Hi, myself Anagha Mohan. I am a B-Tech student of Thangal Kunju Musaliar College of Engineering, Kollam, Kerala. Currently I’m a first-year student of Computer Science Engineering department, TKMCE.

# MAJOR PROJECT 1

AIM

To choose any dataset and apply a suitable ML technique and deploy it.

THE ML MODEL

import pandas as pd

df =pd.read\_csv('train.csv')

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 51054 entries, 0 to 51053

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 lyric 51054 non-null object

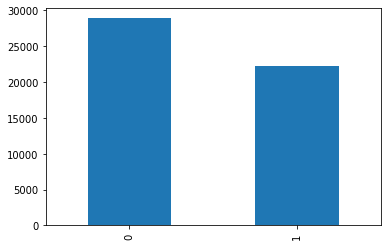
1 class 51054 non-null int64

dtypes: int64(1), object(1)

memory usage: 797.8+ KB

df['class'].value\_counts().plot(kind='bar')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fa3fe5673d0>



#input

x=df.iloc[:,0].values

#output

y=df.iloc[:,1].values

y

array([1, 0, 0, ..., 1, 0, 1])

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test= train\_test\_split(x,y,random\_state=0)

from sklearn.feature\_extraction.text import TfidfVectorizer

vect = TfidfVectorizer()

x\_train\_v = vect.fit\_transform(x\_train)

x\_test\_v = vect.transform(x\_test)

#apply SVC

from sklearn.svm import SVC

model = SVC()

model.fit(x\_train\_v,y\_train)

SVC()

#Pipeline

from sklearn.pipeline import make\_pipeline

text\_model =make\_pipeline(TfidfVectorizer(),SVC())

text\_model.fit(x\_train,y\_train)

Pipeline(steps=[('tfidfvectorizer', TfidfVectorizer()), ('svc', SVC())])

32s

#predict the variable

y\_pred =text\_model.predict(x\_test)

y\_pred

array([0, 0, 1, ..., 0, 1, 0])

y\_test

array([0, 0, 1, ..., 0, 1, 1])

#checking accuracy

from sklearn.metrics import accuracy\_score

accuracy\_score(y\_pred,y\_test)\*100

86.96333437793795

#random prediction test

c ='Cant drink without thinkin about you  '

c =vect.transform([c])

model.predict(c)

array([1])

HERE, 1 REPRESENTS RAP SONGS AND 0 REPRESENTS NON RAP SONGS.

import joblib

joblib.dump(text\_model,'train')

['train']

TEMPORARLY DEPLOYING

!pip install streamlit --quiet

%%writefile app.py

import streamlit as st

import joblib

model =joblib.load('train')

st.title('PREDICT THE TYPE')

ip=st.text\_input('ENTER THE SONG')

op= model.predict([ip])

if st.button('PREDICT'):

 st.title(op[0])

Writing app.py

#temporary deployment

!streamlit run app.py & npx localtunnel --port 8501

2022-07-09 11:16:41.372 INFO numexpr.utils: NumExpr defaulting to 2 threads.

**You can now view your Streamlit app in your browser.**

Network URL: [**http://172.28.0.2:8501**](http://172.28.0.2:8501/)

External URL: [**http://35.185.210.223:8501**](http://35.185.210.223:8501/)

