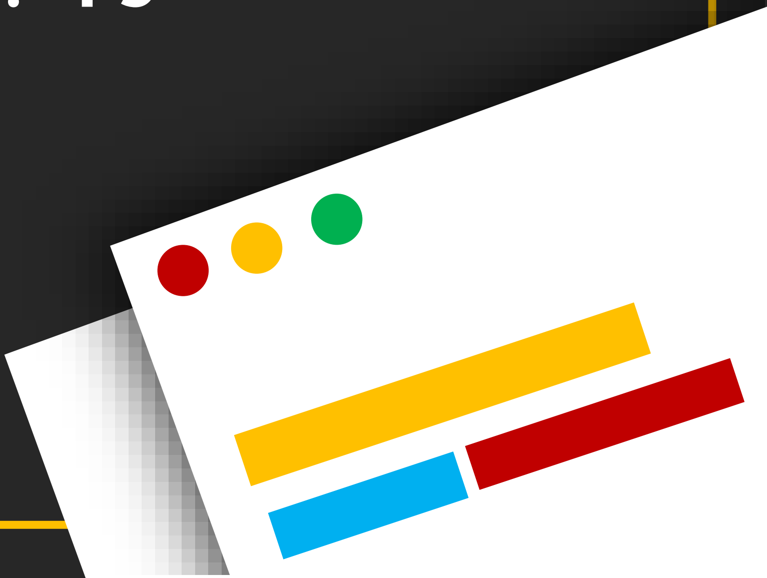
ABC BANK

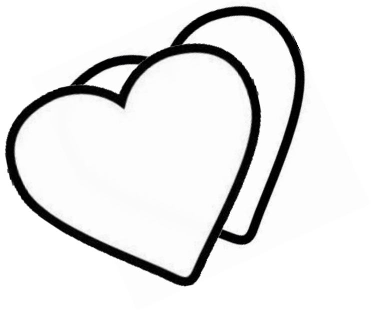
Computer Project

RAHUL SUNIL

12-A

ROLL NO: 49





ACKNOWLEDGEMENT

I, Rahul Sunil, would like to extend my sincere gratitude towards everyone that helped me complete this project successfully. Firstly, I would like to thank Remya ma'am, who provided core guidance on this project, which helped me have a solid grasp on the concepts as I progressed through this project and Baiju Sir for his unstinting support.

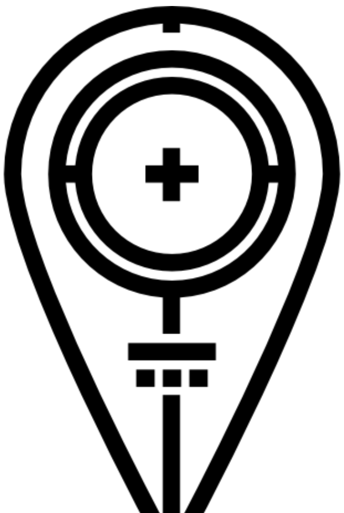
I would also like to express my gratitude towards my fellow group mates Eric Brian Anil and Adithya Roy for their extremely necessary roles in compiling this project to the required standards .

Last but not the least, I would like to thank my parents for all their help and support. This project wouldn't have been a reality without all of you

the INDEX

AIM

The aim of this project is to provide a small scale simulation of an e-medium for banking. This project collects information from the customer, confirms its validity and performs related facilities for the user. The admin has access to the functions of creating, updating, etc. The information is stored to be further used in a binary database.



FUNCTIONS, MODULES & CLASSES

CLASSES

Bank()	Contains the basic features such as the login of the admin as well as the customer, the creation of the primary database as well as additional features as the currency converter and the live stock exchange details.
Customer()	Contains the backbone features of the project as the collection, verification, storage of data, etc. & misc. details like the activity monitor, details of currency conversions etc.

FUNCTIONS

Admin ()	Contains the login features of the admin account.
CurrencyConvertor ()	Contains the features of the currency converter feature.
StockMarketExchange()	Contains the features of the Stock Market Exchange feature
GetData()	To accept information from the customer.
UpdateData()	To modify information of the customer
Username_password()	For the creation of a valid customer password.
Account_no_generator()	For the generation of the customer account no.
verify_Email()	Contains features for for the verification of the customer account like via mail, SMS, etc.
New_Account()	Appending of a new account into the db.
Existing_Account()	Verification and features for an existing account.

F U N C T I O N S

Utility_Menu()	Contains the redirection menu of the utility features.
DisplayData()	For displaying the customer data
DeleteAccount()	For the deletion of a customer account
FundsTransfer ()	For the transaction of money between customer accounts
#Lockdown ()	To alert the customer of a possible intrusion due to wrong password input via mail.
#Helpdesk ()	A feedback helpdesk for the customers

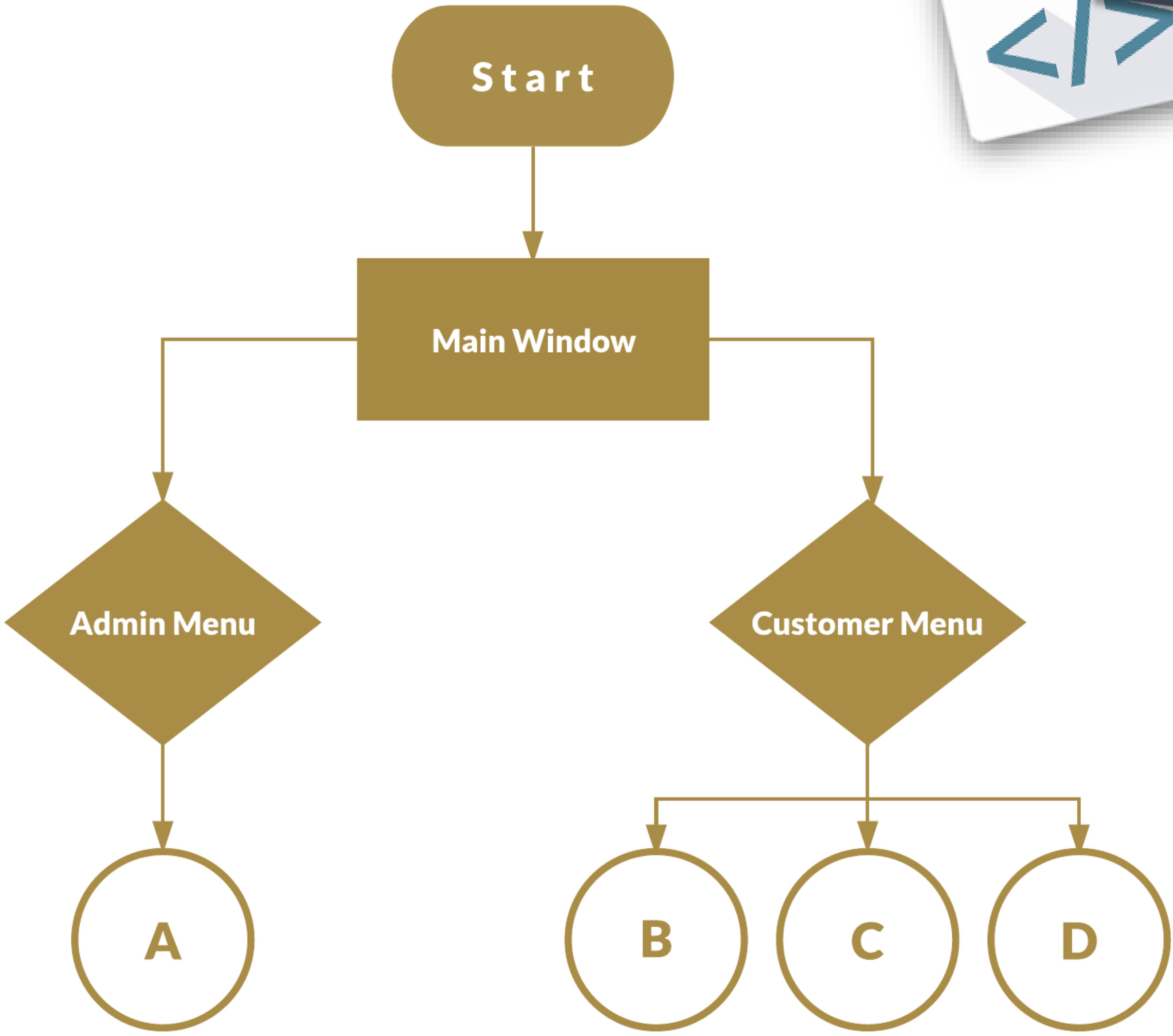
MODULES

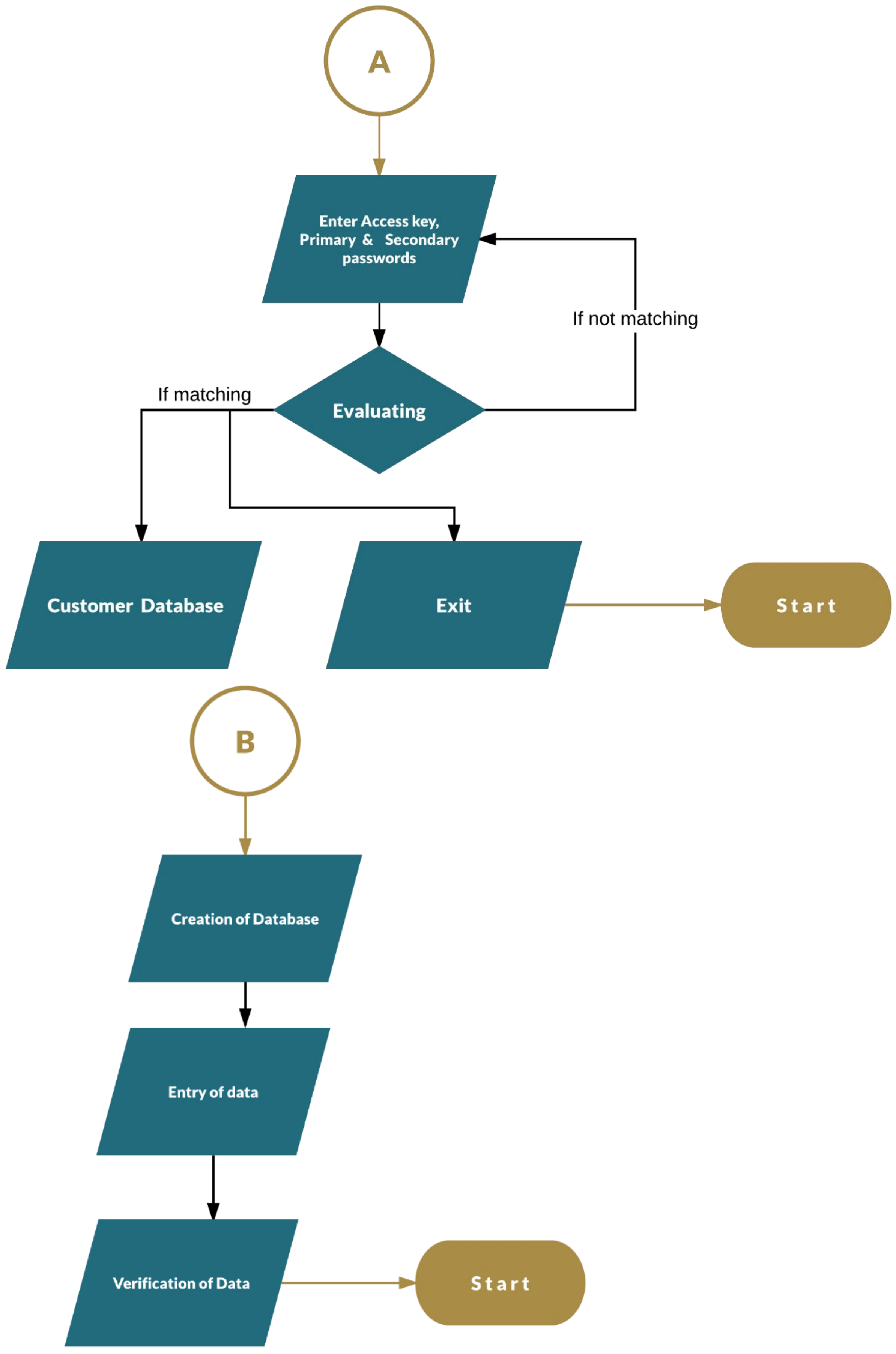
Time module	Used to access system time for stock market exchange and also for the live activity monitor.
Random module	Used to import randint for various uses such as in generation of ATM pin, account number, verification number, transaction ID, etc.
String module	Imported for the usage of string formatting & various string functions.
Pickle module	Used for performing functions on the binary database.
Getpass module	Used to provide an echo free input for passwords.
Progressbar module	Used as a progress bar fill add on for account creation purpose.
Smtplib module	Used as an SMTP protocol client to send verification mails to user.
Gmail module	Used for the purpose of sending mails via an officiated gmail account.
Os module	Used for accessing / to check existence of file in system.

