

the **PandaChams**

Panda Poke

Group 5

Project 3





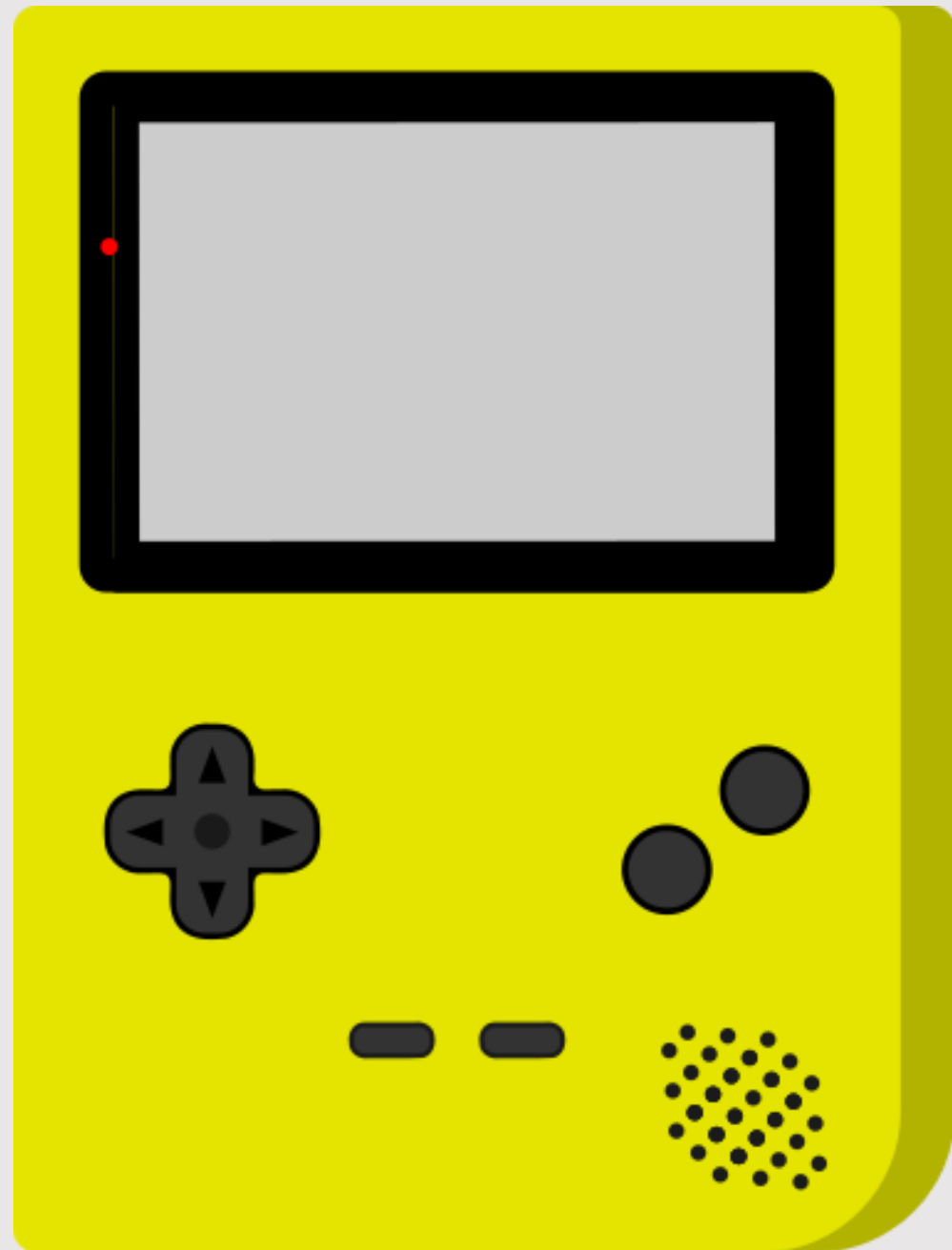
A brief introduction to Pokémon

Originated in Japan.

The original game was a role playing game

Pokémon are divided into various types. Eg. Fire and Water.

Why Pokémon?



