

# XD PROJEKT – Dokumentacja aplikacji

Dariusz Kołodziejczyk, Sebastian Bek, Mikołaj Maliszewski

22 listopada 2025

## 1 Krótki opis

Aplikacja służy do śledzenia cen akcji CD PROJEKT S.A. w wybranych okresach czasu, umożliwia prezentację wykresów oraz bieżącej ceny papierów wartościowych.

## 2 Cele i założenia projektu

Głównym celem aplikacji jest stworzenie narzędzia umożliwiającego użytkownikowi:

- bieżące monitorowanie kursu akcji CD Projekt S.A.,
- analizę zmian wartości w wybranych okresach czasu,
- zestawienie kursu z poziomem zainteresowania w Internecie (Google Trends),
- pobieranie podstawowych dokumentów finansowych udostępnianych przez spółkę.

## 3 Środowisko i wymagania

Aplikacja została stworzona w języku Python w wersji 3.11.9 i działa w środowisku graficznym opartym na bibliotece CustomTkinter.

Do prawidłowego działania wymagane są następujące biblioteki:

- `yfinance` – pobieranie danych giełdowych z Yahoo Finance,
- `matplotlib` – generowanie wykresów kursu,
- `Pillow` – obsługa grafiki,
- `customtkinter` – tworzenie nowoczesnego interfejsu użytkownika,
- `CTkMessageBox` – obsługa okien komunikatów,
- `tkcalendar` – wybór zakresu dat,
- `pytrends` – komunikacja z Google Trends.

Instalacja pakietów za pomocą pliku `req.txt`:

```
pip install -r req.txt
```

## 4 Opis funkcjonalności

Aplikacja realizuje następujące funkcje:

- pobieranie aktualnej ceny akcji CD Projekt S.A.
- wyświetlanie wykresów kursu z zakresu:
  - ostatniego dnia (1d)
  - ostatniego tygodnia (7d)
  - ostatniego miesiąca (1m)
  - dowolnego zakresu wybranego z kalendarza
- pobieranie i prezentacja wykresów z Google Trends z zakresu:
  - ostatniego dnia (1d)
  - ostatniego tygodnia (7d)
  - ostatniego miesiąca (1m)
- możliwość wyboru słowa kluczowego (np. CD Projekt”, Cyberpunk 2077”, “Wiedźmin”)
- możliwość pobierania raportów finansowych spółki w formacie PDF lub XLSX
- zapisywanie pobranych danych w pamięci podręcznej (cache), aby zmniejszyć liczbę zapytań do API

## 5 Struktura kodu

Aplikacja podzielona jest na moduły logiczne zgodnie z zasadą rozdziału odpowiedzialności.

### 5.1 Główna klasa aplikacji – App

Klasa ta inicjalizuje wszystkie ekrany aplikacji oraz zarządza ich wyświetlaniem. Po uruchomieniu programu użytkownik widzi ekran wykresu kursu akcji.

### 5.2 Ekran wykresów giełdowych – Screen1

Odpowiada za:

- wyświetlanie wykresu kursu,
- wybór zakresu danych (1 dzień, 1 tydzień, 1 miesiąc),
- podawanie zakresu użytkownika za pomocą kalendarza,
- aktualizację:
  - bieżącej ceny akcji,
  - minimalnej i maksymalnej ceny w zakresie.

### **5.3 Ekran Google Trends – Screen2**

Udostępnia:

- wybór słowa kluczowego (np. CD Projekt, Wiedźmin),
- wygenerowanie wykresu popularności,
- odświeżanie danych,
- wykorzystanie cache w celu minimalizacji obciążenia API.

### **5.4 Ekran sprawozdań – Screen3**

Pozwala na:

- wybór kwartału z listy raportów,
- pobranie:
  - raportu finansowego PDF,
  - raportu XLSX z wynikami finansowymi,
  - informacji prasowej w PDF.

### **5.5 Moduł danych – CDPdata**

Zawiera logikę:

- pobierania danych z Yahoo Finance,
- obsługi Google Trends poprzez PyTrends,
- zapisu i wczytywania cache w formacie JSON.

### **5.6 Moduł wykresów – CDPplot**

Odpowiada za renderowanie wykresów biblioteki matplotlib i osadzanie ich w komponentach interfejsu.

### **5.7 SplashScreen**

Odpowiada za wyświetlanie startowego ekranu i ładowanie danych przed uruchomieniem aplikacji.

### **5.8 App**

Główne okno aplikacji, odpowiedzialne za przełączanie ekranów.

### **5.9 Screen1**

Ekran obsługujący wykresy kursu giełdowego.

## 5.10 Screen2

Obsługuje wykresy danych z Google Trends.

## 5.11 Screen3

Obsługuje dane z sprawozdań kwartalnych CD Projekt S.A.

