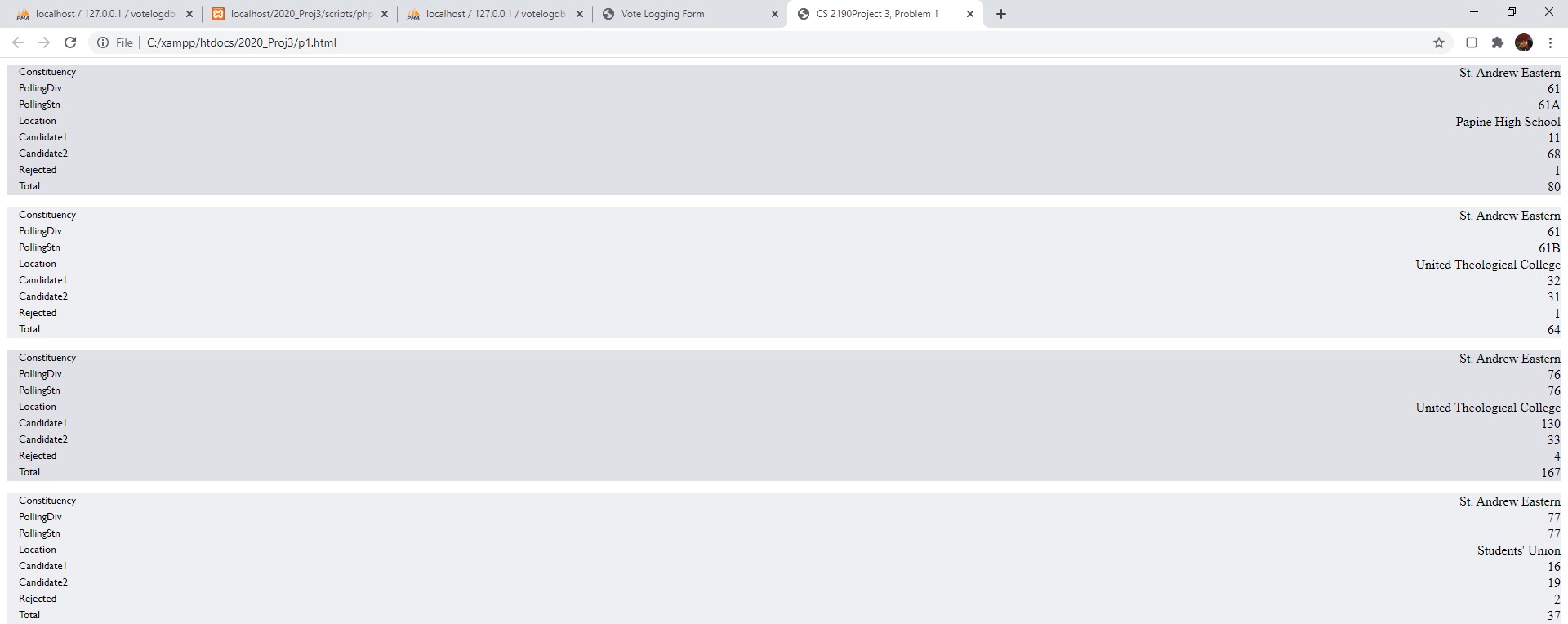
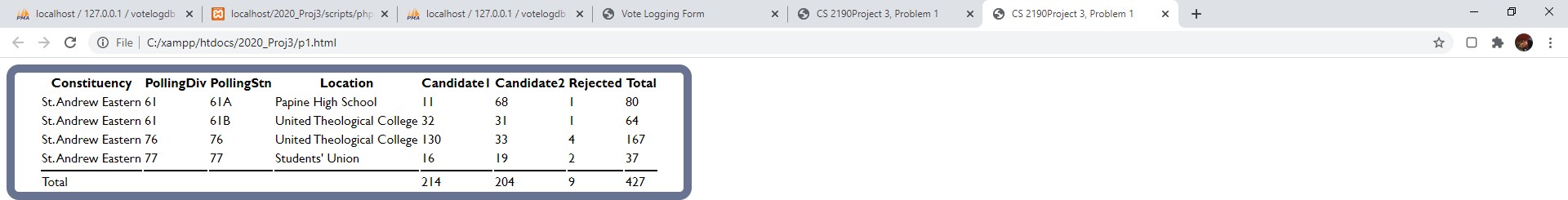
Web Programming Design Document

