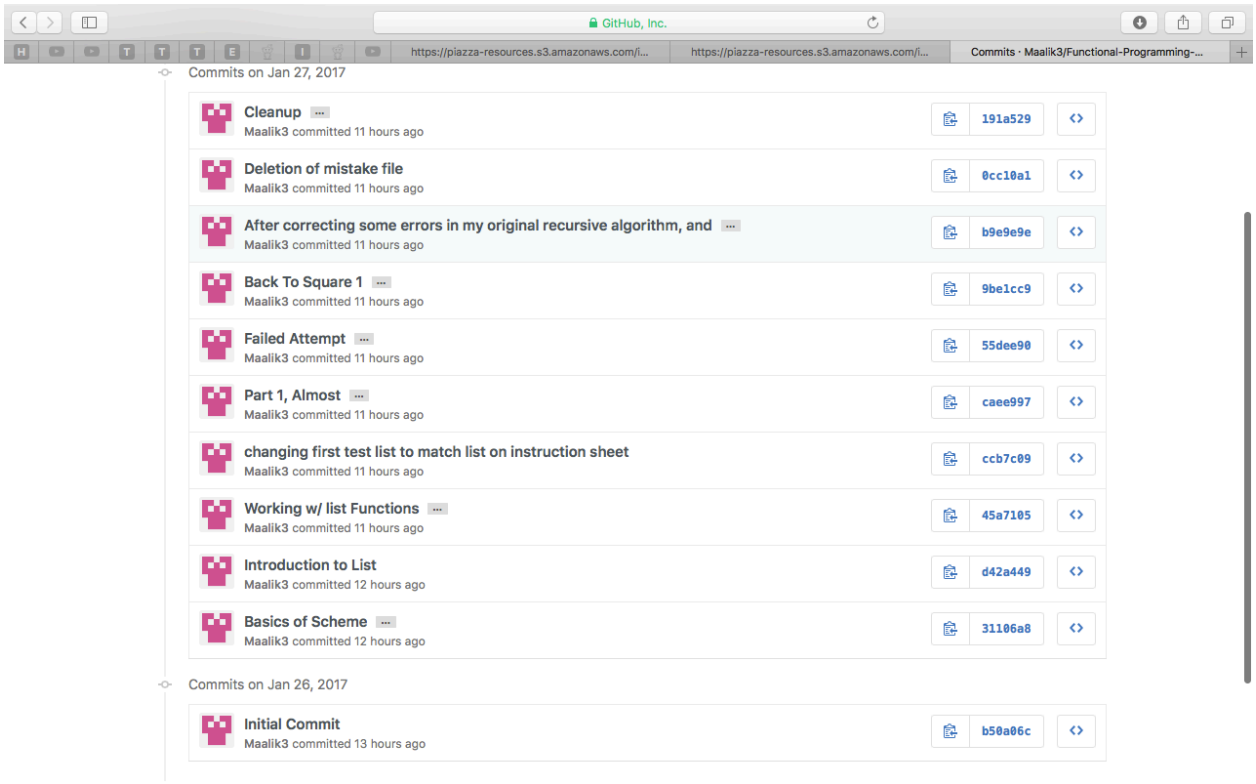
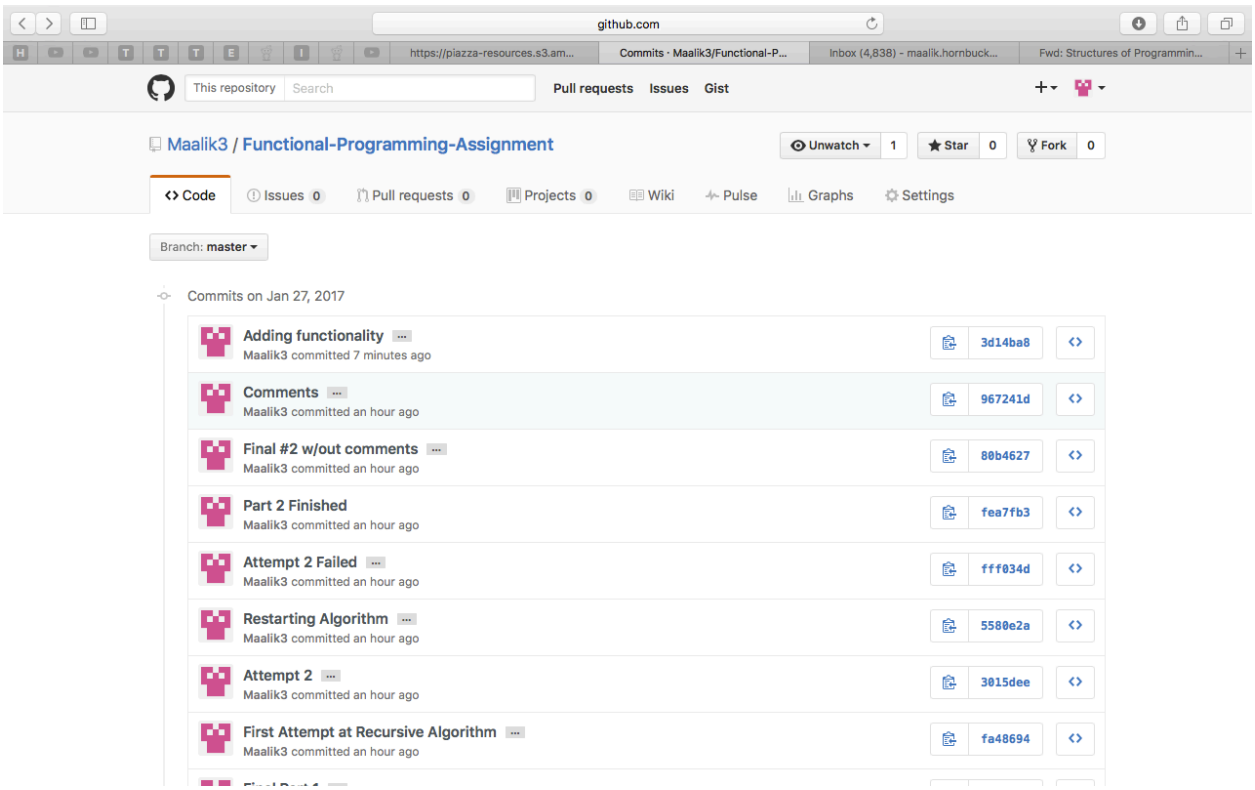
https://github.com/Maalik3/Functional-Programming-Assignment

