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 : React Start – DApp Frontend Scaffolding

Objective/Aim:

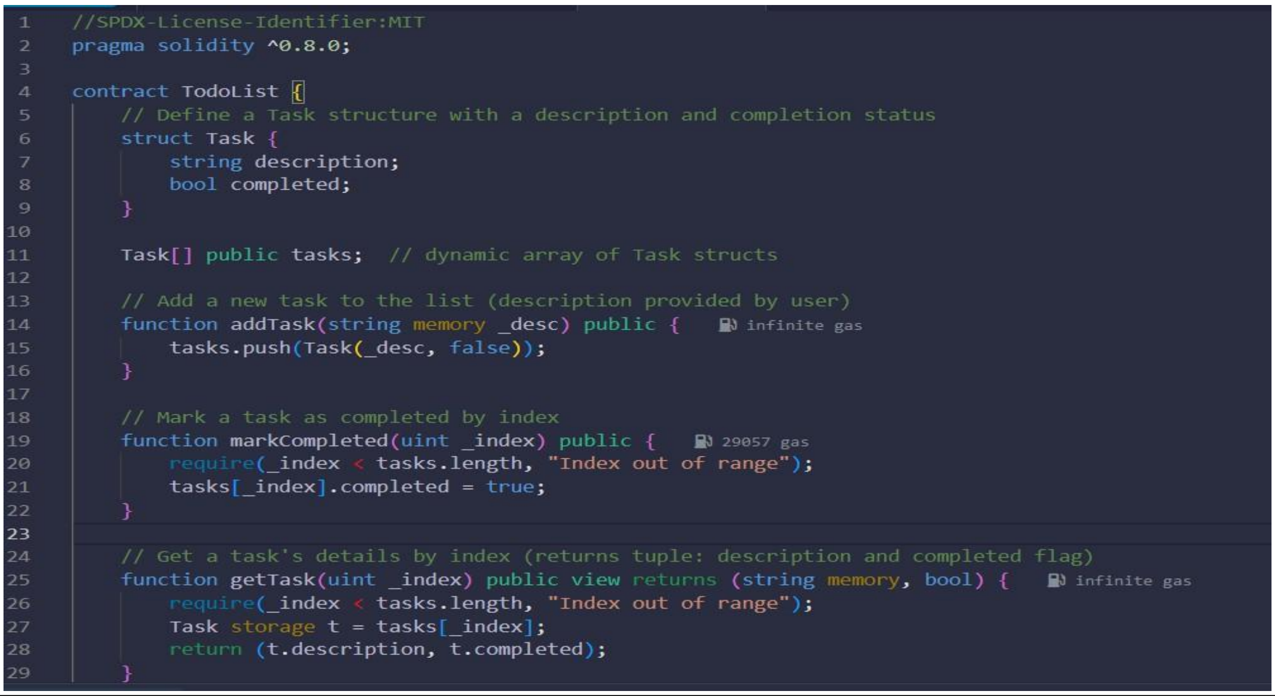
- To build a **TodoList DApp frontend** in React that interacts with a deployed smart contract, allowing users to add tasks and mark them as completed.

Apparatus/Software Used:

- MetaMask
- Remix IDE
- Vs code

Procedure:

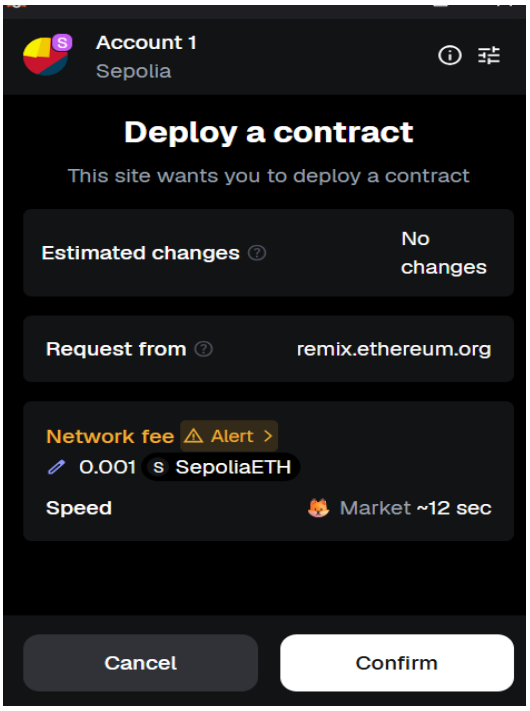
1.Open Remix IDE and write a simple TodoList.sol smart contract.

