

Dokumentacja projektu z przedmiotu  
Podstawy Teleinformatyki

Temat projektu

# **Coinscraper Webscraper/Metawyszukiwarka**

POLITECHNIKA POZNAŃSKA  
WYDZIAŁ ELEKTRYCZNY

Prowadzący  
mgr inż. Przemysław Walkowiak

Skład zespołu

Arkadiusz Kołodyński  
Jakub Kaszczyński  
Maciej Palicki  
Piotr Popiołek

Poznań, 28.06.2018

# Spis treści

<b>1. O projekcie</b>	<b>3</b>
1.1 Podobne projekty	4
<b>2. Podział prac</b>	<b>4</b>
<b>3. Metodyka pracy</b>	<b>4</b>
<b>4. Zasada działania</b>	<b>5</b>
4.1 Funkcjonalność	6
4.2 Prezentacja aplikacji	8
4.3 Baza danych	10
<b>5. Wybrane technologie i narzędzia</b>	<b>12</b>
5.1 Serwer VPS	12
5.2 PuTTY	13
5.3 Apache Tomcat	14
5.4 MySQL	14
5.5 Eclipse	14
5.6 Java	14
5.7 Hibernate	15
5.8 Apache Maven	15
5.9 Selenium IDE	16
5.10 Brackets	16
5.11 Progressive Web App	16
5.12 GitHub	18
<b>6. Testy</b>	<b>18</b>
<b>7. Problemy</b>	<b>19</b>
<b>8. Instrukcja obsługi</b>	<b>19</b>
<b>9. Instrukcja uruchomienia aplikacji serwerowej</b>	<b>26</b>
<b>10 . Rozbudowa projektu</b>	<b>26</b>
<b>11. Doświadczenie wyniesione z projektu</b>	<b>28</b>
<b>12. Podsumowanie</b>	<b>28</b>
<b>13. Wnioski</b>	<b>29</b>
<b>14. Źródła materiałów i narzędzi</b>	<b>30</b>

## 1. O projekcie

Realizacja webscrappera, na przykładzie projektu zbierającego i monitorującego ceny kryptowalut na giełdach oferujących handel kryptowalutami w czasie rzeczywistym.

Czym jest webscrapping? Webscrapping jest to zbierania wszelkich możliwych danych udostępnionych w sieci lub Internecie. Możemy zbierać takie dane jak np: zdjęcia, statystyki, wydarzenia, adresy kontaktowe, ceny produktów, recenzji produktów, dane pogodowe i wiele innych. Bardzo istotne w webscrappingu jest indeksowanie zbieranych danych. Pozwala to zbudować strukturę czy też bazę, z często nie uporządkowanych danych dostępnych online najczęściej w zwykłym formacie HTML. Przygotowana w ten sposób baza, daje możliwości operacji na zebranych informacjach. Przykładowe operacje: sortowanie, porównywanie, monitorowanie zmian, znajdowanie wartości minimalnej i maksymalnej itd. Odpowiedzialnym za zbieranie i indeksowanie jest bot lub crawler.

Projekt wybraliśmy ze względu na możliwość poznania technologii jaką jest webscrapping. W naszym przypadku, postanowiliśmy wykorzystać webscrapping do analizy i monitorowania wahań cen na giełdach w czasie rzeczywistym. Rynek kryptowalut jest bardzo duży i dynamiczny, gdzie ilość transakcji na sekundę sięga kilku tysięcy. Natomiast liczba samych kryptowalut przekracza na tą chwilę 1600 i z każdym dniem rośnie. Przy tak dużej ilości danych nie jest możliwe śledzenie zmian i zachowań rynku. Bez wykorzystanie specjalistycznych narzędzi, jakimi są np. metawyszukiwarki czy też boty. Wykorzystując takie programy, jesteśmy w stanie wydobyć dla nas aktualnie najważniejsze informacje, lub też zautomatyzować handel na rynku. Przykładem jest chociażby informacja o cenie aktywa, gdzie tanio kupić i gdzie drożej sprzedać, lub też sama historia ceny aktywa, która pozwala nam wnioskować czy cena danej kryptowaluty aktualnie rośnie czy też spada. Zbieranie takich informacji jest konieczne do tworzenia wykresów i wizualizacji trendów na rynku. Właśnie z tego względu jak duży wachlarz możliwości daje odpowiednio stworzone narzędzie do webscrappingu, postanowiliśmy zrealizować ten projekt. Temat ten wydaje nam się na tyle ciekawy i przyszłościowy aby się z nim zapoznać. Możemy zauważyć trend w dzisiejszych czasach, gdzie pozyskanie danych nie jest już problemem. Natomiast prawdziwym wyzwaniem jest odpowiednie analizowanie i przetwarzanie danych w czasie na tyle krótkich, aby interesująca nas informacja nie zdażyła stracić na wartości.

## 1.1 Podobne projekty

W internecie istnieją już projekty, które zbierają podobne informacje. Każdy z nich kładzie nacisk na wybrane przez siebie dane i odpowiednią analizę ich. Potwierdza to fakt, że jest zapotrzebowanie rynku na tego rodzaju specjalistyczne narzędzia monitorujące i gromadzące dane. Naszym celem było nauczenie się jak tworzyć od podstaw tego typu projekty.

Przykłady takich platform:

- [coinmarketcap.com](https://coinmarketcap.com)
- [coincheckup.com](https://coincheckup.com)
- [livecoinwatch.com](https://livecoinwatch.com)
- [cryptocompare.com](https://cryptocompare.com)
- [tradingview.com](https://tradingview.com)
- [kursykryptowalut.com.pl](https://kursykryptowalut.com.pl)
- [coinpaprika.com](https://coinpaprika.com)
- [personaltokens.io](https://personaltokens.io)

## 2. Podział prac

Arek Kołodyński

- implementacja aplikacji klienckiej wykorzystującej podejście Progressive Web Application
- interfejs graficzny aplikacji

Jakub Kaszczyński

- tworzenie bazy danych
- zarządzanie zbieranymi danymi
- implementacja triggerów

Maciej Palicki

- konfiguracja i zarządzanie serwerem VPS
- tworzenie metawyszukiwarki

Piotr Popiołek

- tworzenie metawyszukiwarki
- selekcjonowanie danych pobieranych z giełd

## 3. Metodyka pracy

Przed rozpoczęciem pracy, przygotowaliśmy harmonogram w którym zaplanowaliśmy co będzie realizowane w dwutygodniowych interwałach pomiędzy prezentacjami. Każdy miał określone zadania, które musiał zrealizować. W tym podejściu najlepiej sprawdziła się metodyka pracy z tablicą Kanban. Stosowaliśmy ją

również w innych projektach zespołowych. Jej plusem jest na pewno to, że jest bardzo prosta w wdrożeniu i zastosowaniu. Dzięki tablicy mogliśmy w bardzo prosty sposób kontrolować postęp prac. Zachowując podział na kolumny To Do, Doing, Done.

#### **4. Zasada działania**

Nasz crawler i baza danych uruchomiony jest na serwerze VPS, cyklicznie pobieramy określone przez nas parametry. Do istniejącej listy monitorowanych giełd, możemy niezależnie od działania programu, dodać kolejne giełdy. Crawler nie wymaga restartu, aby rozpocząć śledzenie nowej giełdy. Do pobierania danych wykorzystujemy Selenium IDE. Wcześniejsze pomysły nie pozwalały nam pobierać wszystkich interesujących nas parametrów. Problemem był nie wykonujący się javascript. Po pobraniu danych, następuje ich selekcja oraz konwersja do wartości dolara. Dopiero takie dane trafiają do bazy. Baza danych ze względu na dużą ilość rekordów przechowuje tylko ostatnie 24 godziny. Wykorzystując trigger następuje usuwanie starszych rekordów. Zebrane informacje prezentowane są użytkownikom na stronie internetowej, która też może pełnić funkcję aplikacji na smartfonach.

Nasza baza danych przechowuje następujące informacje z giełdy:

- nazwa kryptowaluty
- symbol kryptowaluty
- wartość w dolarach
- wartość w bitcoinach
- ASK ( jest to oferta z najlepszą ceną sprzedaży w arkuszu zleceń )
- BID ( jest to oferta z najlepszą ceną kupna wystawioną w arkuszu zleceń )
- Volume 24h (wolumen jest po ilość akcji lub kontraktów terminowych sprzedanych i kupionych w jakiejś określonej jednostce czasu)
- wartość ATH
- nazwę danej giełdy
- aktualny czas

Na podstawie tych danych jesteśmy w stanie określić następujące parametry:

- średnią wartość danej kryptowaluty na wszystkich monitorowanych przez nas giełdach
- najmniejszą wartość danej kryptowaluty wraz z nazwą giełdy, na której aktualnie występuje
- największą wartość danej kryptowaluty wraz z nazwą giełdy, na której aktualnie występuje
- przedstawić wykres z ostatnich 24 godzin, który pokazuje zmienność ceny w czasie

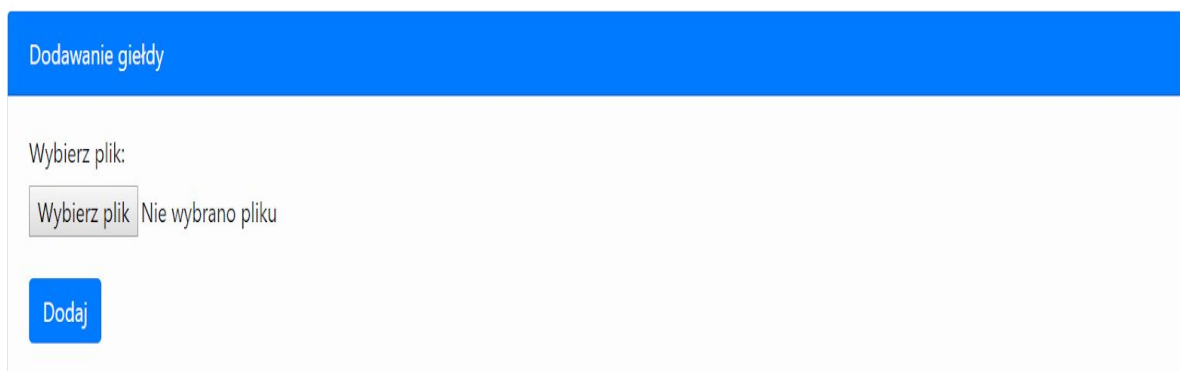
- procent od wartości ATH
- różnicę arbitrażu
- określić zainteresowanie danym aktywem, czy popyt rośnie w ostatnim czasie czy też spada

