

Everyone Votes - Online Voting System

Project Report

Team Name: CivicCoders

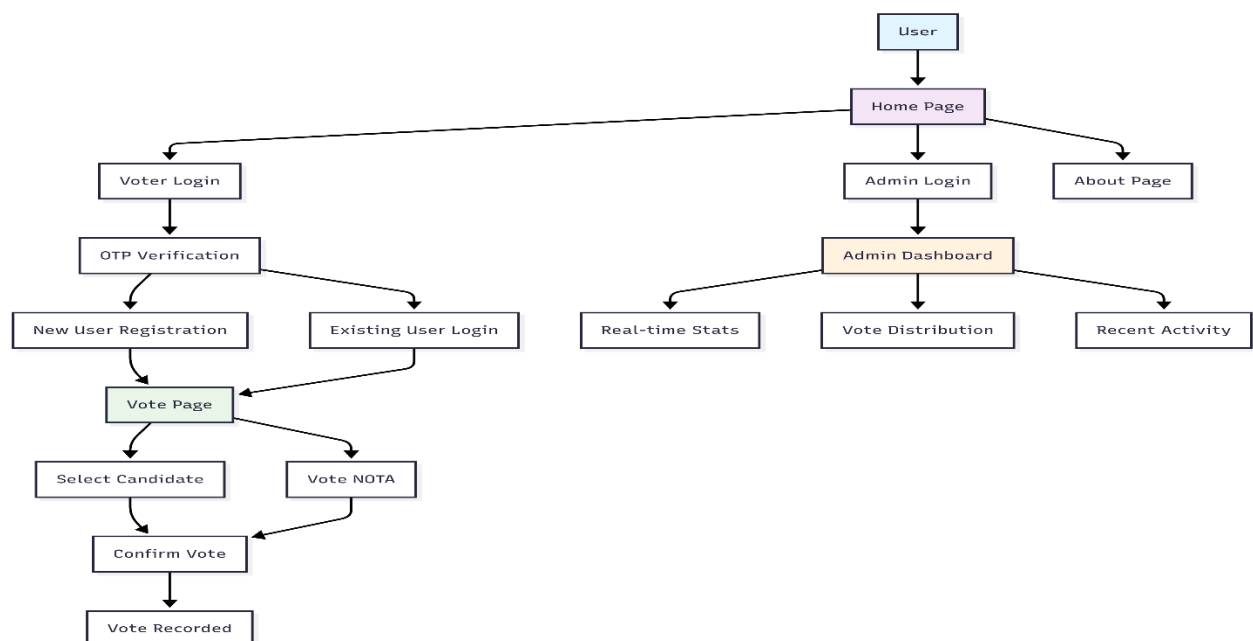
Front End By: Satwik Pandey

Back End By: Satwik Pandey

Date: July 16, 2025

OVERVIEW

High Level Summary (Executive Summary)



Everyone Votes is a modern, secure online voting platform designed to revolutionize how Indians participate in democratic processes. Think of it as a digital voting booth that citizens can access from anywhere using their smartphones or computers.

How it works in simple terms:

1. **Voter Experience:** A citizen visits the website, enters their mobile number, receives an OTP (One-Time Password), and then either logs in (if they're a returning user) or registers with their Aadhar and Voter ID details

