulSoup, яка здобула широку популярність серед розробників.

Основні можливості бібліотеки BeautifulSoup:

*Парсинг HTML і XML.* BeautifulSoup надає потужні засоби для ефективного парсингу і аналізу структури HTML і XML документів.

*Навігація по дереву DOM.* Бібліотека дозволяє здійснювати навігацію по дереву DOM-елементів, знаходити теги за ім'ям, класом, атрибутами та іншими критеріями.

*Вилучення даних.* За допомогою BeautifulSoup можна легко вилучати текст, атрибути, посилання, зображення та інші дані з HTML-сторінок.

*Маніпуляція з даними.* Бібліотека дозволяє змінювати, додавати та видаляти елементи, атрибути та текст в HTML-документі.

*Підтримка різних парсерів.* BeautifulSoup підтримує різні парсери, включаючи вбудований парсер Python (html.parser), парсер lxml і парсер html5lib. Це дає більшу гнучкість і швидкість парсингу залежно від потреб проекту.

*Переваги BeautifulSoup:*

1. Простота використання. BeautifulSoup має простий інтерфейс API, що робить його легким у використанні, особливо для початківців.
2. Гнучкість. Бібліотека добре працює з різними типами HTML-коду і дозволяє витягувати дані зі складних структур.
3. Підтримка парсерів. BeautifulSoup може працювати з різними парсерами, що дозволяє вибирати найоптимальніший парсер для конкретного проекту.
4. *Недоліки BeautifulSoup:*
5. Швидкість. У порівнянні з деякими іншими бібліотеками для парсингу, BeautifulSoup може бути трохи повільнішим, особливо при обробці великих обсягів даних. У таких випадках рекомендується розглянути альтернативні швидкісні парсери, наприклад, lxml.



- б. Недостатня підтримка CSS-селекторів. BeautifulSoup не надає повну підтримку CSS-селекторів, які є потужним інструментом для вибору елементів на веб-сторінці. Однак, можна поєднувати BeautifulSoup з іншими бібліотеками, такими як CSSSelector, для зручного використання CSS-селекторів.

### *Puppeteer*

Бібліотека Puppeteer є потужним інструментом для контролю та автоматизації веб-браузера Chrome, розробленим командою Google Chrome. Puppeteer був вперше випущений у 2017 році командою розробників Google Chrome [12]. Оригінальною метою створення бібліотеки було надання розробникам зручного способу автоматизувати браузерні задачі, такі як тестування, збір даних та взаємодія з веб-сайтами. Puppeteer був швидко прийнятий спільнотою розробників і швидко здобув популярність завдяки своїм потужним можливостям і зручному API.

Основні можливості бібліотеки Puppeteer:

*Керування браузером Chrome.* Puppeteer дозволяє створювати нові випадки браузера Chrome і повністю керувати ними за допомогою програмного коду. Ви можете відкривати сторінки, навігувати по ним, заповнювати форми, клікати на елементи та виконувати різні дії, які зазвичай роблять користувачі.

*Автоматизація веб-скрапінгу.* Puppeteer надає засоби для створення скриптів, які можуть зчитувати дані з веб-сторінок, виконувати пошук, вилучати і зберігати інформацію. Ви можете легко вибирати елементи на сторінці за допомогою селекторів і витягувати потрібні дані.

*Генерація знімків екрана.* Puppeteer дозволяє створювати знімки екрана веб-сторінок, які можуть бути корисні для відладки, створення скріншотів або створення візуальних тестів. Також знімки можна поширювати між користувачами проекту або за допомогою мережі та зберігати для власного використання.

*Обробка асинхронних подій.* Puppeteer підтримує асинхронні операції, такі як очікування завантаження сторінки, обробка подій, запуск JavaScript на сторінці тощо.

*Переваги Puppeteer:*

1. Повний контроль над браузером. Puppeteer дозволяє повністю контролювати веб-браузер Chrome, що відкриває широкі можливості для автоматизації різних задач.
2. Широкий функціонал. Бібліотека надає багато вбудованих методів та функцій для взаємодії з браузером і веб-сторінками.
3. Підтримка різних платформ. Puppeteer доступний для різних платформ, включаючи Windows, macOS і Linux.

*Недоліки Puppeteer:*

1. Обмежена підтримка інших браузерів. Puppeteer спеціально розроблений для контролю над браузером Chrome і не надає такої ж розширеної підтримки для інших веб-браузерів.
2. Великий розмір. Бібліотека Puppeteer має великий розмір, оскільки вона включає в себе повний функціонал браузера Chrome. Це може вплинути на час завантаження та виконання скриптів.

*Guzzle*

Бібліотека Guzzle є однією з найпопулярніших бібліотек для роботи з HTTP-запитами в мові програмування PHP. Guzzle була створена Майклом Доусоном (Michael Dowling) у 2011 році. Він розпочав розробку Guzzle з метою створення простої, потужної та елегантною бібліотеки для роботи з HTTP-запитами. Guzzle поступово набула популярності серед PHP-розробників і стала однією з основних бібліотек для роботи з HTTP в PHP-екосистемі.

*Основні можливості бібліотеки Guzzle:*

*Виконання HTTP-запитів.* Guzzle дозволяє легко виконувати HTTP-запити, такі як GET, POST, PUT, DELETE і т. д. в PHP-проектах. Вона надає простий та зрозумілий API для створення та налаштування запитів.

*Підтримка різних протоколів.* Бібліотека підтримує різні протоколи, такі як HTTP, HTTPS, REST, SOAP і багато інших, що робить її універсальним рішенням для комунікації з веб-серверами.

*Робота з файлами та потоками.* Guzzle дозволяє відправляти файли на сервер та отримувати файли від сервера. Вона також підтримує роботу з потоками даних, що дозволяє ефективно обробляти великі об'єми даних.

*Автоматична обробка перенаправлень та автентифікація.* Guzzle автоматично обробляє перенаправлення, а також надає зручні можливості для автентифікації, включаючи базову автентифікацію, OAuth і багато інших методів.

*Зручна обробка помилок.* Guzzle має вбудовану обробку помилок, що дозволяє легко виявляти та обробляти помилки, пов'язані з HTTP-запитами.

Переваги Guzzle:

1. Простота використання. Guzzle має зрозумілий та дружній API, що робить його легким у використанні.
2. Розширюваність. Бібліотека дозволяє розширювати свої можливості шляхом використання плагінів та розширень.
3. Широкі можливості налаштування. Guzzle надає розширені можливості налаштування HTTP-запитів, включаючи налаштування заголовків, параметрів, тайм-аутів тощо.

Недоліки Guzzle:

1. Велика кількість коду. Завдяки своїм багатьом можливостям і функціоналу, Guzzle може мати велику кількість коду, що може бути необхідно враховувати при розгортанні на веб-сервері.
2. Вимоги до пам'яті. У випадку обробки великих обсягів даних, Guzzle може вимагати більше пам'яті, особливо якщо використовується потоковий режим.

У висновку проаналізувавши технології для створення веб додатків та телеграм ботів, враховуючи всі їх плюси і мінуси мною були обрані наступні технології. Django фреймворк для створення інтерфейсу користувача та

аналізатора коефіцієнтів на маркети спортивного матчу. Бібліотеку Puppeteer (її API “Puppeteer Sharp”) для створення парсера веб сайту. Бот BotFather для створення телеграм боту для користувача.

