

Suraj Verma

Wireframe Document

News Article Sorting



Document Version Control

Date Issued	Version	Description	Author
31/03/23	1	Initial WD – V1	Suraj Verma

Contents

Document Version Control	2
1. Introduction	4
1.1 Why this Wireframe Document?	4
1.2 What is the scope?	4
2. Prediction Webpage.....	5
2.1 Homepage.....	5
2.2 Giving form inputs.....	6
2.3 Prediction... ..	6
2.4 Testing with incorrect ranged input	7
3. API Response.....	8
3.1 Homepage after deploying the app	8
3.2 Testing with Postman	9
3.3 Providing input in json format.....	9
3.4 Prediction using API	11
3.4 Testing API Response with intended exception	12

1 Introduction

What is Wireframe Document?

A wireframe is a document that outlines the structure of a website or app. It's a tool that clearly indicates what needs to be on every page. More importantly it establishes the relationship between everything on each page.

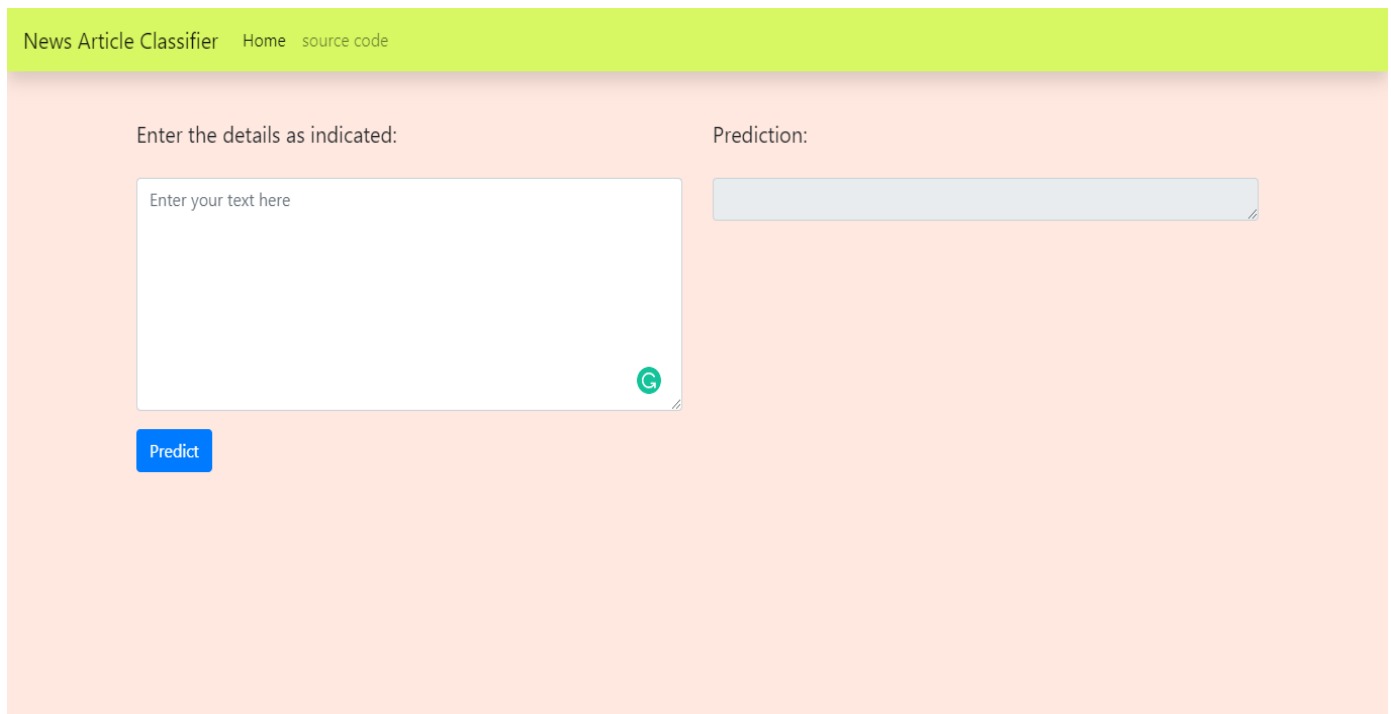
What is Scope?

