### News Tracker Using Bloomberg Market And Financial News API Using IBM Cloud

**1. Introduction:**

**1.1 Overview:-**

Reading newspaper is a good habit that can provide a great sense of educational value. It carries information about politics, economy, entertainment, sports, business, industry, trade and commerce. With this habit, it will not only enhance your knowledge about general information but it will likewise improve your language skills and vocabulary. Many people have habits of reading daily newspapers that their days seem incomplete without taking hold of early morning newspapers.But finding time to read all the news can't be possible and people has to search for their most wanted useful news.

**1.2 Purpose:-**

As our lives are very busy these days, we often feel we need more than 24 hrs a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Futures, Rates, Bonds, etc… as on official websites.

**2.LITERATURE SURVAY:-**

**2.1.Existing Problem:-**

* Technological evolution has transformed the way that we used to read the news. Today, we can easily roam the waves of the Web and find the news of our interests.
* You will find numerous e-newspaper apps out there that deliver the news. It can be really tricky to keep track of them all. With so many options choosing one newspaper app as the best option is slightly difficult, as one may have good content and latest updates, other may have a good speed or user-friendly interface.
* people need an app that delivers complete news fastly for business purposes like stock marketing etc.
* Existing apps may either provide a whole variety of news or one or two categories of news statically.

**2.2.Proposed Solution:-**

* The proposed solution to the problem of reading the news effectively and at the same time quickly involves fetching the news on the user's choice that too complete news in no time.
* For this the famous Bloomberg Financial and market news API is used to fetch the news from all over the world.
* Bloomberg News is One of the world’s leading financial news organizations,which produces roughly 5,000 stories a day and has earned more than 800 awards since it was founded in 1990 – including the 2015 Pulitzer Prize for explanatory reporting.
* It helps to reduce the time as well as to read the news completely.

**3.THEORTICAL ANALYSIS:-**

**3.1.Block Diagram:-**

Stop

USER

UI

News API

**3.2.Hardware/software designing**

**SoftwareRequirements:**

* OS - WindowsXP,7,8,10
* Jupyter notebook
* Flask library
* Anaconda Command Prompt

**Hardware Components:**

* Processor -i3 or above
* Hard disk storage - 10GB
* RAM - 1GB

**4.EXPERIMENTAL INVESTIGATIONS:-**

The main objective of this project is to build a news app that requests news from API.A POST request is made to the Bloomberg News API with the api key and the category the user is interested to fetch.On succesful response the link ,URL of news and thumbnail is retreived.All this information is structured into a pandas DataFrame and presented to the user.If the response doesn't contain the required information proper error handling is written .

**5.Flow Chart:-**

Stop

Proper message is shown

Start

Stop

Proper message is shown

Response from API is empty

Fetch From API

Display on webpage

Enter Category

**6.Results:-**

This application consists of 3 states mentioned below.

* First, input is taken from the user which is the category of the news, user is interested and want to know the updates about.It is used in the GET request payload field.
* Then a GET request is made with the users input as query string which yields a response object.
* This response object is parsed into json object(dict object) and a pandas DataFrame is created which is then rendered on the webpage in a neat GUI.

**7.ADVANTAGES AND DISADVANTAGES:-**

**Advantages:-**

* News reading is benificial and essential in many ways
* Reading newspaper is a good habit that can provide a great sense of educational value.
* It carries information about politics, economy, entertainment, sports, business, industry, trade and commerce.
* With this habit, it will not only enhance your knowledge about general information but it will likewise improve your language skills and vocabulary
* Allowing users to choose the interested category in real time gives flexibility.
* As the news is fetched form Bloomberg news API it is very accurate and world wide
* No need to scroll/search all categories news

**Disadvantages:-**

* User has to take of entering the right categor,sometimes appriopriate news can't be fetched from the API
* User can't enter multiple categories at a time ,because it will mess up.
* User can enter categories one by one to know about multiple categories.
* The GET request only fetches the original URL and thumbnail of the news,that is why it is fast.

**8.APPLICATIONS:-**

* **Investing in Stock Market:**

To invest in Stock Market,knowing the latest trends and market situation is very important.This involves to understand about a specific field(category)

* **Knowing NEWS around the world:**

This application can cover the news from all over the world as it fetches the news from the well reputed Bloomberg News API.