npx: installed 22 in 7.005s

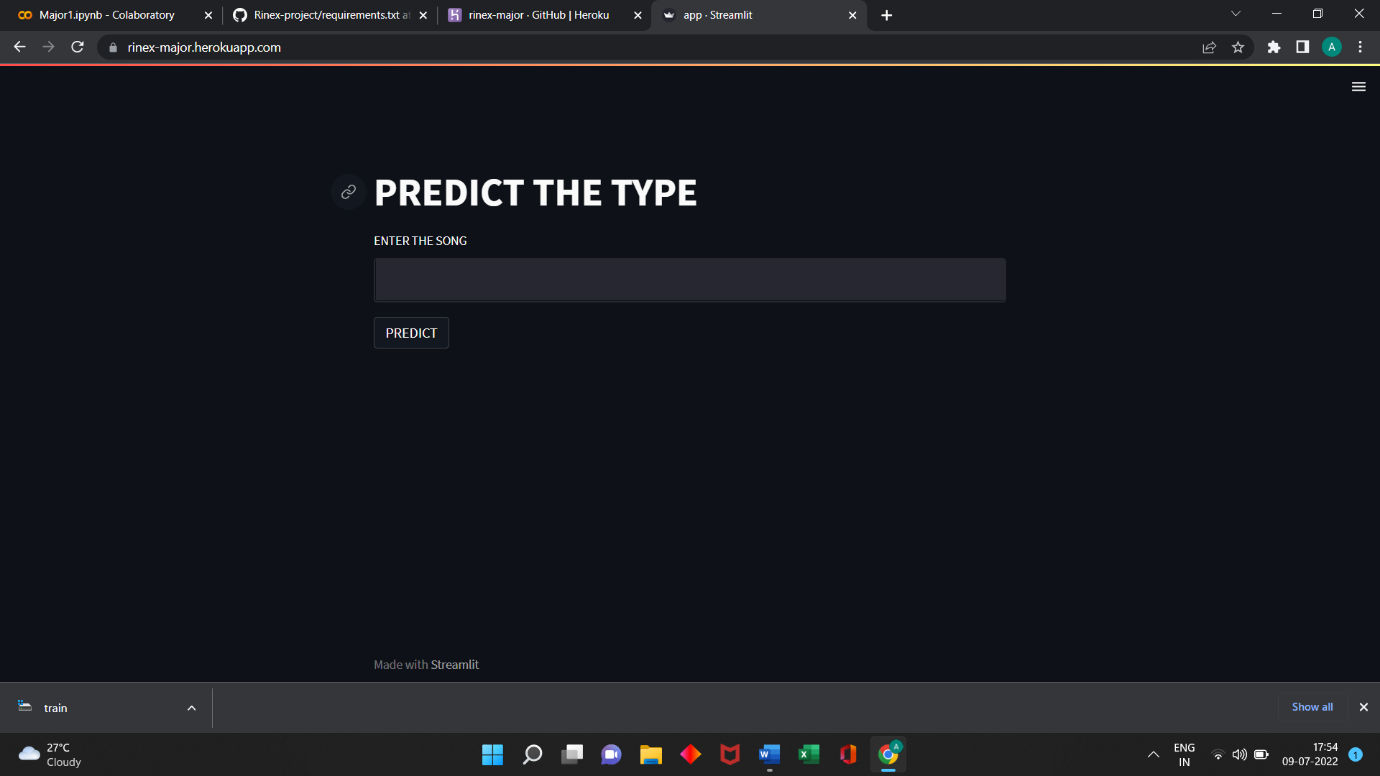
your url is: [https://thick-socks-greet-35-185-210-223.loca.lt](https://thick-socks-greet-35-185-210-223.loca.lt/)

