



LOVELY
PROFESSIONAL
UNIVERSITY

School of Computer Science and Engineering

CAPSTONE PROJECT REPORT
(Project Term January-May 2022)

SECURE E-VOTING SYSTEM

Submitted to Lovely Professional University
In partial fulfillment of the requirement for the award of

DEGREE OF BACHELOR OF TECHNOLOGY
[COMPUTER SCIENCE AND ENGINEERING]

Submitted By

NAME: Aniket Singh
NAME: Chandan Chirag

Registration Number: 11805162
Registration Number: 11812444

Research Supervisor
Kavita Devi
UID: 27344
Assistant Professor
School Of Computer Science and Engineering



TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering (SCSE)

Program: P132: B.Tech. (Computer Science & Engineering)

COURSE CODE: CSE445

REGULAR/BACKLOG: Regular

GROUP NUMBER: CSERGC0107

Supervisor Name: Kavita Devi

UID: 27344

Designation: Assistant Professor

Qualification: _____

Research Experience: _____

SR.NO.	NAME OF STUDENT	Prov. Regd. No.	BATCH	SECTION	CONTACT NUMBER
1	Chandan Chirag	11812444	2018	K18FG	8578878672
2	Aniket Singh	11805162	2018	K18FG	8979766132

SPECIALIZATION AREA: Networking and Security-I

Supervisor Signature: _____

PROPOSED TOPIC: Secure E-Voting System

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	5.82
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	6.09
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	6.36
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.00
5	Social Applicability: Project work intends to solve a practical problem.	6.09
6	Future Scope: Project has potential to become basis of future research work, publication, or patent.	5.73

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Harwant Singh Arri	UID: 12975	Recommended (Y/N): Yes
PAC Member (Allied) Name: Dr.Max Bhatia	UID: 16870	Recommended (Y/N): Yes
PAC Member 3 Name: Ravishanker	UID: 12412	Recommended (Y/N): Yes

Final Topic Approved by PAC: Secure E-Voting System

Overall Remarks: Approved

PAC CHAIRPERSON Name: 13897:: Dr. Deepak Prashar

Approval Date: 04 Mar 2022

4/27/2022 10:58:37 PM

DECLARATION

We hereby declare that the project work entitled in International Journal of Scientific Research in Engineering and Management is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Ms. Kavita Devi during January to May 2022. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student 1: Aniket Singh
Registration Number: 11805162

Name of Student 2: Chandan Chirag
Registration Number: 11812444

Aniket Singh
(Signature of Student 1)
Date: 28/4/2022

Chandan Chirag
(Signature of Student 2)
Date: 28/4/2022

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Name: Ms. Kavita Devi

Designation: Assistant Professor

School of Computer Science and Engineering,

Lovely Professional University,
Phagwara, Punjab.

Date: 28/4/2022

ACKNOWLEDGEMENT

We gratefully take this occasion to express our gratitude to all the guideposts who served as lightening pillars in guiding us through this project, resulting in the successful and satisfied completion of this research.

We would like to express our special thanks of gratitude to Ms. Kavita Devi as she gave us this golden opportunity to do this wonderful project which also helped us in doing a lot of research, gave us guidelines. Support at times when we were confused or lost in the process and made us familiar with so many new aspects around us that were new to us and now, they have grown into us, and we would be able to use it in the future for a better use. We appreciate his active participation, important time and counsel, wholehearted direction, true cooperation, and painstaking commitment throughout the research and in finishing the capstone project on schedule.

We are grateful to all those, particularly our friends, who have been instrumental in creating a proper, healthy, and conductive environment for us during the project, as well as including new and fresh innovative ideas for us; without their assistance, it would have been extremely difficult for us to prepare the project within a time frame.

Table of Contents

Contents	Page No.
First Page	1
PAC Form	2
Declaration	3
Certificate	4
Acknowledgement	5
Table of Content	6
Abstract	7
Problem Analysis	7
Introduction	7-8
Project Description	8
Design and Implementation	9
Flow Chart	9-11
Requirements	11
Tools Used	12
How to Login	12
Important Module	12-13
Project Result	13
Project Plan	13
Source Code	14-25
Stepwise Output / Test Cases	26-31
Server Output	32
Database	32-33
Discussion	33
Conclusion	33
Bibliography	33-34
Complete Flow Chart	35

ABSTRACT

Secure E-Voting System is a python-based project. It is an efficient and financially savvy way of leading a democratic method, which has the characteristics of being large-scale data and real-time, as well as requiring a high level of security. Nonetheless, there are concerns about the security of systems administration and protection of correspondence for e-casting a ballot have been developed. Here, the client establishes a connection with the server, this implies that the TCP protocol is being used. For each new arriving Client, the Server should create a new thread to accomplish this feature we took care of concurrent thread, that is, when the number of connections is made with the server, that time each thread doesn't interfere with one another. Therefore, we synchronized the threads. Securing e-casting a ballot is very urgent and has turned into a famous theme in correspondences and systems administration. This system puts a lot of trust in the central authority. It manages the security of information during the enrolment of the electors while surveying on the final voting day and then some and examining the votes to guarantee an impartial democratic climate. It additionally guarantees that the citizen is an enlisted and interesting elector who is qualified to cast a ballot and its polling form for casting a ballot is customized and gotten. The execution result shows that it is a practical and secure e-casting ballot framework, which takes care of the issue of falsification of votes during e-casting a ballot

PROBLEM ANALYSIS

A few weeks ago, there was an election in the five states of India. Among these five states one state is where I live in i.e., U.P. So, when I have gone to the polling booth to vote that day, I have seen an excessively big queue and, in this pandemic, we should follow the covid protocol but due to heavy rush no one was even trying to follow the covid protocol which seems to be dangerous and even worried about the future covid waves. Apart from these, standing in a big queue and voting is a bit time-consuming in this fast-moving world, so E-voting can help people save time.

E-casting a ballot is a proficient and financially savvy way of conducting a voting procedure, which has the characteristic of always being forthcoming with information and stating a high level of security. In any case, worry Issues have been expressed concerning the integrity of the Internet as well as the safety of correspondence. The level of secrecy required for e-casting a ballot cannot be achieved only by encryption. In e-casting a ballot, for instances, a vote should never be traceable back to that same voter. E-casting a ballot utilizes PCs, and cryptography with the fundamental of encryption and signature algorithms. Instruction to plan a safer and more practical e-casting ballot framework has turned into a well-known topic in the industry and data integrity. To work on the security and secrecy of e-casting a ballot, we present methods to construct e-voting systems.

INTRODUCTION

Voting is a method to settle on an aggregate opinion among a group or a meeting or electorates. Casting a ballot is normally following discussions, conversations, and political races. During casting a ballot, the individual to be chosen is the applicant of a political race, and the individual who casts a ballot for their chosen candidate is a voter. Usually, the voter can vote following the list of candidates or vote for any other persons he/her prefers. Casting a ballot voting form should be unsigned and set apart by the electors in private booths with the goal that no other person can find out for whom a person is voting. Since the seventeenth century, casting a ballot has been the typical system by which present-day delegates a majority rules government has worked. Likewise, voting is utilized in numerous other private associations and gatherings, like clubs, corporations, organizations, and

deliberate affiliations.

Many old offline services, including voting, mail, payment, and installment, are migrating to digital websites due to improvements on the Internet, information technologies, and data breakthroughs developed. E-voting is another name for online voting (e-casting a ballot). It's an electronic voting system that allows you to cast and count your votes. Users of e-casting a ballot are citizens and political race specialists i.e., election authorities. The citizen can give his/her votes electronically to the political party's specialists from any area through e-voting. The election authorities are answerable for gathering votes from electors. E-casting a ballot can save time and effort with high proficiency and adaptability, which is getting more and more attention instead of the conventional way of voting. With the advancement of the Internet, e-voting turned into a significant method for some organizations.

Casting a ballot cycle is significant irrespective of where you are in any corner of the globe in public elections, private or independent bodies for choosing somebody. Due to the present scenario of pandemic crisis, it will be hard to hold elections in the physical way in the foreseeable future. Electronic voting (E-casting a ballot) is an option for getting sorted out by casting a ballot online through which an elector can make their choice remotely from any side of the globe via a computer or a smartphone. Electronic voting is a moving assignment to guarantee to cast a ballot prerequisite like eligibility, Accuracy, Straightforward, sturdiness, security, and dependability of the client and the framework. In Electronic voting to empower remote voting, all the political information as well as the application modules should be put in the Cloud framework, and it should be completely secure that unauthorized persons cannot access it. Data confidentiality and integrity have become a big concern across numerous enterprises as the use of cloud - based services have expanded. The focus of this research is to use Crypt DB to provide security and dependability in the computerized election process. Voter's data, Candidate data, and votes initiated by voters are all stored in a secure encrypted database called Crypt DB with in proposed online electronic voting system. Crypt DB has already been verified to be a safe and secure database that encrypts data before saving it and uses various degrees of encryption. At any stage of the polling system, the proposed solutions do not reveal any information to outsiders, and thus the results of the election process can be reached utilizing the online mechanism with security, confidentiality, and integrity are all important considerations.

PROJECT DESCRIPTION

The main objective of this paper is to progress an online secure E-voting system it is a requirement nowadays. The web-based framework ought to fulfill the essential prerequisites like the product utilized ought to be relied upon should be regarded as trustworthy and safe. This is a computerized system. It will be a secure framework because the client can cast a vote only once as the database will not allow multiple votes for every user as everyone eligible to cast a vote had their information in the database. This system should be able to handle big databases. This online voting system is extremely easy to use and very efficient. In comparison to traditional voting, this technique does not necessitate much work. It will be the greatest way to vote once the method is understood by all of us. It has some negatives, such as software troubles, internet problems, and so on, in addition to all these benefits.

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's name and password and match it with the txtfile.
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.

