

## Lista 2

Link do repozytorium `cw_praca_z_kodem` oraz rozwiązania opisowe zadań prześlij na Platformie Moodle.

1. Pobierz pliki: **app.py**, **index.html**, **hello.html**, **requirements.txt** z Moodla (zakładka Lista2). Przekopiuj zawartość plików do swojego repozytorium utworzonego na ostatnich zajęciach (`cw_praca_z_kodem`). Jeśli nie masz zainstalowanego Pythona – zainstaluj (w wersji nie niższej niż 3.8). Utwórz wirtualne środowisko dla Pythona zgodnie z dokumentacją <https://docs.python.org/3/tutorial/venv.html> i aktywuj je (tak jak mówi dokumentacja)
2. Utwórz folder **templates** i umieść tam pliki `index.html` oraz `login.html`. Struktura projektu powinna wyglądać następująco:

`praca_z_kodem` (folder główny projektu)

|-- templates (folder)

    |-- index.html

    |-- hello.html

|-- app.py

|-- requirements.txt

|-- README.md

Zmiany wrzuć do repozytorium Github.

3. Utwórz plik **Makefile** w folderze głównym projektu. Dodaj do pliku dwie komendy:
  - a. Instalacja bibliotek: `pip install -r requirements.txt`
  - b. Uruchomienie aplikacji flask: `flask run` (jeśli nie zadziała spróbuj `python -m flask run`)

Zmiany wrzuć do repozytorium Github.

4. Uruchom komendy z pkt 3 za pomocą Makefile. Komendy: `pip...` (pkt 3a) żeby zainstalować biblioteki, kolejno uruchom komendę `flask run` (pkt 3b), aby uruchomić aplikację. Sprawdź, czy wszystko działa.
5. Po uruchomieniu aplikacji, wykonaj następujące polecenia:
  - a. Wejdź na adres <http://127.0.0.1:5000/> i wykonaj w terminalu polecenie:  
`curl http://127.0.0.1:5000/`
  - b. Wejdź na adres <http://127.0.0.1:5000/hello> i wykonaj w terminalu polecenie:  
`curl http://127.0.0.1:5000/hello`
  - c. Wejdź na adres `http://127.0.0.1:5000/hello/<name>` i wykonaj w terminalu polecenie:  
`curl http://127.0.0.1:5000/hello/<name>`

\*w miejsce `<name>` wpisz swoje lub przykładowe imię

Odpowiedzi jakie dostaniesz po wykonaniu komend wklej do pliku PDF jako odpowiedź na ćwiczenie.

6. W pliku app.py znajdź funkcję hello, zamień parametr **"name"**, który przyjmuje metoda render\_template na swoje imię (uwaga to musi być string). Uruchom aplikację, żeby zobaczyć co się zmieniło i wykonaj polecenie:

- a. curl <http://127.0.0.1:5000/hello>

Odpowiedź jaką dostaniesz po wykonaniu komendy wklej na Moodle jako odpowiedź na ćwiczenie. Napisz też jaka jest różnica w porównaniu do poprzedniego wykonania polecenia curl (co się zmieniło w odpowiedzi curl jak dokonałeś/dokonałaś zmianę).

7. Zainstaluj narzędzie pylint (`pip install pylint`).
8. Dodaj w pliku Makefile komendę sprawdzającą kod: `pylint` [plik python] - np. **pylint app.py**. Pamiętaj, aby wszystkie zmiany wrzucać do repozytorium Github.
9. Wykonaj za pomocą Makefile polecenie z pkt 8, a odpowiedź wklej na Moodle.
10. Uzupełnij plik README.md instrukcją jak poprawnie zainstalować zależności i uruchomić aplikację. Zmiany wrzuć do repozytorium Github. Dodatkowo napisz odpowiedź, dlaczego ważne jest README w projekcie.

Wszystkie odpowiedzi powinny znaleźć się na platformie Moodle.