

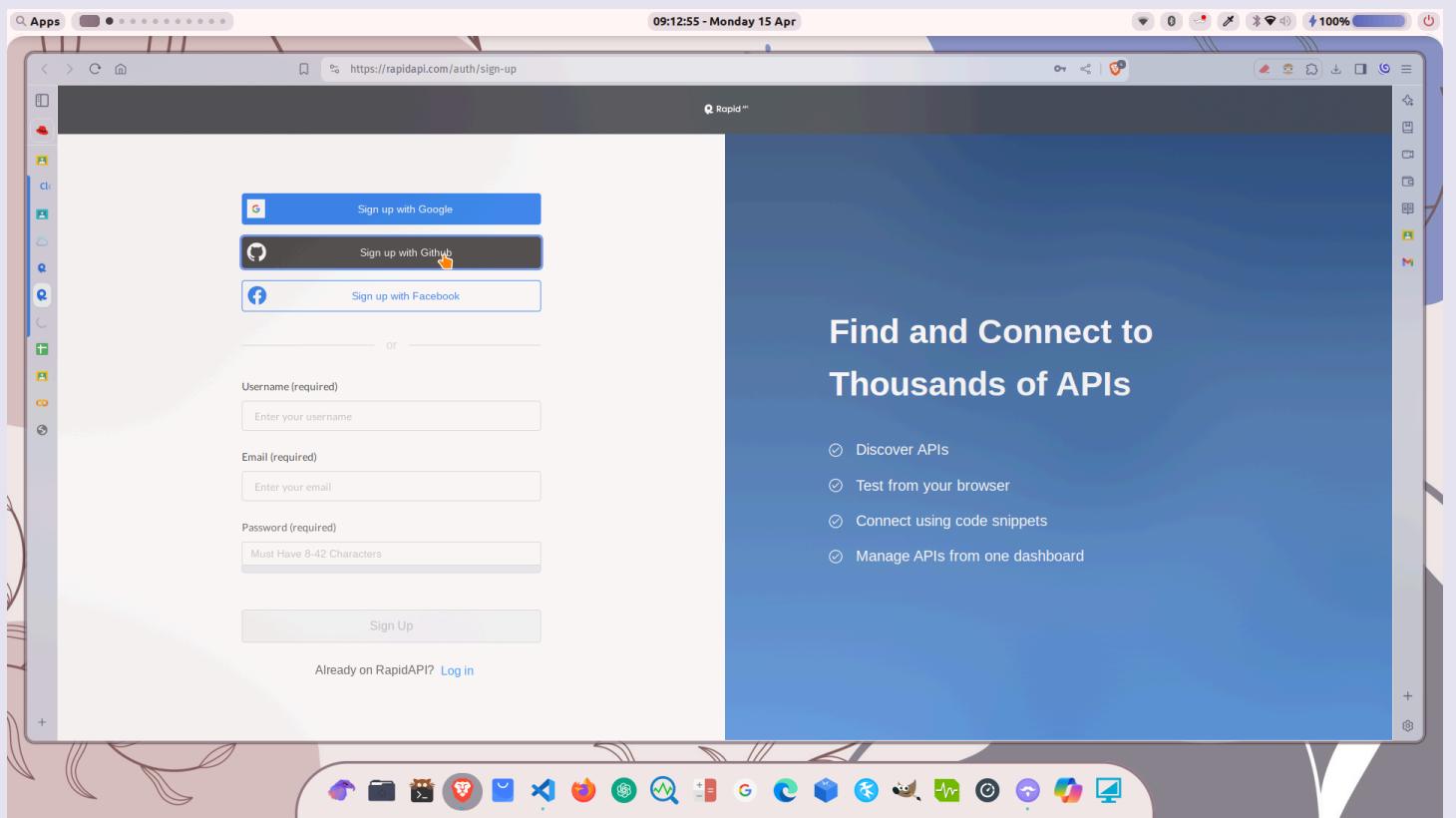
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

**Aim :** Using IBM Cloud, develop an application to implement a Nodejs Application and enrich your Application with Cloud Services

- 1. Create an Application to handle data from 3rd party API(here IMDB API).**
- 2. Integrate NLU into your application to perform sentiment analysis on the feed captured from IMDB.**
- 3. Integrate database service to store captured feeds and their results.**

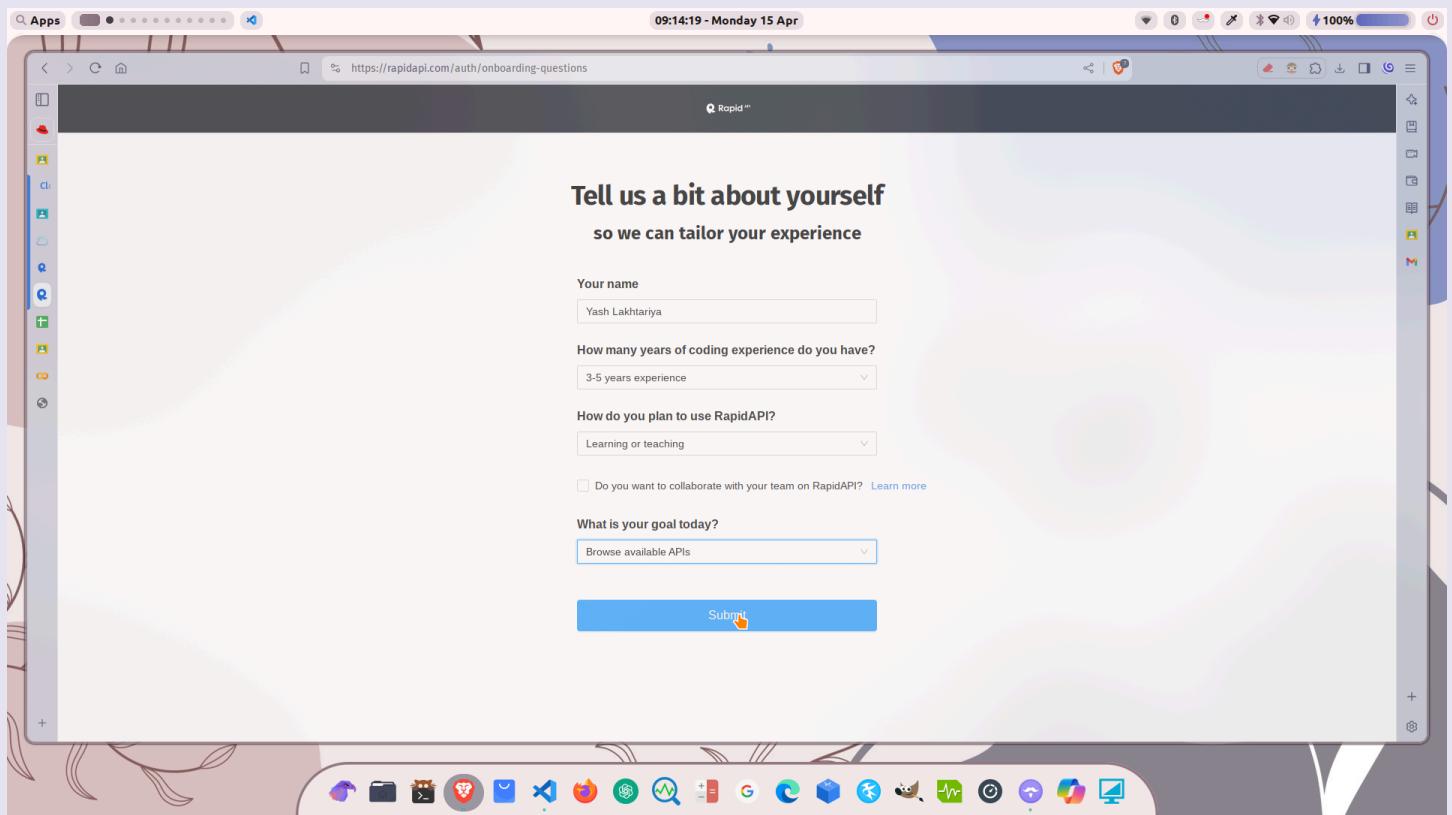
**Steps and screenshots :**

1. Signup with Github or preferred option if account doesn't exists on RapidAPI



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 14**

## 2. Finish creating the account adding necessary details



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

3. Test **Urban Dictionary API (by community)** and click Test Endpoint after entering the term to search in dictionary (here devotion) and the results in JSON format response will be displayed

The screenshot shows the RapidAPI interface for the Urban Dictionary API. The top navigation bar includes 'API Hub', 'Organizations', 'Apps', 'My APIs', and a user profile icon. The main page displays the 'Urban Dictionary' API by community, updated 4 days ago. It shows a popularity score of 9.9 / 10, latency of 84ms, and service level of 100%. A 'Test Endpoint' button is visible next to the API name.

