

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

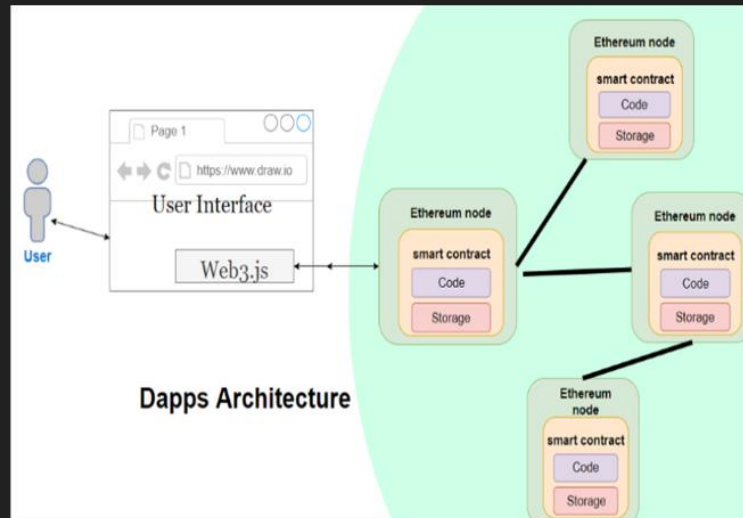
**Тема: “Дослідження, проектування та розробка програмних компонентів для токенизації освітніх активів на основі смарт-контрактів Ethereum”**

ст. г. СПМ-20-2  
Сербін О.М.

## Перелік технологій

- 1) Мова Solidity
- 2) React js
- 3) web3 js

# Архітектура



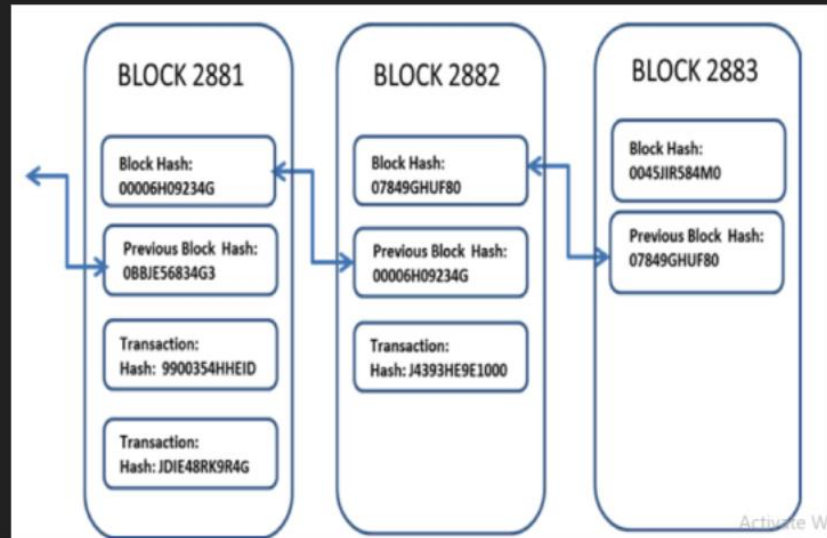
# Реалізація смарт контракту

```

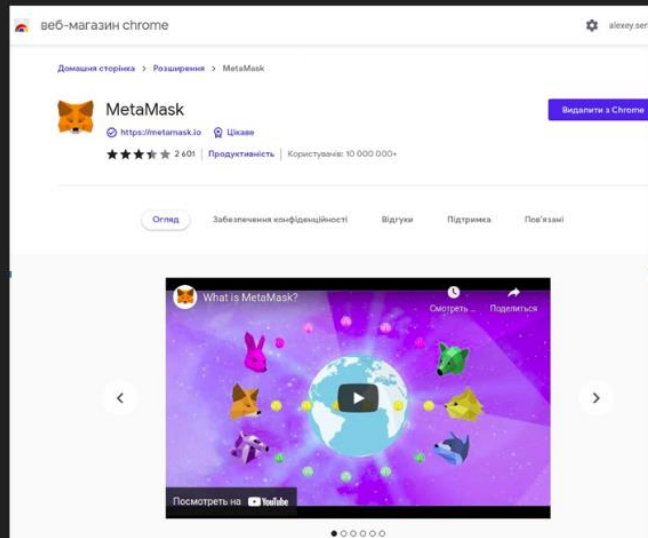
6  contract Storage {
7      struct message{
8          string text;
9          uint256 createdAt;
10         address sender;
11     }
12
13     message[] store;
14
15     function set(string calldata _text) public payable {
16         message memory messageStruct = message({text: _text, createdAt: block.timestamp, sender: msg.sender});
17
18         store.push(messageStruct);
19     }
20 }