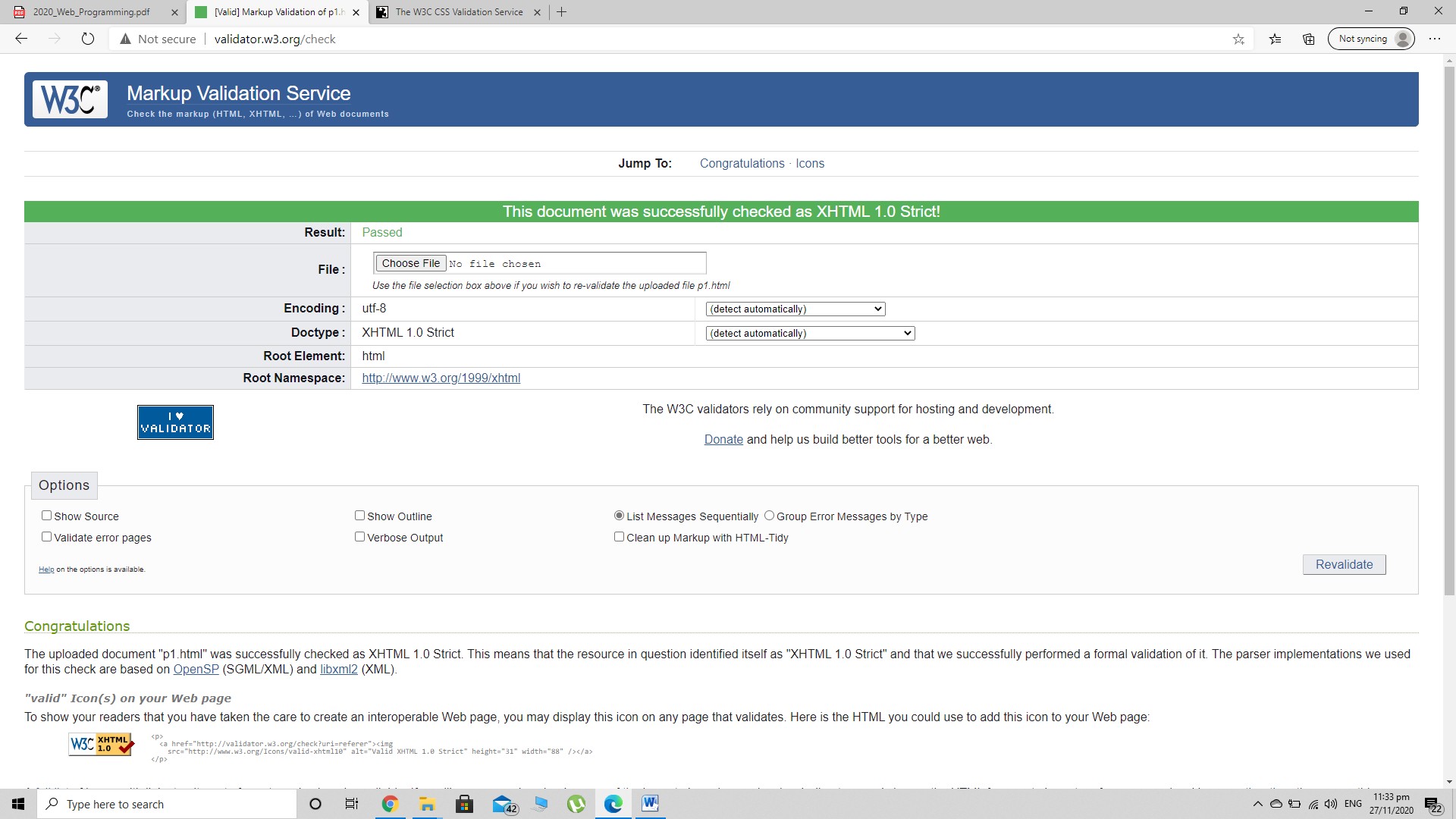
**By: Glenroy Logan**



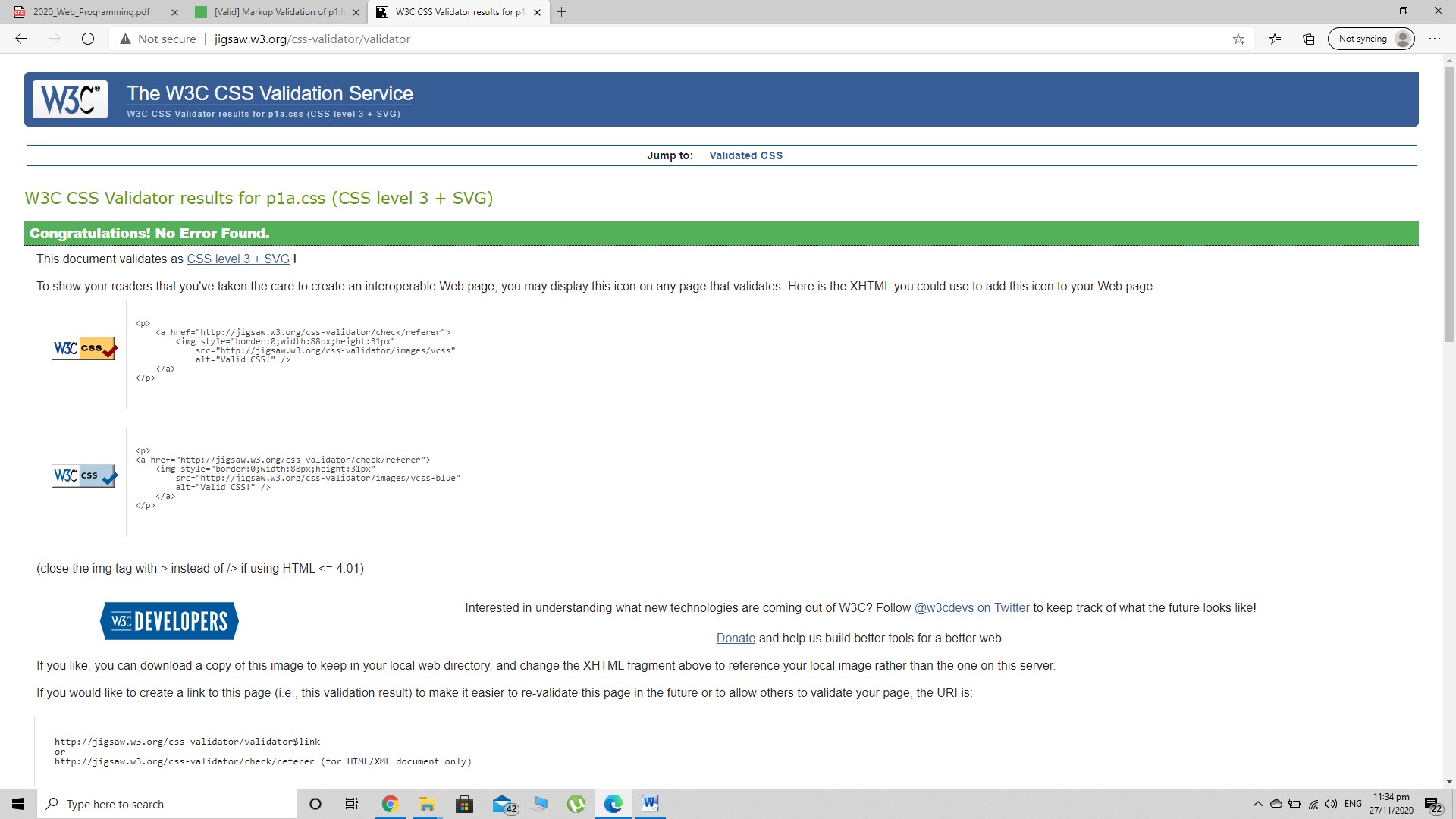
P1a: The screenshot above displays all the headings on the left and the data for each heading on the right



P1b: The screenshot above displays all the headings above in bold and the data for each heading on the below as well as the total for each candidates as well as the totals overall.

