Project Name: Keyword Extraction From The News Feed Or Blogposts With IBM Cloud

1.Introduction:

Keyword extraction is the automated process of extracting the most relevant words and expressions from the text. With more than 290 billion emails sent and received on a daily basis, and half a million tweets posted every single minute, using machines to analyze huge sets of data and extract important information is definitely a game-changer. We are building a flask application by using a keywords extraction API. This API uses advanced Natural Language Processing techniques to extract the most important 'Keywords' and 'Phrases' from a LIST of text or URL that you provide. It can take any type of text as input and will perform best on URLs corresponding to news, blog, content, etc. It can also take MULTIPLE text or URLs as input and you can specify the number of keywords that need to be extracted.

a.Overview:

**Keyword extraction** (also known as **keyword** detection or **keyword** analysis) is a text analysis technique that automatically extracts the most used and most important words and expressions from a text. It helps summarize the content of texts and recognize the main topics discussed.

b.Purpose:

The **keyword extraction** process helps us in identifying the important words. It also effective in topic modeling tasks. You can know a lot about your text data by only a few **keywords**. These **keywords** will help you to determine whether you want to read an article or not.

2.Literature Survey:

a.Existing Problem:

The main **problem** that researchers are facing is how to efficiently and accurately **extract** the core **keywords** from a document. However, previous **keyword extraction** approaches have utilized the text and graph features, there is the lack of models that can properly learn and combine these features in a best way.

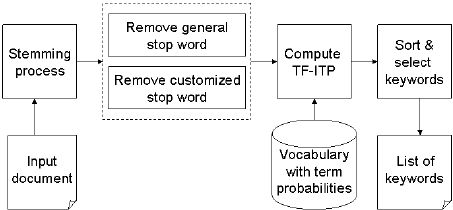
b.Proposed Solution:

Keyword extraction simplifies the task of finding relevant words and phrases within [unstructured text](https://monkeylearn.com/unstructured-data/). This includes emails, social media posts, chat conversations, and any other types of data that are not organized in any predefined way.Keyword extraction can automate workflows, like tagging incoming survey responses or responding to urgent customer queries, allowing you to save huge amounts of time. It also provides actionable, data-driven insights to help make better business decisions. But the best thing about keyword extraction models is that they are easy to set up and implement.There are different techniques you can use for automated keyword extraction. From simple statistical approaches that detect keywords by counting word frequency, to more advanced machine learning approaches that create even more complex models by learning from previous examples.

In this section, we’ll review the different approaches to keyword extraction, with a focus on machine learning-based models.

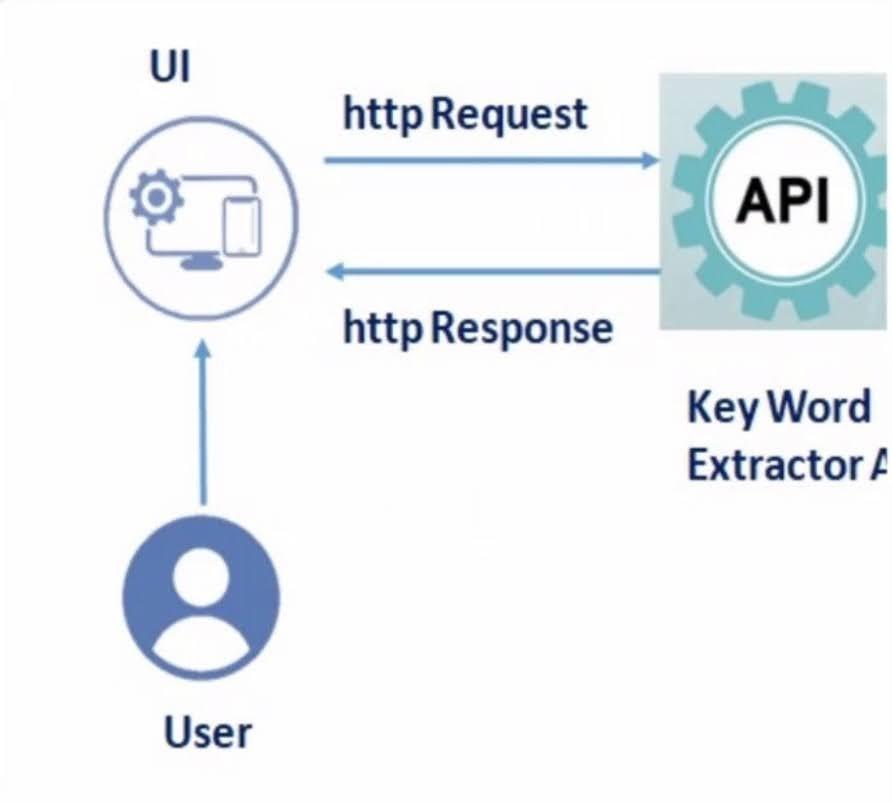
3.Theoretical Analysis:

a.Block Diagram:



b.Hardware Designing and software designing:

In Keyword Extraction we uses every Hardware designings and Software desinings



4.Experimental Investigation:

Keyword extraction (also known as *keyword detection* or *keyword analysis*) is a [text analysis](https://monkeylearn.com/text-analysis/) technique that automatically extracts the most used and most important words and expressions from a text. It helps summarize the content of texts and recognize the main topics discussed.

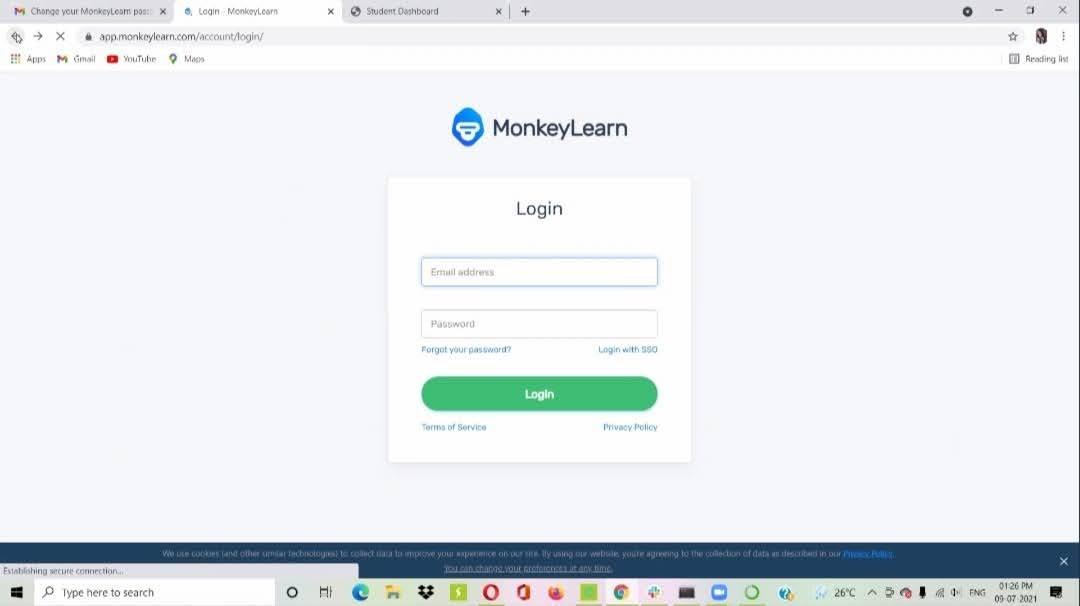
Keyword extraction uses [machine learning](https://monkeylearn.com/blog/gentle-guide-to-machine-learning/) artificial intelligence (AI) with [natural language processing (NLP)](https://monkeylearn.com/natural-language-processing/) to break down human language so that it can be understood and analyzed by machines. It’s used to find keywords from all manner of text: regular documents and business reports, social media comments, online forums and reviews, news reports, and more.First, we’ll recommend some books and academic papers for more in-depth explanations of keyword extraction methods and algorithms. Then, we’ll share some [APIs for keyword extraction](https://monkeylearn.com/blog/keyword-extraction-apis/), including open-source libraries and SaaS APIs.

Finally, we’ll provide some keyword extraction tutorials you can follow to get you up and running. Some of the tutorials show you how to run keyword extraction with open-source libraries with Python and R. However, if you prefer to save time and resources, you may find it useful to try a ready-made solution.

