

# Voting App with Node.js

A secure voting application built with Node.js that allows users to vote for candidates while ensuring one-time voting through role-based access control.

## Features

- **User Authentication:** Secure sign-up and login with JWT tokens
- **Role-Based Access Control:** Separate permissions for voters and admins
- **One-Time Voting:** Prevents multiple votes per user
- **Candidate Management:** Admins can add, update, and remove candidates
- **Live Vote Counts:** Real-time display of voting results
- **Profile Management:** Users can view and update their profiles

## Technologies Used

- Node.js
- Express.js
- MongoDB (or your preferred database)
- JWT for authentication
- bcrypt for password hashing

## Installation

1. Clone the repository:

```
git clone <repository-url>
cd voting-app-with-nodejs
```

2. Install dependencies:

```
npm install
```

3. Set up your environment variables (create a `.env` file):

PORT=3000

MONGODB\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_jwt\_secret

#### 4. Start the server:

```
npm start
```

## Usage

### User Authentication

- **Sign Up:** POST /api/v1/signup
  - Body: { "aadhaar": "user\_aadhaar", "password": "user\_password" }
- **Login:** POST /api/v1/login
  - Body: { "aadhaar": "user\_aadhaar", "password": "user\_password" }
  - Returns: JWT token

### User Profile

- **Get Profile:** GET /api/v1/profile (Auth required)
- **Change Password:** PUT /api/v1/profile/password (Auth required)
  - Body: { "newPassword": "new\_password" }

### Voting

- **Get Candidates:** GET /api/v1/candidates
- **Vote for Candidate:** POST /api/v1/vote/:candidateId (Auth required, Voter role)
- **Get Vote Counts:** GET /api/v1/vote/counts

### Admin Management (Admin role required)

- **Add Candidate:** POST /api/v1/candidates
  - Body: { "name": "candidate\_name", "party": "party\_details" }
- **Update Candidate:** PUT /api/v1/candidates/:candidateId
  - Body: { "name": "updated\_name", "party": "updated\_party" }
- **Delete Candidate:** DELETE /api/v1/candidates/:candidateId

# Data Models

## User

- Name
- Aadhaar Card Number (unique identifier)
- Password (hashed)
- Role (voter/admin)
- Voted Status (boolean)

## Candidate

- Name
- Party/Details
- Vote Count

## Middleware

- **Authentication:** Verifies JWT tokens for protected routes
- **Authorization:** Checks user roles for access control
- **Single Vote Check:** Ensures users can only vote once

## Contributing

1. Fork the repository
2. Create a feature branch
3. Commit your changes
4. Push to the branch
5. Open a Pull Request

## License

This project is licensed under the ISC License.