Oczywiście! Oto pełne podsumowanie kroków, które pozwoliły Ci zbudować działającą, samodzielną aplikację .exe ze Streamlit, bez potrzeby instalowania Pythona, Streamlit czy posiadania pliku .py na komputerze użytkownika.

⸻

✅ CEL:

🔹 Zbudować pojedynczy plik .exe, który:

• uruchamia aplikację Streamlit

• nie wymaga zewnętrznego Pythona ani streamlit zainstalowanego na komputerze

• nie wymaga obecności oryginalnego .py ani POC.xlsx obok .exe

• działa nawet na komputerze bez Pythona

⸻

✅ JAK TO ZOSTAŁO OSIĄGNIĘTE

1. Spakowanie kodu Streamlit do zmiennej w Pythonie

• Twój oryginalny kod nowy skrypt wersja 2.py został zapisany jako łańcuch tekstowy (app\_code) w innym pliku (run\_autonomous.py).

• Dzięki temu plik .py z kodem nie musi być obecny obok .exe.

2. Tymczasowe uruchomienie aplikacji z pamięci

• Plik run\_autonomous.py:

• zapisuje app\_code do tymczasowego pliku .py (np. app\_tmp.py)

• uruchamia streamlit run app\_tmp.py

• Wszystko odbywa się wewnątrz .exe bez zależności.

3. Dołączenie pliku Excel (POC.xlsx) do aplikacji

• Użyto opcji “Add File” w auto-py-to-exe, aby dołączyć POC.xlsx do .exe.

• Ustawiono destination na . – więc plik trafił do bieżącego katalogu pracy .exe.

4. Zbudowanie .exe z pomocą PyInstaller przez auto-py-to-exe

• W auto-py-to-exe ustawiono:

• ✅ Onefile

• ✅ Console (dla debugowania)

• ➕ Additional File: POC.xlsx

• ➕ Hidden import: streamlit

5. Poprawki techniczne

• Zmieniono sposób zapisu pliku tymczasowego, aby działał bez uprawnień administratora (np. zapis do lokalnego folderu, nie Temp).

• Upewniono się, że .exe nie wymaga requireAdministrator ani nie korzysta z lokalizacji systemowych.

⸻

🔄 Finalny efekt:

• ✅ Jeden plik run\_autonomous.exe

• ✅ Nie potrzeba Pythona ani streamlit

• ✅ Nie trzeba mieć .py ani POC.xlsx

• ✅ Działa nawet na nowym koncie użytkownika

• ✅ Działa bez instalacji czegokolwiek

⸻

🛠 Co warto zachować na przyszłość?

1. Twój plik run\_autonomous.py jako template

2. Auto-py-to-exe z odpowiednimi ustawieniami

3. Jeśli coś nie działa – uruchamiaj .exe z konsoli (& .\app.exe) by zobaczyć błędy

⸻

Jeśli chcesz, mogę Ci przygotować gotowy szablon projektu (.zip) do takich aplikacji na przyszłość – chcesz?  Świetnie, oto kompletne podsumowanie procesu budowy samodzielnego .exe ze Streamlit — krok po kroku:

⸻

✅ Cel:

Stworzyć pojedynczy plik .exe, który:

• uruchamia aplikację Streamlit

• nie wymaga Pythona

• nie potrzebuje osobnego pliku .py ani biblioteki Streamlit

⸻

🧱 1. Pliki potrzebne do zbudowania .exe

W katalogu projektu użyliśmy 2 plików:

Plik Rola

run\_autonomous.py Główny plik uruchamiający aplikację (wszystko w jednym)

POC.xlsx Plik z danymi używany przez aplikację

Nie potrzeba już osobnego pliku .py z aplikacją — cały kod jest wbudowany w run\_autonomous.py.

⸻

⚙️ 2. Konfiguracja w auto-py-to-exe

Otworzyliśmy auto-py-to-exe i ustawiliśmy:

Ustawienie Wartość

Script location run\_autonomous.py

Onefile ✅ (zaznaczone)

Console ✅ (zaznaczone na razie — wersja bez konsoli też możliwa)

Add Data POC.xlsx → destination .

Hidden Imports streamlit

Dzięki temu PyInstaller:

• dodał ukrytą bibliotekę Streamlit

• spakował plik danych Excel

• zbudował jeden .exe

⸻

📁 3. Struktura pliku run\_autonomous.py

To specjalny launcher, który wbudowuje Twoją aplikację Streamlit jako string i uruchamia ją lokalnie.

📦 Co zawiera run\_autonomous.py?

import subprocess

import os

# Twój kod Streamlit jako string

app\_code = """

import streamlit as st

import pandas as pd

st.title("Twoja aplikacja")

df = pd.read\_excel("POC.xlsx")

st.dataframe(df)

"""

# Zapisz kod jako tymczasowy plik

app\_path = os.path.join(os.getcwd(), "app\_tmp.py")

with open(app\_path, "w", encoding="utf-8") as f:

f.write(app\_code)

# Uruchom streamlit run app\_tmp.py

subprocess.run(["streamlit", "run", app\_path])

⸻

🔍 Co robi?

