

**McCOY ENERGY**

**WEB DASHBOARD SYSTEM**

**&**

**DATA WAREHOUSING**

**22sum - CIS 5369**

**INDEPENDENT STUDY CIS**

**Project Work Report**

*By:*

**NEERAJ KUMAR REDDY PANTA**

**(gdf30@txstate.edu)**

*Under the guidance of*

**Dr. Sam Lee**

**(Professor - CIS & Quantitative Methods)**

**Table of Contents**

[Introduction](#_762zz8qwu9me) 2

[Softwares, Programming Languages & Libraries Used](#_23pz1s8qmcl2) 2

[My Work & Implementation](#_48lyegyjodp9) 3

[Appendices](#_lqu8o7jafpy5) 5

[Results](#_l2k1ganz0tcl) **13**

# **Introduction**

For the CIS5369 Independent Study project, I have opted to work on the McCoy Energy web dashboard to visualize and display the trends of the electricity consumption and the temperature trends through the sensor provided data for the McCoy College of Business Building. I have also opted to work on the Data Warehousing model which is to be constructed for better storing of the Consumption and Temperature data.

# **Softwares, Programming Languages & Libraries Used**

1. *Google Cloud Platform (GCP)* offered by Google is a cloud service that offers computing, storage, big data, networking, and many more, as well as the cloud management, security protocols and services and developer tools. *Google Cloud SDK or CLI* gives the ability to create and manage App Engine
2. *JetBrains Pycharm* is a dedicated Python Integrated Development Environment (IDE) that provides a wide range of essential tools for Python developers that are tightly integrated to create a convenient environment for productive Python, web, and data science development.
3. *Python* is the computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
4. *HTML* provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript. *CSS* is used to control presentation, formatting, and layout. *JavaScript* is used to control the behavior of different elements.
5. *JSON, FLASK, PLOTLY EXPRESS, SQLALCHEMY* are the other libraries being used.
6. *Pandas is the* high-level data manipulation tool that is built on the numpy package library.*Numpy* is a library for Python that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
7. *Structured Query Language (SQL)* a programming language used to manage relational databases and perform operations on data.
8. *MySQL Workbench* is a Oracle Corporation based visual database design tool that integrates the SQL development, design, administration and maintenance of SQL relational database management system.

# **My Work & Implementation**

For the **McCoy Energy Web Dashboard System** The deployment of the system is through the GCP App Engine. Here I have studied and successfully modeled the FLASK app module structure as the layout connection that renders all the HTML pages into one single website using python. Alongside the FLASK App part in the python file, SQLAlchemy library is being used to access the data from the GCP based MySQL Server’s database. For accessing the database after the SQLAlchemy library is imported I specify the MySQL server hostname from the GCP console along with the specific database and table names and we have to mention the username and password we have specified while creating the SQL Database in the GCP console. After entering all the details in the python code file we then create a sqlalchemy engine by adding the drivename alongside other details.

As we have constructed a Data Warehousing Model I used the PANDAS library along with SQL Query statements to segregate the whole data based on the type\_key as we have the three different types of energy consumptions which are Chilled Water, Steam and Electricity data in a single column labeled energy consumption. After the SQL Query to read the data I assigned 3 different dataframes to 3 different Energies . In the case of the Flask App it is divided into multiple routes which are responsible for a specific path and the function under that path renders a specific HTML page. The “\dashboard” route uses the data frame that contains the results and I convert the values of some columns based on the given data dictionary and then use it to Plot the various graphs based on the respective data by using PLOTLY Express Library in python. For plotting of the figure, first I defined a figure with the number of subplots by mentioning the number of rows and columns. Along with this I use the NUMPY library function for modeling and plotting one of my subplots which shows the weekends and working days.

In the case of the HTML pages in common I have designed a CUSTOM logo for the Dashboard system, and clicking on the home logo will redirect back to home page and have made sure that each Page has a title in the tab bar and also style each page by adding some BOOTSTRAPPING elements and styling them with CSS. The index / home page shows the Title and some description on what the entire dashboard is all about.

In detail, The Dashboard Html page is divided into 4 sections, one section contains Date, Time and Weather of San marcos. The Second Section of the page consists of Two split sections with the sections on the left containing only 1 graph and the one on the right containing 4 graphs in an OverScroll mode. The third section holds some Energy consumption tips and the final section holds the logo of other university buildings if this is planned for future use. Furthermore, Specifically for the Dashboard Page of the Web Dashboard System I have used a total of three JavaScript elements. The first JavaScript element takes the data from the dataframe that contains the data from the database and uses the GOOGLE Visualization charts to display the total consumption of energies in a pie chart using dictionaries and for loops. The Second JavaScript Element is used to display the LIVE Date and Time by making use of the system's local time it is on. For the third and final JS element I am using a Weather Crossing WEB API which uses an API KEY, Location and Unitgroup and have specified the forecast days to display the weather forecast.

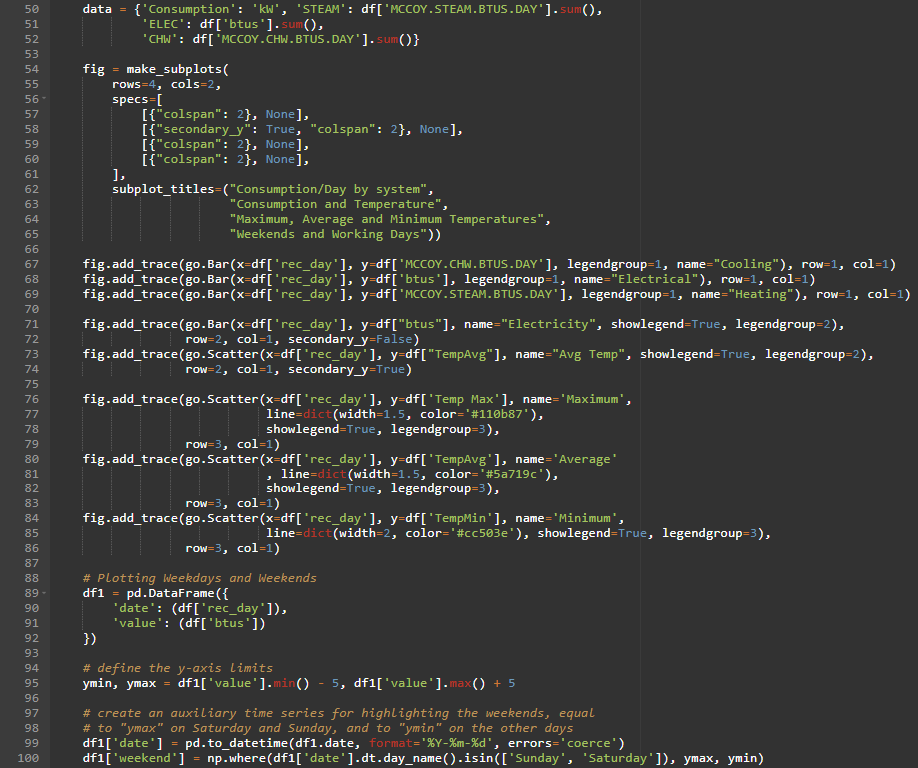
The Data page has 3 column layout with each column having a heading being a hyperlink which redirects to my Texas State OneDrive where I have stored the Data we are using for Visualization, predictions and analysis part for access to others. The Contact Us page consists of 2 columns with a column with a mail address and the other with the phone number. The predictions page has a 2 Column layout with each specific image having a description about it. Also each of the pages has a footer at the bottom with the Copyright label mentioned.

