Baselinker

Autor: Gniewomir Bartkowiak

Link do programu: https://github.com/Gniewo1/Baselinker

Wstęp:

Aplikacja powstała w ramach zaliczenia jednego z przedmiotów na 7 semestrze studiów. Zaliczenie polegało na napisaniu dwóch programów. U mnie zostały one połączone w ramach jednej strony internetowej.

Pierwszym zadaniem było stworzenie strony internetowej albo aplikacji mobilnej (w tym przypadku strony internetowej), która łączy się przez API z wcześniej założonym kontem na platformie BaseLinker (stąd też nazwa programu). Strona/aplikacja miała posiadać system rejestracji oraz logowania. Zapisywać dane zamówień z BaseLinkera w bazie danych i wyświetlać je. Strona miała także być wstanie zmieniać status zamówień, zarówno w bazie danych strony jak i na koncie BaseLinker.

Strona powstała przy użyciu ***Django Rest Framework***(backend), biblioteki ***React***(frontend) oraz ***SQLite***(baza danych). Jeżeli się nie mylę (nie testowałem tego), żeby aplikacja działała należy zainstalować Pythona, Django oraz Reacta. Strona korzysta z innych bibliotek, ale jeśli chodzi o frontend są one w plikach projektu. Backend należy przed uruchomieniem serwera włączyć virtual envornment, inaczej będą wyskakiwać błędy o braku zainstalowanej biblioteki(więcej).

Strona korzysta także z API dostępnej w BaseLinkerze. https://api.baselinker.com . Dostępna jest tam dokumentacja oraz pozwala wypróbować różne zapytania.

Nie będę się rozwodził nad systemem logowania oraz rejestracji, bo jest to gorsza kopia z innego projektu(tutaj). Najważniejsze funkcje to pobieranie zamówień z BaseLinkera, wyświetlanie ich oraz aktualizacja. Pragnę dodać, że BaseLinker daje tylko 14 darmowych dni korzystania, więc token w programie może nie działać.

Strona pobiera zamówienia poprzez funkcję fetch_orders w views.py. Dane pobrane wyglądają następująco

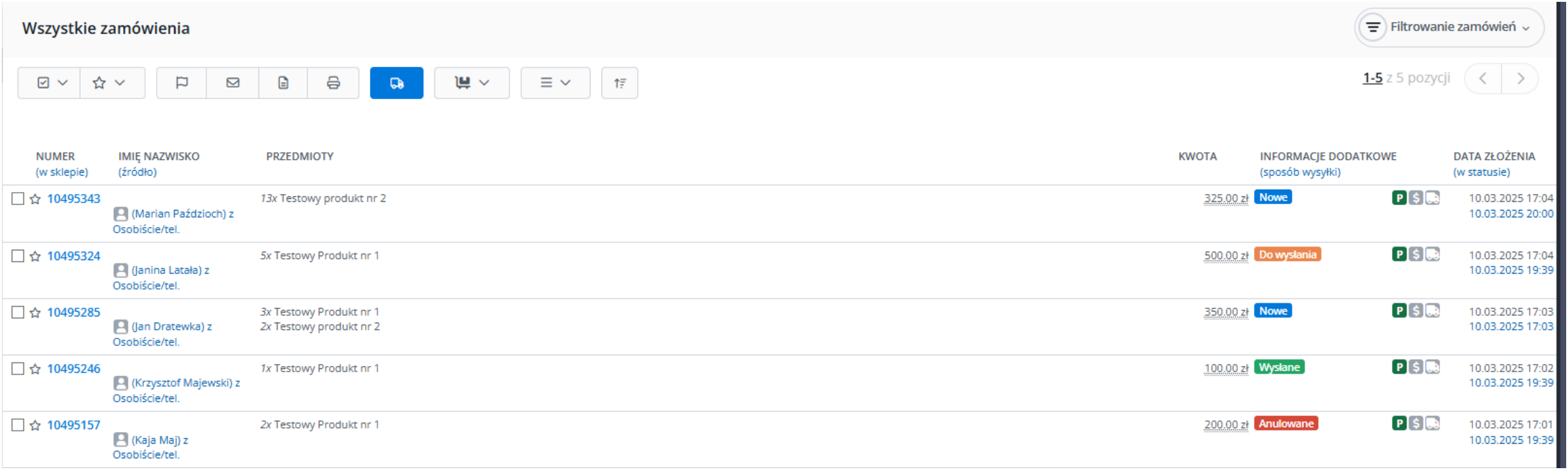

```
{'status': 'SUCCESS', 'orders': [{'order_id': 10495157, 'shop_order_id': 0, 'external_order_id': '', 'order_source': 'personal', 'order_source_id': 0, 'order_source_info': '-', 'order_status_id': 215954, 'confirmed': True
, 'date_confirmed': 1741632229, 'date_add': 1741622467, 'date_in_status': 1741631996, 'user_login': 'Kaja Maj', 'phone': '+48909879873', 'email': 'Kajka@emial.com', 'user_comments': '', 'admin_comments': '', 'currency': '
PLN', 'payment_method': 'Karta', 'payment_method_cod': '0', 'payment_done': 0, 'delivery_method': '', 'delivery_price': 0, 'delivery_package_module': '', 'delivery_package_nr': '', 'delivery_fullname': '', 'delivery_compa
ny': '', 'delivery_address': 'Gronowska 90/9', 'delivery_city': 'Szczecin', 'delivery_state': '', 'delivery_postcode': '70-120', 'delivery_country_code': 'PL', 'delivery_point_id': '', 'delivery_point_name': '', 'delivery
_point_address': '', 'delivery_point_postcode': '', 'delivery_point_city': '', 'invoice_fullname': '', 'invoice_company': '', 'invoice_nip': '', 'invoice_address': '', 'invoice_city': '', 'invoice_state': '', 'invoice_pos
tcode': '', 'invoice_country_code': '', 'want_invoice': '0', 'extra_field_1': '', 'extra_field_2': '', 'order_page': 'https://orders-f.baselinker.com/10495157/fb0y7ur80n/', 'pick_state': 0, 'pack_state': 0, 'delivery_coun
try': 'Polska', 'invoice_country': '', 'products': [{'storage': 'db', 'storage_id': 46175, 'order_product_id': 17288882, 'product_id': '224353145', 'variant_id': '0', 'name': 'Testowy Produkt nr 1', 'attributes': '', 'sku
': '', 'ean': '', 'location': '', 'warehouse_id': 70840, 'auction_id': '0', 'price_brutto': 100, 'tax_rate': 23, 'quantity': 2, 'weight': 100, 'bundle_id': 0}]}, {'order_id': 10495246, 'shop_order_id': 0, 'external_order_
```


