

A
Project Report On
NEWS CATEGORIZATION SYSTEM
Using
MULTINOMIAL NAIVE BAYES ALGORITHM

B.Tech (CE) Sem-VI
Subject :- SYSTEM DESIGN PRACTICE (CE-621)

By:-

Ronak Padaliya	CE096	19CEUEG041
Bhumit Navadiya	CE090	19CEUOS044
Visrut Navadiya	CE091	19CEUES029

Under the Guidance of
Prof. Apurva A. Mehta
Assistant Professor
Dept. of Comp. Engg.



Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University, Nadiad

DHARMSINH DESAI UNIVERSITY

College Road, NADIAD-387001 (Gujarat)



Certificate

This is to certify that the term work carried out in the subject of **System Design Practice** and recorded in this report is the bonafide work of

Ronak Padaliya	CE096	19CEUEG041
Bhumit Navadiya	CE090	19CEUOS044
Visrut Navadiya	CE091	19CEUES029

of **B.Tech Semester 6th** in the branch of Computer Engineering during the academic year 2021-22.

Prof. Apurva A. Mehta
Assistant Professor,
Dept. of Computer Engg.
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia
Head of the Department,
Dept. of Computer Engg.
Faculty of Technology
Dharmsinh Desai University, Nadiad

Index

1. Abstract.....	1
2. Introduction.....	2
2.1 Brief Introduction.....	2
2.2 Technologies/Tools/Platforms used.....	2
3. Software Requirement Specifications.....	4
3.1 Project Scope.....	4
3.2 Types of User.....	4
3.3 System Functional Requirements.....	4
3.4 Other Non-Functional Requirements.....	5
4. Design Documents.....	6
4.1 Use Case Diagram.....	6
4.2 Class Diagram.....	7
4.3 State Diagram.....	7
4.4 Activity Diagram.....	8
4.5 Sequence Diagram.....	9
5. Implements Details.....	10
5.1 Use of Dataset.....	10
5.2 Text Cleaning/Preprocessing.....	11
5.2.1 Remove Tags.....	11
5.2.2 Removing Punctuations and Special Characters.....	12
5.2.3 Convert to lower cases.....	12
5.2.4 Filtering Stop Word.....	13
5.2.5 Lemmatize word.....	13
5.3 Class Representation.....	14
5.4 News Prediction Module.....	15
5.5 Front-End Design Implementation Detail.....	16
5.6 Back-End Design Implementation Detail.....	18

5.7 News Prediction model Implementation.....	19
6. Testing.....	20
6.1 Testing Method Used.....	20
6.2 Test Cases.....	20
7. ScreenShots.....	21
7.1 Home Screen.....	21
7.2 Search news based on category.....	23
7.3 Explore News.....	24
7.4 News Prediction.....	25
7.5 About Page.....	26
7.6 Contact Page.....	27
8. Conclusion.....	28
9. Limitations and Future Extensions.....	29
9.1 Limitations.....	29
9.2 Future Extensions.....	29
10. Bibliography.....	30
10.1 Reference use for develop web application.....	30
10.2 Tools user for develop web application.....	30

List of Figures

Figure 4.1 Use Case Diagram of News Categorization System.....	6
Figure 4.2 Class Diagram of News Prediction.....	7
Figure 4.3 State Diagram.....	7
Figure 4.4 Activity Diagram.....	8
Figure 4.5 Sequence Diagram.....	9
Figure 5.1 Daset and shape of dataset.....	10
Figure 5.2 Remove Tags From dataset.....	11
Figure 5.3 Removing Punctuations and Special Characters from dataset.....	12
Figure 5.4 Convert dataset news to lower cases.....	12
Figure 5.5 Filtering Stop Word from dataset.....	13
Figure 5.6 Lemmatize word of dataset.....	13
Figure 5.7 Use of Label encoding on dataset.....	15
Figure 5.8 React App folder structure.....	16
Figure 5.9 Django App folder structure.....	18
Figure 5.10 News Prediction Method.....	19
Figure 7.1 Loading Screen.....	21
Figure 7.2 Home Page Navbar and Carousel.....	21
Figure 7.3 Home Page with latest news.....	22
Figure 7.4 Mobile View of home page.....	22
Figure 7.5 Mobile view responsive navbar.....	22
Figure 7.6 Dropdown for search news by category.....	23
Figure 7.7 Page view related to search category.....	23
Figure 7.8 Page view of news in more details.....	24
Figure 7.9 News related to news that open for more details.....	24
Figure 7.10 News prediction page.....	25
Figure 7.11 Pop-up for showing news category.....	25
Figure 7.12 About page - top.....	26

Figure 7.13 About page - bottom.....	26
Figure 7.14 Contact page.....	27

1. Abstract

Now a days News Categorization System is very needed for categorise the news to some category, so that people can easily search a news based on particular category.

In this project we make a such type of system. User can enter a news description and System gives the category in which news is lying.

For predict news category we make a Django Backend and React.js Frontend. Which have trained model on news dataset. We make API call to django application which provide us a Predicted Category in result.

We also Gather data of news from public API (currentsapi.services/en). It makes our system attractive.

2. Introduction

2.1 Brief Introduction

News Categorization System is application which is used for predict category of particular news. We do this task using Machine Learning model which is trained using Multinomial Naive Bayes Algorithm. So, our main focus on train Machine Learning Model using various Algorithms. So that it is useful to guess particular News Category.

This system predict news category based on model trained using dataset.

2.2 Technologies/Tools/Platforms used :

→ Django for backend:

Django is a open source python web framework used for rapid development, maintainable, pragmatic ,secure and clean design.

→ React.js for Frontend:

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

→ Github:

This is platform is used were more than one user can develop Project and share using concurrently.

→ Programing Language:

Python, Javascript

➔ Visual Studio Code:

Used to develop and maintain projects.

➔ Python library used:

Numpy, Pandas, nltk, sklearn.

3. Software Requirement Specifications

3.1 Project Scope

The System is designed to perform News Prediction task. Scope of the system is global and open for all users. System provides various functionalities to the users like predict news and explore news.

3.2 Types of User

(1)User:- Which is use this system for predict news category and explore News.

3.3 System Functional Requirements:-

R1. News Prediction:-

Description:- This is used for predict news category. System first get news from user after that it predict news category based on Machine Learning Model which is trained.

Input:- News Description

Output:- Predicted category

3.4 Other Non-Functional Requirements:-

1) Performance:

The application should run efficiently. It must be interactive and user friendly in nature.

2) Reliability:

The application must ensure that the system is reliable in its news prediction operation.

