

Documentation

**2022 FBLA NLC
Coding and Programming**

Updated June 2022

**Thomas Lu
Dheeraj Vislawath
Vishwa Muruguppan**

Documentation

Table of Contents

	Page #
1. Source Documentation	3
1.1 Sources	
1.2 Image Documentation	
1.3 Data Documentation	
1.4 Licenses	
2. System Requirements	5
2.1 Windows	
2.2 Macintosh	
3. Requirements and Accessibility	6
3.1 Application Required Programs	
3.2 Packages and Modules	
3.3 Accessing Packages	
3.4 Accessing Source Code	
4. Program Architecture	18
4.1 Program Structure	
4.2 PyQt5	
4.3 Module Purposes	
5. Version History	20
5.1 Previous Versions	
5.2 Current Version	
5.3 Future Versions	

1. Source Documentation

1.1 Sources

Our application makes use of various files, libraries and images.

- Text files are generated from running the code
- Images were taken with due credit, for more information see 1.2
- Libraries were imported from the PyCharm IDE, for more information see 1.4

1.2 Image Documentation

Our application makes use of attraction images, application logos and icons.

- Bookmark icons
 - <https://www.awicons.com/stock-icons/symbol-black/bookmark/>
 - Available to the public for free use
 - All rights belong to original author
- Star ratings
 - <https://thenounproject.com/icon/star-rating-766721/>
 - Available to the public for free use
 - All rights belong to original author
- Search icon
 - https://www.iconfinder.com/icons/172546/search_icon
 - Available to the public for free use
 - All rights belong to original author
- Title image
 - <https://www.istockphoto.com/photos/chicago-night>
 - Available to the public for free use
 - All rights belong to original author
- Logo image
 - Credits to [Vislawath Corporation](#) for image use
 - Available to our project for free use
 - All rights belong to original author
- Sources for attraction images are available in code files, titled **sources.txt**
 - Attraction sources are also available for viewing in the application
 - All attraction images were available to the public for free use
 - Any image rights belong to original author

1.3 Data Documentation

Information on tourist attractions was gathered for use in our application.

- Data and information of tourist attractions acquired from Google Maps

- Available to public for free use, with all content publicly displayed with proper attribution
- All data rights belong to Google

1.4 Licenses

Our application makes use of various programs, libraries and modules. We used such resources under specific rules and licenses, of which are outlined below.

- **Python:** All Python releases are Open Source, and is also GPL-compatible, which guarantees anyone the freedom to run, study, share and modify.
- **PostgreSQL:** PostgreSQL is released under the PostgreSQL License, a liberal Open Source license. Permission to use, copy, modify, and distribute its software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

Portions Copyright © 1996-2022, The PostgreSQL Global Development Group
Portions Copyright © 1994, The Regents of the University of California

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

- **pgAdmin:** pgAdmin 4 is released under the above PostgreSQL licence.
- **Pycharm:** PyCharm Community Edition is free, open-source and can be used without any license.
- **PyQt5:** released under the GPL v3 license and under a commercial license that allows for the development of proprietary applications.

2. System Requirements

2.1 Windows

- Operating system must be **Windows 8** or above
- Device must be able to install all of the necessary programs and libraries.
- Device must have at least **1 GB of RAM**
- Device must have at least **500 MB of storage space**

2.2 Macintosh

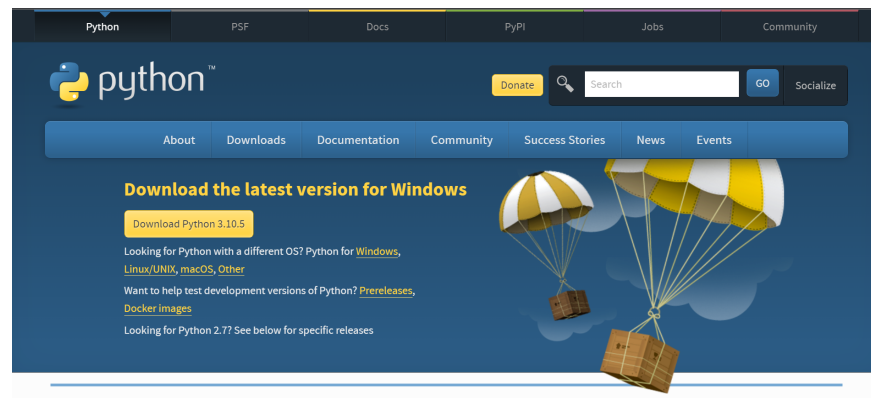
- Operating system must be **macOS 10.14** or above
- Device must be able to install all of the necessary programs and libraries.
- Device must have at least **1 GB of RAM**
- Device must have at least **500 MB of storage space**

3. Requirements and Accessibility

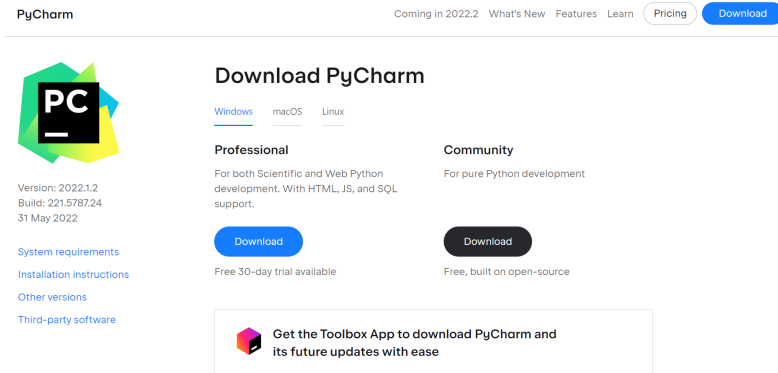
3.1 Application Required Programs

In our application, various programs and libraries are utilized. For the code to properly run, all of the following must be properly installed and functioning.