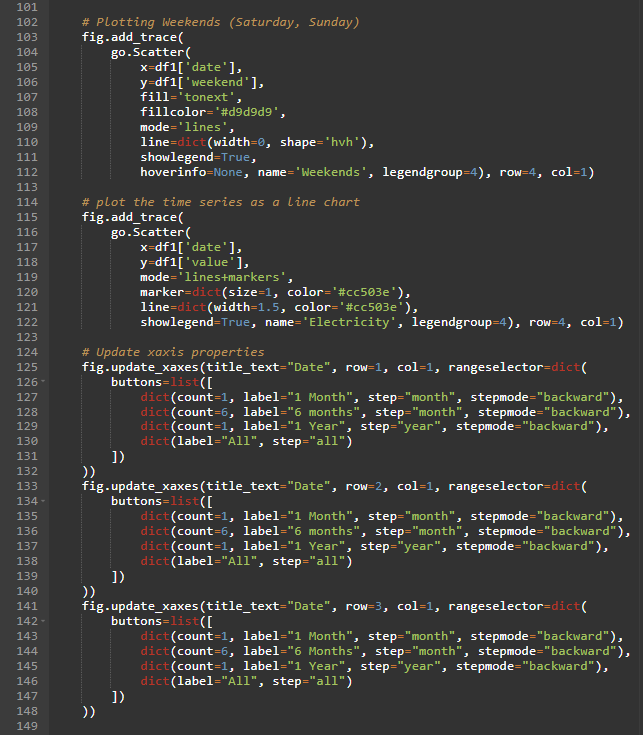
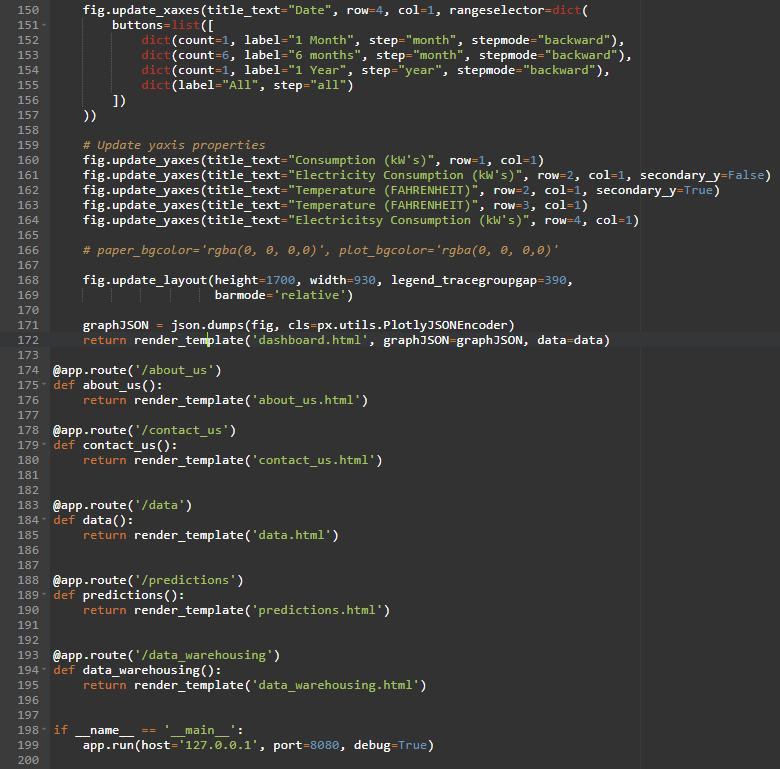
There is also a python file where I worked on loading the data from the WEB API to the SQL database. For doing this I learned and worked on JSON file i.e. the link that I get from the Visual Crossing Website is in the form of JSON, so first I had to check if it is forecasting and if it is then decode the data from JSON into “UTF-8” then connect to the MySQL database in the same method that I worked in the FLASK App and then load the elements of the JSON decoded data into the specific columns by specifying the column names.

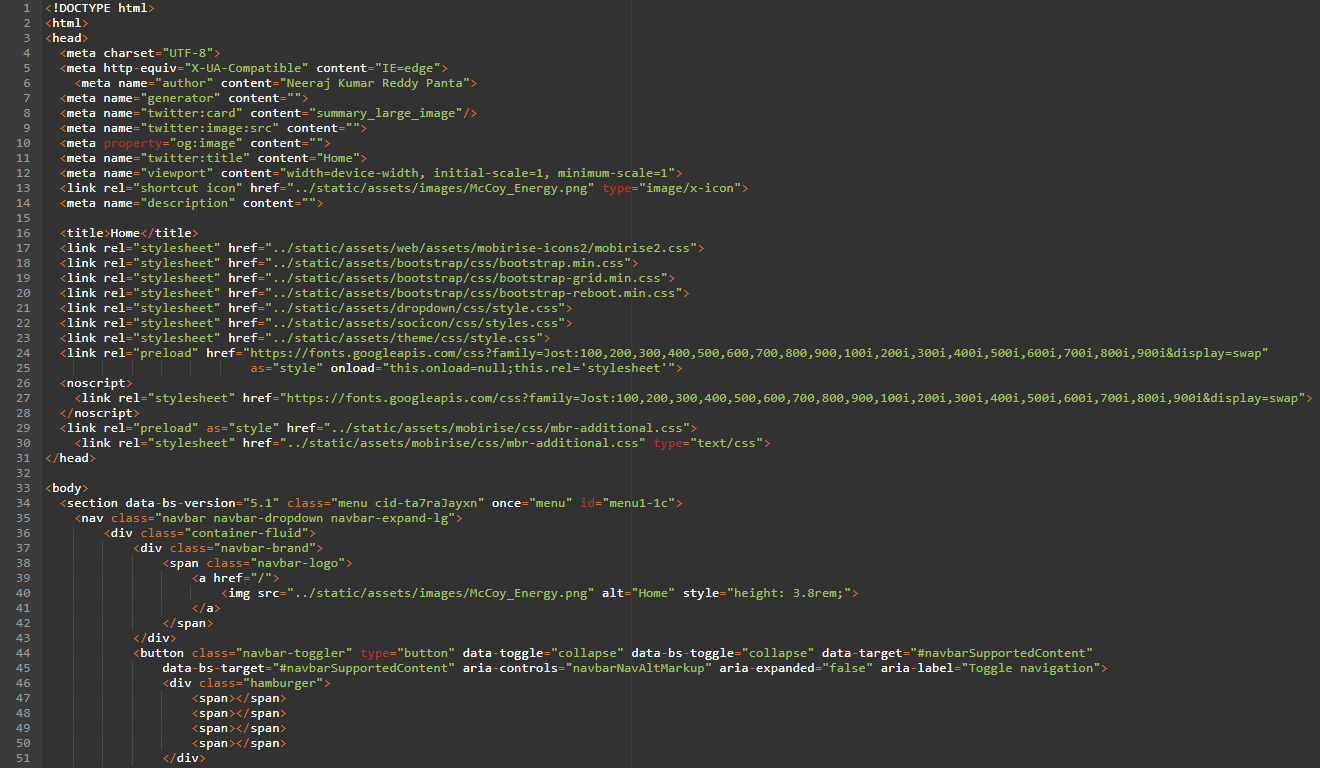
Before deploying FLASK App on Google Cloud I had to create a Google Cloud Account, then Install the gcloud Software Development Kit (SDK) or the Command Line Interface (CLI) this gives me the ability to create and manage App engine and after doing a bit of research I found that I could use it for everything from creating projects, storage buckets which are similar to Amazon S3 and database instances similar to the Amazon RDBMS and many more. After the Installation of the gcloud SDK I had to use it to install the App Engine Extension to deploy our app. Before deploying the app on gcloud I had to create a “app.yaml” file, this specific file tells the App engine what version of the programming language we are using such as “*runtime: python39*” which is the runtime version we are using in this project. When trying to deploy we have to set the project we are working on as the default and also enable billing, cloud build API, only by doing all this I was able to deploy the google cloud app.

