

MikroInwestor - Platforma Symulacji Inwestycyjnych

Dokumentacja projektu zaliczeniowego. Aplikacja webowa oparta na wzorcu **MVC** (Model-View-Controller), służąca do symulacji handlu akcjami i kryptowalutami w czasie rzeczywistym oraz symulowanym.

1. Opis Projektu

System umożliwia użytkownikom zarządzanie wirtualnym portfelem inwestycyjnym. Aplikacja obsługuje dwa tryby danych rynkowych:

- **Tryb Symulowany:** Generowany algorytmicznie (dla celów testowych).
- **Tryb Realny:** Pobierający dane z zewnętrznego API (Finnhub) w czasie rzeczywistym.

Kluczowe aspekty techniczne to wykorzystanie konteneryzacji (Docker), bazy danych PostgreSQL z zaawansowaną logiką (Triggery) oraz czystej architektury PHP bez frameworków.

2. Architektura Systemu

Projekt zrealizowano w architekturze warstwowej, co zapewnia separację logiki biznesowej od warstwy prezentacji i danych.

Struktura Katalogów

```
| .env.example      # Szablon zmiennych środowiskowych
| docker-compose.yaml # Konfiguracja kontenerów
| index.php        # Front Controller
| readme.md       # Dokumentacja
|
|---docker         # Konfiguracja środowiska
|   |---db          # Skrypty SQL i Dockerfile bazy
|   |---nginx        # Konfiguracja serwera WWW
|   |---php          # Konfiguracja obrazu PHP
|
|---public         # Zasoby statyczne (dostępne publicznie)
|   |---scripts      # JS (Timer, Chart.js logic)
|   |---styles        # CSS
```

```

|   └--views      # Szablony widoków (HTML/PHP)
|
└--src          # Logika aplikacji
    ├──controllers  # Kontrolery (App, Project, Security)
    ├──database     # Połączenie z PDO
    ├──factories    # Wzorzec Fabryki (MarketProvider)
    ├──models       # Modele danych (User)
    ├──repository   # Warstwa dostępu do danych (DAO)
    └--services     # Logika biznesowa i integracje API

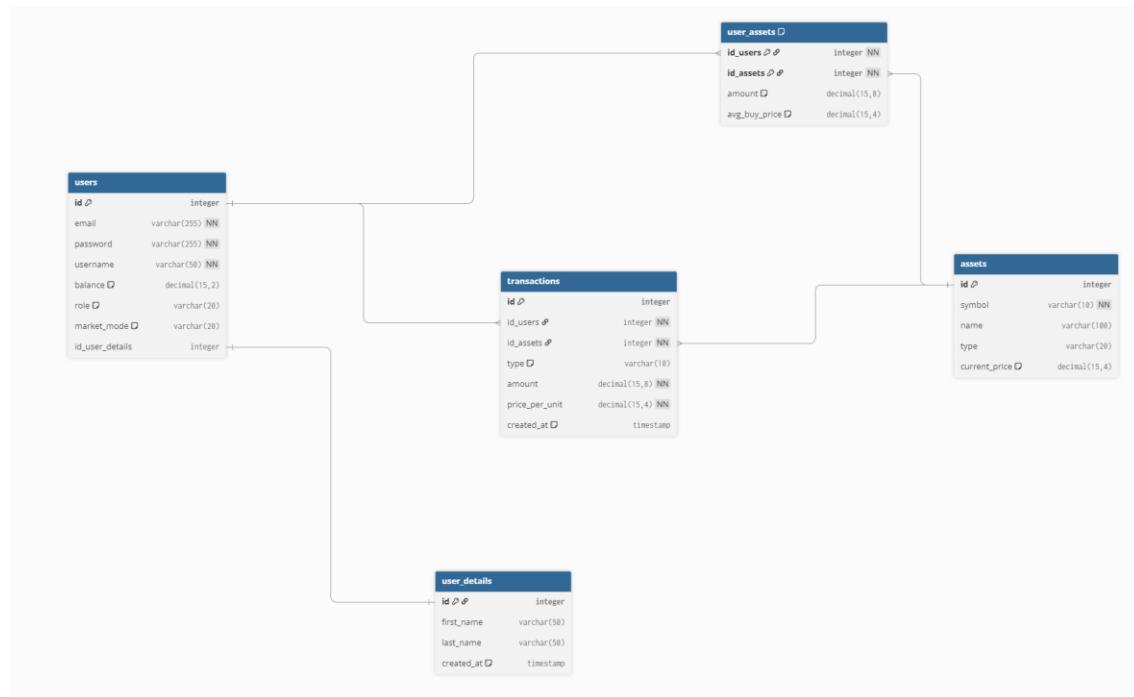
```

3. Baza Danych (ERD)

Baza danych PostgreSQL składa się z 5 powiązanych tabel. Kluczowym elementem jest wykorzystanie **Triggerów** do zapewnienia atomowości operacji finansowych.

