**NEWS ARTICLES CLASSIFIER**

Problem Statement: Classify News Articles into categories - With information overload today users are inundated with news articles of all topics, even the ones which may not be relevant to users. Design a system which can classify incoming news articles and appropriately tag the corresponding category.

1.Data Ingestion

2.Data Preparation

3.Data segregation & Model Training

4.Model Deployment

5.Model Prediction

Languages used: Python, javascript, html

1. DATA INGESTION

* We have used the mediastack news API to get the news from the web.
* The news that we get from the mediastack API has in it the category of the news that is received(general, sports, business, entertainment, health, science, technology)
* In addition to getting the data from the API we made sure that data we get is balanced through the get\_balanced\_mediastack function in the sendNews.py file of the bin folder.
* Kafka is used to send the data from the producer to the consumer.
* The receiveNews.py contains the code for the producer and the sendNews.py has the code for the consumer
* The data is then sent to the DB, where it is stored for later use
* We are using MongoDB to store the data

1. DATA PREPARATION (PRE-PROCESSING), SEGREGATION AND MODEL TRAINING

* For training the model firstly the preprocessing is done
* While getting the data from the mediastack API itself we have made sure that the data has equal number of data in each category so that the data is distributed equally so that the model is not biased.
* We have trained the data using SGDClassifier, as this classifier has the partial\_fit function which helps us retrain the model.
* We first normalize the text and then process the data.
* Then we have trained the data . This trained data has been stored in a path similar to bin/fastapi/saved\_data/
* We are saving the model instead of training it every time we run the application. We retrain the model instead .
* Retraining the model by saving the model helps us improve the accuracy of the model as the time passes.

1. MODEL PREDICTION

* We have used FastApi to build the APIs
* In addition to the fastApi framework to incorporate Apis we have incorporated some UI as well



* The UI allows the user to enter the title of the news and the description. Then the model in the backed would predict the category when the “Get Prediction” button is clicked
* The user is also given an option to retrain the model with the latest data

