Baselinker

Author: Gniewomir Bartkowiak

Link to program: https://github.com/Gniewo1/Baselinker

Entry:

The application was created as part of the final project for one of the courses in the 7th semester of my studies. The project involved writing two programs, which I combined into a single website.

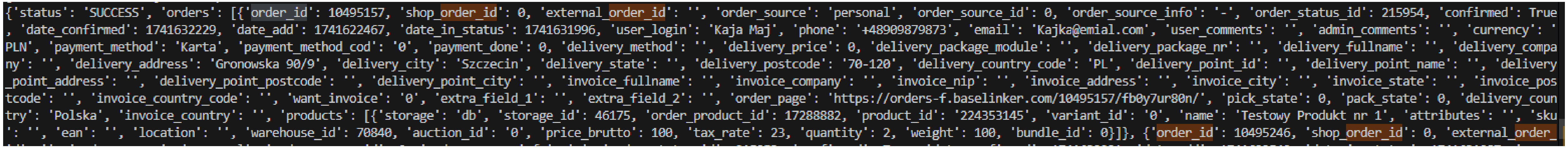
The first task was to create a website or mobile application (in this case, a website) that connects via an API to an account previously created on the BaseLinker platform (hence the name of the program). The website/application was required to have a registration and login system. It needed to store order data from BaseLinker in the database and display it. The website should also be able to change the order status, both in the website's database and on the BaseLinker account.

The website was built using ***Django Rest Framework*** (backend), the ***React*** (frontend), and ***SQLite*** (database). If I'm not mistaken (I haven't tested this), in order for the application to work, Python, Django, and React need to be installed. The website uses other libraries, but as for the frontend, they are included in the project files. The backend needs to activate the virtual environment before starting the server; otherwise, errors will occur regarding missing libraries.

The website also uses the API available in BaseLinker: https://api.baselinker.com. Documentation is available there, and it allows you to try out various queries.

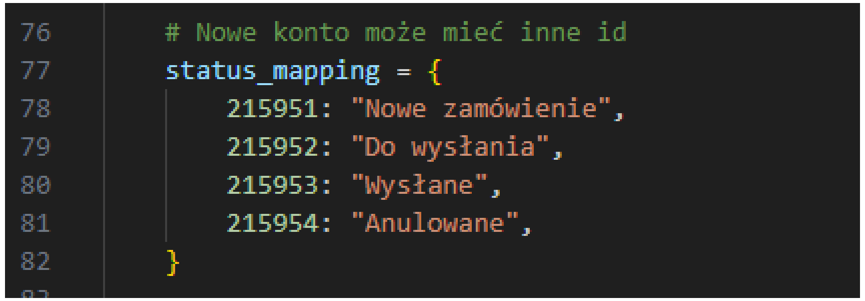
I won’t go into details about the login and registration system, as it’s just a poor copy from another project (here). The most important features are fetching orders from BaseLinker, displaying them, and updating them. I should also mention that BaseLinker only offers 14 free days of usage, so the token in the program may no longer work.

The website fetches orders using the ***fetch_orders*** function in ***views.py***. The retrieved data looks as follows:

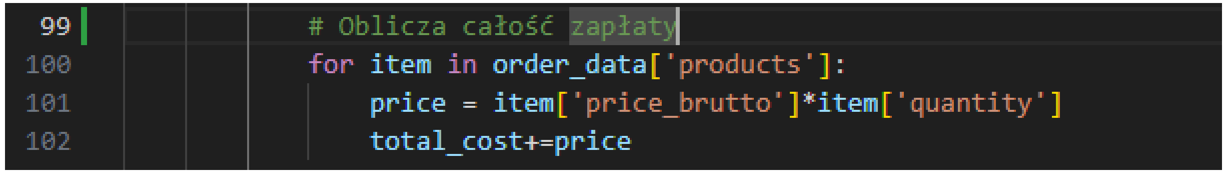


```
{'status': 'SUCCESS', 'orders': [{'order_id': 10495157, 'shop_order_id': 0, 'external_order_id': '', 'order_source': 'personal', 'order_source_id': 0, 'order_source_info': '-', 'order_status_id': 215954, 'confirmed': True
, 'date_confirmed': 1741632229, 'date_add': 1741622467, 'date_in_status': 1741631996, '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_
```

