**Redbus Data Scraping with Selenium & Dynamic Filtering using Streamlit**

This documentation outlines the process for scraping bus data from the Redbus website, storing it in a MySQL database, and creating a Streamlit application for data filtering and analysis. The project involves the following key components:

1. **Web Scraping using Selenium**: Automating the browser to scrape data from the Redbus website.
2. **Database Interaction**: Using MySQL to store the scraped data.
3. **Streamlit Application**: Creating an interactive application to filter and analyse the data.

**Prerequisites**

* Python 3.x
* Selenium
* MySQL
* Streamlit
* WebDriver for the browser (Chrome Driver for Google Chrome)

**Setup Instructions**

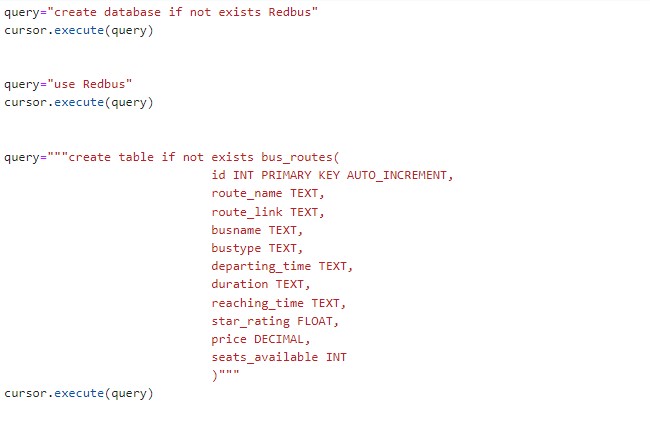
1. Install Dependencies

Installed the required Python packages using pip:

pip install selenium mysql-connector-python streamlit

1. MySQL Database Setup

Created a MySQL database named Redbus and a table named bus\_routes using the following SQL script:



**Code Explanation**

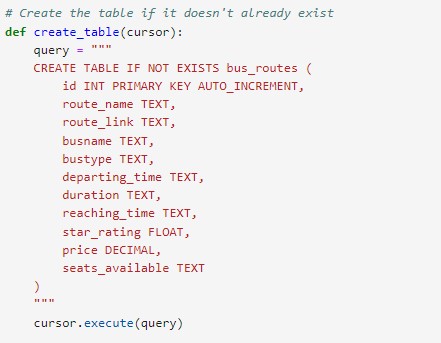
Database Connection

The **get\_database\_connection** function establishes a connection to the MySQL database.



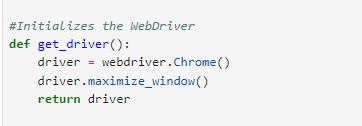
**Creating the Table**

The **create\_table** function creates the bus\_routes table if it does not already exist.



**Initializing the WebDriver**

The **get\_driver** function initializes and returns a Selenium WebDriver instance.



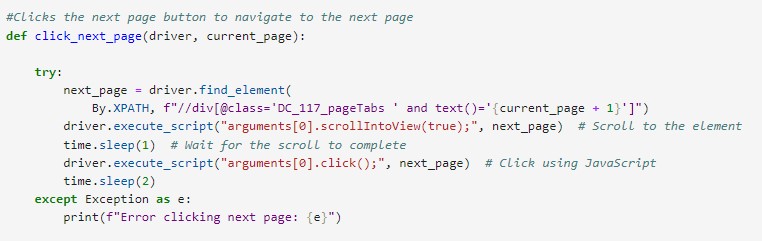
**Scraping Bus Routes and Links**

The **scrape\_current\_page** function scrolls through the current page to load all bus routes and extracts the route names and links.



