# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

*This project aims to create an interactive dashboard for the Austin Animal Center Outcomes data set. The dashboard allows users to filter and explore the data, view it in a data table, and visualize it using charts. It also includes a geolocation chart and a pie chart displaying dog breed distribution.*

## Motivation

*The motivation behind this project is to provide a user-friendly interface for exploring and analyzing the Austin Animal Center Outcomes data. By creating this dashboard, we make it easier for users to understand and interact with the data, helping them make informed decisions related to animal rescues.*

## Getting Started

*To get a local copy up and running, follow these simple example steps:*

1. *Upload the project repository to your local machine.*
2. *Upload the Grazioso Salvare Logo and CVS file into the same directory with ProjectTwoDashboard.ipynb file.*
3. *Run the ProjectTwoDashboard.ipynb file in a Jupyter Notebook environment.*

## Installation

*To run the Austin Animal Center Outcomes Dashboard, you need:*

1. *Jupyter Notebook: Access the lab first, then find Jupyter Notebook on desktop screen.*
2. *Project Files: Upload the ProjectTwoDashboard.ipynb file into Jupyter Notebook, then open it to work on.*
3. *Grazioso Salvare Logo: Download the logo from the assignment and upload to same directory in Jupyter Notebook.*
4. *Austin Animal Center Outcomes Spreadsheet: Download the CVS file from the assignment and upload to same directory in Jupyter Notebook.*
5. *Apporto Lab: Use Apporto Lab for a cloud-based Jupyter Notebook environment. Upload the project files to your Apporto Lab session.*

## Usage

*Here’s a concise code example that demonstrates the core functionality of the project:*

### Code Example

*# Import necessary modules*

*from jupyter\_dash import JupyterDash*

*import dash\_leaflet as dl*

*from dash import dcc, html, dash\_table*

*import base64*

*from animal\_shelter import AnimalShelter*

*import pandas as pd*

*# Create a JupyterDash app*

*app = JupyterDash(\_\_name\_\_)*

*# Add Grazioso Salvare's logo*

*encoded\_image = base64.b64encode(open('Grazioso Salvare Logo.png', 'rb').read())*

*app.layout = html.Div([*

*html.Img(src='data:image/png;base64,{}'.format(encoded\_image.decode())),*

*html.Center(html.B(html.H1('CS-340 Dashboard - Hong Luu'))),*

*dcc.RadioItems(*

*id='filter-type',*

*options=[*

*{'label': 'Water Rescue', 'value': 'water'},*

*{'label': 'Mountain Rescue', 'value': 'mountain'},*

*{'label': 'Disaster Rescue', 'value': 'disaster'},*

*{'label': 'Reset', 'value': 'reset'}*

*],*

*value='reset',*

*labelStyle={'display': 'inline-block'}*

*),*

*dash\_table.DataTable(*

*id='datatable-id',*

*columns=[{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns],*

*data=df.to\_dict('records'),*

*page\_size=10,*

*),*

*html.Div(className='row', style={'display': 'flex'}, children=[*

*html.Div(id='graph-id', className='col s12 m6'),*

*html.Div(id='map-id', className='col s12 m6'),*

*])*

*])*

*# Run the app*

*app.run\_server(debug=True, port=30825)*

### Tests

*To test the project, follow these steps:*

1. *Run the file.*
2. *Navigate to Dash app running on* [*http://127.0.0.1:30825/*](http://127.0.0.1:30825/) *and click on* [*http://127.0.0.1:30825/*](http://127.0.0.1:30825/)*.*
3. *You will see the dashboard interface with four radio buttons labeled ‘Water Rescue,’ ‘Mountain Rescue,’ ‘Disaster Rescue,’ and ‘Reset.’*

*Now, to test the functionality, click on these radio buttons one by one:*

* *Click ‘Water Rescue’: This should filter the data in the table and update the displayed data accordingly.*
* *Click ‘Mountain Rescue’: This should apply a different filter and update the table again.*
* *Click ‘Disaster Rescue’: Similarly, this should apply a different filter and update the table.*
* *Click ‘Reset’: This should reset the filters and display the original unfiltered data in the table.*

*By clicking on these radio buttons, you can test how the dashboard responds to different filter options, ensuring that the data table updates correctly based on your selections.*

### Screenshots

*Grazioso Salvare Logo*

*A red outline of a person

Description automatically generated*

*A red line with a white background

Description automatically generated*

*Starting state.*

*A screenshot of a computer

Description automatically generated*

*Water Rescue is selected.*

*A screenshot of a computer

Description automatically generated*

*Mountain Rescue is selected.*

*A screenshot of a computer

Description automatically generated*

*Disaster Rescue is selected.*

*A screenshot of a computer

Description automatically generated*

*Reset mode.*

*A screenshot of a computer

Description automatically generated*

*Pie chart and Map for Water Rescue.*

*A map with a purple pin

Description automatically generated*

*Pie chart and Map for Mountain Rescue.*

*A screenshot of a map

Description automatically generated*

*Pie chart and Map for Disaster Rescue.*

*A screenshot of a map

Description automatically generated*

*Pie chart and Map for Reset mode.*

*A screenshot of a map

Description automatically generated*

## Contact

Your name: Hong Luu