Schemat relacji:

1. users (Konta użytkowników + saldo)
2. user_details (Dane osobowe - relacja 1:1)
3. assets (Dostępne instrumenty finansowe)
4. transactions (Historia operacji - relacja 1:N)
5. user_assets (Aktualny stan portfela - relacja N:M)



4. Instalacja i Uruchomienie

Wymagania: Docker oraz Docker Compose.

Krok 1: Konfiguracja

Utwórz plik .env na podstawie przykładu:

Bash

```
cp .env.example .env
```

Upewnij się, że w pliku .env wpisałeś swój klucz API do Finnhub (opcjonalne dla trybu realnego).

Krok 2: Uruchomienie kontenerów

Zbuduj i uruchom środowisko w tle:

Bash

```
docker-compose up -d --build
```

Krok 3: Inicjalizacja Bazy Danych

Baza danych zainicjalizuje się automatycznie przy pierwszym uruchomieniu dzięki skryptowi docker/db/init.sql. Zawiera on strukturę tabel, triggery oraz przykładowe dane.

Aplikacja dostępna jest pod adresem: **<http://localhost:8080>**

5. Scenariusz Testowy (Walkthrough)

Poniższa lista kroków pozwala zweryfikować wszystkie funkcjonalności systemu.

A. Logowanie i Rejestracja

1. Wejdź na stronę główną. Zostaniesz przekierowany do /login.
2. **Test Rejestracji:** Kliknij "Zarejestruj się". Wybierz "Tryb Realny" lub "Symulowany".
 - **Weryfikacja:** Utworzony konto powinno mieć startowy balans **\$151,401.00** (zgodnie z numerem albumu).
3. **Logowanie (Konto Testowe):**
 - User: user@mikroinwestor.pl / Hasło: 1234
 - Admin: admin@mikroinwestor.pl / Hasło: 1234

B. Rynek i Zakupy (Core Feature)

1. Przejdź do zakładki **RYNEK**.
2. Zwróć uwagę na licznik czasu (Timer) w rogu. Ceny są cache'owane w sesji i odświeżają się co 60 sekund (synchronizacja JS + PHP).
3. Wybierz aktywo (np. AAPL) i kliknij "KUP".

4. Wpisz ilość i potwierdź.
 - *Weryfikacja:* System sprawdza, czy masz wystarczające środki.

C. Portfel i Logika Biznesowa (Trigger)

1. Przejdz do zakładki **PORTFEL**.
2. Sprawdz nowo zakupione aktywo.
3. **ROI Test:** Zaraz po zakupie ROI powinno wynosić **0.00%** (cena zakupu = cena rynkowa z sesji).
4. Dokup więcej tego samego aktywa.
 - *Weryfikacja:* Średnia cena zakupu (avg_buy_price) zostanie przeliczona automatycznie przez **Trigger w bazie danych**.

D. Panel Administratora (Role & Security)

1. Wyloguj się i zaloguj jako Admin (admin@mikroinwestor.pl).
2. W menu pojawi się czerwony przycisk **ADMIN PANEL**.
3. Wejdź w panel. Zobaczysz listę użytkowników.
4. **CRUD Test:** Usuń użytkownika testowego.
 - *Weryfikacja:* Usunięcie działa kaskadowo (ON DELETE CASCADE) – znikają też transakcje i portfel tego użytkownika.

E. Obsługa Błędów i Bezpieczeństwo

1. Spróbuj wejść na /admin_panel jako zwykły użytkownik.
 - *Oczekiwany rezultat:* Przekierowanie do dashboardu (Ochrona Guard Clause).
2. Spróbuj wejść na nieistniejący adres np. /random123.
 - *Oczekiwany rezultat:* Obsługa błędu 404 w Routerze.

6. Checklist - Co udało się zrealizować?

Backend (PHP)

- [x] **Wzorzec MVC:** Pełna separacja logiki (Controller) od widoku (View) i danych (Repository).
- [x] **Routing:** Własna implementacja routera obsługująca metody GET/POST.
- [x] **Wzorzec Fabryki (Factory Method):** Klasa MarketProviderFactory dobierająca serwis (Real/Simulation) w zależności od ustawień użytkownika.
- [x] **Wzorzec Strategii/Interfejs:** Wspólny interfejs MarketServiceInterface dla różnych źródeł danych.

