Implementing Nowcasting Intelligent System for Media

[Team Members](#_2et92p0)

[Part 1: Designing the REST API with Logic Enhancement](#_tyjcwt)

[Operations supported by API](#_1t3h5sf)

[GeoPy distance logic and Nearest Location](#_dv8czbhmjxs)

[Positive Test Cases](#_4d34og8)

[Negative Test Cases](#_3rdcrjn)

[Part 2: New Client (Streamlit)](#_lnxbz9)

[Conclusion](#_1ksv4uv)

[References](#_44sinio)

# 

Implementing Nowcasting Intelligent System for Media

# 

# Team Members

| NAME | NUID |
| --- | --- |
| PRIYANKA DILIP SHINDE | 001524484 |
| KSHITIJ ZUTSHI | 001021288 |
| DWITHIKA SHETTY | 002114630 |

# Part 1: Designing the REST API with Logic Enhancement

# 

## Operations supported by API

1. Endpoint 1: **/nowcast/{img\_Id}**

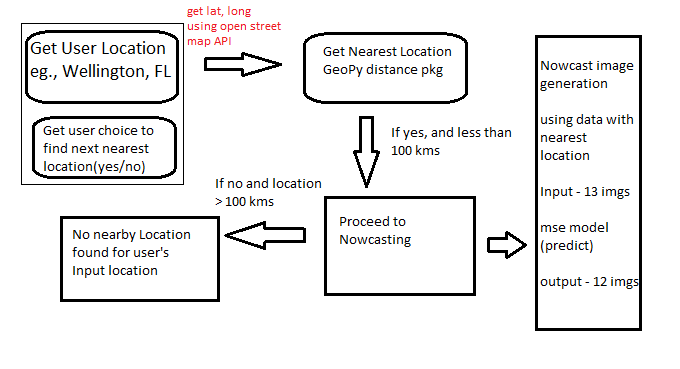
To render next hour’s nowcasting 12 output images based on their image id that ranges from 0-11 where each

image id represents 5 min interval of nowcasted image.

