Logan Hammond

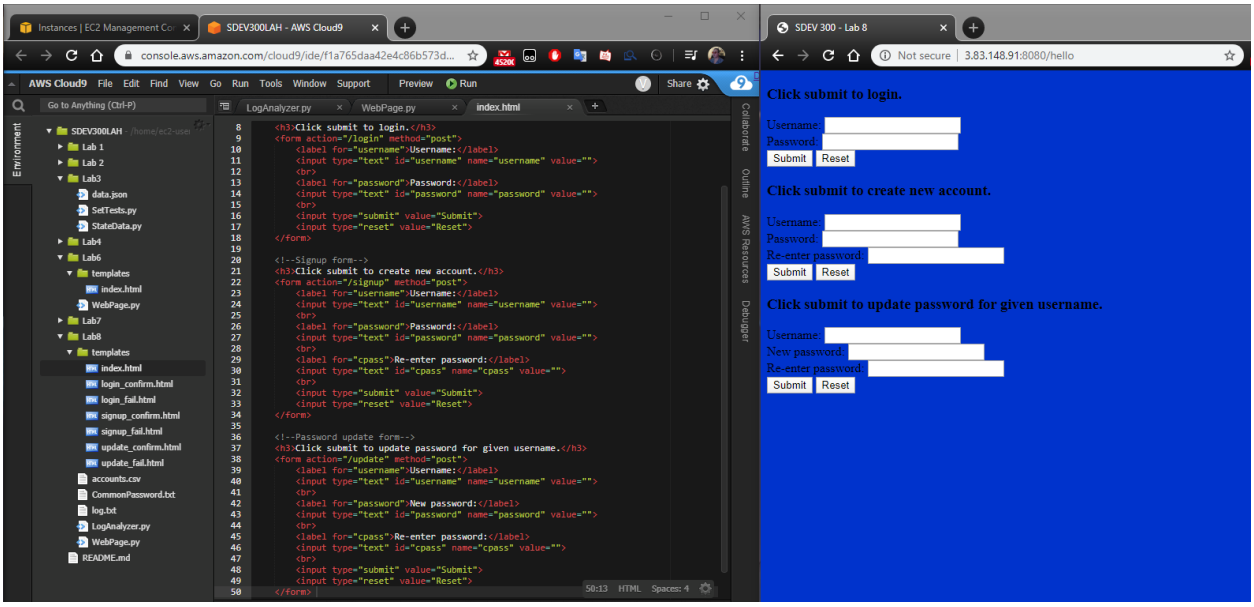
3 March 2020

SDEV 300, Fair

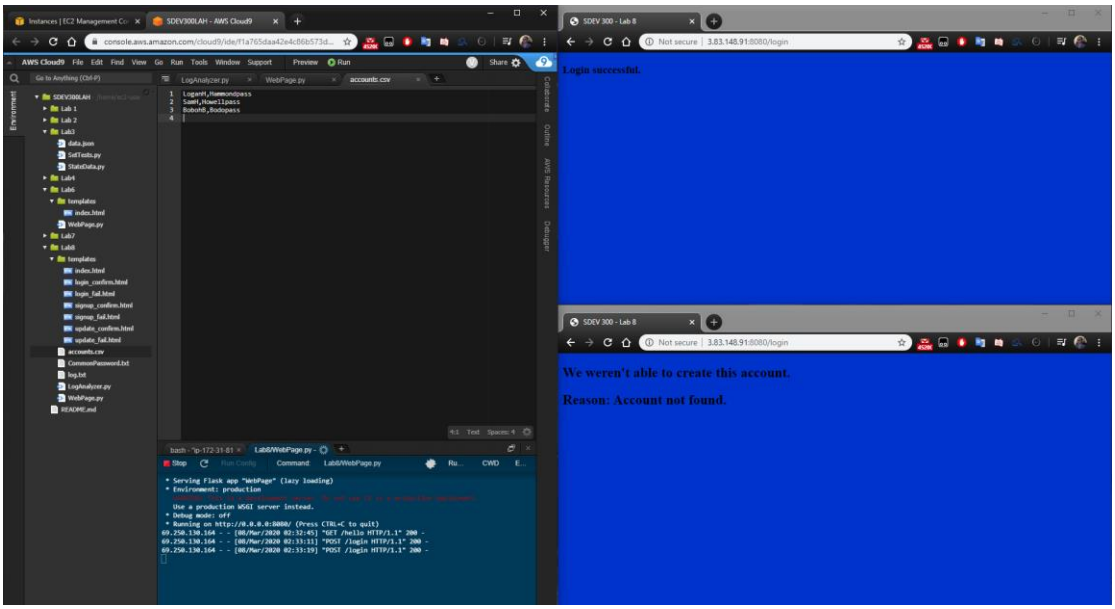
Lab 8 Results Document

Website

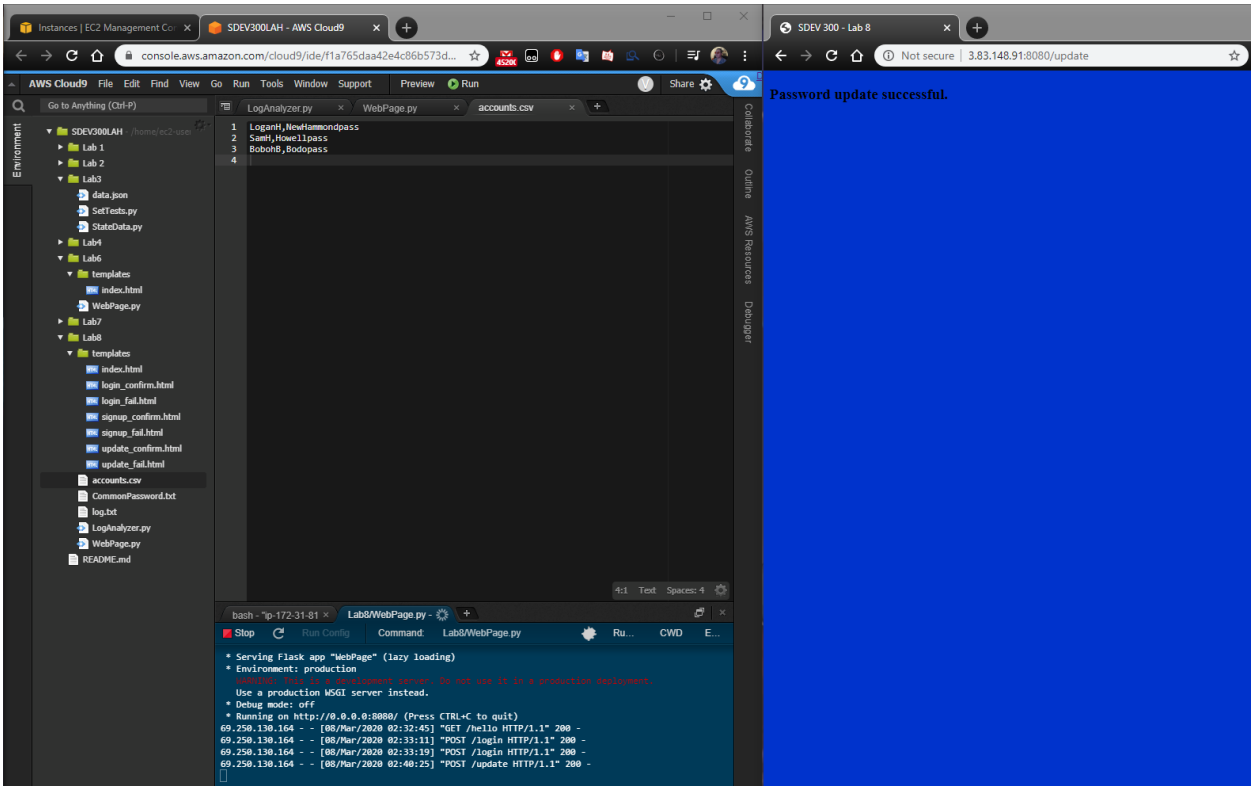
A – B



Main page of website (/hello) shows 3 forms to the user: a login form, a register form, and a password update form. Left image shows the index.html file centered on these forms.