The 'Endpoints' section lists the 'GET Define' endpoint. The 'Header Parameters' section shows 'X-RapidAPI-Key' as an ENUM required parameter with the value 'd971976965msh86cc63f664a1ca02p10db23jsne5dd51351a67'. The 'Required Parameters' section shows 'term' as a STRING required parameter with the value 'devotion'.

The 'Results' tab shows a successful JSON response for the term 'devotion'. The response includes:

```
200 Success
{
  "0": {
    "definition": "[Loyalty] [motivated] by [love].",
    "permalink": "http://devotion.urbanup.com/686694",
    "thumbs_up": 134,
    "author": "Downstrike",
    "word": "devotion",
    "defid": 686694,
    "current_vote": "",
    "written_on": "2004-05-25T02:48:45.000Z",
    "example": "[Devotion] is an [important] part of a happy [relationship].",
    "thumbs_down": 33
  },
  "1": {
    "definition": "Loyalty, love, deep [commitment]. << and will if you're reading this, I am absolutely [devoted] to you [with all my heart] and soul ::)",
    "permalink": "http://devotion.urbanup.com/686695"
  }
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

4. From Code Snippets, copy the RapidAPI key and paste in the below code to check

Code :

```
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
  params: {
    term: 'devotion'
  },
  headers: {
    'X-RapidAPI-Key': 'd971976965msh86c63f664a1ca02p10db23jsne5dd51351a67',
    'X-RapidAPI-Host': 'mashape-community-urban-dictionary.p.rapidapi.com'
  }
};

async function getData() {
  try {
    const response = await axios.request(options);
    console.log('\n\t Definition : ' +
response.data.list[0].definition);
    console.log('\n\t Example : ' + response.data.list[0].example);
  } catch (error) {
    console.error(error);
  }
}

getData();
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

→ Here definition and example of 1st result is displayed in console

The screenshot shows the RapidAPI website interface for the Urban Dictionary API. At the top, the URL is https://rapidapi.com/community/api/urban-dictionary. The page displays the API's popularity (9.9 / 10), latency (84ms), service level (100%), and health check status (N/A). Below this, the 'Endpoints' tab is selected, showing the 'GET Define' endpoint. The endpoint details include 'Header Parameters' (X-RapidAPI-Key, X-RapidAPI-Host) and 'Required Parameters' (term). On the right side, there is a 'Code Snippets' section with Node.js Axios code examples. The code uses axios to make a GET request to the endpoint with the provided parameters and headers. The response is logged to the console.

```
(Node.js) Axios
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
  params: {term: 'devotion'},
  headers: {
    'X-RapidAPI-Key': 'd97197695msh86c63f664a1ca02p10db23jne5dd51351a67',
    'X-RapidAPI-Host': 'mashape-community-urban-dictionary.p.rapidapi.com'
  }
};

try {
  const response = await axios.request(options);
  console.log(response.data);
} catch (error) {
  console.error(error);
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

5. Run the code and check output for list[1] and list[0] to check for both 1st and 2nd results in dictionary

The screenshot shows a Mac OS X desktop environment. In the top right corner, the system status bar displays the date and time as "09:24:09 - Monday 15 Apr" and battery level at "100%". The main window is a terminal application with the title bar "09:24:09 - Monday 15 Apr". The terminal content is as follows:

```
yash ~/Practical_14 $ main ? @ v18.18.2 09:21 node "/home/yash/Documents/sem6practicals/EADC/Practical_14/imdb.js"
Definition : Loyalty, love, deep [commitment]. << and Will if you're reading this, I am absolutely [devoted] to you [with all my heart] and soul :)
Example : the [amount] of devotion I have for him is [amazing] <3

yash ~/Practical_14 $ main ? @ v18.18.2 09:21 node "/home/yash/Documents/sem6practicals/EADC/Practical_14/imdb.js"
Definition : [Loyalty] [motivated] by [love].
Example : [Devotion] is an [important] part of a happy [relationship].
```

Below the terminal, the Dock at the bottom of the screen shows various application icons.

The code editor window, titled "imdb.js", contains the following JavaScript code:

```
const options = {
  headers: {
    'X-RapidAPI-Key': 'd971976965msh86c63f664a1ca02p10db23jsne5dd51351a67',
    'X-RapidAPI-Host': 'mashape-community-urban-dictionary.p.rapidapi.com'
  }
}; <- #4-14 const options =
15
16
17 async function getData(){
18   try {
19     const response = await axios.request(options);
20     console.log(`\n\t Definition : ${response.data.list[0].definition}`);
21     console.log(`\n\t Example : ${response.data.list[0].example}`);
22   } catch (error) {
23     console.error(error);
24   }
25 } <- #17-25 async function getData()
26
27 getData();
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

## 6. Now, to analyze that text, create NLU service on IBM cloud

The screenshot shows the IBM Cloud Catalog interface. The URL in the address bar is <https://cloud.ibm.com/catalog/services/natural-language-understanding?planId=e45ae97c-4140-4a9b-8db9-c419169e9be5>. The page title is "Natural Language Understanding". The summary on the right indicates a "Natural Language Understanding" service is being created, located in London, Plan: Lite, Service name: Natural Language Understanding-YSL, and Resource group: Default. The "Create" button is highlighted.

**Natural Language Understanding**

Analyze text to extract meta-data from content such as concepts, entities, emotion, relations, sentiment and more.

**Create**      **About**

Select a location

London (eu-gb)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

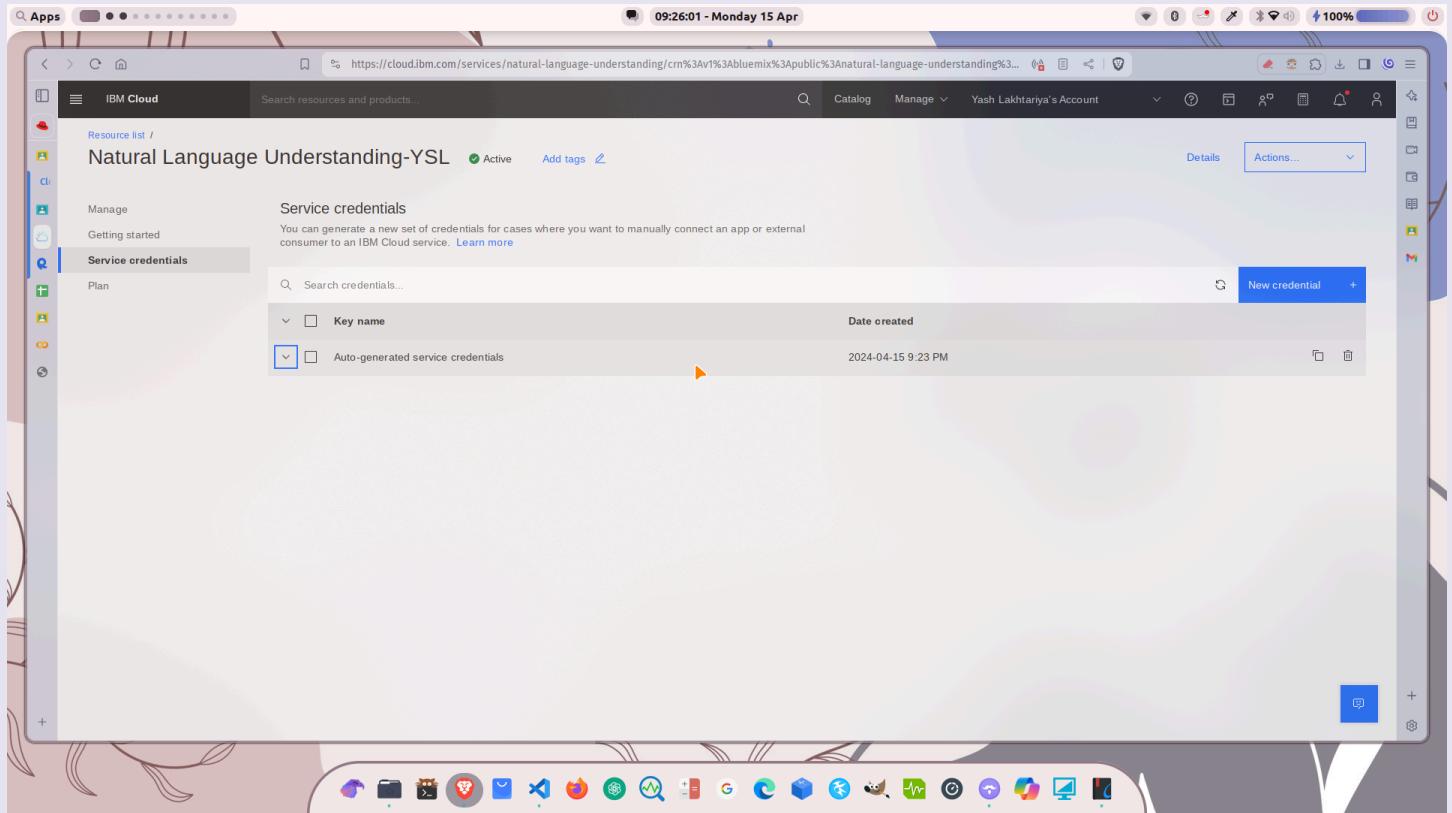
Plan	Features and capabilities	Pricing
Lite	<b>30,000 NLU Items Per Month</b> 1 Custom Model Fixed API Rate Limit. See Standard plan for higher API Rate Limit NOTE: A NLU item is based on the number of data units enriched and the number of enrichment features applied. A data unit is 10,000 characters or less. For example: extracting Entities and Sentiment from 15,000 characters of text is (2 Data Units * 2 Enrichment Features) = 4 NLU Items. A custom model refers to an annotation model developed with Watson Knowledge Studio.	Free
Standard	<b>Unlimited NLU Items Per Month</b> You will be charged per NLU Item You will be charged per Custom Model	<a href="#">Click to view tiers and pricing detail</a>

