- REST

- <https://realpython.com/api-integration-in-python/>

- Python testing with PyTest

<https://realpython.com/pytest-python-testing/>

- DJANGO

<https://realpython.com/digging-deeper-into-migrations/>

<https://realpython.com/django-user-management/>

<https://realpython.com/django-view-authorization/>

<https://www.django-rest-framework.org/>

https://django-allauth.readthedocs.io/en/latest/

- Python

<https://realpython.com/inheritance-composition-python/>

<https://realpython.com/modeling-polymorphism-django-python/>

<https://realpython.com/python-interface/>

Wzorce projektowe:

https://refactoring.guru/design-patterns/decorator

1. Wybór technologii:

- Flask vs **Django** vs FastAPI

<https://www.section.io/engineering-education/choosing-between-django-flask-and-fastapi/>

1. Docker compose? Mikroserwisy? Czy mikroserwisy można umieścić w kontenerach i użyć docker composa?
2. Test Driven Development?
3. Szablony maili – format, przechowywanie, zmienne do podstawienia?
4. Jeśli autoryzacja ma być osobnym mikroserwisem, to czy też dwa osobne repozytoria na githubie?
5. Jeśli TDD, to czy testowanie przy pomocy Postmana?
6. Mikroserwisy vs kontenery vs api w Django
7. Wzorce projektowe

- class MailSender(ABC)

Def send\_mail():

- class

1. Providerzy serwisu mailowego
2. Django allauthlib?
3. ClassBasedViews czy to drugie?
4. Czy jak loguję się przez serwis zewnętrzny, to potrzebuję bazę danych użytkowników?

PLAN DZIAŁANIA:

1. Serwis autoryzacyjny Django
2. Postawienie aplikacji Django
3. Wysyłanie najprostszego testowego maila poprzez gmaila
4. Wysyłanie najprostszego maila poprzez dwóch innych providerów
5. …