1. Endpoint 2: **/weatherviz/{location, nearest\_location}**

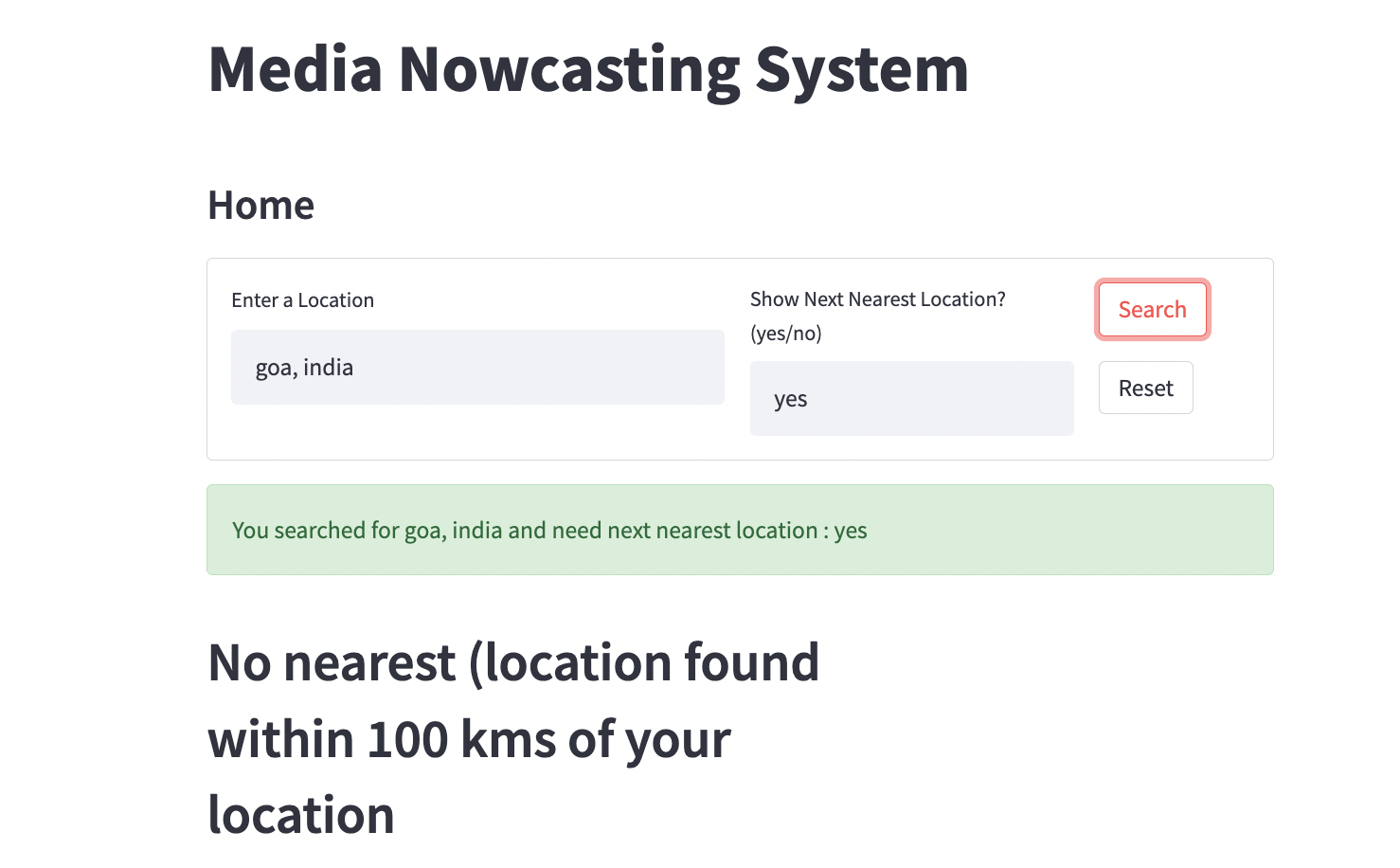
## GeoPy distance logic and Nearest Location

To display the nowcast images gif based on the location and if the location does not exist displaying the nowcast images gif for nearest location based on user request.

