Voting System Project

Harsh Pratap Singh
Roll No: K24ML-30
Reg. No. 12417799
Section: K24ML

Scenario: Voting System

Implement a voting system where users can vote for candidates. Use exception handling to manage cases where the user tries to vote for a candidate that doesn't exist (KeyError) or tries to vote multiple times (raise a custom exception).

Objective: Validate user input and prevent multiple voting using exceptions.

Source code

```
class MultipleVotingError(Exception):
    def __init__(self, message="You have already voted!"):
        self.message = message
        super().__init__(self.message)

class VotingSystem:
    def __init__(self, candidates):
        self.candidates = {candidate: 0 for candidate in
candidates}
        self.voters = set()

    def vote(self, user, candidate):
        if user in self.voters:
            raise MultipleVotingError(f"{user} has already
voted!")
        if candidate not in self.candidates:
            raise KeyError(f"Candidate '{candidate}' does
not exist.")
        self.candidates[candidate] += 1
        self.voters.add(user)
        print(f"{user} successfully voted for
{candidate}!")

    def get_results(self):
        return self.candidates
try:
    candidates_list = ["Alice", "Bob", "Charlie"]
    voting_system = VotingSystem(candidates_list)

    voting_system.vote("user1", "Alice")
    voting_system.vote("user2", "Bob")

    voting_system.vote("user1", "Charlie")

except KeyError as ke:
    print(f"Error: {ke}")
except MultipleVotingError as mve:
    print(f"Error: {mve}")
finally:
    print("Voting Results:", voting_system.get_results())
```

Output

```
user1 successfully voted for Alice!
user2 successfully voted for Bob!
Error: user1 has already voted!
```

```
Voting Results: {'Alice': 1, 'Bob': 1, 'Charlie': 0}
```