Table of contents

About the Project

Problem:

In a democratic system, ensuring transparent, secure, and tamper-proof elections is critical. Traditional voting methods, whether paper-based or electronic, face challenges such as voter fraud, tampering, and a lack of transparency. The problem is further compounded by the difficulty of maintaining voter confidentiality while ensuring that only eligible voters participate and their votes are accurately counted.

Key Challenges:

1. **Voter Registration Fraud**: Verifying voter eligibility and maintaining an up-to-date and secure voter list.
2. **Vote Tampering**: Ensuring that votes are not altered or miscounted during transmission or tabulation.
3. **Transparency vs. Privacy**: Balancing the need for election transparency with voter anonymity.
4. **Lack of Trust**: Limited public trust in election processes due to historical malpractices and inefficiencies.
5. **Auditing and Accountability**: Ensuring secure auditing of votes while maintaining voter confidentiality.

Solution:

The proposed Blockchain-Based Voting System addresses these challenges using Hyperledger Fabric, a permissioned blockchain framework. The system involves collaboration between:

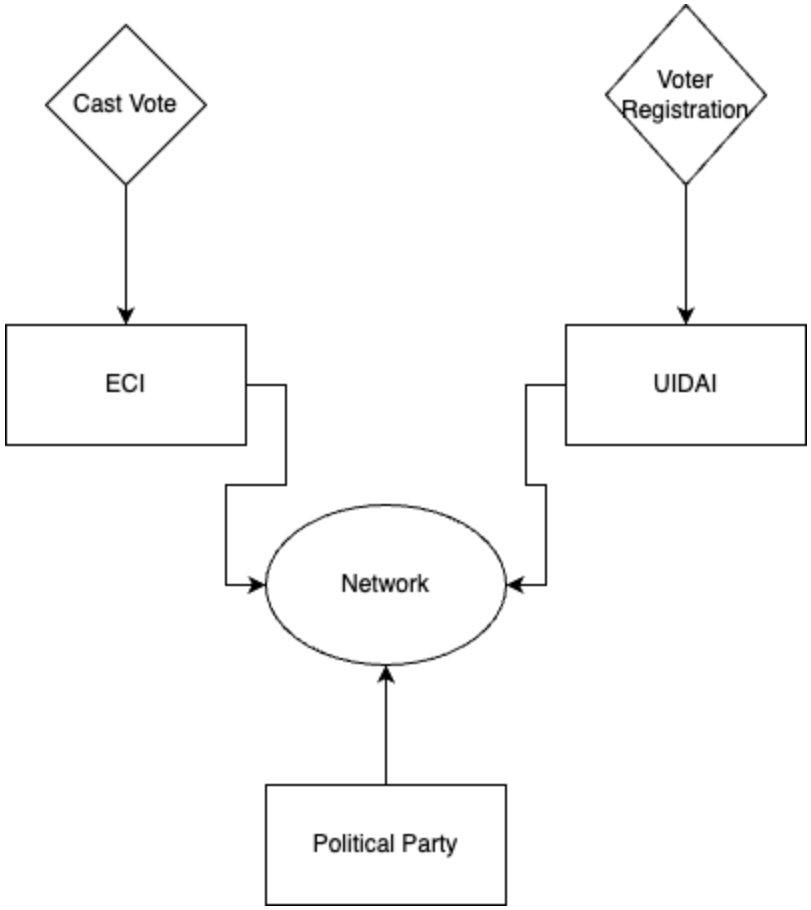
1. Election Commission of India (ECI) – Responsible for overseeing the election process and managing vote casting.
2. Unique Identification Authority of India (UIDAI) – Responsible for securely verifying voter identities and registering eligible voters.
3. Political Parties – Observers of vote aggregation and results verification.

Features of the Solution:

1. Voter Registration: Managed by UIDAI using the Aadhaar database to verify and register eligible voters securely.
2. Vote Casting: Managed by ECI through a secure and transparent blockchain-enabled voting platform.
3. Immutable Ledger: Blockchain ensures that all transactions (voter registration, vote casting) are tamper-proof and auditable.
4. Anonymity: Votes are stored using cryptographic methods to ensure voter privacy.
5. Real-Time Monitoring: Political parties and auditors can monitor aggregated voting data and audit the system in real-time.

Workflow:

1. Voter Creation by UIDAI
2. Caste Vote based on election by ECI
3. Party would be the observer



Running the Application:

Prerequisite:

Docker, docker-compose and fabric binaries should be present and installed on the system.

