

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Ковальова Артема Андрійовича*  
(ПІБ)

академічної групи *122-20ск-1*  
(шифр)

спеціальності *122 Комп'ютерні науки*  
(код і назва спеціальності)

освітньої програми *Бакалавр*  
(назва освітньої програми)

на тему: *Розробка комп'ютерної технології моніторингу появи нових криптовалют*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Лактіонов І.С.</i>			
<b>розділів:</b>				
спеціальний	<i>проф. Лактіонов І.С.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
<b>Рецензент</b>				
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2023

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем  
(повна назва)

(підпис)

(прізвище, ініціали)

« » 2023 року

**ЗАВДАННЯ**

на кваліфікаційну роботу  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-20ск-1 Ковальов А.А.  
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка комп'ютерної технології  
моніторингу появи нових криптовалют

затверджена наказом ректора НТУ «ДП» від 16.05.2023

№ 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми.	19.06.2023 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	26.06.2023 р.

Завдання видав

(підпис)

проф. Лактіонов І.С.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Ковальов А.А.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 28.06.2023 р.

## РЕФЕРАТ

Пояснювальна записка: 90 с., 29 рис., 3 дод., 24 джерела.

Об'єкт розробки: комп'ютерна система моніторингу появи нових крипто валют та збір інформації щодо них.

Мета кваліфікаційної роботи - розробити комп'ютерну систему для відстеження нових крипто валют та збирання інформації про них. Комп'ютерна система буде розроблена з використанням С# та відповідних бібліотек, яке використовуватиме відкриті джерела для отримання актуальної інформації. Додатковою метою є збір технічних характеристик нових крипто валют та посилань на різноманітні джерела. Інформація може бути використана для аналізу тенденцій на ринку крипто валют, визначення перспективних можливостей для інвестування та інших цілей.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі було проаналізовано наявні рішення, вибрано платформи для розробки, проведено проектування та розробку комп'ютерної системи. Описано роботу комп'ютерної системи, її алгоритм та структуру функціонування, а також розглянуто виклик та завантаження комп'ютерної системи, визначено вхідні та вихідні дані і охарактеризовано склад параметрів технічних засобів.

Актуальність розробки комп'ютерної технології моніторингу появи нових крипто валют визначається зростаючим попитом і зацікавленістю ними та їх використання в різних сферах.

Практична цінність розробки полягає в реалізації комп'ютерного автоматичного можливості комп'ютерного автоматичного моніторингу появи нових крипто валют із агрегуванням інформації про них.

Ключові слова: КРИПТОВАЛЮТА, ПАРСЕР, МОДУЛЬ, СИСТЕМА, АЛГОРИТМ, ДАНІ, ТЕХНОЛОГІЯ.

## ABSTRACT

Explanatory note: 90 p., 29 figs., 3 add., 24 sources.

Object of development: a computer system for monitoring the emergence of new crypto currencies and collecting information.

The purpose of the qualification work is to develop a computer system for tracking new crypto currencies and collecting information about them. The computer system will be developed using C# and relevant libraries, which will use open sources to obtain relevant information. An additional goal is to collect technical specifications of new crypto currencies and links to various sources. The information can be used to analyse trends in the cryptocurrency market, identify promising investment opportunities, and other purposes.

In the introduction, the analysis and current state of the problem is considered, the purpose of the qualification work and the field of its application are specified, the justification of the relevance of the topic is given, and the statement of the task is clarified.

In the first section, the subject area is analysed, the relevance of the task and the purpose of the development is determined, the task statement is formulated, and the requirements for software implementation, technologies and software tools are specified.

In the second section, existing solutions were analysed, development platforms were selected, and the computer system was designed and developed. The operation of the computer system, its algorithm and structure of functioning are described, as well as the call and loading of the computer system are considered, the input and output data are determined, and the composition of technical means parameters is characterized.

The relevance of the development of computer technology for monitoring the emergence of new crypto currencies is determined by the growing interest in crypto currencies and their use in various fields.

The practical value of the development lies in the implementation of the computer automatic possibility of computer automatic monitoring of the appearance of new crypto currencies with the aggregation of information about them.

Keywords: CRYPTOCURRENCY, PARSER, MODULE, SYSTEM, ALGORITHM, DATA, TECHNOLOGY.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

КС – комп'ютерна система;

ОС – операційна система.

## ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ....	10
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування.....	15
1.3. Підстава для розробки.....	16
1.4. Постановка завдання.....	16
1.5. Вимоги до програми або програмного виробу.....	17
1.5.1. Вимоги до функціональних характеристик.....	17
1.5.2. Вимоги до інформаційної безпеки.....	17
1.5.3. Вимоги до складу та параметрів технічних засобів.....	18
1.5.4. Вимоги до інформаційної та програмної сумісності.....	18
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	20
2.1. Функціональне призначення системи.....	20
2.2. Опис застосованих математичних методів.....	21
2.3. Опис використаних технологій та мов програмування.....	21
2.3.1. Опис мови програмування C#.....	21
2.3.2. Опис технології ASP.NET.....	23
2.3.3. Опис технології EntityFramework.....	24
2.3.4. Опис технології MVC.....	25
2.3.5. Опис технології Roslyn Api.....	27
2.3.6. Опис формату даних JSON.....	28
2.3.7. Опис технології Microsoft SQL Server.....	28
2.3.8. Опис мови гіпертекстової розмітки HTML.....	30
2.3.9. Опис мови стилів CSS.....	31

2.3.10.	Опис мови програмування JavaScript.....	32
2.3.11.	Опис розподіленої системи керування версіями Git.....	33
2.4.	Опис структури системи та алгоритмів її функціонування.....	35
2.4.1.	Структура системи.....	35
2.4.2.	Алгоритми та функціонування системи.....	36
2.5.	Обґрунтування та організація вхідних та вихідних даних програми	38
2.6.	Опис розробленої системи.....	39
2.6.1.	Використані технічні засоби.....	39
2.6.2.	Використані програмні засоби.....	40
2.6.3.	Виклик та завантаження програми.....	41
2.6.4.	Опис інтерфейсу користувач.....	42
2.6.5.	Опис консольного інтерфейсу.....	46
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....		53
3.1.	Розрахунок трудомісткості та вартості розробки програмного продукту.....	53
3.2.	Розрахунок витрат на створення програми.....	56
ВИСНОВКИ.....		59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		61
Додаток А. Код програми.....		64
Додаток Б. Відгук керівника економічного розділу.....		89
Додаток В. Перелік файлів на диску.....		90

## ВСТУП

Крипто валюти - це електронні активи, що є аналогом традиційних валют і можуть використовуватися як засіб обміну та зберігання вартості. За останні кілька років крипто валюти стали все більш популярними, що призвело до появи нових проектів, які надають нові крипто валюти. Зростання популярності крипто валют призвело до необхідності моніторингу нових крипто валют та їх відслідковування для забезпечення фінансової безпеки та захисту від шахрайства.

Мета цієї кваліфікаційної роботи – розробка комп'ютерної технології моніторингу появи нових крипто валют, що буде забезпечувати безпеку та захист фінансових інтересів в сфері крипто валютного ринку. Галузь застосування цієї технології - фінансовий сектор, де нові крипто валюти з'являються досить часто, що може створювати загрози безпеці та стабільності цього ринку. Таким чином, розробка комп'ютеризованої системи моніторингу, яка буде здатна відслідковувати нові крипто валюти та надавати інформацію про них, є дуже важливою задачею для фінансової сфери. Крім того, така технологія може бути корисною для інвесторів.

Актуальність теми обумовлена швидким розвитком технологій та збільшенням популярності крипто валют. За останні роки кількість нових крипто валют значно збільшилась, що призвело до складнішої ситуації з точки зору безпеки та стабільності ринку крипто валют. Недостатня інформація про нові проекти може призвести до шахрайства та втрати коштів, що підкреслює необхідність розробки технології моніторингу.

Конкретна постановка завдання полягає у розробці комп'ютерної системи для комп'ютеризованого моніторингу появи нових крипто валют та отримання інформації про їх ключові характеристики. Забезпечення доступу до цієї інформації для фінансових установ та інших зацікавлених сторін є також однією з мет цього проекту.



Основна ідея полягає в тому, щоб створити комп'ютерну систему, яка автоматично перевірятиме різні джерела в Інтернеті для виявлення нових крипто валют, що з'являються на ринку. Після виявлення нової крипто валюти, комп'ютерна система буде аналізувати ключові характеристики цієї валюти. Інформація про нові крипто валюти буде зберігатися в базі даних, яка буде оновлюватися в режимі реального часу.

Для досягнення цієї мети будуть використовуватися різні технології для комп'ютеризованого аналізу даних про крипто валюти.

Очікується, що розробка цієї технології допоможе фінансовим установам та іншим зацікавленим сторонам ефективно відслідковувати нові крипто валюти та забезпечити фінансову безпеку. Крім того, така технологія може бути корисною для інвесторів, які шукають нові можливості для інвестування в крипто валюти, оскільки вона надасть їм доступ до оновленої та актуальної інформації про ринок.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Загальні відомості з предметної галузі

Крипто валюти – це цифрові активи, які використовуються як засіб обміну, зберігання та передачі вартості. Вони стали популярними у останні роки, особливо після появи біткоїну у 2009 році [1]. Зараз на ринку існує більше тисячі різних крипто валют, кожна з яких має свої особливості та можливості.

Моніторинг появи нових крипто валют є важливою складовою їх ринку, оскільки нові проекти можуть мати великий потенціал для розвитку та інвестицій. Однак, відслідковувати нові проекти на ринку може бути складно через велику кількість різноманітних джерел інформації та швидкий темп змін.

Розробка комп'ютерної технології моніторингу появи нових крипто валют може допомогти вирішити цю проблему. Ця технологія буде використовувати різні джерела інформації, такі як веб-сайти, соціальні медіа та інші джерела, для відстеження нових крипто валют, які з'являються на ринку. Потім дана технологія буде аналізувати ці дані та надавати користувачам повну інформацію про нові крипто валюти та їх характеристики, такі як обсяг випуску, ринкова капіталізація, ціна та інші параметри .

Розробка комп'ютерної технології моніторингу появи нових крипто валют може бути корисною для багатьох галузей, зокрема для фінансових компаній, які займаються торгівлею крипто валютами, а також для дослідників та інвесторів, які бажають вивчати динаміку розвитку крипто валют на ринку. Крім того, ця технологія може бути використана для аналізу трендів на ринку та прогнозування його розвитку, що дозволить користувачам бути в курсі останніх новин та трендів в галузі крипто валют.

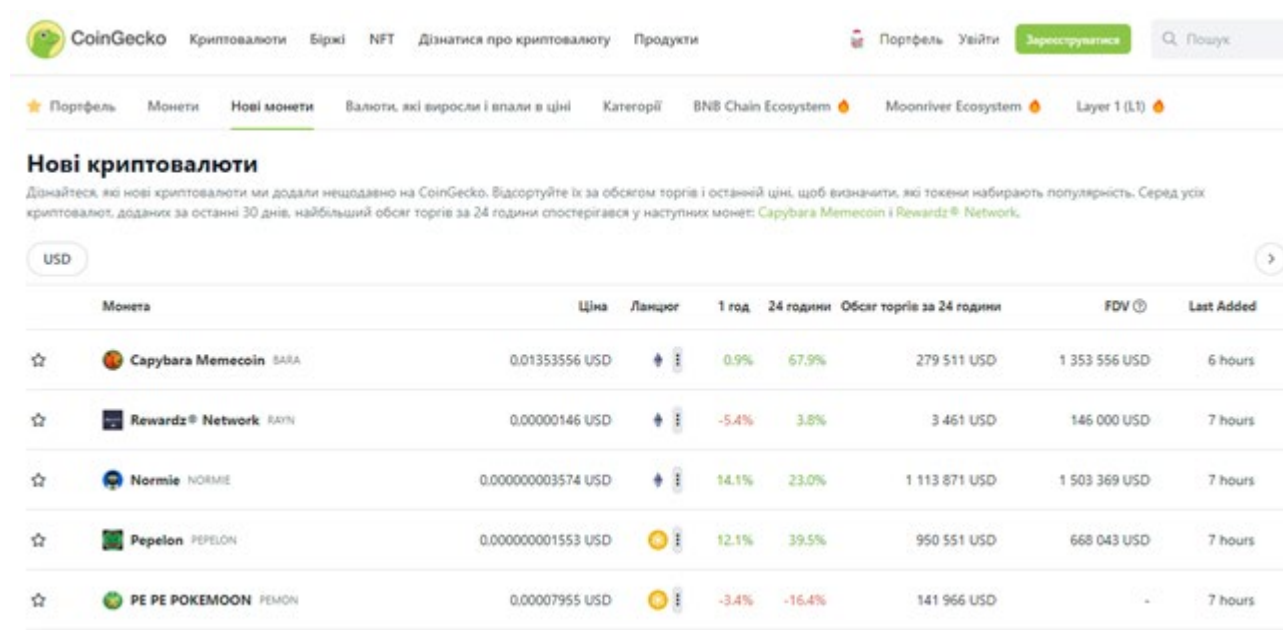
Проте, розробка такої технології може також стикнутися з деякими викликами, зокрема зі збором достовірної та точної інформації з різних джерел та обробкою великої кількості даних. Крім того, ринок крипто валют є досить

непередбачуваним та змінним, що може ускладнити прогнозування розвитку та аналіз ринку.

У цілому, розробка комп'ютерної технології моніторингу появи нових крипто валют може допомогти вирішити важливу проблему в галузі крипто валют та стати корисною для багатьох користувачів, але потребує високої технічної компетентності та врахування специфіки ринку крипто валют.

Аналогічних систем в відкритому доступі не знайдено, але розглянемо дуже наближену до цієї роботи систему. З веб-джерел звідки отримуються дані, є розділ «Нові криптовалюти». Розглянемо функціонал на прикладі декількох веб-ресурсів, а саме: «CoinGecko», «CoinMarketCap» та «CoinPaprika».

Спочатку розглянемо кожну сторінку яка містить нові криптовалюти на кожному веб-ресурсі. Веб-ресурс «CoinGecko» має наступний вигляд сторінки, яка містить нові криптовалюти.



The screenshot shows the 'New Cryptocurrencies' page on CoinGecko. The page title is 'Нові криптовалюти' (New Cryptocurrencies). Below the title, there is a brief description: 'Дізнайтеся, які нові криптовалюти ми додали нещодавно на CoinGecko. Відсортуйте їх за обсягом торгів і останній ціні, щоб визначити, які токени набирають популярність. Серед усіх криптовалют, доданих за останні 30 днів, найбільший обсяг торгів за 24 години спостерігався у наступних монет: Capybara Memecoin і Rewardz® Network.' Below this, there is a table with columns: 'Монета' (Coin), 'Ціна' (Price), 'Ланцюг' (Chain), '1 год' (1 Year), '24 години' (24 Hours), 'Обсяг торгів за 24 години' (24h Volume), 'FDV' (Fully Diluted Valuation), and 'Last Added'. The table lists five cryptocurrencies: Capybara Memecoin (SARA), Rewardz® Network (RAYN), Normie (NORMIE), Pepelon (PEPELON), and PE PE POKEMOON (PEMON).

Монета	Ціна	Ланцюг	1 год	24 години	Обсяг торгів за 24 години	FDV	Last Added
☆ Capybara Memecoin SARA	0.01353556 USD	↑	0.9%	67.9%	279 511 USD	1 353 556 USD	6 hours
☆ Rewardz® Network RAYN	0.00000146 USD	↑	-5.4%	3.8%	3 461 USD	146 000 USD	7 hours
☆ Normie NORMIE	0.00000003574 USD	↑	14.1%	23.0%	1 113 871 USD	1 503 369 USD	7 hours
☆ Pepelon PEPELON	0.00000001553 USD	↑	12.1%	39.5%	950 551 USD	668 043 USD	7 hours
☆ PE PE POKEMOON PEMON	0.00007955 USD	↑	-3.4%	-16.4%	141 966 USD	-	7 hours

Рис. 1.1. Сторінка «Нові криптовалюти» CoinGecko

CoinMarketCap Криптовалюти Біржі Спільнота Продукти Навчайтеся ★ Список Спостереження 👤 Портфель

### Нові Криптовалюти

Нові криптовалюти додані до CoinMarketCap за останні 30 днів

☆ Список Спостереження
👤 Портфель
У Центрі Уваги
Лідери Росту та Лідери Падіння
Популярні
Найбільш Відвідувані
Нещодавно Додані

#	Ім'я	Ціна	1г	24г	Повністю Розбавлена Ринкова Капіталізація	Обсяг	Блокчейн	Додано
1	BrainAI \$BRAIN	UAH0.0...02459	-3.06%	+9.20%	UAH2,459,464	UAH1,592,753	Ethereum	2 годин тому
2	CelebrityCoinV2 CCV2	UAH67.65	-0.42%	+0.37%	UAH4,059,071,486	UAH9,676,856	BNB	4 годин тому
3	MILO Project MILO	UAH8.36	+0.10%	-11.04%	UAH8,360,104,008	UAH2,250,870	Polygon	1 day ago
4	PLAYA3ULL GAMES 3ULL	UAH0.01852	+0.45%	+1.73%	UAH3,389,914	UAH2,729,151	Avalanche	2 днів тому
5	Land Wolf WOLF	UAH0.0...04493	-0.26%	-50.13%	UAH1,890,006	UAH269,017	Ethereum	2 днів тому
6	MARUTARO MARUTARO	UAH0.0...05326	-14.73%	-10.03%	UAH532,575	UAH165,263	Ethereum	2 днів тому
7	KONG KONG	UAH0.0...01172	+6.25%	+23.90%	UAH58,590,336	UAH55,660,127	Ethereum	2 днів тому
8	Boring Coin BORING	UAH0.0...05049	+0.34%	-62.61%	UAH34,836,461	UAH64,123,614	Ethereum	2 днів тому

Рис. 1.2. Сторінка «Нові криптовалюти» CoinMarketCap

coinpaprika Add Coin / Exchange / Update Our Partners English / USD Sign In / Sign Up

API
Coinpaprika API Pro
Market Overview
Currencies <sup>57264</sup>
Portfolio
Exchanges <sup>321</sup>
Transparency
Buy/Sell Crypto
Market Cap: \$1 094 223 644 917 (0.09%)

All News
Halt Operations
Robinhood Drops Support for Three Cryptos
Hong Kong Flings Doors Open for Crypto Exchanges
Binance Nigeria Ordered to Halt Operations
Robinhood Drops Support

Currencies
DeFi
Tokenized Assets
Recently Added
Get rewards for staking 1INCH tokens!

### Top Recently Added based cryptocurrencies

These new cryptocurrencies has been added to coinpaprika within last 30 days.

Favourites
Price Overview
Return rates
Community
All Time High
Filters
USD
Back to top 100

#	Name	Price	1h	24h	7d	Volume (24h)	Liquidity	MarketCap
37	Pulsechain PLS	\$0.00009397	-1.34%	1.4%	-40.65%	\$754 991	\$53.06 K / \$26.56 K	\$1 392 111 970
43	HEX from Pulsechain HEX	\$0.011998	-1.8%	-1.75%	-34.76%	\$3 141 869	\$0 / \$0	\$1 032 886 681
51	PulseX PLSX	\$0.00003715	-0.75%	1.58%	-43.55%	\$8 754 667	\$1.59 K / \$4.36 K	\$782 694 458
63	Wrapped Pulse WPLS	\$0.00009360	-0.74%	-2.02%	-39.63%	\$37 287 796	\$0 / \$0	\$676 732 394
179	Wrapped Beacon ETH WBE...	\$1 752.17	0.25%	0.26%	-8.22%	\$134 586	\$1.25 M / \$67.75 K	\$125 422 026
570	OXBT OXBT	\$0.068064	-7.21%	2.21%	-58.3%	\$129 162	\$0 / \$0	\$13 612 718
634	Orbcity ORB	\$0.067934	-1.76%	-7.84%	-21.99%	\$131 482	\$30.39 K / \$20.90 K	\$9 060 291
735	Verse VER...	\$0.000507	0.22%	-3.35%	-16.22%	\$9 900.33	\$0 / \$0	\$5 063 805
861	Portugal National Team ... POR	\$0.729461	0.46%	0.23%	-12.77%	\$63 800.68	\$14.01 K / \$3.92 K	\$2 263 191

Рис. 1.3. Сторінка «Нові криптовалюти» CoinPaprika

На рисунках 1.1, 1.2 та 1.3 зображені веб-сторінки розділу «Нові криптовалюти» на яких вказано список усіх нещодавно створених

криптовалют, їх поточний курс, його зміни, платформа з якою складено смарт-контракт та час який пройшов з моменту створення [2].

Тепер розглянемо, як ці веб-ресурси відображають детальну інформацію про нову криптовалюту.

**CoinGecko** Криптовалюти Біржі NFT Дізнатися про криптовалюту Продукти Портфель Увійти Зареєструватися Пошук

Криптовалюти > Ціна на Rewardz® Network

Наступний токен має функцію змінюваного податку в смарт-контракті для зміни ставок податку після розгортання. Перевірте самостійно і будьте уважні, якщо ви торгуєте цим токеном.

Rank #N/A

**Rewardz® Network** RAYN

**0,00000146 USD** ▲4.1%

0.00000000 BTC 4.2%↑  
0.00000000 ETH 4.4%↑

Додати у вибране: 7

0.00000140 USD	за 24 години	0.00000159 USD
Ринкова капіталізація		Кількість в обігу
Обсяг торгів за 24 год	3 463 USD	Загальний обсяг пропозиції 68 750 922 192
Повністю розведена оцінка	146 122 USD	Максимальний обсяг пропозиції 100 000 000 000

Огляд Ринки Історичні дані

Загальні відомості Соціальні мережі Розробник Віджети

Рис 1.4. CoinGecko. Детальна інформація

**CoinMarketCap** Криптовалюти Біржі Спільнота Продукти Навчайся Список Спостереження Портфель Шукайте

Криптовалюти > Токени > BrainAI

According to Gopluslabs, the smart contract of the following asset is not renounced and can be modified by the contract owner (for example: disable selling, change fees, mint new tokens, or transfer tokens). Please exercise caution before taking any action and DYOR.

**BrainAI** SBRAIN

Ранг #3770 Токен On 177 watchlists

thebraintoken.com Дослідники Спільнота

Чат

Контракти: Ethereum: 0x6ab0...D1AeFcF

BrainAI Ціна (SBRAIN)

**UAH0.0...04701** ▲108.73%


0.000000000005 BTC ▲107.75%  
0.000000000073 ETH ▲107.93%

Низький: UAH0.0...02253 Високий: UAH0.0...05123

Ринкова Капіталізація	UAH4,701,138 ▲0.05%	Повністю Розбавлена Ринкова Капіталізація	UAH4,701,138 ▲108.73%	Обсяг 24h	UAH2,615,291 ▲0.05%	Self Reported Circulating Supply	1,000,000,000 SBRAIN
24h Обсяг / Ринкова Капіталізація	0.5563	CEX Vol	--	DEX Vol	UAH2,615,291	Максимальна Пропозиція	1,000,000,000,000
						Загальна пропозиція	1,000,000,000,000

Рис. 1.5. CoinMarketCap. Детальна інформація

The SECOTERRA Presale is Live! The Only Recycle-To-Earn Crypto Saving The Planet! Make An Impact Buy Now!








**Pulsechain PLS** Rank 36

★ Add to favourites

Website  
pulsechain.com  
pulse.info

Explorer  
scan.pulsechain.com

Asset Type  
Coin

Similar Coins by MCap  
 Hedera Hashg...  
 Arbitrum  
 Quant  
 Aptos  
 VeChain

**\$0.00009555 (+5.39%)**

1h	24h	Week	Month	Year
+1.45%	+5.39%	-38.48%	+%	+%

0.00000000 BTC (+5.03%) +1.49% +5.03% -35.05% +% +%  
 0.00000005 ETH (+5.15%) +1.51% +5.15% -33.05% +% +%

Operations on Pulsechain provided by Coinpaprika | Are you member of PLS community?

