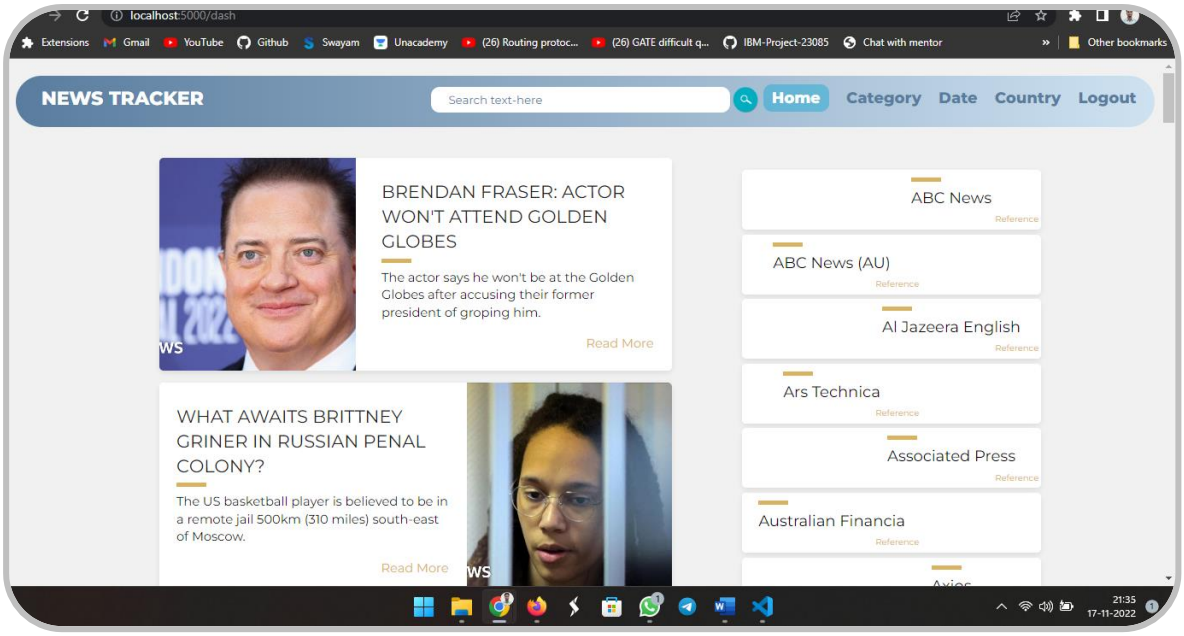
Creation	SPRINT 2
Team ID:	PNT2022TMID06749
Project Name	News Tracker Application

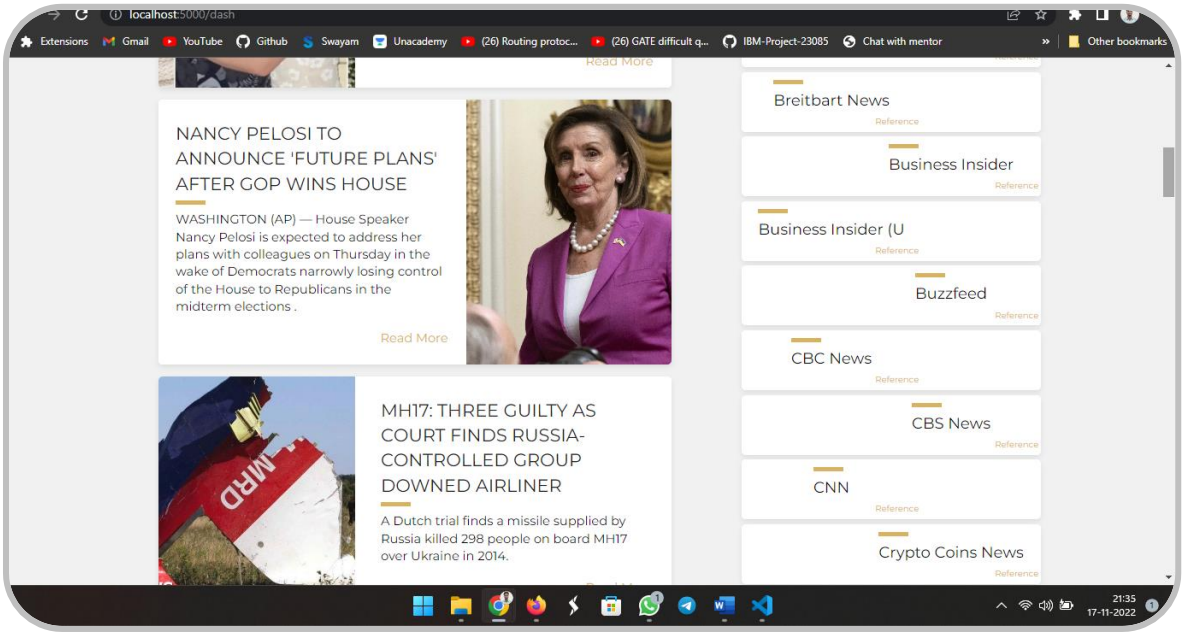
SPRINT-2

Dashboard

Dashboard is used to access the overall top headlines that have been posted by various news media till date and time. Here the user can navigate to his interested news fade by clicking on the “Read More” button and the user will be directed to the news article page from where the source originates.



