MANIPAL UNIVERSITY
JAIPUR

Experiential Learning Report

B.Tech. Program (First Year)
Course: Experiential Learning
Course Code: DA1001

"SIGN-UP INFORMATION VALIDATOR"

by

Esha Baweja (Reg. No.: **209301151**)

Under the guidance

of

Mr. Jayakrishna R.

Assistant Professor (Senior Scale)

Department of Computer Science and Engineering

School of Computing & Information Technology

Faculty of Engineering
Manipal University Jaipur, India
July-November 2020

CERTIFICATE

This is to certify that the project titled "**SIGN-UP INFORMATION VALIDATOR**" is a record of the bonafide work done by **Esha Baweja (Reg. No.: 209301151)**, submitted for the partial fulfilment of the requirements for the completion of the Experiential Learning (DA1001) course in the Department of Computer Science and Engineering of Manipal University Jaipur, during the academic session July-November 2020.

____

Signature of the mentor

Mr. Jayakrishna R.
Assistant Professor (Senior Scale)
Department of Computer Science and Engineering

____

Signature of the HoD

Prof. (Dr.) Sandeep Joshi

Head, Department of Computer Science and Engineering

ACKNOWLEDGEMENT

I would like to express my gratitude to my mentor Mr. Jayakrishna R. who gave me the golden opportunity to do this wonderful project "**SIGN-UP INFORMATION VALIDATOR"**, which made me research this topic.

I would like to thank Mrs. Ruchi Agarwal, who taught me programming in Python. Without her help, this project would not be the same.

Lastly, I would like to thank my friends and family for their unconditional love and support.

-Esha Baweja

TABLE OF CONTENTS

ABSTRACT

This project is a command line based project and is developed using Python. The main purpose of this is to take sign-up information as input from the user, and check the validity of that information. This includes their email address, a strong password, and their phone number. The project contains three separate programs pertaining to each of the above. It will help us investigate the authenticity of the information provided by the user, and would also ensure that the user enters a strong password for the safety of their profile.

INTRODUCTION

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.
- The most recent major version of Python is Python 3.
- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In this project, I have used Python to write three programs as follows.

The first one is for **email address validation**. It takes a string as input from the user.

- It then divides the email into two parts - the local part, and the domain.
- It checks whether the local part is valid. If yes, it moves on to the domain part.
- From a given list of domains, it checks whether the domain name is valid.
- If both the local part and the domain name are valid, the email address is valid. This message is displayed to the user.

The second program **validates the password**.

- It takes a string input from the user.
- It checks the password length is between 8 and 16 characters.
- It checks that the password has no white spaces, and at least one lowercase, one uppercase, one digit, and a special character.
- If all of the above conditions are met, it displays a message that the password is valid. Otherwise, it says "Invalid".

The third program **checks the validity of the phone number**.

- It takes a string input from the user and splits it as follows: the last ten digits are the mobile number, and the rest of it is the country code.
- It checks that the phone number has only ten digits, and that mobile number does not start with 0. If yes, the number is valid.
- From a given list of country codes, it checks whether the country code is valid.
- If both the country code and the mobile number are valid, the "Valid" message is displayed to the user.
- Otherwise, the phone number is deemed to be invalid.

SOFTWARE AND HARDWARE REQUIREMENTS

1. Operating system: Linux- Ubuntu 16.04 or later, or Windows 7 or later.
2. Python 3 installed.
3. A text editor.
4. Python runs on virtually anything from a Raspberry Pi to any top end of the line supercomputer.

METHODOLOGY

ALGORITHMS

EMAIL ADDRESS VALIDATION

1) Start.
2) Input email ID.
3) If there are whitespaces in it, print "Invalid" and stop.
4) Check if exactly one '@' is present in the input string. If yes,split the local part and domain.
5) Check if the local part starts with a letter. If not, print "Invalid" and stop.
6) Check if the domain is present in a given list of domains. If not, print "Invalid" and stop.
7) Email is valid. Print "Valid".
8) Stop.