Pulsechain [PLS] is a cryptocurrency with its own blockchain. The most actual price for one Pulsechain [PLS] is \$0.000096. Pulsechain is listed on 3 exchanges with a sum of 6 active markets. The 24h volume of [PLS] is \$700 658, while the Pulsechain market cap is \$1 415 567 461 which ranks it as #36 of all cryptocurrencies. You can find more information about Pulsechain [PLS] on pulsechain.com pulse.info

Recently Added (3) | Suggest a tag

Social (followers): reddit.com (23 414)

Market Cap	Rank 36	All Time High	Circulating Supply	Volume (24h)	Rank 546	Vol / M Cap (24h)
\$1 415 567 461		\$0.000532	14 814 814 814 815	\$700 658		0.05%
54 883.00 BTC		23 May 23	(11.00%)	27.17 BTC		
		% to ATH (456.26%)	Total: 135 000 000 000 000			
			Max: 135 000 000 000 000			

**Calculator**

PLS to USD calculator

0 PLS ↔ 0 USD

Advanced Calculator

Similar Crypto Pairs

- PLS to HBAR
- PLS to ARB
- PLS to QNT
- PLS to APT
- PLS to VET

Similar Fiat Pairs

- PLS to USD
- PLS to EUR
- PLS to JPY
- PLS to PLN
- PLS to GBP

More crypto and FIATs

**Coins Compare**

Compare Pulsechain with other crypto assets.

Рис. 1.6. CoinPaprika. Детальна інформація

На рисунках 1.4, 1.5 та 1.6 ми бачимо детальну інформацію про криптовалюту. На кожній із сторінок вказана назва криптовалюти, її коротка назва (тікер), більш детальна інформація про курс та його зміни, посилання на соціальні мережі, офіційний сайт, репозиторій, технічну документацію та інші. Також на кожному з веб-ресурсів є унікальна інформація якої немає на інших веб-ресурсах.

Розглянемо переваги та недоліки цих систем.

Переваги:

- Необмежений доступ до інформації без обмежень щодо кількості запитів.
- Можливість відображення інформації, недоступної через офіційне API.
- Відображається уся інформація яка доступна на веб-ресурсі

Недоліки:

- Відсутність можливості отримувати сповіщення про нові криптовалюти одночасно з усіх сайтів.

- Незручність відслідковування інформації з кількох веб-ресурсів одночасно: необхідність постійного перемикання між закладками в браузері, оновлення та перевірка появи нових криптовалют (Рис. 1.7.).

- Відсутня гарантія того, що нові криптовалюти будуть з'являтися саме на даному сайті або на інших джерелах.

- На деяких веб-ресурсах відсутній переклад на українську мову що може утруднити роботу з ним.

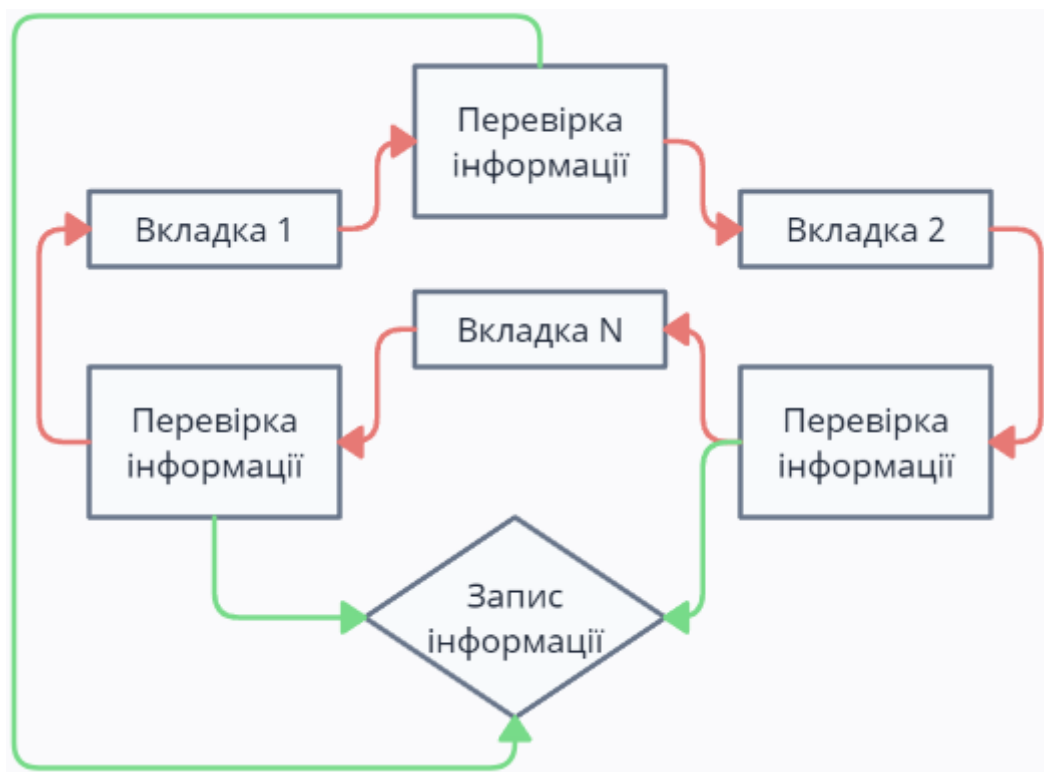


Рис. 1.7. Схема ручного моніторингу появи нових криптовалют

## 1.2 Призначення розробки та галузь застосування

Ця кваліфікаційна робота має на меті розробку комп'ютерної технології, спрямованої на моніторинг появи нових крипто валют та збір інформації про них. Головна мета полягає в розробці програмного забезпечення, яке здатне

автоматично відстежувати та аналізувати нові крипто валюти з використанням відкритих джерел інформації.

Ця технологія має широкий спектр застосувань, зокрема в фінансовому секторі. Завдяки розробленому програмному забезпеченню, фінансові установи, інвестори та трейдери матимуть змогу оперативного отримувати актуальну інформацію про нові крипто валюти, таку як їх характеристики, капіталізація та обсяги торгів. Це дозволить їм проводити більш об'єктивний аналіз, робити інформовані рішення та виявляти перспективні можливості для інвестування.

Безпека та захист фінансових інтересів також є важливими аспектами цієї технології. Завдяки постійному моніторингу, можна виявляти потенційні шахрайські схеми, фінансові злочини та ризикові крипто валюти, що допомагає запобігти втратам і зберегти фінансову безпеку.

Отже, розробка комп'ютерної технології моніторингу нових крипто валют має велике значення для фінансової галузі, сприяючи збереженню стабільності ринку, захисту фінансових інтересів та наданню цінної інформації для інвестиційних рішень.

### **1.3. Підстава для розробки**

В кінці навчання, студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою.

Підставою для розробки кваліфікаційної роботи на тему «Розробка комп'ютерної технології моніторингу появи нових криптовалют» є наказ по Національному технічному університету «Дніпровська політехніка» від 16.05.2023 р. № 350-с

### **1.4. Постановка завдання**

Основною ціллю даної кваліфікаційної роботи є розробка високоефективної комп'ютерної технології, яка надасть користувачам



унікальну можливість систематично відстежувати появу нових криптовалют, а також здійснювати комплексний та всебічний збір важливої інформації, що стосується цих криптовалют.

## **1.5. Вимоги до програми або програмного виробу**

### **1.5.1. Вимоги до функціональних характеристик**

Основні вимоги до функціональних характеристик розроблюваної комп'ютерної технології моніторингу появи нових криптовалют включають:

- відстеження нових криптовалют;
- збір інформації про нові криптовалюти;
- інтеграція з іншими системами.

### **1.5.2. Вимоги до інформаційної безпеки**

Для забезпечення інформаційної безпеки вимоги до віддаленого сервера, на якому буде встановлене програмне забезпечення, включають [3]:

- Використання складного пароля або SSH-ключа для доступу до віддаленого сервера.
- Обмеження доступу зовнішніх мереж до бази даних.
- Використання складного пароля для облікового запису на сервері.
- Неадміністративні повноваження облікового запису бази даних, що використовується програмним забезпеченням.

Щодо доступу до API комп'ютерної системи, певні вимоги до інформаційної безпеки не висуваються, оскільки комп'ютерна система не має кінцевих точок, які можуть впливати на роботу комп'ютерної системи. Планується, що доступ до даних буде публічним і без обмежень. Однак, якщо потрібно обмежити доступ, можна додати API-ключі, які обмежують користувачів у доступі до даних, за винятком тих, які мають дійсний API-ключ.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для повноцінного функціонування програмного забезпечення необхідні:

- персональний комп'ютер або віддалений сервер;
- доступ до інтернет мережі;
- мережеве обладнання;
- біла IP адреса.

Мінімальні вимоги до потужності персонального комп'ютера або віддаленого сервера:

- Процесор: 2 ядра, частота 2 ГГц;
- Оперативна пам'ять – 2 гігабайти DDR4;
- Накопичувач SSD (або NVME) – 55 гігабайт (для ОС на базі Linux) або 70 гігабайт (для ОС Windows);
- Певних вимог до графічних процесорів не висувається оскільки ПЗ не використовує графічних процесор ;
- Доступ до інтернет мережі зі швидкістю 20Мбіт/сек.

Залежно від кількості джерел даних, вимоги можуть змінюватися. Пропонується розраховувати на приблизно 1 ядро процесора на кожні 2 джерела даних, додатково 0,4 гігабайта оперативної пам'яті та швидше інтернет-з'єднання на 10 Мбіт/сек.

### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Розроблювана Комп'ютерна система сумісна з наступними операційними системами:

Windows x64:

- Windows 10 TH1 1507 та вище;
- Windows Server 2016 та вище [4].

Linux x64 [5, 6]:

- Red Hat Enterprise 7.3 та вище;
- Ubuntu 16.04 та вище;
- SUSE Linux Enterprise Server v12 та вище.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 2.1. Функціональне призначення системи

Комп'ютерна технологія для моніторингу появи нових криптовалют та збору інформації має на меті надати користувачам засоби для ефективного відстеження і аналізу ринку криптовалют. Основним функціональним призначенням системи є пошук, відстеження та моніторинг нових криптовалют. Система аналізує різні криптовалютні біржі, щоб виявляти нові проекти і валюти, що з'являються на ринку. Поряд з пошуком нових криптовалют, система здатна збирати детальну інформацію про кожен з них.

Експлуатаційне призначення системи полягає у полегшенні процесу моніторингу криптовалют та наданні користувачам інформаційних засобів для прийняття обґрунтованих рішень. Вона дозволяє користувачу ефективно скорочувати час і зусилля, які раніше потребувалися для пошуку та аналізу нових криптовалют. За допомогою цієї системи можливе автоматизоване відстеження ринку та отримання оперативної інформації. Крім того, вона надає користувачеві можливість встановлювати власні параметри і налаштування для пошуку та моніторингу.

Завдяки використанню передових алгоритмів та штучного інтелекту, система здатна проводити аналіз ринкових тенденцій, прогнозування цін, а також оцінювати потенціал нових криптовалютних проектів. Зібрана інформація допомагає користувачам зробити обґрунтовані інвестиційні рішення та підвищити їх шанси на успіх у криптовалютному ринку. Застосування комп'ютерної технології для моніторингу криптовалют відкриває нові можливості для інвесторів та трейдерів, допомагаючи їм бути постійно в курсі останніх тенденцій і зробити обґрунтовані рішення на основі доступної інформації.

## **2.2. Опис застосованих математичних методів**

Під час розробки комп'ютерної технології в цій кваліфікаційній роботі не математичні методи не застосовувались.

## **2.3. Опис використаних технологій та мов програмування**

При розробці комп'ютерної технології для моніторингу появи нових криптовалют та збору інформації були використані різноманітні технології та мови програмування, що забезпечують функціональність, швидкість та надійність системи. Вибір цих технологій і мов програмування обґрунтовано їх широким спектром можливостей, доброю сумісністю між ними та їхньою популярністю серед розробників. Вони надають зручні інструменти для створення функціональності, веб-інтерфейсу та взаємодії з базою даних для комп'ютерної системи для моніторингу появи нових криптовалют та збору інформації.

### **2.3.1 Опис мови програмування C#**

C# (вимовляється як "сі шарп") – це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Вона базується на попередній мові C і має ряд поліпшень і додаткових можливостей. C# є однією з найпопулярніших мов для розробки програмного забезпечення на платформі .NET.

Одна з головних особливостей мови C# - це її синтаксис, який має структурований і зрозумілий вигляд. Синтаксис C# нагадує синтаксис мови C/C++, але має деякі додаткові конструкції і можливості. Це дозволяє розробникам легко вчитися та розуміти мову C#, особливо тим, хто вже знайомий з іншими мовами C-подібного сімейства.

C# підтримує об'єктно-орієнтовану парадигму програмування, що означає, що програми створюються з використанням класів і об'єктів. Це дозволяє створювати модульні, гнучкі і легко зрозумілі програми. Класи в C# містять поля (змінні) і методи (функції), які можуть виконувати дії над цими полями. Об'єкти - це конкретні екземпляри класів.

Окрім того, C# підтримує інші парадигми програмування, такі як події і делегати для реалізації реактивного програмування, асинхронні операції для роботи з паралельними задачами, лямбда-вирази для створення функцій "на льоту" і багато іншого.

C# має велику стандартну бібліотеку класів (Class Library), яка надає розширений функціонал для роботи з різними аспектами програмування, такими як робота з файлами, мережевими протоколами, базами даних, графічним інтерфейсом користувача та іншими. Це дозволяє розробникам швидко створювати потужні програми без необхідності писати код з нуля для кожного функціонального блоку.

Окрім платформи .NET, C# також підтримує різні інші платформи, такі як Mono і Xamarin, що розширюють його застосування на інші операційні системи, такі як macOS і Linux, а також на мобільні платформи, такі як iOS і Android.

Мова C# відома своєю надійністю і безпекою. Вона надає механізми для контролю типів, обробки винятків і управління пам'яттю, що допомагає уникнути багатьох помилок програмування і забезпечує стабільну роботу програм. Крім того, C# підтримує використання збирача сміття, який автоматично вивільняє пам'ять, використовану непотрібними об'єктами.

Узагальнюючи, C# - це потужна мова програмування, яка поєднує в собі зрозумілий синтаксис, об'єктно-орієнтовану парадигму, багатий функціонал і надійність. Вона широко використовується для розробки різноманітних програмних продуктів, включаючи десктопні додатки, веб-сайти, мобільні додатки, сервіси Інтернету речей та інші [7–10].

## 2.3.2 Опис технології ASP.NET

ASP.NET (Active Server Pages .NET) є фреймворком, розробленим компанією Microsoft, який використовується для створення веб-додатків та веб-сайтів. Він базується на платформі .NET Framework і надає розробникам широкі можливості для побудови потужних, масштабованих і безпечних додатків.

ASP.NET використовує мови програмування, такі як C# або Visual Basic, для реалізації логіки додатку. Цей фреймворк підтримує різні підходи до розробки, включаючи Web Forms, MVC (Model-View-Controller), Web API та SignalR. Кожен з цих підходів надає розробникам різні інструменти та методи для створення веб-додатків залежно від їх потреб і вимог.

ASP.NET пропонує широкі можливості для роботи з базами даних, зокрема використання технології Entity Framework для доступу до даних. Це спрощує взаємодію з базами даних і дозволяє розробникам швидко створювати складні структури даних та виконувати операції з ними. Загальна архітектура ASP.NET зображена на рис 2.1.

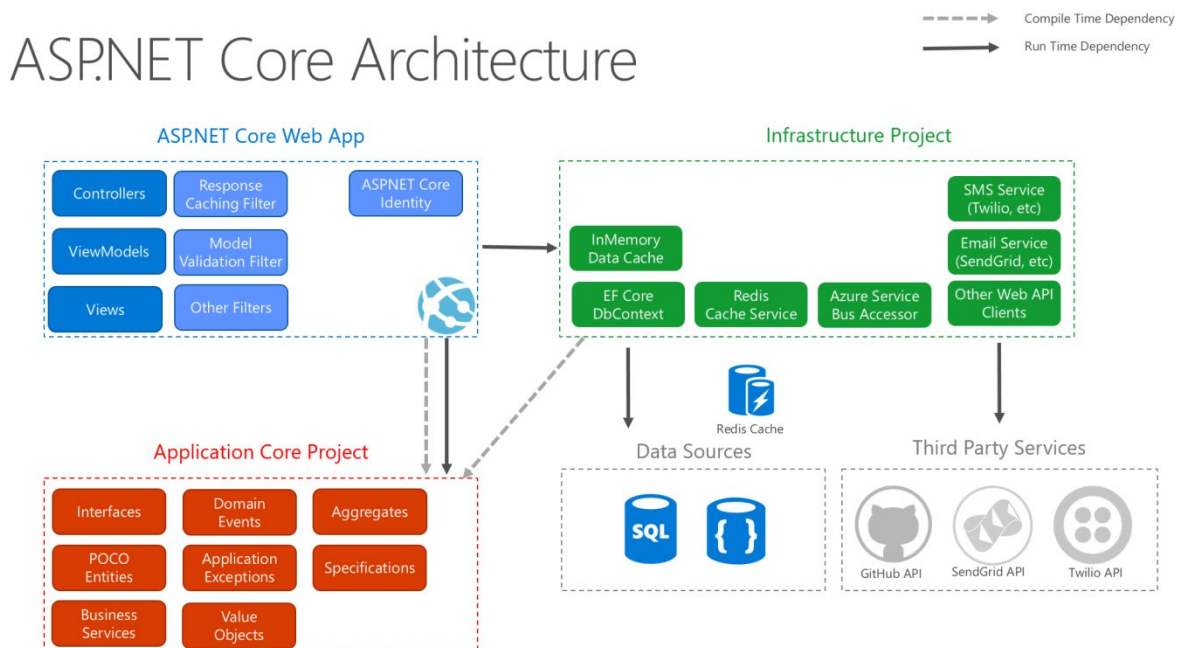


Рис. 2.1. Архітектура ASP.NET Core [11]

Один з ключових аспектів ASP.NET – це його здатність до багато поточності та масштабованості. Фреймворк дозволяє розробникам легко розподіляти завдання на різні потоки, що дозволяє досягати кращої продуктивності та ефективності додатків. Крім того, в ASP.NET є вбудовані механізми кешування та оптимізації продуктивності, які допомагають зменшити навантаження на сервер та підвищити швидкість роботи додатків.

ASP.NET також має вбудовану підтримку безпеки, що робить його відмінним вибором для розробки захищених веб-додатків. Він надає засоби для аутентифікації та авторизації користувачів, контролю доступу до ресурсів, захисту від атак, таких як перехресний сайтівий скриптинг (XSS) і внедрення SQL-команд (SQL injection), а також шифрування даних.

Один зі значних переваг ASP.NET - це його інтеграція з іншими продуктами Microsoft, такими як Microsoft Azure, Visual Studio та SQL Server. Це дозволяє розробникам ефективно використовувати інші інструменти та сервіси компанії Microsoft для побудови повноцінних додатків.

Узагалі, ASP.NET - це потужний фреймворк для розробки веб-додатків, який надає широкі можливості для реалізації функціональності, забезпечує високу продуктивність, безпеку та масштабованість. Його гнучкість і інструменти роботи роблять його популярним вибором серед розробників для створення веб-додатків різного роду. [12, 13]

### **2.3.3 Опис технології EntityFramework**

Entity Framework (EF) - це об'єктно-орієнтований маппер реляційних даних для платформи .NET. Він надає зручний спосіб роботи з базами даних через концепцію "об'єктно-орієнтованого доступу до даних" (Object-Relational Mapping, ORM).

EF дозволяє розробникам працювати з базами даних за допомогою об'єктів, що спрощує взаємодію з даними і робить процес розробки більш продуктивним. Замість прямого написання SQL-запитів, розробнику потрібно



визначити модель даних у вигляді класів C# (або іншої підтримуваної мови програмування) і використовувати EF для взаємодії з базою даних.

Одним з головних компонентів EF є DbContext, який представляє сесію з базою даних і надає можливість виконувати операції збереження, оновлення, видалення та вибірки даних. DbContext визначає набір DbSet, які відображають таблиці бази даних і дозволяють взаємодіяти з ними як з колекціями об'єктів. При цьому EF автоматично створює SQL-запити для виконання операцій над даними.

EF підтримує різні підходи до моделювання даних, включаючи Code First, Database First та Model First. З Code First, розробник визначає модель даних у вигляді класів і EF автоматично створює базу даних за цією моделлю. З Database First, EF використовує наявну базу даних і генерує модель класів на основі схеми бази даних. З Model First, розробник визначає модель даних у вигляді діаграми і EF створює відповідну базу даних та класи.

EF також підтримує міграції бази даних, що дозволяє зберігати історію змін схеми бази даних і автоматично вносити необхідні зміни при релізі нових версій програмного забезпечення. Це робить процес розгортання та оновлення програми більш простим і зменшує ризик втрати даних.

Entity Framework є потужним інструментом для роботи з базами даних у середовищі .NET. Він дозволяє розробникам працювати на вищому рівні абстракції, уникати написання багатої SQL-логіки і зосередитися на логіці програми. Завдяки своїм функціональним можливостям та підтримці різних підходів до моделювання даних, EF став популярним інструментом для розробки програмного забезпечення на платформі .NET. [13]

### **2.3.4 Опис технології MVC**

Модель-представлення-контролер (Model-View-Controller, MVC) є архітектурним шаблоном програмного забезпечення, який використовується для організації та розподілу обов'язків у веб-додатках та інших програмах.

MVC розділяє програму на три основні компоненти: модель, представлення та контролер. Кожен з цих компонентів виконує свої специфічні функції і взаємодіє з іншими компонентами для досягнення бажаної функціональності.

Модель відповідає за представлення даних та бізнес-логіку програми. Вона зберігає дані, які використовуються програмою, та надає методи для їх обробки. Модель може включати в себе структури даних, бази даних, алгоритми обробки даних та інші компоненти, які забезпечують функціональність програми.

Представлення відповідає за відображення даних моделі користувачеві. Воно включає в себе графічний інтерфейс користувача, який дозволяє взаємодіяти з програмою, а також інші елементи, такі як текстові повідомлення, зображення та інші медіа. Представлення може відображати дані з моделі, а також передавати введення користувача до контролера.

Контролер відповідає за обробку введення користувача та керування виконанням програми. Він отримує введення від користувача через представлення, а потім виконує необхідні дії для обробки цього введення. Контролер також може звертатися до моделі для отримання даних або оновлення їх стану. Після обробки введення контролер може оновити представлення для відображення нових даних або результатів.

Ця архітектура дозволяє розділити логіку програми, інтерфейс користувача та обробку даних на окремі компоненти, що полегшує розробку, тестування та підтримку програмного забезпечення. Кожен компонент може бути розроблений незалежно від інших, що забезпечує більшу гнучкість та перевикористання коду.

Модель-представлення-контролер (MVC) є широко використовуваним шаблоном у веб-розробці та розробці програмного забезпечення загалом, оскільки він забезпечує ефективне управління складністю та розділенням відповідальностей між компонентами програми [13].

### 2.3.5 Опис технології Roslyn Api

Roslyn API - це набір програмних інтерфейсів (API) для роботи з кодом на мовах програмування C# і Visual Basic .NET. Розроблений компанією Microsoft, Roslyn API надає розширені можливості для аналізу, трансформації та генерації програмного коду.

Ця бібліотека надає доступ до синтаксичного аналізу, семантичного аналізу, моделі символів та виконання коду. Вона дозволяє розробникам отримувати детальну інформацію про структуру програми, включаючи класи, методи, змінні, параметри, властивості та інші елементи коду.

Основні компоненти Roslyn API включають такі поняття, як синтаксичне дерево, семантичний аналізатор, символи та символні таблиці. Синтаксичне дерево представляє собою ієрархічну структуру, яка відображає синтаксис програми. Воно дозволяє виконувати операції над кодом, такі як вставка, видалення або заміна фрагментів коду.

