Group 7

POKEMON DASHBOARD

Presented by:

Jai Gupta, Jeremy Oliver, Kali Notaras, Siyuan Zhu,Shelly Wong

PROJECT THEME

Entertainment Industry
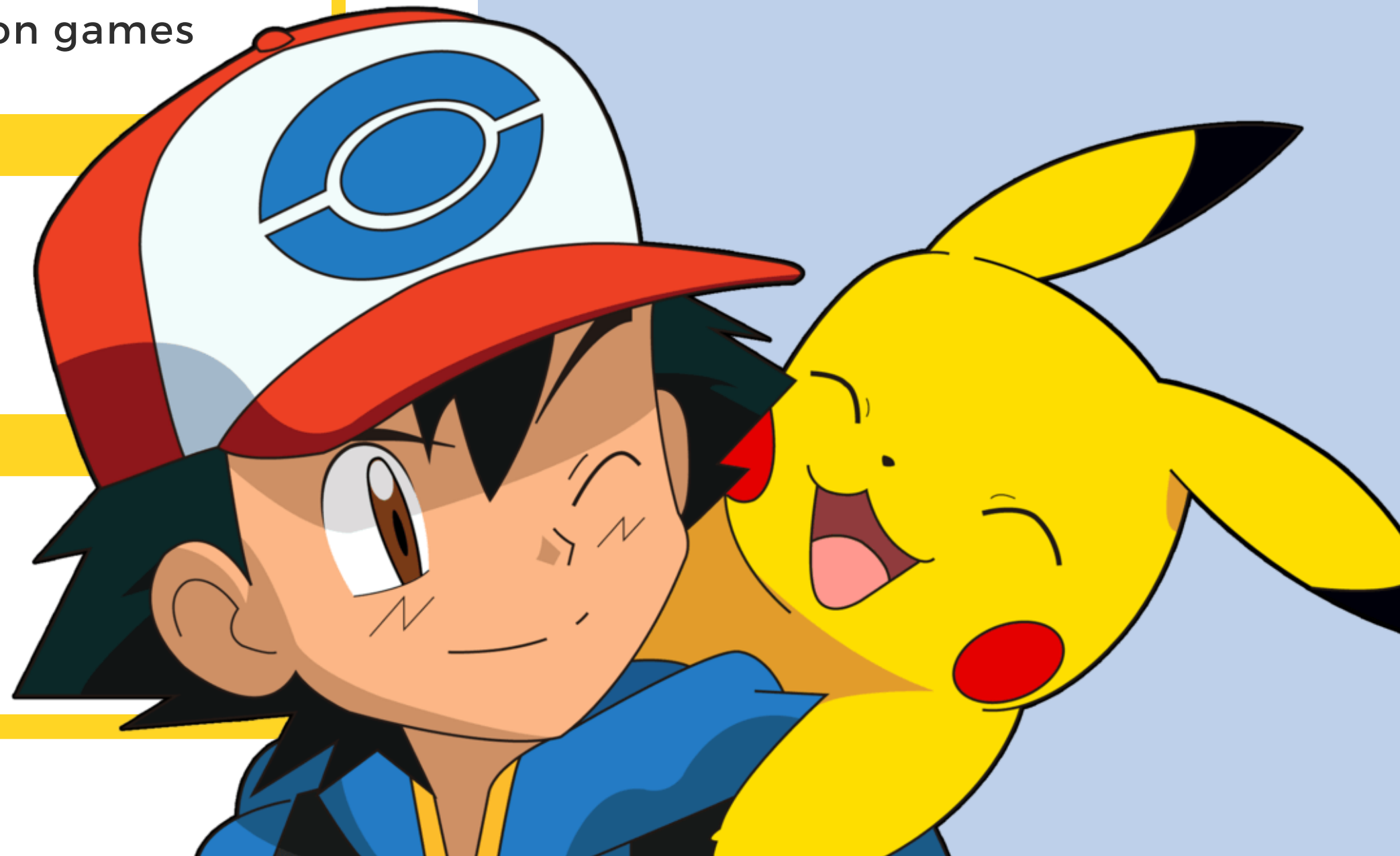
Data Samples: Pokemon

Unique & diverse creatures from the Pokemon games

Target Audience: Pokemon Trainers

People who play any Pokemon games

Rationale: Battling

To win battles & become Champion, you must understand **Statistics** and **Types**

Pokémon
Gotta catch 'em all!

