

CODE CRAFTERS: Everyone Votes

“**Everyone Votes**” is an Online Voting system that leverages the internet and digital platforms to allow voters to cast their ballots remotely, providing convenience and accessibility.

Here's a high-level overview of how an online voting system typically works:

- **Choose Profile:**

The Home page shows two different profiles: **Voter** and **Voting Officer**. Upon selecting the profile, the user will be redirected to the respective page with the use of ‘**useNavigate**’ hooks.

- **Voter/ Voting Officer Login and Authentication:**

1. In this form, the voter has to provide the VoterId after which the ‘Send OTP’ button will be enabled.
2. For voter authentication, the voter is required to generate OTP by clicking on ‘Send OTP’ on which an HTTP POST request is sent with the “**api/users/login**” route with voterId in the request body.
3. In this api call, the voter id is validated to check if the user is eligible to vote and has not already voted.
 - a. If the voterId is valid, then an OTP is sent to the corresponding registered email id.
 - b. Otherwise, an error message is sent to the UI to show that either the voter has already voted or the voterId is invalid.
4. After receiving the OTP user has to enter the received OTP in the “Enter OTP” field, the ‘Check’ button will be enabled.
5. For OTP Verification, voters will click on the “Check” button and a HTTP POST request is sent with “voterID” and “OTP”.
 - a. If the entered OTP is incorrect, the voter will see an error message on screen “Invalid OTP”.
 - b. If OTP is valid, the voter is redirected to the candidate page.

The same flow is to be followed for the login and authentication of the voting officer.

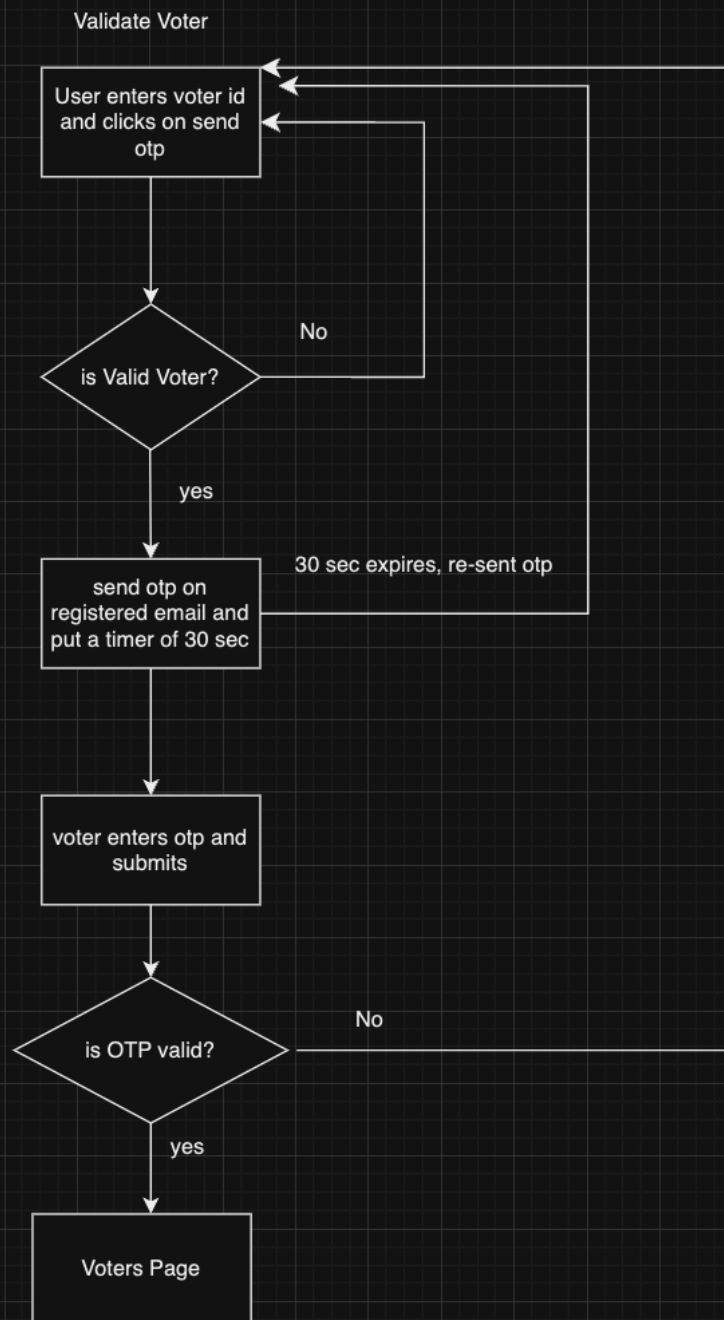
But after OTP verification, the voting officer will be redirected to the “**Area Wise Voting Page**”.

