John Kucera

Prof. Craig Poma

SDEV 300

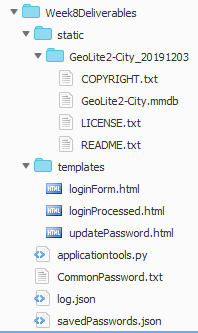
11 December 2019

Week 8: Lab 8

**Python Application Tools**

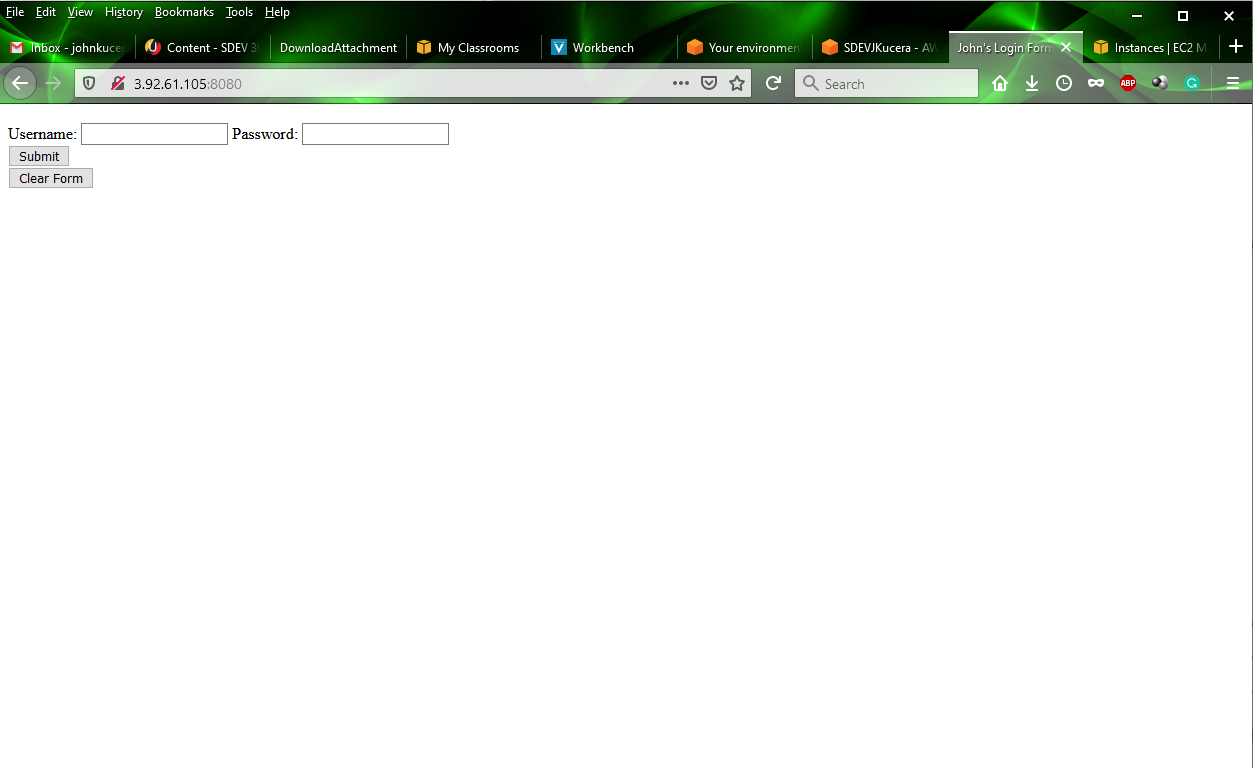
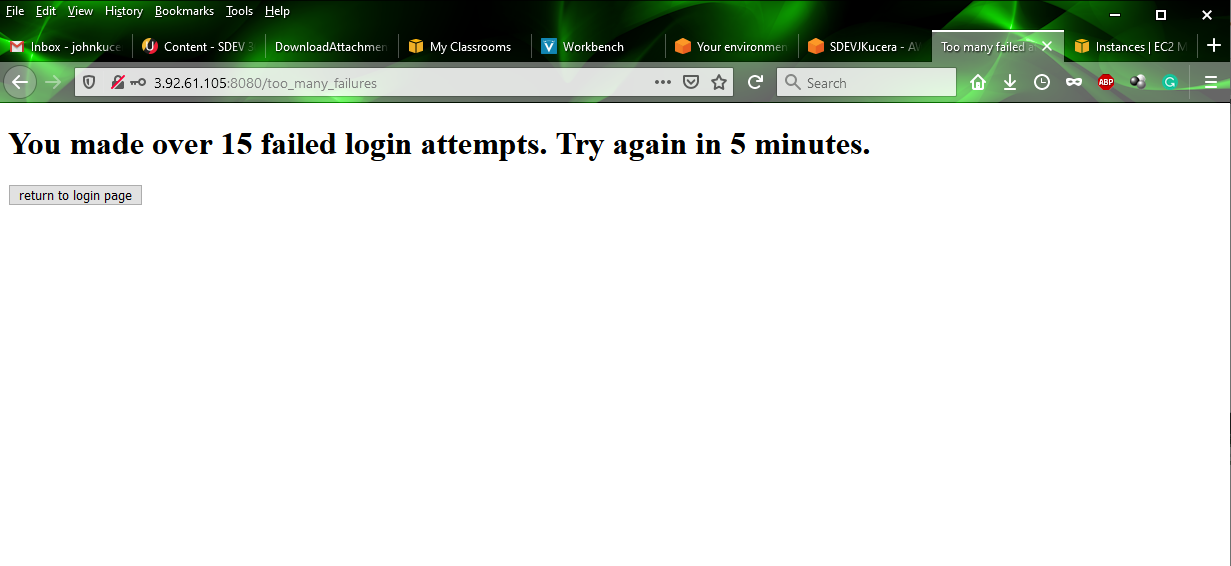
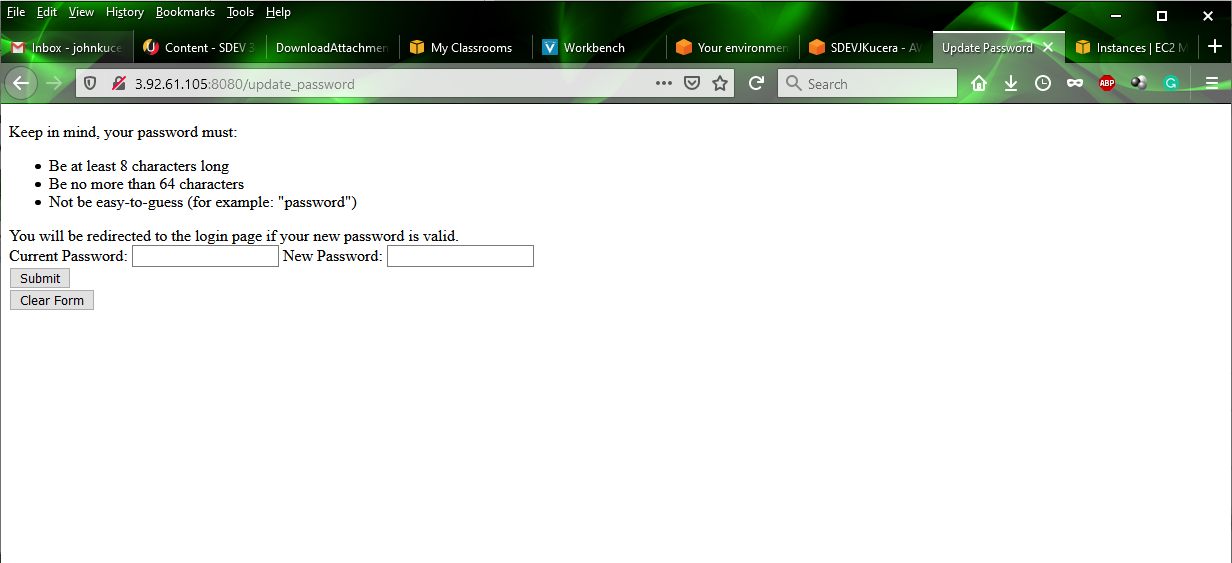
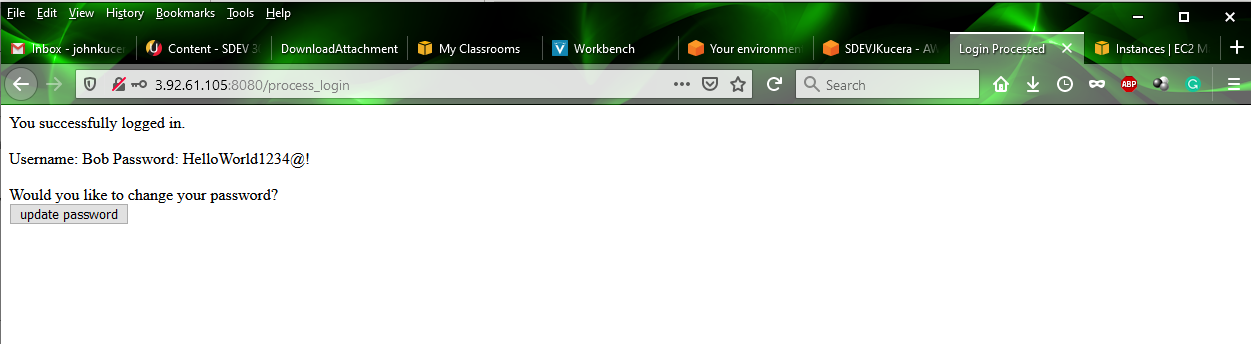
NOTES:

* For this Lab, I used the template you provided and added the following files of my own: **updateForm.html**, **log.json**, **savedPasswords.json**. I also used **CommonPassword.txt**, **GeoLite2-City.mmdb**, **loginForm.html**, and **loginProcessed.html**. My folders look like this:



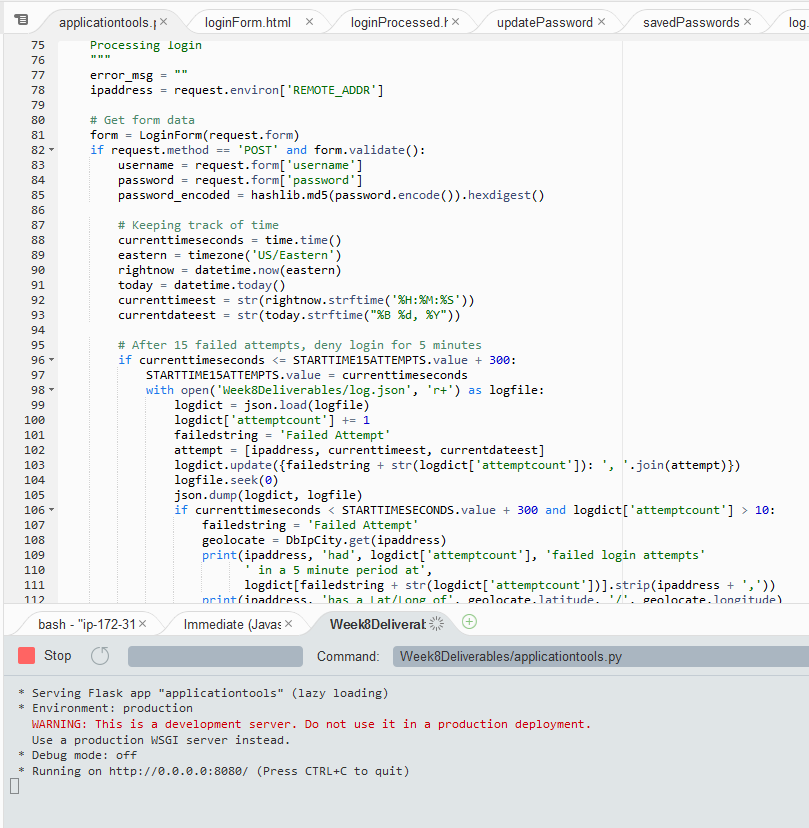
* I included this whole “Week8Deliverables” folder in my submission. The main Python code to run is the file titled **applicationtools.py**.
* Even if **log.json** and **savedPasswords.json** are missing, the program will create them anyways.
* PRESETS: Username = **Bob**, Password = **HelloWorld1234@!**
* I switched to Arial font for this analysis documentation since all the screenshots already show Times New Roman.