[MonkeyLearn](https://monkeylearn.com/), for example, has [pre-trained keyword extraction models](https://monkeylearn.com/keyword-extractor-online/) that you can dive right into. Or learn how to create your own customized models for detecting keywords within texts. We’ll walk you through that process and help you build a keyword extraction model adapted to your needs.

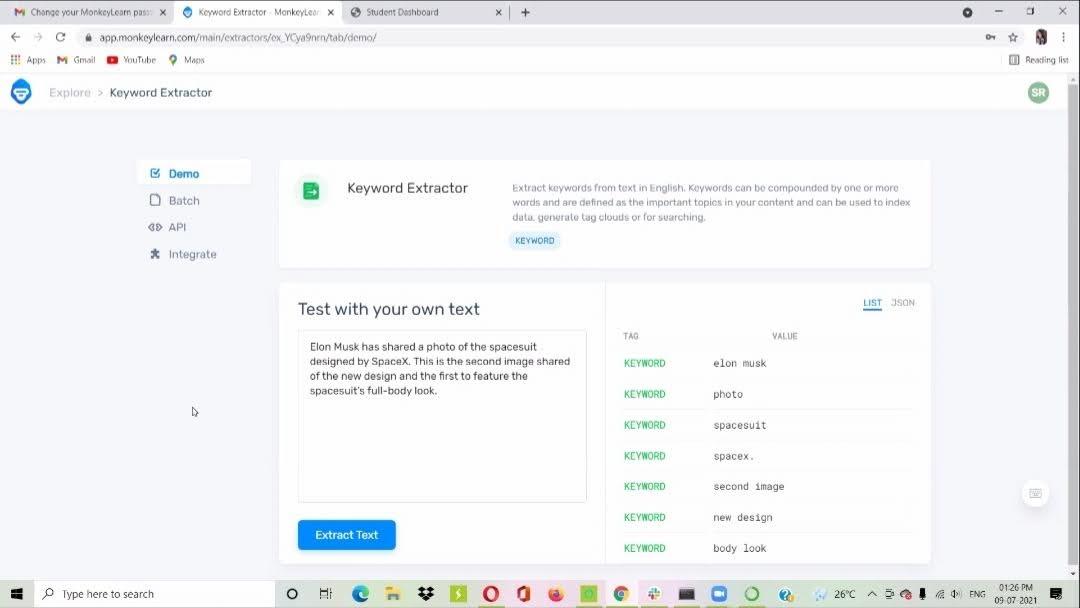
>Here in this monkeylearn is used ..To install monkeylearn we need to type a command i.e, pip install monkey learn.

>we need to create a Monkeylearn account to get API key and Model id.