## 3 РОЗРОБКА АНАЛІЗАТОРА ЛІНІЇ ПОДІЙ БУКМЕКЕРСЬКОЇ КОНТОРИ

### 3.1 Опис предметної області

Розглядаючи сьогоденні тенденції світу азартних ігор та букмекерських контор можна побачити що компанії мають скептичне бачення на системи збору інформації з індивідуальним функціоналом і заохоченням даних з іншими конкурентами. Так як компанії витрачають мільйони на маркетингові проекти та рекламу власного проекту подібні системи виключають можливість спонтанного на трапляння користувача саме на сайт контори через перехід по рекламі або вивіски на білбордах. В той же час системи аналітики та збору інформації про матчі та деталі їх проведення мають місце використовуватись в середині самої букмекерської контори. Кожен підприємець буде бажати мати під рукою інформацію зі всіх його конкурентів, щоб вигідно просувати свої ідеї і захоплювати великі частини ринку без особливих витрат та угод.

Деякі компанії пішли далі і зайняли сторону нейтралітету між звичайними користувачами та букмекерськими конторами. Вони вирішили створювати інформаційні системи з зчитування даних з букмекерських контор та маскувати їх під власні обчислення. Це дає їм змогу безперешкодно викачувати дані з сайтів контор при цьому не боятися кримінального впровадження або санкцій з боку контори [14]. Отримані дані вони або пропонують букмекерським конторам, або продають звичайним користувачам які шукають найкращу пропозицію.

Користуючись послугами даної компанії або виключно аналізатора подій користувач має змогу не лише слідкувати за вибраним матчем а й за усією ситуацією зміни подій в ігровому полі. Перед ним відкриється можливість моніторингу матчів які публікують букмекерські контори без потреби перегляду усіх сайтів контор у пошуку найкращого варіанту. За

бажанням при вправності використання користувач може публікувати дані найкращих варіантів даних матів для власної вигоди в мережах “Facebook” або “Instagram” виймаючи з цього власну вигоду.

### **3.2 Архітектура системи**

Аналізатор лінії подій букмекерських контор представлений в даній роботі, поділяється на три частини:

1. Додаток для парсингу даних з веб сторінки та запису знайдених даних.
2. Телеграм бот для керуванням парсеру та отримання повідомлень
3. Веб інтерфейс для користувача з даними та можливістю відображення відповідних даних. В інтерфейсі користувач має можливість вибрати зі списку вид спорту, країну проведення, турнір з виду спорту країни та сам матч з відповідною інформацією про дату та час проведення, букмекерів та коефіцієнти які він надає до обраного матчу.

На діаграмі розгортання (див. Рис. 5) показано фізичне розташування проекту.

### **3.3 Функціональні вимоги системи**

У відповідності з архітектурою системи вона поділяється на три складові частини, тому необхідно спроектувати функціональні вимоги як для системи в цілому, так і для кожної частини окремо.

Система повинна забезпечувати:

1. Зчитування даних — задача парсера зчитати дані з відповідного сайту й записати отримані дані до відповідного каталогу на зберігаючому пристрої локальної обчислювальної машини. До даних відноситься вид спорту (футбол, баскетбол, волейбол, хокей, тощо), країна проведення матчів з попередньо зчитаного виду спорту, турнір країни з зчитаного виду спорту та кількість матчів які на даний момент будуть в ньому проходити, матч з зчитаного

турніру який в собі повинен мати назву обох команд, дату проведення матчу, час проведення матчу, назву букмекера який видає на своєму сайті даний матч, коефіцієнт який надає букмекер на зчитаний матч.

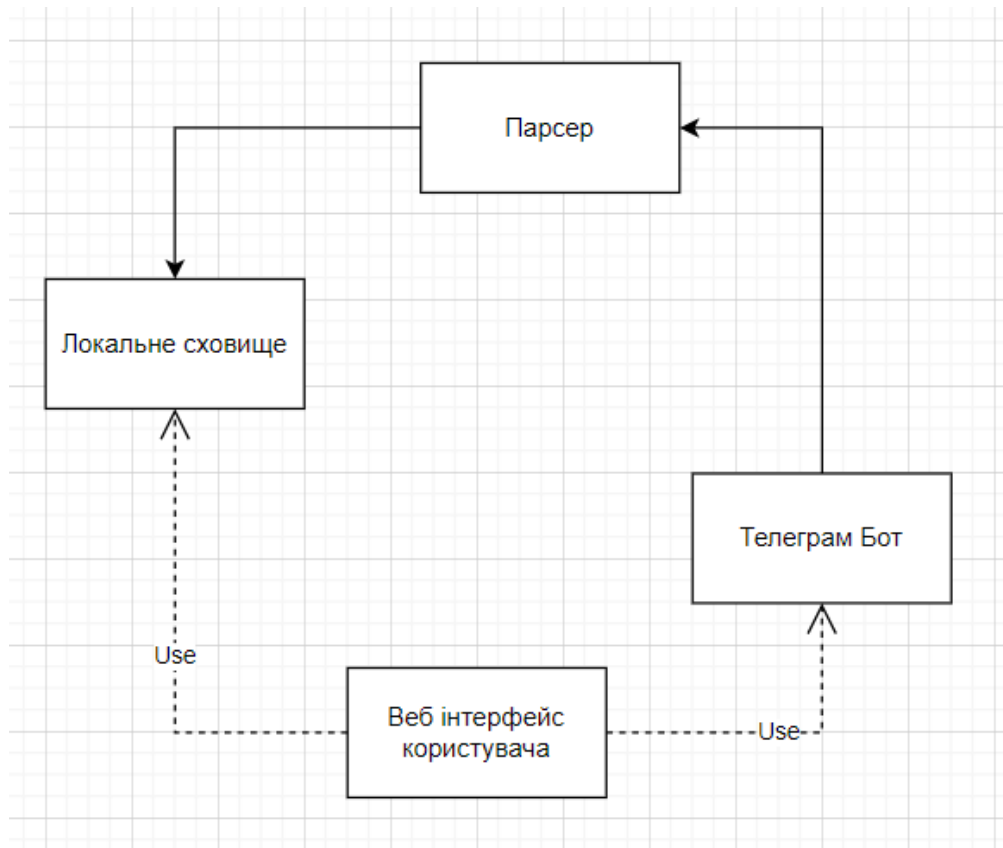


Рис. 5 Діаграма розгортання системи

2. Зчитування даних — задача парсера зчитати дані з відповідного сайту й записати отримані дані до відповідного каталогу на зберігаючому пристрої локальної обчислювальної машини. До даних відноситься вид спорту (футбол, баскетбол, волейбол, хокей, тощо), країна проведення матчів з попередньо зчитаного виду спорту, турнір країни з зчитаного виду спорту та кількість матчів які на даний момент будуть в ньому проходити, матч з зчитаного турніру який в собі повинен мати назву обох команд, дату проведення матчу, час проведення матчу, назву букмекера який видає на своєму сайті даний матч, коефіцієнт який надає букмекер на зчитаний матч.

3. Коректний запис отриманих даних до JSON файлу. Усі отримані дані повинні бути записані до файлу у доступному форматі для цього подальшого безпроблемного зчитування та ілюстрування на веб інтерфейсі.

