$HOWUP

See what shows are up

ANNABEL CHEONG, DIANE YANEZA, VANESSA JULIANA

What events are on?

What data are we looking for?

Event names

Event dates

Event locations

Event categories

Event descriptions

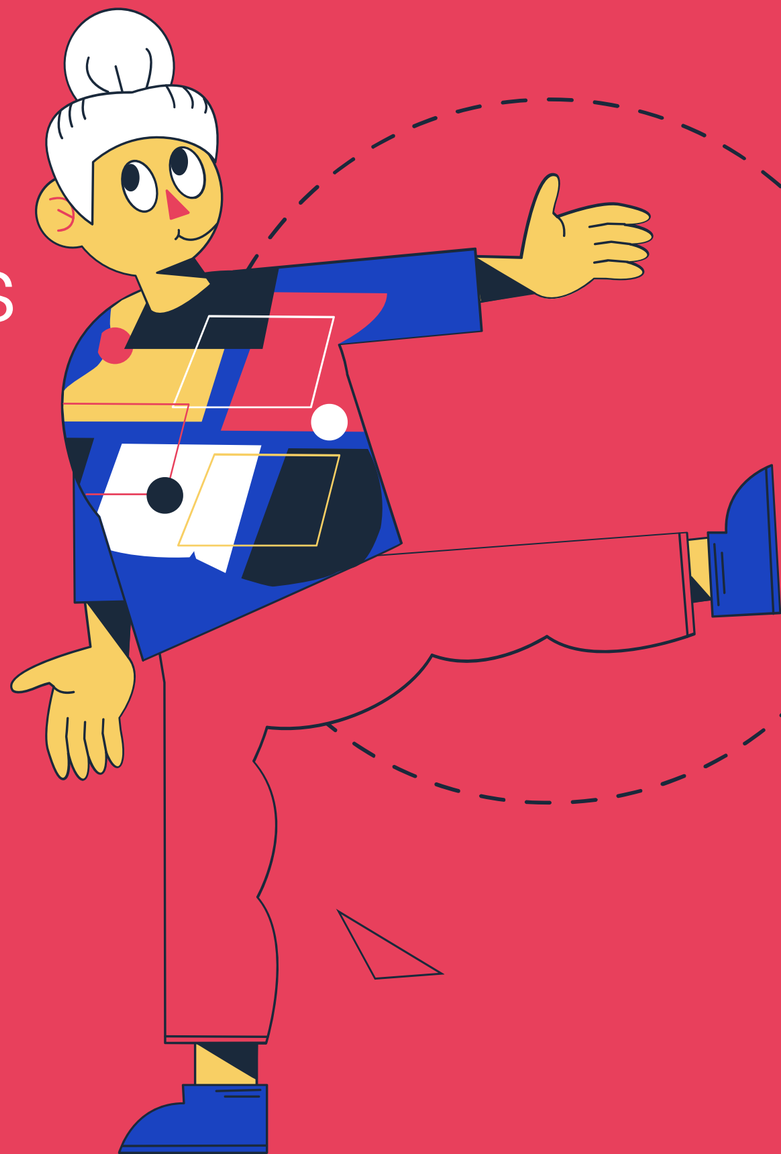
selecting a topic

pulling events dataset

visualising data according to data values

wireframing

Preview



Work Flow

ETL

Extraction

PredictHQ

Free Events API: up to 5000 events

Transformation

python

Jupyter Notebook

Pandas

Library

Load

PostgreSQL

Database

python

Python file: app.py

Flask

Web Server

call data via endpoint

Home endpoint, render html page

reference css file

HTML

reference js file scripts

CSS

JAVASCRIPT

Extraction and Transformation

PredictHQ

Free Events API: up to 5000 events

Challenges

```
data['results'][0]

{'relevance': 1.0,
 'id': 'iv5AMr7UmVxSghaa2N',
 'title': 'Todd Mall Markets',
 'description': 'The Todd Mall Marke
lar venue for tourists and locals fo
ere are a wide range of stalls selli
s: 8am to 1pm from October to April
 'category': 'community',
 'labels': ['community'],
 'rank': 45,
 'local_rank': 74,
 'aviation_rank': None,
 'phq_attendance': 562,
 'entities': [{'entity_id': 'U7KwfB6
   'name': 'Todd Mall Markets',
   'type': 'event-group',
   'category': 'community',
   'labels': ['community', 'event-gr
 'duration': 14400,
 'start': '2021-07-24T23:30:00Z',
 'end': '2021-07-25T03:30:00Z',
 'updated': '2021-04-13T05:07:30Z',
 'first_seen': '2020-12-15T06:40:28Z
 'timezone': 'Australia/Darwin',
 'location': [133.882699, -23.69983]
 'scope': 'locality',
 'country': 'AU',
```

Step 1. Read json from API. Analyse data.

