NEWS TRACKER APPLICATION

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

This project is a web application that returns the news which the user need to know about the current. This is achieved by using Flask which is a famous python library. I have also used IBM DB to store credentials and APIs that provide me with news information. News Tracking Service is to analyze the new what is happen now.

1.2 APPLICATIONS

Applications for a News Tracker Application are vast. Some of the more common ones are:

- Easy to search news
- Can able to know more about the world.

1.3 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1.Easier to Access

2.Cost Efficient

3.Time Saving

DISADVANTAGE:

1. The Android applications are not so good.
2. The battery life of Android powered devices drains quickly.
3. Android applications contain virus.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Similar kinds of applications available on the internet are text based. They return you valuable news based on a search which helps to gain knowlwge about the particular information we need.

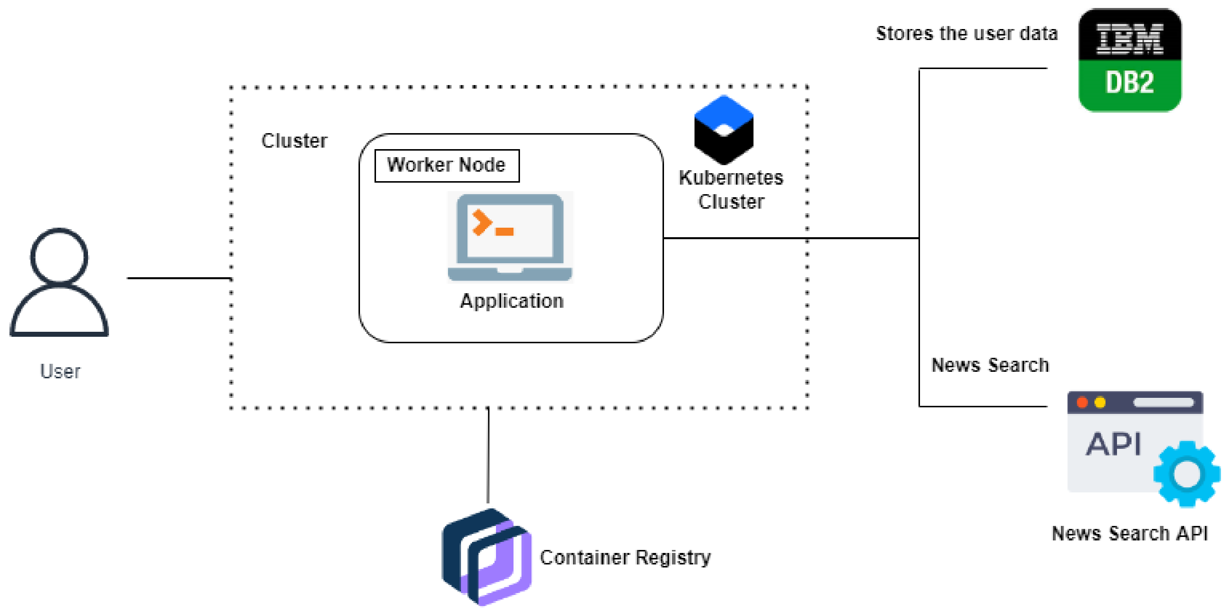
2.2 PROPOSED SOLUTION

As the user can search the news, they can access the news of the particular information what they need to know. It makes the user to do in eco-friendly. They can gain knowledge at the short duration of time.