# 6 Opisy funkcji

## CDPdata

- **GetCdpData(period)** - pobiera dane o kursie z serwisu Yahoo Finance (yfinance) dla danego okresu
- **getCustomCDPData(start\_time, end\_time)** - pobiera dane o kursie dla wybranego przez użytkownika zakresu
- **GetCurrentPrice()** - pobiera i wyświetla aktualną cenę akcji
- **getMinMaxPrice(data)** - pobiera najniższą i najwyższą cenę w danym zakresie
- **ensure\_cache\_dir()** - upewnia się że istnieje folder plików JSON jeśli nie tworzy go
- **load\_cache()** - wczytuje cache a jeśli plik jest uszkodzony zwraca pusty
- **save\_cache(cache)** - zapisuje dane do pliku JSON
- **invalidate\_trends\_plot(keyword, period)** - usuwa cache przy odświeżaniu wykresów
- **timeframe\_for(period)** - zamienia nazwy okresów na te używane przez pytrends
- **df\_from\_entry(entry)** - tworzy DataFrame ze wczytanego cache
- **getTrendsData(period)** - wczytuje dane z pliku JSON a jeśli nie ma pobiera dane z Google Trends

## CDPplot

- **CreateCdpPlot(frame, period)** - tworzy i wyświetla wykres cen
- **createCustomDataCdpPlot(frame, start\_date, end\_date, data)** - tworzy wykres dla wybranego przez użytkownika zakresu
- **createTrendsPlot(frame, period, data, keyword)** - tworzy wykres zainteresowania z Google Trends

## SplashScreen

- `__init__(self)` - ustawia wysokość szerokość i zawartość splash screena
- `center_x(self, window_width)`, `center_y(self, window_width)` - pomocnicze funkcje do ustawienia splash screena na środku ekranu
- `load_data_with_delay(self)` - pobiera dane podczas splash screena i czeka co najmniej 2,5 sekundy
- `open_main_app(self, data)` - zamyka splash screen i otwiera główne okno aplikacji

## App

- `__init__(self, master, preloaded_data, preloaded_trends)` - ustawia wysokość szerokość i zawartość głównego okna aplikacji
- `showFrame(self, frame, text)` - zmienia widoczny ekran i nazwę aplikacji

## Screen1

- `__init__(self, master, preloaded_data, preloaded_trends)` - ustawia ramki okna wykresu akcji
- `showPlot(self, period)` - aktualizuje wykres po kliknięciu któregoś z przycisków
- `showCustomDatePlot(self)` - aktualizuje wykres po wpisaniu własnego zakresu
- `updatePriceLabel(self)` - aktualizuje cenę po kliknięciu przycisku
- `updateMinMaxLabels(self, data)` - aktualizuje min i max ceny po kliknięciu przycisku
- `showError(self, title, message, icon)` - pokazuje okienko z błędem

## Screen2

- `__init__(self, master, preloaded_data, preloaded_trends)` - ustawia okna wykresów Google Trends
- `showTrendsPlot(self, period)` - tworzy wykres dla Google Trends z określonego zakresu
- `refreshTrends(self)` - odświeża dane używane do rysowania wykresu Google Trends

## 7 Zagrożenia

### 7.1 Błąd 429 Google Trends

Wynika z zbyt dużej liczby zapytań do serwisu Google Trends w krótkim czasie.

**Rozwiązanie:** korzystanie z zapisanych danych w JSON (cache) oraz ograniczenie liczby zapytań

## 7.2 Blokowanie Yahoo Finance

Serwis Yahoo Finance wprowadza dzienne limity około 2000 zapytań na IP

**Rozwiązanie:** ograniczanie częstotliwości zapytań, cache danych historycznych oraz stosowanie interwałów

## 7.3 Obsługa plików JSON (cache)

Możliwe błędy serializacji i deserializacji, np. uszkodzony plik JSON lub niepoprawny format danych

**Rozwiązanie:** obsługa wyjątków przy wczytywaniu i zapisie JSON, atomowy zapis do pliku tymczasowego, backup uszkodzonego pliku

## 7.4 Path traversal w cache

Niepoprawnie sformatowane słowa kluczowe mogą prowadzić do zapisania/usunięcia plików poza katalogiem cache

**Rozwiązanie:** sanitacja słów kluczowych (tylko litery, cyfry i podkreślenia), użycie `os.path.basename` oraz bezpieczne tworzenie nazw plików

## 7.5 Pobieranie plików z internetu

Pobieranie plików PDF/XLSX bez weryfikacji rozszerzenia, MIME type czy rozmiaru może prowadzić do pobrania złośliwego pliku

**Rozwiązanie:** sprawdzanie typu i rozmiaru pliku, obsługa wyjątków, zapis w bezpiecznym folderze

## 7.6 Brak timeoutów w zapytaniach sieciowych

Funkcje `requests.get`, `yfinance.download` i Google Trends mogą blokować aplikację przy problemach sieciowych

**Rozwiązanie:** ustawienie timeoutów i obsługa wyjątków przy pobieraniu danych

## 7.7 Duże rozmiary danych

Pobieranie dużych zbiorów danych (Google Trends, Yahoo Finance) może prowadzić do problemów z pamięcią

**Rozwiązanie:** ograniczenie liczby pobieranych danych lub filtrowanie niepotrzebnych kolumn

## 7.8 Interfejs użytkownika

Brak pełnej kontroli nad ścieżką folderu wyboru w Tkinter może prowadzić do zapisu w systemowych katalogach

**Rozwiązanie:** domyślny folder w katalogu aplikacji oraz walidacja ścieżki

## **8 Aktualizacje**

### **27.10.2025**

- Podział aplikacji na moduły
- Dodanie funkcji do obsługi zakresów dat
- Nowe biblioteki: CtkMessageBox i tkcalendar

### **02.11.2025**

- Dodano pytrends
- Obsługa wykresów Google Trends

### **09.11.2025**

- Dodano obsługę zapisu trendów do JSON
- Podział aplikacji na 3 ekrany

### **16.11.2025**

- Dodano wykresy Google Trends dla Cyberpunk 2077 i Wiedźmin
- Dodano pobieranie sprawozdań finansowych CD Projekt SA

### **23.11.2025**

- Poprawienie dokumentacji