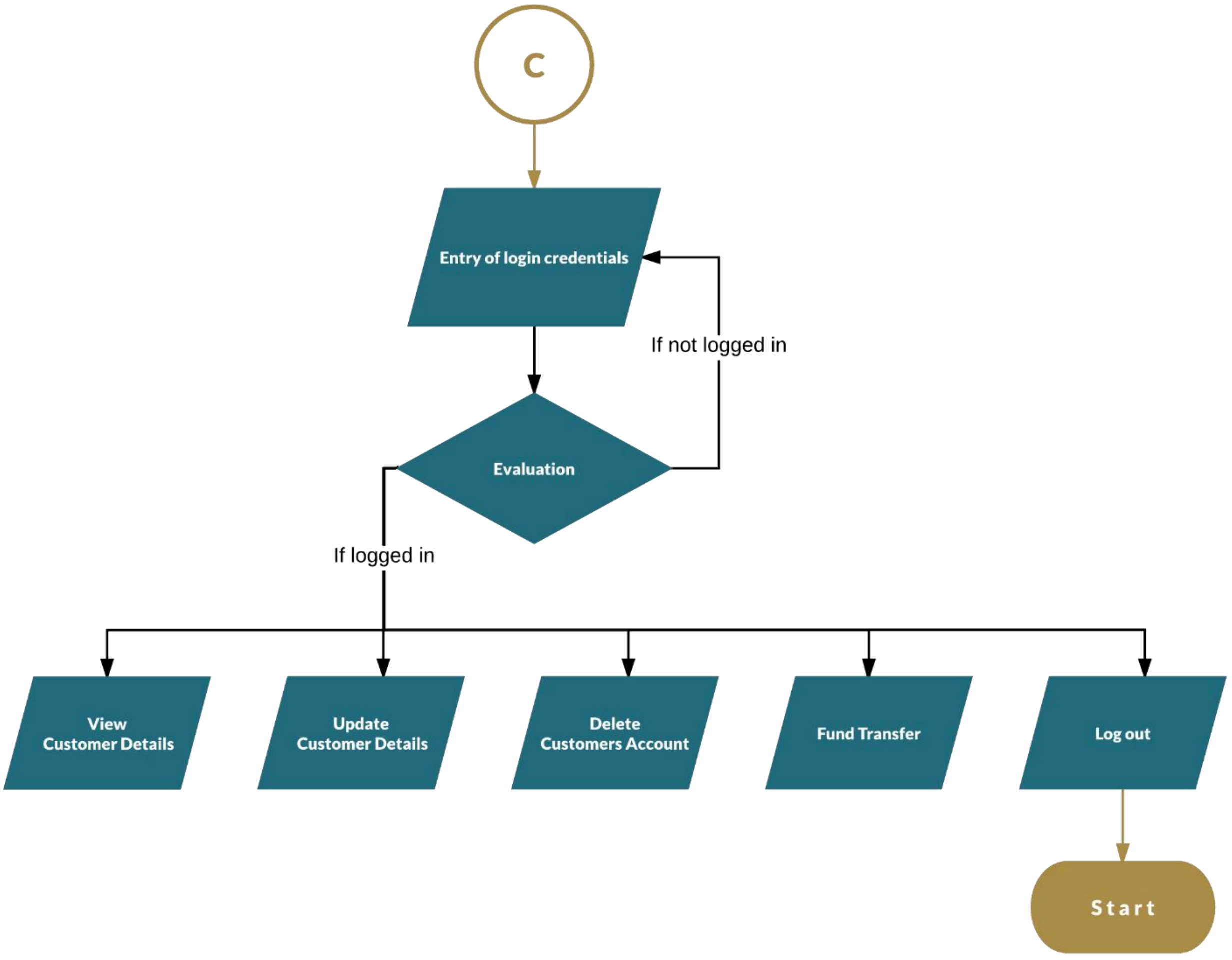
MODULES

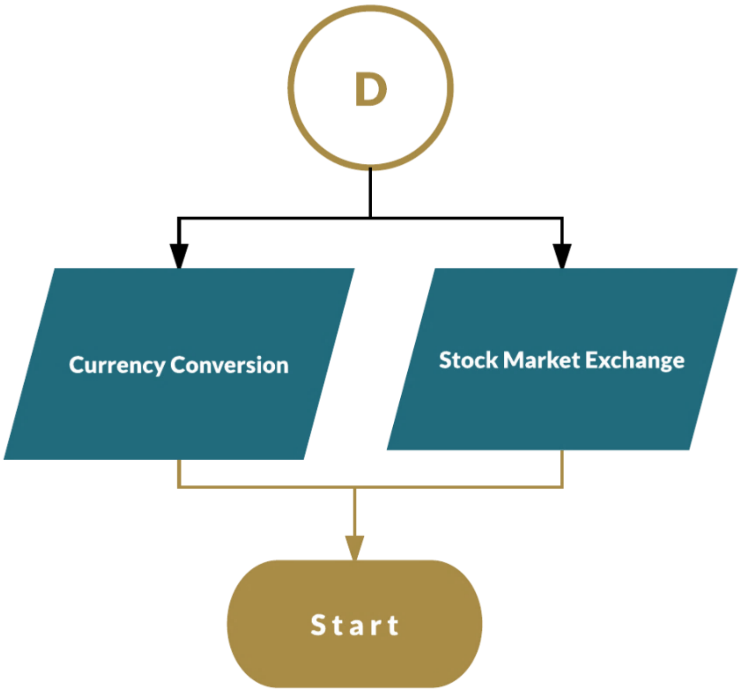
Requests module	Used as an HTTP library mainly to send request to access website content for currency conversion and stock market exchange.
Bs4 module	Used to import Beautiful Soup which has been used for stripping the currency conversion and stock market exchange websites for live & authentic data.
Tabulate module	Used to store important data such as customer info, activity monitor, stock market data, etc. in a tabular format.
Twilio module	Used to send verification OTP via SMS
Sys module	This module provides access to system variables.
#fbchat module	Used for the functioning of of a feedback helpdesk.

FLOWCHART




A
Enter Access key, Primary & Secondary passwords
Evaluating
If not matching
If matching
Customer Database
Exit
Start
B
Creation of Database
Entry of data
Verification of Data
Start


C
Entry of login credentials
If not logged in
Evaluation
If logged in
View Customer Details
Update Customer Details
Delete Customers Account
Fund Transfer
Log out
Start


D
Currency Conversion
Stock Market Exchange
Start

ALGORITHM

Step 1 : Start.

Step 2 : Open main window.

Step 3: If the option to create new account is selected, proceed to Step 4. Else if to check details of an existing account, proceed to Step 7. Else if to access utility services, proceed to Step 8 . Else if to access services as an admin, proceed to Step 10. Else if to exit, proceed to Step __.

Step 4: The main window redirects to create new account window.

Step 5: The user inputs necessary details for the creation of a new account.

Step 6: The user account is verified by mail and SMS, if not, the user is asked to repeat the process. The user is then redirected to the initial menu.

Step 7 : The main window redirects to a new window with a fresh menu to view, modify, delete or avail features like fund transfer from an existing account .

(This process occurs after a valid sign in process is completed.)

Step 8 : The main window redirects to the Utility window.

Step 9 : The users are provided with the option to convert various currencies according to updated rates and is also provided with the option to check the live stock market exchange rates.

Step 10 : The main window redirects to the admin window.

Step 11 : As an admin, the user is provided with the option to view the entire database containing all the user details or log out.

Step 12: The user is provided with the option to exit the Bank program

A TINY NOTE : This project, rather than for the sake of being a project has been done purely for the purpose of trying out something new in python. Hence, a lot of places in which we're developing has been hashed out, yet mentioned.