4. Коректна робота телеграм бота — при запуску парсера з телеграм бота користувач повинен мати змогу як запустити парсер так і при його виконанні зчитування даних та запису їх до файлу зробити аналіз даних на момент так званої “букмекерської вилки”. Після виконання аналізу користувач повинен побачити результат перевірки у зручній для нього формі з виводом інформації про вид спорту, країну проведення турнір та дані про сам матч з посиланням на нього на сайт джерело.

5. Аналізатор даних матчу — окрім збору даних додаток повинен мати змогу аналізувати отримані дані порівнюючи коефіцієнти букмекерів на матч методом перебору їх значень і у випадку передбаченим заделегідь встановленою формулою пошуку дефективних коефіцієнтів виносити їх дані на веб інтерфейс для користувача або в телеграм бот.

6. Веб інтерфейс — для користувача має бути реалізований веб інтерфейс з можливістю відображення усієї зчитаної інформації занесеної до JSON формату. Уся інформація повина подаватися у зручній та зрозумілій для користувача формі. Він повинен мати змогу оперативно та легко знаходити відповідний матч за допомогою списків з видами спорту, країни, турніру та матчів. Для матчів необхідно відображати дату та час проведення, також необхідним є представлення для користувачеві повного списку букмекерів та їх виставлених коефіцієнтів на цей матч. З функціональної точки зору для користувача повинна бути реалізована механіка аналізатора матчів на предмет некоректних коефіцієнтів для цього повинна бути представлена необхідна кнопка. Немало важливим критерієм є додавання кнопки з посиланням на телеграм бота для того щоб користувач мав можливість через бота запустити веб скрапер сторінки та переглянути список знайдених матчів та результат їх аналізу.



### 3.4 Проектування системи проекту

Під час етапу проектування було проведено структурно-функціональне моделювання, котре дозволило прослідити виконання всіх етапів проектування застосунку, проаналізувати напрямки удосконалення роботи.

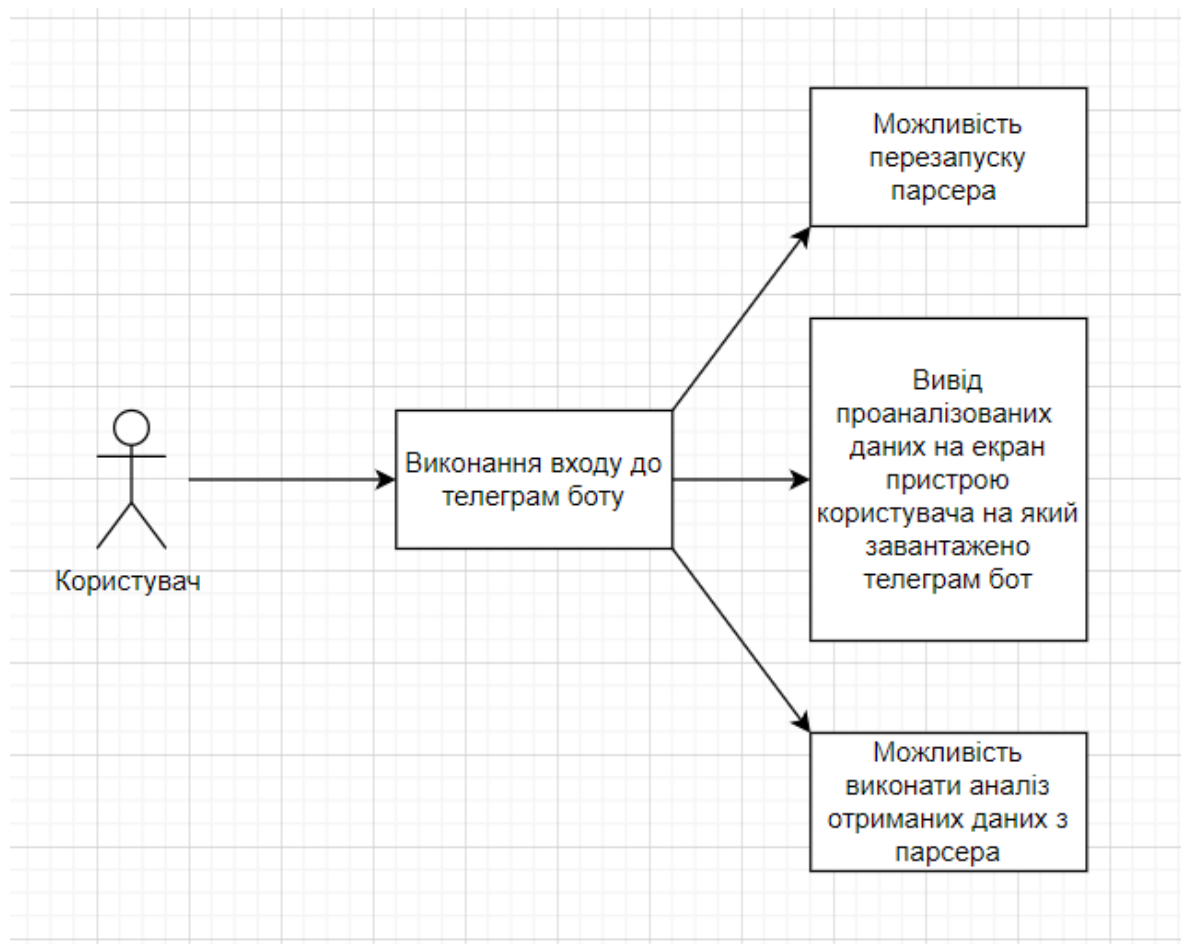


Рис. 6 Діаграма варіантів використання для користувача

Головною метою проектування є забезпечення ефективного функціонування застосунку у відповідності до поставлених вимог та при наявності заданих обмежень та технології. Для виділення основних можливостей системи також було створено діаграми варіантів використання для користувача. Діаграми варіантів використання потрібні для відображення дій, які зовнішній користувач (актор) може виконувати у системі [15].

Діаграми варіантів використання для користувача наведені на рисунках 6 та 7 відповідно.

