**LAB EXPERIMENT # 09**

|  |
| --- |
| **Practical Data Anonymization Using Python and Streamlit** |

|  |  |
| --- | --- |
| **Student Name:** | **Roll No:** |

**Objective:**

This lab helps students:

* To understand the importance and methods of anonymizing sensitive data in big data environments.
* To develop a functional data anonymization tool using Python libraries such as hashlib, pandas, and faker.
* To gain hands-on experience in deploying a data privacy solution using a user-friendly interface built with Streamlit.

### **Introduction:**

In the era of big data, protecting sensitive user information is critical to maintaining privacy and complying with data protection regulations such as GDPR and HIPAA. This lab introduces students to data anonymization techniques by implementing a web-based tool using Python and Streamlit. The tool hashes identifiers like customer IDs and phone numbers and replaces text-based fields with synthetic data using the Faker library, simulating real-world practices in securing personal data before analysis or sharing.

**Procedure:**

**Requirements:**

Ensure the following Python libraries are installed:

**pip install streamlit pandas faker**

**Sample Code: Data Anonymization Tool using Python and Streamlit**

import streamlit as st

import pandas as pd

import hashlib

from faker import Faker

# Initialize Faker

faker = Faker()

# Function to anonymize data

def anonymize\_data(df):

df\_anonymized = df.copy()

for col in df.columns:

col\_lower = col.strip().lower()

# Hash customer id or phone number if found

if 'customer id' in col\_lower or 'phone number' in col\_lower:

df\_anonymized[col] = df[col].apply(lambda x: hashlib.sha256(str(x).encode()).hexdigest())

# Anonymize other string columns

elif df[col].dtype == object:

df\_anonymized[col] = df[col].apply(lambda x: faker.text(max\_nb\_chars=12))

return df\_anonymized

# Streamlit UI

st.set\_page\_config(page\_title="🔐 Data Anonymizer", layout="centered")

st.title("🔐 Data Anonymization Tool")

uploaded\_file = st.file\_uploader("📁 Upload a CSV file", type=["csv"])

if uploaded\_file is not None:

df = pd.read\_csv(uploaded\_file)

st.subheader("📋 Original Data")

st.dataframe(df.head())

if st.button("🚀 Anonymize Data"):

df\_anonymized = anonymize\_data(df)

st.subheader("🛡 Anonymized Data")

st.dataframe(df\_anonymized.head())

csv = df\_anonymized.to\_csv(index=False).encode("utf-8")

