2018

# Project Report

## PYTHON LOGIN FUNCTION WITH MONGODB

NAME - M.F.F AHAMED

INDEX - COHDSE182F-015

# Table of Contents

## Introduction

This is a simple login form. If user entered correct user name and password. User can login to the system otherwise display an error message.

- Front end created using Python.
- Back end created using MongoDB.

  - ➢ Username-test
  - ➢ Password-pass@123

# Source Codes

## Create DB, Collection & Data Insertion through the python

```
from pymongo import MongoClient

Client=MongoClient()

db=Client[“Login”]

collection=db[“Login_Details”]

log={}

log[“UserName”]=”test”

log[“Password”]=”dc06698f0e2e75751545455899adccc3”

collection.insert(log)
```

- Create db call Login.
- Create Collection call Login_Details.
- Create document UserName & Password.
- Insert 1 record.

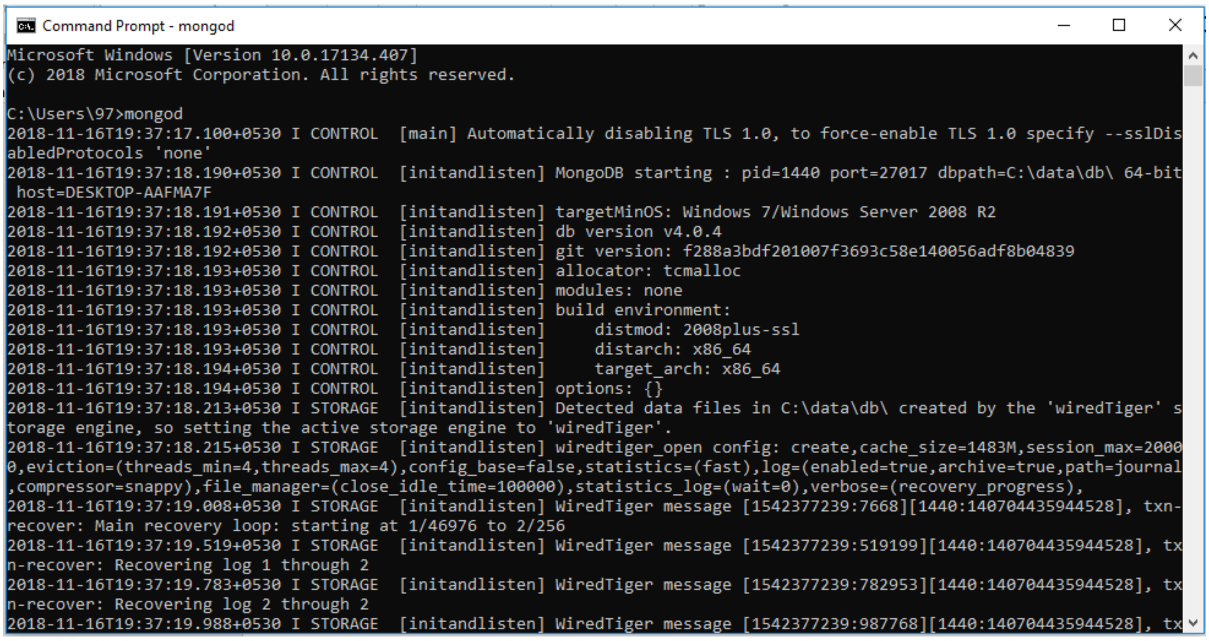
## Create Console Application

```
import pymongo

import hashlib

print(“##########  User Login  ##########”)

un=input(“enter User Name : “)

pwd=hashlib.md5(input(“enter Password : “).encode()).hexdigest()

uri=”mongodb://127.0.0.1:27017”

client=pymongo.MongoClient(uri)

database=client[‘Login’]

collection=database[‘Login_Details’]

users=collection.find({})

for user in users:

  if un==user[‘UserName’] and pwd==user[‘Password’]:

    print(“$$$ Successful Loged IN $$$”)

    print(“Hello “+un)

  else:

    print(“UserName Or Password Incorrect”)
```

- Import pymongo library to connect db.
- Import hashlib to get hashing methods.
- Use md5 algorithm  to hash password.

