Decentralized Voting System

This project is a complete blockchain-based voting application built using React, Solidity, Hardhat, and Ganache, designed to provide a secure, transparent, and tamper-proof voting process through Ethereum smart contracts. The system features a decentralized voting mechanism where voters can self-register and candidates can be added dynamically without predefined entries. It includes owner controls to start or stop the election, supports real-time vote counting and winner display, and provides progress tracking through visual charts. The application also integrates seamlessly with MetaMask for wallet-based authentication, operates in a Ganache-powered local blockchain environment for testing and development, and maintains comprehensive event logging to ensure transparency and reliability. With its responsive UI, the platform delivers a smooth voting experience across both desktop and mobile devices.

Prerequisites

Required Software

1. **Node.js (v16.0.0 or higher)**
   - Download: https://nodejs.org/
   - Verify installation: `node --version` and `npm -version`
2. **Ganache (Ethereum blockchain simulator)**
   - Download GUI: https://trufflesuite.com/ganache/
   - Alternative CLI: `npm install -g ganache-cli`
3. **MetaMask Browser Extension**
   - Install from: https://metamask.io/
   - Supported browsers: Chrome, Firefox, Edge, Brave

Required Libraries & Dependencies

```
{
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "ethers": "^6.8.0",
    "web-vitals": "^3.4.0"
```

```
  },
  "devDependencies": {
    "@nomicfoundation/hardhat-toolbox": "^3.0.0",
    "@nomicfoundation/hardhat-ethers": "^3.0.0",
    "hardhat": "^2.17.2"
  }
}
```

Installation Guide

Step 1: Environment Setup

1.1 Install Node.js

```
# Check if Node.js is installed
node --version
npm --version
# If not installed, download from https://nodejs.org/
```

1.2 Setup Ganache

```
# Download and install Ganache GUI from the official website
# OR install Ganache CLI globally
npm install -g ganache-cli

# Start Ganache GUI:
# - Create new workspace
# - Set RPC Server to http://127.0.0.1:7545
# - Set Network ID to 1337
# - Note down account addresses and private keys
```

1.3 Setup MetaMask

```
# Install MetaMask browser extension
# Create or import wallet
# Add custom network with these settings:
# - Network Name: Ganache Local
# - RPC URL: http://127.0.0.1:7545
# - Chain ID: 1337
# - Currency Symbol: ETH
# - Block Explorer URL: (leave blank)
```

Step 2: Project Creation

2.1 Create Project Directory

```
mkdir voting-system
cd voting-system
```

2.2 Initialize Project (Option 1 - Manual Setup)

```
# Initialize npm project
npm init -y

# Install React dependencies
npm install react react-dom react-scripts web-vitals

# Install Ethereum dependencies
npm install ethers

# Install Hardhat and tools
npm install --save-dev @nomicfoundation/hardhat-toolbox
@nomicfoundation/hardhat-ethers hardhat

# Create directory structure
mkdir contracts scripts src src/contracts public
```

2.2 Alternative Setup (Option 2 - Using Create React App)

```
# Create React app first
npx create-react-app voting-system
cd voting-system

# Install additional dependencies
npm install ethers
npm install --save-dev @nomicfoundation/hardhat-toolbox
@nomicfoundation/hardhat-ethers hardhat

# Create additional directories
mkdir contracts scripts
```

Step 3: File Creation

You need to create the following files:

3.1 Smart Contract (`contracts/VotingSystem.sol`)

- Solidity smart contract with pre-loaded candidates

3.2 Hardhat Configuration (`hardhat.config.js`)

- Configure Hardhat to work with Ganache network

3.3 Deployment Script (`scripts/deploy.js`)

- Script to deploy the smart contract to Ganache

3.4 React Frontend (`src/App.js`)

- Complete React application with voting interface

3.5 Styling (`src/App.css`)

- Modern responsive CSS for the application

3.6 Configuration Files

- `package.json`, `src/index.js`, `public/index.html`

Step 4: Configuration

4.1 Update Hardhat Configuration

```
// Replace with your Ganache private keys in hardhat.config.js
accounts: [
  "0xYOUR_GANACHE_PRIVATE_KEY_1",
  "0xYOUR_GANACHE_PRIVATE_KEY_2",
  "0xYOUR_GANACHE_PRIVATE_KEY_3"
]
```

4.2 Get Ganache Private Keys

1. Open Ganache GUI
2. Go to "Accounts" tab
3. Click the key icon next to each account
4. Copy the private key
5. Update `hardhat.config.js`