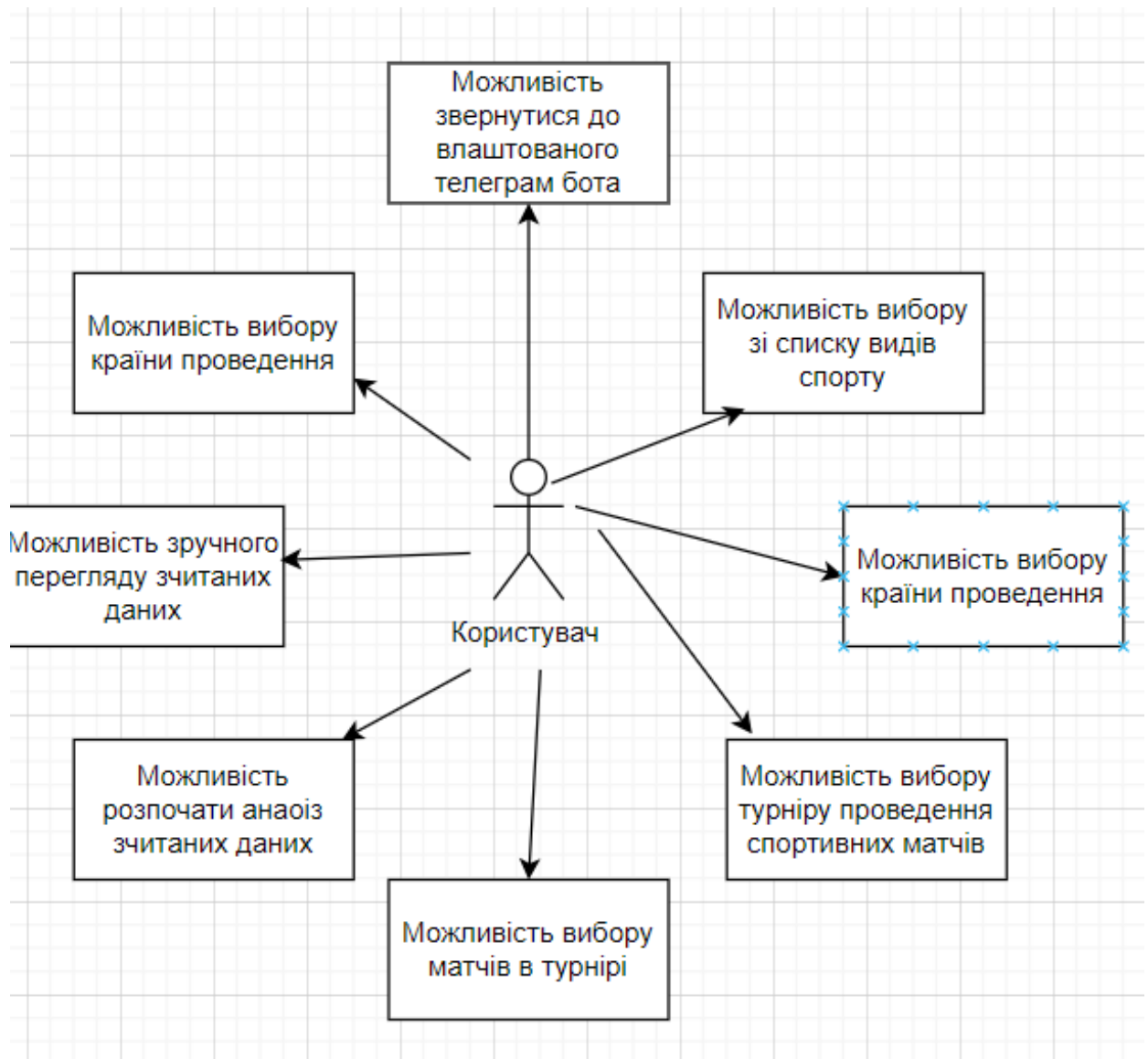


Рис. 7 Діаграма варіантів використання для користувача

Для проектування системи парсера було використано метод об'єктно орієнтованого програмування який включає в себе зв'язки між осередкованими класами для їх взаємної працездатності (див. Рис. 8).

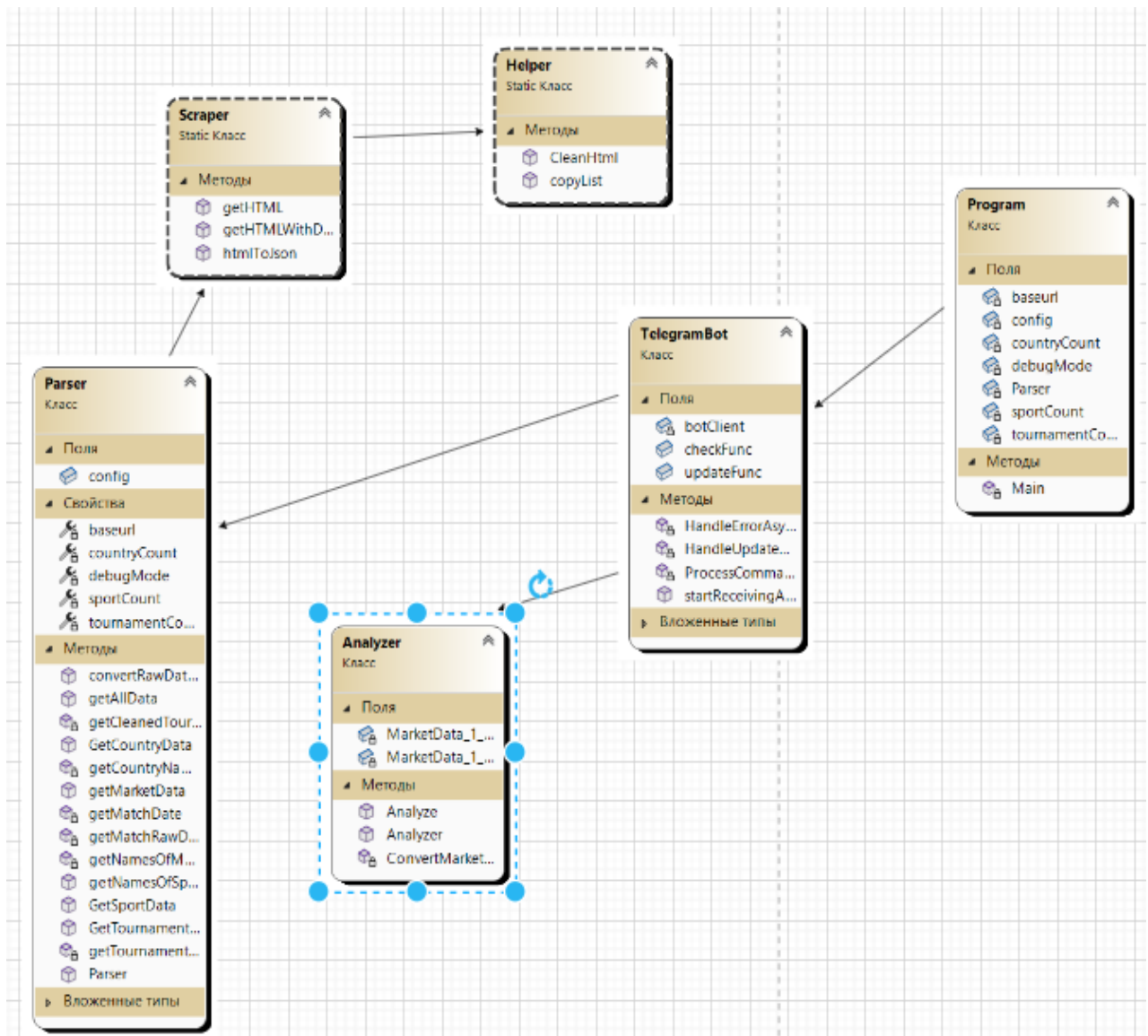


Рис. 8 Діаграма класів для системи парсера

Для створення веб інтерфейсу для користувача окрім використання Django фреймворку та його загальних можливостей, також була побудована спеціалізована структура файлів розмітки у вигляді дерева, що дозволяє їй не мати у подальшому проблем у випадку масштабування та розвитку проекту (див. Рис. 9).

