School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning
(Learning by Doing and Discovery)

Name of the Experiement : Know Your TX – Dissecting a Transaction

Objective/Aim :

To analyze the internal structure and behavior of a blockchain transaction by using a transaction hash and exploring it through a blockchain explorer (Sepolia Testnet)

Apparatus/Software Used:

- MetaMask Wallet (Testnet enabled)
- Sepolia Testnet ETH
- Sepolia Etherscan
- Web browser and internet
- Ethereum test contract (for interaction)

Theory/Concept:

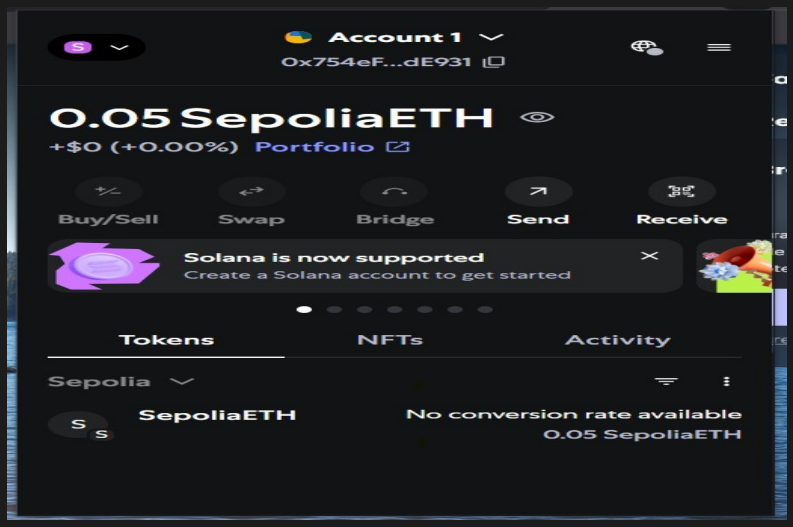
A **transaction (TX)** on the Ethereum blockchain represents an action initiated by an externally owned account (EOA). Transactions may transfer ETH or interact with smart contracts. Key parts of a transaction include:

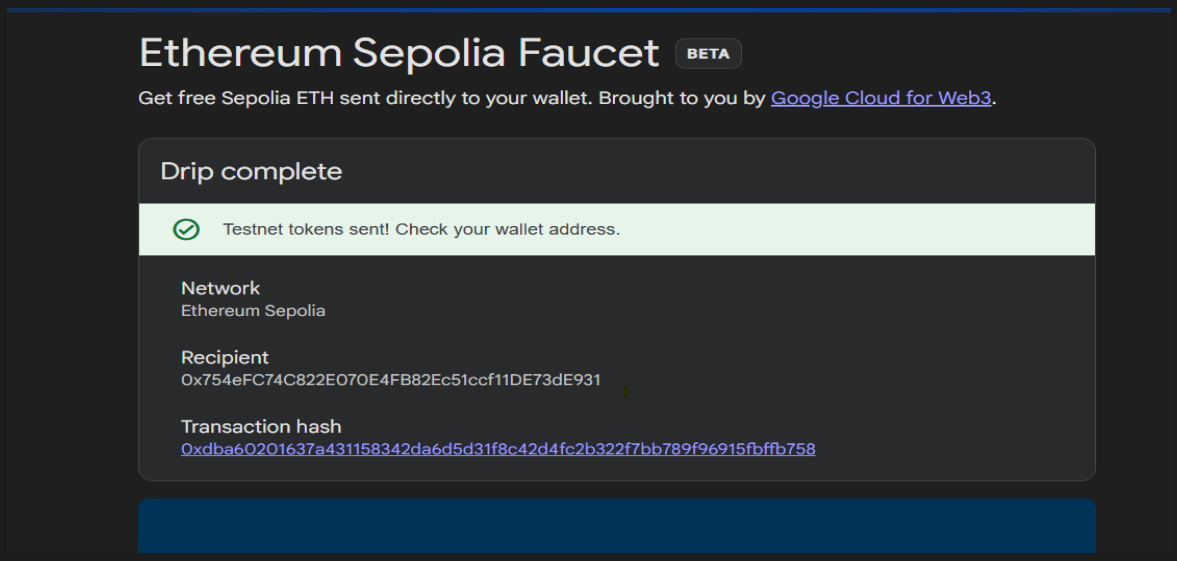
- **TX Hash** – unique ID of the transaction
- **From** – sender address
- **To** – receiver or contract address
- **Value** – amount of ETH transferred
- **Gas Fee** – amount paid to miners/validators
- **Status** – success or failure
- **Block** – block number where TX was recorded

Procedure:

1. Setup MetaMask for Sepolia Testnet:

- Open the **MetaMask** browser extension or mobile app.
- Enable **Sepolia Test Network** from the network list.
- Make sure you have some **test ETH** in your wallet





.