4. Design Documents

4.1 Use Case Diagram :-

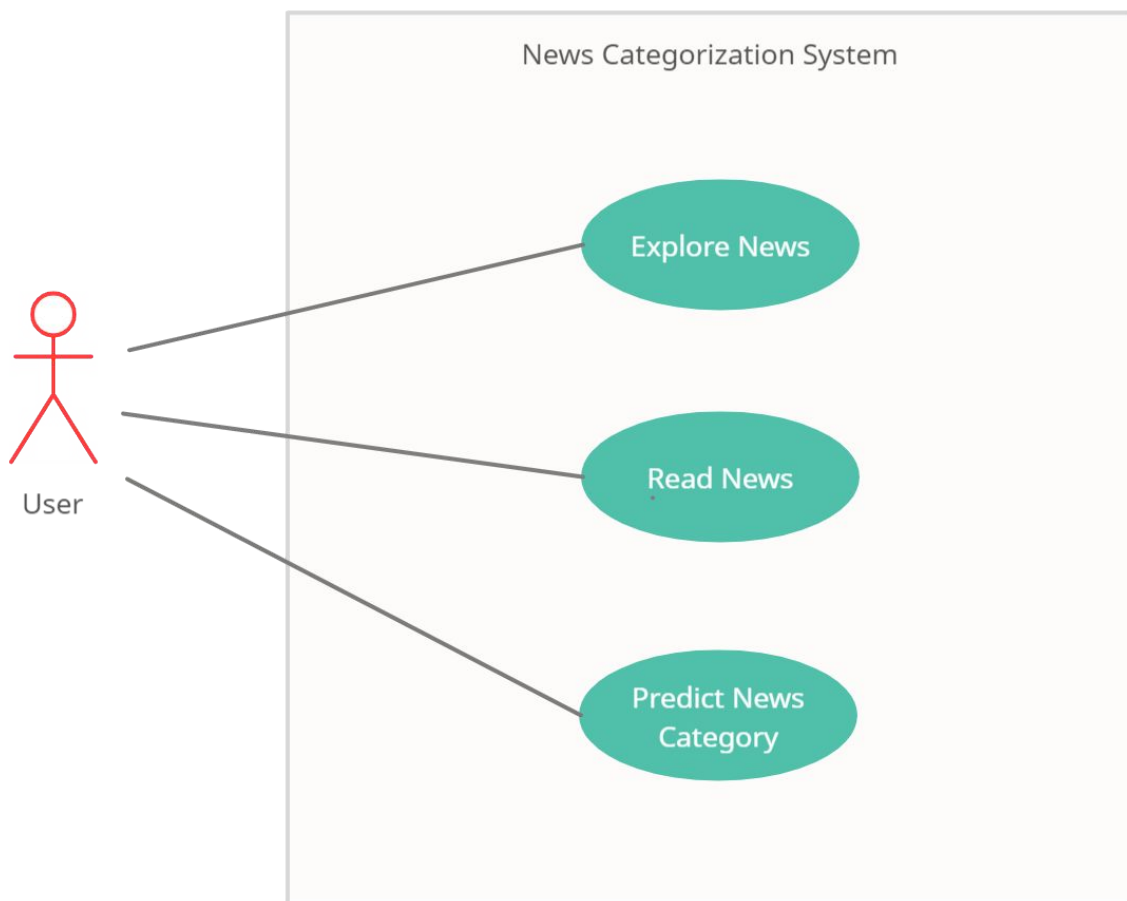


Figure 4.1 Use Case Diagram of News Categorization System

4.2 Class Diagram:-

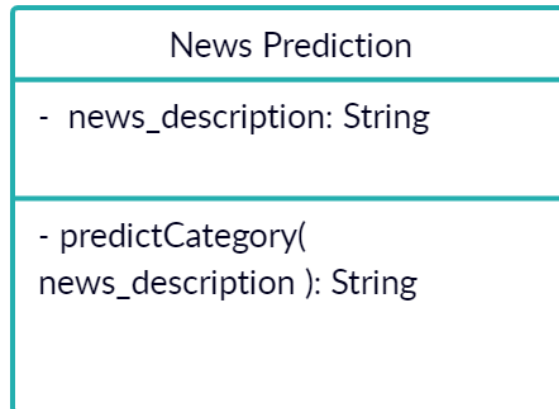


Figure 4.2 Class Diagram of News Prediction

4.3 State Diagram:-

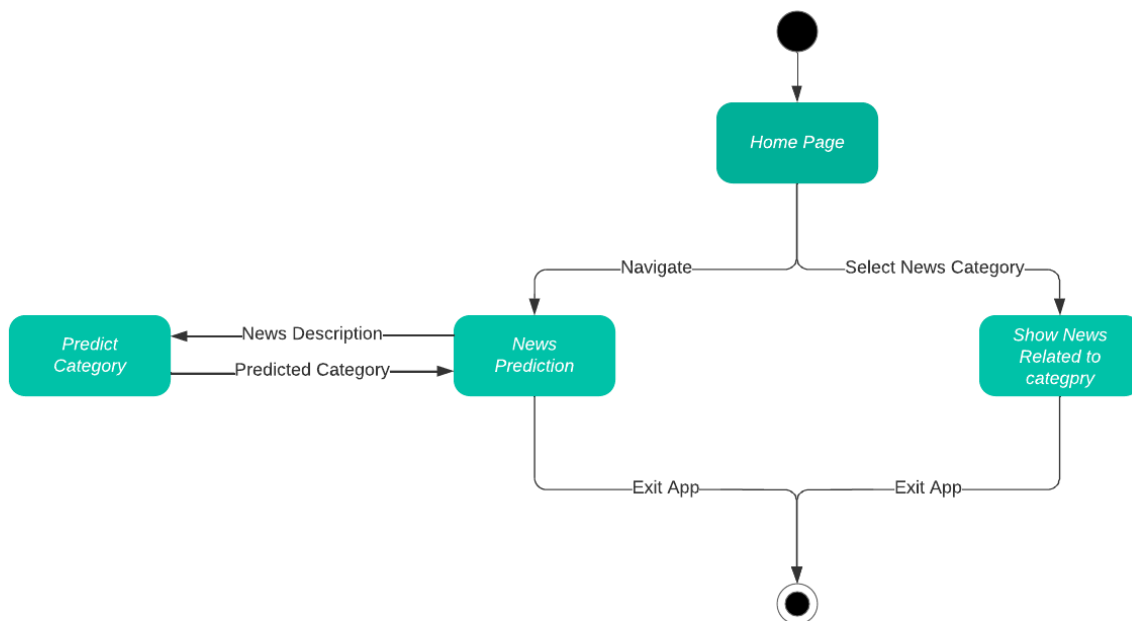


Figure 4.3 State Diagram

4.4 Activity Diagram:-

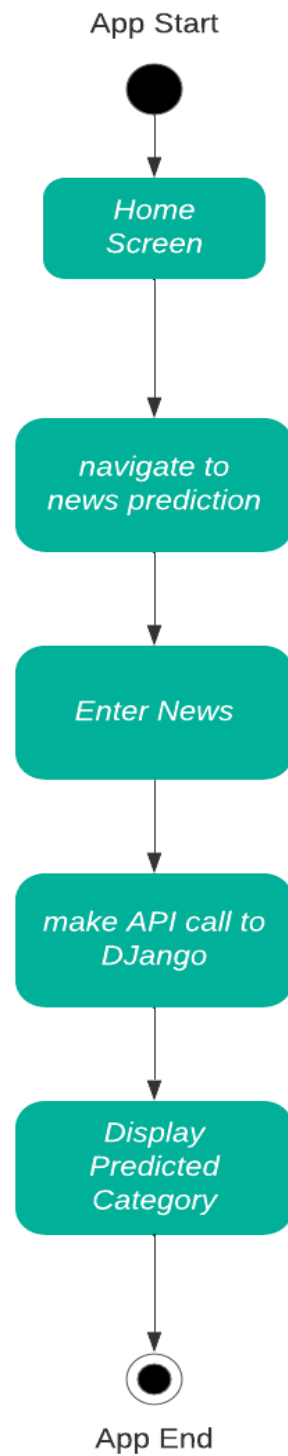


Figure 4.4 Activity Diagram

4.5 Sequence Diagram:-

→ News Prediction

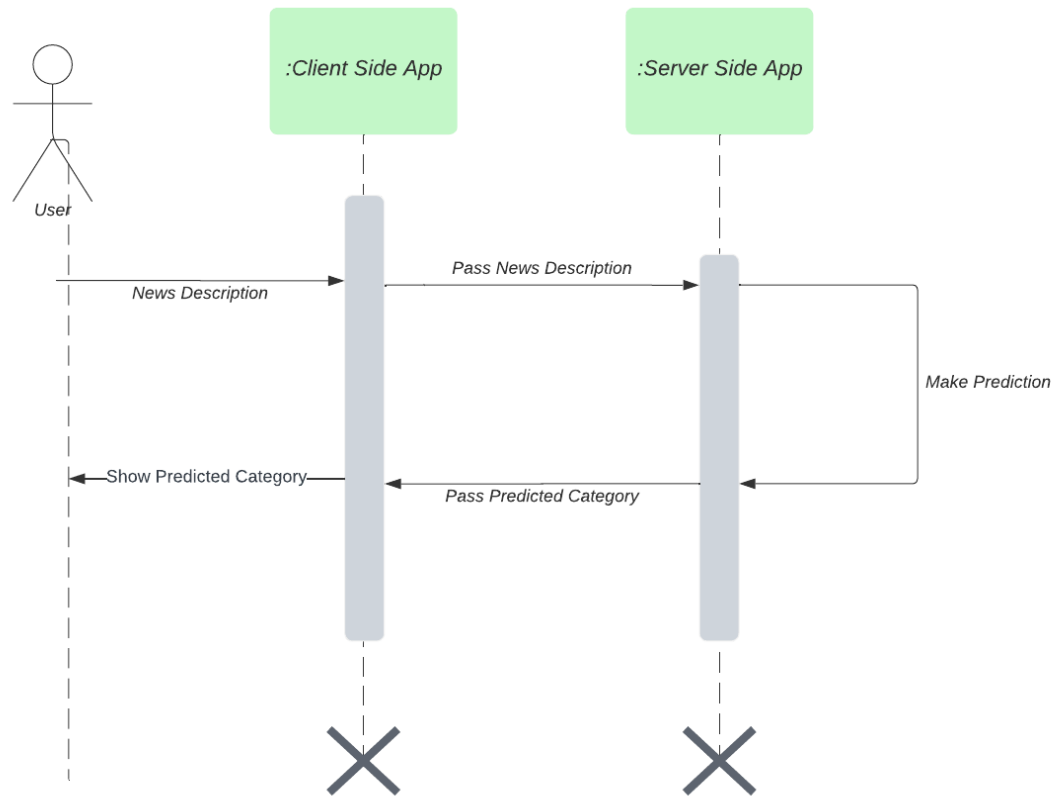


Figure 4.5 Sequence Diagram

5. Implements Details

5.1 Use of dataset

In this project, a news data set was used, which is obtained from Kaggle. It consists of 2 lacks rows but we use only **50,000** rows because the out laptop system not sufficient create ML model. In this news corresponding to stories in 10 typical areas: Wellness, politics, entertainment, travel, style & beauty, parenting, food & drink, world, business, and sports. The distribution of classes plays an important role in classification, and balanced datasets result in better learning models. In this study, the dataset is broken into (80%) records for training and (20%) testing.

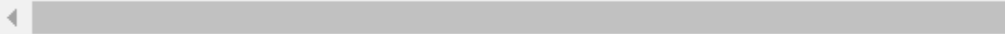
	Category	Text
0	WELLNESS	143 Miles in 35 Days: Lessons LearnedResting i...
1	WELLNESS	Talking to Yourself: Crazy or Crazy Helpful?Th...
2	WELLNESS	Crenezumab: Trial Will Gauge Whether Alzheimer...
3	WELLNESS	Oh, What a Difference She Madelf you want to b...
4	WELLNESS	Green SuperfoodsFirst, the bad news: Soda brea...
		