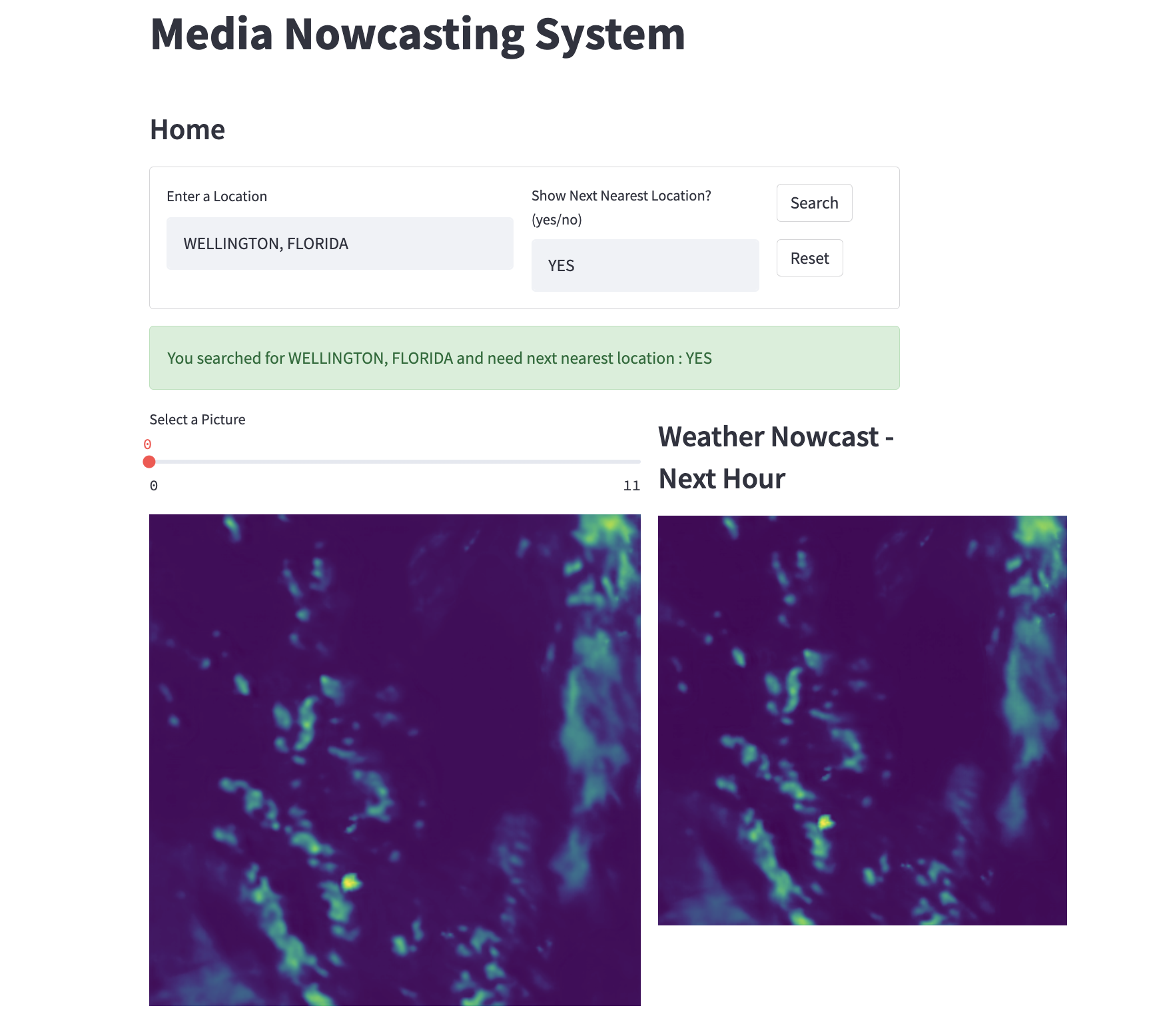


## Positive Test Cases

### If the location is beyond 100 km it will display no nearest location found.

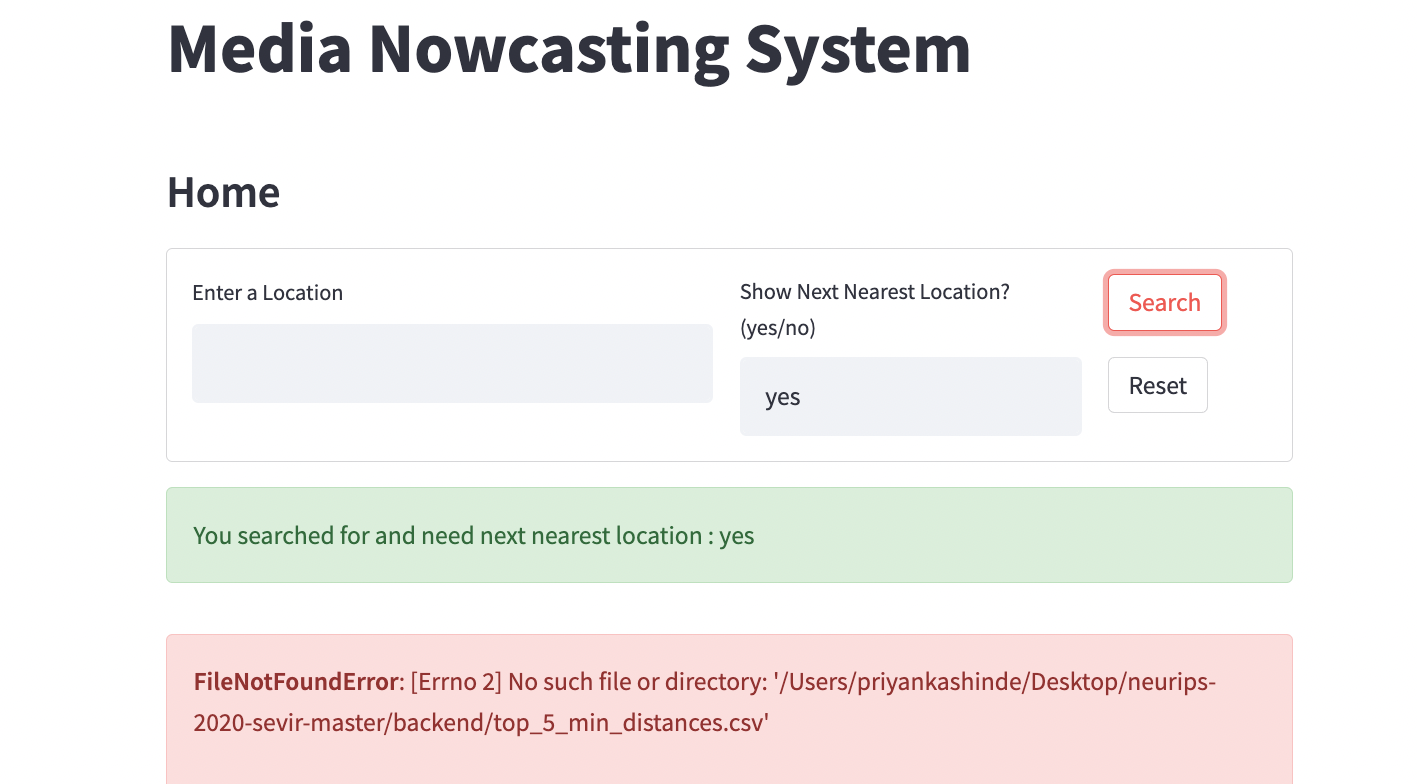


### Giving inputs in caps still gives results.



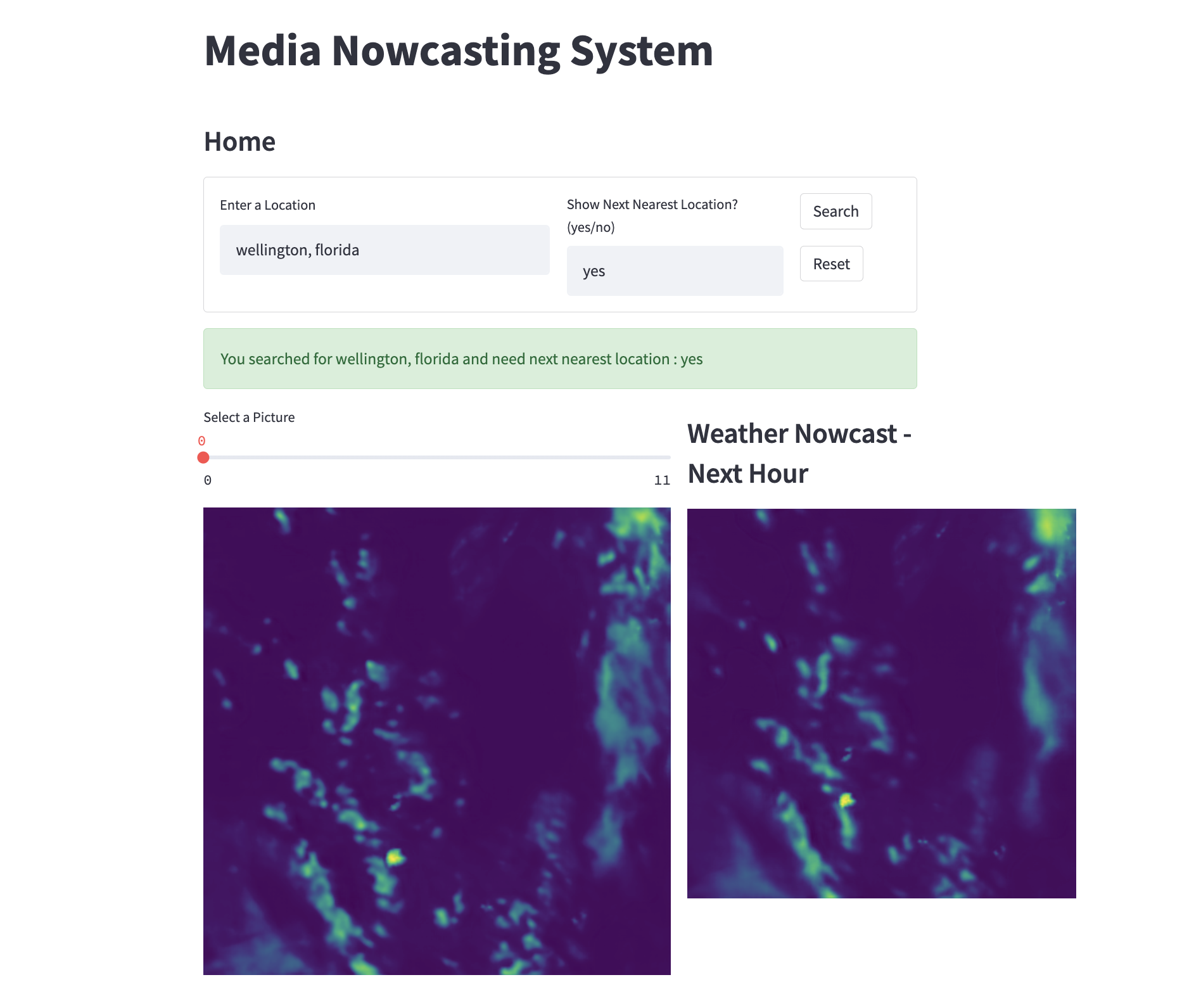
## Negative Test Cases

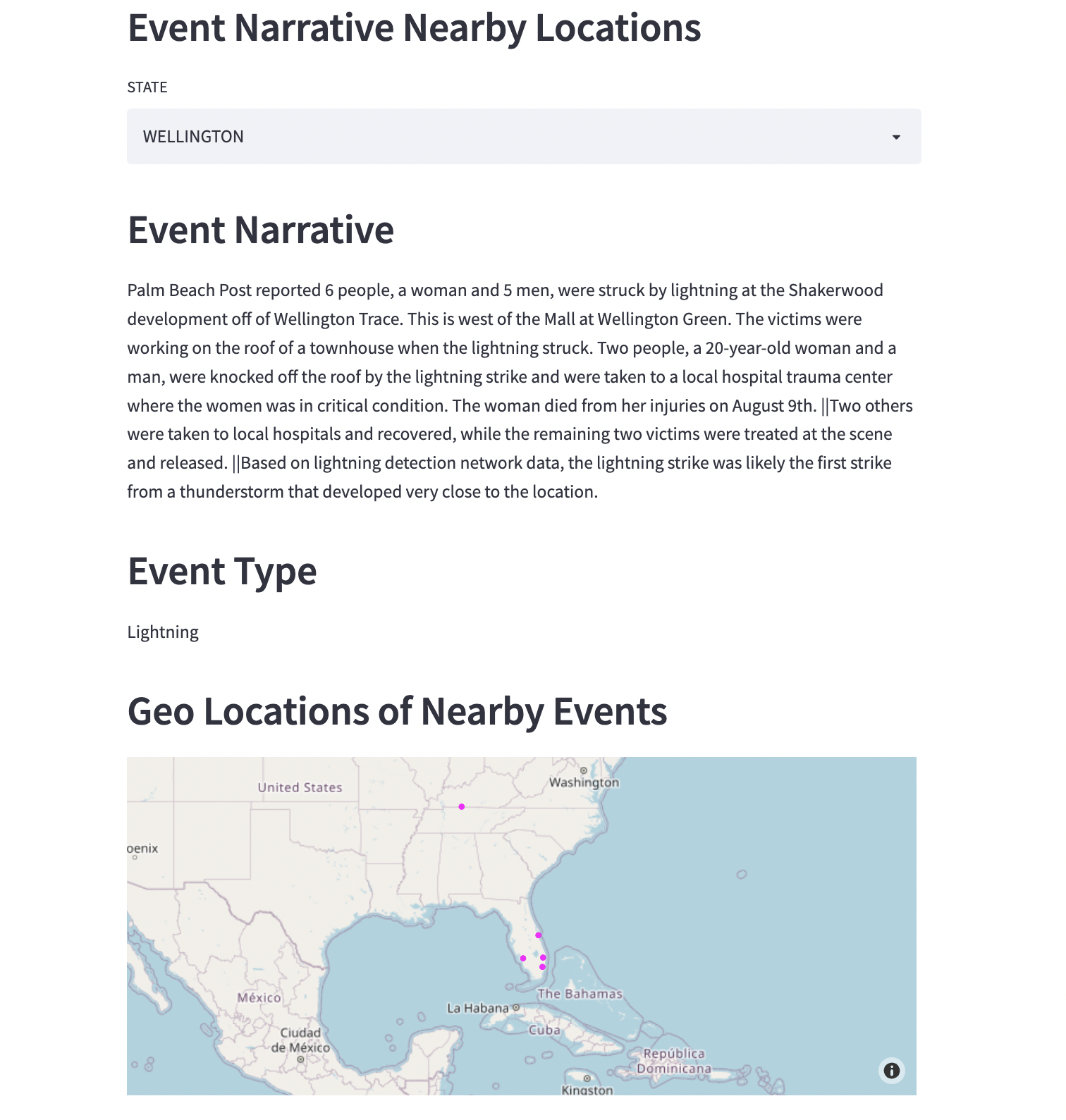
### Giving null values in input field throws error.



# Part 2: New Client (Streamlit)

Integrating Fast API with Streamlit where users can generate nowcast, filter