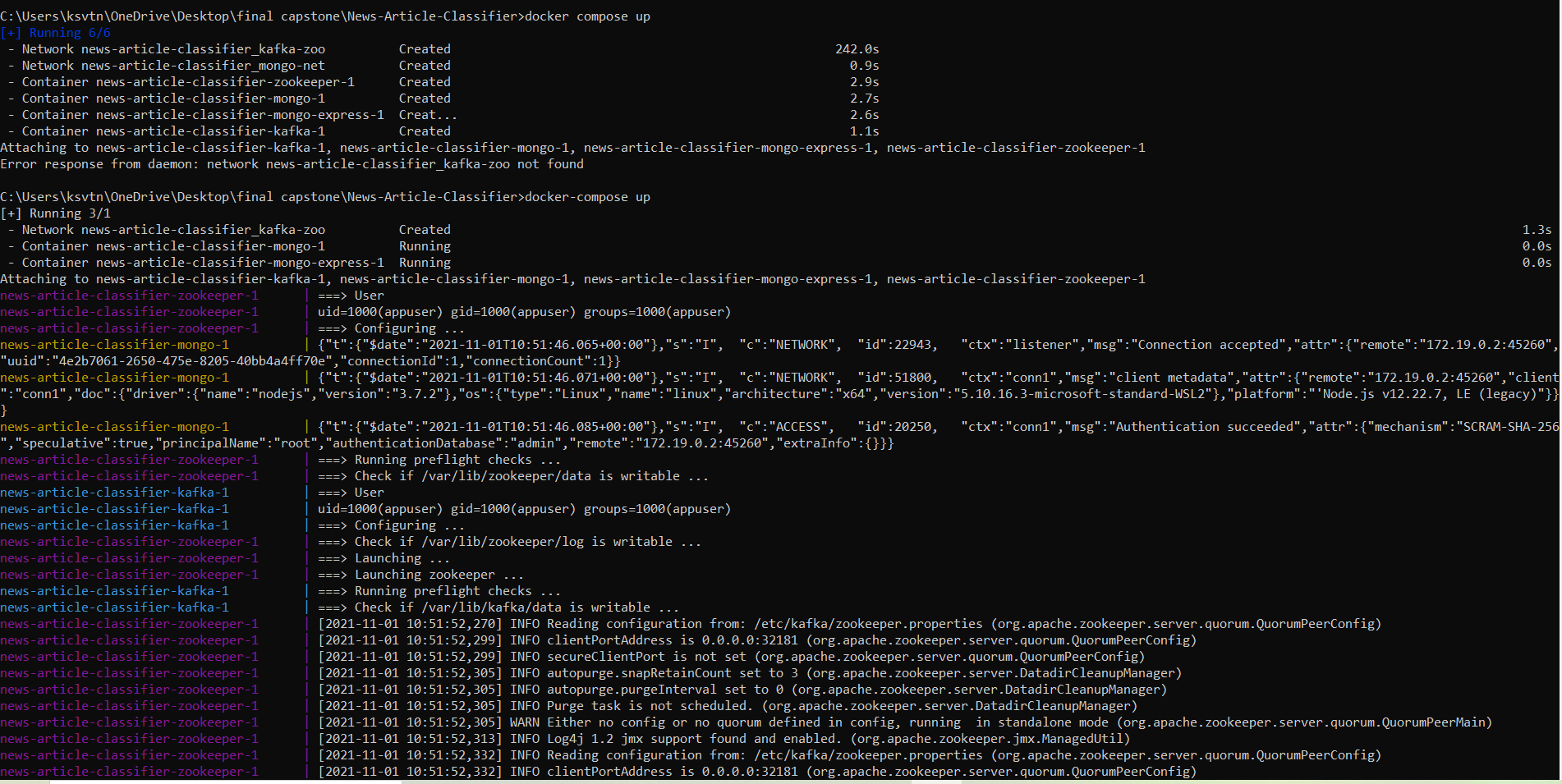
4 MODEL INFRASTRUCTURE AND DEPLOYMENT

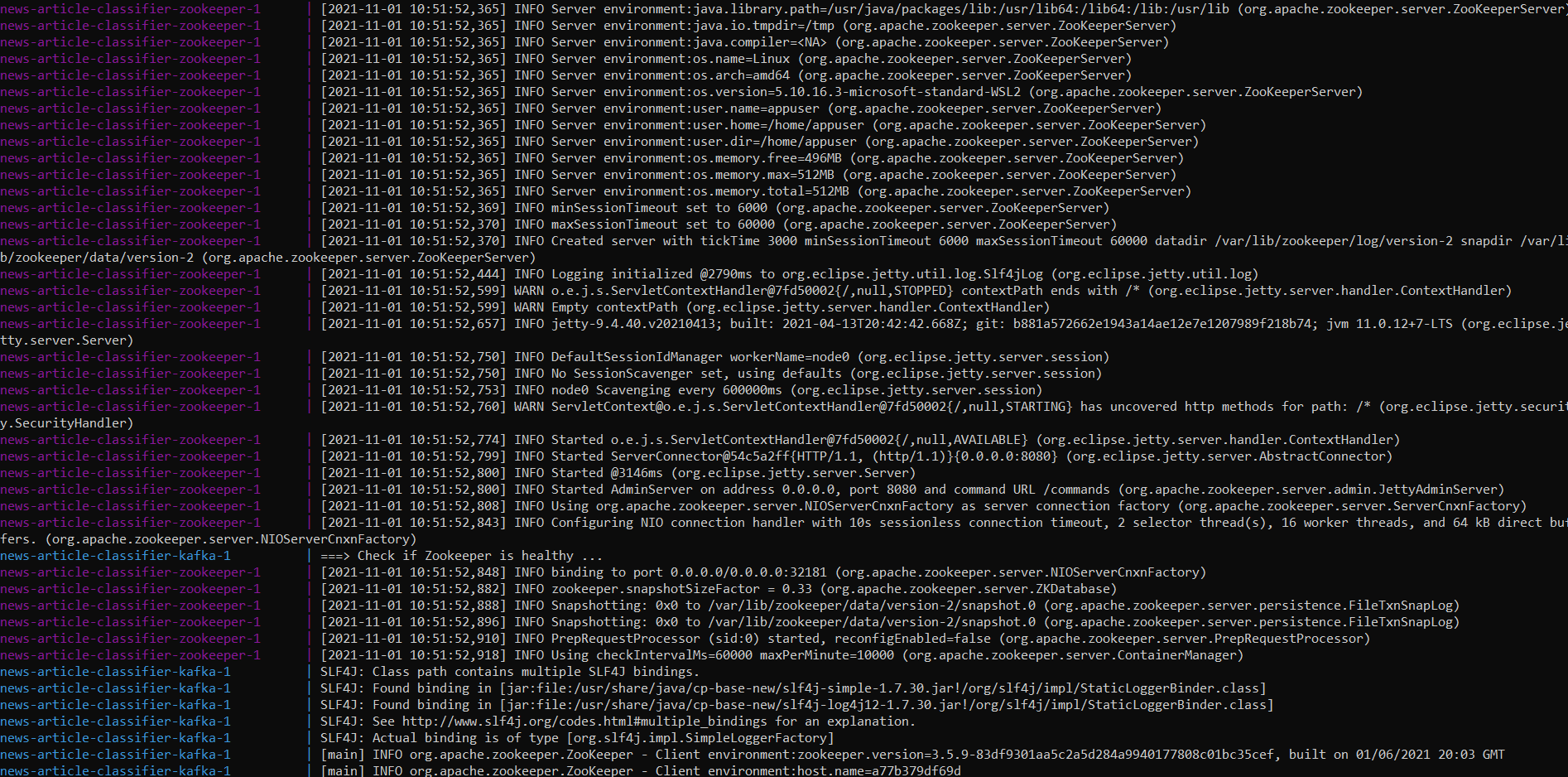
* We have used the FastAPI and docker to deploy the model

