

A Project Report
on
Online Voting System

Submitted by:

BANDARU ASHOK

Reg No:2251926003



Dr Br Ambedkar University, Srikakulam

2022-2024

Table of Contents

Contents	Page No.
Project Analysis	3
Requirements	3
Tools Used	4
How to Run	4
How to Login	4
Workflow Description	5
Stepwise Output / Test Cases	6
Server Output	13
Database	14
Conclusion	14
Flow Chart	15

Project Analysis

The topic allocated to us for the socket programming project is “e-Voting System”. Here, the client establishes a connection with the server, this implies that the TCP protocol is being used. The Server should allocate a new thread for every new incoming Client, to accomplish this feature we took care of concurrent thread, that is, when the number of connections are made with the server, at that time each thread doesn't interfere with one another. Therefore, we synchronized the threads.

❖ Design and Implementation :

1. A secure server that only allows clients with authentic names and passwords to cast votes.
2. Server checks for authenticity of the client & also checks if client has already voted. It returns a message to the client according to the security check.
3. Votes are registered by admin and the vote list is stored in a csv file.
4. Server can take the client name and password and match it with the txt file.
5. If details match, then the vote is redirected to the secured Voting page.
6. The votes will then cast the vote by mentioning the poll symbol of the candidate from the candidate list provided by the server.
7. The system (server) can handle multiple clients and creates a new thread for each of them.
8. One client can cast a vote once and only once.

Requirements

Python Libraries Required :

- Pandas
- Tkinter
- Socket
- Subprocess

Tools Used

- Programming : Python
- Connection : Socket Programming
- Protocol : TCP
- User Interface : python-tkinter
- Data Storage : Using CSV files
- Data Updates : python-pandas
- OS Calls : python-subprocess

How to Run

1. Open terminal/command prompt on your PC.
2. Navigate to 'Voting' folder
3. Run command :
`python homePage.py`
4. A new home page window should open. If this doesn't happen, check your installations.
5. Login into Admin using given details in 'How to Login' part.
6. Click on the 'Run Server' Button.
7. Use the rest of the Buttons as per your need.

How to Login

❖ Admin Login :

- Admin ID : Ashok
- Password : Ashok1928

❖ **Votei Login:**

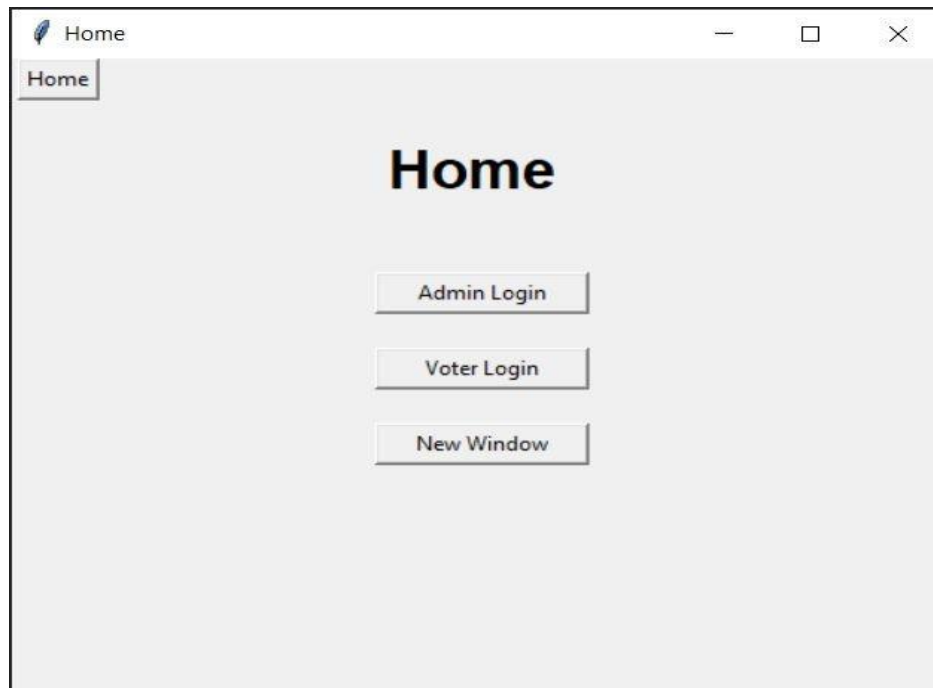
- ❑ Seivei should be iunning foi voteis to be able to login.
- Alieady iegisteied votei I.Ds : 10001 to 10005
- Passwoid (foi alieady iegisteied voteis) : abcd