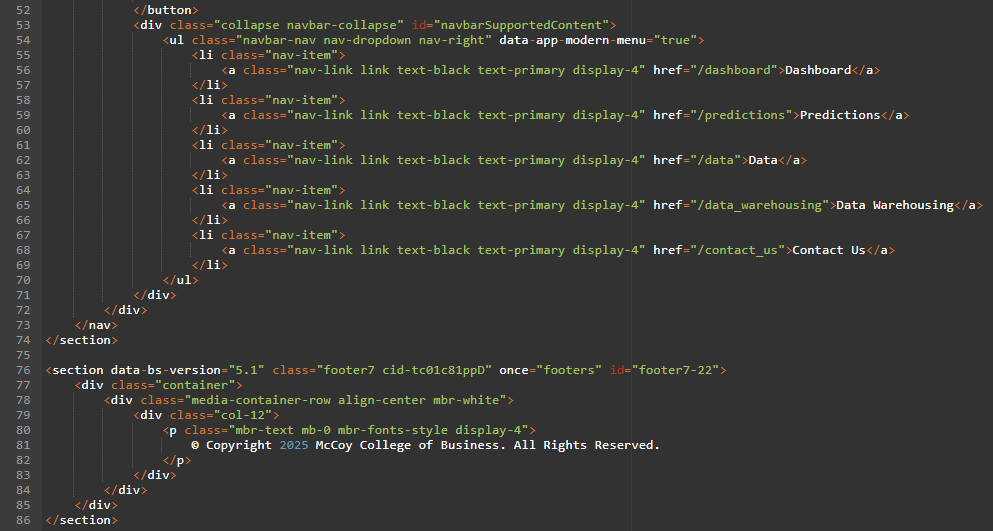
I have uploaded all my source code in my Github Account, the following is the corresponding link: <https://github.com/Neeraj441/mccoy_dashboard>. I have also worked on Data Warehousing along with pooja where I learned the theoretical and practical usages of the schema, ETL scripts, also I’m using the Data Warehousing Model in the dashboard to visualize the data. While working on this I have also enhanced my skills on the basic concepts of SQL, PANDAS and NUMPY. I have also improvised myself with MySQL Workbench Software. I have also worked on altering and updating and merging multiple tables.

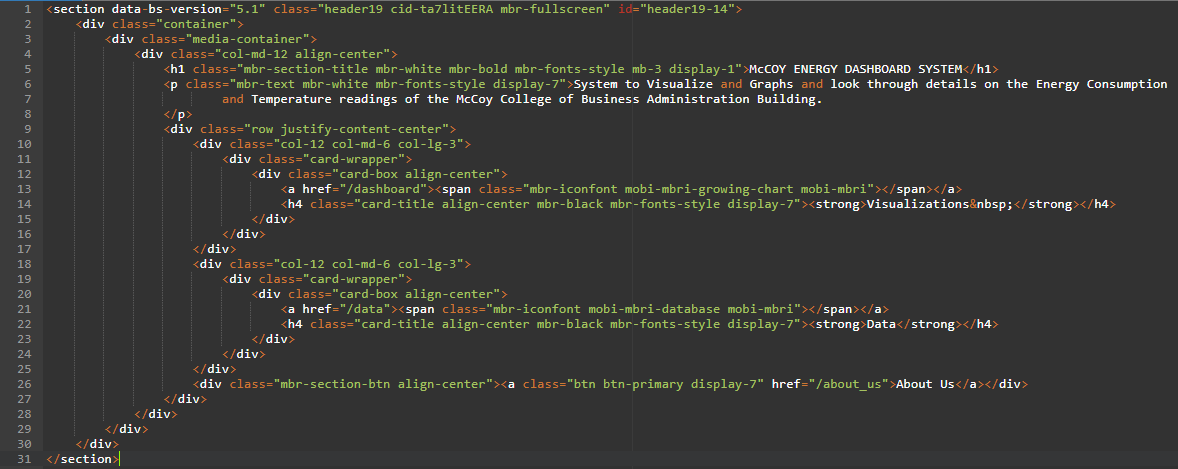
# **Appendices**

*Pages 5 & 6 are the images of main.py python code which corresponds to the FLASK App, database connectivity, retrieving data and also using PLOTLY to add elements of graphs to a particular app route html page.*