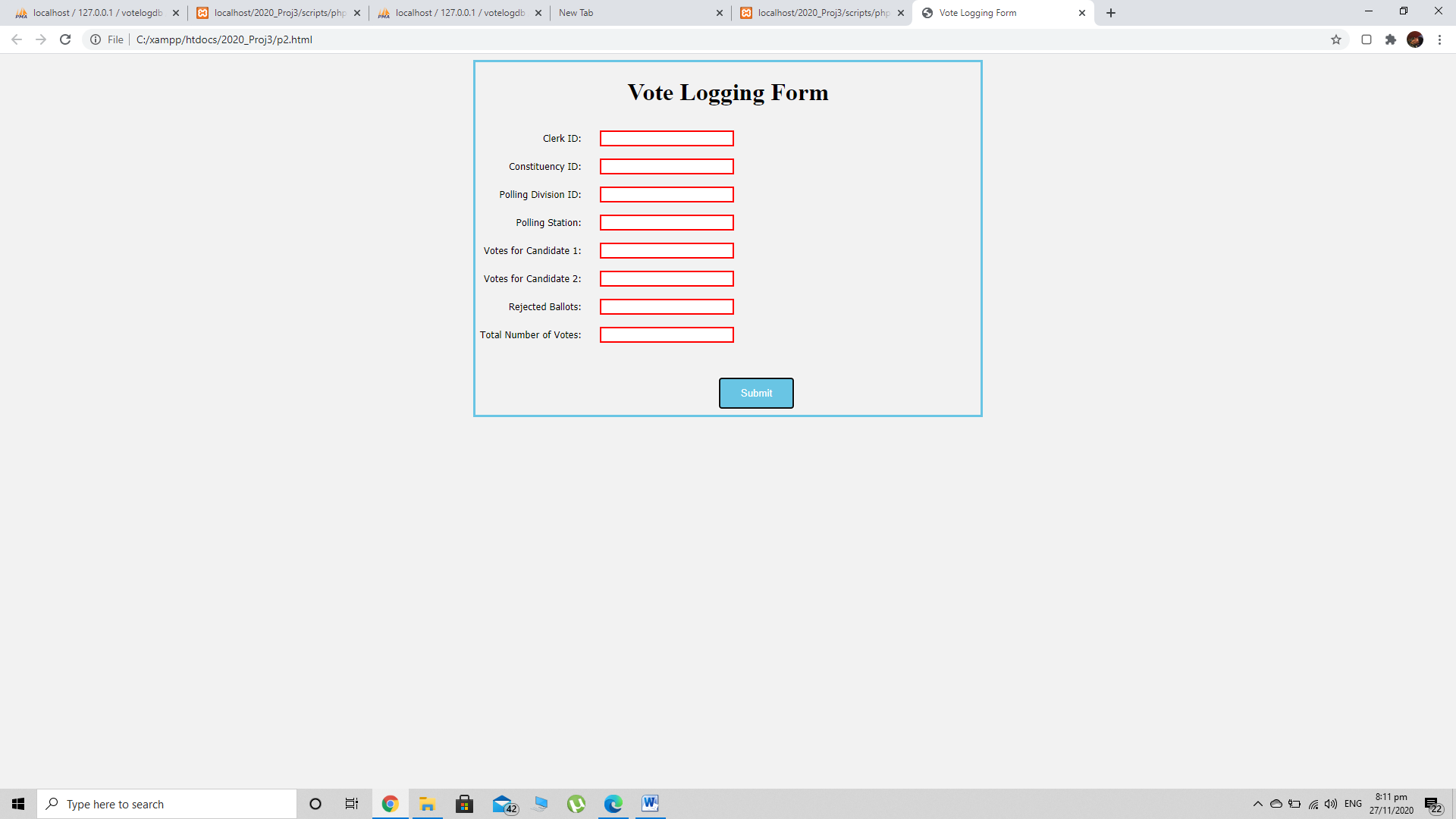


Screenshot showing validation test for html files

