MYSQL

BOOSTRAP

PYTHON

Amrit Subramanian C

XII B

CALIBER

FINANCES

Bonafide Certificate

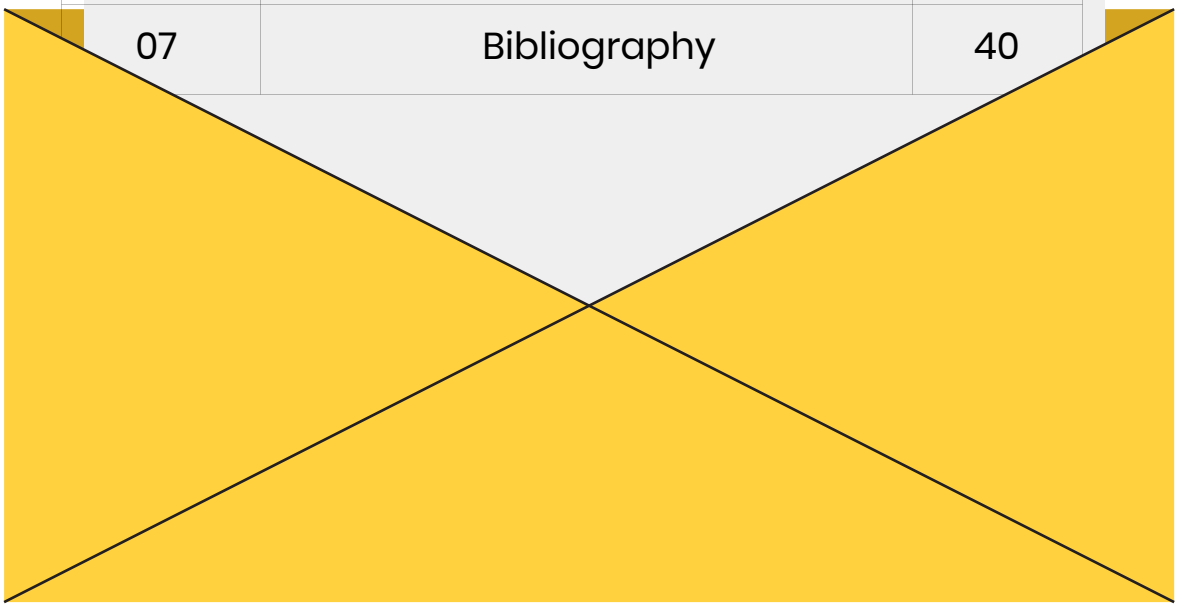
Certified that _____ of Standard _____ Studying in Amrita Vidyalayam, Chennai has Successfully Completed the project on _____ _____ _____during the academic year 2021-22 under my guidanve for AISSCE (All India Senior Secondary Certificate Examination) 2021-2022.

Date:

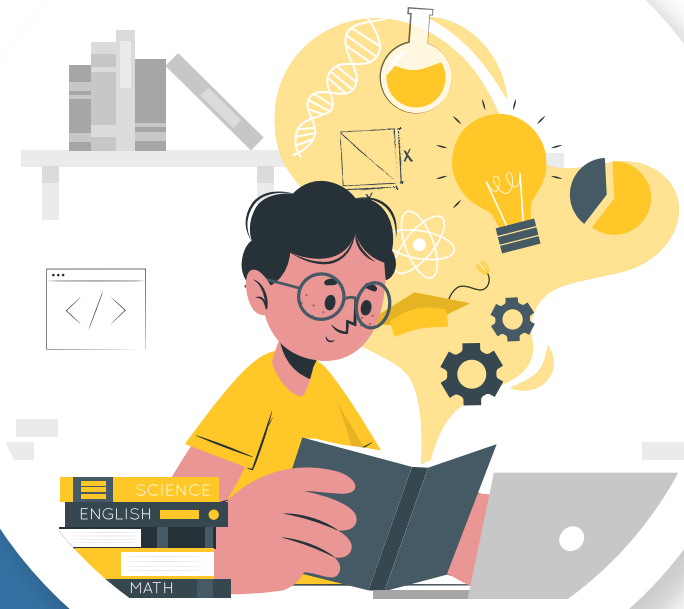
Teacher in charge. **Principal's Signature**

(Mrs. Sangeetha Karat) (Mrs Subashini Haridas)

Index:



ACKNOWLEDGEMENT



- We would like to take this opportunity to thank our Principal, Mrs. Subashini Haridas for her constant motivation support, even during tough times like this pandemic.

- We would also like to extend our heartfelt gratitude to our Computer Science Teacher, Mrs. Sangeetha Karat for guiding us and helping us complete our project successfully.

- A special thanks to our parents and our classmates, who helped us from time to time and encouraged us to think out of the box.

Fintech has been a buzzword in the world of finance and has the capability to extend financial inclusion, improve the daily lives of people, and spur growth.

COVID-19 has become an unexpected catalyst for tech adoption globally. A recent report* shows that we have vaulted five years forward in consumer and business digital adoption in a matter of around eight weeks. Online activities and transactions are no longer a matter of convenience but a necessity. Banks are launching digital channels so that customers can bank from home, and they can provide extra support to borrowers in distress. Senior citizens are adopting QR payments and digital banks for the first time. Schools are conducting classes through teleconferences. Grocery stores have shifted to online ordering and delivery. Doctors have begun delivering telemedicine. The list goes on.

As people adapt to a digital lifestyle, so too will they expect the same convenience and seamless experience from other areas of life, including financial services. Fintech has been a buzzword in the world of finance and has significantly shaped various areas, including banking, insurance, and investments. It also has a unique capability to extend financial inclusion, improve the daily lives of people, and spur growth. Anticipations are that the Fintech domain is to play a very crucial role in a post-COVID-19 world. And, for that to happen, the Fintech industry needs to evolve and adapt to this new scenario.