I have read and agree to the following license agreements:  
[Terms](#)

**Create**

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 14**

## 7. There are already auto generated credentials present there



The screenshot shows the IBM Cloud interface for managing service credentials. The URL in the browser is https://cloud.ibm.com/services/natural-language-understanding/crn%3Av1%3Abuemix%3Apublic%3Anatural-language-understanding%3... . The page title is "Natural Language Understanding-YSL" and it shows "1 Active" credential. The "Service credentials" section contains a table with one row:

Key name	Date created
Auto-generated service credentials	2024-04-15 9:23 PM

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

## 8. Copy the credentials and using CURL request, analyze the webpage's text

```
curl -X POST -u "apikey:525_IRJ1zd1sGmM4Ga-yGEekmqUeLw5NPg-HLHVT24_I" --header "Content-Type: application/json" --data "{\\"url\\": \"https://newsroom.ibm.com/2024-04-09-IBM-to-Help-Canadian-Enterprises-Leverage-Generative-AI-and-Address-their-Data-Sovereignty-Requirements\", \"features\": {\"sentiment\": {}, \"categories\": {}, \"concepts\": {}, \"entities\": {}, \"keywords\": {}}, \"text\": \"IBM has announced a new Cloud Multizone Region, which will enable Canadian enterprises to leverage Generative AI and address their Data Sovereignty Requirements. The new region will provide businesses with access to a range of cloud services, including Watson Natural Language Understanding, which can help them analyze and understand their data more effectively. The announcement comes as IBM continues to invest in its Canadian operations, and as the company looks to expand its presence in the country. The new region will be available to Canadian businesses starting in Q3 2024.\", \"language\": \"en\", \"keywords\": [{\"text\": \"opening of IBM Cloud\", \"relevance\": 0.628346, \"count\": 1}, {\"text\": \"new Cloud Multizone Region\", \"relevance\": 0.617738, \"count\": 1}]}"
```

"https://api.eu-gb.natural-language-understanding.watson.cloud.ibm.com/instances/47885ddf-d3c1-4ca0-a438-57be5bb5e6f3/v1/analyze?version=2018-09-21"

Webpage URL :

<https://newsroom.ibm.com/2024-04-09-IBM-to-Help-Canadian-Enterprises-Leverage-Generative-AI-and-Address-their-Data-Sovereignty-Requirements>

```
curl -X POST -u "apikey:525_IRJ1zd1sGmM4Ga-yGEekmqUeLw5NPg-HLHVT24_I" --header "Content-Type: application/json" --data "{\\"url\\": \"https://newsroom.ibm.com/2024-04-09-IBM-to-Help-Canadian-Enterprises-Leverage-Generative-AI-and-Address-their-Data-Sovereignty-Requirements\", \"features\": {\"sentiment\": {}, \"categories\": {}, \"concepts\": {}, \"entities\": {}, \"keywords\": {}}, \"text\": \"IBM has announced a new Cloud Multizone Region, which will enable Canadian enterprises to leverage Generative AI and address their Data Sovereignty Requirements. The new region will provide businesses with access to a range of cloud services, including Watson Natural Language Understanding, which can help them analyze and understand their data more effectively. The announcement comes as IBM continues to invest in its Canadian operations, and as the company looks to expand its presence in the country. The new region will be available to Canadian businesses starting in Q3 2024.\", \"language\": \"en\", \"keywords\": [{\"text\": \"opening of IBM Cloud\", \"relevance\": 0.628346, \"count\": 1}, {\"text\": \"new Cloud Multizone Region\", \"relevance\": 0.617738, \"count\": 1}]}"
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

## 9. Redirect output to text file for storing it and as seen, the text file can be opened in any viewer (VS code here) to check output of curl request

The screenshot shows a Mac desktop with three main windows:

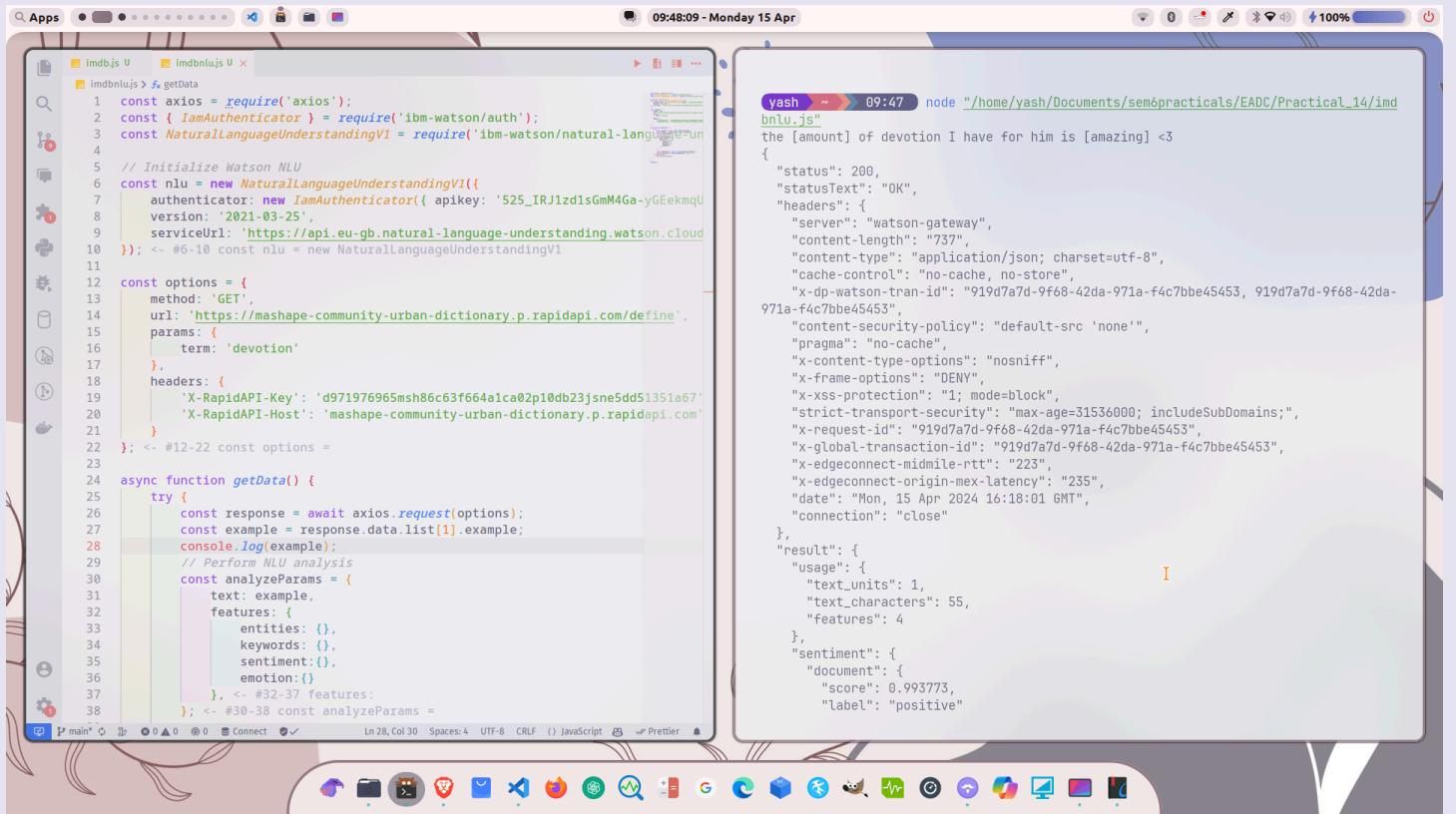
- Code Editor:** A window titled "eadc14.txt" containing JSON code. The code is a response from an IBM Watson API, detailing sentiment analysis and entity extraction for a given text. It includes sections for "usage", "sentiment", "document", "retrieved\_url", "language", "keywords", and "entities".
- Terminal 1:** A terminal window with the command: 