A large dataset

Variety of characters and attributes

It is one of the most played games in the world

Required Dependencies and Libraries

- Pandas
- Requests
- Json
- SQLAlchemy
- Config
- Warnings
- Flask

The Libraries which have been used for this project are

- D3 <https://d3js.org/>
- Plotly <https://plotly.com/>
- Chart.js <https://www.chartjs.org/>
- APEXCharts <https://apexcharts.com/>
- Bootstrap & Star Admin 2 – Bootstrap Admin Dashboard

Sources of Data

There were 3 different endpoints used

- Pokemon: <https://pokeapi.co/api/v2/pokemon/>
- Pokemon Species: <https://pokeapi.co/api/v2/pokemon-species/>
- Growth Rate: <https://pokeapi.co/api/v2/growth-rate/>

Dependencies for ETL

Import Dependencies

```
|: import requests
import json
import pandas as pd

from sqlalchemy import create_engine, ForeignKey, Column, String, Integer
from sqlalchemy.ext.declarative import declarative_base

# Saved password in config file which will be gitignored
from config import pw

# Turn off warning messages
import warnings
warnings.filterwarnings("ignore")
```

Data extraction

Extract Pokemon Information

```
In [2]: #Use the Pokemon Species APIs to populate Lists
poke_name = []
poke_happy = []
poke_catch = []
poke_color = []
poke_evolve = []
poke_gender = []
poke_generation = []
poke_growth = []
poke_habitat = []
poke_id = []
poke_shape = []
poke_baby = []
poke_leg = []
poke_myth = []

for s in range(1000):
    url = "https://pokeapi.co/api/v2/pokemon-species/"+str(s+1)
    response = requests.get(url).json()
    poke_name.append(response["name"])
    poke_happy.append(response["base_happiness"])
    poke_catch.append(response["capture_rate"])
    poke_color.append(response["color"]["name"])
    poke_evolve.append(response["evolves_from_species"])
    poke_gender.append(response["gender_rate"])
    poke_generation.append(response["generation"]["name"])
    poke_growth.append(response["growth_rate"]["name"])
    try:
        poke_habitat.append(response["habitat"]["name"])
    except TypeError:
        poke_habitat.append("N/A")
    poke_id.append(response["id"])
    try:
        poke_shape.append(response["shape"]["name"])
    except TypeError:
```

```
In [5]: #Use the Pokemon API to populate additional Lists
poke_id2 = []
poke_ability = []
poke_exp = []
poke_height = []
poke_sprite = []
poke_shiny = []
poke_hp = []
poke_attack = []
poke_def = []
poke_spatk = []
poke_spdef = []
poke_speed = []
poke_type1 = []
poke_type2 = []
poke_weight = []

for p in range(1000):
    url = "https://pokeapi.co/api/v2/pokemon/"+str(p+1)
    response = requests.get(url).json()
    poke_id2.append(response["id"])
    poke_ability.append(response["abilities"][0]["ability"]["name"])
    poke_exp.append(response["base_experience"])
    poke_height.append(response["height"])
    poke_sprite.append(response["sprites"]["front_default"])
    poke_shiny.append(response["sprites"]["front_shiny"])
    poke_hp.append(response["stats"][0]["base_stat"])
    poke_attack.append(response["stats"][1]["base_stat"])
    poke_def.append(response["stats"][2]["base_stat"])
```


Merging and Cleaning

```
[15]: #Remove and Rename unwanted columns
df_poke_named = poke_merge_2.rename(columns={'id_x' : 'poke_id', 'name_x' : 'name', 'height
      'type_1_y' : 'type_1', 'type_2' : 'type_2', 'color' : 'color', 'sh
      'base_hp' : 'base_hp', 'base_attack': 'base_attack', 'base_def': 'ba
      'name_y' : 'evolves_from', 'habitat': 'habitat', 'catch_rate' : 'c
      'standard_pic' : 'standard_pic' , 'shiny_pic' : 'shiny_pic'})

df_poke_named.head(10)
```

df_poke_named.head(10)

	poke_id	name	height	weight	gender_rate	type_1	type_2	color	shape	growth_rate	...	base_sp
0	1	bulbasaur	7	69	1	grass	poison	green	quadruped	medium-slow	...	
1	2	ivysaur	10	130	1	grass	poison	green	quadruped	medium-slow	...	
2	3	venusaur	20	1000	1	grass	poison	green	quadruped	medium-slow	...	
3	4	charmander	6	85	1	fire	NaN	red	upright	medium-slow	...	
4	5	charmeleon	11	190	1	fire	NaN	red	upright	medium-slow	...	
5	6	charizard	17	905	1	fire	flying	red	upright	medium-slow	...	
6	7	squirtle	5	90	1	water	NaN	blue	upright	medium-slow	...	
7	8	wartortle	10	225	1	water	NaN	blue	upright	medium-slow	...	
8	9	blastoise	16	855	1	water	NaN	blue	upright	medium-slow	...	
9	10	caterpie	3	29	4	bug	NaN	green	armor	medium	...	

