AI Powered News Search App with slack bot integration

Project Report

**INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | |  | **INTRODUCTION** | |  |  |  |
|  | |  | 1.1 Overview | |  |  |  |
|  | |  | 1.2 Purpose | |  |  |  |
| **2** | |  | **LITERATURE SURVEY** | |  |  |  |
|  | |  | 2.1 Existing problem | | |  |  |
|  | |  | 2.2 Proposed solution | | |  |  |
| **3** | |  | **THEORITICAL ANALYSIS** | |  |  |  |
|  | |  | 3.1 Block diagram | |  |  |  |
|  | |  | 3.2 Hardware / Software designing | | |  |  |
| **4** | |  | **EXPERIMENTAL INVESTIGATIONS** | | |  |  |
| **5** | |  | **FLOWCHART** | |  |  |  |
| **6** | |  | **RESULT** |  |  |  |  |
| **7** | |  | **ADVANTAGES & DISADVANTAGES** | | |  |  |
| **8** | |  | **APPLICATIONS** | |  |  |  |
| **9** | |  | **CONCLUSION** | |  |  |  |
| **10** | |  | **FUTURE SCOPE** | | **`** |  |  |
| **11** | |  | **BIBILOGRAPHY** | |  |  |  |
|  | |  | **APPENDIX** | |  |  |  |
|  |  | | A. Source code | |  |  |  |

**1. INTRODUCTION**

* 1. **Overview**

**Project Summary:**

The web is home to massive amounts of data, with more being created every day. Organizations can harness this constant stream of information to gain understanding, plan strategies, and find opportunities. Enriched news data can help your application make dynamic connections across current events faster.

In this project we are going to build an AI Powered News Search application starting with the basics by creating an IBM account and Node-Red account and designing the UI through the nodes and using the IBM Watson discovery service for extracting the latest news articles. Enriched news data can help our application make dynamic connections across current events faster. The user interacts with the app UI to request the relevant news content. The app sends user requests to Watson Discovery News. The Watson Discovery Service is continually crawling the web to update its Discovery News collection.

The Watson Discovery Service responds to Slack search requests. And in our application users will be able to view the latest news according to the query he/she searched.

**1.2 Purpose:**

The purpose of the report is to improve news fetching methodologies. In this guide to sentiment analysis, we learn how by using node-red application we can fetch news that has a sentimental score or value to it. This node red flow is made such that we get the input as the text of the news, url of main page, author of the website, the sentimental score by using gauge, and bar graphs which show us the sentimental analysis like if the searched news is positive, negative or normal.

**2. LITERATURE SURVEY**

**2.1 Existing problem**

Too much information - It goes without saying that the distributed content production of individuals is an amazing and powerful phenomenon, but relying on search engines to index, sort and rank externally-produced content that isn't standardized for format doesn't work for the newsreader. There's still too much information and the organization is distorted by advertising dollars.

News is time-sensitive information – There should be constant organizing externally-produced content and figure out that new content was produced and, then organize all of the content that comes in many different structural formats and it has to passively (arithmetically) distinguish between news information (time-sensitive) and historical information that isn't time-sensitive.

News is geographically sensitive information - If you live in Dallas, TX do you want to have to sort through local content from Boston, MA or New York, NY or vice versa to find local news? No, you want to see time-sensitive information about what's happening in your community and NOW! While GeoRSS is out there, it's still nowhere near being adopted as a widely-used content standard.

**2.2 Proposed solution**

IBM has upgraded its Watson AI platform to boost the ability to understand business lingo that could be used to analyze the English language text and conversations.

We solved the problem of too much information by just giving three to four links of news that is related unlike providing too much information or content then we solved the problem of time-sensitive using the information posted at that particular time span like not the information that is posted previously that is irrelevant so as to distinguish between news information and historical information that isn't time-sensitive, the solution for geographically sensitive information like if we need information on covid-19 in India but we are getting information of Spain, America then that would be useless so to overcome this we can either specify the place of which information is necessary or the software needs to get (by satellites/simply from where the hardware part is bought) from where we are searching the news.

Finally we proposed solution as using the sentimental analysis in node-red application so as it can categorize the news searched as positive, negative or normal this is done firstly by using gauge which shows on the scale of -2 to +2 what number is given to the news if the score of news is between -2 to 0 then the label of the news is negative, if the news is 0 then the news is normal that is neither positive nor negative and if the news is between 0 to +2 then the news is labeled as positive.

**3. THEORITICAL ANALYSIS**

**3.1 Block diagram**

**SLACK**

**WATSON**

**DISCOVERY**

**NEWS**

**APP UI**

**USER**

**RSS**

**ONLINE**

**NEWS CONTENT**

**3.2 Hardware / Software designing**

* IBM Cloud
* Node-Red Application
* IBM Watson Discovery
* Node.js
* UI/UX designing
* GitHub
* Zoho Writer
* Slack
* Python/Java
* HTML
* RSS
* Windows

**4. EXPERIMENTAL INVESTIGATIONS**

This project used an experiment to the use of Watson Discovery to query news and display it to the user via UI and a slack bot. It was taken into consideration after examining differences in credibility assigned to news read in paper form and the same news read on a web site.