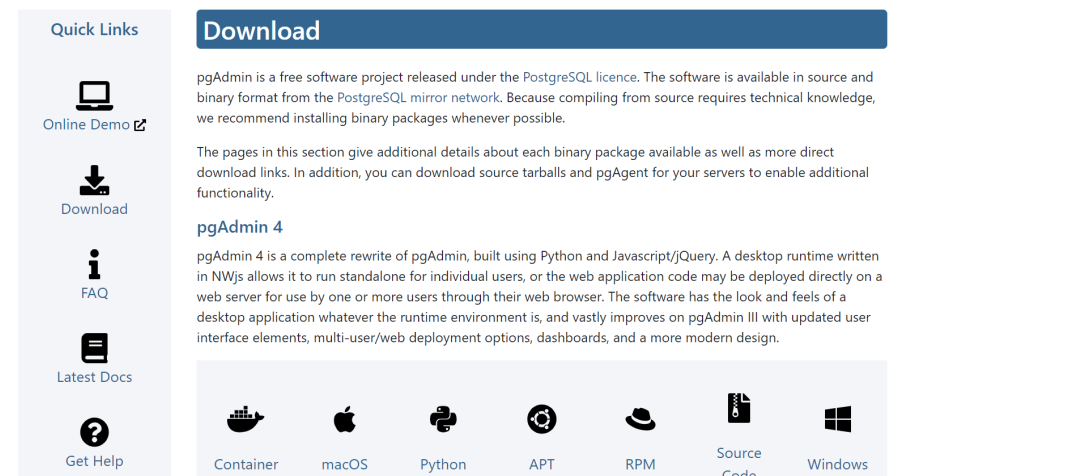
- Python, v3.8
 - Python is the language in which our code runs. Its usage follows a high-level, interpreted, and general-purpose programming philosophy, and emphasis on code readability with significant use of indentation.
 - To install python:
 - Head to the official homepage for the Python language:
<https://www.python.org/downloads/>



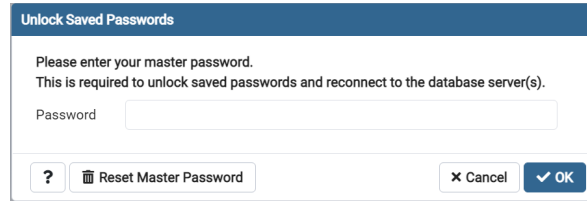
- Download the python installer, and follow its instructions until completion.
- Pycharm, v2022.1
 - For this project, PyCharm was used for the development of the application. To find the system requirements and installation process:
<https://www.jetbrains.com/help/pycharm/installation-guide.html>
 - To install Pycharm:
 - Head to the official homepage for Pycharm.



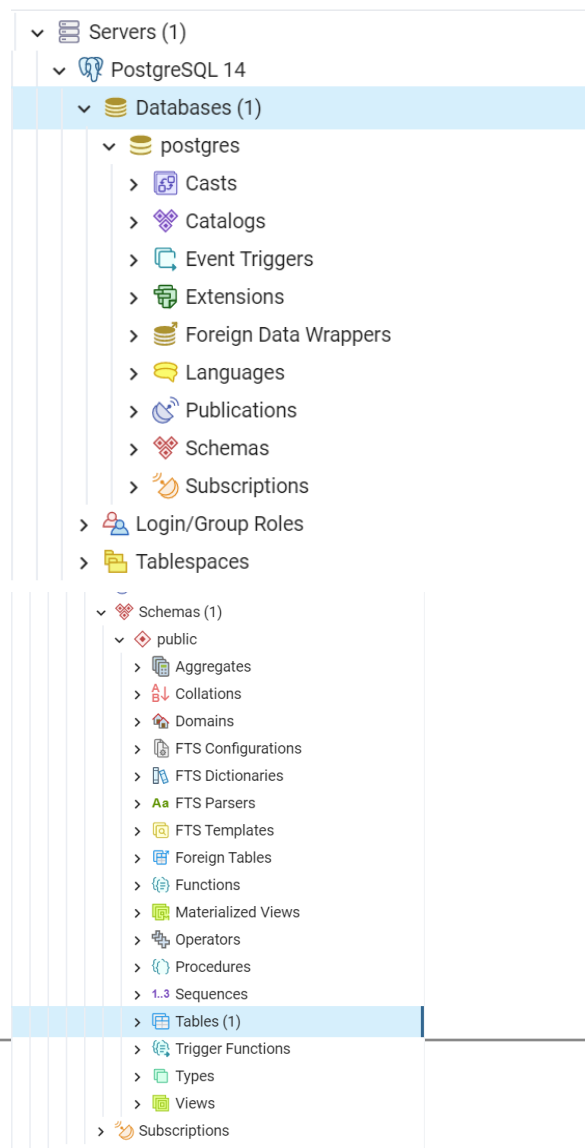
- Download the Pycharm installer and follow its instructions until completion.
- psychopg2, v2.8.6
 - This library is used to connect PostgreSQL to the Python project. Over the course of the project, versions 2.8.6 and 2.9.1 were used. To find more about the library and its installation: <https://pypi.org/project/psycopg2/>
 - To install psycopg2:
 - Run the following command `pip install psycopg2` in your command prompt.
 - Follow all instructions until completion.
- pgAdmin, v14.2
 - This library is used to hold all attraction data, and to dynamically store filtered attractions within this project. For this project only version 14.2 was tested and implemented. To find information about the library and the installation process: <https://www.postgresql.org/download/>
 - To install pgAdmin:
 - Head to the pgAdmin download website.