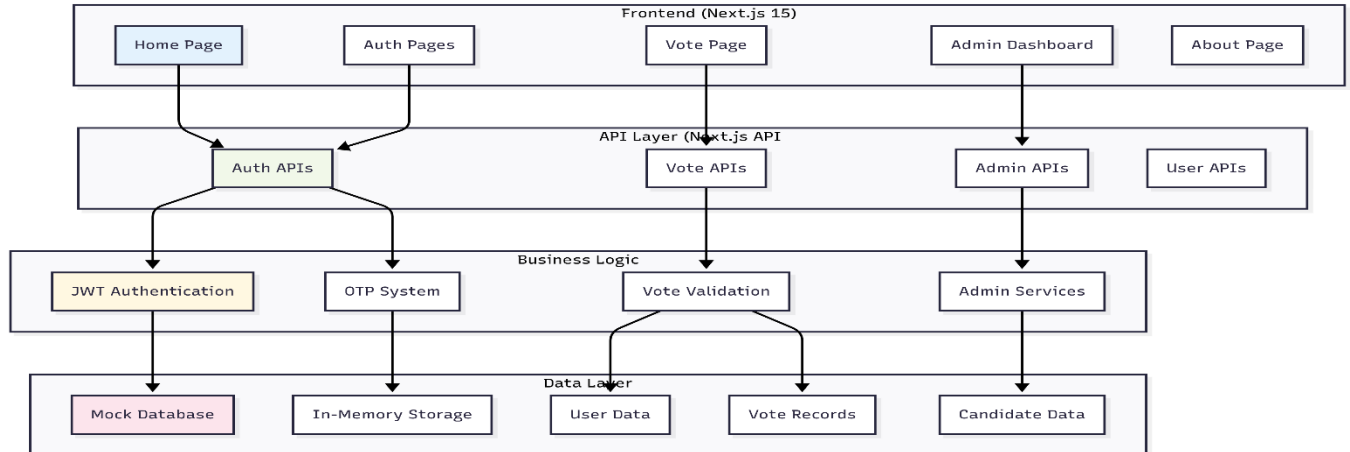
2. **Secure Voting:** Once authenticated, they can view all candidates in their constituency and cast their vote - including an option for "NOTA" (None of the Above)
3. **One Vote Policy:** The system ensures each person can only vote once, preventing duplicate voting
4. **Real-time Monitoring:** Election officials can monitor live voting statistics, see vote distribution, and track recent voting activity through an admin dashboard

Key Benefits:

- **Accessibility:** Vote from anywhere, eliminating travel barriers
- **Security:** Multi-layer authentication with OTP + Aadhar + Voter ID verification
- **Transparency:** Real-time vote counting and statistics
- **Efficiency:** Instant result compilation and reporting

DESIGN DETAILS

High Level Design Considerations



1. Security-First Architecture

- **Multi-Factor Authentication:** OTP verification combined with Aadhar and Voter ID validation
- **JWT Token Security:** Encrypted session management with time-based expiration
- **Input Validation:** Comprehensive data sanitization and format validation
- **One-Vote Enforcement:** Database-level constraints preventing duplicate voting

2. Scalability & Performance

- **Modern Tech Stack:** Next.js 15 with server-side rendering for optimal performance
- **Mock Database:** In-memory storage for demo purposes (easily replaceable with real database)
- **API-First Design:** RESTful APIs enabling future mobile app integration
- **Responsive Design:** Mobile-first approach for smartphone users

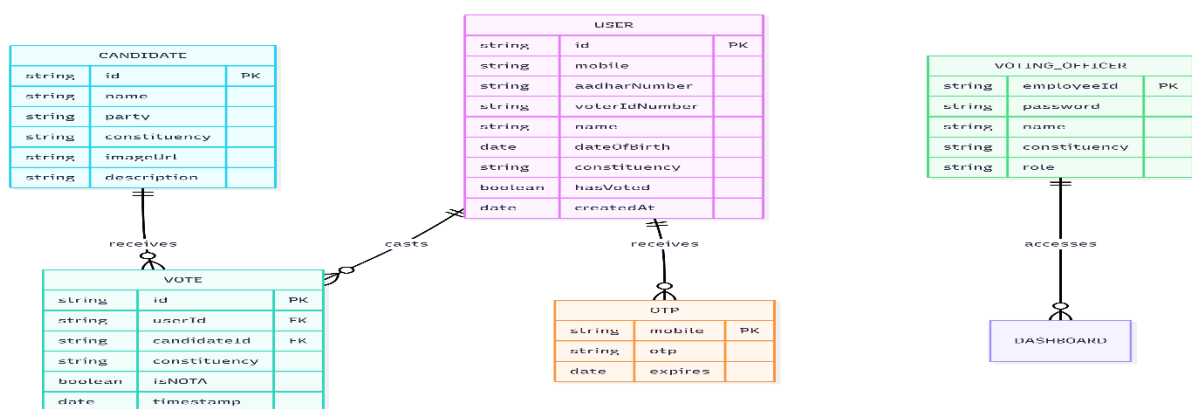
3. User Experience Focus

- **Intuitive Navigation:** Clear breadcrumb navigation and logical flow
- **Accessibility:** High contrast colors, readable fonts, and screen reader compatibility
- **Progressive Enhancement:** Works on all devices and browsers
- **Real-time Feedback:** Instant validation and status updates