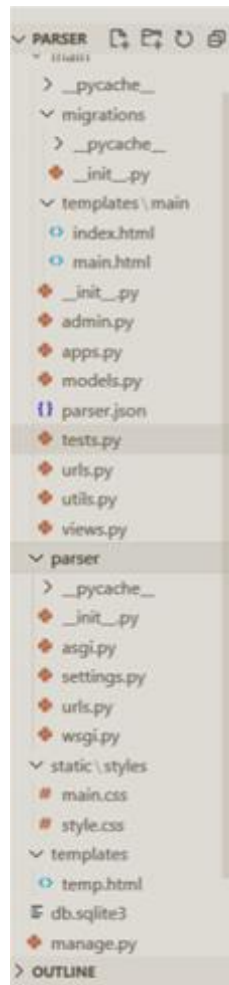


Рис. 9 Схема програми інтерфейсу користувача

В подальшому для більшого розуміння мети проектування та вимог до проекту були сформульовані таблиці предметно-наслідкових дій для користувача. Таблиця дій та наслідків, також відома як матриця дій та наслідків або таблиця вагомості, є інструментом прийняття рішень, який допомагає візуалізувати можливі варіанти дій та їхні наслідки. Ця таблиця зазвичай використовується для аналізу і порівняння альтернативних варіантів і допомагає прийняти обгрунтоване рішення. Серед переваг такого методу дослідження можна визначити його структурованість, віртуалізацію та аналіз ризиків. Для свого проекту метод був використаний виключно для наглядного проектування наслідків дій користувача на системи застосунку. Усі можливі дії користувача в системі телеграм боту, та прояви поведінки застосунку на них зазначені в таблиці 1.

Таблиця 1

*Варіації поведінки системи “телеграм бота”*

<b>Номер дії</b>	<b>Дія користувача</b>	<b>Наслідки дії користувача</b>
1	Користувач реєструє телеграм бот.	Бот вітає користувача та пропонує йому свої послуги.
2	Користувач бажає розпочати парсинг та вводить необхідну команду.	Бот отримує необхідну команду яка відловлюється системою парсингу та починає процес веб скрапінгу сторінки джерела.
3	Користувач очікує результат завершення парсингу.	По звершенню парсер повідомляє про це бот, який в свою чергу виводить на екран пристрою користувача про завершення процесу.
4	Користувач бажає провести аналіз отриманих даних і вводить відповідну команду.	Бот отримує команду яка відловлюється системою парсера та починає перевіряти останні дані які були отримані.
5	Користувач очікує результат перевірки даних.	По завершенню перевірки парсер повідомляє бот про завершення та відсилає йому результати перевірки які в свою чергу бот виставляє у зручному для розуміння користувачу форматі.

Усі можливі дії користувача в системі веб інтерфейсу, та прояви поведінки застосунку на них зазначені в таблиці 2.

Таблиця 2

*Варіації поведінки системи “веб інтерфейсу”*

<b>Номер дії</b>	<b>Дія користувача</b>	<b>Наслідки дії користувача</b>
1	Користувач заходить до веб інтерфейсу застосунку.	Інтерфейс відображає наявні дані зібрані з останнього парсингу сайту джерела. У

		випадку відсутності останніх, застосунок пропонує посилання на телеграм бота для користувача, щоб той зміг запусити парсер.
2	Користувач бажає переглянути список видів спорту і відкриває відповідний список.	Веб застосунок розгортає список видів спорту виставляючи список країн в який на даний момент заплановані турніри та матчі.
3	Користувач бажає переглянути турніри для вибраної країни натиснувши на бажану країну.	Веб застосунок розкриває список турнірів які проходять на даний момент у цій країні з попередньо вибраного виду спорту.
4	Обравши турнір користувач бажає переглянути дані про бажаний матч та натискає на нього.	Веб застосунок розкриває інформацію про вибраний матч до якої входить: обидві назва команд суперниць, дата проведення мату, час проведення матчу.
5	Користувач бажаючи переглянути додаткову інформацію про матч натискає відповідну кнопку для додаткової інформації.	Веб застосунок показує перед користувачем додаткову інформацію про матч до якої входить: список букмекерів які пропоную обраний матч на своїх сайтах, коефіцієнти цих букмекерів на даний матч які висвітленні на момент останнього парсингу сайту джерела.
6	Користувач бажає проаналізувати отримані дані з обраного матчу, і натискає на відповідну кнопку для аналізу даних.	Веб застосунок перевіряє наявні коефіцієнти букмекерів на даний матч, та при виявленні некоректних коефіцієнтів залишить зі списку тільки ці коефіцієнти та букмекерів які їх надали.

### 3.5 Програмна реалізація застосунку

#### 3.5.1 Програмна реалізація парсингової частини

Розглядаючи структуру застосунку можна виділити основні компоненти та їх взаємозв'язок:

Клас `Program` — основний клас серверного застосунку. В собі передбачає ліміти для зчитування сторінки, створенні для пришвидшення зчитування інформації та перешкоджанню надлишкового навантаження на системи пристрою виконавця. Також в собі клас має усі базові посилання на веб-сайт джерело, що полегшує їх використання так як всі вони знаходяться в одному місці. Більш того в класі знаходяться формули для аналізу коефіцієнтів спортивних матчів. Але основними функціями класу є методи створення файлу після завершення процесу зчитування інформації, занесення усіх даних в нього та початок прослуховування команд телеграм ботом.

Клас `TelegramBot` — клас відповідальний за роботу та коректність роботи команд телеграм бота. В своєму розпорядженні клас має прослуховувати дві команди на перезапуск парсера та аналіз зчитаних даних. При перехопленні однієї з команд через метод та делегат виконується один з анонімних методів в класі `Program`.

Клас `Scraper` — клас відповідальний за отримання HTML та JSON даних з веб сайту джерела. Перед парсингом самих даних клас `Program` звертається саме до методів класу `Scraper` який в свою чергу методом переходу на сайт джерело копіює його код сторінки у двох випадках або HTML код або JavaScript код у випадку коли на сторінки є приховані дані.

Клас `Parser` — основний клас обробки даних сторінки сайту джерела. В його розпорядженні знаходяться схожі за структурою та призначенням методи спрямовані на те щоб витягнути з об'єму даних коду сторінки саме потрібні дані. Виконується це наступним чином:

1. Серед коду сторінки знаходиться маркер по якому можна знайти бажану назву або частину коду сторінки (маркер ще називають якорем і найчастіше це повторюваний HTML тег або збірник контексту в кінці сторінки).
2. Після знаходження маркера необхідно за допомогою методів обробки рядків, обрізати необхідний матеріал до прийняттого формату та

помістити його в список який в подальшому буде використовуватись в ієрархічній системі списків даного проекту.

Клас `Helper` — допоміжний клас методи якого використовуються для копіювання списків у випадку формулювання коректної назви турніру та обох команд суперниць.

Клас `Analyzer` — функціональний клас для створення аналізатору отриманих даних, та відправки їх для обробки в телеграм бот. Користуючись формулами які знаходяться в класі `Program`, задача методів класу перевірити коефіцієнти матчу у правильному порядку та коректно сформулювати результат для виведення в телеграм боті. Серед відправлених даних повинні бути посилання на матч на сайт джерело, дані про вид спорту , країну, турнір та коефіцієнти на матч.

Перш за все розробка почалась зі створення функціональної системи телеграм боту, так як він є відправною точкою у працездатності застосунку. Для цього був створений метод “`ProcessCommandAsync`”, який після відлову команди від телеграм боту запускає делегований метод для обробки інформації (див. Лістинг 1).