st.download\_button(

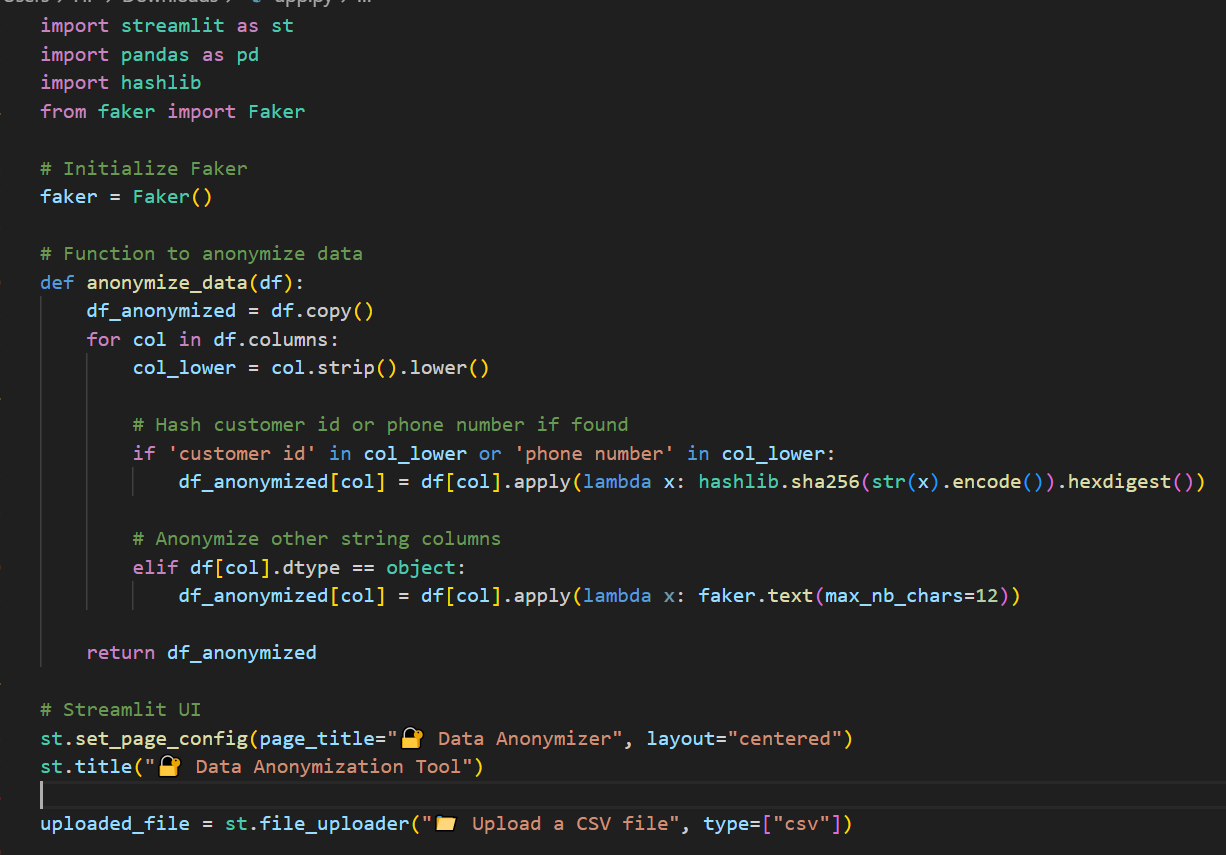
label="⬇ Download Anonymized CSV",

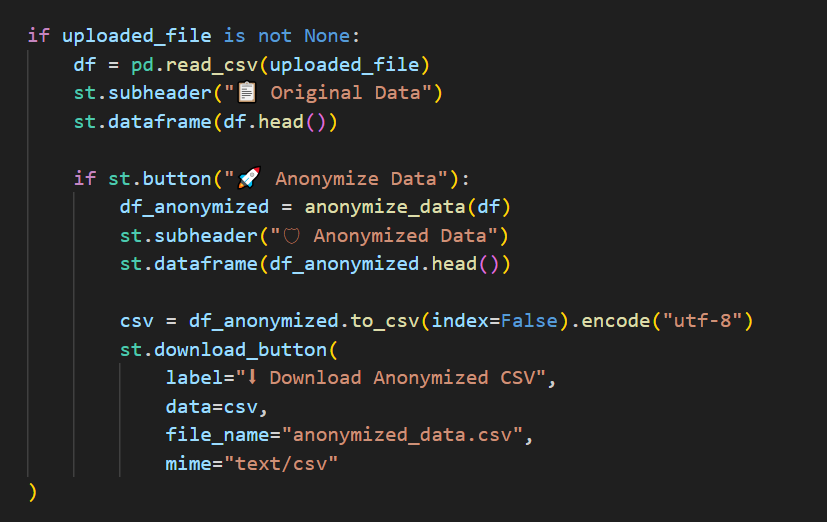
data=csv,

file\_name="anonymized\_data.csv",

mime="text/csv"

)





**Steps to Run:**

* Save the above code as app.py.
* Place a sample CSV file (e.g., with columns like Name, Customer ID, Phone Number).
* Run the Streamlit app:

**python -m streamlit run app.py**

* Upload the CSV file using the interface.
* Click “Anonymize Data” to view and download the anonymized version.

**Output:**