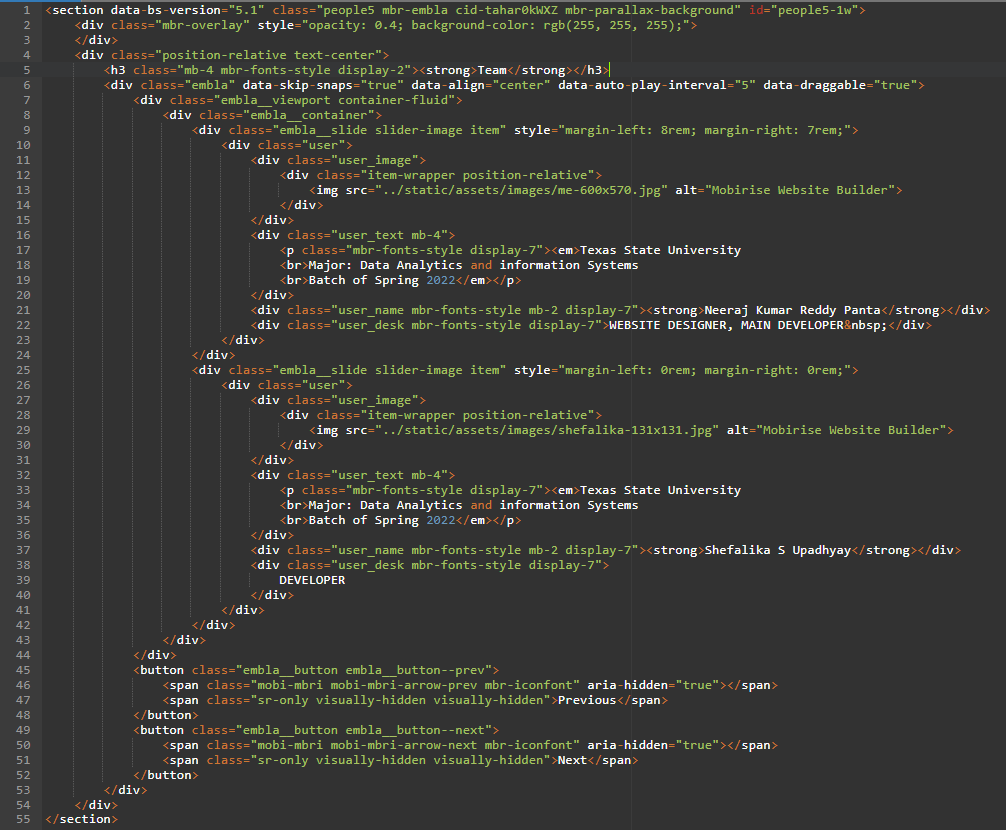


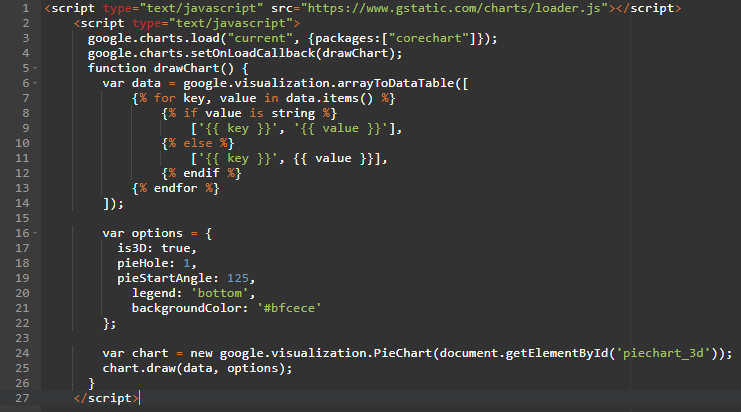
*The <head>, <footer> HTML Code Elements are Common to all HTML Pages and one or two of the pages have an addition of any <scripts> attached to them. Also the Navigation Bar section from the <body> Element is also common across all the HTML Pages. Below is the Image of the Code Holding the common Elements.*



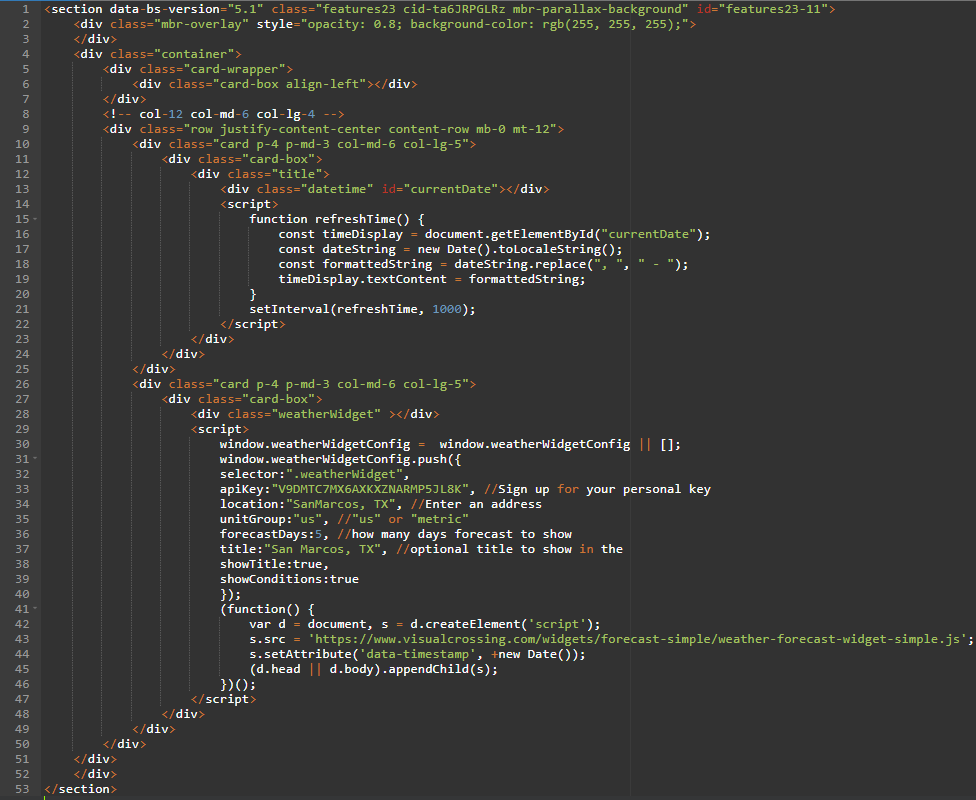
*Below is the Figure containing Home / Index HTML Page <body> Code that displays the Content of the Page.*

*Below Figure Holds the main <body> content of About US HTML page.*

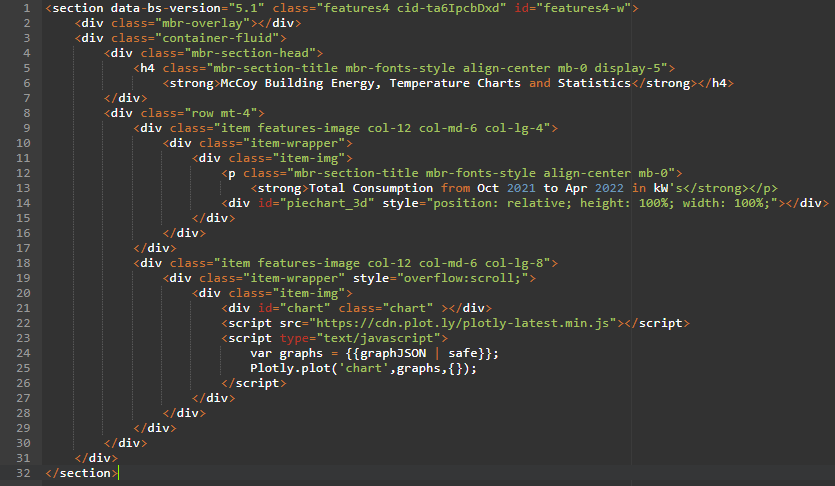


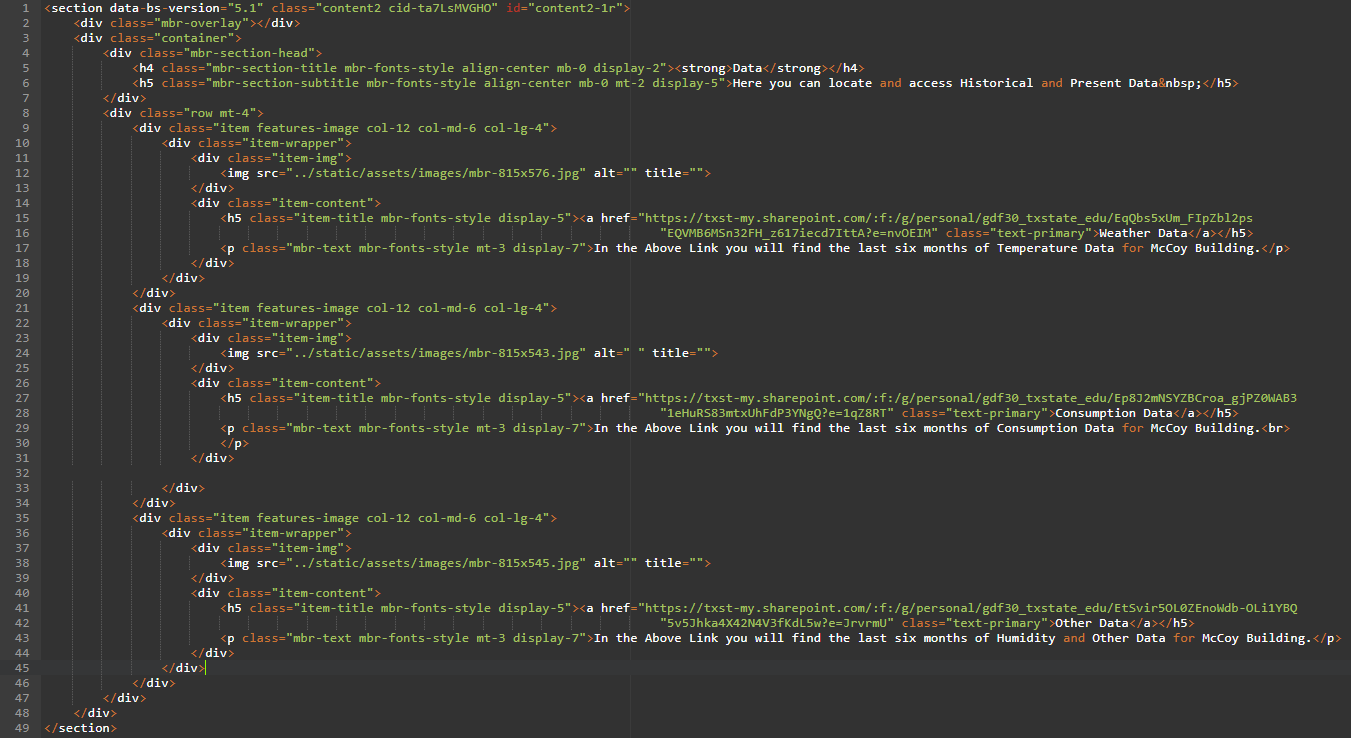
*Figure containing Dashboard HTML page <script> code responsible for the Total Consumption PIE Chart.*

