Testing: UserLogins

R. Costa-Tré

10 June 2022

Module tested	userLogins.py
Version	v0.3.1
File Path	/PLAYGROUND/PYTHON/PYTHON_CODE/
Last written by	R. Costa-Tré
Tested by	R. Costa-Tré

1 Results

No.	Input	Expected output	Output given	Punch list
1	CreateUser(”User1”, ”Pass1”)			
2	print(IsValidUser(”User1”, ”Pass1”))	True	False	1
3	print(IsValidUser(”User1”, ”pass1”)	False	False	
4	-	-	-	2
5	CreateUser(”User2”, ”Pass2”)			
6	print(IsValidUser(”User2”, ”Pass2”))	True	True	
7	print(IsValidUser(”User2”, ”pass2”)	False	False	
8	CreateUser(”User3”, ”Pass3”)			
9	print(IsValidUser(”User3”, ”Pass3”))	True	True	
10	print(IsValidUser(”User3”, ”pass3”)	False	False	
11	CreateUser(”User4”, ”Pass4”)			
12	print(IsValidUser(”User4”, ”Pass4”))	True	False	1
13	print(IsValidUser(”User4”, ”pass4”)	False	False	
14	CreateUser(”User5”, ”Pass5”)			
15	print(IsValidUser(”User5”, ”Pass5”))	(err)		3

Note: if a cell is left blank that implies that not issues occurred during the test and thus not comment or action is required.
(err): ’utf-8’ codec can’t decode byte 0xdc in position 2: invalid continuation byte.

2 Punch list

No.	Description	Resolved	Developer
1	Failed: Encrypted Bytes contain ’\n’ causing ecnryted bytes to split across lines. Information is line sensitive.	10-Jun-22	R. Costa-Tré
2	Cleared users.txt file to get rid of corrupted data.	10-June-22	R. Costa-Tré
3	Failure: Error thrown due to reading second split of code mentioned in (2) as UserID	10-Jun-22	R. Costa-Tré

3 Concerns and future considerations

3.1 No defensive coding

The current code does not account for the '\n' character int he encoded password. As such, this error will always occur when the hash results with said character. There is also currently no defensive programming that catches codes with '\n' characters. This should be alleviated by either using a database to store informatin that ignores new-line or by creating a new encrypted key if the code contains a new-line character.