Семантичний аналізатор дозволяє визначати типи, вирішувати імена, перевіряти правила типізації та здійснювати інші види аналізу на основі контексту програми. Він забезпечує більш високорівневий доступ до інформації про код, що допомагає при рефакторингу, автоматичному завершенні коду та іншому інструментарії розробника.

Символи представляють семантичні об'єкти в програмі, такі як класи, методи, змінні тощо. Розробники можуть отримувати доступ до властивостей символів, таких як типи, модифікатори доступу, параметри, атрибути та інші. Символьні таблиці допомагають здійснювати пошук та взаємодію зі символами в програмі.

Roslyn API може бути використаний для різноманітних сценаріїв, включаючи рефакторинг коду, перевірку якості коду, генерацію автоматичної документації, створення інструментів розробника та багато іншого. Завдяки гнучкості і потужності Roslyn API розробники можуть створювати розширення

та інструменти, які полегшують процес розробки програмного забезпечення на мовах C# і Visual Basic .NET.

Усі ці компоненти дозволяють розробникам аналізувати, змінювати та генерувати код на рівні дерева синтаксису і символів, надаючи багато можливостей для автоматизації та розширення функціональності редакторів коду та інших інструментів розробника. Roslyn API є важливим інструментом для розробників .NET, які працюють з мовами C# і Visual Basic .NET [14].

### **2.3.6 Опис формату даних JSON**

JSON (JavaScript Object Notation) - це стандартний формат обміну даними, який використовується для передачі структурованих інформаційних об'єктів між комп'ютерними системами. JSON заснований на підмножині мови JavaScript і є текстовим форматом, що його легко читати і записувати як людиною, а також обробляти машинами.

Використання JSON дозволяє ефективно обробляти, передавати та зберігати дані, пов'язані з криптовалютами, такі як ціни, обсяги торгів, символічні дані тощо [15].

### **2.3.7 Опис технології Microsoft SQL Server**

Microsoft SQL Server є однією з провідних систем управління базами даних (СУБД), розробленою і підтримуваною компанією Microsoft. Ця програмна система надає потужні та надійні можливості для зберігання, обробки та управління даними у різних сценаріях, від невеликих підприємств до великих корпорацій.

Одна з головних характеристик Microsoft SQL Server - його реляційна модель даних, що дозволяє організувати інформацію у вигляді таблиць зі стовпцями та рядками. Вся база даних розбивається на окремі таблиці, які

взаємодіють між собою за допомогою зв'язків. Кожна таблиця складається зі стовпців, що визначають типи даних, такі як числа, рядки, дати тощо.

SQL Server надає багатий набір функціональних можливостей. Він підтримує мову запитів SQL (Structured Query Language), яка використовується для створення запитів до бази даних. Запити можуть бути виконані для отримання, вставки, оновлення або видалення даних з таблиць. SQL Server також дозволяє створювати збережені процедури, тригери, функції та інші об'єкти бази даних, що допомагають у виконанні складних операцій та забезпечують цілісність даних.

Одним з ключових аспектів SQL Server є його можливості забезпечення безпеки даних. Він підтримує різні рівні доступу до бази даних, включаючи рівні користувачів, ролі та дозволи. Це дозволяє адміністраторам баз даних контролювати, які користувачі мають доступ до яких даних та які операції вони можуть виконувати.

SQL Server також має вбудовану підтримку для резервного копіювання та відновлення даних, що дозволяє створювати резервні копії бази даних і відновлювати їх в разі потреби. Це забезпечує захист від втрати даних внаслідок несправності апаратного забезпечення, помилок користувачів або природних катастроф.

SQL Server може працювати в різних режимах, включаючи одиночний сервер, розподілену систему та кластери. Він підтримує високу доступність та масштабованість, дозволяючи розгортати бази даних на декількох серверах та розподіляти навантаження між ними.

SQL Server також надає інструменти для розробки та адміністрування баз даних. Це включає графічні інтерфейси для створення та управління об'єктами бази даних, а також скриптові мови, такі як Transact-SQL, для автоматизації завдань та виконання складних операцій.

Загалом, Microsoft SQL Server - це потужна та надійна система управління базами даних, яка надає широкий набір функцій для ефективного зберігання, обробки та управління даними. Він широко використовується в багатьох

сферах, включаючи бізнес, науку, громадські служби та інші галузі, де важливо мати доступ до потужних та надійних інструментів для роботи з даними [16].

### 2.3.8 Опис мови гіпертекстової розмітки HTML

HTML (HyperText Markup Language) - це стандартна мова розмітки, використовувана для створення веб-сторінок. Вона визначає структуру та семантику веб-сторінки за допомогою тегів і атрибутів. HTML-код складається з різних елементів, які описують зміст сторінки, включаючи текст, зображення, посилання, таблиці, форми і багато іншого.

HTML-файл починається з тегу `<!DOCTYPE html>`, який вказує на версію HTML і тип документа. Нижче цього тегу розташовується відкриваючий тег `<html>`, який вказує початок HTML-документа, і закриваючий тег `</html>`, який позначає кінець документа.

Усередині тега `<html>` розташовуються два основні блоки: `<head>` і `<body>`. Тег `<head>` використовується для опису документа, включаючи заголовки сторінки, підключення зовнішніх файлів CSS і JavaScript, метадані і багато іншого. Тег `<body>` містить основний зміст сторінки, такий як текст, зображення, посилання і інші елементи.

Елементи HTML представляються за допомогою тегів. Кожен тег починається з відкриваючого символу `<` і закінчується закриваючим символом `>`. Наприклад, тег `<p>` використовується для відображення абзацу тексту, а тег `<img>` використовується для вставки зображення.

Теги можуть мати атрибути, які надають додаткову інформацію про елемент. Атрибути зазвичай знаходяться всередині відкриваючого тегу і мають значення, заключене у лапки. Наприклад, атрибут `src` використовується для вказівки джерела зображення в тезі `<img>`, а атрибут `href` використовується для вказівки посилання в тегу `<a>`.

HTML також підтримує створення списків, таблиць, форм, вкладених елементів і багато іншого. Завдяки широкому набору доступних тегів та їх

комбінацій, HTML надає розробникам гнучкість у створенні різноманітних веб-сторінок з багатим функціоналом.

В цьому описі HTML наведені основні концепції та елементи мови, що дозволяють створювати веб-сторінки. HTML дозволяє використовувати ці елементи разом з CSS (Cascading Style Sheets) і JavaScript для створення багатофункціональних інтерактивних веб-сторінок [17–20].

### **2.3.9 Опис мови стилів CSS**

CSS (Cascading Style Sheets) - це мова, що використовується для опису зовнішнього вигляду веб-сторінок. Вона використовується для встановлення стилів і форматування елементів HTML, надаючи можливість контролювати кольори, шрифти, розміри, розташування та інші аспекти відображення вмісту.

CSS використовує селектори для вибору елементів HTML, до яких будуть застосовуватися стилі. Селектори можуть вибирати елементи за їх тегом, класом, ідентифікатором або іншими атрибутами.

Одним з основних понять CSS є властивості (properties) і значення (values). Властивості визначають аспекти візуального вигляду елементів, такі як колір тексту (color), розмір шрифту (font-size), відступи (margin), рамки (border) тощо. Значення встановлюють конкретні параметри для властивостей, наприклад, "червоний" для властивості color або "12px" для властивості font-size.

CSS також надає можливості групування і вкладеності правил. Групування дозволяє застосовувати однакові стилі до кількох селекторів одночасно. Вкладені правила дають можливість зазначити стилі для елементів, що знаходяться всередині інших елементів.

CSS також підтримує різні одиниці виміру, такі як пікселі (px), відсотки (%), ем (em) та інші, які можна використовувати для задання розмірів і відступів.

Застосування CSS дозволяє розділити структуру та вміст веб-сторінки від її вигляду, що спрощує редагування та зміну стилів без потреби змінювати сам HTML-код. Це робить CSS потужним інструментом для розробки веб-дизайну, який забезпечує гнучкість і контроль над зовнішнім виглядом веб-сторінок [19, 20].

### **2.3.10 Опис мови програмування JavaScript**

JavaScript є високорівневою, інтерпретованою мовою програмування, що зазвичай використовується для створення динамічного контенту на веб-сторінках. Він був розроблений компанією Netscape та спочатку називався LiveScript, але пізніше був перейменований в JavaScript. JavaScript є однією з найпоширеніших мов програмування веб-розробки і використовується як на стороні клієнта (у браузері), так і на стороні сервера (за допомогою платформи Node.js).

JavaScript є скриптовою мовою, що означає, що виконання коду відбувається по одному рядку в кожен момент часу. Він не потребує попередньої компіляції і може бути виконаний безпосередньо в середовищі браузера. Це робить його вельми зручним для розробки інтерактивних веб-сторінок, які реагують на дії користувача.

JavaScript є мовою з подієво-орієнтованим підходом. Це означає, що ви можете визначити функції, які будуть викликатися відповідно до певних подій. Наприклад, ви можете створити функцію, яка виконується, коли користувач клацає на кнопку або наводить курсор миші на певний елемент сторінки.

Однією з особливостей JavaScript є його здатність маніпулювати елементами сторінки HTML та змінювати їх вміст або властивості. За допомогою JavaScript ви можете динамічно змінювати текст, стилізацію, розміщення елементів на сторінці та багато іншого.

Мова також має велику кількість вбудованих функцій та об'єктів, які полегшують роботу з масивами, рядками, датами та іншими типами даних.



Вона підтримує об'єктно-орієнтовану парадигму програмування, що дозволяє створювати класи та об'єкти з власними методами та властивостями.

JavaScript також має можливість взаємодіяти з сервером за допомогою технології AJAX (асинхронного JavaScript і XML). Це дозволяє веб-сторінкам взаємодіяти з сервером у фоновому режимі без перезавантаження сторінки, що дозволяє створювати більш потужні та динамічні веб-додатки.

JavaScript підтримує модульну архітектуру, що дозволяє розбивати програму на окремі модулі та використовувати їх у інших частинах програми. Це полегшує підтримку та розвиток великих проектів.

Однак, як і в усіх мовах програмування, JavaScript має свої обмеження та певні проблеми, такі як проблеми з безпекою, розбіжності в реалізації між різними браузерами та інші. Проте, завдяки постійному розвитку та підтримці великою спільнотою розробників, JavaScript стає все потужнішою мовою програмування для веб-розробки [19, 20].

### **2.3.11 Опис розподіленої системи керування версіями Git**

GIT - це розподілена система керування версіями, призначена для ефективного відстеження змін в програмному коді та спільної роботи над проектами. Її основна функція - зберігання, керування та злиття версій файлів, що використовуються в розробці програмного забезпечення.

У GIT кожен проект зберігається в репозиторії, який може бути локальним або розташовуватися на віддаленому сервері. Репозиторій зберігає всі файли, що належать до проекту, а також повну історію змін цих файлів. Кожна зміна в репозиторії записується як коміт (commit) - невеликий пакет змін, який містить інформацію про змінений файл, автора, дату і повідомлення, що описує зміни.

GIT дозволяє розгалужувати розробку, що означає, що кожен учасник команди може мати власну гілку (branch), в якій він може працювати над функціоналом чи виправленням помилок, не впливаючи на основну гілку

розробки. Після завершення роботи над функцією або виправленням можна злити гілку з основною гілкою, об'єднавши зміни.

Загальна архітектура GIT зображена на рис 2.2.

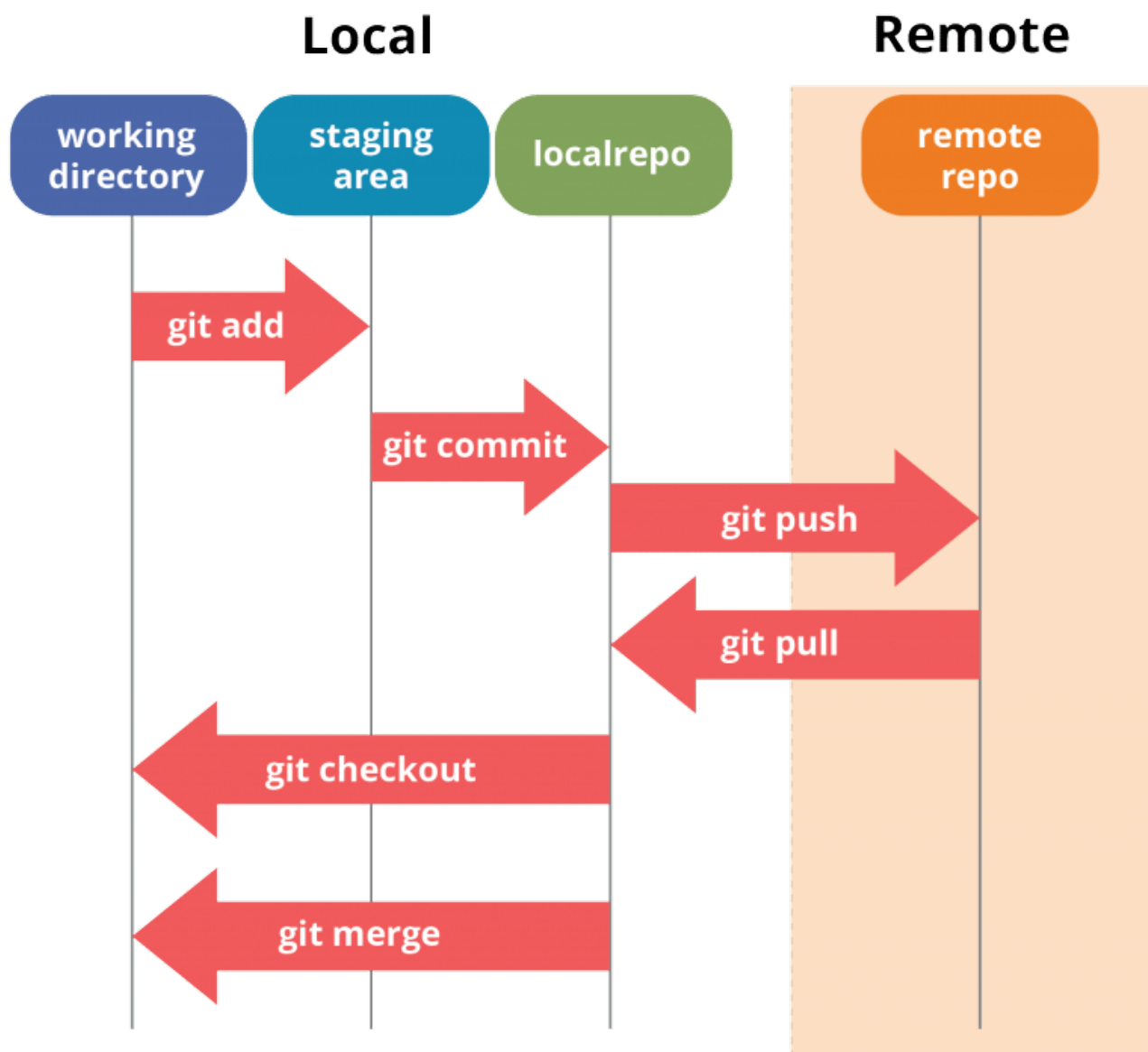


Рис. 2.2. Архітектура Git [21]

GIT надає різноманітні команди та операції для роботи з репозиторієм. Наприклад, команда "git add" додає змінені файли до індексу, команда "git commit" реєструє зміни, команда "git push" відправляє коміти на віддалений сервер, а команда "git pull" оновлює локальний репозиторій з віддаленого.

Крім базових операцій, GIT також має можливості для розв'язання конфліктів злиття, відновлення втрачених даних, стеження за історією змін, аналізу коду та багато іншого. Ця система дозволяє ефективно співпрацювати в команді, відстежувати зміни в коді та керувати версіями проекту.

У підсумку, GIT є потужною інструментальною системою, яка забезпечує збереження і керування версіями файлів, ефективне спільне редагування та співпрацю над проектами розробки програмного забезпечення [22].

## **2.4. Опис структури системи та алгоритмів її функціонування**

Комп'ютерна технологія для моніторингу появи нових криптовалют та збору інформації має наступну структуру та алгоритми функціонування, які забезпечують його ефективність та продуктивність.

### **2.4.1 Структура системи**

Комп'ютерна технологія складається з наступних складових частин:

- Модуль збору, збереження та надання даних: Цей модуль відповідає за моніторинг появи нових криптовалют, збір відповідної інформації та надання цих даних за допомогою API. Він використовує API для отримання даних з різних джерел, таких як біржі криптовалют, веб-форуми або соціальні мережі. Модуль збору даних забезпечує збір, структурування та збереження цих даних для подальшої роботи з ними.

- Модуль візуалізації даних: Цей модуль відповідає за візуалізацію зібраних та аналізованих даних. Він забезпечує користувачам зручний та зрозумілий спосіб відображення. Модуль візуалізації даних допомагає користувачам швидко отримувати інформацію та зробити обґрунтовані рішення.

- Набір розробника для створення парсерів: Цей набір використовується у модулі збору, збереження та надання даних для коректного запуску парсерів та

обробки отриманих даних від них. У модулі візуалізації даних цей набір використовується для відображення даних.

В загальному вигляді, зв'язок між усіма модулями зображено на рис 2.3.



Рис. 2.3. Зв'язок між модулями

#### 2.4.2 Алгоритми та функціонування системи

Перед початком парсингу, парсер завантажує список криптовалютних проектів за допомогою методу `GetCryptocurrencyList()` та в подальшому він буде використовувати цей список для перевірки нових проектів. Далі запускається безкінечний цикл з встановленим інтервалом у якому виконується наступне:

- Отримання списку нових криптовалют за допомогою методу `GetCryptocurrencyList()`;
- Перевірка отриманих криптовалютних проектів
- Якщо криптовалютний проект є новим – викликається метод `GetCryptocurrencyInfo(string projectId, CryptocurrencyInfo cryptocurrencyInfo)` який отримує детальну інформацію про криптовалюту.
- Після того як усі проекти перевірені та отримана додаткова інформація, дані відправляються на перевірку із загальною базою даних. Якщо такого криптовалюти нема в базі даних, то цей проект додається до бази даних.

Доступ до даних здійснюється за допомогою API. Існують наступні кінцеві точки:

[GET] /cryptocurrency

- /getLatest?count=100 – отримання останніх криптовалют у необхідній кількості. Кількість задається у параметрі count.

- /getById?id=123 – отримання криптовалюти за допомогою унікального номеру у базі даних. Унікальний номер задається у параметрі id.

- /search?query=text – отримання проектів які у містять частину тексту у назві та/або у короткій назві в параметрі query.

В загальному вигляді, робота модулю збору, збереження та надання даних зображена на рис 2.4.

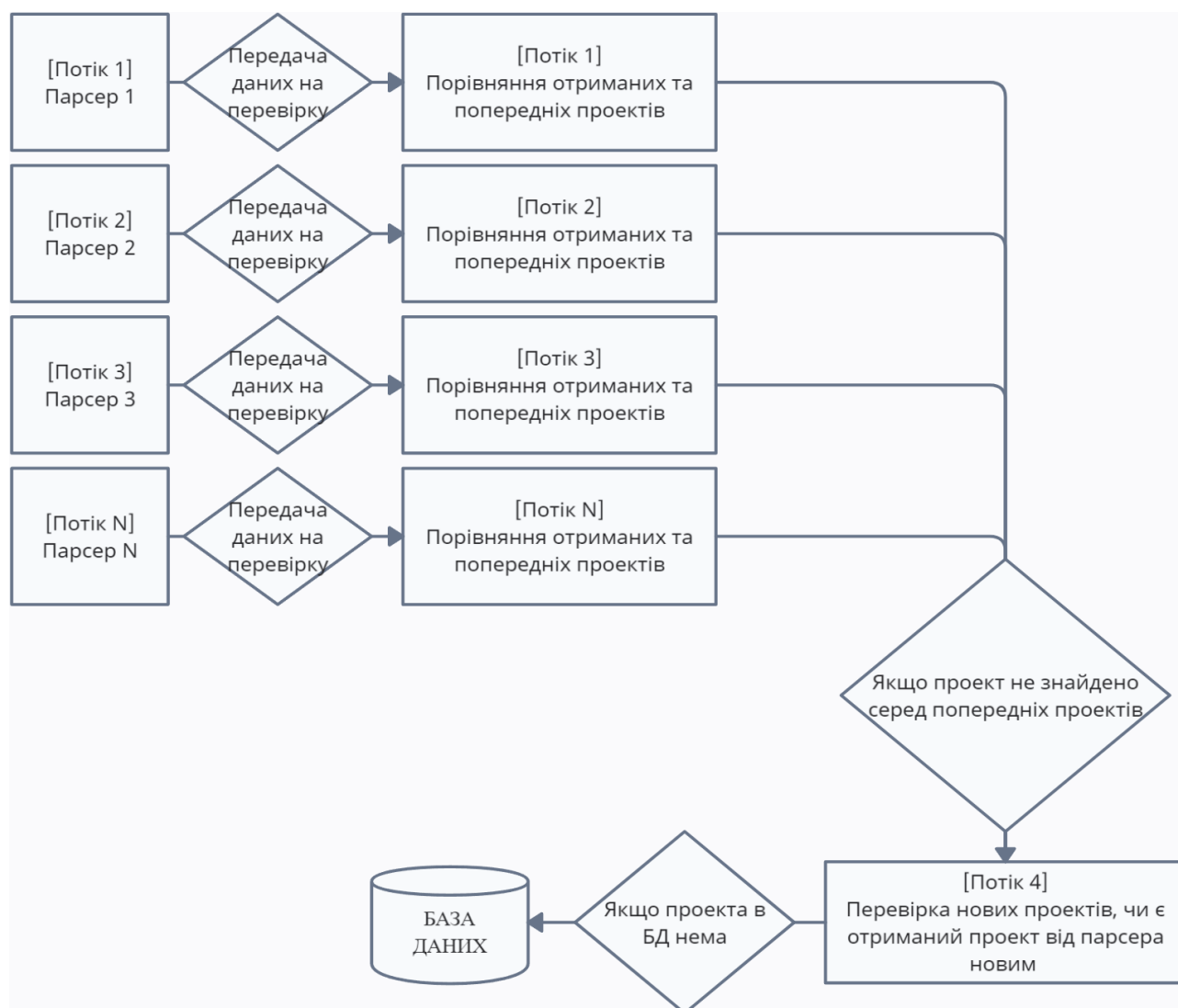


Рис. 2.4. Загальний вигляд роботи модулю збору, збереження та надання даних

## 2.5 Обґрунтування та організація вхідних та вихідних даних програми

В комп'ютерній системі, яка була розроблена, використовуються вхідні дані, які отримуються автоматично з різних джерел за допомогою парсингу. Ці дані мають різноманітний вигляд, формат та структуру. Також до вхідних даних можна віднести такі файли як:

- Конфігураційний файл для модулю збору, збереження та надання даних (\*.json)
- Конфігураційний файл для NLog бібліотеки (\*.config)
- Конфігураційний файл для візуалізації даних (\*.json)
- Файли які містять код парсеру та підключаються окремо до модулю збору, збереження та надання даних за допомогою Roslyn Api (\*.cs)

Вихідними даними є записами в базі даних, які створюються під час роботи комп'ютерної системи. Також вихідними даними вважаються логи які пишуться у файли (\*.log) та у консоль. Загальна схема, як користувач отримує вихідні дані за допомогою усіх описаних модулів наведена на рис. 2.5.