<pre>[21] print("shape :- ", dataset.shape, "\n\n") dataset.info() shape :- (50000, 2)</pre>		

Figure 5.1 Daset and shape of dataset

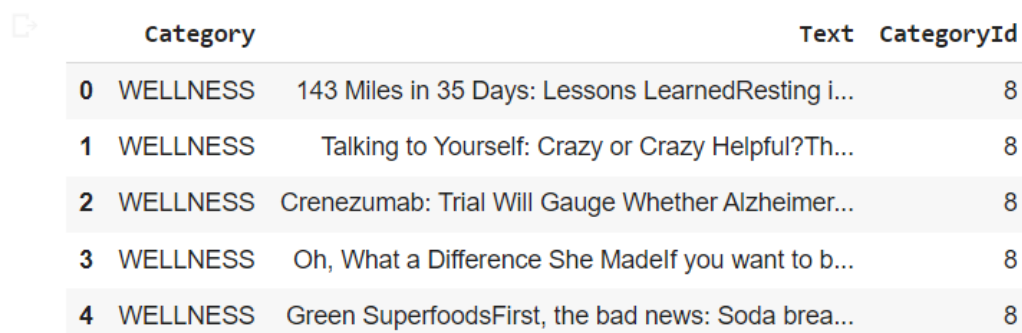
5.2 Text Cleaning/Preprocessing

Text preprocessing or cleaning is a preliminary and crucial step of news classification, which reduces the required space and makes the classification more efficient. Most of the times, the dataset is unstructured in combinations of useful and useless data. Unnecessary information such as stop words, punctuations, special characters, irrelevant sentences, quotations, and dates do not add any predictive power to the classifier/model. They only consume space and can distort the ML model. Therefore, before extracting any feature from the raw dataset, a cleaning process should be performed to minimise the distortions introduced to the model. In this, several steps have been followed to preprocess the news text.

5.2.1 Remove Tags

In our dataset we first remove tags. So that it is not affect out model accuracy and create proper model.

```
[28] def remove_tags(text):  
      remove = re.compile(r'<.*>')  
      return re.sub(remove, '', text)  
  
dataset['Text'] = dataset['Text'].apply(remove_tags)  
  
dataset.head()
```



	Category	Text	CategoryId
0	WELLNESS	143 Miles in 35 Days: Lessons LearnedResting i...	8
1	WELLNESS	Talking to Yourself: Crazy or Crazy Helpful?Th...	8
2	WELLNESS	Crenezumab: Trial Will Gauge Whether Alzheimer...	8
3	WELLNESS	Oh, What a Difference She Madelf you want to b...	8
4	WELLNESS	Green SuperfoodsFirst, the bad news: Soda brea...	8

Figure 5.2 Remove Tags From dataset

5.2.2 Removing Punctuations and Special Characters

Characters such as ?, !, ; and . are disposed of, this process simplifies computations in the next steps. Any special character and unnecessary whitespaces are also removed because they don't contribute to prediction power.

```
[29] def special_char(text):  
    reviews = ''  
    for x in text:  
        if x.isalnum():  
            reviews = reviews + x  
        else:  
            reviews = reviews + ' '  
    return reviews  
dataset['Text'] = dataset['Text'].apply(special_char)  
  
dataset.head()
```

	Category	Text	CategoryId
0	WELLNESS	143 Miles in 35 Days Lessons LearnedResting i...	8
1	WELLNESS	Talking to Yourself Crazy or Crazy Helpful Th...	8
2	WELLNESS	Crenezumab Trial Will Gauge Whether Alzheimer...	8
3	WELLNESS	Oh What a Difference She Madelf you want to b...	8
4	WELLNESS	Green SuperfoodsFirst the bad news Soda brea...	8

Figure 5.3 Removing Punctuations and Special Characters from dataset

5.2.3 Convert to lower cases

Some of the character in upper and some of this are lower case so we can convert all characters to lower case. It helps to get rid of unhelpful parts of the data, or noise.

```
def convert_lower(text):  
    return text.lower()  
dataset['Text'] = dataset['Text'].apply(convert_lower)  
dataset
```

	Category	Text	CategoryId
0	WELLNESS	143 miles in 35 days lessons learnedresting i...	8
1	WELLNESS	talking to yourself crazy or crazy helpful th...	8
2	WELLNESS	crenezumab trial will gauge whether alzheimer...	8
3	WELLNESS	oh what a difference she madeif you want to b...	8
4	WELLNESS	green superfoodsfirst the bad news soda brea...	8

Figure 5.4 Convert dataset news to lower cases

5.2.4 Filtering Stop Word

This technique is mainly used to remove unnecessary words or words with no specific meaning, such as “the”, “an”, “a”, “what”, etc. so that classifier cannot co-relate stop words and important class features. Furthermore, the most frequent or rarely used words do not contribute to the predictive power model. Therefore, they must be removed from the training set. In this study, we have downloaded a list of English stop words from the nltk library and then removed them from the dataset.

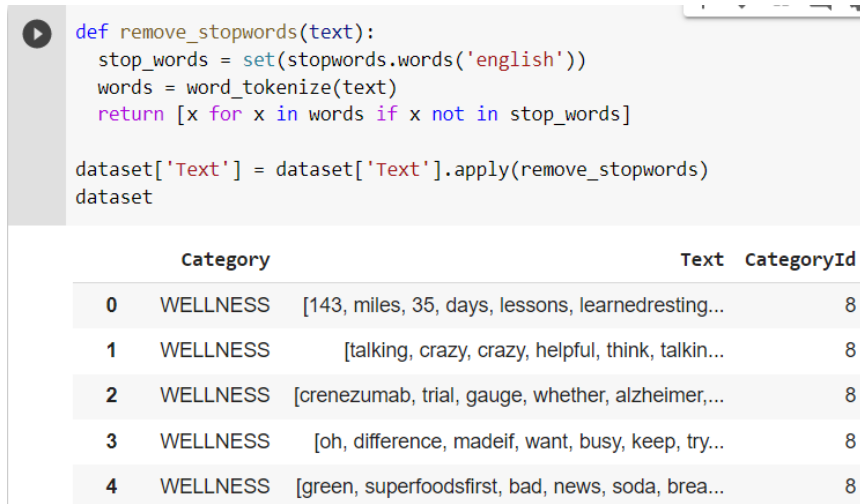


Figure 5.5 Filtering Stop Word from dataset

5.2.5 Lemmatize word

After removing stop words the text of news converted into array of words. So, we need to lemmatize this and make again string which contain all related words of that news.

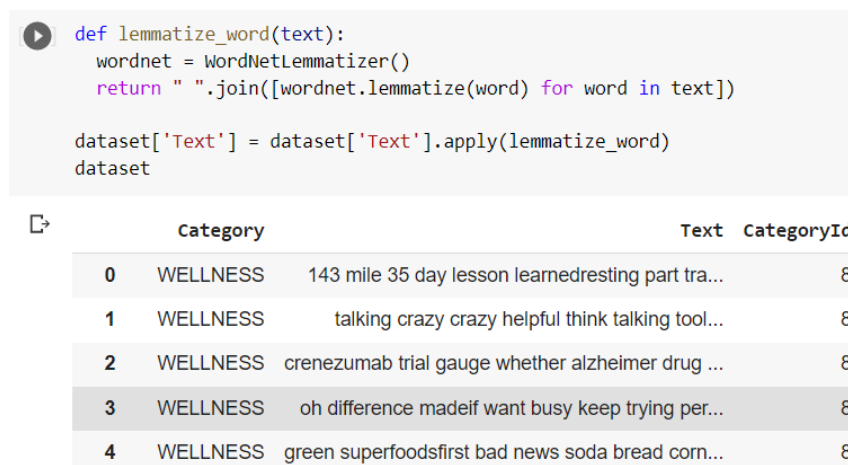


Figure 5.6 Lemmatize words of dataset

5.3 Class Representation

News category prediction is multi-class classification. For instance, the dataset used in this study corresponds to 10 classes: Wellness, politics, entertainment, travel, style & beauty, parenting, food & drink, world, business, and sports. Each class is labelled to make it more understandable. For ML models, label encoding is used to transform labels into numeric values. It can be done by a Label Encoder, which converts class labels into values between 0 and $n-1$, (\because n = total number of unique class labels).

The actual and encoded labels of the dataset used in this study are given in Table.

