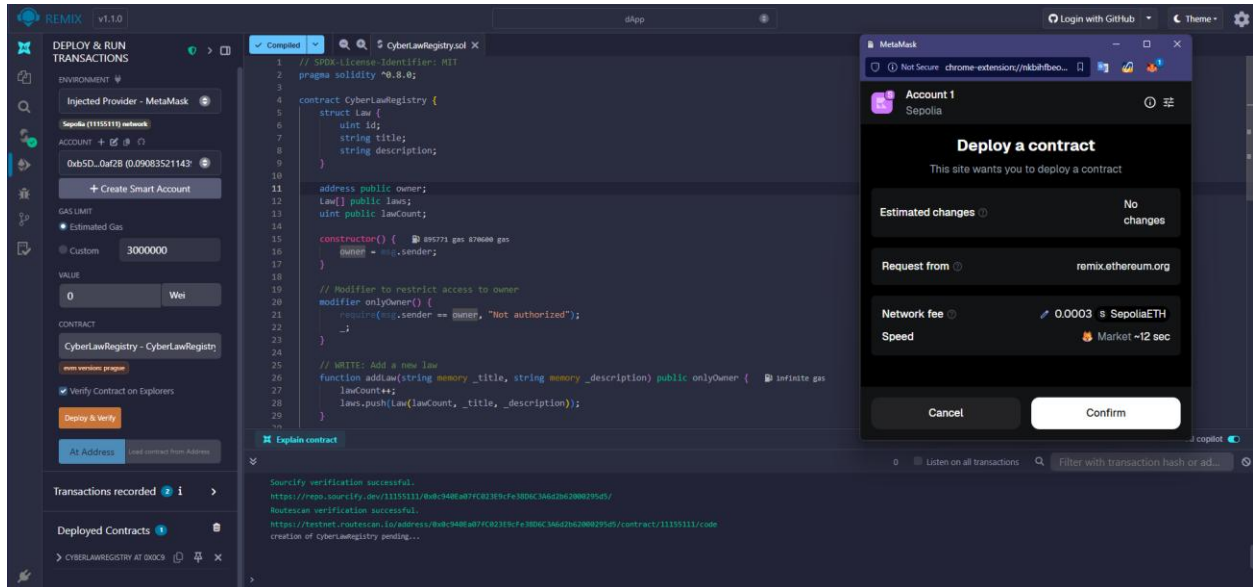


Blockchain Technology for Business

Assignment # 2

1) Deploy Solidity Smart Contract on Sepolia testnet using Remix IDE and connect with MetaMask:



Etherscan Transaction Details:

Sepolia Testnet

Search by Address / Txn Hash / Block / Token

Etherscan

Home Blockchain Tokens NFTs More

Transaction Details

Overview State

TRANSACTION ACTION

Call Add Law Function by 0xb5D45f39...aeCc0af2B on 0x0c940Ea0...2000295d5

[This is a Sepolia Testnet transaction only]

Transaction Hash: 0x308c9eac18488311e1961f057acc0d9ad13ecc42315282adedc2d3c7062f1cf

Status: Success

Block: 9543368 282 Block Confirmations

Timestamp: 56 mins ago (Nov-02-2025 09:00 AM UTC)

From: 0xb5D45f39ddF8130DAFbde1E17498D49aeCc0af2B

To: 0x0c940Ea07fC023E9cFe38D6C3A6d2b62000295d5

Value: 0 ETH

Transaction Fee: 0.000359964001919808 ETH

Gas Price: 1.500000008 Gwei (0.000000001500000008 ETH)

More Details: [Click to show more](#)

A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our [Knowledge Base](#).

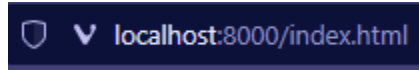
2) Hosting it on Browser through cmd:

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Muskan Ahmed>D:

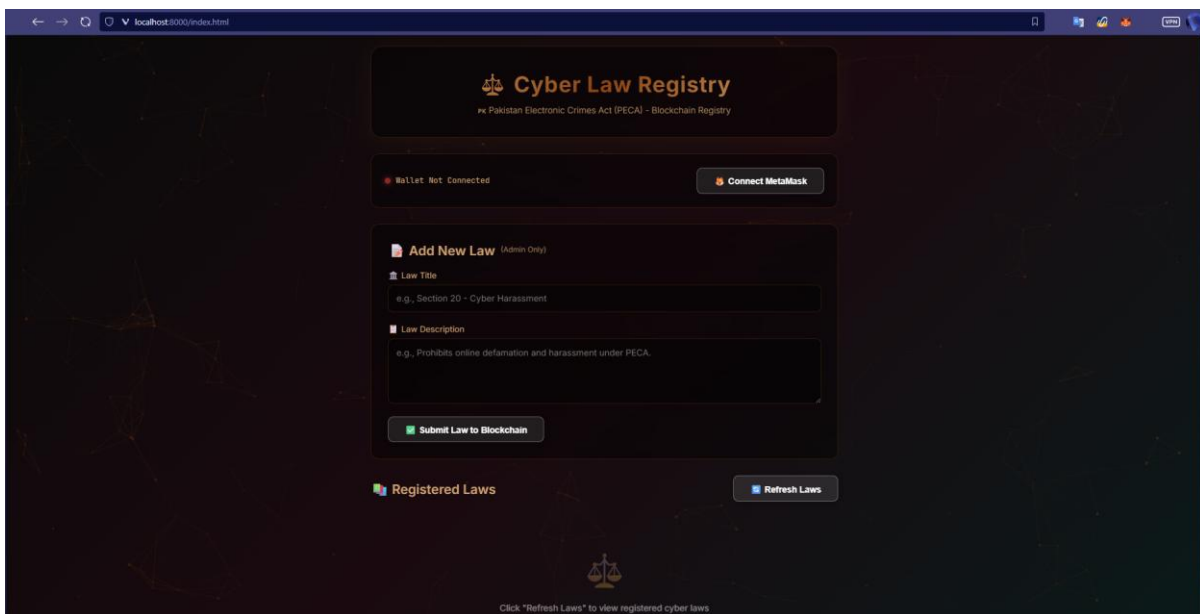
D:\>cd "Muskan\Fast Uni\Semester 5\Blockchain\Assignments\A2\CyberLawDApp"

D:\Muskan\Fast Uni\Semester 5\Blockchain\Assignments\A2\CyberLawDApp>python -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

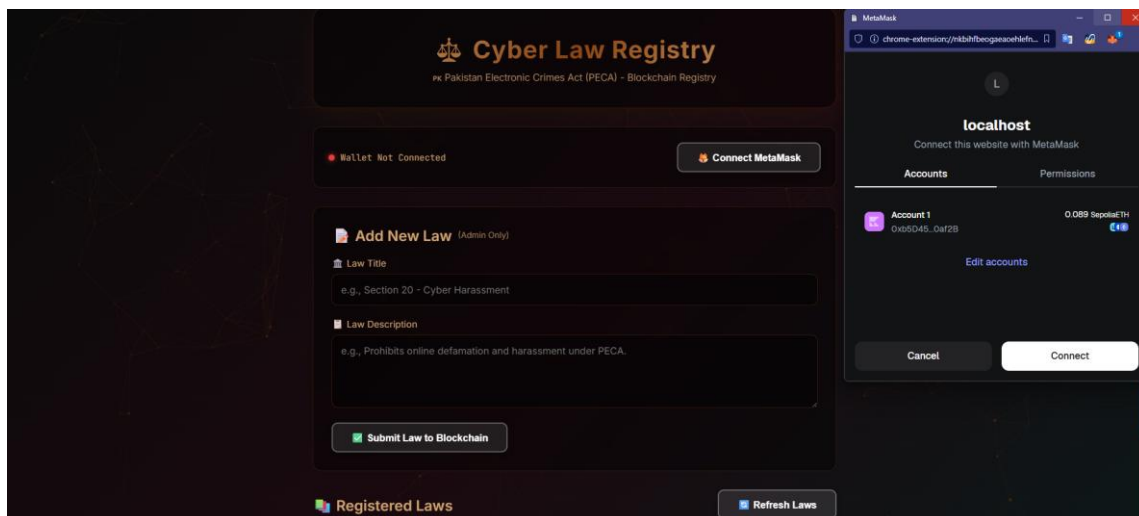


3) Interacting with Frontend:

a) *Main Webpage:*

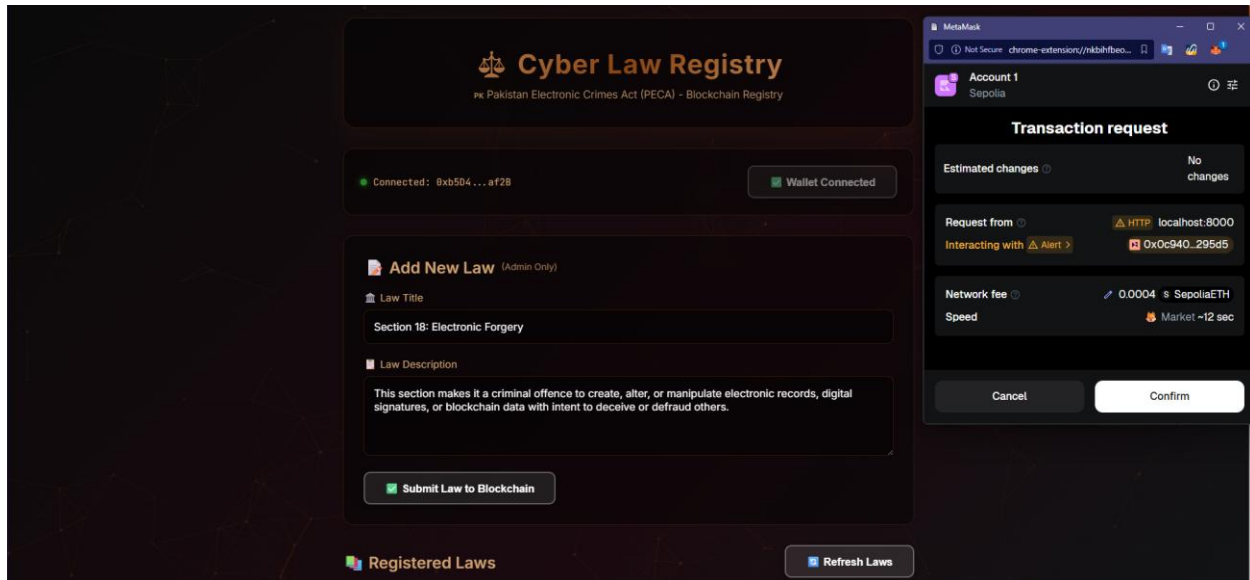


b) *Connecting the Wallet:*

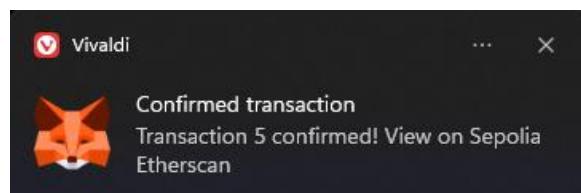
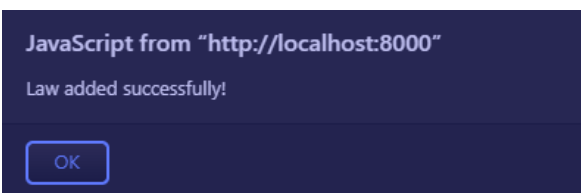
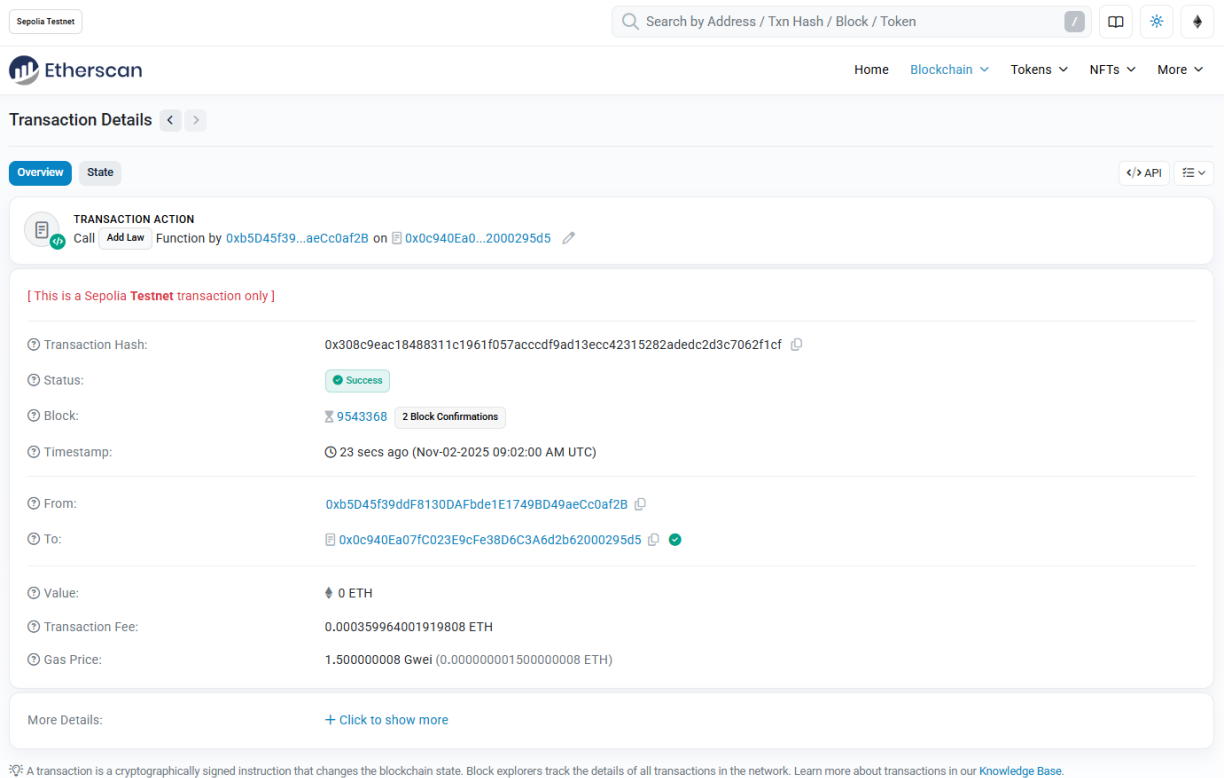