4. Administrative Control

- **Real-time Dashboard:** Live monitoring of voting statistics
- **Constituency Management:** Filter data by electoral regions
- **Activity Tracking:** Comprehensive audit trail of all voting activities
- **Role-based Access:** Separate interfaces for voters and election officials

Low Level Components



UI Components Layer

1. Authentication Components

- **LoginPage:** Mobile number input with OTP generation

- RegisterPage: Multi-step registration with Aadhar/Voter ID validation
- OTPVerification: Secure OTP input with expiration timer

2. Voting Components

- VotePage: Candidate selection interface with NOTA option
- CandidateCard: Individual candidate display with party information
- VoteConfirmation: Final vote confirmation dialog

3. Admin Components

- AdminDashboard: Real-time statistics and analytics
- VoteDistribution: Live vote counting charts
- RecentActivity: Timeline of voting activities

4. Navigation Components

- Navbar: Top navigation with clickable logo
- Breadcrumbs: Contextual navigation breadcrumbs
- Footer: Site information and links

Web Service/Backend Components

1. Authentication APIs

- POST /api/auth/send-otp: Generate and send OTP to mobile number
- POST /api/auth/verify-otp: Validate OTP and initiate session
- POST /api/auth/register: Register new voter with document verification

2. Voting APIs

- GET /api/candidates: Retrieve candidates by constituency
- POST /api/vote: Submit vote with validation
- GET /api/user/profile: Get current user information

3. Admin APIs

- POST /api/admin/login: Officer authentication
- GET /api/admin/dashboard: Real-time voting statistics

4. Utility APIs

- `POST /api/seed`: Initialize sample data for testing

Database Layer

Mock Database Implementation:

- **In-Memory Storage**: JavaScript objects storing all data
- **Data Models**: User, Vote, Candidate, OTP, VotingOfficer
- **Relationships**: Foreign key constraints and data integrity
- **CRUD Operations**: Create, Read, Update, Delete functionality

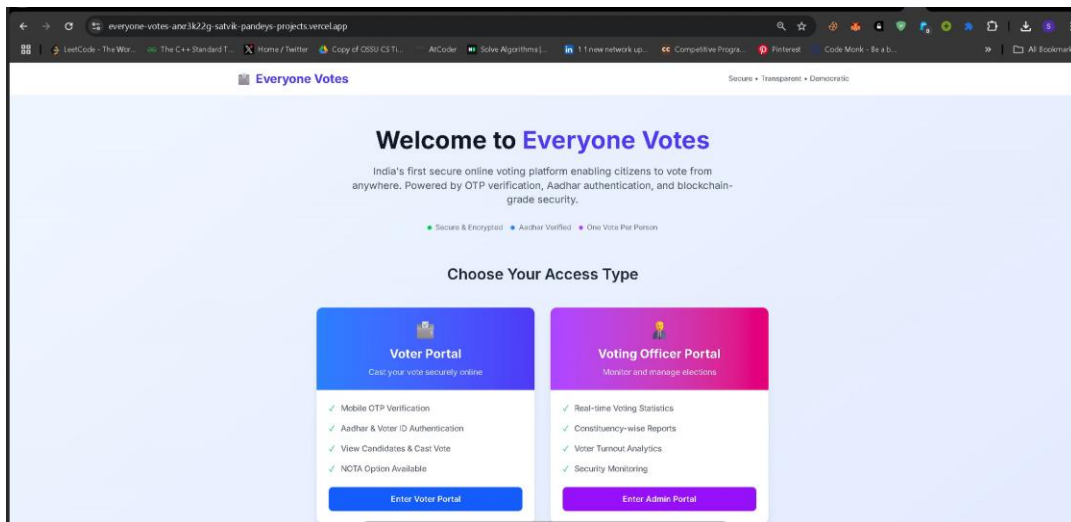
Data Structure:

```
interface User {  
  id: string;  
  mobile: string;  
  aadharNumber: string;  
  voterIdNumber: string;  
  name: string;  
  dateOfBirth: Date;  
  constituency: string;  
  hasVoted: boolean;  
  createdAt: Date;  
}
```

```
interface Vote {  
  id: string;  
  userId: string;  
  candidateId: string;  
  constituency: string;  
  isNOTA: boolean;  
  timestamp: Date;  
}
```