Since BaseLinker returns a lot of data (in my case, often empty), and the task didn’t specify which data I should save, the amount has been limited. I’d like to point out the ***order_status_id***. I implemented a mapping to change the ID to the status name (fun fact: the second account created in BaseLinker has a different ***order_status_id***. The first one, if I remember correctly, started at 189…). Another thing is that BaseLinker doesn’t provide the total amount due, so that has to be calculated manually.

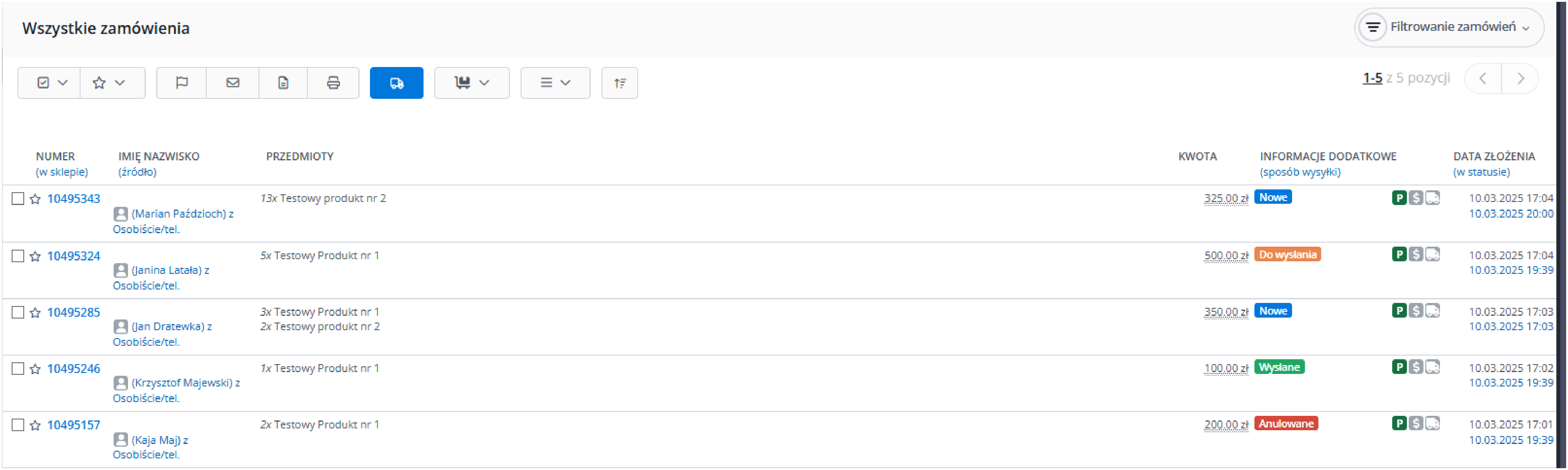


```
# Nowe konto może mieć inne id
status_mapping = {
    215951: "Nowe zamówienie",
    215952: "Do wysłania",
    215953: "Wysłane",
    215954: "Anulowane",
}
```