Workflow Description

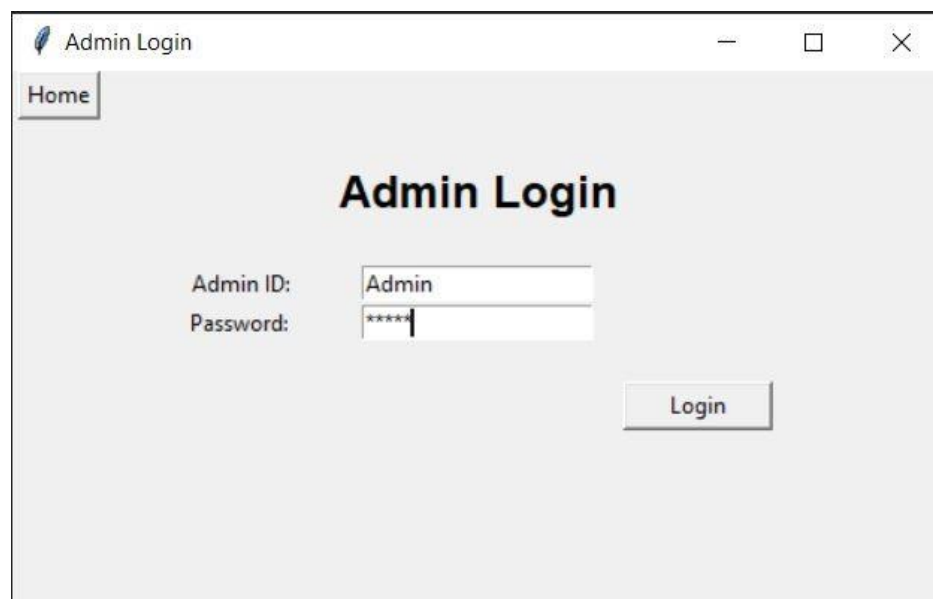
❖ Inoidei Desciption to iun & test this pioject :

1. Open teiminal & iun python homePage.py to open Home Page Window.
2. Log into Admin and piess 'Run Seivei'. This will iun the Seivei in a new console window.
3. Now that the seivei is iunning, ietuin to the admin home page window.
4. Piess 'Registei Votei' and entei details to iegistei a new votei. Remembei oi note down the 'Votei ID' that you will ieceive on successful iegistiation.
5. Piess 'Home' to ietuin to the Home. Now, piess 'Votei Login' to open the votei login page.
6. Entei the login details and you aie iediiected to the Voting Page. You will ieceive an eiioi message if the Votei is invalid oi has alieady cast a vote.
7. Cast a Vote. Now on ieceiving a success message, piess home to ietuin to home.
8. Login into Admin again. Piess 'Show Votes' to check the votes that all paities have ieceived so fai.
9. Retuin to Home. You can piess 'New Window' to open multiple pages and cast a vote concuiiently fiom multiple voteis.