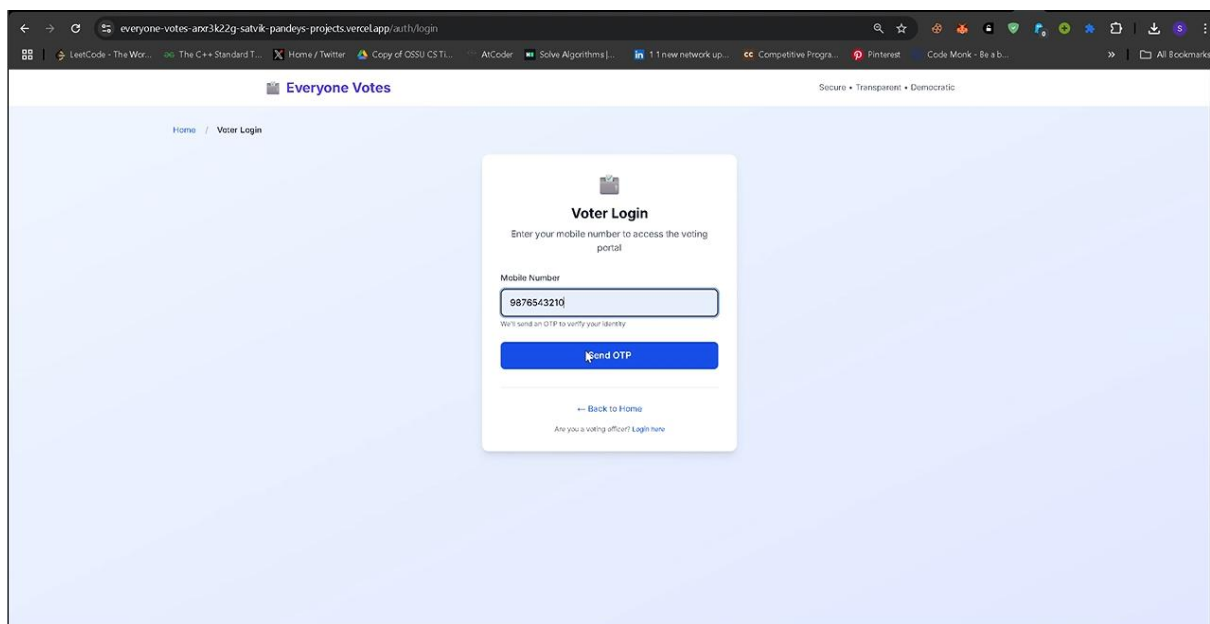
WORKING APPLICATION

UI Screenshots



1. Home Page

- **Features:** Clean landing page with navigation to voter and admin portals
- **Components:** Hero section, feature highlights, call-to-action buttons
- **Navigation:** Breadcrumb: Home



2. Voter Login Page

- **Features:** Mobile number input, OTP generation, security messaging
- **Components:** Form validation, loading states, error handling
- **Navigation:** Breadcrumb: Home → Login

everyone-votes-anr3k22g-satvik-pandey's-projects.vercel.app/auth/register/mobile=7390017863

Everyone Votes

Secure • Transparent • Democratic

Home / Login / Register

Voter Registration

Complete your registration to access the voting portal

Mobile Number *

7390017863

Aadhar Number *

Enter 12-digit Aadhar number

Voter ID Number *

Enter Voter ID (e.g., ABC1234567)

Format: 3 letters followed by 7 digits

Full Name *

Enter your full name as per Aadhar

Date of Birth *

dd-mm-yyyy

Constituency *

Select your constituency

3. Registration Page

- **Features:** Multi-field form with Aadhar/Voter ID validation
- **Components:** Date picker, dropdown selectors, form validation
- **Navigation:** Breadcrumb: Home → Login → Register

everyone-votes-anr3k22g-satvik-pandey's-projects.vercel.app/vote

- Select one candidate from the list below or choose NOTA
- Review your selection carefully before confirming
- You can vote only once - your choice cannot be changed
- Your vote is completely anonymous and secure

Select Your Candidate

☐ **Rajesh Kumar**
Indian National Congress
Experienced leader focused on urban development and infrastructure.

