ASSIGNMENT-2

NAME-ARSHA VARDHINI

ROLL_NO:2303A51600

BATCH-29

QUESTION:

A smart contract is a self-executing program stored on the blockchain.

The Simple Storage contract is a beginner-level Solidity contract that allows users to:

• Store a value on the blockchain

• Retrieve the stored value

-interface should contain:

1.money send

2.previous Hash

3.Current Hash

4.Transaction output

CODE:

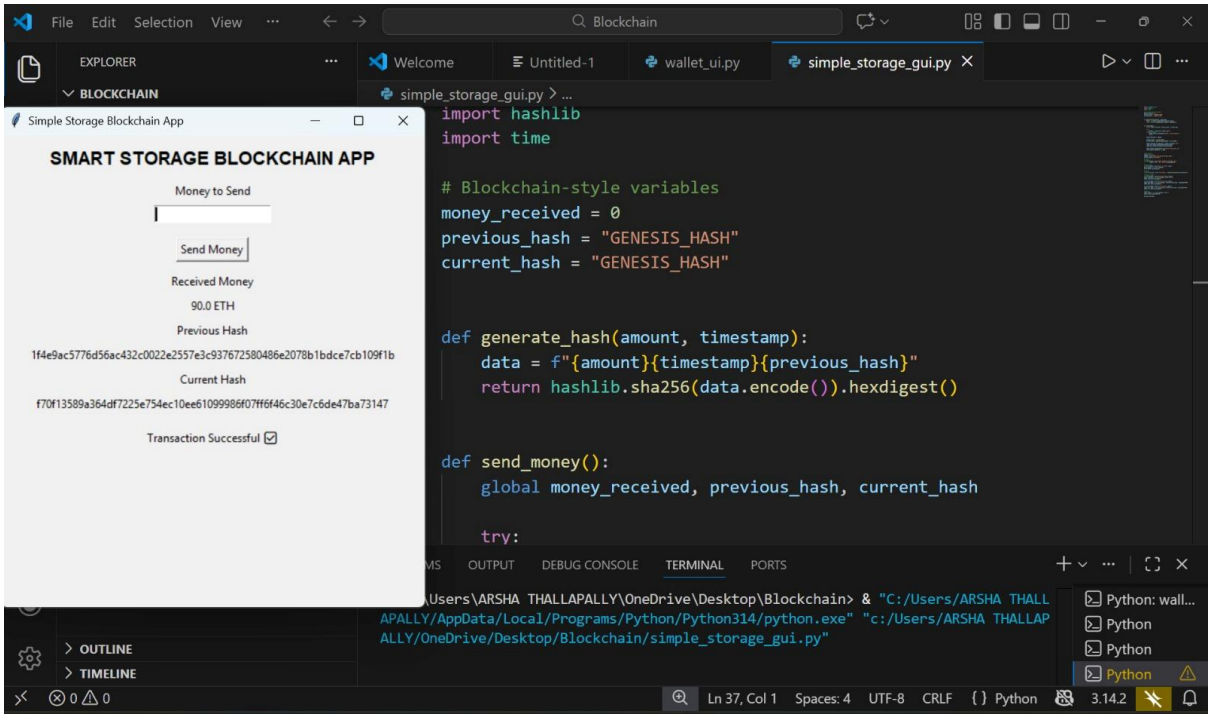
```
import tkinter as tk
import hashlib
import time
```

```
# Blockchain-style variables
money_received = 0
previous_hash = "GENESIS_HASH"
current_hash = "GENESIS_HASH"

def generate_hash(amount, timestamp):
    data = f"{amount}{timestamp}{previous_hash}"
    return hashlib.sha256(data.encode()).hexdigest()

def send_money():
    global money_received, previous_hash, current_hash

    try:
        amount = float(entry_amount.get())
    except ValueError:
        label_status.config(text="Enter a valid number!")
        return

    money_received += amount

    previous_hash = current_hash
    current_hash = generate_hash(amount, time.time())
```

```
    label_received.config(text=f"{money_received} ETH")
    label_prev_hash.config(text=previous_hash)
    label_curr_hash.config(text=current_hash)

    label_status.config(text="Transaction Successful ✅ ")
    entry_amount.delete(0, tk.END)

# GUI Window
window = tk.Tk()
window.title("Simple Storage Blockchain App")
window.geometry("450x500")

# Heading
tk.Label(window, text="SMART STORAGE BLOCKCHAIN APP",
         font=("Arial", 14, "bold")).pack(pady=10)

# Money input
tk.Label(window, text="Money to Send").pack()
entry_amount = tk.Entry(window)
entry_amount.pack(pady=5)

# Button
```

```
tk.Button(window, text="Send Money",
command=send_money).pack(pady=10)

# Display fields

tk.Label(window, text="Received Money").pack()

label_received = tk.Label(window, text="0 ETH")

label_received.pack(pady=5)

tk.Label(window, text="Previous Hash").pack()

label_prev_hash = tk.Label(window, text=previous_hash,
wraplength=400)

label_prev_hash.pack(pady=5)

tk.Label(window, text="Current Hash").pack()

label_curr_hash = tk.Label(window, text=current_hash,
wraplength=400)

label_curr_hash.pack(pady=5)

# Status

label_status = tk.Label(window, text="")

label_status.pack(pady=10)

window.mainloop()
```

OUTPUT:


Simple Storage Blockchain App
SMART STORAGE BLOCKCHAIN APP
Money to Send
Send Money
Received Money
90.0 ETH
Previous Hash
1f4e9ac5776d56ac432c0022e2557e3c937672580486e2078b1bdce7cb109f1b
Current Hash
f70f13589a364df7225e754ec10ee61099986f07ff6f46c30e7c6de47ba73147
Transaction Successful
import hashlib
import time
# Blockchain-style variables
money_received = 0
previous_hash = "GENESIS_HASH"
current_hash = "GENESIS_HASH"
def generate_hash(amount, timestamp):
data = f"{amount}{timestamp}{previous_hash}"
return hashlib.sha256(data.encode()).hexdigest()
def send_money():
global money_received, previous_hash, current_hash
try:
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Users\ARSHA THALLAPALLY\OneDrive\Desktop\Blockchain> & "C:/Users/ARSHA THALLAPALLY/AppData/Local/Programs/Python/Python314/python.exe" "c:/Users/ARSHA THALLAPALLY/OneDrive/Desktop/Blockchain/simple_storage_gui.py"
Python: wall...
Python
Python
Python
OUTLINE
TIMELINE
Ln 37, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.14.2