Moreover, the user also has access to the news channels that are list over the right pane which can be clicked over “Reference” to access the news only from that particular News channel.

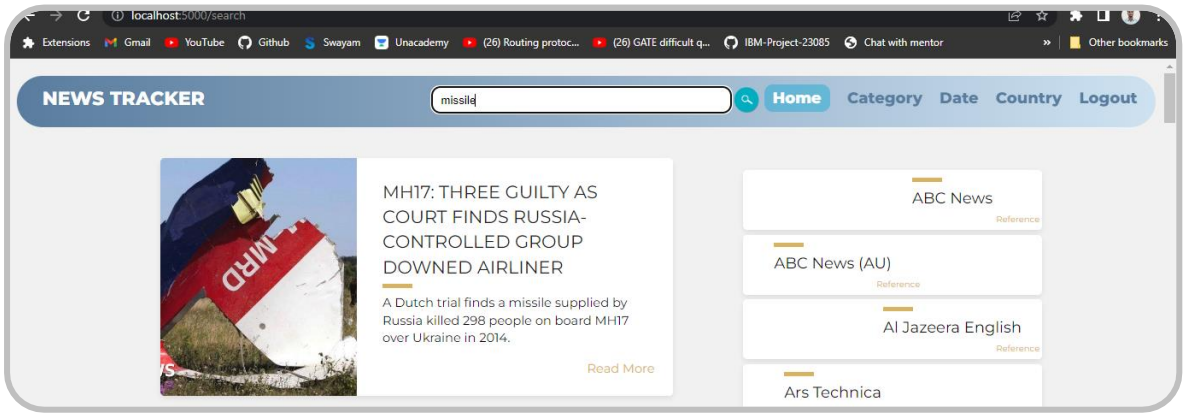


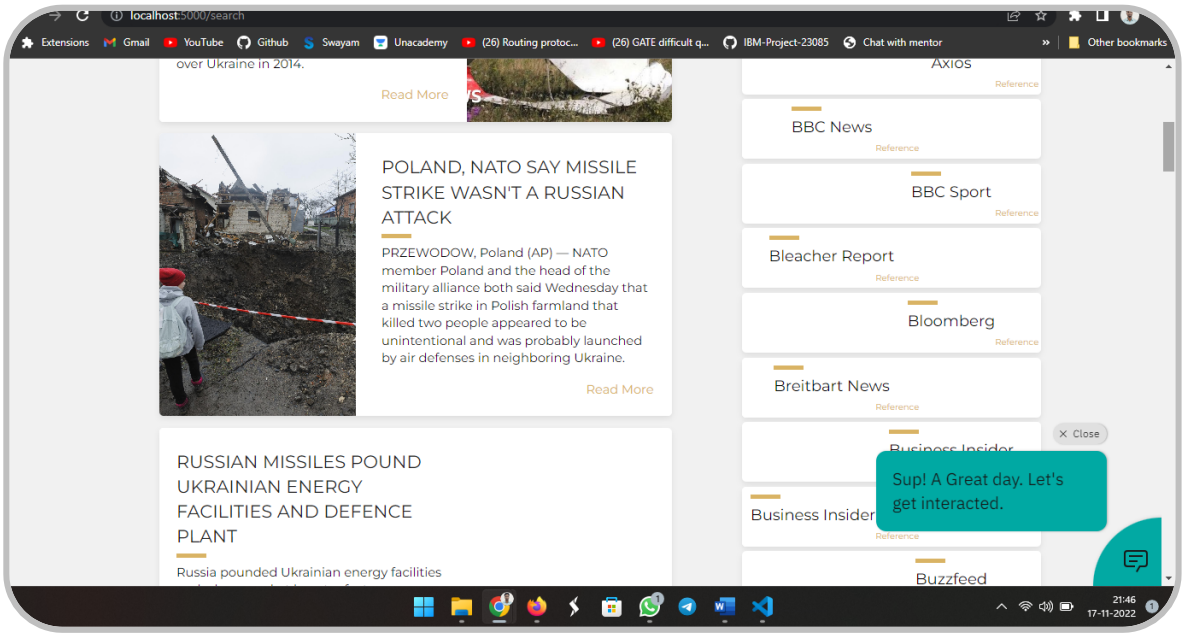
Code for Dashboard:<app.py>

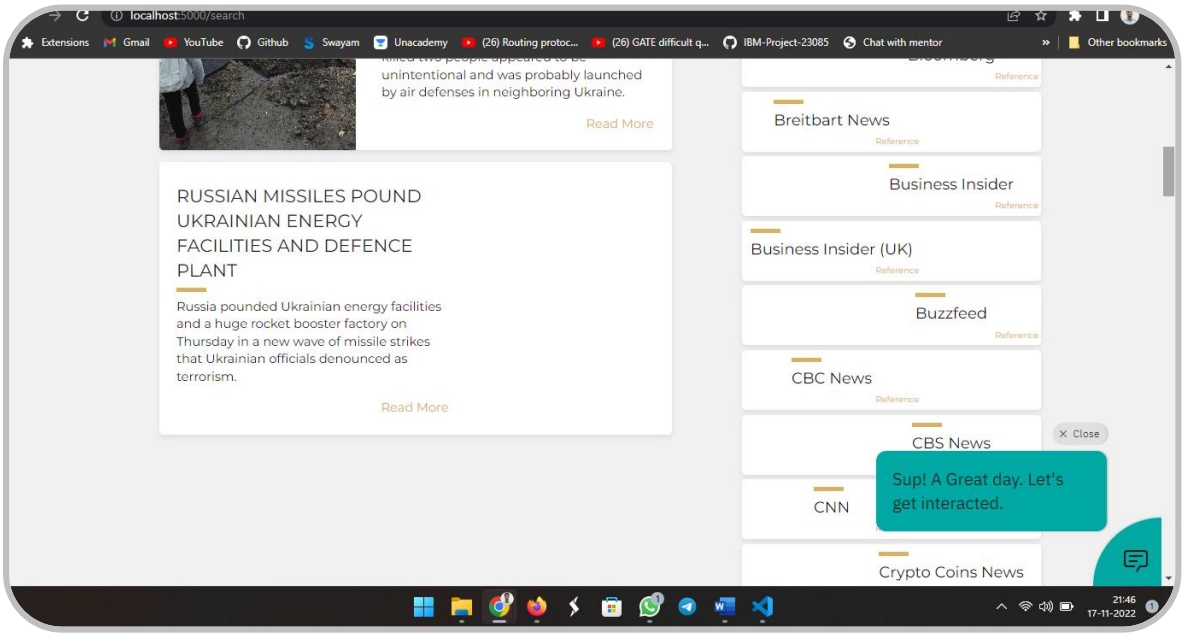
```
@app.route('/dash')
def dash():
    topheadlines=newsapi.get_top_headlines(sources='bbc-news,the-
verge,google-news')
    # from_param="2022-11-08",to="2022-11-09",page=2,
    articles=topheadlines['articles']
    desc=[]
    news=[]
    img=[]
    timing=[]
    more=[]
    auth=[]
    for i in range(len(articles)):
        myarticles=articles[i]
        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        x=myarticles['publishedAt']
        timing.append(x[0:10]+" @ "+x[11:16])
        more.append(myarticles['url'])
        au=myarticles['author']
        if au==None or au[0:3]=="htt":
            au="Media News"
        auth.append(au)

    myList=zip(desc,news,img,timing,more,auth)


    topsources=newsapi.get_sources(language="en")
    articles=topsources['sources']
    desc=[]#category
    name=[]
    url=[]
    id=[]
    # timing=[]
    coun=[]
    for i in range(len(articles)):
        myarticles=articles[i]
        id.append(myarticles['id'])
        name.append(myarticles['name'][0:19])
        desc.append(myarticles['category'].capitalize())
        url.append(myarticles['url'])
        coun.append(myarticles['country'].upper())

    mySources=zip(id,name,desc,url,coun)
    return
render_template('dashboard.html',context=myList,channels=mySources)
```

Search Bar

Search bar as usual is used to search for news over the entered text. The news feed returned as a result has a wide range of news feed over the text entered.







Code for Search Bar :<app.py>

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

    if request.method=='POST':
        try:
            # time=datetime.datetime.now()
            # print("time: "+time)