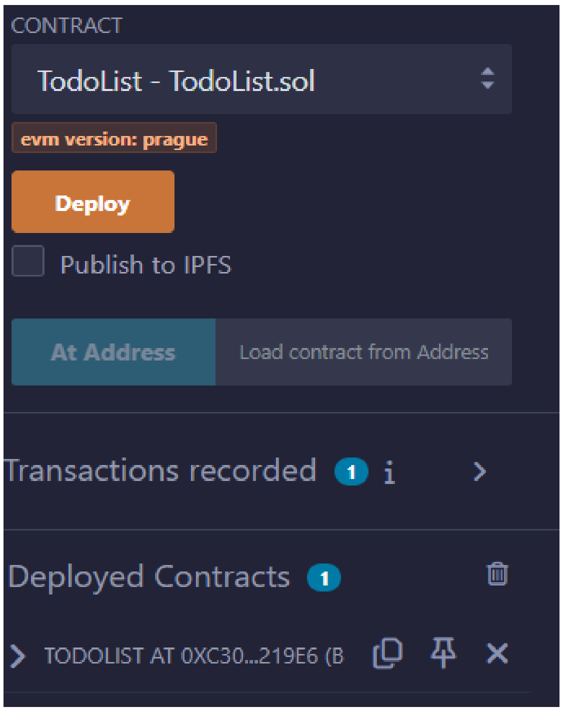


```
//SPDX-License-Identifier:MIT
pragma solidity ^0.8.0;

contract TodoList {
    // Define a Task structure with a description and completion status
    struct Task {
        string description;
        bool completed;
    }

    Task[] public tasks;  // dynamic array of Task structs

    // Add a new task to the list (description provided by user)
    function addTask(string memory _desc) public {    infinite gas
        tasks.push(Task(_desc, false));
    }

    // Mark a task as completed by index
    function markCompleted(uint _index) public {    29057 gas
        require(_index < tasks.length, "Index out of range");
        tasks[_index].completed = true;
    }

    // Get a task's details by index (returns tuple: description and completed flag)
    function getTask(uint _index) public view returns (string memory, bool) {    infinite gas
        require(_index < tasks.length, "Index out of range");
        Task storage t = tasks[_index];
        return (t.description, t.completed);
    }
```

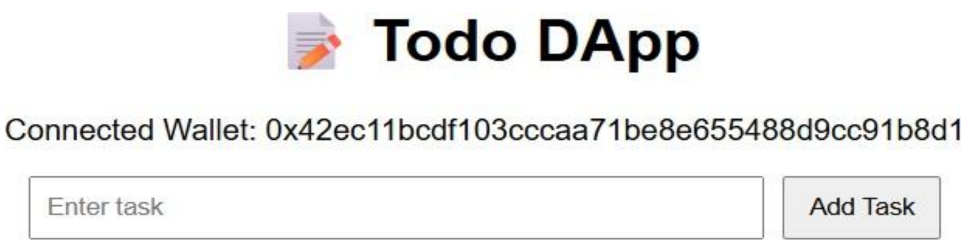
Procedure:

2. After writing smart contract compile it and copy the abi .
3. Then go to the deploye section and delpoye it and confirm transaction on MetaMask.


Account 1
Sepolia
Deploy a contract
This site wants you to deploy a contract
Estimated changes: No changes
Request from: remix.ethereum.org
Network fee ⚠ Alert >
0.001 SepoliaETH
Speed: Market ~12 sec
Cancel
Confirm

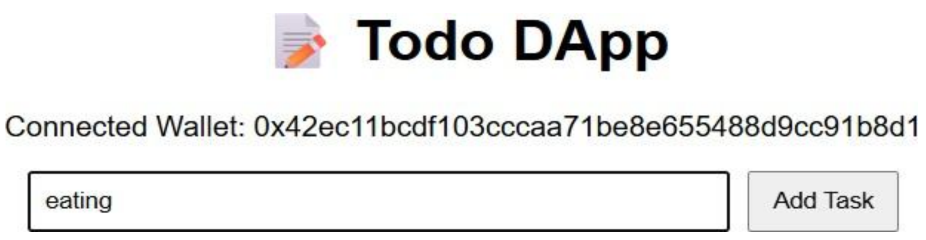

CONTRACT
TodoList - TodoList.sol
evm version: prague
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts 1
TODOLIST AT 0XC30...219E6 (B