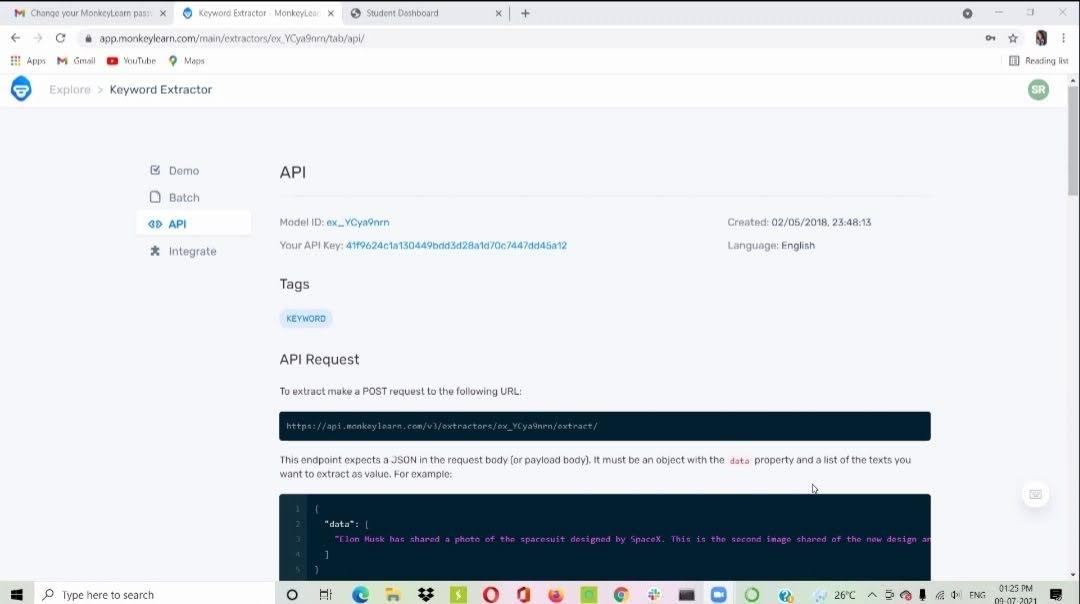


>Then after creating Monkeylearn account

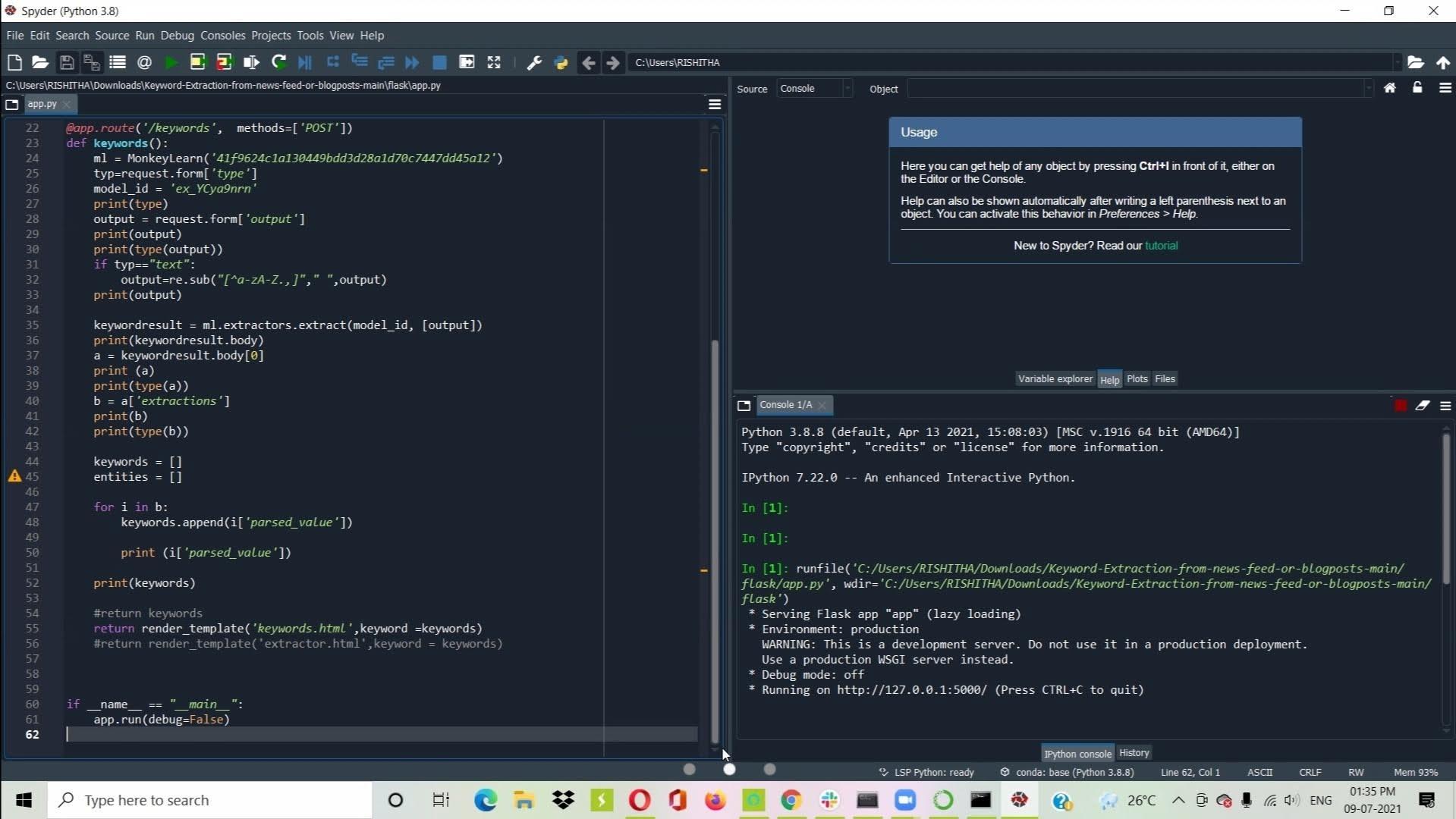
>Here select Keyword Extractor.. Then



> Next here Api key and Model id:

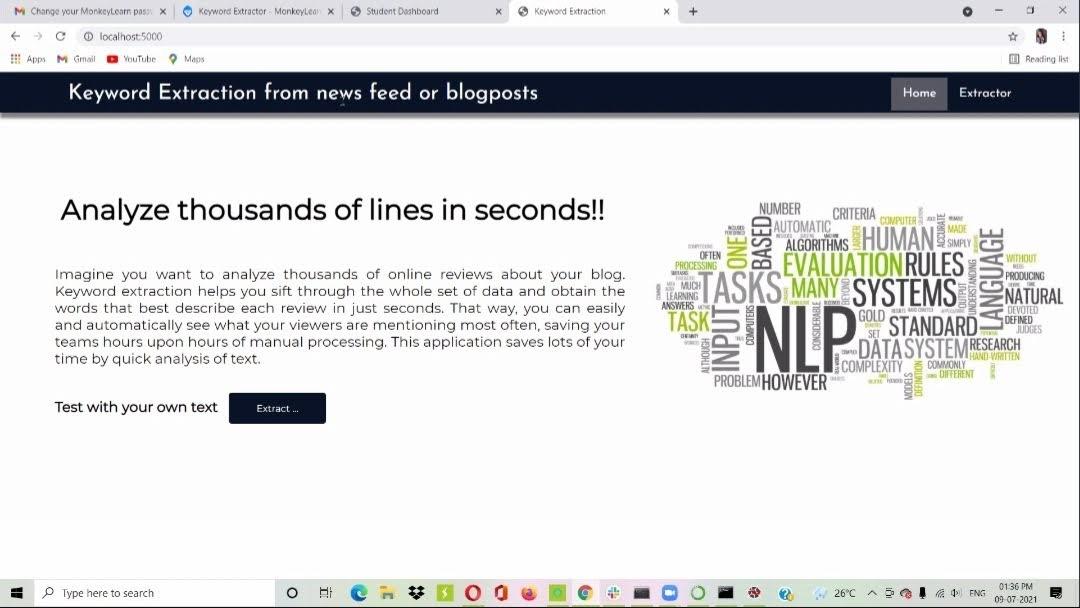


>After typing in code in spyder. We need to replace that Api Key and Model id.



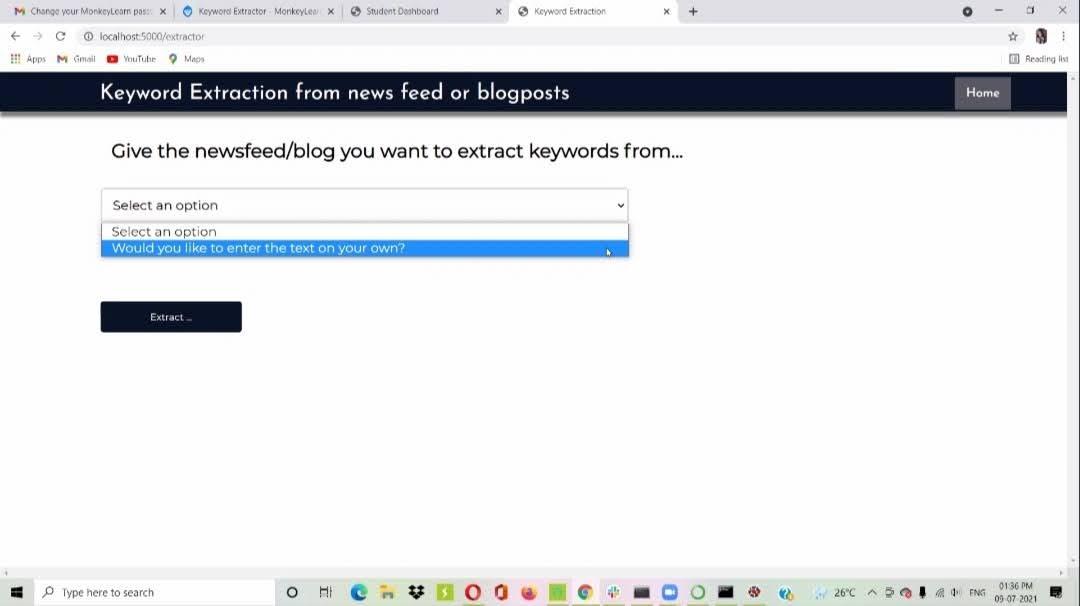
> Then run the code and Navigate to localhost:5000