FLOW CHART

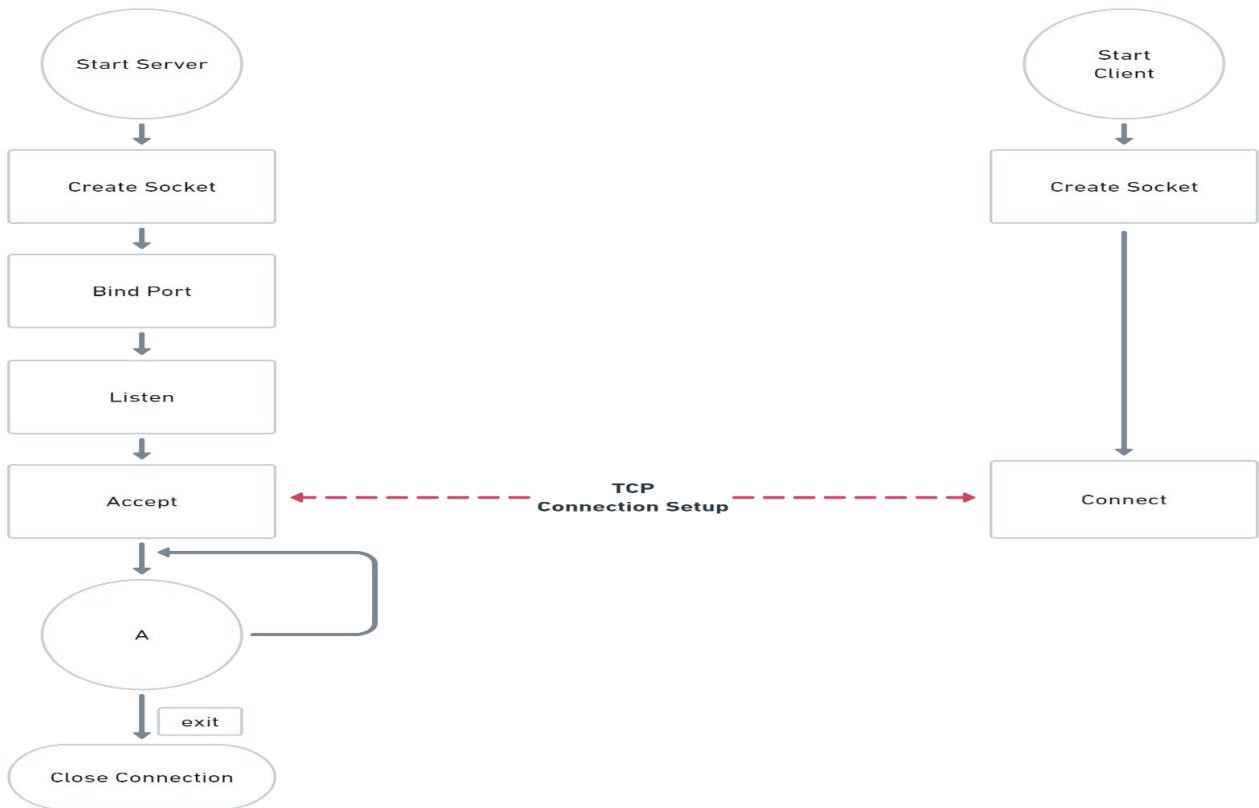


Figure 1: Starting and Creating the Server

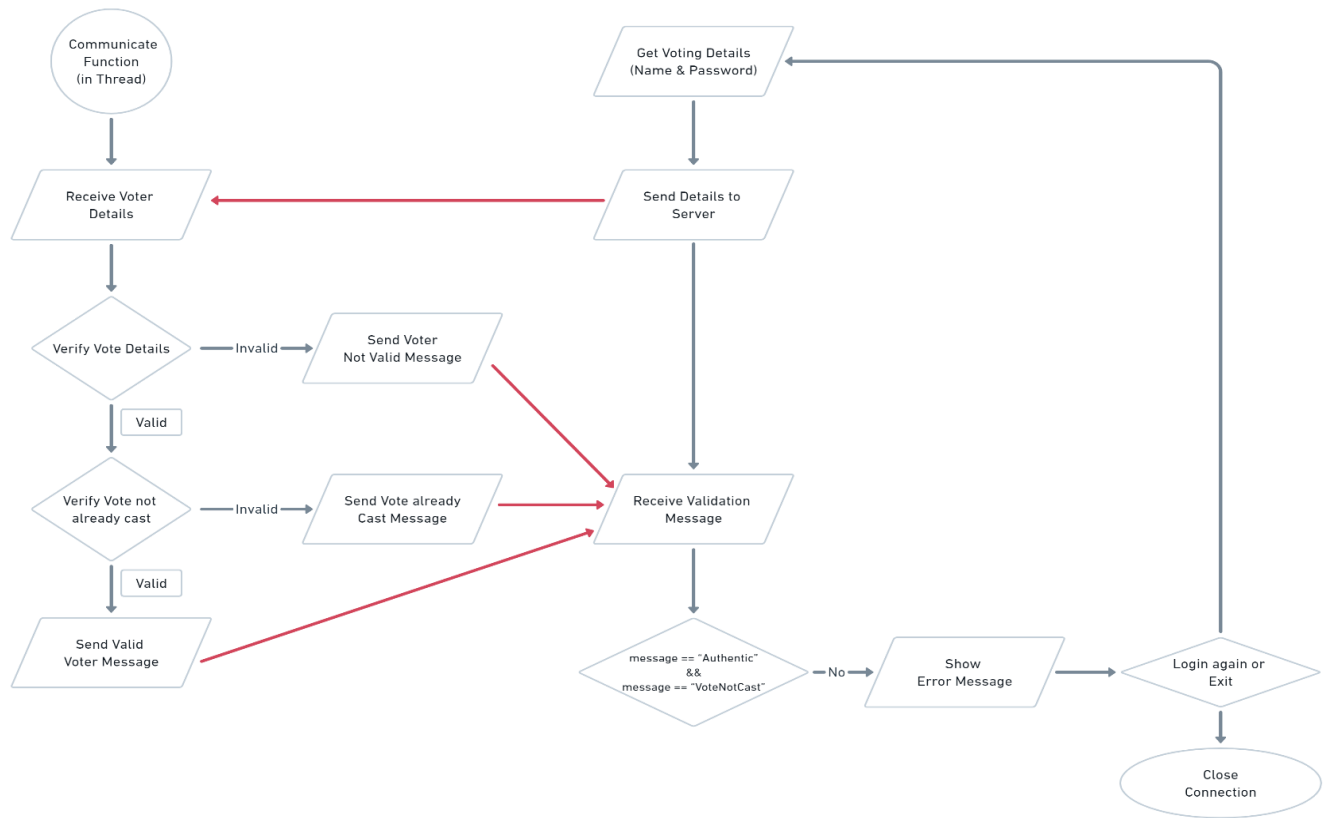


Figure 2: Casting the Vote

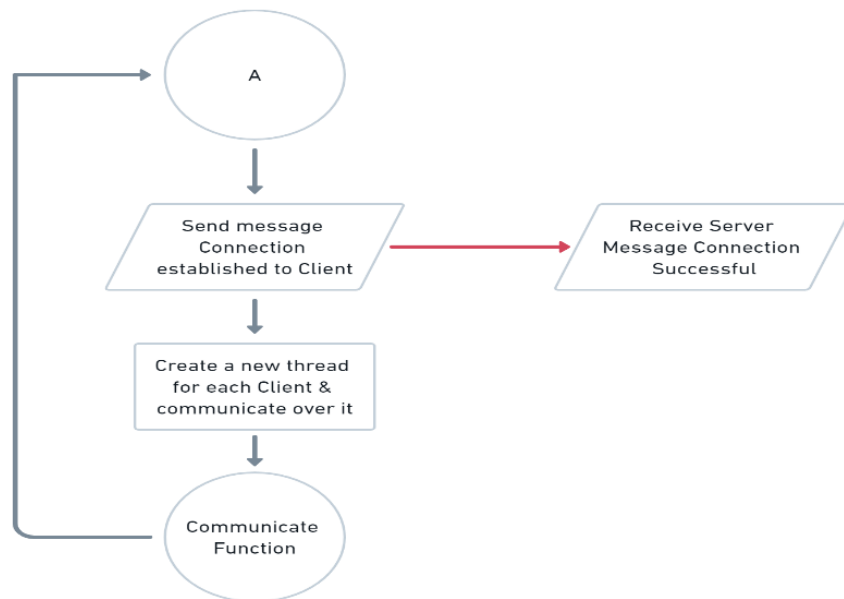


Figure 3: Creating the Thread

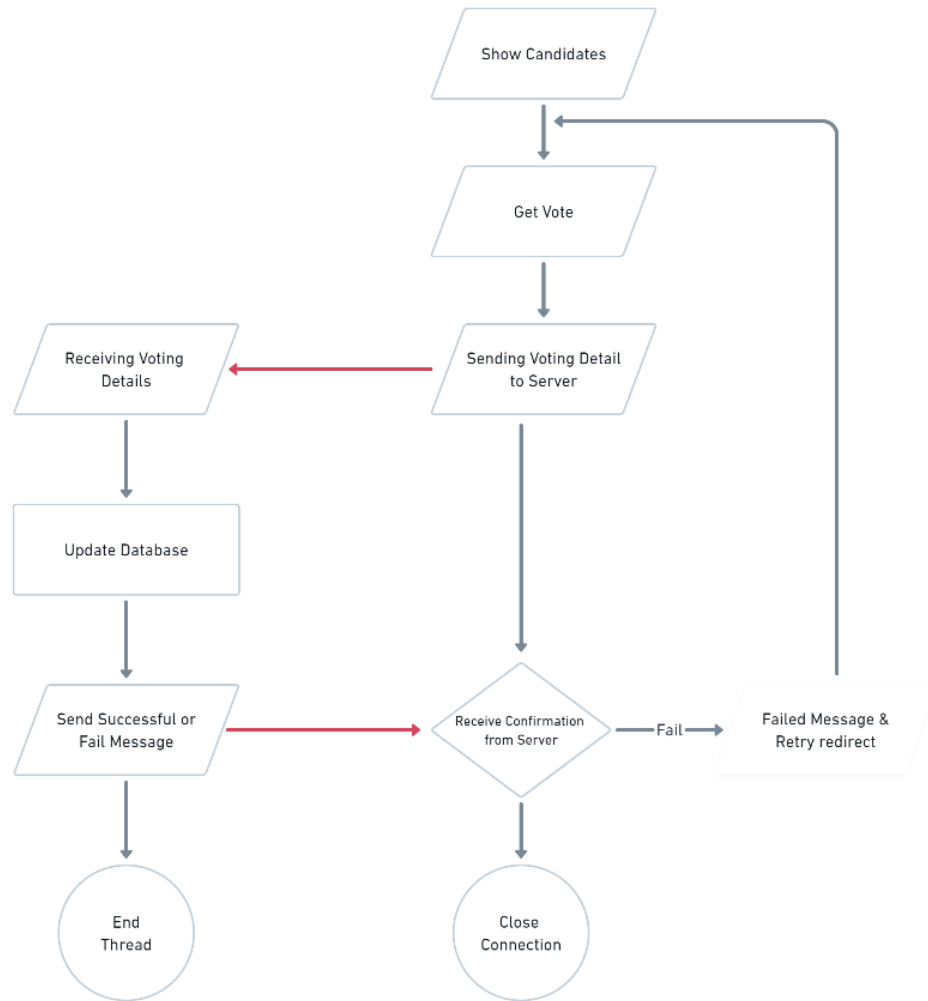


Figure 4: Closing the thread and Server

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 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.D.s: 10001 to 10005
- Password (for already registered voters): ab@12

IMPORTANT MODULES

- I. CLIENT MODULE – It includes a login name and a specific private key that he or she can use to access the web-based voting system. The admin will deliver this to the client. After the client has successfully signed in, he will be able to get a list of the candidates who have been recorded by the administrator, as well as the results after the vote has been cast.
- II. ADMIN LOGIN – It comprises a login name and password of the admin through which the system will be connected to the server and further processes will be executed such as registration of the new user and creating multiple home page windows.
- III. USER LOGIN – The Moderator physically provides each voter with a one-of-a-kind username and private key, i.e., password. The voter logs in with the username and private key and performs the essential right of voting. If the mistaken username and the private key are entered, the message of the invalid user will be given on the screen.
- IV. VOTE – This provides the voter with a slate of nominees within his or her body electorate, as well as the option of selecting their favorite candidate from the list which is stored by the admin who is competing in the election and after the successful voting user will even get the confirmation.
- V. VOTER LIST – The voter's list can be found here. Every voter will have a separate voter ID

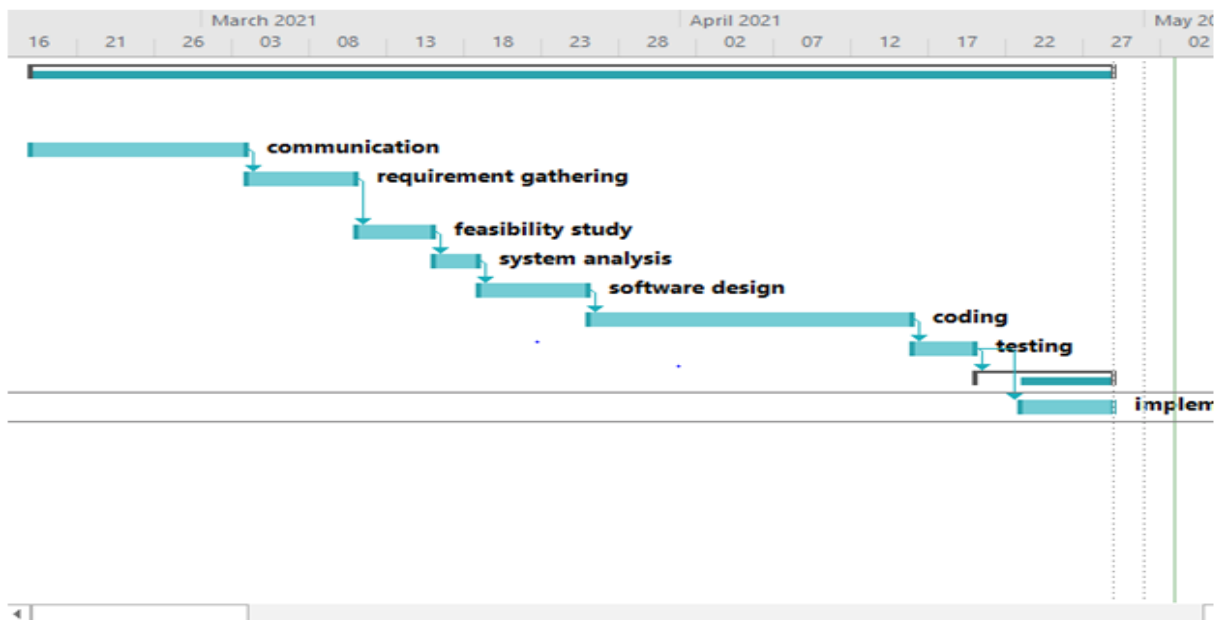
password. It comprises with few more details like gender, city, zone, and whether he/she has cast the vote or not.

- VI. CANDIDATE LIST** – It is possible to examine a list of race contestants. It incorporates the competitor's name, party names, and gathering image along with vote count.

PROJECT RESULTS

1. Open the project and run the homePage.py file 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 home.
8. Login into Admin again. Press 'Show Votes' to check the votes that all parties have received so far.
9. Return Home. You can press 'New Window' to open multiple pages and cast a vote concurrently from multiple voters.

PROJECT PLAN



SOURCE CODE

I. HomePage.py

```
import subprocess as sb_p
import tkinter as tk
from tkinter import *
from Admin import AdmLogin
from voter import voterLogin

def Home(root, frame1, frame2):

    for frame in root.winfo_children():
        for widget in frame.winfo_children():
            widget.destroy()