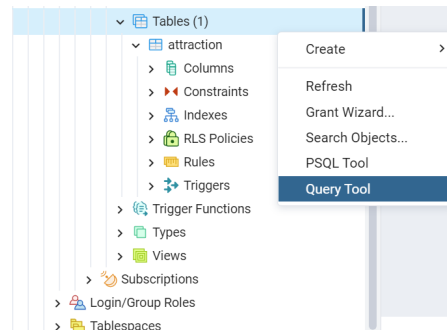
- Find the version that matches your device, and follow instructions until completion.
- Our application makes use of a data table in PostgreSQL, which can be interacted with using pgAdmin. A copy of the data our group used is required in an identical table for the application to run on your own device.
- To set up the same data:
 - Open pgAdmin. Ensure PostgreSQL is installed and running.
 - You will be prompted to create a master password. Create a new password and store it somewhere safe.

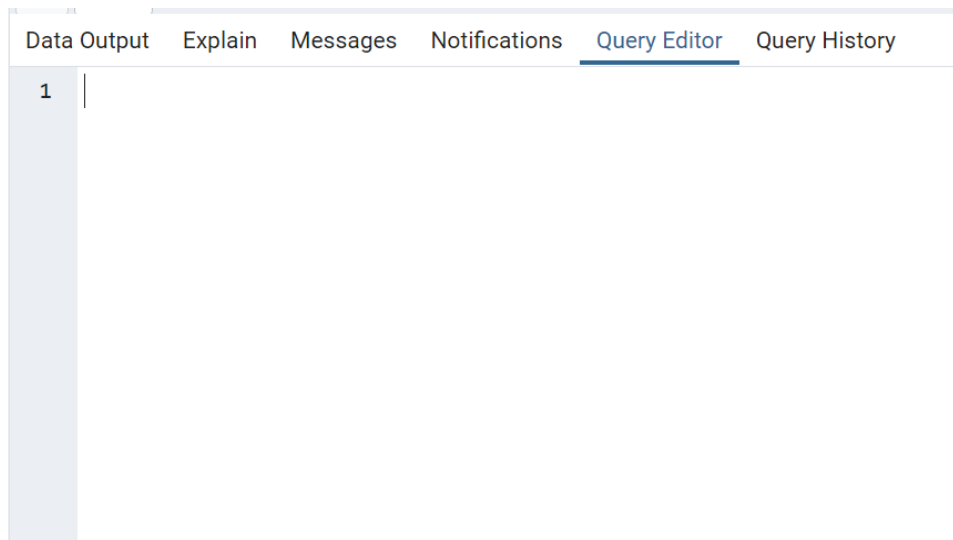
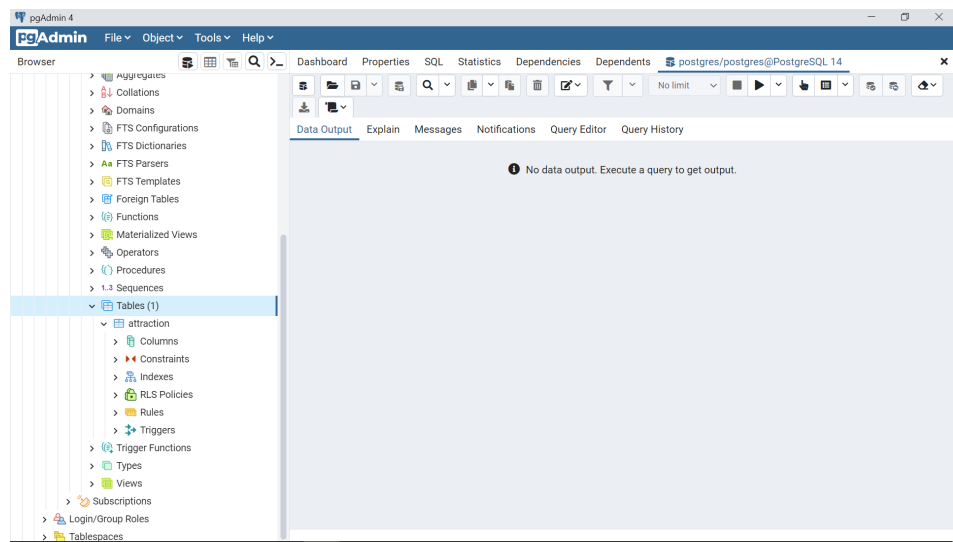


- Next, navigate on the menu on the left from Servers › Postgres › Databases › postgres › Schemas › Tables



- From there, navigate from Tables > Query Tool.