10 rows x 24 columns

10 rows x 24 columns

```
[16]: #Remove Null Values
df_poke_named["type_2"].fillna("None",inplace=True)
df_poke_named["evolves_from"].fillna("Base",inplace=True)
df_poke_named.head(10)
```

df_poke_named.head(10)

	poke_id	name	height	weight	gender_rate	type_1	type_2	color	shape	growth	...	base_sp
0	1	bulbasaur	7	69	1	grass	poison	green	quadruped	me	...	
1	2	ivysaur	10	130	1	grass	poison	green	quadruped	me	...	
2	3	venusaur	20	1000	1	grass	poison	green	quadruped	me	...	
3	4	charmander	6	85	1	fire	None	red	upright	me	...	
4	5	charmeleon	11	190	1	fire	None	red	upright	me	...	
5	6	charizard	17	905	1	fire	flying	red	upright	me	...	
6	7	squirtle	5	90	1	water	None	blue	upright	me	...	
7	8	wartortle	10	225	1	water	None	blue	upright	me	...	
8	9	blastoise	16	855	1	water	None	blue	upright	me	...	
9	10	caterpie	3	29	4	bug	None	green	armor	me	...	

10 rows x 24 columns

DB connetion and Table creation

LOADING DATA INTO DATABASE

```
In [26]: protocol = 'postgresql'
username = 'postgres'
password = pw
host = 'localhost'
port = 5432
database_name = 'pandachams_db'
rds_connection_string = f'{protocol}://{username}:{password}@{host}:{port}'
engine = create_engine(rds_connection_string)

Base = declarative_base()

In [27]: # Check for existing tables before creation
engine.table_names()

Out[27]: []

In [28]: print(final_poke_df.columns.tolist())

['poke_id', 'name', 'height', 'weight', 'male_rate', 'female_rate', 'gender', 'growth_rate', 'base_hp', 'base_attack', 'base_def', 'base_sp_attack', 'base_sp_def', 'catch_rate', 'is_baby', 'is_legendary', 'is_mythical', 'standard_picture']