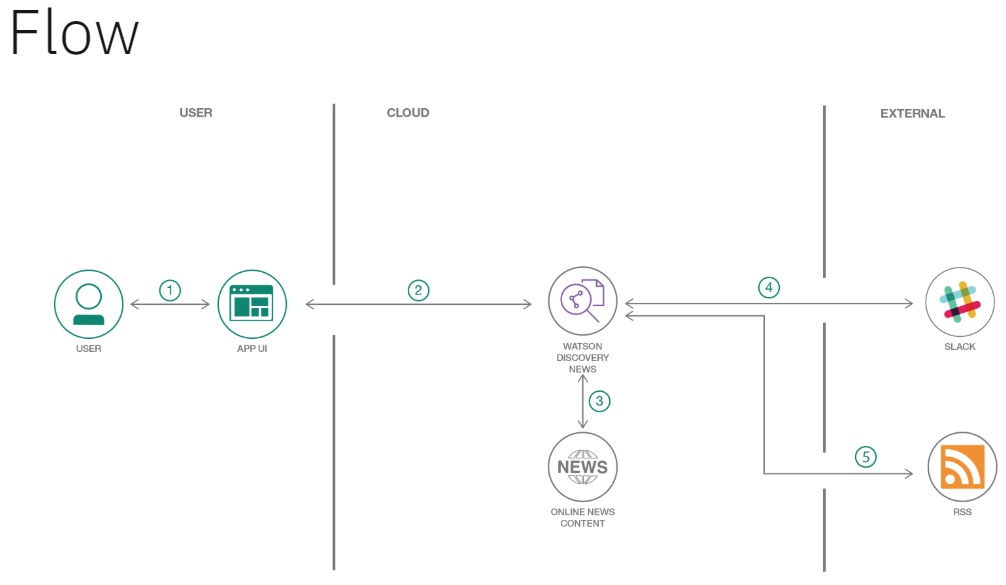
This project includes a bot. It is the same as a regular app: it can access the same range of APIs and do all of the magical things that a Slack App can do. A bot is a type of Slack App designed to interact with users via conversation. However, when you build a bot for your Slack App, you're giving that app a face, a name, and a personality and encouraging users to talk to it.

The bot can send DMs, it can be mentioned by users, it can post messages or upload files, and it can be invited to channels - or kicked out. Since bot is capable of doing everything that a Slack App can do, we're going to limit our focus to a common use-case for bots.

In the new digital era, brands and governments are targets for criticism and comment. They are challenged with how to manage large volumes of data and controversial news and how to address followers and skeptics. The beauty of the Cognizable solution is that it can deliver immediate insight with great precision and a remarkable level of detail.

Cognitive can offer qualitative data about people’s opinions and feelings. The solution processes huge amounts of structured and unstructured data. The process is automated, so it’s fast. To compile a similar sentiment analysis report manually would take approximately a week, and by that time, the data would be old. Additionally, beyond just “listening,” Cognizable can understand sarcasm, satire and emotion.

**5. FLOWCHART**



**6. RESULT**

The final application has

A user-interface is used to request the extraction of required news content.

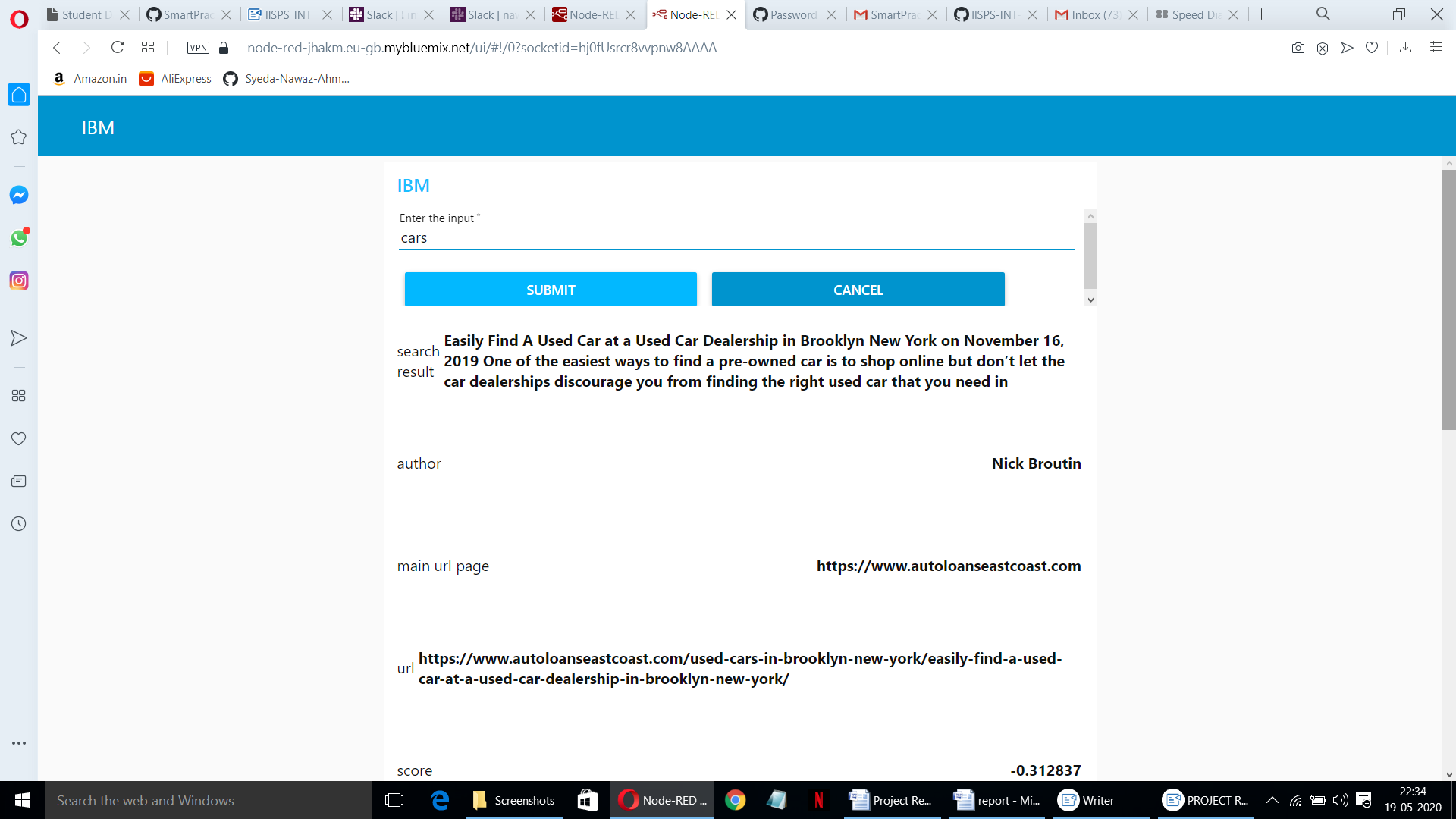
The app generates the latest news regarding request generated by user using the Watson Discovery News.