PASSWORD VALIDATION

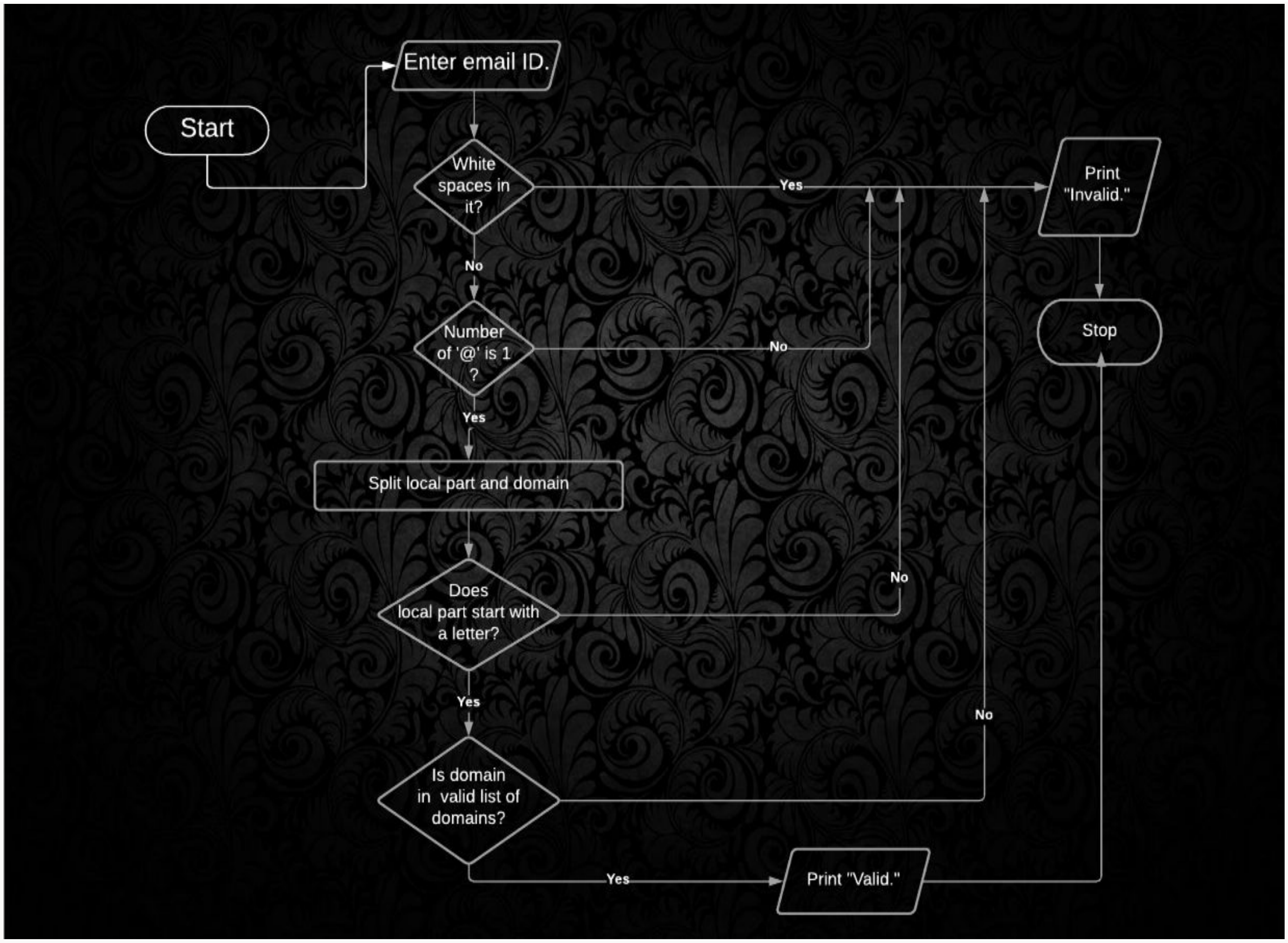
1) Start. Take user input Password.
2) Check whether password length is between 8 to 16 characters. If not, print "Invalid" and stop.
3) Traversing through the password, check whether the password contains any space. If yes, print "Invalid" and stop.