On the bar at the middle top of the new area, navigate to the Query Editor.

- In this query editor text entry, paste the following code to create a table with the correct columns and their attributes. The raw code can also be found [here](#). Run by clicking the execute button.

```
CREATE TABLE ATTRACTION
(
  ID SERIAL PRIMARY KEY,
  NAME VARCHAR(200) NOT NULL,
  DESCRIPTION VARCHAR(2000) NOT NULL,
  STATE VARCHAR(50) NOT NULL,
  CITY VARCHAR(50) NOT NULL,
  TYPE VARCHAR(50) NOT NULL,
```

```

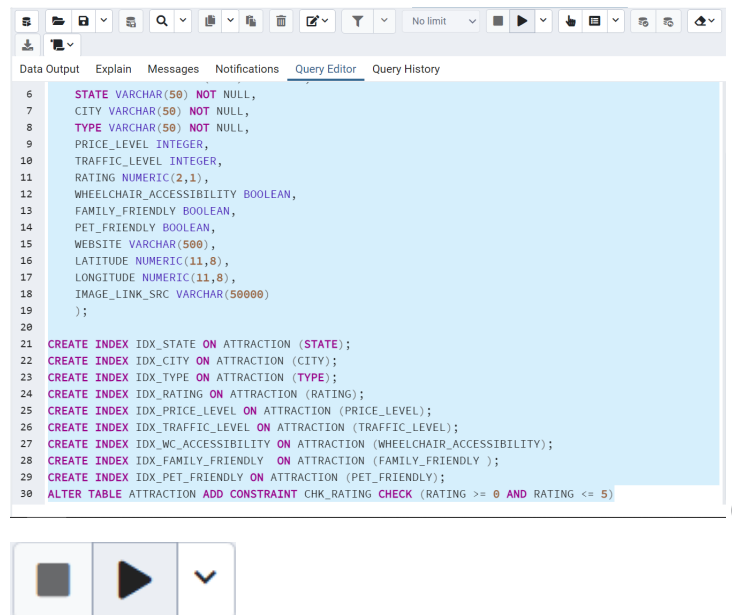
PRICE_LEVEL INTEGER,
TRAFFIC_LEVEL INTEGER,
RATING NUMERIC(2,1),
WHEELCHAIR_ACCESSIBILITY BOOLEAN,
FAMILY_FRIENDLY BOOLEAN,
PET_FRIENDLY BOOLEAN,
WEBSITE VARCHAR(500),
LATITUDE NUMERIC(11,8),
    LONGITUDE NUMERIC(11,8),
    IMAGE_LINK_SRC VARCHAR(50000)
);