**The setup**

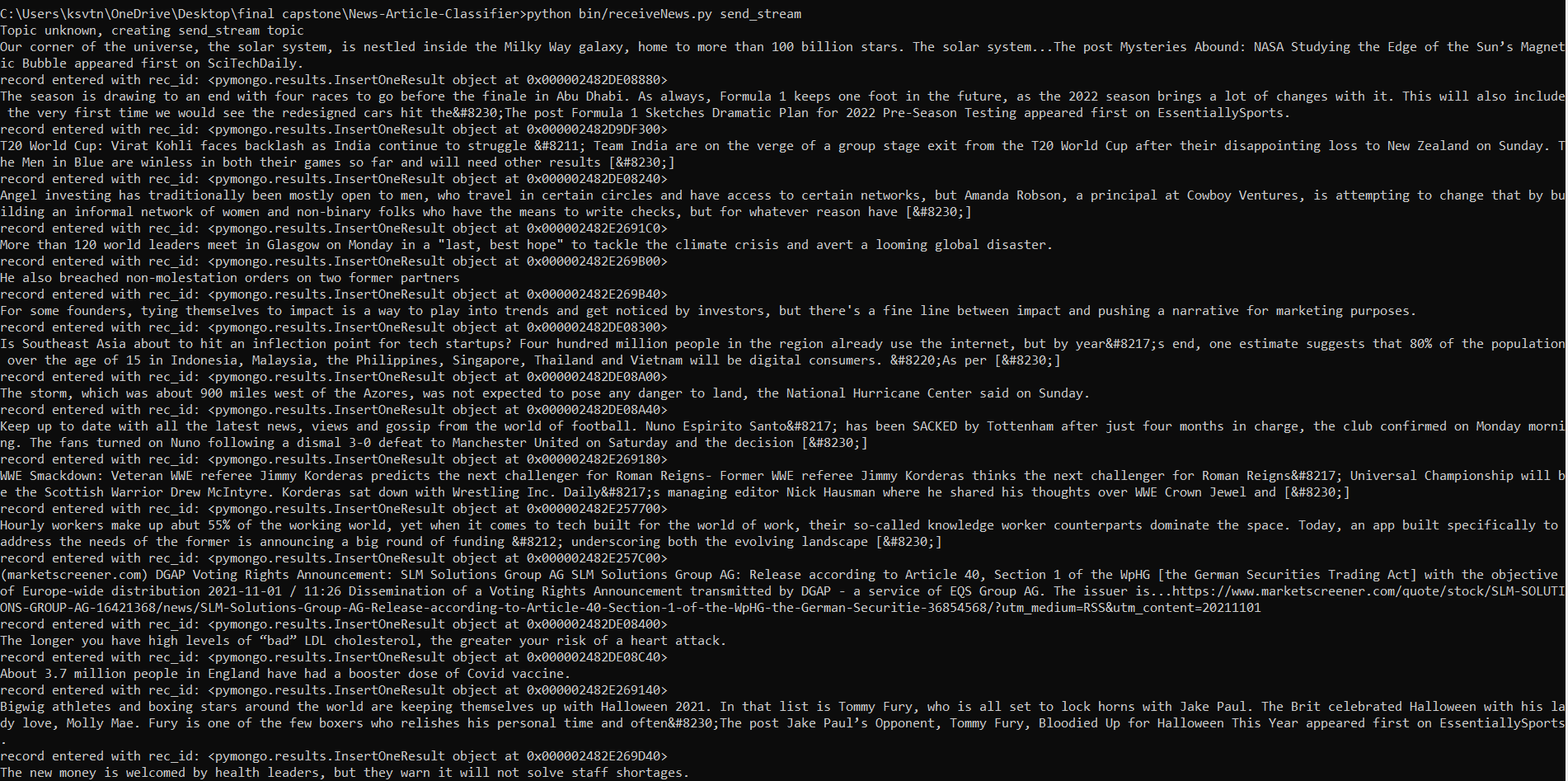
* Clone the repo
* Run docker-compose up (Sets up kafka and Zookeeper for data streaming + MongoDB and Mongo express)
* In a new terminal run python bin/receiveNews.py send\_stream (Kafka consumer)
* In a new terminal run python bin/sendNews.py send\_stream (Kafka Producer)
* In a new terminal navigate to bin/fastapi directory. Run python fastapi\_endpoints.py (FastAPI endpoints)
* Navigate to the UI folder, and run npm start to start the react application.
* Use the application for all your News Article Classification needs.