event narrative, and get geographic representation of storm events happening based on location provided and if the location does not exist user can choose to view the nowcast of nearest location.





# Part 3: Host the API

Deployment of FAST API over Heroku/ Deta Platform

Issues with deployment -

1. Deployment of the branch fails for **heroku** while installing dependencies, default max. Size of Slur(pip packages install on memory) is limited to **512 MB**, but in our use case using tensorflow along with other dependencies itself hogs more than **650 MB**  of data
2. Deployment on **Deta** fails because of **packages install timeout** - default timeout for installing of new package in requirements.txt is 10 seconds, if it takes more than this time to fetch the package, it times out. Dependency installation fails.

# 

# Conclusion

1. Did logic enhancement by allowing the user to generate a nowcast based on choice of location and if the location does not exist asked the user if they want the nearest location nowcast.
2. Integrated Fast API with Streamlit.

# References

1. <https://realpython.com/fastapi-python-web-apis/#path-parameters-get-an-item-by-id>
2. <https://learning.postman.com/docs/getting-started/introduction/>
3. <https://www.youtube.com/watch?v=vpTAqnAbowo>
4. <https://github.com/gretelai/gretel-blueprints>
5. <https://nbviewer.org/github/MIT-AI-Accelerator/eie-sevir/blob/master/examples/SEVIR_Tutorial.ipynb>
6. <https://gretel.ai/blog/walkthrough-create-synthetic-data-from-a-dataframe-or-csv>