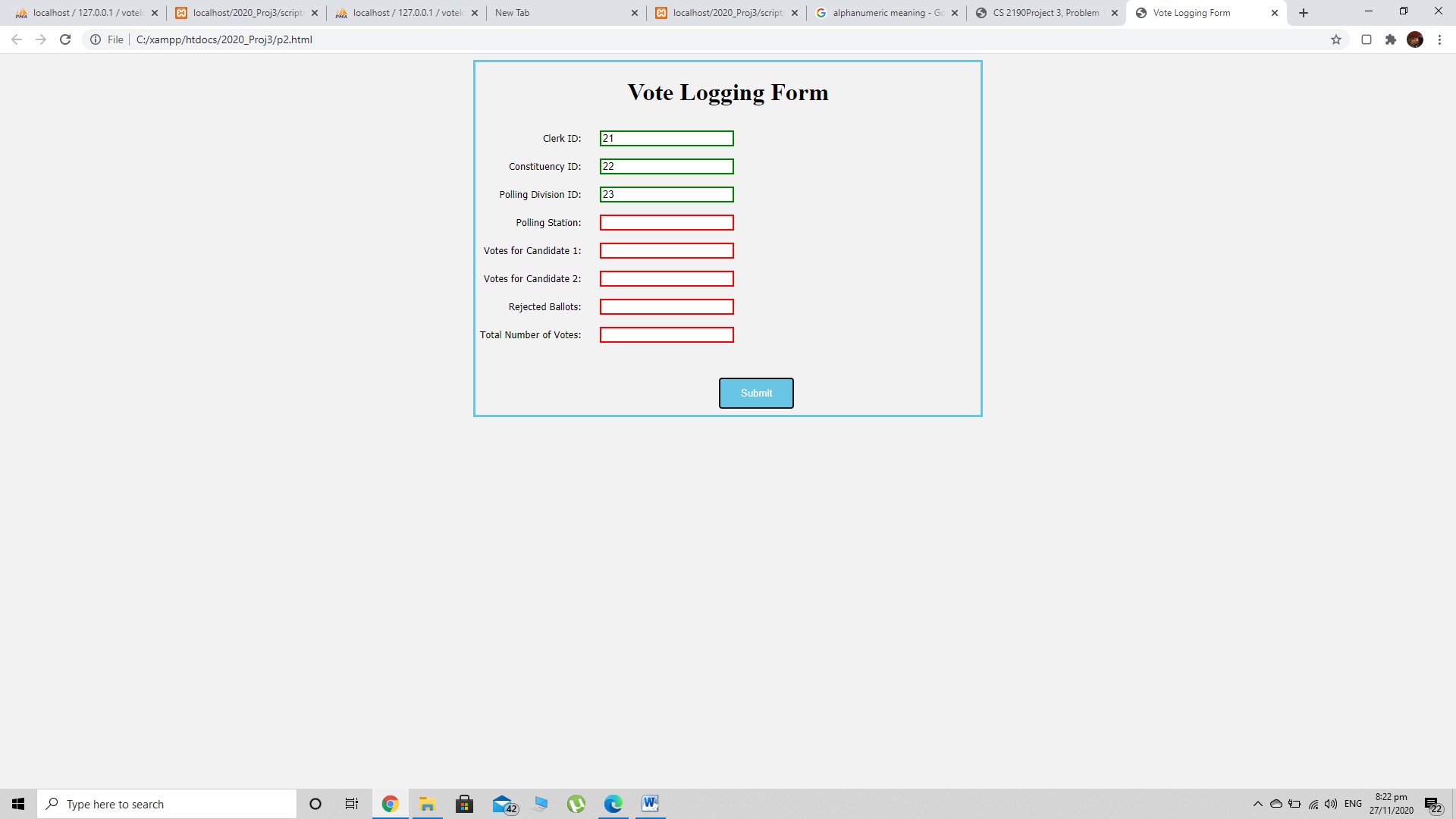


Screenshot showing validation for css files.