Jako, że BaseLinker zwraca dużo danych (w moim przypadku często pustych), a zadanie nie precyzowało, które dane mam zapisać. Ilość ich została ograniczona. Pragnę zwrócić uwagę na order_status_id. Zastowowałem mapping, żeby zmieniło id na nazwę statusu (Ciekawostka: drugie konto założone w Baselinkerze ma inne order_status_id. Pierwsze z tego co pamiętam zaczynał się na 189...). Drugą, rzeczą jest to, że BaseLiner nie podaje ile łącznie było do zapłaty, dlatego trzeba samemu to obliczyć.

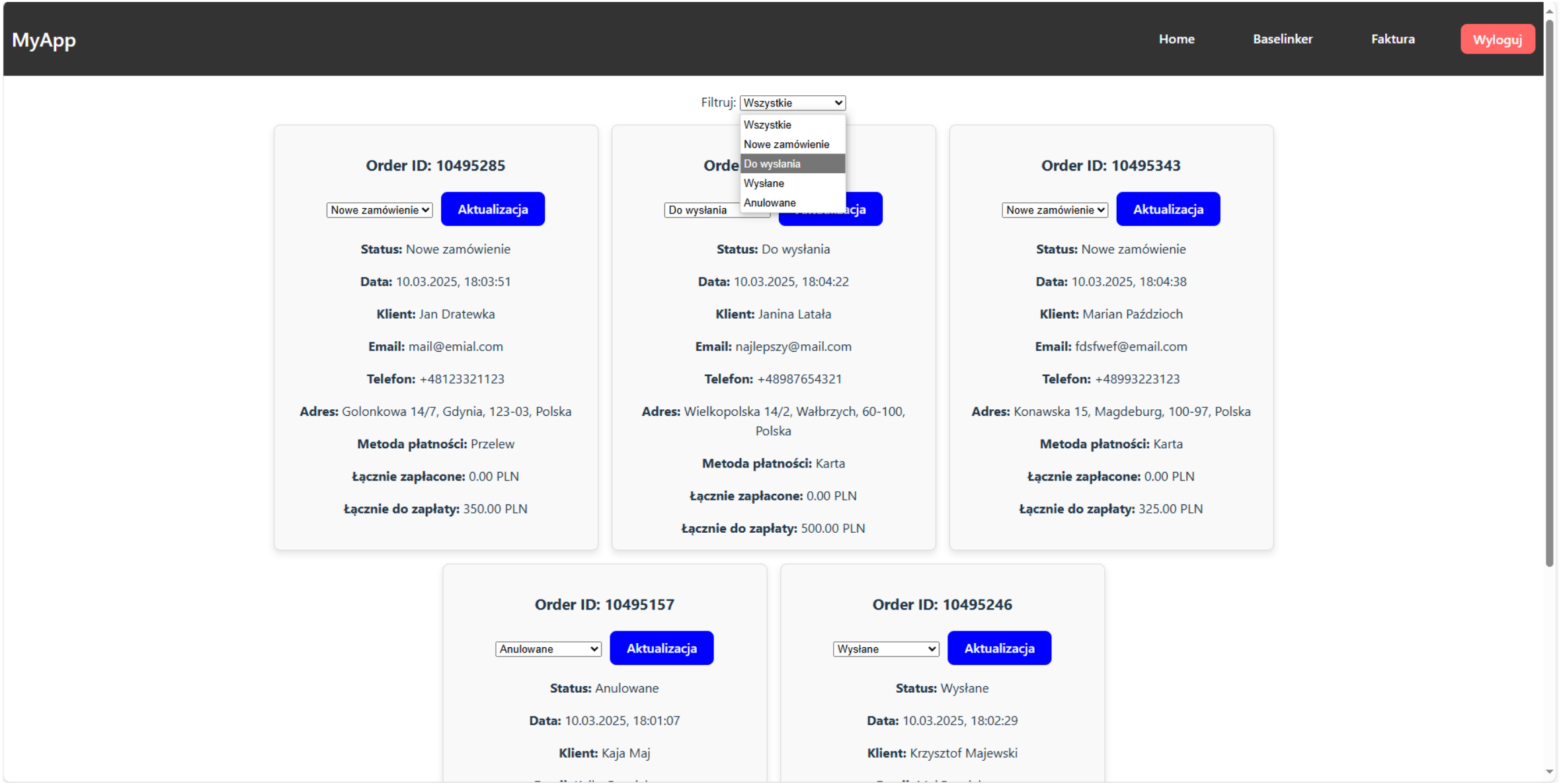
```
76      # Nowe konto może mieć inne id
77      status_mapping = {
78          215951: "Nowe zamówienie",
79          215952: "Do wysłania",
80          215953: "Wysłane",
81          215954: "Anulowane",
82      }
```

```
99          # Oblicza całość zapłaty
100         for item in order_data['products']:
101             price = item['price_brutto']*item['quantity']
102             total_cost+=price
```