## 4.1 Funkcjonalność

Funkcjonalność możemy podzielić na dwie strony:

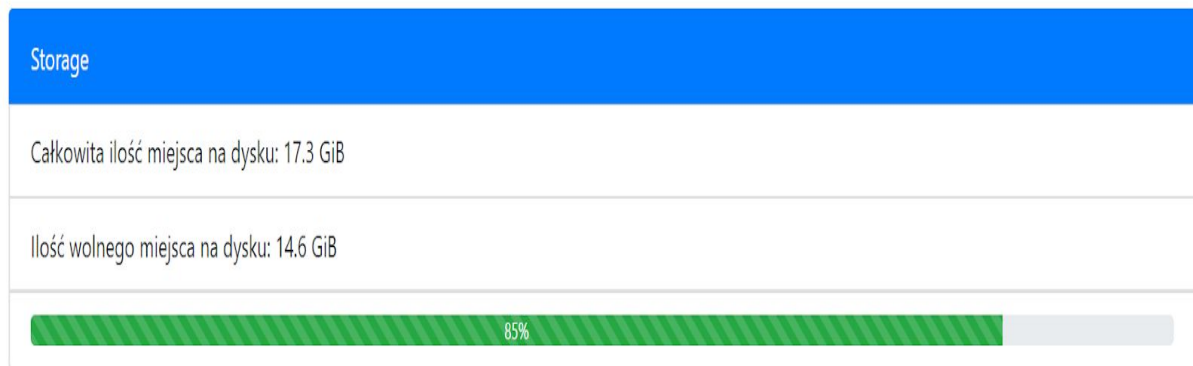
Funkcjonalności serwera

- moduł pozwalający na dodanie nowej giełdy do webscrappera, bez konieczności ingerencji w już działający program



- moduł dodania do monitorowanych kryptowalut, nowo powstałej kryptowaluty, która trawiła w obieg
- dostęp do panelu administratora, gdzie możemy aktywować, restartować i dezaktywować crawlera, a także monitorować stan wolnego/zajętego miejsca na serwerze/bazie danych

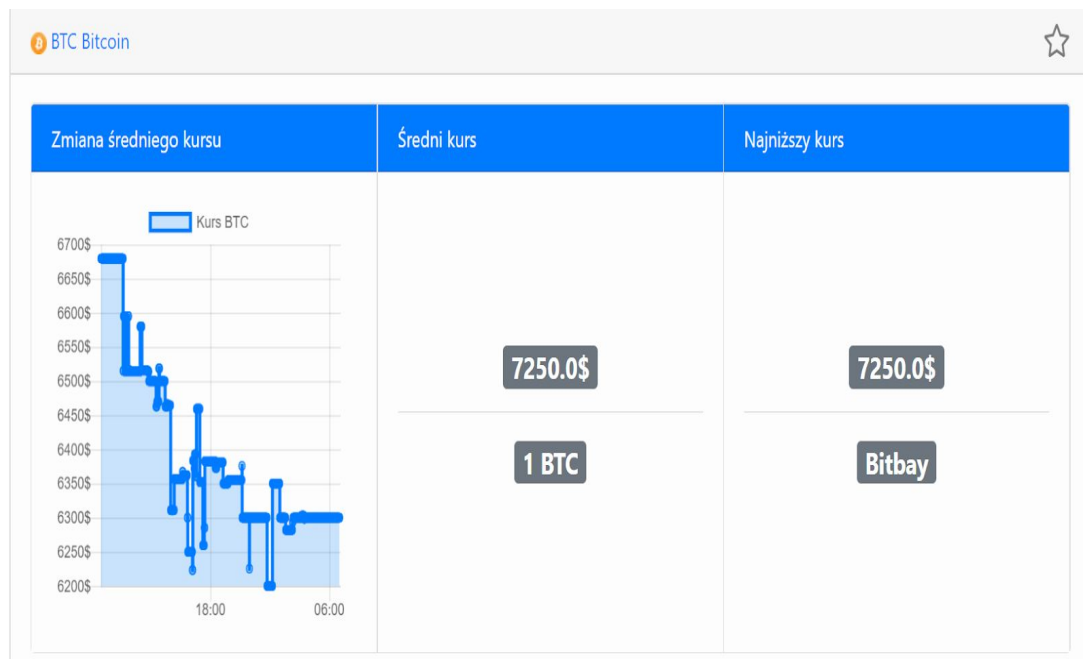




- wykonywanie operacji na rekordach bazy danych

#### Funkcjonalności klienta











- prezentacja zebranych parametrów na stronie <https://coinscraper.naberius.pl/> lub też możliwość instalacji aplikacji mobilnej na telefonie z systemem android, gdzie dzięki wykorzystaniu podejścia Progressive Web Application strona zachowuje się jak aplikacja, a w przypadku braku internetu, mamy podgląd ostatnich zapisach informacji z pamięci cache



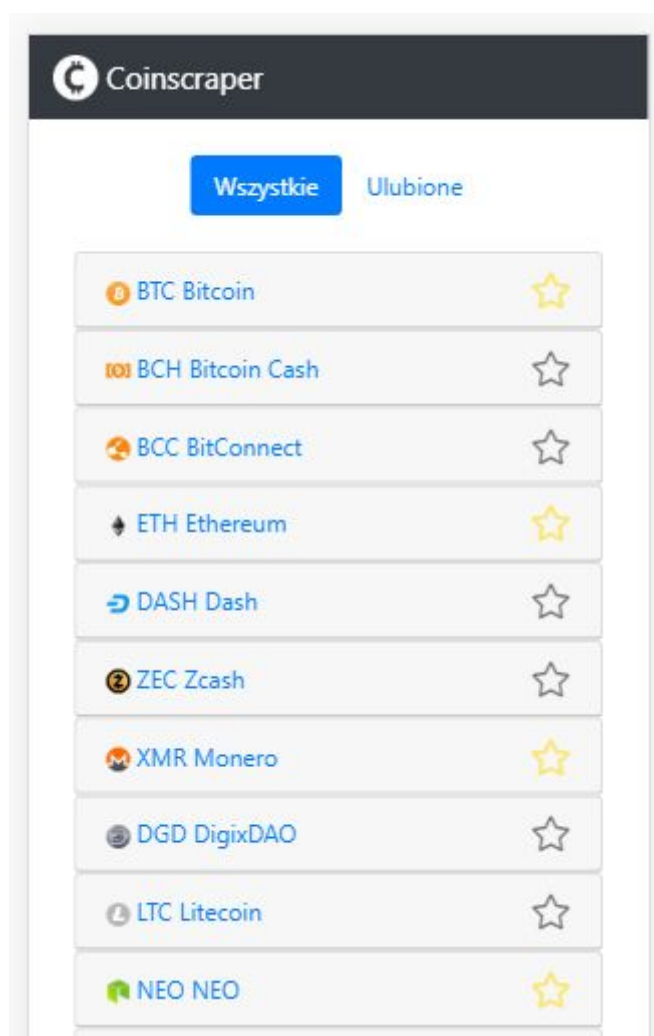
- monitorowanie historii ceny kryptowalut z ostatnich 24 godzin
- informowanie użytkownika, gdzie aktualnie cena kryptowaluty jest najmniejsza lub największa
- możliwość wybrania, które kryptowaluty chcemy obserwować, na zasadzie dodanie jej do ulubionych

Wszystkie

Ulubione

 NEO NEO	
 XMR Monero	
 GVT Genesis Vision	
 BTG Bitcoin Gold	
 ETH Ethereum	
 BTC Bitcoin	

## 4.2 Prezentacja aplikacji

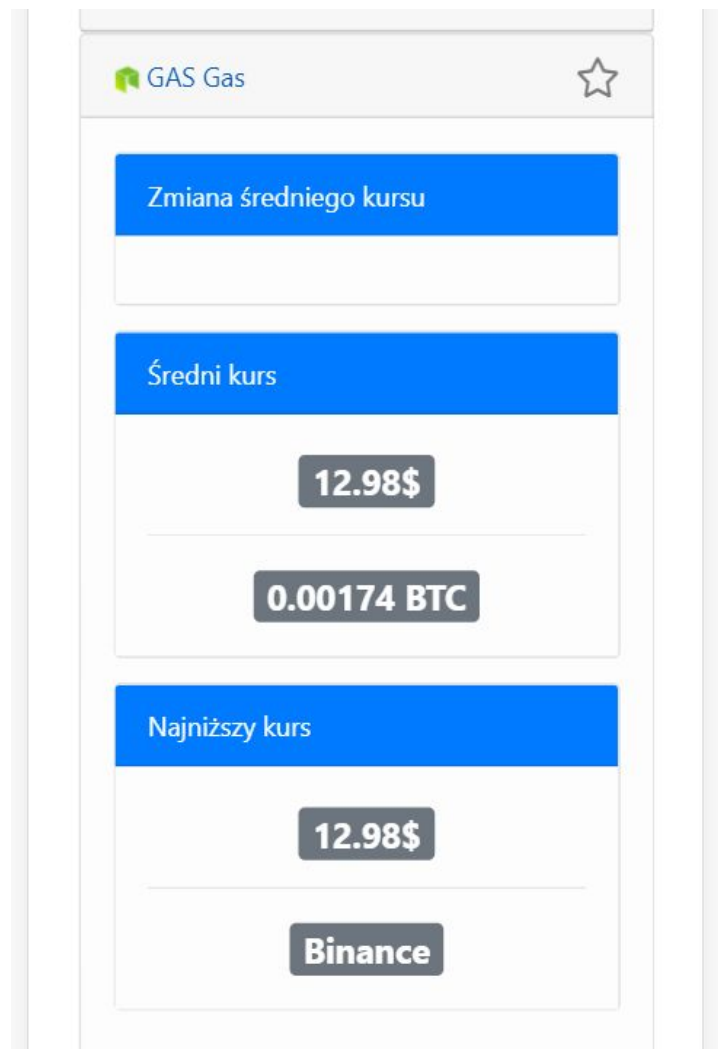


Widok 1



## Widok 1

Jest to lista wszystkich monitorowanych kryptowalut w naszej bazie. Możemy przeglądać jakie kryptowaluty są w bazie oraz dodać do ulubionych interesujące nas pozycje.