6.Result:



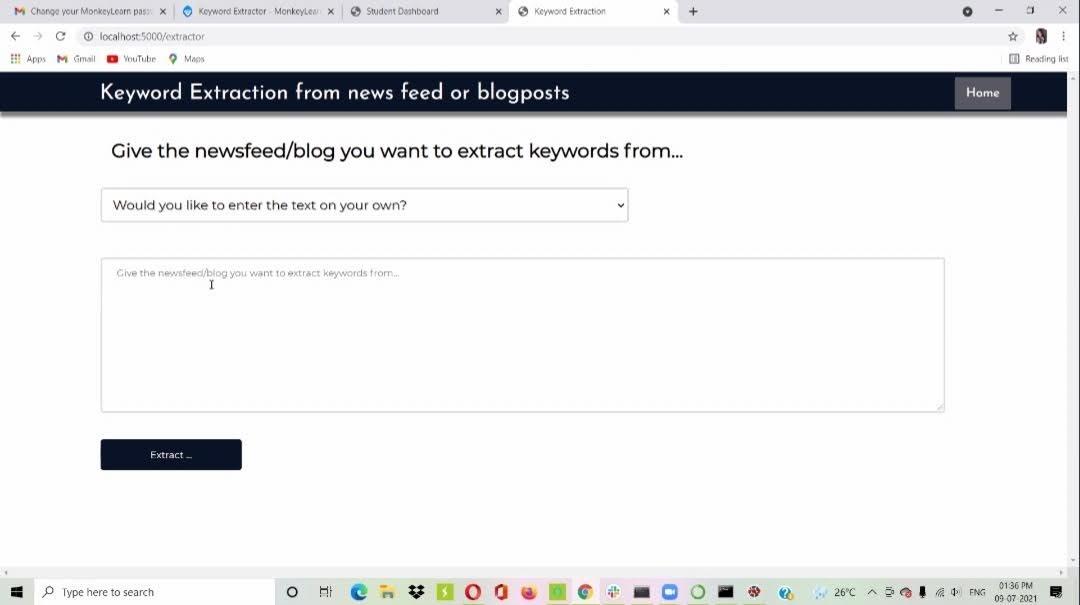
After Navigating to localhost:5000. Then we get this

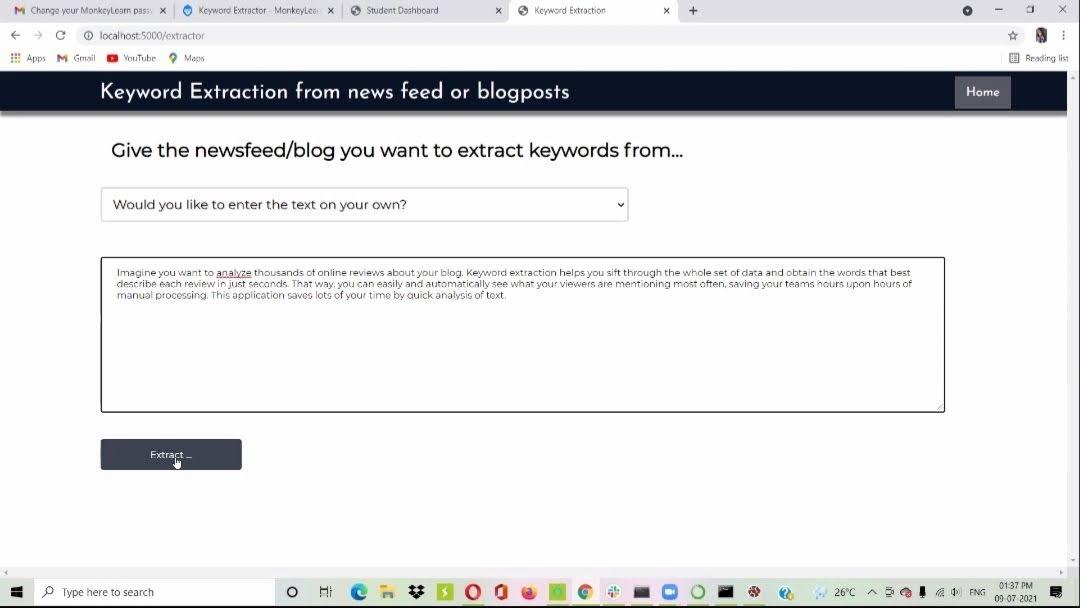
> Here we need to extract.