```
yash ~ 09:41 | curl -X POST -u "apikey:525_IRJ1zd1sGmM4Ga-yGEekmqUeLw5NPg-HLHVT24_1" --header "Content-Type: application/json" --data "{ \"url\": \"https://newsroom.ibm.com/2024-04-09-IBM-to-Help-Canadian-Enterprises-Leverage-Generative-AI-and-Address-their-Data-Sovereignty-Requirements\", \"features\": {\"sentiment\": {}, \"categories\": {}, \"concepts\": {}, \"entities\": {}, \"keywords\": {}}, \"url\": \"https://api.eu-gb.natural-language-understanding.watson.cloud.ibm.com/instances/47885ddf-d3c1-4ca0-a438-57be5bb5e6f3/v1/analyze?version=2018-09-21\" } >> eadc14.txt
```
- Terminal 2:** A terminal window showing the completion of the curl command with a progress bar and final statistics:

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
100	11734	100	11488	100	246	6225	133 0:00:01 0:00:01 --:--:-- 6356

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 14**

10. Now, in nodejs code, paste the URL, apikey credentials and analyse the text of example of dictionary (RapidAPI) through NLU service



```

const axios = require('axios');
const { IamAuthenticator } = require('ibm-watson/auth');
const NaturalLanguageUnderstandingV1 =
require('ibm-watson/natural-language-understanding/v1');

// Initialize Watson NLU
const nlu = new NaturalLanguageUnderstandingV1({
  authenticator: new IamAuthenticator({ apikey: '525_IRJ1zd1sGmM4Ga-yGEekmqUeLw5NPg-HLHVT24_I' }), // Replace
  'YOUR_API_KEY' with your actual API key
  version: '2021-03-25',
  serviceUrl: 'https://api.eu-gb.natural-language-understanding.watson.cloud.ibm.com'
}); <- #6-10 const nlu = new NaturalLanguageUnderstandingV1()

const options = {
  method: 'GET',
  url: 'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
  params: {
    term: 'devotion'
  },
  headers: {
    'X-RapidAPI-Key': 'd971976965msh86c63f664a1ca02p10db23jsne5dd51351a67',
    'X-RapidAPI-Host': 'mashape-community-urban-dictionary.p.rapidapi.com'
  }
}; <- #12-22 const options = 

async function getData() {
  try {
    const response = await axios.request(options);
    const example = response.data.list[1].example;
    console.log(example);
    // Perform NLU analysis
    const analyzeParams = {
      text: example,
      features: {
        entities: {},
        keywords: {},
        sentiment: {},
        emotion: {}
      }, <- #32-37 features:
    }; <- #30-38 const analyzeParams = 
  }
}

```

```

yash ~ 09:47 | node ./home/yash/Documents/sem6practicals/EADC/Practical_14/imdbnlu.js
the [amount] of devotion I have for him is [amazing] <3
{
  "status": 200,
  "statusText": "OK",
  "headers": {
    "server": "watson-gateway",
    "content-length": "737",
    "content-type": "application/json; charset=utf-8",
    "cache-control": "no-cache, no-store",
    "x-dp-watson-tran-id": "919d7a7d-9f68-42da-971a-f4c7bbe45453, 919d7a7d-9f68-42da-971a-f4c7bbe45453",
    "content-security-policy": "default-src 'none'",
    "pragma": "no-cache",
    "x-content-type-options": "nosniff",
    "x-frame-options": "DENY",
    "x-xss-protection": "1; mode=block",
    "strict-transport-security": "max-age=31536000; includeSubDomains",
    "x-request-id": "919d7a7d-9f68-42da-971a-f4c7bbe45453",
    "x-global-transaction-id": "919d7a7d-9f68-42da-971a-f4c7bbe45453",
    "x-edgeconnect-midmile-rtt": "223",
    "x-edgeconnect-origin-mex-latency": "235",
    "date": "Mon, 15 Apr 2024 16:18:01 GMT",
    "connection": "close"
  },
  "result": {
    "usage": {
      "text_units": 1,
      "text_characters": 55,
      "features": 4
    },
    "sentiment": {
      "document": {
        "score": 0.993773,
        "label": "positive"
      }
    }
  }
}

```

Code :

```

const axios = require('axios');
const { IamAuthenticator } = require('ibm-watson/auth');
const NaturalLanguageUnderstandingV1 =
require('ibm-watson/natural-language-understanding/v1');

// Initialize Watson NLU
const nlu = new NaturalLanguageUnderstandingV1({
  authenticator: new IamAuthenticator({ apikey: '525_IRJ1zd1sGmM4Ga-yGEekmqUeLw5NPg-HLHVT24_I' }), // Replace
  'YOUR_API_KEY' with your actual API key
  version: '2021-03-25',
  serviceUrl: 'https://api.eu-gb.natural-language-understanding.watson.cloud.ibm.com'
}); <- #6-10 const nlu = new NaturalLanguageUnderstandingV1()

const options = {
  method: 'GET',
  url: 'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
  params: {
    term: 'devotion'
  },
  headers: {
    'X-RapidAPI-Key': 'd971976965msh86c63f664a1ca02p10db23jsne5dd51351a67',
    'X-RapidAPI-Host': 'mashape-community-urban-dictionary.p.rapidapi.com'
  }
}; <- #12-22 const options = 

async function getData() {
  try {
    const response = await axios.request(options);
    const example = response.data.list[1].example;
    console.log(example);
    // Perform NLU analysis
    const analyzeParams = {
      text: example,
      features: {
        entities: {},
        keywords: {},
        sentiment: {},
        emotion: {}
      }, <- #32-37 features:
    }; <- #30-38 const analyzeParams = 
  }
}

```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
    serviceUrl:  
'https://api.eu-gb.natural-language-understanding.watson.cloud.ibm.com/instances/47885ddf-d3c1-4ca0-a438-57be5bb5e6f3' , // Replace  
'YOUR_INSTANCE_ID' with your actual instance ID  
});  
  
const options = {  
  method: 'GET',  
  url:  
'https://mashape-community-urban-dictionary.p.rapidapi.com/define',  
  params: {  
    term: 'devotion'  
  },  
  headers: {  
    'X-RapidAPI-Key':  
'd971976965msh86c63f664a1ca02p10db23jsne5dd51351a67',  
    'X-RapidAPI-Host':  
'mashape-community-urban-dictionary.p.rapidapi.com'  
  }  
};  
  
async function getData() {  
  try {  
    const response = await axios.request(options);  
    const example = response.data.list[1].example;  
    console.log(example);  
    // Perform NLU analysis  
    const analyzeParams = {  
      text: example,  
      features: {  
        entities: {},  
        keywords: {},  
        sentiment: {}  
      }  
    };  
  } catch (error) {  
    console.error(error);  
  }  
}
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

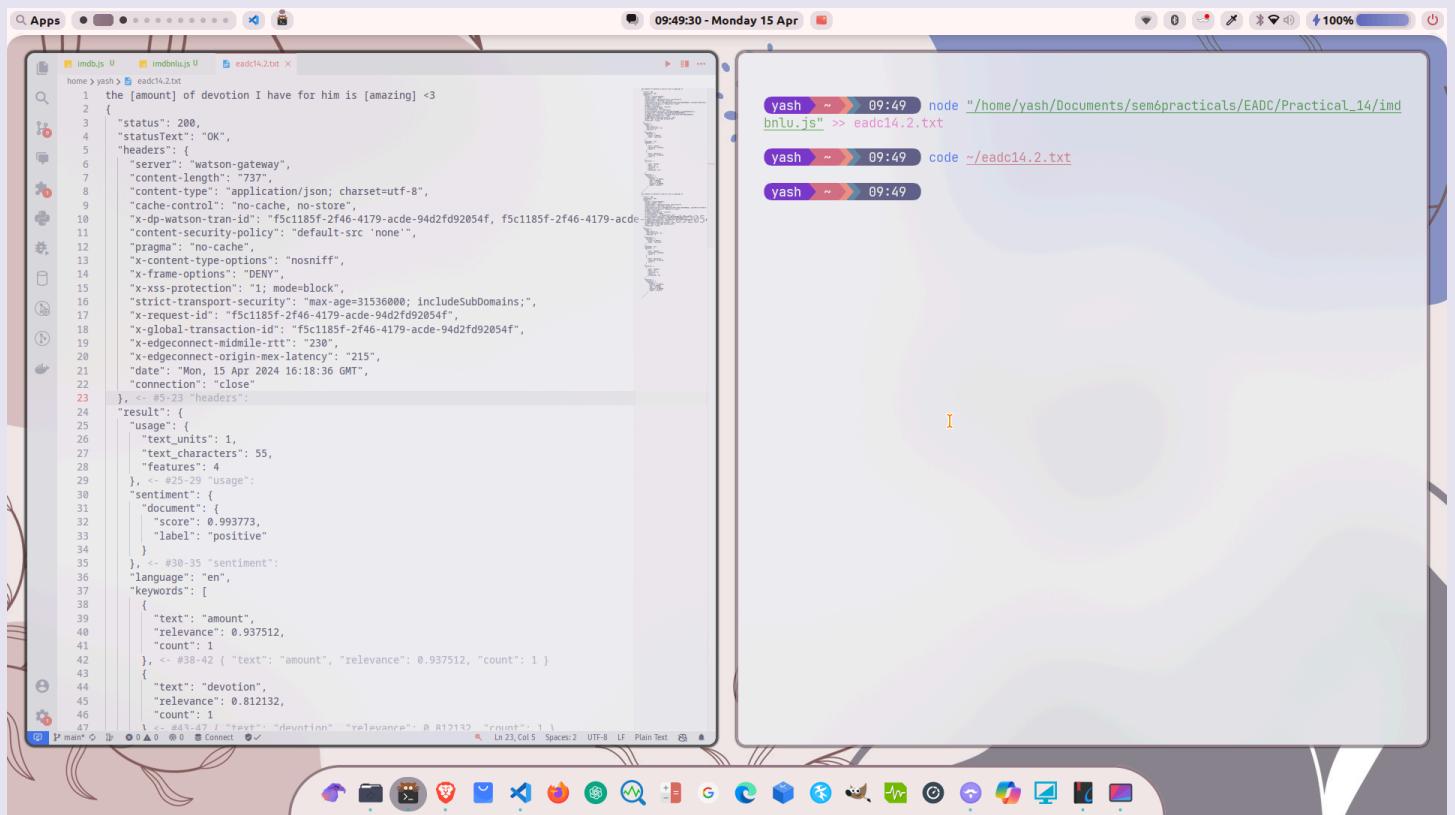
Branch - CBA      Batch - 61

EADC Practical 14