Installation:

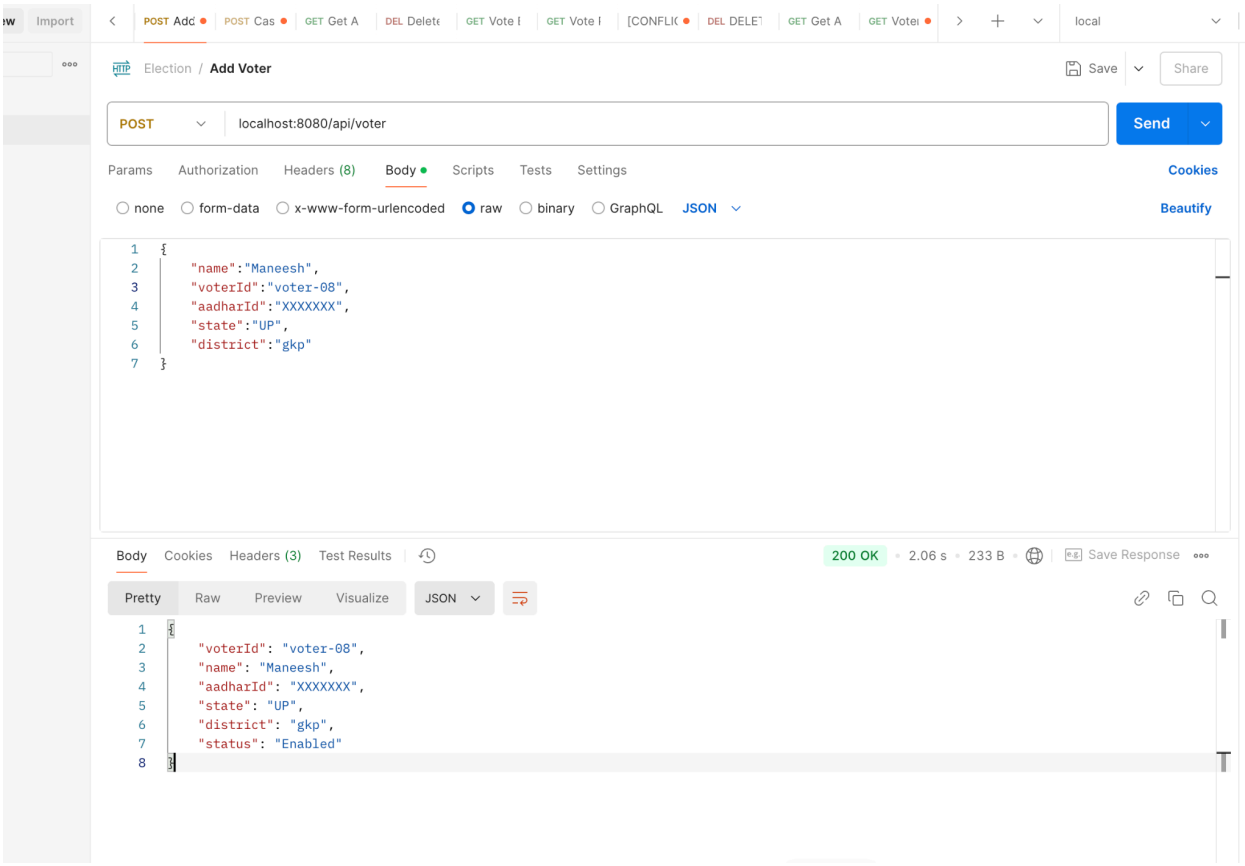
Clone the repo:

git clone https://github.com/KBA-Learning/KBA_CHF_NPCI_B7.git

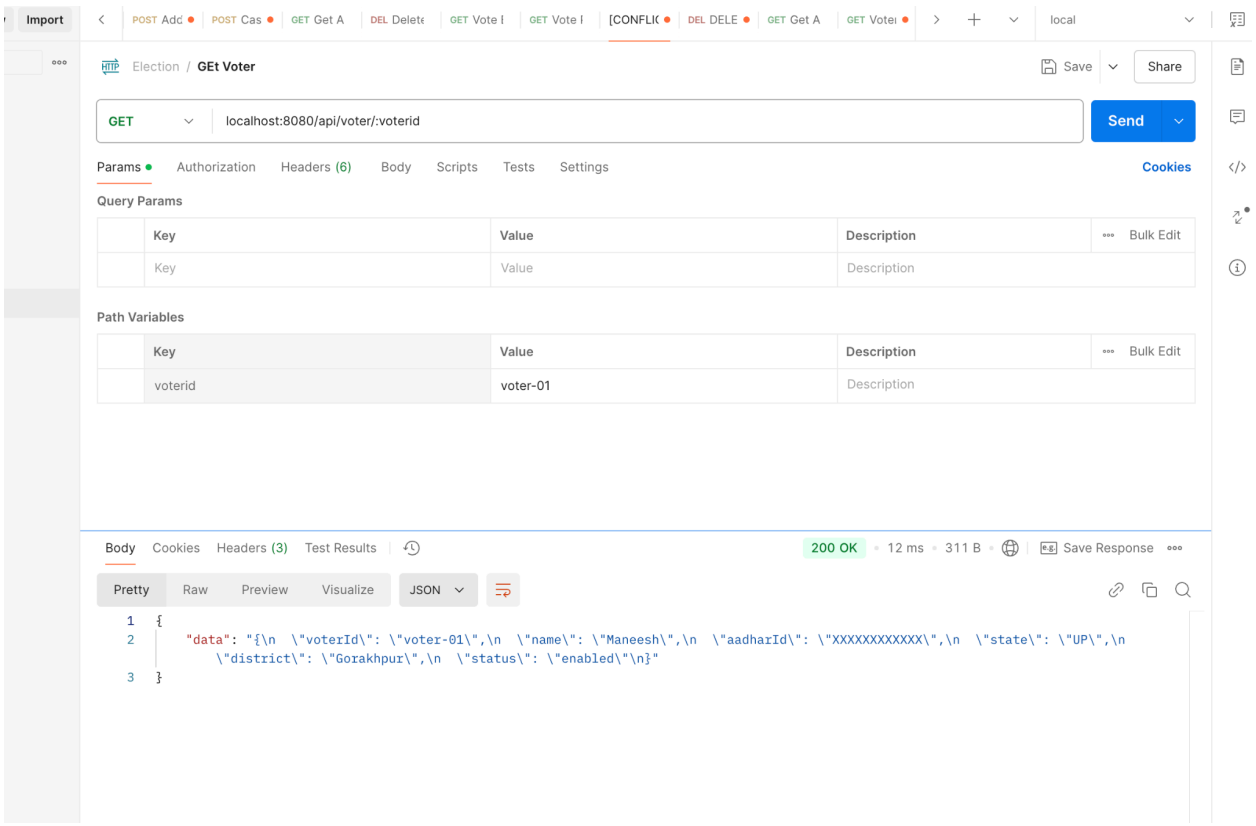
Install the binaries and dependencies and follow the instructions mentioned in readme file

OUTPUTs

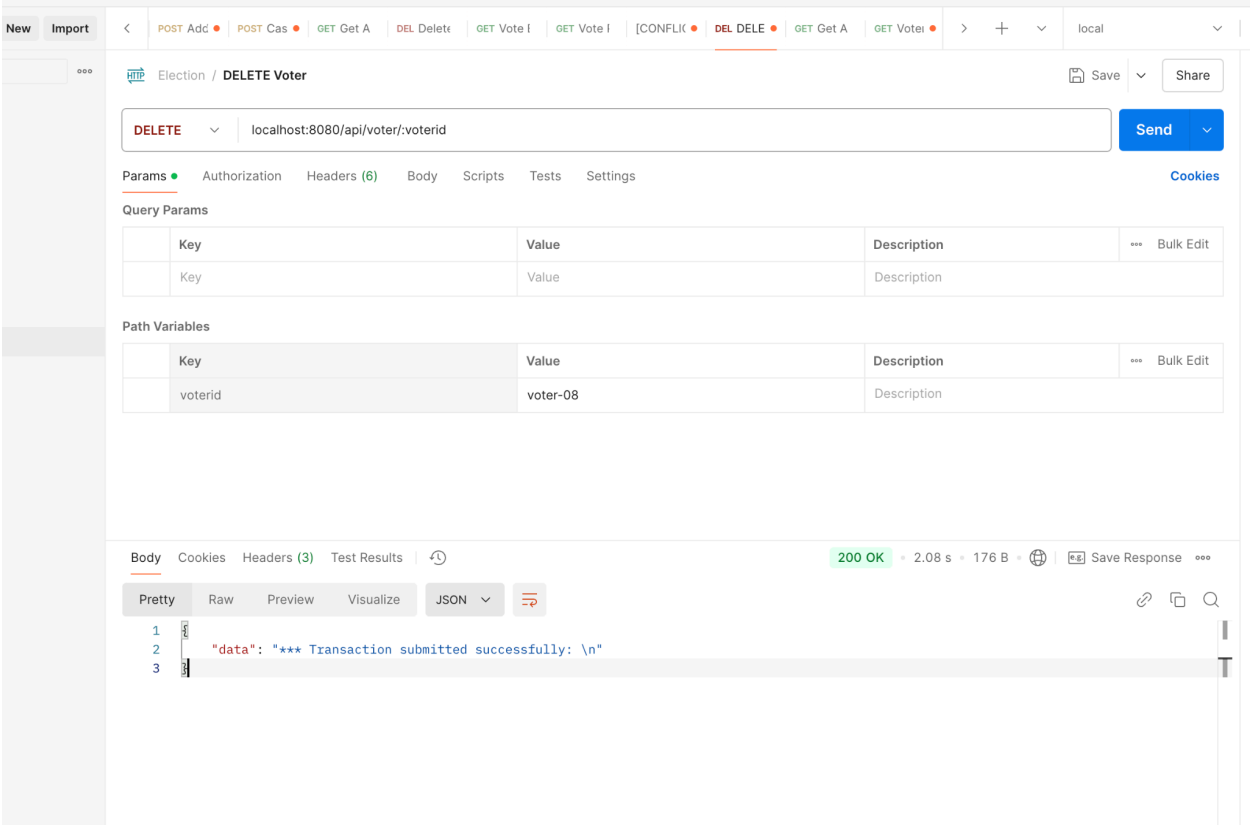
1. Add Voter:



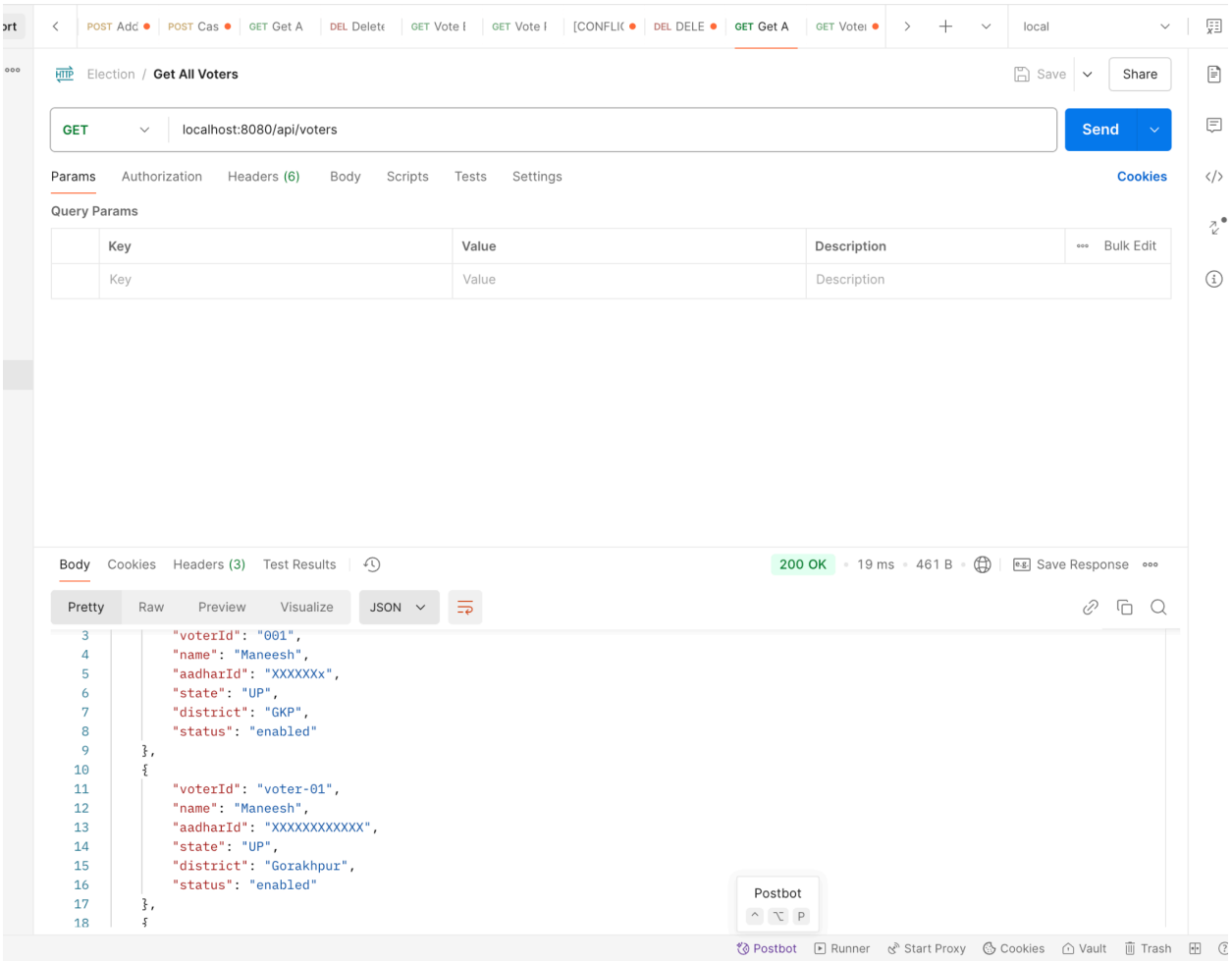
2. Get Voter



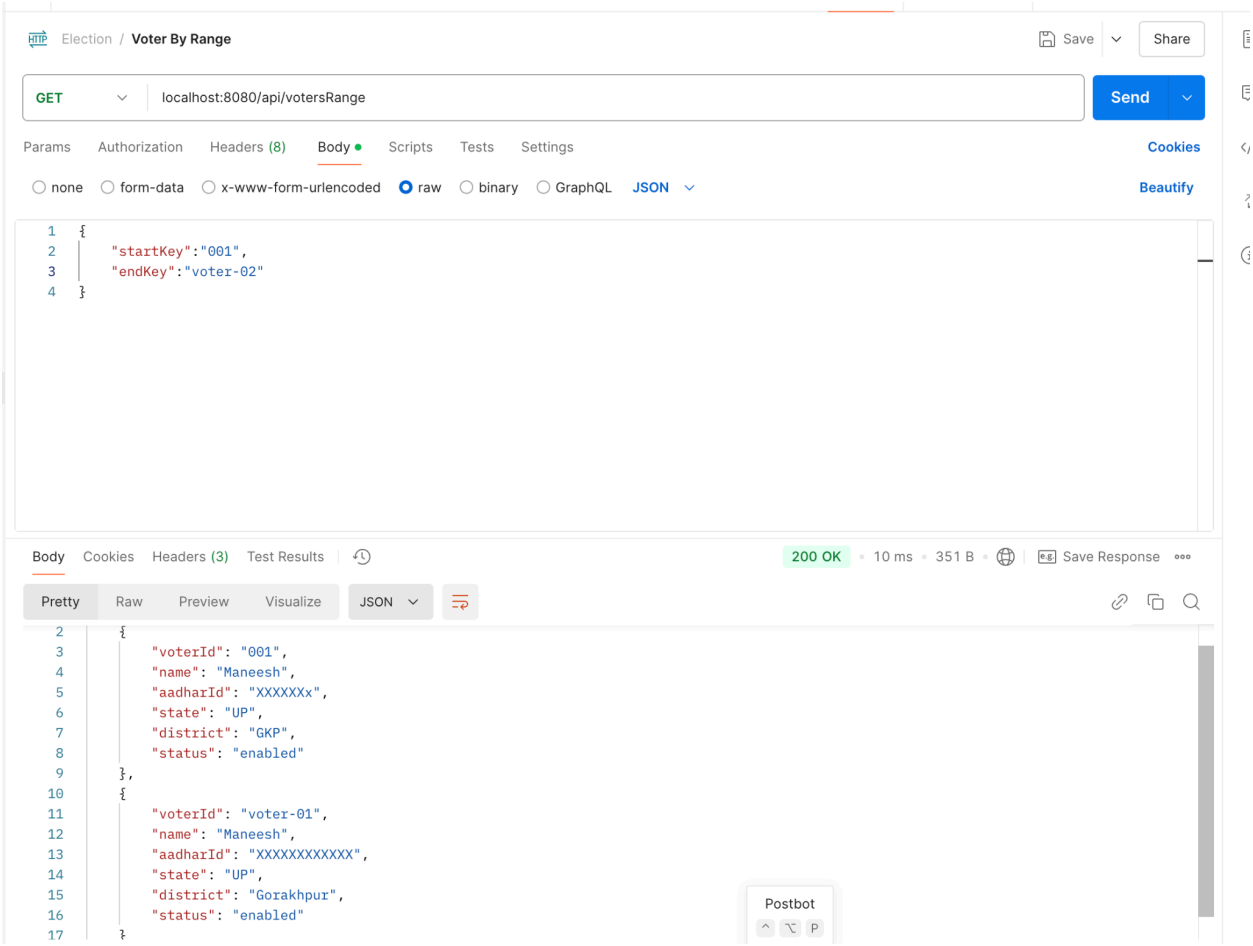
3. Delete Voter



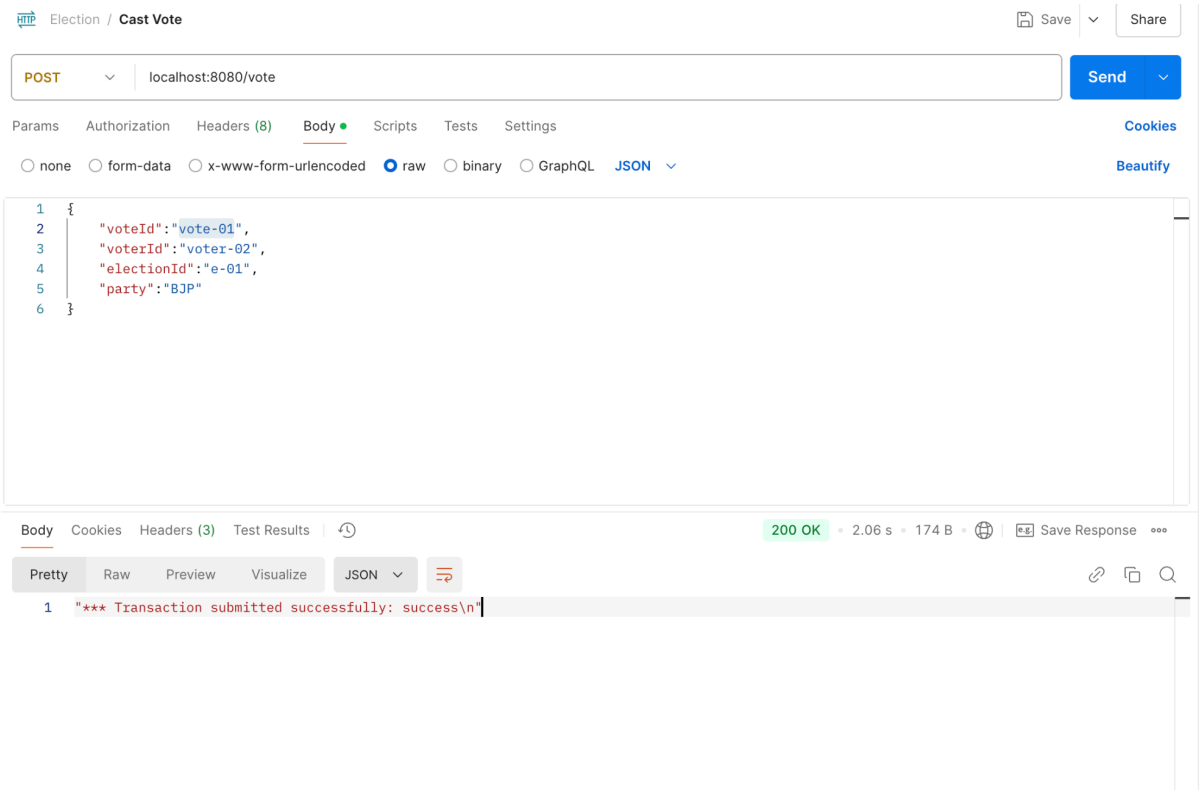
4. Get All voters



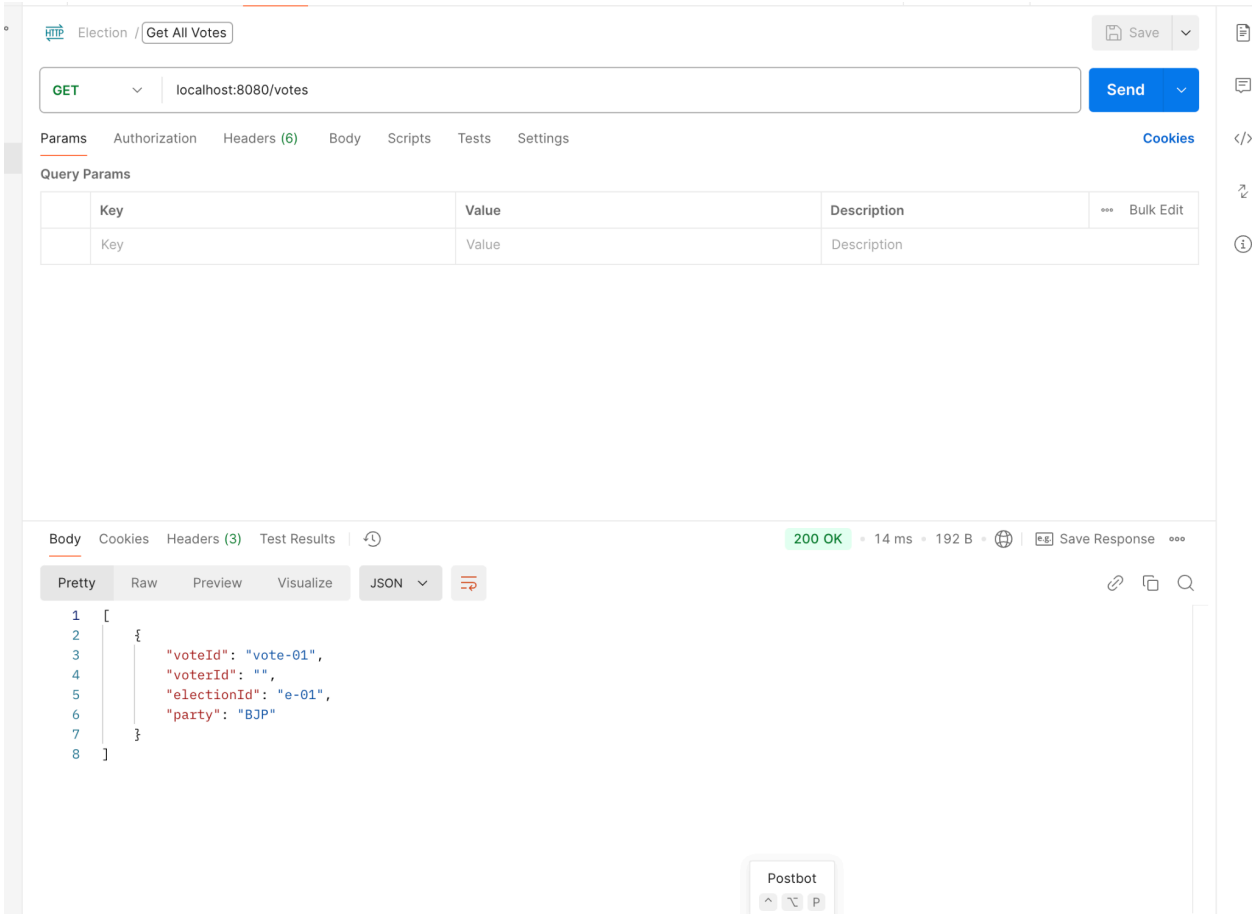
6. Get Voter By range


Election / Voter By Range
Save
Share
GET
localhost:8080/api/votersRange
Send
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"startKey":"001",
"endKey":"voter-02"
}
Body
Cookies
Headers (3)
Test Results
200 OK
10 ms
351 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"voterId": "001",
"name": "Maneesh",
"aadharId": "XXXXXXx",
"state": "UP",
"district": "GKP",
"status": "enabled"
},
{
"voterId": "voter-01",
"name": "Maneesh",
"aadharId": "XXXXXXXXXXX",
"state": "UP",
"district": "Gorakhpur",
"status": "enabled"
}
Postbot