    Button(frame2, text="Home", command = lambda: Home(root, frame1,
frame2)).grid(row=0,column=0)
    Label(frame2, text=" ").grid(row = 0,column = 1)
    Label(frame2, text=" ").grid(row = 0,column = 2)
    Label(frame2, text=" ").grid(row = 1,column = 1)
    frame2.pack(side=TOP)

    root.title("Home")

    Label(frame1, text="Home", font=('Helvetica', 25, 'bold')).grid(row = 0,
column = 1, rowspan=1)
    Label(frame1, text="").grid(row = 1,column = 0)
    #Admin Login
    admin = Button(frame1, text="Admin Login", width=15, command = lambda:
AdmLogin(root, frame1))

    #Voter Login
    voter = Button(frame1, text="Voter Login", width=15, command = lambda:
voterLogin(root, frame1))

    #New Tab
    newTab = Button(frame1, text="New Window", width=15, command = lambda:
sb_p.call('start python homePage.py', shell=True))

    Label(frame1, text="").grid(row = 2,column = 0)
    Label(frame1, text="").grid(row = 4,column = 0)
    Label(frame1, text="").grid(row = 6,column = 0)
    admin.grid(row = 3, column = 1, columnspan = 2)
    voter.grid(row = 5, column = 1, columnspan = 2)
    newTab.grid(row = 7, column = 1, columnspan = 2)

    frame1.pack()
    root.mainloop()
```

```

def new_home():
    root = Tk()
    root.geometry('500x500')
    frame1 = Frame(root)
    frame2 = Frame(root)
    Home(root, frame1, frame2)

if __name__ == "__main__":
    new_home()

```

II. AdminFunction.py

```

import tkinter as tk
import dframe as df
from tkinter import *
from dframe import *
from PIL import ImageTk, Image

def resetAll(root, frame1):
    #df.count_reset()
    #df.reset_voter_list()
    #df.reset_cand_list()
    Label(frame1, text="").grid(row = 10, column = 0)
    msg = Message(frame1, text="Reset Complete", width=500)
    msg.grid(row = 11, column = 0, columnspan = 5)

def showVotes(root, frame1):
    result = df.show_result()
    root.title("Votes")
    for widget in frame1.winfo_children():
        widget.destroy()

    Label(frame1, text="Vote Count", font=('Helvetica', 18, 'bold')).grid(row = 0,
    column = 1, rowspan=1)
    Label(frame1, text="").grid(row = 1, column = 0)

    vote = StringVar(frame1, "-1")

    bjpLogo = ImageTk.PhotoImage ((Image.open("img/bjp.png")).
    resize((35,35),Image.ANTIALIAS)) bjpImg = Label(frame1,
    image=bjpLogo).grid(row = 2, column = 0)

    congLogo = ImageTk.PhotoImage ((Image.open("img/cong.jpg")).
    resize((25,38),Image.ANTIALIAS)) congImg = Label(frame1,
    image=congLogo).grid(row = 3, column = 0)

    aapLogo = ImageTk.PhotoImage ((Image.open("img/aap.png")).
    resize((45,30),Image.ANTIALIAS)) aapImg = Label(frame1,
    image=aapLogo).grid(row = 4, column = 0)

    ssLogo = ImageTk.PhotoImage ((Image.open("img/ss.png")).
    resize((40,35),Image.ANTIALIAS)) ssImg = Label(frame1,
    image=ssLogo).grid(row = 5, column = 0)

    notaLogo = ImageTk.PhotoImage ((Image.open("img/nota.jpg")).