CHAPTER 3

THEORETICAL ANALYSES

3.1 BLOCK DIAGRAM



3.2 SYSTEM SPECIFICATIONS

HARDWARE REQUIREMENTS

- Windows (minimum 10), Mac OS, Ubuntu
- Ram - 4GB (minimum)
- Hard disk - 256GB (minimum)
- Processor - Intel i3 (minimum), Mac M1

SOFTWARE REQUIREMENTS

- Anaconda Navigator
- VS Code
- Docker
- IBM Cloud

CHAPTER 4

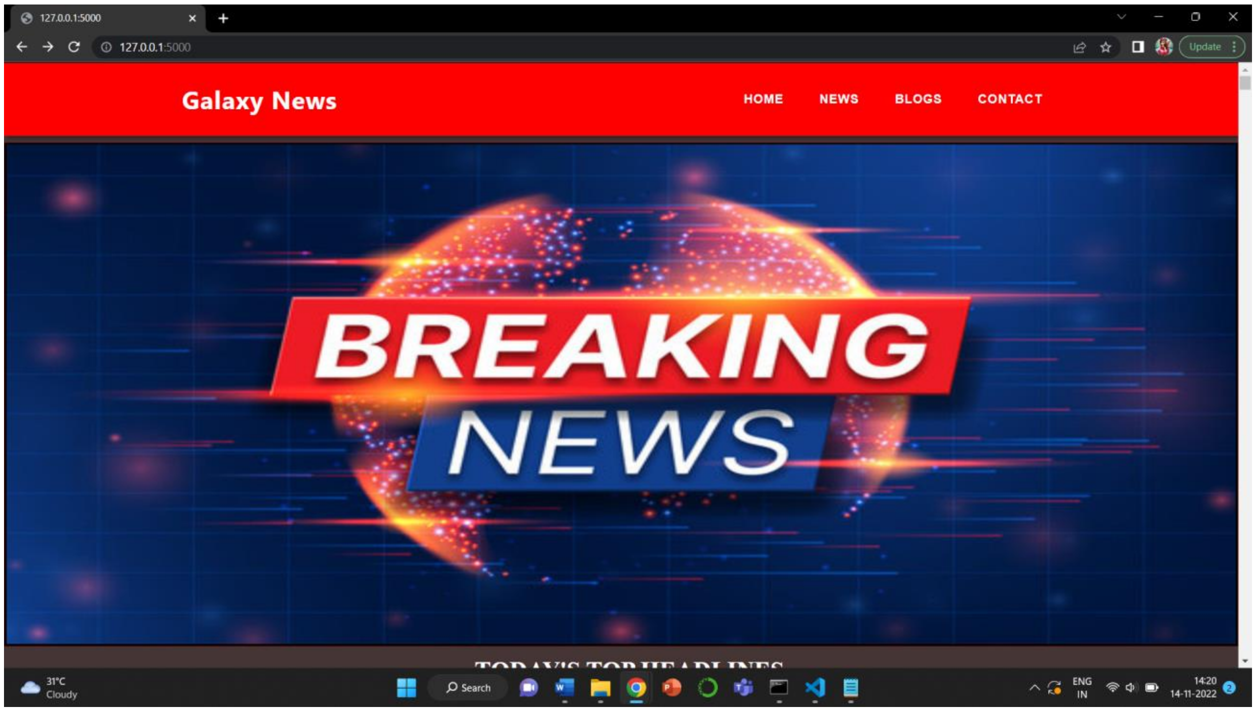
EXPERIMENTAL INVESTIGATIONS

1. Understanding how to insert and fetch data from IBM DB
2. Understanding how to work with APIs - fetch response from various APIs and parse the response they returned (mostly in .json format)
3. Learning how to style using HTML and CSS