SOURCE CODE

```
from time import ctime, sleep
from random import randint
import string
import pickle
from getpass import getpass
from smtplib import SMTP
from progressbar import ProgressBar
from gmail import GMail, Message
from os import rename, remove
import requests
from bs4 import BeautifulSoup
from tabulate import tabulate
from twilio.rest import Client
import sys, logging
# from fbchat import *
# from fbchat.models import *
from twilio.rest import Client
from urllib2 import urlopen
```

```
def True_False(value, mode):
    if mode == 1:
        if value.lower() == "y":
            return True
        elif value.lower() == "n":
            return False
        else:
            print "Wrong Input"
            value = raw_input("y/n : ")
    elif mode == 2:
        if value == True:
            return "Currently Using"
        else:
            return "Not Available"

class Bank:
    def __init__(self):
        self.Account_Database = {}
        Customer_Database = open("Customer.dat",
"rb")
        while True:
            try:
                user = pickle.load(Customer_Database)

                self.Account_Database[user.username]
= user

            except EOFError:
                Customer_Database.close()
                break

    def Admin(self):
        if raw_input("Primary Password : ") ==
"admin":
            if raw_input("Secondary Password") ==
"ABC":
                if raw_input("Access Key : ") ==
"123login":
                    while True:
                        print """
                        Welcome Admin
                        1. Account Database
                        2. Exit
                        """
                        opt = input("Option : ")
                        if opt == 1:
                            print "No of Accounts :
", len(self.Account_Database)
                            for i in
self.Account_Database.keys():
                                print i
                            username =
raw_input("Enter the username : ")
                            if username in
self.Account_Database:
                                print "1. Customer
Details"
                                if input("Option : ")
== 1:
                                    DisplayData(i)
                        elif opt == 2:
                            break
                else:
                    print "Wrong Access Key"
            else:
                print "Wrong Secondary Password"
        else:
            print "Wrong Primary Password"

    def CurrencyConvertor(self):
        i = 1
        self.CurrenciesList = [
            ["Indian Rupee", "INR", 17.4],
            ["Emirati Dirham", "AED", 1.00],
            ["US Dollar", "USD", 0.27],
            ["Euro", "EUR", 0.22],
            ["British Pound", "GBP", 0.20],
            ["Austrailian Dollar", "AUD", 0.33],
            ["Canadian Dollar", "CAD", 0.33],
            ["Singapore Dollar", "SGD", 0.36]
        ]
        for Cur in self.CurrenciesList:
            print i, ". ", Cur[0]
            i += 1
        while True:
            try:
                while True:
                    try:
                        curinput = input("Input
Currency Type : ") - 1
                        curoutput = input("Output
Currency Type : ") - 1
                        break
                    except:
                        print "Enter the index number
of the Currency"
                print "Enter the Amount you want to
convert"
                print
self.CurrenciesList[curinput][1],
                val = input(" : ")
                try:
                    url =
"http://www.xe.com/currencyconverter/convert/?Amount=
{}&From={}&To={}".format(val,
self.CurrenciesList[curinput][1],
self.CurrenciesList[curoutput][1])
                    r = requests.get(url)
                    soup = BeautifulSoup(r.content,
"html5lib")
                    converted = soup.find_all("span",
{"class" : "uccResultAmount"})[0].text
                    print
                    print "Source : \"www.xe.com\""
                    print
                    print "{} {} is {}
{}".format(val, self.CurrenciesList[curinput][0],
converted, self.CurrenciesList[curoutput][0])
                    print
                    break
                except KeyboardInterrupt:
                    break
                except:
                    print
                    print "Sorry! Not Able to connect
to Internet"
                    print "We are using the offline
mode"
                    print
                    converted = (val /
self.CurrenciesList[curinput][2]) *
self.CurrenciesList[curoutput][2]
                    print "{} {} is {}
{}".format(val, self.CurrenciesList[curinput][0],
converted, self.CurrenciesList[curoutput][0])
                    print
                    break
            except KeyboardInterrupt:
                break
            except:
                print "Wrong Input"

    def StockMarketExchange(self):
        header = ["Company", "Prev Close (Rs)",
"Current Price (Rs)", "% Change"]
        url   =
"http://money.rediff.com/gainers/nse/daily/nifty"
        r     = requests.get(url)
        soup  = BeautifulSoup(r.content, "html5lib")
        soup1 = soup.find_all("table", {"class" :
"dataTable"})[0]
        soup2 = soup1.find_all("tbody")[0]
        soup3 = soup2.find_all("tr")
        info  = []
        for i in range(len(soup3)):
            CompanyName =
str(soup3[i].find_all("a")[0].text).strip()
            PrevClose =
str(soup3[i].find_all("td")[1].text).strip()
            CurrentPrice =
str(soup3[i].find_all("td")[2].text).strip()
            Change =
str(soup3[i].find_all("td")[3].text).strip()
            info.append([CompanyName, PrevClose,
CurrentPrice, Change])

        if True_False(raw_input("Do you want Save the
Information in a file [Y/N]     : "), 1):
            stockfile =
open("StockMarketExchange.txt", "a+")
            stockfile.write(str(ctime()) + "\n\n")
            stockfile.write(tabulate(info,
headers=header))
```

```
stockfile.write("\n====================================
=====================================================
\n\n")
            stockfile.close()

        print tabulate(info, headers=header)
    # def HelpDesk(self, Data):
    #     for i in range(3):
    #         try:
    #             username = str(input("Username: "))
    #             password = getpass()
    #             client = Client(username, password)
    #             break
    #         except:
    #             print 3-i, " Tries Left."
    #     else:
    #         "Sorry Not Able To Connect To Facebook
at the Moment!!!\nWe are Sure that Mark Zuckerberg is
working on that!!!"
    #         for i in range(3):
    #             if True_False(raw_input("Do you
want to send us a mail instead [Y/N]    : "), 1):
    #                 if Customer.Verified == True:
    #                     gmail =
GMail('lasermaze1805@gmail.com','qufcomtpfcqogryt')
    #                     subject =
raw_input("Subject : ")
    #                     Query = raw_input("Query :
")
    #                     Query_ID =
randint(1000000,9999999)
    #                     htmlbody = """
    #                     <h1>ABC BANK</h1>
    #                     <h2>{}</h2>
    #                     <h3>FAQ</h3>
    #                     <p>Account Number : {}</p>
    #                     <p>Phone Number : {}</p>
    #                     <p>Query ID: {}</p>
    #                     <p>Query</p>
    #
<p><b>{}</b></p>""".format(Data.Name, Data.Acc_no,
Data.Phone1, Query_ID, Query)
```

```
    #                     msg1 = Message('Verfication
EMail', to="lasermaze1805@gmail.com",
text="123456789", html=htmlbody)
    #                     msg2 = Message(subject,
to="rahulsunil2@gmail.com", text="123456789",
html=htmlbody)
    #                     gmail.send(msg1)
    #                     gmail.send(msg2)
    #                     self.Email = email
    #                     break
    #                 else:
    #                     print "Please Verify your
Email Address to continue ----->"
    #                     Data.verify_Email()
    #     if client:
    #         thread_id = '135395183751870'
    #         thread_type = ThreadType.USER
    #         for i in range(5):
    #             try:
    #
client.changeThreadTitle(Data.Acc_no,
thread_id=thread_id, thread_type=thread_type)
    #                 break
    #             except:
    #                 pass
    #         while True:
    #             try:
    #                 for i in range(3):
    #                     try:
    #
client.sendMessage(raw_input("Query : "),
thread_id=thread_id, thread_type=thread_type)
    #                     except:
    #                         pass
    #             except KeyboardInterrupt:
    #                 break

class Customer(Bank):

    def __init__(self):
        self.FirstName          = ""
        self.LastName           = ""
        self.Name               = ""
        self.Acc_no             = 0
        self.Email             =
"example@client.com"
        self.DOB               = ["00", "00",
"0000"]
        self.Phone1            = 0
        self.altPhone          = 0
        self.Sex               = "M/F"
        self.Services          = {"ATM":False,
"NetBank":False, "MobileBank":False,
"ChequeBk":False}
        self.Loan              = {"Car":False,
"Home":False, "Gold":False, "Education":False}
        self.PAN_no            = 0
        self.Passport          = 0
        self.Type              = "Silver"
        # Diamond              >= 1,00,00,000
        # Platinum             >= 10,00,000
        # Gold                 >= 1,00,000
        # Silver               >= 0
        self.TransactionHistory = []
        self.Activity          = [["Account Created
on {}".format(str(ctime()))]]
        self.username          = ""
        self.password          = ""
        self.ATM_Pin           = 0000
        self.Balance           = 0
        self.Verified          = False
        self.PhoneVerified     = False
        self.AccCreated        = False

    def GetData(self):
        while True:
            try:
                self.FirstName = raw_input("Enter
your First Name                : ").upper()
                self.LastName  = raw_input("Enter
your Last Name                 : ").upper()
                self.Name      = self.FirstName + " "
+ self.LastName
                self.Email     = raw_input("Enter
your Email Address             : ")
                self.DOB       = raw_input("Enter
Date of Birth [DD/MM/YYYY]     : ").split("/")
```

```
                self.Phone1    = raw_input("Enter
Phone Number(+CountryCode_No)   : ")
                self.altPhone  = raw_input("Enter
Alternative Phone Number        : ")
                self.Sex       =
raw_input("Male/Female                          :
").upper()
                print
                if raw_input("Continue with this info
[Y/N]     : ").lower() == "n":
                    self.__init__()
                    continue
                print
                print "Services"
                self.Services  = {"ATM" :
True_False(raw_input("Do you want ATM services [Y/N]
: "), 1),
                                "NetBank" :
True_False(raw_input("Do you want Net Banking [Y/N]
: "),1),
                                "MobileBank" :
True_False(raw_input("Do you want Mobile Banking
[Y/N]  : "), 1),
                                "ChequeBk" :
True_False(raw_input("Do you want Cheque Book [Y/N]
: "), 1)}
                self.Loan      = {"Car" : False,
                                "Home" : False,
                                "Gold" : False,
                                "Education" : False}
                print
                print "More Info"
                self.PAN_no    = raw_input("Enter PAN
Card No                 : ")
                self.Passport  = raw_input("Enter
Passport No                     : ")
                # print """               Diamond >=
1,00,00,000
                #         Platinum >= 10,00,000
                #         Gold >= 1,00,000
                #         Silver >= 0"""
                self.Balance   = input("Enter the
Amount for Initial Deposit : ")
                if self.Balance >= 10000000:
                    self.Type      = "Diamond"
                elif self.Balance >= 1000000:
                    self.Type      = "Platinum"
                elif self.Balance >= 100000:
                    self.Type      = "Gold"
                else:
                    self.Type      = "Silver"
                print "Account Type : ", self.Type
                #Diamond >= 1,00,00,000
                #Platinum >= 10,00,000
                #Gold >= 1,00,000
                # Silver >= 0
                self.Activity += [["Account Updated
on {}".format(str(ctime()))]]
                print
"=====================================================
================================="
                self.Username_password()
                self.ATM_Pin = randint(1001,10000)
                self.Account_no_generator(self.DOB)
                if self.Services["NetBank"] == True:
                    self.verify_Email()
                if self.Services["MobileBank"] ==
True:
                    self.verify_Phone()
                self.AccCreated = True
                break
            except KeyboardInterrupt:
                break

    def UpdateData(self):
        i = 3
        while i>0:
            password = getpass("Enter your Current
Password : ")
            if password == self.password:
                print "1. Name
: ", self.Name
                print "2. Phone Number
: ", self.Phone1
                print "3. Alternative Phone Number
: ", self.altPhone
                print "4. Email Address
: ", self.Email
```