*Below Figure holds the Code for the Date, Time & Weather <section> of Dashboard html Page.*

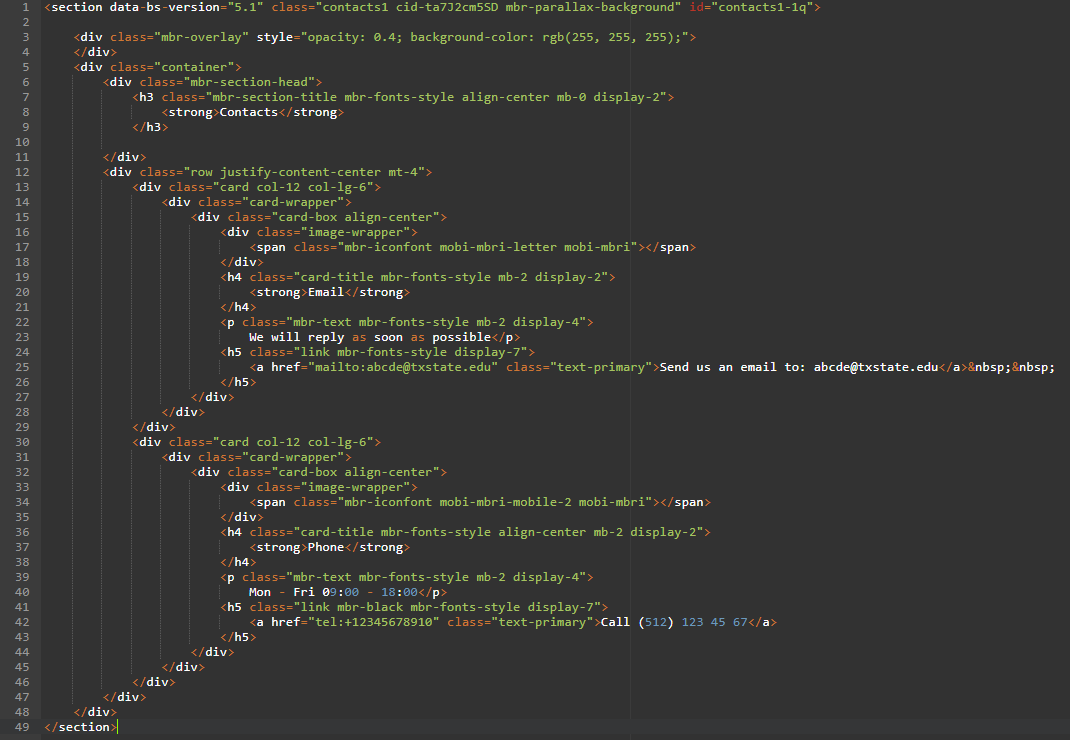


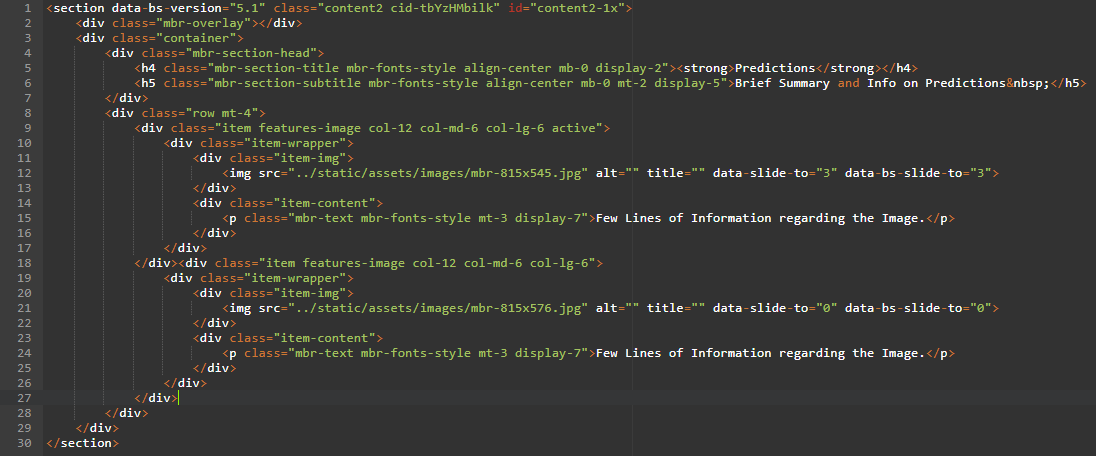
*Below Figure is the HTML <section> Code of the Dashboard Page that Integrates the main.py that contains FLASK App which has the plotly code for plotting the Graphs.*