Each call = 50 entries

Each entry = 1 event

Step 2. Convert json to a dataframe using pandas.

Select specific variables.

Step 3. A 'for loop' to attain 1500 entries.

On each loop, offset by another 50.

Step 4. Concatenate dataframes.

Issue

Only extracts keys in the first level of the dictionary

JSON Data: 2nd level (dictionary in a dictionary)

Resolution

Created function used to return 'venue_name' variable

(dictionary in a dictionary).

Created df, calling in function.

```
def getEntitiesVenue(entities):
    try:
        return entities[1]['name']
    except IndexError:
        return 'no venue'
events_df['venue_name'] = events_df.entities.apply(getEntitiesVenue)
```

Load

PostgreSQL, Python (Flask, Pandas, SQLAlchemy), Javascript

We used:

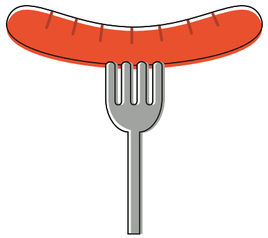
- PostgreSQL to host our database.
- Python Flask to run our application.
- SQLAlchemy to pull the data from Postgres, convert to JSON format, and run our application.
- Pandas to read the database and make last minute changes (e.g. coordinates data).
- Javascript to load our JSONified datasets and render our webpage and three visualisations.

Challenges

Where do we start.....

Key challenges:

- Script to host our database on PostgreSQL ran without a hitch on Mac, but hit a snag when it ran on Windows!
- When the API data was extracted, transformed and loaded using various programs, it converted the coordinates data we needed into a string.

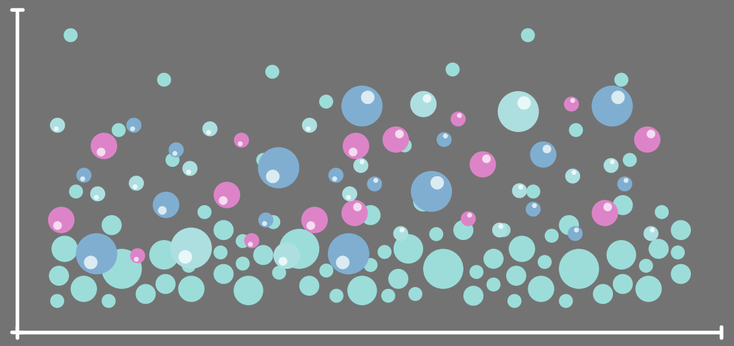


Visualisations

Interactive Map

- D3.js
- Leaflet for interactive map
  - Marker Cluster Group
  - Extra Markers
  - GeoJSON Polygon

Bubble Chart



- D3.js
- Plotly and filter for interactive Bubble Chart
- Litepicker JS library (not covered in class) for Calendar plug in

Table*

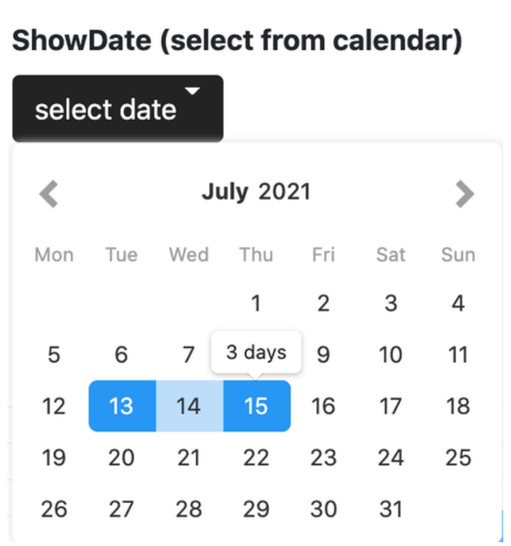
*sorry, I know this is a really bad pun

- D3.js
- Filter for interactive table
- Litepicker JS library (not covered in class) for Calendar plug in