☐ **Priya Sharma**
Bharatiya Janata Party
Young politician advocating for digital India and entrepreneurship.

☐ **Mohammed Ali**
Aam Aadmi Party
Community leader working for transparent governance.

☐ **NOTA**
None of the Above
Choose this option if you don't want to vote for any candidate

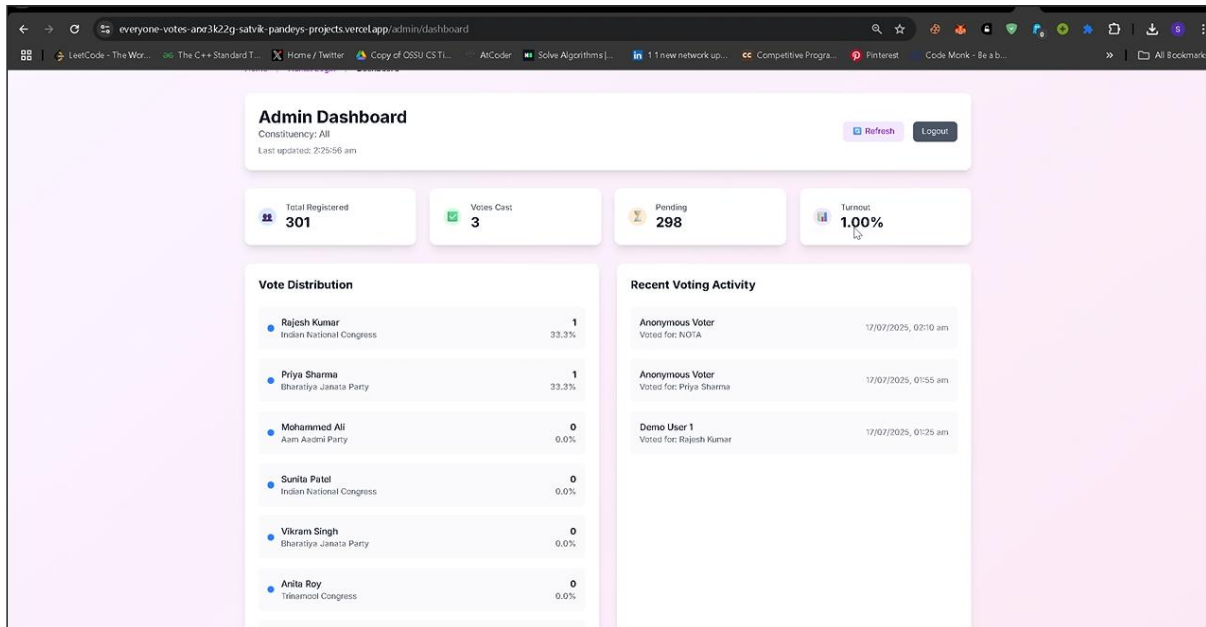
Cast Your Vote

Once you click this button, your vote will be recorded and cannot be changed

4. Vote Page

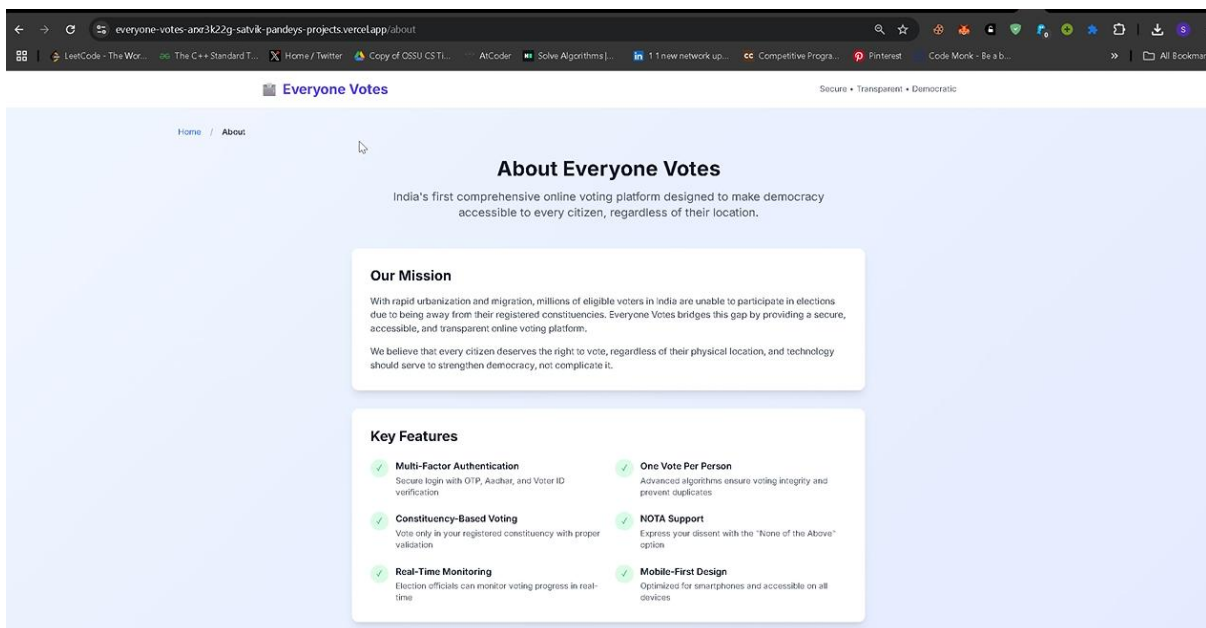
- **Features:** Candidate listing, NOTA option, vote confirmation
- **Components:** Candidate cards, radio selection, submit button

- **Navigation:** Breadcrumb: Home → Vote



5. Admin Dashboard

- **Features:** Real-time statistics, vote distribution charts, recent activity
- **Components:** Statistics cards, data tables, refresh functionality
- **Navigation:** Breadcrumb: Home → Admin → Dashboard



6. About Page

- **Features:** System information, security details, contact information
- **Components:** Feature grid, security badges, contact forms
- **Navigation:** Breadcrumb: Home → About

API Endpoints - Request/Response Examples

Authentication Endpoints

1. Send OTP

POST /api/auth/send-otp

Content-Type: application/json

```
{  
  "mobile": "9876543210"  
}
```

Response (200):

```
{  
  "message": "OTP sent successfully",  
  "expiresAt": "2025-07-16T19:30:00.000Z"  
}
```

2. Verify OTP

POST /api/auth/verify-otp