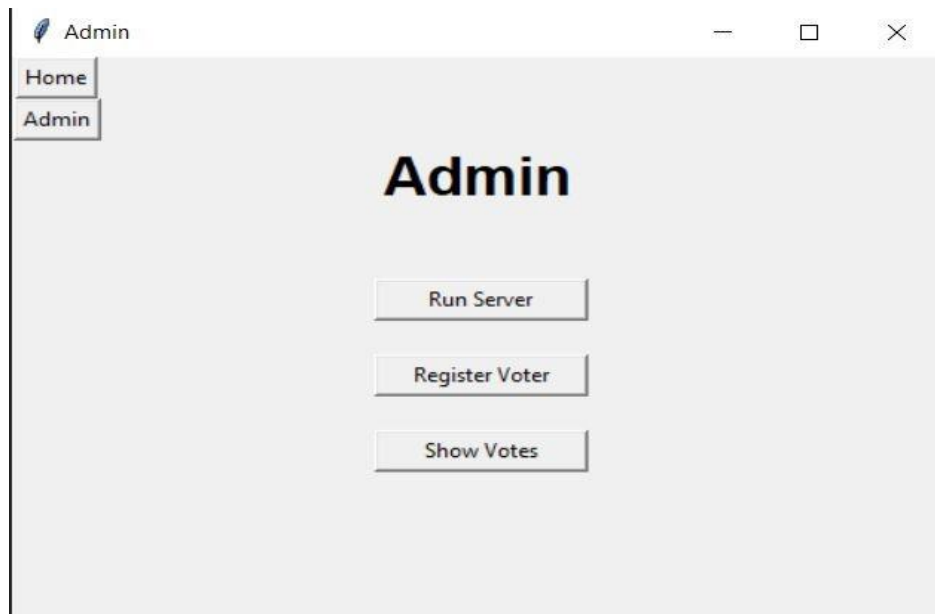
Stepwise Output / Test Cases



Home Page

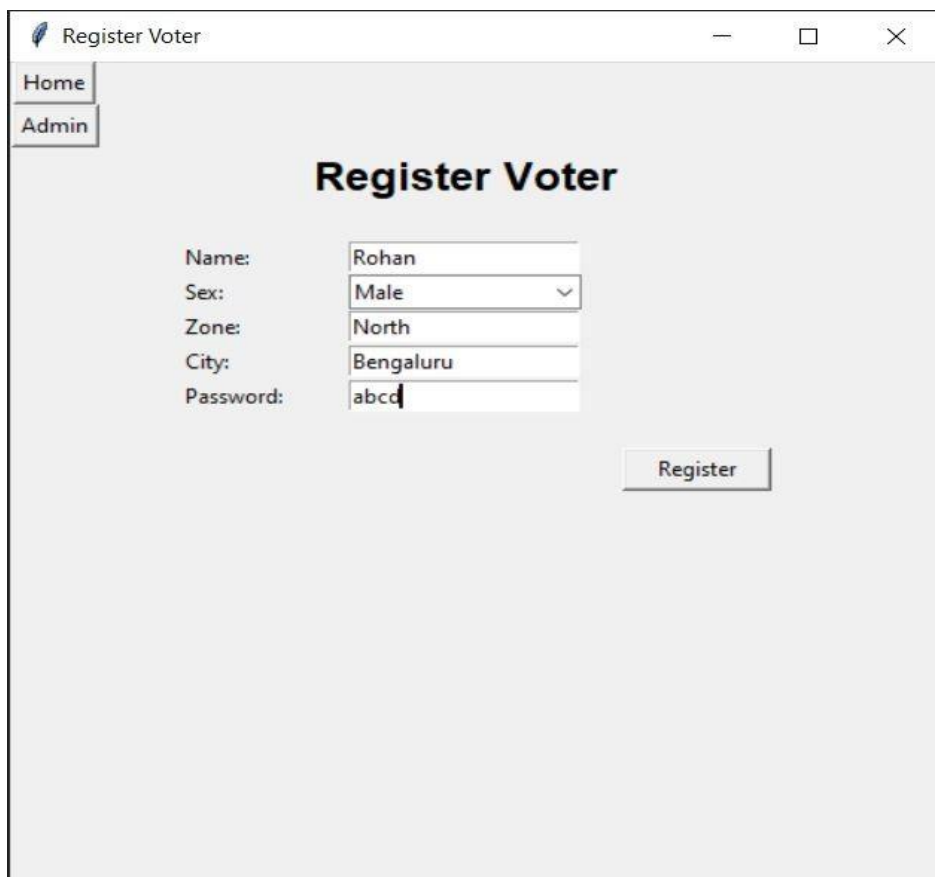


Admin Login



A screenshot of a web application window titled "Admin". The window has a standard OS-style title bar with minimize, maximize, and close buttons. On the left side, there is a vertical navigation menu with two buttons: "Home" and "Admin", with "Admin" currently selected. The main content area has a light gray background and features the word "Admin" in a large, bold, black font. Below the title, there are three rectangular buttons stacked vertically: "Run Server", "Register Voter", and "Show Votes".

