Mini Project

Name : Rahul Sampatrao Patil

Roll No : 4176

Subject : Blockchain Technology

Aim : - To create a secure, transparent, and efficient decentralized electronic voting system using Blockchain technology that enables users to cast their votes online while maintaining the integrity and privacy of the voting process.

Working :-

- The voter must first login to the app using their Aadhar number and OTP confirmation.
- After authentication, the voter can now begin the process of voting.
- If the voter is valid then the voting palliate will be opened with candidate names.
- Now, the voter can cast their vote by clicking on the vote button next to their preferred candidate.
- A voter can cast their vote only once. After which, the voter's automatically logged out.
- Same process continues for many more voter's irrespective of their voting wards.
- Vote count for each candidate is displayed on the portal, next to the candidate.

Technology Stack :-

- React.js
- Truffle
- Ganache
- Metamask
- Web3.js

- **Implementation :-**

A.contracts

1.Migrations.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Migrations {
  address public owner = msg.sender;
  uint public last_completed_migration;

  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
    );
    _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

2.Voting.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Voting {
    uint256 public candidateIdTracker;
    struct Candidate {
        uint256 id;
        string name;
        uint256 votes;
    }

    mapping(uint256 => Candidate) public candidateList;

    function createCandidate(string memory name) private {
        Candidate memory candidate;
        candidate.id = candidateIdTracker;
        candidate.name = name;
        candidate.votes = 0;
        candidateList[candidateIdTracker] = candidate;
        candidateIdTracker++;
    }

    constructor(string[] memory candidates) {
        candidateIdTracker = 0;
        // ["Candidate A", "Candidate B", "Candidate C"]
        for (uint256 i = 0; i < candidates.length; i++) {
            createCandidate(candidates[i]);
        }
    }

    mapping(uint40 => bool) public hasVoted;

    function voteCandidate(uint40 aadhaarId, uint256 candidateId) public {
        require(hasVoted[aadhaarId] == false, "ALREADY VOTED");
        hasVoted[aadhaarId] = true;
        candidateList[candidateId].votes++;
    }
}
```

B. migrations

1.1_initial_migration.js

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

module.exports = function (deployer) {
  deployer.deploy(Migrations);
};
```

2.2_deploy_voting.js

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

module.exports = function (deployer) {
  deployer.deploy(Voting, ["Candidate A", "Candidate B", "Candidate C"]);
};
```

C.UI

1.index.html

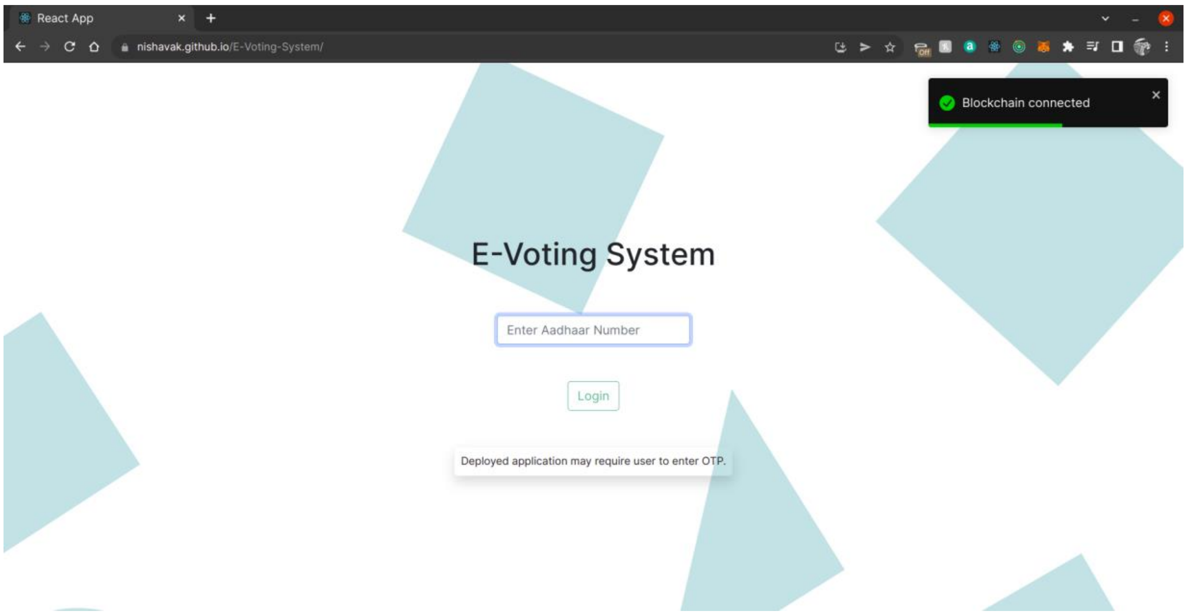
```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta name="theme-color" content="#000000" />
  <meta name="description" content="Web site created using create-react-app" />
  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-
manifest/
    -->
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
  <title>React App</title>
</head>