            topheadlines=newsapi.get_top_headlines(sources='bbc-news,the-
verge,google-news')
            articles=topheadlines['articles']
            desc=[]
            news=[]
            img=[]
            timing=[]
            more=[]
            auth=[]
            ser=request.form['search']
            # print(ser)
            for i in range(len(articles)):
                myarticles=articles[i]

                chck=myarticles['title']
                chck_b=myarticles['description']
                # print(ser+chck)
                if ser.lower() in chck.lower() or ser.lower() in
chck_b.lower():
                    # print("check\n")
                    news.append(chck)
                    desc.append(chck_b)
                    # print(ser)
                    img.append(myarticles['urlToImage'])
                    x=myarticles['publishedAt']
                    timing.append(x[0:10]+" @ "+x[11:16])
                    more.append(myarticles['url'])
                    au=myarticles['author']
                    if au==None or au[0:3]=="htt":
                        au="Media News"
                    auth.append(au)
            myList=zip(desc,news,img,timing,more,auth)

            topsources=newsapi.get_sources(language="en")
            articles=topsources['sources']
            desc=[]#category
            name=[]
            url=[]
            # timing=[]
            coun=[]
```

```
            id=[]
            for i in range(len(articles)):
                myarticles=articles[i]
                id.append(myarticles['id'])
                name.append(myarticles['name'])
                desc.append(myarticles['category'].capitalize())
                url.append(myarticles['url'])
                coun.append(myarticles['country'].upper())

            mySources=zip(id,name,desc,url,coun)

            return
render_template('dashboard.html',context=myList,channels=mySources)

        except Exception as e:
             print("Exception : "+str(e))
```