Eric Boilard

Southern New Hampshire University

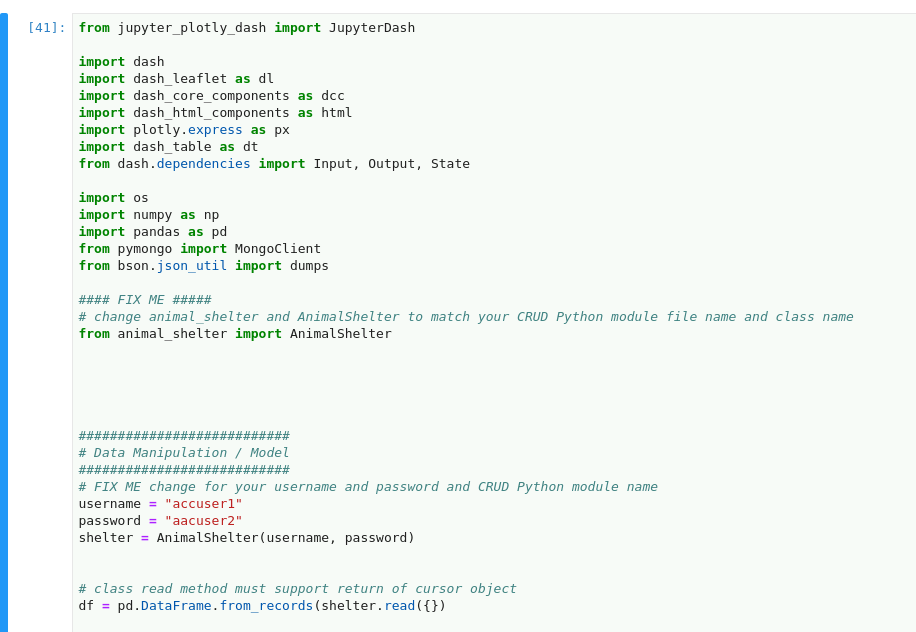
CS-340 Final Project Screenshots

8/20/21

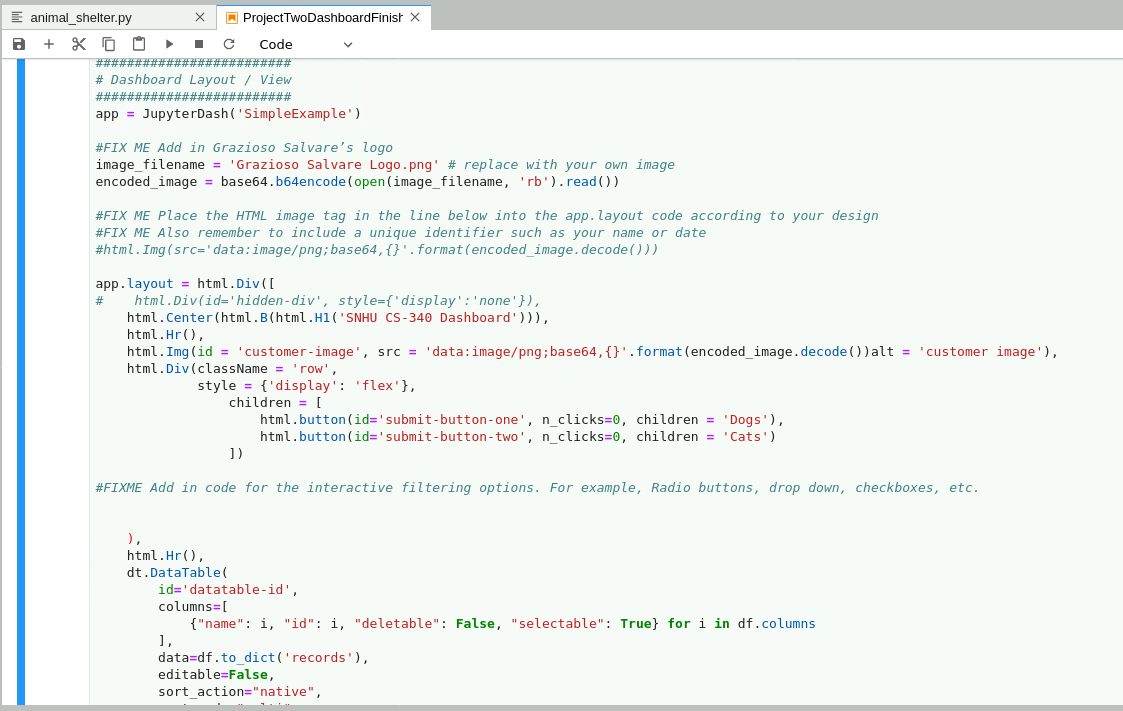
1. Before developing the Python code for the dashboard, be sure to review the Dashboard Specifications Document provided by the UI/UX developer at Global Rain. This document is located in the Supporting Materials section and will provide you with examples of the different dashboard widgets you will create:
   * Interactive options to filter the Austin Animal Center Outcomes data set
   * A data table which dynamically responds to the filtering options
   * A geolocation chart and a second chart of your choice that dynamically respond to the filtering options

* In addition to the widgets, you have been asked to include the **Grazioso Salvare logo and a unique identifier** containing your name somewhere on the dashboard. A high-resolution copy of the logo is included in the Supporting Materials section.