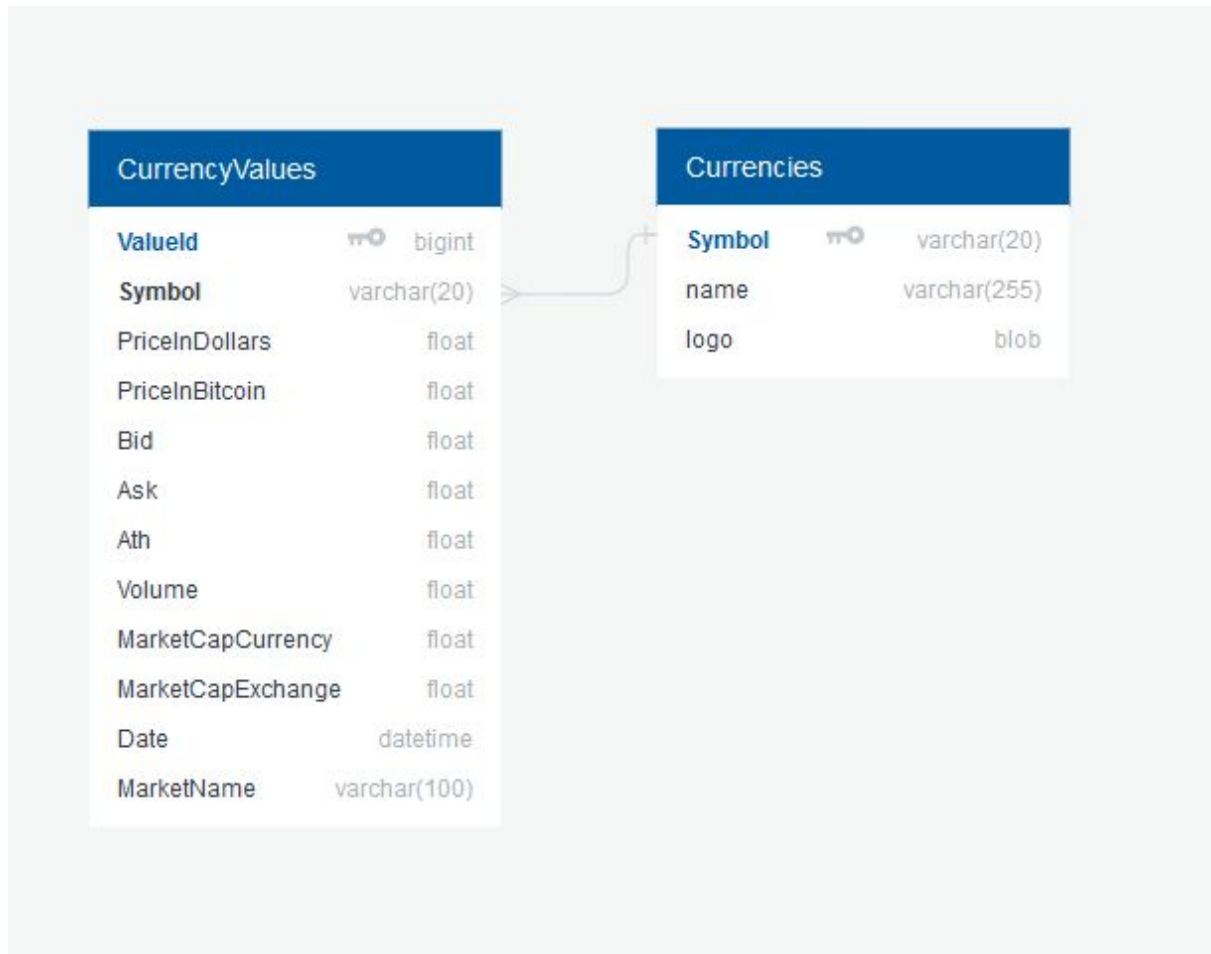


Widok 2

## Widok 2

Tutaj mamy dostęp do szczegółowych parametrów danej kryptowaluty. Możemy obejrzeć wykres z ostatnich 24 godzin, sprawdzić aktualny kurs w dolarach, kurs wyrażony w BTC, oraz gdzie jest najlepsza oferta sprzedaży.

### 4.3 Baza danych



### Struktura bazy danych

```
97
98 DELIMITER $$
99 • CREATE EVENT IF NOT EXISTS delete_old_records_graph
100 ON SCHEDULE EVERY 5 MINUTE
101 DO BEGIN
102     DELETE FROM GraphValues
103     WHERE Date < ((select * from(select max(date) from GraphValues) as t)- INTERVAL 24 HOUR);
104 END$$
105
```

Trigger usuwający starsze rekordy niż ostatnie 24 godziny.

```

WebDriverWait wait = new WebDriverWait(driver, 30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.className("ant-table-row ant-table-row-level-0")));
List<WebElement> rows = driver.findElements(By.className("ant-table-row ant-table-row-level-0"));

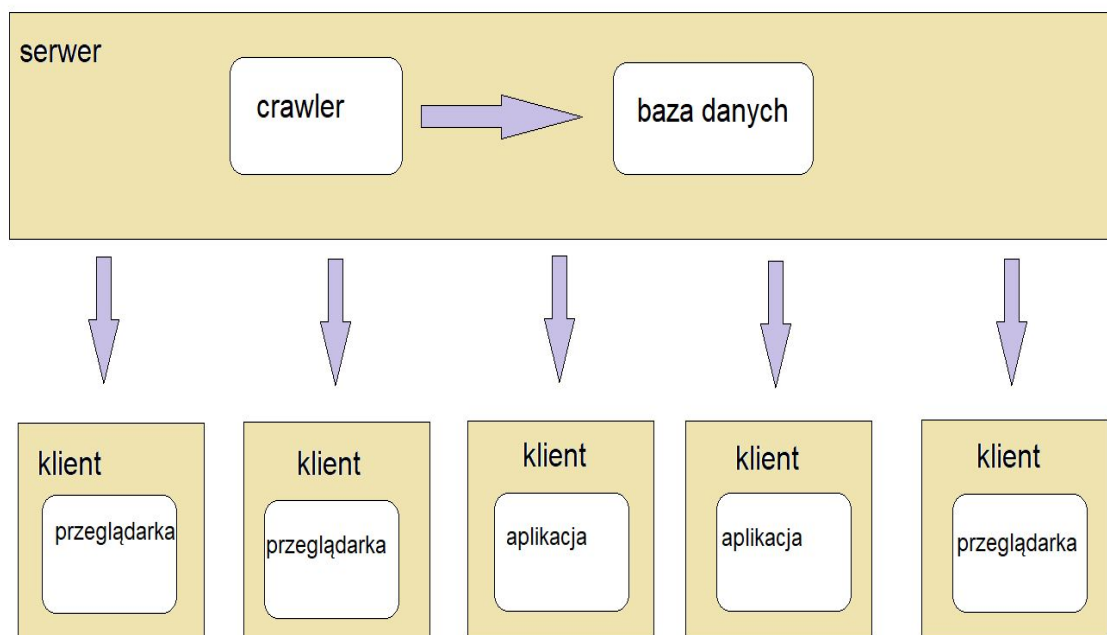
if (rows.size() > 0) {
    for (WebElement e : rows) {
        String symbol = e.findElement(By.xpath("./td[1]/div/div/span")).getText().replace("/BTC", "");
        System.out.println("\n" + symbol + "\n");
        CurrencyValue cv = new CurrencyValue(symbol);
        cv.setPriceInDollars(Double.parseDouble(e.findElement(By.xpath("./td[2]/div/span[2]")).getText().replace("$", "").replace(",", "")));
        cv.setPriceInBitcoin(Double.parseDouble(e.findElement(By.xpath("./td[2]/div/span[1]")).getText().replace(",", "")));
        cv.setAsk(Double.parseDouble(e.findElement(By.xpath("./td[2]/div/span[1]")).getText().replace(",", "")));
        cv.setBid(Double.parseDouble(e.findElement(By.xpath("./td[2]/div/span[1]")).getText().replace(",", "")));
        cv.setVolume(Double.parseDouble(e.findElement(By.xpath("./td[5]")).getText().replace(",", "")));
        cv.setMarketName("Kucoin");
        CurrencyManagement management = CurrencyManagement.getInstance();
        management.addCurrencyValue(cv);
    }
} else {
    System.out.println("Kucoin: Currencies not found!");
}

```

Fragment implementacji odpowiedzialny za pobieranie danych z jednej z monitorowanych przez nas giełd.

#### 4.4 Architektura rozwiązania

Projekt bazuje na architekturze typu klient - serwer. Mamy jeden serwer na którym znajduje się baza danych i crawler. Natomiast klientami może być każde urządzenie z systemem operacyjnym.



Architektura Coinscraper. Strzałki wskazują kierunek przepływu danych.