**9.CONCLUSION:-**

This applications uses Bloomberg News API which is one of the world's leading financial news organization which produces roughly 5,000 stories a day and has earned more than 800 awards since it was founded in 1990 – including the 2015 Pulitzer Prize for explanatory reporting and thus provides the users with wide variety of news to explore.

This application basically enables users to choose a category and get the latest news on that from all over the world.Thus it gives a wide exposure to the users and helps in investing,fututre decisions,knowledge etc.

**10.FUTURE SCOPE:-**

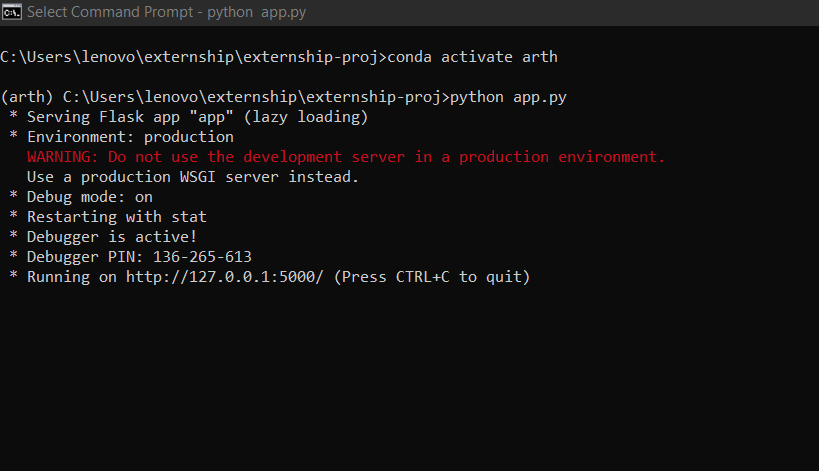
This application has a great advantage in business industry and to take economic decisions wisely.As one needes to aware of present economy and world staus ,this application can be found in many fileds.This makes the news reading easy and can be improved to include full articles rather than links.