```
        emotion: {}  
    } ,  
};  
  
const nluResponse = await nlu.analyze(analyzeParams);  
console.log(JSON.stringify(nluResponse, null, 2));  
} catch (error) {  
    console.error(error);  
}  
}  
  
getData();
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 14**

11. Here also, the output of node can be redirected to text file for storing it

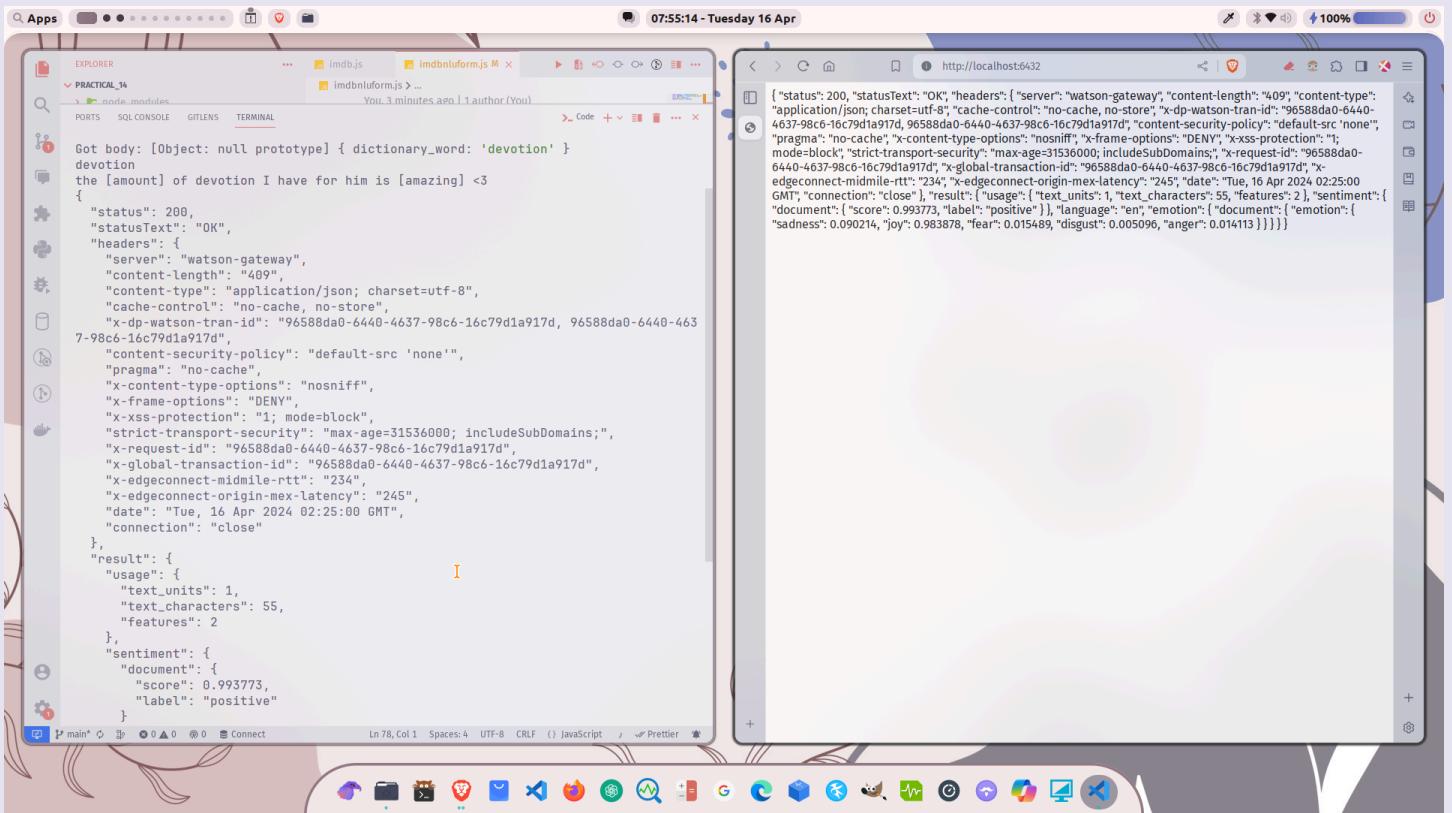


```
home/yash> cat eadc14.2.txt
1 the [amount] of devotion I have for him is [amazing] <3
2 {
3   "status": 200,
4   "statusText": "OK",
5   "headers": {
6     "server": "watson-gateway",
7     "content-length": "737",
8     "content-type": "application/json; charset=utf-8",
9     "cache-control": "no-cache, no-store",
10    "x-dp-watson-tran-id": "f5c1185f-2f46-4179-acde-94d2fd92054f",
11    "content-security-policy": "default-src 'none'",
12    "pragma": "no-cache",
13    "x-content-type-options": "nosniff",
14    "x-frame-options": "DENY",
15    "x-xss-protection": "1; mode=block",
16    "strict-transport-security": "max-age=31536000; includeSubDomains",
17    "x-request-id": "f5c1185f-2f46-4179-acde-94d2fd92054f",
18    "x-global-transaction-id": "f5c1185f-2f46-4179-acde-94d2fd92054f",
19    "x-edgeconnect-midmile-rtt": "230",
20    "x-edgeconnect-origin-mex-latency": "215",
21    "date": "Mon, 15 Apr 2024 16:18:36 GMT",
22    "connection": "close"
23  }, <- #5-23 "headers":
24  "result": {
25    "usage": {
26      "text_units": 1,
27      "text_characters": 55,
28      "features": 4
29    }, <- #25-29 "Usage":
30    "sentiment": {
31      "document": {
32        "score": 0.993773,
33        "label": "positive"
34      }
35    }, <- #30-35 "sentiment":
36    "language": "en",
37    "Keywords": [
38      {
39        "text": "amount",
40        "relevance": 0.937512,
41        "count": 1
42      }, <- #38-42 { "text": "amount", "relevance": 0.937512, "count": 1 }
43      {
44        "text": "devotion",
45        "relevance": 0.812132,
46        "count": 1
47      }
48    ]
49  }
50  <- #48-47 J "text", "devotion", "relevance": 0.812132 "count": 1
51  Ln 23, Col 5  Spaces:2  UTF-8  LF Plain Text 89
```

```
yash ~ 09:49 | node "/home/yash/Documents/sem6practicals/EADC/Practical_14/imdblu.js" >> eadc14.2.txt
yash ~ 09:49 | code ~/eadc14.2.txt
yash ~ 09:49
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 61  
EADC Practical 14

## 12. Run the NLU form node code to get input from HTML page



Code :