## 5. Wybrane technologie i narzędzia

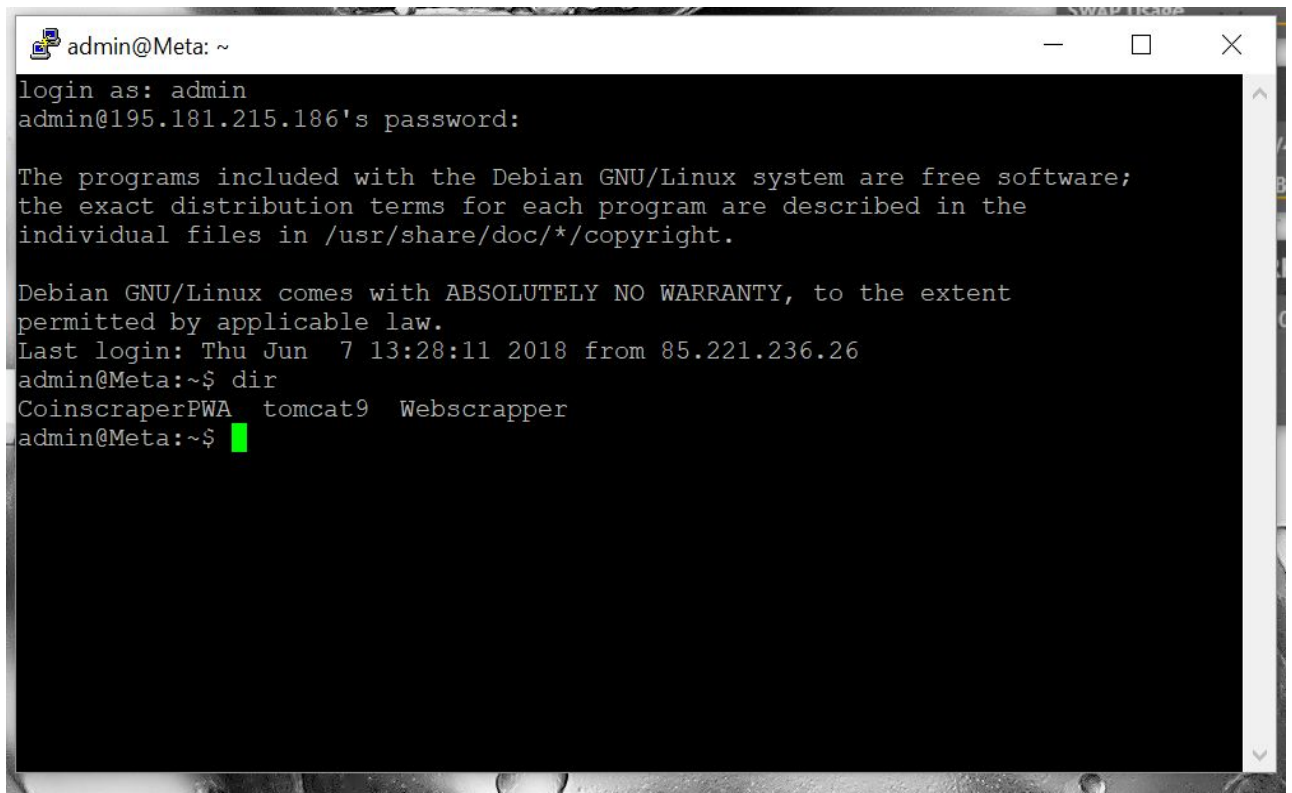
### 5.1 Serwer VPS

Jest to wirtualny serwer pracujący na systemie Debian. Łączymy się za pomocą programu putty. System Debian ma skonfigurowane konta użytkowników z odpowiednimi uprawnieniami do zarządzania serwerem.

Specyfikacje techniczne serwera.

Linux	
1 vCPU 1 GB RAM 20 GB SSD Storage 2 TB/miesiąc transferu danych	
Typ Storage	Full SSD
Publiczne IP (IPv4)	1 niezmienny (darmowy)
Publiczne IP (IPv6)*	klasa/64 darmowe
Łącze sieciowe	1

Przy wyborze serwera ważnymi dla nas były następujące czynniki liczba miejsca na dysku, brak limitu transferu danych lub też wystarczająco wysoki limit, oraz publiczny adres IP konieczny do stworzenia aplikacji klienckiej w technologii Progressive Web Application.



```
admin@Meta: ~  
login as: admin  
admin@195.181.215.186's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Jun  7 13:28:11 2018 from 85.221.236.26  
admin@Meta:~$ dir  
CoinscraperPWA  tomcat9  Webscraper  
admin@Meta:~$
```

Przykładowe nawiązanie połączenia z serwerem, adres serwera to 195.181.215.186. Połączenie nawiązujemy wykorzystując bezpieczny protokół SSH. Autoryzacja następuje przez podanie loginu i hasła.

## 5.2 PuTTY

Bezpłatny program będący klientem usług TELNET, SSH i rlogin, działający pod systemami operacyjnymi Microsoft Windows oraz Unix/Linux. Aplikacja została stworzona przez Simona Tathama i rozpowszechniana jest na licencji MIT.

PuTTY emuluje terminal tekstowy, co pozwala w łatwy sposób łączyć się z serwerem za pomocą jednego z następujących protokołów: TELNET, rlogin, SSH-1.

W naszym przypadku wykorzystujemy protokół SSH-1.

### 5.3 Apache Tomcat

Kontener aplikacji webowych rozwijany w ramach projektu Apache. Jako kontener aplikacji jest serwerem, który umożliwia uruchamianie aplikacji internetowych w technologiach Java Servlets i Java Server Pages (JSP). Jest to jeden z bardziej popularnych kontenerów Web. Tomcat jest wykorzystywany w takich serwerach aplikacji JEE (J2EE) jak Apache Geronimo. Jest również bardzo popularnym kontenerem dla samodzielnych aplikacji (niewymagających pełnego serwera aplikacji) pisanych w środowisku Spring Framework.

### 5.4 MySQL

MySQL to najpopularniejsza na świecie baza danych z otwartym dostępem do kodu źródłowego. Dzięki potwierdzonej wydajności, niezawodności i łatwości użycia baza danych MySQL jest najczęściej wybierana do współpracy z aplikacjami internetowymi w serwisach o wysokim znaczeniu typu Facebook, Twitter, YouTube oraz we wszystkich pięciu najważniejszych witrynach internetowych. Ponadto jest niezwykle popularna jako wbudowana baza danych, rozpowszechniana przez tysiące niezależnych twórców oprogramowania (ISV) i producentów OEM.

Wybraliśmy to rozwiązanie ze względu właśnie na niezawodność i wysoką wydajność. Było to konieczne aby nasz system działał bez problemów i bez awaryjnie przez 24 godziny na dobę. Zależało nam też na jak najmniejszym obciążeniu serwera i zajętości dysku.

### 5.5 Eclipse

Platforma (framework) napisana w 2004 roku w Javie do tworzenia aplikacji typu rich client. Na bazie Eclipse powstało zintegrowane środowisko programistyczne do tworzenia programów w Javie, które jest razem z tą platformą rozpowszechniane. Projekt został stworzony przez firmę IBM, a następnie udostępniony na zasadach otwartego oprogramowania. W chwili obecnej jest on rozwijany przez Fundację Eclipse.

### 5.6 Java

Java to dynamiczny, obiektowy język programowania opracowany w firmie Sun Microsystems przez grupę dowodzoną przez Jamesa Goslinga oraz Patricka Naughtona. W swej istocie cechuje się obiektywnym podejściem do programowania kładąc szczególny nacisk na niezależność od platformy systemowej oraz silne typowanie oraz dużą wydajnością. Java zaliczana jest do języków, w których kod kompilowany jest do postaci pośredniej tzw. kodu bajtowego (ang. b-code, byte-code), a następnie interpretowany przez wirtualną maszynę Javy (ang. Java

Virtual Machine, JVM) pozwalającą na uruchomienie aplikacji na różnych maszynach.

Jako język programowania postanowiliśmy wybrać Javę. Każdy z nas zna już podstawy tego języka. Kolejną rzeczą było wsparcie Javy dla zintegrowanego środowiska deweloperskiego jakim jest Selenium IDE. Język Java wybraliśmy też ze względu na możliwość pisania programów wykorzystujących wielowątkowość. Docelowo chcieliśmy stworzyć crawler, który w przypadku umieszczenia na odpowiednim serwerze, będzie wykorzystywał do monitorowania i zbierania informacji z giełdy osobne wątek.

## 5.7 Hibernate

Jest to framework do realizacji warstwy dostępu do danych (ang. persistence layer). Zapewnia on przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiekowym (ang. O/R mapping). Opiera się na wykorzystaniu opisu struktury danych za pomocą języka XML, dzięki czemu można rzutować obiekty, stosowane w obiektowych językach programowania, takich jak Java bezpośrednio na istniejące tabele bazy danych. Dodatkowo Hibernate zwiększa wydajność operacji na bazie danych dzięki buforowaniu i minimalizacji liczby przesyłanych zapytań. Jest to projekt rozwijany jako open source.

Ze względu na ciągły zapis do bazy danych nowych rekordów, wykorzystaliśmy ten framework, zapewniając tym sposobem odpowiednią wydajność.

## 5.8 Apache Maven

Narzędzie automatyzujące budowę oprogramowania na platformę Java. Maven wywodzi się z projektu Jakarta. Tak jak i inne produkty fundacji Apache, Maven jest rozprowadzany na licencji Apache License. Mianem głównego cyklu życia projektu określa się uszeregowanych kolejno osiem najważniejszych z punktu widzenia budowy aplikacji celów. Powodzenie każdego kolejnego celu uzależnione jest od pomyślnej realizacji celów znajdujących się wcześniej w cyklu.

- validate - sprawdzenie, czy projekt jest poprawny i czy wszystkie niezbędne informacje zostały określone
- compile - kod źródłowy jest kompilowany
- test - przeprowadzane są testy jednostkowe
- package - budowana jest paczka dystrybucyjna
- integration-test - zbudowany projekt umieszczany jest w środowisku testowym, gdzie przeprowadzane są testy integracyjne
- verify - sprawdzenie, czy paczka jest poprawna
- install - paczka umieszczana jest w repozytorium lokalnym - może być używana przez inne projekty jako zależność
- deploy - paczka umieszczana jest w repozytorium zdalnym (opublikowana)

Wykorzystując narzędzie jakim jest Maven, mogliśmy zapewnić sobie zgodność projektu. Pracując w czteroosobowym zespole, ważne było aby każdy mógł rozwijać swój moduł czy też wyznaczoną mu funkcjonalność, nie przejmując się czy jego korzysta z tych samych bibliotek co pozostali. Dodatkowo mogliśmy wykorzystać testy jednostkowe, np. sprawdzając połączenie z bazą danych.