Technology Stack:

- **Backend**
  - PYTHON 3.9
  - Django Framework
- **Front End**
  - HTML5, CSS3, Bootsrap, JS and allied technologies.
- **Database**
  - MySQL 8

• External Python Libraries used

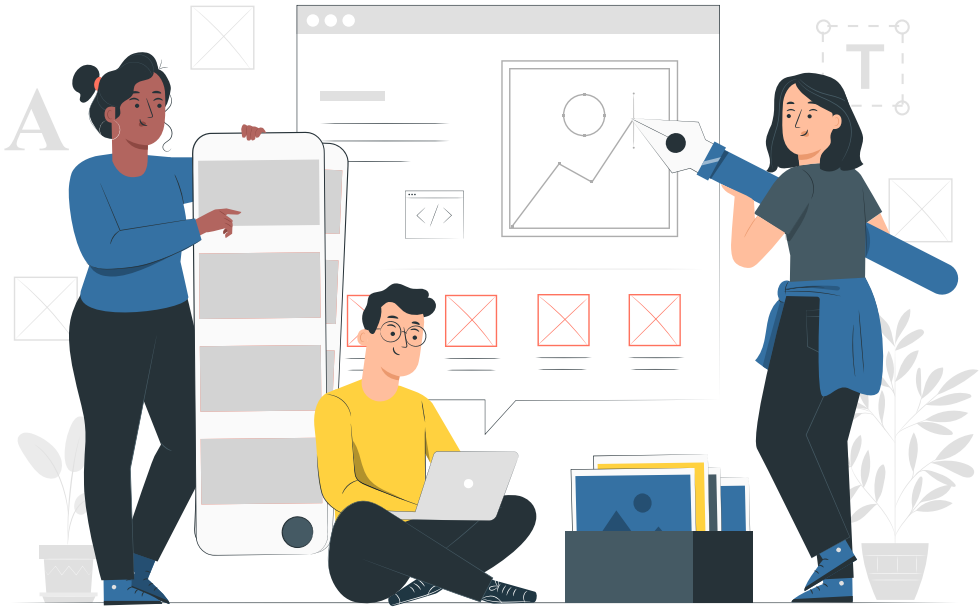
S.No	Library	Usage
1	smtplib	To send mail
2	Qrcode	To Generate QR codes.
3	opencv python	Used for all sorts of image and video analysis
4	csv	To read and write tabular data in CSV format
5	json	To store and exchange data
6	getpass	To read the input from the user as Password.
7	os	To Delete File
8	email	To read, write, and send simple email messages
9	sys	To manipulate different parts of the Python
10	datetime	To manipulate date and time
11	random	To generate a random number
12	cv2	Used for Computer Visioning
13	matplotlib	Used for data visualization and graphical plotting
14	mysql connector	Python driver for communicating with MySQL servers
15	imghdr	To determines the type of image contained in a file or byte stream

• HARDWARE Requirement

- **Processor :** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- **RAM :** 16gb
- **Storage :** 1Tb ssd
- **Operating System :** Windows 11 Home Single Language

• **IDE :** VSCODE

• **API :** UTILITY API



To ease the pieces of stuff, complex programs have been subdivided into tiny packages which contain imperative modules

Total Estimated Lines of Code ~ 8000
Total Packages Built : 8

PACKAGE 1 - cod_gen_bot_aisle

- Total Modules (2)
    - otp_gen_bot.py
    - unique_code_gen.py
- This library is being deployed to ease the chore of crafting unique id's for bank users without any clashes in private credentials. Viz.
    - OTP generation bot constructs a unique OTP and mails it to the given email address
    - Generates unique User ID/ Account Number
    - Creating multiple bank accounts without using identical email-id

PACKAGE 2 - db_managment

- Total Modules (3)
    - sv_int.py
    - man_vald.py
    - validative.py

• csv_int module have been created to deal with the registration of transactional history in a user's unique history- tabulated CSV file

• man_vald.py makes the code super-efficient by automating user authentication

•validative.py is a ground breaking snippet of code that becomes hyper-handy to manipulate database objects using the aspect of Oops[object-oriented-programming], thus resulting in the amplification of code-run pace

PACKAGE 3 - email_aisle

- Total Modules (1)
    - send_mail.py

• E-Mailing is a huge process... which is being smartly handled by this package. It works splendidly by providing plethora of choices to mail users depending upon the occasion.

PACKAGE 4 - input_aisle

- Total Modules (1)
    - inputs.py

• Acquiring info from users has been a tedious process... in a vision of simplifying it this package detects trashy outputs and raises exception to alert user and admin about the issue popped in.

PACKAGE 5 - json_aisle

- Total Modules (2)
    - son_manipulation.py
    - json_read_bot.py

• Passwords have been an absolute menace nowadays. Protecting them is a must necessary action to implement... thus this package helps in creating new hash algos for storing passcodes in a safe and non-human readable manner.

• Algos generated are tagged to its parent key using json files as its freaking fast at fetching data

PACKAGE 6 - qr_code_aisle

- Total Modules (2)
    - qr_gen_bot.py
    - qr_recognizer_bot.py

• Humans hate spoiling time in securing data.. that's where qrcode-authentication comes into play.

• qr_gen_bot modules helps in manufacturing of unique qr-codes for bank users

• qr_recognizer_bot translates the info encrypted in displayed qr-code to authenticate user

PACKAGE 7 - user_codes

- Total Modules (1)
    - sample_outs.py
    - json_read_bot.py

• Getting frustrated with input seeking bots have been a massive issue... to sort this out sample_outs module provides spotless human readable input seeking snippets

PACKAGE 8 - usr_shrs

- Total Modules (1)
    - users.py

• users module helps in fetching and forecasting total worth of the firm and provides the appropriate price of equity shares.

Directory Structure

DIRECTORY 1 -
csv_history_aisle

• This is the destination where all transaction-al-history related csv files gets stored

DIRECTORY 2 -
json_stuff

• This is the destination where the encrypting algo for password hash is stored

DIRECTORY 3 -
usr_qr_codes

• This is the destination where user's encrypted qr-codes are stored for user validation and processes that sort

Employee Panel

• This web interface helps employees to fetch or identify users and their bank details in a flashy and convenient way
Used Django to integrate MySql with Boostrap (UI).