HOW IT LOOKS (webpage in browser, 4 routes):

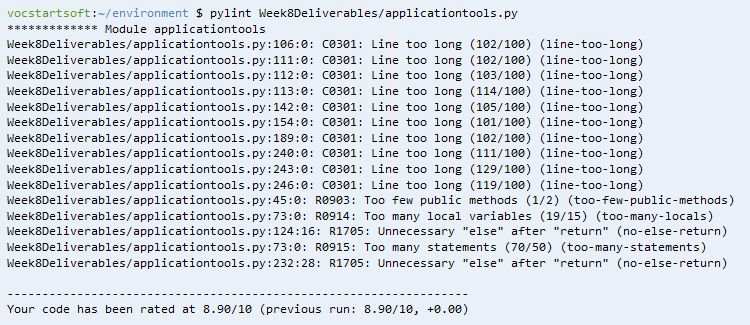
 

The IPv4 Public IP was 3.92.61.105, so I typed 3.92.61.105.205:8080/ into the search bar. My webpage successfully loaded in my browser.

HOW IT LOOKS (code in Cloud9 IDE)



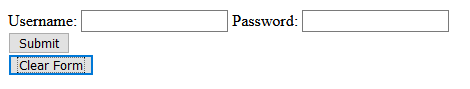
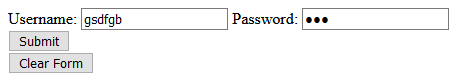
pylint analysis of Python Application Tools code: 8.90/10 (PASSING)



NOTE: Screenshots are under each respective test case. I should’ve started doing this a long time ago.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Actual Output** | **Pass?** |
| 1 | (into login form) Press “Clear Form” | Text fields are cleared | Text fields are cleared | Yes |

Screenshots before and after pressing “Clear Form”:



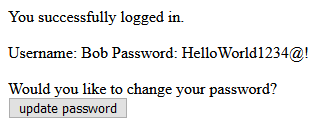
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | (into login form) Enter wrong username: dfgdf | Return to main login form with error message “Username does not exist. Please try again.” | Return to main login form with error message “Username does not exist. Please try again.” | Yes |



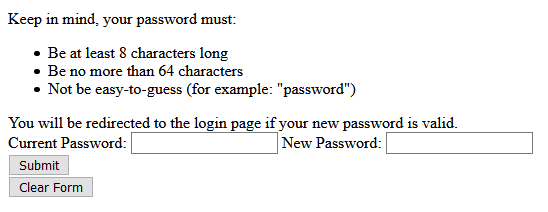
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | (into login form) Enter wrong password:  asdfasd | Return to main login form with error message “Password is incorrect. Please try again.” | Return to main login form with error message “Password is incorrect. Please try again.” | Yes |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | (into login form) Enter correct username and password: “Bob”, “HelloWorld1234@!” | Redirect to “loginProcessed” form with correct username and password listed | Redirect to “loginProcessed” form with correct username and password listed | Yes |

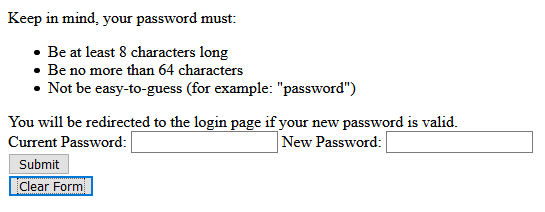
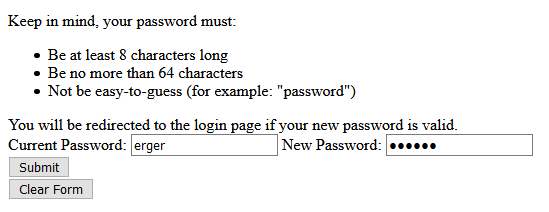


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | (into loginProcessed page) Press “update password” button | Redirect to “updatePassword” form | Redirect to “updatePassword” form | Yes |

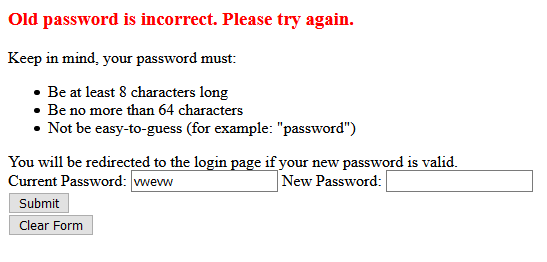


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | (into updatePassword form) Press “Clear form” button | Text fields are cleared | Text fields are cleared | Yes |

