

Jyosthna Gandhodi
801254449

Final Project Documentation

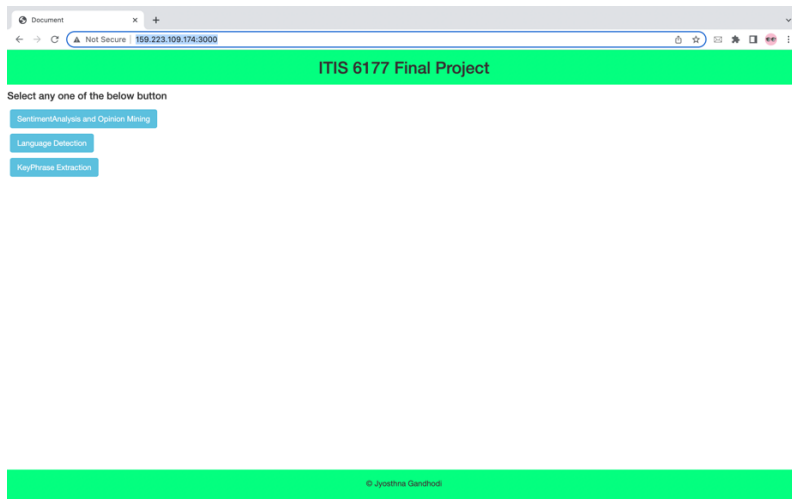
API Provided : <https://azure.microsoft.com/en-us/services/cognitive-services/language-service/>

Using this API I have created endpoints to perform Sentiment Analysis, Language Detection and Key Phrase extraction.

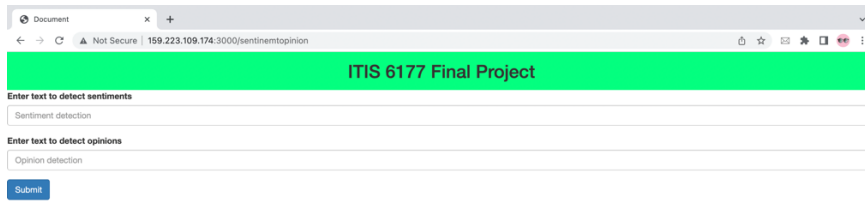
1.Sentiment Analysis : Sentiment analysis, often known as opinion mining, is a type of natural language processing (NLP) technique for determining the emotional tone of a body of text.

Testing via UI :

Step 1 : On the browser go to the website <http://159.223.109.174:3000/> which opens the home page.



Step 2 : Select the button “Sentiment Analysis and Opinion Mining” which redirects to <http://159.223.109.174:3000/sentinemtopinion>



Step 3 : Under Enter text to detect sentiments and Enter text to detect opinions please give any text input.

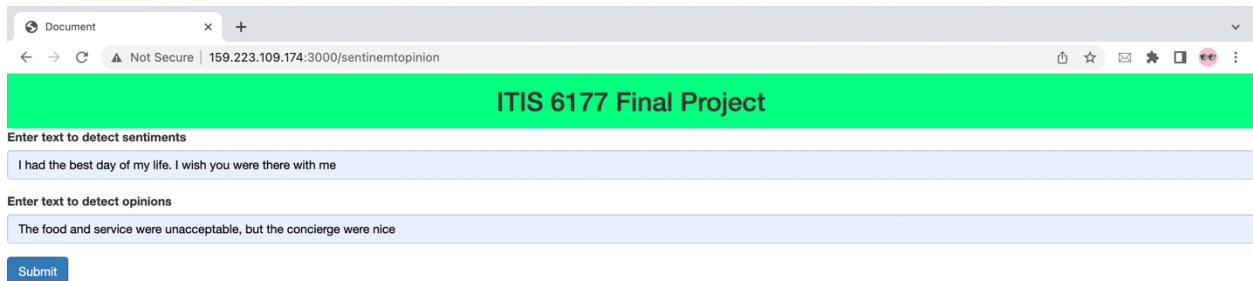
Sample input :

Enter text to detect sentiments :