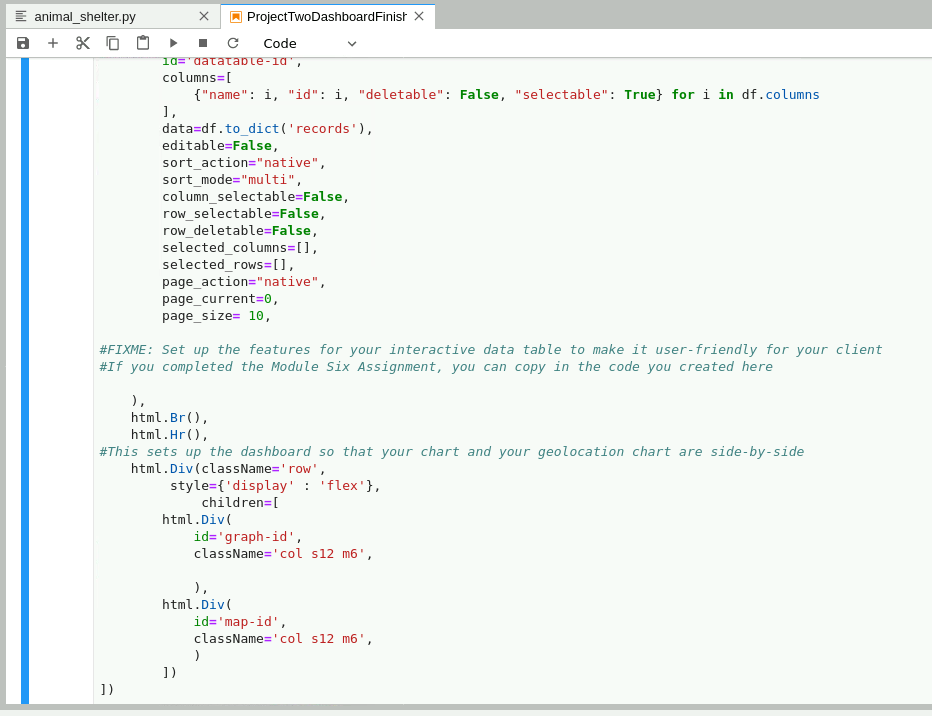
1. Next, you will begin developing the Python code for your dashboard. Starter code is contained in the ProjectTwoDashboard.ipynb file, linked in the What to Submit section. Start by creating a **data table** on the dashboard which shows an *unfiltered* view of the Austin Animal Center Outcomes data set. To populate the data onto your table, **utilize your previous CRUD Python module** from Project One to run a “retrieve all” query and bring in the data from MongoDB.  
     
   Tip: Be sure to consider your client when creating the interactive data table. Consider optional features that will make the table easier to use, such as limiting the number of rows displayed, enabling pagination (advanced), enabling sorting, and so on.  
     
   Note: If you completed the Module Six Milestone, you have already completed this step. Copy your code for the data table into the ProjectTwoDashboard.ipynb file.
2. Next, you will make sure that the dashboard filter options can properly retrieve data from the database. Start by **developing database queries that match the required filter functionality**. Refer to the Rescue Type and Preferred Dog Breeds Table, located in the Dashboard Specifications Document, to help you construct these queries.  
     
   Note: Be sure to **utilize your previous CRUD Python module** (a PY file) from Project One to develop these database queries. You will need to hard code in the username/password for the “aacuser” account as part of the CRUD Python module class instantiation.



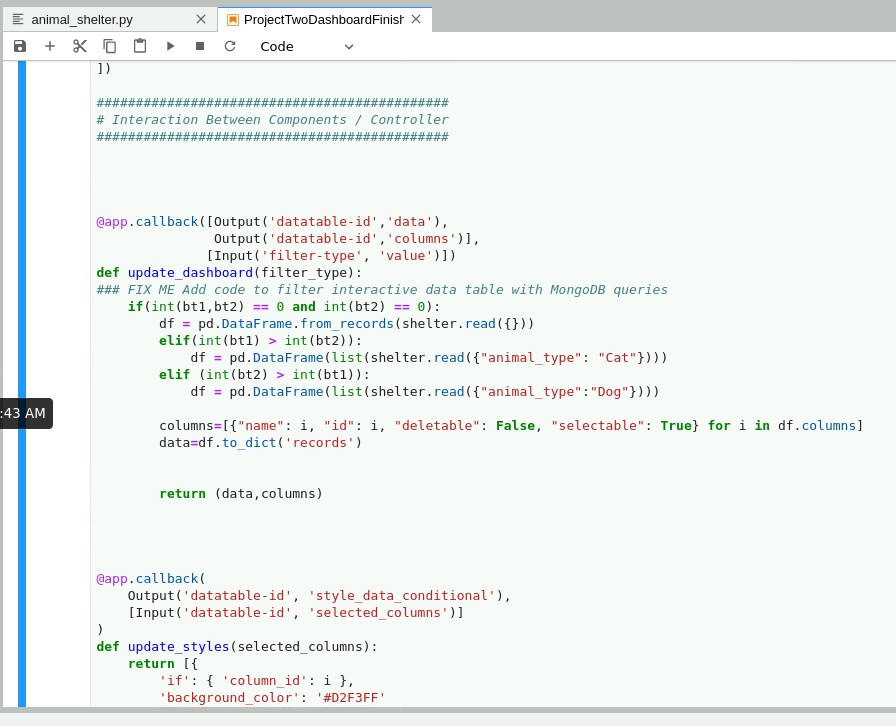
1. You must develop the controller pieces to **create interactive options that allow for the selection of data based on your filtering functions** (such as radio items or drop-downs). Develop these pieces in your IPYNB file, and be sure to import and use your CRUD Python module queries from Step 3. These interactive options will enable the **control of other dashboard widgets**, such as the data table and charts.