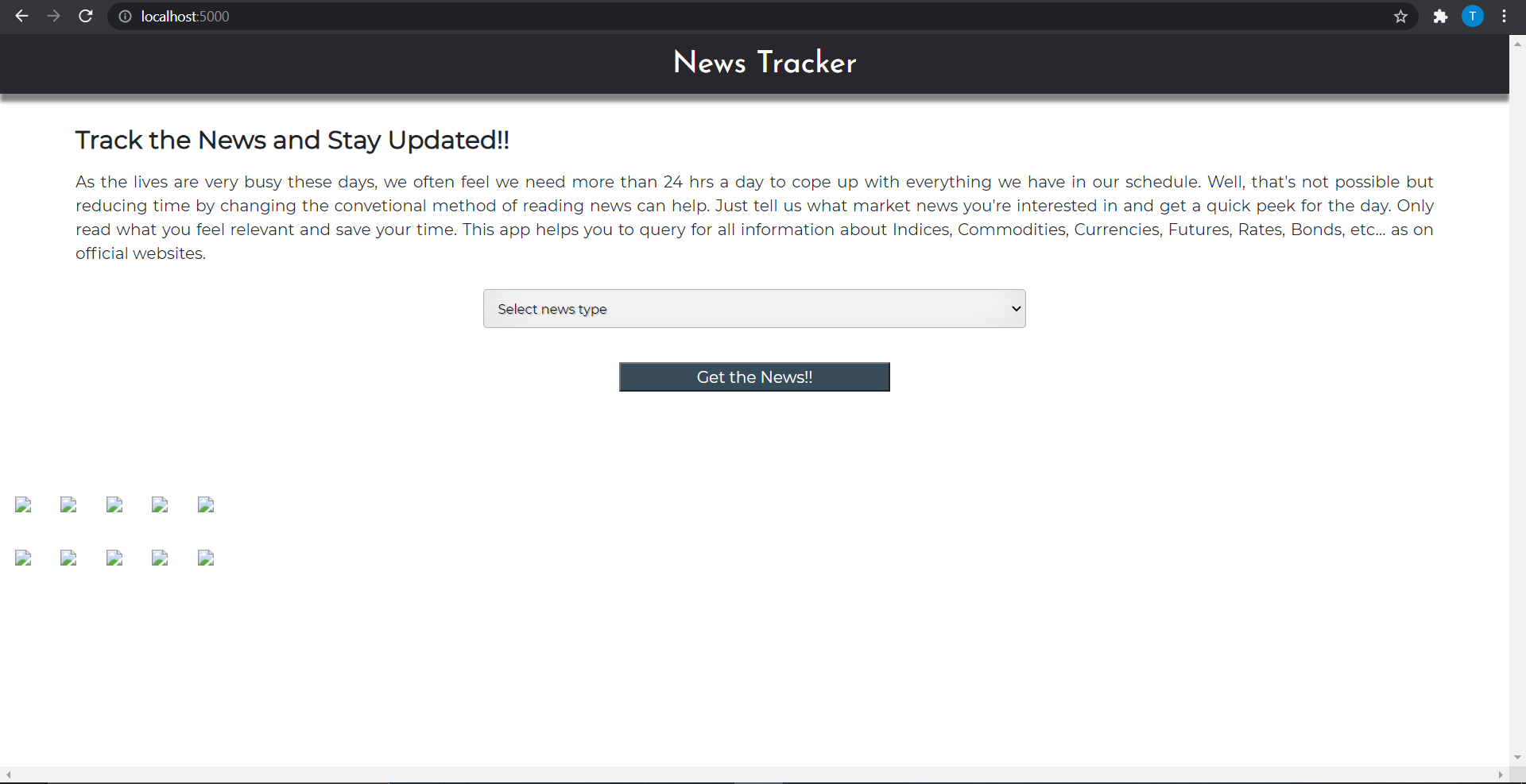
**11.BIBILOGRAPHY:-**

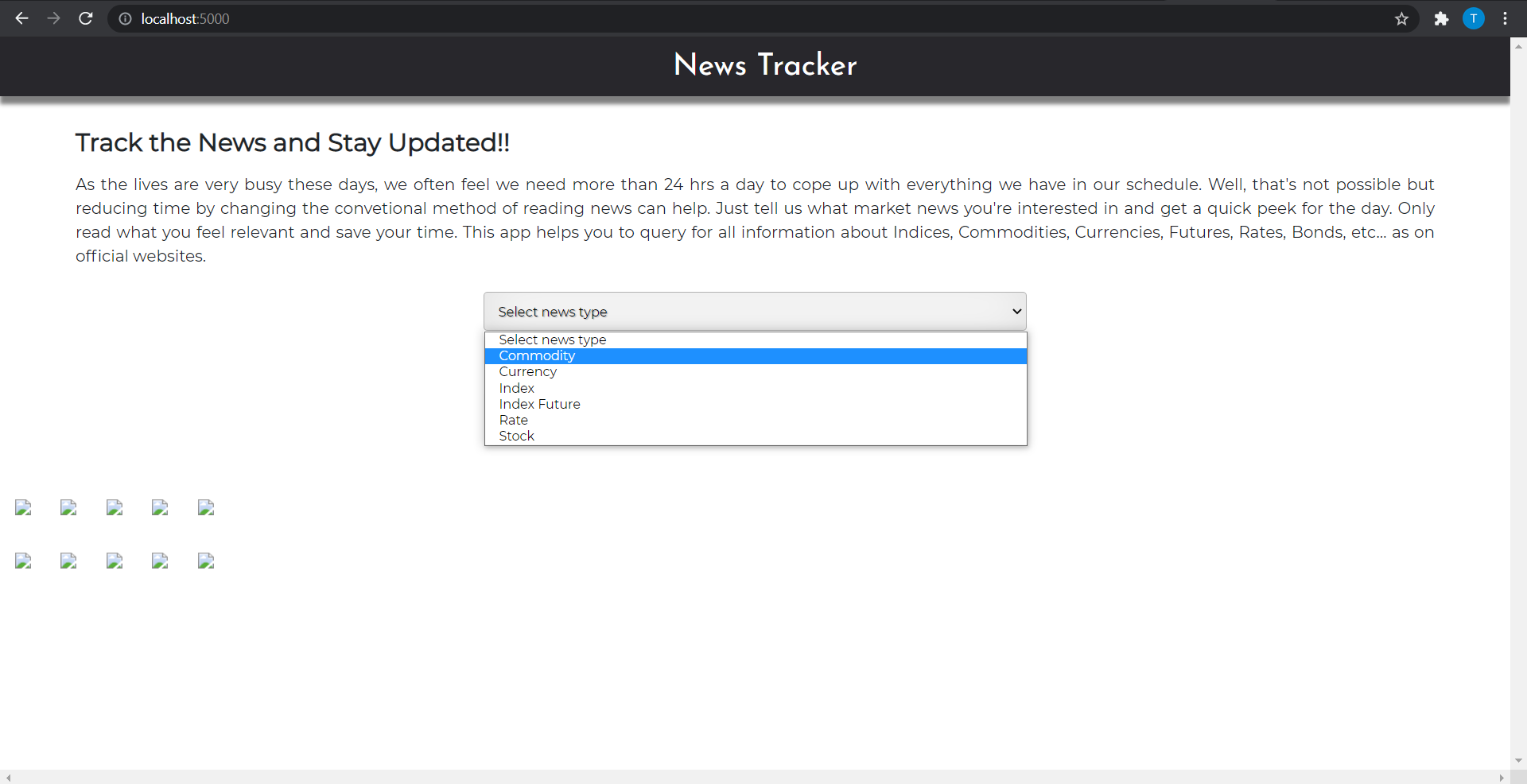
# 1.Flask Web Development: Developing Web Applications with Python by [Miguel Grinberg](https://www.amazon.com/Miguel-Grinberg/e/B00J23SQ34/ref=dp_byline_cont_book_1)

