Simple To-Do list Dapp

Objective: Create a DApp to store and manage to-do lists.
Key Features:

Smart contract to add, delete, and view tasks.
DApp frontend to interact with the contract.
Tasks are stored immutably on the blockchain.

Objectives:

Solidity events, state management, and frontend-backend interaction.

Tools:

Solidity, Truffle, Ganache, web3.js, and React/HTML

Step 1: Install Prerequisites

1. **Install Node.js**:
   Download and install Node.js from nodejs.org. This will also install npm (Node Package Manager).
2. **Install Visual Studio Code (VS Code)**:
   Download and install VS Code from code.visualstudio.com.
3. **Install MetaMask**:
   Install the MetaMask extension from metamask.io.
4. **Install Ganache**:
   Download and install Ganache from trufflesuite.com/ganache

Step 2: creating the project

1. **Install Truffle**:
   Open a terminal in VS Code and install Truffle globally by running:

```
npm install -g truffle
```

2. **Initialize Truffle Project**:
   In the terminal, navigate to your project folder and run:

```
truffle init
```

This will set up a basic Truffle project structure.

3. **Configure Truffle for Ganache**:
   In the project folder, open the truffle-config.js file and configure the development network to use Ganache:

truffle-config.js:

```
module.exports = {
  networks: {
        development: {
        host: "127.0.0.1",
        port: 7545, // Ganache default port
        network_id: "*" // Match any network id
        }
  },
  compilers: {
        solc: {
        version: "0.8.0"
        }
  }
};
```

4.add solidity code

Inside your project folder, create a file named SimpleToDoList.sol.write the following Solidity code into SimpleToDoList.sol:

SimpleToDoList.sol :

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract SimpleToDoList {
        string public task;
        // Function to set a task
        function setTask(string memory _task) public {

        task = _task;

        }

        // Function to get the current task

        function getTask() public view returns (string memory) {

        return task;

        }

}
```

5.Deploy Contract:
In the migrations folder, create a file 2_deploy_contracts.js:

```
const SimpleToDoList = artifacts.require("SimpleToDoList");

module.exports = function (deployer) {

  deployer.deploy(SimpleToDoList);

};
```

Now, open Ganache and start a new workspace to run a local blockchain.

6.Run Migration:
In the terminal, run the following command to deploy the contract to Ganache:

```
truffle migrate --network development
```

Step 3: Set Up the Frontend

1. **Create an index.html File**:
   In the project folder, create a file named index.html and paste the following code:

```
<!DOCTYPE html>

<html lang="en">

<head>

        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Simple To-Do List DApp</title>
        <script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>
</head>
<body>
        <h1>Simple To-Do List DApp</h1>
        <p>Connect MetaMask and interact with the blockchain!</p>
        <!-- Set Task -->
        <h3>Set Task</h3>
        <input id="taskInput" type="text" placeholder="Enter a task..." />
        <button onclick="setTask()">Set Task</button>
        <!-- Get Task -->
        <h3>Current Task</h3>
        <button onclick="getTask()">Get Task</button>
        <p id="taskOutput">Task will appear here...</p>
        <script>
        let contract;
        let account;
        const loadBlockchain = async () => {
            if (window.ethereum) {
                const web3 = new Web3(window.ethereum);
                await window.ethereum.request({ method: "eth_requestAccounts" });
                const accounts = await web3.eth.getAccounts();
```

```
                    account = accounts[0];

                    const abi = [

                    { "inputs": [{ "internalType": "string", "name": "_task", "type": "string" }],
"name": "setTask", "outputs": [], "stateMutability": "nonpayable", "type": "function" },

                    { "inputs": [], "name": "getTask", "outputs": [{ "internalType": "string",
"name": "", "type": "string" }], "stateMutability": "view", "type": "function" },

                    { "inputs": [], "name": "task", "outputs": [{ "internalType": "string", "name":
"", "type": "string" }], "stateMutability": "view", "type": "function" }

                    ];

 const contractAddress = " 0x6293FF1Ed36d69D1923b4A1F7f3CB1DE0e5F453f";

                    contract = new web3.eth.Contract(abi, contractAddress);

          } else {

                    alert("MetaMask not installed!");

          }

          };

          const setTask = async () => {

          const taskInput = document.getElementById("taskInput").value;

          await contract.methods.setTask(taskInput).send({ from: account });

          alert("Task set successfully!");

          };

          const getTask = async () => {

          const task = await contract.methods.getTask().call();

          document.getElementById("taskOutput").innerText = task;

          };

          window.addEventListener("load", loadBlockchain);
```

```
        </script>

</body>

</html>
```