Admin Home

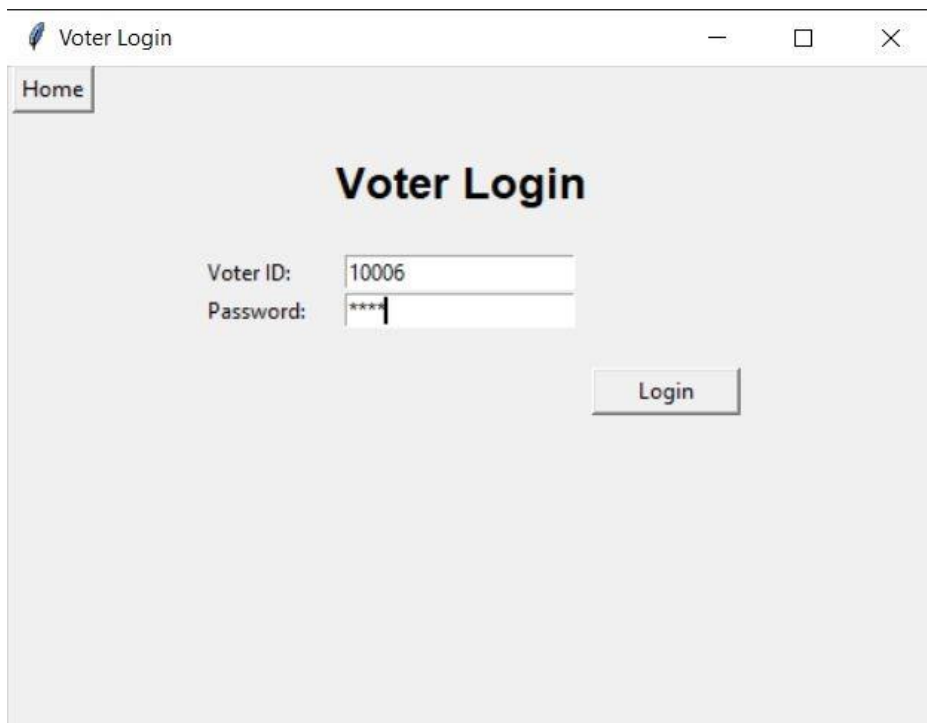


A screenshot of a web application window titled "Register Voter". The window has a standard OS-style title bar with minimize, maximize, and close buttons. On the left side, there is a vertical navigation menu with two buttons: "Home" and "Admin", with "Admin" currently selected. The main content area has a light gray background and features the text "Register Voter" in a bold, black font. Below the title, there is a registration form with the following fields: "Name:" with the value "Rohan", "Sex:" with a dropdown menu showing "Male", "Zone:" with the value "North", "City:" with the value "Bengaluru", and "Password:" with the value "abcd". To the right of these fields is a rectangular "Register" button.