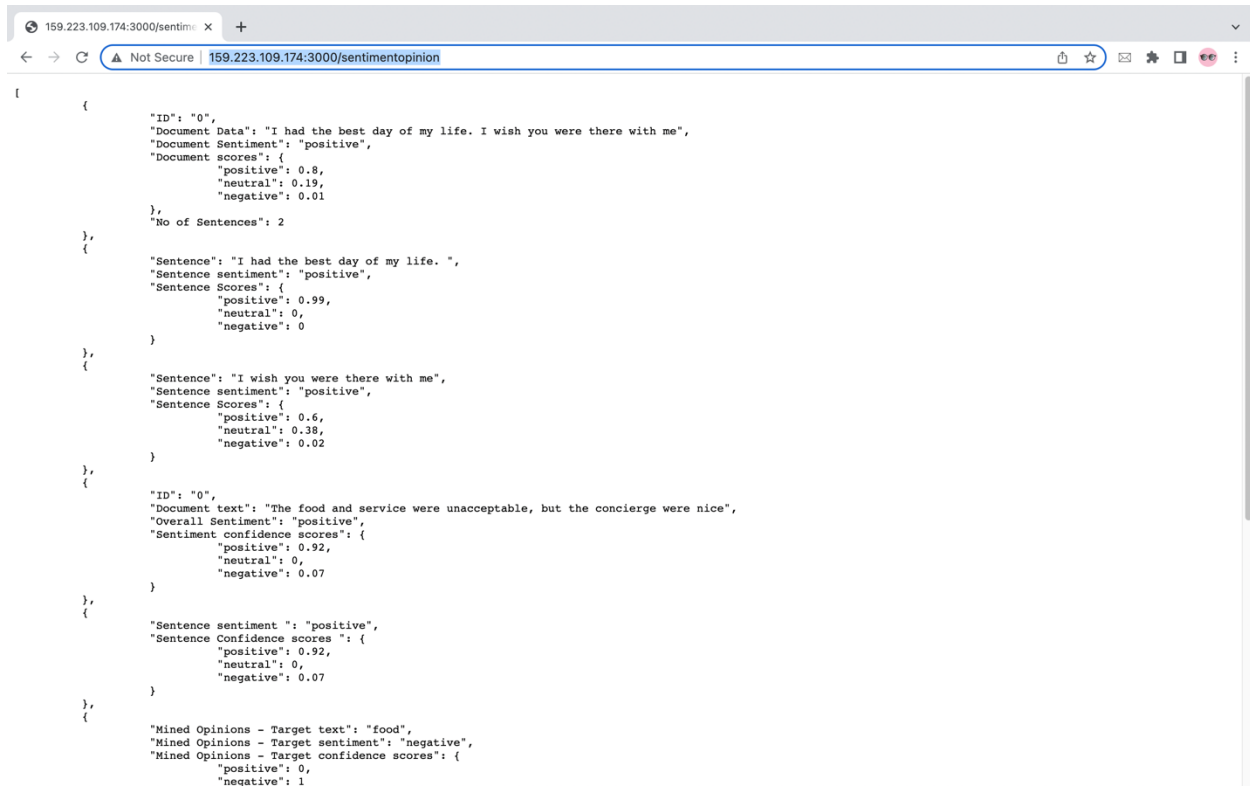
I had the best day of my life. I wish you were there with me

Enter text to detect opinions :

The food and service were unacceptable, but the concierge were nice.



Step 4: Press Submit which redirects to <http://159.223.109.174:3000/sentimentopinion> that contains JSON output of document Sentiment and its scores