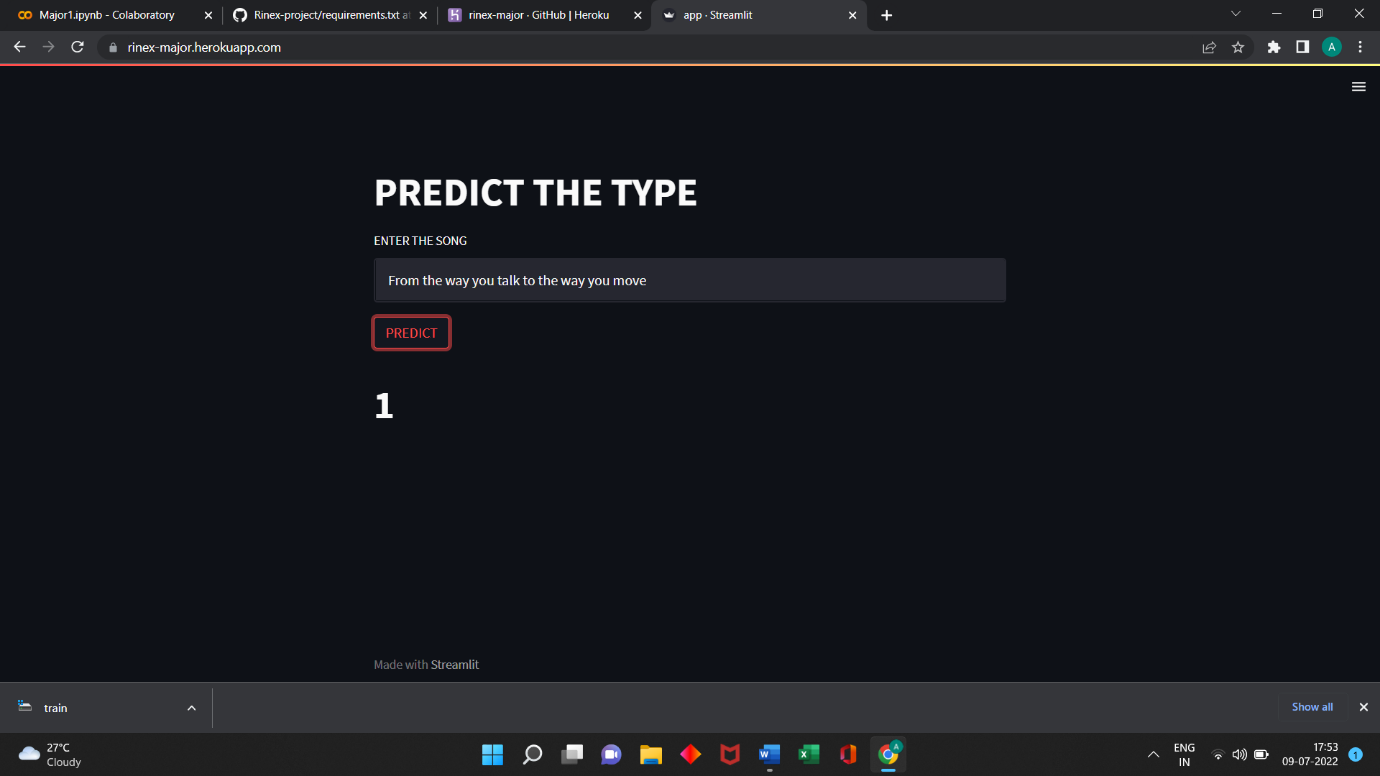
Stopping...

