INSTITUTE OF PRINTING TECHNOLOGY
AND
GOVERNMENT POLYTECHNIC COLLEGE
SHORANUR

OPEN-ENDED PROJECT REPORT

SYSTEM ADMINISTRATION LAB

COURSE CODE: 5138

TOPIC :

VOTING SYSTEM

SUBMITTED BY:

ANANTHAKRISHNAN (2101131194) – 10

BHARATHCHANDHRAN T.S (2101131201) – 16

JISHNU A.P (2101131207) - 21

SUBMITTED TO :

BABY SALINI.C

LECTURER

DEPARTMENT OF COMPUTER ENGINEERING

IPT & GPTC SHORANUR

ABSTRACT

The provided Shell script implements a simple command-line voting system. It allows users to cast votes for predefined candidates, display election results, and exit the voting system. The script maintains a list of candidates and tracks the number of votes each candidate receives. Users can choose a candidate to vote for by selecting their candidate number from the menu. After casting votes, they can view the election results, showing the number of votes each candidate has received. The script provides a straightforward and interactive way to simulate a basic voting system using Shell scripting, making it suitable for small-scale elections or demonstrations.

The Voting System Project Using Shell Script is a versatile and adaptable solution that can be deployed in various environments, including universities, organizations, and communities, to conduct elections, surveys, and decision-making processes. It leverages the power of Shell scripting to automate and streamline the voting process while maintaining the highest standards of security and accuracy.

PROGRAM CODE :

```
#!/bin/bash

# Initialize variables to store candidates and their vote counts
candidates=("Candidate A" "Candidate B" "Candidate C")
votes=(0 0 0)

# Function to display the list of candidates
display_candidates() {
  echo "Candidates:"
  for ((i=0; i<${#candidates[@]}; i++)); do
    echo "$((i+1)). ${candidates[i]}"
  done
}

# Function to cast a vote
cast_vote() {
  display_candidates
  read -p "Enter the number of the candidate you want to vote for: " choice

  if [ $choice -ge 1 ] && [ $choice -le ${#candidates[@]} ]; then
    ((votes[choice-1]++))
    echo "You voted for ${candidates[choice-1]}"
  else
    echo "Invalid choice. Please enter a valid candidate number."
  fi
}
```

```
# Function to display the results
display_results() {
  echo "Election Results:"
  for ((i=0; i<${#candidates[@]}; i++)); do
    echo "${candidates[i]}: ${votes[i]} votes"
  done
}

# Main loop to manage the voting system
while true; do
  echo "Voting System Menu:"
  echo "1. Cast a vote"
  echo "2. Display results"
  echo "3. Quit"
  read -p "Enter your choice: " option

  case $option in
    1)
      cast_vote
      ;;
    2)
      display_results
      ;;
    3)
      echo "Exiting the voting system."
      exit 0
      ;;
    *)
      echo "Invalid option. Please select a valid option."
      ;;
  esac
done
```

<u>OUTPUT</u> :

```
Voting System Menu:
1. Cast a vote
2. Display results
3. Quit
Enter your choice: 1
Candidates:
1. Candidate A
2. Candidate B
3. Candidate C
Enter the number of the
candidate you want to vote
for: 2
You voted for Candidate B

Voting System Menu:
1. Cast a vote
2. Display results
3. Quit
Enter your choice: 1
Candidates:
1. Candidate A
2. Candidate B
3. Candidate C
Enter the number of the
candidate you want to vote
for: 1
You voted for Candidate A
```

```
Voting System Menu:
1. Cast a vote
2. Display results
3. Quit
Enter your choice: 2
Election Results:
Candidate A: 1 votes
Candidate B: 1 votes
Candidate C: 0 votes

Voting System Menu:
1. Cast a vote
2. Display results
3. Quit
Enter your choice: 3
Exiting the voting system.
```