The screenshot shows a web browser window with the address bar displaying `159.223.109.174:3000/sentimentopinion`. The page content is a JSON object representing sentiment analysis results for three different documents. The first document is a positive statement about a life event, the second is another positive statement, and the third is a mixed statement about food and service. Each document entry includes an ID, the original text, an overall sentiment label, and a set of confidence scores for positive, neutral, and negative sentiments. The third document also includes mined opinions for the target 'food'.

```
[
  {
    "ID": "0",
    "Document Data": "I had the best day of my life. I wish you were there with me",
    "Document Sentiment": "positive",
    "Document scores": {
      "positive": 0.8,
      "neutral": 0.19,
      "negative": 0.01
    },
    "No of Sentences": 2
  },
  {
    "Sentence": "I had the best day of my life. ",
    "Sentence sentiment": "positive",
    "Sentence Scores": {
      "positive": 0.99,
      "neutral": 0,
      "negative": 0
    }
  },
  {
    "Sentence": "I wish you were there with me",
    "Sentence sentiment": "positive",
    "Sentence Scores": {
      "positive": 0.6,
      "neutral": 0.38,
      "negative": 0.02
    }
  },
  {
    "ID": "0",
    "Document text": "The food and service were unacceptable, but the concierge were nice",
    "Overall Sentiment": "positive",
    "Sentiment confidence scores": {
      "positive": 0.92,
      "neutral": 0,
      "negative": 0.07
    }
  },
  {
    "Sentence sentiment ": "positive",
    "Sentence Confidence scores ": {
      "positive": 0.92,
      "neutral": 0,
      "negative": 0.07
    }
  },
  {
    "Mined Opinions - Target text": "food",
    "Mined Opinions - Target sentiment": "negative",
    "Mined Opinions - Target confidence scores": {
      "positive": 0,
      "negative": 1
    }
  }
]
```

```
159.223.109.174:3000/sentiment X +
Not Secure | 159.223.109.174:3000/sentimentopinion
{
  "ID": "0",
  "Document text": "The food and service were unacceptable, but the concierge were nice",
  "Overall Sentiment": "positive",
  "Sentiment confidence scores": {
    "positive": 0.92,
    "neutral": 0,
    "negative": 0.07
  },
  {
    "Sentence sentiment ": "positive",
    "Sentence Confidence scores ": {
      "positive": 0.92,
      "neutral": 0,
      "negative": 0.07
    }
  },
  {
    "Mined Opinions - Target text": "food",
    "Mined Opinions - Target sentiment": "negative",
    "Mined Opinions - Target confidence scores": {
      "positive": 0,
      "negative": 1
    }
  },
  {
    "Target assesments - Text": "unacceptable",
    "Target assesments - Sentiment": "negative"
  },
  {
    "Mined Opinions - Target text": "service",
    "Mined Opinions - Target sentiment": "negative",
    "Mined Opinions - Target confidence scores": {
      "positive": 0,
      "negative": 1
    }
  },
  {
    "Target assesments - Text": "unacceptable",
    "Target assesments - Sentiment": "negative"
  },
  {
    "Mined Opinions - Target text": "concierge",
    "Mined Opinions - Target sentiment": "positive",
    "Mined Opinions - Target confidence scores": {
      "positive": 1,
      "negative": 0
    }
  },
  {
    "Target assesments - Text": "nice",
    "Target assesments - Sentiment": "positive"
  }
}
```

Testing via POSTMAN :

1. Give URL: <http://159.223.109.174:3000/sentimentopinion> ,
2. Select POST request and under body select **x-www-form-urlencoded** and give KEY as “analyzeSenti” and VALUE as “I had the best day of my life. I wish you were there with me “
3. Click send and the response redirected will be JSON output that contains document Sentiment and its scores and the mined opinions focused on targeted words.

HomeWorkspacesAPI NetworkReportsExplore

Search Postman

Invite

Upgrade

My Workspace

NewImport

OverviewGET http://locPOST http://loPOST http://loPOST http://loPOST http://loPOST http://loPOST http://lo

No Environment

Collections

My first collection

First folder inside collection

Second folder inside collection

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create collection

Start working with APIs

67%

Next: Save a request. Show me

http://159.223.109.174:3000/sentimentopinion

Save

Send

POSThttp://159.223.109.174:3000/sentimentopinion

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> analyzeSenti	I had the best day of my life. I wish you were there with me		
<input checked="" type="checkbox"/> analyzeOpinion	The food and service were unacceptable, but the concierge were nice		

BodyCookiesHeaders (6)Test Results

Status: 200 OKTime: 202 msSize: 3.43 KBSave Response

PrettyRawPreviewVisualizeText

```
1 {
2   "ID": "0",
3   "Document Data": "I had the best day of my life. I wish you were there with me",
4   "Document Sentiment": "positive",
5   "Document scores": {
6     "positive": 0.8,
7     "neutral": 0.19,
8     "negative": 0.01
9   },
10  "No of Sentences": 2
11 },
12 {
13   "Sentence": "I had the best day of my life. ",
14   "Sentence sentiment": "positive",
15   "Sentence Scores": {
16     "positive": 0.99,
17     "neutral": 0,
18     "negative": 0
19   }
20 },
21 {
22   "Sentence": "I wish you were there with me",
```

