TFAL TERMINAL

Przewodnik po projekcie

Lokalny system wyceny spółek i liczenia wyniku portfela

Autor: Kacper Teichert · **GitHub:** github.com/kicporr · **Kontakt:** kacper.teichert6790@gmail.com

Wersja 1.0 · listopad 2026 · dokument dla osób przeglądających projekt

Projekt w skrócie

Obszar	Wartość
Kod	17 modułów Pythona, 28 endpointów REST, ~9 600 linii (backend + frontend)
Model	3 metody wyceny (P/E, P/S, EV/EBITDA), 16 kryteriów oceny jakości
Dane	FactSet przez API Scraba, notowania Yahoo, kursy NBP; 5 lat historii
Raporty	Karta spółki PDF (3 strony), dokumentacja PDF (31 stron) z kodu

W tym dokumencie

1. Czym jest ten projekt 2
2. Jak działa — przepływ danych 2
3. Przewodnik po sekcjach aplikacji 3
4. Jak liczone są liczby 4
5. Jak uruchomić 5
6. Jak czytać wynik — przykład 6
7. Struktura repozytorium 6
8. Ograniczenia 7
9. Summary in English 7
10. Jak sprawdzić projekt w pięć minut 7

Pełna dokumentacja techniczna znajduje się w repozytorium w katalogu docs/ (model, architektura, API, operacje) oraz w wersji do druku: docs/Dokumentacja-TFAL.pdf.

Sekcje 1–3 opisują produkt i jego ekrany, 4–6 metodykę i uruchomienie, 7–10 zaplecze techniczne i sposób weryfikacji projektu.

1. Czym jest ten projekt

TFAL Terminal to lokalna aplikacja desktopowa do analizy spółek i prowadzenia własnego portfela. Zastępuje arkusz kalkulacyjny, w którym te same obliczenia trzeba było wykonywać ręcznie przy każdej zmianie danych: pobiera dane fundamentalne, wycenia spółkę trzema metodami mnożnikowymi, ocenia ją na 16 kryteriów, prowadzi dziennik transakcji i liczy realny wynik portfela.

Powstał z konkretnej potrzeby: chciałem mieć jedno miejsce, w którym mogę sprawdzić, ile spółka jest warta przy własnych historycznych mnożnikach, jak wypada na tle własnej historii i czy mój wynik to zasługa doboru spółek, czy ruchu waluty. Program działa offline na lokalnych plikach, uruchamia się dwuklikiem i wygląda jak program desktopowy, nie strona internetowa.

Co ten projekt pokazuje

- **Model i dane —** wycena mnożnikowa, prognozy 12M, scenariusze, siatki wrażliwości;
- **Inżynieria danych —** API FactSet/Scrab, Yahoo i NBP, cache, obsługa sesji, braków danych i błędów;
- **Programowanie —** Python, Flask, JSON, matplotlib, generowanie PDF, okno WebView2 bez ramki;
- **Projektowanie informacji —** interfejs, który tłumaczy liczby: werdykt, przegląd zmian, panel „co zakłada cena".

Jak powstał

Projekt rozwijałem iteracyjnie: najpierw model wyceny w Pythonie i arkusz wynikowy, potem aplikacja desktopowa, na końcu analiza ryzyka, dziennik i raporty. Każdą zmianę w modelu weryfikowałem liczbowo — porównywałem wynik z obliczeniami ręcznymi i arkuszem źródłowym. Część kodu powstawała z asystentem AI, ale architektura, założenia modelu i kryteria akceptacji były moje; asystent przyspieszał implementację, nie podejmował decyzji.

2. Jak program działa — przepływ danych

Program jest zbudowany jak mały system, a nie pojedynczy skrypt. Dane przechodzą przez cztery warstwy, a każda z nich jest osobno testowalna:

1. Pobranie danych — FactSet przez API Scraba (kurs, prognozy EPS i sprzedaży, mnożniki P/E, P/S, EV/EBITDA, bilans), notowania i zdarzenia z Yahoo, kursy walut z NBP.
2. Cache — odpowiedzi lądują na dysku (bundle na ticker, TTL 4 h), więc aplikacja działa szybko i offline; nie pobiera tych samych danych przy każdym kliknięciu.
3. Model — wycena (cena docelowa z mnożników × prognozy), ocena 16 kryteriów, kryteria TFL, ryzyko, wynik portfela z dziennika transakcji.
4. Interfejs i raporty — widoki w oknie WebView2 oraz PDF-y generowane z tych samych danych (karta spółki, portfel, dokumentacja).

