



SI 507

# PROJECT DOCUMENT

Ashruti Tuteja

University of Michigan



## Table of Contents

<b><i>Project code.....</i></b>	<b><i>2</i></b>
<b>Link to Github repo: .....</b>	<b>2</b>
<b>README .....</b>	<b>2</b>
Preparation of the execution environment.....	2
Activate .....	2
<b><i>Brief Interaction.....</i></b>	<b><i>2</i></b>
<b><i>Data sources.....</i></b>	<b><i>3</i></b>
<b><i>Data Structure.....</i></b>	<b><i>4</i></b>
<b>README .....</b>	<b>4</b>
<b>JSON .....</b>	<b>4</b>
<b>Screenshot of the data and data structures.....</b>	<b>4</b>
Flu data.....	4
.....	4
Data Structure .....	4
Covid-19 data .....	5
Data Structure .....	5
<b><i>Interaction and Presentation Options .....</i></b>	<b><i>6</i></b>
<b>Displaying data .....</b>	<b>6</b>
<b>Interactive and presentation technologies .....</b>	<b>6</b>
<b>Interaction.....</b>	<b>7</b>
<b><i>Demo Video link .....</i></b>	<b><i>7</i></b>

## Project code

Link to Github repo: [https://github.com/AshrutiT/CDC\\_Covid-Flu.git](https://github.com/AshrutiT/CDC_Covid-Flu.git)

### README

The following section provides an overview of the steps for executing the programming and how to interact with the web app:

#### Preparation of the execution environment

- Open VS Code, clear cache (if needed)
- Open the working directory then go to Terminal > New
- Type “flask run”

Still if doesn't work for you, then prepare the system to support flask for your working directory:

- Create a virtual environment in a folder of your choosing using, *python -m venv flask*
- Navigate to your working directory folder and activate your virtual environment.
- Activate the virtual environment using the appropriate command.

Or use the link for guidance to flask: <https://flask.palletsprojects.com/en/1.1.x/installation/#install-create-env>

#### Activate

- Copy the app.py file and the templates and static folders to the scripts directory.
- Install the required packages in the working directory. These packages include *flask*, *plotly*, *lxml*, and *pandas*.
- Open a terminal in VSCode and execute the following command, *flask run*

The README file: [ReadMe\\_tashruti.txt](#)

## Brief Interaction

After executing the flask run command, the program will call the tree which contains some user interactions requiring the user to select the data collection option and whether the user would like to proceed with the execution.

- User is here supposed to supply the required responses to the interactive questions.

After supplying the required input, a link will appear as shown in the snip here which the user will click to be directed to the visualization app.

```
Would you like to continue with the analysis? 1 (Yes) or 0 (no)
1
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

## Data sources

Table 1 below provides an overview of the aspects of the collected datasets.

Aspect		Flu	Covid
Source		<a href="https://www.cdc.gov/flu/weekly/weeklyarchives2021-2022/data/whoAllregt_cl13.html">https://www.cdc.gov/flu/weekly/weeklyarchives2021-2022/data/whoAllregt_cl13.html</a> while the covid-19	from <a href="https://data.cdc.gov/api/views/9mfq-cb36/rows.csv?accessType=DOWNLOAD">https://data.cdc.gov/api/views/9mfq-cb36/rows.csv?accessType=DOWNLOAD</a>
Formats		HTML	CSV
Brief Description of how the data was accessed		Includes information related to flu vaccinations, tests, and positive cases that were recorded in 2021 and 2022	Includes information related to covid-19 cases including confirmed cases, new cases, total cases, mortalities, etcetera since the onset of the Covid-19 pandemic
Summary of the data	Records available	6 variables and 26 observations	15 variables and 49560 observations
	Records retrieved	6 variables and 26 observations	15 variables and 49560 observations
	Description of records	Total_A (total number of cases suspected to have Flue type A), Total_B (total number of cases suspected to have Flue type B), Percent_Positive_A (percentage of individuals who had Flue type A), Percent_Positive_B (percentage of individuals who had Flue type B), Total_#_Tested (total number of tested cases), and %_Positive (percentage of tested cases that were positive).	tot_cases (shows the total number of cases per state), the conf_cases (number of confirmed cases in that state), and prob_cases (cases that are suspected but not confirmed), tot_death (fatality cases), new_death (new deaths), state (state from which the cases were recorded),

## Data Structure

### README

A tree was defined which includes both user prompts and displaying options. The first part of the tree included prompting the user which kind of data they would prefer i.e., a cached or a new dataset. The second part introduces the user to what the visualization is all about and whether they are interested in viewing the expected displays before accessing the actual app and lastly, an option of whether they would like to proceed with the analysis. If they choose to, the app collects the data and generates a link. The following files were used for storing the information contained in the tree as well as reading the contents of the tree.

### JSON

The following is a JSON file with the tree used for user interaction: [tree.json](#)

The following file reads the JSON file into a tree: [read\\_json.py](#)

### Screenshot of the data and data structures

#### Flu data

Week	Total A	Total B	Percent Positive A	Percent Positive B	Total # Tested	% Positive
202140	38	28	0.07	0.05	49177	0.13
202141	22	31	0.04	0.06	45787	0.11
202142	26	38	0.05	0.07	48173	0.13
202143	98	33	0.19	0.06	51355	0.25
202144	123	36	0.21	0.06	56373	0.28
202145	309	36	0.49	0.05	62342	0.55
202146	684	45	0.94	0.06	72164	1.01
202147	1265	39	1.78	0.05	70936	1.83
202148	2351	67	2.78	0.07	84270	2.86
202149	3840	49	3.97	0.05	96645	4.02

#### Data Structure

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49620 entries, 0 to 49619
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   submission_date     49620 non-null  object 
 1   state               49620 non-null  object 
 2   tot_cases           49620 non-null  int64  
 3   conf_cases         27420 non-null  float64 
 4   prob_cases         27348 non-null  float64 
 5   new_case           49620 non-null  int64  
 6   pnew_case          45918 non-null  float64 
 7   tot_death          49620 non-null  int64  
 8   conf_death         27007 non-null  float64 
 9   prob_death         27007 non-null  float64 
10   new_death          49620 non-null  int64  
11   pnew_death         45950 non-null  float64 
12   created_at         49620 non-null  object 
13   consent_cases      41345 non-null  object 
14   consent_deaths     42177 non-null  object 
dtypes: float64(6), int64(4), object(5)
memory usage: 5.7+ MB
```

