# Incident Tracker Project

Navanti Group x GW

# TABLE OF CONTENTS

# Background:

- Understand conflicts, societal sentiment, and other trends in countries of interest in a granular level efficiently
- Web scraping news articles with machine learning algorithms to track certain matrices in order to indicate societal, political, economical, kinetic environments in Ukraine

# Problems:

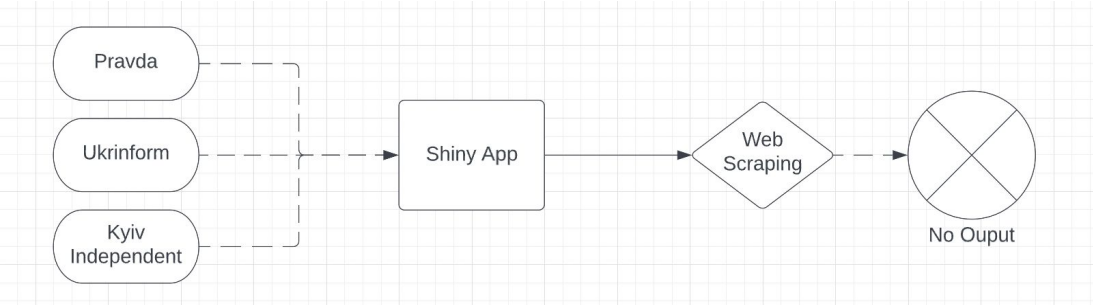
- Existing data model is prone to errors, and cannot create an output
- Need to replace sitemap methodology
- Documentation is vague and disorganized which makes implementation and replication difficult

# Methodology

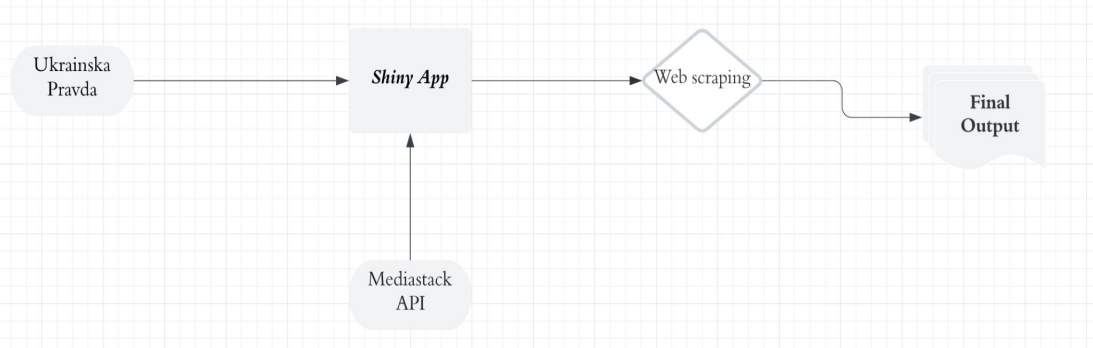
- **Group 1: Improve Original**
  - Debugging and annotating the data model
  - Implementing deep learning and named entity recognition
  - Recreate the desired table for 2023 output
- **Group 2: API Addition**
  - Apply API method to find articles to scrape
  - Implement into original data model
- **Group 3: Usage and Replicability**
  - Documentation of set-up, usage for replicability
  - Organize and update the github