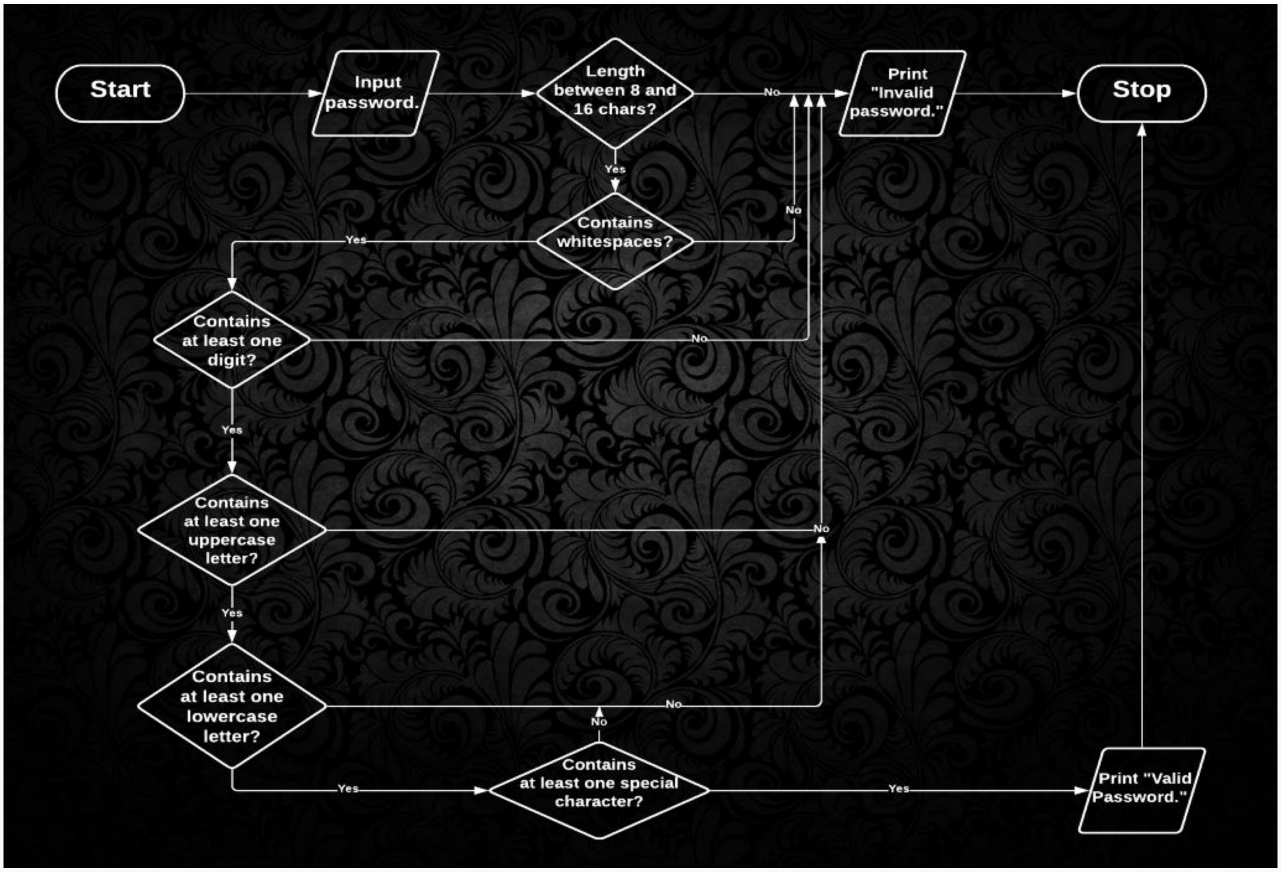
4) Check whether the password contains at least one digit(0-9). If not, print "Invalid" and stop.
5) Check whether the password contains at least one lowercase letter(a-z). If not, print "Invalid" and stop.
6) Check whether the password contains one uppercase letter(A-Z). If not, print "Invalid" and stop.
7) Check whether password contains at least one special character (@, #, %,&, !, $, etc....). If not, print "Invalid" and stop.
8) Password is Valid. Print "Valid".
9) Stop.