DATABASE STRUCTURE

DB NAME : MAINSTRUCTURE

- Total Tables (3)
    - user_main
    - transactions
    - bankrevenue

Table Structures :

- user_main

Field	Type	Null	Key	Default
name	varchar(120)	YES		NULL
gender	char(1)	YES		NULL
DOB	varchar(120)	YES		NULL
nationality	varchar(120)	YES		NULL
unique_id	char(10)	NO	PRI	NULL
phone_no	varchar(20)	YES		NULL
email	varchar(200)	YES		NULL
date_created	varchar(252)	YES		NULL
balance	float	YES		NULL
pincode	char(4)	YES		NULL
password_hash	varchar(2500)	YES		NULL

- transactions

Field	Type	Null	Key	Default
unique_id	varchar(100)	YES		NULL
span	varchar(120)	YES		NULL
amount	float	YES		NULL
cord	char(1)	YES		NULL

- bankrevenue

Field	Type	Null	Key	Default
cash	float	YES		NULL
span	varchar(100)	YES		NULL

- Mysql commands used :

```
Create database Mainstructure;
create table if not exists user_main(
name varchar(120),
gender char(1),
DOB varchar(120),
nationality varchar(120),
unique_id char(10),
phone_no varchar(20),
email varchar(200),
date_created varchar(252),
balance float,
pincode char(4),
password_hash varchar(2500),
primary key(unique_id));


create Table If Not Exists transactions(
        unique_id varchar(100),
        span varchar(120),
        amount float,
        cord char(1));


CREATE TABLE IF NOT EXISTS bankrevenue(
        cash float,
        span varchar(100));
```

```
#importin stuffs required
from datetime import datetime
from source_aisle.cod_gen_bot_aisle import otp_gen_bot as send_otp, unique_code_gen as
pecu_id
from source_aisle.db_managment import validative as db_mani, man_vald as man_user_vali-
dation
from source_aisle.email_aisle import send_mail as mailup
from source_aisle.input_aisle import inputs as enterup
from source_aisle.qr_code_aisle import qr_gen_bot as make_qr, qr_recognizer_bot as
read_qr
from source_aisle.user_codes import sample_outs as outputs
from source_aisle.json_aisle import json_read_bot as hashpas
from source_aisle.db_managment import csv_int as csvmaker
from getpass import getpass
from source_aisle.usr_shrs import users
from os import remove
#---------------------------------------->
#mail csv
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from email.mime.text import MIMEText
import smtplib
import yagmail

def acc_post_email(reciever_email: str,
                   usdid: str):
    right_from = 'amritsubramanian.c@gmail.com'
    act_receiver = reciever_email
    titly = outputs.Email_head_Title
    filename = F'csv_history_aisle\\{usdid}.csv'
    mrit = yagmail.SMTP(right_from, 'amma@@1953')
    mrit.send(
        to=act_receiver,
        subject=titly,
        contents='Your transactional history upto now has been written to the csv file
pinged down:',
        attachments=filename
```

```
     )
     print('History has been mailed successfully!!')
#---------------------------------------->
def withdraw_funds(amount, id_m):

    withdraw_db = db_mani.MainDb('MainStructure')
    det = list(filter(lambda x: True if id_m[4] in x else False,withdraw_db.display_-
dat('user_main')))[0]
    # print(id_m)
    # for i in withdraw_db.display_dat('user_main'):
    #     if 'o!0$491bk$' in i: print(i)

    if amount<det[8] and amount<=100001 and det:
        min_balance = det[8]-amount-1
        #updating db
        withdraw_db.upd_data('user_main', 'balance', min_balance, 'n', det[4])
        withdraw_db.insert_dat(1,str(datetime.now()), tablename='bankrevenue')
        withdraw_db.insert_dat(det[4], str(datetime.now()), amount, 'd', table-
name='transactions')
          # print(outputs.user_debition.format(det[0], det[4], amount, 1, min_balance,
str(datetime.now())))

        csvmaker.csv_input('d', amount, str(datetime.now()), min_balance, id_m[4])

        #sending email
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.suc_emailand-
res_debition+'\n'+outputs.user_debition.format(det[0], det[4], amount, 1, min_balance,
str(datetime.now())))

        #printing output
        print(outputs.suc_emailandres_debition+'\n'+outputs.user_debition.format(det[0],
det[4], amount, 1, min_balance, str(datetime.now())))
    else:
        print(outputs.unsuc_emailandres_debition)
        print('Insufficient balance/withdrawal limit exceeded')
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.unsuc_emailand-
res_debition)
```

```
    withdraw_db.close_db()
    print()
def deposit_funds(amount, id_m):
    amnt = amount
    print()
    deposit_db = db_mani.MainDb('MainStructure')
    det = list(filter(lambda x: True if id_m[4] in x else False, deposit_db.display_-
dat('user_main')))[0]

    if 1<amnt and det:
        min_balance = det[8]+amnt-1
        #updating Balance
        deposit_db.upd_data('user_main', 'balance', min_balance, 'n', det[4])
        deposit_db.insert_dat(1,str(datetime.now()), tablename='bankrevenue')
        deposit_db.insert_dat(det[4], str(datetime.now()), amnt-1, 'c',tablename='trans-
actions')



        csvmaker.csv_input('c', amount, str(datetime.now()), min_balance, id_m[4])

        #sending email
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.suc_emailandres_-
credition+'\n'+outputs.User_credition.format(det[0], det[4], amnt, 1, min_balance, str(-
datetime.now())))
        #printing output
        print( outputs.suc_emailandres_credition+'\n'+outputs.User_credi-
tion.format(det[0], det[4], amnt, 1, min_balance, str(datetime.now())))

    else:
        print(outputs.unsuc_emailandres_credition)
        print('Insufficient balance/withdrawal limit exceeded')
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.unsuc_emailand-
res_credition)
    deposit_db.close_db()
    print()

def transfer_funds(senders_id, receivers_id, amount):
```

```
    #1$ should be deducted from the funds tranfered totaly from sender to recivers
100[debited]->99[credited]
    transferal = db_mani.MainDb('Mainstructure')
    amnt = amount
    print()
    det = list(filter(lambda x: True if senders_id[4] in x else False, transferal.dis-
play_dat('user_main')))[0]
    rec = list(filter(lambda x: True if receivers_id in x else False, transferal.dis-
play_dat('user_main')))[0]

    if amnt<=det[8] and det and receivers_id in rec:
        #reducing
        min_balance = det[8]-amnt
        #updating Balance
        transferal.upd_data('user_main', 'balance', min_balance, 'n', det[4])
        transferal.insert_dat(1,str(datetime.now()), tablename='bankrevenue')
        #maximising
        max_balance = rec[8]+amnt-1
        #updating Balance
        transferal.upd_data('user_main', 'balance', max_balance, 'n', rec[4])
        transferal.insert_dat(det[4], str(datetime.now()), amnt, 'd',tablename='transac-
tions')
        transferal.insert_dat(rec[4], str(datetime.now()), amnt-1, 'c',tablename='trans-
actions')

        #printing output
        print(outputs.suc_emailandres_transcation)
        print(outputs.User_Transaction_info.format(det[0], det[6], det[4],str(date-
time.now()),rec[0], rec[6], rec[4], amnt, min_balance))

        csvmaker.csv_input('c', amnt, str(datetime.now()), max_balance, receivers_id)

        csvmaker.csv_input('d', amnt, str(datetime.now()), min_balance, senders_id[4])

        #sending email sender
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.suc_emailand-
res_transcation+'\n'+outputs.User_Transaction_info.format(det[0], det[6], det[4],str(da-
```

```
res_transcation+'\n'+outputs.User_Transaction_info.format(det[0], det[6], det[4],str(da-