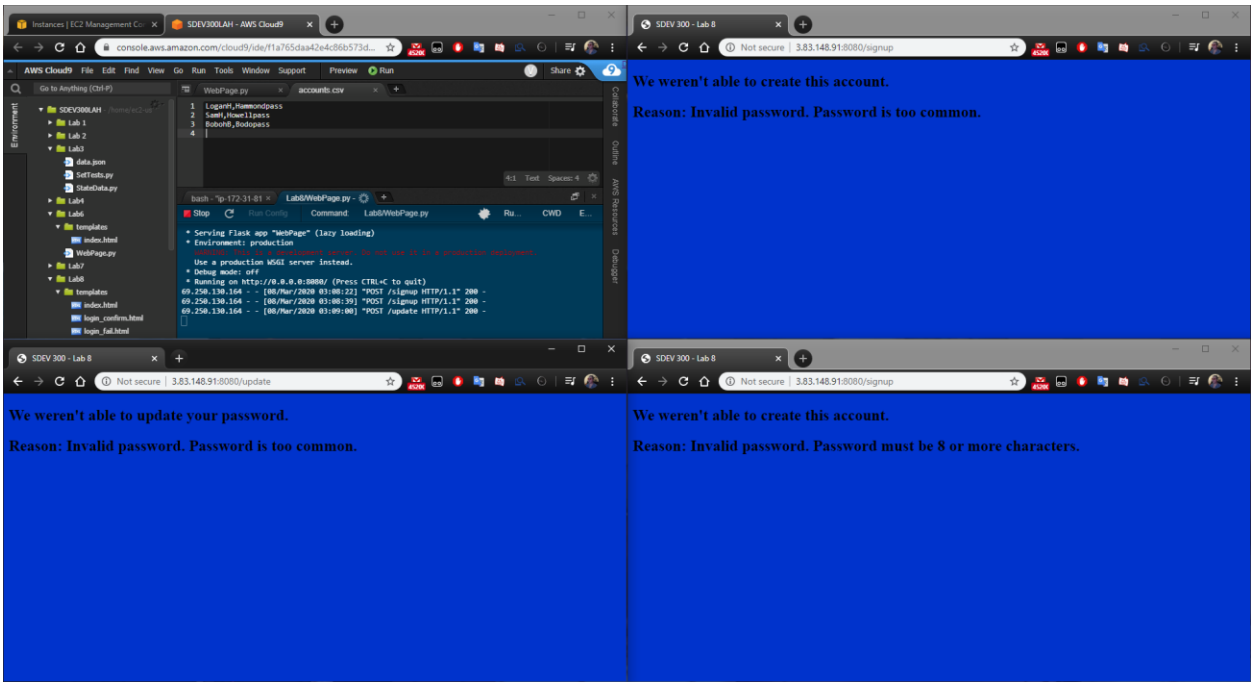


Submitting the first form will allow the user to login with a set of credentials. If the credentials are found in the master list of accounts (accounts.csv) then the user is redirected to a separate page confirming the login was successful. If the credentials were not found the user is submitted to a page alerting them that the credentials were not valid i.e. the account was not found in the master list of accounts. The credentials used were “LoganH” and “Hammondpass” for the username and password respectively.

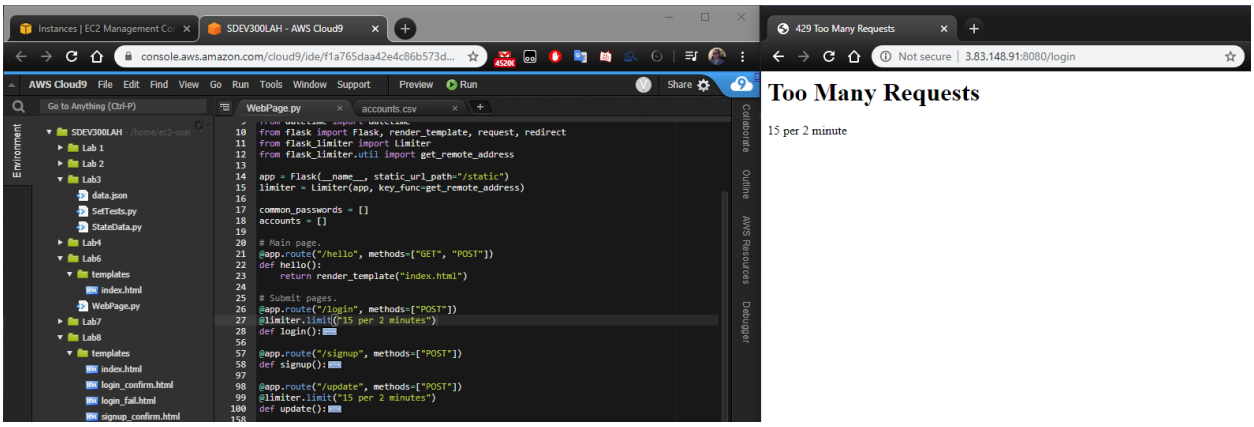


From the homepage a user may submit their username and a new password for their account. When this is done the master list updates accordingly without changing the chronological ordering of the accounts.