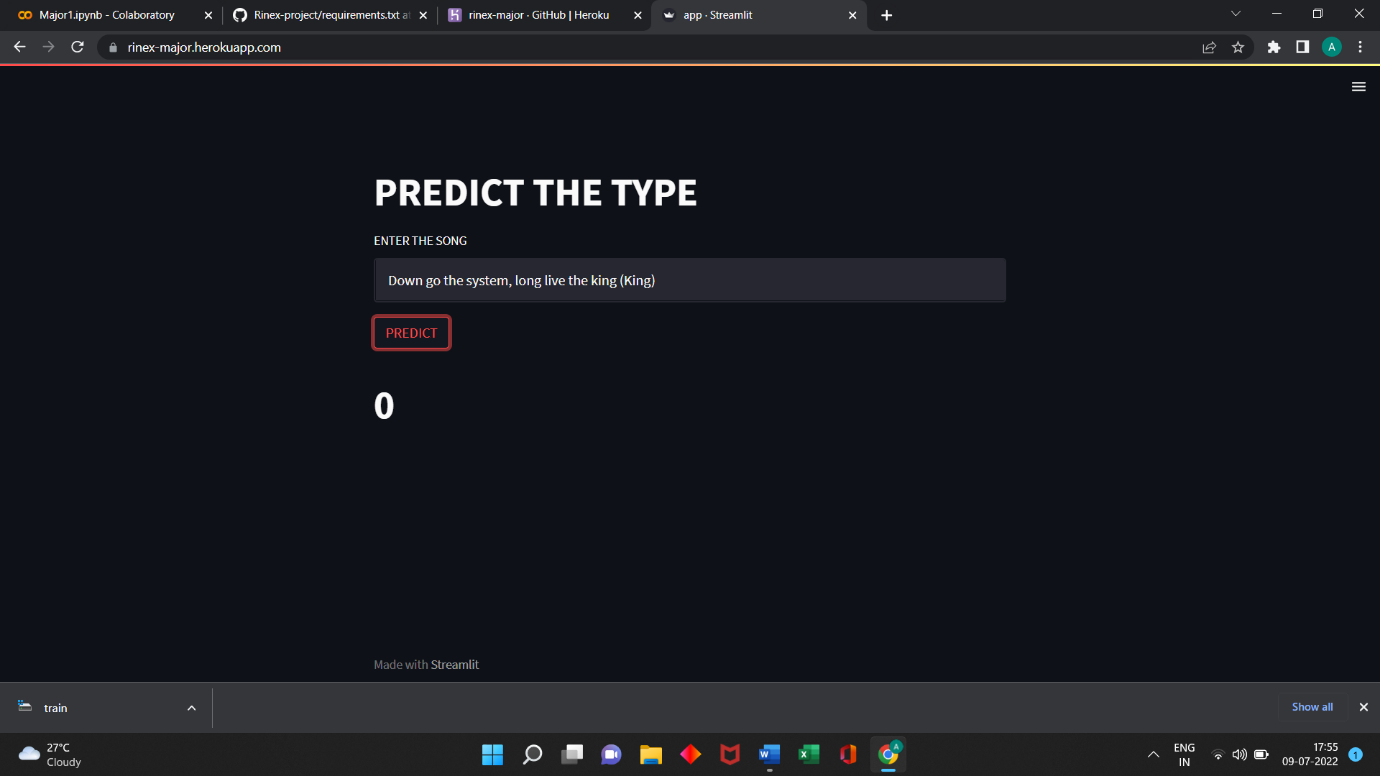
^C

OUTPUT

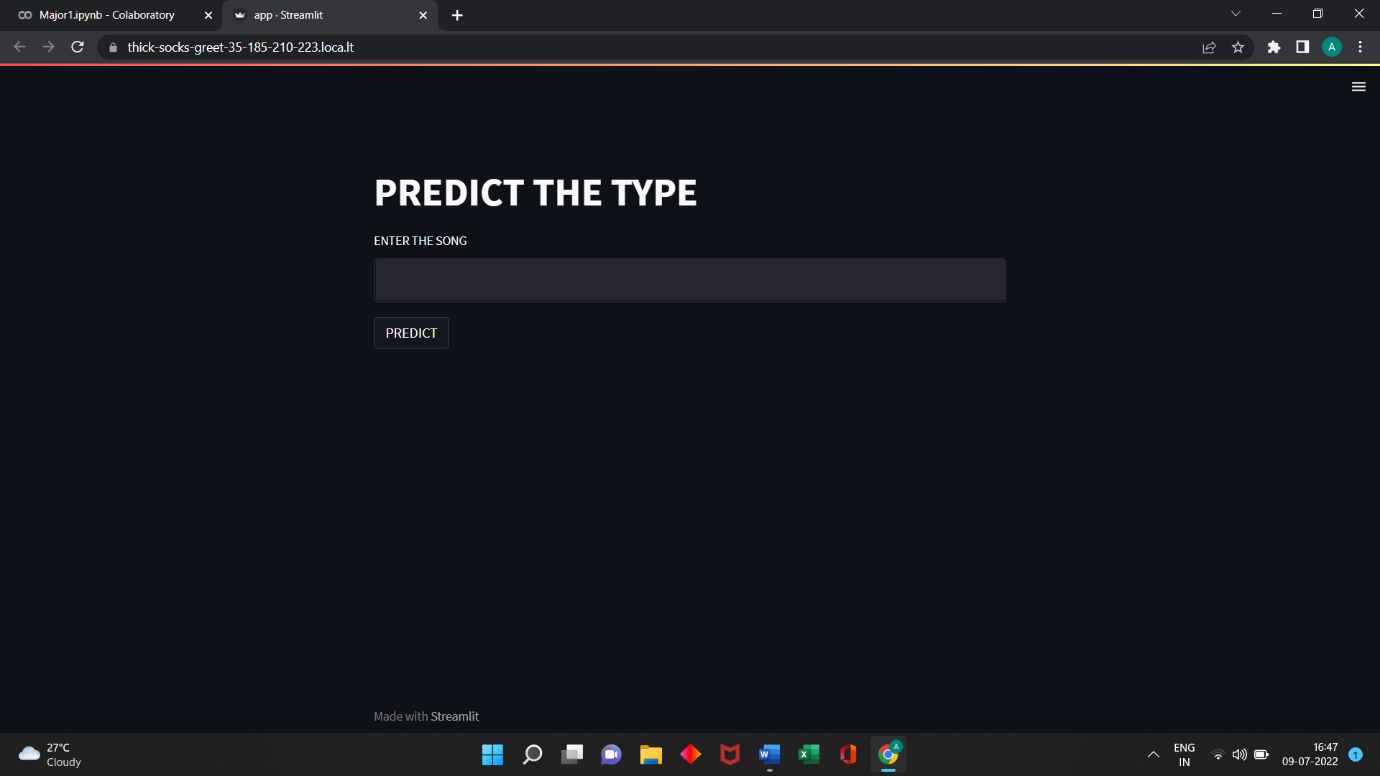
(Permanent Deployment- https://rinex-major.herokuapp.com/)