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

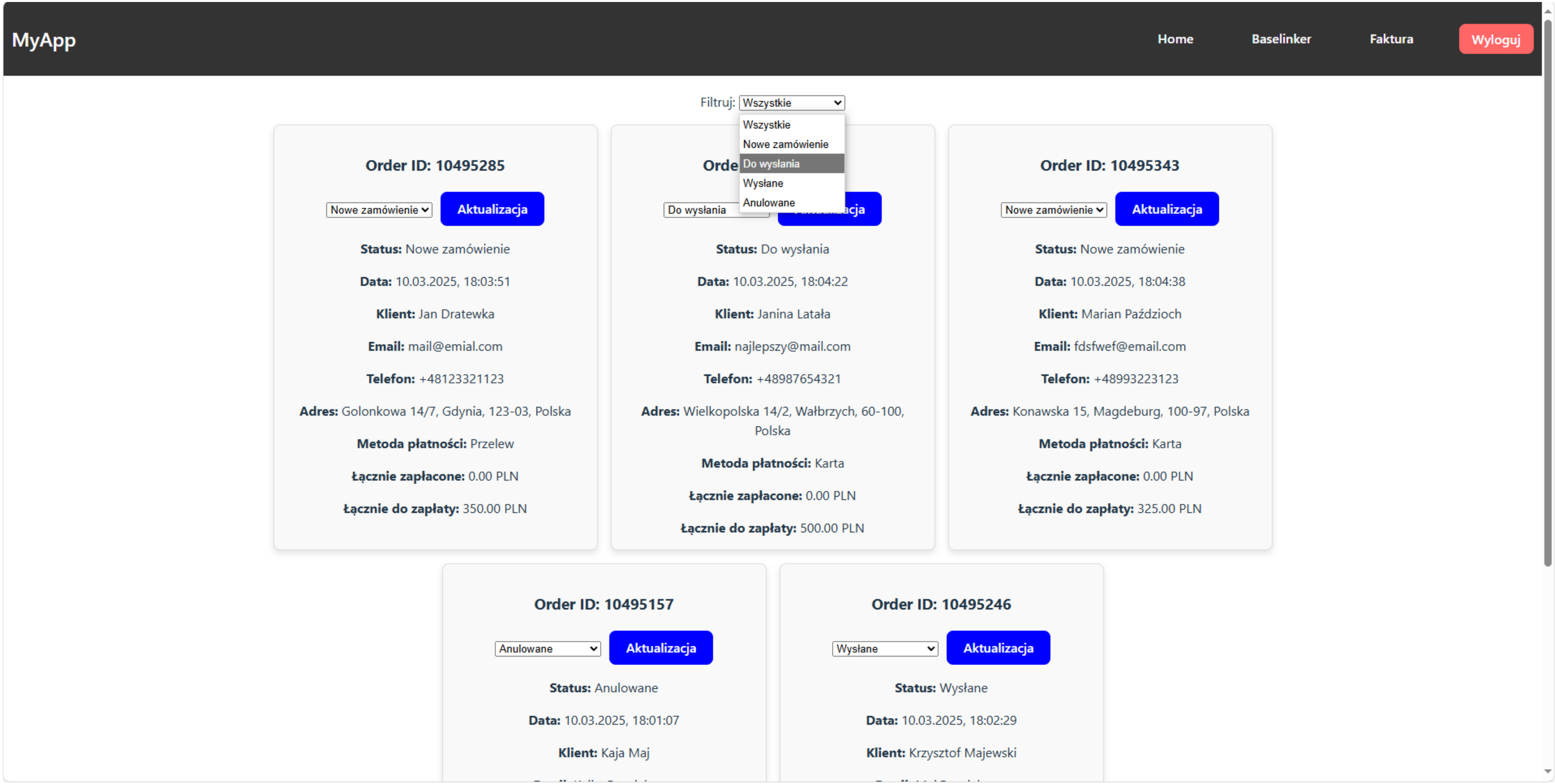
View of orders in Baselinker



Wszystkie zamówienia

NUMER (w sklepie)	IMIĘ NAZWISKO (źródło)	PRZEDMIOTY	KWOTA	INFORMACJE DODATKOWE (sposób wysyłki)	DATA ZŁOŻENIA (w statusie)
□ ☆ 10495343	(Marian Paździoch) z Osobiście/tel.	*13x* Testowy produkt nr 2	325.00 zł	Nowe	10.03.2025 17:04 10.03.2025 20:00
□ ☆ 10495324	(Janina Latała) z Osobiście/tel.	*5x* Testowy Produkt nr 1	500.00 zł	Do wysłania	10.03.2025 17:04 10.03.2025 19:39
□ ☆ 10495285	(Jan Dratewka) z Osobiście/tel.	*3x* Testowy Produkt nr 1 *2x* Testowy produkt nr 2	350.00 zł	Nowe	10.03.2025 17:03 10.03.2025 17:03
□ ☆ 10495246	(Krzysztof Majewski) z Osobiście/tel.	*1x* Testowy Produkt nr 1	100.00 zł	Wysłane	10.03.2025 17:02 10.03.2025 19:39
□ ☆ 10495157	(Kaja Maj) z Osobiście/tel.	*2x* Testowy Produkt nr 1	200.00 zł	Anulowane	10.03.2025 17:01 10.03.2025 19:39