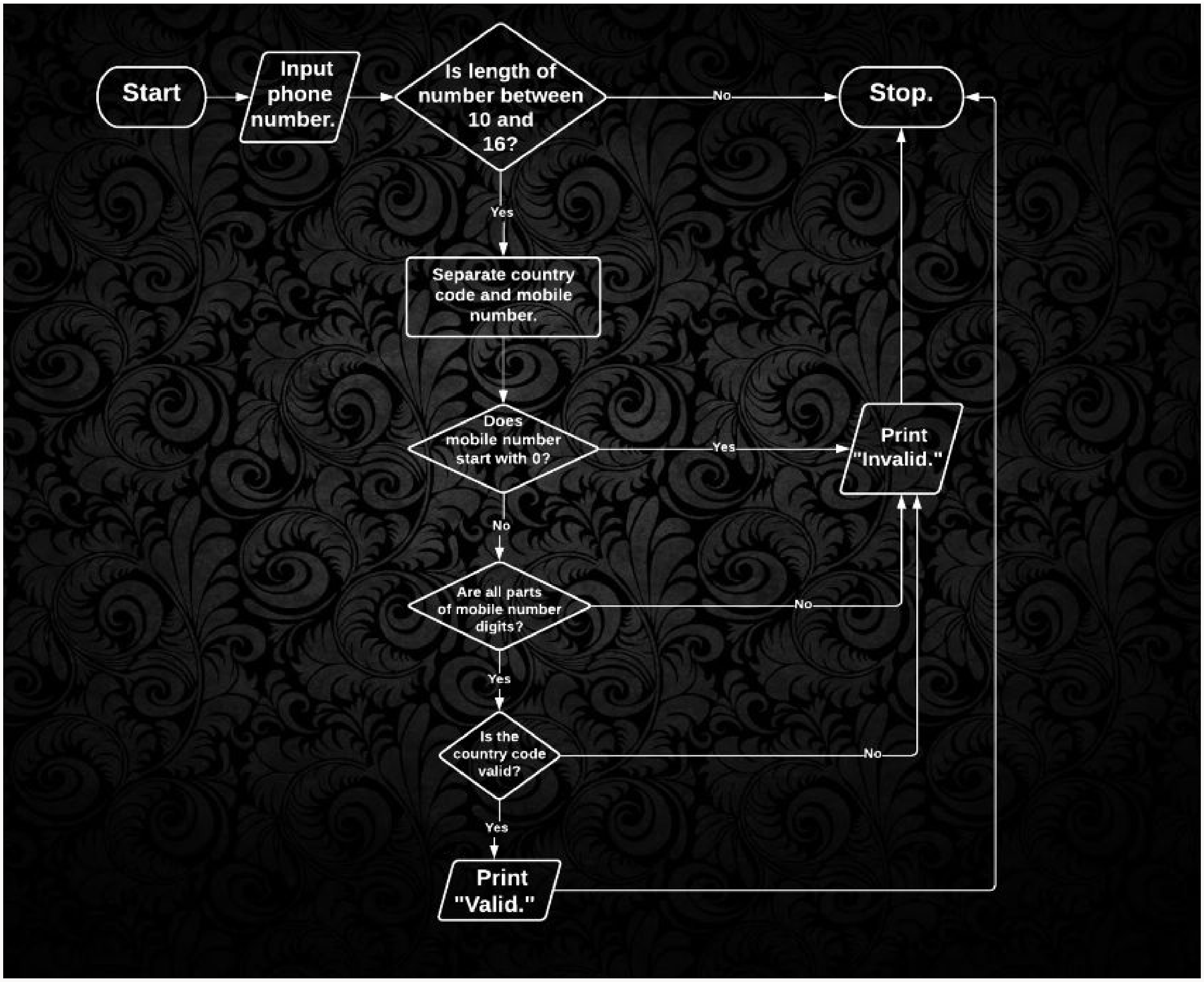
PHONE NUMBER VALIDATION

1) Start.
2) Input contact number.
3) Split the number.The last ten digits would be the phone number, and the rest would be the country code.
4) Check whether the mobile number has only digits. If not, print "Invalid". Stop.
5) Check if the country code is valid, using the list of codes. If not, print "Invalid". Stop.
6) Check whether phone number starts with 0. If it does, print "Invalid". Stop.
7) Phone number is valid. Print "Valid phone number."
8) Stop.