Content-Type: application/json

```
{  
  "mobile": "9876543210",  
  "otp": "123456"  
}
```

Response (200):

```
{  
  "message": "OTP verified successfully",  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "isNewUser": false,  
  "user": {
```

```
"id": "user123",
"name": "John Doe",
"mobile": "9876543210",
"constituency": "Mumbai North",
"hasVoted": false
}
}
```

3. Register User

POST /api/auth/register

Content-Type: application/json

Authorization: Bearer <token>

```
{
  "mobile": "9876543210",
  "aadharNumber": "123456789012",
  "voterIdNumber": "ABC1234567",
  "fullName": "John Doe",
  "dateOfBirth": "1990-01-01",
  "constituency": "Mumbai North"
}
```

Response (201):

```
{
  "message": "Registration successful",
  "user": {
    "id": "user123",
    "name": "John Doe",
    "mobile": "9876543210",
    "constituency": "Mumbai North",
    "hasVoted": false
  }
}
```

Voting Endpoints

4. Get Candidates

GET /api/candidates?constituency=Mumbai%20North

Authorization: Bearer <token>

Response (200):

```
{
  "candidates": [
    {
      "id": "candidate1",
      "name": "Candidate A",
      "party": "Party X",
      "constituency": "Mumbai North",
      "imageUrl": "/images/candidate1.jpg",
      "description": "Experienced leader..."
    }
  ]
}
```

5. Submit Vote

POST /api/vote

Content-Type: application/json

Authorization: Bearer <token>

```
{
  "candidateId": "candidate1",
  "isNOTA": false
}
```