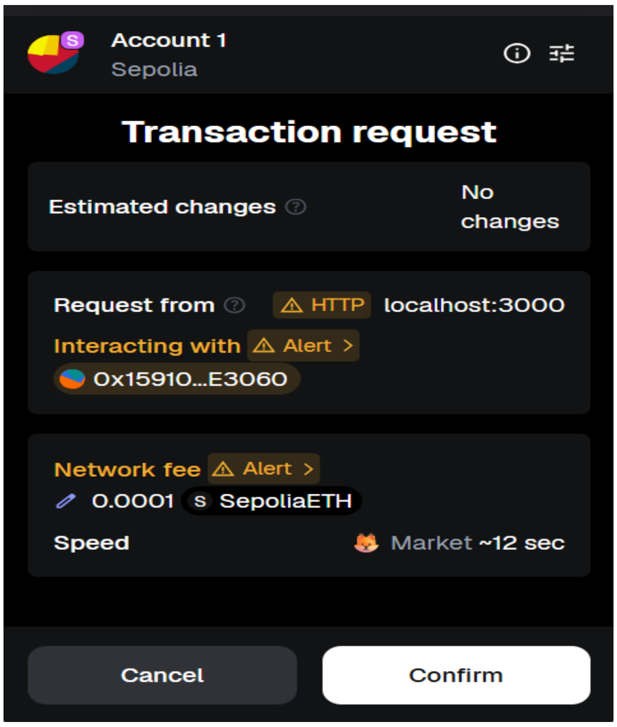
3. Now open Vs code and past the abi and write the frontend in app.js and ether.js after connecting our smart contract with frontend run the the code .


Todo DApp
Connected Wallet: 0x42ec11bcdf103cccaa71be8e655488d9cc91b8d1
Enter task
Add Task

4. Now we test our functions:

- Enter a task and click **Add Task** → MetaMask will pop up for confirmation.


Todo DApp
Connected Wallet: 0x42ec11bcdf103cccaa71be8e655488d9cc91b8d1
eating
Add Task


Account 1
Sepolia

Transaction request

Estimated changes — No changes

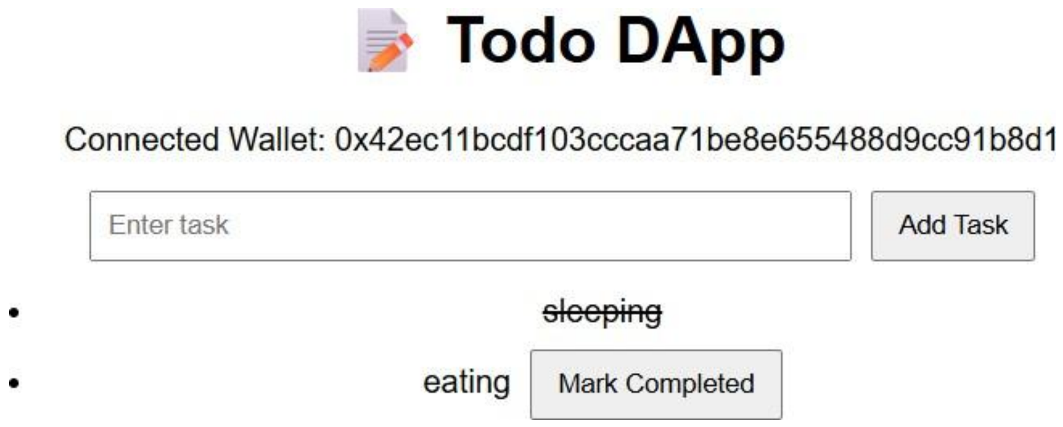
Request from — HTTP localhost:3000

Interacting with — Alert
0x15910...E3060

Network fee — Alert
0.0001 SepoliaETH

Speed — Market ~12 sec

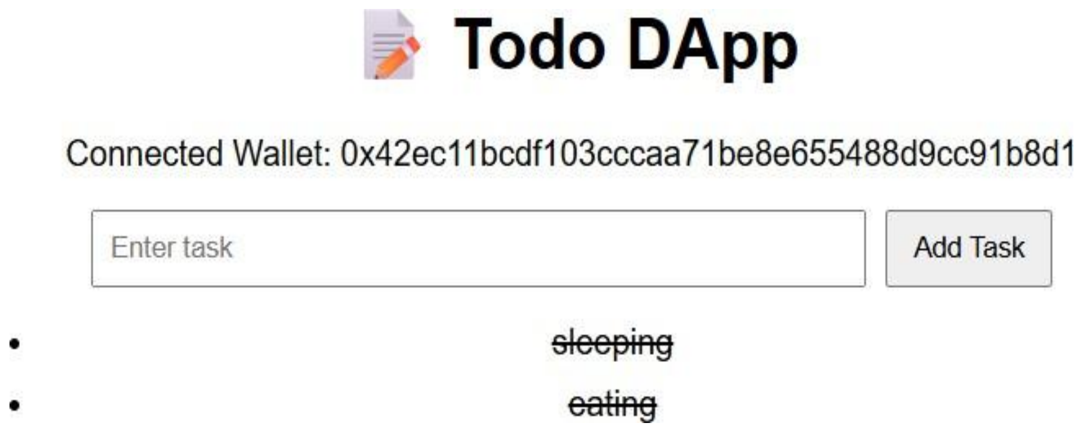
- After confirming, the task appears in the **task list** with (not completed).


