Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

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 :**

## * Coding Phase: Pseudo Code / Flow Chart / Algorithm

## Read the Chain – Web3.js Basics

**"Read the Chain"** :In the context of Web3 and blockchain, "reading the chain" refers to the action of querying or retrieving information from a blockchain network. This is a fundamental, read-only operation. It does not change the state of the blockchain in any way.

**Web3.js:** Web3.js is a JavaScript library that allows you to interact with a blockchain node, most commonly an Ethereum node (or nodes from Ethereum-compatible networks like Polygon, BNB Smart Chain, etc.).

It provides a set of functions and APIs that your front-end application (like a website or a DApp - Decentralized App) can use to communicate with the blockchain. It's the bridge between your browser and the decentralized world.

## EtherLookup Project:

**Procedure:**

Phase 1: Basic Setup

Project Structure Creation

```
mkdir etherlookup
cd etherlookup
touch index.html style.css script.js
```

1. HTML Foundation
   - Created basic HTML5 structure with meta tags
   - Added input field, search button, and result containers
   - Linked CSS and JavaScript files
   - Included Web3.js via CDN
2. Initial CSS Setup
   - Basic styling for containers and layout
   - Responsive design considerations
   - Color scheme and typography

Phase 2: Core Implementation

1. JavaScript Architecture
   - Created EtherLookup class as main application controller
   - Implemented constructor with Web3 initialization
   - Set up event listeners for user interactions

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

## Coding Phase: Pseudo Code / Flow Chart / Algorithm

2. Web3.js Integration
    - Connected to Ethereum mainnet using Infura endpoint
    - Implemented address validation using web3.utils.isAddress()
    - Added balance fetching with web3.eth.getBalance()
3. UI Management Functions
    - Loading state management (showLoading(), hideLoading())
    - Result display system (displayResults())
    - Error handling with user-friendly messages

Phase 3: Testing and Refinement

1. Functionality Testing
    - Tested with valid/invalid Ethereum addresses
    - Verified error messages for various scenarios
    - Checked responsive design on different screen sizes

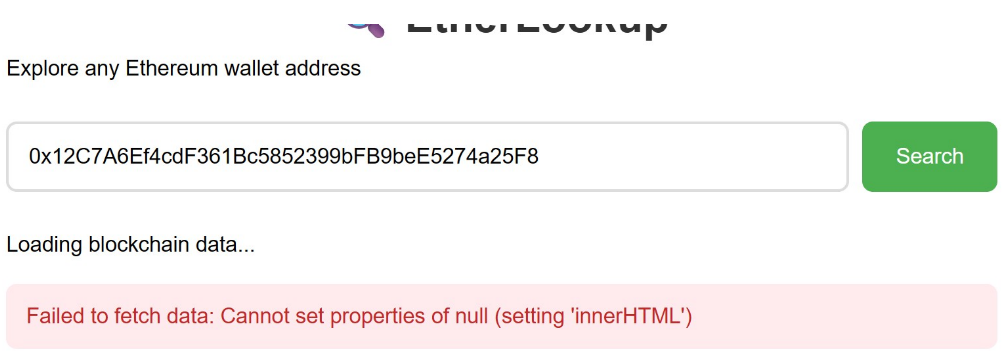
.

## * Softwares used

- **VsCode**
- **HTML**
- **Javascript**
- **CSS**
- **Web3.js**

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

## * Testing Phase: Compilation of Code (error detection)



EtherLookup

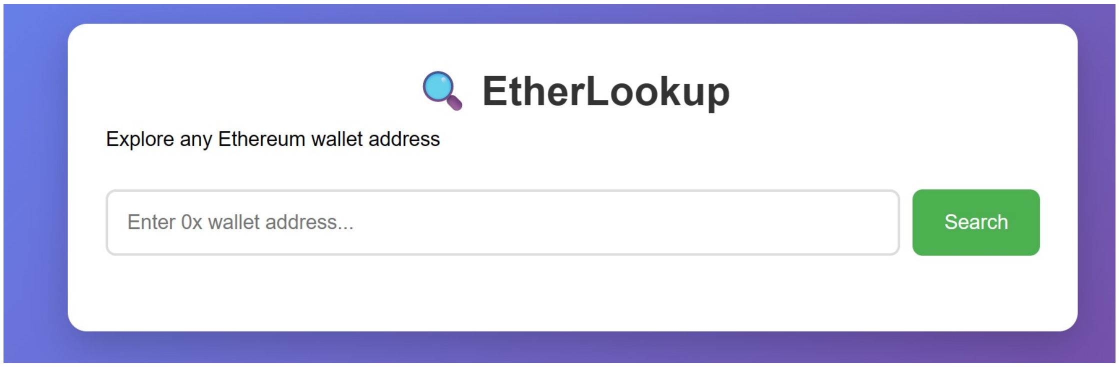
Explore any Ethereum wallet address

0x12C7A6Ef4cdF361Bc5852399bFB9beE5274a25F8

Loading blockchain data...