Find and Replace

Console

CookiesCapture requestsBootcampRunnerTrash

HomeWorkspacesAPI NetworkReportsExplore

Search Postman

Invite

Upgrade

My Workspace

NewImport

OverviewGET http://loPOST http://loPOST http://loPOST http://loPOST http://loPOST http://loPOST http://loPOST http://lo

No Environment

Collections

My first collection

First folder inside collection

Second folder inside collection

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create collection

Start working with APIs

67%

Next: Save a request. Show me

http://159.223.109.174:3000/sentimentopinion

Save

Send

POSThttp://159.223.109.174:3000/sentimentopinion

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> analyzeSenti	I had the best day of my life. I wish you were there with me	
<input checked="" type="checkbox"/> analyzeOpinion	The food and service were unacceptable, but the concierge were nice	

BodyCookiesHeaders (6)Test Results

Status: 200 OKTime: 202 msSize: 3.43 KBSave Response

PrettyRawPreviewVisualizeText

```
21 },
22 {
23   "Sentence": "I wish you were there with me",
24   "Sentence sentiment": "positive",
25   "Sentence Scores": {
26     "positive": 0.6,
27     "neutral": 0.38,
28     "negative": 0.02
29   }
30 },
31 {
32   "ID": "0",
33   "Document text": "The food and service were unacceptable, but the concierge were nice",
34   "Overall Sentiment": "positive",
35   "Sentiment confidence scores": {
36     "positive": 0.92,
37     "neutral": 0,
38     "negative": 0.07
39   }
40 },
41 {
42   "Sentence sentiment ": "positive",
43   "Sentence Confidence scores ": {
```

Find and Replace

Console

CookiesCapture requestsBootcampRunnerTrash

Output Explanation :

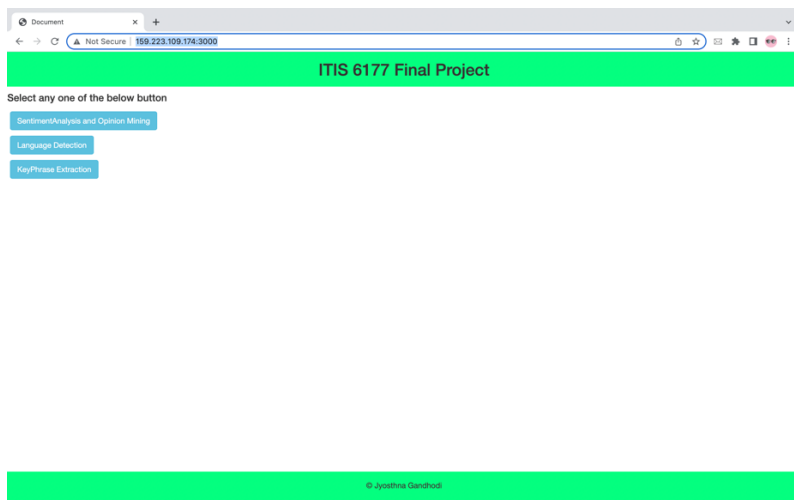
The response output displays the emotional tone of the text provided.

For instance for the text input “**I had the best day of my life. I wish you were there with me**” it displays the document Sentiment and its scores and for the text input “**The food and service were unacceptable, but the concierge were nice**” the output displays Sentiment and its scores and also the mined opinions focused on targeted words as shown in the above output images.