```

```

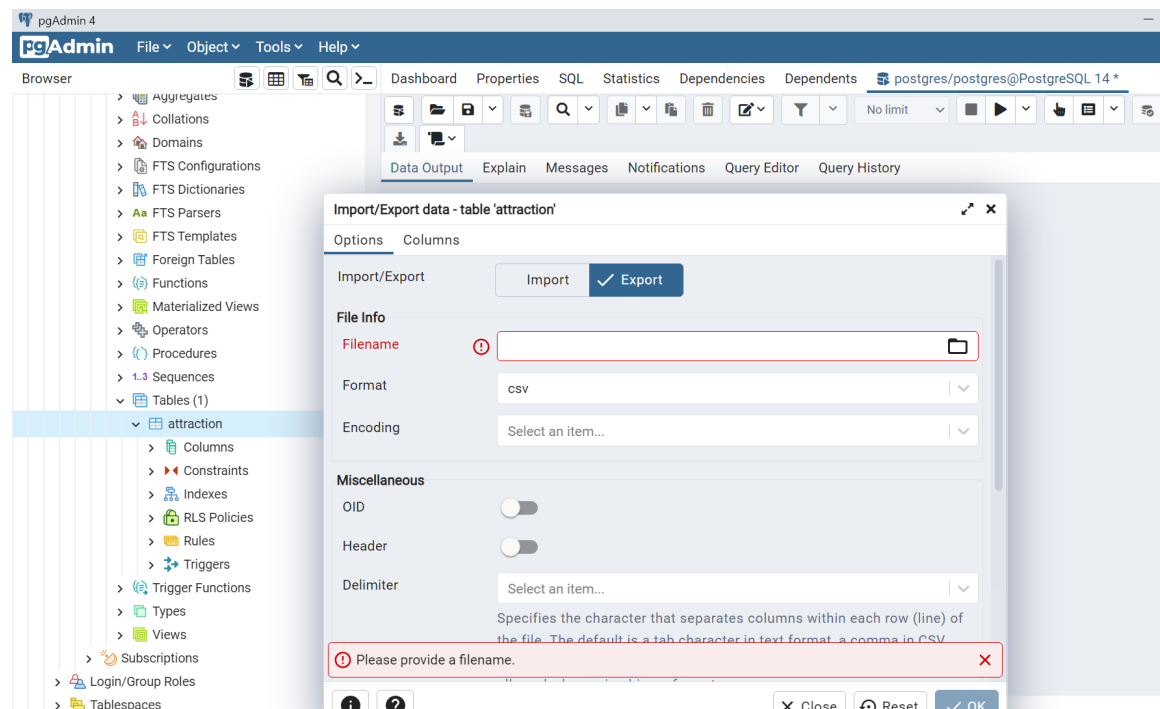
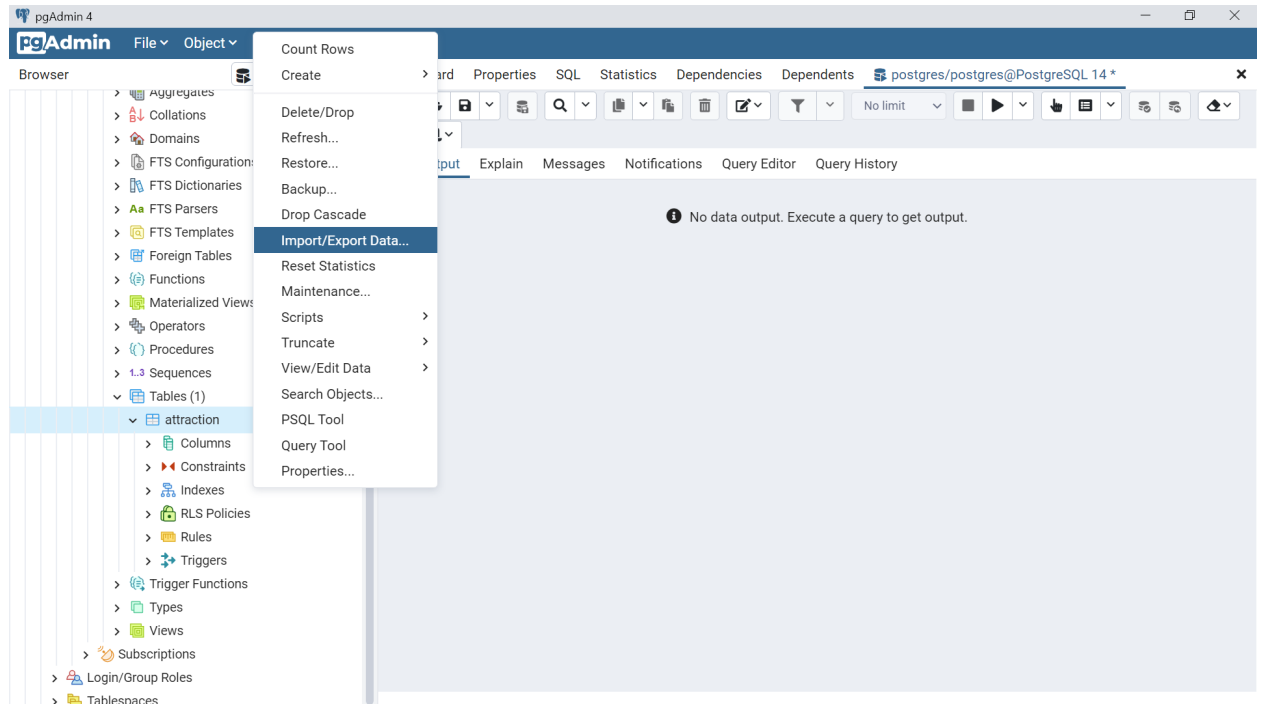
CREATE INDEX IDX_STATE ON ATTRACTION (STATE);
CREATE INDEX IDX_CITY ON ATTRACTION (CITY);
CREATE INDEX IDX_TYPE ON ATTRACTION (TYPE);
CREATE INDEX IDX_RATING ON ATTRACTION (RATING);
CREATE INDEX IDX_PRICE_LEVEL ON ATTRACTION (PRICE_LEVEL);
CREATE INDEX IDX_TRAFFIC_LEVEL ON ATTRACTION (TRAFFIC_LEVEL);
CREATE INDEX IDX_WC_ACCESSIBILITY ON ATTRACTION
(WHEELCHAIR_ACCESSIBILITY);
CREATE INDEX IDX_FAMILY_FRIENDLY ON ATTRACTION (FAMILY_FRIENDLY);
CREATE INDEX IDX_PET_FRIENDLY ON ATTRACTION (PET_FRIENDLY);
ALTER TABLE ATTRACTION ADD CONSTRAINT CHK_RATING CHECK RATING >=
0 AND RATING <= 5)

```



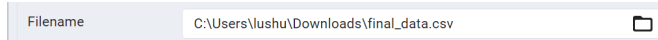
Now that you have created a table with the correct columns, download the data sheet. The raw csv can be found [here](#).

- Next, import the data sheet.



- Ensure that:

- File name is correct.



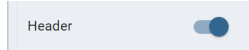
Filename C:\Users\lusu\Downloads\final_data.csv

- Import is selected rather than export.



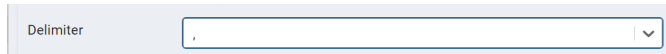
Import/Export

- Header option is selected.



Header ☒

- Delimiter is set to a comma.



Delimiter ,

- Columns are the same and in the correct order.



Columns to import

id	name	description	state	city	type
price_level	traffic_level	rating	family_friendly		
wheelchair_accessibility	pet_friendly	website			
latitude	longitude	image_link-src			

- The database is now functioning. All extraneous programs should be functioning at this point.