**

*Below Figure contains the <section> code for the Data HTML Page.*

*Below Figure shows the <section> code for the Contact US HTML Page.*

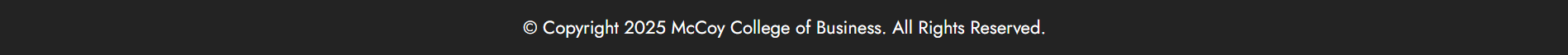
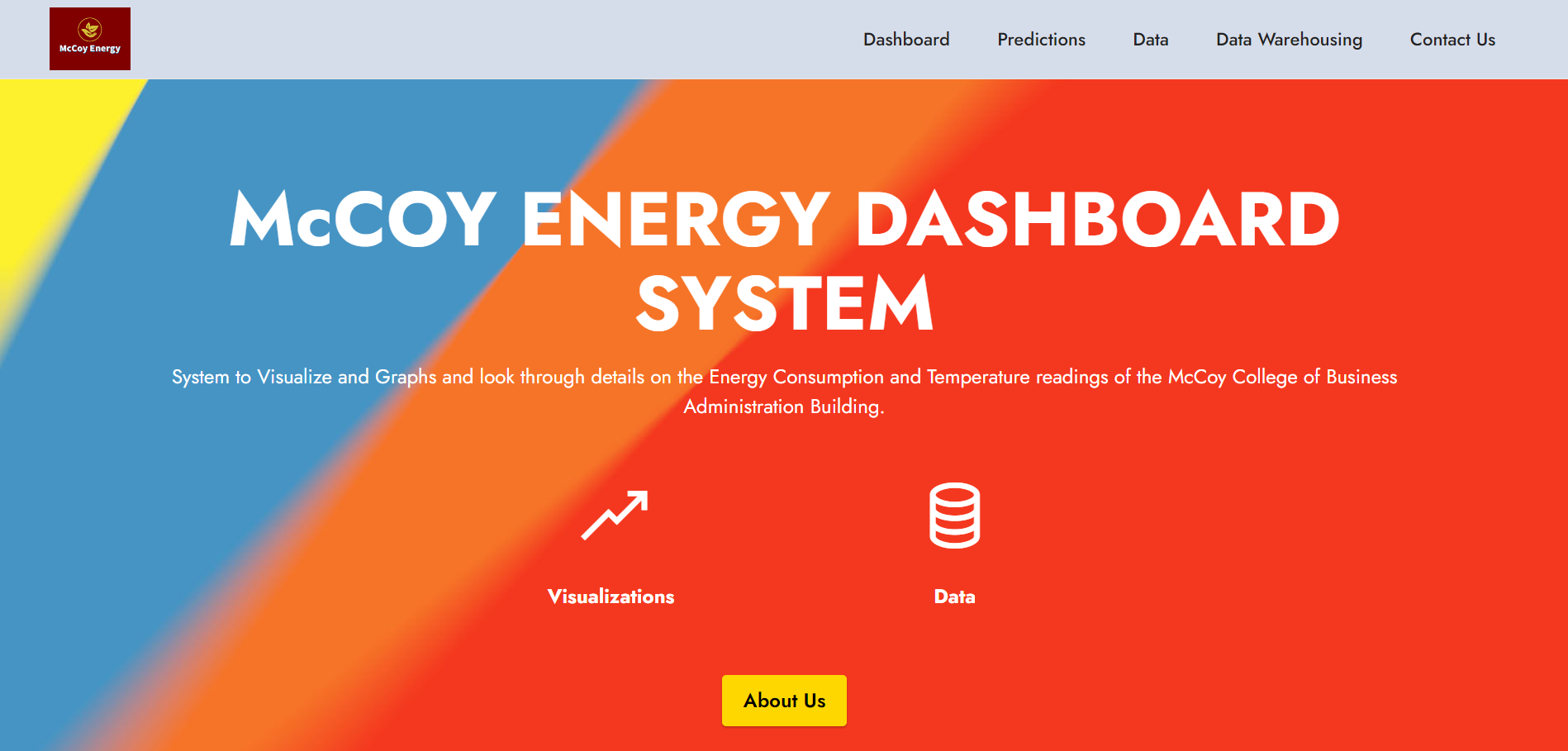


*Below Figure is the <section> Code for the Predictions HTML Page.*

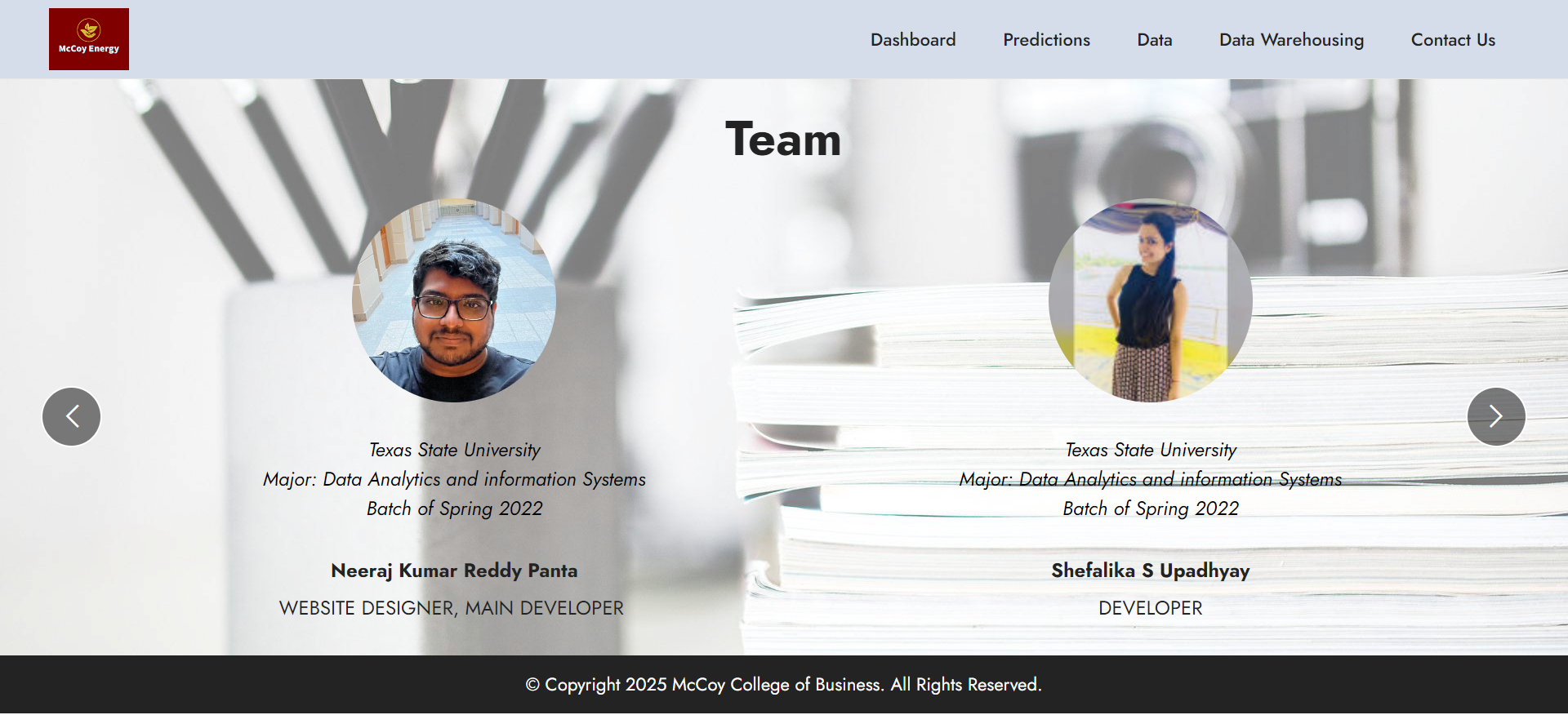
# **Results**

*Link for the Website:* [*https://turnkey-crowbar-351921.uc.r.appspot.com/*](https://turnkey-crowbar-351921.uc.r.appspot.com/)