- [x] **Cache/Proxy:** Buforowanie cen w sesji (`$_SESSION`) zsynchronizowane z ciasteczkami (cookie), aby zredukować zapytania do API.

Baza Danych (PostgreSQL)

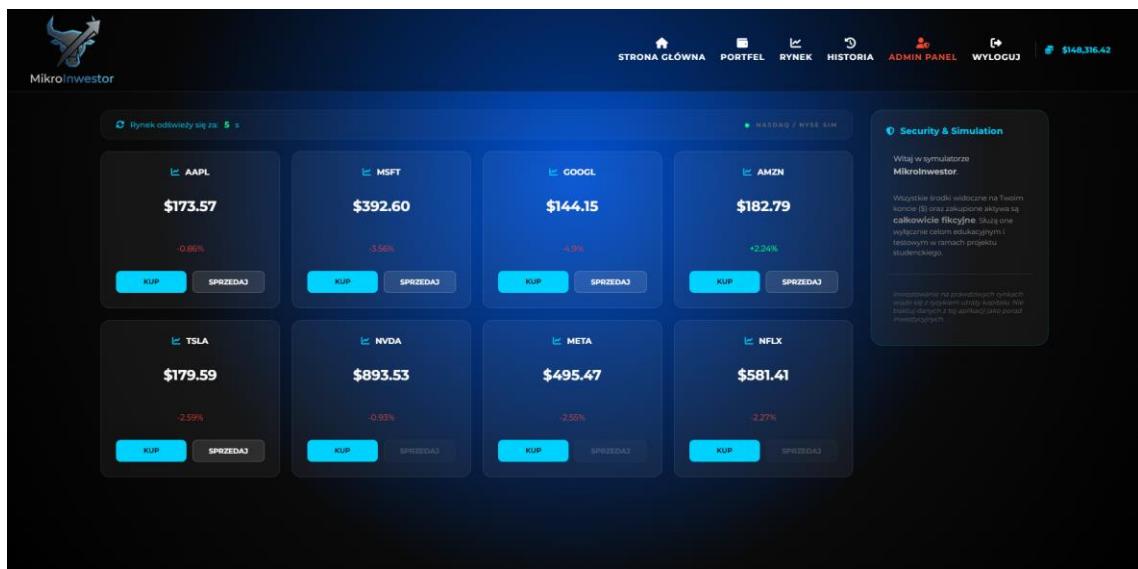
- [x] **Relacyjna struktura:** 5 tabel połączonych kluczami obcymi.
- [x] **Triggery (PL/pgSQL):** Automatyczne obliczanie średniej ceny zakupu (`avg_buy_price`) po każdej transakcji `INSERT`.
- [x] **Kaskadowe usuwanie:** `ON DELETE CASCADE` zapewniające spójność danych.
- [x] **Bezpieczeństwo:** Hasła hashowane algorymem **Bcrypt**.

Frontend

- [x] **Dynamiczne Wykresy:** Integracja **Chart.js** z danymi pobieranymi asynchronicznie (AJAX/Fetch).
- [x] **Responsywność:** CSS Grid/Flexbox (Własne style, brak Bootstrapa).
- [x] **UX:** Licznik czasu do odświeżenia rynku, walidacja formularzy.