1. Tworzy plik app\_tmp.py zawierający kod Twojej aplikacji

2. Wykonuje streamlit run app\_tmp.py — czyli uruchamia lokalny webserver

3. Wszystko dzieje się automatycznie — użytkownik nie widzi żadnego kodu

⸻

🧪 Efekt końcowy

• Plik run\_autonomous.exe działa samodzielnie

• Nie wymaga:

• Pythona

• streamlit

• .py z kodem aplikacji

• Aplikacja otwiera się w przeglądarce jak klasyczny Streamlit

⸻

✅ BONUS: Co możesz teraz?

• Dodać ikonę .ico w auto-py-to-exe

• Przełączyć się na wersję bez konsoli (okienkową)

• Spakować .exe + POC.xlsx do ZIP i udostępniać jako gotowy program

⸻

Chcesz, żebym Ci też przygotował wersję bez konsoli lub z ikoną?   Świetne pytanie — i bardzo trafna obserwacja.

⸻

🔍 Gdzie „zniknął” streamlit.exe, który musiałeś wcześniej wyciągać?

Już go nie potrzebujesz osobno, bo:

✅ PyInstaller spakował go automatycznie do .exe, dzięki opcji:

--hidden-import streamlit

⸻

🔧 Co się działo wcześniej?

🔴 Wcześniej, kiedy próbowałeś uruchomić Streamlit z poziomu .exe, bez tej opcji:

• streamlit nie był dołączony

• Twój launcher (run.py) robił coś takiego:

subprocess.run(["streamlit", "run", "plik.py"])

• Ale .exe nie wiedział, co to streamlit, bo nie było go w środku

Więc próbowałeś ręcznie podmieniać lub wskazywać ścieżki do streamlit.exe, kopiując go z:

C:\Users\<nazwa>\AppData\Local\Programs\Python\Python3XX\Scripts\streamlit.exe

⸻

✅ A teraz?

W run\_autonomous.py, kod uruchamia:

subprocess.run(["streamlit", "run", app\_path])

I działa, ponieważ:

• streamlit jest spakowany w środku

• pyinstaller dzięki --hidden-import streamlit dołącza wszystkie potrzebne biblioteki i CLI

• exe potrafi odpalić streamlit bez żadnego Pythona zainstalowanego

⸻

📦 Gdzie jest ten streamlit teraz?

Ukryty głęboko w .exe – dokładnie:

run\_autonomous.exe → zawiera archiwum .pkg → zawiera kod streamlit + jego zależności

Nie musisz się tym przejmować — PyInstaller stworzył własną wersję środowiska w środku .exe.

⸻

🧼 Co możesz zrobić teraz?

Możesz:

• usunąć lokalnego streamlit.exe z Scripts — już nie jest potrzebny

• przekopiować .exe na inny komputer bez Pythona i zobaczyć, że działa

⸻

Chcesz, żebym pokazał Ci jak zobaczyć, co dokładnie jest spakowane w .exe?  🟩 Nie, pliku streamlit.exe nie musisz dołączać ręcznie do auto-py-to-exe – ani przez Add File, ani Add Data.

⸻

✅ Dlaczego?

streamlit.exe to tylko skrócona forma wywołania całego pakietu Streamlit (CLI).

To, co naprawdę ważne:

• to nie ten .exe z Scripts\, tylko

• cały pakiet streamlit w katalogu site-packages

PyInstaller i auto-py-to-exe automatycznie znajdą i spakują wszystko, o ile:

⸻

✅ Masz dodane --hidden-import streamlit

To jest jedyne co musisz dodać ręcznie, aby PyInstaller wiedział, że streamlit ma być częścią .exe.

W auto-py-to-exe:

• zakładka Advanced

• pole Hidden imports

• wpisz streamlit

⸻

❌ Czego NIE robimy:

Niepotrzebne czynności Dlaczego?

❌ Nie dodajesz streamlit.exe w Add Files Ten plik sam nic nie znaczy, to tylko “skrót do CLI”

❌ Nie kopiujesz ręcznie folderu streamlit PyInstaller weźmie go sam, jeśli --hidden-import jest ustawione

❌ Nie zmieniasz ścieżek Wystarczy, że masz zainstalowany streamlit w swoim Pythonie

⸻

📦 PyInstaller znajdzie to tutaj automatycznie:

C:\Users\<ty>\AppData\Local\Programs\Python\Python311\Lib\site-packages\streamlit

⸻

✨ Podsumowanie:

❗W auto-py-to-exe, jedyne co musisz zrobić to:

w Hidden imports wpisać: streamlit

✅ Żadnych ręcznych dodawań pliku streamlit.exe

✅ Nie dodajesz folderów ani wpisów w Add Files

⸻

Chcesz przykład komendy pyinstaller jaką auto-py-to-exe tworzy w tle?