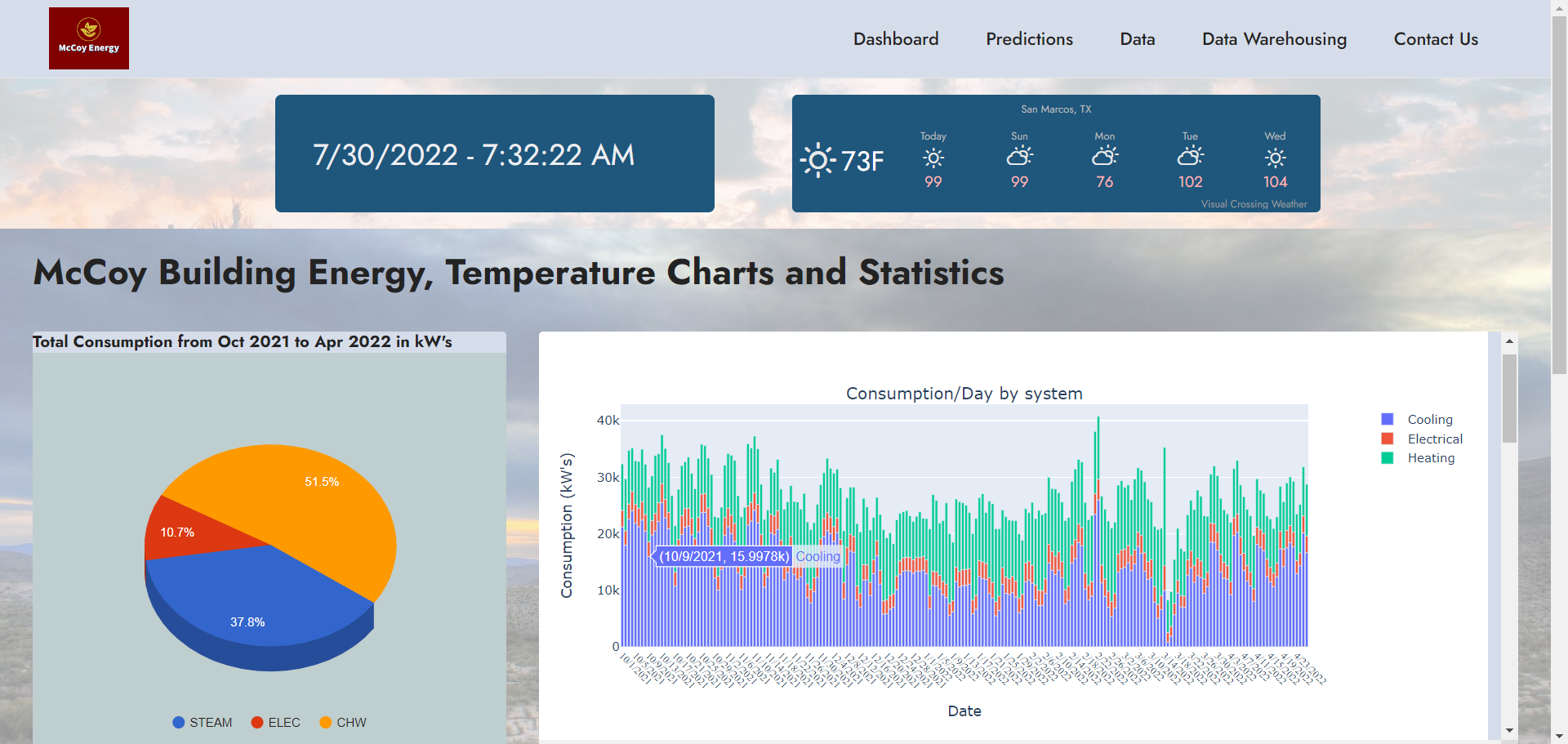
*Home / Index Page*

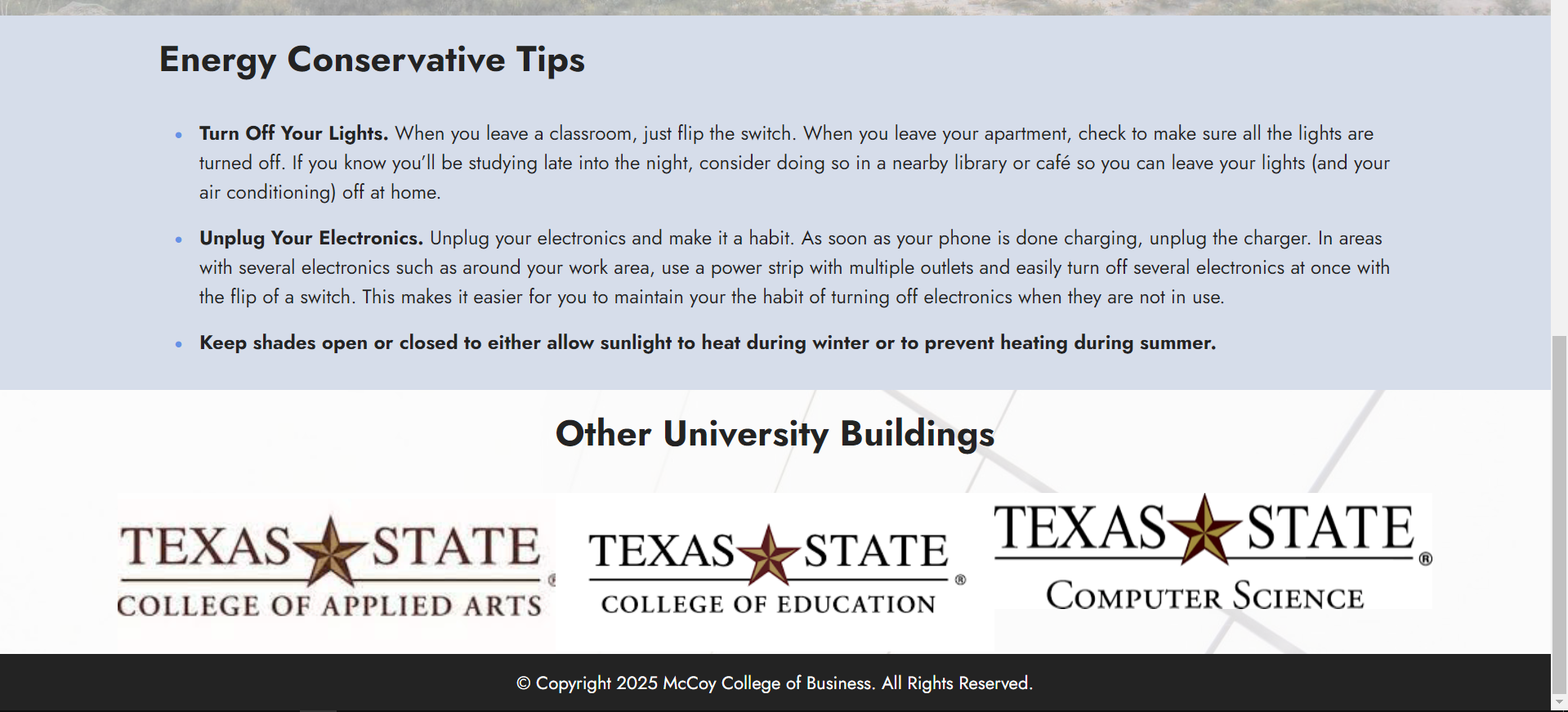


*About Us Page*

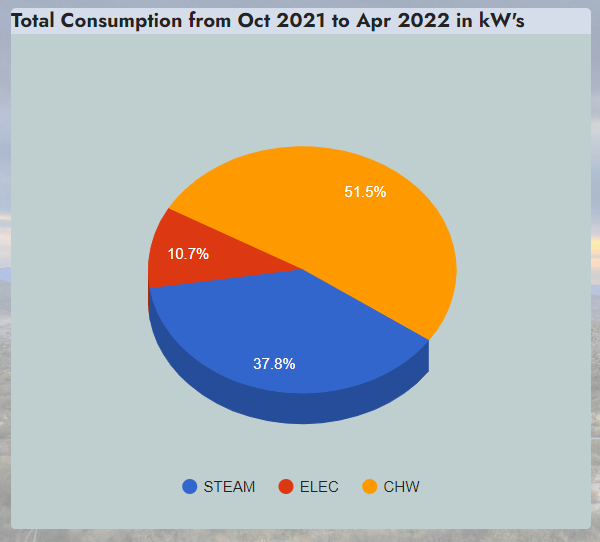


*Dashboard Page*