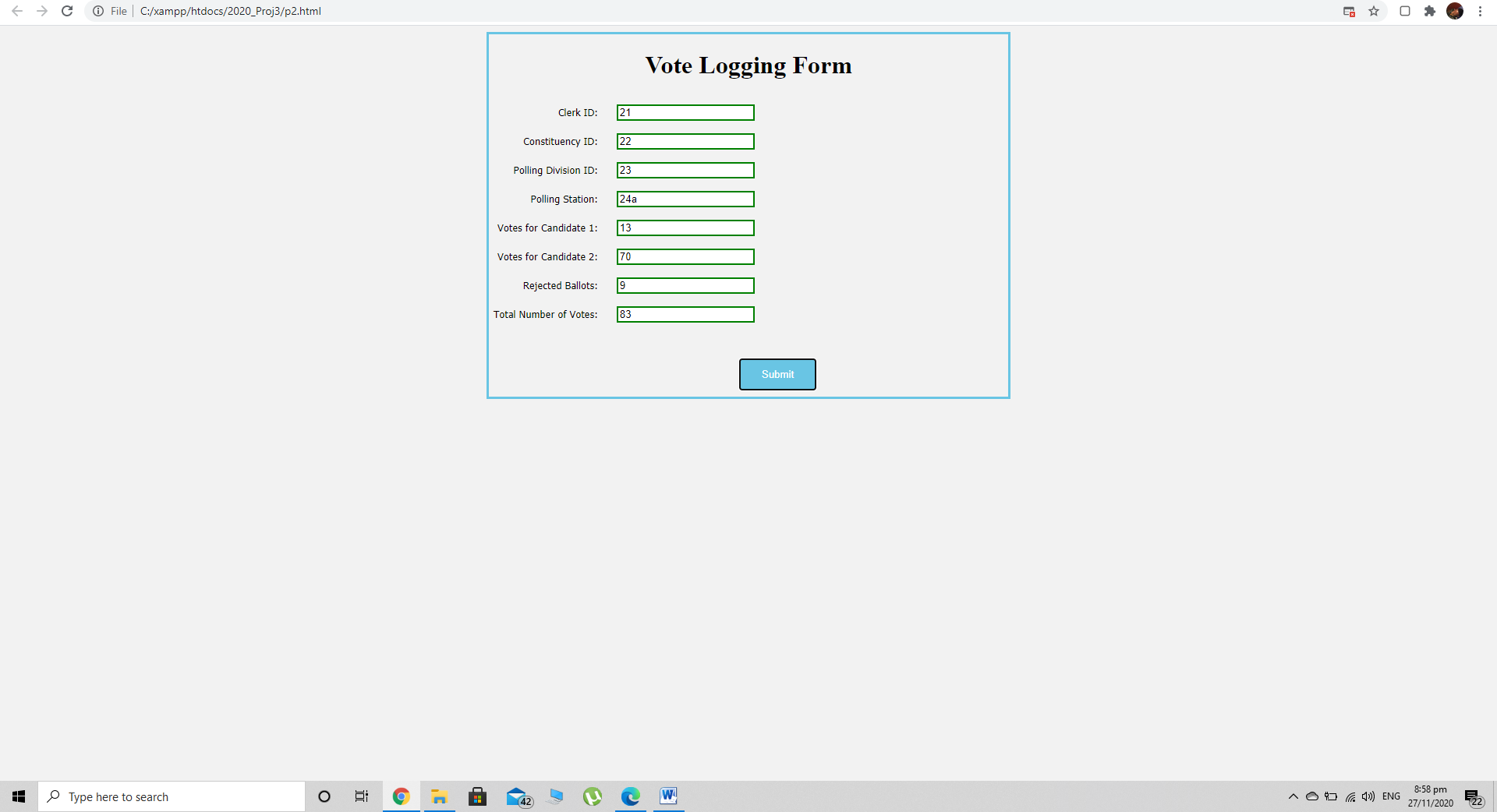
# Test Plan

Test plan for validation input

Screenshot 1 Checks the validation to see if the fields listed are empty and if so will display a red border around the fields indicating to the user that more data needs to be entered



Screenshot 2 Shows what happens when a user inputs valid data but still has not completely filled out the remaining fields if the correct data type for each field is entered correctly then a green field is displayed to indicate to the user that they have entered the data correctly



In Screenshot 3 This shows where a user enters data into the respective fields including the Polling Station where the input accepts alphanumeric values as seen with the entry of “24a”. All entries have been completed and filled out properly as indicated by the green borders around the boxes. The form will then accept the entries of the user after they click the submit button.

