1. Run backend

cd backend

node server.js

1. Frontend

cd frontend

python -m http.server 8000

or if node.js install  
npx serve -s . -l 8000

2. go to this website now  
[**http://localhost:8000**](http://localhost:8000)

# Decentralized Voice Authentication System - Complete Documentation

## Overview

This project implements a decentralized voice authentication system that combines voice biometrics with blockchain technology. Users can register their voiceprints and authenticate themselves by speaking a passphrase. All authentication attempts are recorded on the Polygon Amoy testnet, providing a tamper-proof audit trail.

## Prerequisites & Required Installations

### 1. Software Installation

#### Node.js and npm

* **Download**: <https://nodejs.org/>
* **Version**: v16 or higher
* **Installation**:
  + Download the installer for your operating system
  + Follow the installation wizard
  + Verify installation:
  + node --version
  + npm --version

#### MetaMask Browser Extension

* **Download**: <https://metamask.io/>
* **Installation**:
  + Install the extension in your browser (Chrome, Firefox, Brave, etc.)
  + Create a new wallet or import an existing one
  + Save your seed phrase in a secure location

#### FFmpeg

* **Download**: <https://ffmpeg.org/download.html>
* **Installation**:
  + **Windows**:
    - Download from <https://www.gyan.dev/ffmpeg/builds/>
    - Extract to a location like C:\ffmpeg
    - Add C:\ffmpeg\bin to your system PATH
    - Verify installation:
    - ffmpeg -version
  + **macOS**:
  + brew install ffmpeg
  + **Linux**:
  + sudo apt update
  + sudo apt install ffmpeg

#### Git (Optional but recommended)

* **Download**: <https://git-scm.com/>
* **Installation**:
  + Download and install Git for your operating system
  + Verify installation:
  + git --version

### 2. Account Setup

#### Alchemy Account

* **Sign Up**: <https://www.alchemy.com/>
* **Create App**:
  + Log in to your Alchemy dashboard
  + Click "Create App"
  + Select "Polygon" as the chain
  + Select "Amoy" as the network
  + Give your app a name (e.g., "Voice Auth")
  + Note down the API key (found in the app details)

#### MetaMask Configuration

* **Add Polygon Amoy Testnet**:
  + Open MetaMask
  + Click on the network selector at the top
  + Click "Add Network"
  + Enter these details:
    - Network Name: Polygon Amoy Testnet
    - RPC URL: https://polygon-amoy.g.alchemy.com/v2/YOUR\_ALCHEMY\_API\_KEY
    - Chain ID: 80002
    - Currency Symbol: MATIC
    - Block Explorer: https://amoy.polygonscan.com/
* **Get Test MATIC**:
  + Visit <https://faucet.polygon.technology/>
  + Enter your wallet address
  + Complete the verification process
  + Request test MATIC

## Technologies Used

### Blockchain & Smart Contracts

* **Blockchain Platform**: Polygon Amoy Testnet (Chain ID: 80002)
* **Smart Contract Language**: Solidity
* **Development Framework**: Hardhat
* **Contract Interaction**: Ethers.js

### Backend

* **Runtime**: Node.js
* **Framework**: Express.js
* **Audio Processing**: FFmpeg via fluent-ffmpeg
* **File Upload**: Multer
* **Blockchain Interaction**: Ethers.js
* **Environment Management**: dotenv

### Frontend

* **Languages**: HTML5, CSS3, Vanilla JavaScript
* **Audio Recording**: Web Audio API
* **Server**: Static file serving via serve package

### Infrastructure

* **RPC Provider**: Alchemy
* **Wallet**: MetaMask
* **Audio Processing**: FFmpeg

## System Architecture

The system consists of three main components:

1. **Smart Contract**: Deployed on Polygon Amoy testnet
   * Stores user voiceprints as hashes
   * Records authentication attempts
   * Provides functions for registration and authentication
2. **Backend Server** (Node.js/Express)
   * Processes audio files using FFmpeg
   * Extracts voiceprint features
   * Interacts with the blockchain via smart contract
   * Serves as API for the frontend
3. **Frontend Interface** (Web-based)
   * Provides UI for registration and authentication
   * Captures audio via browser's microphone
   * Communicates with backend API

## Project Setup & Package Installation

### 1. Clone or Create Project

If using Git:

git clone [repository-url]

cd voice-auth-system

Or create manually:

mkdir voice-auth-system

cd voice-auth-system

mkdir backend frontend smart-contract

### 2. Smart Contract Setup

cd smart-contract

npm init -y

npm install --save-dev hardhat @nomicfoundation/hardhat-toolbox dotenv

Create .env:

PRIVATE\_KEY=081ac2fef5931bf8cb01a5c992836852b31679b6046fa8f41ab847d3fd016a7f

RPC\_URL=https://polygon-amoy.g.alchemy.com/v2/pXsYxdVEMCwq7QJkAnWtT

Compile and deploy:

npx hardhat compile

npx hardhat run scripts/deploy.js --network polygonAmoy

### 3. Backend Setup

cd ../backend

npm init -y

npm install express multer ethers dotenv fluent-ffmpeg

Create .env:

PRIVATE\_KEY=081ac2fef5931bf8cb01a5c992836852b31679b6046fa8f41ab847d3fd016a7f

RPC\_URL=https://polygon-amoy.g.alchemy.com/v2/pXsYxdVEMCwq7QJkAnWtT

CONTRACT\_ADDRESS=0x... # Replace with your contract address

Start backend:

node server.js

### 4. Frontend Setup

cd ../frontend

npm install -g serve

serve -s index.html -l 8000

## Complete Package List

### Smart Contract

* hardhat
* @nomicfoundation/hardhat-toolbox
* dotenv

### Backend

* express
* multer
* ethers
* dotenv
* fluent-ffmpeg

### Frontend

* serve

## Usage Instructions

### Registration

1. Open http://localhost:8000
2. Ensure MetaMask is connected to Polygon Amoy
3. In **Register** tab:
   * Enter wallet address
   * Click **Start Recording**
   * Allow microphone access
   * Speak passphrase
   * Click **Stop Recording**
4. Wait for processing & transaction confirmation
5. Success message with transaction hash

### Authentication

1. Switch to **Authenticate** tab
2. Enter same wallet address
3. Record same passphrase
4. Stop recording
5. Wait for verification
6. Success or failure message with transaction hash

### Verify on Blockchain

* Visit [Polygon Amoy Explorer](https://amoy.polygonscan.com/)
* Enter wallet address or transaction hash

## Project Structure

voice-auth-system/

├── backend/

│ ├── node\_modules/

│ ├── uploads/

│ ├── .env

│ ├── audio-processor.js

│ ├── package.json

│ ├── server.js

│ └── Dockerfile

├── frontend/

│ ├── index.html

│ ├── script.js

│ └── style.css

├── smart-contract/

│ ├── .env

│ ├── artifacts/

│ ├── cache/

│ ├── contracts/

│ │ └── VoiceAuth.sol

│ ├── hardhat.config.js

│ ├── node\_modules/

│ ├── package.json

│ └── scripts/

│ └── deploy.js

└── README.md

## Troubleshooting

1. **FFmpeg Not Found**: Ensure installed and PATH set
2. **Contract Deployment Fails**: Check MATIC balance, RPC URL, private key
3. **Backend Fails to Start**: Install dependencies, check .env, port availability
4. **Frontend Issues**: Ensure files exist, use -s index.html, check port
5. **Mic Access Denied**: Use HTTPS/localhost, check browser permissions

Debugging:

* Check browser console
* Check backend terminal
* Verify on Polygonscan
* Ensure MetaMask network

## Future Enhancements

1. Enhanced Voice Biometrics
2. Security Improvements
3. User Experience Enhancements
4. Production Considerations

## License

This project is for educational purposes only.