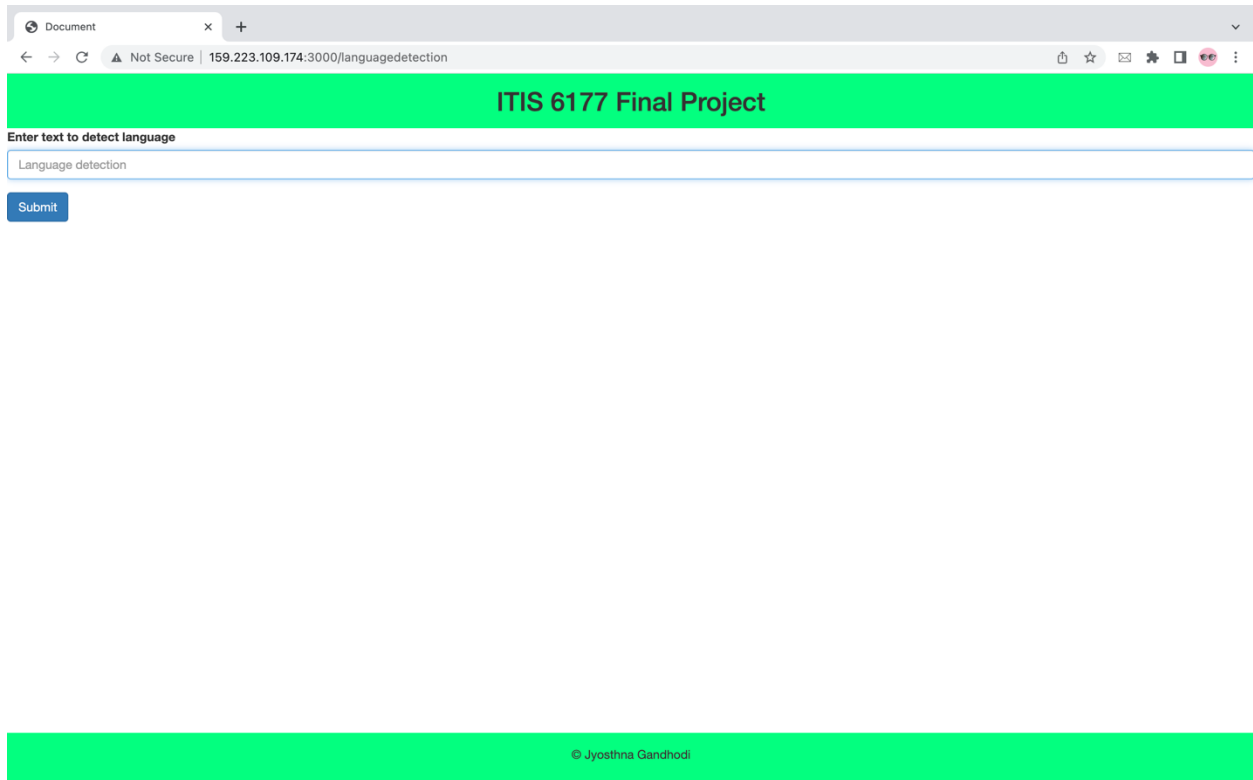
2. Language Detection: The language detection function detects which natural language the text is written in.

Testing via UI :

Step 1: On the browser go to the website <http://159.223.109.174:3000/> which opens the home page.



Step 2 : Select the button “**Language detection**” which redirects to <http://159.223.109.174:3000/language-detection>



Step 3 : Under Enter text to detect language, enter any sample language input

Sample input :

Enter text to detect language

Ce document est rédigé en Français.

Document x +

← → ↻ Not Secure | 159.223.109.174:3000/ languagedetection

ITIS 6177 Final Project

Enter text to detect language

Ce document est rédigé en Français.