DATA SOURCE

API JSON Formatted Data

Extracted only the data required for our dashboard visualisations

API Resource:
https://pokeapi.co/

Software:
Python, Jupyter Notebook

Packages:
requests, pprint

Pokemon (endpoint)

Pokémon are the creatures that inhabit the world of the Pokémon games. They can be caught using Pokéballs and battling with other Pokémon. Each Pokémon belongs to a specific species but may take on a variant which makes other Pokémon of the same species, such as base stats, available abilities and typings. See Bulbapedia for greater

`GET` `https://pokeapi.co/api/v2/pokemon/{id or name}/`

```
id: 35
name: "clefairy"
base_experience: 113
height: 6
is_default: true
order: 56
weight: 75
▼ abilities: [] 1 item
  ▼ 0: {} 3 keys
      is_hidden: true
      slot: 3
    ▼ ability: {} 2 keys
        name: "friend-guard"
        url: "https://pokeapi.co/api/v2/ability/132/"
▼ forms: [] 1 item
  ▼ 0: {} 2 keys
      name: "clefairy"
      url: "https://pokeapi.co/api/v2/pokemon-form/35/"
▼ game_indices: [] 1 item
  ▼ 0: {} 2 keys
      game_index: 35
    ▼ version: {} 2 keys
        name: "white-2"
        url: "https://pokeapi.co/api/v2/version/22/"
```

☐ View raw JSON (14.059 kB, 278 lines)

Pokemon (type)

Name	Description	Type
id	The identifier for this resource.	*integer*
name	The name for this resource.	*string*
base_experience	The base experience gained for defeating this Pokémon.	*integer*

ETL PROCESS

1 **Extract**

Jupyter Notebook

Raw data

previous: null
▼ results: Array(20)
  ▶ 0: {name: 'bulbasaur', url: 'https://pokeapi.co/api/v2/pokemon/1/'}
  ▶ 1: {name: 'ivysaur', url: 'https://pokeapi.co/api/v2/pokemon/2/'}
  ▶ 2: {name: 'venusaur', url: 'https://pokeapi.co/api/v2/pokemon/3/'}
  ▶ 3: {name: 'charmander', url: 'https://pokeapi.co/api/v2/pokemon/4/'}
  ▶ 4: {name: 'charmeleon', url: 'https://pokeapi.co/api/v2/pokemon/5/'}
  ▶ 5: {name: 'charizard', url: 'https://pokeapi.co/api/v2/pokemon/6/'}
  ▶ 6: {name: 'squirtle', url: 'https://pokeapi.co/api/v2/pokemon/7/'}
  ▶ 7: {name: 'wartortle', url: 'https://pokeapi.co/api/v2/pokemon/8/'}
  ▶ 8: {name: 'blastoise', url: 'https://pokeapi.co/api/v2/pokemon/9/'}
  ▶ 9: {name: 'caterpie', url: 'https://pokeapi.co/api/v2/pokemon/10/'}

2 **Transform**

Pandas Dataframes

Final Output - CSV Files

	name	order_id	hp	attack	defense	special_attack
0	bulbasaur	1	45	49	49	65
1	ivysaur	2	60	62	63	80
2	venusaur	3	80	82	83	100
3	charmander	4	39	52	43	60
4	charmeleon	5	58	64	58	80
5	charizard	6	78	84	78	109
6	squirtle	7	44	48	65	50

3 **Load**

Pgadmin4 SQL Database

Clean data

```
Query   Query History
1  SELECT * FROM public.metadata
2  ORDER BY order_id ASC
```

Data Output   Messages   Notifications

	name character varying (255)	order_id [PK] integer	weight integer	type_1 character varying (255)	type_2 character varying (2!
1	bulbasaur	1	69	grass	poison
2	ivysaur	2	130	grass	poison

