

Redbus Data Scraping with Selenium & Dynamic Filtering using Streamlit

**Prepared
ALIYAS
MD113**

**By
THOMAS**

Project Illustration

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

Objective: Develop a web scraper to automate the extraction of bus route details, schedules, and relevant information from the RedBus website for multiple states. Store the data in an SQL database and visualize it using a Streamlit app.

Scope:

- **Data Extraction:** Scrape bus route links, names, and detailed information for each route, including bus name, type, departing time, duration, reaching time, star rating, price, and seat availability.
- **Automation:** Automate navigation through multiple pages and states on the RedBus website.
- **Data Storage:** Store the scraped data in an SQL database.

- **Visualization:** Develop a Streamlit app to visualize and analyze the stored data.

Solution Overview

The solution involves three main components: web scraping, SQL database integration, and Streamlit app development.

1. Web Scraping

Approach:

- Use Selenium to automate web browsing and data extraction from the RedBus website.
- Handle dynamic content loading, pagination, and potential pop-ups.

Steps:

- Initialize Web Driver:** Open and maximize the browser, and navigate to the RedBus website.
- Load Web Page:** Load the specific URL for the target state, handling any loading delays.
- Scrape Bus Routes:** Identify and extract all bus route links and names on the page, managing pagination to capture all routes.
- Scrape Bus Details:** Navigate to each bus route link and extract detailed information about available buses, such as name, type, departing time, duration, reaching time, star rating, price, and seat availability.
- Handle Errors:** Implement error handling for missing elements or loading failures, logging errors and continuing the scraping process.

2. SQL Database Integration

Approach:

- Use Python's sqlite3 or another SQL database connector (like mysql-connector-python for MySQL) to store the scraped data.

Steps:

- i. **Database Setup:** Create a database and define a table structure to store bus route and schedule details.
- ii. **Data Insertion:** Insert the scraped data into the SQL database, ensuring data integrity and handling duplicates or errors.

Column names	Data types
Id	INT PRIMARY KEY AUTO INCREMENT
Route Name	TEXT
Route Link	TEXT
Bus Name	TEXT
Bus Type	TEXT
Departing Time	VARCHAR (100)
Duration	TEXT
Reaching Time	VARCHAR (100)
Star Rating	FLOAT
Price	FLOAT
Seat availability	INT

3. Streamlit App Development

Approach:

- Develop a Streamlit app to query and visualize the data from the SQL database.

Steps:

- i. **Database Connection:** Establish a connection to the SQL database.
- ii. **Query Data:** Fetch data from the database to be displayed in the app.
- iii. **Filtering:** Use Streamlit components to filter the bus route name, price 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.

Implementation Steps

1. Web Scraping

A. Initialize Web Driver:

- Use Selenium to open the browser and navigate to the target URL.
- Handle page loading and pop-ups.

B. Scrape Bus Routes and Details:

- Identify elements containing bus route links and names.
- Navigate to each route link and extract detailed information.

2. SQL Database Integration

A. Setup Database:

- Use `sqlite3` or `mysql-connector-python` to create a database and define table structures.
- Whether check datatypes are same.

3. Streamlit App Development

A. Setup Streamlit:

- Install Streamlit (`pip install streamlit`).
- Create a Streamlit script (`app.py`).

B. Database Connection:

- Use `sqlite3` or another connector to connect to the database.

C. Query and Display Data:

- Fetch data from the database and use Streamlit components to display it.

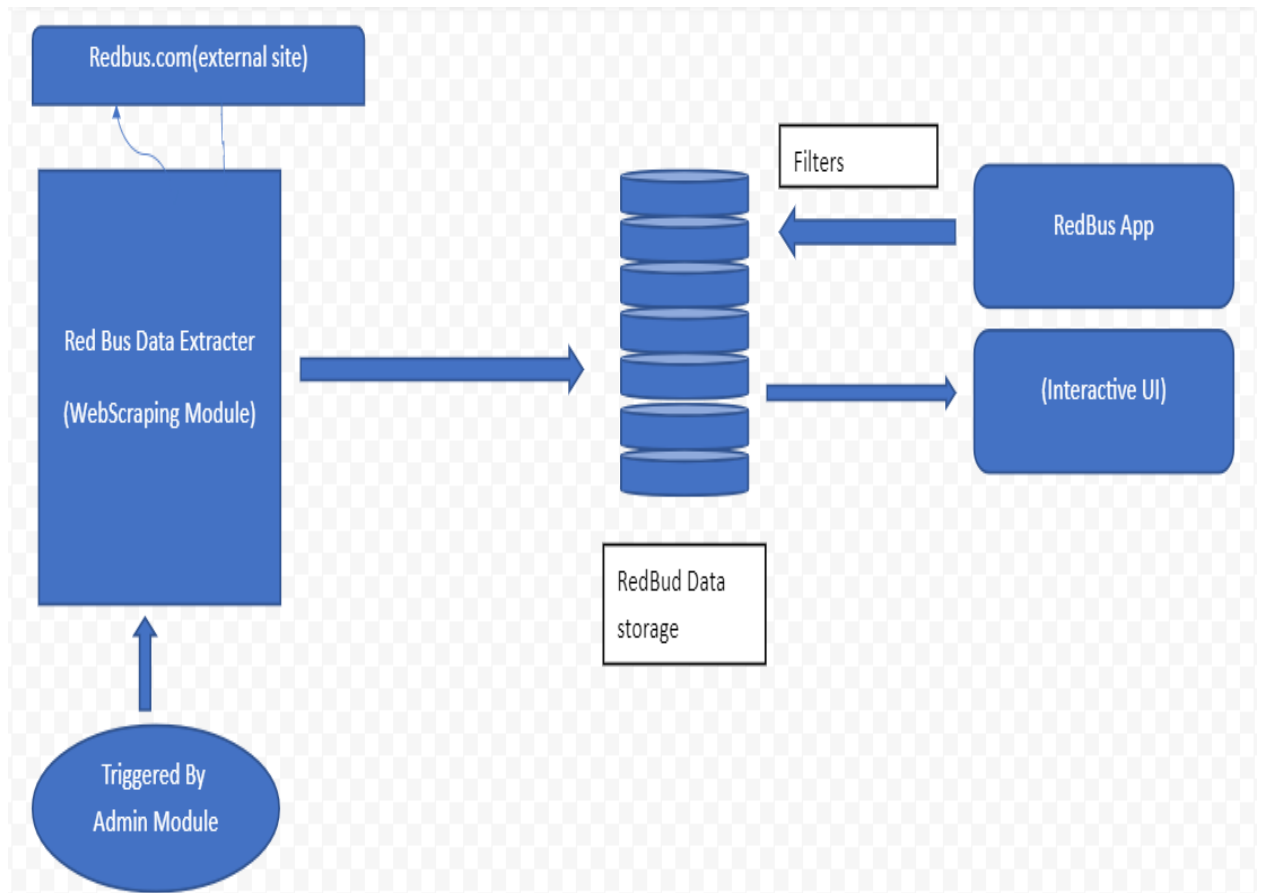
D. Add Filters and Analysis:

- Use Streamlit widgets to add filters for route name, bus type, etc.

- Use Streamlit to analyse data

By following this approach, you can automate the scraping of bus route details, store the data in an SQL database, and create an interactive Streamlit app for data analysis.

3. RedBus Interactive Filter : Components



Prepared
ALIYAS
MD113

By
THOMAS