In [29]: final_poke_df.dtypes

Out[29]: poke_id          int64
name              object
height           int64
```

```
In [32]: # Creating growth_rate_species table
class poke(Base):
    __tablename__ = "growth_rate_species"
    extend_existing=True

    id = Column("id", Integer, primary_key = True, autoincrement = True)
    growth_rate = Column("growth_rate", String)
    species_name = Column("species_name", String)

In [33]: growth_rate_levels.dtypes

Out[33]: growth_rate    object
levels              int64
exp                 int64
dtype: object

In [34]: # Creating growth_rate_levels table
class poke(Base):
    extend_existing=True
    __tablename__ = "growth_rate_levels"

    id = Column("id", Integer, primary_key = True, autoincrement = True)
    growth_rate = Column("growth_rate", String)
    levels = Column("levels", Integer)
    exp = Column("exp", Integer)

In [35]: Base.metadata.create_all(bind = engine)

In [36]: # Checking for existing tables after creation
engine.table_names()

Out[36]: ['poke', 'growth_rate_species', 'growth_rate_levels']

In [37]: final_poke_df.to_sql(name='poke', con=engine, if_exists='append', index=False)

Out[37]: 1000

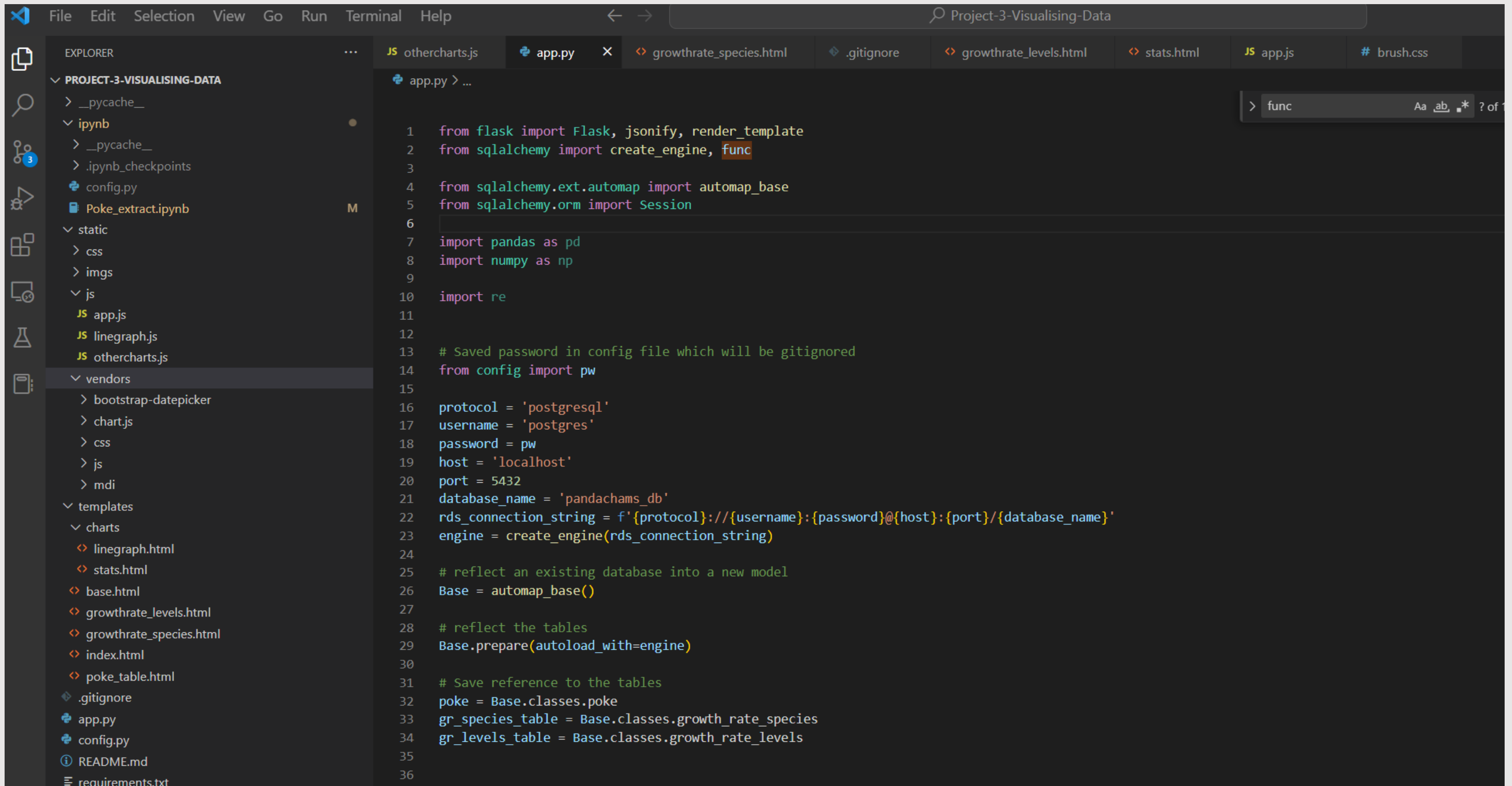
In [38]: growth_rate_species.to_sql(name='growth_rate_species', con=engine, if_exists='append', index=False)

Out[38]: 8

In [39]: growth_rate_levels.to_sql(name='growth_rate_levels', con=engine, if_exists='append', index=False)

Out[39]: 600
```