```

```

resize((35,25),Image.ANTIALIAS)) notaImg = Label(frame1,
image=notaLogo).grid(row = 6,column = 0)

Label(frame1, text="BJP          :          ", font=('Helvetica', 12,
'bold')).grid(row = 2, column = 1)
Label(frame1, text=result['bjp'], font=('Helvetica', 12, 'bold')).grid(row =
2, column = 2)

Label(frame1, text=" Cong          :          ", font=('Helvetica', 12,
'bold')).grid(row = 3, column = 1)
Label(frame1, text=result['cong'], font=('Helvetica', 12, 'bold')).grid(row =
3, column = 2)

Label(frame1, text=" AAP          :          ", font=('Helvetica', 12,
'bold')).grid(row = 4, column = 1)
Label(frame1, text=result['aap'], font=('Helvetica', 12, 'bold')).grid(row =
4, column = 2)

Label(frame1, text=" Shiv Sena      :          ", font=('Helvetica', 12,
'bold')).grid(row = 5, column = 1)
Label(frame1, text=result['ss'], font=('Helvetica', 12, 'bold')).grid(row = 5,
column = 2)

Label(frame1, text=" NOTA          :          ", font=('Helvetica', 12,
'bold')).grid(row = 6, column = 1)
Label(frame1, text=result['nota'], font=('Helvetica', 12, 'bold')).grid(row =
6, column = 2)

frame1.pack()
root.mainloop()

```

III. Admin.py

```

import subprocess as sb_p
import tkinter as tk
import registerVoter as regV
import admFunc as adFunc
from tkinter import *
from registerVoter import *
from admFunc import *

def AdminHome(root,frame1,frame3):
    root.title("Admin")
    for widget in frame1.winfo_children():widget.destroy()

    Button(frame3, text="Admin", command = lambda: AdminHome(root, frame1,
frame3)).grid(row = 1, column = 0)
    frame3.pack(side=TOP)

    Label(frame1, text="Admin", font=('Helvetica', 25, 'bold')).grid(row = 0,
column = 1)
    Label(frame1, text="").grid(row = 1,column = 0)

    #Admin Login
    runServer = Button(frame1, text="Run Server", width=15, command = lambda:
sb_p.call('start python Server.py', shell=True))

```



```

#Voter Login
registerVoter = Button(frame1, text="Register Voter", width=15, command =
lambda: regV.Register(root, frame1))

#Show Votes
showVotes = Button(frame1, text="Show Votes", width=15, command = lambda:
adFunc.showVotes(root, frame1))

#Reset Data
reset = Button(frame1, text="Reset All", width=15, command = lambda:
adFunc.resetAll(root, frame1))

Label(frame1, text="").grid(row = 2,column = 0)
Label(frame1, text="").grid(row = 4,column = 0)
Label(frame1, text="").grid(row = 6,column = 0)
Label(frame1, text="").grid(row = 8,column = 0)
runServer.grid(row = 3, column = 1, columnspan = 2)
registerVoter.grid(row = 5, column = 1, columnspan = 2)
showVotes.grid(row = 7, column = 1, columnspan = 2)
# reset.grid(row = 9, column = 1, columnspan = 2)

frame1.pack()
root.mainloop()

def log_admin(root,frame1,admin_ID,password):

    if(admin_ID=="Admin" and password=="admin"):
        frame3 = root.winfo_children()[1]
        AdminHome(root, frame1, frame3)
    else:
        msg = Message(frame1, text="Either ID or Password is Incorrect",
width=500)
        msg.grid(row = 6, column = 0, columnspan = 5)

def AdmLogin(root,frame1):

    root.title("Admin Login")
    for widget in frame1.winfo_children():widget.destroy()

    Label(frame1, text="Admin Login", font=('Helvetica', 18, 'bold')).grid(row =
0, column = 2, rowspan=1)
    Label(frame1, text="").grid(row = 1,column = 0)
    Label(frame1, text="Admin ID:      ", anchor="e", justify=LEFT).grid(row =
2,column = 0)
    Label(frame1, text="Password:      ", anchor="e", justify=LEFT).grid(row =
3,column = 0)

    admin_ID = tk.StringVar()
    password = tk.StringVar()

    e1 = Entry(frame1, textvariable = admin_ID)
    e1.grid(row = 2,column = 2)
    e2 = Entry(frame1, textvariable = password, show = '*')
    e2.grid(row = 3,column = 2)

    sub = Button(frame1, text="Login", width=10, command = lambda: log_admin(root,
frame1, admin_ID.get(), password.get()))

```

```

Label(frame1, text="").grid(row = 4, column = 0)
sub.grid(row = 5, column = 3, columnspan = 2)

frame1.pack()
root.mainloop()

```

IV. Dframe.py

```

import pandas as pd
from pathlib import Path

# path = Path("C:/Users/Desktop/Sem-5/CS301 CN/Project/Voting/database")
path = Path("database")

def count_reset():
    df=pd.read_csv(path/'voterList.csv')
    df=df[['voter_id', 'Name', 'Gender', 'Zone', 'City', 'Passw', 'hasVoted']]
    for index, row in df.iterrows():
        df['hasVoted'].iloc[index]=0
    df.to_csv(path/'voterList.csv')

    df=pd.read_csv(path/'cand_list.csv')
    df=df[['Sign', 'Name', 'Vote Count']]
    for index, row in df.iterrows():
        df['Vote Count'].iloc[index]=0
    df.to_csv (path/'cand_list.csv')

def reset_voter_list():
    df =
pd.DataFrame(columns=['voter_id', 'Name', 'Gender', 'Zone', 'City', 'Passw', 'hasVoted']
)
    df=df[['voter_id', 'Name', 'Gender', 'Zone', 'City', 'Passw', 'hasVoted']]
    df.to_csv(path/'voterList.csv')

def reset_cand_list():
    df = pd.DataFrame(columns=['Sign', 'Name', 'Vote Count'])
    df=df[['Sign', 'Name', 'Vote Count']]
    df.to_csv(path/'cand_list.csv')

def verify(vid,passw):
    df=pd.read_csv(path/'voterList.csv')
    df=df[['voter_id', 'Passw', 'hasVoted']]
    for index, row in df.iterrows():
        if df['voter_id'].iloc[index]==vid and df['Passw'].iloc[index]==passw:
            return True
    return False

def isEligible(vid):
    df=pd.read_csv(path/'voterList.csv')
    df=df[['voter_id', 'Name', 'Gender', 'Zone', 'City', 'Passw', 'hasVoted']]
    for index, row in df.iterrows():
        if df['voter_id'].iloc[index]==vid and df['hasVoted'].iloc[index]==0:
            return True
    return False

```

```

def vote_update(st,vid):
    if isEligible(vid):
        df=pd.read_csv (path/'cand_list.csv')
        df=df[['Sign','Name','Vote Count']]
        for index, row in df.iterrows():
            if df['Sign'].iloc[index]==st:
                df['Vote Count'].iloc[index]+=1

        df.to_csv (path/'cand_list.csv')

        df=pd.read_csv(path/'voterList.csv')
        df=df[['voter_id','Name','Gender','Zone','City','Passw','hasVoted']]
        for index, row in df.iterrows():
            if df['voter_id'].iloc[index]==vid:
                df['hasVoted'].iloc[index]=1

        df.to_csv(path/'voterList.csv')

    return True
return False

def show_result():
    df=pd.read_csv (path/'cand_list.csv')
    df=df[['Sign','Name','Vote Count']]
    v_cnt = {}
    for index, row in df.iterrows():
        v_cnt[df['Sign'].iloc[index]] = df['Vote Count'].iloc[index]
    # print(v_cnt)
    return v_cnt

def taking_data_voter(name,gender,zone,city,passw):
    df=pd.read_csv(path/'voterList.csv')
    df=df[['voter_id','Name','Gender','Zone','City','Passw','hasVoted']]
    row,col=df.shape
    if row==0:
        vid = 10001
        df = pd.DataFrame({"voter_id":[vid],
                           "Name":[name],
                           "Gender":[gender],
                           "Zone":[zone],
                           "City":[city],
                           "Passw":[passw],
                           "hasVoted":[0]},)
    else:
        vid=df['voter_id'].iloc[-1]+1
        df1 = pd.DataFrame({"voter_id":[vid],
                            "Name":[name],
                            "Gender":[gender],
                            "Zone":[zone],
                            "City":[city],
                            "Passw":[passw],
                            "hasVoted":[0]},)

        df=df.append(df1,ignore_index=True)

    df.to_csv(path/'voterList.csv')

```

```
return vid
```

V. Registervoter.py

```
import tkinter as tk
import dframe as df
import Admin as adm
from tkinter import ttk
from Admin import *
from tkinter import *
from dframe import *

def reg_server(root, frame1, name, sex, zone, city, passw):
    if (passw == '' or passw == ' '):
        msg = Message(frame1, text="Error: Missing Fileds", width=500)
        msg.grid(row = 10, column = 0, columnspan = 5)
        return -1

    vid = df.taking_data_voter(name, sex, zone, city, passw)
    for widget in frame1.winfo_children():
        widget.destroy()

    txt = "Registered Voter with\n\n VOTER I.D. = " + str(vid)
    Label(frame1, text=txt, font=('Helvetica', 18, 'bold')).grid(row = 2, column = 1, columnspan=2)

def Register(root, frame1):

    root.title("Register Voter")
    for widget in frame1.winfo_children():
        widget.destroy()

    Label(frame1, text="Register Voter", font=('Helvetica', 18, 'bold')).grid(row = 0, column = 2, rowspan=1)
    Label(frame1, text="").grid(row = 1, column = 0)
    #Label(frame1, text="Voter ID:      ", anchor="e", justify=LEFT).grid(row = 2, column = 0)
    Label(frame1, text="Name:      ", anchor="e", justify=LEFT).grid(row = 3, column = 0)
    Label(frame1, text="Sex:      ", anchor="e", justify=LEFT).grid(row = 4, column = 0)
    Label(frame1, text="Zone:      ", anchor="e", justify=LEFT).grid(row = 5, column = 0)
    Label(frame1, text="City:      ", anchor="e", justify=LEFT).grid(row = 6, column = 0)
    Label(frame1, text="Password:  ", anchor="e", justify=LEFT).grid(row = 7, column = 0)

    #voter_ID = tk.StringVar()
    name = tk.StringVar()
    sex = tk.StringVar()
    zone = tk.StringVar()
    city = tk.StringVar()
    password = tk.StringVar()
```

```

e1 = Entry(frame1, textvariable = voter_ID).grid(row = 2, column = 2)
e2 = Entry(frame1, textvariable = name).grid(row = 3, column = 2)
e5 = Entry(frame1, textvariable = zone).grid(row = 5, column = 2)
e6 = Entry(frame1, textvariable = city).grid(row = 6, column = 2)
e7 = Entry(frame1, textvariable = password).grid(row = 7, column = 2)

e4 = ttk.Combobox(frame1, textvariable = sex, width=17)
e4['values'] = ("Male", "Female", "Transgender")
e4.grid(row = 4, column = 2)
e4.current()

reg = Button(frame1, text="Register", command = lambda: reg_server(root,
frame1, name.get(), sex.get(), zone.get(), city.get(), password.get()),
width=10)
Label(frame1, text="").grid(row = 8, column = 0)
reg.grid(row = 9, column = 3, columnspan = 2)

frame1.pack()
root.mainloop()