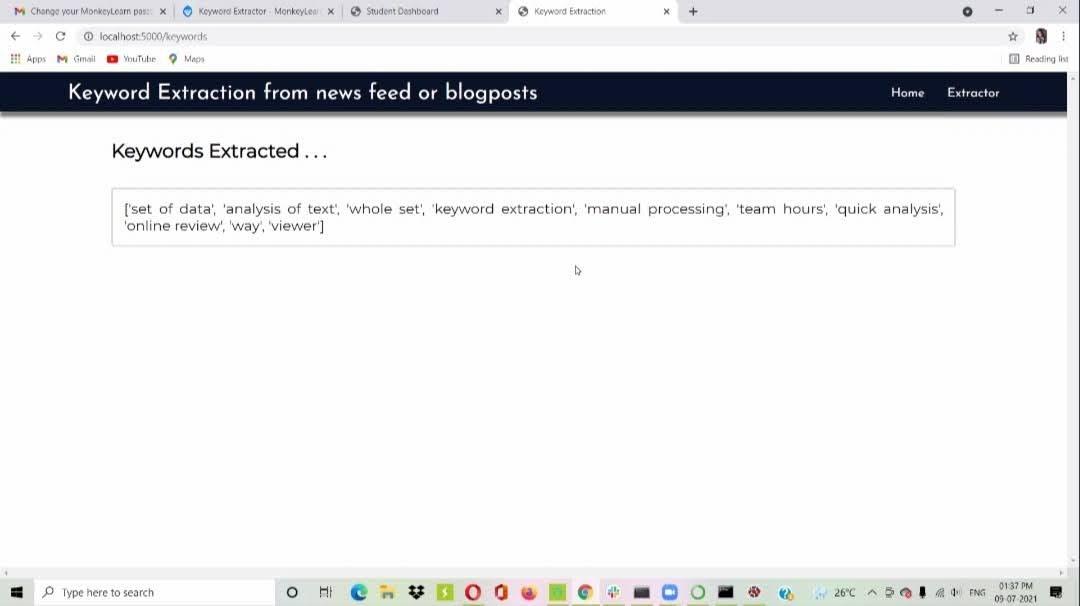
>Keywords extraction from news feed or blogposts it asks to newsfeed/blog you want to extract Keywords from…

>Then select an option and give some newsfeed or blog to it. Then it extracts the Keywords.





>Here keywords are extracted.



Advantages and Disadvantages:

**Advantages of keyword Extraction:**

* Scalability. Automated **keyword extraction** allows you to analyze as much data as you want. ...
* Consistent criteria. ...
* Real-time analysis. ...
* Word Frequency. ...
* Word Collocations and Co-occurrences. ...
* TF-IDF. ...
* RAKE. ...
* Conditional Random Fields.

Disadvantages of Keyword Extraction:

Because the **keyword** not only determine by frequency but also by it's mean. In addition this algorithm makes compound word get higher weight so they are nor accurate enough. 2.2. **Keyword extraction** for text characterization We use quadgram-based to **extraction keyword**.

Applications:

* [Social media monitoring](https://monkeylearn.com/keyword-extraction/#social-media-monitoring)
* [Brand monitoring](https://monkeylearn.com/keyword-extraction/#brand-monitoring)
* [Customer service](https://monkeylearn.com/keyword-extraction/#customer-service)
* [Customer feedback](https://monkeylearn.com/keyword-extraction/#customer-feedback)
* [Business intelligence](https://monkeylearn.com/keyword-extraction/#business-intelligence)
* [Search engine optimization (SEO)](https://monkeylearn.com/keyword-extraction/#seo)
* [Product analytics](https://monkeylearn.com/keyword-extraction/#product-analytics)
* [Knowledge management](https://monkeylearn.com/keyword-extraction/#knowledge-management).