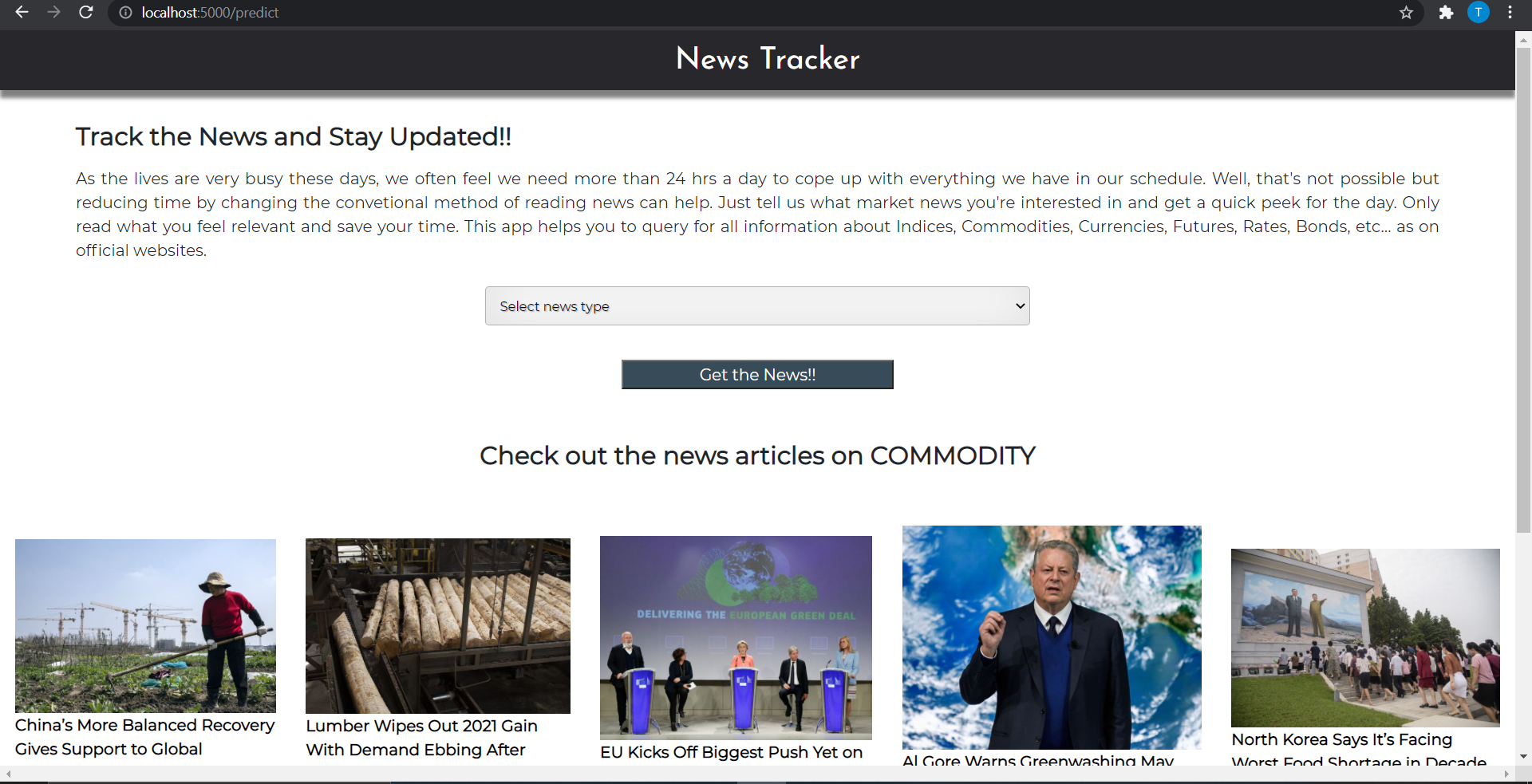
2.[RESTful API Design](https://www.amazon.com/dp/1514735164?tag=uuid10-20) by Matthias Biehl