1.



- **Basics of Scheme** - Practicing basic concepts of scheme: arithmetic & functions
- **Working w/ list Functions -** Working with list functions to obtain an understanding of CAR, CONS, CDR
- **Part 1, Almost-** Understand the ideology of reversing the list. We must recursively pop
- the beginning of the list and then add it to the end of the list
- **Failed Attempt -** Failed attempt to reverse list
- **After correcting some errors in my original recursive algorithm -** adding an if statement to catch empty list, I was able to successfully reverse the list by adding the first element to the tail of the list recursively
- **Cleanup -** Removed CDR and CAR practice. Created test list to match assignment. Added comments. Ran Test Cases
- **Final Part 1 -** Removal of Test Cases

2.



- **First Attempt at Recursive Algorithm -** My first attempt to recursively sum the list is returning an error "mcdr: contract violation expected: mpair? given: 1"
- **Attempt 2 -** Created if else statement in function to c2atch empty list. Still getting error, "expected mpair"
- **Final #2 w/out comments -** Using list (1,2,3), I popped the first item in the list using CAR (1) then added it to the second item of the list by using CDR to take the list without the first item (2,3) and recursively calling function (sum-up-general) to retrieve first item (2)
- **Comments -** Added comments to explain idealogy of algorithm
- **Adding functionality -** Changed if else to handle non numeral items, as well as adding conditional statement if list is null