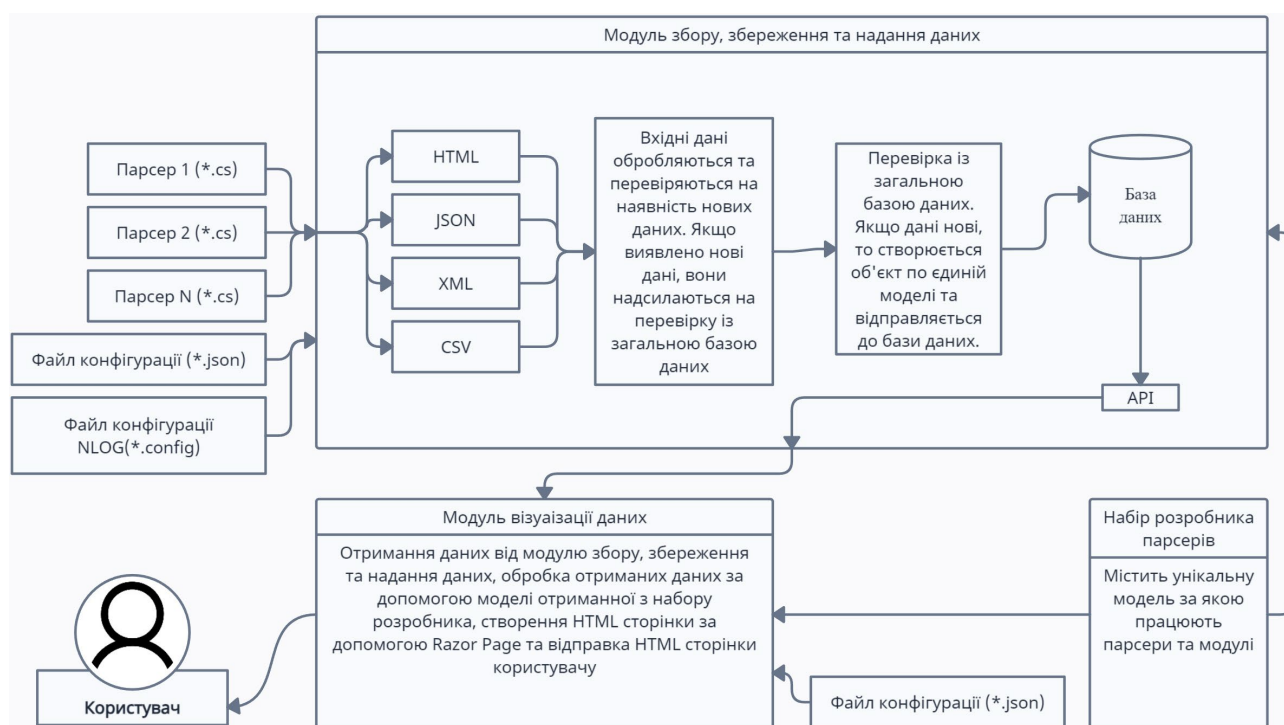


Рис. 2.5. Загальна схема вхідних-вихідних даних

## **2.6 Опис розробленої системи**

### **2.6.1 Використані технічні засоби**

Для успішної розробки даної комп'ютерної системи було необхідно скористатися різноманітними технічними засобами. Зокрема, перелічимо основні компоненти, які було задіяно:

1. Персональний комп'ютер: Це основний робочий інструмент, який використовується для розробки та програмування. Він має достатній обсяг пам'яті, потужний процесор і достатню кількість роз'ємів для підключення різноманітних пристроїв.

2. Мережеве обладнання: Для побудови мережевого середовища, в якому можуть взаємодіяти різні комп'ютери та пристрої, було використано спеціальне мережеве обладнання. Це можуть бути маршрутизатори, комутатори, міжмережеві екрани, а також кабельна інфраструктура.

3. Клавіатура: Клавіатура є основним засобом введення інформації до комп'ютера. Вона використовується для набору тексту, введення команд, керування програмами тощо. Для зручності розробки було важливо мати надійну та комфортну клавіатуру.

4. Миша: Миша є пристроєм курсорного керування, який дозволяє точно вказувати об'єкти на екрані комп'ютера. Вона була використана для точного позиціонування, вибору та керування об'єктами під час розробки системи.

5. Монітор: Монітор є пристроєм виведення інформації, який відображає графічний інтерфейс, текст, зображення тощо. У процесі розробки було важливо мати великий та якісний монітор, щоб зручно спостерігати за робочим простором та перевіряти результати своєї роботи.

Ці технічні засоби були використані як складові частини розробки комп'ютерної системи, забезпечуючи комфортну та продуктивну роботу розробника. Важливо мати належне обладнання, яке відповідає вимогам проекту та забезпечує ефективну взаємодію з комп'ютерною системою.

## 2.6.2 Використані програмні засоби

Для успішної розробки даної комп'ютерної системи було використано різноманітні програмні засоби, які забезпечили ефективну та зручну роботу розробників. Основні програмні засоби, що були використані, включають:

1. Операційна система Windows 11 Pro: Це операційна система, яка надає робоче середовище для виконання програм та керування ресурсами комп'ютера. Windows 11 Pro була обрана як платформа для розробки даної комп'ютерної системи.

2. Інтегроване середовище розробки Visual Studio 2022 Community 17.5.1: Це потужне середовище розробки, яке надає широкі можливості для програмування, налагодження та тестування програмного забезпечення. Visual Studio 2022 Community було використано для розробки та збирання компонентів комп'ютерної системи.

3. Інтегроване середовище розробки Visual Studio Code 1.78.2: Це легке та розширюване середовище розробки, яке підтримує різні мови програмування. Visual Studio Code було використано для редагування та перегляду джерела коду компонентів системи.

4. Система керування та моніторингу серверів SQL Server Management Studio 19: Ця система надає інструменти для керування базами даних SQL Server, виконання запитів та моніторингу роботи серверів. SQL Server Management Studio було використано для розробки та управління базами даних, які використовує комп'ютерна система.

5. Набір розробника .Net Core 6.0.410: Це набір інструментів та бібліотек для розробки програмного забезпечення на платформі .NET. .Net Core було використано для розробки та розгортання компонентів системи.

6. Система контролю версій Git: Git є розподіленою системою контролю версій, яка дозволяє відстежувати зміни в коді, спільно працювати над проектами та керувати версіями програмного забезпечення. Git було



використано для зберігання та управління кодовою базою комп'ютерної системи.

Ці програмні засоби надали потужний інструментарій для створення та управління комп'ютерною системою, забезпечивши ефективність та продуктивність у процесі розробки.

### **2.6.3 Виклик та завантаження програми**

Залежно від параметрів компіляції, спосіб виклику програми може варіюватися. Якщо програма була скомпільована з параметрами "Режим розгортання: Переносний" та "Цільова середа виконання: Переносна версія" або іншим значенням, для запуску потрібно ввести наступну команду у консоль або термінал: `dotnet NewCryptoParser.dll`. Перед цим необхідно встановити на цей комп'ютер комплект розробника .Net Core 6.0.xxx. Якщо було обрано режим розгортання "Автономний" і вказано операційну систему в параметрі "Цільова середа виконання", для запуску програми необхідно виконати ті самі дії, які потрібні для запуску інших програм на даній операційній системі. Наприклад, для Windows це означає запуск файлу `NewCryptoParser.exe`, а для Linux - команда `./NewCryptoParser`, попередньо встановивши права на запуск за допомогою команди `sudo chmod 755 NewCryptoParser`.

### **2.6.4 Опис інтерфейсу користувача**

Коли користувач заходить на веб-сайт, він одразу побачить сторінку з останніми 100 новими криптовалютами. Ця сторінка зображена на рис 2.6.

Головна Пошук... Знайти

### Останні 100 отриманих криптовалют

#id	Назва	Тікер	Дата парсінгу	Знайдено на	
206	<a href="#">Father Pepe Inu</a>	FAPENU	15.06.2023 20:06:18	CoinMarketCap	
205	<a href="#">Uncle Benis</a>	BENIS	15.06.2023 20:06:18	CoinMarketCap	
204	<a href="#">JennyCo</a>	JCO	15.06.2023 19:41:53	CoinPaprika	
203	<a href="#">FraxEther</a>	FRXETH	09.06.2023 14:48:42	CoinPaprika	
202	<a href="#">TAJQ</a>	TAJQ	09.06.2023 00:53:14	CoinPaprika	
201	<a href="#">BUSD Hong Kong</a>	BUSDHK	09.06.2023 00:43:09	CoinPaprika	
200	<a href="#">The boss</a>	Boss	09.06.2023 00:21:56	CoinCatapult	
199	<a href="#">Pepe Apes</a>	Apes	08.06.2023 23:57:13	CoinCatapult	
198	<a href="#">BlueProfitBot</a>	BlueBot	08.06.2023 23:00:37	CoinCatapult	
197	<a href="#">Junji Chu</a>	JUCHU	08.06.2023 22:47:14	CoinPaprika	
196	<a href="#">STRAY CAT COIN</a>	SCAT	08.06.2023 22:21:36	CoinCatapult	
195	<a href="#">Pepito</a>	PITO	08.06.2023 22:14:32	CoinCatapult	
194	<a href="#">Delphi Privacy Protocol</a>	DPHI	08.06.2023 21:27:45	CoinMarketCap	
193	<a href="#">Kakapo Protocol</a>	KAKAPO	08.06.2023 21:27:45	CoinMarketCap	

Рис. 2.6. Головна сторінка

У випадку невдалого з'єднання з сервером, користувач побачить сторінку, на якій буде відображено повідомлення про невдачу у з'єднанні з сервером. Ця сторінка зображена на рис 2.7

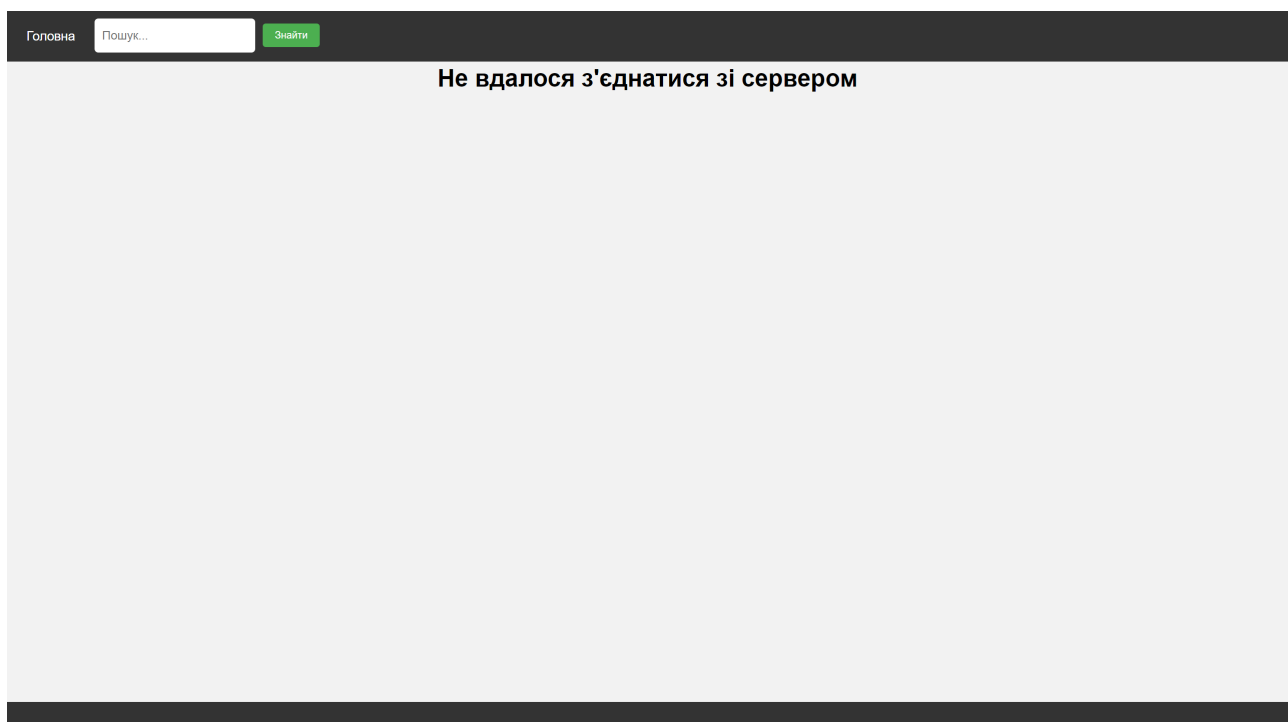


Рис. 2.7. Сторінка при невдалій спробі з'єднатися зі сервером

Сторінка «Детальна інформація» містить наступне:

- список платформ з якими складено угоду;
- список посилань на соціальні мережі, офіційний сайт, тощо;
- опис криптовалюти.

Ці сторінки зображені на рис. 2.8, рис. 2.9, та рис. 2.10.

Головна Пошук... Знайти

### Детальна інформація про знайдену криптовалюту

Назва	Father Pepe Inu
Тікер	FAPENU
Дата парсингу	15.06.2023 20:06:18
Знайдено на	<a href="#">CoinMarketCap</a>

#### Опис

Father Pepe Inu (FAPENU) is a cryptocurrency launched in 2023 and operates on the BNB Smart Chain (BEP20) platform. Father Pepe Inu has a current supply of 1,000,000,000 with 0 in circulation. The last known price of Father Pepe Inu is 0.00003598 USD and is down -3.79 over the last 24 hours. It is currently trading on 2 active market(s) with \$0.00 traded over the last 24 hours. More information can be found at <https://fatherpepeinu.com>.

#### Посилання

Назва	Посилання
Website	<a href="https://fatherpepeinu.com">https://fatherpepeinu.com</a>
Explorer	<a href="https://bscscan.com/token/0xd639b209eD72b475DE848F8e9173135CBe8edbD">https://bscscan.com/token/0xd639b209eD72b475DE848F8e9173135CBe8edbD</a>
SourceCode	<a href="https://github.com/fatherpepeinu/FatherPepeInu/commit/cfc8d87b8daa724a3f778dec9b53d4b8fb90aeb5">https://github.com/fatherpepeinu/FatherPepeInu/commit/cfc8d87b8daa724a3f778dec9b53d4b8fb90aeb5</a>
Twitter	<a href="https://twitter.com/fatherpepeinu">https://twitter.com/fatherpepeinu</a>
Medium	<a href="https://medium.com/@fatherpepeinu/introducing-father-pepe-inu-unleash-the-power-of-blockchain-gaming-97415d1fd0c6">https://medium.com/@fatherpepeinu/introducing-father-pepe-inu-unleash-the-power-of-blockchain-gaming-97415d1fd0c6</a>
Unknown	<a href="https://father-pepe-inu.gitbook.io/father-pepe-inu/getting-started/introduction">https://father-pepe-inu.gitbook.io/father-pepe-inu/getting-started/introduction</a>
Telegram	<a href="https://t.me/Fatherpepeinu">https://t.me/Fatherpepeinu</a>
Discord	<a href="https://discord.gg/3J483W2Hv8">https://discord.gg/3J483W2Hv8</a>

#### Платформи

Назва	Тип	Smart Contract
BNB		0xd639b209eD72b475DE848F8e9173135CBe8edbD

Рис. 2.8. Сторінка «Детальна інформація»

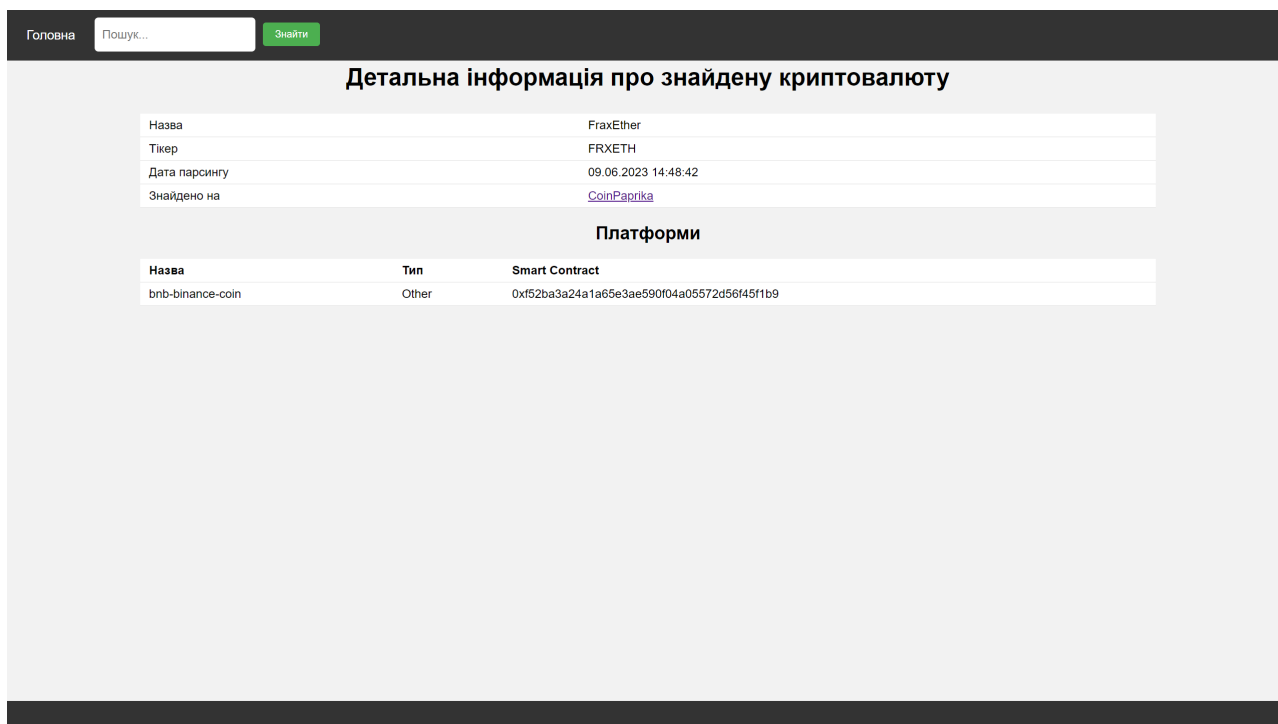


Рис. 2.9. Сторінка «Детальна інформація» (Не містить посилань та опису)

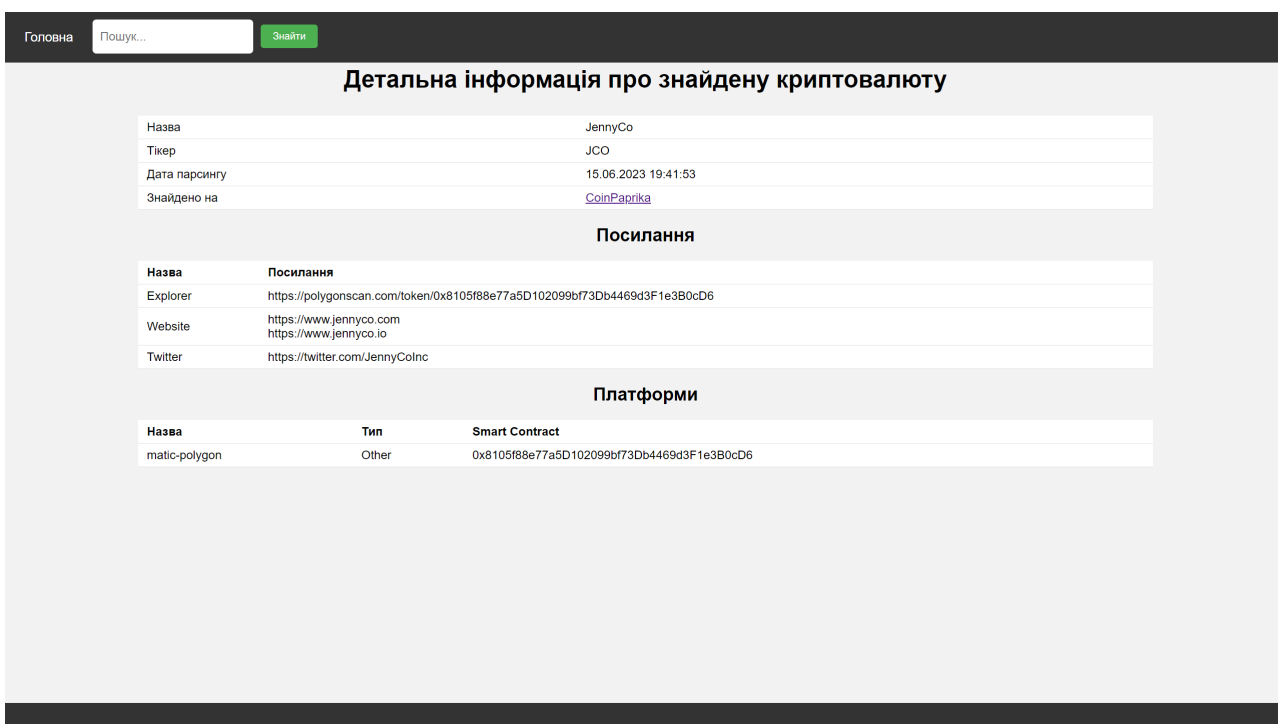


Рис. 2.10. Сторінка «Детальна інформація» (Не містить опису)

На веб-сайті є пошук криптовалют за назвою або скороченою назвою. Він розташований у верхньому лівому куті. Ця сторінка з пошуковим запитом «AB» (латинськими) зображена на рис. 2.11.

Головна

### Результат пошуку











#id	Назва	Тікер	Дата парсінгу	Знайдено на	
180	<a href="#">HongKong Baby</a>	HKBABY	08.06.2023 18:29:45	CoinPaprika	
177	<a href="#">Profit Blue Baby</a>	BLUEBABY	08.06.2023 18:19:40	CoinPaprika	
168	<a href="#">Abyss World</a>	awt	08.06.2023 12:54:47	CoinGecko	
151	<a href="#">BabySafuu</a>	BBYSFU	08.06.2023 12:17:11	CoinPaprika	
128	<a href="#">Relation Labs</a>	REL	06.06.2023 13:30:33	CoinPaprika	
126	<a href="#">Dexter Labs</a>	DLABS	06.06.2023 13:30:33	CoinPaprika	
100	<a href="#">Rabbit Peter</a>	PETER	06.06.2023 12:13:52	CoinPaprika	
90	<a href="#">Babadoge</a>	BABADOGE	05.06.2023 21:58:08	CoinMarketCap	
36	<a href="#">BabyFlokiZilla</a>	BABYFLOKIZILLA	05.06.2023 21:45:25	CoinCodex	
28	<a href="#">BabyDogeZilla</a>	BABYDOGEZILLA	05.06.2023 21:45:25	CoinCodex	

Рис. 2.11. Сторінка «Результат пошуку»

Якщо користувач введе текст запиту на який не знайдеться жодної криптовалюти, веб-сайт відобразить сторінку яка зображена на рис. 2.12.