```
const express = require('express');

const bodyParser = require('body-parser');

const app = express();

require('dotenv').config();
```

```
const port = 6432;

var urlencodedParser = bodyParser.urlencoded({ extended: false })

const axios = require('axios');

const { IamAuthenticator } = require('ibm-watson/auth');

const NaturalLanguageUnderstandingV1 =
```

```
require('ibm-watson/natural-language-understanding/v1');
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
// Initialize Watson NLU
const nlu = new NaturalLanguageUnderstandingV1({
  authenticator: new IamAuthenticator({ apikey: process.env.NLU_apikey
}), // Replace 'YOUR_API_KEY' with your actual API key
  version: '2021-03-25',
  serviceUrl: process.env.NLU_serviceURL, // Replace 'YOUR_INSTANCE_ID'
with your actual instance ID
});

app.get('/', (req, res) => {
  res.sendFile(__dirname + '/public/index.html');
});

app.post('/', urlencodedParser, (req, res) => {
  console.log('Got body:', req.body);
  x = req.body.dictionary_word;
  console.log(x);
  /////////
  const options = {
    method: 'GET',
    url:
'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
    params: {
      term: req.body.dictionary_word
    },
    headers: {
      'X-RapidAPI-Key': process.env.RapidAPI_key,
      'X-RapidAPI-Host':
'mashape-community-urban-dictionary.p.rapidapi.com'
    }
  };
});
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
async function getData() {  
    try {  
        const response = await axios.request(options);  
        const example = response.data.list[1].example;  
        console.log(example);  
        // Perform NLU analysis  
        const analyzeParams = {  
            text: example,  
            features: {  
                // entities: {},  
                //keywords: {},  
                sentiment: {},  
                emotion: {}  
            },  
        };  
  
        const nluResponse = await nlu.analyze(analyzeParams);  
        console.log(JSON.stringify(nluResponse, null, 2));  
        res.send(JSON.stringify(nluResponse, null, 2));  
        // res.send(nluResponse.result.sentiment.document.label);  
        //console.log(nluResponse.result.sentiment.document.label);  
    } catch (error) {  
        console.error(error);  
    }  
}  
  
getData();  
//////////
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
app.listen(port, () => {
  console.log(`Server running on port ${port}`);
}) ;
```

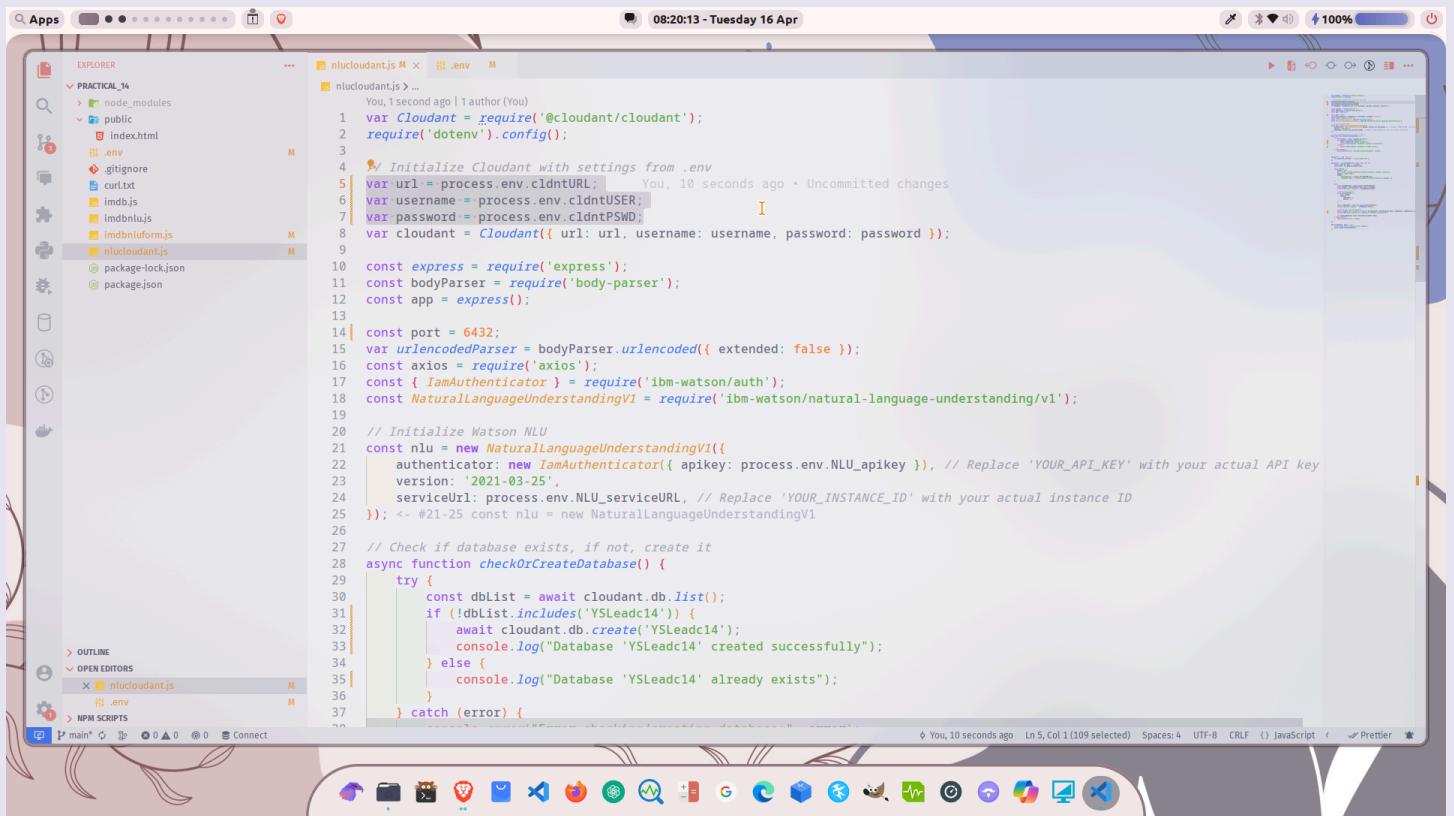
13. If only label of sentiment is printed in HTML, then uncomment that line commenting the full response printing line

The screenshot shows a Mac desktop environment. On the left, a code editor window is open, displaying a file named `nlucloudant.js`. The code contains several lines of JavaScript, including an `app.listen` call at the bottom. On the right, a web browser window is open at the URL `http://localhost:6432`, showing the word "positive". The system tray at the bottom of the screen displays various icons.

```
08:07:34 - Tuesday 16 Apr
nlucloudant.js  imdbsnluform.js M .env M
imdbsnluform.js > f app.post('/').callback > f getData
25 app.post('/', urlencodedParser, (req, res) => {
42   async function getData() {
57     const nluResponse = await nlu.analyze(analyzeParams);
58     console.log(JSON.stringify(nluResponse, null, 2));
59     // res.send(JSON.stringify(nluResponse, null, 2));
60     res.send(nluResponse.result.sentiment.document.label);
61     //console.log(nluResponse.result.sentiment.document.label);
62   } catch (error) {
63     console.error(error);
64   }
65 }
66 } <- #42-66 async function getData()
67
68 getData();
69 } <- #25-69 app.post
70 //////////
71
72
73
74
75 app.listen(port, () => {
  },
  "language": "en",
  "emotion": {
    "document": {
      "emotion": {
        "sadness": 0.090214,
        "joy": 0.983878,
        "fear": 0.015489,
        "disgust": 0.005096,
        "anger": 0.014113
      }
    }
  }
});
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