Submit

© Jyosthna Gandhodi

Step 4 : Press submit which redirects to <http://159.223.109.174:3000/language>

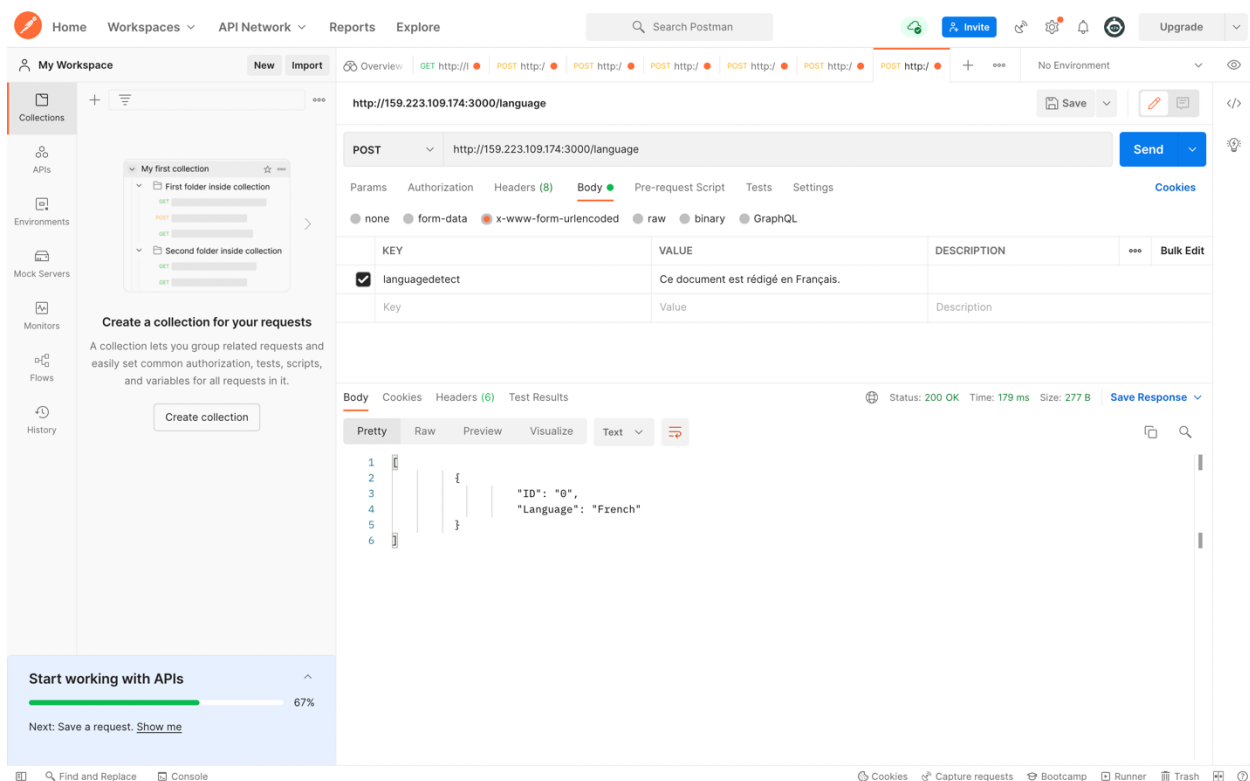
159.223.109.174:3000/language x +

← → ↻ Not Secure | 159.223.109.174:3000/language

```
[
  {
    "ID": "0",
    "Language": "French"
  }
]
```


Testing via POSTMAN :

1. Give URL : <http://159.223.109.174:3000/language>
2. Select POST request and under body select **x-www-form-urlencoded** and give KEY as “**languagedetect**” and VALUE as “**Ce document est rédigé en Français**”.
3. Click send and the response redirected will be JSON output that contains output of the language for the input given.



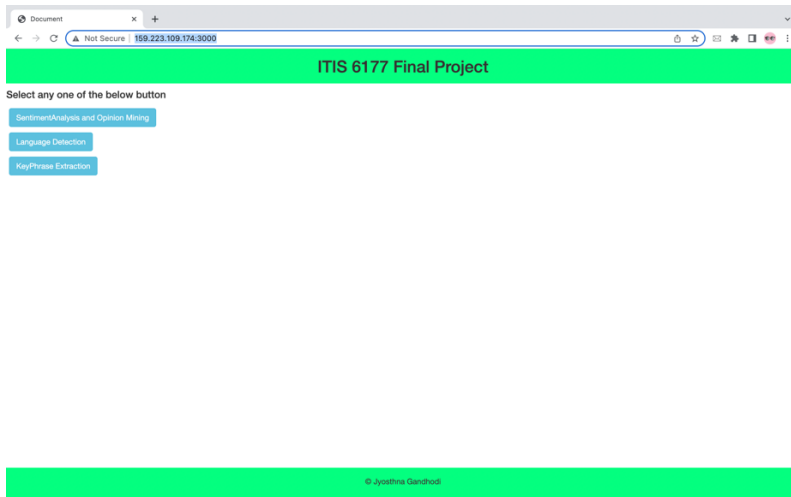
Output Explanation : For the input “**Ce document est rédigé en Français.**”, the response of the output is the language of the input given .