```

VI. Server.py

```

import socket
import threading
import dframe as df
from threading import Thread
from dframe import *

lock = threading.Lock()

def client_thread(connection):

    data = connection.recv(1024)          #receiving voter details          #2

    #verify voter details
    log = (data.decode()).split(' ')
    log[0] = int(log[0])
    if(df.verify(log[0], log[1])):          #3 Authenticate
        if(df.isEligible(log[0])):
            print('Voter Logged in... ID:'+str(log[0]))
            connection.send("Authenticate".encode())
        else:
            print('Vote Already Cast by ID:'+str(log[0]))
            connection.send("VoteCasted".encode())
    else:
        print('Invalid Voter')
        connection.send("InvalidVoter".encode())

    data = connection.recv(1024)          #4 Get Vote
    print("Vote Received from ID: "+str(log[0])+" Processing...")
    lock.acquire()

```

```

#update Database
if(df.vote_update(data.decode(),log[0])):
    print("Vote Casted Sucessfully by voter ID = "+str(log[0]))
    connection.send("Successful".encode())
else:
    print("Vote Update Failed by voter ID = "+str(log[0]))
    connection.send("Vote Update Failed".encode())

lock.release()
connection.close()

#5

def voting_Server():

    serversocket = socket.socket()
    host = socket.gethostname()
    port = 4001

    ThreadCount = 0

    try :
        serversocket.bind((host, port))
    except socket.error as e :
        print(str(e))
    print("Waiting for the connection")

    serversocket.listen(10)

    print( "Listening on " + str(host) + ":" + str(port))

    while True :
        client, address = serversocket.accept()

        print('Connected to :', address)

        client.send("Connection Established".encode())    ### 1
        t = Thread(target = client_thread,args = (client,))
        t.start()
        ThreadCount+=1
        # break

    serversocket.close()

if __name__ == '__main__':
    voting_Server()

```

VII. Voter.py

```

import tkinter as tk
import socket
from tkinter import *
from VotingPage import votingPg

```

```

def establish_connection():
    host = socket.gethostname()
    port = 4001
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))
    print(client_socket)
    message = client_socket.recv(1024) #connection establishment message #1
    if(message.decode()=="Connection Established"):
        return client_socket
    else:
        return 'Failed'

def failed_return(root, frame1, client_socket, message):
    for widget in frame1.winfo_children():
        widget.destroy()
    message = message + "... \nTry again..."
    Label(frame1, text=message, font=('Helvetica', 12, 'bold')).grid(row = 1,
        column = 1)
    client_socket.close()

def log_server(root, frame1, client_socket, voter_ID, password):
    message = voter_ID + " " + password
    client_socket.send(message.encode()) #2

    message = client_socket.recv(1024) #Authentication message
    message = message.decode()

    if(message=="Authenticate"):
        votingPg(root, frame1, client_socket)

    elif(message=="VoteCasted"):
        message = "Vote has Already been Cast"
        failed_return(root, frame1, client_socket, message)

    elif(message=="InvalidVoter"):
        message = "Invalid Voter"
        failed_return(root, frame1, client_socket, message)

    else:
        message = "Server Error"
        failed_return(root, frame1, client_socket, message)

def voterLogin(root, frame1):
    client_socket = establish_connection()
    if(client_socket == 'Failed'):
        message = "Connection failed"
        failed_return(root, frame1, client_socket, message)

    root.title("Voter Login")
    for widget in frame1.winfo_children():
        widget.destroy()

```

```

Label(frame1, text="Voter Login", font=('Helvetica', 18, 'bold')).grid(row =
0, column = 2, rowspan=1)
Label(frame1, text="").grid(row = 1,column = 0)
Label(frame1, text="Voter ID:      ", anchor="e", justify=LEFT).grid(row =
2,column = 0)
Label(frame1, text="Password:     ", anchor="e", justify=LEFT).grid(row =
3,column = 0)

voter_ID = tk.StringVar()
name = tk.StringVar()
password = tk.StringVar()

e1 = Entry(frame1, textvariable = voter_ID)
e1.grid(row = 2,column = 2)
e3 = Entry(frame1, textvariable = password, show = '*')
e3.grid(row = 3,column = 2)

sub = Button(frame1, text="Login", width=10, command = lambda:
log_server(root, frame1, client_socket, voter_ID.get(), password.get()))
Label(frame1, text="").grid(row = 4,column = 0)
sub.grid(row = 5, column = 3, columnspan = 2)

frame1.pack()
root.mainloop()

```

VIII. VotingPage.py

```

import tkinter as tk
import socket
from tkinter import *
from PIL import ImageTk,Image

def voteCast(root,frame1,vote,client_socket):

    for widget in frame1.winfo_children():
        widget.destroy()

    client_socket.send(vote.encode()) #4

    message = client_socket.recv(1024) #Success message
    print(message.decode()) #5
    message = message.decode()
    if(message=="Successful"):
        Label(frame1, text="Vote Casted Successfully", font=('Helvetica', 18,
'bold')).grid(row = 1, column = 1)
    else:
        Label(frame1, text="Vote Cast Failed... \nTry again", font=('Helvetica',
18, 'bold')).grid(row = 1, column = 1)

    client_socket.close()