14. Now, to add response data to Cloudant database, paste the URL, username and password credentials to the code below



```
You, 1 second ago | 1 author (You)
1 var Cloudant = require('@cloudant/cloudant');
2 require('dotenv').config();
3
4 // Initialize Cloudant with settings from .env
5 var url = process.env.cldntURL; You, 10 seconds ago * Uncommitted changes
6 var username = process.env.cldntUSER;
7 var password = process.env.cldntPSWD;
8 var cloudant = Cloudant({ url: url, username: username, password: password });
9
10 const express = require('express');
11 const bodyParser = require('body-parser');
12 const app = express();
13
14 const port = 6432;
15 var.urlencodedParser = bodyParser.urlencoded({ extended: false });
16 const axios = require('axios');
17 const { IamAuthenticator } = require('ibm-watson/auth');
18 const NaturalLanguageUnderstandingV1 = require('ibm-watson/natural-language-understanding/v1');
19
20 // Initialize Watson NLU
21 const nlu = new NaturalLanguageUnderstandingV1({
22   authenticator: new IamAuthenticator({ apikey: process.env.NLU_apikey }), // Replace 'YOUR_API_KEY' with your actual API key
23   version: '2021-03-25',
24   serviceUrl: process.env.NLU_serviceURL, // Replace 'YOUR_INSTANCE_ID' with your actual instance ID
25 }); <- #21-25 const nlu = new NaturalLanguageUnderstandingV1
26
27 // Check if database exists, if not, create it
28 async function checkOrCreateDatabase() {
29   try {
30     const dbList = await cloudant.db.list();
31     if (!dbList.includes('YSLeadc14')) {
32       await cloudant.db.create('YSLeadc14');
33       console.log("Database 'YSLeadc14' created successfully");
34     } else {
35       console.log("Database 'YSLeadc14' already exists");
36     }
37   } catch (error) {
38
39   }
40 }
41
42 // Create a new document in the database
43 const createDocument = async () => {
44   const document = {
45     title: "Hello World",
46     content: "This is a test document."
47   };
48
49   const result = await cloudant.db('YSLeadc14').post(document);
50   console.log(result);
51 }
52
53 // Read a document from the database
54 const readDocument = async () => {
55   const result = await cloudant.db('YSLeadc14').get('56789');
56   console.log(result);
57 }
58
59 // Update a document in the database
60 const updateDocument = async () => {
61   const document = {
62     _id: '56789',
63     title: "Hello World Updated",
64     content: "This is a test document updated."
65   };
66
67   const result = await cloudant.db('YSLeadc14').update(document);
68   console.log(result);
69 }
70
71 // Delete a document from the database
72 const deleteDocument = async () => {
73   const result = await cloudant.db('YSLeadc14').remove('56789');
74   console.log(result);
75 }
76
77 // Get all documents in the database
78 const getAllDocuments = async () => {
79   const result = await cloudant.db('YSLeadc14').all();
80   console.log(result);
81 }
82
83 // Get a specific document by ID
84 const getDocumentById = async () => {
85   const result = await cloudant.db('YSLeadc14').get('56789');
86   console.log(result);
87 }
88
89 // Delete the database
90 const deleteDatabase = async () => {
91   const result = await cloudant.db('YSLeadc14').remove();
92   console.log(result);
93 }
94
95 // Export the app
96 module.exports = app;
```

Code :

```
var Cloudant = require('@cloudant/cloudant');
require('dotenv').config();

// Initialize Cloudant with settings from .env
var url = process.env.cldntURL;
var username = process.env.cldntUSER;
var password = process.env.cldntPSWD;
var cloudant = Cloudant({ url: url, username: username, password: password });

const express = require('express');
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
const bodyParser = require('body-parser');

const app = express();

const port = 6432;

var urlencodedParser = bodyParser.urlencoded({ extended: false });

const axios = require('axios');

const { IamAuthenticator } = require('ibm-watson/auth');

const NaturalLanguageUnderstandingV1 =
require('ibm-watson/natural-language-understanding/v1');

// Initialize Watson NLU

const nlu = new NaturalLanguageUnderstandingV1({
  authenticator: new IamAuthenticator({ apikey: process.env.NLU_apikey }),
  // Replace 'YOUR_API_KEY' with your actual API key
  version: '2021-03-25',
  serviceUrl: process.env.NLU_serviceURL, // Replace 'YOUR_INSTANCE_ID'
  with your actual instance ID
});

// Check if database exists, if not, create it

async function checkOrCreateDatabase() {
  try {
    const dbList = await cloudant.db.list();
    if (!dbList.includes('ysleadc14')) {
      await cloudant.db.create('ysleadc14');
      console.log("Database 'ysleadc14' created successfully");
    } else {
      console.log("Database 'ysleadc14' already exists");
    }
  } catch (error) {
    console.error("Error checking/creating database:", error);
  }
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

```
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/public/index.html');
});

app.post('/', urlencodedParser, async (req, res) => {
  console.log('Got body:', req.body);
  const term = req.body.dictionary_word;

  const options = {
    method: 'GET',
    url:
'https://mashape-community-urban-dictionary.p.rapidapi.com/define',
    params: { term: term },
    headers: {
      'X-RapidAPI-Key': process.env.RapidAPI_key,
      'X-RapidAPI-Host':
'mashape-community-urban-dictionary.p.rapidapi.com'
    }
  };

  try {
    const axiosResponse = await axios.request(options);
    const example = axiosResponse.data.list[1].example;
    console.log("Axios response:", axiosResponse.data);

    // Perform NLU analysis
    const analyzeParams = {
      text: example,
      features: {
        sentiment: {},
        emotion: {}
      },
    },
  }
});
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 14

```
};

const nluResponse = await nlu.analyze(analyzeParams);
console.log("NLU response:", nluResponse.result);

// Insert data into the database
await cloudant.use('ysleadc14').insert({ axiosResponse:
axiosResponse.data, nluResponse: nluResponse.result });
    console.log("Data inserted into 'ysleadc14' database
successfully");

    res.send(nluResponse.result.sentiment.document.label);
} catch (error) {
    console.error("Error:", error);
}
});

app.listen(port, async () => {
    console.log(`Server running on port ${port}`);
    await checkOrCreateDatabase();
});
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

## 15. Run the code and enter the term to search dictionary and add data to cloudant

The screenshot displays a macOS desktop environment with two windows open. On the left is a terminal window titled 'main' with the command `node nlucloudant.js` running. The output shows the script initializing a Cloudant client, starting a server on port 6432, and receiving a POST request with the word 'transcendental'. It then performs a database query and returns a definition from a Watson Natural Language Understanding API response. On the right is a web browser window titled 'Dictionary word submission for Analysis' with the URL `http://localhost:6432`. The page contains a single input field labeled 'Dictionary Word' with the value 'transcendental' and a 'Submit' button.

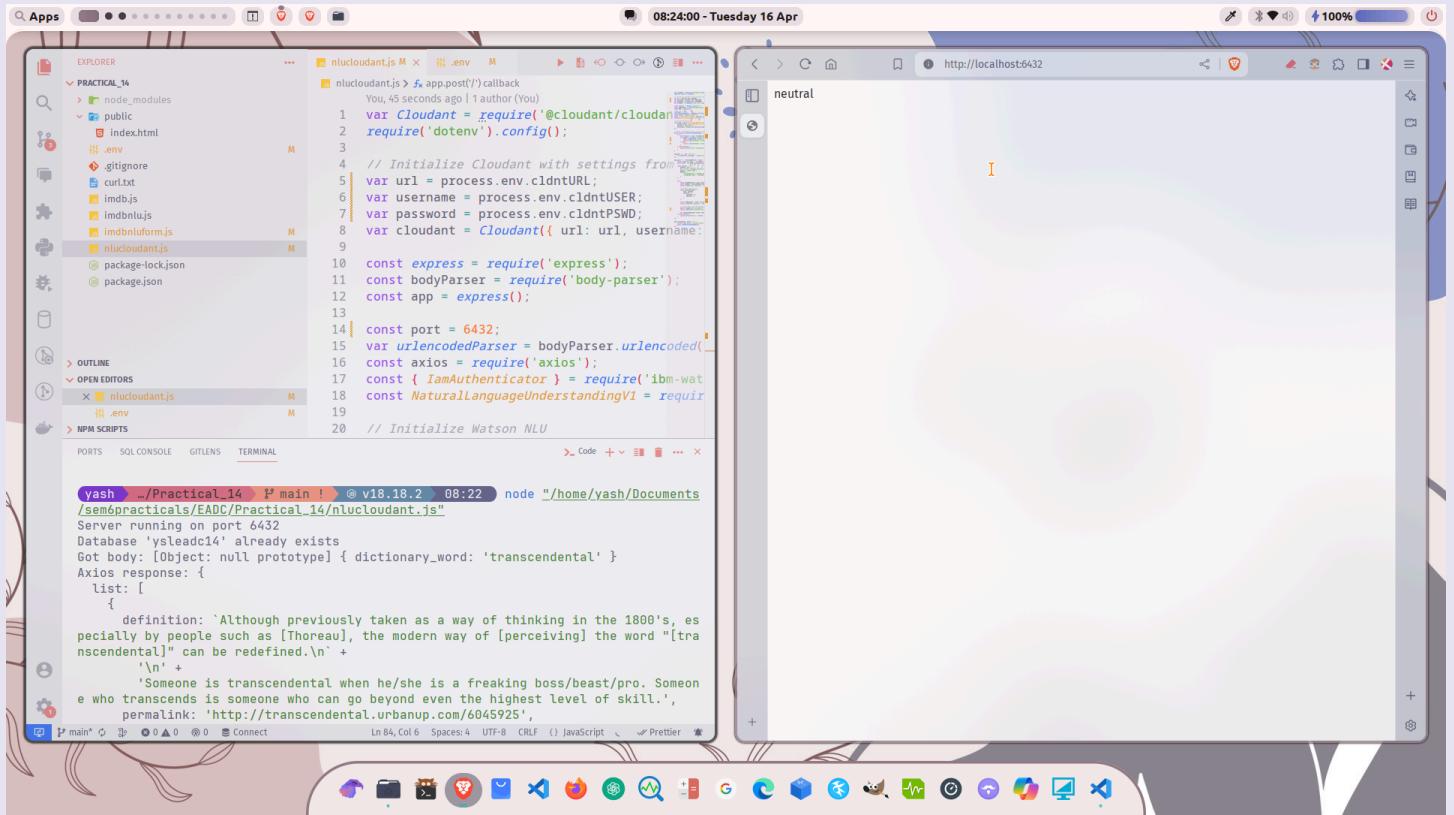
```
08:23:29 - Tuesday 16 Apr
nlucloudant.js M .env M
nlucloudant.js > f app.post() callback
You, 45 seconds ago | 1 author (You)
1 var Cloudant = require('@cloudant/cloudant');
2 require('dotenv').config();
3
4 // Initialize Cloudant with settings from .env
5 var url = process.env.cldntURL;
6 var username = process.env.cldntUSER;
7 var password = process.env.cldntPSWD;
8 var cloudant = Cloudant({ url: url, username: username, password: password });
9
10 const express = require('express');
11 const bodyParser = require('body-parser');
12 const app = express();
13
14 | const port = 6432;
15 | var urlencodedParser = bodyParser.urlencoded({
16 |   extended: true
17 | });
18 const axios = require('axios');
19 const { IamAuthenticator } = require('ibm-watson/natural-language-understanding/v1');
20 const NaturalLanguageUnderstandingV1 = require('ibm-watson/natural-language-understanding/v1');