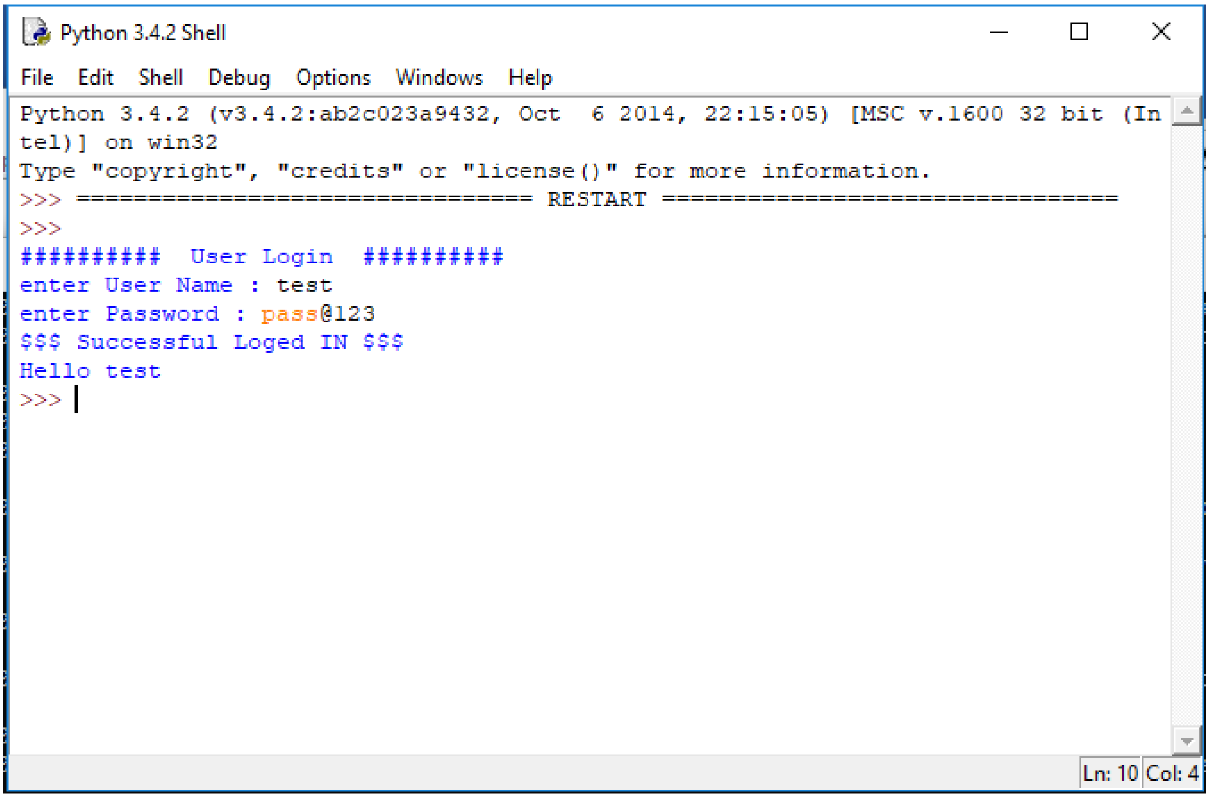
# Program Outputs

## Running the db server MongoDB



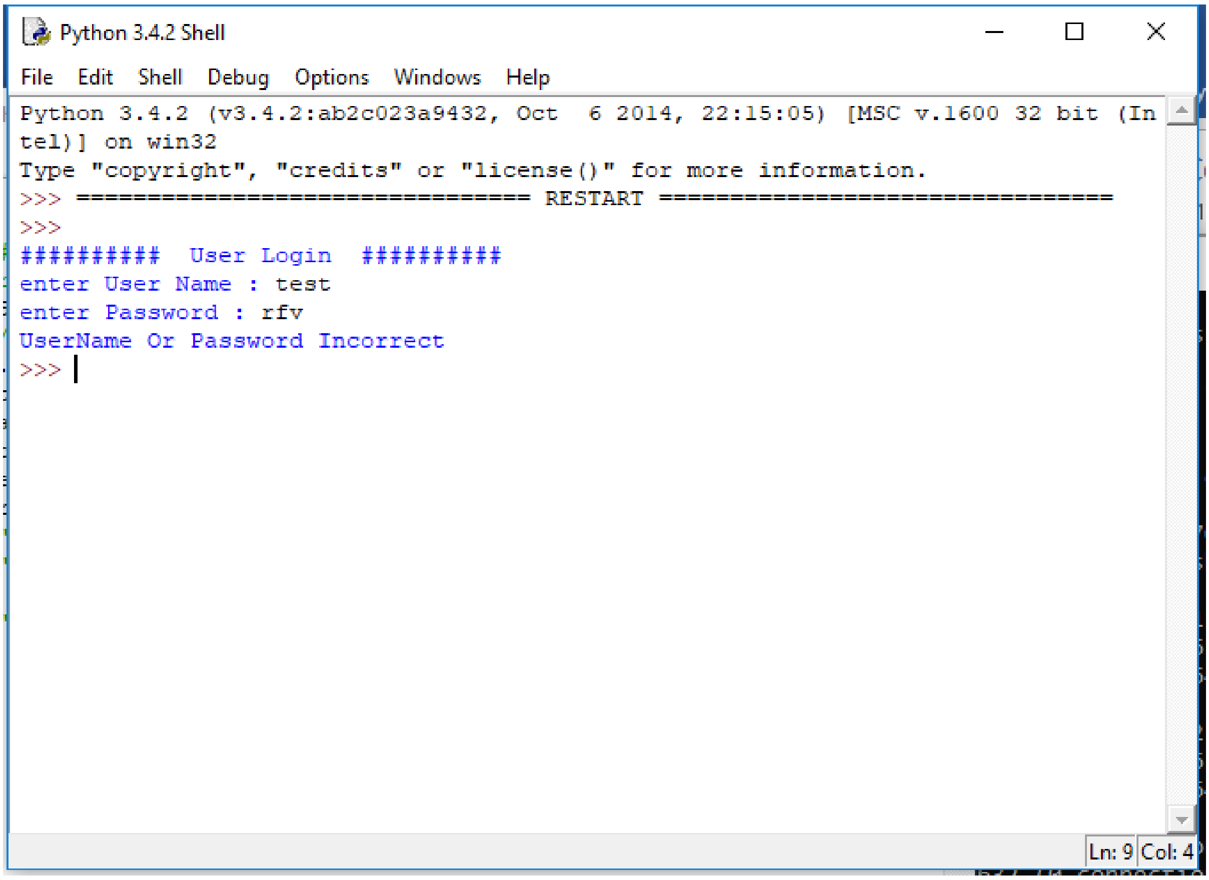
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\97>mongod
2018-11-16T19:37:17.100+0530 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDis
abledProtocols 'none'
2018-11-16T19:37:18.190+0530 I CONTROL  [initandlisten] MongoDB starting : pid=1440 port=27017 dbpath=C:\data\db\ 64-bit
 host=DESKTOP-AAFMA7F
2018-11-16T19:37:18.191+0530 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-11-16T19:37:18.192+0530 I CONTROL  [initandlisten] db version v4.0.4
2018-11-16T19:37:18.192+0530 I CONTROL  [initandlisten] git version: f288a3bdf201007f3693c58e140056adf8b04839
2018-11-16T19:37:18.193+0530 I CONTROL  [initandlisten] allocator: tcmalloc
2018-11-16T19:37:18.193+0530 I CONTROL  [initandlisten] modules: none
2018-11-16T19:37:18.193+0530 I CONTROL  [initandlisten] build environment:
2018-11-16T19:37:18.193+0530 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2018-11-16T19:37:18.193+0530 I CONTROL  [initandlisten]     distarch: x86_64
2018-11-16T19:37:18.194+0530 I CONTROL  [initandlisten]     target_arch: x86_64
2018-11-16T19:37:18.194+0530 I CONTROL  [initandlisten] options: {}
2018-11-16T19:37:18.213+0530 I STORAGE  [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' s
torage engine, so setting the active storage engine to 'wiredTiger'.
2018-11-16T19:37:18.215+0530 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=1483M,session_max=2000
0,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal
,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2018-11-16T19:37:19.008+0530 I STORAGE  [initandlisten] WiredTiger message [1542377239:7668][1440:140704435944528], txn-