# App Functionality - User Interface

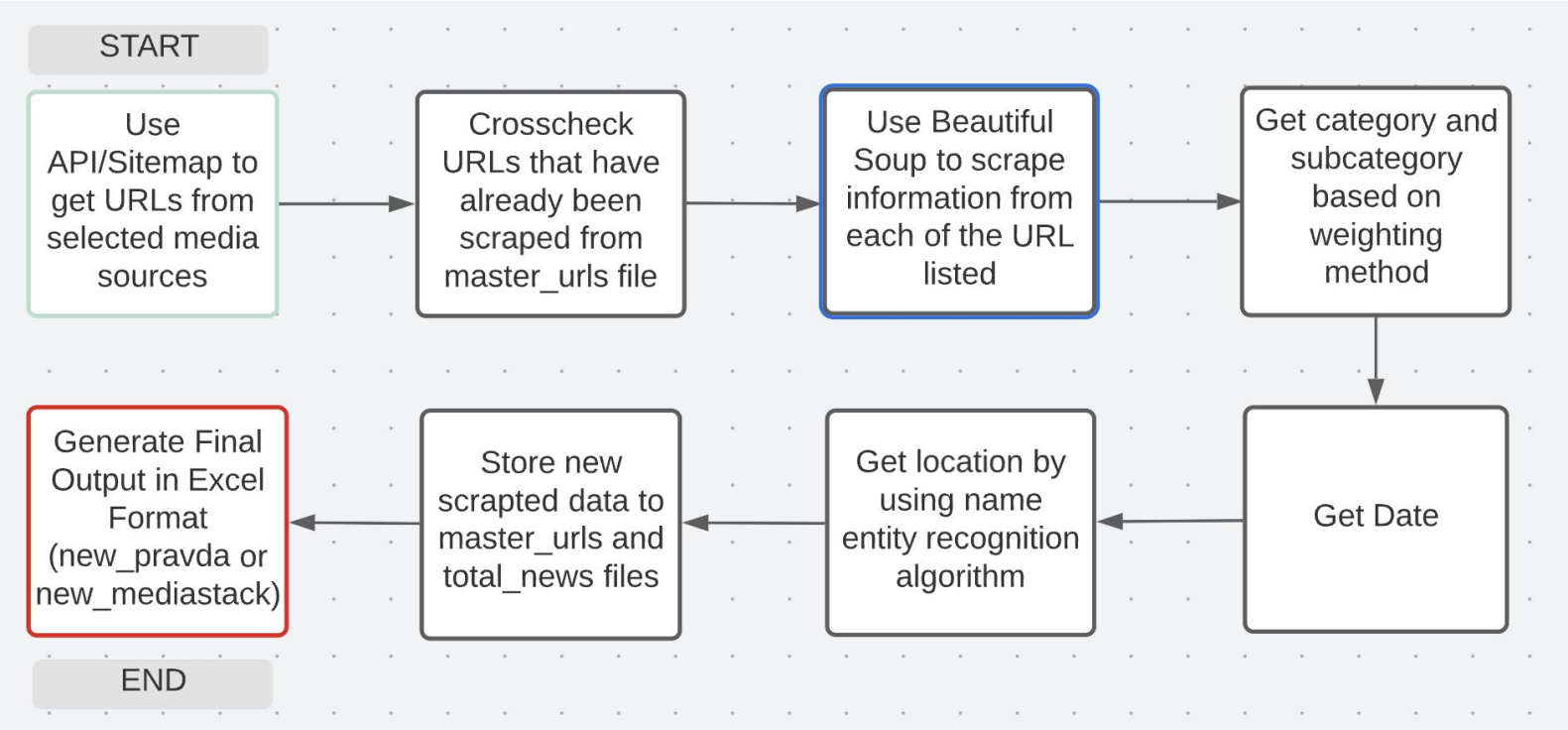
## BEFORE




## AFTER



# App Functionality



# Improve Usability of Data Model

- **Split the code to test simpler runs that don't take so long**
- **Debug the data model and identify and fix all errors that arise**
- **Implement deep learning and named entity recognition**
  - Replace original method to find the location of each incident
  - Create a new function using deep learning and named entity recognition
- **Reduce code repetition by creating separate functions**
  - Get Location
  - Get Category
  - Get Sub_Category
  - Get Date

# Name Entity Recognition (SPACY)

Make this slide later!