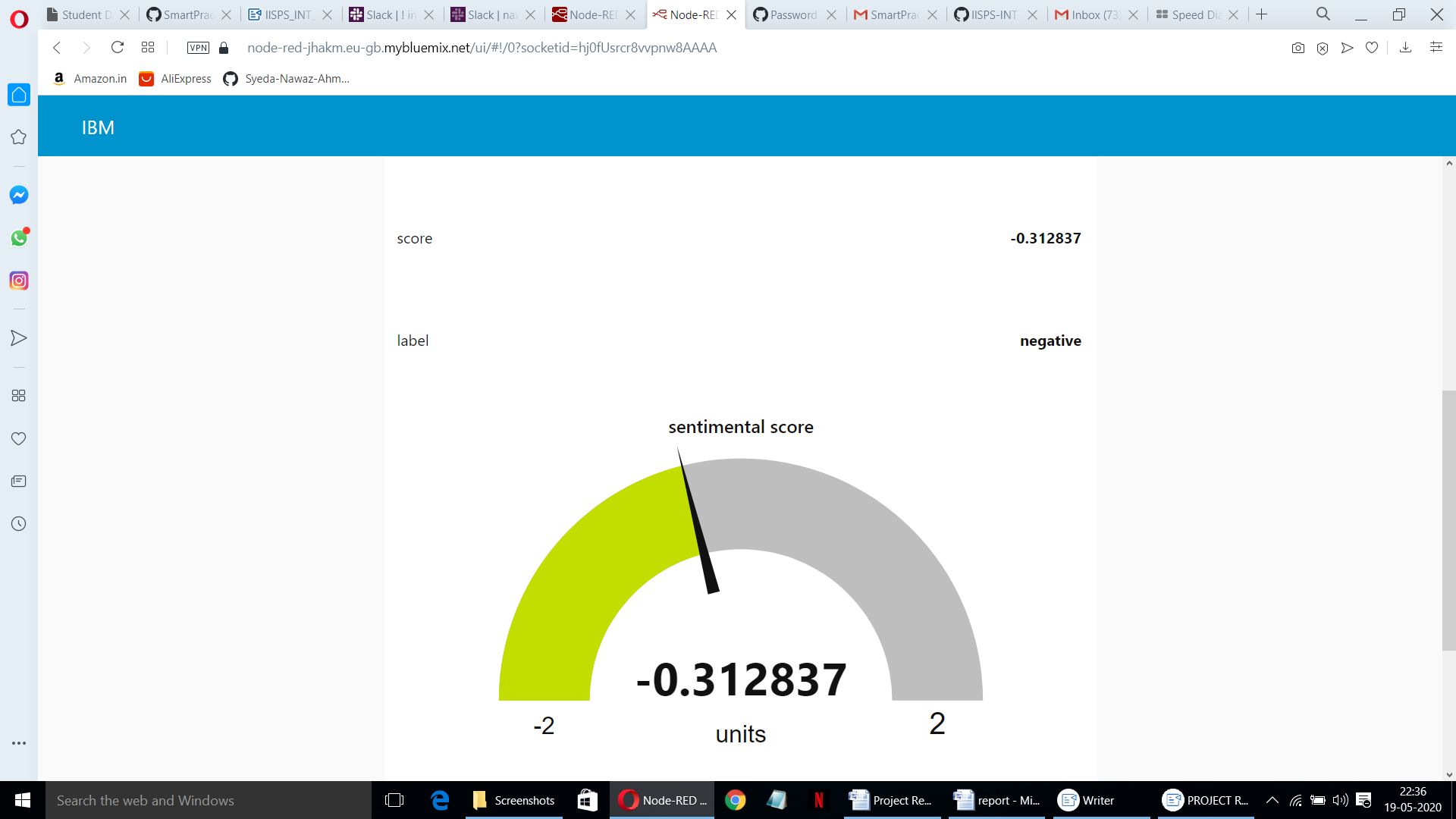
It generates the URL it used to extract the news from.