✅ **Benefits of ETL**

Filtered to only the data you need

Tailored & Error-free; most usable format
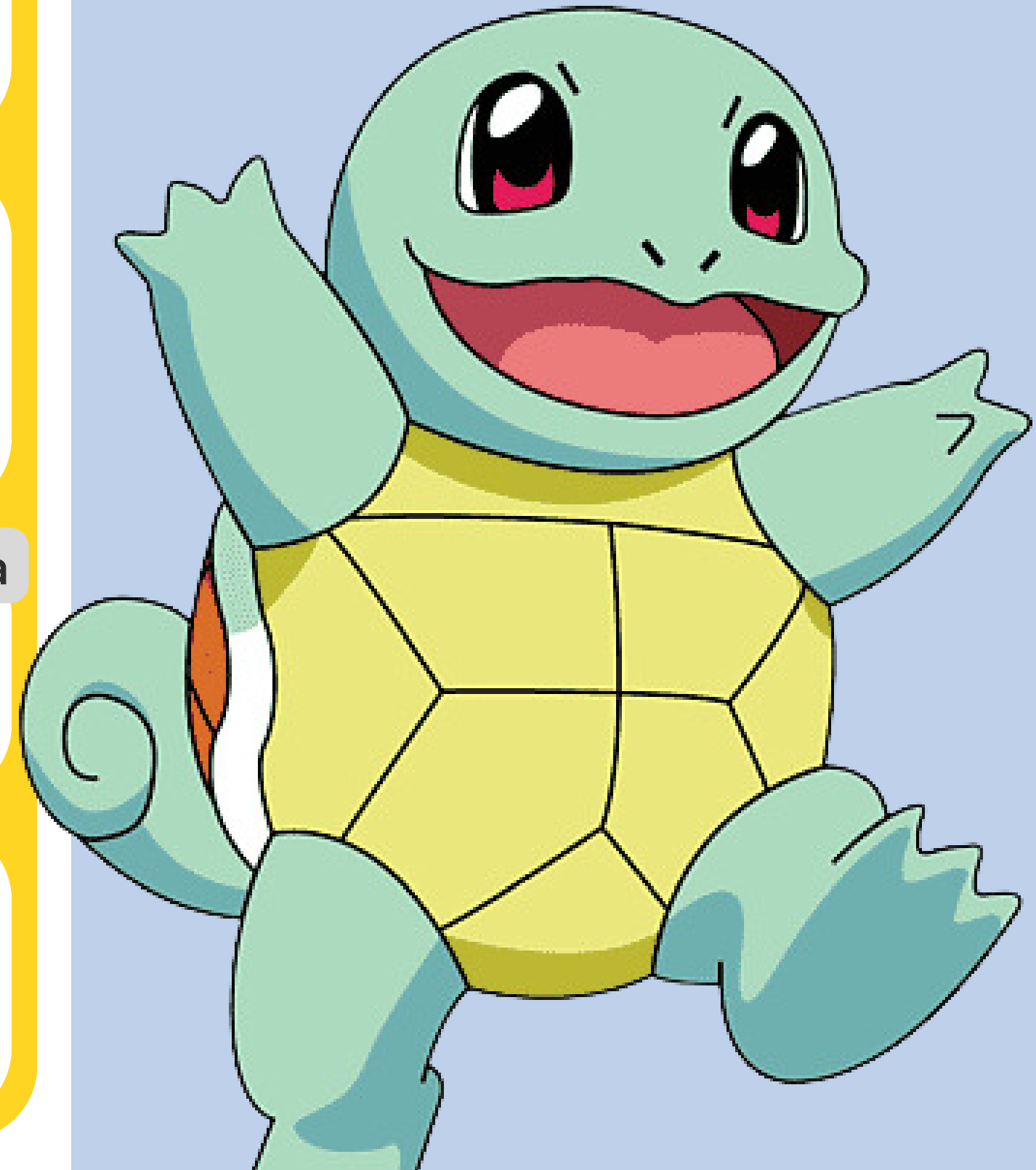
API FLASK

1 **Connect SQL to Flask Python**

SQLAlchemy, Flask, config.py

2 **App Route & Engine**

Summary Route: API of Jsonified SQL data

Index Route: Dashboard HTML

3 **Connect JavaScript & HTML**

Render template via app.py

Final Data

```
▼ Object ℹ
  ▶ metadata: (100) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}
  ▶ sprites: (100) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}
  ▼ statistics: Array(100)
    ▶ 0: {Attack: 49, Defense: 49, Hp: 45, Name: 'bulbasaur', Special_attack: 65, …}
    ▶ 1: {Attack: 62, Defense: 63, Hp: 60, Name: 'ivysaur', Special_attack: 80, …}
    ▶ 2: {Attack: 82, Defense: 83, Hp: 80, Name: 'venusaur', Special_attack: 100, …}
    ▶ 3: {Attack: 52, Defense: 43, Hp: 39, Name: 'charmander', Special_attack: 60, …}
```