Category	CategoryId
Business	0
ENTERTAINMENT	1
FOOD & DRINK	2
PARENTING	3
POLITICS	4
SPORTS	5
STYLE & BEAUTY	6
TRAVEL	7
WELLNESS	8
WORLD NEWS	9

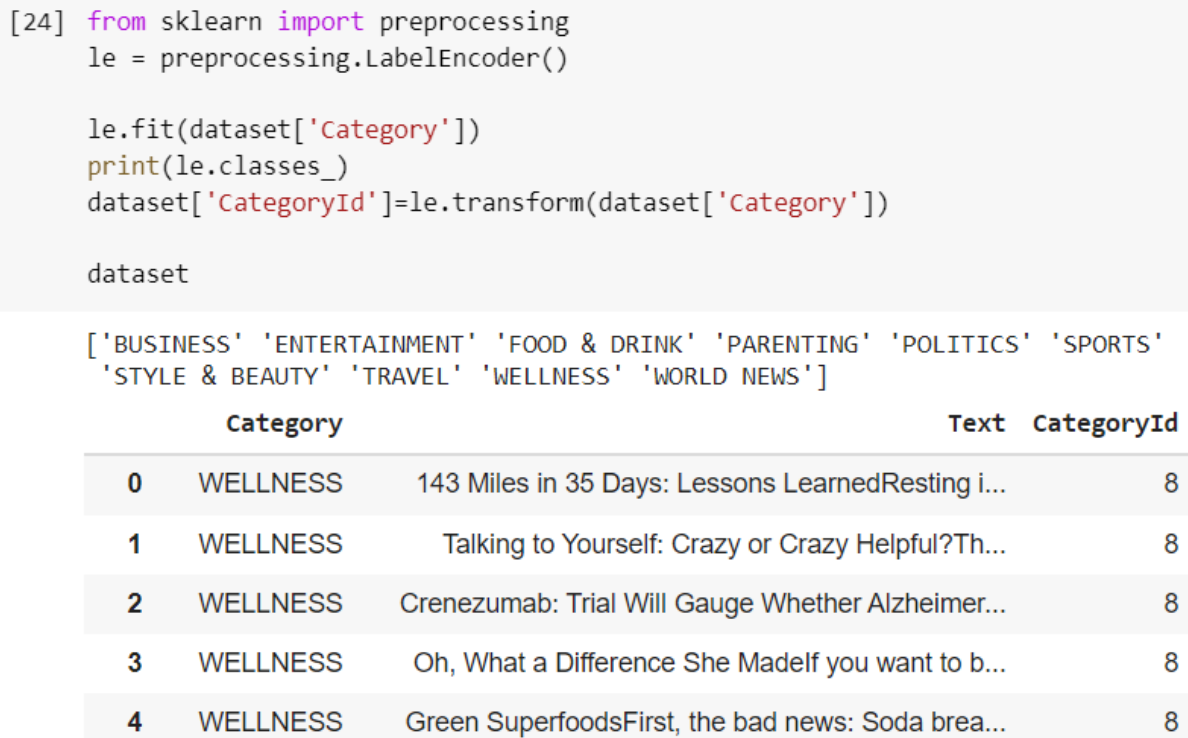


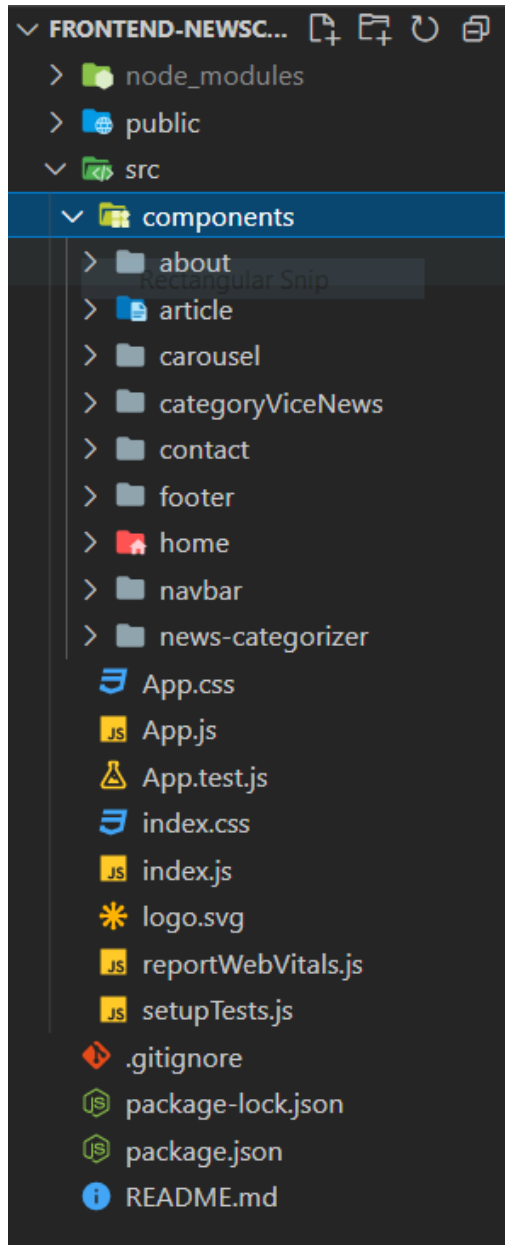
Figure 5.7 Use of Label encoding on dataset

5.4 News Prediction Module

In this module we tack news description from user and pass it to django Server. It will predict news category and get predicted category. Which is shown to user From React Client side application.

5.5 Front-End Design Implementation Detail:-

Frontend design in React.Js which is use javascript as a main programming language. Package.json include necessary packages which project needed.



public :-

This folder contain all public item like images, indedx.html, etc...

Components:-

This Folder contain all components which is needed to use.

App.js:-

This file is a root component to all component.

Package.json:-

This file contain all information related to libraries which is needed to application.

**Figure 5.8 React App
folder structure**

We have basically 4 types of view,

(1)Home View:-

Which contain carousel for news, responsive navbar, and near about 200 news which is fetch from currentsapi.services/en public API.

(2)Search News by category view:-

Based on user select category fetch news related to that category and shown to user.

(3)Explore news view:-

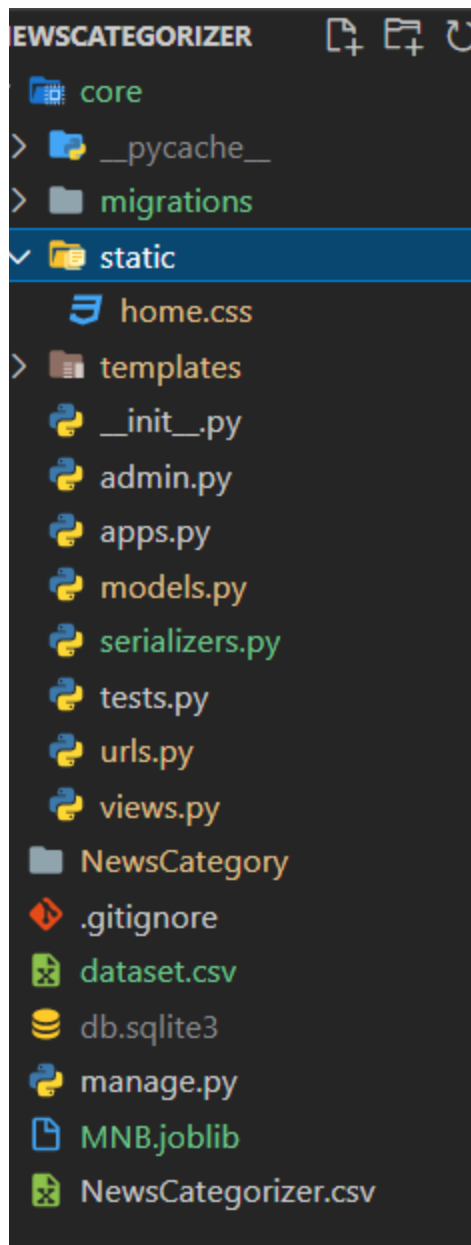
If user want to know more about to news user can simply click on news and redirected to it's detail view section. And based on user selected news category system show related news to that category.

(4)News Prediction View:-

Here system provide a textarea box where user can enter news description. After submit this news backend server give response with predicted category. Which is shown after to user.

5.6 Back-End Design Implementation Detail:-

We create backend in Django which use python as a main programming language.



NewsCategory:-

This folder is the main project folder.

Core:-

This folder is an application folder. Which is created under project **NewsCategory**.

Static:-

This folder contains all static files.

Dataset.csv:-

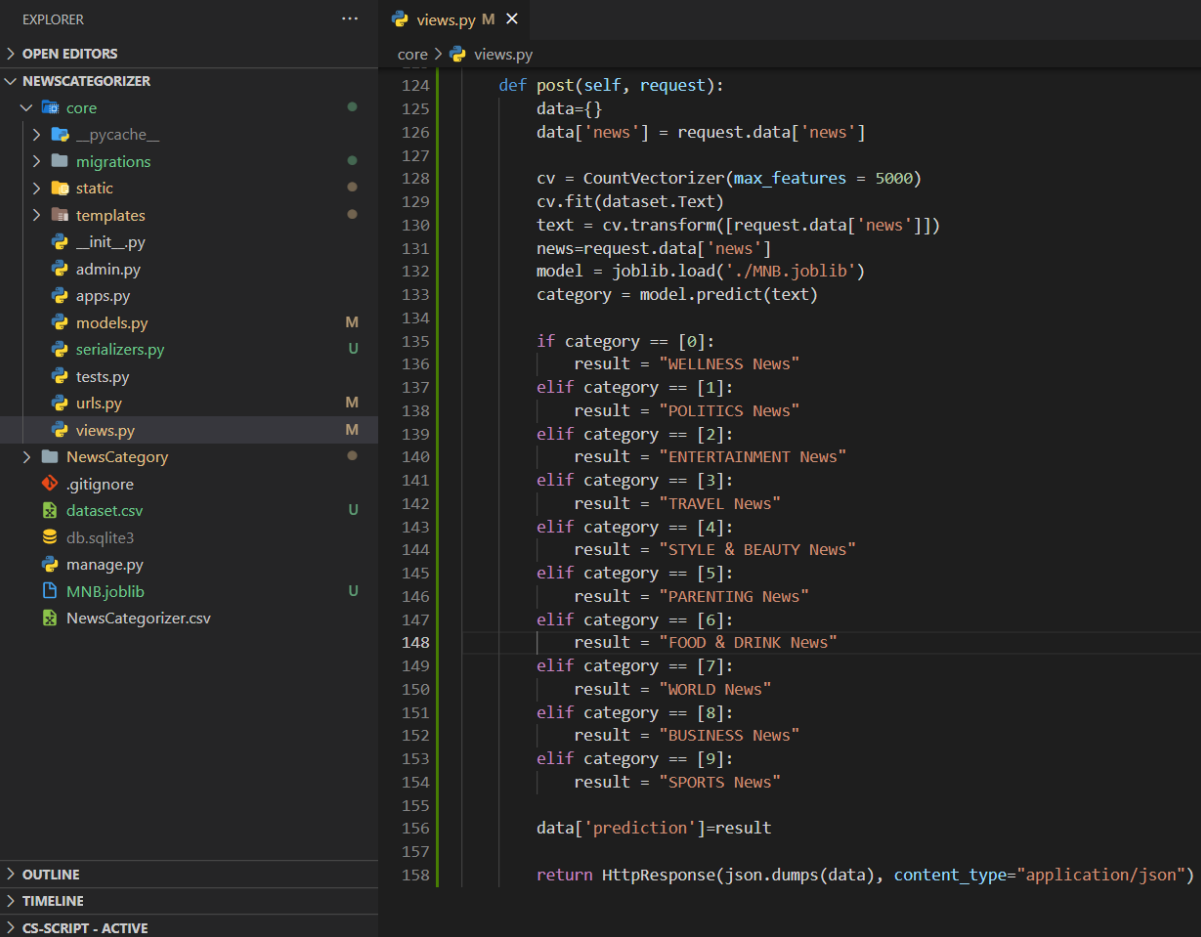
Subpart of NewsCategorizer.csv. Which contain near about 50,000 Records of news.