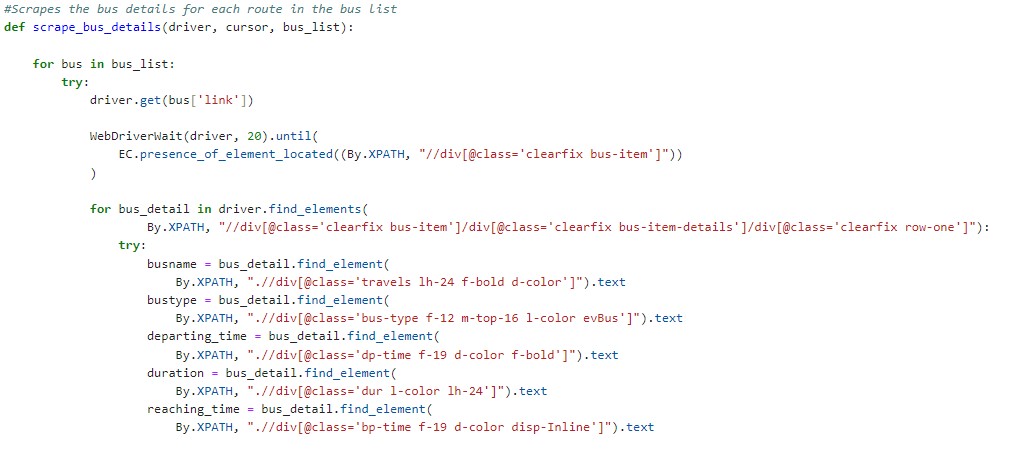
**Clicking the Next Page Button**

The **click\_next\_page** function navigates to the next page of bus routes.



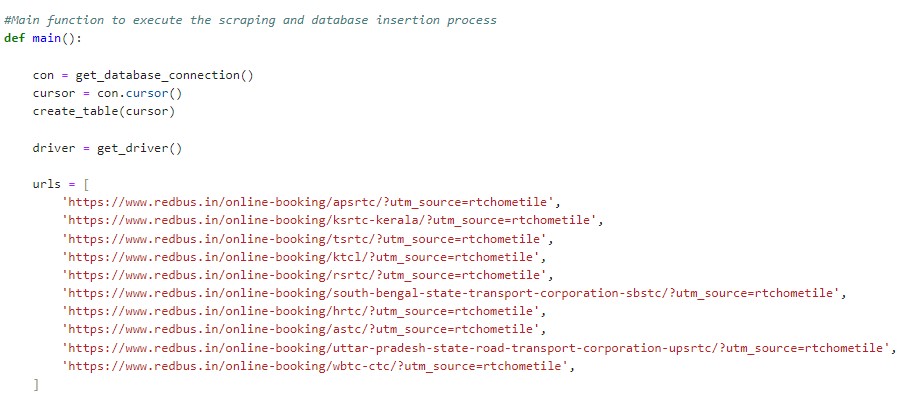
**Scraping Bus Details**

The **scrape\_bus\_details** function extracts detailed information about each bus on the route.



**Main Function**

The main function manages the entire process, from initializing the database and WebDriver to scraping data and inserting it into the database.





**Technical Tags:**

* Web Scraping
* Selenium
* Streamlit
* SQL
* Data Analysis
* Python
* Interactive Application

**Running the Script**

To run the script, executed the following command terminal:

python scrap.py

**Streamlit Application**

Created a streamlit.py file for the Streamlit application:

**Data Analysis/Filtering using Streamlit:**

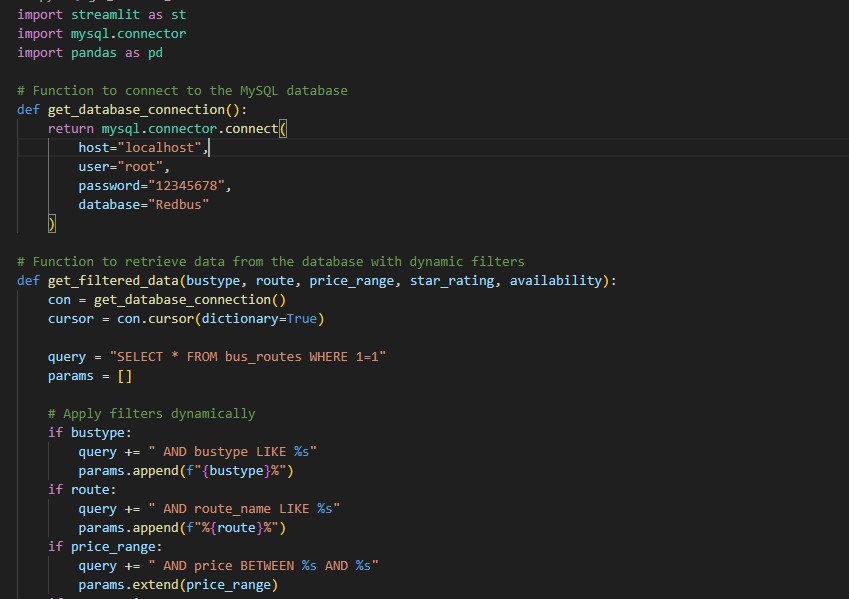
Used SQL queries to retrieved and filtered data based on user inputs.