Register Voter

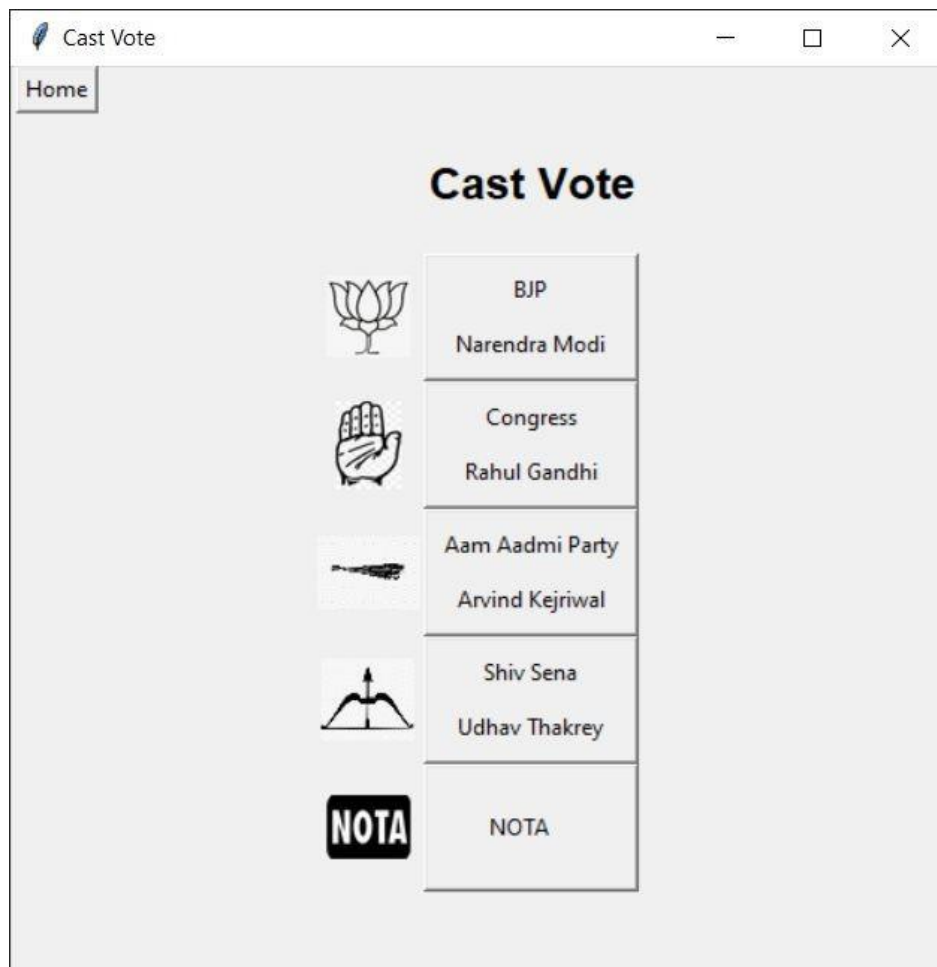


Register Success Message

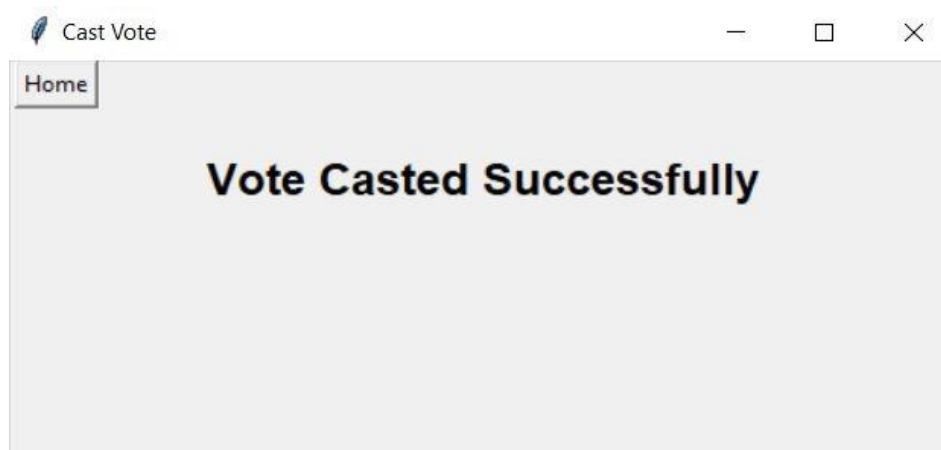


Voter Login

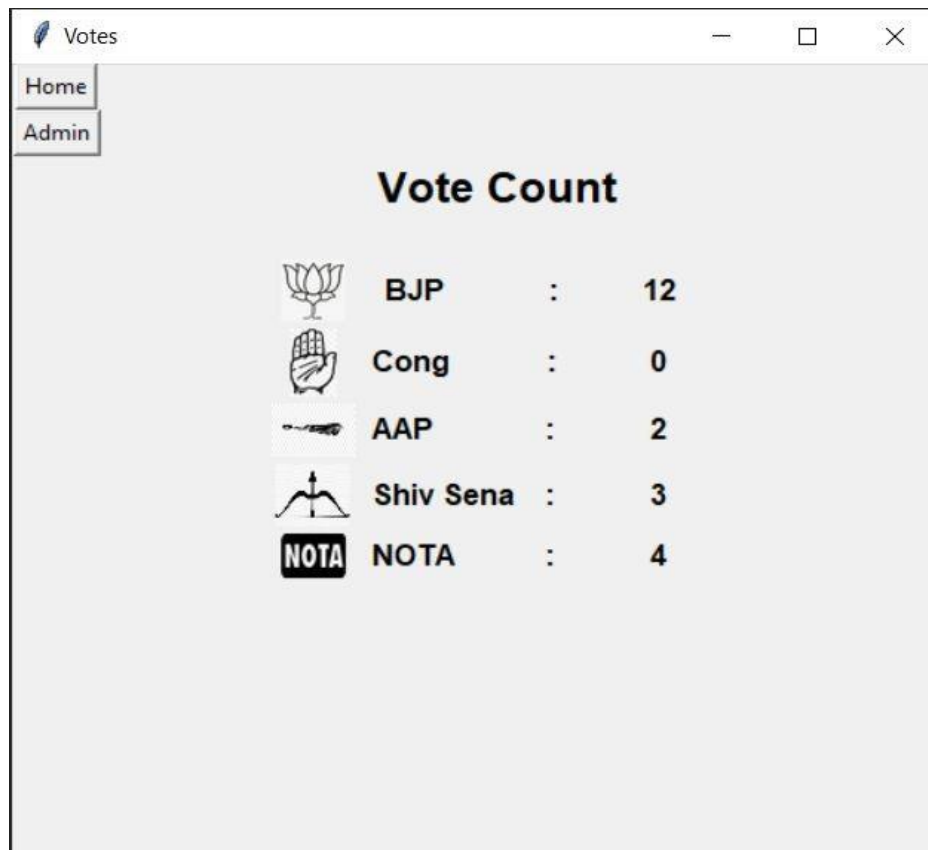
Test Case 1 : If detail matches, then it welcomes the voter and displays the name and poll symbol of the candidates.