To protect the routes, after login, the user will be marked as ‘loggedIn’, and therefore only then he can go to the ‘votePage’ or ‘area wise voting page’ according to the user identification as a voter or voting officer.

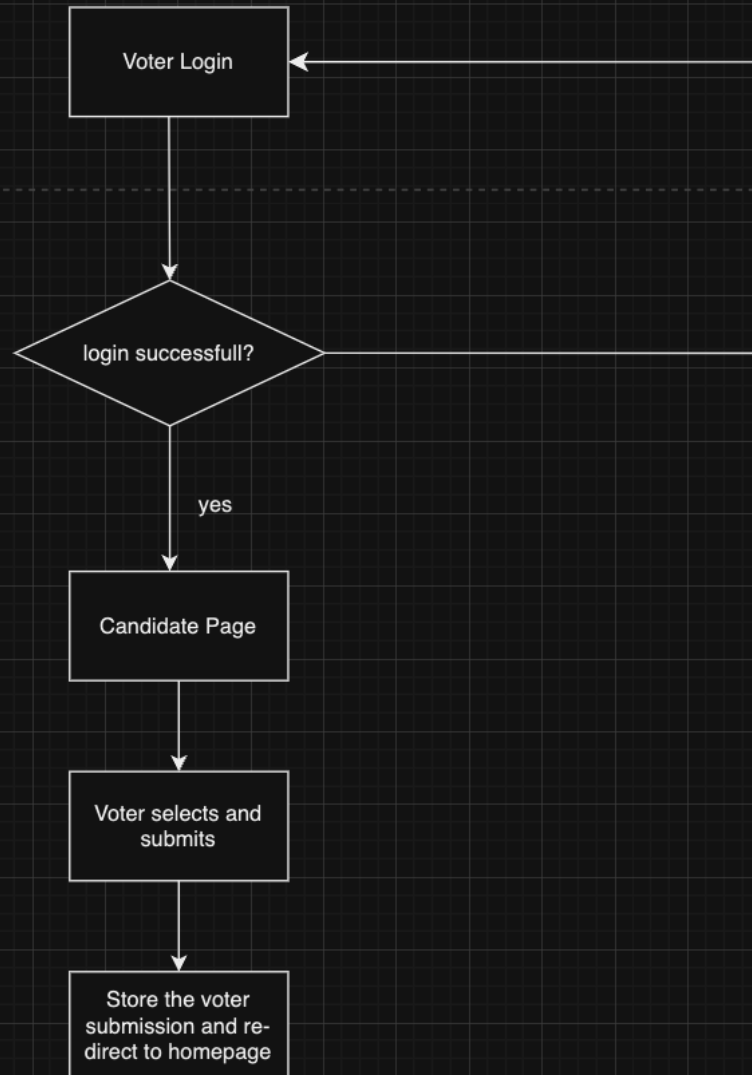
- **Place the Vote:**

1. On the 'votePage' multiple candidates will be shown in the form of cards with their name and vote button. This data is fetched from HTTP GET request "**/api/candidate/get**".
 2. After clicking on the vote button, the 'onClick' event handler will call the increment function where an HTTP POST request is made to "**/api/vote**" route which sets the status of 'has voted' for voterID in the DB of that particular voterId as YES.
- **Final page:**
After voting, the voter will be redirected to the page for a successful vote and a button for going back to the home page and the state of loggedIn will to false.
 - **Area-wise voting page:**
On this page, the officer will select the required constituency from the dropdown menu and then three cards will be shown representing the count of voters registered in that area, the count of voters who have voted, and the count of pending voters.

Voter Authentication Flow



Complete Voter Flow



API Endpoint Details

Send OTP Endpoint

Api Endpoint	https://localhost:8080/api/users/login
Method	POST
Headers	{Content-Type: application/json}
Body	{ "aadharNumber": "adfmsd" }
Response	{ "validVoter": false, "message": "Invalid Voter Id" "voter has already voter" }

Validate OTP Endpoint

Api Endpoint	https://localhost:8080/api/users/verify
Method	POST
Headers	{Content-Type: application/json}
Body	{ "aadharNumber": "adfmsd", "Otp": "4324" }
Response	{ "validVoterOTP": true }

Get Candidates Endpoint

Api Endpoint	https://localhost:8080/api/candidates/get
Method	GET
Response	{ "_id": "", }

	<pre> "candidateName": "" } </pre>
--	------------------------------------

Vote Endpoint

Api Endpoint	https://localhost:8080/api/vote
Method	POST
Headers	Content-Type: application/json}
Body	<pre> { "voterID": "adfmsd", "candidateId": "4324" } </pre>
Response	200 status code