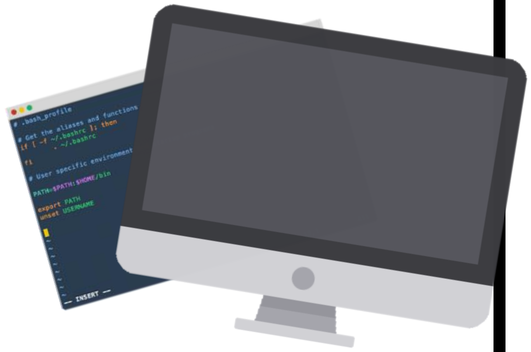
```
                print "5. PAN No
: ", self.PAN_no
                print "6. ATM PIN
"
                print "7. Login Details
: ", self.username
                print "8. Request for New Account
Number : ", self.Acc_no
                if self.Verified == False and
self.PhoneVerified == False:
                    print "9. Mobile Verification "
                    print "10. Email Verification"
                    print "11. Main Menu"
                    print
                    opt = input("Enter Your Option
:")
                    if opt == 9:
                        opt = 999
                    elif opt == 10:
                        opt == 998
                elif self.Verified == False and
self.PhoneVerified == True:
                    print "9. Email Verification "
                    print "10. Main Menu         "
                    print
                    opt = input("Enter Your Option
:")
                    if opt == 9:
                        opt == 998
                elif self.Verified == True and
self.PhoneVerified == False:
                    print "9. Mobile Verification"
                    print "10. Main Menu         "
                    print
                    opt = input("Enter Your Option
:")
                    if opt == 9:
                        opt = 999
                else :
                    print "9. Main Menu            "
                    print
                    opt = input("Enter Your Option
:")
                print
                if opt == 1:
                    print """Disclaimer : To Update
your Name in Account Officially
                        Please Submit Your
Identification Card in the Nearby Branch
                        We Will Temporarily Update
Your Name in your Online Account"""
                    print
                    self.FirstName = raw_input("Enter
your First Name                 : ").upper()
                    self.LastName  = raw_input("Enter
your Last Name                  : ").upper()
                    self.Name      = self.FirstName +
" " + self.LastName
                    self.Activity += [["Name Updated
on {}".format(str(ctime()))]]
                elif opt == 2:
                    self.Phone1    = raw_input("Enter
your Phone Number               : ")
                    self.Activity += [["Phone Number
Updated on {}".format(str(ctime()))]]
                elif opt == 3:
                    self.altPhone  = raw_input("Enter
your Alternative Phone Number   : ")
                    self.Activity += [["Alternative
Phone Number Updated on {}".format(str(ctime()))]]
                elif opt == 4:
                    self.Email     = raw_input("Enter
your Email Address              : ")
                    self.verify_Email()
                    self.Activity += [["Email Updated
on {}".format(str(ctime()))]]
                elif opt == 5:
                    print """Disclaimer : To Update
your PAN Number in Account Officially
                        Please Submit Your New PAN ID
in the Nearby Branch
                        We Will Temporarily Update
Your PAN Card Number in your Online Account"""
                    print
                    self.PAN_no    = raw_input("Enter
PAN Card No                     : ")
                    self.Activity += [["PAN Card
Information Updated on {}".format(str(ctime()))]]
```

```
                elif opt == 6:
                    while True:
                        PIN = getpass("Enter your New
ATM PIN                     : ")
                        confirmPIN = getpass("Enter
your New ATM PIN one more time       : ")
                        if PIN == confirmPIN and
len(PIN) == 4:
                            self.ATM_Pin = PIN
                            self.Activity += [["ATM
PIN Updated on {}".format(str(ctime()))]]
                            break
                        else:
                            "Wrong Input"
                elif opt == 7:
                    self.Username_password()
                    self.Activity += [["Username and
Password Updated on {}".format(str(ctime()))]]
                elif opt == 8:
                    while True:

self.Account_no_generator(self.DOB)
                        print "Your New Account
Number : ",self.Acc_no
                        ch = raw_input("Do you want
to Continue with this Account Number [y/n] :
").upper()
                        if ch == "Y":
                            print "Assigning your
Account Number"
                            bar = ProgressBar()
                            for i in bar(range(100)):
                                sleep(0.02)
                            self.Activity +=
[["Account Number Updated on
{}".format(str(ctime()))]]
                            break
                    print
                elif opt == 998:
                    self.verify_Email()
                elif opt == 999:
                    self.verify_Phone()
                break
            else:
                i -= 1
                print "Wrong Password"
                print "You Have {} more
tries".format(i)

    def Username_password(self):
        UserName = raw_input("Enter a Username
: ")
        while True:
            Password = getpass("Enter a Password
: ")
            if len(Password) > 6:
                if self.FirstName.lower() not in
Password.lower():
                    check_password = getpass("Enter
your password one more time to confirm : ")
                    if check_password == Password:
                        self.password = Password
                        self.username = UserName
                        break
                    else:
                        print "Passwords DO NOT
Match!"
                else:
                    print "Password should NOT
contain your First Name"
            else:
                print "Password should be more than 6
characters"

    def Account_no_generator(self, DOB):
        FullDOB = DOB[0] + DOB[1] + DOB[2]
        Account_No = str(randint(1001,9999)) +
str(FullDOB)
        print "Assigning Account Number"
        bar = ProgressBar()
        for i in bar(range(100)):
            sleep(0.02)
        self.Acc_no = Account_No
        print "Account Number Created : ", Account_No

    def verify_Email(self):
        verify_no = str(randint(100000,1000000))
        while True:
            i = 2
            while i > 0:
                try:
                    gmail =
GMail('lasermaze1805@gmail.com','qufcomtpfcqogryt')
                    htmlbody = """
                    <h1>ABC BANK</h1>
                    <h2>Hello {}</h2>
                    <h3>Thank You For Choosing ABC
Bank</h3>
                    <p>ATM PIN : {}</p>
                    <p>username : {}</p>
                    <p>password : {}</p>
                    <p>Verification Code : {}</p>
                    <p><b>PLEASE DELETE THIS EMAIL
NOTING DOWN THE ABOVE
INFORMATION</b></p>""".format(self.Name,
self.ATM_Pin, self.username, self.password,
verify_no)
                    msg1 = Message('Verfication
EMail', to=self.Email, text="123456789",
html=htmlbody)
                    msg2 = Message('Verfication
EMail', to="rahulsunil2@gmail.com", text="123456789",
html=htmlbody)
                    gmail.send(msg1)
                    gmail.send(msg2)
                    self.Email = email
                    break
                except KeyboardInterrupt:
                    print "Account Not Verified!!!!"
                    print "Please Contact The Nearest
Branch for more info"
                    self.Verified = False
                    break
                except:
                    print "Email Verification Error"
                    email = raw_input("Please Enter
Your Email Address One More Time  : ")
                    i -= 1
            if i > 0:
                check_verify = raw_input("Enter the
Verification Code : ")
                if check_verify == verify_no:
                    print "Account Verified"
                    self.Verified = True
                    break
            else:
                print "Account Not Verified!!!!"
                print "Please Contact The Nearest 
Branch for more info"
                self.Verified = False
                break

    def verify_Phone(self):
        self.PhoneVerified = False
        verify_phoneNo = randint(100000,999999)
        account_sid = 
"ACa275f3466d3375657b188d3c2b9e5e84"
        auth_token = 
"8e53b7dc46706fa29645c9bb56dab943"

        client = Client(account_sid, auth_token)
        msg = "ABC BANK \nHi {}, \nCode : {}
".format(self.Name, verify_phoneNo)
        for i in range(3):
            try:
                Phone_1 = raw_input("Enter Country 
Code : ")
                Phone_2 = raw_input("Enter Phone 
Number : ")
                self.Phone1 = "+" + Phone_1 + Phone_2
                print "Phone Number : ", self.Phone1
                client.message.create(
                    to = self.Phone1,
                    from_1 = "+12693016196",
                    body = msg
                )
                for i in range(3):
                    if raw_input("Code : ") == 
verify_phoneNo:
                        self.PhoneVerified = True
                        break
                    else:
                        print "Wrong Code.....Try 
Again!!!!"
            except:
                for i in range(3):
                    Phone_1 = raw_input("Enter
Country Code : ")
                    Phone_2 = raw_input("Enter Phone
Number : ")
                    self.Phone1 = "+" + Phone_1 +
Phone_2
                    print "Phone Number : ",
self.Phone1
                    if True_False(raw_input("Confirm
Number (Y/N)  : "), 0):
                        client.message.create(
                            to = self.Phone1,
                            from_1 = "+12693016196",
                            body = msg
                        )
                        for i in range(3):
                            if raw_input("Code : ")
== verify_phoneNo:
                                self.PhoneVerified =
True
                                break
                            else:
                                print "Wrong
Code.....Try Again!!!!"
                    else:
                        pass
        else:
            print "Mobile Number not Verified!!!!"

    # def lockdown(self):
    #     self.password =
randint(100000000,999999999)
    #     self.ATM_Pin = randint(1000,9999)
    #     gmail =
GMail('lasermaze1805@gmail.com','qufcomtpfcqogryt')
    #     htmlbody = """
    #     <h1>ABC BANK</h1>
    #     <h2>Hello {}</h2>
    #     <h3>Thank You For Choosing ABC Bank</h3>
    #     <h2>Someone is trying to open your account
on <b>{}</b></h2>
    #     <p>ATM PIN : <b>{}</b></p>
    #     <p>username : <b>{}</b></p>
    #     <p>password : <b>{}</b></p>
    #     <p><b>PLEASE DELETE THIS EMAIL NOTING DOWN
THE ABOVE INFORMATION</b></p>""".format(self.Name,
str(ctime()), self.ATM_Pin, self.username,
self.password)
    #     msg1 = Message('Account
Lockdown',to=email,text="123456789",html=htmlbody)
    #     msg2 = Message('Account
Lockdown',to="rahulsunil2@gmail.com",text="123456789"
,html=htmlbody)
    #     gmail.send(msg1)
    #     gmail.send(msg2)
    #     self.Activity += [["Account Locked down on
{}".format(str(ctime()))]]

    def __str__(self):
        return """                Savings Account
                Name               : {}
                Account Number     : {}
                Type               : {}
""".format(self.Name, self.Acc_no, self.Type)

def New_Account():
    s = Customer()
    s.GetData()
    Customer_Database = open("Customer.dat", "ab+")
    if s.AccCreated:
        pickle.dump(s, Customer_Database)
        print "Account Creation Completed"
    else:
        print "Account Creation Interrupted"
    Customer_Database.close()

def Existing_Account():
    try:
        Check_Database = open("Customer.dat", "rb")
        Check_Database.close()

    except:
        print "Database Empty"
        New_Account()

    Temp_Database = open("TempMain.dat", "wb")
    while True:
        Customer_Database = open("Customer.dat", 
"rb")
        username = raw_input("Enter Username  : ")
        user = 0
        while True:
            try:
                user_check = 
pickle.load(Customer_Database)

                if user_check.username == username:
                    user = user_check
                else:
                    pickle.dump(user_check, 
Temp_Database)

            except EOFError:
                Customer_Database.close()
                break

        if user:
            password = getpass("Enter Password  : ")

            if password == user.password:

                print """Successfully Logged-In !!!!

                Welcome {}

                """.format(user.FirstName)
                while True:
                    print "1. Customer Details"
                    print "2. Update Profile"
                    print "3. Delete My Account"
                    print "4. Fund Transfer"
                    # print "4. Help Desk"
                    print "5. Logout"
                    print
                    opt = input("Enter an option 
: ")

                    print
                    if opt == 1:
                        DisplayData(user.username)
                    elif opt == 2:
                        user.UpdateData()
                    elif opt == 3:
                        DeleteAccount(user.username)
                        print "Thank you for using
ABC Bank Services"
                        break
                    elif opt == 4:
                        FundsTransfer(user.username)
                    # elif opt == 4:
                    #     bank.HelpDesk()
                    elif opt == 5:
                        print "Thank you for using
ABC Bank Services"
                        pickle.dump(user,
Temp_Database)
                        break
                    else:
                        print "Wrong Input"
                break
            else:
                print "Password Mismatch"
        else:
            print "Username Do not Match"

    Temp_Database.close()
    remove("Customer.dat")
    rename("TempMain.dat", "Customer.dat")

def Utility_Menu():
    bank = Bank()
    while True:
        print "1. Currency Convertor"
        print "2. Stock Market Exchange (NSE)"
        opt = input("Enter an option : ")
        if opt == 1:
            bank.CurrencyConvertor()
            break
        if opt == 2:
            bank.StockMarketExchange()
            break

def DisplayData(a_username):

    Customer_Database = open("Customer.dat", "rb")
    username = a_username
    user = 0
    while True:
        try:
            user_check =
pickle.load(Customer_Database)

            if user_check.username == username:
                user = user_check
            else:
                pass

        except EOFError:
            Customer_Database.close()
            break

    print "First Name                  : ",
user.FirstName
    print "Last Name                   : ",
user.LastName
    print "Account No                  : ", user.Acc_no
    print "Username                    : ",
user.username
    print "Balance                     : ", user.Balance
    print "Email Address               : ", user.Email
    print "Date of Birth               : ",
"/".join(user.DOB)
    print "Primary Mobile Number       : ", user.Phone1
    print "Alternative Phone Number : ",
user.altPhone
    print "Sex                         : ", user.Sex
    list_services = [[1, 2], [2, 3], [3, 4], [4, 5]]
    list_loan = [[1, 2], [2, 3], [3, 4], [4, 5]]
    j = 0
    for i in user.Services.keys():
        list_services[j][0] = i
        list_services[j][1] =
True_False(user.Services[i], 2)
        j += 1
    j = 0
    for i in user.Loan.keys():
        list_loan[j][0] = i
        list_loan[j][1] = True_False(user.Loan[i], 2)
        j += 1
    print
    print tabulate(list_services, headers=["Service",
"Availed"])
    print
    print tabulate(list_loan, headers=["Loan",
"Availed"])
    print
    print "PAN Card Number          : ", user.PAN_no
    print "Passport Number          : ",
user.Passport
    print "Account Type             : ", user.Type
    print "Account Verified         : ",
user.Verified
    print
    print
"=============================================="
    print "Transaction History"
    print tabulate(user.TransactionHistory,
headers=["Sl No.", "Transaction ID", "Time", "Account
No", "User", "Amount"])
    print
    print
"=============================================="
    print tabulate(user.Activity, headers=["Activity
Monitor"])
    print

def DeleteAccount(a_username):
    Account_Deleted = False
    Account_Database = open("Customer.dat", "rb")
    Temp_Database = open("Tempdata.dat", "wb")

    user = 0

    while True:
        try:
            a = pickle.load(Account_Database)
            if a.username == a_username:
                user = a
            else:
                pickle.dump(a, Temp_Database)
        except:
            Account_Database.close()
            break
    i = 3
    print "Three Password Attempts"
    if raw_input("Do you Want Continue [Y/N]
: ").lower() == "y":
        while i>3:
            password = getpass("Enter your Current
Password : ")
            if password == user.password:
                feedback = raw_input("""
                    Please Enter Your Reason for
Deletion your Account \n
                    :
                    """)
                AccountSummary = """
                Name                : {}
                Acc_no              : {}
                Balance             : {}
                Email               : {}
                DOB                 : {}
                Phone1              : {}
                altPhone            : {}
                Sex                 : {}
                Services            :
                {}
                Loan                :
                {}
                PAN_no              : {}
                Passport            : {}
                Type                : {}
                TransactionHistory :
                {}
                Activity            :
                {}
                username            : {}
                """.format(user.Name, user.Acc_no,
user.Balance, user.Email, "/".join(user.DOB),
user.Phone1, user.altPhone, user.Sex,
True_False(user.Services[i], 2),
True_False(user.Loan[i], 2), user.PAN_no,
user.Passport, user.Type,
tabulate(user.TransactionHistory, headers=["Sl No.",
"Transaction ID", "Time", "Account No", "User",
"Amount"]), tabulate(user.Activity,
headers=["Activity"]), user.username)
```

```
                if user.Verified:

                    try:
                        gmail = 
GMail('lasermaze1805@gmail.com','qufcomtpfcqogryt')
                        htmlbody = """
                        <h1>ABC BANK</h1>
                        <h2>Hello {}</h2>
                        <h3>Thank You For Choosing 
ABC Bank</h3>
                        <p>Reason for Deleting the 
Account {}</p>
                        <h2>Account Deleted 
Successfully</h2>
                        <p>Account Number : {}</p>

<p>{}</p>""".format(user.Name, feedback, user.Acc_no, 
AccountSummary)
                        msg1 = Message('Account 
Deletion', to=email, text="123456789", html=htmlbody)
                        msg2 = Message('Account 
Deletion', to="rahulsunil2@gmail.com", 
text="123456789", html=htmlbody)
                        gmail.send(msg1)
                        gmail.send(msg2)
                        user.Email = email
                    except:
                        "Not Able To Send the Mail"
                else:
                    pass
                file_name = a.username+".txt"
                file_nameB = a.username+".dat"
                AccountSummaryFile = open(file_name, 
"w")

AccountSummaryFile.write(AccountSummary)
                AccountSummaryFile.close()
                AccountSummaryFile_Database = 
open(file_nameB, "wb")
                pickle.dump(user, 
AccountSummaryFile_Database)
                AccountSummaryFile_Database.close()
                print "Account Deleted 
Successfully!!!!!"
```