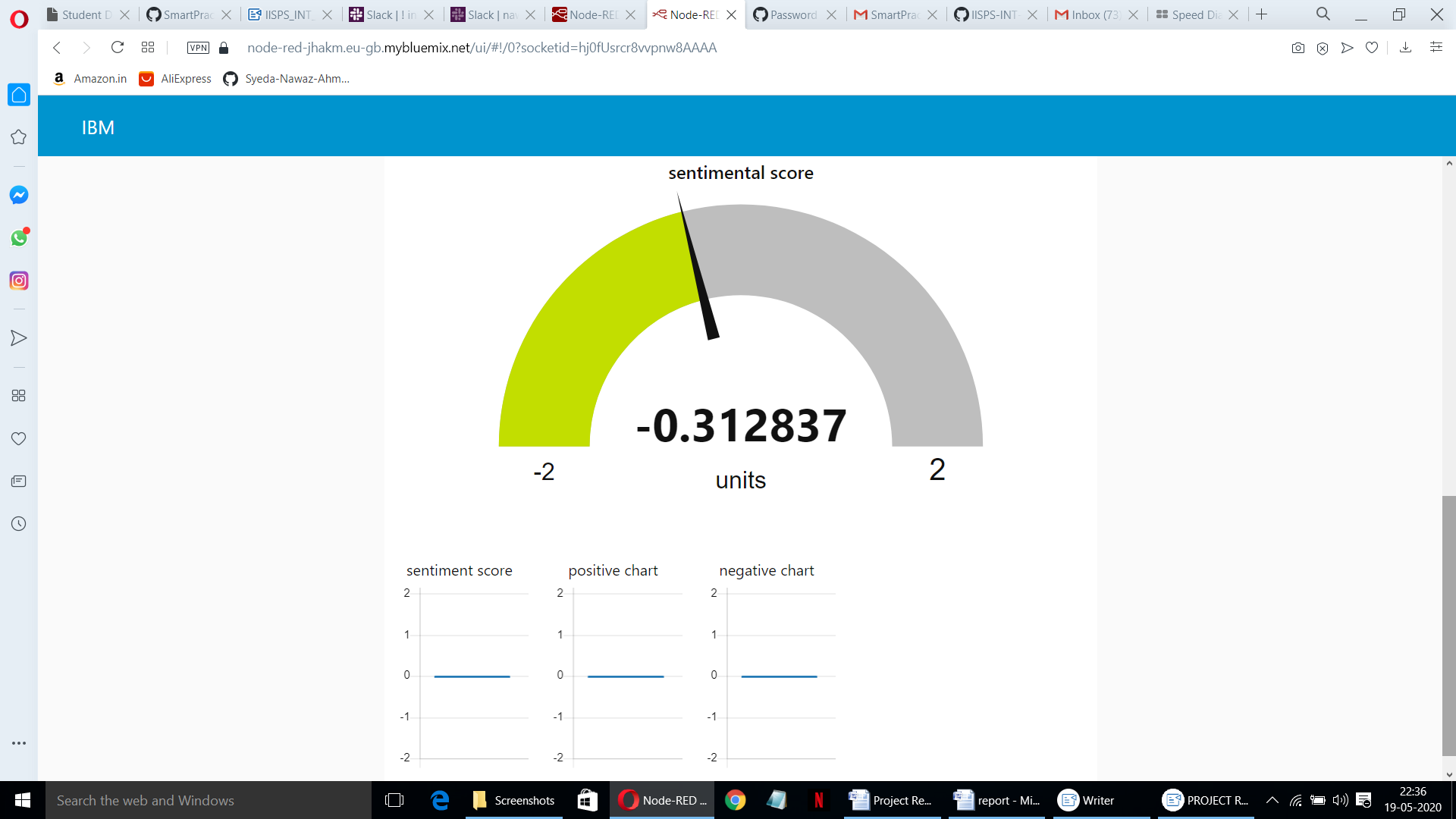
It displays the author who had submitted the news report.

It provides with sentimental analysis that separate positive or negative opinion from good or bad news.