(Temporary deployment)



# 

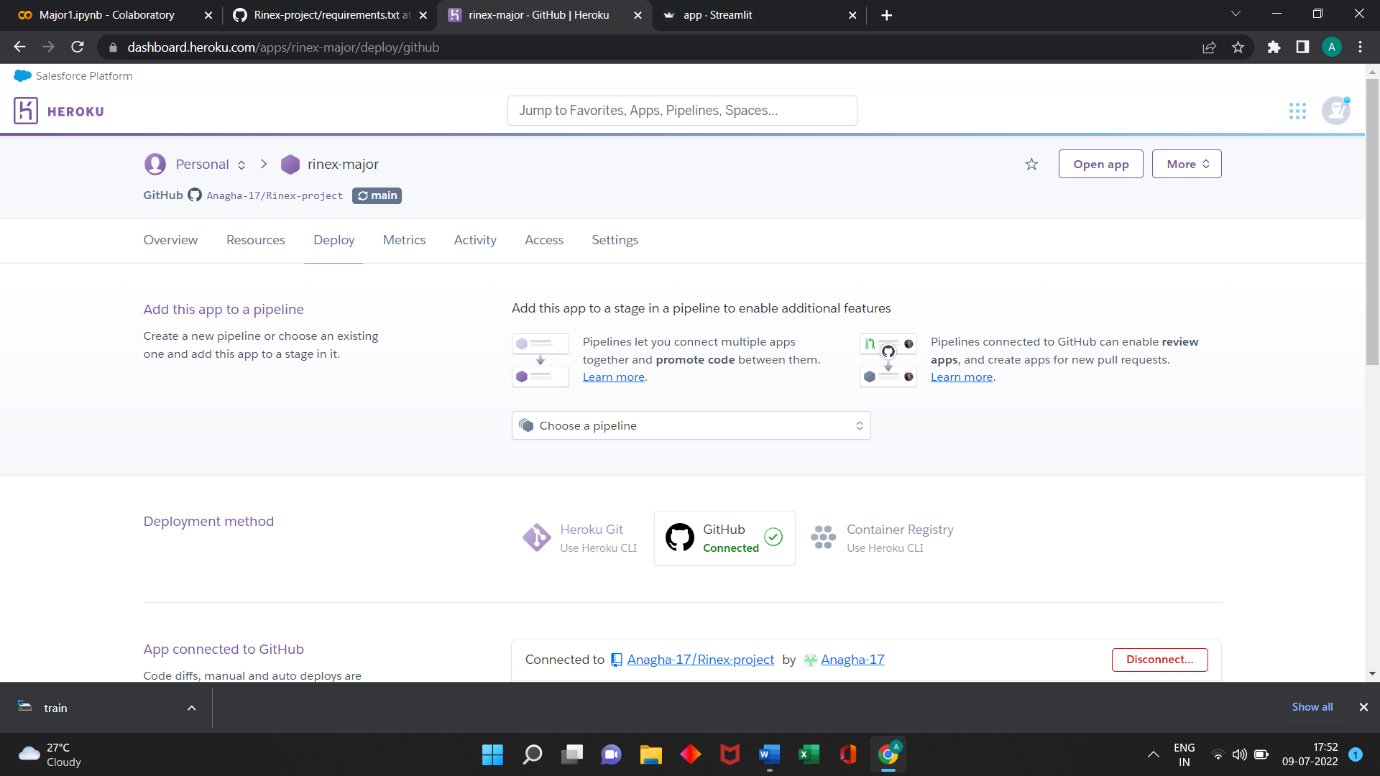
# 

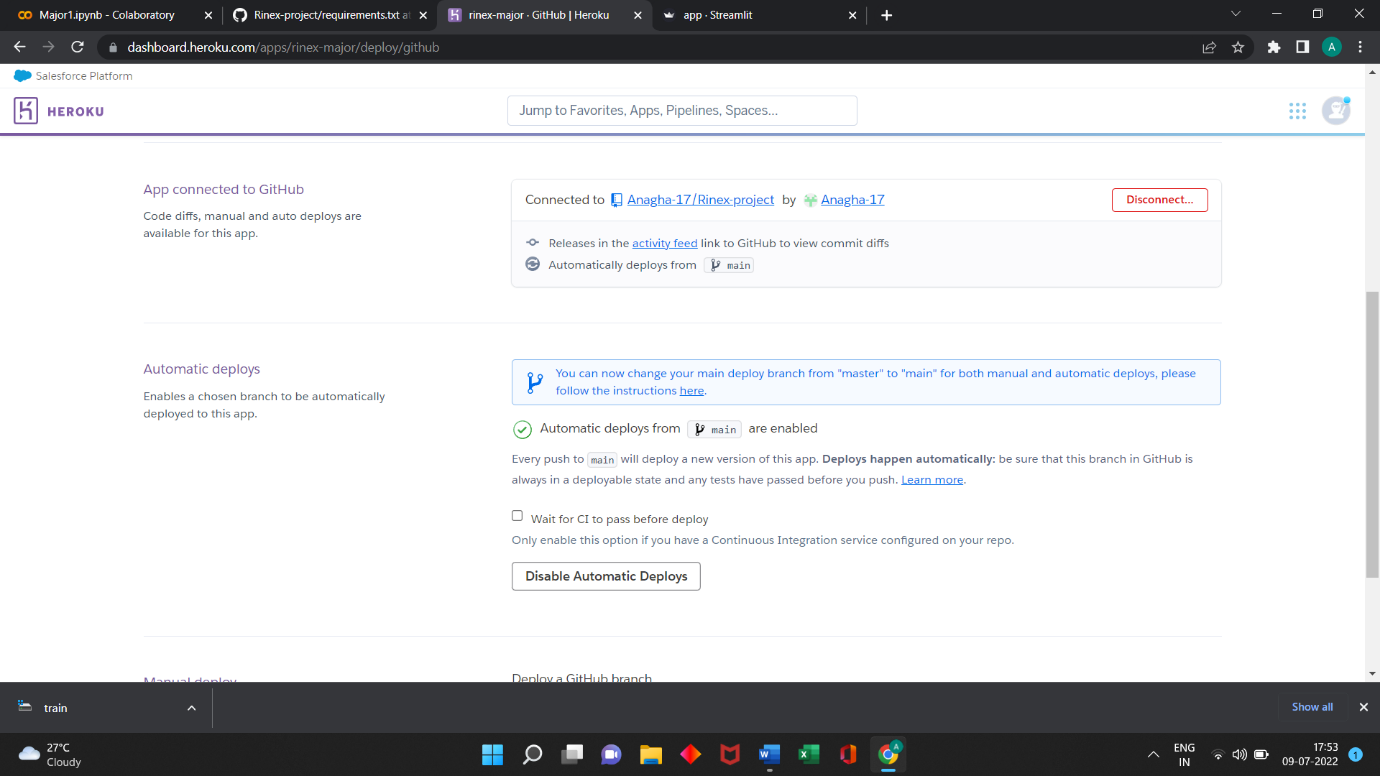
HEROKU

Heroku is **a container-based cloud Platform as a Service (PaaS)**. Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