```

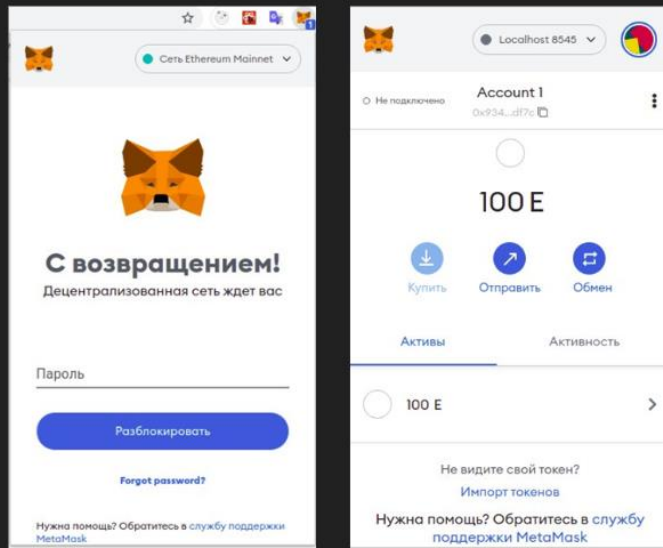
## Графічне уявлення блоків



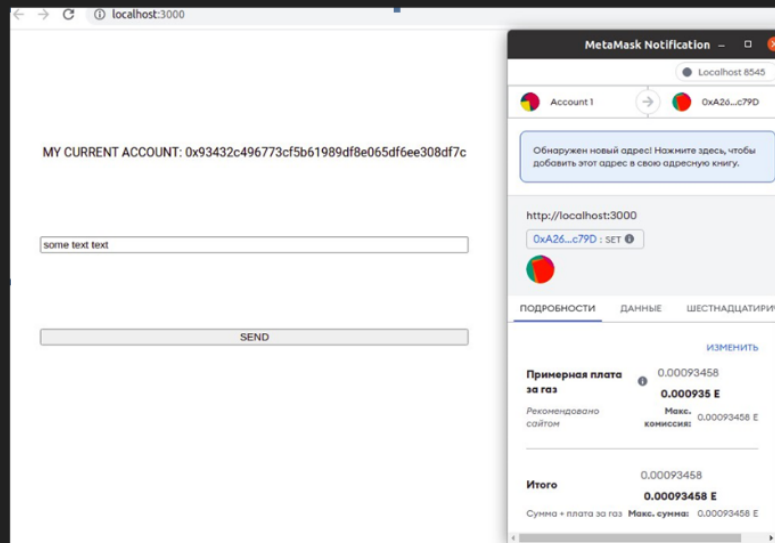
## Авторизація у додатку та інтеграція MetaMask



# MetaMask авторизация



# Додавання блоку у блокчейн





## ДОДАТОК Б

### Програмна реалізація

#### Б.1 Реалізація React js додатку

```

import './App.css';
import Web3 from 'web3/dist/web3.min.js';
import Storage from 'contracts/Storage.json';
import { useEffect, useState } from 'react';

const initWeb3 = async () => {
  if (window.ethereum) {

    return new Web3(window.ethereum);
  }
};

const getAccount = async () => {
  if (window.ethereum) {

    const accounts = await window.ethereum.request({method:
'eth_requestAccounts'});

    return accounts[0];
  }
};

const send = async (web3, text, account) => {
  const id = await web3.eth.net.getId();

  const storageContract = new web3.eth.Contract(
    Storage.abi,
    Storage.networks[id].address
  );

  console.log(await
storageContract.methods.set(text).send({from: account}));
};

function App() {
  const [account, setAccount] = useState()
  const [web3, setWeb3] = useState()
  const [text, setText] = useState('');

  useEffect(async () => {
    setWeb3(await initWeb3());
  });
}

```

```

    }, []);

    useEffect(() => {
      getAccount().then((result) => {
        setAccount(result);
      });
    }, []);

    const onChange = (e) => {
      setText(e.target.value);
    };

    const onSend = () => {
      send(web3, text, account);
    }

    return (
      <div className="App">
        <div className="flex">
          MY CURRENT ACCOUNT: {account}
          <input
            value={text}
            onChange={onChange}
          />
          <button
            onClick={onSend}
          >
            SEND
          </button>
        </div>
      </div>
    );
  }

  export default App;

```

## Б.2 Реалізація смарт-контракту

```

pragma solidity >=0.4.22 <0.9.0;

contract Storage {
  struct message{
    string text;
    uint256 createdAt;
    address sender;
  }

  message[] store;

```

```
function set(string calldata _text) public payable {
    message memory messageStruct = message({text: _text,
createdAt: block.timestamp, sender: msg.sender});

    store.push(messageStruct);
}
}
```