7. Zrzuty Ekranu

Dashboard i Rynek



MikroInwestor

STRONA GŁÓWNA PORTFEL RYNEK HISTORIA ADMIN PANEL WYLOGU \$148,316.42

Rynek odwzajemnia się za: \$1 s

NASDAQ / NYSE SIM

Eksploruj Rynek
Wybierz aktywa i zacznij budować swoją historię inwestycyjną

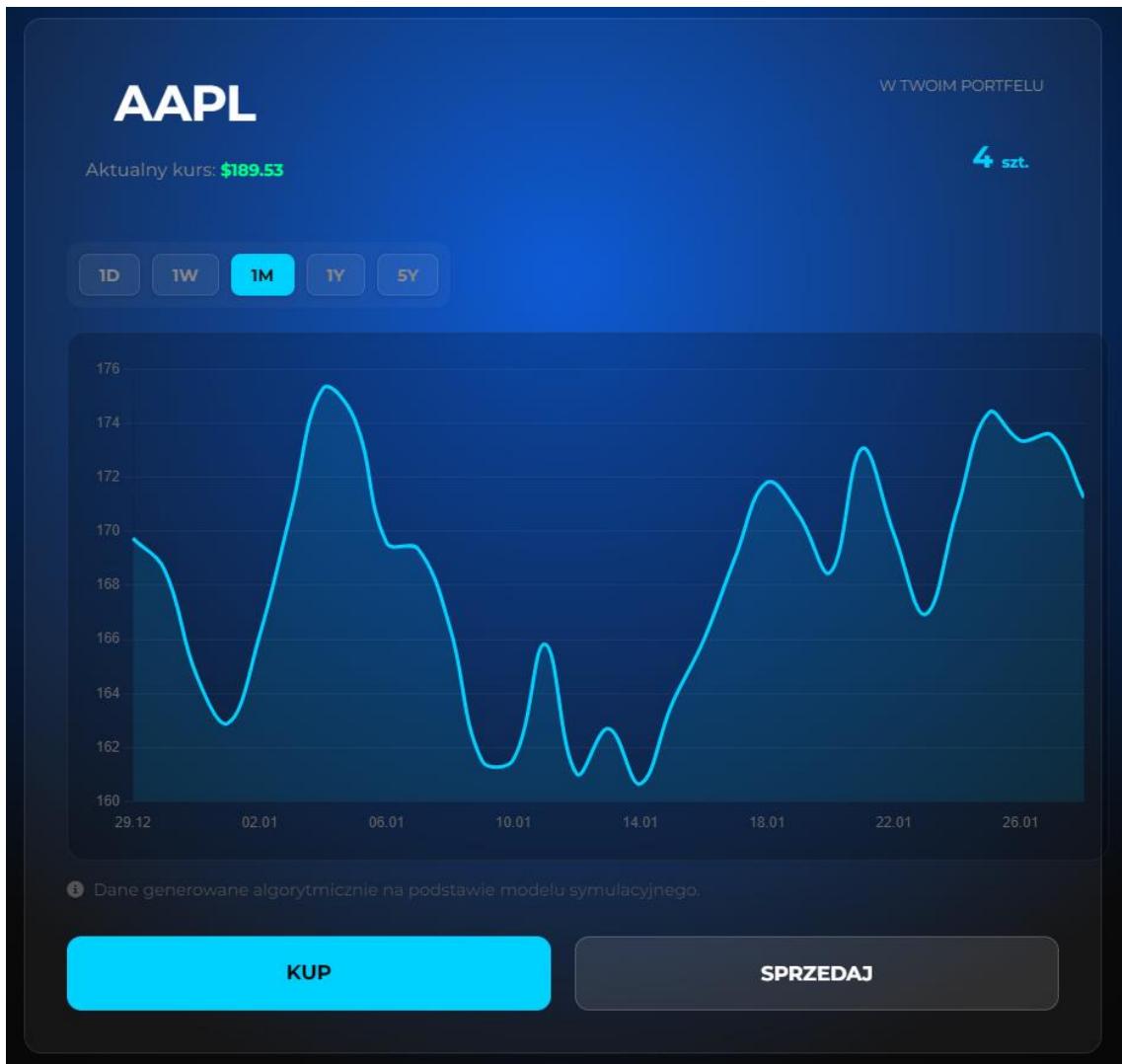
AKTYWO	CENA (USD)	ZMIANA	HANDEL
AAPL	\$189.53	+ 4.39%	KUP SPRZEDAJ
MSFT	\$418.23	+ 1.19%	KUP SPRZEDAJ
GOOGL	\$143.87	+ 2.09%	KUP SPRZEDAJ
AMZN	\$174.54	- 4.57%	KUP SPRZEDAJ
TSLA	\$164.52	+ 3.32%	KUP SPRZEDAJ
NVDA	\$882.72	+ 3.96%	KUP SPRZEDAJ

Security & Simulation
Witaj w symulatorze MikroInwestor.

Możesz teraz przejść do Twoim koniecznym lub zakupionym aktywów całkowicie fikcyjne. Składa one wyłącznie celom edukacyjnym i testowym w ramach projektu studenckiego.

Inwestowanie na przedstawionych rynkach może się zyskiem utrać kapitału. Aby traktować danymi z tej aplikacji jako poziom bieżącego.

Szczegóły Aktywa (Wykres)



Panel Administratora

The screenshot shows the MikroInvestor website interface. At the top, there is a navigation bar with icons for Home, Portfolio, Market, History, Admin Panel, and Logout, along with a balance of \$149,316.42. On the left, there is a sidebar titled "Zarządzanie Użytkownikami" with a note about managing all registered investors. The main content area displays a table of users:

ID	USERNAME	EMAIL	ROLA	AKCJE
1	admin	admin@example.com	ADMIN	EDYCJA
4	real	real@gmail.com	USER	USUN