The algorithm that I chose to do my project on was the Manacher Algorithm and the author of the project is Kylle Waldie. The algorithm is used to find the longest palindromic substring. It works by splitting the input string and adding in a symbol between each letter and at the beginning and the end and storing it as an array. In this case the symbol was a ‘#’. The algorithm then iterates through the array and checks if the left and right side are similar and stores the longest palindrome. We then reform the old string by removing the symbol and output the string to the consol or form.

The algorithm was discovered by Glenn Manacher in 1975, He published it in a paper called “A Linear-Time ‘On-Line’ Algorithm for Finding the Smallest Initial Palindrome of a String” The algorithm is O(n) or linear time complexity because it processes each character of the string once.

The program opens up into a menu that allows you to either click a start button to continue on with the program or an exit button to close the window and exit the program. When you click the start button it closes the menu while also bringing you to another window where you are able to enter a string to find the longest palindrome in it. If there is not string input then a new window pops up declaring that you did not input a string while also playing the Pedro racoon gif. I added this because I thought it would be funny. When you input a string it calls the function string longestPalindrome = ManacherAlgorithm.LongestPalindrome(userInput);. This goes through the ManacherAlgorithm class and calls the LongestPalindrome function passing the user input as the argument. It then out makes the second box visible and outputs the palindrome in it so the user can see it. If you want to test another string. You can type it into the box again and it clears the second box to output the new string.

To exit the program you simply click the Quit buttons or you can close out of them with the X in the corner.