Wireframing is a way to design a website service at the structural level. A wireframe is commonly used to layout content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content is added.

2 Prediction Webpage

2.1 Homepage

Homepage of the deployed app.



The screenshot shows the homepage of the 'News Article Classifier' application. At the top, there is a green navigation bar with the text 'News Article Classifier' and two links: 'Home' and 'source code'. The main content area has a light orange background. On the left, under the heading 'Enter the details as indicated:', there is a large white text input field with the placeholder text 'Enter your text here'. Below this field is a blue button labeled 'Predict'. To the right of the input field, under the heading 'Prediction:', there is a long, empty light blue rectangular box intended for the prediction result. A small green circular icon with a white 'G' is located at the bottom right corner of the text input field.

- **Home** – Homepage for Metro Interstate Traffic Prediction app.
- **Source Code** – Redirects to the GitHub source code page.
- **Input fields** – Input with respect to model trained features .
- **Input ranges** – Placeholder guiding the valid ranges for input. Any values outside these ranges will lead to the error page.
- **Prediction** – Prediction will be displayed here.
- **Predict** – Button to display result of prediction.

2.2 Giving form inputs

News Article Classifier Home source code

Enter the details as indicated:

I own a company that values at nearly 2 billion dollars. But due to some financial breakdowns, I want to sell my company

Predict

Prediction:

The values entered in each field should lie between the valid input range, otherwise upon prediction the application will be redirecting the error page with the respective error.

2.3 Prediction

The Predicted result are too ranged between appropriate analyzed values, but as far as the input value is within the domain range, the prediction out of range exception is less likely to be triggered.

News Article Classifier [Home](#) [source code](#)

Enter the details as indicated:

Prediction:

Enter your text here

Buisness

Predict

2.4 Testing with incorrect ranged input

The application is tested with intended exceptions such as invalid columns, invalid datatypes and invalid ranges.

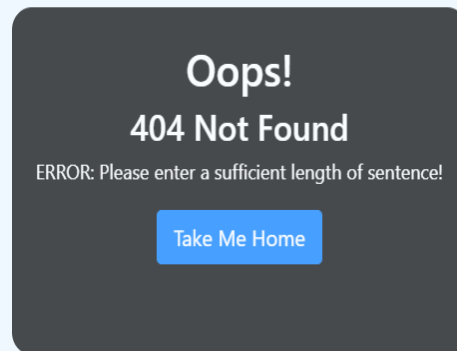
News Article Classifier [Home](#) [source code](#)

Enter the details as indicated:

Prediction:

hello

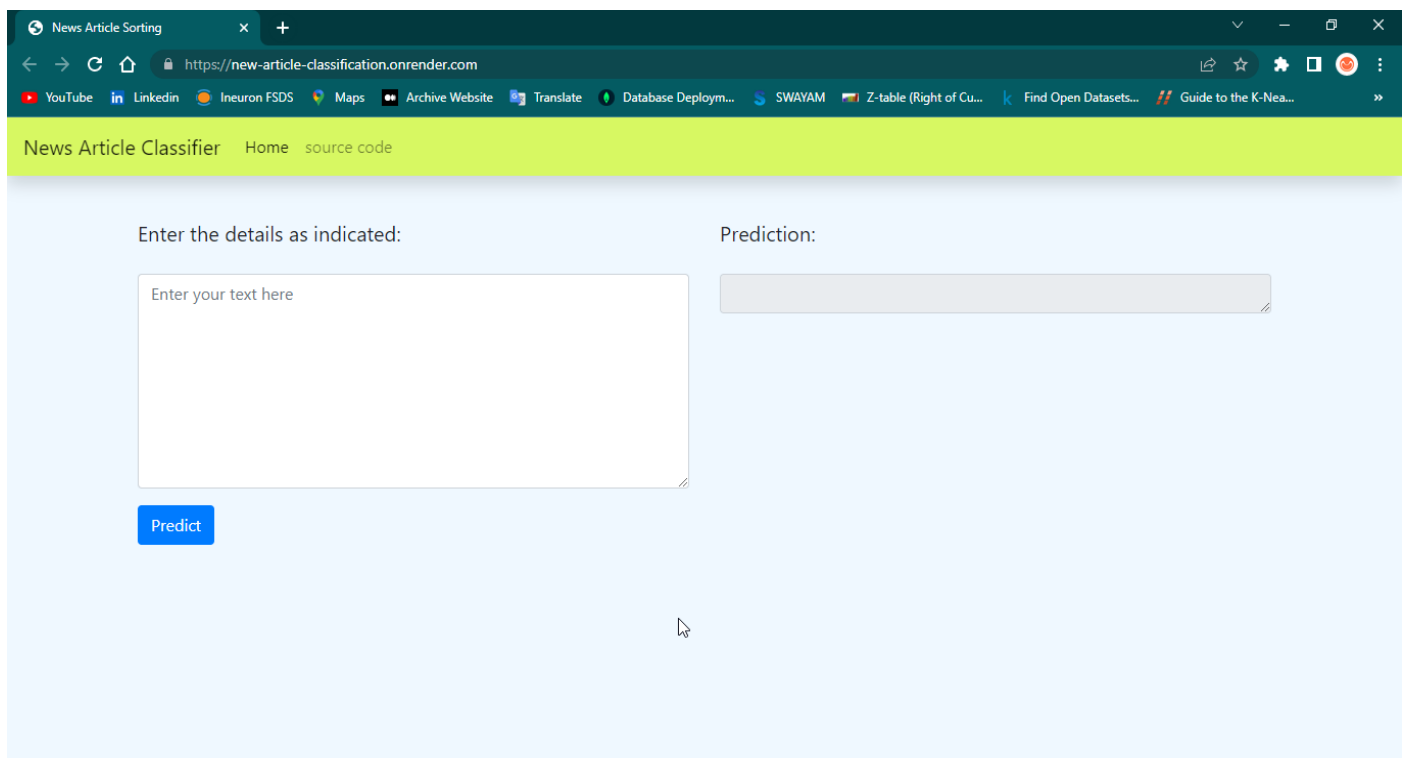
Predict



3 API Response

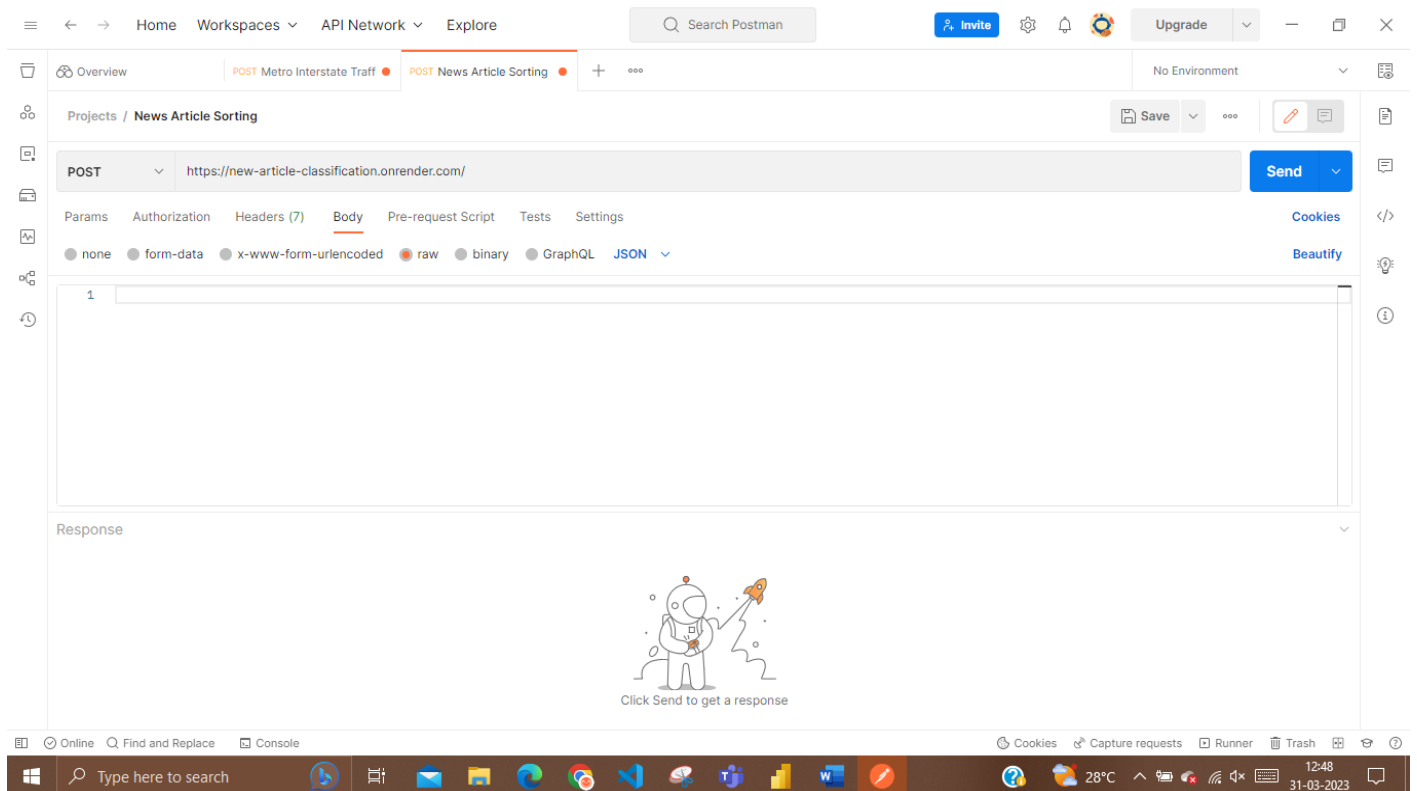
3.1 Homepage after deploying app

Predicting traffic volume through API response is the second feature of our prediction app. The deployed app link can be used for API response.



3.2 Testing with Postman

For testing Prediction using API request we will use Postman. Postman is an API platform for developers to design, build, test and iterate their APIs.



3.3 Providing input in json format

As per our model, for API response prediction we accept only json formatted response. So go to Body>raw and write json formatted inputs.

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is present. On the right, there are buttons for 'Invite', 'Upgrade', and a dropdown menu. Below the navigation bar, the 'Overview' tab is active, showing a list of projects. The 'News Article Sorting' project is selected. The main workspace shows a POST request to the URL 'https://new-article-classification.onrender.com/'. The request body is in JSON format, with a single key 'txt' containing a long paragraph of text. The response section is empty, showing a cartoon character and the text 'Click Send to get a response'. The bottom status bar shows the system clock as 12:48 on 31-03-2023, along with various system icons.

Home Workspaces API Network Explore Search Postman Invite Upgrade

Overview POST Metro Interstate Traff POST News Article Sorting No Environment

Projects / News Article Sorting Save

POST https://new-article-classification.onrender.com/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
 "txt": "Hour ago, I contemplated retirement for a lot of reasons. I felt like people were not sensitive enough to my injuries. I felt like a lot of people were backed, why not me? I have done no less. I have won a lot of games for the team, and I am not feeling backed, said Ashwin"
}