MNB.joblib:-

This is a trained Machine Learning model using the Multinomial Naive Bayes algorithm.

Figure 5.9 Django App folder structure

5.7 News Prediction model Implementation



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'core' directory containing 'views.py'. The code editor shows the implementation of a 'post' method in a Django view. The method takes a request and returns a JSON response with a predicted news category. The prediction is based on a pre-trained model loaded from 'MNB.joblib'.

```
124 def post(self, request):
125     data={}
126     data['news'] = request.data['news']
127
128     cv = CountVectorizer(max_features = 5000)
129     cv.fit(dataset.Text)
130     text = cv.transform([request.data['news']])
131     news=request.data['news']
132     model = joblib.load('./MNB.joblib')
133     category = model.predict(text)
134
135     if category == [0]:
136         result = "WELLNESS News"
137     elif category == [1]:
138         result = "POLITICS News"
139     elif category == [2]:
140         result = "ENTERTAINMENT News"
141     elif category == [3]:
142         result = "TRAVEL News"
143     elif category == [4]:
144         result = "STYLE & BEAUTY News"
145     elif category == [5]:
146         result = "PARENTING News"
147     elif category == [6]:
148         result = "FOOD & DRINK News"
149     elif category == [7]:
150         result = "WORLD News"
151     elif category == [8]:
152         result = "BUSINESS News"
153     elif category == [9]:
154         result = "SPORTS News"
155
156     data['prediction']=result
157
158     return HttpResponse(json.dumps(data), content_type="application/json")
```

Figure 5.10 News Prediction Method

This method work on **POST** request. It will predict the news category by loading the **MNB.joblib** file. The total prediction category is 10.

6. Testing

6.1 Testing Method Used:-

For testing purpose, we have used black box testing method.

For black box testing, we have designed the test cases and have tested it in our application. Also, we have observed the output and note down the results in the last section.

6.2 Test Cases:-

6.2.1 For News Prediction

Test Case ID	Test Data (News Description)	Expected output	Actual Output	Pass / Fail
1	As youngsters flock to physiotherapy centers, experts reveal how one can avoid the pain.	WELLNESS	WELLNESS	Pass
2	Twitter is finally working on an edit button, after years of calls asking for one. The company has only shared a few details of how it works.	BUSINESS	BUSINESS	Pass
3	Congress leader Salman Nizami calls 'The Kashmir Files' actor Anupam Kher an 'Islamophobic', 'bigot' after he wishes Ramadan Mubarak.	POLITICS	POLITICS	Pass
4	From Madison Square Garden to 'Howdy Modi': A look back at Narendra Modi's visits to the US	POLITICS	FOOD & DRINK	Fail

