1. Open **app.py** and change “sqlite:///MBase.db” to the database you are using. (e.g. sqlite:///musicdatabase.db”.

Graphical user interface, website

Description automatically generated

Figure 1

1. Open **db\_setup.py** and change “sqlite:///MBase.db” to the database you are using
2. Open **settings.py** and **models.py**
3. The settings and model classes are the Classes in Python being mapped to the tables in your database. If your database is using different tables, change these classes for your needs. (Model => Music etc.) Remember to change \_\_tablename\_\_ to the table name of your database table.

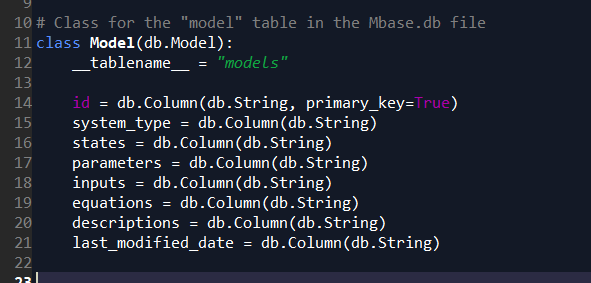


Figure 2

1. Within the **model** and **setting** classes, the columns of the database tables are mapped to db.Column objects (Figure 2). Ensure that the variable names are the same as the column names of your database table.
2. Open **db\_setup.py** again and in the **init\_db()** function, change **import models** and **import settings** to the names of your new table classes (e.g. import models => import artists).

Graphical user interface

Description automatically generated

Figure 3

1. Open **tables.py** and go to class Results and Settings. These are the html tables that are shown as the search results and settings respectively. Map the columns from **models.py** and **settings.py** to the Col objects.

Text

Description automatically generated

Figure 4

1. The **classes** attribute can be changed to use different html classes for css formatting purposes.
2. The Col objects can also be subclasses to format the contents within the Col (e.g. FormatCol, ParameterBoundsCol)

Text

Description automatically generated

Figure 5

1. There are also other Col such as LinkCol which allows us to put a link in the Col. (Refer to the flask-table documentation at: <https://flask-table.readthedocs.io/en/stable/>)
2. Open **forms.py.** This is the class handling the search form for the website. The **choices** dictionary provides the option to search the respective columns of the database.
3. Open **main.py.** The app.route line defines the url and the supported methods. The **index()** function defined is then executed when the url is accessed.

Graphical user interface

Description automatically generated with medium confidence

Figure 6

1. When the search is executed, the flask app will access the /results url, which will then call the **search\_results()** function. Change the “System Type”, “Last Modified Date” and “Description” to the updated choices options from **forms.py**.

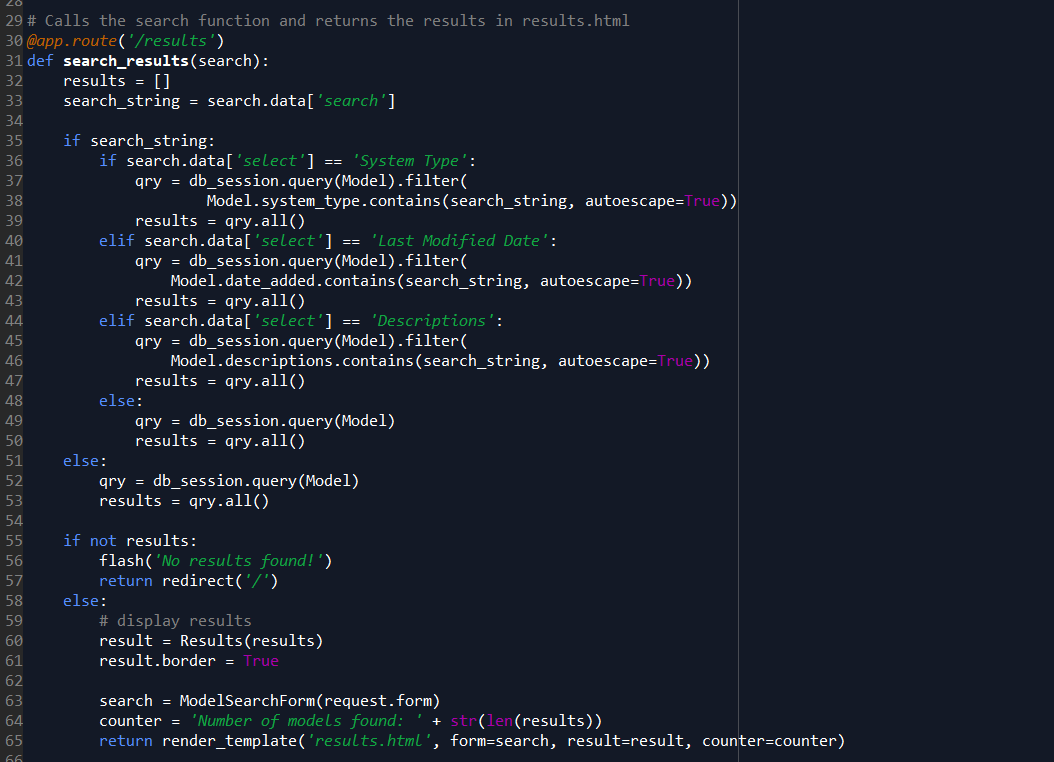


Figure 7

1. Change the iterations of **Model** to the new database class that was defined in step 4.
2. Navigate to the templates folder and open **index.html.** Navigate to BMSS Search.



Figure 8

1. Note that the images used for Flask must be static and the links are different from how it would be typically referenced.
2. Note that Flask uses the Jinja 2 templating language.

**Appendix**

The base of the website was done following the tutorial on <https://www.blog.pythonlibrary.org/2017/12/15/flask-101-filtering-searches-and-deleting-data/>