2. Interact with a Smart Contract:

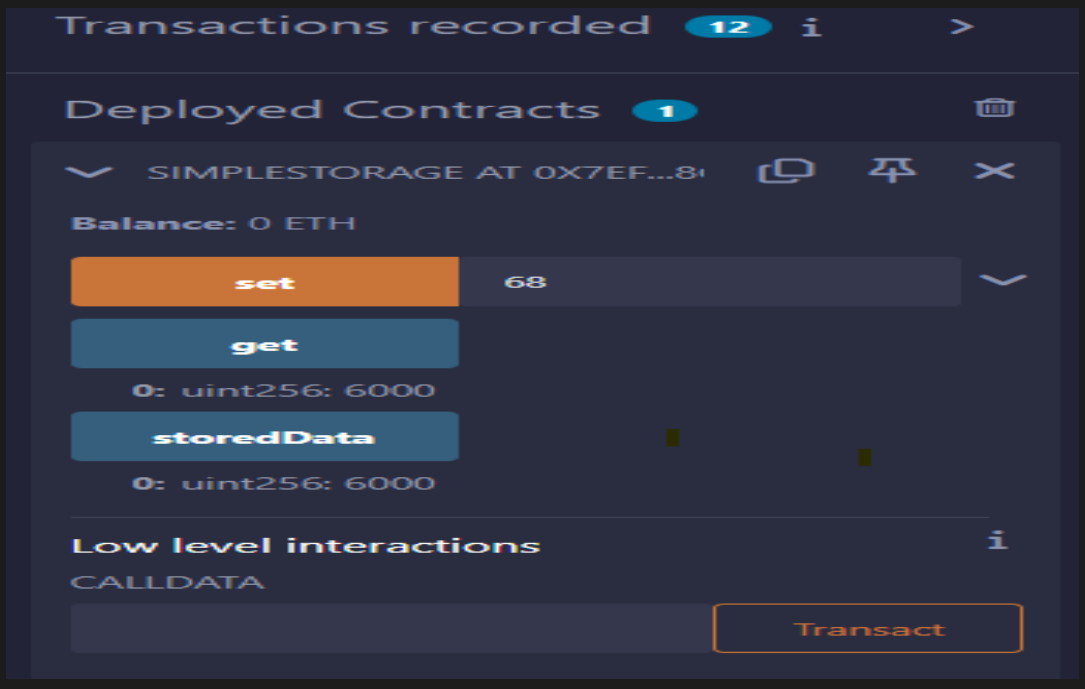
- Visit a platform like **Remix IDE** or any dApp connected to Sepolia.
- Use MetaMask to **call a function** on a deployed **smart contract** (e.g., a counter contract or any public method).
- Submit the transaction

Home | First.sol | storage.test.js | Ballot_test.sol | 1_Storage.sol

```
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.0;
3  contract SimpleStorage{
4      uint public storedData;
5
6
7      constructor(uint _data) {    infinite gas 73800 gas
8          storedData = _data;
9      }
10
11     function set(uint x) public {    22514 gas
12         storedData = x;    DUP1 costs 3 gas - this line costs 5023 gas - 9019 gas left
13     }
14
15     function get() public view returns (uint) {    2453 gas
16         return storedData;
17     }
18
19 }
20
```

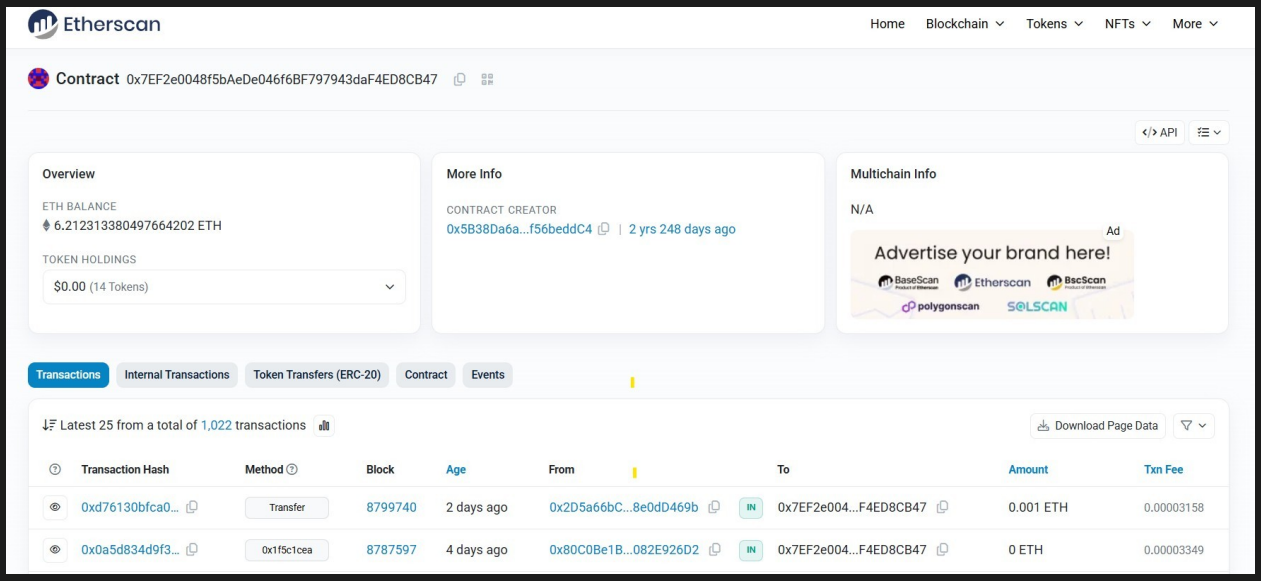
3. Copy the Transaction Hash (TX Hash):

- After sending the transaction, MetaMask will show a **TX hash** (a long alphanumeric string).
- This hash is a **unique identifier** for your transaction on the blockchain.