3.2 Packages and Modules

To run our application, various packages and modules were used.

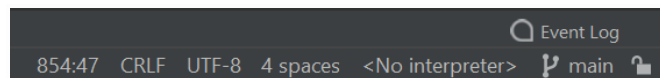
- The **PyQt5** module is a comprehensive set of Python bindings, implementing high-level APIs for accessing many aspects of modern desktop and mobile systems, covering more than 35 extension modules.
- The **ast** module helps Python applications to process trees of the Python abstract syntax grammar. The abstract syntax itself might change with each Python release; this module helps to find out programmatically what the current grammar looks like.
- The **os** module provides a portable way of using operating system dependent functionality.
- The **webbrowser** module provides a high-level interface to allow displaying web-based documents to users.
- The **io** module provides Python's main facilities for dealing with various types of Input/Output.

- The **folium** module builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library. Manipulate your data in Python, then visualize it in a Leaflet map via folium.
- The **ipregistry** is the official Python client library for the ipregistry IP geolocation and threat data API, allowing you to lookup your own IP address or specified ones.
- The **time** module provides various time-related functions.
- The **operator** module exports a set of efficient functions corresponding to the intrinsic operators of Python.
- The **geopy** module is a Python client for several popular geocoding web services.

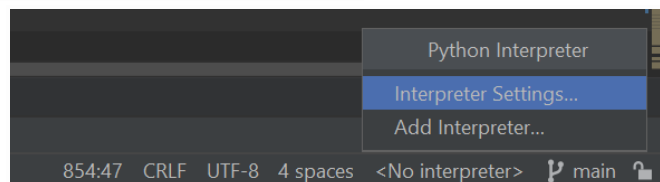
3.3 Accessing Packages

Our application requires various packages and modules to operate. You will need to import the aforementioned packages into your Pycharm IDE.

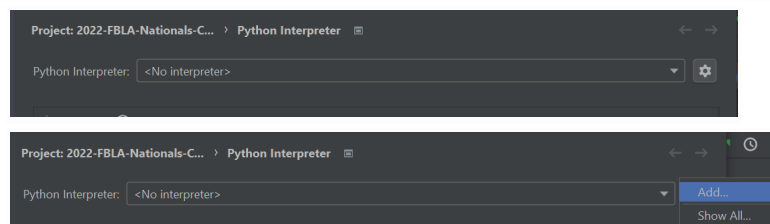
- To import packages:
 - You will first need to select an interpreter. In the bottom right, click on the < No Interpreter > tag.



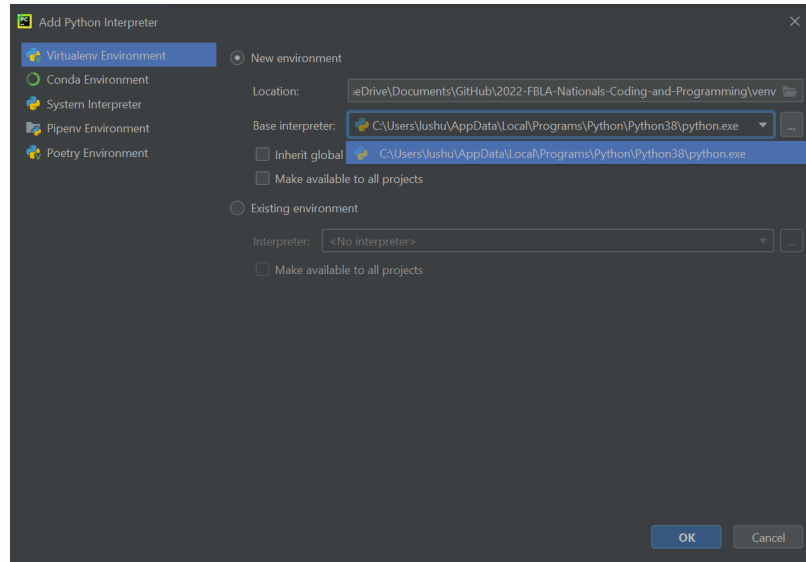
- Select Interpreter Settings...



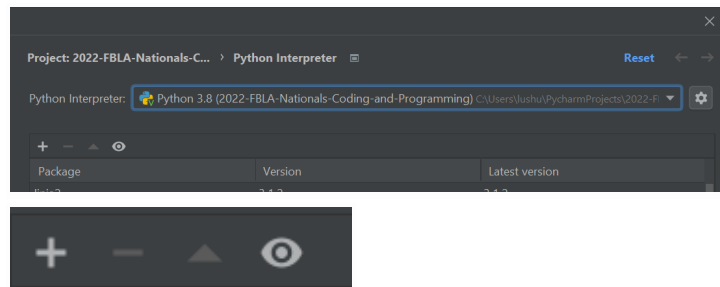
- In the top right, select the settings icon and then select Add...