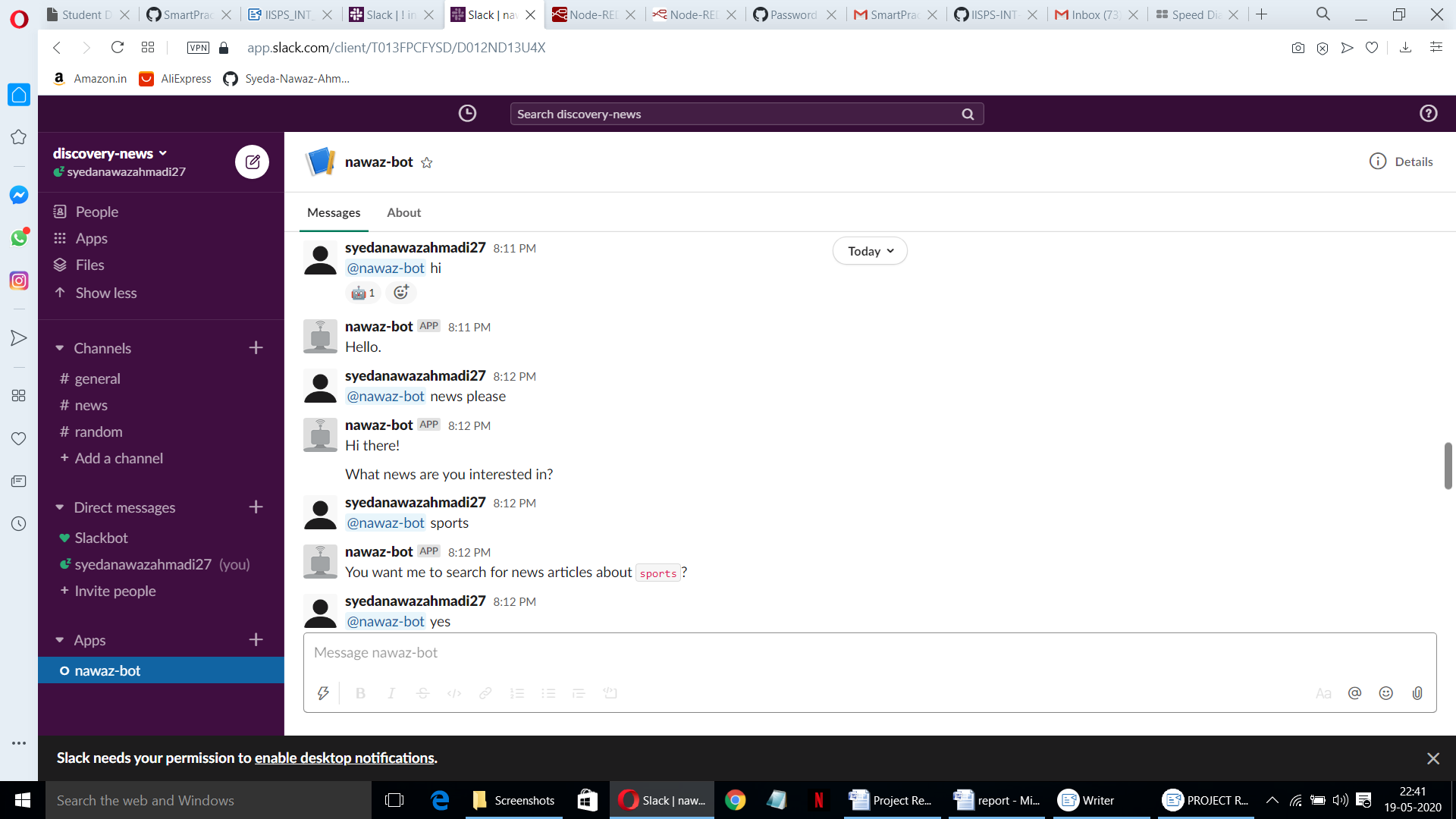
Node-Red UI result :

