

Sprawozdanie z projektu: TaskFlow

Nazwa projektu: TaskFlow

Autor: Pawel Duda

Cel projektu:

Celem aplikacji jest umożliwienie tworzenia zadań, zarządzania nimi w ramach projektów, filtrowania ich oraz oznaczania jako ukończone.

Technologie:

- Python
- Pytest (testowanie)
- UML (diagramy)
- brak bazy danych - dane w pamięci

Struktura klas:

- Task (tytuł, opis, deadline, status)
- Project (nazwa, lista zadań)

Zrealizowane funkcjonalności:

- Dodawanie zadania
- Usuwanie zadania
- Oznaczanie jako ukończone
- Filtrowanie

- Przypisanie zadania do projektu

Diagramy UML dołączone w folderze 'uml' Testy

jednostkowe znajdują się w folderze 'tests'

<https://www.plantuml.com/plantuml/uml/SyfFKj2rKt3>

[CoKnELR1Io4ZDoSa700001](#) - LINK DO PUML

Instrukcja testów:

```
python -m venv venv
```

```
.\venv\Scripts\activate
```

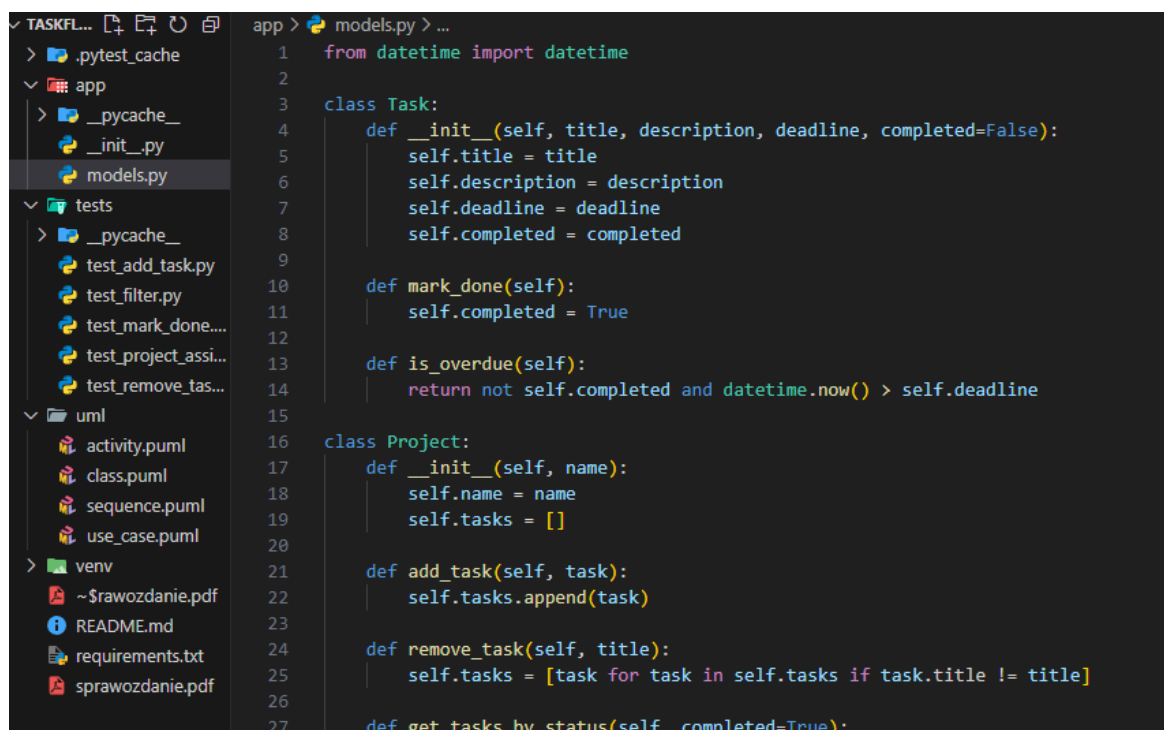
```
pip install -r requirements.txt
```

```
python -m pytest tests/
```

```
Projekt wykonany samodzielnie
>> nv) PS C:\Users\Paweł\Downloads\taskflow_final>
=====
platform win32 -- Python 3.11.9, pytest-8.4.0, pluggy-1.6.0
rootdir: C:\Users\Paweł\Downloads\taskflow_final
collected 5 items

tests\test_add_task.py .
tests\test_filter.py .
tests\test_mark_done.py .
tests\test_project_assign.py .
tests\test_remove_task.py .

=====
(venv) PS C:\Users\Paweł\Downloads\taskflow_final>
```



```
app > models.py > ...
1  from datetime import datetime
2
3  class Task:
4      def __init__(self, title, description, deadline, completed=False):
5          self.title = title
6          self.description = description
7          self.deadline = deadline
8          self.completed = completed
9
10     def mark_done(self):
11         self.completed = True
12
13     def is_overdue(self):
14         return not self.completed and datetime.now() > self.deadline
15
16     class Project:
17         def __init__(self, name):
18             self.name = name
19             self.tasks = []
20
21         def add_task(self, task):
22             self.tasks.append(task)
23
24         def remove_task(self, title):
25             self.tasks = [task for task in self.tasks if task.title != title]
26
27         def get_tasks_by_status(self, completed=True):
```