#### Лістинг 1 Код методу `ProcessCommandAsync`

```
private async Task ProcessCommandAsync(string command,
Message message)
{
    switch (command)
    {
        case "/refresh":
            {
                if(updateFunc != null)
                {
                    await
botClient.SendTextMessageAsync(message.Chat, "I am start
refresh! ");
```



```

        Console.WriteLine("refresh
started");
        updateFunc();
        Console.WriteLine("refresh
ended");
        await
botClient.SendTextMessageAsync(message.Chat, "I am
refreshed! ");
        await
botClient.SendTextMessageAsync(message.Chat, data);
    } else
    {
        await
botClient.SendTextMessageAsync(message.Chat, "No special
action");
    }
    break;
}
case "/check":
{
    if (checkFunc != null)
    {
        await
botClient.SendTextMessageAsync(message.Chat, "I am start
check! ");
        Console.WriteLine("check
started");
        List<string> data =
checkFunc();

```

```

        Console.WriteLine("check
ended");

        await
botClient.SendTextMessageAsync(message.Chat, "I am a
checked! ");

        data.ForEach(async (value)
=>
        {
            int maxMessageLength =
4096; // Maximum message length allowed by Telegram Bot API

            while (value.Length >
0)
            {
                string
currentMessage = value.Substring(0,
Math.Min(maxMessageLength, value.Length));

                await
botClient.SendTextMessageAsync(message.Chat,
currentMessage);

                value =
value.Remove(0, currentMessage.Length);
            }
        });
    } else
    {
        await
botClient.SendTextMessageAsync(message.Chat, "No special
action");
    }
    break;
}
default:

```

```

        await
botClient.SendTextMessageAsync(message.Chat, "No special
action");

        break;
    }
}

```

Другим не менш важливим компонентом системи є аналізатор збережених коефіцієнтів. Його реалізація проводиться після створення файлу з даними для того щоб мінімізувати ризики застримки в часі розробки та подальших проблем. Для цього був створений метод “Analyze”, який методом перебору даних виділяє потрібну інформацію яка потрібно перевірити та збирає її у список який передається на вивід для користувача (див. Лістинг 2).

#### Лістинг 2 Код методу *Analyzer*

```

public List<string> Analyze(MainData data)
{
    var res = new List<string>();

    foreach(var sport in data.sportsDatas)
    {
        Console.WriteLine(sport.title + "
started");
        foreach(var country in sport.countries)
        {
            Console.WriteLine(country.title + "
started");

            foreach (var tournament in
country.tournaments)
            {
                Console.WriteLine(tournament.title
+ " started");

                var dataString = "";
                dataString += sport.title + " " +
country.title + " " + tournament.title +
Environment.NewLine;
                dataString += tournament.link +
Environment.NewLine;
            }
        }
    }
}

```

```

        dataString += Environment.NewLine;
        bool needAdd = false;
        foreach (var match in
tournament.matches)
        {
            Console.WriteLine(match.title +
" started");

            IAnalyzableMarketData?
optAnalyzableData = ConvertMarketData(match.marketData);
            if (optAnalyzableData != null)
            {
                string matchData = "";
                IAnalyzableMarketData
marketData = optAnalyzableData!;

                var indexes =
marketData.Analyze();

                matchData += " " +
match.title + Environment.NewLine;
                matchData += match.date +
Environment.NewLine;
                matchData += match.link +
Environment.NewLine;

                if (indexes.Count <= 0)
matchData += "No suitable data";
                else needAdd = true;
                indexes.ForEach((index) =>
                {
                    matchData +=
match.marketData.ToString(index) + Environment.NewLine;
                });
                matchData +=
Environment.NewLine;

                if (indexes.Count > 0)
dataString += matchData;

            }
            dataString +=
Environment.NewLine;

            Console.WriteLine(match.title +
" finished");
        }
        if(needAdd) res.Add(dataString);
        Console.WriteLine(tournament.title
+ " finished");
    }

```

```

        Console.WriteLine(country.title + "
finished");
    }
    Console.WriteLine(sport.title + "
finished");
}
return res;
}

```

Важливим етапом в розробці парсера було створити функціонал який буде зчитувати саме маркети та коефіцієнти матчів методом обробки відповідного коду сторінки та відтинання саме потрібної інформації. Усі відповідні дані записуються в списки для подальшої обробки. Для цього був створений метод “getMarketData”, (див. Лістинг 3).

### Лістинг 3 код методу *getMarketData*

```

public MarketData getMarketData(string content)
{
    string[] splittedcontent =
content.Split("\n");
    List<string> headers = new List<string>();
    headers.Add("Bookmakers");
    bool headersCheackout = false;
    int count = 0;
    MarketData marketData = new MarketData();
    List<string> temp = new List<string>();
    for (int i = 0; i < splittedcontent.Length;
i++)
    {
        if (!headersCheackout)
        {
            if (splittedcontent[i] == "Payout")
            {

```

```

        headersCheackout = true;
        marketData.headers = headers;
    }
    else
    {
headers.Add(splittedcontent[i]);
        }
    }
    else
    {
        if (count == 0)
        {
            temp = new List<string>();
        }

        string data = splittedcontent[i];
        if (data.Split(" ").Length > 3)
        {
            break;
        }
        temp.Add(data);
        count++;
        if (count >= headers.Count)
        {

marketData.data.Add(Helper.copyList(temp));
            count = 0;
        }
    }
}
return marketData;
}

```

### 3.5.2 Програмна реалізація інтерфейсу користувача

Основні компоненти веб інтерфейсу користувача можна поділити на три основні частини:

URL-адреси — власноруч прописані адреси маршрутизації, для взаємодії між застосунком та користувачем. За їх допомоги користувач може бачити інформацію яка формується з цих посилань.

View-функції — функції для обробки даних що знаходяться на сторінці сайту. За їх допомоги перед потраплянням на головну сторінку дані обробляються та формуються належним чином.

HTML-темплейт — компонент що динамічно обробляється фреймворком Django. View-функції створюють контент для обробки в HTML-темплейтах.

Окрім основної веб сторінки для користувача, необхідно було створити функціонал перебору зчитаних елементів та виводу їх на головний екран (див. Лістинг 4).

#### Лістинг 4 Код методу перебору зчитаних даних

```
% if not sports %}
    <li><a href="https://t.me/manual_parser_bot">gen
new data</a></li>
    {% else %}
        {% for sport in sports %}
            <li><button class="button-85"
role="button"><a href="{% url 'discipline'
name=sport%}">{{sport}}</a></button></li>
        {% endfor %}
    {% endif %}
```

### 3.6 Демонстрація веб-інтерфейсу користувача

Відповідно до функціональних вимог була розроблена клієнтська програма, де реалізовані всі зазначені функції (див. Рис. 10).

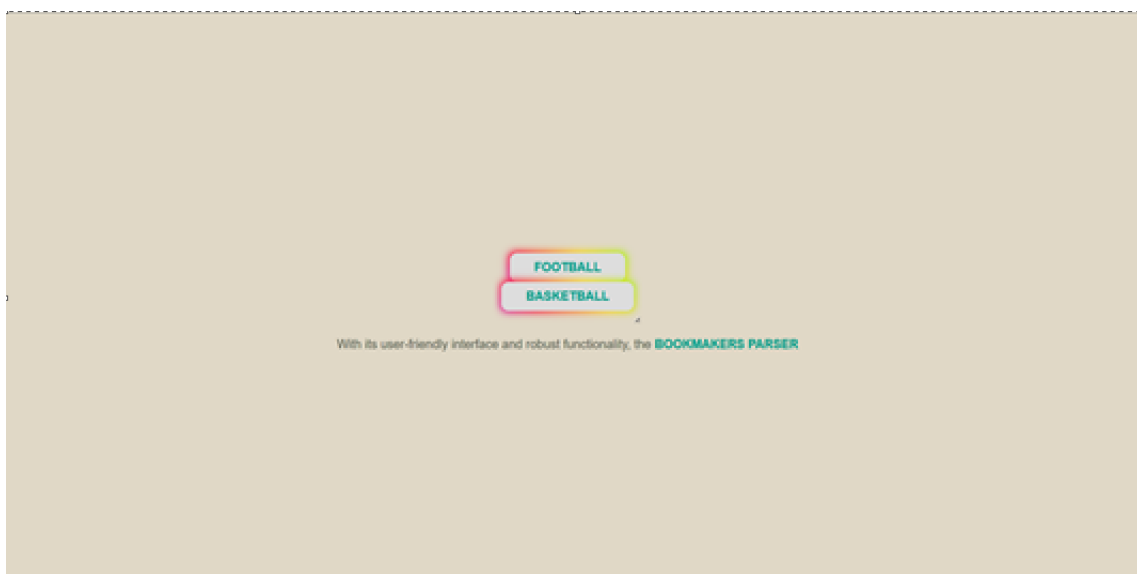


Рис. 10 Головне меню інтерфейсу

Далі користувач має змогу обрати країну проведення яку забажає (див. Рис. 11).