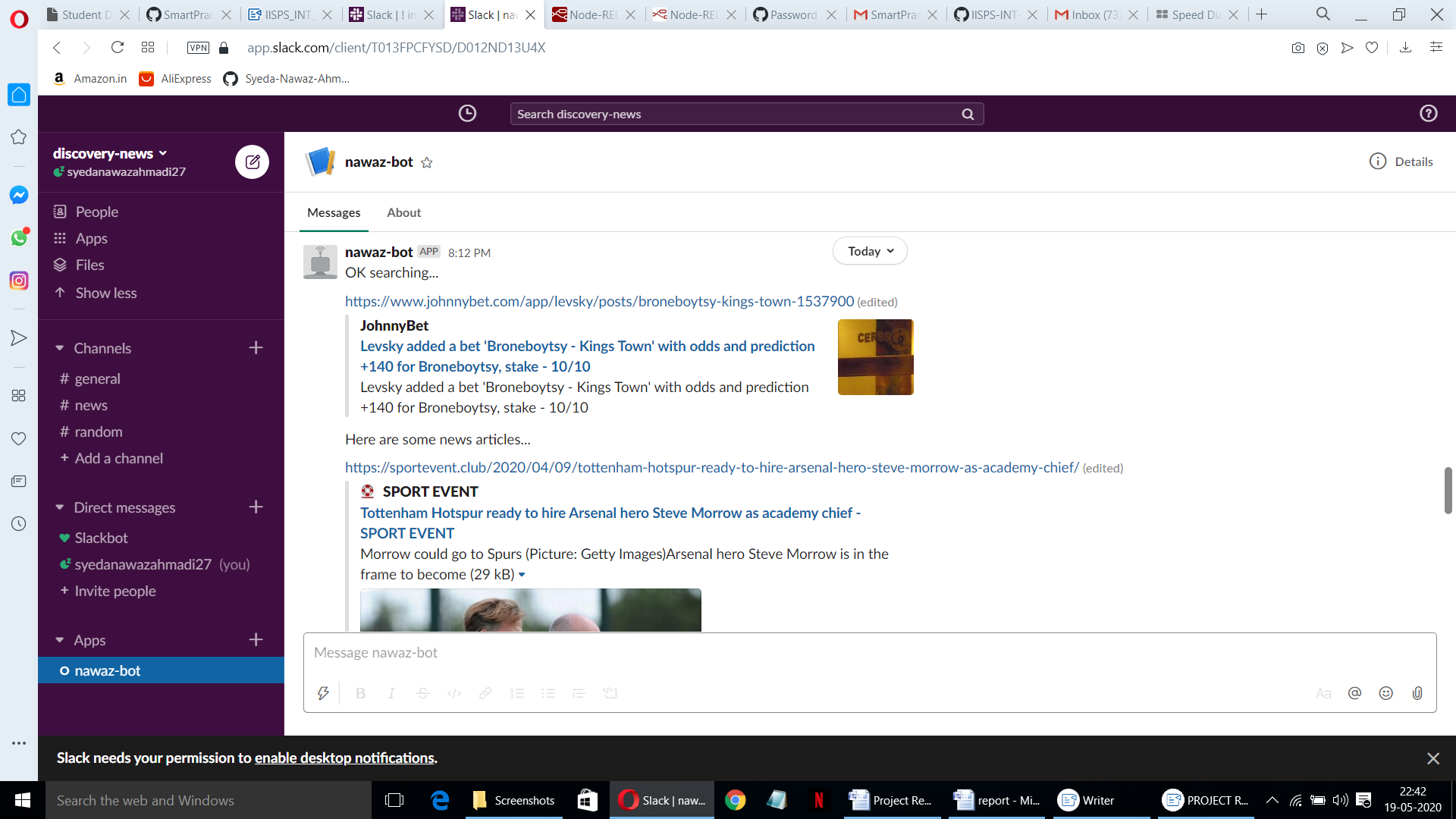


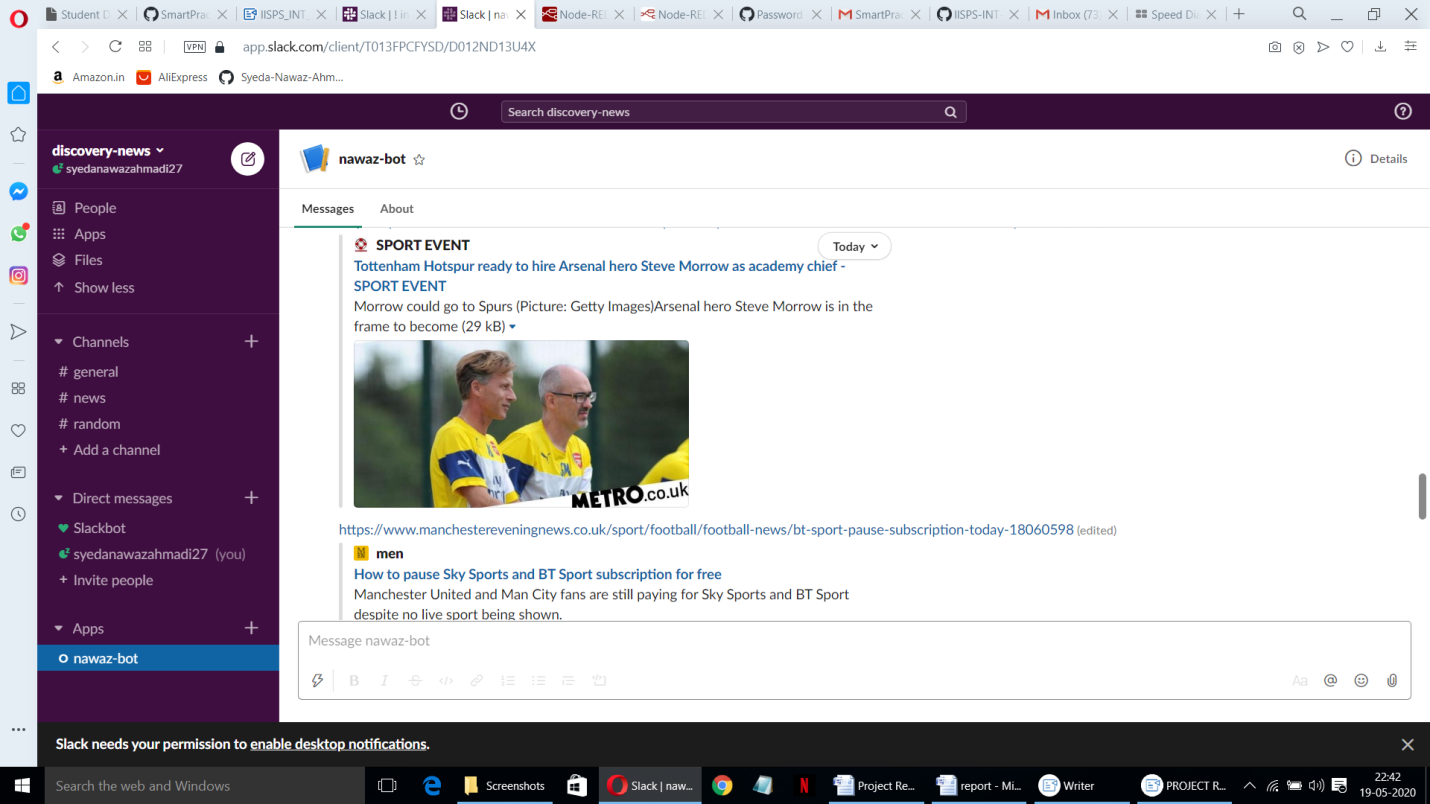


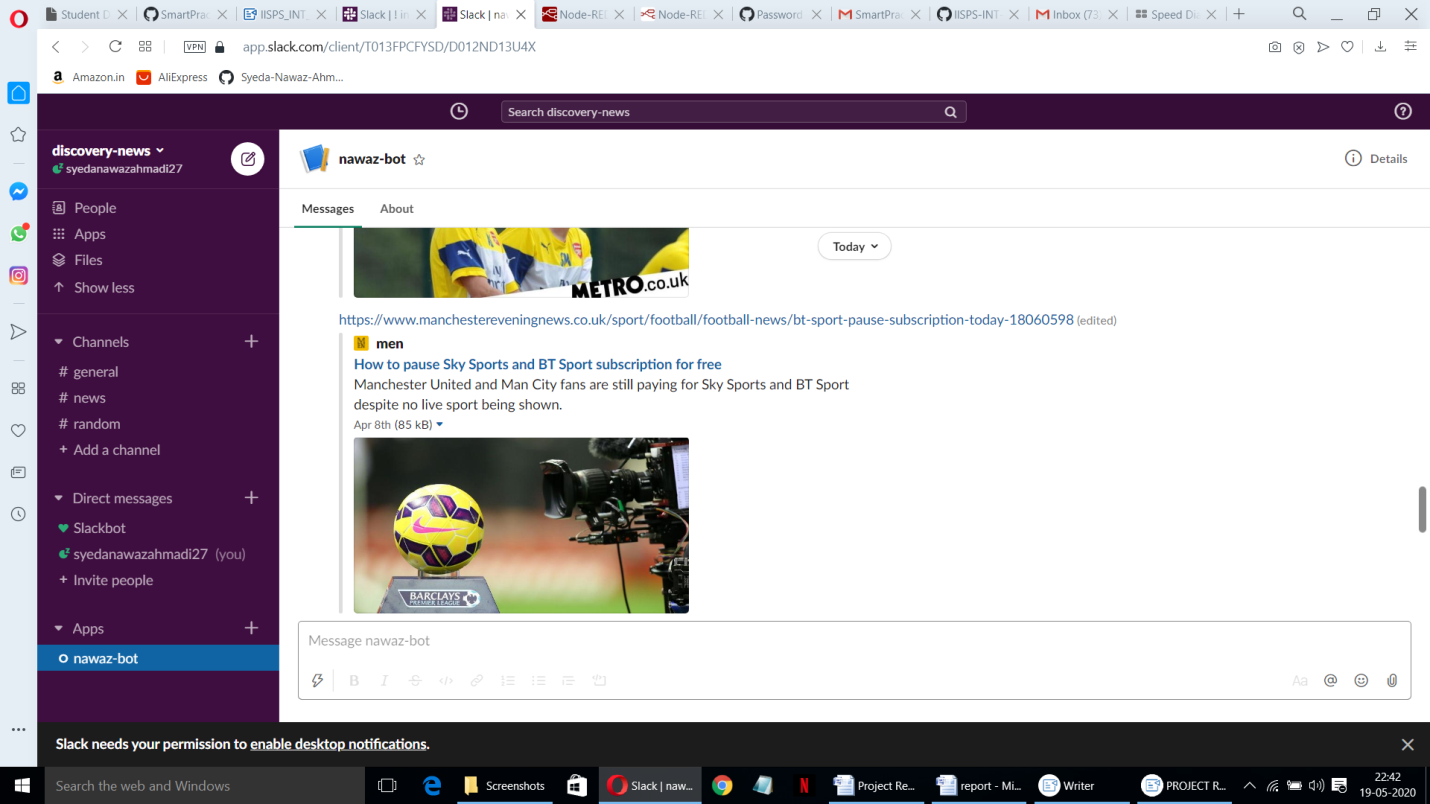


Slack Bot result :









**7. ADVANTAGES & DISADVANTAGES**

Advantages:

The Watson news search application includes following advantages:

* Easy to create.
* Less time required to develop.
* High performance with IBM cloud.

By querying and manipulating enriched data, you can build a more insightful search interface. This code pattern provides a Node.js app built on the Watson Discovery Service that does just that. The pattern demonstrates how you can use individual out-of-the-box UI components to extract and visualize the enriched data provided by the Discovery analytics engine.