7. ScreenShots

7.1 Home Screen

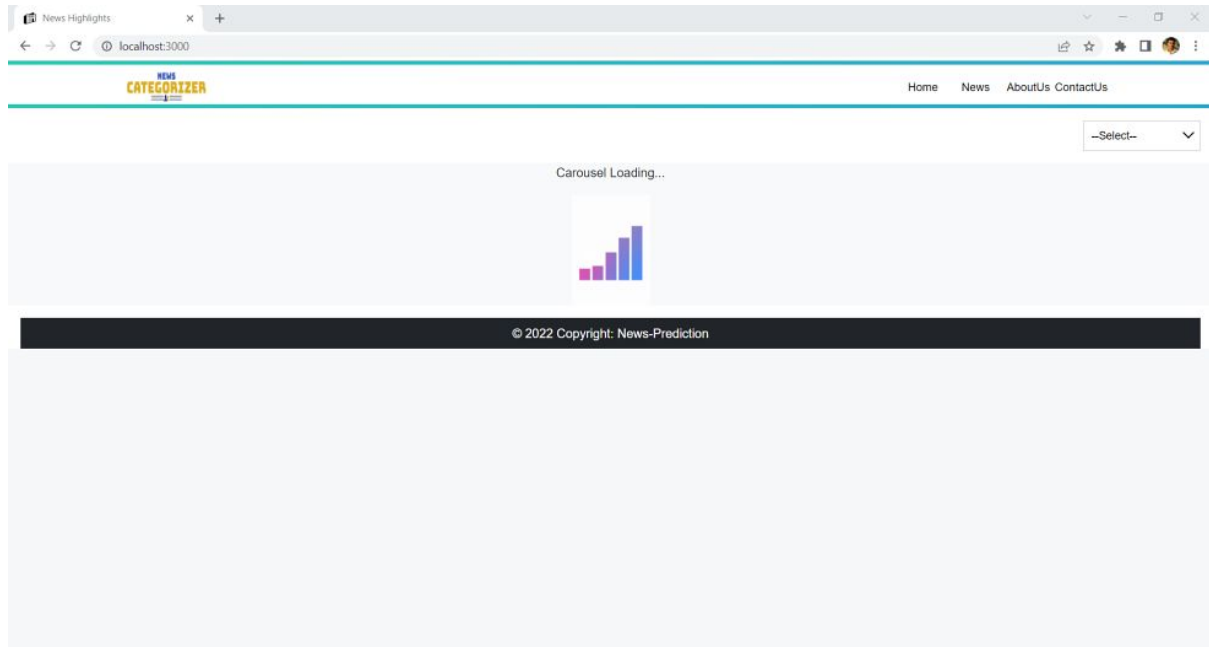


Figure 7.1 Loading Screen

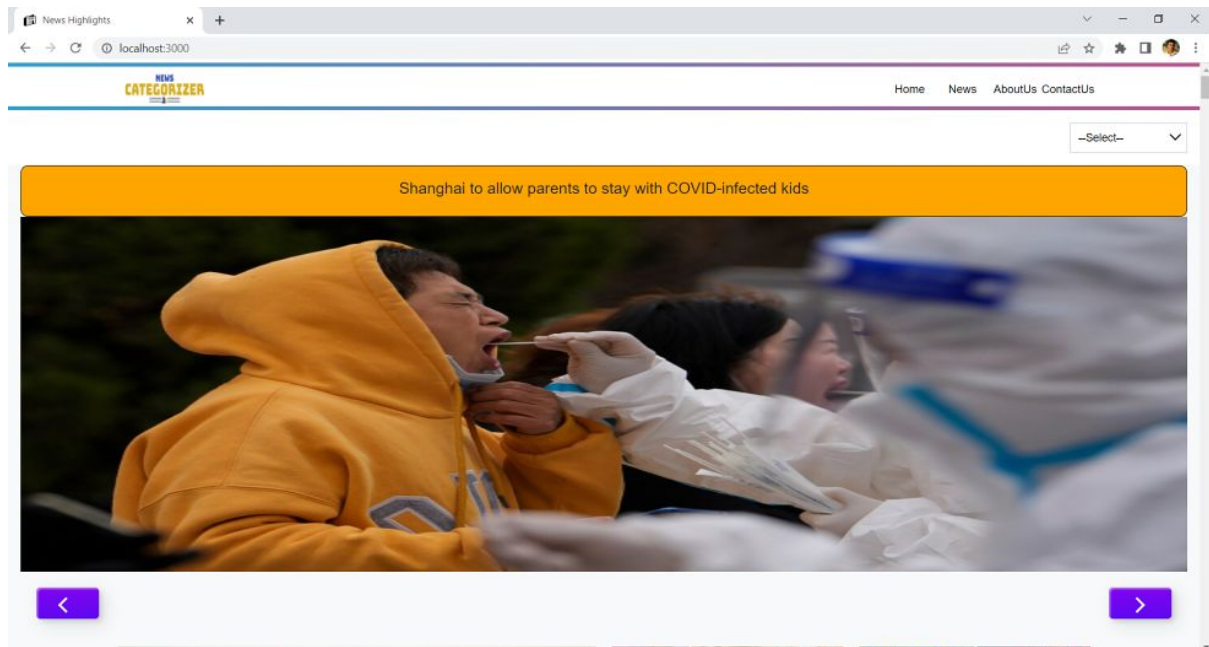


Figure 7.2 Home Page Navbar and Carousel

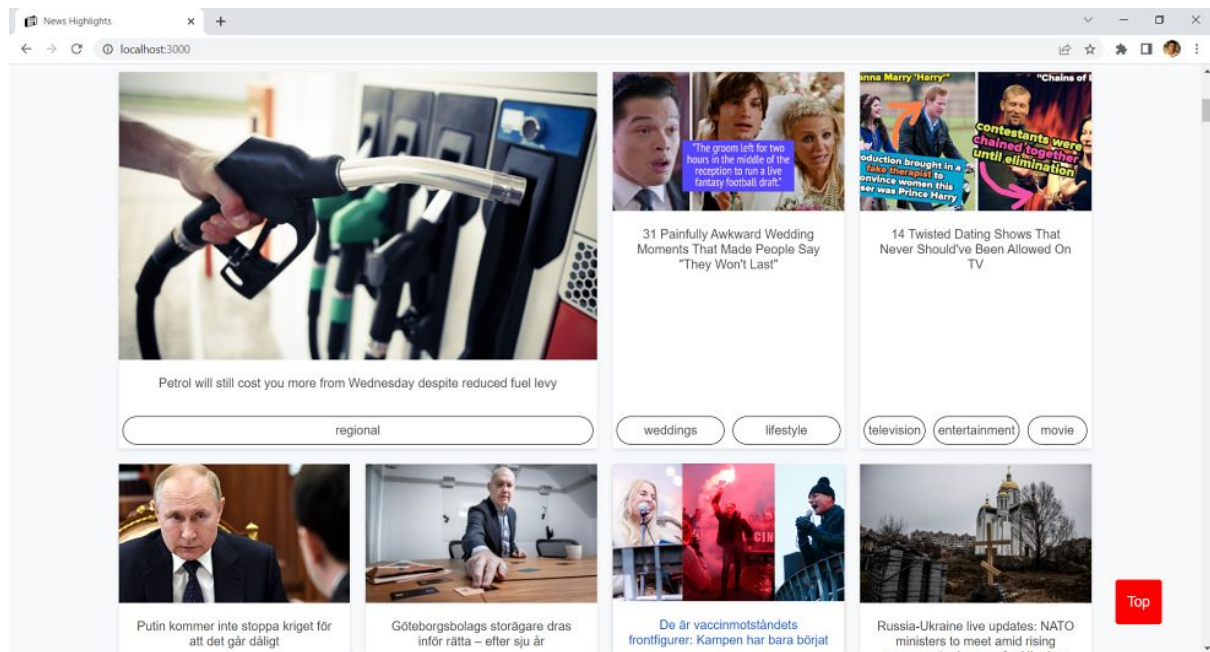


Figure 7.3 Home Page with latest news

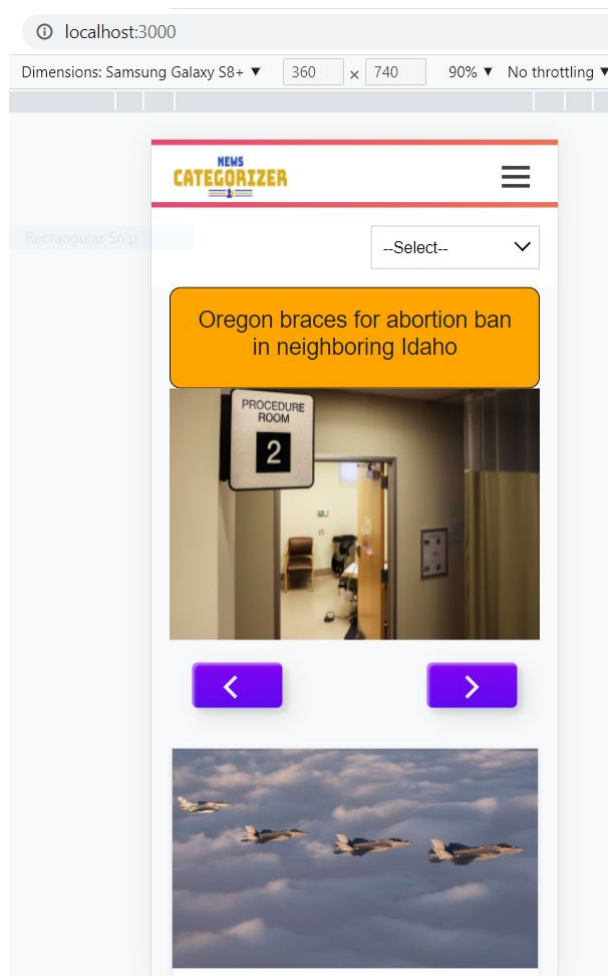


Figure 7.4 Mobile View of home page

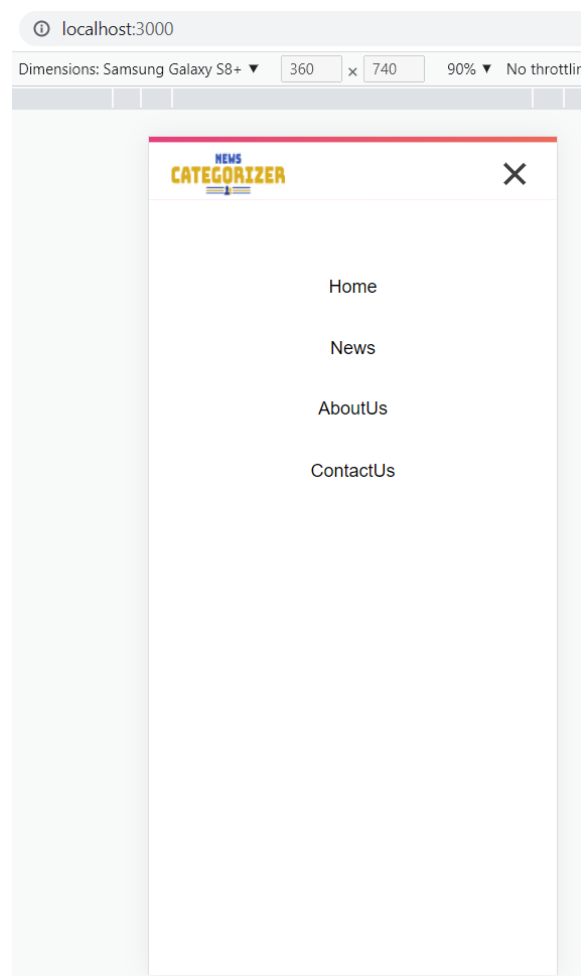


Figure 7.5 Mobile view responsive navbar

7.2 Search news based on category:-

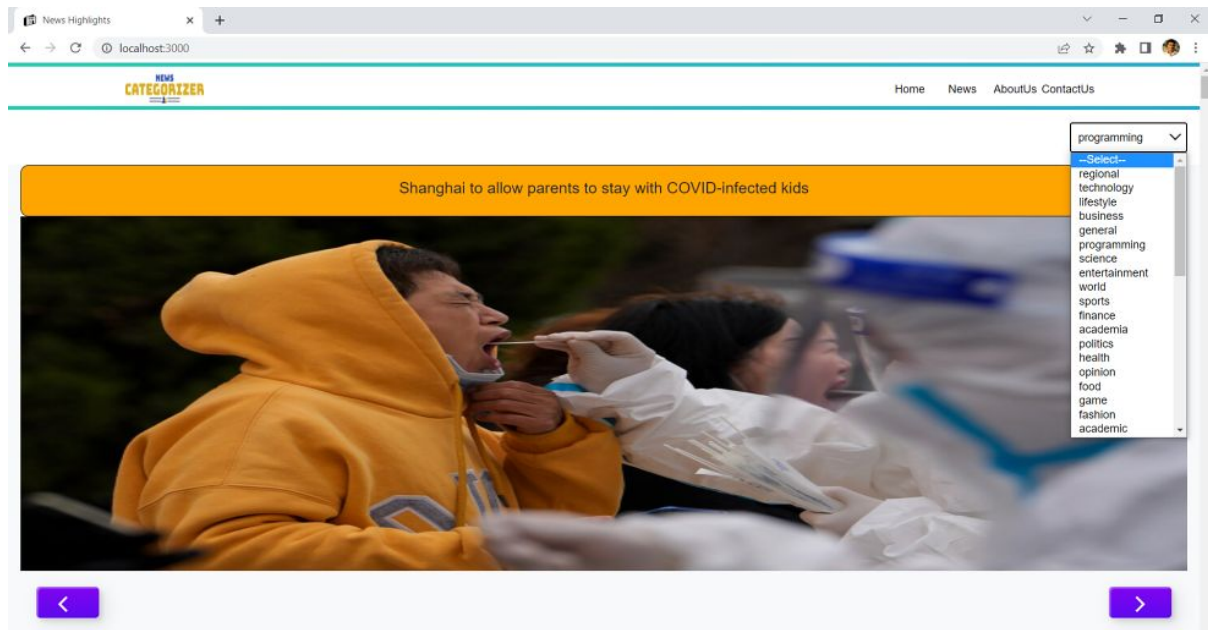


Figure 7.6 Dropdown for search news by category

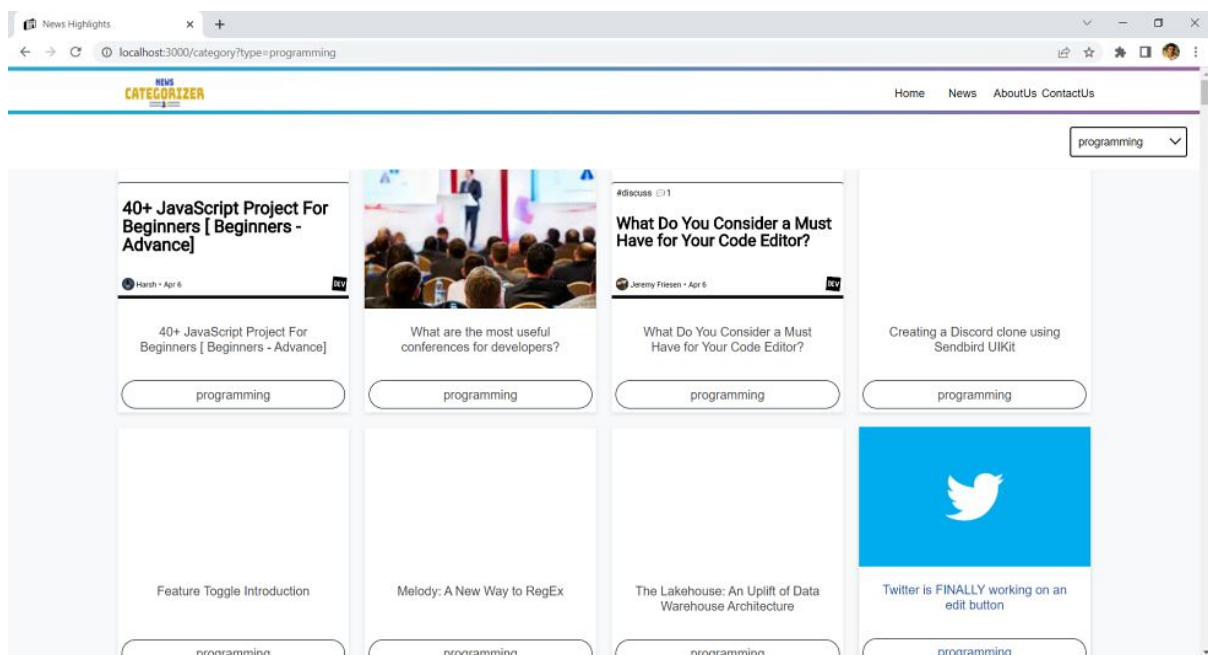


Figure 7.7 Page view related to search category

7.3 Explore News

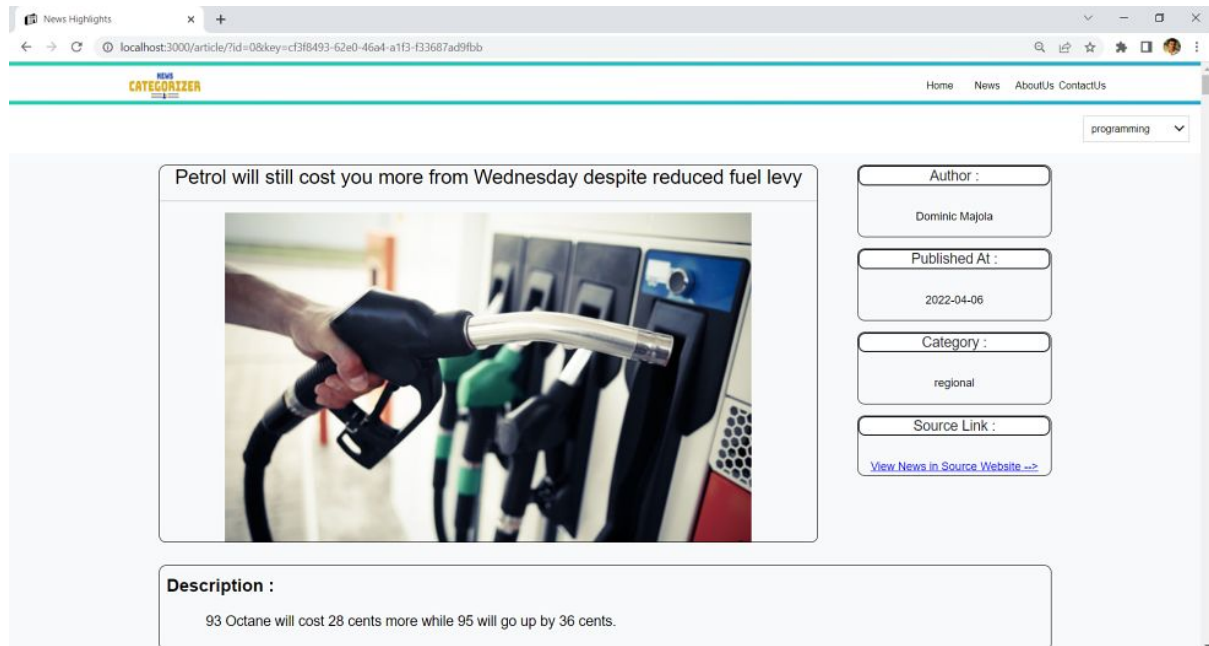


Figure 7.8 Page view of news in more details

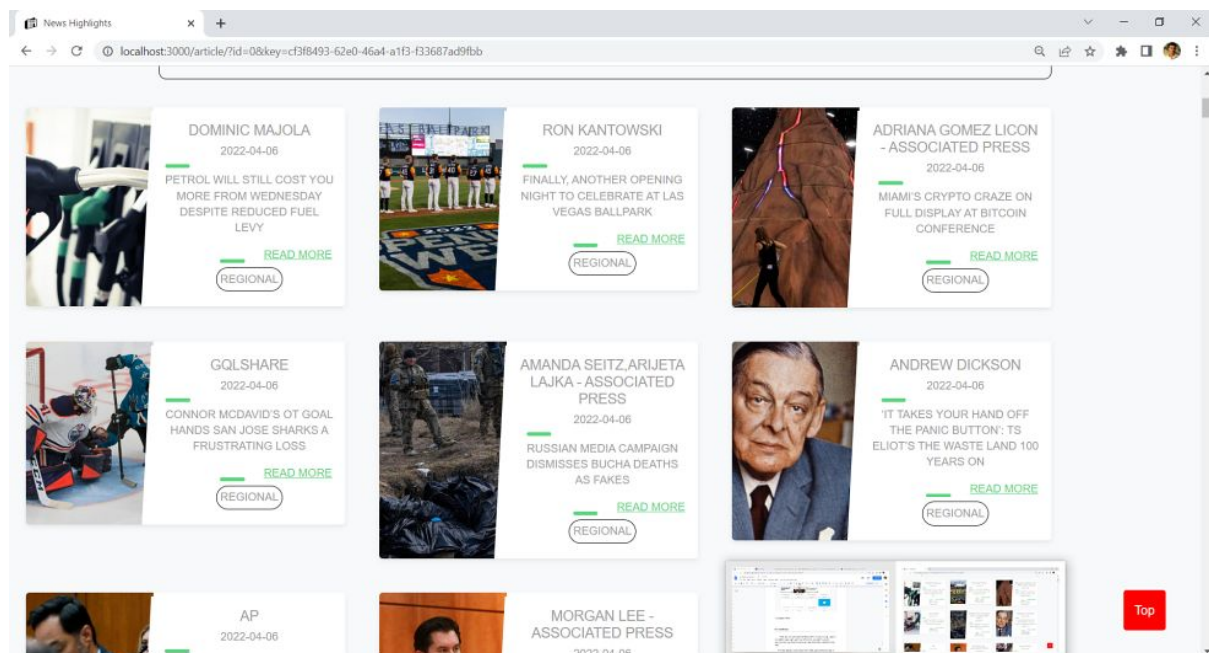


Figure 7.9 News related to news that open for more details

7.4 News Prediction

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/news-categorizer'. The page has a header with a logo 'NEWS CATEGORIZER' and navigation links: Home, News, AboutUs, ContactUs. Below the header is a dropdown menu with '--Select--'. The main content area is titled 'Enter News :-' and contains a text input field with the placeholder text: 'Twitter is finally working on an edit button, after years of calls asking for one. The company has only shared a few details of how it works.' Below the input field is a yellow 'Predict' button. Underneath the button is a section titled 'Available News Categories for Prediction' with a row of buttons: WELLNESS, POLITICS, ENTERTAINMENT, TRAVEL, STYLE & BEAUTY, PARENTING, FOOD & DRINK, WORLD, BUSINESS, and SPORTS. At the bottom of the page is a black footer with the text '© 2022 Copyright: News-Prediction'.

Figure 7.10 News prediction page

This screenshot shows the same web application as Figure 7.10, but with a modal pop-up window open. The pop-up is titled 'Prediction' and contains the text 'BUSINESS News'. It has a 'Close' button in the bottom right corner. The background of the page is dimmed. The rest of the interface, including the header, navigation links, dropdown menu, text input field, 'Predict' button, category buttons, and footer, remains visible and unchanged from the previous figure.