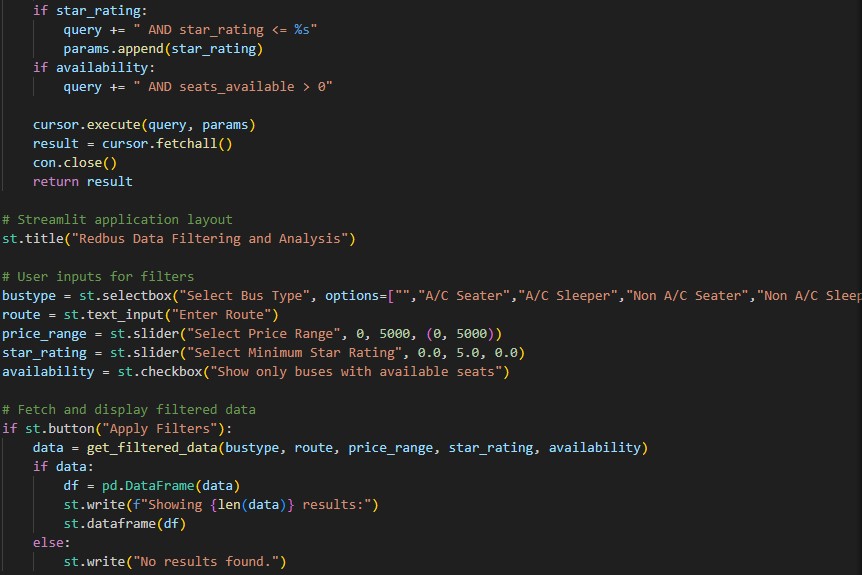
Used Streamlit to allow users to interact with and filter the data through the application.

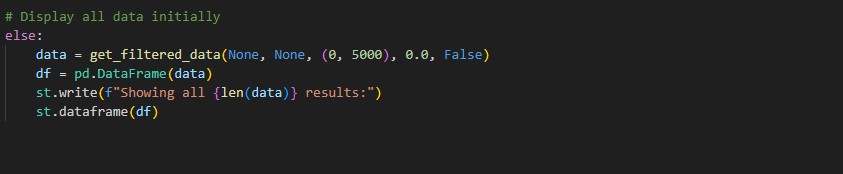
**Streamlit Application:**

The Streamlit application to display and filter the scraped data.

Implemented various filters such as bustype, route, price range, star rating, availability.

****





**Database Schema:**

**Table Name: bus\_routes**

**Primary Key:** To ensure each record is unique, an auto-incrementing primary key (id) is used.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | INT | Primary Key (Auto-increment) |
| route\_name | TEXT | Bus Route information for each state transport |
| route\_link | TEXT | Link to the route details |
| busname | TEXT | Name of the bus |
| bustype | TEXT | Type of the bus |
| departing\_time | TEXT | Departure time |
| duration | TEXT | Duration of the journey |
| reaching\_time | TEXT | Arrival time |
| star\_rating | FLOAT | Rating of the bus |
| price | DECIMAL | Price of the ticket |
| seats\_available | INT | Number of seats available |

GitHub Link:

https://github.com/vishnupriya08-hub/redbus-data-scraping