## 5.9 Selenium IDE

Selenium to przenośna platforma do testowania oprogramowania dla aplikacji internetowych . Selenium zapewnia narzędzia do tworzenia testów bez potrzeby uczenia się języka skryptowego (Selenium IDE). Dostarcza również testowy język specyficzny dla domeny (Selenium) do pisania testów w wielu popularnych językach programowania, w tym C#, Groovy, Java, Perl, PHP, Python, Ruby i Scala . Testy można następnie przeprowadzić na większości nowoczesnych przeglądarek internetowych. Jest to oprogramowanie typu open-source , wydane na licencji Apache 2.0.

W naszym przypadku użyliśmy Selenium do zbierania potrzebnych danych z giełdy. Proces wdrożenia do naszego projektu nowej giełdy, to napisanie indywidualnego skryptu dla giełdy.

## 5.10 Brackets

Darmowy edytor tekstu dedykowany tworzeniu dokumentów HTML (edytor HTML) dystrybuowany na licencji MIT o otwartym kodzie źródłowym. Spośród innych edytorów webowych *Brackets* wyróżnia się wbudowaną opcją podglądu dokumentu na żywo podczas jego tworzenia.

Wykorzystywany był podczas implementacji aplikacji klienta.

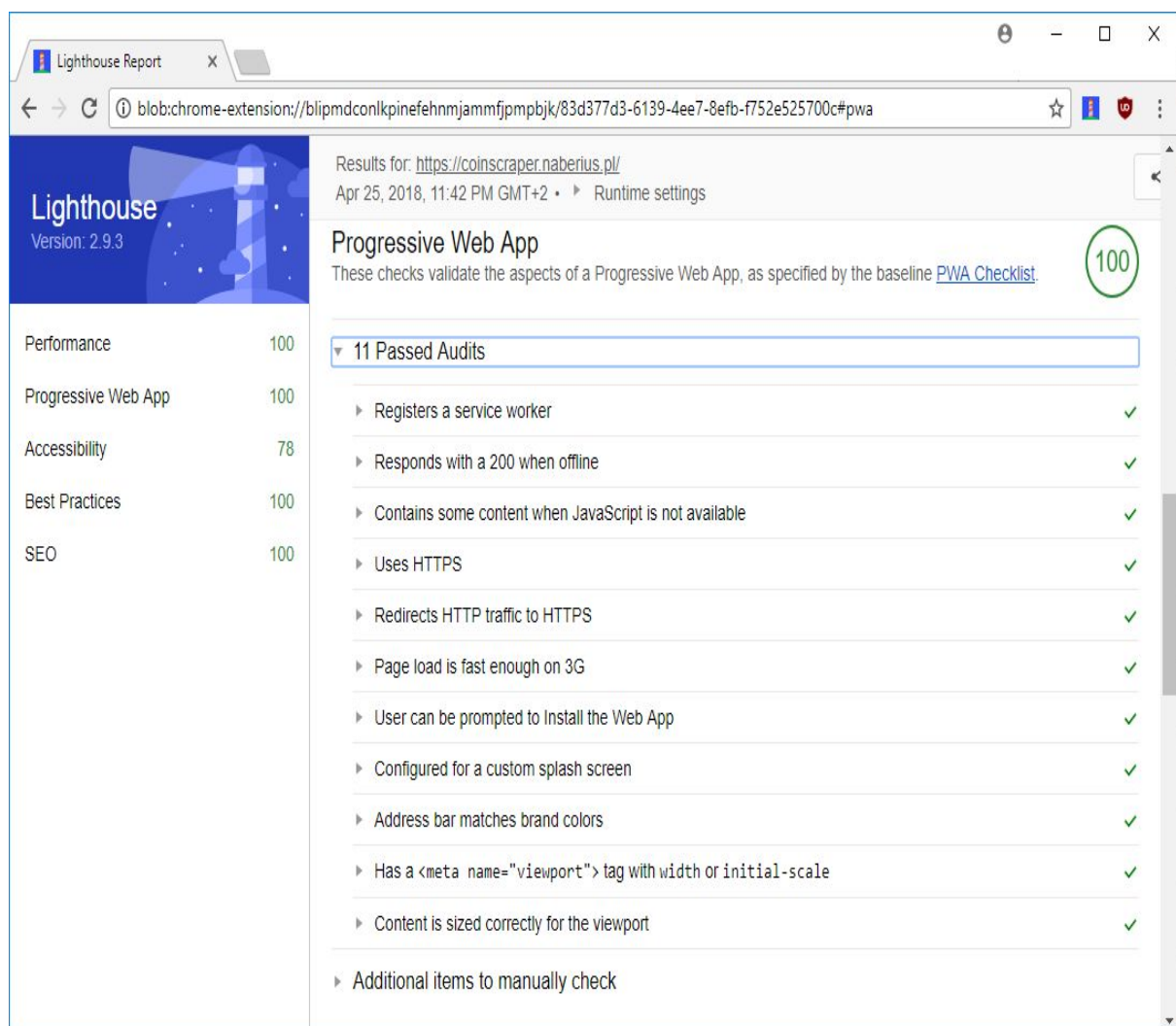
## 5.11 Progressive Web App

PWA zostały opracowane przez firmę Google'a w 2015. Są to aplikacje pisane w językach webowych (HTML + CSS + JS). Korzyścią jest, że zadziałają na zarówno na Chrome OS, Androidzie, IOS czy Windows. Niezależnie czy jest to telefon, tablet, komputer czy też konsola. Aplikacja ta potrafi działać natywnie w systemie.



Aby nasza aplikacja była zgodna ze standardem PWA, musieliśmy spełnić określone warunki:

- posiadać dedykowany adres IP
- zainstalować certyfikat HTTPS
- zaprojektować stronę internetową zgodnie z techniką RWD (Responsive Web Design), czyli nasza strona musiała dostosowywać układ i wygląd w zależności z jakiego urządzenia jest wyświetlana
- musiała być lekka i szybko się ładować
- posiadać widoki jak typowa aplikacja i reagować na interakcję użytkownika
- wyposażyć ją w mechanizm Service Workers, który odpowiedzialny jest za aplikację w przypadku utraty połączenia z Internetem, oraz za możliwość instalacji na danym urządzeniu
- stworzyć plik manifest.json



Jednym z najważniejszych aspektów PWA jest działanie w trybie offline. Można to osiągnąć na kilka sposobów. W minimalnej wersji można wyświetlić ekran, z informacją o braku dostępności sieci. Można również zapisywać w pamięci urządzenia dane, które udało nam się pobrać podczas ostatniej wizyty, a następnie prezentować je w aplikacji zamiast błędu o braku dostępności sieci. Ostatnią możliwością jest pobieranie danych w tle. Można tego dokonać z pomocą Service Worker API. Dzięki SW można określić, które zasoby powinny zostać zapisane do cache'a, a po które należy zawsze pobierać z serwera. To, co najważniejsze zapisujemy na później, tak, aby przy kolejnym odpaleniu aplikacji użytkownik nie musiał czekać aż pliki zostaną ściągnięte z naszego serwera. Oprócz zarządzalnego cache'owania Service Worker pozwala także na implementację Push-Notifications i Background Sync.

## 5.12 GitHub

Cały nasz kod był przechowywany na repozytorium GitHuba. Nie korzystaliśmy z mechanizmu rozgałęziania. Wszystkie commity trafiały bezpośrednio na mastera.

## 6. Testy

Strona coinscraper.naberius.pl testowana była na przeglądarkach Chrome, Firefox, Opera i Brave. Na wszystkich wyświetlała się poprawnie. Natomiast aplikacja testowana była na telefonach z systemem android. Tutaj również wszystko było poprawne.

W przyszłości można by przeprowadzić testy jak aplikacja zachowa się w systemie Windows poza przeglądarką.

Crawler natomiast testowany był przez miesiąc ciągłej pracy. Przez cały ten okres działał stabilnie, nie było konieczności restartowania go.