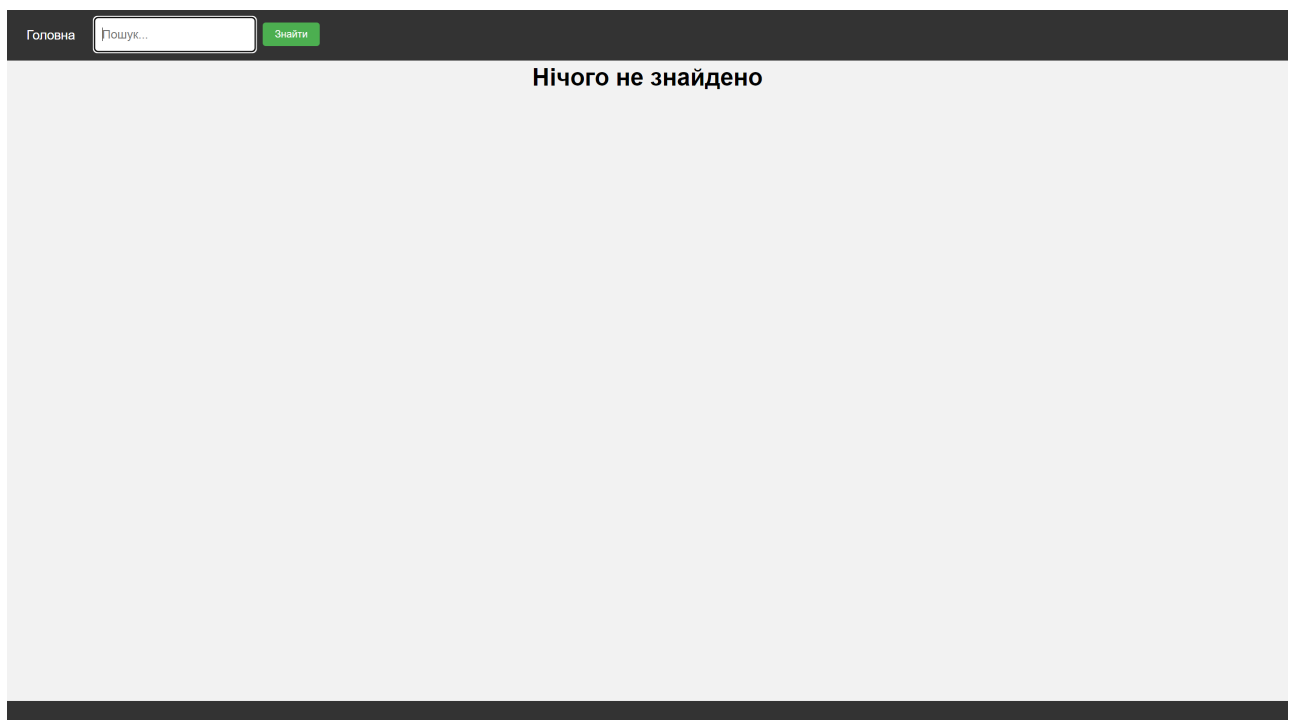
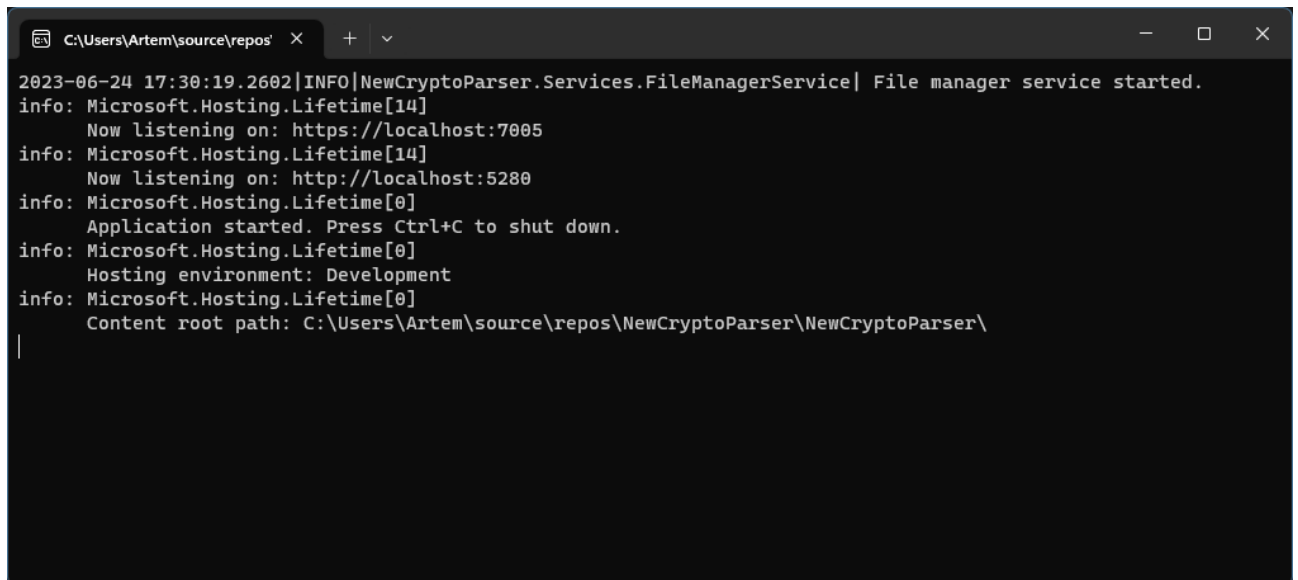


Рис. 2.12. Сторінка «Невдалий результат пошуку»

## 2.6.5 Опис консольного інтерфейсу

Консольний інтерфейс в першу чергу призначений для розробників або для технічних адміністраторів. У цьому підрозділі буде розглянуто як виглядає консольний інтерфейс, які повідомлення від відображає та які помилки можуть виникати під час експлуатації.

Спершу буде описано консольний інтерфейс модулю збору, збереження та надання даних. При успішному запуску програми, текст у консолі буде як на рис. 2.13.



```
C:\Users\Artem\source\repos' x + v - □ x
2023-06-24 17:30:19.2602|INFO|NewCryptoParser.Services.FileManagerService| File manager service started.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7005
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5280
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\
|
```

Рис. 2.13. Успішний запуск модулю збору, збереження та надання даних

Якщо при запуску модулю виникне непередбачувана помилка то у консолі з'явиться повідомлення про помилку та модуль модулю збору, збереження та надання даних завершить свою роботу.

При додаванні парсеру до модулю збору, збереження та надання даних, у консолі з'явиться повідомлення про додання нового парсеру. Це повідомлення зображено на рис. 2.14.

```
C:\Users\Artem\source\repos' x + v
2023-06-24 17:30:19.2602|INFO|NewCryptoParser.Services.FileManagerService| File manager service started.
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:7005
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5280
info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
  Content root path: C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\
2023-06-24 17:30:46.2452|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinMarketCap] added
|
```

Рис. 2.14. Успішне додання парсеру до модулю збору, збереження та надання даних.

Якщо парсер знайде нову криптовалюту, з'явиться відповідне повідомлення у консолі. Це повідомлення зображено на рис. 2.15.

```
C:\Users\Artem\source\repos' x + v
2023-06-24 17:30:19.2602|INFO|NewCryptoParser.Services.FileManagerService| File manager service started.
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:7005
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5280
info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
  Content root path: C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\
2023-06-24 17:30:46.2452|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinMarketCap] added
2023-06-24 17:30:49.3264|INFO|NewCryptoParser.Services.ParserManagerService| [CoinMarketCap] Found 1 new project(-s)
|
```

Рис. 2.15. Повідомлення про знайдену нову криптовалюту.

При видаленні парсеру із модулю, у консолі з'явиться повідомлення про видалення. Це повідомлення зображено на рис. 2.16.

```
C:\Users\Artem\source\repos' x + v
2023-06-24 17:30:19.2602|INFO|NewCryptoParser.Services.FileManagerService| File manager service started.
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:7005
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5280
info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
  Content root path: C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\
2023-06-24 17:30:46.2452|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinMarketCap] added
2023-06-24 17:30:49.3264|INFO|NewCryptoParser.Services.ParserManagerService| [CoinMarketCap] Found 1 new project(-s)
2023-06-24 17:31:16.4546|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinMarketCap] removed
```

Рис. 2.16. Повідомлення про видаленний парсер.

Якщо парсер, що додається містить синтаксичну помилку, у консолі з'явиться повідомлення про помилку в якому буде описана синтаксична помилка. Це повідомлення зображено на рис 2.17.

```
C:\Users\Artem\source\repos' x + v
2023-06-24 17:30:49.3264|INFO|NewCryptoParser.Services.ParserManagerService| [CoinMarketCap] Found 1 new project(-s)
2023-06-24 17:31:16.4546|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinMarketCap] removed
2023-06-24 17:35:45.3167|ERROR|NewCryptoParser.Services.FileManagerService| An error occurred while adding a parser from file [C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Parsers\CoinMarketCap.cs]. NewCryptoParser.Exceptions.ParserAddException: Parser contains error
--> NewCryptoParser.Exceptions.CompilerException: One or more errors occurred. (} expected)
--> System.Exception: } expected
--- End of inner exception stack trace ---
at NewCryptoParser.CodeCompiler.CompileCodeAndGetObject[T](String code) in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\CodeCompiler.cs:line 37
at NewCryptoParser.Services.ParserManagerService.createCryptoParserScheduledTask(String code, String name) in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Services\ParserManagerService.cs:line 54
at NewCryptoParser.Services.ParserManagerService.AddParser(String code, String name) in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Services\ParserManagerService.cs:line 22
--- End of inner exception stack trace ---
at NewCryptoParser.Services.ParserManagerService.AddParser(String code, String name) in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Services\ParserManagerService.cs:line 26
at NewCryptoParser.Services.FileManagerService.<ExecuteAsync>b__5_1(Object o, FileSystemEventArgs e) in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Services\FileManagerService.cs:line 56
```

Рис. 2.17. Повідомлення про синтаксичну помилку в парсері.

Якщо в парсері під час виконання станеться помилка, у консолі з'являється повідомлення що саме трапилось під час виконання. Приклад однієї з помилок зображено на рис. 2.18.



```
C:\Users\Artem\source\repos' X + v - □ X
at NewCryptoParser.Services.ParserManagerService.<>c__DisplayClass6_0.<<createCryptoParserSch
eduledTask>b__0>d.MoveNext() in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Serv
ices\ParserManagerService.cs:line 79
--- End of inner exception stack trace ---
2023-06-27 14:11:14.8580|ERROR|NewCryptoParser.Services.ParserManagerService| Parser error [Coin
Gecko] NewCryptoParser.Exceptions.ParserRuntimeException: One or more errors occurred. (Response
status code does not indicate success: 429 (Too Many Requests).)
---> System.AggregateException: One or more errors occurred. (Response status code does not ind
icate success: 429 (Too Many Requests).)
---> System.Net.Http.HttpRequestException: Response status code does not indicate success: 429
(Too Many Requests).
at System.Net.Http.HttpResponseMessage.EnsureSuccessStatusCode()
at System.Net.Http.HttpClient.GetStringAsyncCore(HttpRequestMessage request, CancellationToke
n cancellationToken)
--- End of inner exception stack trace ---
at System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions)
at System.Threading.Tasks.Task'1.GetResultCore(Boolean waitCompletionNotification)
at System.Threading.Tasks.Task'1.get_Result()
at ParserPlugin.CoinGecko.GetCryptocurrencyList()
at NewCryptoParser.Services.ParserManagerService.<>c__DisplayClass6_0.<<createCryptoParserSch
eduledTask>b__0>d.MoveNext() in C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Serv
ices\ParserManagerService.cs:line 79
--- End of inner exception stack trace ---
2023-06-27 14:11:14.9502|ERROR|NewCryptoParser.Services.ParserManagerService| Parser error [Coin
Gecko] NewCryptoParser.Exceptions.ParserRuntimeException: One or more errors occurred. (Response
status code does not indicate success: 429 (Too Many Requests).)
---> System.AggregateException: One or more errors occurred. (Response status code does not ind
icate success: 429 (Too Many Requests).)
---> System.Net.Http.HttpRequestException: Response status code does not indicate success: 429
(Too Many Requests).
at System.Net.Http.HttpResponseMessage.EnsureSuccessStatusCode()
at System.Net.Http.HttpClient.GetStringAsyncCore(HttpRequestMessage request, CancellationToke
n cancellationToken)
--- End of inner exception stack trace ---
at System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions)
at System.Threading.Tasks.Task'1.GetResultCore(Boolean waitCompletionNotification)
at System.Threading.Tasks.Task'1.get_Result()
```

Рис. 2.18. Помилка під час роботи парсеру.

Якщо парсер було успішно завантажено до комп'ютерної системи та було успішно скомпільовано, але під час ініціалізації станеться помилка, парсер буде зупинено та видалено із системи. У консолі з'являться відповідні повідомлення. Ці повідомлення зображені на рис. 2.19.

```
C:\Users\Artem\source\repos' x + v
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7005
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5280
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\
2023-06-27 14:13:04.3916|INFO|NewCryptoParser.Services.ParserManagerService| Parser [CoinGecko] added
2023-06-27 14:13:04.3916|INFO|NewCryptoParser.Services.FileManagerService| Parser [C:\Users\Artem\source\repos\NewCryptoParser\NewCryptoParser\Parsers\CoinGecko.cs] successfully loaded
2023-06-27 14:13:04.3916|INFO|NewCryptoParser.Services.FileManagerService| File manager service started.
2023-06-27 14:13:04.4909|ERROR|NewCryptoParser.Services.ParserManagerService| [CoinGecko] System.Net.Http.HttpRequestException: Response status code does not indicate success: 429 (Too Many Requests).
      at System.Net.Http.HttpResponseMessage.EnsureSuccessStatusCode()
      at System.Net.Http.HttpClient.GetStringAsyncCore(HttpRequestMessage request, CancellationToken cancellationToken)
2023-06-27 14:13:04.4909|WARN|NewCryptoParser.Services.ParserManagerService| [CoinGecko] Task stopped
```

Рис. 2.19. Помилка при ініціалізації парсера.

Далі буде описано консольний інтерфейс модулю візуалізації даних. При успішному запуску модулю, у консолі з'являться повідомлення, як на рис. 2.20.

```
C:\Users\Artem\source\repos' x + v
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7240
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5107
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Artem\source\repos\NewCryptoParser\Website\
```

Рис. 2.20. Успішний запуск модулю візуалізації даних.

Також у консолі будуть відображатися усі HTTP запити від користувачів. Такі повідомлення зображені на рис. 2.21.

```
C:\Users\Artem\source\repos' X + v - □ X
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7240
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5107
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Artem\source\repos\NewCryptoParser\Website\
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[100]
      Start processing HTTP request GET https://localhost:7005/cryptocurrency/getLatest?count=100
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[100]
      Sending HTTP request GET https://localhost:7005/cryptocurrency/getLatest?count=100
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[101]
      Received HTTP response headers after 549.5825ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[101]
      End processing HTTP request after 556.8058ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[100]
      Start processing HTTP request GET https://localhost:7005/cryptocurrency/GetById?id=209
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[100]
      Sending HTTP request GET https://localhost:7005/cryptocurrency/GetById?id=209
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[101]
      Received HTTP response headers after 59.5807ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[101]
      End processing HTTP request after 59.6609ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[100]
      Start processing HTTP request GET https://localhost:7005/cryptocurrency/search?query=AB
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[100]
      Sending HTTP request GET https://localhost:7005/cryptocurrency/search?query=AB
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[101]
      Received HTTP response headers after 15.6521ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[101]
      End processing HTTP request after 15.7195ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[100]
      Start processing HTTP request GET https://localhost:7005/cryptocurrency/search?query=ABC
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[100]
      Sending HTTP request GET https://localhost:7005/cryptocurrency/search?query=ABC
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[101]
      Received HTTP response headers after 12.1328ms - 200
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[101]
      End processing HTTP request after 12.2124ms - 200
```

Рис. 2.21. Повідомлення про зроблені HTTP запити від користувачів

Якщо користувач запитує дані у модуля візуалізації даних, а модуль збору, збереження та надання даних буде недоступний у цей час, у консолі з'явиться повідомлення про помилку з'єднання з модулем (рис. 2.22.), а користувач побачить сторінку як на рис. 2.7.

```
C:\Users\Artem\source\repos' X + v - □ X
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7240
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5107
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Artem\source\repos\NewCryptoParser\Website\
info: System.Net.Http.HttpClient.NewCryptoParser.LogicalHandler[100]
      Start processing HTTP request GET https://localhost:7005/cryptocurrency/getLatest?count=
100
info: System.Net.Http.HttpClient.NewCryptoParser.ClientHandler[100]
      Sending HTTP request GET https://localhost:7005/cryptocurrency/getLatest?count=100
fail: Website.Pages.IndexModel[0]
      OnGet error
      System.Net.Http.HttpRequestException: No connection could be made because the target mac
hine actively refused it. (localhost:7005)
      --> System.Net.Sockets.SocketException (10061): No connection could be made because th
e target machine actively refused it.
         at System.Net.Sockets.Socket.AwaitableSocketAsyncEventArgs.ThrowException(SocketError
error, CancellationToken cancellationToken)
         at System.Net.Sockets.Socket.AwaitableSocketAsyncEventArgs.System.Threading.Tasks.Sou
rces.IValueTaskSource.GetResult(Int16 token)
         at System.Net.Sockets.Socket.<ConnectAsync>g__WaitForConnectWithCancellation|277_0(Aw
aitableSocketAsyncEventArgs saea, ValueTask connectTask, CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.ConnectToTcpHostAsync(String host, Int32 port,
HttpRequestMessage initialRequest, Boolean async, CancellationToken cancellationToken)
      --- End of inner exception stack trace ---
         at System.Net.Http.HttpConnectionPool.ConnectToTcpHostAsync(String host, Int32 port,
HttpRequestMessage initialRequest, Boolean async, CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.ConnectAsync(HttpRequestMessage request, Boolea
n async, CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.CreateHttp11ConnectionAsync(HttpRequestMessage
request, Boolean async, CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.AddHttp11ConnectionAsync(HttpRequestMessage req
uest)
         at System.Threading.Tasks.TaskCompletionSourceWithCancellation`1.WaitWithCancellation
Async(CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.GetHttp11ConnectionAsync(HttpRequestMessage req
uest, Boolean async, CancellationToken cancellationToken)
         at System.Net.Http.HttpConnectionPool.SendWithVersionDetectionAndRetryAsync(HttpReque
stMessage request, Boolean async, Boolean doRequestAuth, CancellationToken cancellationToken)
         at System.Net.Http.DiagnosticsHandler.SendAsyncCore(HttpRequestMessage request, Boole
an async, CancellationToken cancellationToken)
         at System.Net.Http.RedirectHandler.SendAsync(HttpRequestMessage request, Boolean asyn
c, CancellationToken cancellationToken)
         at Microsoft.Extensions.Http.Logging.LoggingHttpMessageHandler.SendAsync(HttpRequestM
essage request, CancellationToken cancellationToken)
         at Microsoft.Extensions.Http.Logging.LoggingScopeHttpMessageHandler.SendAsync(HttpReq
uestMessage request, CancellationToken cancellationToken)
         at System.Net.Http.HttpClient.GetStringAsyncCore(HttpRequestMessage request, Cancell
ationToken cancellationToken)
         at Website.Pages.IndexModel.OnGet() in C:\Users\Artem\source\repos\NewCryptoParser\We
bsite\Pages\Index.cshtml.cs:line 37
```

Рис. 2.22. Невдале з'єднання з модулем збору, збереження та надання даних

## РОЗДІЛ 3

### ЕКОНОМІЧНИЙ РОЗДІЛ

Вихідні дані розробки програмного забезпечення:

- 1) передбачуване число операторів – 1528
- 2) коефіцієнт складності програми – 1,25
- 3) коефіцієнт кореляції програми в ході її розробки - 0,1
- 4) середня годинна заробітна плата програміста, грн/рік – 156,25
- 5) коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,25
- 6) коефіцієнт кваліфікації програміста, обумовлений від стажу роботи даної кваліфікації – 1,4
- 7) вартість машино-години ЕОМ, грн/рік – 21,32

#### **3.1 Розрахунок трудомісткості та вартості розробки програмного продукту**

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_a$  – витрати праці на розробку блок-схеми алгоритму;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{oml}$  – витрати праці на налагодження програми на ЕОМ;

$t_d$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C(1 + p), \text{ де} \quad (3.2)$$

$q$  – передбачуване число операторів;

$C$  – коефіцієнт складності програми;

$p$  – коефіцієнт кореляції програми в ході її розробки.

$$Q = 1528 \cdot 1,25 (1 + 0,1) = 2101$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85)K}, \text{ людино-годин,} \quad (3.3)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі,  $B=1.2 \dots 1.5$ ;

$K$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. до 2 – 0,8;

$$t_u = \frac{2101 \cdot 1,25}{85 \cdot 1,4} = 22,07, \text{ людина – годин}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20..25)K} \quad (3.4)$$
$$t_a = \frac{2101}{25 \cdot 1,4} = 60,03, \text{ людина – годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25)K} \quad (3.5)$$
$$t_n = \frac{2101}{25 \cdot 1,4} = 60,03, \text{ людина – годин}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5)K} \quad (3.6)$$
$$t_{oml} = \frac{2101}{5 \cdot 1,4} = 300,14 \text{ людина – годин}$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,2 \cdot t_{oml} \quad (3.7)$$
$$t_{oml}^k = 1,2 \cdot 300,14 = 360,17, \text{ людина – годин}$$

Витрати праці на підготовку документації:

$$t_d = t_{dp} + t_{do} \quad (3.8)$$

де  $t_{dp}$  – трудомісткість підготовки матеріалів і рукопису

$$t_{дp} = \frac{Q}{(15..20)K} \quad (3.9)$$

$$t_{дp} = \frac{2101}{20 \cdot 1,4} = 75,04, \text{ людина-годин}$$

$t_{до}$  – трудомісткість редагування, печатки й оформлення документації

$$t_{до} = 0,75 \cdot t_{дp} \quad (3.10)$$

$$t_{до} = 0,75 \cdot 75,04 = 56,28$$

$$t_{\phi} = 75,04 + 56,28 = 131,32, \text{ людина – годин}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 22,07 + 60,03 + 60,03 + 360,17 + 131,32 = 683,62, \text{ людино-годин.}$$

У результаті ми розрахували, що в загальній складності необхідно 683,62 людино-годин для розробки даного програмного забезпечення.

### 3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ  $K_{но}$  включають витрати на заробітну плату виконавця програми  $Z_{зп}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{но} = Z_{зп} + Z_{мв}, \text{ грн,} \quad (3.11)$$

де  $Z_{зп}$  – заробітна плата виконавців, яка визначається за формулою:

$$Z_{зп} = t \cdot C_{сп}, \text{ грн,} \quad (3.12)$$



де  $t$  – загальна трудомісткість, людино-годин;

$C_{ПР}$  – середня годинна заробітна плата програміста, грн/година  
(<https://ua.jobble.org/salary/junior-c%23%2F.net-developer/%D0%9A%D0%B8%D1%97%D0%B2>)

$$З_{ЗП} = 683,62 \cdot 156,25 = 106815,63, \text{грн.}$$

$З_{МВ}$ – Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{МВ} = t_{омл} \cdot C_M, \text{ грн,} \quad (3.13)$$

де  $t_{омл}$  – трудомісткість налагодження програми на ЕОМ, год.

$C_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$З_{МВ} = 360,17 \cdot 21,32 = 7678,82, \text{ грн.}$$

$$К_{ПО} = 106815,63 + 7678,82 = 114494,45, \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.} \quad (3.14)$$

де  $B_k$  - число виконавців;

$F_p$  - місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p=176$  годин).

$$T = \frac{683,62}{1 \cdot 176} = 3,88 \text{ міс.}$$

**Висновки.** Час розробки даного програмного забезпечення складає 683,62 людино-годин. Таким чином, очікувана тривалість розробки складе 3,88 місяця при 40 годинному робочому тижні (місячний фонд робочого часу 176 годин), а витрати на створення програмного забезпечення складатимуть 114494,45 грн.

## ВИСНОВКИ

Виконана кваліфікаційна робота підтверджує, що розроблена комп'ютерна технологія моніторингу появи нових криптовалют є ефективним інструментом для аналізу та спостереження за динамікою криптовалютного ринку. Її ефективність як інструменту для аналізу та спостереження за динамікою криптовалютного ринку наголошується підтвердженими результатами. Ця технологія дозволяє оперативно виявляти нові криптовалюти та робити прогнози щодо їх потенціалу та ринкової вартості, що є надзвичайно цінною інформацією для інвесторів, трейдерів та фахівців з фінансового аналізу.

Проте важливо усвідомити, що ринок криптовалют є надзвичайно динамічним і піддається значним змінам. Тому розроблена технологія потребує постійного вдосконалення та оновлення, а також постійного доступу до актуальних даних для досягнення оптимальних результатів. Лише таким чином можна забезпечити надійну та точну інформацію про ринок криптовалют, що є важливим фактором для успішних інвестиційних рішень.