Kluczowa decyzja projektowa: model i interfejs korzystają z jednego źródła obliczeń. Karta PDF nie jest osobnym raportem „na boku" — powstaje z tych samych liczb, które widzisz na ekranie, więc nie może się z nimi rozjechać.

3. Przewodnik po sekcjach aplikacji

Aplikacja ma osiem widoków. Poniżej: co robi każdy z nich i na co warto zwrócić uwagę.

3.1 Portfel

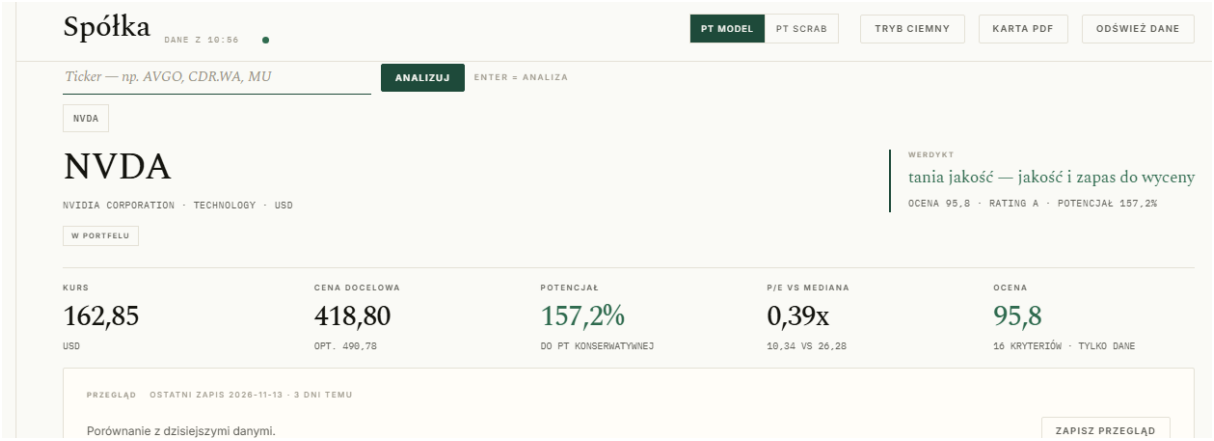
Lista spółek z oceną, potencjałem do ceny docelowej i wartością pozycji. Jeden przełącznik zmienia źródło ceny docelowej: model własny albo konsensus analityków (Scrab).

- **Co widzisz:** ocena i rating (A/B/C/D), rewizje prognoz EPS w 3 miesiące, waga pozycji;
- **Co możesz zrobić:** sortowanie po każdej kolumnie, dodawanie i usuwanie tickerów, lista obserwowanych;
- **Na co uważać:** kolumna „Potencjał" liczy się do wybranej ceny docelowej — po przełączeniu na Scrab zmieniają się wszystkie liczby, bez ponownego pobierania danych.

3.2 Spółka — karta analizy

Najważniejszy ekran. Dla dowolnego tickera (także spoza portfela) pokazuje pełną analizę w jednym miejscu.

- **Nagłówek:** werdykt jednym zdaniem, cena docelowa konserwatywna i optymistyczna, potencjał, ocena;
- **Przegląd:** porównanie z ostatnim zapisem: kurs, prognozy, składniki ceny docelowej, mnożniki i ocena — z informacją, gdy z wyceny wypadła jedna z metod;
- **Scoring:** 16 kryteriów z punktacją i wykresem, obok surowe metryki (ROIC, FCF/zysk, dług netto/EBITDA);
- **Kryteria TFL:** jedenaście pytań z listy kontrolnej inwestora, na które odpowiadają dane: przychody, marże, zadłużenie, wycena vs historia, trend, rewizje ceny docelowej i prognoz EPS;
- **Wykres:** cena + panele mnożników (P/E, P/S, PT upside, autorski PS 2Y / wzrost przychodów); kurs logarytmicznie, panele liniowo;
- **Checklista:** checklista przed decyzją (5 pytań) zapisywana przy spółce i wchodząca do przeglądu;
- **Analiza odwrotna:** „co zakłada cena" — odwrócona wycena: ile musiałby wynieść zysk, sprzedaż i marża, żeby uzasadnić dzisiejszy kurs;
- **Eksport:** trzystronicowa karta PDF z tymi samymi liczbami.



Nagłówek karty spółki: werdykt, cena docelowa, potencjał, mnożnik względem mediany i ocena.

3.3 Dziennik

Transakcje wprowadzane ręcznie albo importowane z XTB (plik xlsx/csv). Program liczy koszt FIFO w walucie instrumentu, a wynik przelicza na złote po kursie z dnia transakcji — nie po dzisiejszym, żeby wynik nie zmieniał się bez sprzedaży.