## 7. Problemy

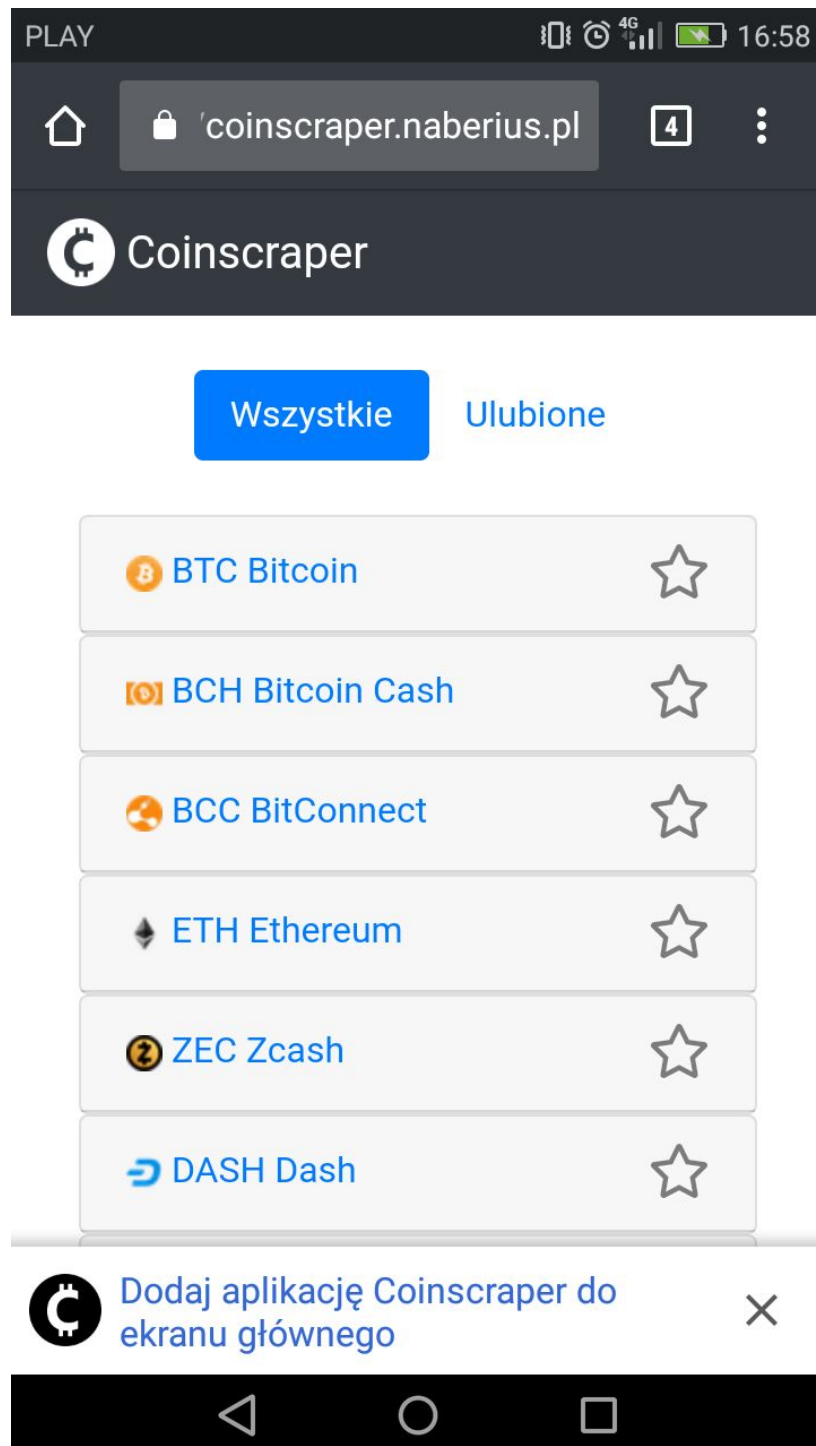
Pierwszym z problemów na jakie natrafiliśmy był wybór biblioteki do tworzenia crawlera. Chcieliśmy wykorzystać bibliotekę JSoup. Jest to bardzo popularna biblioteka w javie do pobierania i parsowania źródła strony. Szybko okazało się, że w naszym przypadku biblioteka ta nie sprawdzi się. W źródle strony nie było informacji które nas interesowały. Musieliśmy zmienić rozwiązanie na takie, które pobierze dane ze strony dopiero po wykonaniu się na niej plików JS. Postanowiliśmy w takim razie wykorzystać narzędzie Selenium, które wykorzystywane jest do testów automatycznych.

Kolejnym problemem był wybór serwera, nasz pierwszy serwer w ciągu miesiąca wykorzystał cały dostępny transfer danych. Musieliśmy zmienić serwer na taki, który nie posiada limitów transferu oraz oferuje dostatecznie dużo miejsca dla bazy danych. Baza danych ze względu na ciągły zapis nowych rekordów rosła bardzo szybko. Z tego względu też przechowujemy dane tylko z ostatnich 24 godzin. Aby archiwizować dane w dłuższej perspektywie czasu potrzeba serwerów o bardzo dużej pojemności. Pojemności mogą dochodzić tutaj do terabajtów.

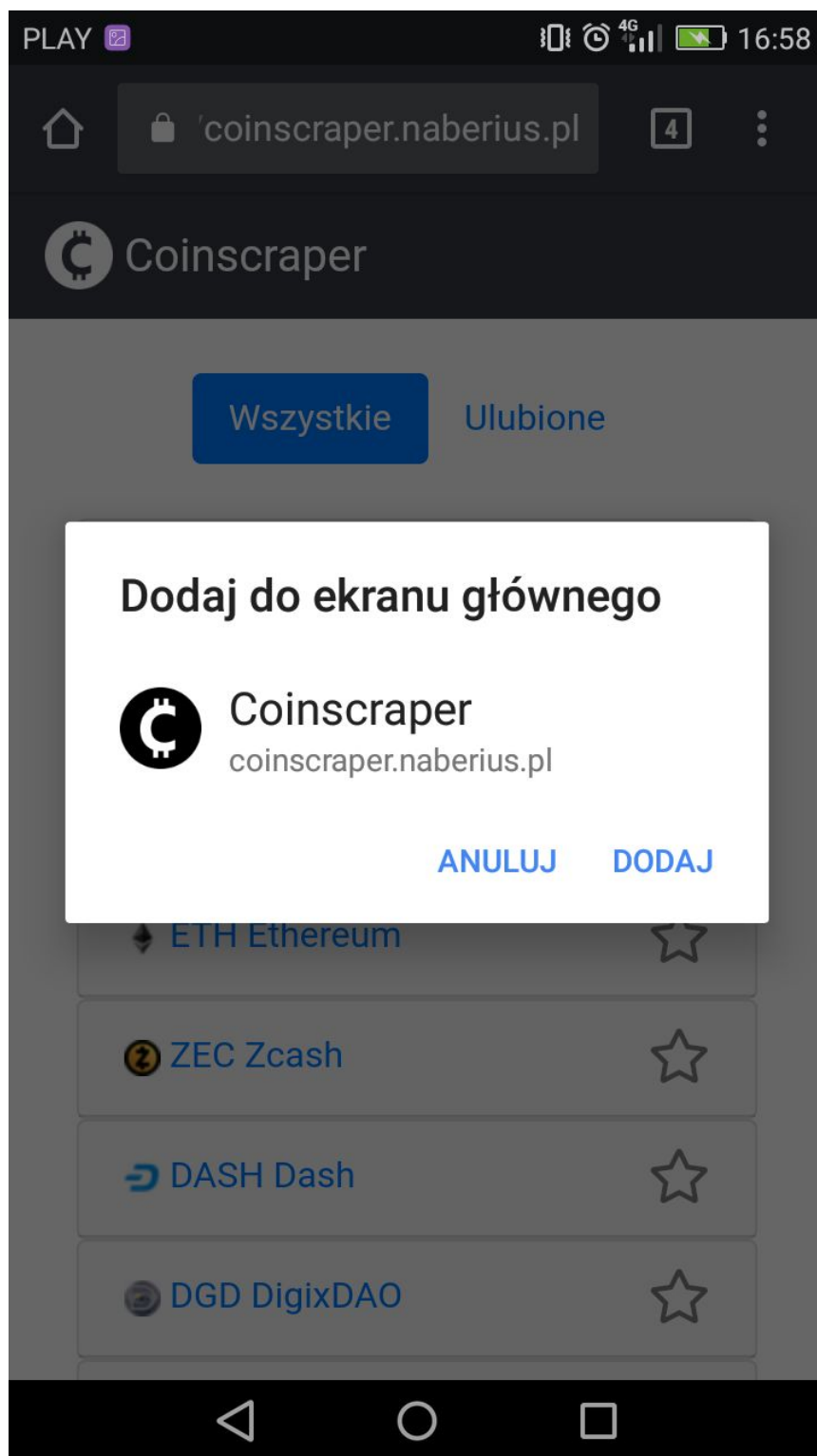
Nie byliśmy w stanie opracować rozwiązania, które byłoby uniwersalne dla każdej giełdy. Posiadamy za to moduł, który w prosty sposób dodaje klasę giełdy do crawlera, a co za tym idzie monitoruje nową giełdę. Jednak przygotowując taką klasę, trzeba podchodzić do każdej giełdy w sposób indywidualny. Każda giełda ma inną hierarchię i strukturę strony. Tak więc z każdej giełdy dane wydobywane są w inny sposób, dopiero po ich wydobyciu możemy zastosować uniwersalne podejście.

## 8. Instrukcja obsługi

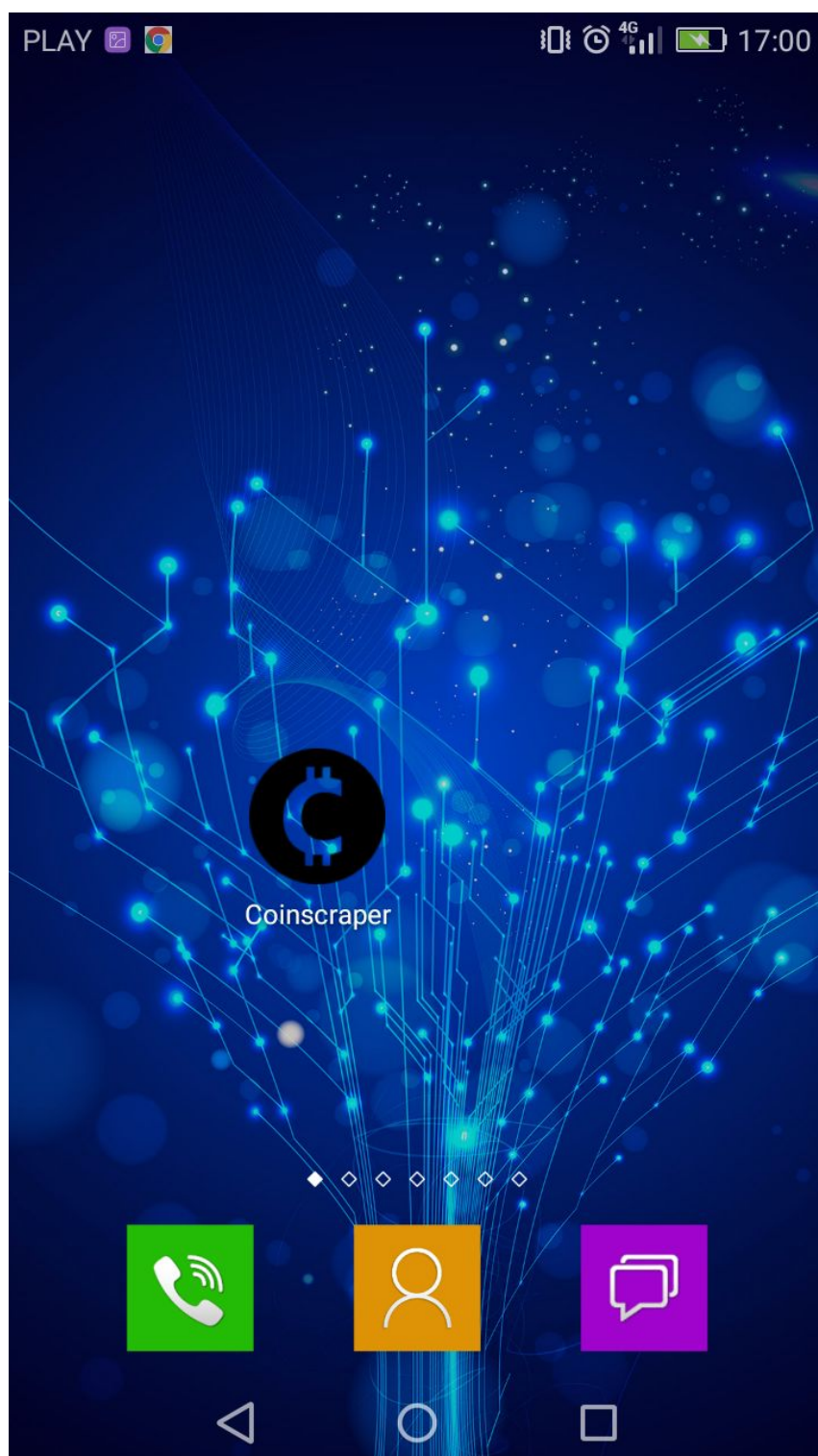
Korzystanie z klienta jest bardzo proste wystarczy wejść z dowolnego urządzenia i dowolnej przeglądarki na adres [coinscraper.naberius.pl](https://coinscraper.naberius.pl) i mamy dostęp do naszego projektu. W przypadku gdy wchodzimy na naszą stronę używając smartfonu zostaniemy zapytani czy chcemy zainstalować aplikację, zapewni to w przyszłości szybszy dostęp do danych.