4. Open Sepolia Etherscan:

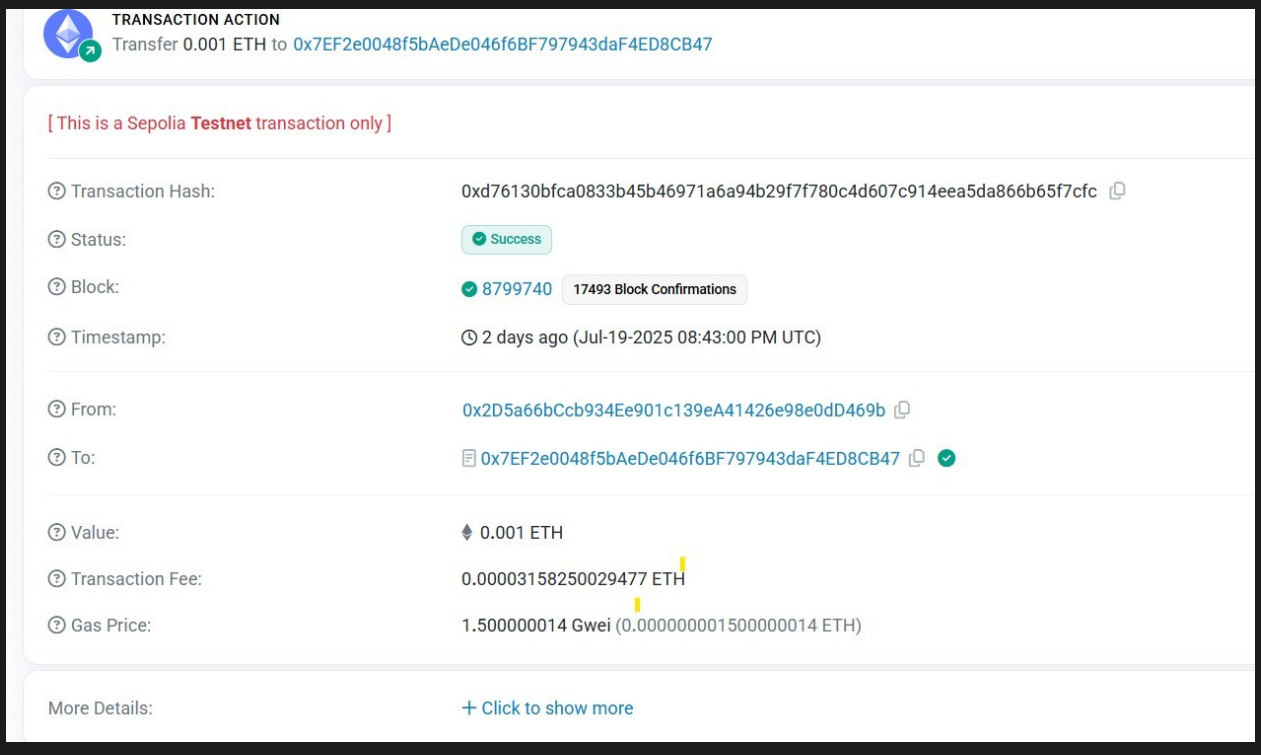
- Go to https://sepolia.etherscan.io.
- Paste the **transaction hash** in the search bar and hit **Enter**



5. Analyze the Transaction Details:

On the transaction page, observe the following:

- **Status**: Success or Failure of the transaction.
- **Block Number**: The block that included your TX.
- **Timestamp**: Exact time and date of confirmation.
- **From & To Address**: Sender (your wallet) and receiver (wallet or contract).
- **Value**: Amount of ETH transferred.
- **Transaction Fee**: Calculated as `Gas Used × Gas Price`.
- **Gas Price**: Fee per unit of gas set by the sender.
- **To (Contract)**: If the transaction is a contract call, it shows the contract address.



Observation:

A transaction was sent on the **Sepolia Testnet** and completed successfully. It was included in **block 8787597** and had over **29,000 confirmations**. The **sender address** was `0x80C0...`, and the **receiver** was a smart contract at `0x7EF2....` No ETH was sent (value = 0), but a **transaction fee of 0.00003349 ETH** was paid. The **gas price** was **1.5 Gwei**, and the transaction status was **successful**.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No. :

Signature of the Faculty:

** As applicable according to the experiment. Two sheets per experiment (10-20) to be used.*