- **Co widzisz:** zrealizowany i niezrealizowany P&L, dywidendy, wpłaty netto, skuteczność;

- **Statystyki:** trafność per spółka, średni czas trzymania i zmiana kursu 60 dni po sprzedaży — odpowiedź na pytanie „czy sprzedaję za wcześnie”;
- **Import:** import z XTB pomija duplikaty i ostrzega o sprzedaży bez pokrycia w dzienniku.

3.4 Wyniki

Realny wynik portfela policzony z transakcji — od pierwszego wpisu do dziś.

- **Miary:** TWR (wynik strategii, niezależny od wpłat) i MWR (XIRR — wynik Twoich pieniędzy), obsunięcie, zmienność, Sharpe, zwroty miesięczne;
- **Benchmark:** S&P 500, Nasdaq 100, MSCI World albo WIG20TR, przeliczone na PLN; beta, alfa, tracking error, capture;
- **Atrybucja:** tabela „kto zrobił wynik” — zrealizowane, niezrealizowane i dywidendy per spółka.

3.5 Ryzyko

Ryzyko portfela liczone z 3-letniej historii wspólnych notowań.

- **Miary:** zmienność annualizowana, VaR 5% (parametryczny i historyczny), największy wkład w ryzyko;
- **Struktura:** macierz korelacji 5×5, ekspozycja sektorowa, koncentracja (HHI, efektywna liczba pozycji);
- **Scenariusze:** USD/PLN −10%, największa pozycja −20%, wszystkie spółki −20% — strata w złotych i procentach portfela; liniowo, bez zmiany korelacji.

3.6 Kalendarz

Najbliższe zdarzenia spółek z portfela (wyniki, ex-dividend) oraz — obok — status przeglądów: kiedy ostatni zapis i które spółki przekroczyły próg 90 dni. Kliknięcie wiersza otwiera kartę spółki.

3.7 Rebalans

Wagi docelowe według wybranej metody (równa, odwrotność zmienności, ocena), z limitem na pozycję i na sektor. Program pokazuje różnicę do dzisiejszego stanu i listę zleceń w sztukach; limity sektorowe nie przelewają pieniędzy do innej branży — nadwyżka zostaje w gotówce.

3.8 Screener

Filtrowanie portfela i listy obserwowanych po ocenie, wycenie (P/E vs mediana), potencjale, sektorze i ratingu. Służy do odpowiedzi na pytanie „co kupić za nowe wpłaty”, a nie do przeglądania całego rynku — program nie pobiera danych hurtowo, tylko dla wskazanych tickerów.

4. Jak liczone są liczby

Trzy rzeczy warto rozumieć, żeby czytać ekrany świadomie: skąd bierze się cena docelowa, z czego powstaje ocena i jak liczony jest wynik portfela.

4.1 Cena docelowa

Wycena jest mnożnikowa i horyzontowa: bierzemy historyczne mnożniki spółki i przykładamy je do prognoz na najbliższe 12 miesięcy.

- **Konserwatywna:** mediana dziennych mnożników P/E i P/S z 3-letniego okna;
- **Optymistyczna:** mediana rocznych maksimów z tego samego okna — „poziom dobrych czasów”, a nie jednorazowy szczyt, który potrafi zafałszować wycenę na lata;

- **Cena docelowa:** średnia z metody P/E i P/S (opcjonalnie także EV/EBITDA); przy stracie spółki zostaje sama metoda P/S.

Mnożniki są forward (na kolejne 12 miesięcy), bo za rok dzisiejsza prognoza stanie się wynikiem raportowanym. Przy krótkiej historii model schodzi do surowego maksimum i mówi o tym w karcie.

4.2 Ocena jakości (0–100)

Szesnaście kryteriów w czterech blokach: wzrost (przychody 3Y i prognoza, EPS, wzrost EPS), jakość i bilans (marża, ROA, ROIC, konwersja zysku w gotówkę, dług netto/EBITDA, płynność), wycena (P/E, P/S i EV/EBITDA względem własnej mediany) oraz rewizje (cena docelowa i prognozy EPS w 3 miesiące).

- **Progi:** punkty 1–10 z interpolacją między progami — 2,49 i 2,50 nie różnią się o cztery punkty;
- **Braki:** brak danych nie zeruje oceny: wagi renormują się do dostępnych kryteriów;
- **Rating:** A ≥ 75, B ≥ 60, C ≥ 45, D poniżej. Ocena jest liczona wyłącznie z danych — analiza miękka została wyłączona z modelu.