```



```

def votingPg(root,frame1,client_socket):

    root.title("Cast Vote")
    for widget in frame1.winfo_children():
        widget.destroy()

    Label(frame1, text="Cast Vote", font=('Helvetica', 18, 'bold')).grid(row = 0,
    column = 1, rowspan=1)
    Label(frame1, text="").grid(row = 1,column = 0)

    vote = StringVar(frame1,"-1")

    Radiobutton(frame1, text = "BJP\n\nNarendra Modi", variable = vote, value =
    "bjp", indicator = 0, height = 4, width=15, command = lambda:
    voteCast(root,frame1,"bjp",client_socket)).grid(row = 2,column = 1)
    bjpLogo = ImageTk.PhotoImage((Image.open("img/bjp.png"))).
    resize((45,45),Image.ANTIALIAS))
    bjpImg = Label(frame1, image=bjpLogo).grid(row = 2,column = 0)

    Radiobutton(frame1, text = "Congress\n\nRahul Gandhi", variable = vote, value
    = "cong", indicator = 0, height = 4, width=15, command = lambda:
    voteCast(root,frame1,"cong",client_socket)).grid(row = 3,column = 1)
    congLogo = ImageTk.PhotoImage((Image.open("img/cong.jpg"))).
    resize((35,48),Image.ANTIALIAS))
    congImg = Label(frame1, image=congLogo).grid(row = 3,column = 0)

    Radiobutton(frame1, text = "Aam Aadmi Party\n\nArvind Kejriwal", variable =
    vote, value = "aap", indicator = 0, height = 4, width=15, command = lambda:
    voteCast(root,frame1,"aap",client_socket)).grid(row = 4,column = 1)
    aapLogo = ImageTk.PhotoImage((Image.open("img/aap.png"))).
    resize((55,40),Image.ANTIALIAS))
    aapImg = Label(frame1, image=aapLogo).grid(row = 4,column = 0)

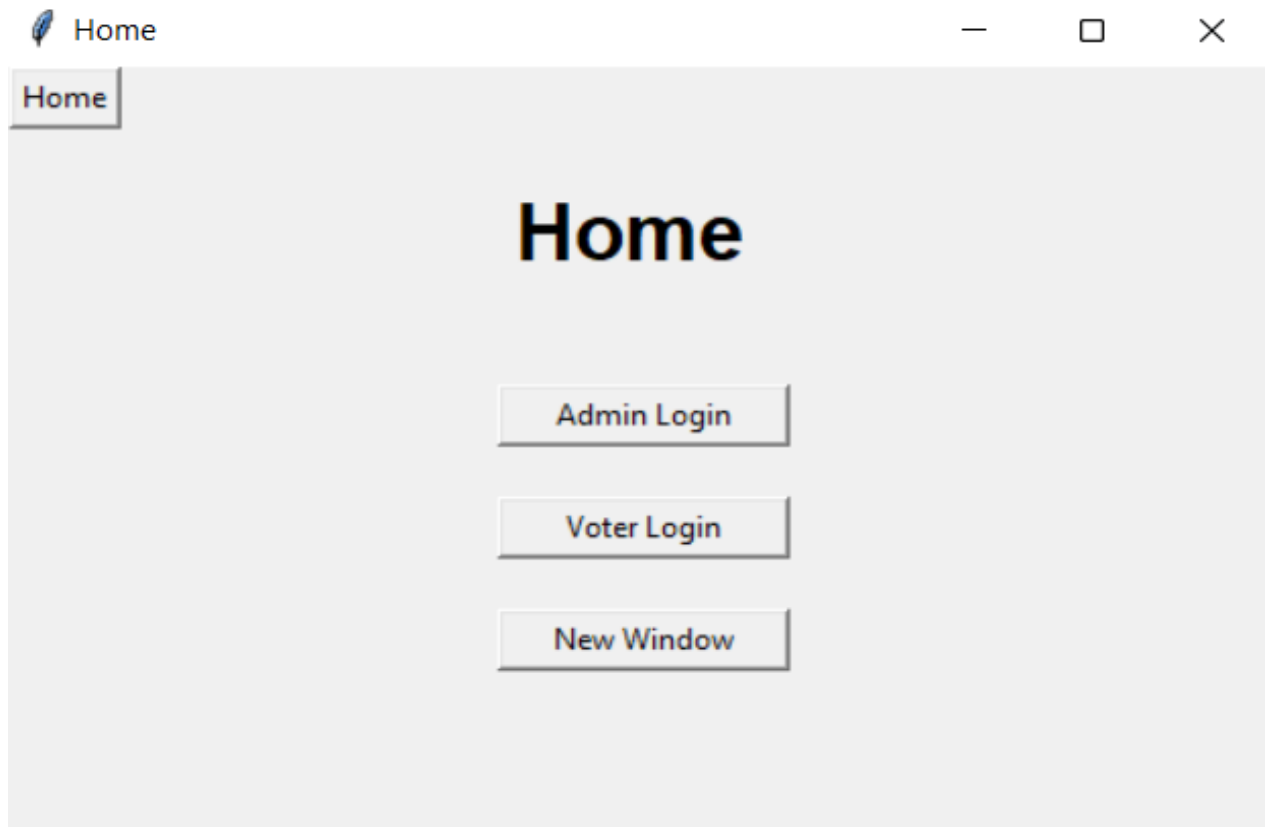
    Radiobutton(frame1, text = "Shiv Sena\n\nUdhav Thakrey", variable = vote,
    value = "ss", indicator = 0, height = 4, width=15, command = lambda:
    voteCast(root,frame1,"ss",client_socket)).grid(row = 5,column = 1)
    ssLogo = ImageTk.PhotoImage((Image.open("img/ss.png"))).
    resize((50,45),Image.ANTIALIAS))
    ssImg = Label(frame1, image=ssLogo).grid(row = 5,column = 0)

    Radiobutton(frame1, text = "\nNOTA \n ", variable = vote, value = "nota",
    indicator = 0, height = 4, width=15, command = lambda:
    voteCast(root,frame1,"nota",client_socket)).grid(row = 6,column = 1)
    notaLogo = ImageTk.PhotoImage((Image.open("img/nota.jpg"))).
    resize((45,35),Image.ANTIALIAS))
    notaImg = Label(frame1, image=notaLogo).grid(row = 6,column = 0)

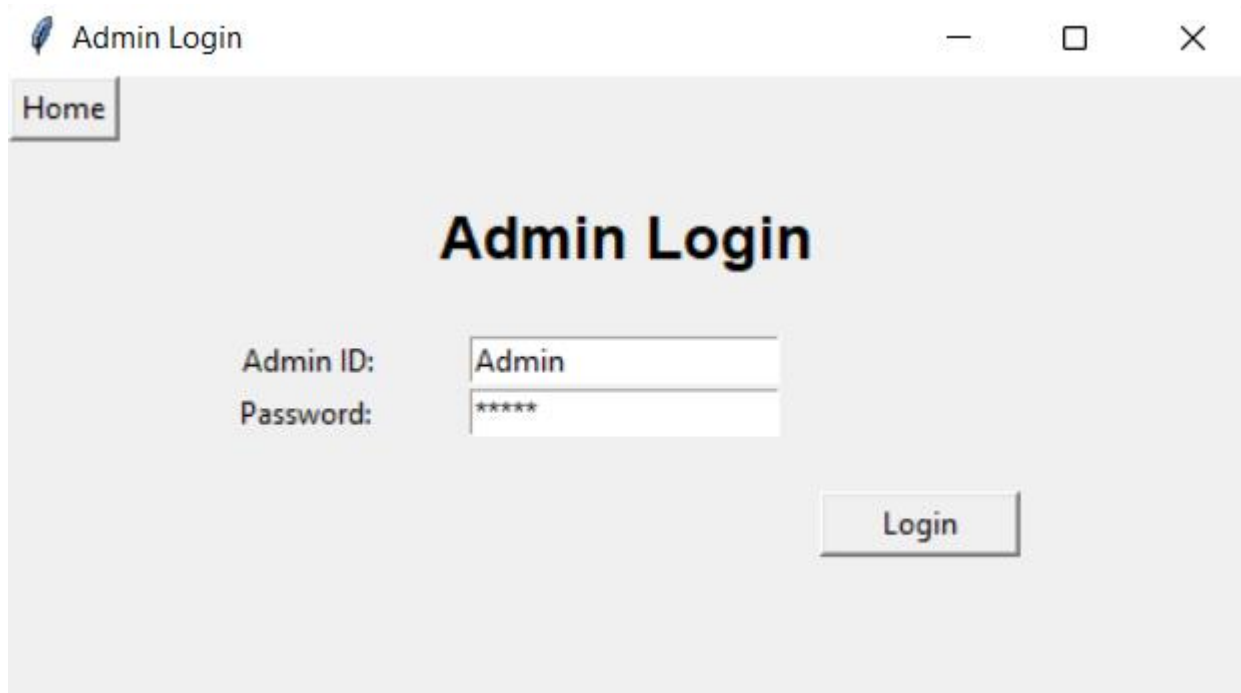
    frame1.pack()
    root.mainloop()

```


STEPWISE OUTPUT / TEST CASES



Home Page



Admin Login

 Admin—□×

HomeAdmin


Admin

Run Server

Register Voter

Show Votes

Admin Home

 Register Voter—□×

HomeAdmin

Register Voter

Name:

Aniket Singh

Sex:

Male

▼

Zone:

West

City:

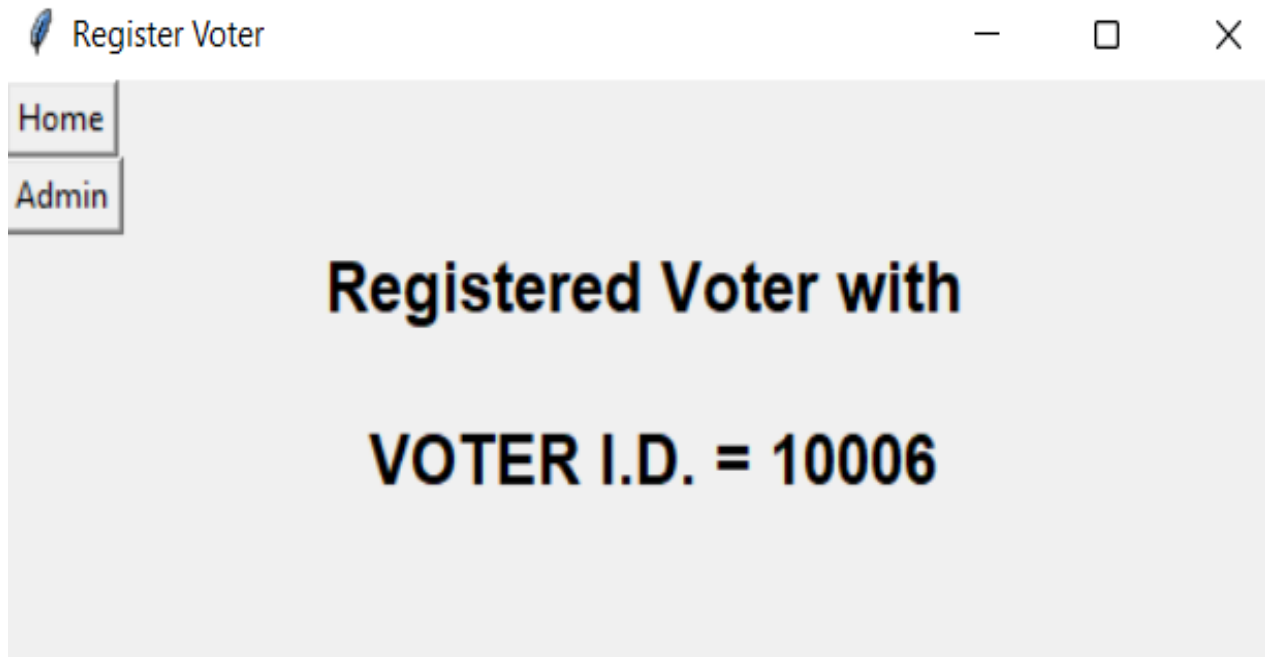
Saharanpur

Password:

ab@12

Register

Register Voter

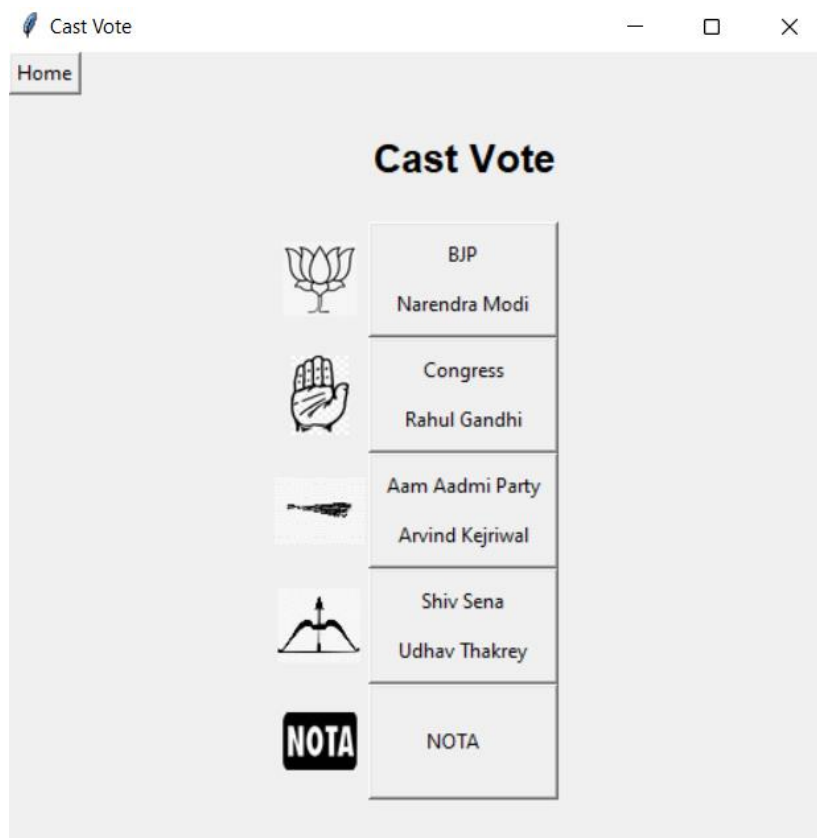


Register Success Message

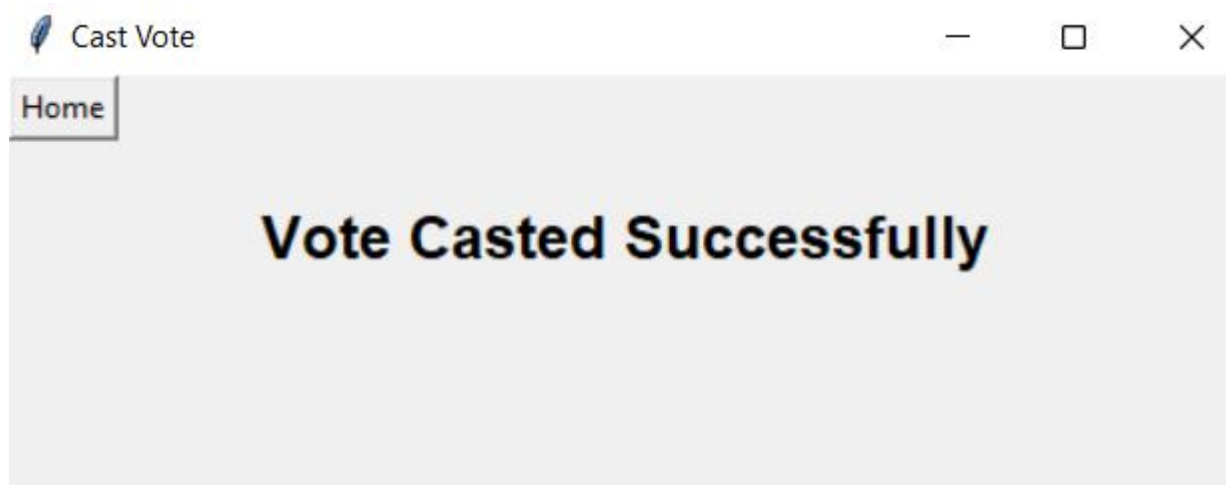


Voter Login

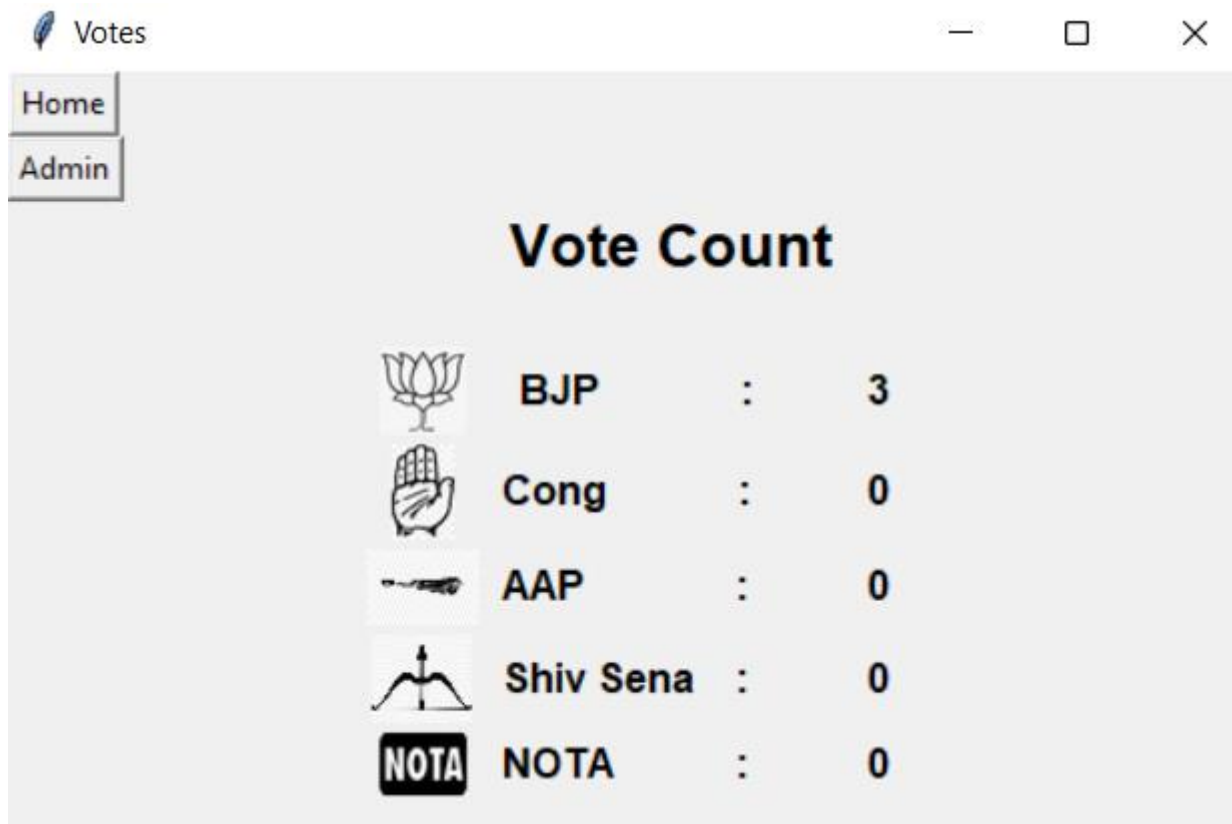
Test Case 1 : If detail matches, then it welcomes the voter and displays the name and poll symbol of the candidates.