3.Keyphrase Extraction :

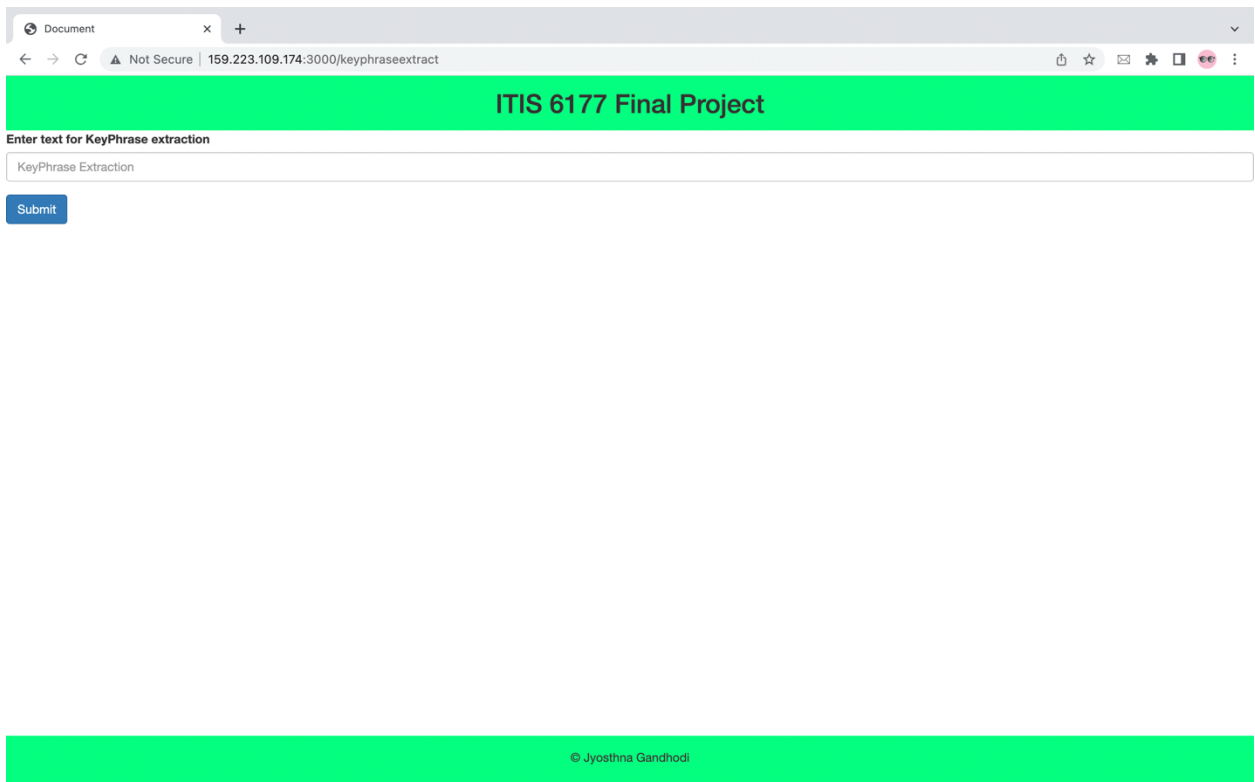
Keyphrase extraction is to automatically identify phrases that best describe the content of a text.

Testing via UI :

Step 1 : On the browser go to the website <http://159.223.109.174:3000/> which opens the home page.



Step 2 : Select the button “**KeyPhrase Extraction**” which displays a form to enter the KeyPhrase



Step 3 : Under “**Enter text for KeyPhrase extraction**”, enter any text input

Sample input for Enter text for KeyPhrase extraction :
“My cat might need to see a veterinarian.”

Document x +

← → ↻ ⚠ Not Secure | 159.223.109.174:3000/keyphraseextract

📄 ☆ 📧 ⚙ 🖨 👤 ⋮

ITIS 6177 Final Project

Enter text for KeyPhrase extraction

My cat might need to see a veterinarian.