4.3 Wynik portfela

- **Dziennik:** koszt liczony metodą FIFO w walucie instrumentu, a wynik przeliczany na PLN po kursie z dnia każdej transakcji (nie po dzisiejszym);
- **Miary:** TWR — wynik strategii, odporny na wpłaty i wypłaty; MWR (XIRR) — wynik pieniędzy inwestora, zależny od momentu wpłat;
- **Atrybucja:** atrybucja per spółka (zrealizowane + niezrealizowane + dywidendy) i osobny wiersz różnic kursowych, żeby suma zgadzała się z wynikiem portfela.

4.4 Ryzyko

Zmienność i korelacje liczone z 3-letniej historii wspólnych sesji; VaR parametryczny (1,645 × σ) i historyczny (5. percentyl dziennych stóp). Wkład w ryzyko to udział pozycji w zmienności portfela (w·σ/σ_p), a koncentracja — HHI (suma kwadratów wag) i efektywna liczba pozycji.

5. Jak uruchomić

Program jest przewidziany na Windows 10/11 z WebView2 (jest razem z Edge).

Krok	Co zrobić
1	Zainstaluj Python 3.12 i pobierz repozytorium (git clone albo ZIP).
2	W katalogu projektu: python -m venv venv
3	venv\Scripts\pip install -r requirements.txt
4	Uruchom Terminal.bat (albo venv\Scripts\pythonw.exe terminal.py).
5	Przy pierwszym uruchomieniu program utworzy własne pliki stanu.

Ważne: dane nie są częścią repozytorium

Dane fundamentalne pochodzą z FactSetu przez API Scraba i wymagają własnej sesji (plik scrab_session.json z ciasteczkami zalogowanej przeglądarki). Sesja nie jest dołączona do repozytorium — to dane dostępowe, traktowane jak hasło. Bez niej aplikacja uruchomi się, ale zamiast danych pokaże komunikat o brakującej sesji; zrzuty ekranu i dokumentacja pokazują, jak wygląda wypełniona.

Dlatego w repozytorium nie ma też stanu portfela ani dziennika transakcji — są prywatne. Program nie zawiera żadnych danych osobowych ani cudzych.

Skąd wziąć dane — sesja Scraba

Dane fundamentalne wymagają sesji: pliku scrab_session.json z cookies z zalogowanej przeglądarki. Do repozytorium dołączony jest skrypt, który robi to za Ciebie:

1. Zaloguj się na app.scrab.com w Chrome lub Edge.
2. Naciśnij F12 → zakładka Network → odśwież stronę → kliknij dowolne zapytanie do app.scrab.com/data/metric/.
3. W sekcji Request Headers znajdź Cookie i skopiuj całą wartość (ciąg nazwa=wartość; …).
4. Uruchom Sesja-Scrab.bat w katalogu projektu, wklej skopiowany ciąg i naciśnij Enter.
5. Skrypt zapisze plik i sprawdzi sesję zapytaniem do API — zobaczysz „Test API: OK".
6. Zrestartuj aplikację albo kliknij „Odśwież dane".

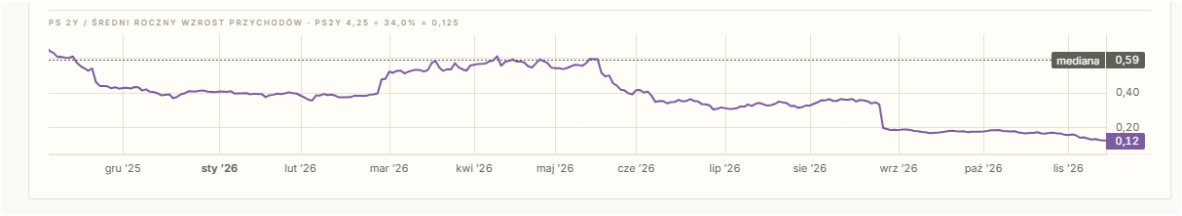
Sesja wygasa po około dwóch tygodniach — wtedy wystarczy powtórzyć kroki 1–6. Plik jest wpisany do .gitignore: zostaje na Twoim komputerze i nigdy nie trafia do repozytorium. Bez niego aplikacja uruchomi się, ale zamiast danych pokaże komunikat o brakującej sesji.