Figure 7.11 Pop-up for showing news category

7.5 About Page

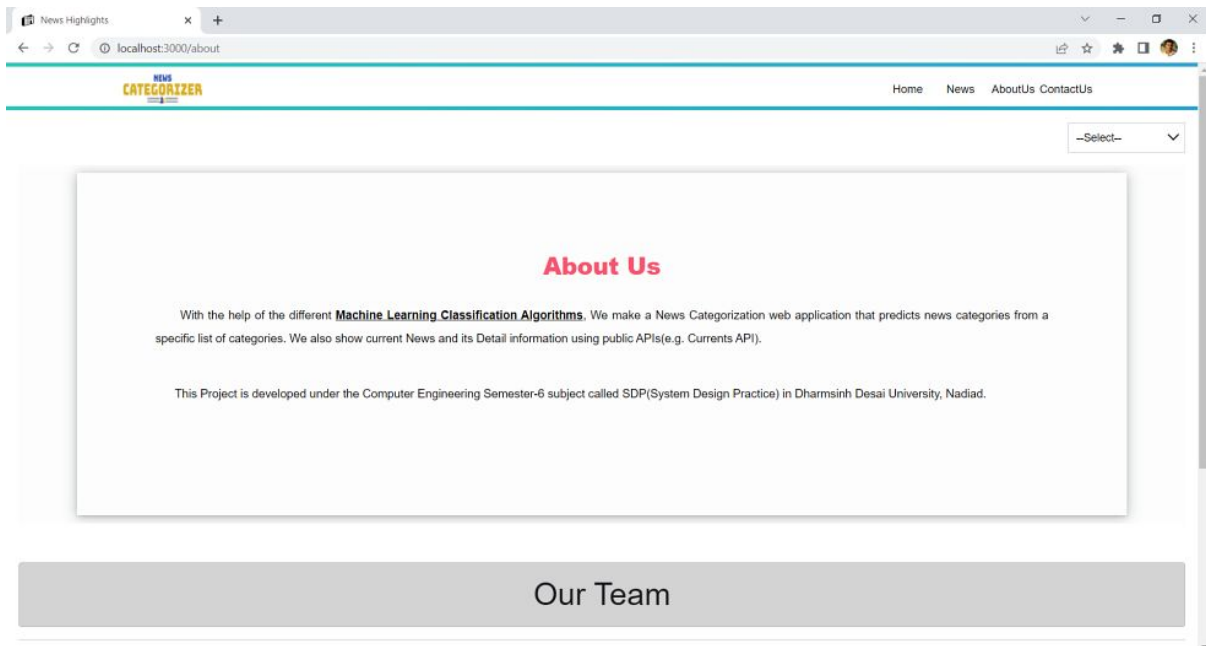


Figure 7.12 About page - top

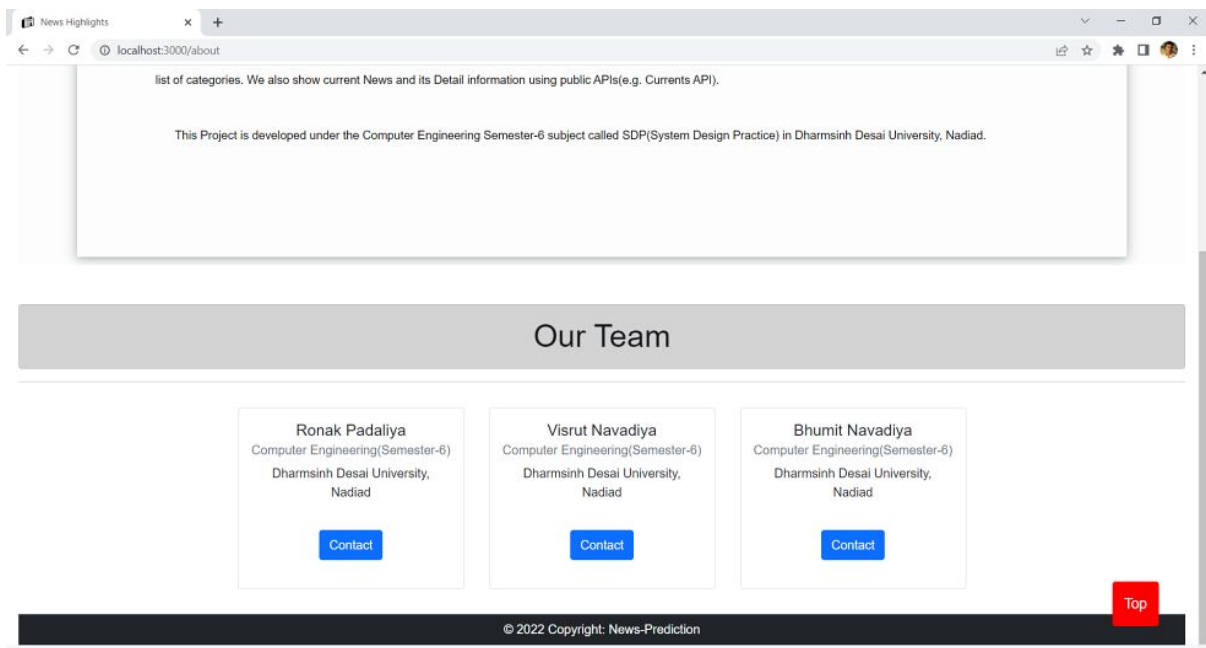


Figure 7.13 About page - bottom

7.6 Contact Page

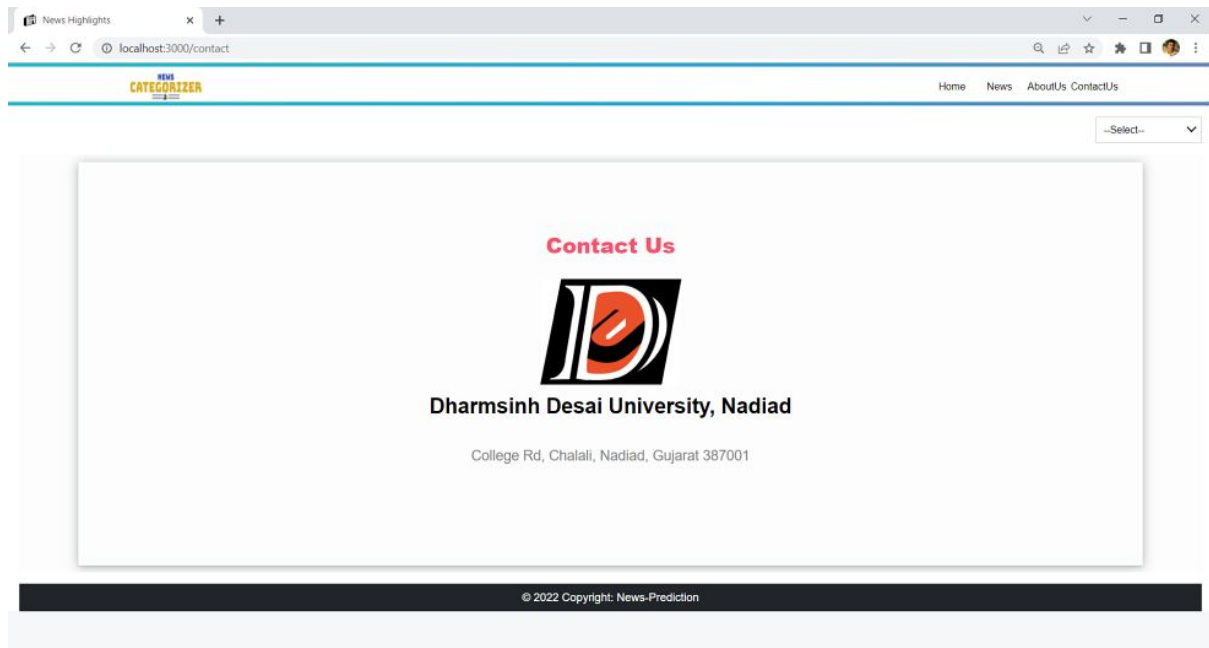


Figure 7.14 Contact page

8. Conclusion

This project is aimed at developing an application which is predict News category using Machine Learning and Multinomial Naive Bayes Algorithm.

For this project, we selected News Categorization functionality. To achieve this first we train the Multinomial Naive Bayes model using news dataset which is present in **Kaggle**. This dataset have near about 2 lacks rows. After that we make a MNB.joblib file which contain trained model.

This file is used to predict the news category in Django application. First, we tack input from the user after that we pass this input to Backend API call. In response it gives predicted category.

The accuracy of our Multinomial Naive Bayes Machine Learning model is **78.12 %**.

So, after performing various comprehensive tests, we conclude that our project is working successfully. But of course, there is always a scope for improvement and learning. This was our first ML project but yes in the end, we did learn something new from it. We are now looking forward to overcome the existing limitations and to add new possible extensions which are discussed in the next section.

9. Limitations and Future Extensions

9.1 Limitations:

- For some news description application not give correct Prediction.
- Accuracy of ML model is **78.12%**.
- Application not predict more then 10 category.
- All types of category is not cover.
- In news exploration full description is not provided.

9.2 Future Extensions:

- Increase accuracy of ML model.
- When time goes need to regular update dataset to current latest news and again create ML model
- We can also do news recommendation based on user interested news category.
- We can use different ML algorithms to train and predict news.
- Increase News Categories in our trained model.

10. Bibliography

10.1 Reference use for develop web application:

- <https://stackoverflow.com/>
- <https://www.djangoproject.com/>
- <https://reactjs.org/>
- <https://codepen.io/>
- Fetch News Data from public API:-
<https://currentsapi.services/en>
- Connect React with Django:-
<https://www.digitalocean.com/community/tutorials/build-a-to-do-application-using-django-and-react>
- Used News dataset link:-
<https://www.kaggle.com/datasets/rmisra/news-category-dataset>

10.2 Tools user for develop web application:

- Google Colab
- Visual Studio Code
- GitHub