**SCREENSHOTS:-**

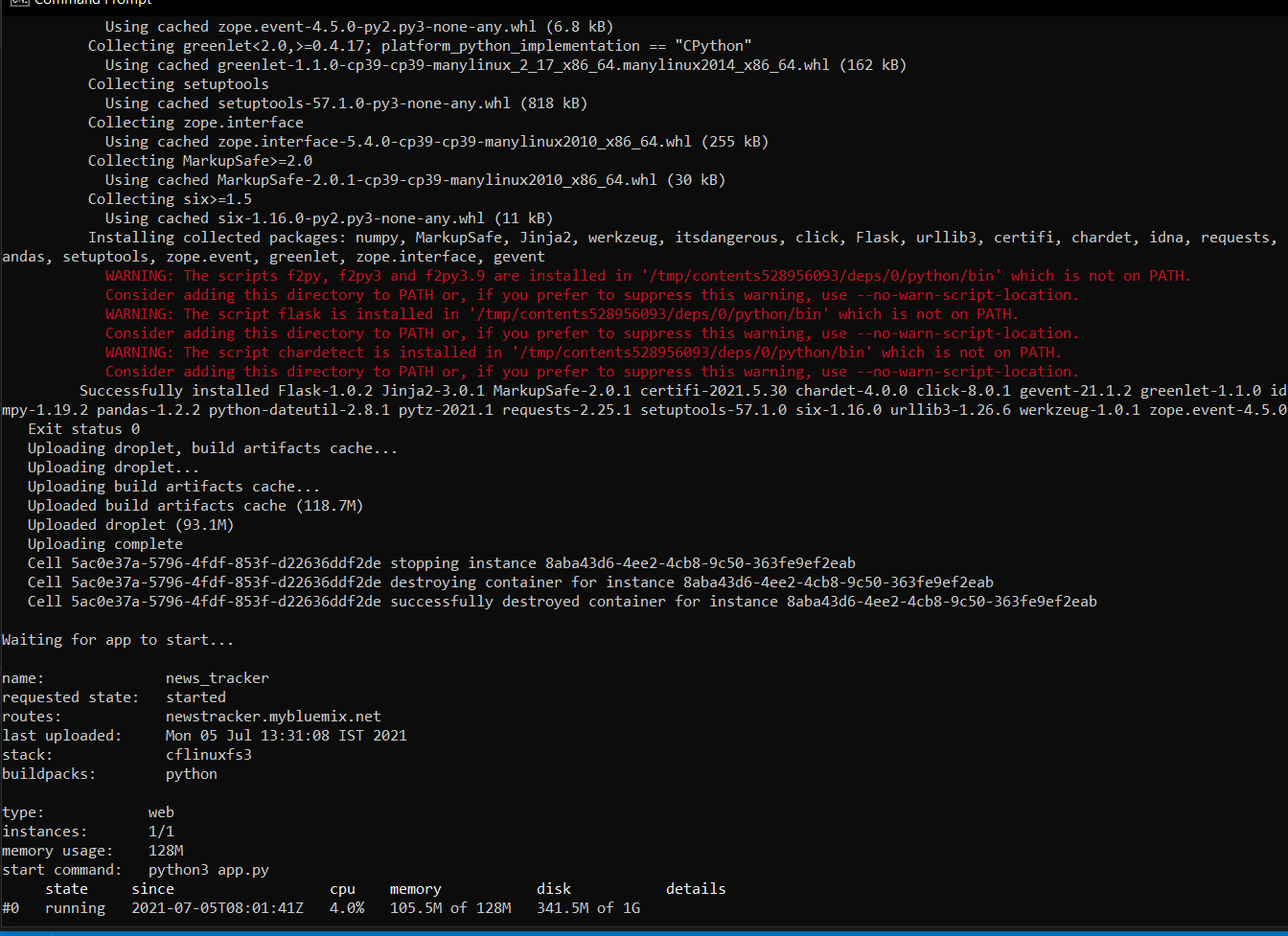




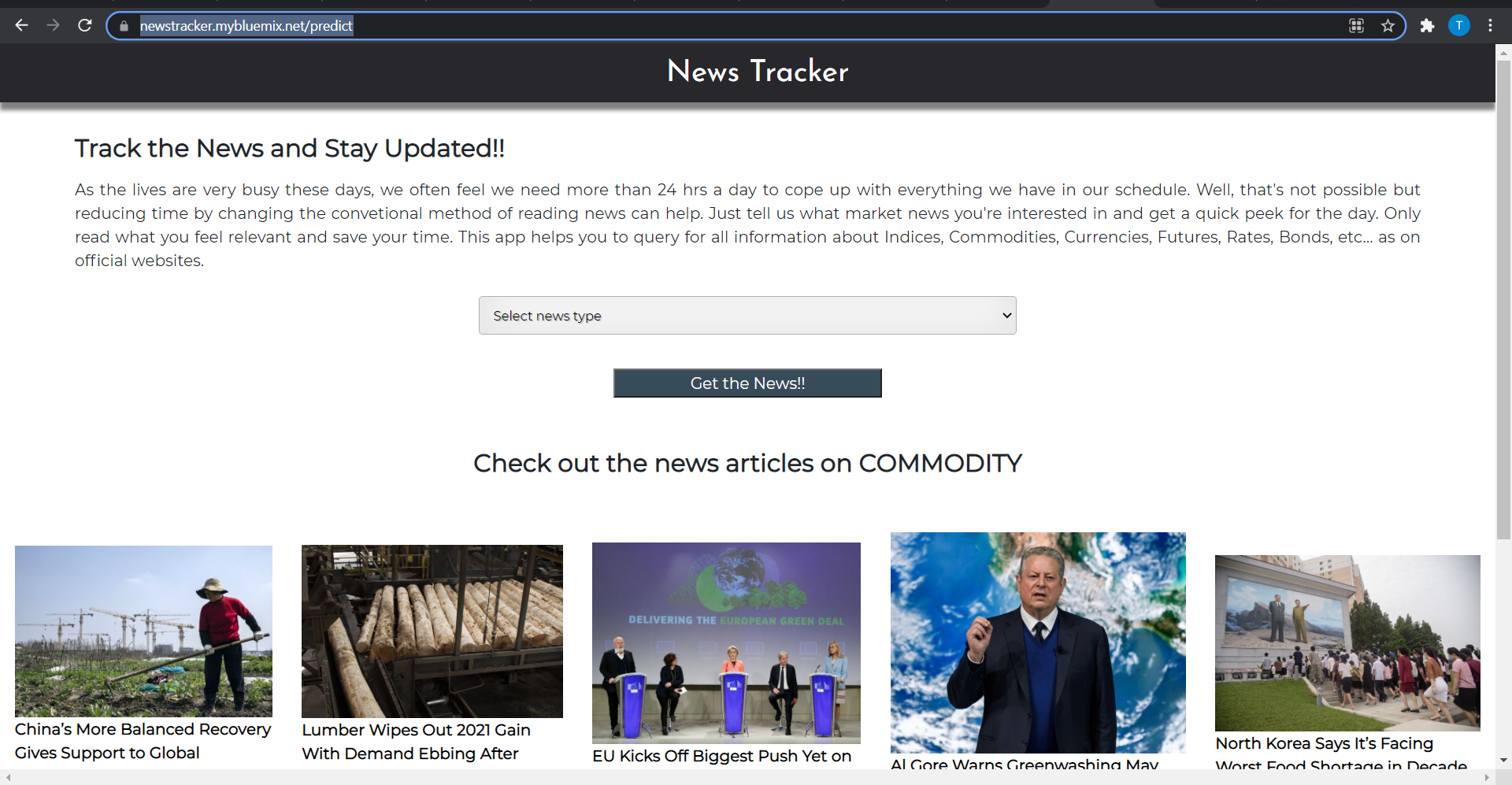




**Deploying On IBM CloudFoundry:**



**Accessing the deployed app With App URL:**



**Source Code:**

Flask App-app.py

# -\*- coding: utf-8 -\*-

#importing required libraries

from flask import Flask, request, render\_template

import numpy as np

import re

import requests

import pandas as pd

import json

# from gevent.pywsgi import WSGIServer

# import os

#initializing the flask app

app = Flask(\_\_name\_\_)

def check(user\_inp):

url = "https://bloomberg-market-and-financial-news.p.rapidapi.com/stories/list"

querystring = {"template":user\_inp,"id":"usdjpy"}

headers = {

'x-rapidapi-key': "d730d682demshbfaf2fc584babdap15184djsnd2eec3eec4be",

'x-rapidapi-host': "bloomberg-market-and-financial-news.p.rapidapi.com"

}

response = requests.request("GET", url, headers=headers, params=querystring)

result = response.json()

df = pd.DataFrame(columns = ['Card', 'Title', 'Thumbnail Link','Link to source of news'])

if result['stories']:

for story in result['stories']:

# print(story['thumbnailImage'])

story.setdefault('card','')

story.setdefault('title','')

story.setdefault('thumbnailImage','')

story.setdefault('longURL','')

# print(len(story))

df.loc[len(df.index)] =[story['card'],story['title'],story['thumbnailImage'],story['longURL']]

print(df)

return df

#home page

@app.route('/')

def home():