Voting Page



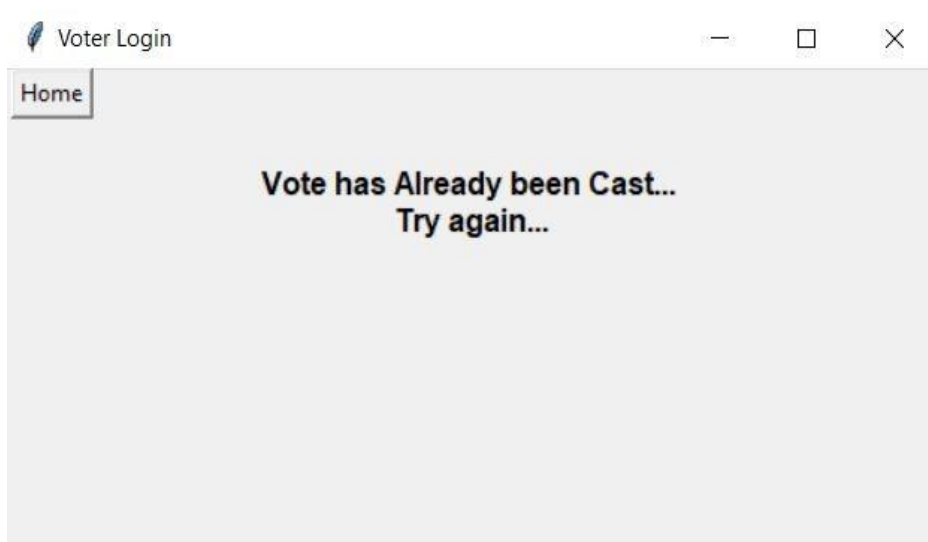
Vote Casted Successfully Message



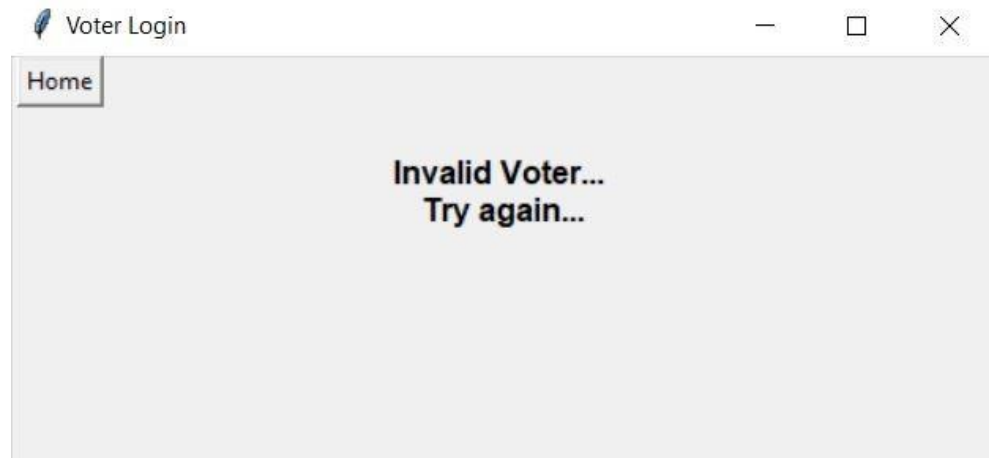
Show Votes

❖ **Error Handling :**

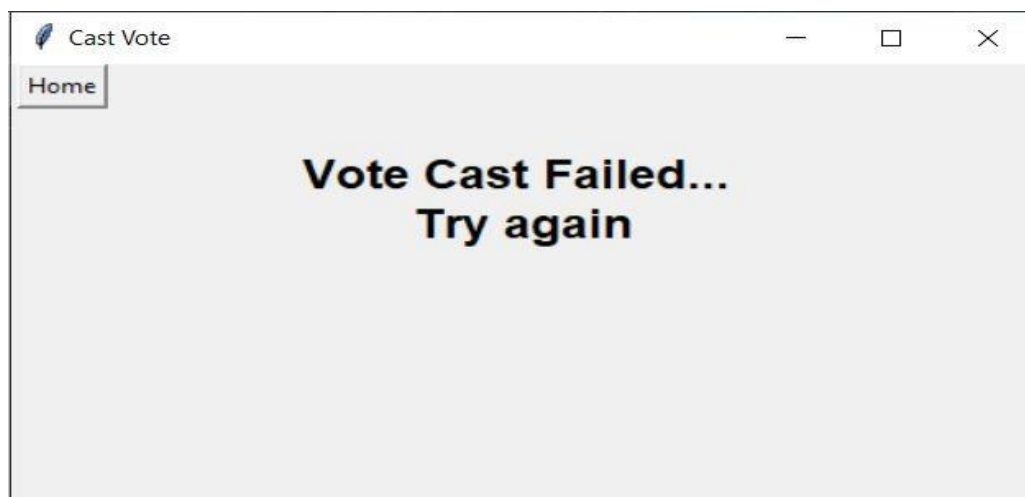
