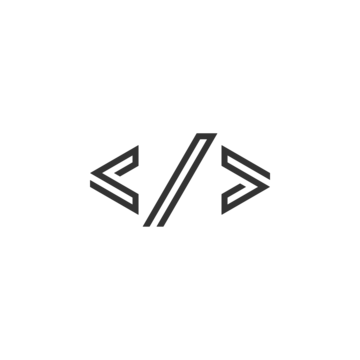


**Introduction**

School council elections are an essential part of fostering leadership and decision-making skills among students. Traditionally, these elections have been conducted manually, which can be time-consuming, error-prone, and less engaging for students in today's digital age. To address these challenges, this project introduces a Python-based voting system tailored for school council elections.

This system is designed to streamline the election process, ensuring efficiency, accuracy, and transparency. It provides a user-friendly interface for casting votes, managing candidates, and displaying results in real-time. By leveraging the simplicity and power of Python, this solution aims to modernize the voting process, making it accessible and reliable for schools of all sizes.

The project not only demonstrates practical programming skills but also highlights the importance of technology in solving real-world problems. This voting system serves as a step toward integrating digital solutions into everyday school activities, enhancing the overall experience for students and staff alike.



**User Manual**

**Overview**

This Python-based voting system is designed to conduct school council elections efficiently and transparently. The system consists of multiple scripts that handle candidate registration, post segregation, voting, and results compilation. This document explains the usage of each component and guides you through the election process.

**System Requirements**

1. Python 3.x installed on your system.
2. A text editor or IDE for running Python scripts.
3. Required CSV files will be generated automatically during execution.

**Main Menu**

To start, execute the main\_menu.py script. The main menu provides the following options:

|  |  |
| --- | --- |
| **Option** | **Description** |
| 100 | Read the procedures of PVSSCE. |
| 101 | View the list of candidates. |
| 102 | View the election results (only available after voting). |
| x | Quit the program. |

**Step-by-Step Instructions**

**Step 1: Register Candidates**

1. Run the v1\_candidates.py script.
2. Enter the number of candidates.
3. For each candidate, provide the following details:
   * Name
   * Class and Section
   * House
   * Post Applied (Choose from available options such as Head Boy, Head Girl, etc.)
4. Candidate information will be saved in candidates.csv.

**Step 2: Segregate Candidates by Post**

1. Run the v2\_segregation\_of\_posts.py script.
2. The system will segregate candidates into separate CSV files based on their applied posts.
   * Examples: hb.csv for Head Boy, hg.csv for Head Girl, etc.

**Step 3: Conduct Voting**

1. Run the v3\_voting.py script.
2. Enter the total number of voters.
3. For each voter:
   * Vote for candidates in each post by choosing the candidate's number.
   * You can also select "None of the Above (NOTA)" by entering 0.
4. Votes are tallied, and results are saved in result.csv.
5. After voting is complete, election coordinators can access the final vote count by entering the correct password.

**Step 4: View Results**

1. Run the v4\_results.py script.
2. View detailed results, including:
   * Candidate information.
   * Vote count for each candidate.
   * Results are displayed on the screen and saved in result.csv.

**Additional Information**

* Ensure all scripts are run in sequence.
* Do not modify or delete generated CSV files during the election process.
* Default election coordinator password: 9876543210 (can be changed in the v3\_voting.py script).