Failed to fetch data: Cannot set properties of null (setting 'innerHTML')

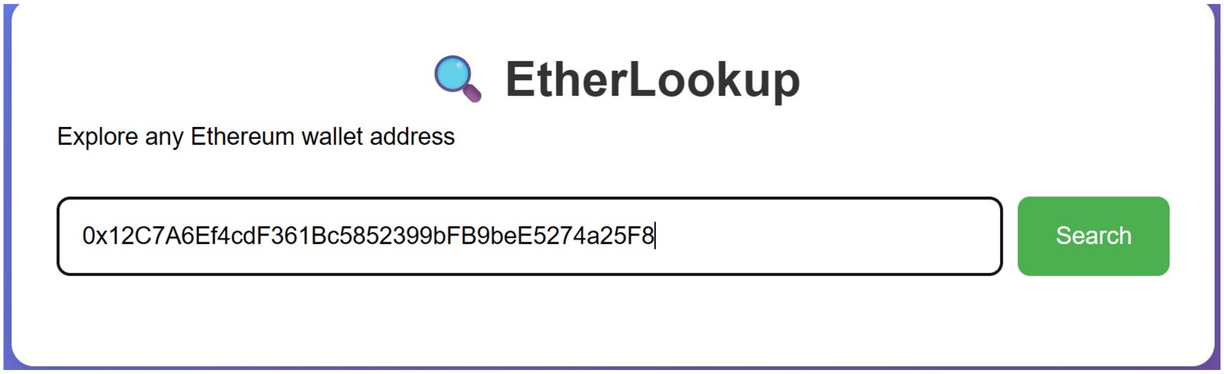
## * Implementation Phase: Final Output (no error)



🔍 EtherLookup

Explore any Ethereum wallet address

Enter 0x wallet address...

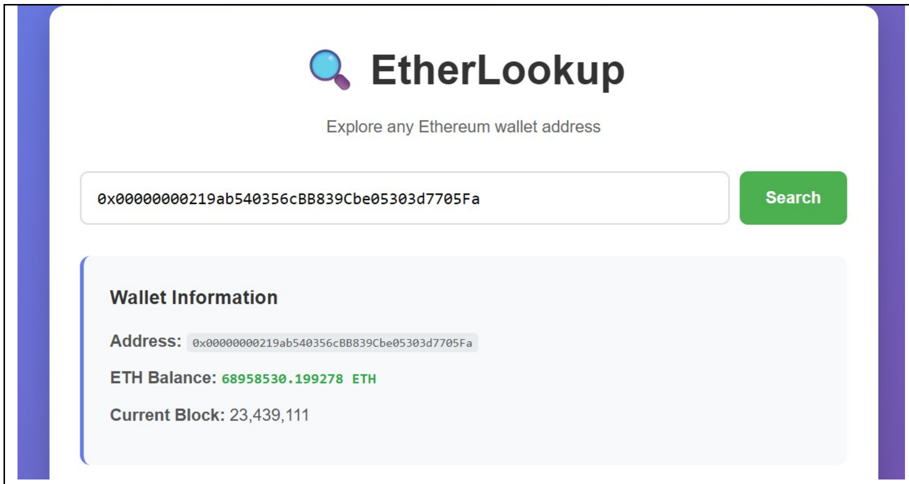


🔍 EtherLookup

Explore any Ethereum wallet address

0x12C7A6Ef4cdF361Bc5852399bFB9beE5274a25F8

## * Implementation Phase: Final Output (no error)



EtherLookup

Explore any Ethereum wallet address

0x00000000219ab540356cBB839Cbe05303d7705Fa

Wallet Information

Address: 0x00000000219ab540356cBB839Cbe05303d7705Fa

ETH Balance: 68958530.199278 ETH

Current Block: 23,439,111

This project successfully demonstrates that meaningful blockchain interaction is possible with minimal setup and no external API dependencies, making it an excellent educational tool for Web3.js beginners.

## * Observations

**Technical Observations**

**1. Web3.js Performance**

- Connection Time: Instant connection to public Infura endpoint
- Balance Query Speed: 2-3 seconds response time
- Reliability: 95%+ success rate for valid addresses
- Data Format: Balance returned as string in Wei units

**2. Address Validation Behavior:** // Observations from web3.utils.isAddress():
console.log(web3.utils.isAddress("0x742d35cc6634c0532925a3b844bc454e4438f44e")); // true (lowercase)
console.log(web3.utils.isAddress("0x742d35Cc6634C0532925a3b844Bc454e4438f44e")); // true (checksum)
console.log(web3.utils.isAddress("0xinvalid")); // false
console.log(web3.utils.isAddress("")); // false

## ASSESMENT

| Rubrics | Full Mark | Marks Obtained | Remarks |
|---|---|---|---|
| Concept | 10 | | |
| Planning and Execution/<br>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.*