Assignment – 03

To understand the fundamentals of Solidity smart contract development by implementing a simple Ethereum smart contract that accepts a message from the user, stores it on the blockchain, and retrieves it securely.

Problem Statement

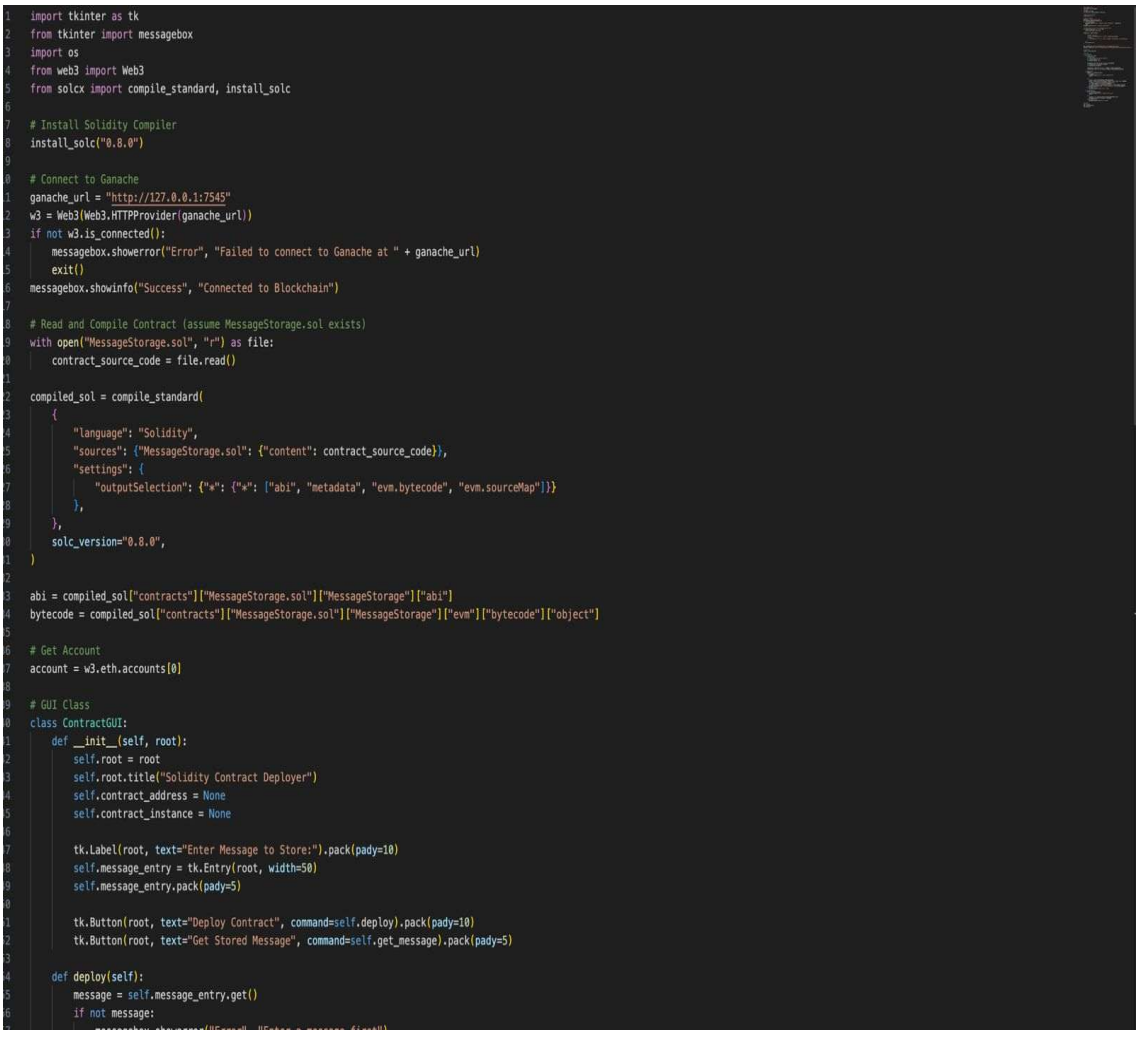
Develop a basic Solidity smart contract that allows users to:

• Store a message on the blockchain

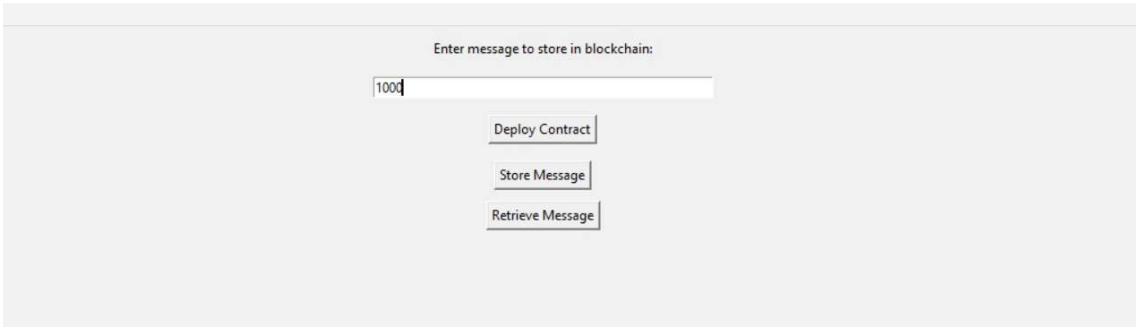
• Update the message

• Retrieve the stored message

This practical helps understand state variables, functions, constructors, and data types in Solidity.



```python
import tkinter as tk
from tkinter import messagebox
import os
from web3 import Web3
from solcx import compile_standard, install_solc

# Install Solidity Compiler
install_solc("0.8.0")

# Connect to Ganache
ganache_url = "http://127.0.0.1:7545"
w3 = Web3(Web3.HTTPProvider(ganache_url))
if not w3.is_connected():
    messagebox.showerror("Error", "Failed to connect to Ganache at " + ganache_url)
    exit()
messagebox.showinfo("Success", "Connected to Blockchain")

# Read and Compile Contract (assume MessageStorage.sol exists)
with open("MessageStorage.sol", "r") as file:
    contract_source_code = file.read()

compiled_sol = compile_standard(
    {
        "language": "Solidity",
        "sources": {"MessageStorage.sol": {"content": contract_source_code}},
        "settings": {
            "outputSelection": {"*": {"*": ["abi", "metadata", "evm.bytecode", "evm.sourceMap"]}}
        },
    },
    solc_version="0.8.0",
)

abi = compiled_sol["contracts"]["MessageStorage.sol"]["MessageStorage"]["abi"]
bytecode = compiled_sol["contracts"]["MessageStorage.sol"]["MessageStorage"]["evm"]["bytecode"]["object"]

# Get Account
account = w3.eth.accounts[0]

# GUI Class
class ContractGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Solidity Contract Deployer")
        self.contract_address = None
        self.contract_instance = None

        tk.Label(root, text="Enter Message to Store:").pack(pady=10)
        self.message_entry = tk.Entry(root, width=50)
        self.message_entry.pack(pady=5)

        tk.Button(root, text="Deploy Contract", command=self.deploy).pack(pady=10)
        tk.Button(root, text="Get Stored Message", command=self.get_message).pack(pady=5)

    def deploy(self):
        message = self.message_entry.get()
        if not message:
```


Enter message to store in blockchain:
1000
Deploy Contract
Store Message
Retrieve Message