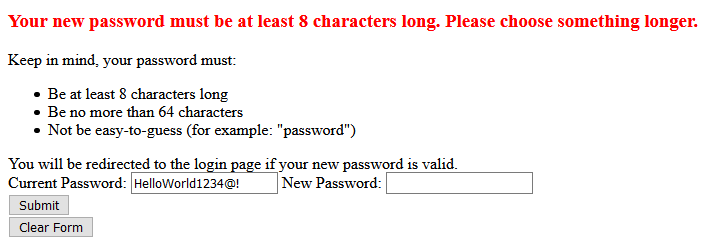
Screenshots before and after pressing “Clear Form”:



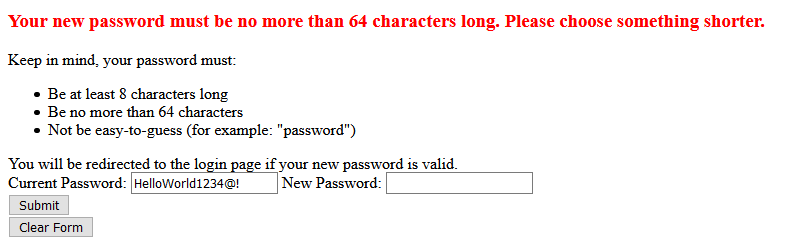
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | (into updatePassword form) enter incorrect old password | Return to update form with error message “Old password is incorrect. Please try again.” | Return to update form with error message “Old password is incorrect. Please try again.” | Yes |



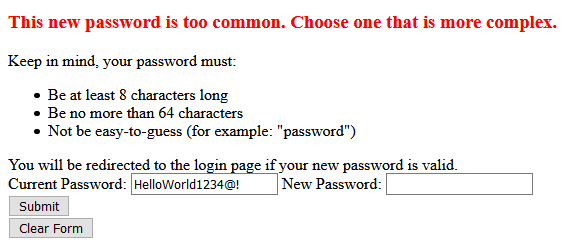
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | (into updatePassword form) enter new password below 8 characters | Return to update form with error message “Your new password must be at least 8 characters long. Please choose something longer.” | Return to update form with error message “Your new password must be at least 8 characters long. Please choose something longer.” | Yes |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9 | (into updatePassword form) enter new password over 64 characters | Return to update form with error message “Your new password must be no more than 64 characters long. Please choose something shorter.” | Return to update form with error message “Your new password must be no more than 64 characters long. Please choose something shorter.” | Yes |

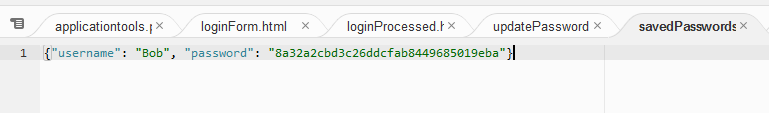


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | (into updatePassword form) enter new password that is in CommonPassword.txt (I entered “jennifer”) | Return to update form with error message “This new password is too common. Choose one that is more complex.” | Return to update form with error message “This new password is too common. Choose one that is more complex.” | Yes |

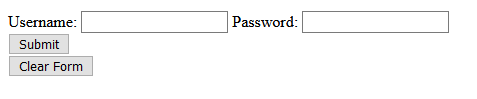


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | (into updatePassword form) enter password that meets requirements (I entered “yeahyeahyeah”) | New password is stored, encrypted. Redirect to main login form. New password “yeahyeahyeah” works for logging in. | New password is stored, encrypted. Redirect to main login form. New password “yeahyeahyeah” works for logging in. | Yes |

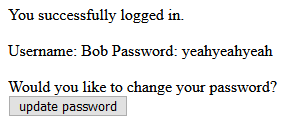
New password is stored, encrypted (this is the correct encryption for “yeahyeahyeah”)



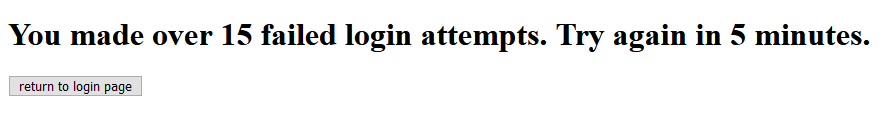
Redirected to main login form:



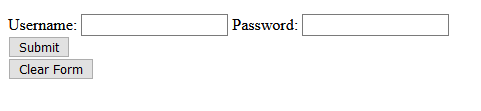
New password “yeahyeahyeah” works for logging in:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12 | (into login form) make 16 failed login attempts | redirect to “too\_many\_attempts” page, displays “You made over 15 failed login attempts. Try again in 5 minutes.” with a button to return to login page | redirect to “too\_many\_attempts” page, displays “You made over 15 failed login attempts. Try again in 5 minutes.” with a button to return to login page | Yes |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | (into “too many attempt” page) press “return to login page” button | redirect to login form | redirect to login form | Yes |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 14 | (into login form) make another login attempt before 5 minutes have passed | Return to main login form with error message “It has not been at least 5 minutes since previous failed attempt. Please wait 5 minutes.” | Return to main login form with error message “It has not been at least 5 minutes since previous failed attempt. Please wait 5 minutes.” | Yes |

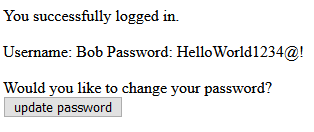
Timer for denying login for 5 minutes starts here. Until the 5 minutes is up, whenever another attempt is made, the timer resets and the subscriber must wait for 5 minutes again.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | (into login form) WAIT 5 minutes after previous failed attempt, make a failed attempt | Return to main login form with error message “Password is incorrect. Please try again.” No error message about waiting 5 minutes. | Return to main login form with error message “Password is incorrect. Please try again.” No error message about waiting 5 minutes. | Yes |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 16 | (into login form) WAIT 5 minutes after previous failed attempt, make a correct attempt | Login successful, Redirect to “loginProcessed” form with correct username and password listed | Login successful, Redirect to “loginProcessed” form with correct username and password listed | Yes |

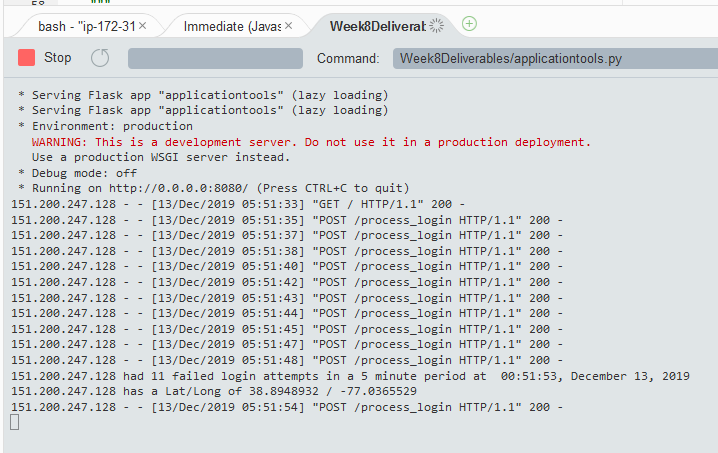


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17 | (into login form) make 1 failed attempt | log.json shows the date, time, and IP address for Failed Attempt 1 | log.json shows the date, time, and IP address for Failed Attempt 1 | Yes |



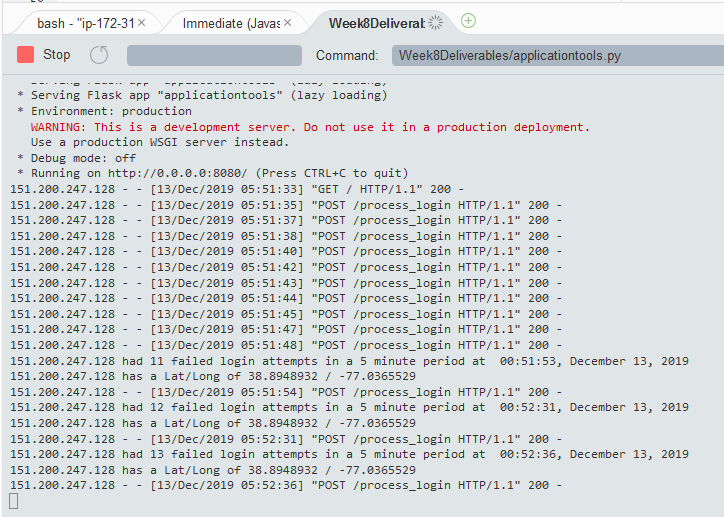
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 18 | (into login form) make 11th failed attempt in 5 minute period | log.json shows the date, time, and IP address for Failed attempt 11.  Logger statements are printed into the console:  151.200.247.128 had 11 failed login attempts in a 5 minute period at 00:51:53, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | log.json shows the date, time, and IP address for Failed attempt 11.  Logger statements are printed into the console:  151.200.247.128 had 11 failed login attempts in a 5 minute period at 00:51:53, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | Yes |