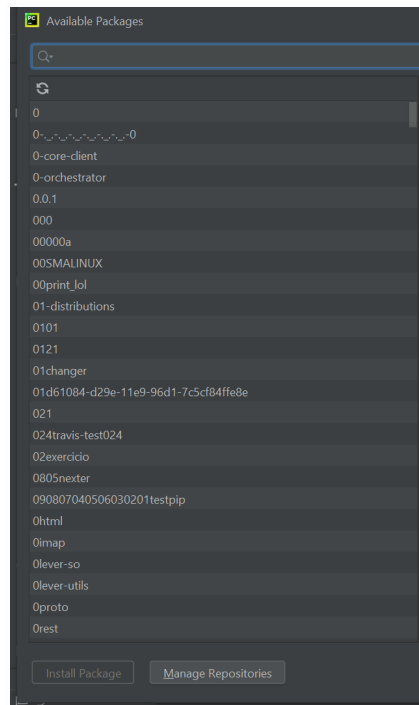
- Select the base interpreter that matches your device and your version of Python.



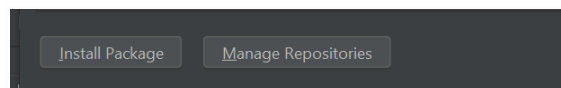
- Now that you have selected an interpreter, install the packages listed above.
 - To do so, from the packages menu, select the + button.



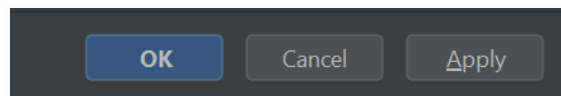
- From the menu on the left, you can search for the modules above.



- Select Install Package in the bottom left for each of the required modules listed above.



- In the bottom right, select OK when you have installed all the required modules.

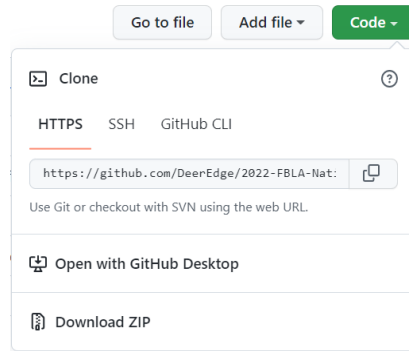


- With that, all of the modules needed to run the application should be installed.

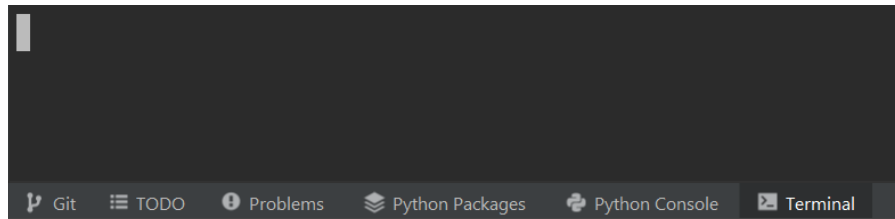
3.4 Accessing Source Code

- All necessary files for the application are stored on Github, a hosting site for project management and version control. The source code includes everything related to running our application and algorithms to find tourist attractions. Commentary is provided to assist the process of running the source code.
- To access the source code and open it on your own device, several approaches can be taken.
 - a. Clone the code from github onto your local Pycharm IDE. This can be done:

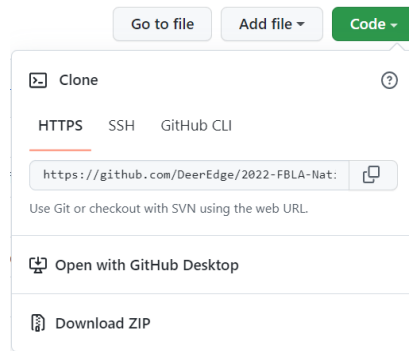
- i. Through the github.com interface. The link to our github repository can be found [here](#).

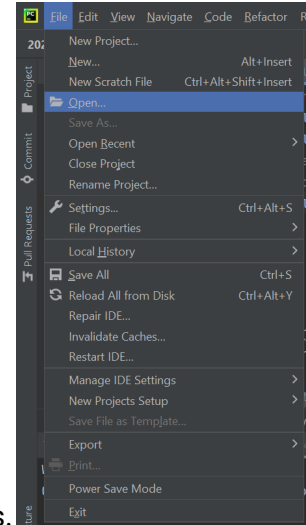


- ii. Directly through the Pycharm terminal. In the terminal, input the command `git clone`.



- b. Download the program files and open them on your local IDE.
 - i. Download the ZIP file with all program files.





- ii. In your Pycharm IDE, open those downloaded files.
- With all that, all of the source code should be available on your own device.

4. Program Architecture

4.1 Program Structure

- Our application makes use of filtering as well as sorting algorithms to suggest tourist attractions to the user in a visual application. Additional intelligent features are also supported.
 - The application begins on the title page, in which the basic attributes of state, city and type of attraction are offered, as well as the option of searching for an attraction.
 - When the user hits search, all of their chosen attributes are stored in program memory. From there, the code will then run through the database to find all attractions that match the given user's given attributes. These attractions are then displayed along with their attributes. Further filtering by the user is always allowed.
 - Our application also allows the user to sort the suggested attractions. When such an option is selected, the stored attractions are run through the algorithm that corresponds to the selected sorter, and redisplayed to the user. A special sorter of distance to the user is offered, with support for user input of coordinates, an option to automatically fill in location, and a preview of their location on maps.
 - Alongside filtering and sorting, other intelligent features are also offered to the user. These include a preview map of the attraction's location as well as an expanded map option; the ability to navigate to each attraction's homepage; and a help menu that includes FAQ, documentation, and the ability to create a user report.
 - Another unique feature of our application is the ability to bookmark certain tourist attractions. These are stored in another tab window for bookmarks. This tab is similar to the main window in which attributes of each bookmarked attraction are displayed. A search bar and a button to clear all bookmarks is also offered to the user.
 - Additionally, the application includes a third tab for listing all sources, licenses are references used in the creation of this application.