Test Case 2 : One client can cast a vote ONCE AND ONLY ONCE.



If the vote already been casted



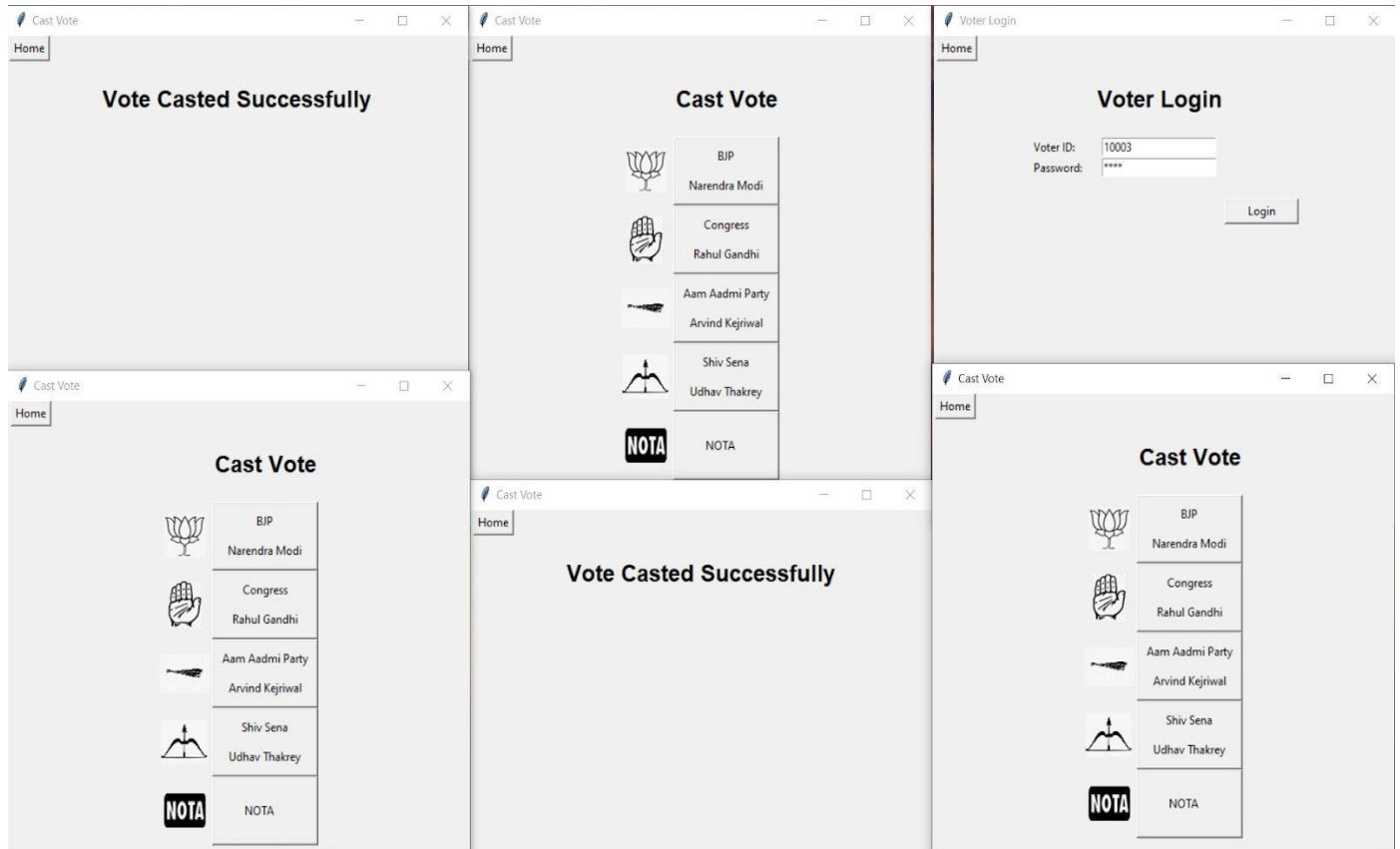
If a voter is not registered/invalid voter