<body oncontextmenu="return false;">
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
</body>

</html>
```

2.manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

System UI :-

1.Aadhaar authentication



React App

nishavak.github.io/E-Voting-System/

Blockchain connected

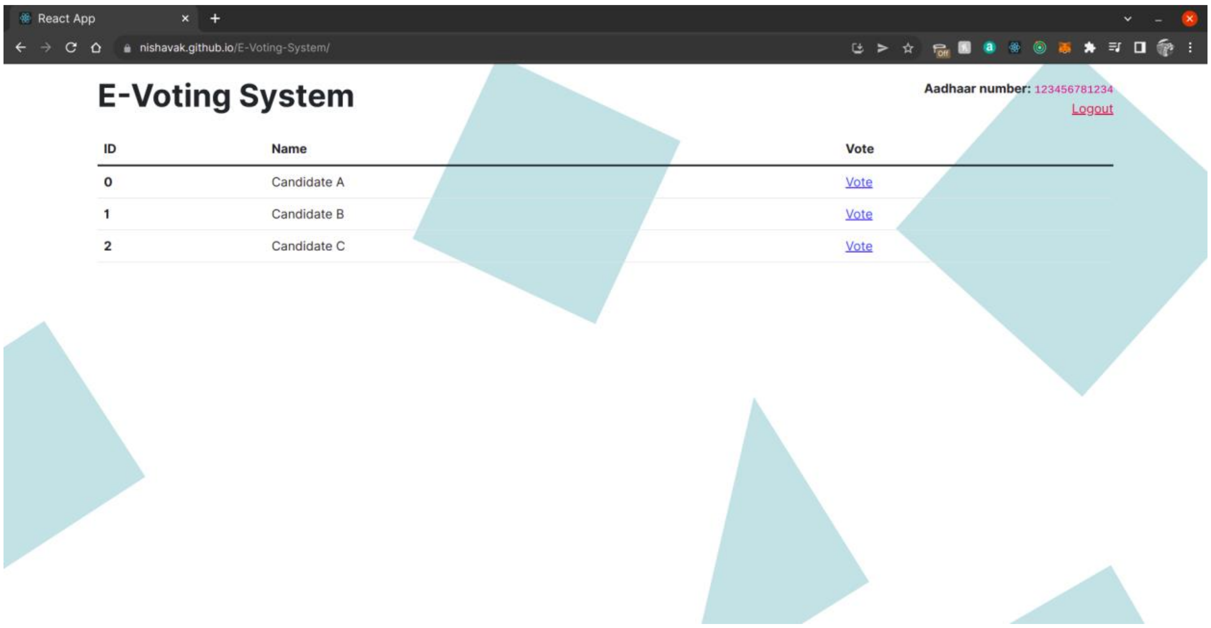
E-Voting System

Enter Aadhaar Number

Login

Deployed application may require user to enter OTP.

2.After successful Aadhaar authentication



React App

nishavak.github.io/E-Voting-System/