Disadvantages:

Like other APIs and integrations, bot users are free. Unlike regular users, the actions they can perform are somewhat limited. For workspaces on the Free plan, each bot user counts as a separate integration.

Application Programming Interface (API) is Not Stable one of the key problems that most of the developers encounter is the Application Programming Interface (API) keeps on changing at frequent intervals and does not remain stable.

At times, a new API appears having a number of backwards-incompatible changes. As a result the developers are forced to make changes in the accessible code bases to match the compatibility with the latest version of the Node.js API.

The Watson news search application includes following disadvantages:

* We need to pay for IBM platform service.
* The services which we use require amount for space we use.

**8. APPLICATIONS**

Applications of Chatbots

A Chatbot is a program that can have a conversation with a person using rules and Artificial Intelligence (AI) in a way that mimics human-like conversations and interactions. Chatbots have become popular in the past few years as businesses discover innovative ways to put them to use. Having a Chatbot today has numerous benefits for businesses – they make life easier for customers, are available 24/7, save time (no more long waits to talk to a service rep) and they are easy to use.

These benefits have led to increased adoption of Chatbots by both businesses and consumers. This is great news for businesses and entrepreneurs as they can develop Chatbots to augment or improve the efficiency of any aspect of their business operations, improve customer engagement and enhance User Experience.

The watson news search application is used to search top trending news from news collections. It can be used for sentimental analysis of search result which is been gathered by queuing the search key provided by user respectively. With the help of slack bot integration we can fetch the news directly from slack channel by just inviting the slack bot to the channel and searching for interested news

**9. CONCLUSION**

This application uses individual out-of-the-box UI components to extract and visualize the enriched data provided by the Discovery analytics engine. The code pattern includes Watson Discovery series. It gets the customer sentiment insights from product reviews. And this project gave us some basic working knowledge of the Watson Discovery Service and shows how to use Discovery along with JavaScript and Node Red to build your own news mining web application. This project also tells about the integration process of slack with the Watson services to access and create our own bot to query news from the channel.

**10. FUTURE SCOPE**

Making more complex bots

we expected that users would respond with a very specific spelling, we assumed a test environment where there were no other conversations happening, and so on.

For a real bot in production, some of these assumptions would break the behavior of the bot. So let's cover some situations that you should address for your own bots - think of these as best practices rather than specific instructions to follow.

Tracking conversations

In our example bot, we've used a mention as the triggering point for a specific conversation, but you'll notice that your bot will still respond if you skip some of the steps - for example if you type Who's there?, your bot will respond to this message with a bot user, even if you didn't mention the bot or start at the beginning of the conversation.

A solution to this might involve tracking the beginning of a conversation, the participants involved, and the progress through the flow. For example, when the user first mentions the bot, a database entry is created that identifies that user and the open workflow with them.

As the user progresses through the flow, the database records this, and the user is unable to repeat earlier steps in the conversation (unless of course that is a desired behavior). Once the workflow is completed, the database entry is also marked as complete, and the bot waits for another mention before starting a new.

Threaded messages

Be aware that a user might choose to reply to your bots messages in a thread rather than at the channel-level. Your bot will still receive message events for these threaded replies, but you will have to add some extra logic to ensure that your bot responds to the user in the relevant location.

Check out the Threading Messages docs for more information on how to spot the difference between messages and threaded messages.

Variations in phrasing

Because your bot will be interacting with humans, it is unlikely that you can expect consistent spelling and phrasing across messages from different people that might be trying to invoke the same thing. For example, our simple example bot used the phrase tell me a joke to trigger the start of the workflow, but at a very basic level a user might also try to type what's a good joke? or make me laugh.

Your bot can get more complex by broadening its understanding of natural language queries to capture a wider range of potential trigger phrases. Alternatively you can be more prescriptive about the exact phrasing to use, and provide user education to train correct usage.

Integrating with other services

The real magic of a bot comes when it is connected with external services, providing a seamless conversational interface for them from within Slack. There is a huge range of possibilities for what your bot could do, so it might help to start with a great resource like the Bot kit Community.

**11. BIBLIOGRAPHY**

Name : Syeda Nawaz Ahmadi.

College Name : Muffakham Jah college of Engineering and Technology.

Work Title : AI Powered News Search App with slack bot integration.

References :

<https://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be>

<https://github.com/>

<https://www.youtube.com/watch?v=w3jLJU7DT5E&feature=youtu.be>

<https://www.youtube.com/watch?v=7YUTc4Cigc8&feature=youtu.be>

<https://www.ibm.com/cloud/get-started>

<https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>

<https://www.w3schools.com/howto/howto_make_a_website.asp>

<https://www.youtube.com/watch?v=W3iPbFTAAds&feature=youtu.be>

<https://www.youtube.com/watch?v=W3iPbFTAAds&feature=youtu.be>

<https://discovery-news-demo.ng.bluemix.net/>

APPENDIX :

Link to Node-Red work space :

https://node-red-jhakm.eu-gb.mybluemix.net/red/#flow/ccc750c1.534f2

Link to Node-Red UI :

https://node-red-jhakm.eu-gb.mybluemix.net/ui/#!/0?socketid=hj0fUsrcr8vvpnw8AAAA