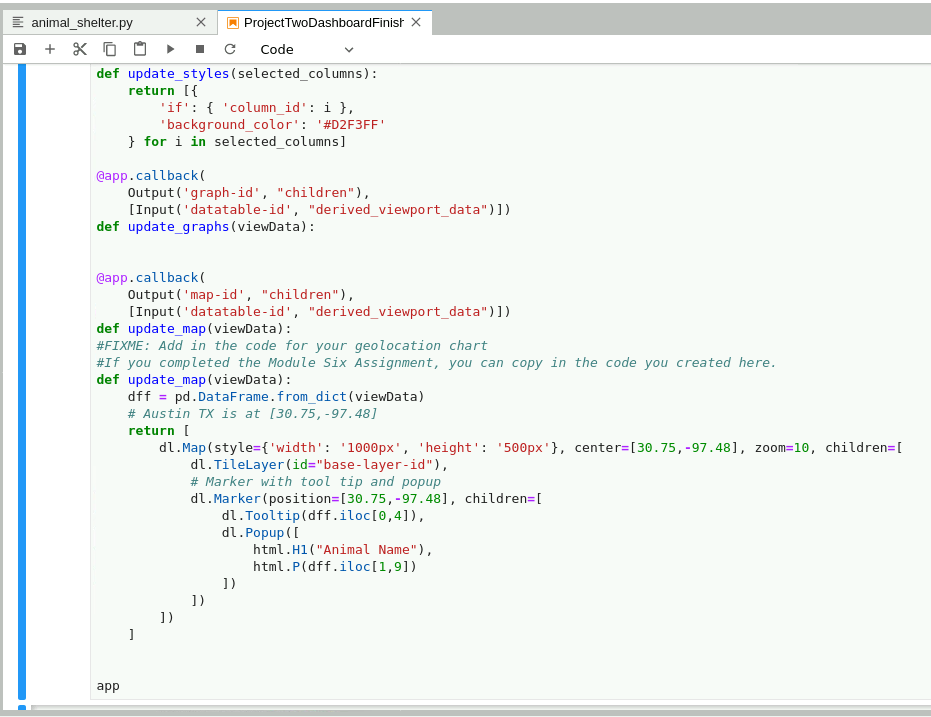
  
Tip: You may choose any interactive option that you wish, such as radio items, a drop-down menu, and so on, as long as the client is able to intuitively use the interactive option to filter the data. Refer to the Dash Core Components reading from the module resources to help you set up these options.

1. Next, you must modify or create the dashboard widgets that receive input from the interactive options and present those dynamic updates to the client. Be sure to modify or create these widgets in your IPYNB file. Specifically, you must do the following:
   * Modify the data table you created in Step 2 so that it is **an interactive data table that responds to input** from the interactive options.
   * **Create charts that display data in response to updates** from the data table. As outlined in the Dashboard Specifications Document, you are required to create, at minimum, a geolocation chart and a second chart of your choice.

  
Note: If you completed the Module Six Milestone, you have already begun work on this step by creating the geolocation chart. Copy your code into the ProjectTwoDashboard.ipynb file. You will need to make sure that this chart receives updates from the interactive options.

1. Finally, after developing all of your code, you must test and deploy the dashboard to make sure that all of your components work. To complete this step, run your IPYNB file. You must either **take screenshots or create a screencast of your dashboard and widget functionality**. Each of your screenshots or your screencast should contain the **Grazioso Salvare logo** and your **unique identifier**. Your screenshots or screencast must show the following:
   * The starting state of your dashboard, which should include your widgets for the **interactive options to filter data (such as radio items or drop-downs), the interactive data table, and the charts**
   * Executions of your dashboard, showing the widgets after **each** of the following data filters has been applied (four screenshots total):
     + Water Rescue
     + Mountain or Wilderness Rescue
     + Disaster or Individual Tracking
     + Reset (returns all widgets to their original, unfiltered state)



  
You will include all of these screenshots, or your screencast, in your README file when describing the functionality of your project. These screenshots are **required** as they demonstrate proof of your dashboard’s functionality.