FLOWCHARTS

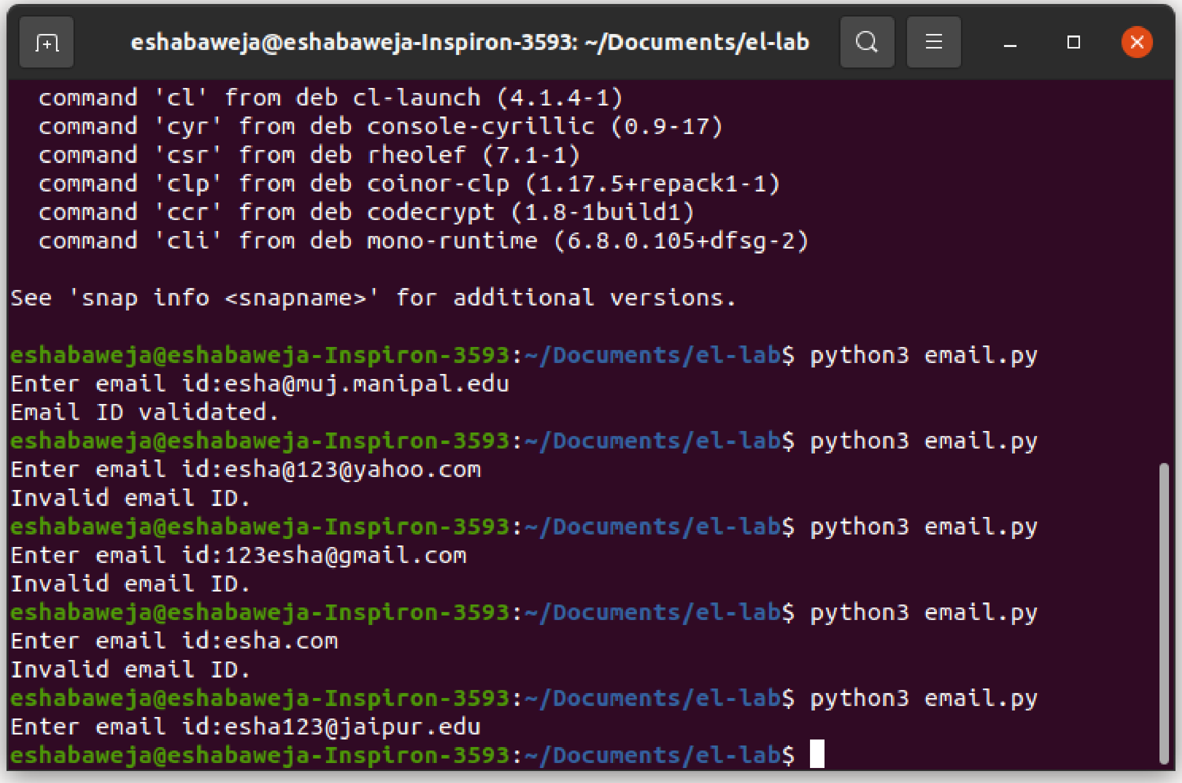

Start
Enter email ID.
White spaces in it?
Yes
Print "Invalid."
Stop
No
Number of '@' is 1 ?
No
Yes
Split local part and domain
Does local part start with a letter?
No
Yes
Is domain in valid list of domains?
No
Yes
Print "Valid."


Start
Input password.
Length between 8 and 16 chars?
No
Print "Invalid password."
Stop
Yes
Contains whitespaces?
No
Yes
Contains at least one digit?
No
Yes
Contains at least one uppercase letter?
No
Yes
Contains at least one lowercase letter?
No
No
Yes
Contains at least one special character?
Yes
Print "Valid Password."


Start
Input phone number.
Is length of number between 10 and 16?
No
Stop.
Yes
Separate country code and mobile number.
Does mobile number start with 0?
Yes
Print "Invalid."
No
Are all parts of mobile number digits?
No
Yes
Is the country code valid?
No
Yes
Print "Valid."