## Covid-19 data

submission_date	state	tot_cases	conf_case	prob_case	new_case	pnew_case	tot_death	conf_deat	prob_deat	new_deat	pnew_deat	created_at	consent_c	consent_deaths
04/25/202	WA	1489269			1287	68	12668			11	0	04/27/2022 12:00:00 AM		
08/17/202	MD	100715			503	0	3765	3616	149	3	0	08/19/2020 12:00:00	Agree	
03/28/202	VT	107785			467	35	585			0	0	03/29/202	Not agree	Not agree
03/18/202	ME	44	44	0	12	0	0	0	0	0	0	03/20/202	Agree	Agree
#####	NE	0			0		0			0		03/26/202	Agree	Agree
#####	DE	134690	124148	10542	444	45	1977	1810	167	2	0	#####	Agree	Agree
#####	WA	440016			366	42	5856			20	0	#####		
#####	VT	1009			10	0	54			0	0	#####	Not agree	Not agree
06/15/202	WI	25480	22932	2548	185	11	700	694	6	2	0	06/16/202	Agree	Agree
03/17/202	GU	46695			45	70	341			0	0	03/18/202	Not agree	Not agree

## Data Structure

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49620 entries, 0 to 49619
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   submission_date        49620 non-null  object
1   state                  49620 non-null  object
2   tot_cases              49620 non-null  int64
3   conf_cases             27420 non-null  float64
4   prob_cases             27348 non-null  float64
5   new_case               49620 non-null  int64
6   pnew_case              45918 non-null  float64
7   tot_death              49620 non-null  int64
8   conf_death            27007 non-null  float64
9   prob_death            27007 non-null  float64
10  new_death              49620 non-null  int64
11  pnew_death             45950 non-null  float64
12  created_at             49620 non-null  object
13  consent_cases          41345 non-null  object
14  consent_deaths         42177 non-null  object
dtypes: float64(6), int64(4), object(5)
memory usage: 5.7+ MB
```

## Interaction and Presentation Options

### Displaying data

There are up to 11 options for displaying data with 7 options for the covid data and 4 for the flu data. The following table includes the options for displaying the covid-19 and flu datasets

Table 2: Data displaying options

Data	Options for displaying the data
<b>Covid-19</b>	<ul style="list-style-type: none"><li>i. Total number of covid cases per year</li><li>ii. Change of new cases over time</li><li>iii. Distribution of confirmed covid cases</li><li>iv. Overview of Covid Data</li><li>v. Examine the correlation between Covid-19 attributes</li><li>vi. Total Number of cases per state</li></ul>
<b>Flu</b>	<ul style="list-style-type: none"><li>i. Number of flu cases per year</li><li>ii. Relationship between the number of tested individuals for flu and positive flu cases</li><li>iii. Distribution of the percentage of positive flu cases</li><li>iv. View Flu Data</li></ul>

Once the user has followed all the steps stipulated in the project code section for setting up the environment, the user needs to only follow the server link to access the application. In the application, the user is provided with an option to use a dark or light mode for the app depending on their preference. There is also a drop-down menu that is given from where the user can select the kind of display they want to interact with.

### Interactive and presentation technologies

The interactive technology used was *Flask* while the presentation technologies that were employed mainly involved *plotly*. Moreover, some basic interactive functionality was designed for the *command line* where the user specifies the data collection options.

## Interaction

The interaction component involves making decisions regarding (a) whether to use cached or collect new files for both flu and covid-19, (b) whether to view the type of visualizations to expect beforehand, (c) whether to continue or quit the analysis.

When executing the program, the user will be prompted to supply some input including a choice of using the cached files or generating new files. Using cached files will prompt the app to use previously downloaded files but overwriting will generate new files that are updated if the period between the last and current execution exceeds 24 hours. The user will also be prompted as to whether you would like to see the kind of visualizations to expect. Depending on the supplied answer, the user will be prompted as to whether to continue with the analysis after which you will see a link to the app or quit.

## Demo Video link

[Demonstration\\_video.mp4](#)

### **GoogleDrive Link:**

<https://drive.google.com/file/d/1QOQEYHzxyhcwQhbUTwyCSyQT-Zc-hae/view?usp=sharing>



**THANK YOU!**