Calendar Plugin

Litepicker

HTTPS://LITEPICKER.COM



HTML

1. Ensure script provided by Litepicker is referenced in html.
2. Create a button for user to click on.
3. Create an id.

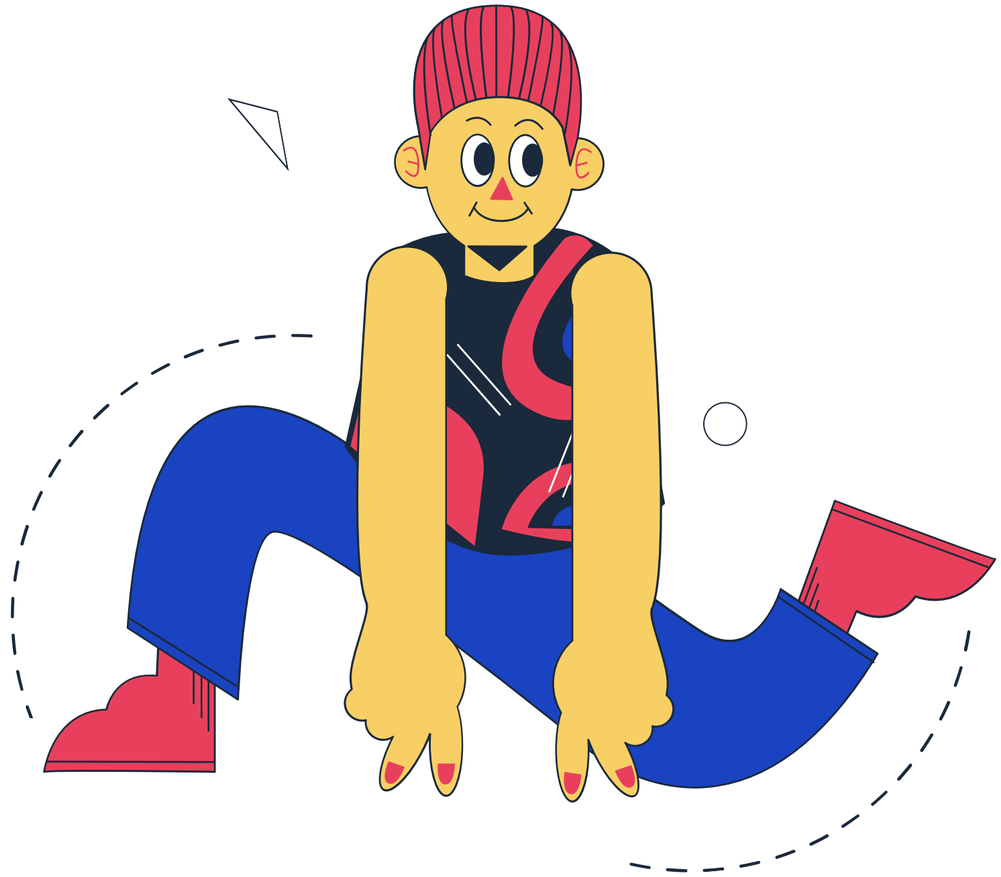
```
<button id="datepicker"
```

JAVASCRIPT

1. Create event listener (code provided in litepicker.com)
2. Extract out the **user selected start and finish date** and convert it to a date format. Use js function: newDate() and getTime()

```
Tue Jul 13 2021 00:00:00 GMT+0800
(Australian Western Standard Time)
```

3. Use filter function to filter for data entries (dictionaries) within date range. (Includes all keys)
4. Use map to assign data to variables. (1 key per variable)
5. Append variables to html table.



Features and Functionality

ShowMap
- map of event locations
- zoom, drag, click: clusters, markers, popups
- displays event name, date/time, category

ShowChart
- bubble chart of event rankings
- zoom, pan, hover, filter by category
- displays event date/time, venue, name

ShowTable
- further details for a list of ten events
- filter by event category, date, name
- displays event date/time, category, name, description, venue, address

Got any questions?