6. Jak czytać wynik — przykład

Karta NVDA z listopada 2026: ocena 95,8 (A), cena docelowa konserwatywna 418,80 USD, optymistyczna 490,78 USD, potencjał +157%. To nie jest prognoza kursu, tylko raport z założeń, i warto to przeczytać w całości:

- **Skąd +157%:** mediana P/E z 3 lat jest znacznie wyższa od dzisiejszego mnożnika — model „dopisuje" powrót do historycznej wyceny; to główne źródło wysokiego potencjału;
- **Wycena:** P/E 0,39× i EV/EBITDA 0,44× własnej mediany — spółka jest tania względem własnej historii, ale to nie znaczy tania w wartościach bezwzględnych;
- **Jakość:** ROIC 72%, konwersja zysku w gotówkę 66% — jakość biznesu wysoka, choć ostatni wskaźnik ciągnie ocenę w dół;
- **Rewizje:** prognozy EPS w górę o 23%, cena docelowa w dół o 10% — analitycy podnoszą zyski, ale tną wycenę, czyli kurs wyprzedził prognozy.

Karta pokazuje też sytuacje nietypowe: brak prognozy EPS (wycena tylko z P/S), wypadnięcie jednej metody z ceny docelowej czy zmianę wersji modelu — zamiast ukrywać je w liczbach, opisuje je słowami.



Panel PS 2Y / wzrost przychodów — wskaźnik autorski: 4,25 ÷ 34,0% = 0,12 przy medianie 0,59.

7. Struktura repozytorium

Plik / katalog	Co zawiera
app.py	API (28 endpointów) i logika widoków
scoring.py, tfl.py	ocena 16 kryteriów, kryteria TFL i rewizje

scrab_client.py	klient FactSet/Scrab: sesja, cache, mnożniki
wyniki.py, perf.py	TWR/MWR, benchmarki, kursy NBP
rebalans.py	wagi docelowe, limity, lista zleceń
karta.py	karta spółki PDF (matplotlib)
web/	frontend: HTML, CSS, fonty, wykresy (bez frameworka)
desktop.py, terminal.py	okno WebView2 bez ramki i launcher
pt.py, batch.py, analiza.py	wcześniejsze skrypty CLI — dziś zastąpione przez aplikację
docs/	dokumentacja: model, architektura, API, operacje + PDF (31 stron)
PRZEWODNIK-po-projekcie.docx	ten dokument

8. Ograniczenia — czego program nie robi

- nie liczy podatku (PIT-38, Belka); wynik portfela jest brutto;
- dziennik nie obsługuje splitów akcji — po splicie trzeba poprawić transakcje;
- zakres dat w widoku Wyniki zmienia wykresy, ale KPI liczone są za cały okres;
- dane są dzienne (zamknięcie), to nie terminal intraday;
- program jest jednoosobowy i lokalny: brak logowania, chmury i wielodostępu;
- wymaga płatnego dostępu do Scraba/FactSet, żeby pobierać nowe dane.

Te punkty są opisane także w dokumentacji technicznej — wolę je nazwać, niż zostawić, żeby odkrył je rekruter.

9. Summary in English

TFAL Terminal is a local desktop application for equity research and personal portfolio accounting, built in Python (Flask) with a WebView2 window.

- **Valuation:** valuation from historical P/E, P/S and EV/EBITDA multiples (3-year window) applied to 12-month forward estimates, with conservative and optimistic scenarios;
- **Scoring:** a 16-criterion quality score across growth, balance sheet, capital returns, cash conversion, relative valuation and estimate revisions;
- **Portfolio:** FIFO cost basis in the instrument currency with P&L converted at the transaction-date FX rate, TWR and MWR (XIRR), attribution, risk (VaR, correlations, HHI) and PDF reports;
- **Scale:** 31-page documentation generated from Markdown, 28 API endpoints, ~9.6k lines of code.

Data comes from FactSet (via the Scrab API), Yahoo Finance and the NBP (Polish central bank) FX rates. Market data and credentials are not included in the repository. Educational project, not investment advice.

10. Jak sprawdzić projekt w pięć minut

- **Przeczytaj** README.md — co robi, jak uruchomić, jakie ma ograniczenia;
- **Zajrzyj do** docs/MODEL.md — wzory wyceny, scoringu i wyników, napisane językiem zrozumiałym bez kodu;
- **Otwórz** docs/Dokumentacja-TFAL.pdf — 31 stron: architektura, API, dane, operacje;
- **Zobacz** screenshots/ — karta spółki i panele wykresów, jeśli nie chcesz pobierać kodu.

Kontakt

Kacper Teichert · kacper.teichert6790@gmail.com · github.com/kicporr

Repozytorium: github.com/kicporr/tfal-terminal · dokumentacja do druku: docs/Dokumentacja-TFAL.pdf