1. Stworzenie repozytorium:

- Utwórz nowe repozytorium na GitHubie i sklonuj je na swój komputer.
- Dodaj prostą aplikację Flask do repozytorium (np. app.py), która wyświetla "Hello, World!" na stronie głównej.

```
licrosoft Windows [Version 10.0.22000.2538]
[c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

l:\Users\klape\OneDrive\Pulpit\cw_zaliczenie>git clone https://github.com/AdrjanZ/repo2.git
lloning into 'repo2'...
uarning: You appear to have cloned an empty repository.

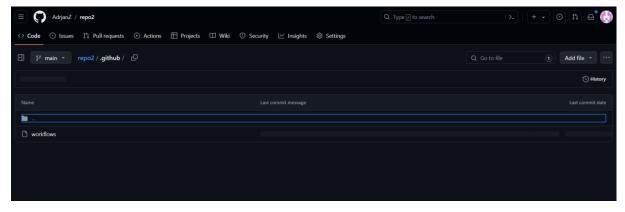
l:\Users\klape\OneDrive\Pulpit\cw_zaliczenie>_
```

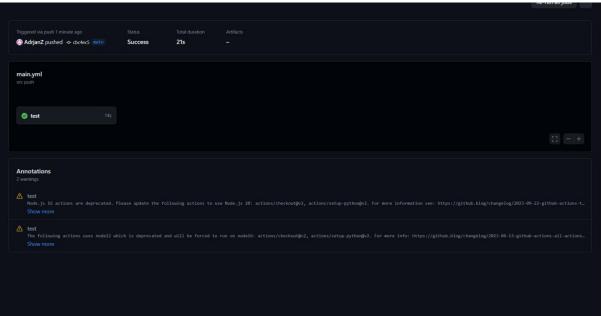
```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\klape\OneDrive\Pulpit\cw_zaliczenie\repo2>python3 main.py
* Serving Flask app 'main'
* Debug mode: on
MARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000

Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 380-218-298
```

```
C:\Users\klape\OneDrive\Pulpit\cw_zaliczenie\repo2>git push -u origin main Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 329.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AdrjanZ/repo2.git
* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
```



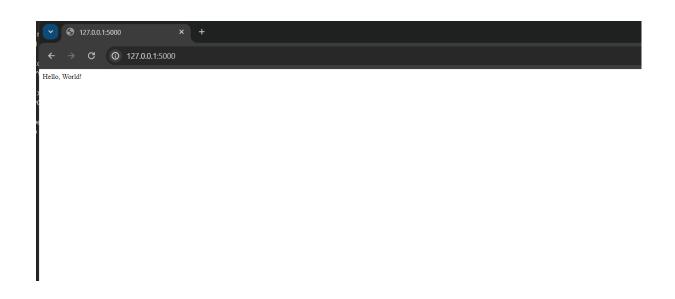


```
L:\Windows\System32\cmd.exe - docker run -p 5000:5000 tiask-app
                                                                                                                                                0 05
 => [1/5] FROM docker.io/library/python:3.10-slim@sha256:31b1e4b1c4a1903649c95fa5e46c1066baf6acdd33167813af9fa790
=> [internal] load build context
                                                                                                                                                0.1s
 => > transferring context: 2.78kB

=> CACHED [2/5] WORKDIR /app

=> CACHED [3/5] COPY requirements.txt requirements.txt

=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt
 >> => writing image sha256:1e0bd049f0d2fa99ba7a56a7046cdd266e947177ea92bb6795aa8496cdf61759
                                                                                                                                                0.0s
iew build details: docker-desktop://dashboard/build/default/default/2q1n5l9bewk7xwst804g9d1y8/
Mat's Next?
 View a summary of image vulnerabilities and recommendations \rightarrow docker scout quickview
 :\Users\klape\OneDrive\Pulpit\cw_zaliczenie\repo2>docker run -p 5000:5000 flask-app
  Serving Flask app 'main' (lazy loading)
  Environment: production
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  Debug mode: on
  Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
  Restarting with stat
  Debugger is active!
Debugger PIN: 751-942-377
```



C:\Users\klape\OneDrive\Pulpit\cw_zaliczenie\repo2\tests>python3 -m pytest	
test_app.py .	[100%]