

A Project Report
on
Online Voting System

Submitted by:

Arsh Sheikh

Falgun Patel

Dhananjay Navlani

Parth Bhimani

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, 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. Voters are registered by admin and the voter 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 voter is redirected to the secured Voting page.
6. The voters 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 : Admin ➔ Password :
admin



❖ Voter Login:

- ❑ Server should be running for voters to be able to login.
 - Already registered voter I.Ds : 10001 to 10005
 - Password (for already registered voters) : abcd

Workflow Description

❖ Inorder Description to run & test this project :

1. Open terminal & run python homePage.py to open Home Page Window.
2. Log into Admin and press 'Run Server'. This will run the Server in a new console window.
3. Now that the server is running, return to the admin home page window.
4. Press 'Register Voter' and enter details to register a new voter. Remember or note down the 'Voter ID' that you will receive on successful registration.
5. Press 'Home' to return to the Home. Now, press 'Voter Login' to open the voter login page.
6. Enter the login details and you are redirected to the Voting Page. You will receive an error message if the Voter is invalid or has already cast a vote.
7. Cast a Vote. Now on receiving a success message, press home to return to home.
8. Login into Admin again. Press 'Show Votes' to check the votes that all parties have received so far.
9. Return to Home. You can press 'New Window' to open multiple pages and cast a vote concurrently from multiple voters.