PROGRAMS AND OUTPUTS

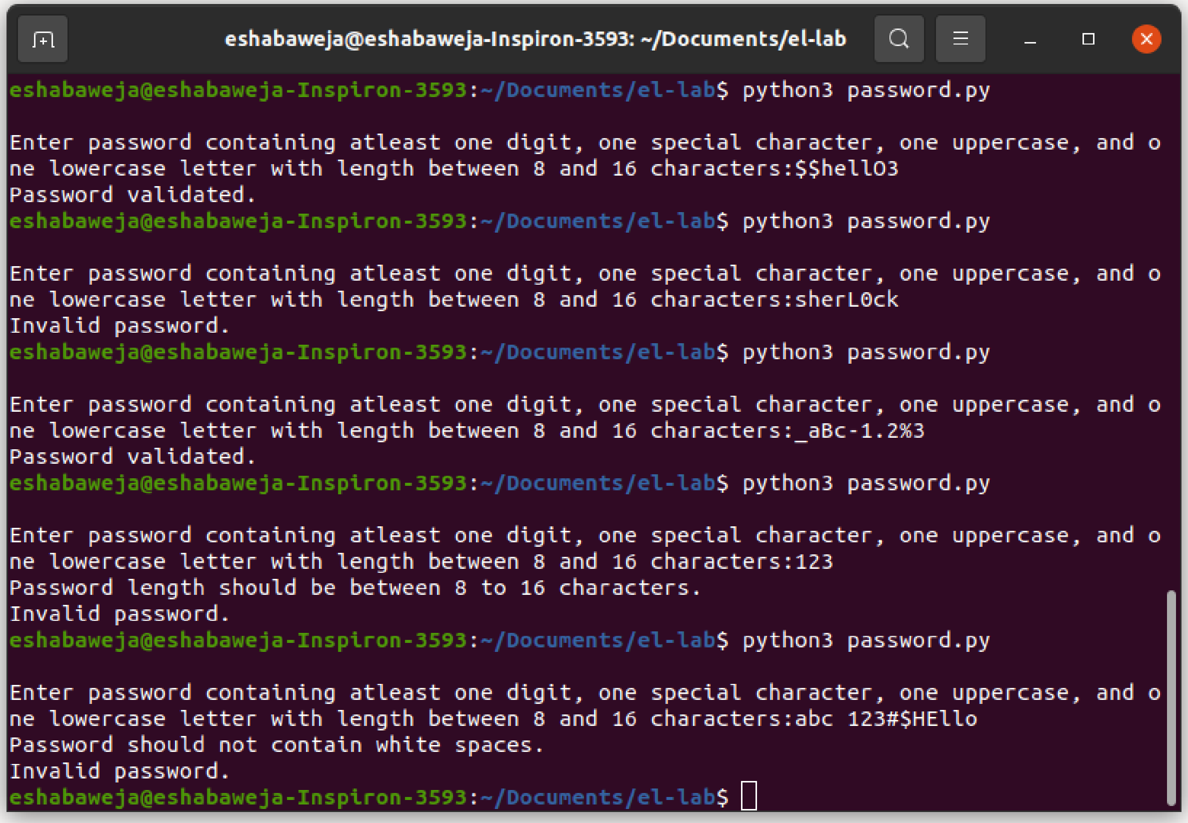
```
#A simple program to validate the email input by the user.
email = str(input("Enter email id:"))
ats=0

for i in email:
        if (i==' '):
        print ("Invalid email ID.")
        if (i=='@'):
         ats=ats+1

domains = [
  "aol.com", "att.net", "comcast.net", "facebook.com", "gmail.com",
"gmx.com", "googlemail.com",
  "google.com", "hotmail.com", "hotmail.co.uk", "mac.com", "me.com",
"mail.com", "msn.com",
  "live.com", "sbcglobal.net", "verizon.net", "yahoo.com",
"yahoo.co.uk",
"email.com", "fastmail.fm", "games.com" , "gmx.net", "hush.com",
"hushmail.com", "icloud.com",
  "iname.com", "inbox.com", "lavabit.com", "love.com" , "outlook.com",
"pobox.com", "protonmail.ch", "protonmail.com", "tutanota.de",
"tutanota.com", "tutamail.com", "tuta.io",
 "keemail.me", "rocketmail.com" , "safe-mail.net", "wow.com" ,
"ygm.com" ,
  "ymail.com", "zoho.com", "yandex.com",
  "bellsouth.net", "charter.net", "cox.net", "earthlink.net", "juno.com",
  "btinternet.com", "virginmedia.com", "blueyonder.co.uk",
"freeserve.co.uk", "live.co.uk",
  "ntlworld.com", "o2.co.uk", "orange.net", "sky.com", "talktalk.co.uk",
"tiscali.co.uk",
  "virgin.net", "wanadoo.co.uk", "bt.com",
  "sina.com", "sina.cn", "qq.com", "naver.com", "hanmail.net",
"daum.net", "nate.com", "yahoo.co.jp", "yahoo.co.kr", "yahoo.co.id",
"yahoo.co.in", "yahoo.com.sg", "yahoo.com.ph", "163.com", "yeah.net",
"126.com", "21cn.com", "aliyun.com", "foxmail.com",
  "hotmail.fr", "live.fr", "laposte.net", "yahoo.fr", "wanadoo.fr",
"orange.fr", "gmx.fr", "sfr.fr", "neuf.fr", "free.fr",
  "gmx.de", "hotmail.de", "live.de", "online.de", "t-online.de", "web.de",
"yahoo.de",
  "libero.it", "virgilio.it", "hotmail.it", "aol.it", "tiscali.it", "alice.it",
"live.it", "yahoo.it", "email.it", "tin.it", "poste.it", "teletu.it",
  "mail.ru", "rambler.ru", "yandex.ru", "ya.ru", "list.ru",
  "hotmail.be", "live.be", "skynet.be", "voo.be", "tvcablenet.be",
"telenet.be",
  "hotmail.com.ar", "live.com.ar", "yahoo.com.ar", "fibertel.com.ar",
"speedy.com.ar", "arnet.com.ar",
  "yahoo.com.mx", "live.com.mx", "hotmail.es", "hotmail.com.mx",
"prodigy.net.mx",
  "yahoo.ca", "hotmail.ca", "bell.net", "shaw.ca", "sympatico.ca",
"rogers.com",
  "yahoo.com.br", "hotmail.com.br", "outlook.com.br", "uol.com.br",