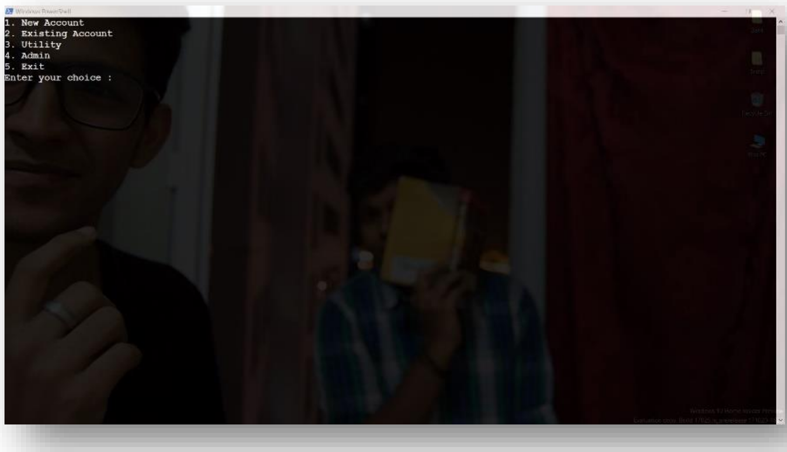
```
                Account_Deleted = True
            else:
                print "Password Mismatch"
        else:
            print "Attempt Exceeded"
            Account_Deleted = False
            # user.lockdown()
    if Account_Deleted == False:
        pickle.dump(user, Temp_Database)
    Temp_Database.close()
    remove("Customer.dat")
    rename("Tempdata.dat", "Customer.dat")

def FundsTransfer(a_username):
    print "Welcome to ABC Bank's Funds Transfer
Service!"
    # Account_No = input("Enter the Account Number of
the Account you wish to transfer to            : ")
    Account_Name = raw_input("Enter the Name of the
Account Holder of the Account you wish to transfer to
: ")
    b_username = Account_Name

    Account_Database = open("Customer.dat", "rb")
    Temp_Database = open("Tempfund.dat", "wb")

    user1 = 0
    user2 = 0

    while True:
        try:
            a = pickle.load(Account_Database)
            if a.username == a_username:
                user1 = a
            elif a.username == b_username:
                user2 = a
            else:
                pickle.dump(a, Temp_Database)
        except:
            Account_Database.close()
            break

    if user2:
        while True:
            PasswordVerification = getpass("Enter
your Password for Verification
: ")
            if PasswordVerification ==
user1.password:
                Amount = input("Enter the Amount you
wish to transfer
: AED ")
                if Amount > user1.Balance:
                    print "INSUFFICIENT Account
Balance to Proceed"
                    break

                else:
                    print "From", user1.username
                    print "To", user2.username
                    print
                    user1.Balance -= Amount
                    user2.Balance += Amount
                    print "After Transaction"
                    print "Your Balance",
user1.Balance
                    print
                    print "Amount Successfully
Transferred! Thank You for using ABC Bank's Funds
Transfer Service!"

                    Transaction_ID =
randint(100001,999998)
                    print "Transaction ID
: ",Transaction_ID

                    Transfer_Time = ctime()
                    Transcation_Gist =
[len(user1.TransactionHistory), Transaction_ID,
Transfer_Time, user2.Acc_no, Account_Name, Amount]

user1.TransactionHistory.append(Transcation_Gist)
                    pickle.dump(user1, Temp_Database)
                    pickle.dump(user2, Temp_Database)

                    Temp_Database.close()
                    remove("Customer.dat")
                    rename("Tempfund.dat",
"Customer.dat")

                break
            else:
                try:
                    print "Incorrect Password"
                    print "Press Enter to Try Again!"
                    print "Press Ctrl+C to Exit"

                except KeyboardInterrupt:
                    break

while True:
    print "1. New Account"
    print "2. Existing Account"
    print "3. Utility"
    print "4. Admin"
    print "5. Exit"
    opt = input("Enter your choice : ")
    if opt == 1:
        New_Account()
    elif opt == 2:
        Existing_Account()
    elif opt == 3:
        Utility_Menu()
    elif opt == 4:
        bank = Bank()
        bank.Admin()
    elif opt == 5:
        print "Thank you for using ABC Bank Services"
        break
    else:
        print "Wrong Input"
```

OUTPUT



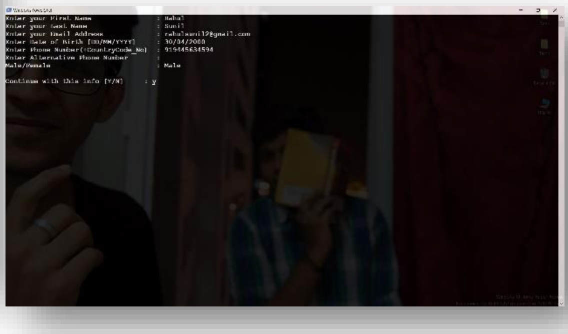
://SCREENS

The Main Page

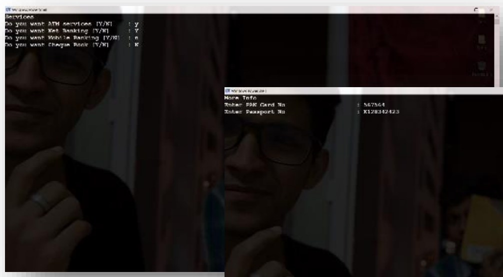


```
1. New Account
2. Existing Account
3. Utility
4. Admin
5. Exit
Enter your choice :
```

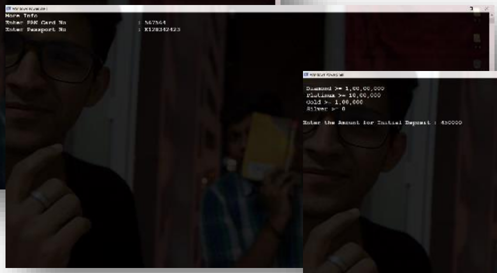
Entry of Customer Details



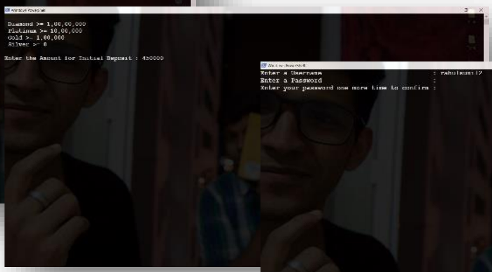
```
Enter your First Name                     : Rahul
Enter your Last Name                      : Sunil
Enter your Email Address                  : rahulsunil2@gmail.com
Enter Date of Birth [DD/MM/YYYY]          : 30/04/2000
Enter Phone Number(+CountryCode_No)       : 919445634594
Enter Alternative Phone Number            :
Male/Female                               : Male

Continue with this info [Y/N]        : y
```



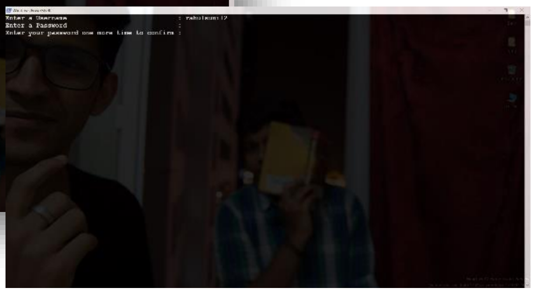
```
Services
Do you want ATM services [Y/N]     : y
Do you want Net Banking [Y/N]      : Y
Do you want Mobile Banking [Y/N]   : n
Do you want Cheque Book [Y/N]      : N
```



```
More Info
Enter PAN Card No                         : 567564
Enter Passport No                         : K128342423
```

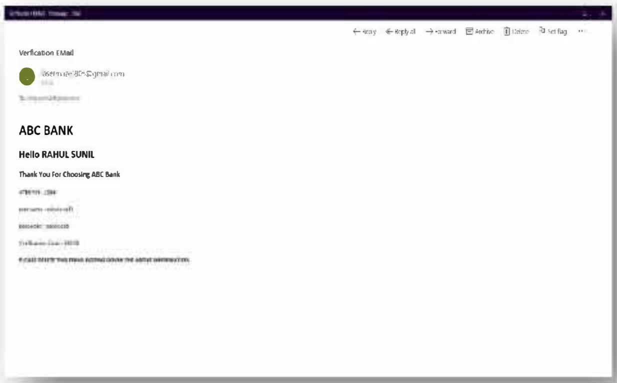


```
 Diamond >= 1,00,00,000
 Platinum >= 10,00,000
 Gold >= 1,00,000
 Silver >= 0

Enter the Amount for Initial Deposit : 450000
```



```
Enter a Username                          : rahulsunil2
Enter a Password                          :
Enter your password one more time to confirm :
```

Email and Mobile verification procedures



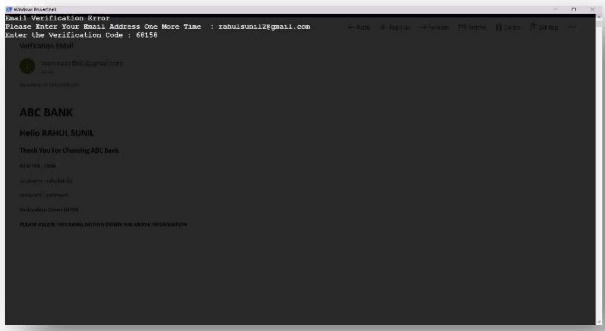
Verfication EMail

ABC BANK

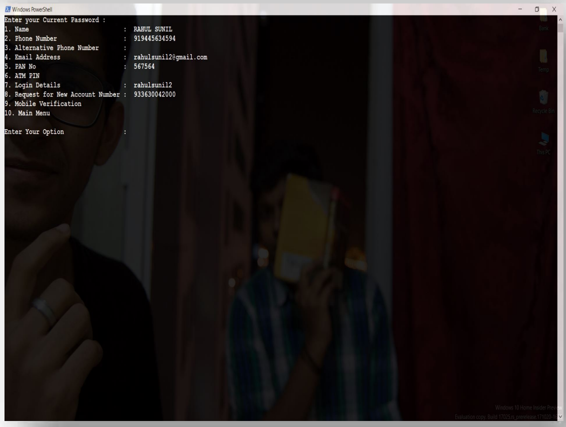
Hello RAHUL SUNIL

Thank You For Choosing ABC Bank

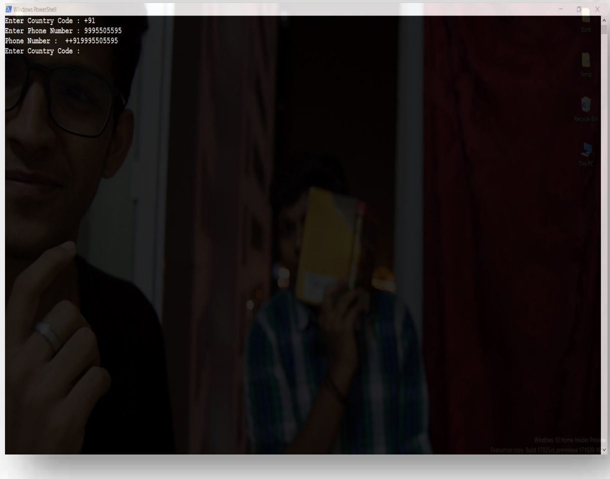
[illegible]



```
Email Verification Error
Please Enter Your Email Address One More Time  : rahulsunil2@gmail.com
Enter the Verification Code : 68158
```