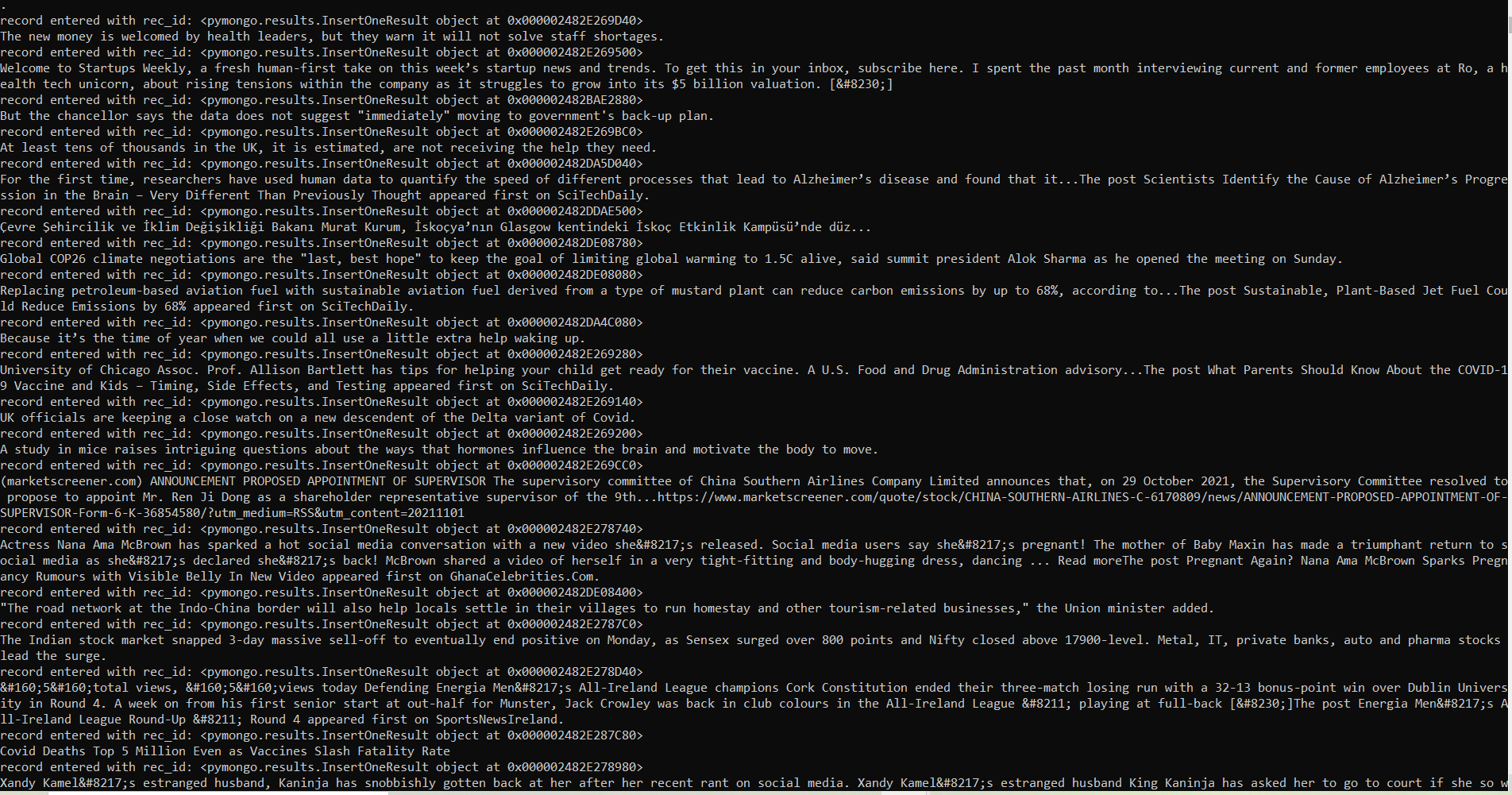
\*The following is the snip of the screen when “docker-compose up” Is run.