Voting Page



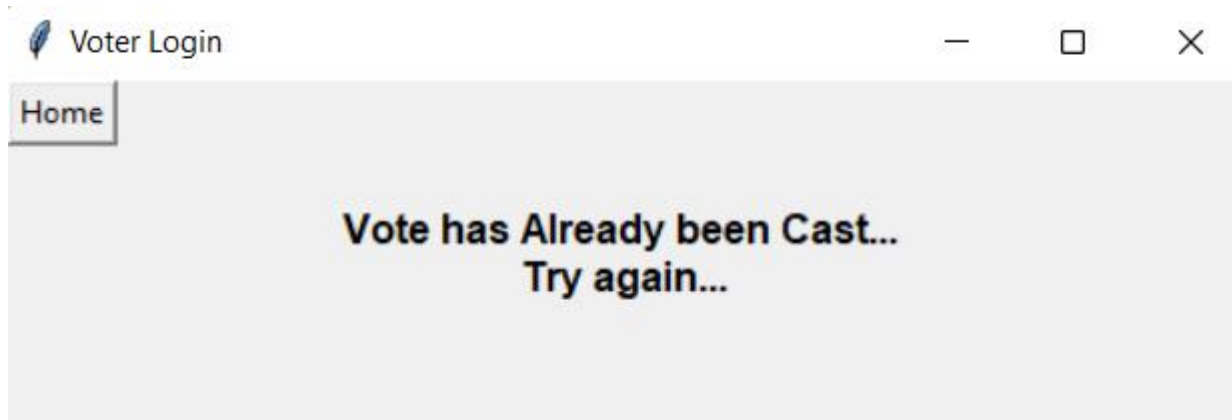
Vote Casted Successfully Message



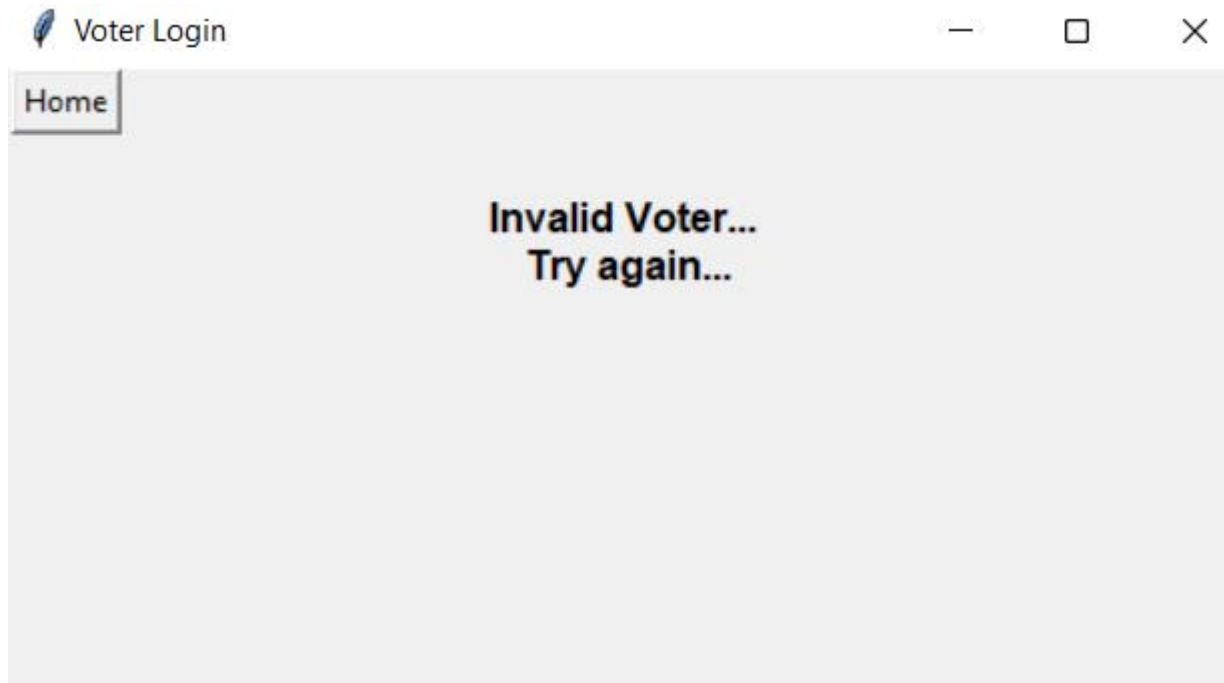
Show Votes

❖ **Error Handling:**

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



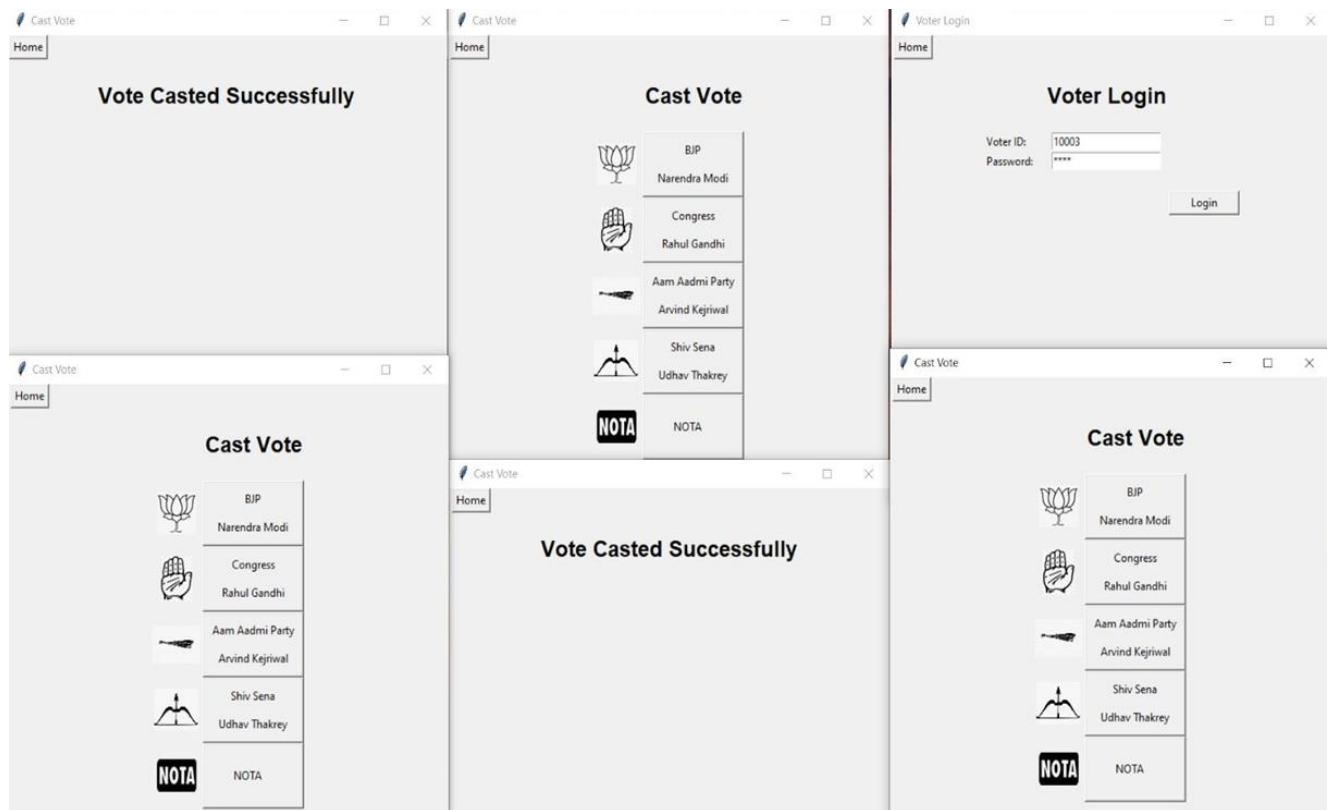
If the vote already been casted



If a voter is not registered/invalid voter

❖ Voter's casting vote concurrently:

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



6 Voters

SERVER OUTPUT

```
C:\Users\anike\Anaconda3\python.exe
Connected to : ('192.168.56.1', 61962)
Voter Logged in... ID:10001
Vote Received from ID: 10001 Processing...
Vote Casted Sucessfully by voter ID = 10001
Connected to : ('192.168.56.1', 61976)
Vote Already Cast by ID:10001
Vote Received from ID: 10001 Processing...
Vote Update Failed by voter ID = 10001
Connected to : ('192.168.56.1', 62000)
Invalid Voter
Vote Received from ID: 10001 Processing...
Vote Update Failed by voter ID = 10001
Connected to : ('192.168.56.1', 62020)
Voter Logged in... ID:10002
Vote Received from ID: 10002 Processing...
Vote Casted Sucessfully by voter ID = 10002
Connected to : ('192.168.56.1', 62029)
Voter Logged in... ID:10003
Connected to : ('192.168.56.1', 62040)
Voter Logged in... ID:10003
Vote Received from ID: 10003 Processing...
Vote Casted Sucessfully by voter ID = 10003
Connected to : ('192.168.56.1', 62044)
Voter Logged in... ID:10004
Vote Received from ID: 10004 Processing...
Vote Casted Sucessfully by voter ID = 10004
Connected to : ('192.168.56.1', 62045)
Voter Logged in... ID:10005
Vote Received from ID: 10005 Processing...
Vote Casted Sucessfully by voter ID = 10005
Connected to : ('192.168.56.1', 62062)
Vote Already Cast by ID:10001
Vote Received from ID: 10001 Processing...
Vote Update Failed by voter ID = 10001
```

DATABASE

	voter_id	Name	Gender	Zone	City	Passw	hasVoted
0	10001	Aniket Singh	Male	West	Saharanpur	ab@12	1
1	10002	Chandan Chirag	Male	East	Ranchi	ab@12	1
2	10003	Kavita Devi	Female	North	Mandi	ab@12	1
3	10004	Vardhan Kumar	Male	West	Roorkee	ab@12	1
4	10005	Yash Tyagi	Male	East	Agra	ab@12	1

Voter Info Database

	Sign	Name	Vote Count
0	bjp	Narendra Modi	7
1	cong	Rahul Gandhi	0
2	aap	Arvind Kejriwal	1
3	ss	Udhav Thakrey	0
4	nota	NOTA	0

Candidate Info Database

DISCUSSIONS

By completing this task, I could bring a new paradigm for online national voting in our country's favor with some more security of cryptography algorithms such as SHAKE-256. We could give fast service for casting a ballot framework to voters both within and outside the country via our digitally casting a ballot framework, thanks to the influx of new technologies and the Internet in our daily lives.

CONCLUSION

We learned how to implement TCP socket programming using Python from the 'E-voting system' project. We also learned how to connect multiple clients with one server. As the requirement of the project was to allocate a new thread by the 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.

The Online Voting Platform includes creative tickets, dazzling topic features, counting of votes, classifications, and disclosure. These resources are pre-programmed and should not be distributed to internal faculty. It also allows leaders to decide things on polls with the objective of ensuring that voters cannot cast illegitimate ballots and that they are not reviewed during tallying. The Portal for Online Casting a ballot is very least time-consuming and most useful method for both employer and voters. For employer, the process of creating a ticket and making a decision is simple and straightforward.

BIBLIOGRAPHY

1. S. B. Khairnar, P. S. Naidu, and R. Kharat, "Secure authentication for online voting system", *2016 International Conference on Computing Communication Control and automation (ICCUBE)*, pp. 1-4, 2016.
2. R. Bhuvanapriya, S. Rozil Banu, P. Sivapriya and V. K. G. Kalaiselvi, "Smart voting", *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 143-

147, 2017.

3. N. H. Sultan, F. A. Barbhuiya, and N. Sarma, "Pair Voting: A secure online voting scheme using Pairing-Based Cryptography and Fuzzy Extractor", *2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6, 2015.
4. S. Sridharan, "Implementation of authenticated and secure online voting system", *2013 Fourth International Conference Computing Communications and Networking Technologies (ICCCNT)*, pp. 1-7, 2013.
5. H. Agarwal and G. N. Pandey, "Online voting system for India based on AADHAAR ID", *Eleventh International Conference on ICT and Knowledge Engineering*, pp. 1-4, 2013.
6. L.Rura, B. Issac, and M. K. Haldar, "Online voting verification with cryptography and steganography approaches", *Proceedings of 2011 International Conference on Computer Science and Network Technology*, pp. 125-129, 2011.
7. V. Chinmay Vishal, R. Garg, and P. Yadav, "Online voting system linked with AADHAR", *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 3239-3240, 2016.
8. Ashwini Walake and Pallavi Chavan, "Efficient Voting system with Secret Sharing Based Authentication", *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 6, no. 1, pp. 410-412, 2015.
9. S. Katiyar, K. R. Meka, F. A. Barbhuiya and S. Nandi, "Online Voting System Powered by Biometric Security Using Steganography", *Second International Conference on Emerging Applications of Information Technology*, pp. 288-291, 2011.
10. K. Kim and D. Hong, "Electronic Voting System using Mobile Terminal" in *World Academy of Science Engineering and Technology*, pp. 33-37, 2007.
11. Buchsbaum, T. M. (2004). E-casting a ballot: International improvements and exercises learnt. *Electronic ballot in Europe Technology, Law, Politics and Society*, 31-34.
12. D. A. Kumar, T. Ummal, and S. Begum, "A novel design of electronic voting system using fingerprint," 2011.
13. Rabinadnan kishor, "Implementation of cloud for online election system," *International journal of advance research in computer science and management studies*, vol.3, March 2015.
14. Tadayoshi Kohno, Adam Stubblefield, "Analysis of an Electronic Voting System," *IEEE computer society press*, July 2003.

Flow Chart