У світлі швидкого розвитку криптовалютного ринку, рекомендується розширювати функціонал системи моніторингу, включаючи додаткові аспекти аналізу криптовалютного ринку. Наприклад, враховувати суспільну думку та вплив новин на вартість криптовалют. Аналізувати реакції громадськості, думки експертів та масових ЗМІ може допомогти у зрозумінні того, як певні події та інформація впливають на курс криптовалют. Крім того, безпека зберігання та передачі даних є критично важливою у цій галузі, оскільки велика кількість грошей та фінансових активів залучена до криптовалютного простору. Тому необхідно приділяти належну увагу заходам безпеки для запобігання крадіжкам та несанкціонованому доступу до цінної інформації.

У цілому, розробка комп'ютерної технології моніторингу появи нових криптовалют має великий потенціал у розвитку фінансової технології. Вона дозволяє отримувати об'єктивну та швидко інформацію про криптовалютний ринок, що сприяє прийняттю кращих рішень щодо інвестування та торгівлі

криптовалютами. Проте необхідно пам'ятати про постійні зміни та вразливість ринку, що вимагає постійного вдосконалення технології та заходів безпеки для забезпечення успіху в цій галузі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system // Decentralized business review. – 2008. – С. 21260.
2. Delmolino K. et al. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab // Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20. – Springer Berlin Heidelberg, 2016. – С. 79-94.
3. Bauer M. D. Linux server security. – " O'Reilly Media, Inc.", 2005.
4. Krause J. Mastering Windows Server 2016. – Packt Publishing Ltd, 2016.
5. Ward B. How Linux works: What every superuser should know. – no starch press, 2021.
6. Camargos F., Girard G., Ligneris B. Virtualization of Linux servers //Proceedings of the Linux Symposium. – 2008. – Т. 2008.
7. C# docs - get started, tutorials, reference. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/uk-ua/dotnet/csharp/> (дата звернення: 23.06.2023).
8. Troelsen A., Japikse P. Pro C# 7: With. net and. net Core. – Apress, 2017. – Т. 1328.
9. Richter J. CLR via c#. – Redmond : Microsoft Press, 2006. – Т. 4.
10. Aroraa G., Chilberto J. Hands-On Design Patterns with C# and. NET Core: Write clean and maintainable code by using reusable solutions to common software design problems. – Packt Publishing Ltd, 2019.
11. A sample ASP.NET Core app's runtime architecture. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/ru-ru/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures> (дата звернення: 27.06.2023).

12. Cole M. R. Hands-on Microservices with C#: Designing a Real-world, Enterprise-grade Microservice Ecosystem with the Efficiency of C# 7. – Packt Publishing Ltd, 2018.
13. Price M. J. C# 10 and .NET 6—Modern Cross-Platform Development: Build apps, websites, and services with ASP.NET Core 6, Blazor, and EF Core 6 using Visual Studio 2022 and Visual Studio Code. – Packt Publishing Ltd, 2021.
14. Vasani M. Roslyn Cookbook. – Packt Publishing Ltd, 2017.
15. Pezoa F. et al. Foundations of JSON schema // Proceedings of the 25th international conference on World Wide Web. – 2016. – С. 263-273.
16. Gorman K. et al. Introducing Microsoft SQL Server 2019: Reliability, scalability, and security both on premises and in the cloud. – Packt Publishing Ltd, 2020.
17. Lawson B., Sharp R. Introducing html5. – New Riders, 2011.
18. Lubbers P. et al. Pro HTML5 programming. – New York, NY, USA: Apress, 2011. – С. 107-133.
19. Robbins J. N. Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. – " O'Reilly Media, Inc.", 2012.
20. Duckett J. Web design with HTML, CSS, JavaScript and jQuery set. – Wiley Publishing, 2014.
21. Phulare A. Commands and Operations in Git - DZone, 2019. dzone.com. URL: <https://dzone.com/articles/git-tutorial-commands-and-operations-in-git> (дата звернення: 25.06.2023).
22. Loeliger J., McCullough M. Version Control with Git: Powerful tools and techniques for collaborative software development. – " O'Reilly Media, Inc.", 2012.
23. Скорочення слів в українській мові у бібліографічному описі. Загальні правила та вимоги: ДСТУ 3582-97. – Чинний від 1998–07–01. – К. : Держстандарт України, 1998. – 24 с. – (Державний стандарт України).
24. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: (ГОСТ 7.1-2003, IDT) : ДСТУ ГОСТ 7.1:2006. – Чинний з 2007–07–01. – К. : Держспоживстандарт України, 2007. – 47 с. – (Система

стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України).

25. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Комп’ютерні системи” / Уклад О.Г.Вагонова, Нікітіна О.Б. Н.Н.Романюк – Дніпропетровськ: Національний гірничий університет. – 2013. – 11 с.

26. Методичні рекомендації до виконання кваліфікаційних робіт здобувачів першого рівня вищої освіти спеціальності 122 Комп’ютерні науки/ В.В. Спирінцев, П.О. Іщук, О.С. Шевцова; Д : НТУ «Дніпровська політехніка», 2021. – 59 с.

## ЛІСТИНГ ПРОГРАМИ

## A.1 Проект «CryptoParserSdk»

## Файл TestParser.cs

```

using CryptoParserSdk.Models;

namespace CryptoParserSdk;

public static class TestParser<T> where T : CryptoParserAbstract, new()
{
    public static async Task Test()
    {
        var parser = new T();
        Console.WriteLine($"Cryptocurrency Exchange Url:
{parser.CryptocurrencyExchangeUrl}");
        Console.WriteLine($"Config:");
        Console.WriteLine($"\\tRequest rate type: {parser.ParserConfig.RequestRateType}");
        Console.WriteLine($"\\tRequest rate: {parser.ParserConfig.RequestsRate}");
        Console.WriteLine($"\\tPrefix url: {parser.ParserConfig.PrefixUrl}");
        PeriodicTimer timer = new
PeriodicTimer(TimeSpan.FromSeconds((double)parser.ParserConfig.RequestRateType /
parser.ParserConfig.RequestsRate));
        var projects = parser.GetCryptocurrencyList();
        await timer.WaitForNextTickAsync();

        if (parser.ParserConfig.MultiQueryInfoSupport)
            if (projects.Count > 0)
            {
                IEnumerable<MultiQueryCryptocurrencyInfo[]> chunks;
                if (parser.ParserConfig.MultiQueryInfoLimit == 0)
                    chunks = new List<MultiQueryCryptocurrencyInfo[]>()
                    {
                        projects
                            .Select(x => new MultiQueryCryptocurrencyInfo() {
ParamToSearchInfo = x.ParamToSearchInfo, CryptocurrencyInfo = x.CryptocurrencyInfo ??
new CryptocurrencyInfo() })
                            .ToArray()
                    };
                else
                    chunks = projects
                        .Select(x => new MultiQueryCryptocurrencyInfo() { ParamToSearchInfo =
x.ParamToSearchInfo, CryptocurrencyInfo = x.CryptocurrencyInfo ?? new
CryptocurrencyInfo() })
                        .Chunk(parser.ParserConfig.MultiQueryInfoLimit);

                foreach (var chunk in chunks)
                {
                    var infos = parser.GetCryptocurrenciesInfo(chunk.ToList());

                    foreach (var item in infos)
                    {
                        var project = projects.First(x => x.ParamToSearchInfo == item.ParamToSearchInfo);
                        Console.WriteLine($"Project name: {project.Name}");
                    }
                }
            }
    }
}

```



```

        Console.WriteLine($"Project symbol: {project.Symbol}");
        Console.WriteLine($"Project url: {project.ProjectUrl}");
        Console.WriteLine($"Project param: {project.ParamToSearchInfo}");
        Console.WriteLine("Info:");
        var info = item.CryptocurrencyInfo;
        await Console.Out.WriteLineAsync($"Description:
{project.CryptocurrencyInfo.Description}");
        await Console.Out.WriteLineAsync("-----LINKS-----");
        foreach (var link in project.CryptocurrencyInfo.Links)
        {
            await Console.Out.WriteLineAsync($"{{(link.LinkType == 0 ? link.OtherLinkType :
link.LinkType)}}");
            foreach (var url in link.Urls)
                await Console.Out.WriteLineAsync(url);
            await Console.Out.WriteLineAsync("=====");
        }
        await Console.Out.WriteLineAsync("-----");
        await Console.Out.WriteLineAsync("-----PLATFORMS-----");
        foreach (var platform in project.CryptocurrencyInfo.Platforms)
        {
            await Console.Out.WriteLineAsync($"Name: {platform.Name}");
            await Console.Out.WriteLineAsync($"Type: {platform.Type}");
            await Console.Out.WriteLineAsync($"Type: {platform.SmartContract}");
            await Console.Out.WriteLineAsync("=====");
        }
        await Console.Out.WriteLineAsync("-----");
    }
    await timer.WaitForNextTickAsync();
}
}
else
    foreach (var project in projects)
    {
        Console.WriteLine($"Project name: {project.Name}");
        Console.WriteLine($"Project symbol: {project.Symbol}");
        Console.WriteLine($"Project url: {project.ProjectUrl}");
        Console.WriteLine($"Project param: {project.ParamToSearchInfo}");
        Console.WriteLine("Info:");
        project.CryptocurrencyInfo =
parser.GetCryptocurrencyInfo(project.ParamToSearchInfo, project.CryptocurrencyInfo ??
new CryptocurrencyInfo());
        await timer.WaitForNextTickAsync();
        await Console.Out.WriteLineAsync($"Description:
{project.CryptocurrencyInfo.Description}");
        await Console.Out.WriteLineAsync("-----LINKS-----");
        foreach (var link in project.CryptocurrencyInfo.Links)
        {
            await Console.Out.WriteLineAsync($"{{(link.LinkType == 0 ? link.OtherLinkType :
link.LinkType)}}");
            foreach (var url in link.Urls)
                await Console.Out.WriteLineAsync(url);
            await Console.Out.WriteLineAsync("=====");
        }
        await Console.Out.WriteLineAsync("-----");
        await Console.Out.WriteLineAsync("-----PLATFORMS-----");
        foreach (var platform in project.CryptocurrencyInfo.Platforms)
        {
            await Console.Out.WriteLineAsync($"Name: {platform.Name}");
            await Console.Out.WriteLineAsync($"Type: {platform.Type}");
            await Console.Out.WriteLineAsync($"Contract: {platform.SmartContract}");
            await Console.Out.WriteLineAsync("=====");
        }
    }
}

```

```

        await Console.Out.WriteLineAsync("-----");
    }
}
}

```

## Файл CryptoParserAbstract.cs

```

using CryptoParserSdk.Models;

namespace CryptoParserSdk;

public abstract class CryptoParserAbstract
{
    public abstract string CryptocurrencyExchangeUrl { get; }
    public abstract ParserConfig ParserConfig { get; }
    public abstract List<ParsingResult> GetCryptocurrencyList();
    public virtual CryptocurrencyInfo? GetCryptocurrencyInfo(string projectId,
CryptocurrencyInfo info) { return default; }
    public virtual List<MultiQueryCryptocurrencyInfo>?
GetCryptocurrenciesInfo(List<MultiQueryCryptocurrencyInfo> multiInfos) { return
default; }
}

```

## Файл Models/CryptocurrencyInfo.cs

```

namespace CryptoParserSdk.Models;

public class CryptocurrencyInfo
{
    public DateTime Start { get; set; }
    public List<Platform> Platforms { get; set; } = new();
    public string? Description { get; set; }
    public List<Link> Links { get; set; } = new();
}

```

## Файл Models/Link.cs

```

namespace CryptoParserSdk.Models;

public class Link
{
    public LinkType LinkType { get; set; }
    public List<string> Urls { get; set; } = new();
    public string? OtherLinkType { set; get; }
}

public enum LinkType
{
    Other = 0, CoinGecko, CoinMarketCap, Discord, Email, Explorer, Facebook, Instagram,
    LinkedIn, Medium, Quora, Website, SourceCode, Reddit, Slack, TecnicalDoc, Telegram,
    TikTok, Twitch, Twitter, Youtube,
}

```

## Файл Models/MultiQueryCryptocurrencyInfo.cs

```

namespace CryptoParserSdk.Models;

public class MultiQueryCryptocurrencyInfo
{
    public string ParamToSearchInfo { get; set; }
    public CryptocurrencyInfo CryptocurrencyInfo { get; set; }
}

```

## Файл Models/ParserConfig.cs

```
namespace CryptoParserSdk.Models;

public class ParserConfig
{
    public enum RequestRate
    {
        Minute = 60,
        Hour = 3600,
        Day = 86400,
        Week = 604800,
        Month = 2592000
    }
    public string? PrefixUrl { get; set; }
    public RequestRate RequestRateType { get; set; }
    public int RequestsRate { get; set; }
    public object? Custom { get; set; }
    public bool MultiQueryInfoSupport { get; set; } = false;
    public int MultiQueryInfoLimit { get; set; }
}
```

## Файл Models/ParsingResult.cs

```
namespace CryptoParserSdk.Models;
public class ParsingResult
{
    public string Name { get; set; } = string.Empty;
    public string Symbol { get; set; } = string.Empty;
    public string ProjectUrl { get; set; } = string.Empty;
    public string ParamToSearchInfo { get; set; } = string.Empty;
    public CryptocurrencyInfo? CryptocurrencyInfo { get; set; }
}
```

## Файл Models/Platform.cs

```
namespace CryptoParserSdk.Models;

public class Platform
{
    public string? Name { get; set; }
    public string? Type { get; set; }
    public string? SmartContract { get; set; }
}
```

## Файл Extensions/LinkHelper.cs

```
using CryptoParserSdk.Models;

namespace CryptoParserSdk.Extensions;

public static class LinkHelper
{
    private static Dictionary<LinkType, string[]> _linkNames = new Dictionary<LinkType, string[]>()
    {
        { LinkType.CoinGecko, new string[] { "coingecko.com" } },
        { LinkType.CoinMarketCap, new string[] { "coinmarketcap.com" } },
        { LinkType.Discord, new string[] { "discord.gg", "discord.com" } },
        { LinkType.Facebook, new string[] { "facebook.com" } },
    }
}
```

```

{ LinkType.Linkedin, new string[] { "linkedin.com" } },
{ LinkType.Quora, new string[] { "quora.com" } },
{ LinkType.Medium, new string[] { "medium.com" } },
{ LinkType.SourceCode, new string[] { "bitbucket.org", "gitlab.com", "github.com" } },
{ LinkType.Reddit, new string[] { "reddit.com" } },
{ LinkType.Telegram, new string[] { "t.me" } },
{ LinkType.Twitch, new string[] { "twitch.tv" } },
{ LinkType.Twitter, new string[] { "twitter.com", "t.co" } },
{ LinkType.Youtube, new string[] { "youtube.com", "youtu.be" } },
{ LinkType.Explorer, new string[]
{
    "explorer", "tokenview.io", "suiscan.xyz", "blockchain.com", "blockchair.com",
    "etherscan.io", "insight.is", "sochain.com", "tokenview.com", "blockstream.info",
    "blocktrail.com", "oxt.me", "amberdata.io", "blockdozer.com", "bloxy.info",
    "etherchain.org", "bscscan.com", "tronscan.org", "cardanoscan.io", "stellar.expert",
    "teztracker.com", "nimiq.watch", "polkascan.io", "viewblock.io/arweave",
    "flowscan.org", "blockchain.info", "bitupper.com", "trxplorer.io", "bloks.io",
    "nearblocks.io", "dogechain.info", "ethplorer.io", "ton.sh", "tonapi.io",
    "tonmoon.org", "youton.org", "tonscan.org", "ton.app", "gastracker.io", "steexp.com",
    "polygonscan.com", "snowtrace.io", "cronos.org", "mintscan.io", "atomscan.com",
    "eospark.com", "explore.cash", "avascan.info", "aptoscan.com", "apscan.io",
    "tracemove.io", "crypto.org", "cdoscan.com", "trivial.co", "elrondscan.com",
    "elrondmonitor.com", "getblock.io", "hash-hash.info", "confluxscan.io", "tezos.id",
    "tezblock.io", "tzstats.com", "thegraph.com", "iotasear.ch", "neotracker.io",
    "arbiscan.io", "ftmscan.com", "solscan.io", "kavascan.com", "viewblock.io",
    "oasisscan.com", "blockscout.com", "ravencoin.network", "moneroblocks.info",
    "kryfi.com", "btgexp.com", "ordinalswallet.com", "coindaddy.io", "ever.live",
    "tonscan.io", "flatqube.io", "everscan.io", "nanode.co", "bkscan.com",
    "mergechain.com", "secretnodes.com", "steemd.com", "wscan.io", "wavescap.com",
    "omnichest.info", "reefscan.com", "wavesgo.com", "bithomp.com", "axelarscan.io",
    "fsnex.com", "ardorportal.org", "cronoscan.com", "thecelo.com", "solanabeach.io",
    "mccashscan.io", "cryptofresh.com", "bitsharescan.com", "bts.ai", "verge-
    blockchain.info", "tellerscan.com", "prohashing.com", "xchain.io", "clvscan.com",
    "eosflare.io", "eosx.io", "hecoinfo.com", "thevoxel.com", "ttcscan.io", "tronscan.io",
    "zkspace.info", "oxen.observer", "scprime.info", "flarescan.org", "simpleledger.info",
    "blockexperts.com", "nexussoft.io", "neoscan.io", "pocketnet.app", "revdefine.io",
    "beamprivacy.community", "electraprotocol.network", "grinscan.net", "grin-fans.org",
    "presstab.pw", "nulscan.io", "cmttracking.io", "jdcoin.us", "oklink.com",
    "bitcointalk.org", "bismuth.online", "bitinfocharts.com", "gridresearchcorp.com",
    "bchain.info", "chainradar.com", "metaverse.network", "ubiqscan.io", "secretscan.io",
    "aurorascan.dev", "cryptoblock.xyz", "guldenblocks.com", "dactual.com", "pacscan.io",
    "cryptobe.com", "bitswift.network", "ardor.tools", "openchains.info", "mynxt.info",
    "vtrchains.com", "dbixscan.io", "wanscan.org", "htdfscan.com", "openledger.io",
    "khashier.com", "democats.org", "starcoinblock.com", "blockchain.mn", "nxtportal.org",
    "hecochain.io", "intensecoin.com", "cnetscan.com", "hscan.org", "zp.io",
    "paichain.info", "peerplaysdb.com", "boschain.org", "binance.org", "tauhq.com",
    "blockscan.com", "vitaetoken.io", "xptx.io", "nolimitcoin.info", "bleutrade.com",
    "github.com", "chancoin.info", "parkgene.io", "artery-network.io", "3dcstats.net",
    "golosd.com", "moonscan.io", "skb-coin.jp", "turtle.land", "tagcoin.org",
    "msrchain.net", "seeleview.net", "growersintl.com", "xtrabytes.global",
    "plcultima.info", "emaratcoin.info", "wavesdesk.com", "nyzo.co", "wxcoins.info",
    "solana.fm", "revolvercoin.org", "gander.tech", "eostracker.io", "poswallet.com",
    "philosopherstones.org", "zpool.ca", "californium.info", "block-chain.com",
    "minemanic.com", "crowdcoin.site", "chainmapper.com", "eaglepay.io", "nasdacoin.info",
    "qubitcoinexplorer.cc", "tusc.network", "adzcoin.net", "bitvier.com", "cryptoguru.tk",
    "goldbcr.io", "htcblockchain.com", "evrice.com", "catcoinwallets.com",
    "ecnblockchain.com", "elementrem.net", "ridemycar.online", "xrpscan.com",
    "xdc.network", "t.co", "cryptap.us", "proxynode.network", "decent-db.com",
    "gcnchain.com", "cntblockchain.org", "npcoinpool.com", "huntercoin.info",
    "popularcoin.com", "akroma.io", "vechainstats.com", "agenor.network", "opcx.info",
    "tokyocoin.xyz", "arionum.info", "fickschnitzel.pw", "blockexp.com", "c2chain.info",
    "florincoin.info", "netrum.info", "lpool.name", "etherscan.com", "bitclassic.org",

```

"smartchain.cc", "straks.info", "minecoins.online", "frazchain.com", "abcc.com",  
"prcoin.info", "nyan.space", "geysercoin.com", "xenixblockchain.com", "cryptogods.net",  
"altmix.org", "bitcoin-21.com", "daxxcoin.org", "semuxchain.info", "creamchain.info",  
"pioneerchain.com", "fantasygold.network", "vivocoin.net", "blolys.com",  
"hiveblocks.com", "blocktube.net", "sorascan.com", "dividend.cash", "altbet.io",  
"cleanblocks.info", "scorecoin.net", "abjcoin.io", "tradecoinv2.com", "bdtcoin.info",  
"amsterdamblockchain.info", "blockmunch.club", "bitc2.org", "galorecoin.org",  
"regalcoin.info", "fazzcoin.org", "hscscan.com", "xrpl.to", "tyzen.live",  
"steadynode.org", "truevisionofsatoshi.com", "c3xcoin.com", "coinex.net", "tzkt.io",  
"mjcoin.co", "ethersocial.net", "ksfswap.info", "futcoin.org", "bittron.io", "better-  
call.dev", "teloscan.io", "lucent-blockchaindata.com", "zakatexp.com",  
"smartscan.cash", "masternodes.online", "dixi.live", "fargochain.org",  
"etherdelta.com", "tomoscan.io", "lottocoin.info", "uniwscan.com", "evrazdex.org",  
"mintme.com", "briscan.com", "vitescan.io", "humanscoin.info", "404block.net",  
"emscan.io", "qtum.info", "crypto-city.com", "cryptoplerer.com", "dunscan.io",  
"bitaichain.co", "memescan.io", "defiscan.live", "2x2block.space", "rebellious.io",  
"iqchain.io", "daefrom.io", "sproutsbloc.com", "iotexscout.io", "hooscan.com",  
"klaytnfinder.io", "5000bitcoin.com", "bcbscan.io", "ton.cx", "fortune1.money",  
"stacksonchain.com", "miamining.com", "moscan.app", "wemetis.com",  
"harablockchain.net", "hfrco.in", "zenonhub.io", "qitchain.org", "digitalmoneybits.io",  
"bitcoinlvx.com", "lcnxp.com", "xscscan.pub", "kalscan.io", "mixin.one", "e-money.net",  
"weecoins.io", "mystakingwallet.com", "documentchain.org", "zeroscan.io", "hnsan.com",  
"hnsxplorer.com", "ycnxp.com", "zdxplorer.info", "hcxscan.io", "triforcecash.com",  
"hdd.cash", "ping.pub", "avescan.io", "vluchain.info", "addictsx.com", "serey.io",  
"amazonasbit.com", "pmgcoin.io", "jmcblock.com", "iotexscan.io", "bitalgochains.info",  
"pegascoin.com", "aliothcoin.com", "bit.ly", "alloyproject.org", "adevplus.info",  
"lumenscan.io", "yottascan.io", "pcsblocchain.com", "berithscan.com", "brtscan.com",  
"altmco.in", "blockgatorsarmy.info", "veoscan.io", "biticascan.com", "rhypton.com",  
"exzoscan.io", "blacknet.xyz", "ordiscan.com", "blocktonscan.com", "w8io.ru",  
"subscan.io", "cloutcontracts.net", "bccschain.com", "smithpool.net", "alltra.global",  
"kcnxp.com", "tyroblockchain.com", "tmyscan.com", "criptoreal.info", "cyb.ai",  
"wyzthscan.org", "rainz.xyz", "dogz.tech", "ethereumx.xyz", "chaincash.me",  
"minepi.com", "ulxscan.com", "cointool.app", "amingpool.com", "umine.eu",  
"mofowallet.com", "bergco.net", "gdccscan.io", "galleon.live", "relianz.info",  
"fivestarcoint.in", "alpturkcoin.com", "bigdipper.live", "techsharescommunity.com",  
"hospitalcoin.net", "repite.es", "stellarchain.io", "indexchain.org",  
"eosauthority.com", "xinfinscan.com", "chainxplorer.io", "cicscan.com", "ardor.world",  
"bytscan.org", "relictum.pro", "fufiscan.com", "hashscan.io", "eticascan.org",  
"metascan.io", "trontokens.org", "bitindiscan.com", "fulascan.io", "ramascan.com",  
"heliosprotocol.io", "cgc.capital", "enjinx.io", "fitoken.org", "fonscan.io",  
"classicbitcoin.info", "aokscan.com", "ihostmn.com", "ektascan.io", "itsfoto.com",  
"mic3coinproject.com", "rubychain.org", "luckydogpool.com", "rbx.network",  
"referencecoin.co", "trustpad.io", "gstchain.io", "stationcoin.net", "neonscan.org",  
"phoenixplorer.com", "platincoin.info", "uniscan.io", "infiniteblocks.space",  
"tpcscan.com", "qooper.space", "kazuexplore.com", "memescan1.io", "terrascope",  
"traaitchain.com", "visionscan.org", "cubescan.network", "orbis.money",  
"recessioncoin.com", "miascoin.net", "yoscan.io", "freecash.info", "incscan.io",  
"bilbotel.fr", "spock.network", "btcixscan.com", "thinkiumscan.net",  
"tronwatch.market", "layerview.io", "powerbalt.com", "viabtc.com", "moneyonchain.com",  
"helpico.tech", "subgamescan.io", "protonscan.io", "needlecoinpool.eu", "twitter.com",  
"velascan.org", "intelligent.community", "exlscan.com", "bearcoin.net", "infinity-  
economics.org", "vestx.online", "gastroadvisor.com", "dpscan.app", "thxchain.com",  
"xlscan.com", "mixin.space", "atoll.finance", "minascan.org", "iostabc.com",  
"dchainscan.com", "plugchain.network", "stepscan.io", "tuber.build", "terra.credit",  
"tenetscan.io", "santatokens.com", "buysellcoin.org", "oasiscoin.team", "fuzzbawls.pw",  
"redlightscan.finance", "bscan.com", "anyswap.net", "hyperstats.info", "mcoinscan.com",  
"lbank.com", "alcor.exchange", "terafoundation.org", "cmpscan.io", "zeescan.io",  
"speedcryptoblock.com", "xenophyte.com", "filestar.info", "bitcoinreferenceline.com",  
"xrpl.services", "blockchain.coinmarketcap.com", "blockchair", "insight.bitpay.com",  
"live.blockcypher.com", "chainz.cryptoid.info", "smartbit.com.au",  
"xrpcharts.ripple.com", "iota.tanglebay.org", "hubble.figment.network",  
"finder.terra.money", "tracker.icon.foundation", "scan.hecochain.com",