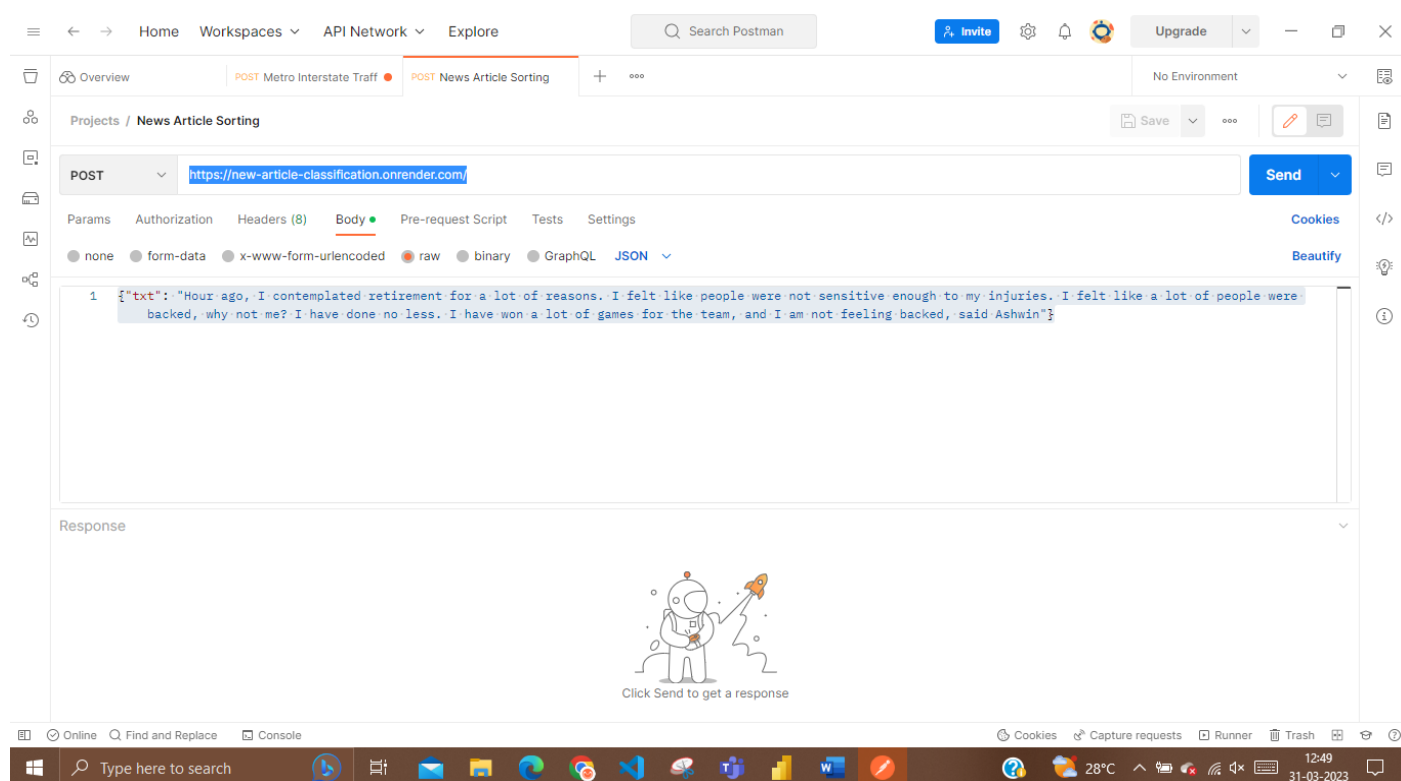
Response

Click Send to get a response

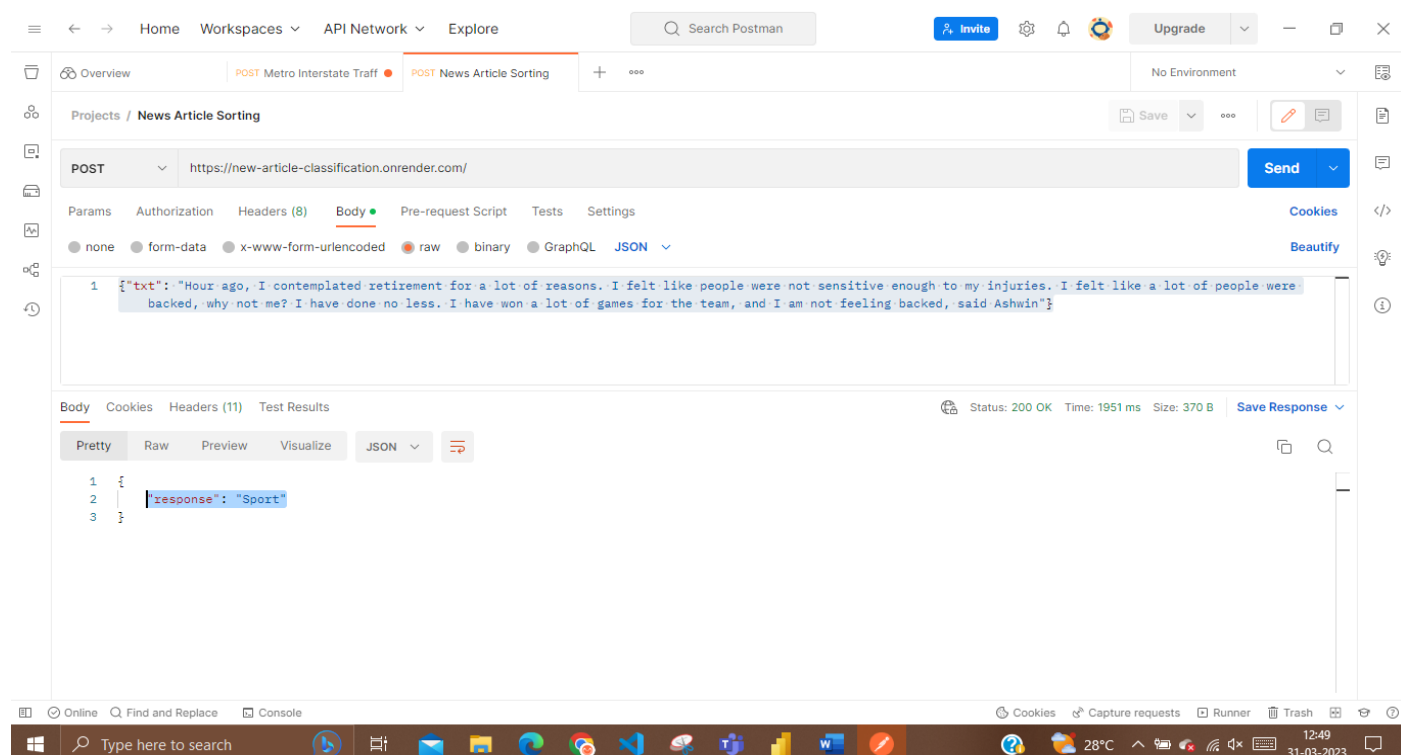
Online Find and Replace Console Cookies Capture requests Runner Trash 12:48 31-03-2023

3.4 Prediction using API

After providing the correctly ranged inputs, we will use the deployed app link as an API for returning prediction response from the Postman.



Click on send to get the prediction result.



3.5 Testing API response with an intended exception

Here we will test the prediction response with an intended irrelative column in our json input.

The image displays two screenshots of the Postman API client interface, showing a POST request to the URL `https://new-article-classification.onrender.com/`.

Top Screenshot: The request body is set to JSON. The input is a long, irrelevant string: `"txt": "He362427384723811o"`. The response status is 200 OK, with a time of 1457 ms and a size of 422 B. The response body is shown in the 'Pretty' view:

```

1 {
2   "response": "Text contains many numbers! Please enter a suitable text."
3 }

```

Bottom Screenshot: The request body is set to JSON. The input is a short string: `"txt": "Hello"`. The response status is 200 OK, with a time of 911 ms and a size of 410 B. The response body is shown in the 'Pretty' view:

```

1 {
2   "response": "Please enter a sufficient length of sentence!"
3 }

```