The File System and Flask



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure for 'PROJECT-3-VISUALISING-DATA'. The main editor area shows the 'app.py' file, which contains Python code for setting up a Flask application with a SQLAlchemy database engine and models.

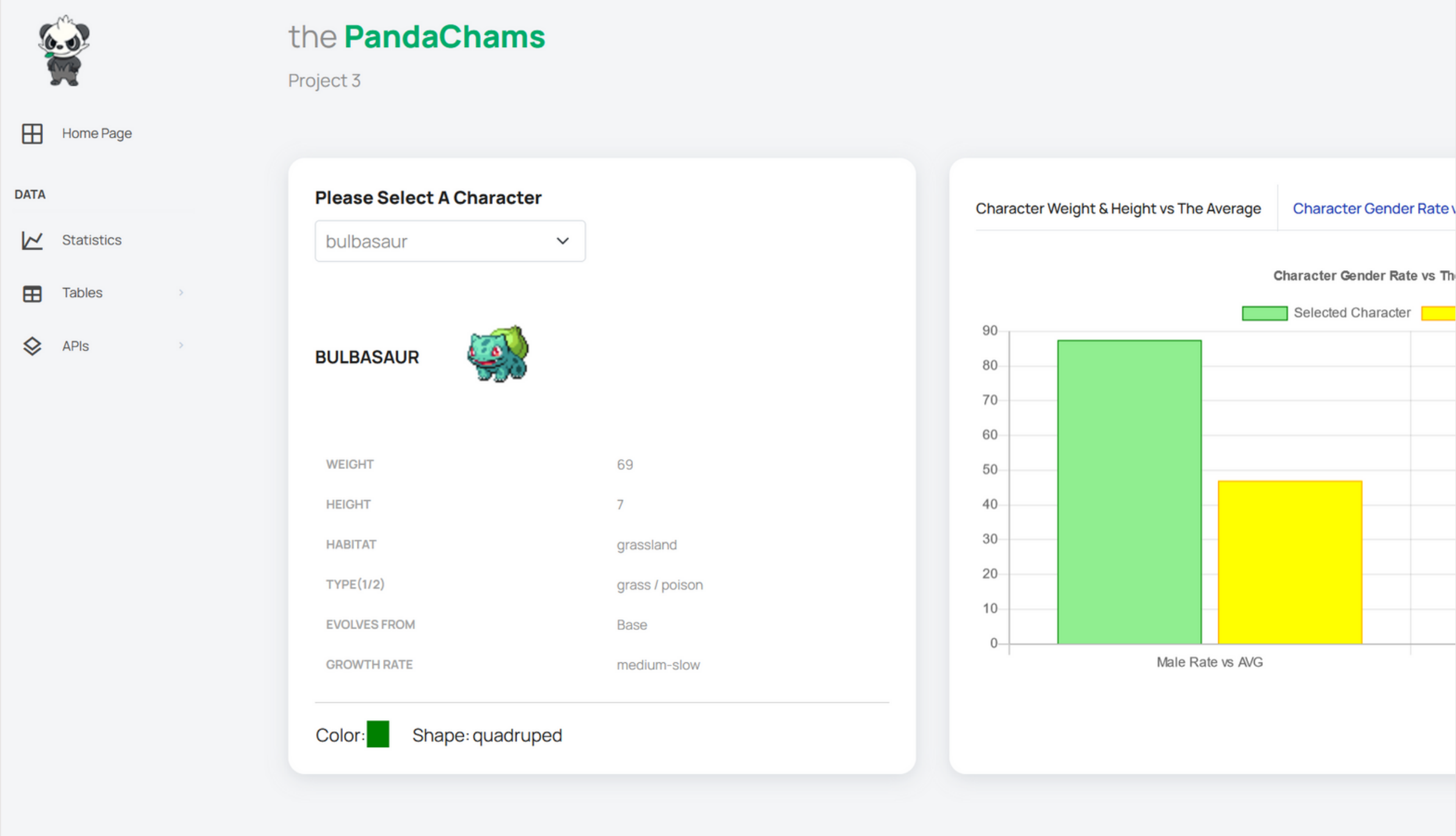
EXPLORER

- PROJECT-3-VISUALISING-DATA
 - > __pycache__
 - > ipynb
 - > __pycache__
 - > .ipynb_checkpoints
 - config.py
 - Poke_extract.ipynb
 - static
 - > css
 - > imgs
 - js
 - app.js
 - linegraph.js
 - othercharts.js
 - vendors
 - > bootstrap-datepicker
 - > chart.js
 - > css
 - > js
 - > mdi
 - templates
 - charts
 - linegraph.html
 - stats.html
 - base.html
 - growthrate_levels.html
 - growthrate_species.html
 - index.html
 - poke_table.html
 - .gitignore
 - app.py
 - config.py
 - README.md
 - requirements.txt

app.py

```
1 from flask import Flask, jsonify, render_template
2 from sqlalchemy import create_engine, func
3
4 from sqlalchemy.ext.automap import automap_base
5 from sqlalchemy.orm import Session
6
7 import pandas as pd
8 import numpy as np
9
10 import re
11
12
13 # Saved password in config file which will be gitignored
14 from config import pw
15
16 protocol = 'postgresql'
17 username = 'postgres'
18 password = pw
19 host = 'localhost'
20 port = 5432
21 database_name = 'pandachams_db'
22 rds_connection_string = f'{protocol}://{username}:{password}@{host}:{port}/{database_name}'
23 engine = create_engine(rds_connection_string)
24
25 # reflect an existing database into a new model
26 Base = automap_base()
27
28 # reflect the tables
29 Base.prepare(autoload_with=engine)
30
31 # Save reference to the tables
32 poke = Base.classes.poke
33 gr_species_table = Base.classes.growth_rate_species
34 gr_levels_table = Base.classes.growth_rate_levels
35
36
```

The Landing Page



Locally Gerated APIs

JSON	Raw Data	Headers
Save Copy Collapse All Expand All (slow) Filter JSON		
0:		
base_attack:	49	
base_def:	49	
base_hp:	45	
base_sp_attack:	65	
base_sp_def:	65	
base_speed:	45	
catch_rate:	45	
color:	"green"	
evolves_from:	"Base"	
female_rate:	12.5	
gender_neutral_rate:	0	
growth_rate:	"medium-slow"	
habitat:	"grassland"	
height:	7	
is_baby:	false	
is_legendary:	false	
is_mythical:	false	
male_rate:	87.5	
name:	"bulbasaur"	
poke_id:	1	
shape:	"quadruped"	
shiny_pic:	"https://raw.githubusercontent.com/pokemonshiny/1.png"	
standard_pic:	"https://raw.githubusercontent.com/sprites/pokemon/1.png"	
type_1:	"grass"	
type_2:	"poison"	
weight:	69	
1:	{...}	
2:	{...}	
3:	{...}	
4:	{...}	
5:	{...}	

