Дашборд для отдела

# Идея

1. Работа.

Сама работа происходит в некоторой среде разработки (jupyter notebook например).

1. Создание фронтенда и бэкенда для дашборда.

По окончанию работы:

* Весь нужный код структурируются, чтобы можно было создать один связный и визуально готовый статический или динамический фронтенд.
* Все исходные данные / запросы в БД переносятся в текстовые файлы, а рабочий py файл модифицируется, чтобы считывать данные из текстовых файлов.

1. Из этого создаётся гитхаб проект либо докер образ, готовый к запуску на сервере.

И потом это развертывается на удаленном сервере, чтобы другие люди могли по ссылке в браузере получить дашборд.

# Сервисы для деплоя проектов

1. Render

<https://render.com/>

Позволяет бесплатно деплоить статические сайты и веб приложения

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Streamlit

* Если достаточно просто показать статические графики из пайтона, то это идеальное решение.

Однако, сделать чтобы данные такого дашборда обновлялись и были актуальные тяжело и лучше не надо. В таком случае лучше со своим бэкендом.

Есть возможность добавить свои react компоненты.

* При таком подходе именно py файл определяет в каком порядке и будут расположен текст и графики.

1. Если работа происходит в JN, то перенести весь код в отдельный py файл.

Вместо print написать st.write() итд

1. сделай файл requirements.txt
2. Создать гитхаб репозиторий с py файлом и requirements.txt. И запустить на <https://share.streamlit.io/>

## Добавить react компонент

1. Из шаблона

<https://docs.streamlit.io/develop/concepts/custom-components/intro#create-a-bi-directional-component>

1. Самостоятельно
2. Сделать предыдущие шаги
3. Создать свой реакт компонент в папке frontend

npm create vite@latest my-react-app -- --template react

1. для использования реакт компонента в *py* файле:

import os

import streamlit as st

import streamlit.components.v1 as components

\_RELEASE = False

if not \_RELEASE:

\_component\_func = components.declare\_component(

"my\_component",

url="http://localhost:3001",

)

else:

build\_dir = os.path.join(os.path.dirname(\_\_file\_\_), "frontend/build")

\_component\_func = components.declare\_component("my\_component", path=build\_dir)

def my\_component(name, key=None):

return \_component\_func(name=name, key=key, default=0)

#\_\_\_\_\_\_Использование компонента\_\_\_\_\_\_\_\_\_

value = my\_component("Streamlit")

st.write(f"Component returned: {value}")

# Статический сайт

# Веб сервис с бэкендом