



**PERIYAR
MANIAMMAI**
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University)
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited

think • innovate • transform

MINI PROJECT : VOTING SYSTEM

NAMES

REGISTER NUMBER

INIYAN.P

123012019014

ABDUL RASIK

123012019033

MUHAMMAD SULTAN

123012019020

CLASS COURSE : II B SEC CSE SPEC WITH AI & ML

NAME : INTRODUCTION TO AI&ML

SUBJECT CODE : XCSHA1

VOTING SYSTEM PROGRAM

```
class VotingSystem:

    def __init__(self):

        self.candidates = {}

        self.voters = set()


    def add_candidate(self, name):

        if name not in self.candidates:

            self.candidates[name] = 0

        else:

            print(f"Candidate '{name}' already exists.")


    def cast_vote(self, voter_id, candidate_name):

        if voter_id in self.voters:

            print("You have already voted!")

            return

        if candidate_name not in self.candidates:

            print(f"Candidate '{candidate_name}' does not exist.")

            return

        self.candidates[candidate_name] += 1

        self.voters.add(voter_id)

        print(f"Vote cast successfully for '{candidate_name}'!")


    def display_results(self):

        if not self.candidates:

            print("No candidates available.")

            return
```

```
print("\nVoting Results:")

for candidate, votes in self.candidates.items():

    print(f"{candidate}: {votes} votes")


winner = max(self.candidates, key=self.candidates.get)

print(f"\nWinner: {winner} with {self.candidates[winner]} votes")


if __name__ == "__main__":

    voting_system = VotingSystem()


    while True:

        print("\n1. Add Candidate")

        print("2. Cast Vote")

        print("3. Display Results")

        print("4. Exit")


        choice = input("Enter your choice: ")


        if choice == "1":

            name = input("Enter candidate name: ")

            voting_system.add_candidate(name)


        elif choice == "2":

            voter_id = input("Enter your voter ID: ")

            candidate_name = input("Enter candidate name to vote for: ")

            voting_system.cast_vote(voter_id, candidate_name)


        elif choice == "3":

            voting_system.display_results()


        elif choice == "4":

            print("Exiting voting system. Thank you!")
```

else:

```
print("Invalid choice. Please try again.")
```

Overview

This program implements a simple voting system where:

1. Candidates can be added.
2. Voters cast their votes for specific candidates.
3. Results are displayed, showing vote counts for all candidates and the winner.

It prevents duplicate candidates and ensures each voter can vote only once

Key Components

1. `VotingSystem` Class

This class contains the main logic of the voting system.

`__init__()`:

- Initializes an empty dictionary `candidates` to store candidates and their vote counts.
- Initializes a set `voters` to track voter IDs and prevent duplicate votes.

`add_candidate(name)`:

- Adds a new candidate to the `candidates` dictionary with an initial vote count of 0.
- Prints a message if the candidate already exists.

`cast_vote(voter_id, candidate_name)`:

- Accepts a voter ID and the name of the candidate the voter wants to vote for.
- Checks if the voter has already voted by verifying `voter_id` in the `voters` set.
- Validates if the candidate exists in the `candidates` dictionary.
- Adds the voter ID to the `voters` set and increments the vote count for the selected candidate.

`display_results()`:

- Iterates through all candidates and their vote counts.
- Displays the total votes for each candidate.
- Declares the candidate with the maximum votes as the winner using the `max()` function.

Program Flow

1. **User Menu**:

- The program runs in a loop, presenting options to add candidates, cast votes, display results, or exit.

2. **Adding Candidates**:

- Users can add candidates by entering their names.
- Prevents duplicate candidate entries.

3. **Casting Votes**:

- Voters enter their voter ID and the name of the candidate they want to vote for.
- Ensures:
 - Each voter can vote only once.
 - Votes can only be cast for valid candidates.

4. **Displaying Results**:

- Shows the total votes for all candidates.
- Declares the winner based on the highest vote count.

5. **Exiting**:

- The user can exit the voting system anytime.

Example Walkthrough

Step 1: Adding Candidates

...

Enter your choice: 1

Enter candidate name: Alice

Enter your choice: 1

Enter candidate name: Bob

- Adds "Alice" and "Bob" to the candidate list.

Step 2: Casting Votes

...

Enter your choice: 2

Enter your voter ID: voter1

Enter candidate name to vote for: Alice

Vote cast successfully for 'Alice'!

...

- Voter ID `voter1` votes for "Alice."

- Adds `voter1` to the `voters` set and increments Alice's vote count.

Step 3: Displaying Results

...

Enter your choice: 3

...

Displays:

...

Voting Results:

Alice: 1 votes

Bob: 0 votes

Winner: Alice with 1 votes

...

Step 4: Exiting

...

Enter your choice: 4

...

The program exits.

Features

1. **Duplicate Voting Prevention**:

- Each voter is uniquely identified by their `voter_id`, ensuring no voter can vote twice.

2. **Duplicate Candidate Prevention**:

- Candidates are stored in a dictionary, preventing duplicate entries.

3. **Real-Time Results**:

- Users can view live voting results at any point.

Customizations

- **Tie Handling**: Add logic to handle cases where multiple candidates have the same number of votes.
- **Voter Registration**: Introduce voter registration to store details like name and ID.
- **Improved UI**: Use a graphical library (like Tkinter) for a better user interface.
- **Data Persistence**: Store candidates and votes in a database or file for reuse.

OUTPUT

```
1. Add Candidate
2. Cast Vote
3. Display Results
4. Exit
Enter your choice: 1
Enter candidate name: Alice
Enter your choice: 1
Enter candidate name: Bob
Enter your choice: 2
Enter your voter ID: voter1
Enter candidate name to vote for: Alice
Enter your choice: 2
Enter your voter ID: voter2
Enter candidate name to vote for: Bob
Enter your choice: 3
```

```
Voting Results:
Alice: 1 votes
Bob: 1 votes

Winner: Alice with 1 votes
```