**Scripts Files Explained**

# Overview

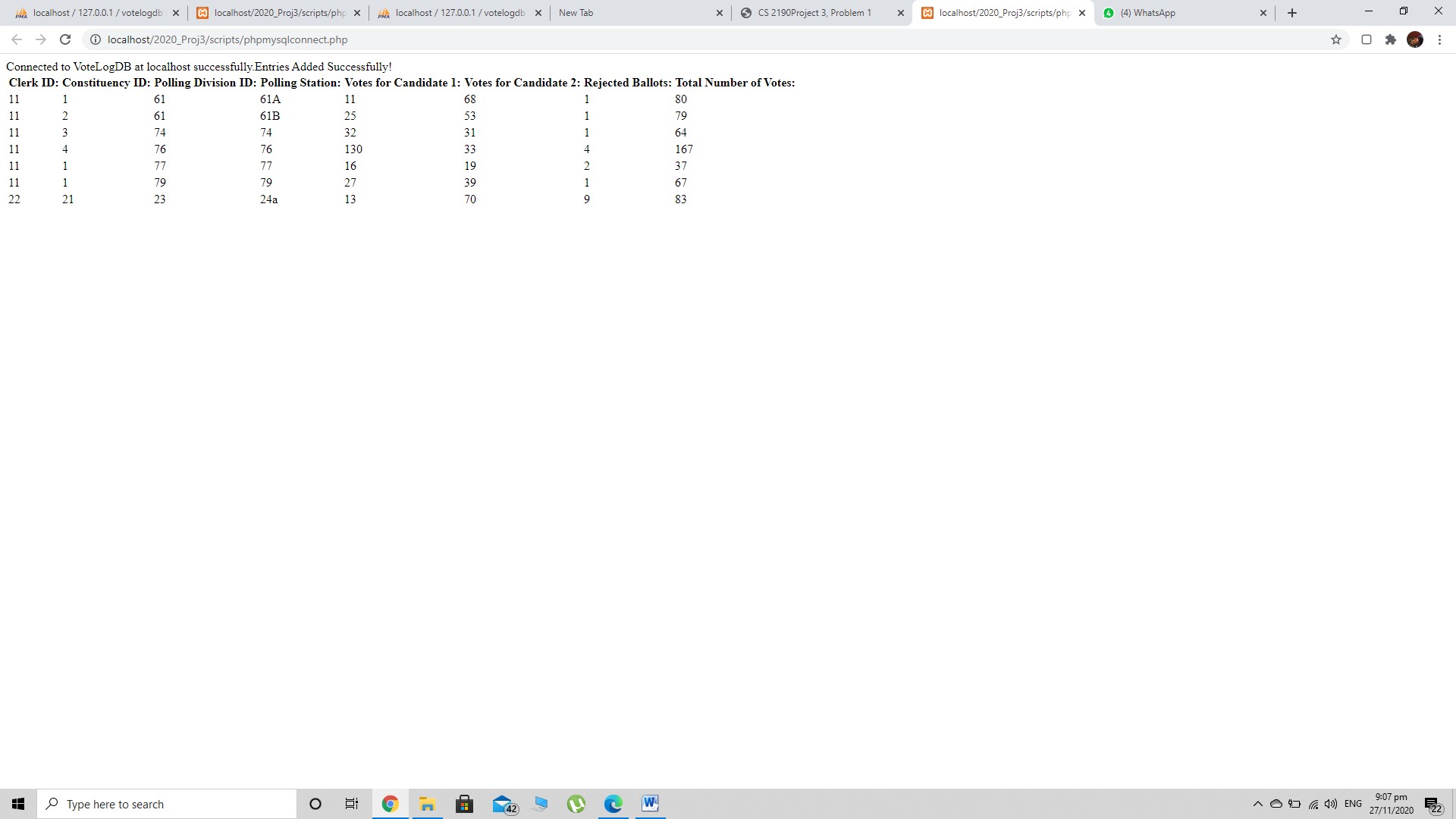
The project requires the use of external PHP scripts to validate user input from a HTML form then insert this data into a xampp database using an apache server if it is valid.

