**Project: DS\_RedBus**

**Title: Redbus Data Scraping with Selenium & Dynamic Filtering using Streamlit.**

**Aim:** Developing a web scrapper to automate the extraction of the bus routes schedules details and information from the redbus website for multiple states. Storing the data in MySQL database and visualize it using a Streamlit.

**Primary Aim for the study:**

* **Extracting data:** Data extracting from each bus routes of the states, collecting details such as route name, bus name, bus type, duration, reaching time, star rating, price and seat availability. Which automatically navigate to multiple pages for each bus route states.
* **Organizing the data:** The data scraped from all the bus routes of state are organized and convoluted into sing all\_bus\_details.csv file.
* **Storing the Data:** New database/schema is created according to requirement “bus\_routes”.Data collected from all the bus routes which was appended and stored in the single csv file is formatted to fit in to required predefined MySQL database.
* **Visualization:** To develop a Streamlit app to visualize and analyze the stored data.

**Solution Approach:**

* Using Selenium to automate web browsing and extracting data from redbus website, redbus.in
* To Handle dynamic content loading and pagination.

1. **Web scraping:** To scrape bus routes and details from RedBus, we need to first initialize a web driver, open a browser window, and navigate to the RedBus website. Then, we should load the specific URL for the target state, handling any loading delays. Once the page is loaded, we can scrape all the bus route links and names from the page, managing pagination to capture all available routes. For each bus route, we need to navigate to its respective link and extract detailed information about the available buses, including their names, types, departing times, durations, reaching times, star ratings, prices, and seat availability. Finally, we should implement error handling to gracefully handle missing elements or loading failures, logging errors and continuing the scraping process.
2. **Stitching multiple CSV files to single CSV File:**

**Approach:**

* + Using glob library to read multiple file csv file and storing it in a DataFrame
  + Converting the DataFrame in to csv file named ‘all\_bus\_details.csv’.

1. **MySql Database Integration:**

**Approach:**

* Using python and mysql-connector-python to create a database / schema according to the specified requirement.
* Creating a table from the python notebook in the specified requirement.

|  |  |  |
| --- | --- | --- |
| **Column names** | **Data types** | **Description** |
| Id | INT PRIMARY KEY AUTO INCREMENT | Primary Key (Auto-increment) |
| Route Name | TEXT | Bus Route information for each state transport |
| Route Link | TEXT | Link to the route details |
| Bus Name | TEXT | Name of the bus |
| Bus Type | TEXT | Type of the bus |
| Departing Time | TIME | Departure time |
| Duration | TEXT | Duration of the journey |
| Reaching Time | TIME | Arrival time |
| Star Rating | FLOAT | Rating of the bus |
| Price | DECIMAL | Price of the ticket |
| Seat availability | INT | Number of seats available |

* Using sqlalchemy library to insert multiple rows of data to the desired table in bus\_routes database.

1. **Streamlit App Development:**

**Approach:**

* Developing a Streamlit app to query the data and visualize the data from MySQL Database.

**Steps:**

* **Database Connection:** Establishing the connection to MySQL Database.
* **Query Data:** Fetch data from the database to be displayed in the app.
* **Filtering:** Use Streamlit components to filter the bus route name, price, bust type and star rating.

**Streamlit App Features:**

* Display a table of bus routes and schedules.
* Provide filters for searching by route name, bus type, departing time, etc.

By following this approach, I have automated the scraping of bus route details and stored the data in an MySQL database, and created an interactive Streamlit app for data analysis.