Todo DApp

Connected Wallet: 0x42ec11bcdf103cccaa71be8e655488d9cc91b8d1

- ~~sleeping~~
- eating

- Mark the task as completed in Remix or extend frontend with a button → reload page → shows completed.

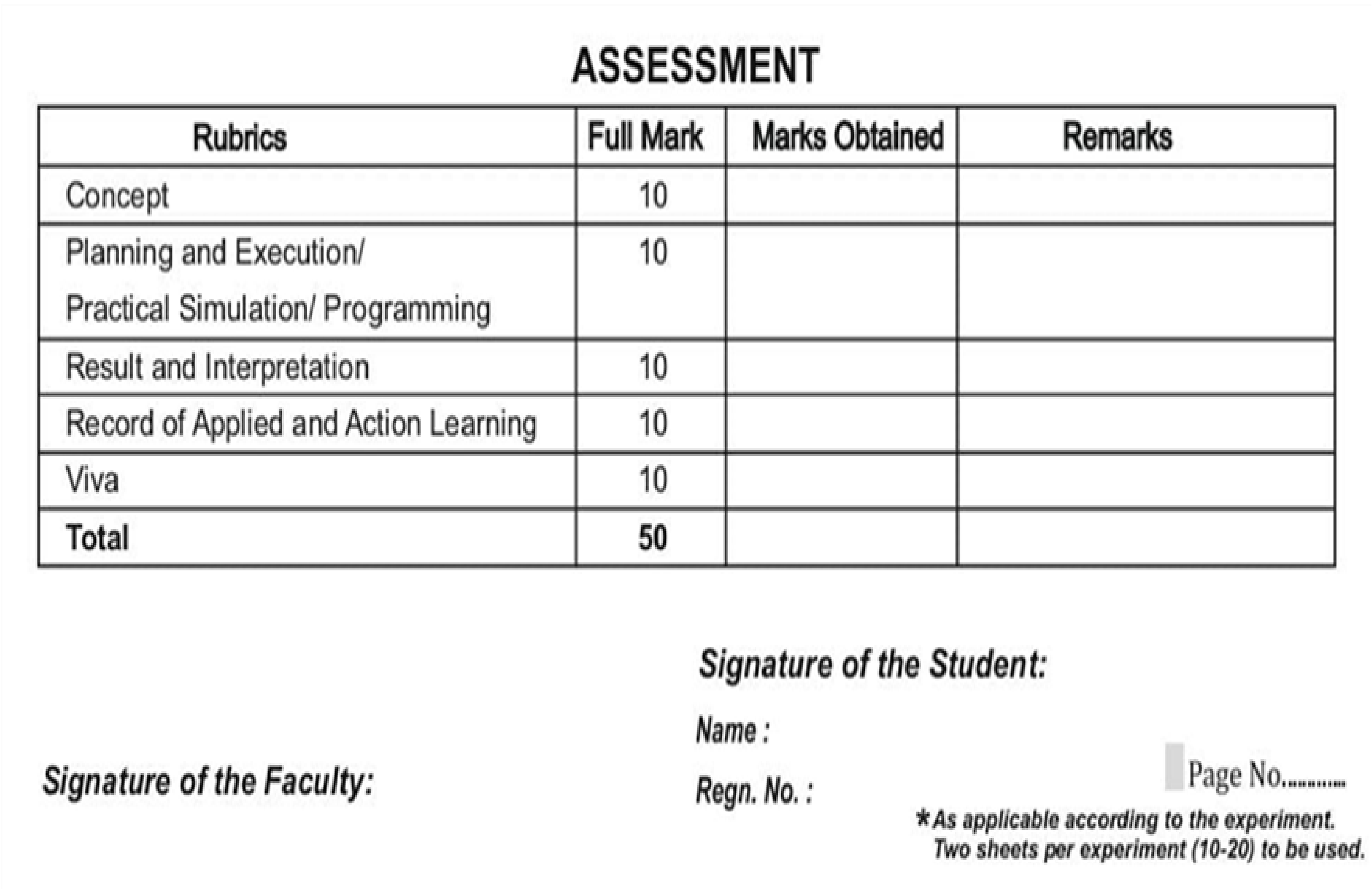

Todo DApp

Connected Wallet: 0x42ec11bcdf103cccaa71be8e655488d9cc91b8d1

- ~~sleeping~~
- ~~eating~~

Observation:

- After **connecting wallet**, the DApp fetches all tasks from blockchain.
- Users can **add tasks** (stored on-chain).
- Clicking **Complete/Undo** updates the task’s status on blockchain.
- UI dynamically updates from the blockchain state.



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:

Page No..........

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