"dashboard.stellar.org", "bch.btc.com", "insight.vecha.in", "hubble.figment.io",  
"app.dragonglass.me", "goalseeker.purestake.io", "scope.klaytn.com", "terra.stake.id",  
"scan.meter.io", "nembex.nem.ninja", "katnip.kaspad.net", "explore.sia.tech",  
"mainnet.decred.org", "moonbase-blockscout.testnet.moonbeam.network",  
"scan.huobichain.com", "dashboard.mainnet.concordium.software", "axelar-  
mainnet.chainode.tech", "finder.kujira.app", "stats.celo.org",  
"blockchain.elastos.org", "45.55.242.48", "scan.platon.network", "explore.vechain.org",  
"mona.chainsight.info", "statemine.statescan.io", "blockbook.groestlcoin.org",  
"groestlsight.groestlcoin.org", "bitciexp.bitcichain.com", "namecoin.webbtc.com",  
"insight.dimensionchain.io", "149.28.91.104", "scan.tomochain.com",  
"creeper.banano.cc", "space.midnightminer.net", "chain.sibcoin.net", "scan.nel.group",  
"browser.credittag.io", "blocks.mix-blockchain.org", "piscan.pchain.org",  
"explore.zenon.network", "scan.nerve.network", "explore.energi.network",  
"explore.signumcoin.ro", "explore.marscoin.org", "live.reddcoin.com",  
"kalkulus.silopool.com", "xplorer.xfccoin.io", "k21.kanon.art",  
"browser.aurorachain.io", "ex.tosblock.com", "jupiter.omlira.com",  
"espv2.miningalts.com", "blocks.malwarechain.com", "explore.duneanalytics.com",  
"121.78.116.108", "scan.idena.io", "yec.safe.trade", "blockchain.gulden.com",  
"insight.garli.co.in", "explore.veforge.com", "orbiter.musicoin.org",  
"sove.cryptoscope.cc", "173.249.13.162", "explore.next.exchange",  
"aurumblocks.cointech.net", "203.128.6.219", "explore.tokesplatform.org",  
"explore.wownero.com", "mainnet.scan.caduceus.foundation", "browser.achain.com",  
"exp.bhpa.io", "comdex.aneka.io", "look.chillvalidation.com", "chain.stealthcoin.com",  
"79.143.186.234", "scc.ccore.online", "chains.trittium.cc", "scan.odinprotocol.io",  
"atlas.phoenixcoin.org", "iquidus.xpcnet.work", "emercoin.mintr.org", "108.61.188.7",  
"185.223.31.170", "144.76.113.28", "testnet.lamden.io", "chain.breakoutcoin.com",  
"104.155.183.25", "transactions.evergreencoin.org", "mutualcoin.dynu.net",  
"chain.quantisnetwork.org", "block.campuscoin.net", "block.ha.cash", "191.101.232.221",  
"be.611.to", "dev.pywaves.org", "explore.safex.io", "45.55.52.85",  
"liverate.tccexchange.org", "blockscout.moonriver.moonbeam.network",  
"info.chainswap.com", "159.69.33.243", "chain.hollywoodcoin.biz", "chain.pepegold.org",  
"217.163.23.222", "microbitcoinorg.github.io", "insight.leocoin.org",  
"asset.burstnation.com", "zenzo.iizzz.net", "mazacoin.thecoin.pw",  
"transactions.sterlingcoin.org", "chain.fair.to", "mtnc.snodo.de",  
"explore.auxilium.global", "insight.quasarcoin.org", "insight.zerocurrency.io",  
"insight.zeromachine.io", "condensate.cryptophi.net", "45.77.214.49",  
"xgs.ccore.online", "ledger.paycoin.com", "crypto.miningalts.com",  
"explore.atccoin.com", "149.28.16.126", "swap.coinscope.cc", "70.83.227.32",  
"ldoge.miningalts.com", "steep.ddclub.org", "steep.overemo.com",  
"sojourn.thecryptochat.net", "be1.centauricoins.info", "104.238.153.140",  
"sphere.iquidus.io", "insight.ritocoin.org", "blockbook.ritocoin.org", "104.238.173.4",  
"chain.phonecoin.space", "91.121.108.101", "chains.sye.host",  
"inflationcoin.thecryptochat.net", "zennies.thecryptochat.net", "158.69.205.238",  
"block.mfcoin.net", "blockchain.nixplatform.io", "insight.parkbyte.com",  
"bplexp.blockpool.io", "remchain.remme.io", "blockchain.bumbacoin.com",  
"162.243.101.66", "blockchain.marijuanacoin.net", "144.172.80.25",  
"beacon.exp.monitorit4.me", "188.226.178.216", "78.153.4.77", "boat.cryptophi.net",  
"dgc.blockr.io", "explore.bit-flowers.com", "block.irishcoin.org", "explorateur-  
qbc.circonference.ca", "chain.projectcoin.net", "45.89.26.10", "5.9.158.101",  
"xuez.donkeypool.com", "mainnet.iop.cash", "blox.slingcoin.rocks", "pool-  
node.hanacoin.com", "block.monacocoin.net", "172.110.9.193", "209.222.30.131",  
"gxx.ccore.online", "66.55.133.82", "144.202.18.54", "beta.wavesplatform.com",  
"93.95.97.96", "178.62.133.174", "blockchain.tpcash.io", "explore.unelmecoin.io",  
"app.bitphantom.io", "explore.beetlecoin.io", "193.70.109.114",  
"insight.axerunners.com", "homeblock.thecryptochat.net", "chain.nem.ninja",  
"nbx.cryptonight.mine.nu", "abe.darkgamex.ch", "212.24.102.92",  
"bitcoinplanet.thecryptochat.net", "grim.reaper.rocks", "piecoin.thecryptochat.net",  
"exp.merebel.org", "45.76.76.105", "blockbook.polispay.org", "exp.deliziaproject.com",  
"bzx.ccore.online", "mri.prenges.rocks", "blocks.jiyo.io", "btb.altcoinwarz.com",  
"chain.cdmcoin.org", "sdp.miningalts.com", "107.170.222.106", "chain.carebit.org",  
"block.ulatech.com", "45.77.143.94", "chain.ragnaproject.io", "go.etitanium.net",  
"95.179.155.62", "bcz.ccore.online", "125.128.182.102", "206.189.160.159",

"block.veltor.org", "bbk.ccore.online", "bbk.overemo.com", "benjiexp.cryptooz.com",  
"cannationcoin.thecryptochat.net", "174.138.74.243", "31.220.50.241",  
"stats.ultranote.org", "aevo.evo.today", "104.238.184.49", "45.32.183.111",  
"bnx.miningalts.com", "explore.tuxtoke.life", "mainnet.hlchain.net", "217.69.14.34",  
"ec2-52-37-136-30.us-west-2.compute.amazonaws.com", "node1.cryptojacks.com",  
"blocks.pennykoin.com", "paxex.dynu.net", "browser.poriot.com",  
"insights.dinerocoin.org", "139.59.163.156", "206.189.140.184", "142.93.162.83",  
"ltb.altcoinwarz.com", "abe.getbitcoininstant.org", "45.32.173.218", "164.132.57.179",  
"blocks.cutcoin.org", "insight.bithereum.network", "mcrn.acc-pool.pw",  
"chains.crevacoin.com", "159.203.112.182", "blockchain.beavercoin.org",  
"pool.agrolifecoin.org", "scan.coredao.org", "192.250.236.182", "167.99.200.1",  
"sperocoin.ddns.net", "blocks.printex.tech", "scan.bitcoinxc.org", "94.176.236.84",  
"znd.ccore.online", "96.44.134.252", "blocks.compcoin.com", "evm.evmos.org",  
"104.223.43.151", "ourcoin.blockexplorer.info", "140.82.4.77", "188.166.182.57",  
"flavorcoin.thecryptochat.net", "pacific.safeseafoodcoin.com",  
"miningpool2.thruhere.net", "my.geekcash.org", "ex.peoplecoin.pw", "162.243.99.178",  
"104.238.171.186", "207.148.126.138", "144.91.87.7", "51.255.6.35", "207.148.1.73",  
"docs.wolk.com", "blog.ethereum.org", "finder.extraterrestrial.money",  
"194.135.92.135", "exp.uralscoin.info", "95.179.153.89", "5.189.162.110",  
"wallet.antimonycoin.com", "185.141.61.248", "blocks.rhfcoin.com", "95.181.230.26",  
"ex.unrc.eu", "193.200.241.196", "fly.dnsalias.com", "chain.uncoin.org",  
"138.197.18.171", "bitcf.mintr.org", "explore.xwinner.io", "blockchain.live9coin.net",  
"xmon.blockexplorer.info", "95.179.194.226", "chain.t-powercoin.com",  
"cropcoin.blockexplorer.info", "endo.overemo.com", "endo.blockrex.info", "66.42.72.101",  
"dxc.overemo.com", "esc.elastos.io", "63.142.255.39", "46.101.21.105",  
"blockchain.bitcoinx.space", "scan.parshrax.com", "api.bscscan.chttps",  
"topazcoin.thecryptochat.net", "blockscout.aqua.signal2noi.se", "e.axiomcoin.xyz",  
"block.hebeblock.com", "scan.octium.io", "hodl.amit.systems", "chain.paws.fund",  
"46.173.218.227", "namocoin.dynns.com", "45.63.111.121", "144.202.54.73",  
"176.123.2.135", "deliv.deliziaproject.com", "block.aos.plus",  
"explore.zenithcoin.net", "207.148.70.103", "scan.microvisionchain.com",  
"95.179.212.95", "207.126.164.144", "cchblocks.counos.org", "scan.bitonechain.com",  
"operator.adshares.net", "mainnet.bityuan.com", "x2.altcoinsteps.com",  
"explore.daikicoin.org", "scan.compverse.io", "browser.ofbank.com",  
"blocks.aurumcrypto.gold", "blocks.blackbill.info", "achilles.lchain.cc",  
"block.aib.one", "block.ggcash.live", "gkiscan.hgeek1.com", "68.183.65.69",  
"insight.eacoin.io", "scan.kingaruru.com", "insight.kevacoin.org", "app.onstacks.com",  
"scan.hupayx.com", "nebula.lyra.live", "block.cdy.one", "dynex.dyndns.org",  
"owo.ccore.online", "forkdelta.github.io", "45.63.91.153", "block.russellcoin.com",  
"tracker.paw.digital", "scan.mdukey.org", "xdc.blocksscan.io", "95.179.147.253",  
"scan.luniverse.io", "neuro.enecuum.com", "68.183.203.170", "bx.mochimo.org",  
"eca.ccore.online", "mainnet.qiblockchain.online", "scan.zktube.io",  
"artax.blockexplorer.info", "149.248.63.235", "explore.myce.world", "107.191.39.190",  
"forum.wavesplatform.com", "78.141.199.29", "ex.soldo.in", "206.189.128.154",  
"testnet.alphabetnetwork.org", "sidescan.luniverse.io", "mainnet.viteview.xyz",  
"116.203.105.245", "expl.xgamingup.live", "scan.rangersprotocol.com", "5.181.49.233",  
"insight.solbit.tech", "scan.alaya.network", "asa.cryptoscope.io",  
"hole.moonrabbit.com", "boxy.blockexplorer.info", "scan.biut.io", "scan.coinbiten.com",  
"azzr.urcryptodepot.com", "scan.oasys.games", "3.230.179.87", "bid.cryptoscope.cc",  
"173.212.196.144", "scan.apitool.io", "block.phuquoc.dog", "fln.ccore.online",  
"bex.zilbercoin.space", "54.37.233.45", "blockchain.bitcoinpositive.org",  
"explore.bitfree.vip", "exp.cashberycoin.com", "45.94.209.55", "bgl.bitaps.com",  
"polkadot.js.org", "bit.ccore.online", "block.bitvote.one", "livenet.xrpl.org",  
"ccxxblocks.counos.org", "188.120.224.54", "beex.zendesk.com", "coin.infinipay.co",  
"data.dcrn.xyz", "45.9.191.168", "insight.tcoin.eu", "47.75.96.91", "46.101.231.40",  
"btca.cryptoscope.io", "80.211.196.242", "block.barc.io", "esr-ex.esrpos.com",  
"chain.frostbytecoin.io", "exp.cryptohashtank.com", "chain.futurexco.com",  
"blockchain.umi.top", "state.jingtum.com", "lkloon123.ddns.net", "chain.cspn.io",  
"149.28.229.20", "blt.cryptoscope.cc", "aioexp.mine2.live", "polkadot.acuity.social",  
"207.180.216.126", "scout.ech.network", "chain.hashbit.org", "dex.globiance.com",  
"block.epchian.com", "info.viz.plus", "cpb.cricket.foundation", "chain.gwaycoin.org",  
"explore.gac.one", "45.76.202.188", "explorateur.heptafranc.com",

```

"blockchain.prizm.space", "45.63.100.148", "149.28.170.184", "157.230.216.65",
"blockchain.bened.cc", "45.76.39.254", "scan.nachain.org", "discover.konjungate.net",
"invoice.cryptonode.online", "185.177.59.35", "142.93.240.228", "80.211.189.203",
"insight.kotocoin.info", "blockchain.proeliocoin.com", "network.protoncoin.org",
"sera.overemo.com", "block.superbtc.org", "chain.thetaspere.com",
"stacsbrowser.gsxgroup.global", "scan.wegochain.io", "jaspersight.jasperpay.io",
"scan.stablepure.com", "chain.hashbro.org", "208.85.21.92", "207.148.70.241",
"ratcoin.rektfreak.io", "taycan-evmscan.hupayx.io", "nynode.electroneropulse.org",
"18.219.212.172", "gls.glsscan.co", "ex.tgichain.com", "scan.saintlignenft.com",
"scan.pulsechain.com", "beacon.pulsechain.com", "159.89.89.240", "scan.rei.network",
"blocks.dstra.io", "95.179.133.118", "scan.metablockchain.id", "be.stx.nl",
"skeinetwork-staging.skeinetwork.dev", "172.105.213.162", "chain.evolution-
network.org", "blocks.myus-coin.com", "scan.zeronelabs.org", "sat3.nerdlabs001.com",
"chain.sprintpay.net", "zettel.dashnetwork.info", "zettel.blockmole.info",
"e.alphacup.io", "avian.cryptoscope.io", "bitok.steepool.com", "explore.smartx.one",
"browser.tongtongchain.io", "ccc.mining4people.com", "testnet.wbtc-chain.com",
"blocks.blurtwallet.com", "insight.bdcashprotocol.com", "live.block.koinon.cloud",
"scan.chain.pixie.xyz", "explorador.petro.gob.ve", "block.tschain.top",
"scope.mooichain.io", "scan.carbon.network", "18.219.109.235", "explore.brixco.in",
"138.68.143.185", "obtc.althashers.com", "blockchain.plusonecoin.org",
"wallet.pandoracash.com", "evm.findorascan.io", "155.138.216.103",
"explore.odinblockchain.org", "regen.aneka.io", "browser.zorochain.com", "51.38.71.12",
"45.79.21.87", "exploreblockchain.clore.ai", "crome.cryptoscope.io",
"blocks.cash2.org", "blocks.mocha.network", "5.196.67.100", "38.242.145.127",
"explo.andes-coin.com", "v1.gwscan.com", "108.160.128.47", "scan.orai.io",
"volrscan.volare.network", "66.42.52.30", "scan.neatio.net", "lilli-
new.cryptoscope.io", "157.245.164.19", "blockchain.mavro.org", "ecoin.03c8.net",
"45.32.175.139", "45.77.13.137", "whivepool.cointest.com", "blocks.flapxcoin.com",
"138.68.44.164", "vapor.blockmeta.com", "halloweencoin.thecryptochat.net",
"5.189.166.55", "freakchain.freakhouse.dev", "scan.fibochain.org",
"blockchain.megacrypton.com", "poolparty.stride.zone", "scan.zenithchain.co",
"chain.anwang.com", "shroud.mastermine.eu", "skull.servep2p.com",
"xplere.satoverse.io", "45.63.77.45", "xscp.bot.nu", "blockchain.aunite.com",
"scan.nollar.org", "info.apeswap.finance", "208.167.249.234", "66.113.234.71",
"chain.fortuneum.io", "info-14792.medium.com", "eth.elastos.io", "217.69.7.182",
"neo.alchemint.io", "81.25.161.17", "block.piratecash.net", "info.mercibq.com",
"46.101.246.110", "explore.stargaze.zone", "subnets.avax.network", "server.msp-
coin.com"
}
}
};

```

```

public static void SortLinks(this List<Link> list, params string[] links)
{
    foreach (string link in links)
    {
        bool isFinded = false;
        foreach (var kvp in _linkNames)
        {
            foreach (string host in kvp.Value)
            {
                if (link.Contains(host))
                {
                    list.AddLink(link, kvp.Key);
                    isFinded = true;
                    break;
                }
            }
            if (isFinded)
                break;
        }
        if (!isFinded)
    }
}

```



```

        list.AddLink(link);
    }
}

public static void AddLink(this List<Link> list, string url, LinkType type = 0, string
otherType = "Unknown")
{
    if (string.IsNullOrEmpty(url))
        return;

    Link? linkObj;

    if (type == 0)
        linkObj = list.FirstOrDefault(x => x.LinkType == type && x.OtherLinkType ==
otherType);
    else
        linkObj = list.FirstOrDefault(x => x.LinkType == type);

    if (linkObj == null)
    {
        var newLink = new Link();
        newLink.Urls.Add(url);
        newLink.LinkType = type;
        if (type == 0)
            newLink.OtherLinkType = otherType;
        list.Add(newLink);
    }
    else
        if (!linkObj.Urls.Contains(url))
            linkObj.Urls.Add(url);
}

public static void AddLinks(this List<Link> list, LinkType type, params string[] urls)
{
    if (urls == null || urls.Length == 0)
        return;
    foreach (var url in urls)
    {
        if (string.IsNullOrEmpty(url))
            continue;
        AddLink(list, url, type);
    }
}
}
}

```

## A.2 Проект «NewCryptoParser»

### Файл Program.cs

```

using Microsoft.EntityFrameworkCore;
using NewCryptoParser.Services;
using NLog;
using NLog.Web;

var logger =
LogManager.Setup().LoadConfigurationFromAppSettings().GetCurrentClassLogger();
logger.Debug("init main");

try
{
    var builder = WebApplication.CreateBuilder(args);

```

```

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddParserManager();
builder.Services.AddNewCryptocurrencyProjectManager();
builder.Services.AddHostedService<FileManagerService>();
builder.Logging.ClearProviders();
builder.Services.AddDbContext<NewCryptocurrencyDbContext>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));
builder.Host.UseNLog();
var app = builder.Build();
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
}
catch (Exception exception)
{
    logger.Error(exception, "Stopped program because of exception");
    throw;
}
finally
{
    LogManager.Shutdown();
}

```

## Файл CodeCompiler.cs

```

using Microsoft.CodeAnalysis;
using Microsoft.CodeAnalysis.CSharp;
using Microsoft.CodeAnalysis.Emit;
using NewCryptoParser.Exceptions;
using System.Reflection;

namespace NewCryptoParser;

internal class CodeCompiler
{
    private static List<MetadataReference> _references = new List<MetadataReference>();

    public static T CompileCodeAndGetObject<T>(string code) where T : class
    {
        SyntaxTree syntaxTree = CSharpSyntaxTree.ParseText(code);

        CSharpCompilation compilation = CSharpCompilation.Create(
            Path.GetRandomFileName(),
            syntaxTrees: new[] { syntaxTree },
            references: _references,
            options: new CSharpCompilationOptions(OutputKind.DynamicallyLinkedLibrary));

        using var ms = new MemoryStream();

        EmitResult result = compilation.Emit(ms);
        if (!result.Success)
        {
            IEnumerable<Diagnostic> failures = result.Diagnostics.Where(diagnostic =>

```

```

        diagnostic.IsWarningAsError ||
        diagnostic.Severity == DiagnosticSeverity.Error);

    var exceptions = new List<Exception>();
    foreach (Diagnostic diagnostic in failures)
    {
        exceptions.Add(new Exception(diagnostic.GetMessage()));
    }
    throw new CompilerException(exceptions.ToArray());
}

ms.Seek(0, SeekOrigin.Begin);

Assembly assembly = Assembly.Load(ms.ToArray());
T? obj = null;
foreach (var item in assembly.GetTypes())
{
    try
    {
        obj = Activator.CreateInstance(item) as T;
        if (obj is not null)
            break;
    }
    catch { } //Trying to create first suitable type
}
if (obj == null)
    throw new Exception("Cannot find class");
return obj;
}
static CodeCompiler()
{
    loadDefaultReferences();
}
private static void loadDefaultReferences()
{
    var assemblies = AppContext.GetData("TRUSTED_PLATFORM_ASSEMBLIES");
    if (assemblies != null)
        foreach (var r in ((string)assemblies).Split(Path.PathSeparator))
            _references.Add(MetadataReference.CreateFromFile(r));
    else
        throw new Exception();
}
}
}

```

## Файл ServiceProviderExtensions.cs

```

using NewCryptoParser.Abstract;
using NewCryptoParser.Services;

public static class ServiceProviderExtensions
{
    public static void AddParserManager(this IServiceCollection services)
    {
        services.AddSingleton<IParserManager, ParserManagerService>();
    }

    public static void AddNewCryptocurrencyProjectManager(this IServiceCollection
services)
    {
        services.AddSingleton<INewCryptocurrencyProjectManager,
NewCryptocurrencyProjectManagerService>();
    }
}