"bol.com.br", "terra.com.br", "ig.com.br", "itelefonica.com.br",
"r7.com", "zipmail.com.br", "globo.com", "globomail.com",
"oi.com.br","muj.manipal.edu"
]

if (ats==1):                    #a valid email address has exactly one 'at'
sign
        x = email.split("@")
        local_part = x[0]
        domain = x[1]

        if local_part[0].isalpha():
        for i in domains:
        if (i == domain):
                print("Email ID validated.")

        else:
        print("Invalid email ID.")

else:
        print("Invalid email ID.")
```



```
  command 'cl' from deb cl-launch (4.1.4-1)
  command 'cyr' from deb console-cyrillic (0.9-17)
  command 'csr' from deb rheolef (7.1-1)
  command 'clp' from deb coinor-clp (1.17.5+repack1-1)
  command 'ccr' from deb codecrypt (1.8-1build1)
  command 'cli' from deb mono-runtime (6.8.0.105+dfsg-2)

See 'snap info <snapname>' for additional versions.

eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 email.py
Enter email id:esha@muj.manipal.edu
Email ID validated.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 email.py
Enter email id:esha@123@yahoo.com
Invalid email ID.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 email.py
Enter email id:123esha@gmail.com
Invalid email ID.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 email.py
Enter email id:esha.com
Invalid email ID.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 email.py
Enter email id:esha123@jaipur.edu
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ 
```

```
#A program to validate password
password = input("\nEnter password containing at least one digit, one
special character, one uppercase, and one lowercase letter with length
between 8 and 16 characters:")

count_up =0
count_low =0
count_num =0
count_spec =0

special_chars = ['!', '@','#','$','%','^','&',
                '*','(',')','-','_','.']

if (len(password)>7 and len(password)<17):
    for i in password:
    if i == ' ':
    print("Password should not contain white spaces.")
    break

    elif i.isupper(): #checking at least one uppercase
    count_up = 1

    elif i.islower(): #checking at least one lowercase
    count_low = 1

    elif i.isdigit(): #checking at least one digit
    count_num = 1

    elif i in special_chars:
    count_spec = 1
```