The appearance of orders in the application (FetchData.jsx). I also added a filter that shows orders based on their status. The website allows changing the order status. The status is updated both on the website and on the BaseLinker account.



Filtruj: Wszystkie

Order ID: 10495285

Nowe zamówienie

Status: Nowe zamówienie

Data: 10.03.2025, 18:03:51

Klient: Jan Dratewka

Email: mail@emial.com

Telefon: +48123321123

Adres: Golonkowa 14/7, Gdynia, 123-03, Polska

Metoda płatności: Przelew

Łącznie zapłacone: 0.00 PLN

Łącznie do zapłaty: 350.00 PLN

Orde

Do wysłania

Status: Do wysłania

Data: 10.03.2025, 18:04:22

Klient: Janina Latała

Email: najlepszy@mail.com

Telefon: +48987654321

Adres: Wielkopolska 14/2, Wałbrzych, 60-100, Polska

Metoda płatności: Karta

Łącznie zapłacone: 0.00 PLN

Łącznie do zapłaty: 500.00 PLN

Order ID: 10495343

Nowe zamówienie

Status: Nowe zamówienie

Data: 10.03.2025, 18:04:38

Klient: Marian Paździoch

Email: fdsfwef@email.com

Telefon: +48993223123

Adres: Konawska 15, Magdeburg, 100-97, Polska

Metoda płatności: Karta

Łącznie zapłacone: 0.00 PLN

Łącznie do zapłaty: 325.00 PLN

Order ID: 10495157

Anulowane

Status: Anulowane

Data: 10.03.2025, 18:01:07

Klient: Kaja Maj

Order ID: 10495246

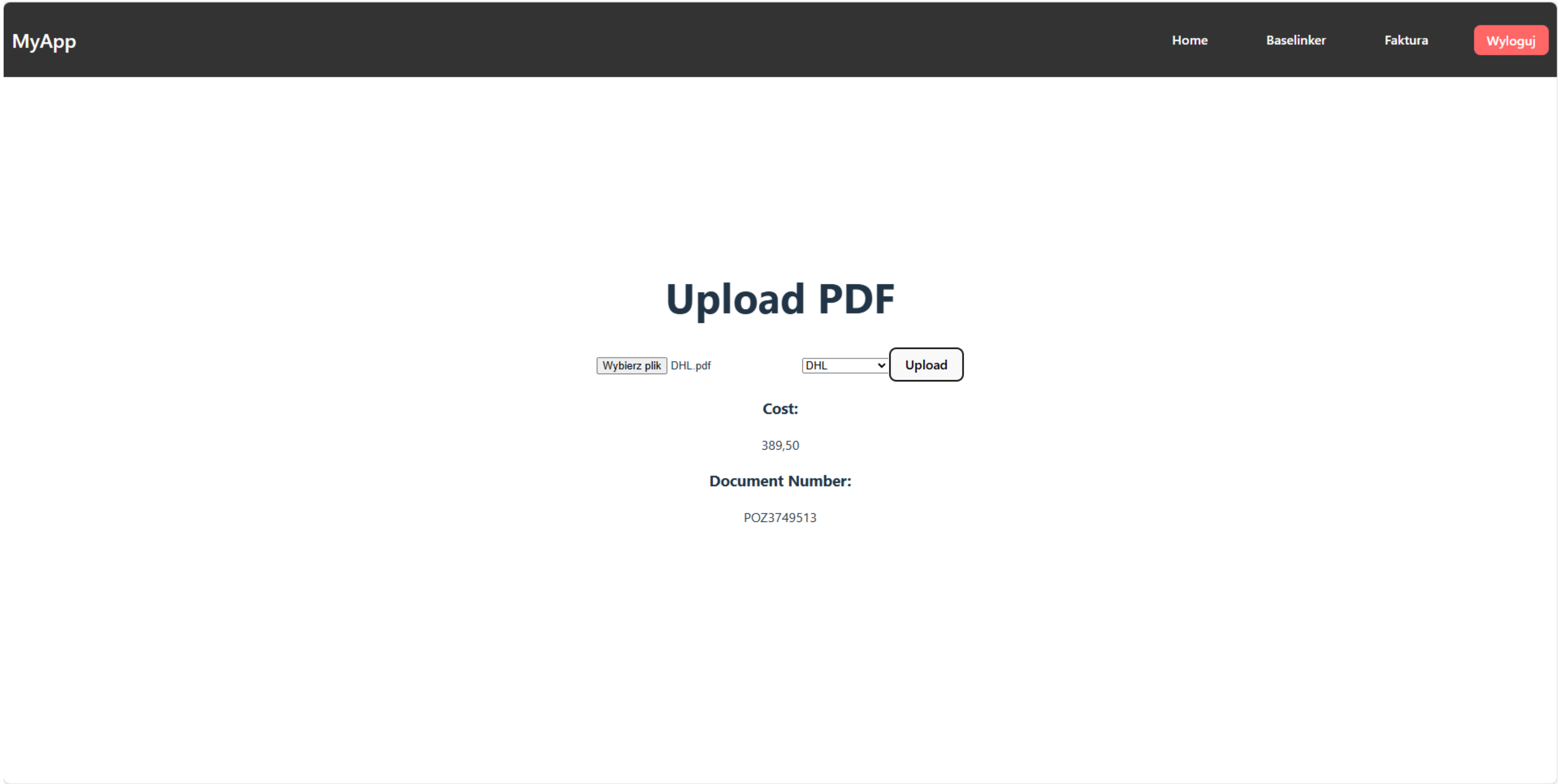
Wysłane

Status: Wysłane

Data: 10.03.2025, 18:02:29

Klient: Krzysztof Majewski

The second part was to write a program that extracts data from a courier company’s invoice PDF, such as the document number and the total shipping cost. The format was up to me, but since I already had a website, I added this feature to it, which is why there is a "Invoice" option in the navigation bar. The page "just works."



Upload PDF

Cost:

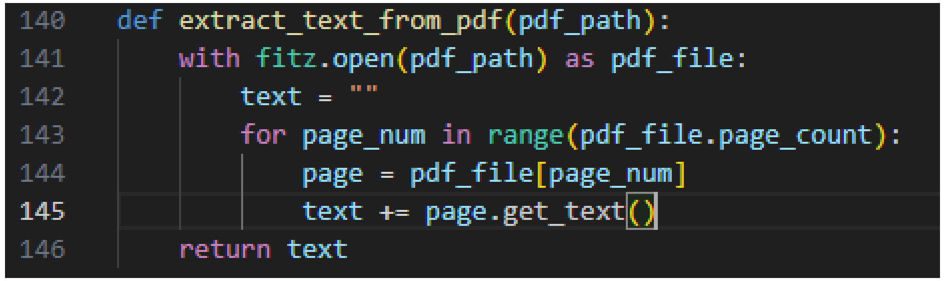
389,50

Document Number:

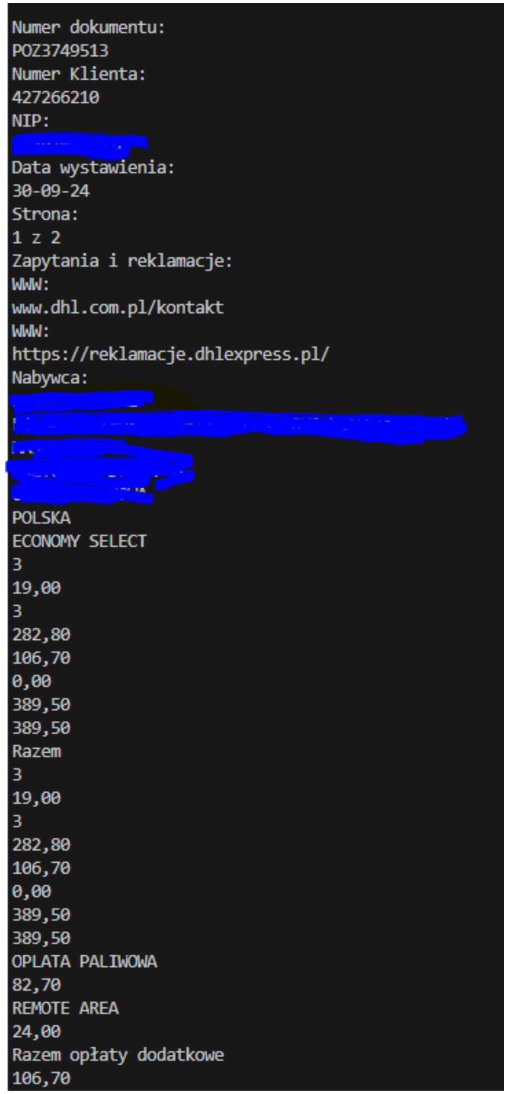
POZ3749513

As for the program itself, the easiest way would probably be to send the PDF through the API to ChatGPT, but that costs money. So, I found another solution.

First, the PDF file needs to be converted to a string using the "fitz" library.



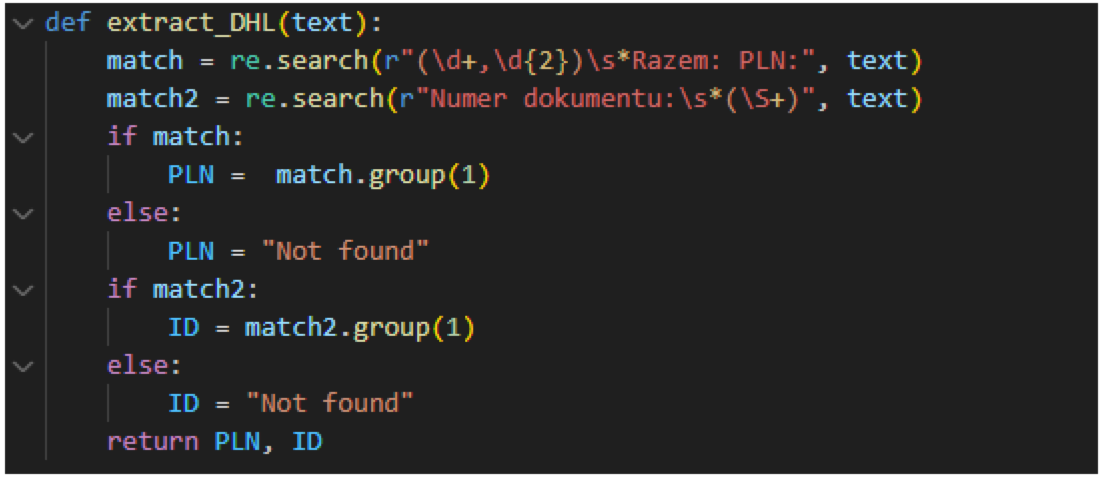
```
def extract_text_from_pdf(pdf_path):
    with fitz.open(pdf_path) as pdf_file:
        text = ""
        for page_num in range(pdf_file.page_count):
            page = pdf_file[page_num]
            text += page.get_text()
    return text
```



```
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

An example result is shown on the left. As you can see, the document number appears right after "Numer dokumentu:". I found the "re" library, which allows you to extract the first word(s) from a string that appear before or after a phrase you specify.



```
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
```