Error while casting vote

❖ Voters casting votes concurrently :

Test Case 3: This system should work perfectly for at least 5 different clients at the same time.



6 Voters

Server Output

```
Select C:\ProgramData\Anaconda3\python.exe
Waiting for the connection
Listening on 0.0.0.0:4001
Connected to : ('192.168.0.113', 56631)
Voter Logged in...
ID:10006
Vote Received from ID: 10006 Processing...
Vote Casted Sucessfully by voter ID = 10006
Connected to : ('192.168.0.113', 56635)
Vote Already Cast by ID:10006
Vote Received from ID: 10006 Processing...
Vote Update Failed by voter ID = 10006
Connected to : ('192.168.0.113', 56636)
Invalid Voter
Vote Received from ID: 10006 Processing...
Vote Update Failed by voter ID = 10006
Connected to : ('192.168.0.113', 56663)
Connected to : ('192.168.0.113', 56664)
Connected to : ('192.168.0.113', 56665)
Connected to : ('192.168.0.113', 56666)
Connected to : ('192.168.0.113', 56667)
Voter Logged in...
ID:10001
Voter Logged in...
ID:10002
Connected to : ('192.168.0.113', 56668)
Voter Logged in...
ID:10006
Vote Received from ID: 10001 Processing...
Vote Casted Sucessfully by voter ID = 10001
Voter Logged in...
ID:10005
Vote Received from ID: 10005 Processing...
Vote Casted Sucessfully by voter ID = 10005
Voter Logged in...
ID:10004
Vote Received from ID: 10004 Processing...
Vote Casted Sucessfully by voter ID = 10004
Connected to : ('192.168.0.113', 56686)
Vote Already Cast by ID:10004
Vote Received from ID: 10004 Processing...
Vote Update Failed by voter ID = 10004
Connected to : ('192.168.0.113', 56687)
Voter Logged in...
ID:10005
Vote Received from ID: 10005 Processing...
Vote Casted Sucessfully by voter ID = 10005
Voter Logged in...
ID:10002
Vote Received from ID: 10002 Processing...
Vote Casted Sucessfully by voter ID = 10002
```

Database

	voter_id	Name	Gender	Zone	City	Passw	hasVoted
0	10001	Deep	Male	West	Gandhinag	abcd	0
1	10002	Prachi	Female	South	Surat	abcd	0
2	10003	Het	Male	East	Surat	abcd	0
3	10004	Shivanshi	Female	East	Gandhinag	abcd	0
4	10005	Rohan	Male	North	Bengaluru	abcd	0

Vote Info Database

	Sign	Name	Vote Count
0	bjp	Narendra Modi	15
1	cong	Rahul Gandhi	0
2	aap	Arvind Kejriwal	3
3	ss	Udhav Thakrey	4
4	nota	NOTA	5

Candidate Info Database

Conclusion

For the 'E-voting system' project we learned how to implement TCP socket programming using Python. We also learned how to connect multiple clients with one server. As the requirement of the project was to allocate a new thread by server for every new incoming Client, thus to accomplish this requirement we learned how to implement synchronized multithreading in python and implemented it in the code of socket programming.

Flow Chart