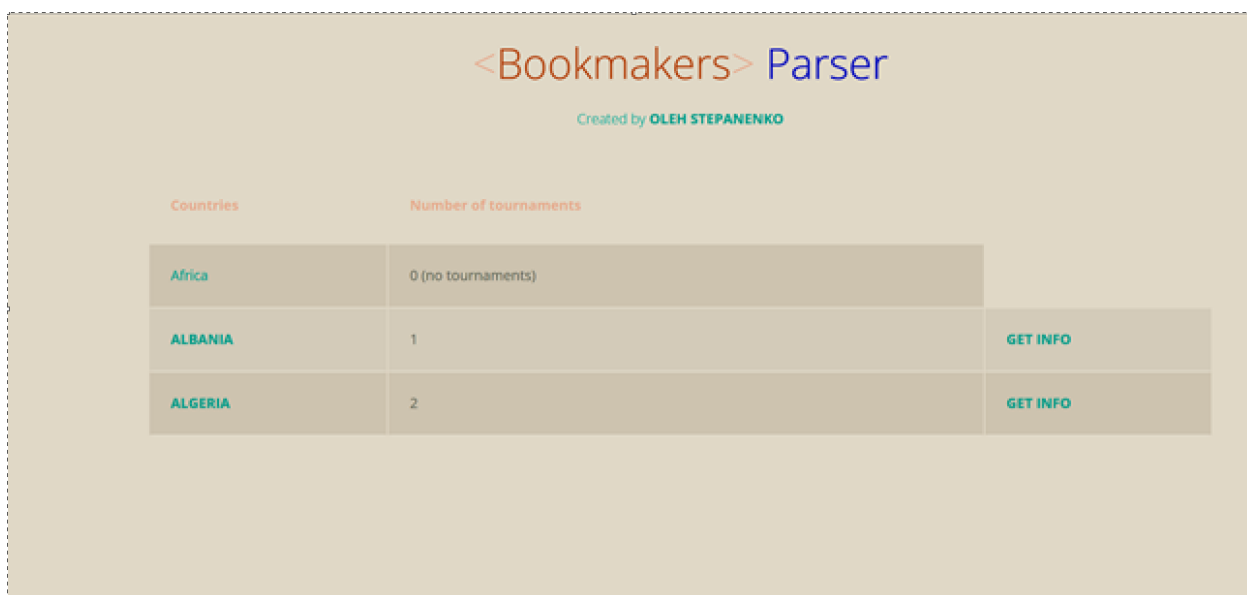


Рис. 11 Меню обраної країни проведення



Наступним кроком користувач може обрати турнір проведення виду спорту (див. Рис. 12).

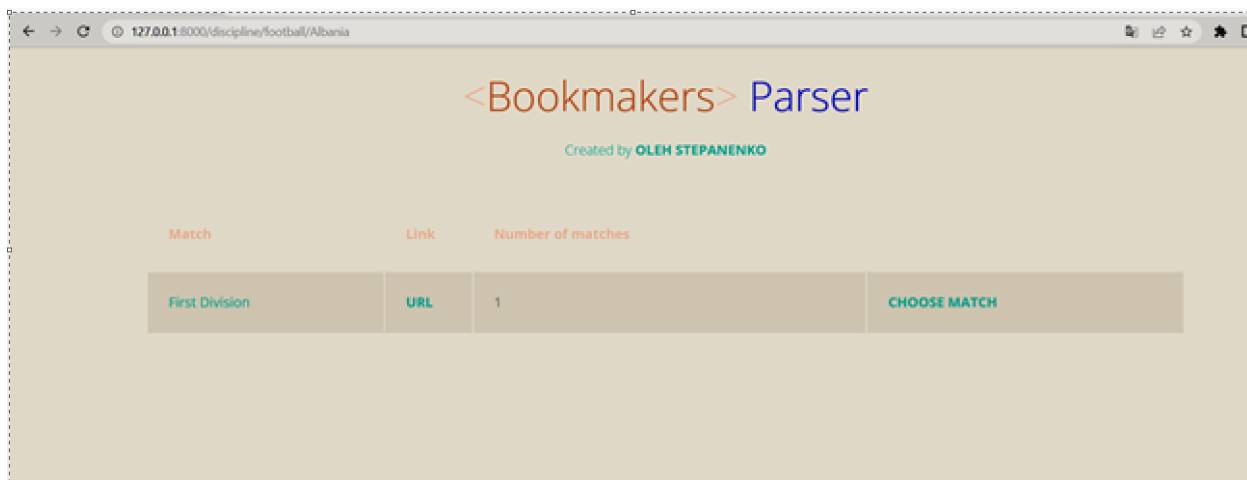


Рис. 12 Меню обраного турніру проведення матчів

Після цього перед користувачем відкривається можливість вибрати один зі списку матчів які проходять у заздалегідь вибраному турнірі (див. Рис. 13).

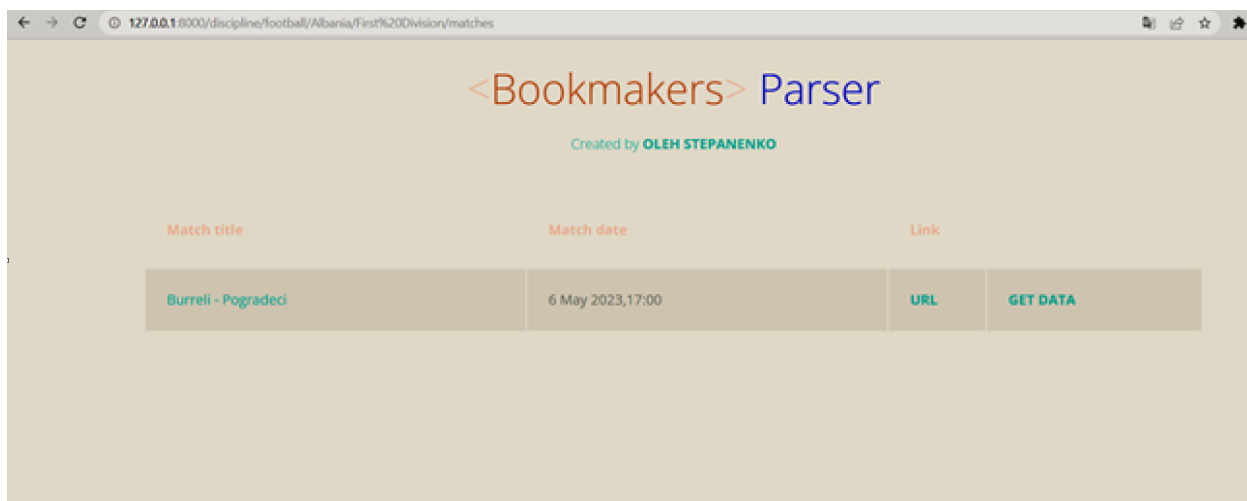


Рис. 13 Меню обраного матчу

Забажавши подивитися розширену інформацію про матч користувач може натиснути відповідну кнопку біля даних про матч (див. Рис. 14).