# API Addition

## We are adding API because

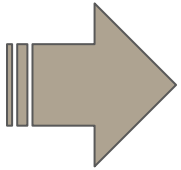
- More and more companies are now adopting APIs on their website for accessing data; more authorized
- The data is already structured
- Less fragile compared to traditional scraping method like beautiful soup
    - Able to handle large amounts of data extraction without any hassle

## Now we are integrating free API into the existing code


- It supports multiple languages
- Able to capture news across countries (7500+ news sources)
- Free plan gives 500 requests per month

# API Output Demo

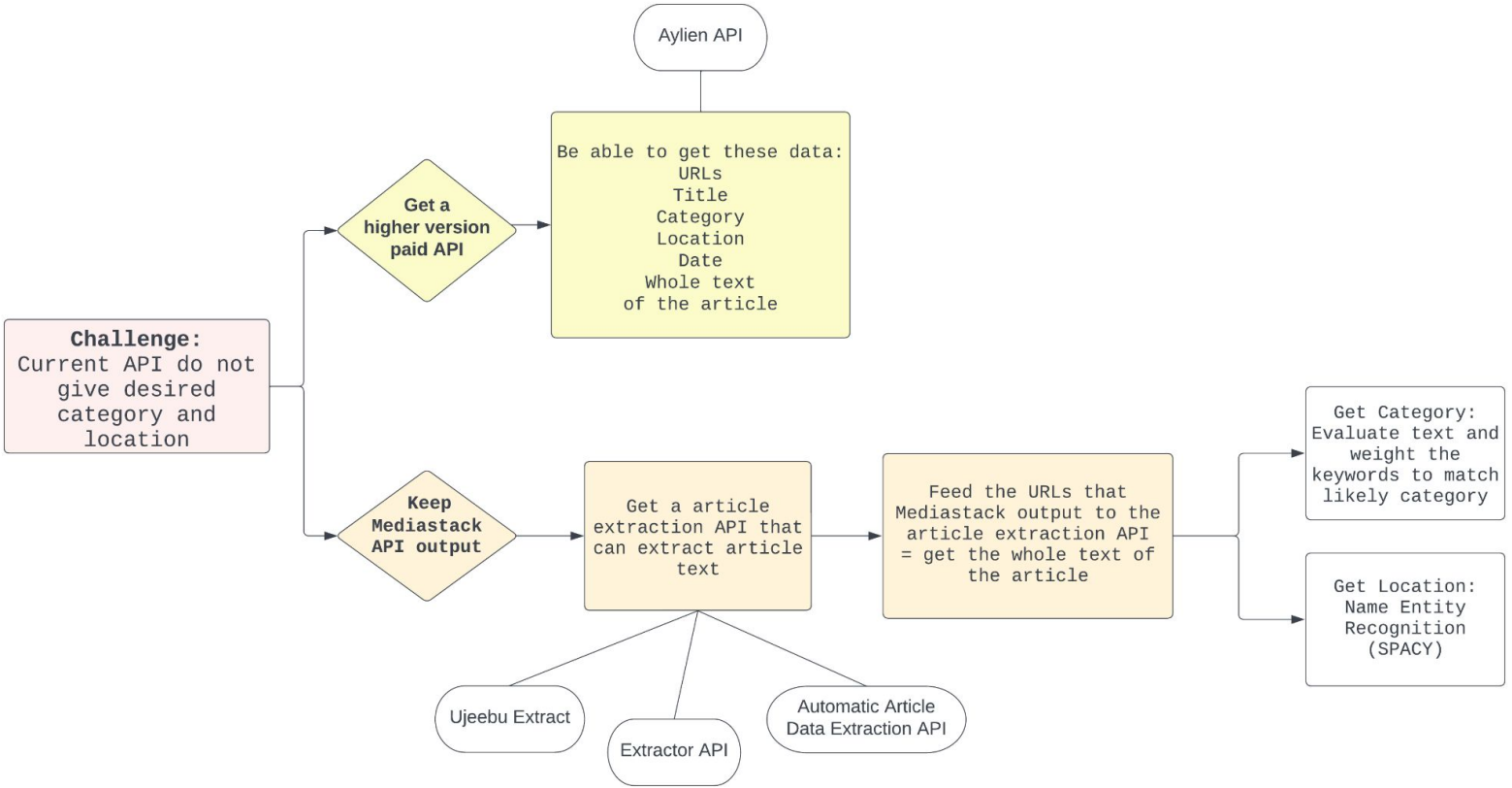
```
import http.client, urllib.parse
import json
import pandas as pd

conn = http.client.HTTPConnection('api.mediastack.com')

params = urllib.parse.urlencode({
    'access_key': 'd9babc2a947d03f9b887715a6df56a7a',
    'categories': 'general',
    'keywords': 'Ukraine Russian',
    'languages': 'ar, en, fr, ru, zh',
    'sort': 'published_desc',
    'limit': 10,
    })

conn.request('GET', "/v1/news?{}".format(params))
res = conn.getresponse()
data = res.read()
```