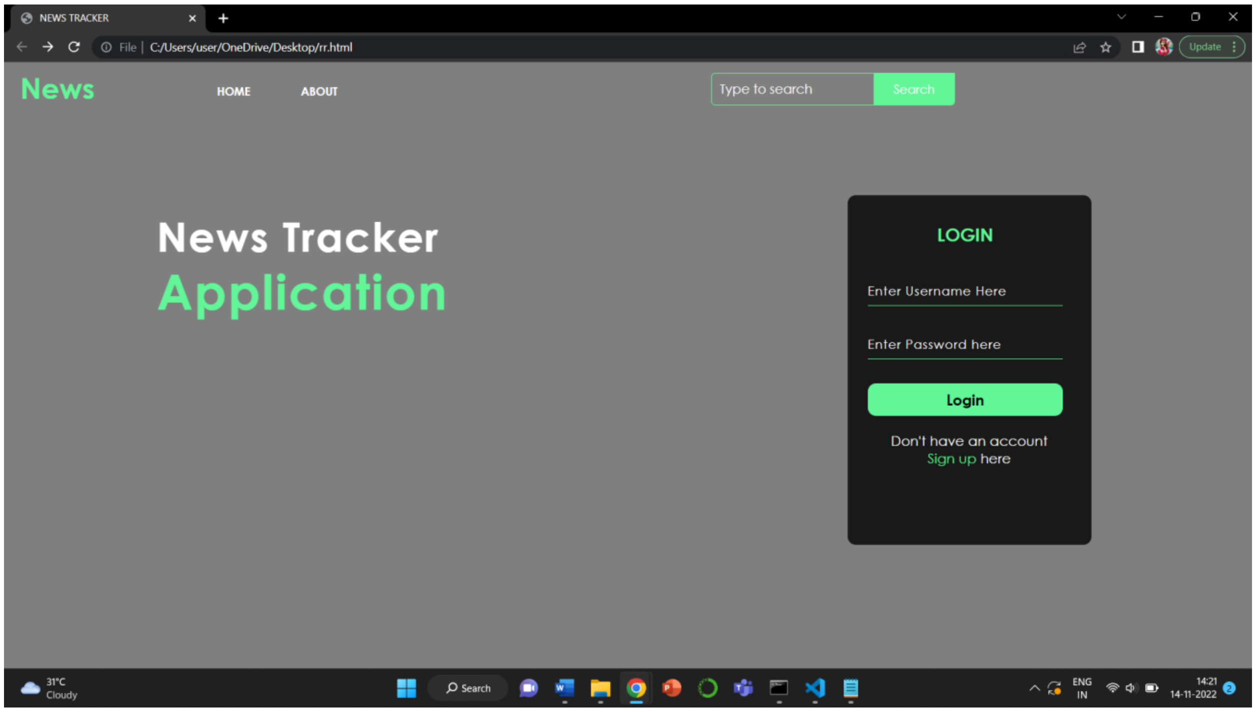
CHAPTER 5

5.1 RESULTS

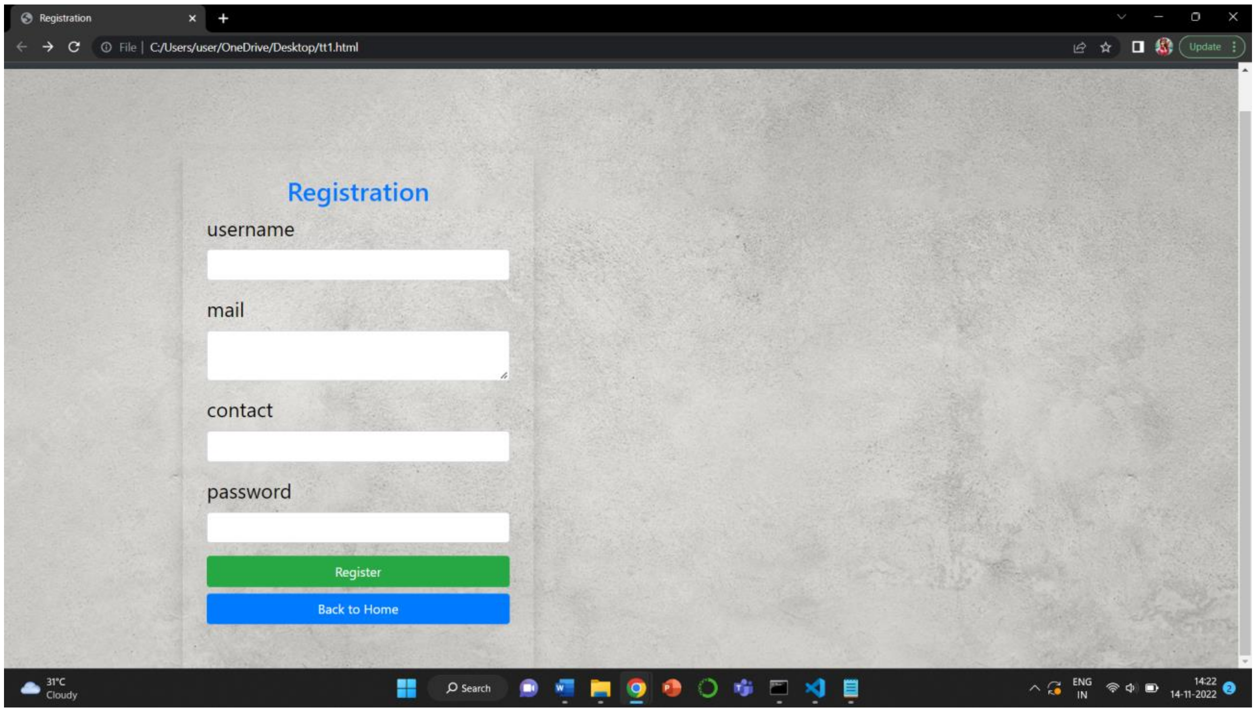
HOME PAGE



LOGIN PAGE



REGISTRATION PAGE



6.2 CONCLUSIONS

This project has been a challenge which gives a specific information. So many information which happen in and around the world. The application gives the real information which the user can access anytime and anywhere. Instead of searching all the news, the user can search the particular which in need.. Besides these constraints the application provides very good information.

6.3 FUTURE SCOPE

Enhancements that can be made in the future:

1. UI Improvement
2. Scaling the application to handle more requests

APPENDIX

SOURCE CODE

```
from flask import Flask,render_template

from newsapi import NewsApiClient



app = Flask(__name__)



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

def login():

    return render_template('index.html')



@app.route("/reg", methods=['GET','POST'])

def reg():

    return render_template('reg.html')



@app.route("/news", methods=['GET','POST'])

def updates():

    api_key = '25560f3cf53c433c9807c60595b373a6'
```

```
newsapi = NewsApiClient(api_key=api_key)


top_headlines = newsapi.get_top_headlines(sources = "bbc-news")

all_articles = newsapi.get_everything(sources = "bbc-news")