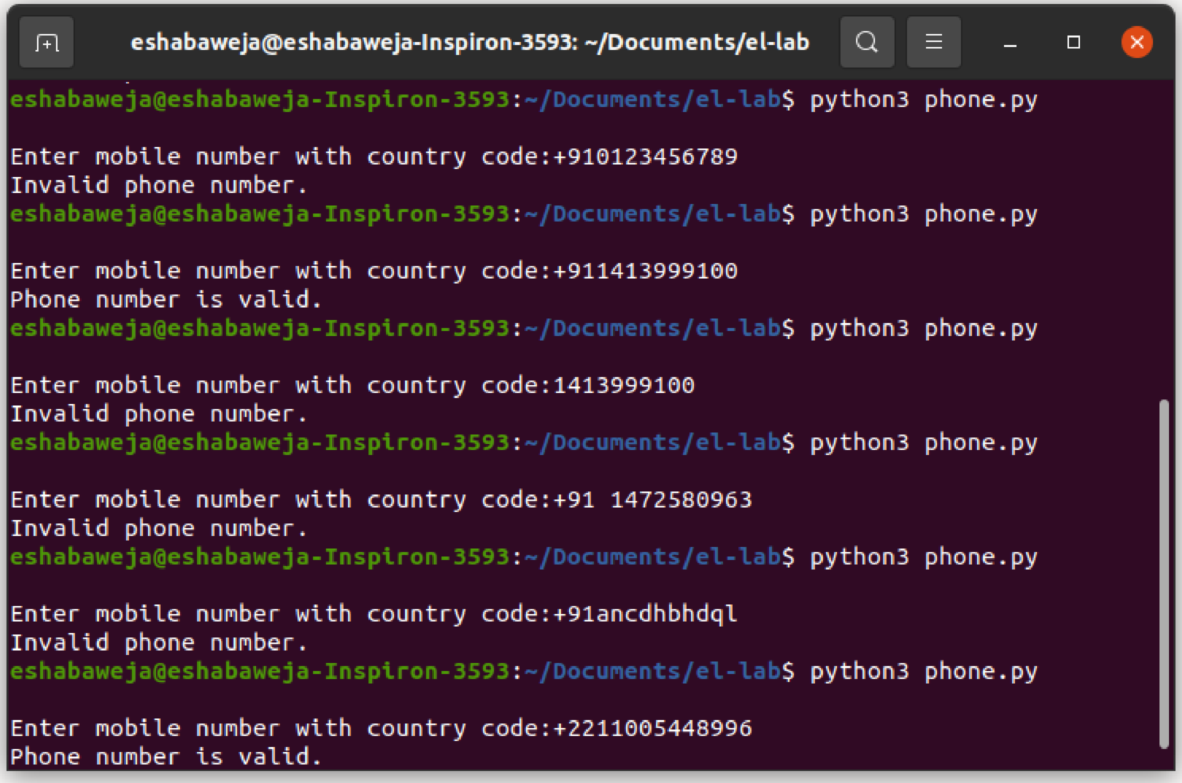
```
else:
        print("Password length should be between 8 to 16 characters.")

if (count_up==1 and count_low==1 and count_num ==1 and
count_spec==1):
        print("Password validated.")

else:
        print("Invalid password.")
```



```
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 password.py

Enter password containing atleast one digit, one special character, one uppercase, and o
ne lowercase letter with length between 8 and 16 characters:$$hellO3
Password validated.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 password.py

Enter password containing atleast one digit, one special character, one uppercase, and o
ne lowercase letter with length between 8 and 16 characters:sherL0ck
Invalid password.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 password.py

Enter password containing atleast one digit, one special character, one uppercase, and o
ne lowercase letter with length between 8 and 16 characters:_aBc-1.2%3
Password validated.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 password.py

Enter password containing atleast one digit, one special character, one uppercase, and o
ne lowercase letter with length between 8 and 16 characters:123
Password length should be between 8 to 16 characters.
Invalid password.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 password.py

Enter password containing atleast one digit, one special character, one uppercase, and o
ne lowercase letter with length between 8 and 16 characters:abc 123#$HEllo
Password should not contain white spaces.
Invalid password.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ 
```

```
#A simple program to check a phone number

num = input("\nEnter mobile number with country code:")
mob,coun = 0,0  #flags
country_codes =
["+93","+355","+213","+1-684","+376","+244","+1-264","+672","+1-2
68","+54","+374","+297",

"+61","+43","+994","+1-242","+973","+880","+1-246","+375","+32","
+501","+229","+1-441",

"+975","+591","+387","+267","+55","+673","+359","+226","+257","+8
55","+237","+1","+238",

"+1-345","+236","+235","+56","+86","+53","+61","+57","+269","+243
","+242","+682","+506",

"+225","+385","+53","+357","+420","+45","+253","+1-767","+1-809","
+1-829","+670","+593",

"+20","+503","+240","+291","+372","+251","+500","+298","+679","+3
58","+33","+594","+689",

"+241","+220","+995","+49","+233","+350","+30","+299","+1-473","+
590","+1-671","+502",

"+224","+245","+592","+509","+504","+852","+36","+354","+91","+62
","+98","+964","+353",

"+972","+39","+1-876","+81","+962","+7","+254","+686","+850","+82
","+965","+996","+856",
"+371","+961","+266","+231","+218","+423","+370","+352","+853","+
389","+261","+265","+60",

"+960","+223","+356","+692","+596","+222","+230","+269","+52","+6
91","+373","+377","+976",

"+1-664","+212","+258","+95","+264","+674","+977","+31","+599","+
687","+64","+505","+227",

"+234","+683","+672","+1-670","+47","+968","+92","+680","+970","+
507","+675","+595","+51",

"+63","+48","+351","+1-787","+1-939","+974","+262","+40","+7","+2
50","+290","+1-869","+263",

"+1-758","+508","+1-784","+685","+378","+239","+966","+221","+248
","+232","+65","+421",

"+386","+677","+252","+27","+34","+94","+249","+597","+268","+46"
,"+41","+963","+886","+967",

"+992","+255","+66","+690","+676","+1-868","+216","+90","+993","+
1-649","+688","+256","+380",

"+971","+44","+1","+598","+998","+678","+418","+58","+84","+1-284
","+1-340","+681","+260"]

if (len(num)>=10 and len(num)<=16):

        country_code = num[0:-10] #using string slices
        mobile_no = num[-10:len(num)]  #the last ten digits would be the
mobile number
```

```
    if (mobile_no.isdigit() and mobile_no[0] != '0'):     #checking mobile number is valid
    mob = mob+1

    for i in country_codes:        #checking country code is valid
    if (i == country_code):
    coun = coun+1

if (mob==1 and coun ==1):
    print("Phone number is valid.")

else:
    print("Invalid phone number.")
```



```
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:+910123456789
Invalid phone number.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:+911413999100
Phone number is valid.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:1413999100
Invalid phone number.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:+91 1472580963
Invalid phone number.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:+91ancdhbhdql
Invalid phone number.
eshabaweja@eshabaweja-Inspiron-3593:~/Documents/el-lab$ python3 phone.py

Enter mobile number with country code:+2211005448996
Phone number is valid.
```

CONCLUSION

Python is very versatile. Being a high-level language, it makes programs easy to develop and understand. The purpose of my project was to verify email-ids, passwords, and phone numbers. This project helped me brush-up on my programming skills, and also helped me practice how to execute commands using the Linux terminal. I gained experience in error-handling in Python as well.

REFERENCES

- https://ubuntu.com/
- www.python.org
- docs.python.org › using › unix
- https://github.com/mailcheck/mailcheck/wiki/List-of-Popular-Domains
- https://www.w3schools.com/python/gloss_python_string_slice.asp