C

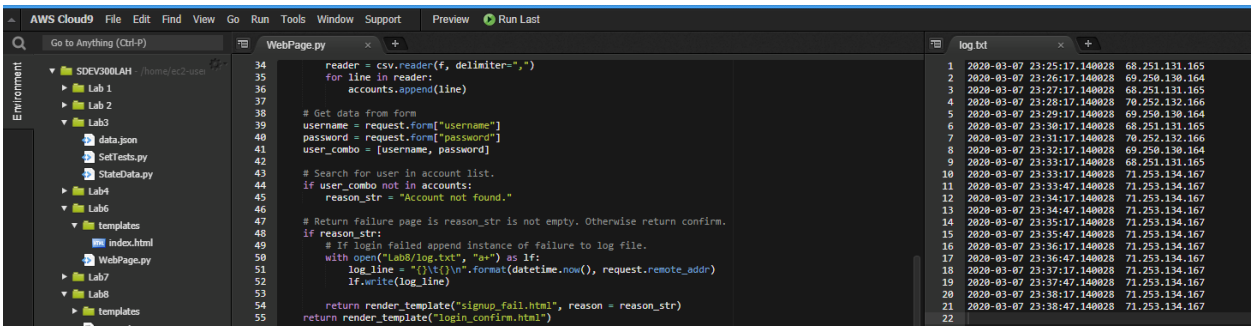


As per NIST SP 800-63B recommendations passwords must meet certain criteria before users can create or update their accounts. The picture above demonstrates a user attempting to create an account twice, failing both times because the password was too common and too short respectively (top right and bottom right), and attempting to update their password and failing because the password was too common. This logic also applies to passwords that are too long in length (64+ characters). Passwords used for this demonstration were “password” and “hello”.



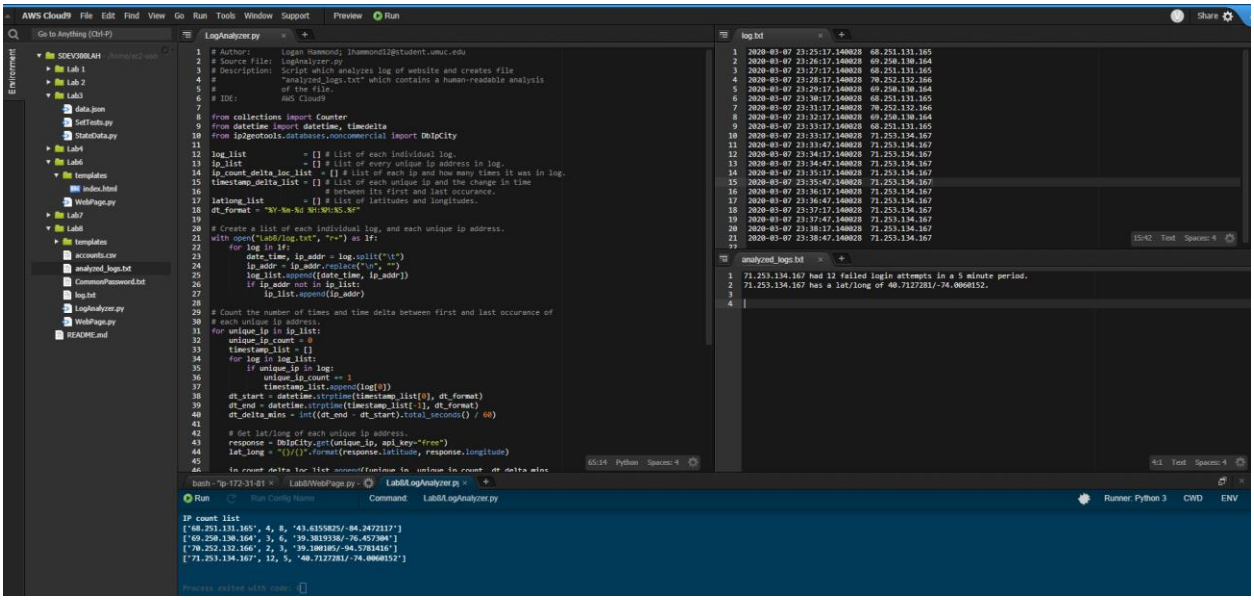
NIST SP 800-63B recommends time-based rate limiting when attempting to login or update passwords. This project will throttle users that fail to login or update their password successfully after 15 attempts per two minutes.

D



Upon a failed login attempt the program logs the date and time of the failed attempt as well as the IP address that the attempt originated from. Note: The log shown above is manufactured but based on real recorded logs in order too maintain anonymity and showcase functionally of the log analyzer in the next section.

E



LogAnalyzer.py is a script which parses the log file and outputs noteworthy data in a human-readable format. If no noteworthy data is found that is reported to the user. Noteworthy for this script is defined as an IP address appearing more than 10 times within a period of 5 minutes. Shown at the bottom is the complete list of each unique IP address and its relevant data which includes the number of times it appears, the time delta (time between first and last appearance) and the latitude and longitude from which the IP originates.

Decryption

a) Dots, dashes, and slashes; spaces => Morse code.
Given: - .... / ... -.. . ...- / ...-- ----- ----- / -.-. .-.. .- ... ... / .... .- ... / ... --- -- . / ... - .-. .- -. --. . / .-. . - -.- ..- . ... - ... .-.-.-
Solution: THIS SDEV 300 CLASS HAS SOME STRANGE REQUESTS.

b) Contains a-z, A-Z, 0-9; no spaces => (probably) Base64.
Solution: So this is base64. Now I know.

c) Formatted like plain language; apparent repetition; (probably) Simple cipher/cryptogram.
Solution: BEGIN KEY I AM SO CLEVER NO ONE COULD POSSIBLY FIGURE THIS OUT END KEY