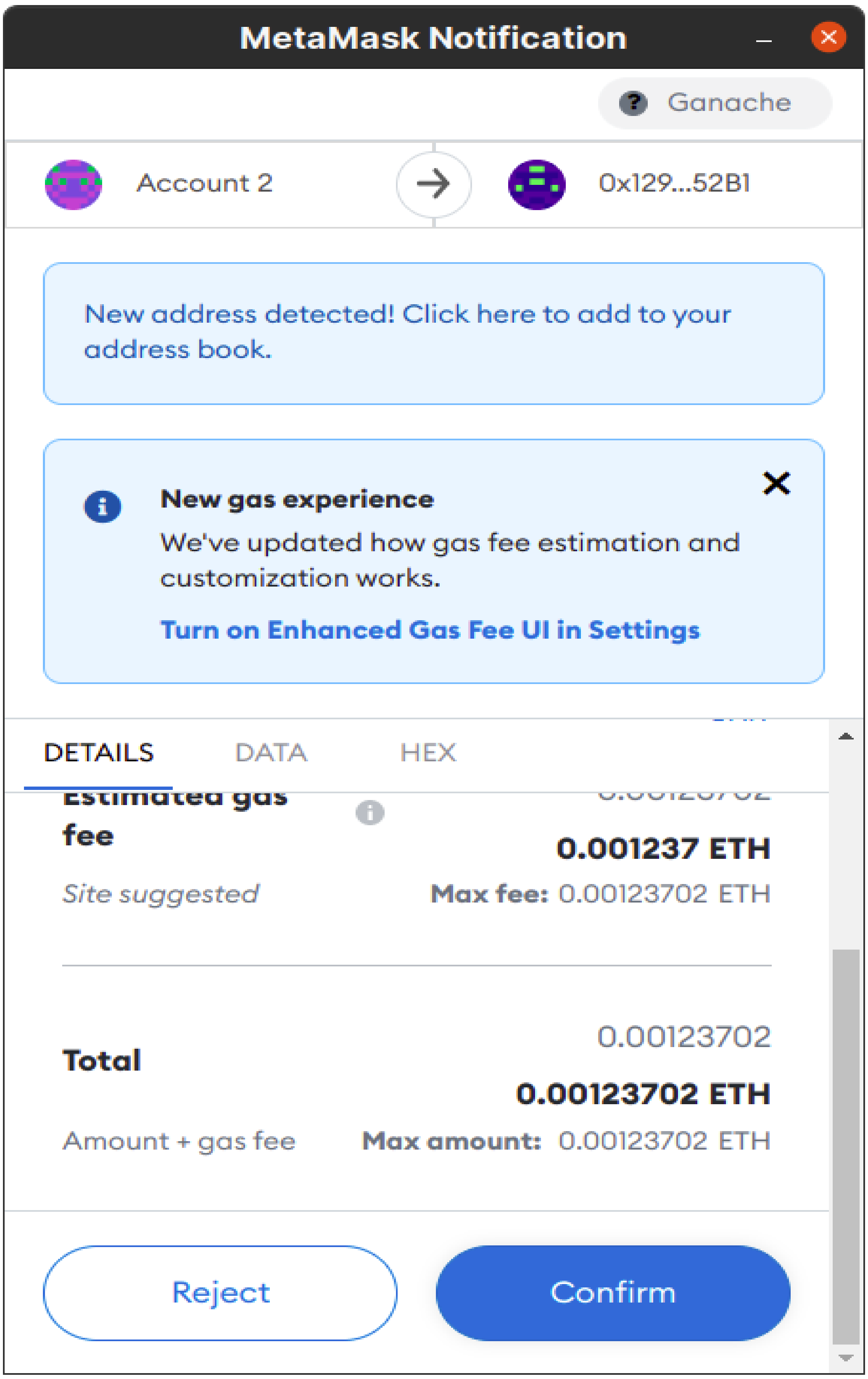
E-Voting System

Aadhaar number: 123456781234

Logout

ID	Name	Vote
0	Candidate A	Vote
1	Candidate B	Vote
2	Candidate C	Vote

3.Metamask confirmation to vote a selected candidate



MetaMask Notification

Ganache

Account 2 → 0x129...52B1

New address detected! Click here to add to your address book.

New gas experience
We've updated how gas fee estimation and customization works.
Turn on Enhanced Gas Fee UI in Settings

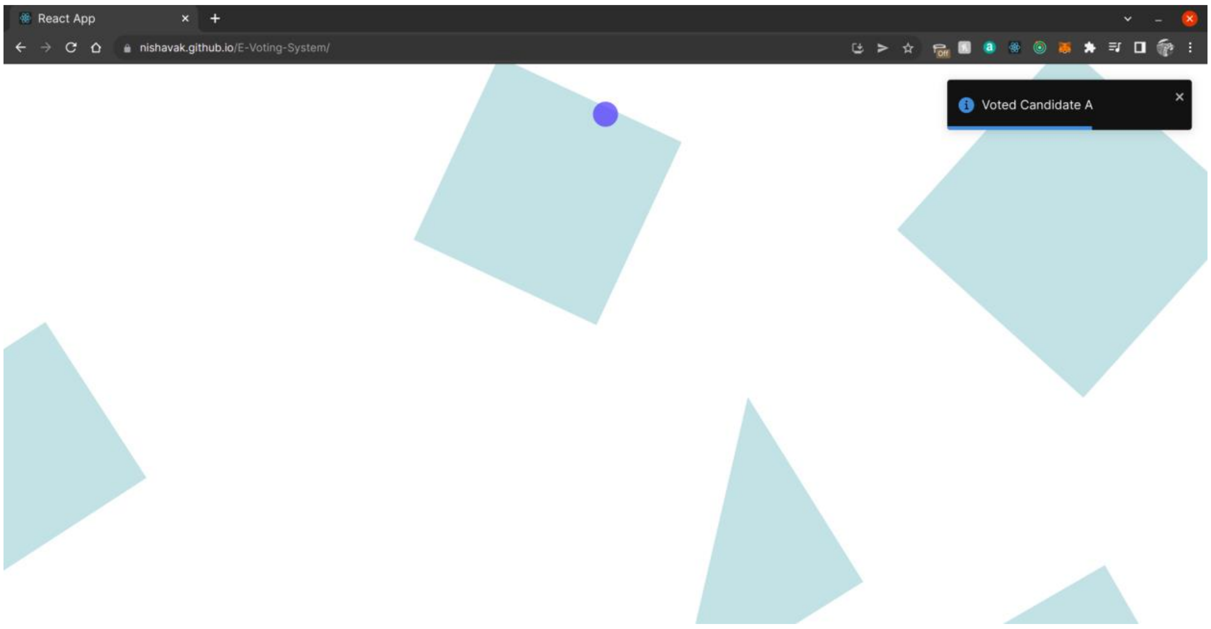
