

Projekt został wykonany przy użyciu PHP 8.3 oraz frameworka Symfony + MySQL. Całość działa w środowisku zbudowanym przy pomocy Dockera, dzięki czemu wszystkie serwisy można uruchomić osobno i komunikują się ze sobą w sieci dockerowej.

W projekcie znajdują się trzy oddzielne serwisy:

1. **Auth Service** – odpowiedzialny za rejestrację oraz logowanie użytkowników. Po poprawnym zalogowaniu generowany jest token JWT, który jest następnie wysyłany do frontendu i później wykorzystywany w zapytaniach do pozostałych serwisów.
2. **Items Service** – realizuje typowe operacje CRUD na przedmiotach (items). Można dodawać nowe przedmioty, edytować istniejące, usuwać je oraz wyświetlać listę wszystkich.
3. **Logs Service** – przechowuje logi wszystkich operacji wykonywanych na itemach. Logowane są takie akcje jak odczyt, tworzenie, modyfikowanie i usuwanie.

Dodatkowo został stworzony **Gateway**, czyli serwis, który stanowi jedno wejście do całej aplikacji. Frontend komunikuje się tylko z gatewayem, a gateway na podstawie ścieżki requestu wysyła zapytanie do odpowiedniego mikroservisu i zwraca jego odpowiedź.

---

## Frontend

Frontend został wykonany w React. Obsługuje logowanie, wyświetlanie danych oraz komunikację z gatewayem. Po zalogowaniu token JWT jest trzymany w cookies, a następnie ustawiany w nagłówkach zapytań HTTP.

Użytkownik po wejściu do aplikacji trafia najpierw na ekran logowania lub rejestracji.

---

Logowanie:

[Home](#) [ZALOGUJ SIĘ](#) [ZAREJESTRUJ SIĘ](#)

Logowanie

Email \*

uzytnik123@mail.pl

Hasło \*

\*\*\*\*\*

ZALOGUJ SIĘ

---

## Rejestracja:

[Home](#) [ZALOGUJ SIĘ](#) [ZAREJESTRUJ SIĘ](#)

Rejestracja

Email \*

Nazwa użytkownika \*

uzytnik123@mail.pl

Hasło \*

\*\*\*\*\*

ZAREJESTRUJ SIĘ

---

Po pomyślnym zalogowaniu można przejść do części aplikacji związanej z itemami. Dostępna jest lista wszystkich przedmiotów, a obok każdego można wykonać akcje:

- edycja,
- usunięcie.
- dodanie nowego

Tworzenie nowego itemu odbywa się poprzez wypełnienie formularza.

---

## Wyświetlanie przedmiotów:

[Home](#) [WYLOGUJ SIĘ](#) [PRZEDMIOTY](#) [LOGI](#)

Lista Przedmiotów

Nazwa \*

Cena \*

DODAJ PRZEDMIOT

Banan  
Cena: 6

EDYTUJ

Kawa  
Cena: 31

EDYTUJ

---

## Dodawanie przedmiotu:

Lista Przedmiotów

Nazwa \*

Przedmiot

Cena \*

12

DODAJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Lista Przedmiotów

Nazwa \*

Cena \*

DODAJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Przedmiot

Cena: 12

EDYTUJ

Usuwanie przedmiotu:

Lista Przedmiotów

Nazwa \*

Cena \*

DODAJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Przedmiot

Cena: 12

EDYTUJ

Lista Przedmiotów

Nazwa \*

Cena \*

DODAJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Edycja przedmiotu:

Lista Przedmiotów

Nazwa \*

Cena \*

DODAJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Lista Przedmiotów

Nazwa \*

Banan

Cena \*

99

EDYTUJ PRZEDMIOT

Banan

Cena: 8

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Lista Przedmiotów

Nazwa \*

Cena \*

ODDAJ PRZEDMIOT

Banan

Cena: 99

EDYTUJ

Kawa

Cena: 31

EDYTUJ

Logi zostały pokazane w formie tabeli. Zawierają takie dane jak ID logu, endpoint, status oraz timestamp wykonanej akcji. Poza tym kolor wiersza zmienia się w zależności od typu operacji (np. CREATE, DELETE, READ itd.).

Wyświetlanie logów:

Logi API

ID	Endpoint	Status	Timestamp
1	/api/items	READ	2025-11-30 15:23:39
2	/api/items/1	READ	2025-11-30 15:23:53
3	/api/items	CREATE	2025-11-30 15:25:02
4	/api/items/2	UPDATE	2025-11-30 15:25:31
5	/api/items/2	DELETE	2025-11-30 15:26:04
6	/api/items	READ	2025-12-02 17:38:45
7	/api/items	READ	2025-12-02 17:38:45
8	/api/items	CREATE	2025-12-02 17:39:26
9	/api/items	READ	2025-12-02 17:39:26
10	/api/items/1	UPDATE	2025-12-02 17:39:33