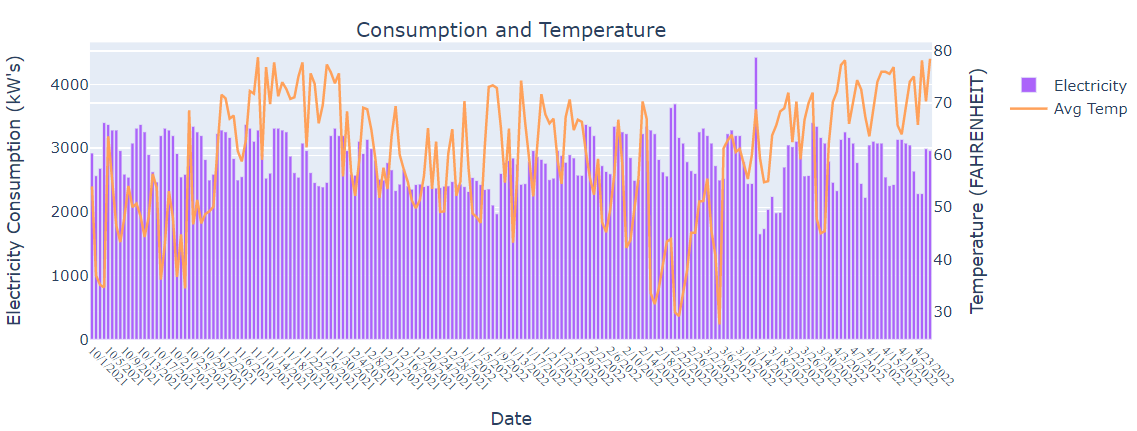
**

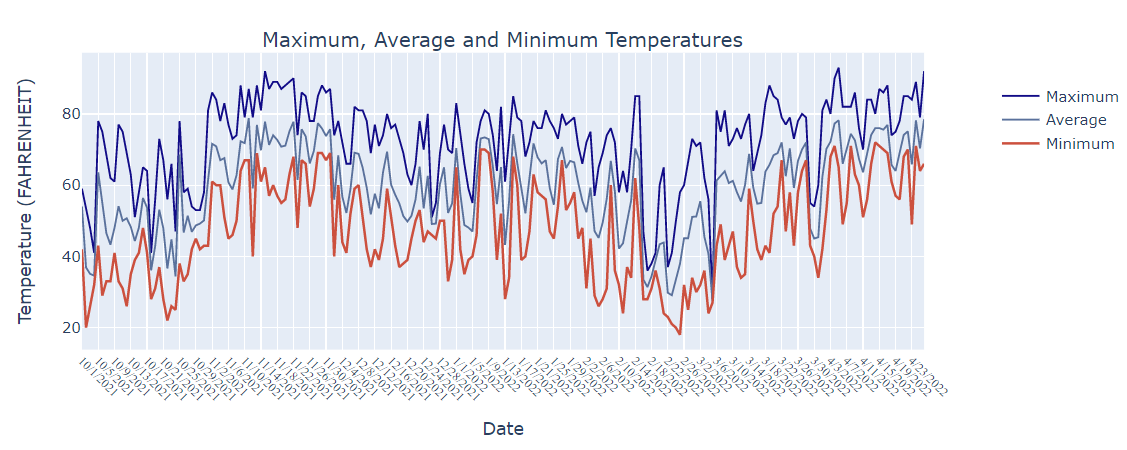


*All Graphs in the Dashboard Page*



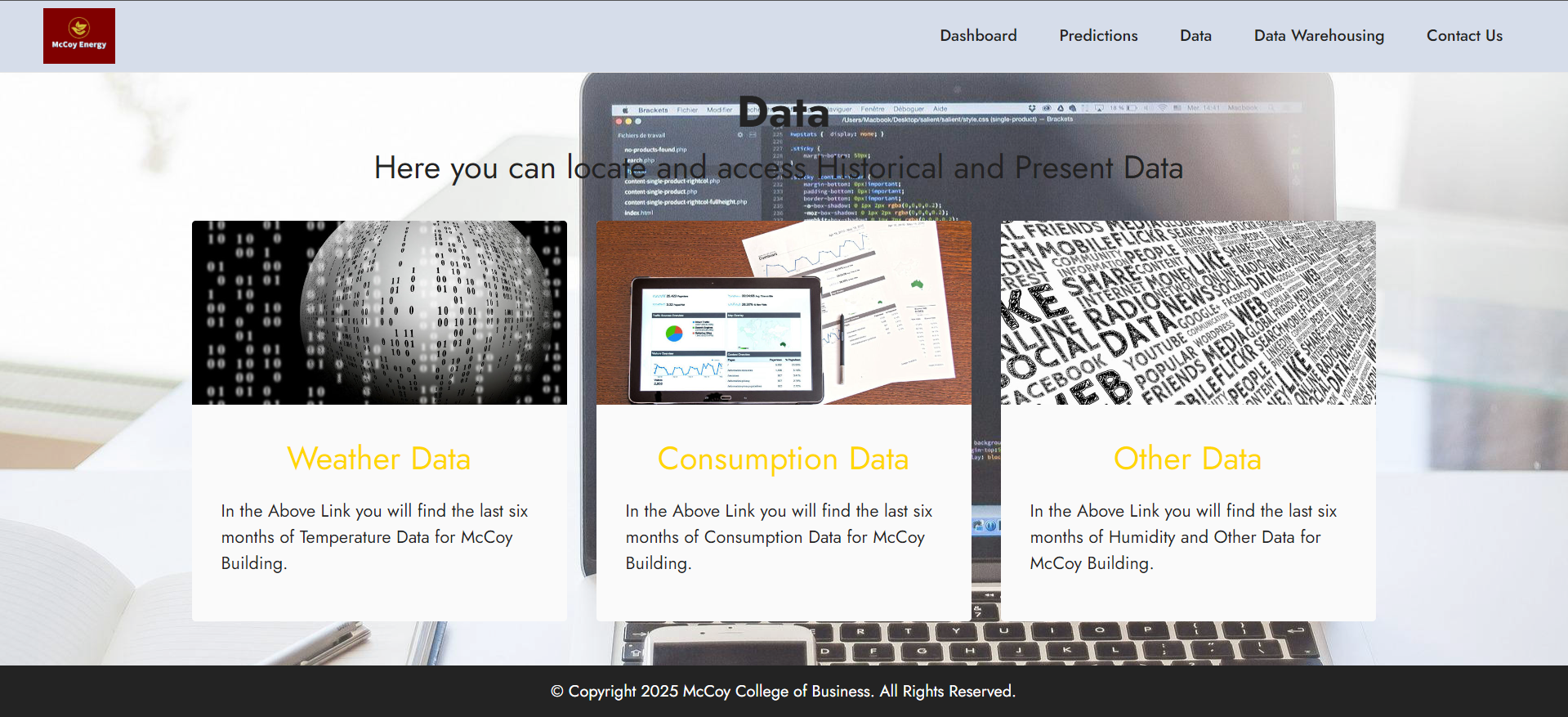








Data Page



Contact Us Page