Step 5: Deployment

5.1 Compile Smart Contract

```
npx hardhat compile
```

5.2 Deploy to Ganache

```
# Make sure Ganache is running first!
npx hardhat run scripts/deploy.js --network development
```

Expected Output:

```
Starting deployment of VotingSystem contract...
VotingSystem deployed to: 0x[CONTRACT_ADDRESS]
Initial Candidates:
   1. Subhash (ID: 1, Votes: 0)
   2. Abhik (ID: 2, Votes: 0)
   3. Pranav (ID: 3, Votes: 0)
Deployment completed successfully!
```

5.3 Start Frontend Application

```
npm start
```

The application will open at http://localhost:3000

Project Structure

```
voting-system/
├── package.json                 # Dependencies and scripts
├── hardhat.config.js           # Hardhat configuration
├── README.md                   # This file
├── contracts/
│   └── VotingSystem.sol        # Smart contract
├── scripts/
│   └── deploy.js               # Deployment script
├── src/
│   ├── App.js                  # Main React component
│   ├── App.css                 # Styling
│   ├── index.js                # React entry point
│   ├── index.css               # Base styles
│   └── contracts/
│       └── VotingSystem.json   # Generated after deployment
├── public/
│   └── index.html              # HTML template
├── cache/                      # Generated by Hardhat
└── artifacts/                  # Generated by Hardhat
```

Configuration

Ganache Settings

- **RPC Server**: http://127.0.0.1:7545
- **Network ID**: 1337
- **Chain ID**: 1337
- **Port**: 7545

MetaMask Network Settings

- **Network Name**: Ganache Local
- **RPC URL**: http://127.0.0.1:7545
- **Chain ID**: 1337
- **Currency Symbol**: ETH

Usage

For Voters

1. Connect Wallet

- Open the application
- Connect MetaMask to Ganache network
- Ensure you're on the correct account

2. Register to Vote

- Click "Register to Vote" button
- Confirm transaction in MetaMask
- Wait for confirmation

3. Cast Your Vote

- Select your preferred candidate (Subhash, Abhik, or Pranav)
- Click "Vote for [Candidate Name]"
- Confirm transaction in MetaMask

4. View Results

- See real-time vote counts
- Check current leader
- View vote percentages

For Owner (Contract Deployer)

1. Manage Voting

- Use owner account (first account used for deployment)
- Start/Stop voting using toggle button
- Monitor voting activity

2. View Analytics

- Total votes cast
- Individual candidate performance
- Winner announcement

Smart Contract Details

Core Functions

Public Functions

- `getAllCandidates()` - Retrieve all candidates with vote counts
- `vote(uint256 candidateId)` - Cast vote for specific candidate
- `registerSelf()` - Self-register as voter
- `getWinner()` - Get current leading candidate
- `isRegistered(address voter)` - Check registration status
- `hasVoted(address voter)` - Check voting status

Owner Functions

- `addCandidate(string name)` - Add new candidate
- `registerVoter(address voter)` - Register specific voter
- `toggleVoting()` - Start/stop voting process

View Functions

- `candidatesCount()` - Total number of candidates
- `totalVotes()` - Total votes cast
- `votingActive()` - Current voting status

Events

- `VoteCast(address voter, uint256 candidateId)` - Vote recorded
- `VoterRegistered(address voter)` - New voter registered
- `CandidateAdded(uint256 candidateId, string name)` - Candidate added
- `VotingStatusChanged(bool active)` - Voting status changed

Troubleshooting

Common Issues & Solutions

1. "react-scripts: command not found"

```
# Solution
npm install
```

```
# or
npm install react-scripts --save
```

2. "Error loading contract"

```
# Make sure contract is deployed
npx hardhat run scripts/deploy.js --network development

# Check if VotingSystem.json exists in src/contracts/
```

3. "Transaction failed"

- Ensure MetaMask is connected to Ganache network (Chain ID: 1337)
- Check if you have sufficient ETH for gas fees
- Verify you're registered to vote
- Confirm voting is active

4. "Network connection error"

- Start Ganache application
- Verify Ganache is running on http://127.0.0.1:7545
- Check MetaMask network configuration

5. "Private key error in Hardhat"

- Copy actual private keys from Ganache accounts
- Update `hardhat.config.js` with correct keys
- Ensure keys start with "0x"

Reset Everything

```
# Clean Hardhat cache
npx hardhat clean

# Remove node modules and reinstall
rm -rf node_modules package-lock.json
npm install

# Recompile and redeploy
npx hardhat compile
npx hardhat run scripts/deploy.js --network development

# Restart React app
npm start
```