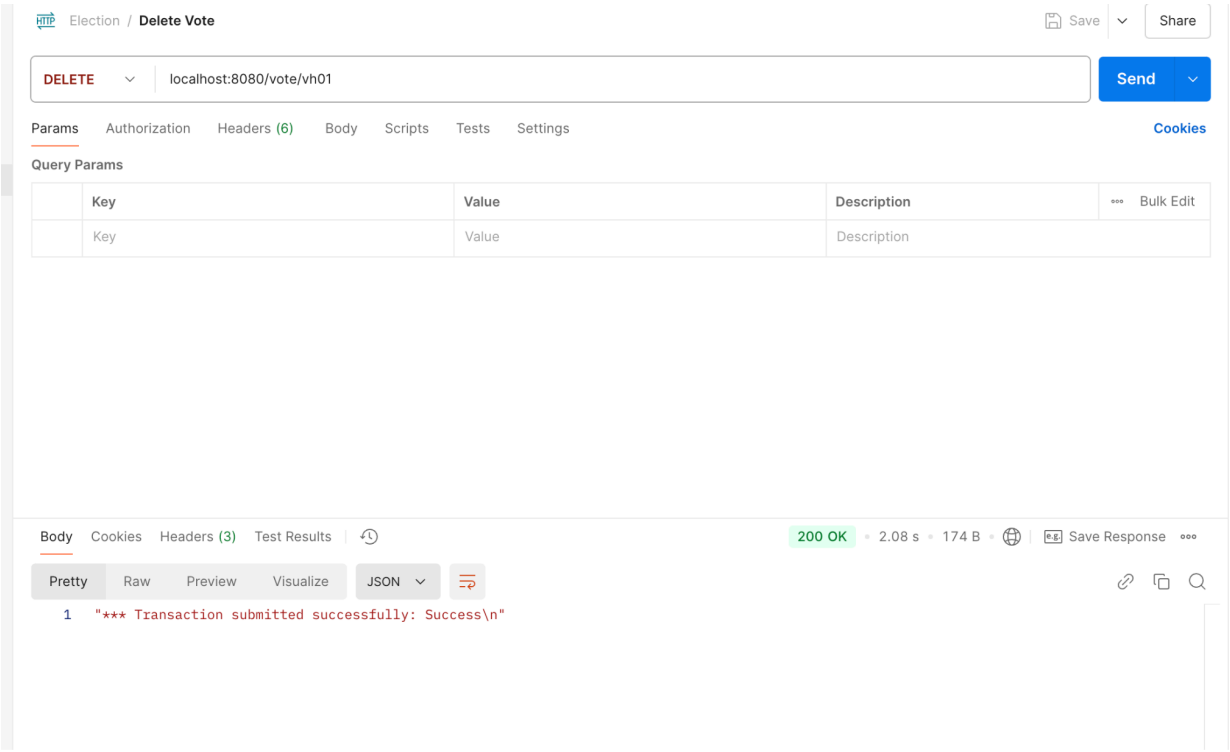
7. Caste Vote



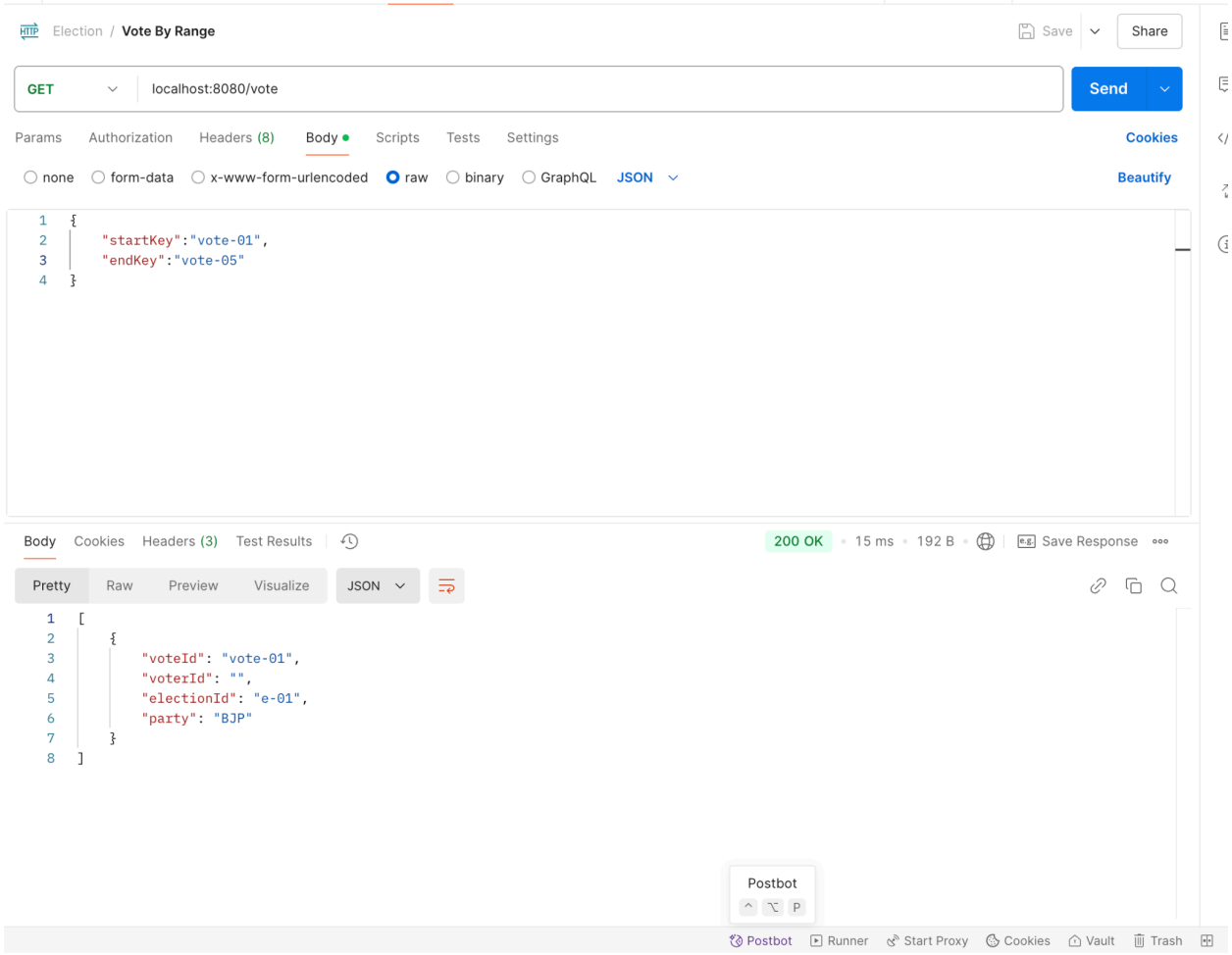
8.Get all votes



9. Delete Vote



10. Vote by Range



Chaincode functions:

1. **CastVote:** Used to cast the vote
2. **RevokeVote:** to revoke the vote
3. **GetAllVotes:** to get all available votes
4. **GetVotesByRange:** to get all votes in a range via start and end key
5. **GetVoteHistory:** to history of the votes
6. **GetVoteWithPagination:** get all votes based on the pagesize and bookmark
7. **AddVoter:** To add new voter into the system
8. **DeleteVoter:** to delete the voter from the system
9. **GetVoter:** get the details of the voter
10. **GetAllVoters:** to get all available voter
11. **GetVotersByRange:** get all voter into the range

Resources and Technologies Used:

1. Hyperledger Fabric
2. Go Lang
3. Grpc
4. Gin
5. Postman
6. Git and github