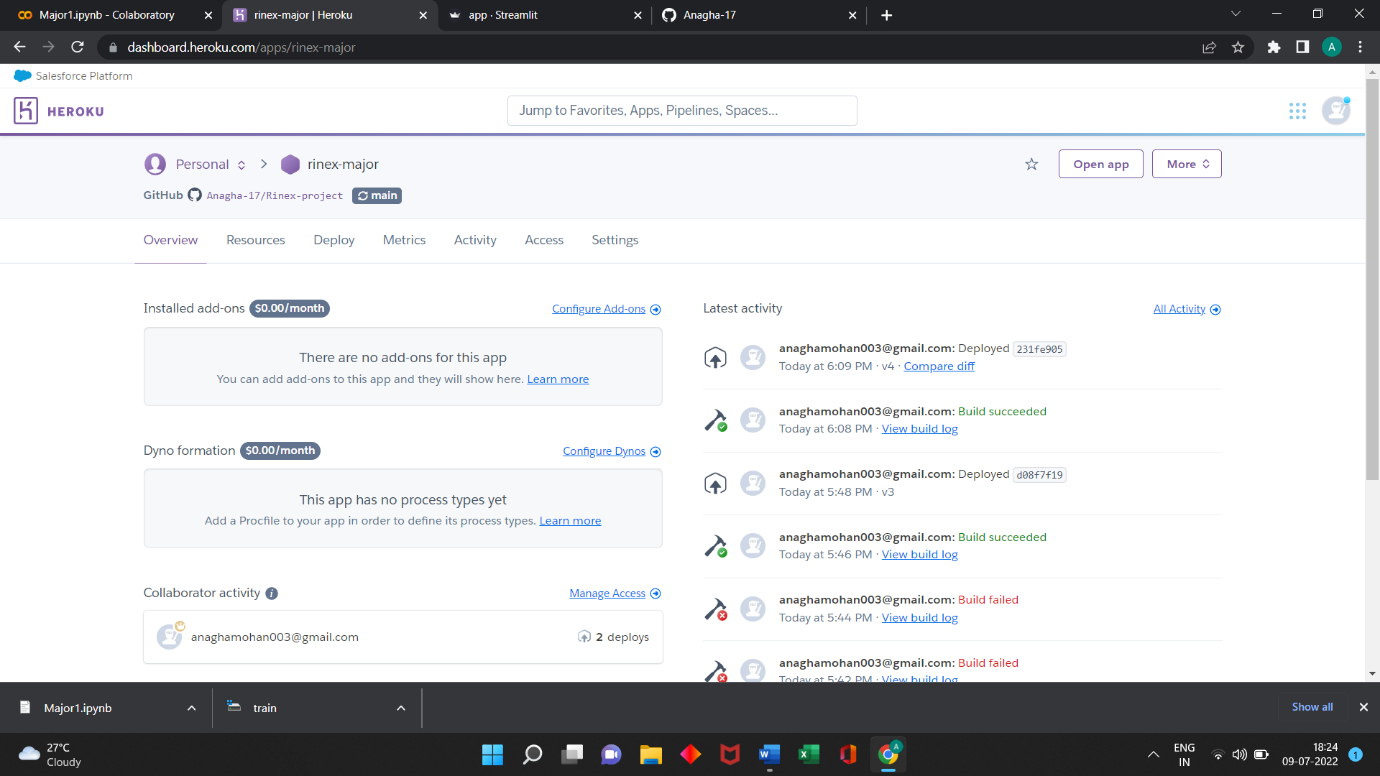
STEPS OF PERMANENT DEPLOYMENT

1.Connect your Git Hub repository with Heroku

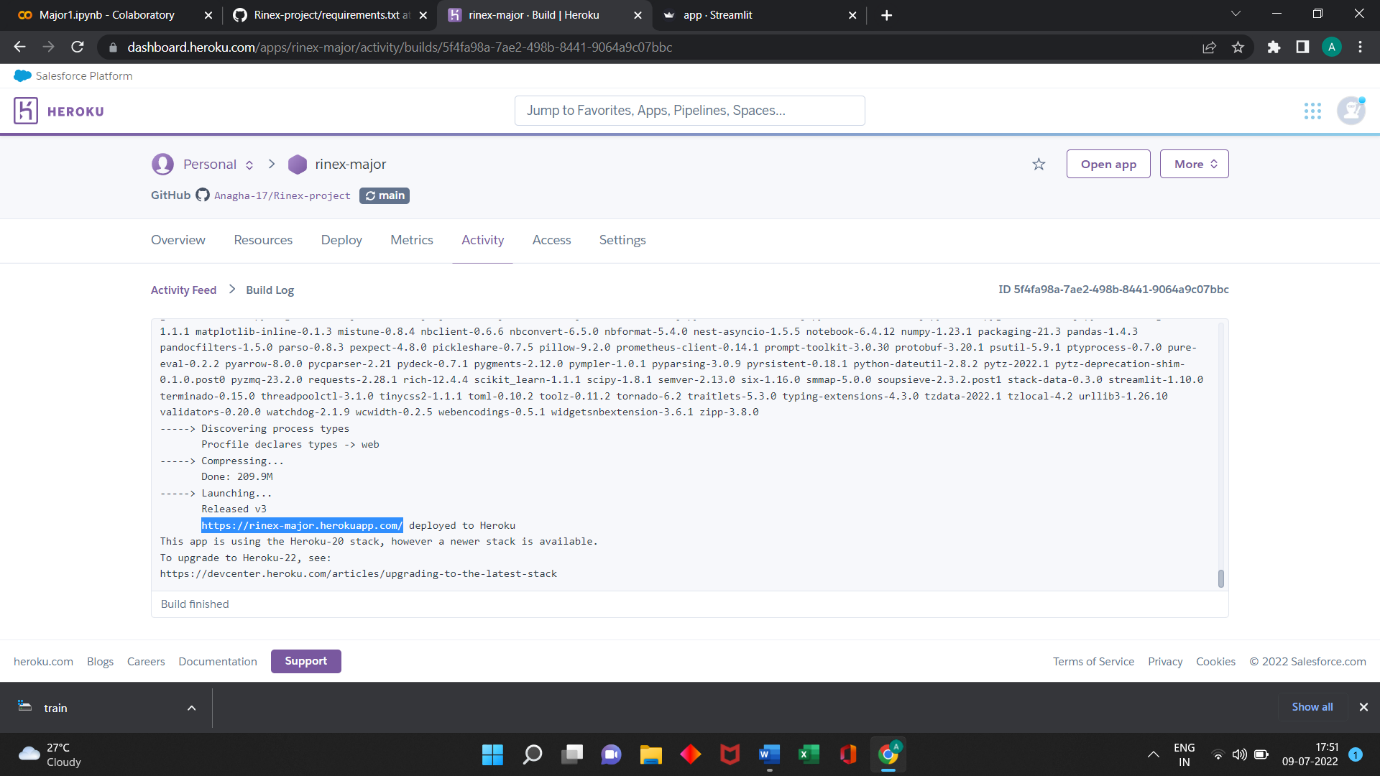




2.Go to over view and let the built of app get finished.



3.Once the built is complete,use the URL generated to open your web app.



MY GITHUB DETAILS

To access my Rinex major project, follow this URL

https://github.com/Anagha-17/Rinex-project

# MAJOR PROJECT 2

AIM

To create any image processing projects using numpy and OpenCV.

PROGRAM CODE

# TO SCALE UP AN IMAGE AND RECOGNISE FACES

import cv2

import numpy as np

img=cv2.imread('mp.jpg')

cv2.imshow('original',img)

cv2.waitKey(2000)

#face recognision

face\_cascade=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

faces=face\_cascade.detectMultiScale(img,scaleFactor=1.1,minNeighbors=1)

for x,y,w,h in faces:

cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),5)

#w,h -width ,height

#resizing

img =cv2.resize(img,None,fx=2,fy=2)

cv2.imshow('new',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

OUTPUT