Screenshots of updated log.json and console:



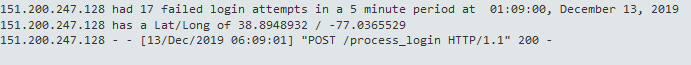
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 19 | (into login form) make 13th failed attempt in 5 minute period | log.json shows the date, time, and IP address for Failed attempt 13.  Logger statements are printed into the console:  151.200.247.128 had 13 failed login attempts in a 5 minute period at 00:52:36, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | log.json shows the date, time, and IP address for Failed attempt 13.  Logger statements are printed into the console:  151.200.247.128 had 13 failed login attempts in a 5 minute period at 00:52:36, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | Yes |

Screenshots of updated log.json and console:



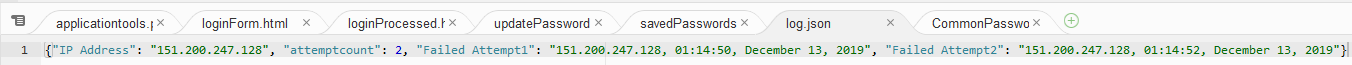
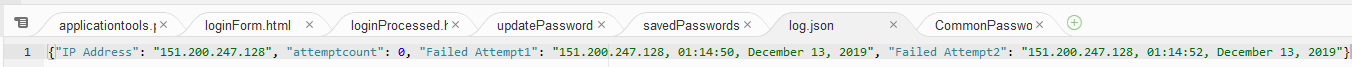
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20 | (into login form) make another login attempt before 5 minutes have passed | Treated as a failed attempt, whether the input is correct or not. log.json shows the date, time, and IP address for new Failed attempt (17).  Logger statements are printed into the console: 151.200.247.128 had 17 failed login attempts in a 5 minute period at 01:09:00, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | Treated as a failed attempt, whether the input is correct or not. log.json shows the date, time, and IP address for new Failed attempt (17).  Logger statements are printed into the console: 151.200.247.128 had 17 failed login attempts in a 5 minute period at 01:09:00, December 13, 2019  151.200.247.128 has a Lat/Long of 38.8948932 / -77.0365529 | Yes |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21 | (into login form) make 2 failed attempts, then correct input | Counter for failed attempts for subscriber reset to 0.  However, details for previous failed attempts in the log REMAIN. | Counter for failed attempts for subscriber reset to 0.  However, details for previous failed attempts in the log REMAIN. | Yes |

Screenshots of BEFORE and AFTER the correct input: Attempt counter is reset to 0, but previous failed attempt details remain.

**Secret Message Decryption**

1. - .... .. ... / ... -.. . ...- / ...-- ----- ----- / -.-. .-.. .- ... ... / .... .- ... / ... --- -- . / ... - .-. .- -. --. . / .-. . --.- ..- . ... - ... .-.-.-

is in **Morse Code**. When decoded, it says:   
  
**THIS/SDEV/300/CLASS/HAS/SOME/STRANGE/REQUESTS.**I narrowed it down to Morse Code from seeing the use of dots and dashes.

1. U28gdGhpcyBpcyBiYXNlNjQuIE5vdyBJIGtub3cu  
   is in **Base64**. When decoded, it says:  
     
   **So this is base64. Now I know.**  
   I narrowed it down to Base64 after seeing that it was a jumble of uppercase and lowercase letters with some numbers.
2. --- Psuwb Ysm ----   
   W oa gc qzsjsf. Bc cbs qcizr dcggwpzm twuifs hvwg cih.   
   --- Sbr Ysm ---  
   is in **Caesarian shift**. When decoded, it says:  
     
   **--- Begin Key ----  
   I am so clever. No one could possibly figure this out.  
   --- End Key ---**  
   I narrowed it down to Caesarian Shift after seeing that it seemed like a normal sentence using the English alphabet, just with substituted letters. Uppercase letters were at the lead of each “sentence”, followed by lowercase letters. Then, it was just a matter of using rumkin.com and going through each value of “N” to see which would turn decode the sentence into some sensible message. The result was **N = 12 for decryption**.