

# Decentralized Greeting DApp

**Objective:** Expand the basic greeting smart contract into a functional DApp.

## Key Features:

Create a smart contract to store and update greetings.

Build a frontend where users can read and update the greeting.

Connect the frontend with Ethereum using web3.js.

## Objectives:

Basics of smart contracts, web3.js integration, and DApp structure.

## Tools:

Solidity, Truffle, Ganache, MetaMask, and HTML/JavaScript.

## 1. Set Up the Development Environment

1. Install **Node.js**:  
Download and install [Node.js](#).
2. Install **Truffle Suite**:  
Run the following command to install Truffle globally:

```
npm install -g truffle
```

3. Install **Ganache**:  
Download and install Ganache from Truffle Suite.
4. Install **MetaMask**:  
Add the [MetaMask extension](#) to your browser.

## 2. Create the Project

1. Open a terminal and create a new Truffle project:

```
mkdir GreetingDApp
```

```
cd GreetingDApp
```

```
truffle init
```

2. Install **web3.js** for frontend interaction:

```
npm install web3
```

### 3. Write the Smart Contract

1. Create a new Solidity file under the contracts folder (e.g., Greeting.sol).
2. Add the following code for the Greeting smart contract:

solidity

```
// SPDX-License-Identifier: MIT
```

```
pragma solidity ^0.8.0;
```

```
contract Greeting {
```

```
    string public greeting;
```

```
    constructor(string memory _greeting) {
```

```
        greeting = _greeting;
```

```
    }
```

```
    function setGreeting(string memory _newGreeting) public {
```

```
        greeting = _newGreeting;
```

```
    }
```

```
    function getGreeting() public view returns (string memory) {
```

```
        return greeting;
```

```
    }
```

```
}
```

### 4. Deploy the Smart Contract

1. Create a deployment script under the migrations folder (e.g., 2\_deploy\_greeting.js):

javascript

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

module.exports = function (deployer) {

    deployer.deploy(Greeting, "Hello, Blockchain!");

};
```

2. Start Ganache:
  - Open Ganache and start a new workspace.
3. Compile the smart contract:

```
truffle compile
```

4. Migrate the smart contract to Ganache:

```
truffle migrate
```

## 5. Connect MetaMask to Ganache

1. Open MetaMask and click the network dropdown (top center).  
Select **"Add Network"** and enter:
  - **Network Name:** Ganache
  - **New RPC URL:** `http://127.0.0.1:7545`
  - **Chain ID:** 1337
  - **Currency Symbol:** ETH
2. Import an account from Ganache into MetaMask:
  - Copy a private key from Ganache (click the key icon next to an account).
  - In MetaMask, go to **"Import Account"**, paste the private key, and click **Import**.

### Index.html:

```
<!DOCTYPE html>

<html>

<head>

    <title>Greeting DApp</title>

    <script src="https://cdn.jsdelivr.net/npm/web3/dist/web3.min.js"></script>

</head>
```

```
<body>
```

```
  <h1>Decentralized Greeting DApp</h1>
```

```
  <p id="currentGreeting"></p>
```

```
  <input type="text" id="newGreeting" placeholder="Enter new greeting" />
```

```
  <button onclick="updateGreeting()">Set Greeting</button>
```

```
<script>
```

```
let web3 = new Web3(Web3.givenProvider || "http://127.0.0.1:7545");
```

```
let contractAddress = "0xF4bF2A1b53505aa12a502484ee41df672c1909e2";
```

```
let abi = [{
```

```
  "inputs": [
```

```
  {
```

```
    "internalType": "string",
```

```
    "name": "_greeting",
```

```
    "type": "string"
```

```
  }
```

```
],
```

```
  "stateMutability": "nonpayable",
```

```
  "type": "constructor"
```

```
},
```

```
{
```

```
  "inputs": [],
```

```
  "name": "greeting",
```

```
"outputs": [  
  {  
    "internalType": "string",  
    "name": "",  
    "type": "string"  
  }  
],  
"stateMutability": "view",  
"type": "function",  
"constant": true  
},  
{  
  "inputs": [  
    {  
      "internalType": "string",  
      "name": "_newGreeting",  
      "type": "string"  
    }  
  ],  
  "name": "setGreeting",  
  "outputs": [],  
  "stateMutability": "nonpayable",  
  "type": "function"
```

```

    },
    {
      "inputs": [],
      "name": "getGreeting",
      "outputs": [
        {
          "internalType": "string",
          "name": "",
          "type": "string"
        }
      ],
      "stateMutability": "view",
      "type": "function",
      "constant": true
    }
  ]];

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

async function loadGreeting() {
  const greeting = await contract.methods.getGreeting().call();

  document.getElementById("currentGreeting").innerText = `Current Greeting:
${greeting}`;
}

async function updateGreeting() {
  const accounts = await web3.eth.getAccounts();

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

```

```
    await contract.methods.setGreeting(newGreeting).send({ from: accounts[0] });  
    loadGreeting();  
  }  
  loadGreeting();  
</script>  
</body>  
</html>
```

Run the above code with live server the output will be seen in the <https://localhost:5500>

## Greeting DApp

Loading greeting...

## Decentralized Greeting DApp

Current Greeting:



# Decentralized Greeting DApp

Current Greeting: congrats

Set Greeting

Ganache

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK7

GAS PRICE2000000000

GAS LIMIT6721975

HARDFORKMERGE

NETWORK ID5777

RPC SERVERHTTP://127.0.0.1:7545

MINING STATUSAUTOMINING

WORKSPACEQUICKSTART

SAVE

SWITCH

MNEMONIC

actress crisp ensure stay cattle output link inhale vessel deputy joy rare

HD PATH

m44'60'0'0account\_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x9B495D763C122068F83d9047ea9C39cea1cF9556	100.00 ETH	6	0	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xa9d0e855c5111c952F6B31F409Bf8cA2B0f71Db4	90.00 ETH	1	1	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x0Ffb1119B4584454979D40DA34d201965bB6A706	100.00 ETH	0	2	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x68Ea8C2542B1282412F631F190a6b902Ae8c1724	110.00 ETH	0	3	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x5cF3522eF5DB9df3204e876679eDf5c31C869455	100.00 ETH	0	4	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x6e4999789F9f543A5aD4da76401BFaa004Fa9655	100.00 ETH	0	5	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x305029b9d87b7cDcD30c8C992ceD330BEec619e0	100.00 ETH	0	6	