t_articles = top_headlines['articles']

a_articles = all_articles['articles']


news = []

desc = []

img = []

p_date = []

url = []


for i in range (len(t_articles)):

    main_article = t_articles[i]


    news.append(main_article['title'])

    desc.append(main_article['description'])
    img.append(main_article['urlToImage'])

    p_date.append(main_article['publishedAt'])

    url.append(main_article['url'])


    contents = zip( news,desc,img,p_date,url)


news_all = []

desc_all = []

img_all = []

p_date_all = []

url_all = []


for j in range(len(a_articles)):

    main_all_articles = a_articles[j]


    news_all.append(main_all_articles['title'])

    desc_all.append(main_all_articles['description'])

    img_all.append(main_all_articles['urlToImage'])

    p_date_all.append(main_all_articles['publishedAt'])
```

```
        url_all.append(main_article['url'])


        all = zip( news_all,desc_all,img_all,p_date_all,url_all)


    return render_template('home.html',contents=contents,all = all)



if __name__ == '__main__':

    app.run(debug=True)
```

BIBLIOGRAPHIES

[1]https://www.w3schools.com/html/ - W3schools html guide

[2]https://www.w3schools.com/w3css/defaulT.asp - W3schools CSS

[3]https://cloud.ibm.com/docs/Db2onCloud?topic=Db2onCloud-getting-started – IBM Cloud DB2 Connection docs