DETAILS DATA HEX

Estimated gas fee 0.00123702
0.001237 ETH
Site suggested **Max fee:** 0.00123702 ETH

Total 0.00123702
0.00123702 ETH
Amount + gas fee **Max amount:** 0.00123702 ETH

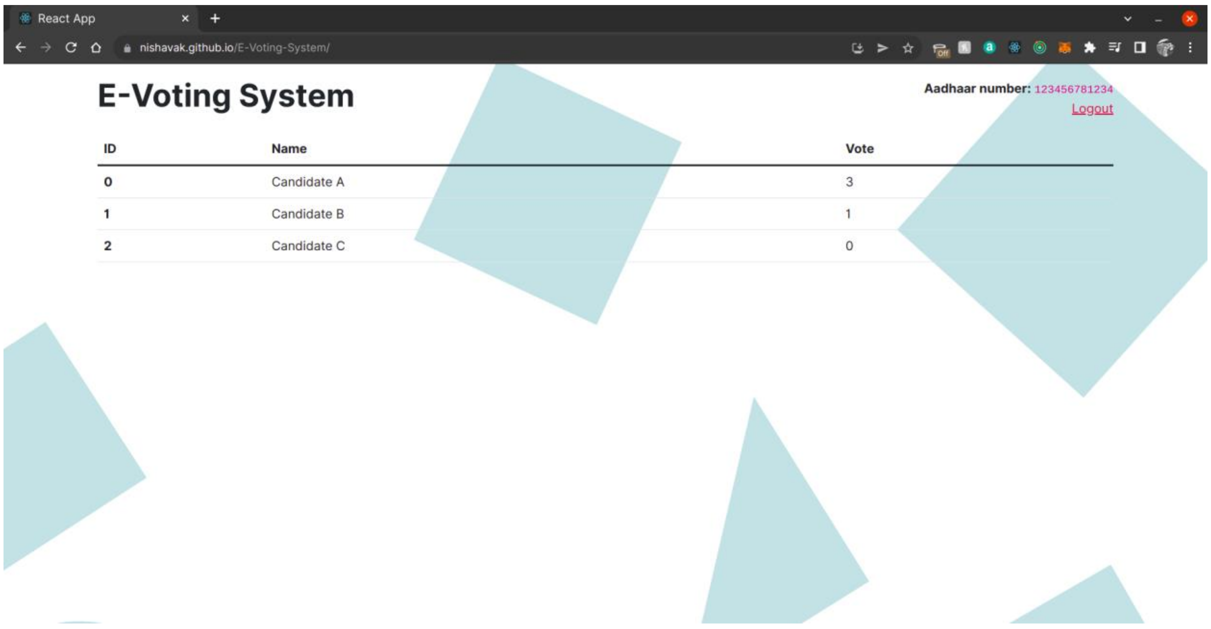
Reject Confirm

4.Successfully voted a candidate prompt



Voted Candidate A

5.Logging in again after voting; candidate vote incrementend. Who votes whom isn't logged.



E-Voting System

Aadhaar number: 123456781234

Logout

ID	Name	Vote
0	Candidate A	3
1	Candidate B	1
2	Candidate C	0