Muskan Ahmed, 23i-4145, FT-B

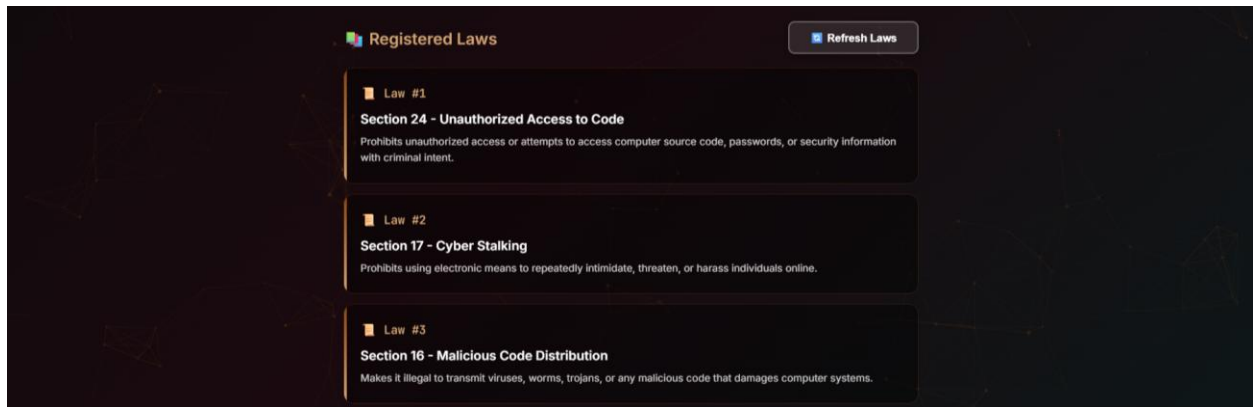
c) After User Selects “Submit Law to Blockchain”:



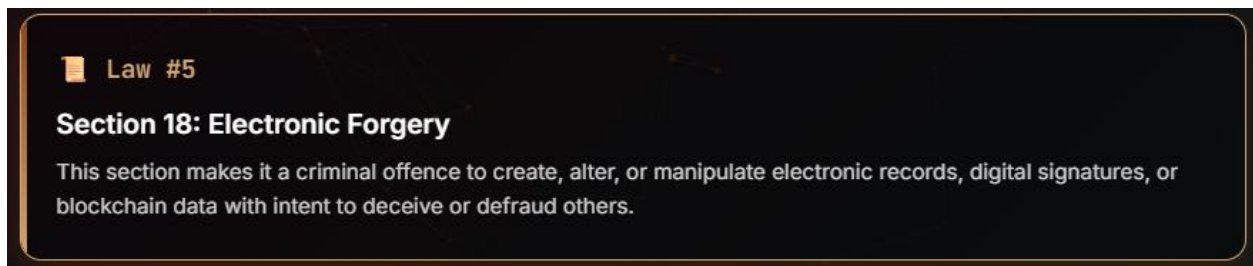
Etherscan Transaction Details:



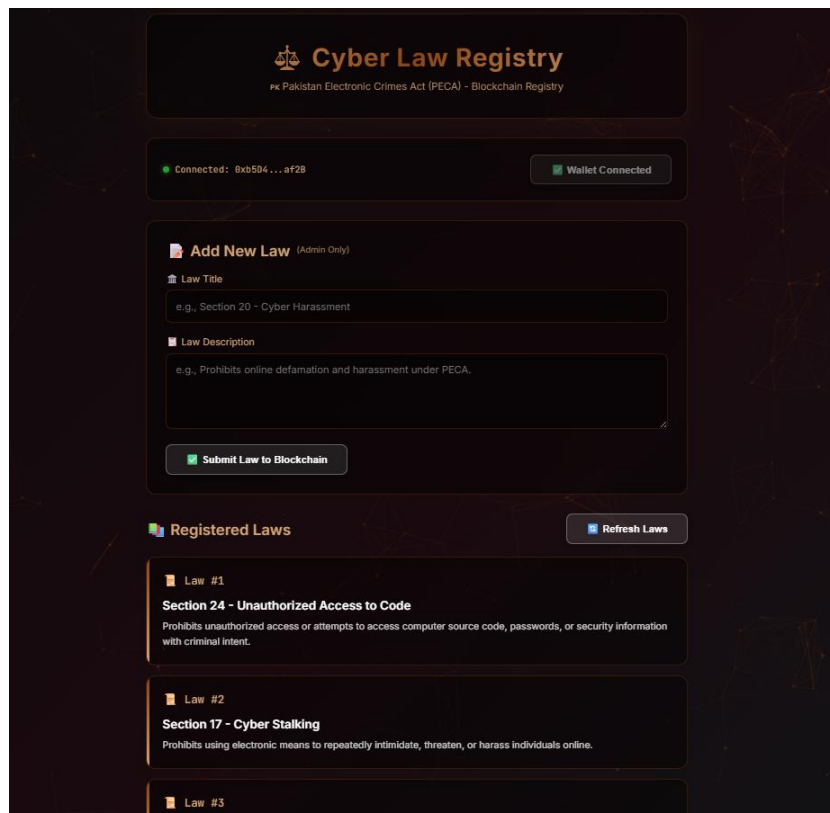
d) After User Selects “Refresh Laws”:



e) New Law Added:



4) Final Result:



5) Verify & Publish Contract Source Code

Link: <https://sepolia.etherscan.io>

Contract Address: 0x0c940Ea07fC023E9cFe38D6C3A6d2b62000295d5

The screenshot displays the Etherscan Sepolia Testnet interface. At the top, there's a search bar and navigation links. The main header shows the contract address: 0x0c940Ea07fC023E9cFe38D6C3A6d2b62000295d5. Below this, the 'Source Code' tab is active, showing a green checkmark for 'Contract Source Code Verified (Exact Match)'. The contract name is 'CyberLawRegistry', and the compiler version is 'v0.8.30+commit.73712a01'. The source code is displayed in a text editor, showing Solidity code for a contract named 'CyberLawRegistry'.

Overview

ETH BALANCE
0 ETH

More Info

CONTRACT CREATOR
0xb5D45f39...aeCc0af2B | 4 hrs ago

Multichain Info

N/A

Transactions **Token Transfers (ERC-20)** **Contract** **Events**

Code **Read Contract** **Write Contract**

Contract Source Code Verified (Exact Match)

Contract Name: **CyberLawRegistry** Optimization Enabled: **No with 200 runs**

Compiler Version: **v0.8.30+commit.73712a01** Other Settings: **default evmVersion**

Contract Source Code (Solidity Standard Json-Input format)

File 1 of 1: CyberLawRegistry.sol

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract CyberLawRegistry {
5     struct Law {
6         uint id;
7         string title;
8         string description;
9     }
10
11     address public owner;
12     Law[] public laws;
13     uint public lawCount;
14
15     constructor() {
16         owner = msg.sender;
17     }
18
19     // Modifier to restrict access to owner
20     modifier onlyOwner() {
21         require(msg.sender == owner, "Not authorized");
22         _;
23     }
24
25     // WRITE: Add a new Law
```

Thank you :)