return render\_template('base.html')

#News results page

@app.route('/predict',methods=['POST'])

def y\_predict():

output=request.form['news1']

df=check(output)

print(df)

if df.empty:

str="data is not available sorry!"

return render\_template('base.html',main=str)

str="Check out the news articles on "+output

return render\_template('base.html',main=str,

img1=df["Thumbnail Link"][0],img2=df["Thumbnail Link"][1],img3=df["Thumbnail Link"][2],

img4=df["Thumbnail Link"][3],img5=df["Thumbnail Link"][4],img6=df["Thumbnail Link"][5],

img7=df["Thumbnail Link"][6],img8=df["Thumbnail Link"][7],img9=df["Thumbnail Link"][8],

img10=df["Thumbnail Link"][9],

src1=df["Link to source of news"][0],src2=df["Link to source of news"][1],src3=df["Link to source of news"][2],

src4=df["Link to source of news"][3],src5=df["Link to source of news"][4],src6=df["Link to source of news"][5],

src7=df["Link to source of news"][6],src8=df["Link to source of news"][7],src9=df["Link to source of news"][8],

src10=df["Link to source of news"][9],

data1=df["Title"][0],data2=df["Title"][1],data3=df["Title"][2],data4=df["Title"][3],data5=df["Title"][4],

data6=df["Title"][5],data7=df["Title"][6],data8=df["Title"][7],data9=df["Title"][8],data10=df["Title"][9])

if \_\_name\_\_ == "\_\_main\_\_":

# port = os.getenv('VCAP\_APP\_PORT','8080')

# app.secret\_key=os.urandom(12)

# app.run(debug=True,port=port,host='0.0.0.0')

app.run(debug=True)