JAVASCRIPT & HTML

1 **Design Draft**

Visualisations, Data Required, End-User Function

2 **HTML Framework**

CSS Stylesheet from Bootstrap
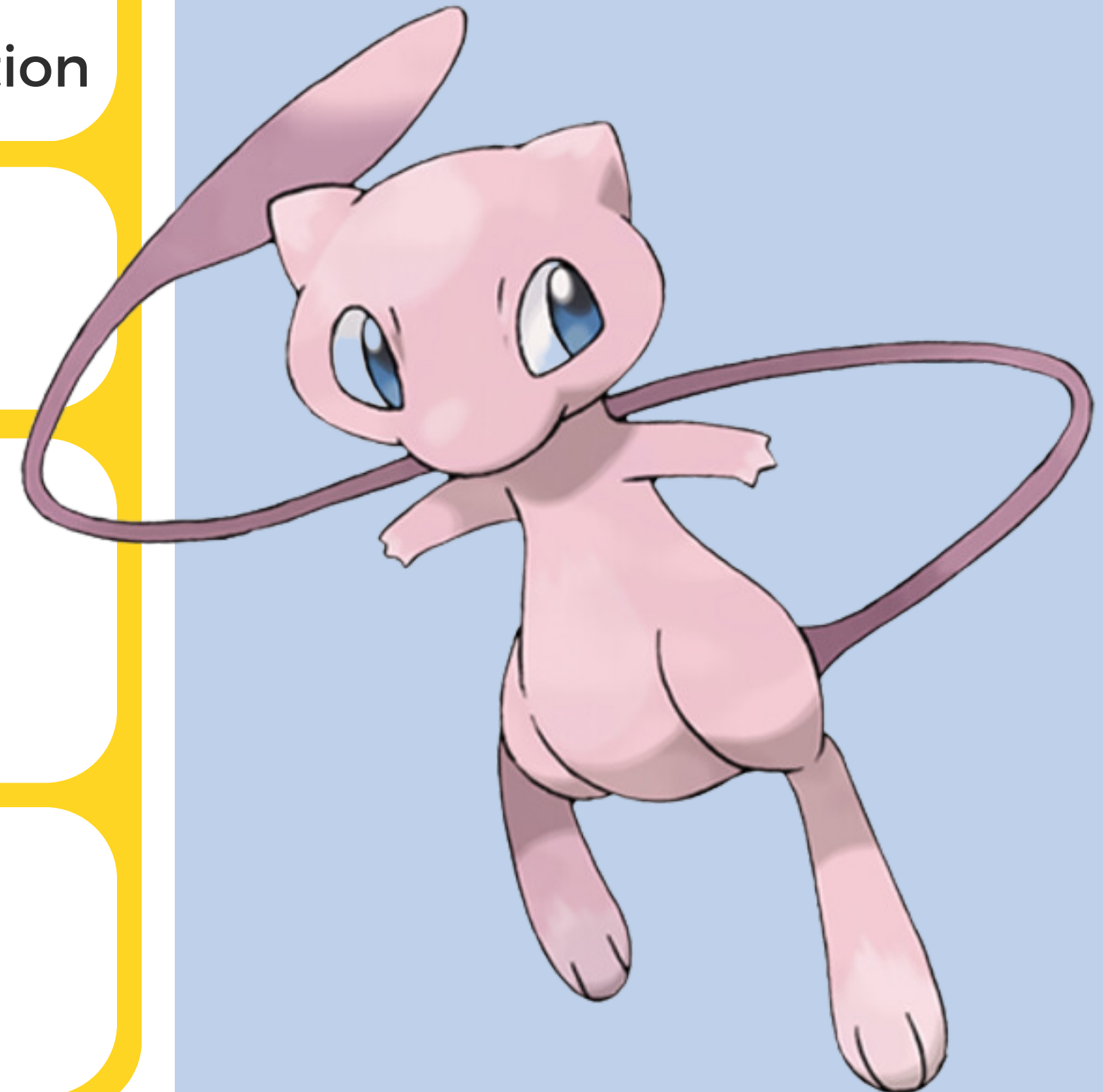
3 **JavaScript**

Visualisations using Plotly.js

Data Wrangling using D3.js

Lets see the final product!

https://github.com/Kopixx/Project-3

LIMITATIONS

- The difficulty to use data extracted directly from external API

- Unable to understand some other data given and its relationships

IMPROVEMENTS

- Include pre-filled information feature for typing pokemon name
- Include all pokemon data

Pokémon

Gotta catch 'em all!

Good luck Trainer!