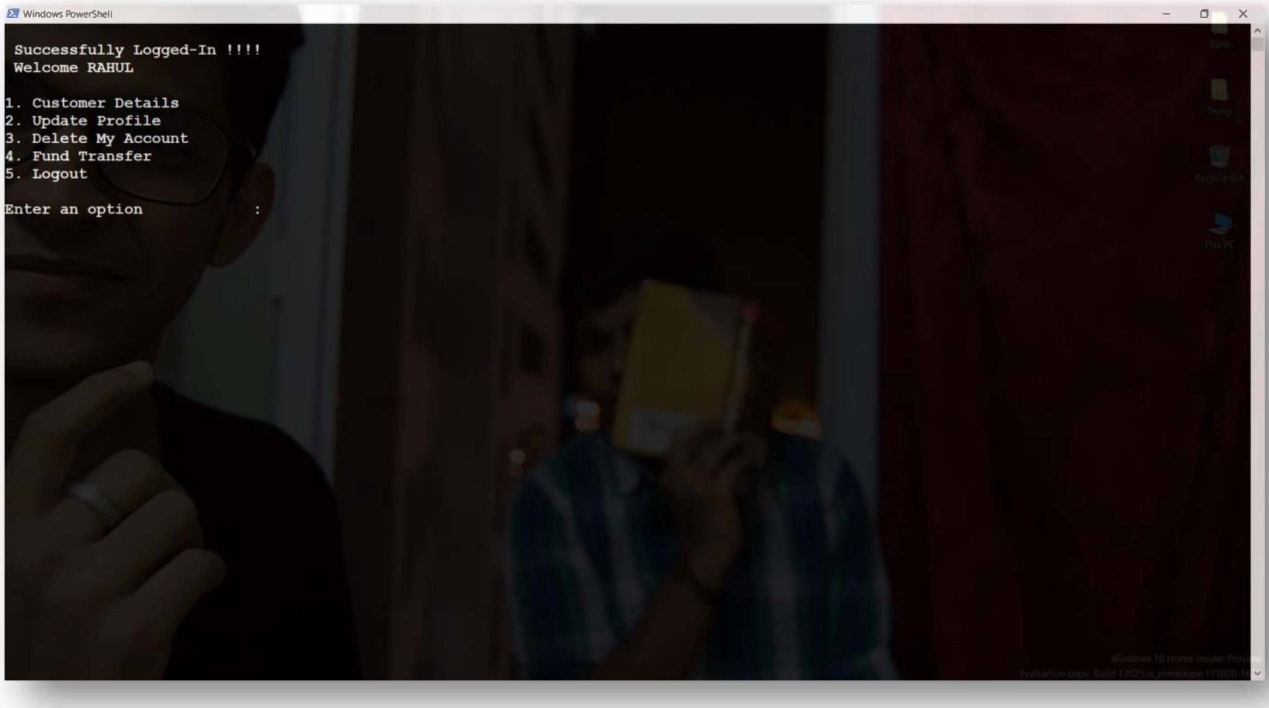


```
Enter your Current Password :
1. Name                           :  RAHUL SUNIL
2. Phone Number                   :  919445634594
3. Alternative Phone Number       :
4. Email Address                  :  rahulsunil2@gmail.com
5. PAN No                         :  567564
6. ATM PIN
7. Login Details                  :  rahulsunil2
8. Request for New Account Number :  933630042000
9. Mobile Verification
10. Main Menu

Enter Your Option                 :
```



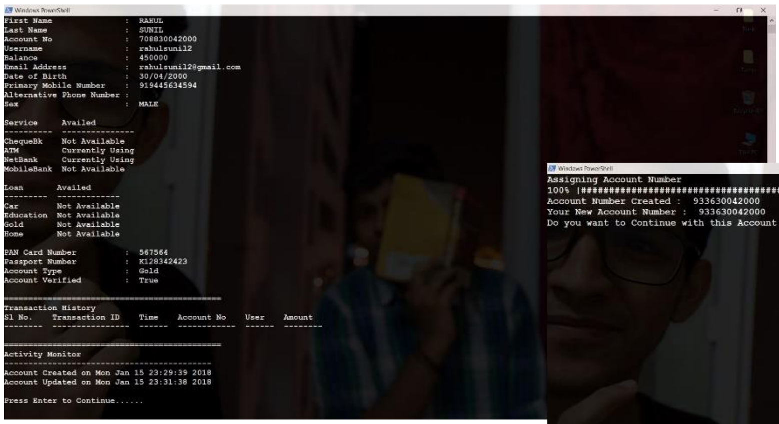
```
Enter Country Code : +91
Enter Phone Number : 9995505595
Phone Number :  ++919995505595
Enter Country Code :
```

Main Menu

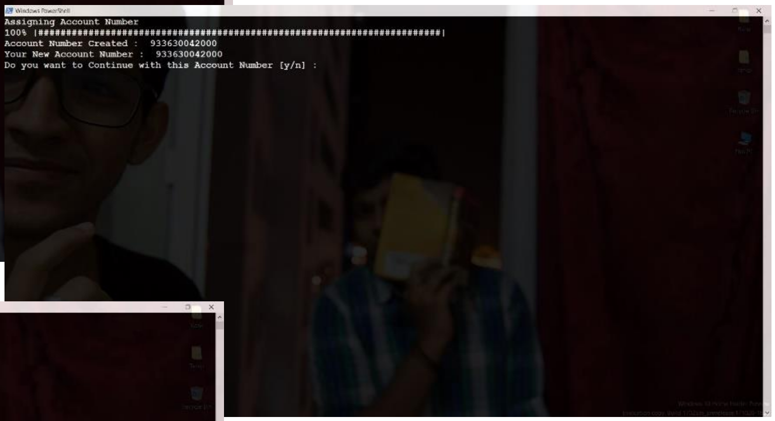


```
 Successfully Logged-In !!!!
 Welcome RAHUL

1. Customer Details
2. Update Profile
3. Delete My Account
4. Fund Transfer
5. Logout

Enter an option                  :
```

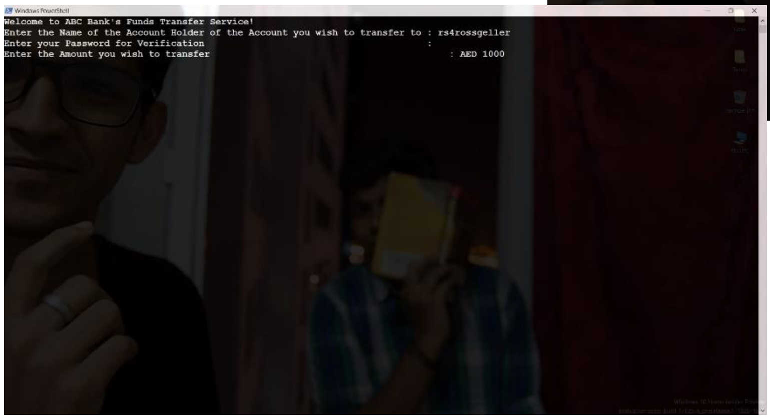
Acccount Options



```
First Name                  :  RAHUL
Last Name                   :  SUNIL
Account No                  :  708830042000
Username                    :  rahulsunil2
Balance                     :  450000
Email Address               :  rahulsunil2@gmail.com
Date of Birth               :  30/04/2000
Primary Mobile Number       :  919445634594
Alternative Phone Number :
Sex                         :  MALE

Service     Availed
----------  ---------------
ChequeBk    Not Available
ATM         Currently Using
NetBank     Currently Using
MobileBank  Not Available

Loan       Availed
---------  ------------
Car        Not Available
Education  Not Available
Gold       Not Available
Home       Not Available

PAN Card Number             :  567564
Passport Number             :  K128342423
Account Type                :  Gold
Account Verified            :  True

=============================================
Transaction History
Sl No.   Transaction ID    Time    Account No   User    Amount
-------  ---------------  ------  -----------  ------  --------

=============================================
Activity Monitor
-------------------------------------------
Account Created on Mon Jan 15 23:29:39 2018
Account Updated on Mon Jan 15 23:31:38 2018

Press Enter to Continue......
```



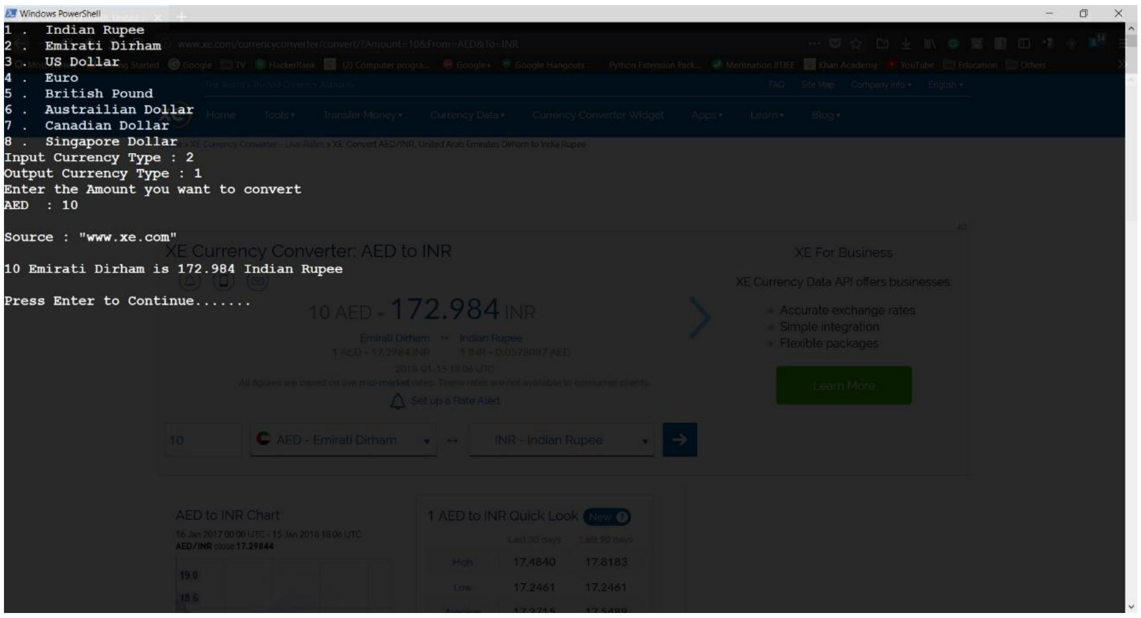
```
Assigning Account Number
100% |###############################################################################|
Account Number Created :  933630042000
Your New Account Number :  933630042000
Do you want to Continue with this Account Number [y/n] :
```



```
Welcome to ABC Bank's Funds Transfer Service!
Enter the Name of the Account Holder of the Account you wish to transfer to : rs4rossgeller
Enter your Password for Verification                                         :
Enter the Amount you wish to transfer                                          : AED 1000
```