recover: Main recovery loop: starting at 1/46976 to 2/256
2018-11-16T19:37:19.519+0530 I STORAGE  [initandlisten] WiredTiger message [1542377239:519199][1440:140704435944528], tx
n-recover: Recovering log 1 through 2
2018-11-16T19:37:19.783+0530 I STORAGE  [initandlisten] WiredTiger message [1542377239:782953][1440:140704435944528], tx
n-recover: Recovering log 2 through 2
2018-11-16T19:37:19.988+0530 I STORAGE  [initandlisten] WiredTiger message [1542377239:987768][1440:140704435944528], tx
```

## Successfully Login



```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:15:05) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
##########  User Login  ##########
enter User Name : test
enter Password : pass@123
$$$ Successful Loged IN $$$
Hello test
>>> 
```

## Invalid Login



```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:15:05) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
##########  User Login  ##########
enter User Name : test
enter Password : rfv
UserName Or Password Incorrect
>>> 
```

## Conclusion

- **MongoDB** is best suitable for hierarchical data storage, but **RDBMS** is not. **MongoDB** supports JSON query language along with SQL but **RDBMS** supports SQL query language only. ... **MongoDB** is almost 100 times faster than traditional database system like **RDBMS**, which is slower in comparison with the NoSQL databases.
- MongoDB’s DO concept is much more flexible than RDBMS technologies and it is a more suitable choice to work with dynamic languages such as Python .

# References

- https://docs.mongodb.com/
- https://www.w3schools.com/python/
- MongoDB and Python O’reily edition - Niall O’Higgins
- learn-python-in-7-days