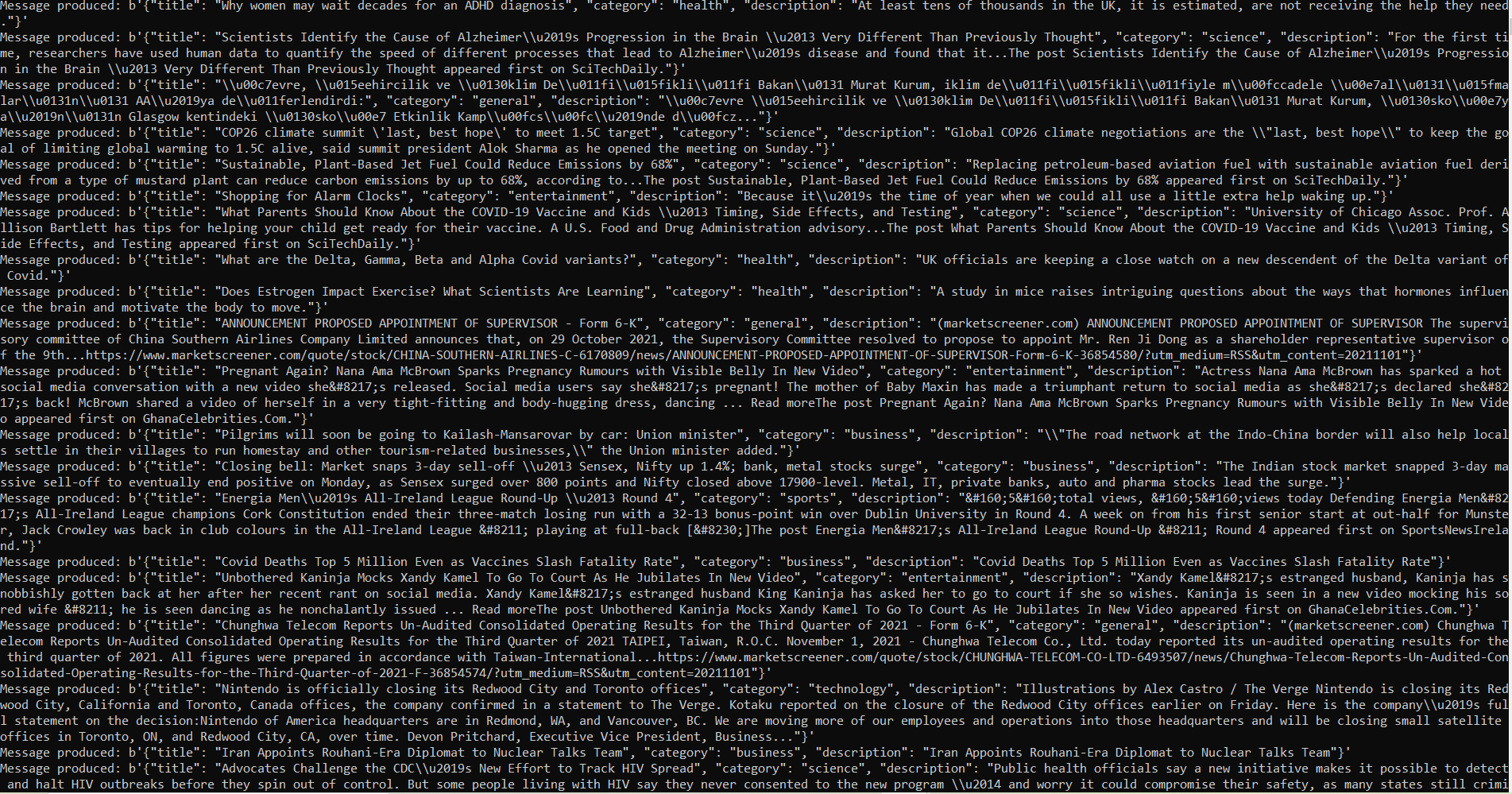
\*The screen when python bin/receiveNews.py send\_stream is run:



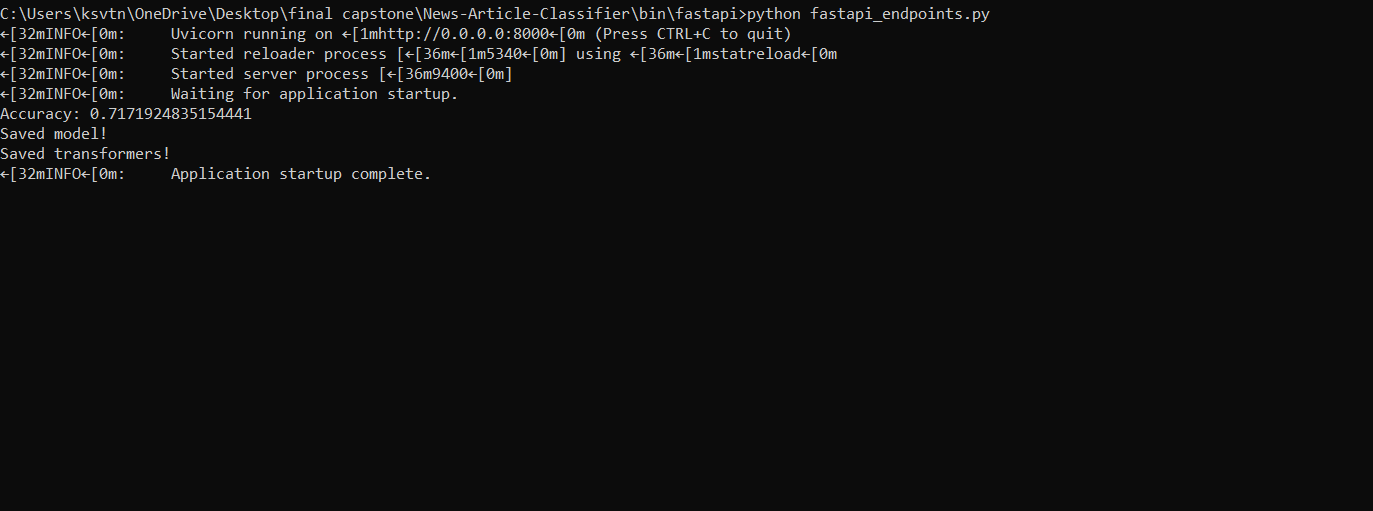


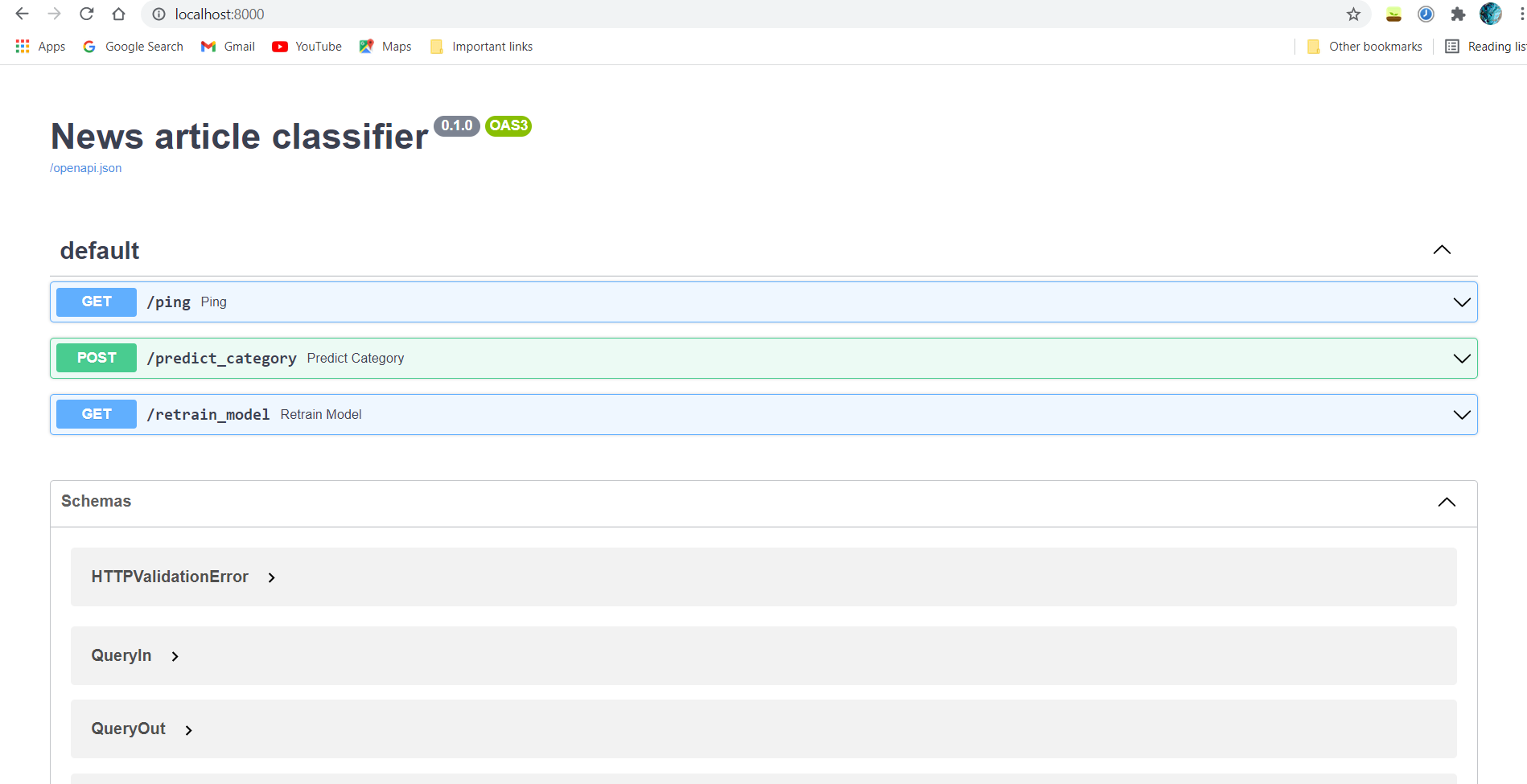
\*The screen when python bin/sendNews.py send\_stream is run:

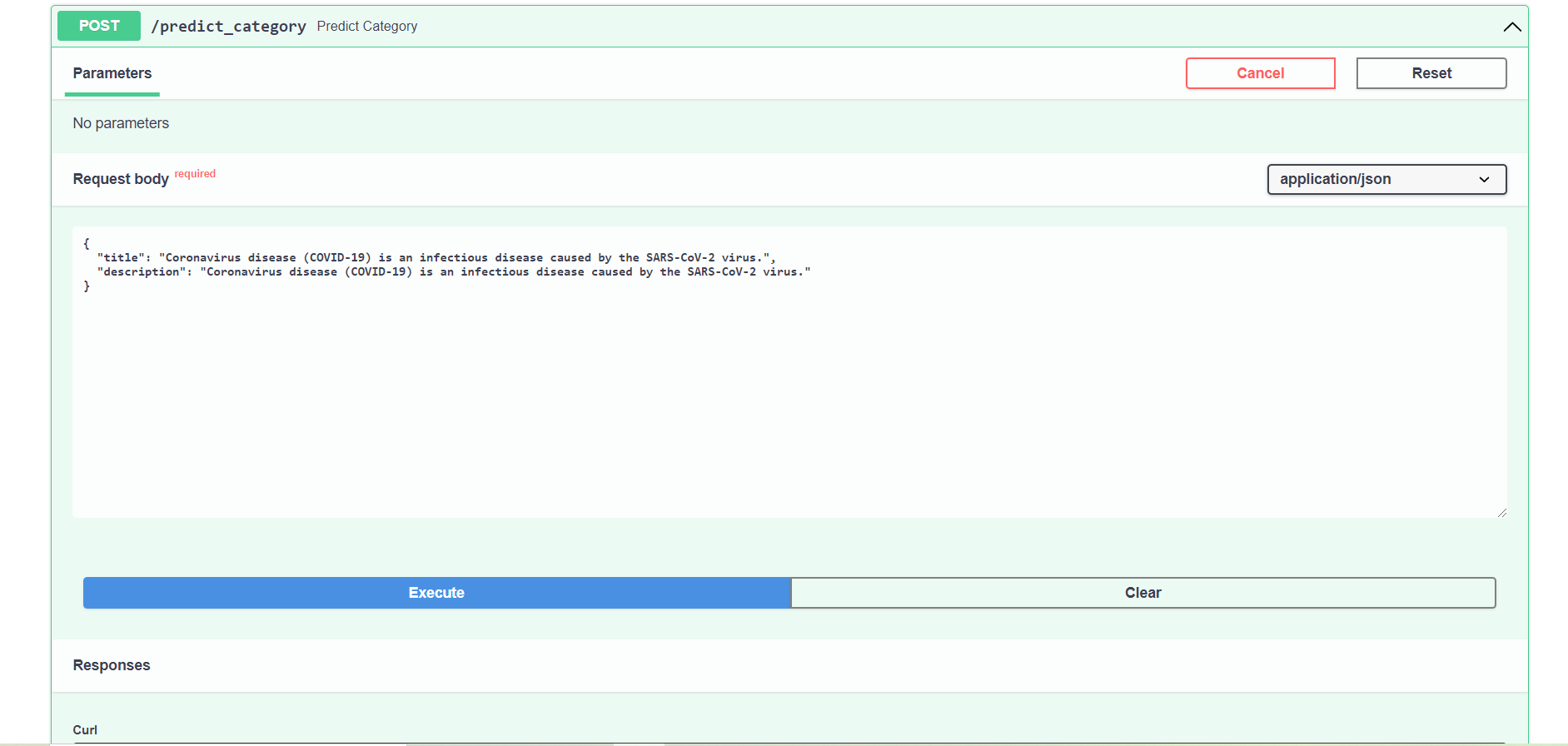


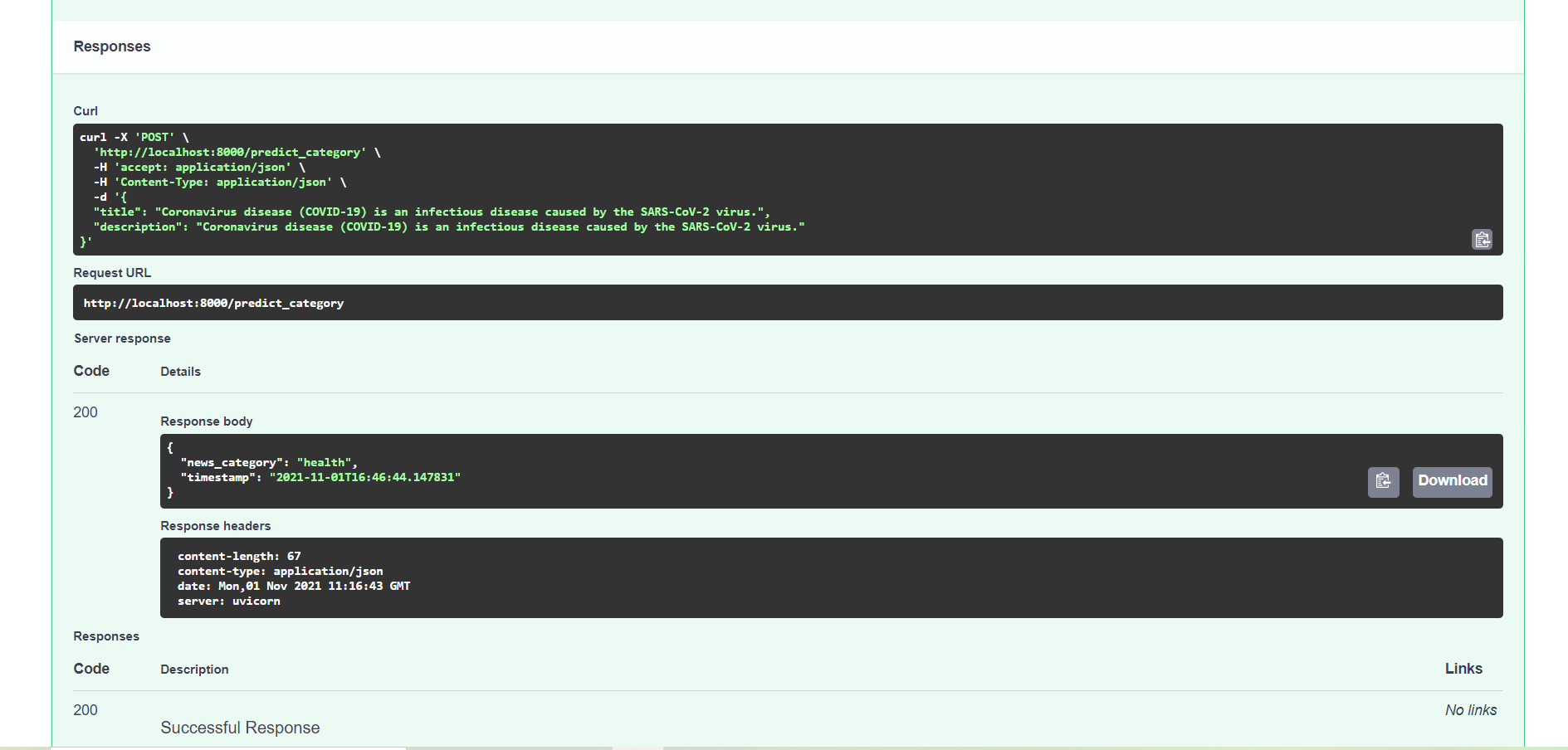


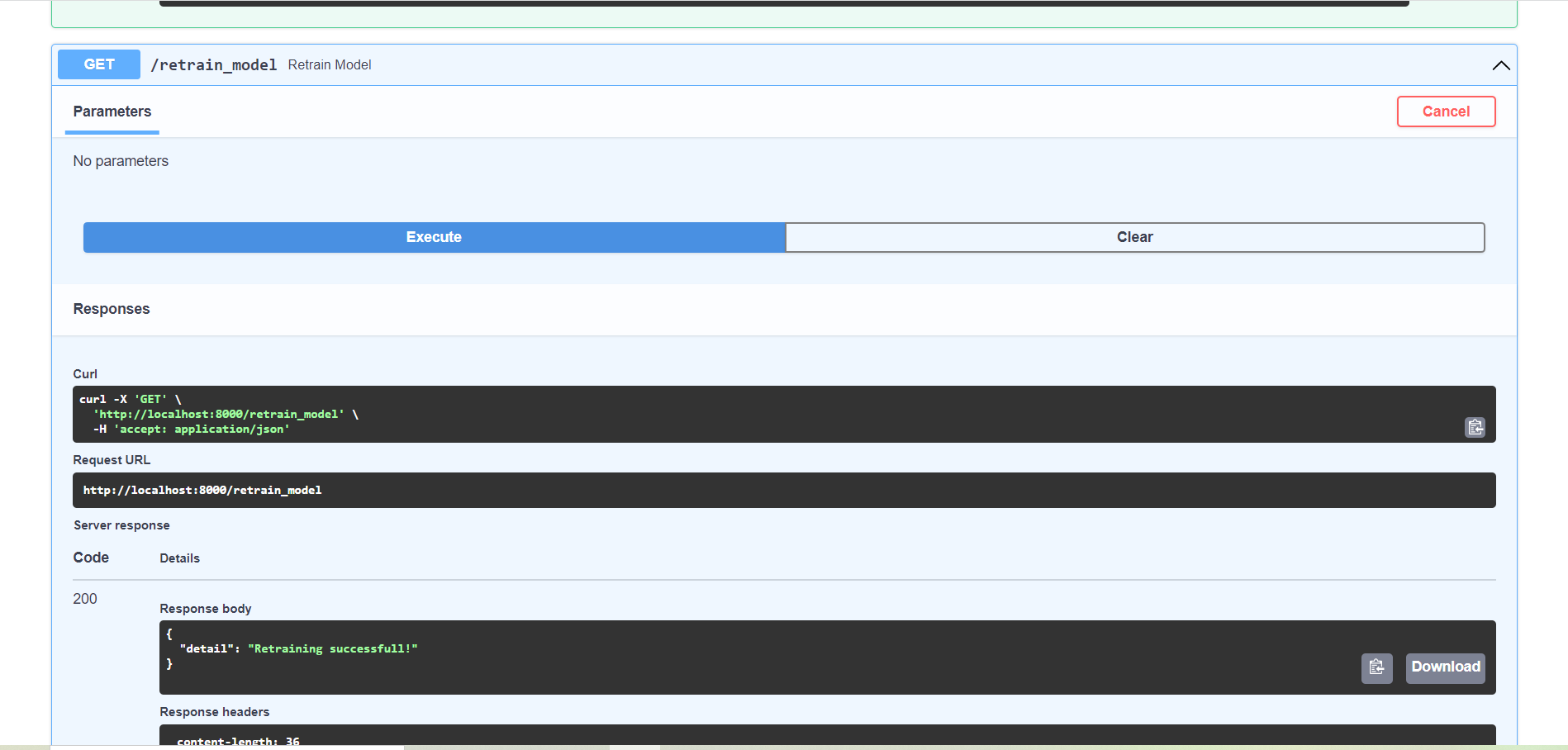
\*When a new terminal is opened in the bin/fastapi directoryand  “python fastapi\_endpoints.py” is run.











\*When we navigate to the bin/ui folder and enter “npm start” in a new terminal.

