# Module 3: PyPoll Challenge

## Deliverable Requirements

|  |  |
| --- | --- |
| **Rqmt ID** | **Description** |
| 1.0 | Election Results Printed to the Command Line: Using repetition statements, conditional statements with logical operators, and print statements, print out the candidate and county election results to the command line. |
| 1.1 | Candidate Results |
| 1.1.1 | Total Votes in the election are printed to the terminal. |
| 1.1.2 | Each candidate’s total votes and percentage of votes are printed to the terminal. |
| 1.1.3 | The winner of the election, winning vote count, and winning percentage of votes are printed to the terminal. |
| 1.2 | County Results |
| 1.2.1 | Each county and its total vote count are printed to the terminal. |
| 1.2.2 | Each county and its percentage of the total votes are printed to the terminal. |
| 1.2.3 | The county with the largest number of voters is printed to the terminal. |
| 2.0 | Election Results Saved to a Text File: Using your knowledge of writing data to a text file, write the winning candidate results and the county election results to the election\_results.txt file. |
| 2.1 | Candidate Results |
| 2.1.1 | Total Votes in the election are saved in the election\_results.txt file. |
| 2.1.2 | Each candidate’s total votes and percentage of votes are saved in the election\_results.txt file. |
| 2.1.3 | The winner of the election, winning vote count, and winning percentage of votes are saved in the election\_results.txt file |
| 2.2 | County Results |
| 2.2.1 | Each county and its total vote count are saved in the elections\_result.txt file. |
| 2.2.2 | Each county and its percentage of the total votes are saved in the elections\_result.txt file. |
| 2.2.3 | The county with the largest number of voters is saved in the elections\_result.txt file. |
| 3.0 | Written Analysis of the Election Audit: Use your repository README to write your analysis of Deliverables 1 and 2. |
| 3.1 | Structure, Organization, and Formatting |
| 3.1.1 | There is a title, and there are multiple sections. |
| 3.1.2 | Each section has a heading. |
| 3.1.3 | Links to images are working, and code is formatted and displayed correctly. |
| 3.2 | Analysis |
| 3.2.1 | Overview of Election Audit: The purpose of this election analysis audit is well defined. |
| 3.2.2 | Election Audit Results: There is a bulleted list where each election outcome is addressed. |
| 3.2.3 | Election Audit Summary: There is a statement to the election commission that explores how this script can be used for any election, with two examples for modifying the script. |
| 4.0 | Submission |
| 4.1 | Upload the following to your Election\_Analysis GitHub repository. |
| 4.1.1 | The PyPoll\_Challenge.py file at project root. |
| 4.1.2 | The analysis folder with the election\_results.txt file. |
| 4.1.3 | The Resources folder with the election\_results.csv file. |

## Goals (Rqmts 1.0 – 2.0)

Pseudocode/Algorithm

|  |  |
| --- | --- |
| **Candidate Flow** | **County Flow** |
| 1) Open the data file. | 1) Open the data file. |
| 2) Get the names of all the candidates. | 2) Write down the names of all the counties. |
| 3) Add a vote count for each candidate. | 3) Add a vote count for each county. |
| 4) Get the total votes for each candidate. | 4) Get the total votes for each county. |
| 5) Get the total votes cast for the election. | 5) Get the total votes cast for the election. |
| 6) Determine the winning count. | 6) Get the county percentage contribution to the election. |
| 7) Determine the winning candidate. | 7) Determine the largest county turnout. |
| 8) Determine the winning percentage. | 8) Print the county name, vote percentage, and total vote. |
| 9) Print all candidates and their final votes. | 9) Print the largest county turnout. |
| 10) Print the winning candidate, their final vote and the winning percentage. |  |

### Flow/Variable Mapping