tetime.now()),rec[0], rec[6], rec[4], amnt, min_balance))

        #sending email receiver
        mailup.send_for_mail(rec[6], outputs.Email_head_Title, outputs.suc_emailand-
res_transcation+'\n'+outputs.User_Transaction_info.format(det[0], det[6], det[4],str(da-
tetime.now()),rec[0], det[6], rec[4], amnt, max_balance))
        # print(outputs.suc_emailandres_transcation+'\n'+outputs.Us-
er_Transaction_info.format(det[0], det[6], det[4],str(datetime.now()),rec[0], rec[6],
rec[4], amnt, max_balance))

    else:
        print(outputs.unsuc_emailandres_transcation)
        print('Insufficient balance or no such user found')
        mailup.send_for_mail(det[6], outputs.Email_head_Title, outputs.unsuc_emailand-
res_transcation)
    print()
    transfera1.close_db()

#---------------------------------------->

print('CALIBER FINANCES - <Indian unicorn-Fin-tech> | WELCOMES YOU!!\n')
while True:
    print(outputs.menu_1)
    while True:
        try:
            main_key = input('(enter your choice)-> ')
            break
        except: print('\nTry again... Enter valid input\n')

    if main_key=='1':#signup

        #name validation
        #usr_name
        while True:
            try:
                usr_name = enterup.name_in(input('Enter your name: '))
```

```
    except: print('\nTry again with valid input\n')
    else:
        print()
        if usr_name[1]:
            usr_name = usr_name[0]
            break
        else: print(usr_name[0])
        print()
    finally: pass


#GENDER
#gender
while True:
    try:
        gender = enterup.gender_in(input('Enter your gender: '))
    except: print('\nTry again with valid input\n')
    else:
        print()
        if gender[1]:
            gender = gender[0]
            break
        else: print(gender[0])
        print()
    finally: pass


#D-O-B
#dob
while True:
    try:
        dob = enterup.dob_in(input('Enter your DOB(dd-mm-yyyy): '))
    except: print('\nTry again with valid input\n')
    else:
        brk_real = False
        print()
        if dob[1]:
```

```
                dob = dob[0]
                break
            else:
                if dob[0][0]=='t':
                    print(dob[0])
                    print('program terminating..')
                    brk_real = True
                    break
                else: print(dob[0])
            print()
        finally: pass

    if brk_real: break
    else: pass


    #Nationality
    #nationality
    while True:
        try:
            nationality = enterup.nationality_in(input('Enter your Nationali-
ty(IND/USA): '))
        except: print('\nTry again with valid input\n')
        else:
            print()
            if nationality[1]:
                nationality = nationality[0]
                break
            else:
                print(nationality[0])
            print()
        finally: pass


    #Phone_n0
    #phone_no
    while True:
```

```
            try:
                phone_no = enterup.phone_no_in(input('Enter your phone-number(10-11
digits): '))
            except: print('\nTry again with valid input\n')
            else:
                print()
                if phone_no[1]:
                    phone_no = phone_no[0]
                    break
                else:
                    print(phone_no[0])
                print()
            finally: pass



        #password
        #password
        while True:
            try:
                password_1 = enterup.pass_in(getpass(prompt='Set Password: '))
                password_2 = getpass(prompt='Confirm Password: ')
            except: print('\nTry again with valid input\n')
            else:
                print()
                if password_1[1]:
                    if password_1[0]==password_2:
                        password = hashpas.json_hash_val(password_1[0])
                        break
                    else: print("passwords didn't match! Try again")
                else:
                    print(password_1[0])
                print()
            finally: pass

        #pincode
        #pincode
        while True:
```

```
            try:
                pincode = enterup.pincode_in(getpass(prompt='Set pincode: '))
                pincode_2 = getpass(prompt='Confirm pincode: ')
            except: print('\nTry again with valid input\n')
            else:
                print()
                if pincode[1]:
                    if pincode_2==pincode[0]:
                        pincode = pincode[0]
                        break
                    else: print("pin's didn't match! Try again")
                else:
                    print(pincode[0])
                print()
            finally: pass


        #Email
        #email
        while True:
            brky = False
            #retrievin email from db
            email_collection = db_mani.MainDb('MainStructure')
            paras = email_collection.display_dat('user_main')
            #6
            paras = list(map(lambda x: x[6], paras))
            email_collection.close_db()
            try:
                email = enterup.email_in(input('Enter your E-Mail: '))
                if email[1]:
                    if email[0] in paras:
                        print('\nEmail has been sacked already...Try a new one\n')
                    else:
                        otp_generated = send_otp.gen_otp()
                        mailup.send_for_mail(email[0], outputs.otp_verification, F'your
otp-> {otp_generated}')
                        print()
                        read_otp = input('enter the otp mailed: ')
```

```
                    print()
                    if otp_generated==read_otp:
                        email = email[0]
                        break
                    else:
                        print("otp didn't match! Try again.")
                        resend = input('resend otp(y/n): ')
                        if resend == 'n':
                            brky = True
                            break
                        else: brky = False
                        print()
            else:  print('\nTry again with valid email\n')

        except:  print('\nTry again with valid input\n')
        else: pass
        finally: pass

    if brky:
        print('program terminating...')
        break

    unique_id = pecu_id.gen_random()

    created_span = str(datetime.now())

    balance = .0

    #qrcode_making
    maxy = make_qr.gen_qr_code(unique_id, F'usr_qr_codes\\{unique_id}.jpg')

    #writtin into db
    main_wrt = db_mani.MainDb('MainStructure')
    x = main_wrt.insert_dat(usr_name, gender, dob, nationality, unique_id, phone_no,
email, created_span, balance, pincode,password, tablename='user_main')
    main_wrt.close_db()
    csvmaker.initialize_user(unique_id)
```

```
          #writtin success
        if x:
            print(outputs.signup_pass)
            z = outputs.User_created_email_info.format(usr_name, unique_id, balance,
created_span, email, gender, phone_no, nationality, dob)
            print(z)
            mailup.send_for_mail(email, outputs.Email_head_Title, z, F'us-
r_qr_codes\\{unique_id}.jpg')

        else: print(outputs.signup_fail)
        print(outputs.splitter)

#------------------------------------------------------------------------------>>>
    elif main_key=='2': #Login qr_code
        main = read_qr.qr_reader()
