3📥 1. Original Data (sample_data.csv)

Name,Email,Phone

Alice,alice@example.com,1234567890

Bob,bob@example.com,9876543210

📄 2. Console Output (when you run main.py)

Original Data:

	Name	Email	Phone
0	Alice	alice@example.com	1234567890
1	Bob	bob@example.com	9876543210

✅ Data encrypted and saved.

🔐 3. Encrypted Data (encrypted/encrypted_data.txt)

Example content (your output will be different due to randomness in encryption):

gAAAAABlqRSbaYlF1v2N4E5fGlfYdMMAX6BtF...

gAAAAABlqRSbrzswA35_WoMKjwLgEiNqJ2bJ...

Each line is an encrypted row of the CSV data using AES encryption via Fernet.

🔓 4. Optional Decryption Output

If you run the decrypt code:

Decrypted row: Alice,alice@example.com,1234567890

Decrypted row: Bob,bob@example.com,9876543210

📦 What You Get

Step	Output Type	Purpose
main.py	Terminal + Encrypted file	Reads, encrypts, and stores data
encrypted_data.txt	Binary Encrypted Text	Safe to store / transmit
decrypt_data()	Plaintext output (optional)	Shows decrypted data for testing

---

Let me know if you'd like:

✅ Login + password protection

✅ AES-256 instead of Fernet (more manual control)

✅ Web-based form to upload and encrypt

✅ Secure storage in a cloud or database

I'm happy to extend this project as per your need.