**Documentation**

**Selenium – To Scrap Data**

As per our requirement Selenium downloaded and imported the required library

The url of the website (selected government bus) is assigned to the variable

Declaration of variable of datatype List to hold all bus details

Calling the function **scrape\_bus\_details** to Scrape routes and details from all pages to get the bus details

Iterating the number of pages in for loop and calling function **initialize\_load\_Webdriver** to open a chrome window, maximize window and load the url .

If condition is executed if page number is greater than one to load the next page until the end page using pagination tab.

Calling the function **fetch\_bus\_routes**  to get all the route name and route link in that particular page

Iterate over each bus route link and route name to scrape the bus details for each route

Now the function **scrape\_bus\_details** by passing the route link and route name is called and the page is loaded and click the "View Buses" button if it exists and scroll down to load all bus items and find bus item details(bus name, bus type, departing time, duration time, reaching time, star rating, price, seat availability)by using the **CLASS\_NAME** and **XPATH**

Declare variable list **bus\_details** and looping the data which are getting from the bus item details(bus name, bus type, departing time, duration time, reaching time, star rating, price, seat availability)which is converted in dictionary format and use append function to append the bus details to **bus\_details**

All the bus details are now extended to the global variable list

Convert the list of dictionaries to a DataFrame

Save the DataFrame to a CSV file.

**SQL**

As per requirement connection between python and database is made by installing **pip install pymysql ,** **./ Scripts/activate** and  **pip install mysql-connector-python**Query to connect with the database is done by passing the hostname, username, password and database details.

Once the connection is successfully created, Create a table and column definitions are done.

All bus csv files names are added in the form as list and import pandas by **pip install pandas (import pandas as pd)** and Now csv files are read, by using the **pd.read** function which is done through a loop. All csv files are combined using pd.concat in **Combined\_df** variable and convert the **combined\_df.to\_csv** to **bus\_routes.csv** to get the all files as one.   
  
Now the bus\_routes.csv file is read and which is in dataframe format.

Iterating the dataframe object to insert into the table which is done by passing column declaration using the variable and the values are taken from dataframe using iloc

**Streamlit**

As per requirement Streamlit is downloaded

Header is created using streamlit as per our wishes for the app.

Connect with sql is done by writing query by passing the hostname, username, password and database details

Sidebar input text box is created and title is given for entering the route name with default value as empty

If the textbox has a value route names are fetched by searching the starting letter from database which is done by calling the function **fetch\_route\_names**. This function will get all the route name that has starting letter of characters we passed in textbox which is fetched from databse using query. By using pandas, data from database are read and converted as list.

In the sidebar for fetched routenames radio button is created on click to get all information related to the selected route\_names.

And we have a select box which will fetch the data from database according to the price filter. We have Low to High and High to Low option. By default we use Low to high which will show price in ascending order option selected which will call a function **fetch\_data**

This function will get all information based on selected **r**oute name and price sort order as per ascending and Descending order as per option selected in price selectbox which is converted into dataframe in the main screen.

Filter box is created in main page for Star Rating and Bus Type where both filters are multiselect and mandatory field to get the required output.  
  
Now by calling the function **filter\_data** which will filter the fetched data accordingly based starting rating and bus type from the dataframe.