

# **BLOCKCHAIN ENGINEERING PRACTICAL-1.**

**NAME: AKSHITHA**

**ROLL NO:2303A51360**

**BATCH NO:29**

## **QUESTION 1:**

Objective:

To learn blockchain interaction by creating a cryptocurrency wallet, checking wallet balance, and simulating transactions using Python and Web3

Requirements:

- Install Python 3.x
- Set up VS Code with Python extension
- Install required Python libraries:
  - pip install web3
- Use a test blockchain network (Ethereum Sepolia / Ganache local blockchain)
- Basic understanding of blockchain wallets and private keys

Practical Description:

Step 1: Environment Setup

- Install Python and VS Code
- Install Web3.py library
- Create a Python file named `wallet_interaction.py`

Step 2: Wallet and Blockchain Interaction Script

Create a Python script that:

- Connects to a blockchain network
  - Loads a wallet using a private key
  - Fetches wallet address
  - Checks wallet balance
  - Demonstrates transaction preparation (without real funds)

## **CODE:**

```
# ----- Functions -----  
  
def check_balance():  
  
    if connected:  
  
        try:  
  
            wei_balance = web3.eth.get_balance(SAMPLE_ADDRESS)  
  
            balance_eth = web3.from_wei(wei_balance, 'ether')  
  
            formatted_balance = f"{balance_eth:.2f}"  
  
  
            messagebox.showinfo(  
                "Wallet Balance",  
                f"Connected to Ganache\n\n"  
                f"Wallet Address:{SAMPLE_ADDRESS}\n\n"  
                f"Balance: {formatted_balance} ETH"  
            )  
  
        except Exception as e:  
  
            messagebox.showerror("Error", str(e))  
  
    else:  
  
        # Simulation Mode  
  
        balance_eth = 10  
  
        formatted_balance = f"{balance_eth:.2f}"  
  
  
        messagebox.showinfo(
```

```
"Wallet Balance",
f"Simulation Mode (Offline)\n\n"
f"Simulated Balance: {formatted_balance} ETH"
)

def simulate_transaction():
    if connected:
        messagebox.showinfo(
            "Transaction",
            "Connected Mode\n\nTransaction feature can be added here."
        )
    else:
        messagebox.showinfo(
            "Transaction",
            "Simulation Mode\n\nTransaction simulated successfully!"
        )

# ----- GUI Setup -----
root = tk.Tk()
root.title("Blockchain Wallet Simulator")
root.geometry("350x300")
root.resizable(False, False)
```

```
# Title
title_label = tk.Label(
    root,
    text="Blockchain Wallet (Python + Web3)",
    font=("Arial", 12, "bold")
)
title_label.pack(pady=10)

# Status
status_text = "Connected to Ganache" if connected else "Simulation
Mode (Offline)"
status_color = "green" if connected else "red"

status_label = tk.Label(
    root,
    text=status_text,
    fg=status_color,
    font=("Arial", 10, "bold")
)
status_label.pack(pady=5)

# Buttons
btn_balance = tk.Button(
```

```
root,  
text="Check Wallet Balance",  
width=25,  
command=check_balance  
)  
btn_balance.pack(pady=10)
```

```
btn_transaction = tk.Button(  
root,  
text="Simulate Transaction",  
width=25,  
command=simulate_transaction  
)  
btn_transaction.pack(pady=10)
```

```
btn_exit = tk.Button(  
root,  
text="Exit",  
width=25,  
command=root.quit  
)  
btn_exit.pack(pady=10)
```

# Run app

```
root.mainloop()
```

## OUTPUT:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `Welcome`, `Blockchain`, and `.venv`.
- Terminal:** Displays the Python code for the blockchain wallet application and a message confirming web3 installation.
- Output Panel:** Shows a success message: "Great! You're all set with web3 installed. You can now start building your blockchain project. Let me know if you need help with anything else!"
- Code Editor:** The `main.py` file contains the code for the blockchain wallet application, including logic for connecting to Ganache and displaying wallet balance.
- Blockchain Wallet Simulator:** A modal window titled "Wallet Balance" shows the simulated balance as 10.00 ETH.
- Status Bar:** Shows the current line (Ln 109), column (Col 1), spaces (Spaces: 4), encoding (UTF-8), CRLF, Python version (venv 3.13.7), and other development tools.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `Welcome`, `Blockchain`, and `.venv`.
- Terminal:** Displays the Python code for the blockchain wallet application and a message confirming web3 installation.
- Output Panel:** Shows a success message: "Great! You're all set with web3 installed. You can now start building your blockchain project. Let me know if you need help with anything else!"
- Code Editor:** The `main.py` file contains the code for the blockchain wallet application, including logic for packing transactions and quitting the application.
- Blockchain Wallet Simulator:** A modal window titled "Transaction" shows the message "Transaction simulated successfully!"
- Status Bar:** Shows the current line (Ln 109), column (Col 1), spaces (Spaces: 4), encoding (UTF-8), CRLF, Python version (venv 3.13.7), and other development tools.