        datas = db_mani.MainDb('MainStructure')
        try:
            main = list(filter(lambda x: True if main in x else False,datas.display_-
dat('user_main')))[0]
        except: print('user not found!!')
        else:
            if main:

#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++

                while True:
                    print('-------------------------------------------')
                    print(F'Logged in as {main[0]}--->')
                    print('-------------------------------------------')
                    print(outputs.menu_Login)

                    while True:
                        try:
                            decision = input('(enter your choice)-> ')
                            break
                        except: print('Try Again.. Enter valid input')
```

```
if decision=='1':
    iny = True
    try:
        pin = getpass('Enter 4 digit pin: ')
        if pin==main[9]: pass
        else: iny=False
    except:
        iny = False
        print('invalid input.. Try Again...')
    if iny:
        try:
            amounty = float(input('Enter amount to be withdrawn: '))
            withdraw_funds(amounty, main)#amnt, id
        except: print('invalid input.. Try Again...')
    else: print('pincode didn\'t match...')


elif decision=='2':
    iny = True
    try:
        pin = getpass('Enter 4 digit pin: ')
        if pin==main[9]: pass
        else: iny=False
    except:
        iny = False
        print('invalid input.. Try Again...')

    if iny:
        try:
            amounto = float(input('Enter amount to be deposited: '))
            deposit_funds(amounto, main)#amnt, id
        except: print('invalid input.. Try Again...')
    else: print('pincode didn\'t match...')


elif decision=='3':
    iny = True
```

```
                        try:
                            pin = getpass('Enter 4 digit pin: ')
                            if pin==main[9]: pass
                            else: iny=False
                        except:
                            iny = False
                            print('invalid input.. Try Again..')
                        if iny:
                            try:
                                amounti = float(input('Enter amount to be transferred:
'))
                                rec_id = (input('Enter receiver\'s id: '))
                                transfer_funds(main, rec_id, amounti)#amnt, id
                            except: print('invalid input... No such user found.. Try
Again...')
                        else: print('pincode didn\'t match...')

                    elif decision=='4': #change pin

                        try:
                            pin = getpass('Enter 4 digit pin: ')
                            if pin==main[9]:
                                pinnew = enterup.pincode_in(getpass('Enter your new pin:
'))
                                pinnewconfirm = getpass('Enter your new pin: ')
                                if pinnew and pinnewconfirm==pinnew[0]:
                                    datas.upd_data('user_main', 'pincode', pinnew[0],
's', main[4])
                                    print('pincode updated successfully...Login
again...')
                                    print(outputs.splitter)
                                    break
                                else:
                                    print('Process Failed... Invalid pin entered or
pincodes didn\'t match')
                        except:
                            print('invalid input.. Try Again..')
```

```
                elif decision=='5': #change password

                    try:
                        pin = getpass('Enter 4 digit pin: ')
                        if pin==main[9]:
                            passcode = getpass('Enter password: ')
                            if hashpas.json_hash_val(passcode) == main[10]:
                                enter1 = enterup.pass_in(getpass('Enter new pass-
word: '))
                                if enter1[1]:
                                    enter2 = getpass('Enter new password: ')
                                    if enter2==enter1[0]:
                                        datas.upd_data('user_main', 'password_hash',
hashpas.json_hash_val(enter2), 's', main[4])
                                        print('password updated successfully...Login
again...')
                                        print(outputs.splitter)
                                        break
                                    else: print('passwords didn\'t match.. Try
again')
                                else: print('invalid password.. Try again')

                            else:print('invalid password.. Try again')
                        else:print('invalid pin.. Try again')


                    except:
                        print('invalid input.. Try Again..')

                elif decision=='6': #email history
                    acc_post_email(main[6], main[4])

                elif decision=='7': #acnt info
                    balancy = list(filter(lambda x: True if x[4]==main[4] else
False,db_mani.MainDb('Mainstructure').display_dat('user_main')))
                    balancy = balancy[0][8]
```

```
                        print(F'''
------------------------------
User's Info                   |
------------------------------
User name: {main[0]}
User id: {main[4]}
User balance: ${balancy}
account created on: {main[7]}
User's email id: {main[6]}
------------------------------
        ''')

                    elif decision=='8':
                        try:
                            pin = getpass('Enter 4 digit pin: ')
                        except:
                            print('Invalid input. Try again!')
                        else:
                            if pin==main[9]:
                                print('pin matched')
                                dece = input('Do you want to delete this account(y/n):
').casefold()
                                if dece=='y':
                                    buby = db_mani.MainDb('MainStructure')
                                    buby.del_data('user_main', main[4])
                                    #-----------------------------------
                                    remove(F'usr_qr_codes\\{main[4]}.jpg')
                                    remove(F'csv_history_aisle\\{main[4]}.csv')
                                    #-----------------------------------
                                    print('Account deleted successfully!!')
                                    buby.close_db()

                                    break
                                else: print('deletion process was aborted')
                            else: print('pincode didn\'t match')
```

```
                elif decision=='9':
                    print('Logging out...')
                    break

                else: print('Invalid input... Try again with a valid one.')

                print(outputs.splitter)


#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++

                #usr pin = main[9]
            else:
                print('No such user found! Try again.')
            datas.close_db()
#------------------------------------------------------------------------------>>>
    elif main_key=='3':#manual login
        try:
            main = man_user_validation.man_authentication(input('Enter usr id: '), get-
pass(prompt='Enter password: '))
            print()
            datas = db_mani.MainDb('MainStructure')
            into = True
        except: into = False
        if into:
            if main:
                main = main[1][0]

#------------------------------------------------------------------------------>
                while True:
                    print('-----------------------------------------')
                    print(F'Logged in as {main[0]}--->')
                    print('-----------------------------------------')
                    print(outputs.menu_Login)
```

```
while True:
    try:
        decision = input('(enter your choice)-> ')
        break
    except: print('Try Again.. Enter valid input')

if decision=='1':
    iny = True
    try:
        pin = getpass('Enter 4 digit pin: ')
        if pin==main[9]: pass
        else: iny=False
    except:
        iny = False
        print('invalid input.. Try Again...')
    if iny:
        try:
            amounty = float(input('Enter amount to be withdrawn: '))
            withdraw_funds(amounty, main)#amnt, id
        except: print('invalid input.. Try Again...')
    else: print('pincode didn\'t match...')


elif decision=='2':
    iny = True
    try:
        pin = getpass('Enter 4 digit pin: ')
        if pin==main[9]: pass
        else: iny=False
    except:
        iny = False
        print('invalid input.. Try Again...')

    if iny:
        try:
            amounto = float(input('Enter amount to be deposited: '))
            deposit_funds(amounto, main)#amnt, id
```

```
                except: print('invalid input.. Try Again...')
            else: print('pincode didn\'t match...')


        elif decision=='3':
            iny = True
            try:
                pin = getpass('Enter 4 digit pin: ')
                if pin==main[9]: pass
                else: iny=False
            except:
                iny = False
                print('invalid input.. Try Again..')
            if iny:
                try:
                    amounti = float(input('Enter amount to be transferred:
'))
                    rec_id = (input('Enter receiver\'s id: '))
                    transfer_funds(main, rec_id, amounti)#amnt, id
                except: print('invalid input... No such user found.. Try
Again...')
            else: print('pincode didn\'t match...')

        elif decision=='4': #change pin

            try:
                pin = getpass('Enter 4 digit pin: ')
                if pin==main[9]:
                    pinnew = enterup.pincode_in(getpass('Enter your new pin:
'))
                    pinnewconfirm = getpass('Enter your new pin: ')
                    if pinnew and pinnewconfirm==pinnew[0]:
                        datas.upd_data('user_main', 'pincode', pinnew[0],
's', main[4])
                        print('pincode updated successfully...Login
again...')
                        print(outputs.splitter)
```

```
                                break
                            else:
                                print('Process Failed... Invalid pin entered or
pincodes didn\'t match')
                    except:
                        print('invalid input.. Try Again..')


                elif decision=='5': #change password

                    try:
                        pin = getpass('Enter 4 digit pin: ')
                        if pin==main[9]:
                            passcode = getpass('Enter password: ')
                            if hashpas.json_hash_val(passcode) == main[10]:
                                enter1 = enterup.pass_in(getpass('Enter new pass-
word: '))
                                if enter1[1]:
                                    enter2 = getpass('Enter new password: ')
                                    if enter2==enter1[0]:
                                        datas.upd_data('user_main', 'password_hash',
hashpas.json_hash_val(enter2), 's', main[4])
                                        print('password updated successfully...Login
again...')
                                        print(outputs.splitter)
                                        break
                                    else: print('passwords didn\'t match.. Try
again')
                                else: print('invalid password.. Try again')

                            else:print('invalid password.. Try again')
                        else:print('invalid pin.. Try again')


                    except:
                        print('invalid input.. Try Again..')
```

```
                    elif decision=='6': #email history
                        acc_post_email(main[6], main[4])

                    elif decision=='7': #acnt info
                        balancy = list(filter(lambda x: True if x[4]==main[4] else
False,db_mani.MainDb('Mainstructure').display_dat('user_main')))
                        balancy = balancy[0][8]
                        print(F'''
------------------------------
User's Info                   |
------------------------------
User name: {main[0]}
User id: {main[4]}
User balance: ${balancy}
account created on: {main[7]}
User's email id: {main[6]}
------------------------------
    ''')

                    elif decision=='8':
                        try:
                            pin = getpass('Enter 4 digit pin: ')
                        except:
                            print('Invalid input. Try again!')
                        else:
                            if pin==main[9]:
                                print('pin matched')
                                dece = input('Do you want to delete this account(y/n):
').casefold()
                                if dece=='y':
                                    buby = db_mani.MainDb('MainStructure')
                                    buby.del_data('user_main', main[4])
                                    #-----------------------------------
                                    remove(F'usr_qr_codes\\{main[4]}.jpg')
                                    remove(F'csv_history_aisle\\{main[4]}.csv')
                                    #-----------------------------------
                                    print('Account deleted successfully!!')
```

```
                                buby.close_db()

                                break
                            else: print('deletion process was aborted')
                        else: print('pincode didn\'t match')


                elif decision=='9':
                    print('Logging out...')
                    break

                else: print('Invalid input... Try again with a valid one.')

                print(outputs.splitter)

#------------------------------------------------------------------------------------>
                datas.close_db()

            else:
                print('No such user found/password didn\'t match! Try again.')
                print()
        else:
            print('Invalid input! Try again.')
            print()
        print(outputs.splitter)

#------------------------------------------------------------------------------------>>>

    elif main_key=='4':
        users.show_shares()
        print(outputs.splitter)

    elif main_key=='5':
        print('Terminating....')
        break

    else: pass
```