9.Conclusion:

Which indicated how well a blog reflected the web context. If a blog includes rich content on a topic word, the word can be considered as a keyword of the blog. Since the web includes various and trendy information a topic, we could choose more popular and trendy keywords than other methods by measuring the richness of blogs,based on the web context. The trendy topic keywords would help the readers to understand the content of a blog more easily.we can conclude that any newsfeed or blogposts can be read simply as Keywords.

10.Future Scope:

**Future Scope** we can conclude that different methods and techniques could be used to **extract** **keywords**.Different approaches such as supervised and unsupervised are used in different scenarios.By **keyword** **extraction** we could

11.Bibilography:

A new **keyword** **extraction** algorithm for news web pages which is bases on Keyword extraction… Experiments on randomly selected web pages have been performed to demonstrate the quality of the **keywords** extracted. Keywords are Extracted.

12.Appendix:

a.Source Code:

from monkeylearn import MonkeyLearn

from flask import Flask, request, render\_template

import re

import requests

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

#home page

@app.route('/')

def home():

return render\_template('home.html')

#home page

@app.route('/extractor')

def extractor():

return render\_template('extractor.html')

#extractor page

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

def keywords():

ml = MonkeyLearn('41f9624c1a130449bdd3d28a1d70c7447dd45a12')

typ=request.form['type']

model\_id = 'ex\_YCya9nrn'

print(type)

output = request.form['output']

print(output)

print(type(output))

if typ=="text":

output=re.sub("[^a-zA-Z.,]"," ",output)

print(output)

keywordresult = ml.extractors.extract(model\_id, [output])

print(keywordresult.body)

a = keywordresult.body[0]

print (a)

print(type(a))

b = a['extractions']

print(b)

print(type(b))

keywords = []

entities = []

for i in b:

keywords.append(i['parsed\_value'])

print (i['parsed\_value'])

print(keywords)

#return keywords

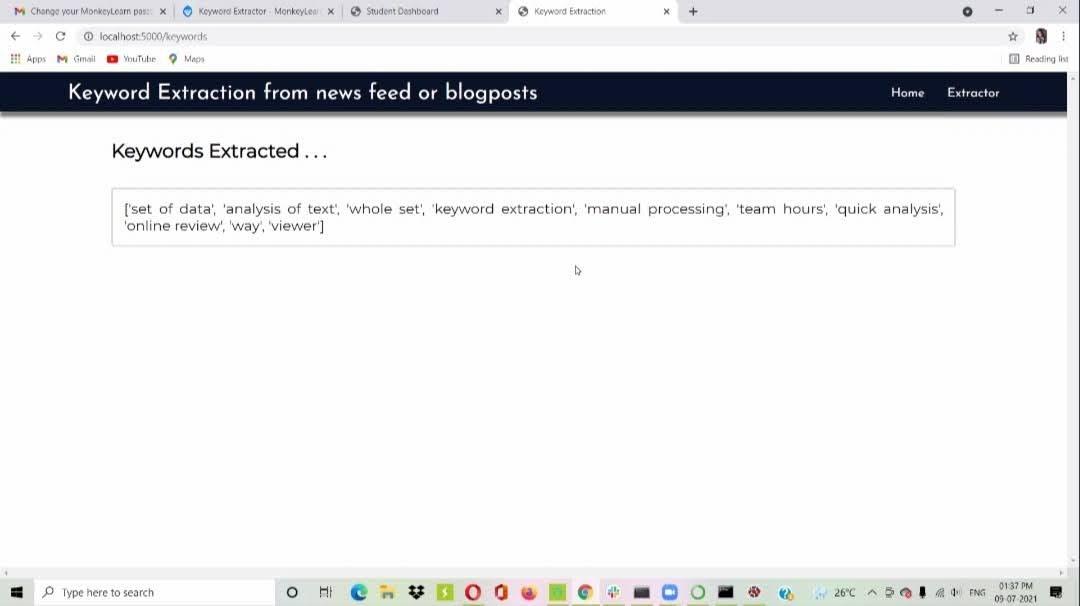
return render\_template('keywords.html',keyword =keywords)

#return render\_template('extractor.html',keyword = keywords)

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

app.run(debug=False)

b.UI Output Screenshot:



sion:

## 