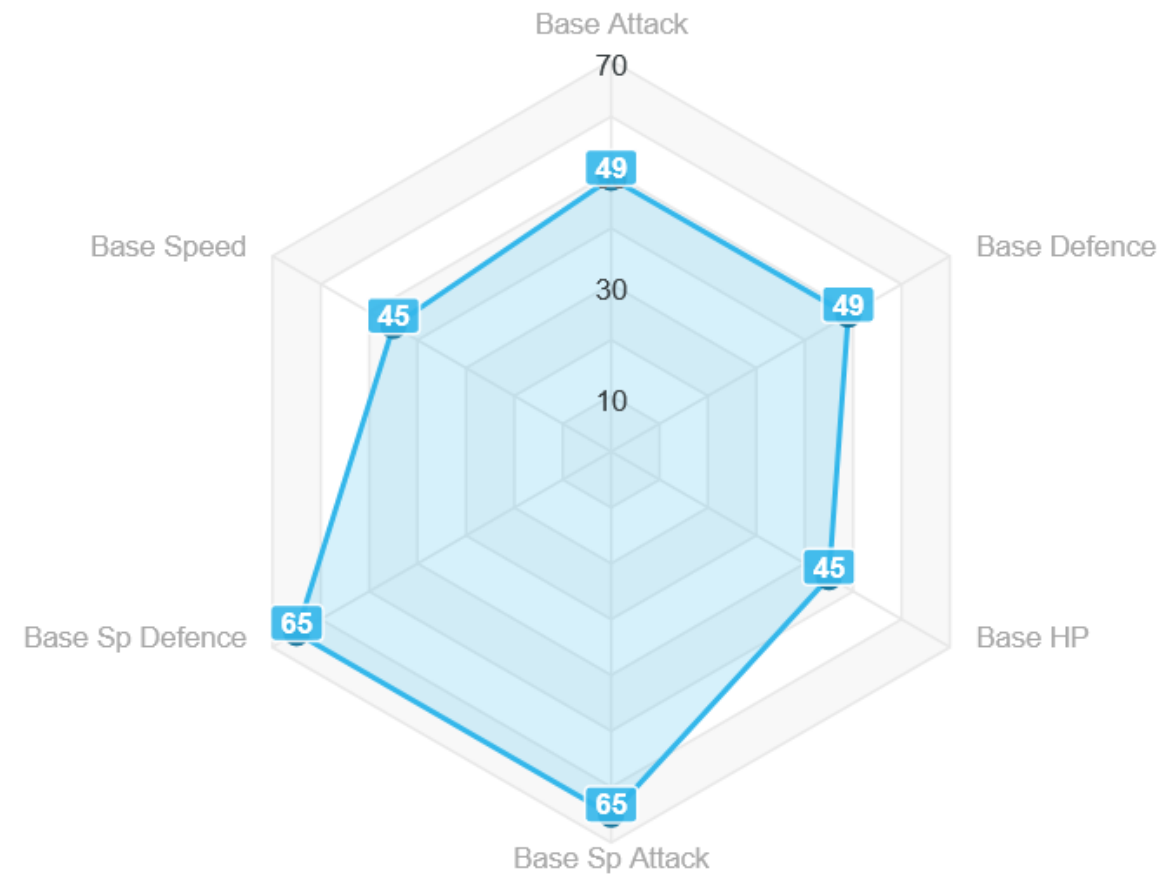
JSON	Raw Data	Headers
Save Copy Collapse All Expand All Filter JSON		
0:		
exp:	0	
growth_rate:	"slow"	
id:	1	
level:	1	
1:		
exp:	10	
growth_rate:	"slow"	
id:	2	
level:	2	
2:		
exp:	33	
growth_rate:	"slow"	
id:	3	
level:	3	
3:		
exp:	80	
growth_rate:	"slow"	
id:	4	
level:	4	
4:		
exp:	156	
growth_rate:	"slow"	
id:	5	
level:	5	
5:		
exp:	270	