User Menu

```
CALIBER FINANCES - <Indian unicorn-Fin-tech> | WELCOMES YOU!!


1-> Signup
2-> Login - Qrcode
3-> Login - Passcode
4-> Shares Composition
5-> Terminate

(enter your choice)->
```

Signup

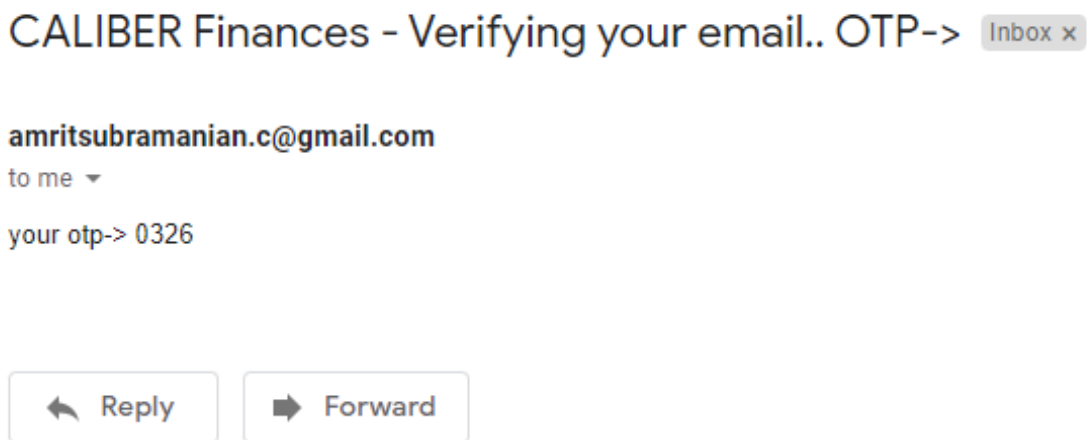
```
(enter your choice)-> 1
Enter your name: C.Amrit Subramananian

Enter your gender: M


Signup-Successful

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Account Created Successfully
------------------------------
User's Name: C.Amrit Subramananian
User's Id: k6%z1q!t!2
User's Balance: $0.0
Date Created: 2022-02-17 06:50:22.355179
User's Email: amritsubramanian.c@gmail.com
User's Gender: M
User's Phone Number: 9003724367
User's Nationality: USA
User's DOB: 11-08-1997
------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

OTP Mailed



EMAIL-Account Created

CALIBER FINANCES Inbox ×

amritsubramanian.c@gmail.com
to me

>>
Account Created Successfully

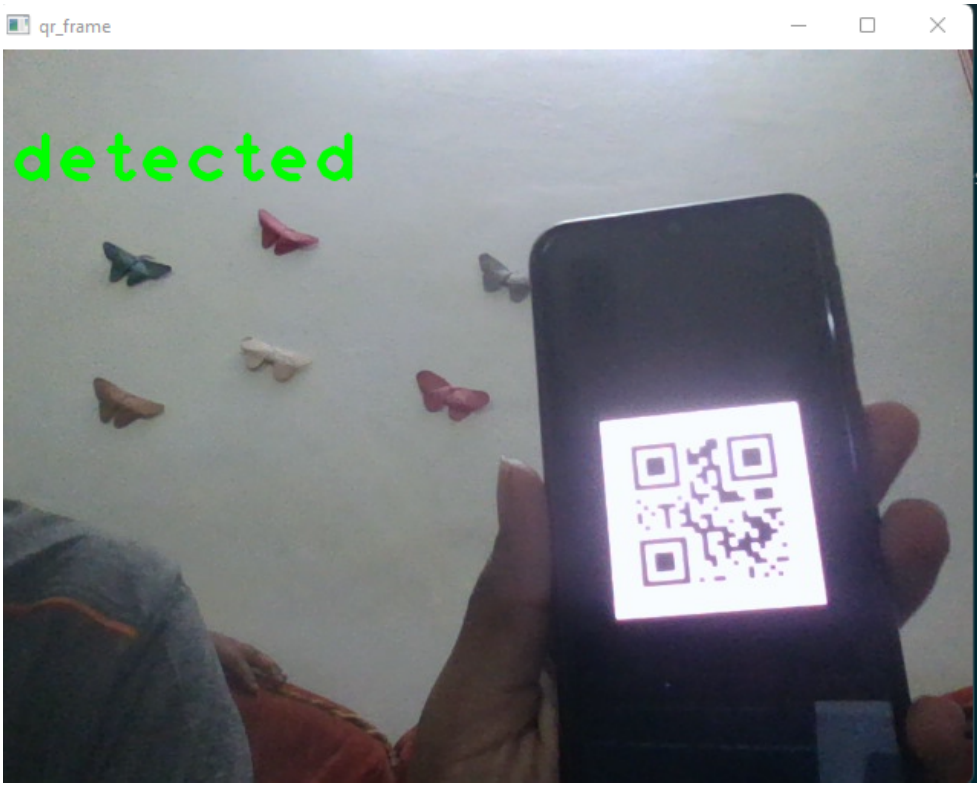
User's Name: C.Amrit Subramananian
User's Id: k6%z1q!t!2
User's Balance: $0.0
Date Created: 2022-02-17 06:50:22.355179
User's Email: amritsubramanian.c@gmail.com
User's Gender: M
User's Phone Number: 9003724367
User's Nationality: USA
User's DOB: 11-08-1997

>>

Login:

- Qrcode-login:
  - Before detection:

- After detection:



Manual Login

```
(enter your choice)-> 3
Enter usr id: a#29485G83
Enter password:
```

After - Login Menu

```
------------------------------------------------
Logged in as C.Amrit Subramananian--->
------------------------------------------------

1-> Withdraw Funds($100000 at max)
2-> Deposit Funds
3-> Transfer Funds
4-> Change Pin
5-> Change Password
6-> Transactional History(Get it Mailed in a CSV format)
7-> Account Info
8-> Delete Account
9-> Log out
------------------------------------------------------------
```

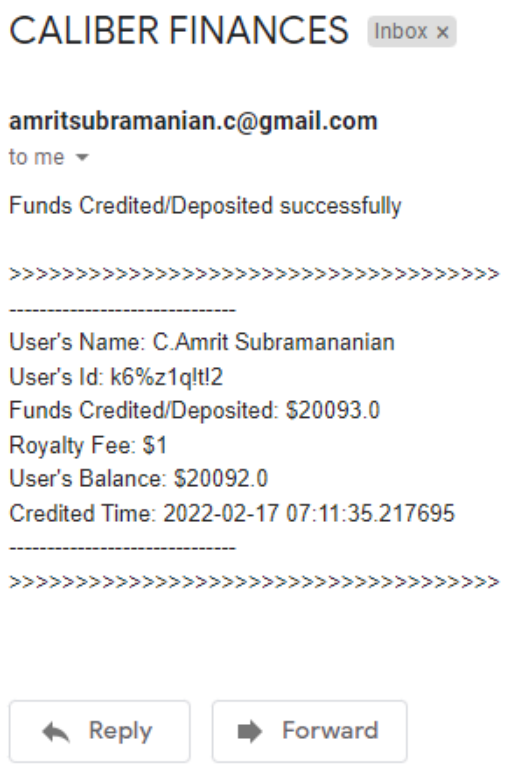
Fund Deposition