Response (200):

```
{
  "message": "Vote recorded successfully",
  "voteId": "vote123",
  "candidateName": "Candidate A",
  "timestamp": "2025-07-16T19:45:00.000Z"
}
```

Admin Endpoints

6. Admin Login

POST /api/admin/login

Content-Type: application/json

```
{
  "employeeId": "VO001",
  "password": "admin123"
}
```

Response (200):

```
{
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "officer": {
    "employeeId": "VO001",
    "name": "Officer Name",
    "constituency": "Mumbai North",
    "role": "Voting Officer"
  }
}
```

7. Dashboard Statistics

GET /api/admin/dashboard?constituency=Mumbai%20North

Authorization: Bearer <token>

Response (200):

```
{
  "totalRegistered": 1250,
  "totalVoted": 875,
  "totalPending": 375,
  "voteDistribution": [
    {
      "candidateId": "candidate1",
      "candidateName": "Candidate A",
      "candidateParty": "Party X",
      "votes": 450,
      "isNOTA": false
    },
    {
      "candidateId": "NOTA",
      "candidateName": "NOTA",
      "candidateParty": "None",
      "votes": 25,
      "isNOTA": true
    }
  ],
  "recentActivity": [
    {
      "voterName": "John Doe",
      "candidateName": "Candidate A",
      "timestamp": "2025-07-16T19:45:00.000Z",
      "isNOTA": false
    }
  ]
}
```

```
}  
]  
}
```

ASSUMPTIONS

Technical Assumptions

1. **Demo Environment:** The system uses mock data and in-memory storage for demonstration purposes
2. **OTP Service:** OTP generation is simulated; in production, this would integrate with SMS gateway services
3. **Document Verification:** Aadhar and Voter ID validation is format-based; real verification would require government API integration
4. **Network Connectivity:** Users have stable internet connection for real-time features

Security Assumptions

1. **JWT Security:** Hardcoded JWT secret is acceptable for demo; production would use environment variables
2. **Data Encryption:** Basic validation is implemented; production would require advanced encryption
3. **Session Management:** Browser-based token storage is used; production might implement more secure methods

Business Assumptions

1. **Constituency Data:** Sample constituencies are provided; real implementation would sync with Election Commission data
2. **Candidate Information:** Mock candidate data is used; production would integrate with official candidate databases
3. **User Registration:** Self-registration is allowed; production might require pre-verification by election officials
4. **Vote Validation:** Single vote per user is enforced; production would require additional fraud prevention measures

Operational Assumptions

1. **Admin Access:** Demo admin credentials are provided; production would have proper role-based access control
2. **Data Persistence:** In-memory storage is used; production would require database backup and recovery systems
3. **System Monitoring:** Basic error handling is implemented; production would require comprehensive monitoring and alerting
4. **Load Handling:** System is designed for demo load; production would require load balancing and scaling considerations

TECHNOLOGY STACK

Frontend

- **Framework:** Next.js 15 with App Router
- **Language:** TypeScript
- **Styling:** Tailwind CSS
- **State Management:** React Hooks (useState, useEffect)
- **Navigation:** Next.js Navigation API

Backend

- **Runtime:** Node.js
- **Framework:** Next.js API Routes
- **Authentication:** JWT (JSON Web Tokens)
- **Password Hashing:** bcryptjs
- **Data Storage:** In-memory mock database

Development Tools

- **Type Checking:** TypeScript
- **Code Quality:** ESLint
- **Build Tool:** Next.js Build System
- **Package Manager:** npm

CONCLUSION

The **Everyone Votes** system demonstrates a complete, secure, and user-friendly online voting platform. It addresses key challenges in traditional voting systems while maintaining the highest security standards. The modular architecture allows for easy scaling and feature additions, making it suitable for various democratic processes from local elections to national polls.

The system successfully balances security, usability, and administrative control, providing a solid foundation for modern digital democracy initiatives.

This report demonstrates the complete implementation of a secure online voting system with comprehensive features for both voters and election officials.

I had also implemented it using blockchain based authentication in the past and presented it in a hackathon too:

<https://github.com/worldisconfusion/Voting-App>