Wygląd zamówień w Baselinkerze



Wygląd zamówień w aplikacji(FetchData.jsx). Dodałem też filtr, który pokazuje zamówienia ze względu na status. Strona pozwala na zmianę statusu zamówienia. Status zmienia się na stronie, jak i na koncie BaseLinkera.



Drugą częścią było napisanie programu, który wyczyta dane z pliku pdf faktury z firm kurierskich. Dane takie jak nr dokumentu oraz łączny koszt przesyłki/łek. Forma była dowolna, ale że miałem już stronę to dodałem to do strony i stąd też w pasku nawigacyjnym „Faktura”. Sama strona „po prostu działa”.

Jeśli chodzi o sam program, to najprawdopodobniej najprostszym sposobem byłoby wysłać pdf-a przez API do ChataGPT, ale to kosztuje. Znalazłem, więc inne rozwiązanie.
Najpierw należy przerobić plik pdf na Stringa, po przez bibliotekę „fitz”.

```
140  def extract_text_from_pdf(pdf_path):
141      with fitz.open(pdf_path) as pdf_file:
142          text = ""
143          for page_num in range(pdf_file.page_count):
144              page = pdf_file[page_num]
145              text += page.get_text()
146      return text
```

```
Numer dokumentu:
POZ3749513
Numer Klienta:
427266210
NIP:
[redacted]
Data wystawienia:
30-09-24
Strona:
1 z 2
Zapytania i reklamacje:
WWW:
www.dhl.com.pl/kontakt
WWW:
https://reklamacje.dhlexpress.pl/
Nabywca:
[redacted]
POLSKA
ECONOMY SELECT
3
19,00
3
282,80
106,70
0,00
389,50
389,50
Razem
3
19,00
3
282,80
106,70
0,00
389,50
389,50
OPLATA PALIWOWA
82,70
REMOTE AREA
24,00
Razem opłaty dodatkowe
106,70
```

Dalej się nie zmieściło

Przykładowy wynik widać po lewej stronie. Jak widać nr dokumenty występuje zaraz po „Numer dokumentu:”. Znalazłem więc bibliotekę „re”, która umożliwia pobrania ze Stringa pierwszego słowa/słów, które pojawiają się przed lub po frazie którą wpiszesz

```
def extract_DHL(text):
    match = re.search(r"(\d+,\d{2})\s*Razem: PLN:", text)
    match2 = re.search(r"Numer dokumentu:\s*(\S+)", text)
    if match:
        PLN =  match.group(1)
    else:
        PLN = "Not found"
    if match2:
        ID = match2.group(1)
    else:
        ID = "Not found"
    return PLN, ID
```