```

```
}  
}
```

## Файл nlog.config

```
<?xml version="1.0" encoding="utf-8" ?>  
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  autoReload="true"  
  internalLogLevel="Info"  
  internalLogFile="c:\temp\internal-nlog-AspNetCore.txt">  
  <extensions>  
    <add assembly="NLog.Web.AspNetCore"/>  
  </extensions>  
  <targets>  
    <target xsi:type="File" name="allfile" fileName="c:\temp\nlog-AspNetCore-all-  
    ${shortdate}.log"  
      layout="${longdate}|${event-  
properties:item=EventId:whenEmpty=0}|${level:uppercase=true}|${logger}|${message}  
    ${exception:format=tostring}" />  
    <target xsi:type="File" name="ownFile-web" fileName="c:\temp\nlog-AspNetCore-own-  
    ${shortdate}.log"  
      layout="${longdate}|${event-  
properties:item=EventId:whenEmpty=0}|${level:uppercase=true}|${logger}|${message}  
    ${exception:format=tostring}|url: ${aspnet-request-url}|action: ${aspnet-mvc-  
action}|${callsite}" />  
    <target xsi:type="Console" name="ParserLog"  
layout="${longdate}|${level:uppercase=true}|${logger}| ${message}  
    ${exception:format=tostring}" />  
    <target xsi:type="Console" name="lifetimeConsole"  
layout="${MicrosoftConsoleLayout}" />  
  </targets>  
  <rules>  
    <logger name="*" minlevel="Trace" writeTo="allfile" />  
    <logger name="Microsoft.Hosting.Lifetime" minlevel="Info" writeTo="lifetimeConsole,  
ownFile-web" final="true" />  
    <logger name="Microsoft.*" maxlevel="Info" final="true" />  
    <logger name="System.Net.Http.*" maxlevel="Info" final="true" />  
    <logger name="NewCryptoParser.Services.ParserManagerService" minlevel="Info"  
final="true" writeTo="ParserLog"/>  
    <logger name="NewCryptoParser.Services.FileManagerService" minlevel="Info"  
final="true" writeTo="ParserLog"/>  
    <logger name="*" minlevel="Trace" writeTo="ownFile-web" />  
  </rules>  
</nlog>
```

## Файл appsettings.json

```
{  
  "AllowedHosts": "*",  
  "ConnectionStrings": {  
    "DefaultConnection":  
"Server=(localdb)\mssqllocaldb;Database=NewCryptocurrencyProjects;Trusted_Connection=T  
true;MultipleActiveResultSets=true"  
  }  
}
```

## Файл Services/FileManagerService.cs

```
using NewCryptoParser.Abstract;  
using NewCryptoParser.Exceptions;
```

```

namespace NewCryptoParser.Services;

public class FileManagerService : BackgroundService
{
    private readonly ILogger<FileManagerService> _logger;
    private readonly IServiceProvider _serviceProvider;
    private readonly IConfigurationProvider _configurationProvider;
    private string parsersFolderPath =>
        Environment.CurrentDirectory +
        Path.DirectorySeparatorChar +
        "Parsers";
    protected override async Task ExecuteAsync(CancellationToken stoppingToken)
    {
        var files = Directory.GetFiles(parsersFolderPath).Where(file =>
file.EndsWith(".cs"));
        foreach (var file in files)
        {
            try
            {
                var _parserManager = _getParserManager();
                using StreamReader sr = new StreamReader(file);
                _parserManager.AddParser(await sr.ReadToEndAsync(),
Path.GetFileNameWithoutExtension(file));
                _logger.LogInformation($"Parser [{file}] successfully loaded");
            }
            catch (ParserAlreadyExist)
            {
                _logger.LogWarning($"File [{file}] added. Parser exist");
            }
            catch (CompilerException ex)
            {
                _logger.LogError(ex, $"Compilation error occurred during auto-add file
[{file}]");
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"An error occurred during auto-add file [{file}]");
            }
        }

        _logger.LogInformation("File manager service started.");
        using var _watcher = new FileSystemWatcher(parsersFolderPath);
        if (!Directory.Exists(parsersFolderPath))
            Directory.CreateDirectory(parsersFolderPath);

        _watcher.Created += (o, e) =>
        {
            try
            {
                _logger.LogDebug($"FileSystemWatcher.Created [{e.FullPath}]");
                if (!e.FullPath.EndsWith(".cs"))
                    return;
                WaitForFile(e.FullPath);
                var _parserManager = _getParserManager();
                using StreamReader sr = new StreamReader(e.FullPath);
                _parserManager.AddParser(sr.ReadToEnd(),
Path.GetFileNameWithoutExtension(e.FullPath));
            }
            catch (CompilerException ex)
            {
                _logger.LogError(ex, $"Error compiling [{e.FullPath}] file.");
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while adding a parser from file
[{e.FullPath}].");
        }
    };
    _watcher.Deleted += (o, e) =>
    {
        _logger.LogDebug($"FileSystemWatcher.Deleted [{e.FullPath}]");
        try
        {
            if (!e.FullPath.EndsWith(".cs"))
                return;
            var _parserManager = _getParserManager();
            _parserManager.RemoveParser(Path.GetFileNameWithoutExtension(e.FullPath));
        }
        catch (Exception ex)
        {
            _logger.LogError(ex,
                $"An error occurred while uninstalling the parser due to deleting file
[{e.FullPath}].");
        }
    };

    _watcher.EnableRaisingEvents = true;

    while (!stoppingToken.IsCancellationRequested)
    {
        await Task.Delay(200, stoppingToken);
    }
    _logger.LogWarning("File manager service stopped.");
    await Task.CompletedTask;
}

private IParserManager? _getParserManager()
{
    using var scope = _serviceProvider.CreateScope();
    return scope.ServiceProvider.GetService<IParserManager>();
}

private bool IsFileReady(string filename)
{
    try
    {
        _logger.LogDebug($"Waiting for file [{filename}]");
        using (FileStream inputStream = File.Open(filename, FileMode.Open,
FileAccess.Read, FileShare.None))
            return inputStream.Length > 0;
    }
    catch (Exception)
    {
        return false;
    }
}

private void WaitForFile(string filename)
{
    while (!IsFileReady(filename)) { }
}

public FileManagerService(ILogger<FileManagerService> logger, IServiceProvider
serviceProvider)

```

```

    {
        _logger = logger;
        _serviceProvider = serviceProvider;
    }
}

```

### Файл Services/NewCryptocurrencyDbContext.cs

```

using Microsoft.EntityFrameworkCore;
using NewCryptoParser.Models;
using Newtonsoft.Json;
using System.Diagnostics.CodeAnalysis;

namespace NewCryptoParser.Services;

public class NewCryptocurrencyDbContext : DbContext
{
    public NewCryptocurrencyDbContext(DbContextOptions<NewCryptocurrencyDbContext>
options) : base(options)
    {
        Database.EnsureCreated();
    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<NewCryptocurrencyProject>().Property(x => x.Info).HasConversion
(
            x => JsonConvert.SerializeObject(x),
            x => JsonConvert.DeserializeObject<NewCryptocurrencyProjectInfo>(x) ??
new()
        );
    }
    [NotNull]
    public DbSet<NewCryptocurrencyProject> NewCryptocurrencyProjects { get; set; }
}

```

### Файл Services/NewCryptocurrencyProjectManagerService.cs

```

using NewCryptoParser.Abstract;
using NewCryptoParser.Models;
using System.Collections.Concurrent;

namespace NewCryptoParser.Services;

public class NewCryptocurrencyProjectManagerService : INewCryptocurrencyProjectManager
{
    private readonly IServiceProvider _provider;
    private readonly ILogger<NewCryptocurrencyProjectManagerService> _logger;
    private ConcurrentQueue<(string parserName, CryptoParserSdk.Models.ParsingResult
project)> _buffer;

    public void AddNewProjects(string parserName,
List<CryptoParserSdk.Models.ParsingResult> projects)
    {
        foreach (var project in projects)
        {
            _buffer.Enqueue((parserName, project));
        }
    }

    public IEnumerable<NewCryptocurrencyProject> GetProjects()
    {

```

```

        var _context = getDbService();
        return _context.NewCryptocurrencyProjects;
    }

    public
    NewCryptocurrencyProjectManagerService(ILogger<NewCryptocurrencyProjectManagerService>
    logger, IServiceProvider provider)
    {
        _provider = provider.CreateScope().ServiceProvider;
        _logger = logger;
        _buffer = new ConcurrentQueue<(string parserName,
    CryptoParserSdk.Models.ParsingResult project)>();
        new Thread(() =>
        {
            while (true)
            {
                try
                {
                    if (!_buffer.IsEmpty)
                    {
                        _logger.LogDebug("Creating db context");
                        var _context = getDbService();
                        while (!_buffer.IsEmpty)
                        {
                            _buffer.TryDequeue(out var data);
                            dataProcessor(data.parserName, data.project, _context);
                        }
                        _context.SaveChanges();
                        _context.Dispose();
                    }
                }
                catch (Exception ex)
                {
                    _logger.LogError(ex, "");
                }
                Thread.Sleep(100);
            }
        }).Start();
    }

    private void dataProcessor(string exchangeUrl, CryptoParserSdk.Models.ParsingResult
    project, NewCryptocurrencyDbContext _context)
    {
        var _projs = _context?.NewCryptocurrencyProjects;
        if (_projs == null)
        {
            _logger.LogError("List of new projects is null");
            return;
        }
        var finded = _projs.FirstOrDefault(p =>
        p.ProjectName.ToLower() == project.Name.ToLower() &&
        p.ProjectSymbol.ToLower() == project.Symbol.ToLower());

        //NewCryptocurrencyProjectInfo info = project.CryptocurrencyInfo as
    NewCryptocurrencyProjectInfo ?? new NewCryptocurrencyProjectInfo();
        NewCryptocurrencyProjectInfo info = new NewCryptocurrencyProjectInfo()
        {
            Links = project.CryptocurrencyInfo.Links,
            Platforms = project.CryptocurrencyInfo.Platforms,
            Start = project.CryptocurrencyInfo.Start,
            Description = project.CryptocurrencyInfo.Description,
            ExchangeUrl = exchangeUrl,
        }
    }

```



```

        ProjectUrl = project.ProjectUrl
    };
    if (finded == null)
    {
        _context.NewCryptocurrencyProjects.Add(new NewCryptocurrencyProject()
        {
            ExchangeUrl = exchangeUrl,
            ProjectName = project.Name,
            ProjectSymbol = project.Symbol,
            Info = info,
            ParseTime = DateTime.Now
        });
    }
}

private NewCryptocurrencyDbContext getDbService()
{
    return
_provider.CreateScope().ServiceProvider.GetService<NewCryptocurrencyDbContext>(!);
}
}

```

## Файл Services/ParserManagerService.cs

```

using CryptoParserSdk;
using NewCryptoParser.Abstract;
using System.Collections.Concurrent;
using NewCryptoParser.Models;
using CryptoParserSdk.Models;
using NewCryptoParser.Exceptions;

namespace NewCryptoParser.Services;

public class ParserManagerService : IParserManager
{
    private ConcurrentDictionary<string, CryptoParserScheduledTask> parsers;
    private readonly ILogger<ParserManagerService> _logger;
    private readonly INewCryptocurrencyProjectManager _projectManager;
    public void AddParser(string code, string name)
    {
        if (parsers.ContainsKey(name))
            throw new ParserAlreadyExist();
        bool successAdd = false;
        try
        {
            successAdd = parsers.TryAdd(name, createCryptoParserScheduledTask(code, name));
        }
        catch (Exception ex)
        {
            throw new ParserAddException("Parser contains error", ex);
        }
        if (!successAdd)
            throw new ParserAddException("ConcurrentDictionary.TryAdd() return false");
        _logger.LogInformation($"Parser [{name}] added");
    }

    public CryptoParserScheduledTask? GetParser(string name)
    {
        var successGet = parsers.TryGetValue(name, out var parser);
        if (!successGet)
            throw new ParserGetException("ConcurrentDictionary.TryGet() return false");
        return parser;
    }
}

```

```

    }

    public void RemoveParser(string name)
    {
        var successRemove = parsers.TryRemove(name, out var parser);
        if (!successRemove)
            throw new ParserRemovalException("ConcurrentDictionary.TryRemove() return
false");
        parser.Dispose();
        _logger.LogInformation($"Parser [{name}] removed");
    }

    private CryptoParserScheduledTask createCryptoParserScheduledTask(string code, string
name)
    {
        var cryptoTask = new CryptoParserScheduledTask();
        cryptoTask.CancellationTokenSource = new CancellationTokenSource();
        cryptoTask.CryptoParser =
CodeCompiler.CompileCodeAndGetObject<CryptoParserAbstract>(code);

        cryptoTask.PeriodicTask = new Task(async _ =>
        {
            var _name = name;
            try
            {
                var _parser = cryptoTask.CryptoParser;
                TimeSpan interval =
                    TimeSpan.FromSeconds((double)_parser.ParserConfig.RequestRateType /
                    _parser.ParserConfig.RequestsRate);

                PeriodicTimer timer = new PeriodicTimer(interval);

                _logger.LogDebug($"[[_name]] Task started");

                var _localProjectsRepository =
                _parser.GetCryptocurrencyList().SkipLast(1).ToList();
                await timer.WaitForNextTickAsync();
                while
                    (!cryptoTask.CancellationTokenSource.IsCancellationRequested &&
                    await timer.WaitForNextTickAsync())
                {
                    try
                    {
                        _logger.LogDebug($"[[_name]] requesting data");
                        var projects = _parser.GetCryptocurrencyList();

                        var newProjects = CheckProjects(projects, _localProjectsRepository);

                        if (_parser.ParserConfig.MultiQueryInfoSupport)
                        {
                            if (newProjects.Count > 0)
                            {
                                IEnumerable<MultiQueryCryptocurrencyInfo[]> chunks;
                                if (_parser.ParserConfig.MultiQueryInfoLimit == 0)
                                    chunks = new List<MultiQueryCryptocurrencyInfo[]>()
                                    {
                                        newProjects
                                            .Select(x => new MultiQueryCryptocurrencyInfo()
{ ParamToSearchInfo = x.ParamToSearchInfo, CryptocurrencyInfo = x.CryptocurrencyInfo ??
new CryptocurrencyInfo() })
                                            .ToArray()
                                    };
                            }
                        }
                    }
                }
            }
            catch { }
        });
    }

```

```

else
    chunks = newProjects
        .Select(x => new MultiQueryCryptocurrencyInfo() {
ParamToSearchInfo = x.ParamToSearchInfo, CryptocurrencyInfo = x.CryptocurrencyInfo ??
new CryptocurrencyInfo() })
        .Chunk(_parser.ParserConfig.MultiQueryInfoLimit);

    foreach (var chunk in chunks)
    {
        await timer.WaitForNextTickAsync();
        var infos = _parser.GetCryptocurrenciesInfo(chunk.ToList());
        foreach (var info in infos)
            newProjects.First(x => x.ParamToSearchInfo ==
info.ParamToSearchInfo).CryptocurrencyInfo = info.CryptocurrencyInfo;
    }
}
else
{
    foreach (var newProject in newProjects)
    {
        if (newProject.CryptocurrencyInfo is null)
            newProject.CryptocurrencyInfo = new CryptocurrencyInfo();
        var info = _parser.GetCryptocurrencyInfo(newProject.ParamToSearchInfo,
newProject.CryptocurrencyInfo);
        if (info != null)
            await timer.WaitForNextTickAsync();
    }
}
if (newProjects.Count > 0)
{
    if (!string.IsNullOrEmpty(_parser.ParserConfig.PrefixUrl))
        foreach (var newProject in newProjects)
            newProject.ProjectUrl = newProject.ProjectUrl.Insert(0,
_parser.ParserConfig.PrefixUrl);
    _localProjectsRepository = projects;
    _logger.LogInformation($"[{{_name}}] Found {newProjects.Count} new
project(-s)");
    _projectManager.AddNewProjects(_parser.CryptocurrencyExchangeUrl,
newProjects);
}
catch (Exception ex)
{
    _logger.LogError(new ParserRuntimeException(ex), $"Parser error
[{{_name}}]");
}
}
}
catch (Exception ex)
{
    _logger.LogError(ex, $"[{{_name}}]");
}
if (!cryptoTask.CancellationTokenSource.IsCancellationRequested)
    _logger.LogWarning($"[{{_name}}] Task stopped");
}, cryptoTask.CancellationTokenSource);
cryptoTask.PeriodicTask.Start();
return cryptoTask;

static List<ParsingResult> CheckProjects(
    List<ParsingResult> parsingResults,
    List<ParsingResult> repository

```

```

    )
    {
        var newProjects = new List<ParsingResult>();
        foreach (var project in parsingResults)
            if (repository.FirstOrDefault(p => p.ProjectUrl == project.ProjectUrl) == null)
                newProjects.Add(new()
                {
                    Name = project.Name,
                    Symbol = project.Symbol,
                    ProjectUrl = project.ProjectUrl,
                    ParamToSearchInfo = project.ParamToSearchInfo,
                    CryptocurrencyInfo = project.CryptocurrencyInfo
                });
        return newProjects;
    }
}

public ParserManagerService(
    ILogger<ParserManagerService> logger,
    INewCryptocurrencyProjectManager projectManager
)
{
    _logger = logger;
    _projectManager = projectManager;
    parsers = new ConcurrentDictionary<string, CryptoParserScheduledTask>();
}
}

```

## Файл Models/CryptoParserScheduledTask.cs

```

using CryptoParserSdk;

namespace NewCryptoParser.Models;

public class CryptoParserScheduledTask : IDisposable
{
    public CryptoParserAbstract CryptoParser { get; set; }
    public CancellationTokenSource CancellationTokenSource { get; set; }
    public Task PeriodicTask { get; set; }

    public void Dispose()
    {
        CryptoParser = null;
        GC.Collect();
        CancellationTokenSource.Cancel();
        PeriodicTask.Dispose();
        CancellationTokenSource.Dispose();
    }

    ~CryptoParserScheduledTask()
    {
        Dispose();
    }
}

```

## Файл Models/NewCryptocurrencyProject.cs

```

using System.ComponentModel.DataAnnotations;

namespace NewCryptoParser.Models;

public class NewCryptocurrencyProject

```

```

{
    [Key]
    public int Id { get; set; }
    public string ExchangeUrl { get; set; } = string.Empty;
    public string ProjectName { get; set; } = string.Empty;
    public string ProjectSymbol { get; set; } = string.Empty;
    public DateTime ParseTime { get; set; }
    public NewCryptocurrencyProjectInfo Info { get; set; } = new();
}

```

### **Файл Models/NewCryptocurrencyProjectInfo.cs**

```

using System.ComponentModel.DataAnnotations;

namespace NewCryptoParser.Models;

public class NewCryptocurrencyProjectInfo : CryptoParserSdk.Models.CryptocurrencyInfo
{
    [Key]
    public int Id { get; set; }
    public string ExchangeUrl { get; set; } = string.Empty;
    public string ProjectUrl { get; set; } = string.Empty;
}

```

### **Файл Exceptions/CompilerException.cs**

```

namespace NewCryptoParser.Exceptions;

public class CompilerException : AggregateException
{
    public CompilerException(params Exception[] innerExceptions) : base(innerExceptions)
    { }
}

```

### **Файл Exceptions/ParserAddException.cs**

```

namespace NewCryptoParser.Exceptions;

public class ParserAddException : Exception
{
    public ParserAddException() { }
    public ParserAddException(string message) : base(message) { }
    public ParserAddException(string message, Exception ex) : base(message, ex) { }
}

```

### **Файл Exceptions/ParserAlreadyExist.cs**

```

namespace NewCryptoParser.Exceptions;

public class ParserAlreadyExist : Exception { }

```

### **Файл Exceptions/ParserGetException.cs**

```

namespace NewCryptoParser.Exceptions;

public class ParserGetException : Exception
{
    public ParserGetException() { }
    public ParserGetException(string message) : base(message) { }
}

```

## Файл Exceptions/ParserRemovalException.cs

```
namespace NewCryptoParser.Exceptions;

public class ParserRemovalException : Exception
{
    public ParserRemovalException() { }

    public ParserRemovalException(string message) : base(message) { }
}
```

## Файл Exceptions/ParserRuntimeException.cs

```
namespace NewCryptoParser.Exceptions;

public class ParserRuntimeException : Exception
{
    public ParserRuntimeException(Exception ex) : base(ex.Message, ex) { }
}
```

## Файл Exceptions/TooManyAttemptsException.cs

```
namespace NewCryptoParser.Exceptions;

public class TooManyAttemptsException : Exception
{
    public TooManyAttemptsException(string? message) : base(message) { }
}
```

## Файл Controllers/CryptocurrencyController.cs

```
using Microsoft.AspNetCore.Mvc;
using NewCryptoParser.Abstract;

namespace NewCryptoParser.Controllers;

[ApiController]
[Route("[controller]")]
public class CryptocurrencyController : ControllerBase
{
    private readonly ILogger<CryptocurrencyController> _logger;
    private readonly INewCryptocurrencyProjectManager _projectManager;

    public CryptocurrencyController(ILogger<CryptocurrencyController> logger,
    INewCryptocurrencyProjectManager projectManager)
    {
        _projectManager = projectManager;
        _logger = logger;
    }

    [HttpGet("GetLatest")]
    public IActionResult GetLatest(int count = 1)
    {
        if (count < 1)
            return UnprocessableEntity("Count must be 1 or greater");
        if (count == 1)
            return Ok(_projectManager.GetProjects().Last());
        var _plist = _projectManager.GetProjects().ToList();
        _plist.Reverse();
        if (_plist.Count() <= count)
            return Ok(_plist);
        return Ok(_plist.Take(count));
    }
}
```

```

}

[HttpGet("GetById")]
public IActionResult GetById(int id)
{
    if (id < 1)
        return UnprocessableEntity();
    var project = _projectManager.GetProjects().FirstOrDefault(x => x.Id == id);
    if (project == null)
        return NotFound();
    return Ok(project);
}

[HttpGet("Search")]
public IActionResult Search(string query)
{
    if (string.IsNullOrEmpty(query))
        return UnprocessableEntity("query parameter cannot be empty");
    var projects = _projectManager.GetProjects().ToList();
    projects.Reverse();
    var selectedProjects = projects.Where(x =>
x.ProjectName.ToLower().Contains(query.ToLower()) ||
x.ProjectSymbol.ToLower().Contains(query.ToLower()));
    return Ok(selectedProjects);
}
}

```

### **Файл Abstract/IFileManager.cs**

```

namespace NewCryptoParser.Abstract;

public interface IFileManager
{
    public string GetWorkPath();
    public string[] GetFiles();
    public void AddFile();
    public void RemoveFile();
}

```

### **Файл Abstract/INewCryptocurrencyProjectManager.cs**

```

using CryptoParserSdk.Models;
using NewCryptoParser.Models;

namespace NewCryptoParser.Abstract;

public interface INewCryptocurrencyProjectManager
{
    void AddNewProjects(string parserName, List<ParsingResult> projects);
    IEnumerable<NewCryptocurrencyProject> GetProjects();
}

```

### **Файл Abstract/IParserManager.cs**

```

using NewCryptoParser.Models;

namespace NewCryptoParser.Abstract;

public interface IParserManager
{

```

```
void AddParser(string code, string name);  
CryptoParserScheduledTask? GetParser(string name);  
void RemoveParser(string name);  
}
```

### **A.3 Проект «Parsers»**

Код цього проекту знаходиться на електронному носії в архіві «Application.rar» в папці «Parsers»

### **A.4 Проект «Website»**

Код цього проекту знаходиться на електронному носії в архіві «Application.rar» в папці «Website»



**ДОДАТОК Б**

**ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ**

## ДОДАТОК В

### ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

<b>Ім'я файлу</b>	<b>Опис</b>
Пояснювальні документи	
Кваліфікаційна робота.doc	Пояснювальна записка до кваліфікаційної роботи Документ Word.
Кваліфікаційна робота.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Application.rar	Архів. Містить коди програм
Builds.rar	Архів. Містить скомпільовані програми
Презентація	
Презентація.ppt	Презентація кваліфікаційної роботи