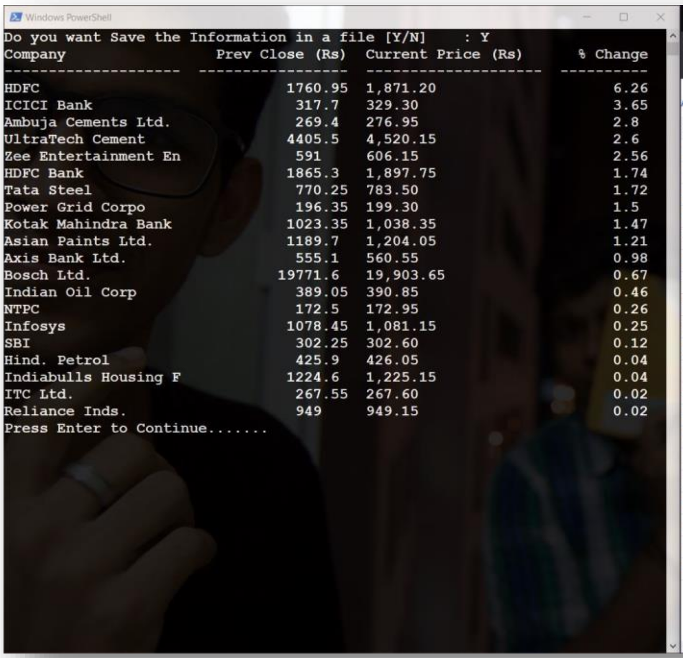
Utility

Currency Convertor

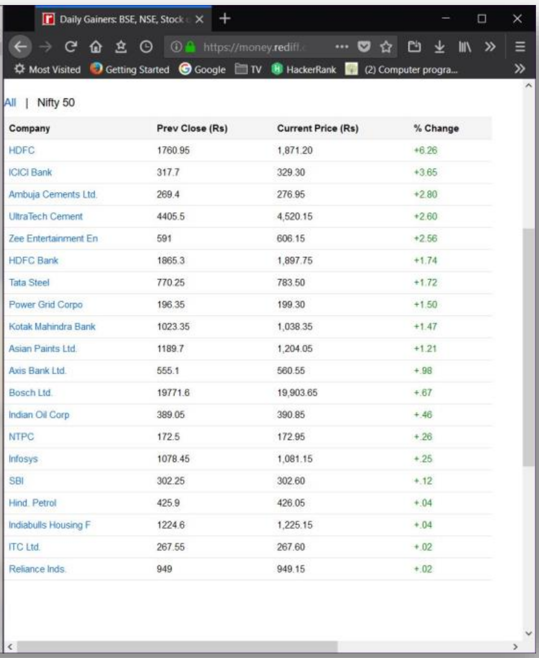


```
1 .  Indian Rupee
2 .  Emirati Dirham
3 .  US Dollar
4 .  Euro
5 .  British Pound
6 .  Austrailian Dollar
7 .  Canadian Dollar
8 .  Singapore Dollar
Input Currency Type : 2
Output Currency Type : 1
Enter the Amount you want to convert
AED  : 10

Source : "www.xe.com"

10 Emirati Dirham is 172.984 Indian Rupee

Press Enter to Continue.......
```

Stock Market Exchange



```
Do you want Save the Information in a file [Y/N]    : Y
Company                        Prev Close (Rs)   Current Price (Rs)      % Change
--------------------   -----------------   -------------------   ----------
HDFC                             1760.95   1,871.20                     6.26
ICICI Bank                         317.7   329.30                       3.65
Ambuja Cements Ltd.                269.4   276.95                       2.8
UltraTech Cement                  4405.5   4,520.15                     2.6
Zee Entertainment En                 591   606.15                       2.56
HDFC Bank                         1865.3   1,897.75                     1.74
Tata Steel                        770.25   783.50                       1.72
Power Grid Corpo                  196.35   199.30                       1.5
Kotak Mahindra Bank              1023.35   1,038.35                     1.47
Asian Paints Ltd.                 1189.7   1,204.05                     1.21
Axis Bank Ltd.                     555.1   560.55                       0.98
Bosch Ltd.                       19771.6   19,903.65                    0.67
Indian Oil Corp                   389.05   390.85                       0.46
NTPC                               172.5   172.95                       0.26
Infosys                          1078.45   1,081.15                     0.25
SBI                               302.25   302.60                       0.12
Hind. Petrol                       425.9   426.05                       0.04
Indiabulls Housing F              1224.6   1,225.15                     0.04
ITC Ltd.                          267.55   267.60                       0.02
Reliance Inds.                       949   949.15                       0.02
Press Enter to Continue.......
```

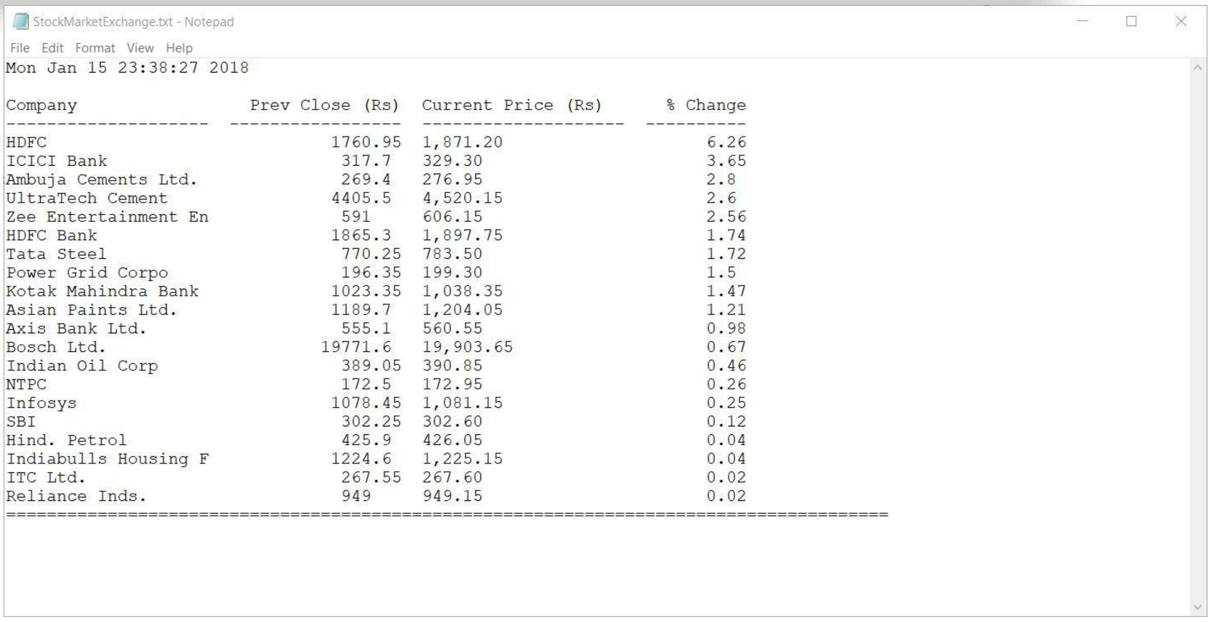


Daily Gainers: BSE, NSE, Stock

https://money.rediff.

All | Nifty 50

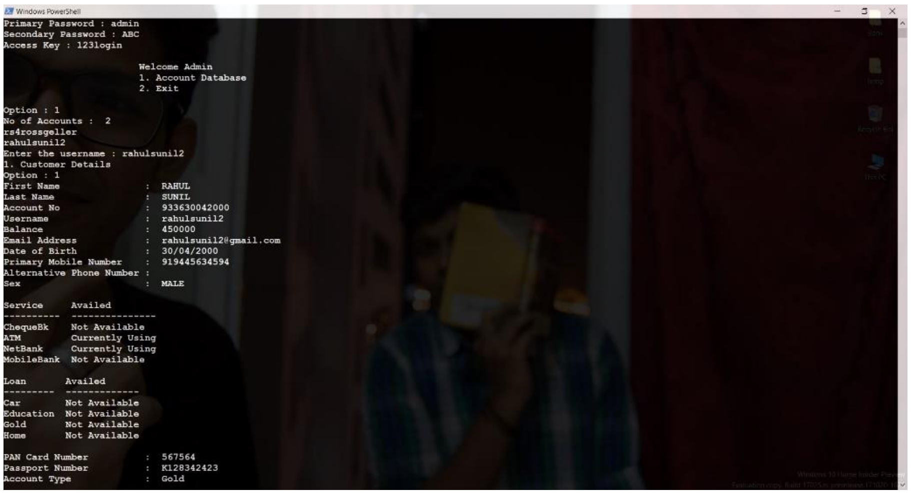
Company	Prev Close (Rs)	Current Price (Rs)	% Change
HDFC	1760.95	1,871.20	+6.26
ICICI Bank	317.7	329.30	+3.65
Ambuja Cements Ltd.	269.4	276.95	+2.80
UltraTech Cement	4405.5	4,520.15	+2.60
Zee Entertainment En	591	606.15	+2.56
HDFC Bank	1865.3	1,897.75	+1.74
Tata Steel	770.25	783.50	+1.72
Power Grid Corpo	196.35	199.30	+1.50
Kotak Mahindra Bank	1023.35	1,038.35	+1.47
Asian Paints Ltd.	1189.7	1,204.05	+1.21
Axis Bank Ltd.	555.1	560.55	+.98
Bosch Ltd.	19771.6	19,903.65	+.67
Indian Oil Corp	389.05	390.85	+.46
NTPC	172.5	172.95	+.26
Infosys	1078.45	1,081.15	+.25
SBI	302.25	302.60	+.12
Hind. Petrol	425.9	426.05	+.04
Indiabulls Housing F	1224.6	1,225.15	+.04
ITC Ltd.	267.55	267.60	+.02
Reliance Inds.	949	949.15	+.02



StockMarketExchange.txt - Notepad

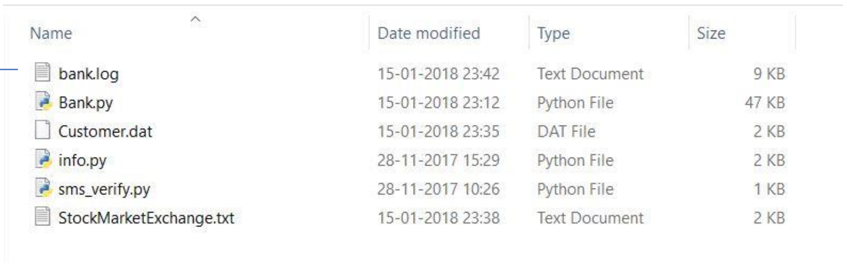
```
Mon Jan 15 23:38:27 2018

Company                  Prev Close (Rs)   Current Price (Rs)     % Change
--------------------   -----------------   -------------------   ----------
HDFC                             1760.95   1,871.20                     6.26
ICICI Bank                         317.7   329.30                       3.65
Ambuja Cements Ltd.                269.4   276.95                       2.8
UltraTech Cement                  4405.5   4,520.15                     2.6
Zee Entertainment En                 591   606.15                       2.56
HDFC Bank                         1865.3   1,897.75                     1.74
Tata Steel                        770.25   783.50                       1.72
Power Grid Corpo                  196.35   199.30                       1.5
Kotak Mahindra Bank              1023.35   1,038.35                     1.47
Asian Paints Ltd.                 1189.7   1,204.05                     1.21
Axis Bank Ltd.                     555.1   560.55                       0.98
Bosch Ltd.                       19771.6   19,903.65                    0.67
Indian Oil Corp                   389.05   390.85                       0.46
NTPC                               172.5   172.95                       0.26
Infosys                          1078.45   1,081.15                     0.25
SBI                               302.25   302.60                       0.12
Hind. Petrol                       425.9   426.05                       0.04
Indiabulls Housing F              1224.6   1,225.15                     0.04
ITC Ltd.                          267.55   267.60                       0.02
Reliance Inds.                       949   949.15                       0.02
=====================================================================================
```

ADMIN

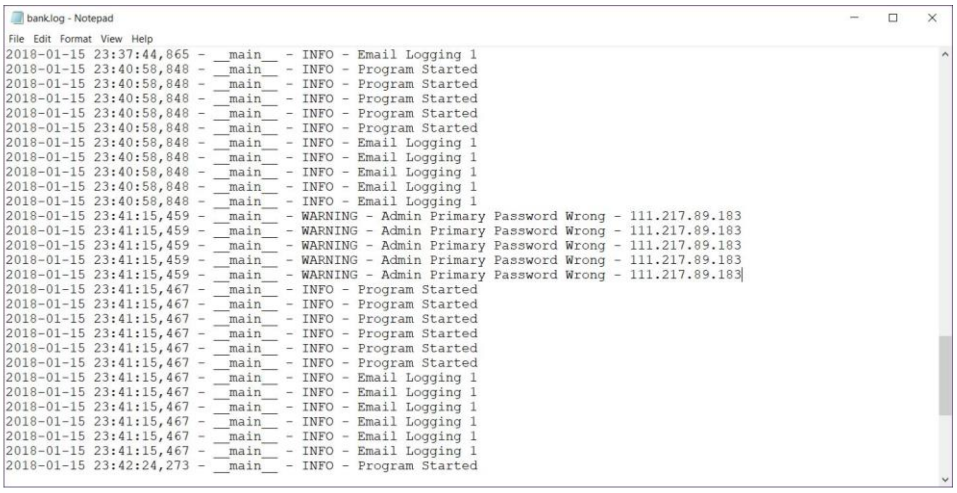


```
Primary Password : admin
Secondary Password : ABC
Access Key : 123login

                        Welcome Admin
                        1. Account Database
                        2. Exit

Option : 1
No of Accounts :  2
rs4rossgeller
rahulsunil2
Enter the username : rahulsunil2
1. Customer Details
Option : 1
First Name                 :  RAHUL
Last Name                  :  SUNIL
Account No                 :  933630042000
Username                   :  rahulsunil2
Balance                    :  450000
Email Address              :  rahulsunil2@gmail.com
Date of Birth              :  30/04/2000
Primary Mobile Number      :  919445634594
Alternative Phone Number :
Sex                        :  MALE

Service     Availed
----------  ---------------
ChequeBk    Not Available
ATM         Currently Using
NetBank     Currently Using
MobileBank  Not Available

Loan       Availed
---------  -------------
Car        Not Available
Education  Not Available
Gold       Not Available
Home       Not Available

PAN Card Number            :  567564
Passport Number            :  K128342423
Account Type               :  Gold
```