Submit

© Jyosthna Gandhodi

Step 4 : Press submit which redirects to <http://159.223.109.174:3000/keyphraseextraction>

159.223.109.174:3000/keyphrase x +

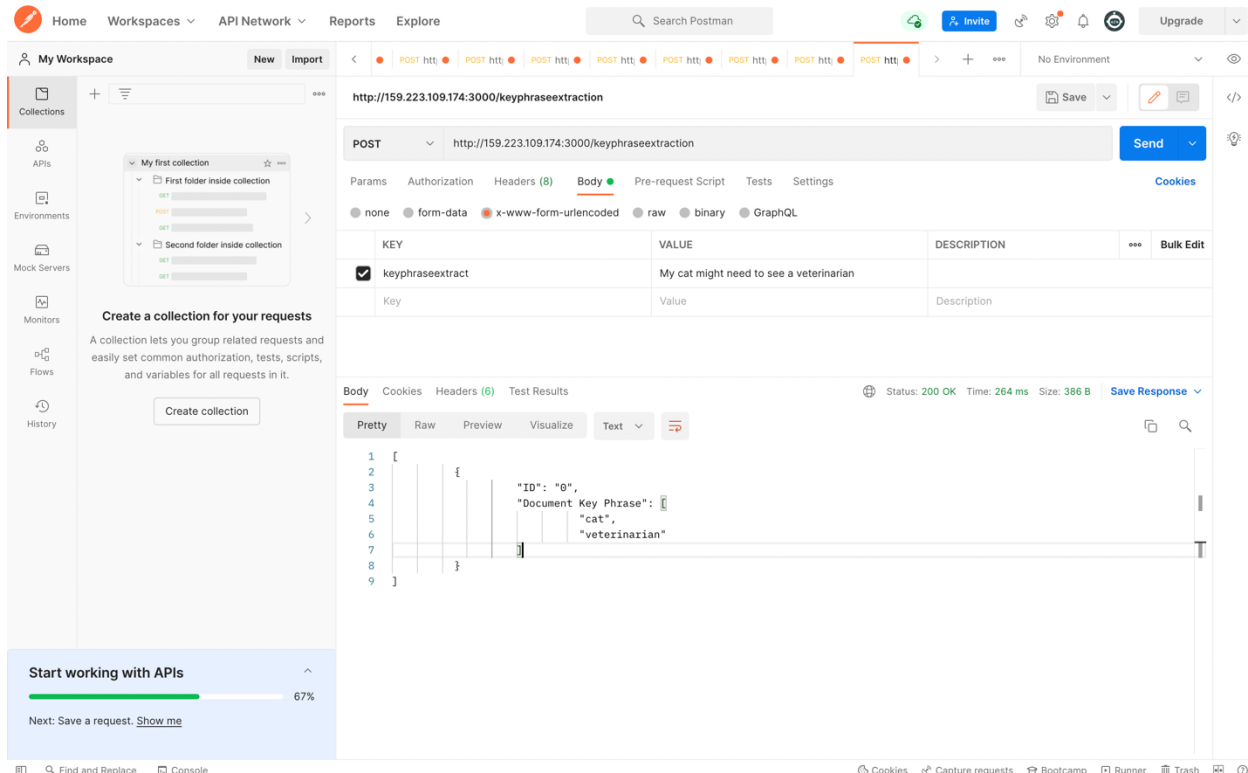
← → ↻ ⚠ Not Secure | 159.223.109.174:3000/keyphraseextraction

📄 ☆ 📧 ⚙ 🖨 👤 ⋮

```
{
  {
    "ID": "0",
    "Document Key Phrase": [
      "cat",
      "veterinarian"
    ]
  }
}
```

Testing via POSTMAN :

- 1 . Give URL as <http://159.223.109.174:3000/keyphraseextraction>
2. Select POST request and under body select **x-www-form-urlencoded** and give KEY as “**keyphraseextract**” and value as “**My cat might need to see a veterinarian**”
3. Click send and the response redirected will be JSON output that contains key phrases for the input text given.



Output Explanation: For the text input given “My cat might need to see a veterinarian” the response of the output displays the keyphrases of the text “cat, veterinarian”.

Tools, Languages and Frameworks Used

node.js - For implementing API

npm - Package manager for Javascript Programming language.

express - For building web applications and API.

azure/ai-text-analytics - Microsoft Azure API used.

body-parser - Used for processing data sent through an HTTP request.

cors - allows you to make requests from one website to another website in the browser.

dotenv - automatically loads environment variables.

ejs - Used for Generating web pages.

digital Ocean - For deploying the project.

References : <https://azure.microsoft.com/en-us/services/cognitive-services/language-service/>