```
(enter your choice)-> 2
Enter 4 digit pin:
Enter amount to be deposited: 20093

Funds Credited/Deposited successfully

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
------------------------------
User's Name: C.Amrit Subramananian
User's Id: k6%z1q!t!2
Funds Credited/Deposited: $20093.0
Royalty Fee: $1
User's Balance: $20092.0
Credited Time: 2022-02-17 07:11:41.419079
------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Deposition Mail



Funds Withdrawal

```
(enter your choice)-> 1
Enter 4 digit pin:
Enter amount to be withdrawn: 1200
Funds Debited/Withdrawn successfully

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
------------------------------
User's Name: C.Amrit Subramananian
User's Id: k6%z1q!t!2
Funds Debited/Withdrawn: $1200.0
Royalty Fee: $1
User's Balance: $18891.0
Debited Time: 2022-02-17 07:14:18.130306
------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Withdrawal Email

CALIBER FINANCES Inbox ×

amritsubramanian.c@gmail.com
to me

Funds Debited/Withdrawn successfully

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
------------------------------
User's Name: C.Amrit Subramananian
User's Id: k6%z1q!t!2
Funds Debited/Withdrawn: $1200.0
Royalty Fee: $1
User's Balance: $18891.0
Debited Time: 2022-02-17 07:14:13.552972
------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

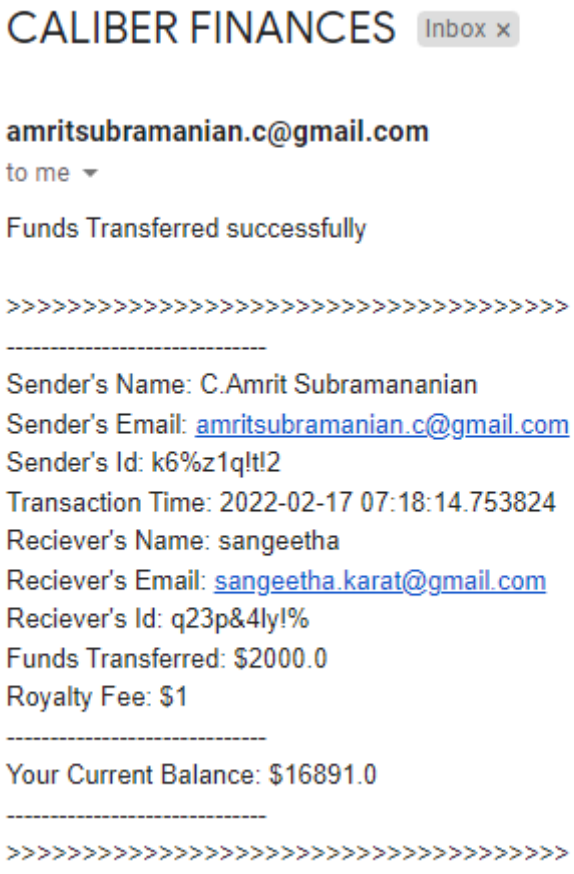
Fund Transfer

```
(enter your choice)-> 3
Enter 4 digit pin:
Enter amount to be transferred: 2000
Enter receiver's id: q23p&4ly!%

Funds Transferred successfully

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
------------------------------
Sender's Name: C.Amrit Subramananian
Sender's Email: amritsubramanian.c@gmail.com
Sender's Id: k6%z1q!t!2
Transaction Time: 2022-02-17 07:18:14.750363
Reciever's Name: sangeetha
Reciever's Email: sangeetha.karat@gmail.com
Reciever's Id: q23p&4ly!%
Funds Transferred: $2000.0
Royalty Fee: $1
------------------------------
Your Current Balance: $16891.0
------------------------------
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Transfer Email



CALIBER FINANCES Inbox ×

amritsubramanian.c@gmail.com

to me

Funds Transferred successfully

>>

Sender's Name: C.Amrit Subramananian
Sender's Email: amritsubramanian.c@gmail.com
Sender's Id: k6%z1q!t!2
Transaction Time: 2022-02-17 07:18:14.753824
Reciever's Name: sangeetha
Reciever's Email: sangeetha.karat@gmail.com
Reciever's Id: q23p&4ly!%
Funds Transferred: $2000.0
Royalty Fee: $1

Your Current Balance: $16891.0

>>

Displaying User's Info

```
(enter your choice)-> 7

-----------------------------
User's Info                  |
-----------------------------
User name: C.Amrit Subramananian
User id: k6%z1q!t!2
User balance: $16891.0
account created on: 2022-02-17 06:50:22.355179
User's email id: amritsubramanian.c@gmail.com
-----------------------------
```

Changing 4 -Digit Pincode

```
(enter your choice)-> 4
Enter old 4 digit pin:
Enter your new pin:
Enter your new pin:
pincode updated successfully...Login again...
```

Changing Password

```
(enter your choice)-> 5
Enter 4 digit pin:
Enter old password:
Enter new password:
Enter new password:
password updated successfully...Login again...
```

Getting Transactional Email in CSV Format

CALIBER FINANCES Inbox ×

amritsubramanian.c@gmail.com <amritsubramanian.c@gmail.com>
to me

Your transactional history upto now has been written to the csv file pinged down:

k6%z1q!t!2.csv

k6%z1q!t!2.csv

	A	B	C	D
1	Credited	Debited	Transaction Time	Account Balance
2	$20093.0	——	2022-02-17 7:11:35	$20092.0
3	——	$1200.0	2022-02-17 7:14:14	$18891.0
4	——	$2000.0	2022-02-17 7:18:15	$16891.0

CBSE Class 12 textbook Computer Science with Python - Preeti Arora.

CBSE Class 11 textbook Computer Science with Python - Preeti Arora

https://docs.python.org/

https://www.djangoproject.com/

https://opencv.org/

https://www.hackerrank.com/

https://codewithmosh.com/

https://www.timbuchalka.com/

https://www.programiz.com/

https://www.w3schools.com/

https://www.geeksforgeeks.org/

https://leetcode.com/

https://en.wikipedia.org/

https://stackoverflow.com/

Thank You