Рис. 14 Меню додаткової інформації про матч

По завершенню користувач може використати аналізатор коефіцієнтів на вибраний матч та побачити тільки тих букмекерів та коефіцієнти які є дефектними (див. Рис. 15).

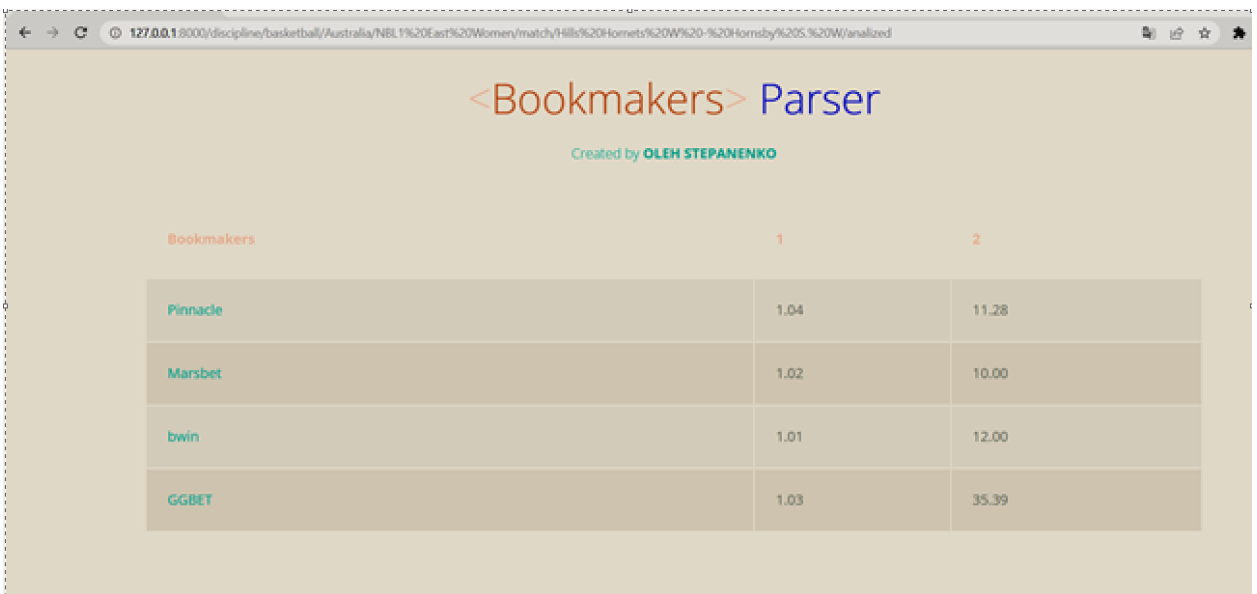


Рис. 15 Екран інтерфейсу після роботи аналізатора за вибраним матчем

## ВИСНОВКИ

1. У ході реалізації кваліфікаційної роботи було опрацьовано літературні джерела за темою розробки парсерів, аналізаторів даних та проаналізовано продукти аналогів, які надають перевагу при використанні як звичайними користувачами так і підприємцями букмекерських контор.
2. Аналіз доступних наявних інформаційних матеріалів та результатів наукових досліджень, показав, що найбільша увага розробників та дослідників приділяється ІТ технологіям, зорієнтованим на швидкісне зчитування інформації з сайтів конкурентів, для власної вигоди. Тому ціллю розробки став аналізатор лінії подій букмекерських контор для широкого спектру користувачів як зі сторони звичайних користувачів, так і букмекерських контор.
3. Після аналізу сучасного стану питання було розроблено технічне завдання та виконане планування робіт. Перед початком реалізації було проведено проектування архітектури системи, та виділення варіантів використання. Розроблена система відповідає усім поставленим функціональним вимогам.
4. Спроектований та створений парсер та аналізатор лінії подій спортивних матчів букмекерських контор, який взаємодіє з функціональним телеграм ботом. Крім цього було реалізовано користувацький інтерфейс з можливістю переходу до телеграм боту та активацією аналізатора коефіцієнтів на спортивні матчі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Thomas A. Runkler. “Data Analytics: Models and Algorithms for Intelligent Data Analysis” : Springer Fachmedien, 2020. 540 p.
2. Siegmund Brandt “Data Analysis Statistical and Computational Methods for Scientists and Engineers” : Springer, 2014. 544p.
3. Wes McKinney “Python for data analysis” : O`REILLY, 2015. 470p.
4. Grune D. Parsing Techniques — A Practical Guide / D.Grune, C. Jacobs — Chichester : Originally published by Ellis Horwood, 1990. 320p.
5. Головна документація до фреймворку Django : веб-сайт. URL: <https://www.djangoproject.com/>.
6. Головна документація до фреймворку ASP.NET : веб-сайт. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet>.
7. Головна документація до фреймворку Laravel : веб-сайт. URL: <https://laravel.com/>.
8. Документація до Telegram Bot API : веб-сайт. URL: <https://core.telegram.org/bots/api>.
9. Документація до BotFather : веб-сайт. URL: <https://botfather.io/docs/>.
10. Документація до Telegraf : веб-сайт. URL: <https://telegraf.js.org/>.
11. Документація до бібліотеки BeautifulSoup : веб-сайт. URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.
12. Документація до бібліотеки Puppeteer : веб-сайт. URL: <https://pptr.dev/>.
13. Документація до бібліотеки Guzzle : веб-сайт. URL: <https://docs.guzzlephp.org/en/stable/>.
14. Сторінка компанії Sportradar : веб-сайт. URL: <https://sportradar.com/>.

15. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. посіб. Харків : ХНУМГ ім. О. М. Бекетова, 2017. 93 с.

16. Степаненко О.І., студент 4 курсу ІННІ ім. Ю.М. Потебні ЗНУ. Наук. кер.: к.т.н., доц. Михайлуца О.М. «Використання фреймворку Django для розробки автоматичного аналізатора лінії подій та результатів букмекерських контор». Збірник наукових праць студентів, аспірантів і молодих вчених «Молода наука-2023» : у 5 т. Запорізький національний університет. Запоріжжя: ЗНУ, 2023. Т.5. С. 121-123.

**Декларація**  
**академічної доброчесності**  
**здобувача ступеня вищої освіти ЗНУ**

Я, Степаненко Олег Ігорович, студент 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz19bd-14@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему **“Використання Django REST фреймворку для розробки автоматичного аналізатора лінії подій та результатів букмекерських контор”** відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 20.06.2023 Підпис \_\_\_\_\_ Степаненко Олег Ігорович  
(студент)

Дата 20.06.2023 Підпис \_\_\_\_\_ Михайлуца Олена Миколаївна  
(науковий керівник)