Dictionary word submission for Analysis
Dictionary Word
transcendental
Submit

yash ~/Practical_14 ② main ! @ v18.18.2 08:22 node ~/home/yash/Documents/sem6practicals/EADC/Practical_14/nlucloudant.js
Server running on port 6432
Database 'ysleadc14' already exists
Got body: [Object: null prototype] { dictionary_word: 'transcendental' }
Axios response:
{
  list: [
    {
      definition: 'Although previously taken as a way of thinking in the 1800\'s, especially by people such as [Thoreau], the modern way of [perceiving] the word "[transcendental]" can be redefined.\n' +
        '\n' +
        'Someone is transcendental when he/she is a freaking boss/beast/pro. Someone who transcends is someone who can go beyond even the highest level of skill.',

      permalink: 'http://transcendental.urbanup.com/6045925',
    },
  ],
}
Ln 84, Col 6  Spaces: 4  UTF-8  CRLF  () JavaScript  ⚡ Prettier  ⌂
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 14**



The screenshot shows a macOS desktop environment with two windows open. On the left is a terminal window titled 'main' with the command 'yash .../Practical\_14 p main ! @ v18.18.2 08:22 node "/home/yash/Documents/\_sem6practicals/EADC/Practical\_14/nlucloudant.js"'. The output of the command is displayed, showing the server starting on port 6432 and a database named 'ysleadc14' already existing. It also shows a dictionary word 'transcendental' and its definition from the Watson NLU service. On the right is a web browser window titled 'neutral' with the URL 'http://localhost:6432'. The browser displays the same content as the terminal, indicating a successful API call.

```
nlucloudant.js M x .env M 08:24:00 - Tuesday 16 Apr
nlucloudant.js > f app.post('/callback
You, 45 seconds ago | 1 author (You)
1 var Cloudant = require('@cloudant/cloudant');
2 require('dotenv').config();
3
4 // Initialize Cloudant with settings from
5 var url = process.env.cldntURL;
6 var username = process.env.cldntUSER;
7 var password = process.env.cldntPWD;
8 var cloudant = Cloudant({ url: url, username: username, password: password });
9
10 const express = require('express');
11 const bodyParser = require('body-parser');
12 const app = express();
13
14 const port = 6432;
15 var urlencodedParser = bodyParser.urlencoded({
16   extended: true
17 });
18 const axios = require('axios');
19 const IamAuthenticator = require('ibm-watson-iam-authenticator');
20 const NaturalLanguageUnderstandingV1 = require('ibm-watson-natural-language-understanding/v1');
21
22 app.use(urlencodedParser);
23
24 app.get('/', (req, res) => {
25   res.send('Hello, World!');
26 });
27
28 app.post('/callback', (req, res) => {
29   const { body } = req;
30   const { definition } = body;
31   res.json({ definition });
32 });
33
34 app.listen(port, () => {
35   console.log(`Server running on port ${port}`);
36 });
37
38 module.exports = app;
```

```
yash .../Practical_14 p main ! @ v18.18.2 08:22 node "/home/yash/Documents/_sem6practicals/EADC/Practical_14/nlucloudant.js"
Server running on port 6432
Database 'ysleadc14' already exists
Got body: [Object: null prototype] { dictionary_word: 'transcendental' }
Axios response: {
  data: {
    definition: 'Although previously taken as a way of thinking in the 1800\'s, especially by people such as [Thoreau], the modern way of [perceiving] the word "[transcendental]" can be redefined.\n' +
      '\n' +
      'Someone is transcendental when he/she is a freaking boss/beast/pro. Someone who transcends is someone who can go beyond even the highest level of skill.',
    permalink: 'http://transcendental.urbanup.com/6045925',
    title: 'transcendental'
  }
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 14

## 16. Check the cloudant database if data is inserted successfully

```
08:25:46 - Tuesday 16 Apr
https://60379ad1-b8cb-4933-bb03-b1bfa6df8a7e-bluemix.cloudant.com/dashboard.html#database/ysleadc14/d8977cd0bd1a97004c659ae7ae96d1b3
Save Changes Cancel Upload Attachment Clone Document Delete Log Out

[{"list": [{"definition": "Although previously taken as a way of thinking in the 1800's, especially by people such as [Thoreau], the modern way of [perceiving] the word \"[transcendental]\" can be redefined.\n\nSomeone is transcendental when they have a sense of being above or beyond the ordinary, especially in terms of spirituality or philosophy.", "permalink": "http://transcendental.urbanup.com/6045925", "author": "[hx]aurora", "word": "Transcendental", "defid": "6045925", "current_vote": "", "written_on": "2011-08-28T17:43:12.000Z", "example": "Person 1: \"Dude, check out that kid, he's [such a beast].\"\r\nPerson 2: \"Nah [mang], he's better than that. He's [transcendental].\"", "thumbs_up": 99, "thumbs_down": 35}, {"definition": "A movement of people who, before the civil war, made it their goal to spawn a body of literature that was wholly American and unique from anything the likes of which Europe had seen. Writers of the movement focused on nature, individualism, and self-reliance.", "permalink": "http://transcendentalism.urbanup.com/4924128", "author": "Sithstress", "word": "Transcendentalism", "defid": "4924128", "current_vote": "", "written_on": "2010-04-29T08:19:49.000Z", "example": "Some authors who were part of the movement of [transcendentalism] include [Ralph Waldo Emerson], Henry David Thoreau, Thomas Wentworth [Higginson], Emily Dickinson, Theodore Parker, Margaret Fuller, and Harriet Marti", "thumbs_up": 18, "thumbs_down": 5}, {"definition": "A concept that was the base of the [Transcendental] Movement of the 1830s. Represents an idealistic thought system that focused on the spirit and felt it more important than mere matter. Followers believed in a spiritual world that could be experienced through intuition and direct contact with God.", "permalink": "http://transcendentalism.urbanup.com/419753", "author": "Sithstress", "word": "Transcendental", "defid": "419753", "current_vote": "", "written_on": "2010-04-29T08:19:49.000Z", "example": "The Transcendentalists believed in the inherent goodness of humanity and the possibility of achieving a state of perfection through personal experience and direct communion with the divine.", "thumbs_up": 62, "thumbs_down": 11}]}]
```