4.2 PyQt5

- This project makes use of the PyQt library, which is a library that lets you use the Qt GUI framework from Python. Our application makes use of different database connections and algorithms to suggest tourist attractions, which is then displayed in a user-friendly manner using the PyQt5 library. PyQt5 allows us to build a graphic user interface for our code and create a visual application.

4.3 Module purposes

- Different modules and libraries are imported with the goal of enhancing our application and improving the user's experience. Each respective library has a specialized purpose.
 - Assisting with basic code operations and functions (**os, io, operator** modules)
 - Creation of application and its architecture (**pyqt5, ast, folium** modules)
 - Improving user experience and elevating program functions (**geopy, webbrowser, ipregistry, time** modules)

5. Version History

5.1 Previous Version(s)

- 1.0
 - Application opened by tkinter window
 - Basic ability to sort attractions based on simple algorithm
 - Added difference factor algorithm to suggest tourist attractions
 - 50 pieces of data stored locally, in program files
 - Data includes various attributes to sort by
 - Limited window design and styling
 - Limited display ability
- 2.0
 - Modules / libraries updated for greater workability and potential in application
 - Tkinter library replaced by PyQt5
 - Other essential libraries imported to support more application features
 - Database created on local machine to store data
 - All attributes fully created in database
 - Attribute properties in database managed
 - Limited data table
 - Ability to filter and sort attractions fully implemented
 - Limited support for further features
- 3.0
 - Multiple tabs in the application added for further program usability
 - Data table for all 50 states finished
 - Added support for all 50 states and their largest cities
 - Help features added
 - Ability to create a user report
 - Output reports that log user activity generated
 - Support for other features added, but not fully implemented
 - Advanced functions implemented
 - Google maps support added
 - Ability to view a preview of attraction location
 - Ability to expand window of preview map
 - Support for sorting by distance to user added
 - Latitude and longitude input created
 - Ability to automatically fill in user location details
 - Added expandable map or user location
 - Further usability implemented

5.2 Current Version

- 4.0
 - A working title page added to improve user experience
 - A tab designated to citing sources for all attraction information fully implemented
 - Application design polished, further styling of application window
 - All attraction images downloaded, cropped and properly displayed in attraction window
 - Bookmark functionality added to bookmark attractions
 - Bookmarks display all attributes
 - Ability to search, clear bookmark
 - Design and styling of application finalized to improve user experience
 - Help menu polished with support for a FAQ menu

5.3 Future Versions

- Increase in data table for more potential attractions to be suggested to user
- Ability to audially play application details for improved user accessibility
- Ability to store data externally so any user can access data table
- Increase performance
- Improve aesthetics of application

Readme files:

- Prereqs
- Hidden tricks
- Need to know
- Description
-
- Manual on how to do everything
- General usage tips

QnA questions

Text box with questions, when clicked display answer, recede when clicked again

Questions:

1. How do I navigate back to the attraction finder page?
 - a. Go [here](#).
2. What does each attribute mean?
 - a. Location
 - b. State
 - c. City
 - d. Type
 - e. Wheelchair
 - f. Family
 - g. Pet
 - h. Sort: nearest, price, rating, traffic
3. Does the attraction finder cover all states?
 - a. Yes, this application suggests attractions from all 50 states, covering each state's largest cities.
4. Can I download my suggested attractions for further use?
 - a. Yes, when running our application, we generate an output report for your use.
5. How can I suggest an edit to an attraction?
 - a. [\[\] > \[\]](#) gives you the option to tell us of an error by generating a user report. The user report will automatically tell you what searches you have done.
6. I need some more information on the program. (sure: documentation)
 - a. Sure, we have created a file for documentation of our program. Visit [\[\] > \[\]](#)
7. How do I use the 'my location' feature?
 - a. Our application is able to take your location and use it when looking for nearest attractions. By selecting, you can sort on the right. Each attraction also tells distance to you
8. Why do some features not work?
 - a. Most likely, you may have some wifi problems that prevent full use of internet features, such as finding your location and redirecting you to each attraction's homepage.
 - b. You can also generate a user report in the event you encounter an error.
 - c. We also have created a documentation file that covers required libraries, system requirements, and more.
9. How do I find attractions close to where I am?
 - a. You can do this by navigating to the location details window and selecting an option in the dropout menu under the area labelled "Desired Distance From You"

10. How do I filter for whether attractions are pet friendly?
 - a. You can do this by navigating to the "Filter By:" area and checking the box on the pet friendly option.
11. How do I find out how expensive it is to visit an attraction?
 - a. This may be done by navigating to the center area where all the attractions are displayed and looking at the price rating. It should be \$, \$\$, \$\$\$, \$\$\$\$.~~\$\$\$~~.

Licences:

License

PyQt5 is released under the GPL v3 license and under a commercial license that allows for the development of proprietary applications.