**Excel file:**

✅URL
✅Title
✅Text (the first paragraph of the article)
✅Author
✅Source (grouped)
✅Date
√Incident Type
√Sub Category
√Latitude
√Longitude

# API Next Steps



# Understanding the Code

- Having annotations (aka reference points)
  - Using docstring (restructured text style)
  - Creating a listing of field with descriptions
  - Creating a diagram of how app functions and how the code works together

# Documentation and Usability

## Transition Packages: User Documentation and Instructions

- For Navanti Group:
  - Entry-level detailed-oriented
  - less technical verbiage

  Examples:
  - how to download certain programs
  - how to get to the dashboard to create the Excel file
  - how the API key was implemented

- For Next Student Group:
  - Detailed-oriented with technical aspects of the code
  - Explanations about what can be improved upon
  - Introduce method to test the code logic without running the entire data model

# Final Deliverables

✅ **Refined existing model:** scraped articles from Pravda news platform

✅ **Timely large-scale data scraping:** scraped articles retrieved by Mediastack API

✅ **Usability:** detailed-oriented user instruction for data model

- one for Navanti and one for the next semester group

✅ **Learning:** in-person learning session for the client

# Future Strategic Risk - Accuracy Concerns

- **Accuracy has not improved due to:**
  - Additional keywords have not been added.
    - The weights of keywords have not been adjusted.
  - There is still repeat and out of scope events being produced by the data model
  - Beautifulsoup
  - Spacy NLP

# Future Strategic Risk - Usage Concerns

- **Key technology used in the data model that needs to be replaced:**
  - Sitemaps → API
  - Beautifulsoup → API
  - Replace function in python → Natural Language Processing
  - Upgrade current API

# Thank you!
# Q&A?

# Appendices

## Code

- App.py here

## GitHub Exhibit

- GitHub here

# Software & Tools Used

- Python
  - Shiny Application
- Visual Studio
- Github

# Goals:

- **Getting the Data Model to create an output**
  - Debugging and rebuilding certain functions
  - Optimizing the code so it is robust to failures
  - Increase functionality and improved performance
- **Implementing API branch to increase web-scraping scope**
  - Will add more news resources
  - Build foundations to use APIs and deep learning
- **Adding user documentation**
  - Writing comments for ease of use and reference

# WHAT WE ARE WORKING ON

## Incidents Tracker

- Keeps tracking certain matrices in order to indicate societal, political, economical, kinetic environments
- Understand conflicts, societal sentiment, and other trends in countries of interest in a granular level efficiently

## Improve Existing Capabilities

- Existing model is prone to errors
  - Make it more robust and be able to scape requested news websites and twitter

## Usability

- Current is hard to use for a user from a non-tech background
  - Create clear annotation and documentation for future implementation

## Accuracy

- Identify events and categorize them based on the ontology provided by Navanti Group