Zapytanie użytkownika czy chce zainstalować aplikację Coinscraper na smartfonie z systemem operacyjnym Android.

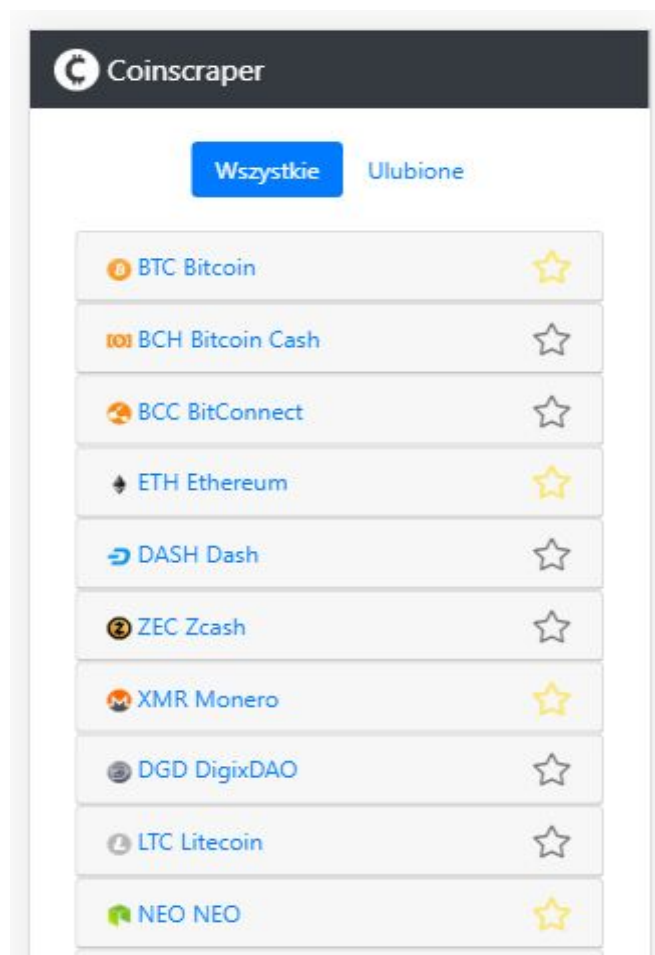


Potwierdzenie zgody na instalację aplikacji oraz dodanie jej do ekranu głównego smartfonu.

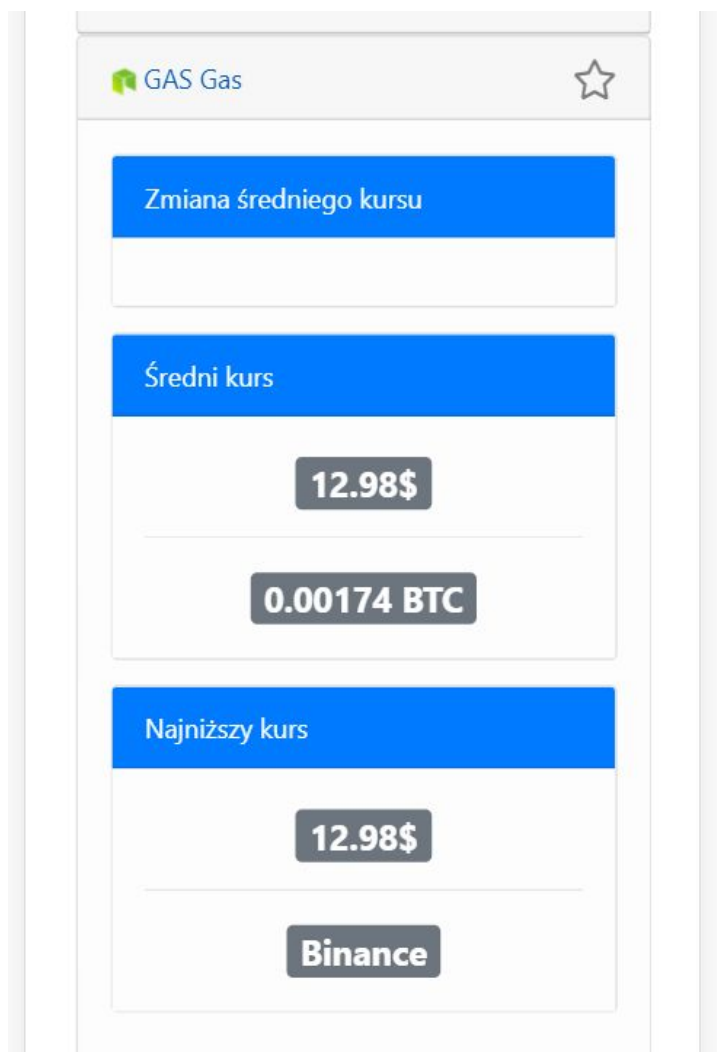


Zainstalowana aplikacja gotowa do użytkowania.

Dodatkowo aplikacja może wysyłać powiadomienia o zmianach wybranych przez nas aktywów. W przypadku gdy stracimy połączenie z internetem aplikacja wyświetli nam ostatnio zapisane wartości w cachu.

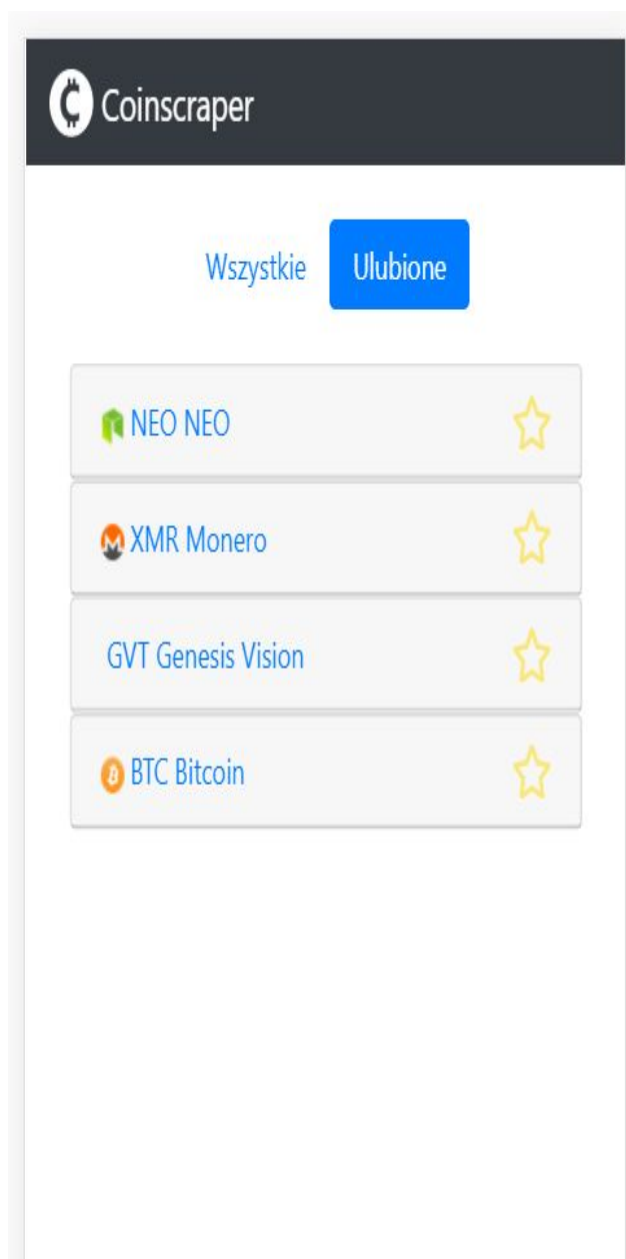


Gdy mamy już zainstalowaną aplikację, uruchamiamy ją poprzez kliknięcie w ikonkę Coinscraper. W pierwszym widoku aplikacji możemy przeglądać listę wszystkich obserwowanych kryptowalut. Wystarczy scrolować góra dół na ekranie.



Aby wejść do szczegółów danej kryptowaluty należy kliknąć button z jej nazwą. Wtedy rozwija się przegląd informacji jakie mamy zawarte w naszej bazie.