Back End



Name	Date modified	Type	Size
bank.log	15-01-2018 23:42	Text Document	9 KB
Bank.py	15-01-2018 23:12	Python File	47 KB
Customer.dat	15-01-2018 23:35	DAT File	2 KB
info.py	28-11-2017 15:29	Python File	2 KB
sms_verify.py	28-11-2017 10:26	Python File	1 KB
StockMarketExchange.txt	15-01-2018 23:38	Text Document	2 KB

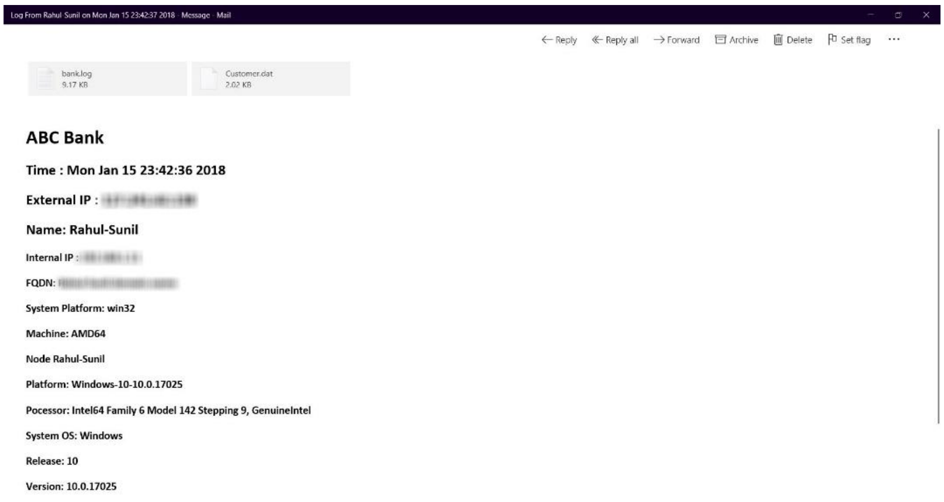


```
2018-01-15 23:37:44,865 - __main__ - INFO - Email Logging 1
2018-01-15 23:40:58,848 - __main__ - INFO - Program Started
2018-01-15 23:40:58,848 - __main__ - INFO - Program Started
2018-01-15 23:40:58,848 - __main__ - INFO - Program Started
2018-01-15 23:40:58,848 - __main__ - INFO - Program Started
2018-01-15 23:40:58,848 - __main__ - INFO - Program Started
2018-01-15 23:40:58,848 - __main__ - INFO - Email Logging 1
2018-01-15 23:40:58,848 - __main__ - INFO - Email Logging 1
2018-01-15 23:40:58,848 - __main__ - INFO - Email Logging 1
2018-01-15 23:40:58,848 - __main__ - INFO - Email Logging 1
2018-01-15 23:40:58,848 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,459 - __main__ - WARNING - Admin Primary Password Wrong - 111.217.89.183
2018-01-15 23:41:15,459 - __main__ - WARNING - Admin Primary Password Wrong - 111.217.89.183
2018-01-15 23:41:15,459 - __main__ - WARNING - Admin Primary Password Wrong - 111.217.89.183
2018-01-15 23:41:15,459 - __main__ - WARNING - Admin Primary Password Wrong - 111.217.89.183
2018-01-15 23:41:15,459 - __main__ - WARNING - Admin Primary Password Wrong - 111.217.89.183
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Program Started
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:41:15,467 - __main__ - INFO - Email Logging 1
2018-01-15 23:42:24,273 - __main__ - INFO - Program Started
```

Log File : A file which contains errors and bugs encountered. This can be used for debugging the program

ReportABug Email : An email is sent to the admin for debugging.

Log file and the Customer Database file is attached with email.



ABC Bank

Time : Mon Jan 15 23:42:36 2018

External IP : [redacted]

Name: Rahul-Sunil

Internal IP : [redacted]

FQDN: [redacted]

System Platform: win32

Machine: AMD64

Node Rahul-Sunil

Platform: Windows-10-10.0.17025

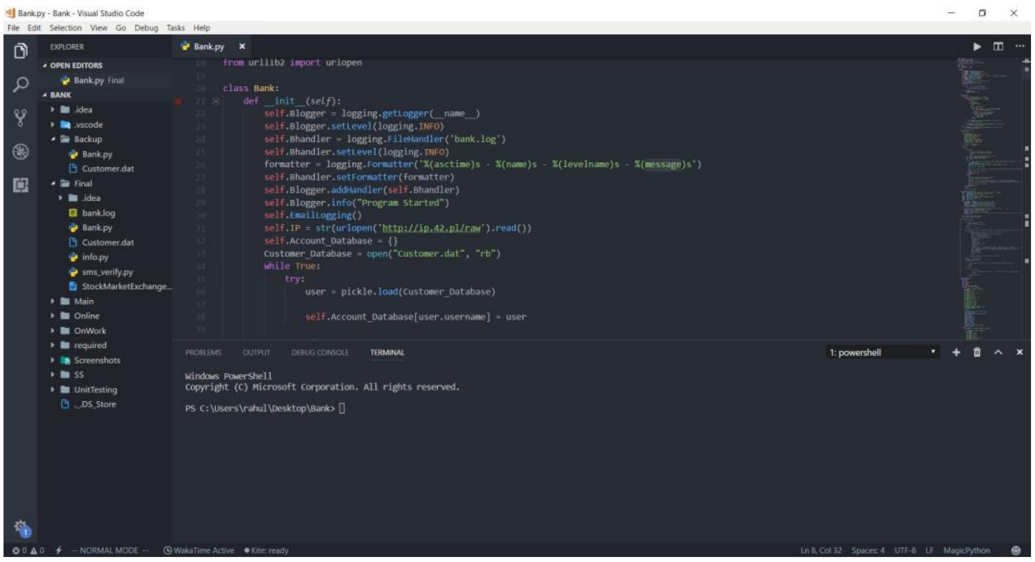
Pocessor: Intel64 Family 6 Model 142 Stepping 9, GenuineIntel

System OS: Windows

Release: 10

Version: 10.0.17025

Programming Environment



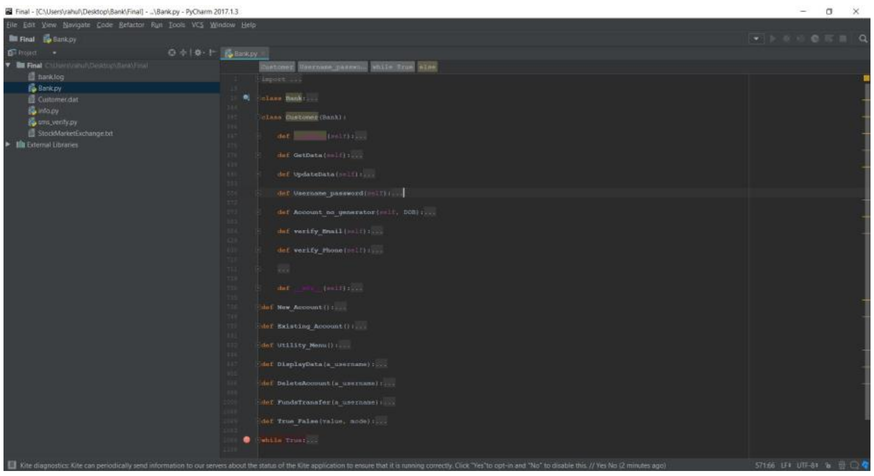
```
from urllib2 import urlopen

class Bank:
    def __init__(self):
        self.Blogger = logging.getLogger(__name__)
        self.Blogger.setLevel(logging.INFO)
        self.Bhandler = logging.FileHandler('bank.log')
        self.Bhandler.setLevel(logging.INFO)
        formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
        self.Bhandler.setFormatter(formatter)
        self.Blogger.addHandler(self.Bhandler)
        self.Blogger.info("Program Started")
        self.EmailLogging()
        self.IP = str(urlopen('http://ip.42.pl/raw').read())
        self.Account_Database = {}
        Customer_Database = open("Customer.dat", "rb")
        while True:
            try:
                user = pickle.load(Customer_Database)

                self.Account_Database[user.username] = user
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\rahul\Desktop\Bank> 
```

Visual

Studio

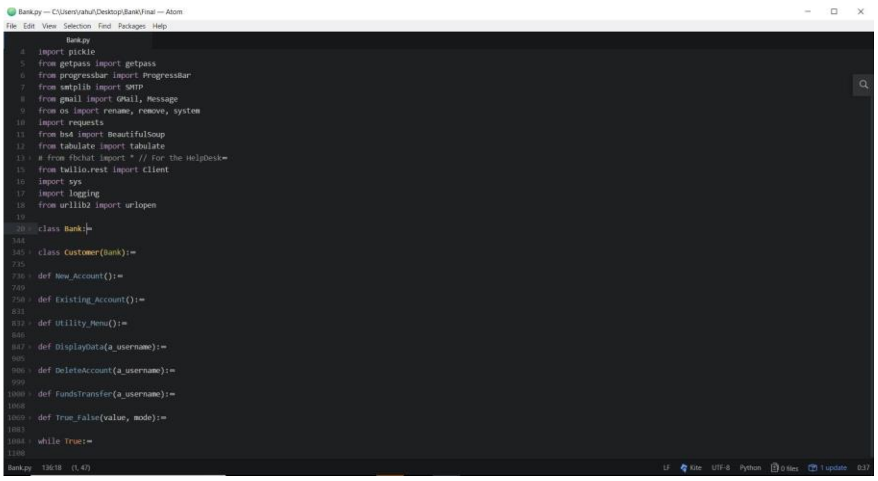
Code



```
import ...

class Bank:...

class Customer(Bank):

    def          (self):...

    def GetData(self):...

    def UpdateData(self):...

    def Username_password(self):...

    def Account_no_generator(self, DOB):...

    def verify_Email(self):...

    def verify_Phone(self):...

    ...

    def __del__(self):...

def New_Account():...

def Existing_Account():...

def Utility_Menu():...

def DisplayData(a_username):...

def DeleteAccount(a_username):...

def FundsTransfer(a_username):...

def True_False(value, mode):...

while True:...
```

JetBrians

Pycharm



```
import pickle
from getpass import getpass
from progressbar import ProgressBar
from smtplib import SMTP
from gmail import GMail, Message
from os import rename, remove, system
import requests
from bs4 import BeautifulSoup
from tabulate import tabulate
# from fbchat import * // For the HelpDesk
from twilio.rest import Client
import sys
import logging
from urllib2 import urlopen

class Bank:

class Customer(Bank):

def New_Account():

def Existing_Account():

def Utility_Menu():

def DisplayData(a_username):

def DeleteAccount(a_username):

def FundsTransfer(a_username):

def True_False(value, mode):

while True:
```

Atom

Bibliography

1. Python.org
2. Xe Currency Authority.

Language used : Python

Environments used : VS Code, Pycharm, Atom

```
>>> if os.page == " Last_Page " :
                              Break
    finally :
        print  " " "
```

THE END

" " "

```
>>> print  " " "
       Teacher :
       Mr. Baiju V " " "
```

[illegible]