JSON	Raw Data	Headers
Save Copy Collapse All Expand All Filter JSON		
0:		
growth_rate:	"slow"	
id:	1	
species_name:	"growlithe"	
1:		
growth_rate:	"slow"	
id:	2	
species_name:	"tentacool"	
2:		
growth_rate:	"slow"	
id:	3	
species_name:	"shellder"	
3:		
growth_rate:	"slow"	
id:	4	
species_name:	"exeggcute"	
4:		
growth_rate:	"slow"	
id:	5	
species_name:	"rhyhorn"	
5:		
growth_rate:	"slow"	
id:	6	
species_name:	"staryu"	
6:		
growth_rate:	"slow"	

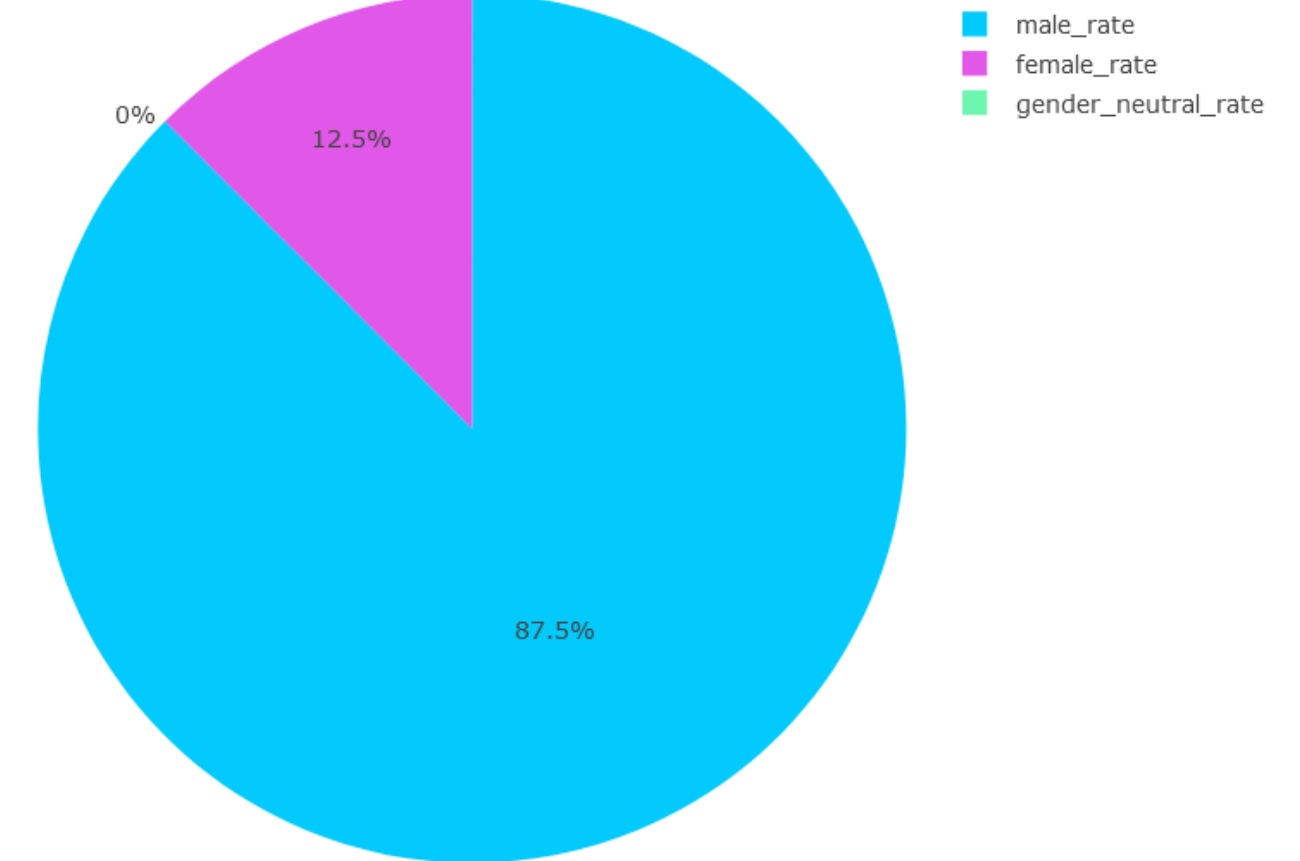
Radar Plot for Base Stats

Pie Chart for Gender Rates

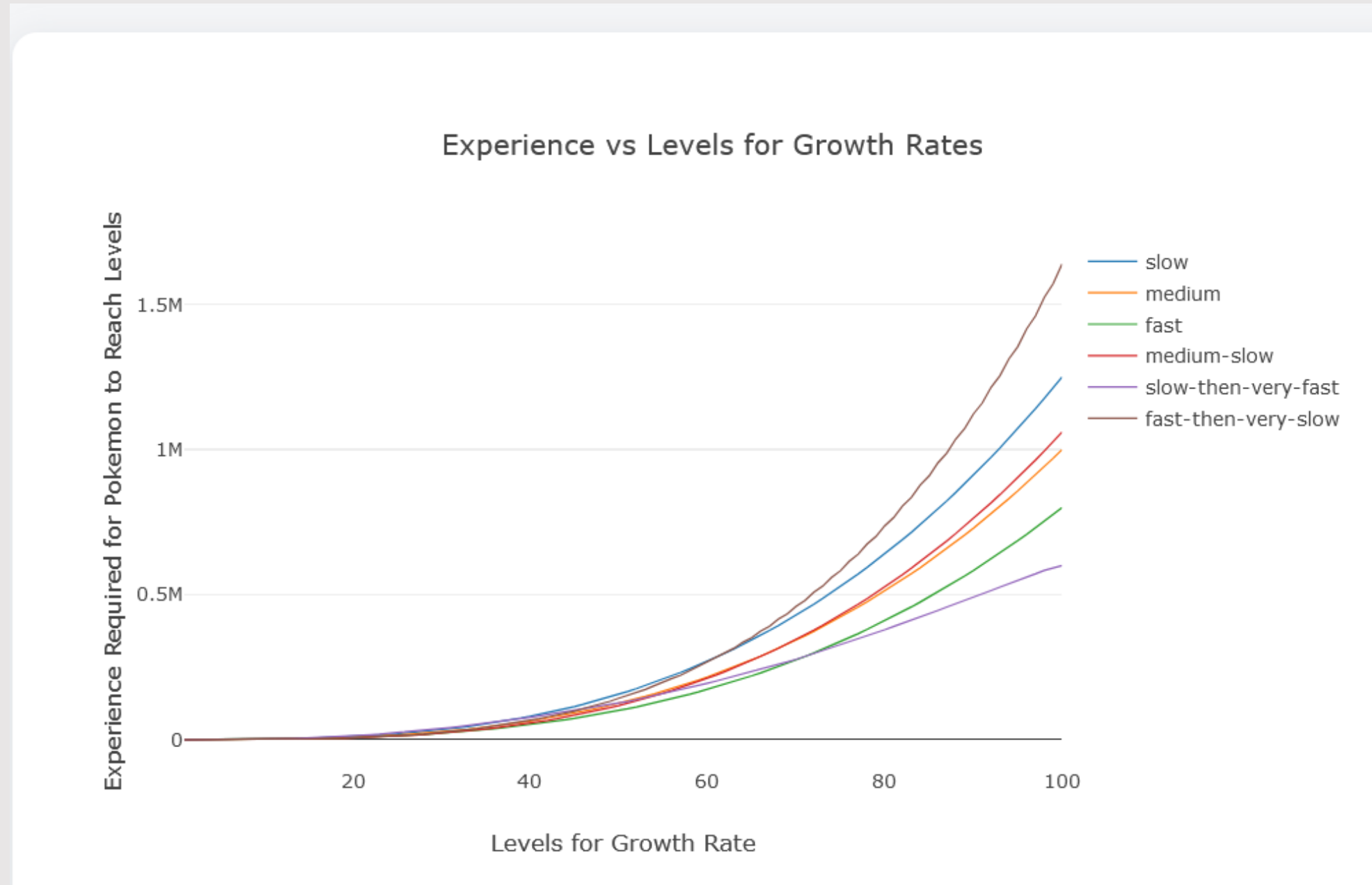
Pokemon Base Stats



Gender Rates



Growth Rates Comparison Line Graph



Tables extracted from local API

Poke Data

CHARACTER COUNT	AVG HEIGHT	AVG WEIGHT	AVG MALE RATE
1000	12	646	47.0







Poke

scroll to view all columns

Show

10

entries

Poke ID	Pic	Name	Height	Weight	male_rate
1		bulbasaur	7	69	87.5
2		ivysaur	10	130	87.5
3		venusaur	20	1000	87.5
4		charmander	6	85	87.5
5		charmeleon	11	190	87.5
6		charizard	17	905	87.5

Growth Rate

Levels

Show

10

entries

ID	Growth Rate	Level	Exper
1	slow	1	0
2	slow	2	10
3	slow	3	33
4	slow	4	80
5	slow	5	156
6	slow	6	270
7	slow	7	428
8	slow	8	640
9	slow	9	911
10	slow	10	1250

Showing 1 to 10 of 600 entries

Previous

1

2

Challenges

- No Location data in dataset as Fictional regions

Next Steps....

- To obtain a map for Pokemon locations generation
- Using more endpoints to incorporate moves and types