Klikając na gwiazdkę z prawej strony dodajemy kryptowalutę do ulubionych. Możemy w ten sposób stworzyć listę aktywów, które nas interesują i to je chcemy śledzić w pierwszej kolejności. Po wybraniu już kryptowalut, które nas interesują. Klikamy na button Ulubione i pokazuje nam się lista tylko tych pozycji wybranych przez nas. Gdy chcemy wrócić do całej listy wystarczy kliknąć button Wszystkie.

## 9. Instrukcja uruchomienia aplikacji serwerowej

W celu łatwego zarządzania aplikacją, z poziomu wyżej opisanego panelu administracyjnego, postanowiliśmy wykorzystać zalety aplikacji dostępnej w systemach UNIX - screen.

Screen jest menedżerem terminali, umożliwiającym uruchamianie i łatwe zarządzanie kolejnymi sesjami powłoki shell. Dzięki niemu można uruchomić wiele poleceń w wielu osobnych "oknach" i szybko się pomiędzy nimi przełączać jak i kontrolować ich działanie (np. zakończenie wykonywania).

Dzięki zastosowaniu powyższego podejścia aplikacją można sterować zarówno z poziomu przeglądarki, jak i bezpośrednio z terminala serwera (np. z wykorzystaniem SSH).

Uruchamianie aplikacji:

```
"screen -dmS NAZWA sudo PATH"
```

gdzie:

NAZWA - nazwa sesji

PATH - ścieżka do skryptu startowego

Ścieżka może być zarówno pośrednia, jak i bezpośrednia.

Zatrzymywanie aplikacji:

```
"screen -S NAZWA -X stuff \"stop^M\""
```

gdzie:

NAZWA - nazwa sesji, do której zostanie przekazany sygnał ENTER, który informuje aplikację o żądaniu zakończenia wykonywania. Po tym sygnale aplikacja nie tworzy już kolejnych wątków, lecz kończy wykonywanie obecnie przetwarzanych. Po zakończeniu wykonywania aplikacji sesja screen kończy się automatycznie.

## 10 . Rozbudowa projektu

Projekt Coinscraper można rozbudowywać na następujące sposoby. Poprzez tworzenie nowych klas giełd, zwiększając w ten sposób liczbę monitorowanych rynków. Dodatkowo można też zwiększyć liczbę zbieranych parametrów danej kryptowaluty. Można by dołożyć np głębokość rynku, liczbę transakcji, różnicę arbitrażu itd. Dzięki zwiększonej liczbie zbieranych parametrów, możemy też poprawić analizę i prezentację wyników. Dodanie wykresu głębokości rynku, wolumenu, procentowych wahań ceny. Określić market cap danej

kryptowaluty, czy też market cap całej monitorowanej giełdy. Dodać parametr ATH (All Time High), wyliczać procent do wartości ATH i wiele innych.

Przeniesienie crawlera i bazy danych na serwer o nieograniczonej pojemności i braku limitów transferowych. Zyskując w ten sposób możliwość archiwizowania danych.

Kolejną możliwością rozwoju jest rozbudowa aplikacji klienta, poprzez wykorzystanie powiadomień. Można by dodać następujące funkcjonalności:

- ustawienie powiadomienia w przypadku zmiany ceny o wartość x
- ustawienie powiadomienia w przypadku zmiany ceny o procent x
- ustawienie powiadomienia w przypadku wzrostu wolumenu o procent x
- ustawienie powiadomienia w przypadku zanotowania wejścia na giełdę nowej kryptowaluty
- ustawienie powiadomienia w przypadku przebicia ATH
- listowanie giełd
- dodanie przekierowań do giełd
- przegląd kryptowalut na wybranej giełdzie
- filtrowanie i sortowanie według wybranych parametrów
- dodanie przelicznika kryptowaluty a na kryptowalutę b
- możliwość budowania własnego portfolio wraz z śledzeniem zmiany jego wartości w czasie
- przeliczanie wartości portfolio na dowolną walutę nie tylko dollar
- dodanie widgetów

Wykorzystując odpowiednio zestaw zbieranych danych można użyć go do uczenia maszynowego w celu tworzenia analiz przyszłych cen. Określać w jakim aktualnie jesteśmy trendzie wzrostowym czy spadkowym, oraz kiedy można spodziewać się jego zmiany.

Kolejną możliwością jest stworzenie bota, który będzie obracał naszymi aktywami. W zależności od konfiguracji będzie podejmował określone decyzje mniej lub bardziej ryzykowne.

Stworzenie bota do arbitrażu, którego zadaniem będzie wyszukiwać giełdy i aktywa, na którym aktualnie można w ten sposób zarobić.

Rozbudowana crawlera o możliwość monitorowania innych aktywów niż kryptowaluty. Dostępnych np na Forexie takich jak kontrakty terminowe na złoto, srebro i wiele innych.

## 11. Doświadczenie wyniesione z projektu

Podczas tworzenia projektu, mogliśmy zapoznać się z nowymi technologiami, udoskonalić się w językach programowania i narzędziach już poznanych. Przeszliśmy ścieżkę budowania od zera narzędzia do webscrappingu. Poznaliśmy z czym to się wiąże i na co należy zwracać uwagę. Po stworzeniu harmonogramu prac, musieliśmy rozdysponować zadania i wybrać metodologię wedle, której będziemy działać. Projekt dał nam okazję pracy w grupie, gdzie byliśmy odpowiedzialni nie tylko za siebie, ale i za całą grupę. Musieliśmy nauczyć się współpracy i przynajmniej pozornego dotrzymywania terminów. Co wcale nie było łatwe. Pogodzenie pozostałych projektów studenckich, nauki, pracy i życia prywatnego. Poznaliśmy swoje lepsze i gorsze strony w używanych technologiach. Nie brakowało rzeczy, które sprawiały nam sporo problemów. Musieliśmy zwracać uwagę na szczegóły, które nie są istotne w przypadku tworzenia małych projektów. Chociażby jak limity transferu danych czy rozmiar dysku. Okazało się też, że nie wszystkie nasze wybory będą prawidłowe i podczas implementacji, trzeba będzie zmieniać technologię i rozwiązania. Wspólnie szukać rozwiązań napotkanych problemów.

Kolejnym zdobytym doświadczeniem było przygotowywanie i prezentowanie projektu przed grupą. Mogliśmy poćwiczyć sprzedaż naszego produktu i udowodnienie, że faktycznie działa.

## 12. Podsumowanie

Webscrapping jest to bardzo ciekawe zagadnienie, dające duże możliwości do działania. Można znaleźć dla niego bardzo dużo zastosowań, a dziedzina w której my go użyliśmy jest tylko jedną z wielu. Stworzenie zindeksowanej struktury danych z informacji często nieuporządkowanych pozwala w prosty sposób zarządzać nimi. Mając do dyspozycji taki zbiór możemy analizować go, wybierać interesujące nas informacje, porównywać określone wartości, czy też sortować je. W dzisiejszych czasach to właśnie właściwa informacja ma największą wartość. Mając odpowiednią informację w odpowiednim czasie, jesteśmy w stanie wykorzystać ją na naszą korzyść. W czasach gdzie jesteśmy bombardowani wszelkimi danymi, coraz trudniej nam samodzielnie je wszystkie przetwarzać. Liczba dostępnych informacji nas przerasta. Potrzebujemy więc tego typu narzędzi, które wykonają część pracy za nas. Często jest to właśnie bardzo żmudna praca i potrzeba jest jej automatyzacja. Tu też przychodzi właśnie z pomocą webscrapping.

Plusem webscrappingu jest też to, że możemy go użyć jako stworzenie jakiejś bazy, podstawy programu. A co dalej już zrobimy z stworzoną przez nas bazą zależy tylko od nas i naszej kreatywności.

Wydaje nam się, że w przyszłości zapotrzebowanie i zastosowanie webscrappingu będzie rosło, ponieważ nie tylko ludzie zaczynają tworzyć informacje ale i maszyny. W obliczu powstania internetu rzeczy, gdzie lodówka będzie komunikować się ze sklepem a okno z robotem sprząającym. Liczba generowanych danych eksploduje, a jakiś bot będzie musiał je indeksować i analizować. Dlatego też był to jeden z powodów dla którego postanowiliśmy nauczyć się tworzyć własne tego typu narzędzia.

## **13. Wnioski**

Jednym z najważniejszych wniosków z tego projektu jest sztuka planowania. Właściwie zaplanowany terminarz pozwala, określić realną do osiągnięcia datę wydania końcowego produktu. Bardzo ważne też jest szczegółowe rozrysowanie co dokładnie chce się stworzyć. Zaoszczędzi to wiele problemów podczas realizacji projektu i zmniejszy liczbę nieudów w zespole. Nie każdy ma taką samą wizję końcową projektu, więc trzeba o takich rzeczach dyskutować od początku.

Na pewno dużo rzeczy podczas realizacji projektu mogliśmy wykonać lepiej, ale dopiero teraz tego jesteśmy świadomi. Praktyka w takich realizacjach jest bardzo ważna i z każdym kolejnym projektem zdobywamy potrzebne doświadczenie.

Z podstawowych rzeczy do poprawy, na przyszłość to na pewno organizacja pracy. Gdybyśmy byli bardziej zorganizowani udało by się wdrożyć więcej giełd i więcej funkcjonalności do aplikacji. Jednak dużo czasu też straciliśmy na poznawaniu i uczeniu się nowych dla nas technologii. Warto też pracować na kilku branchach w repozytorium zamiast na jednym masterze.

## 14. Źródła materiałów i narzędzi

Tablica Kanban

<https://kanbanflow.com>

Serwer VPS

<https://www.arubacloud.pl/>

PuTTY

<https://www.putty.org/>

Apache Tomcat

<http://tomcat.apache.org/>

MySQL

<https://www.mysql.com/>

Eclipse

<https://www.eclipse.org/>

Java

<https://www.java.com/pl/download/>

Hibernate

<http://hibernate.org/>

Apache Maven

<https://maven.apache.org/>

Selenium IDE

<https://www.seleniumhq.org/>

Brackets

<http://brackets.io/>

Progressive Web App

<https://developers.google.com/web/progressive-web-apps/>

GitHub

<https://github.com/>