# How it works

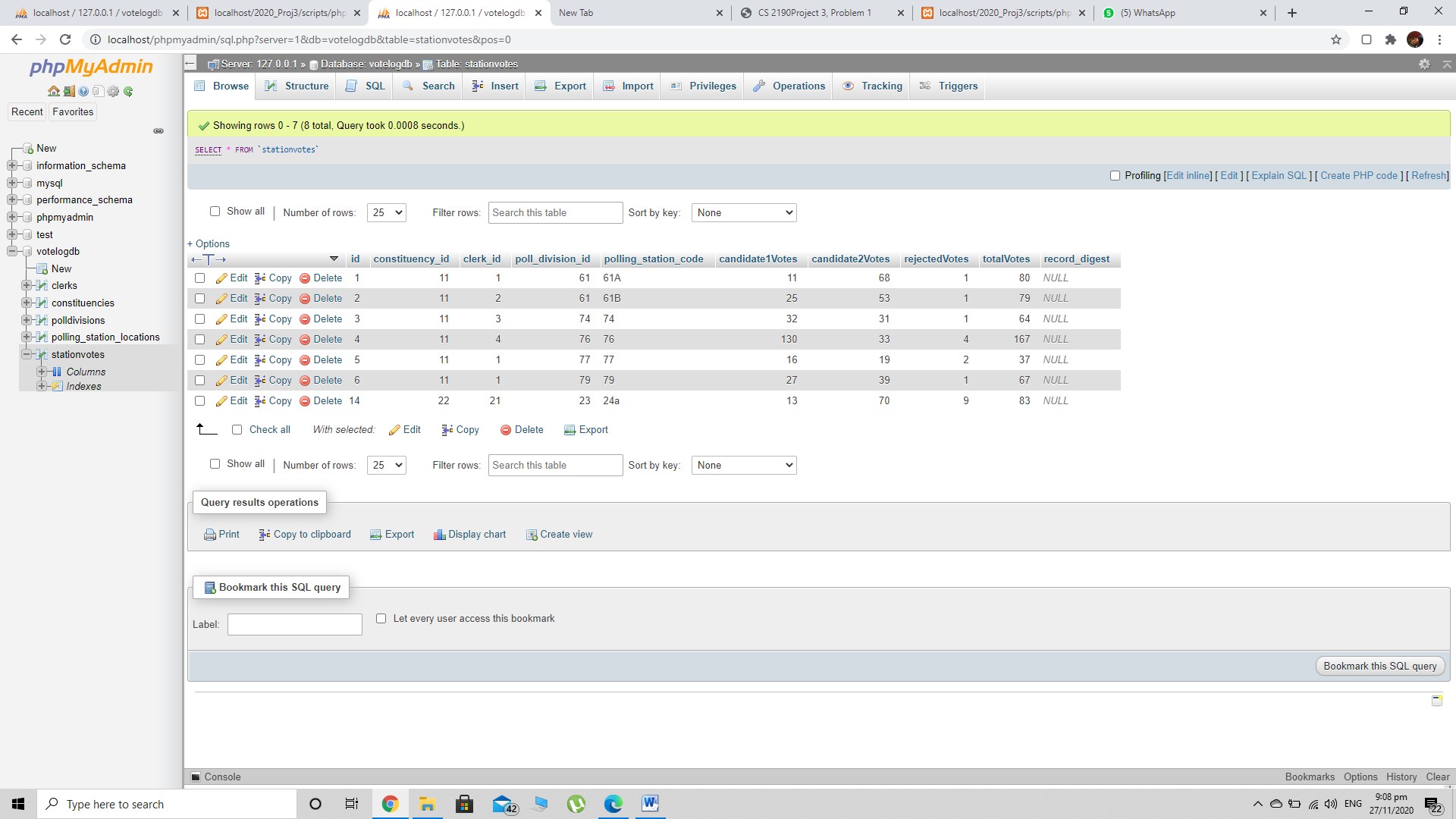
The HTML form displays various input text fields to gather data from a user who is looking to create various voting logging information entries into a system. This data submitted has both client and server side validation. After the data passes the client-side Javascript validation the formis submitted to a PHP script. The script then extracts the data from the super global “\_POST” variable in PHP and places it into a custom ‘form’ object. The script then validates several fields of the data such as:

* 1. None of the fields listed above is empty.
* 2. All the fields except for the polling station are integers.
* 3. The polling station field contains alphanumeric values, i.e., [A-Z,0-9]
* 4. The total votes field must be equal to the sum of the votes for candidate 1, candidate 2, and the rejected votes.

Before the data is validated a connection to the mySQL database is established and validation commences if the connection was successful. After validation, if the data passes all tests, all fields are stored in the “StationVotes” table in the “VoteLogDB” database on the xampp server hosted locally. In order to execute SQL commands the PHP object-oriented ‘PDO’ methods were used to connect and “INSERT” data into the table. After validation and possible insertion into the database the connection between the php script and xampp database is closed.



The screenshot above indicates that the entry from the form section was successful as well as the contents of the database were printed out in a table where the latest entry from the user was added at the last row.



In the final screenshot this displays the latest entry into the xampp database that was entered by the user in the form section as seen in the last row in the StationVotes table