Redbus Data Scraping with Selenium & Dynamic Filtering using Streamlit

Project Overview

This project involves scraping bus route data from the Redbus website using Selenium, storing the data in a SQL database, and developing an interactive Streamlit application for data filtering. This project aims to demonstrate web scraping techniques, data storage, and creating a user-friendly interface for data analysis and filtering.

Technologies Used

- **Python**: For scripting and automation.
- **Selenium**: For web scraping and extracting data from the Redbus website.
- **MySQL**: For storing the scraped data.
- **Streamlit**: For building an interactive web application.
- **Libraries**: Pandas, NumPy, MySQL Connector.

Dataset

The dataset used in this project is scraped from the Redbus website. It includes information about various bus routes, bus types, departure and arrival times, prices, ratings, and seat availability.

Database Schema

The data is stored in a MySQL database with the following schema:

- id: Auto-incrementing primary key
- route_name: TEXT
- route_link: TEXT
- busname: TEXT
- bustype: TEXT
- departing_time: DATETIME
- duration: TEXT
- reaching_time: DATETIME
- star_rating: FLOAT
- price: DECIMAL
- seats_available: INT

Implementation Steps:

1. Web Scraping with Selenium

- **Objective**: Automate the extraction of bus travel data from the Redbus website.
- **Tools Used**: Python, Selenium WebDriver, ChromeDriver
- **Process**:
    1. **Setup**: Install and configure Selenium and ChromeDriver.
    2. **Automation**: Write Python scripts to navigate the Redbus website, locate and extract data fields such as route name, bus type, departure time, duration, arrival time, star rating, price, and seats available.
    3. **Data Extraction**: Use Selenium to dynamically interact with web elements and extract the required data.

2. Data Storage

- **Objective**: Store the extracted data in a structured format for efficient retrieval and analysis.
- **Tools Used**: MySQL, SQLAlchemy
- **Process**:
    1. **Database Setup**: Create a MySQL database named redbus.
    2. **Schema Design**: Define the database schema with appropriate tables and columns.
    3. **Data Insertion**: Write scripts to insert the scraped data into the database.

3. Data Cleaning and Preprocessing

- **Objective**: Ensure the data is clean and ready for analysis.
- **Tools Used**: Python, Pandas
- **Process**:
    1. **Data Cleaning**: Remove any inconsistencies, handle missing values, and correct data types.
    2. **Preprocessing**: Prepare the data for analysis and visualization by normalizing and formatting it.

4. Streamlit Application Development

- **Objective**: Develop an interactive web application for data filtering and visualization.
- **Tools Used**: Streamlit, Python, Pandas
- **Process**:
    1. **Setup**: Install and configure Streamlit.
    2. **Interface Design**: Design a user-friendly interface with options for filtering data based on various parameters (route, bus type, departure time, price).
    3. **Visualization**: Create dynamic visualizations to display filtered data using Matplotlib and Seaborn.

Features

- **Real-time Data Extraction**: Scrapes live data from the Redbus website, ensuring up-to-date information.
- **Interactive Interface**: User-friendly Streamlit application for easy navigation and data filtering.
- **Dynamic Filtering**: Allows users to filter data based on multiple criteria, providing flexibility in data analysis.
- **Data Storage**: Efficiently stores and retrieves data from a MySQL database.

Key Results

- Successfully automated the extraction of bus travel data from the Redbus website.
- Structured and stored the data in a MySQL database for efficient retrieval and analysis.
- Developed an interactive Streamlit application that allows users to filter and visualize bus travel data in real-time.
- Provided valuable insights into travel patterns, helping users make informed travel choices.

Future Work

- Expand the application to include more filtering options and additional data fields.
- Implement real-time data updates to keep the information current.
- Enhance the visualizations with more detailed analysis and insights.

Conclusion

This project demonstrates the effective use of web scraping, data storage, and web application development to create a tool that provides valuable insights into bus route data. The combination of Selenium, MySQL, and Streamlit offers a powerful solution for dynamic data analysis.

Sample Outputs:

Image 1

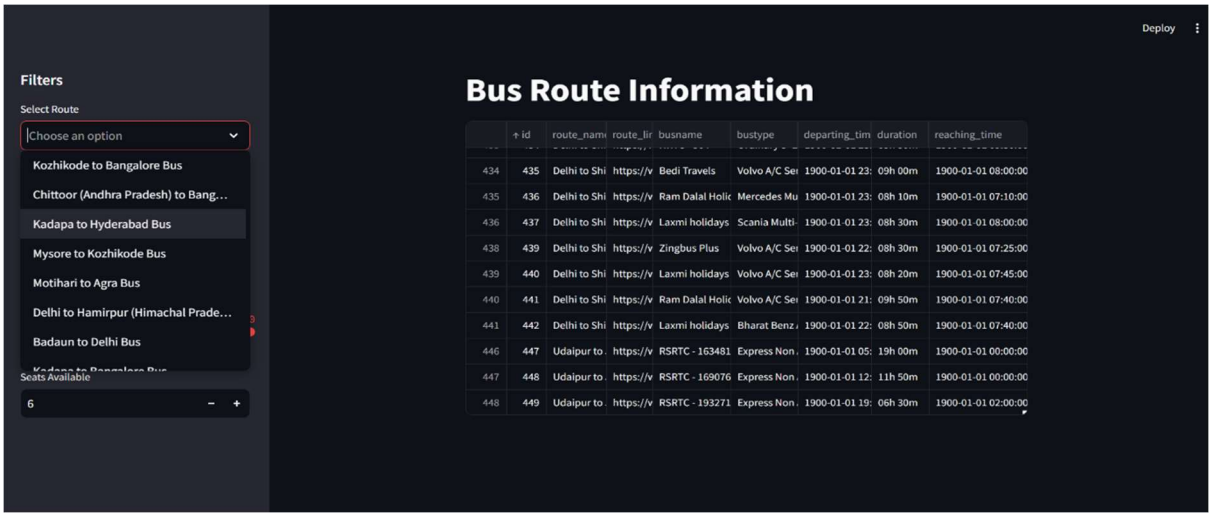

Image 2:

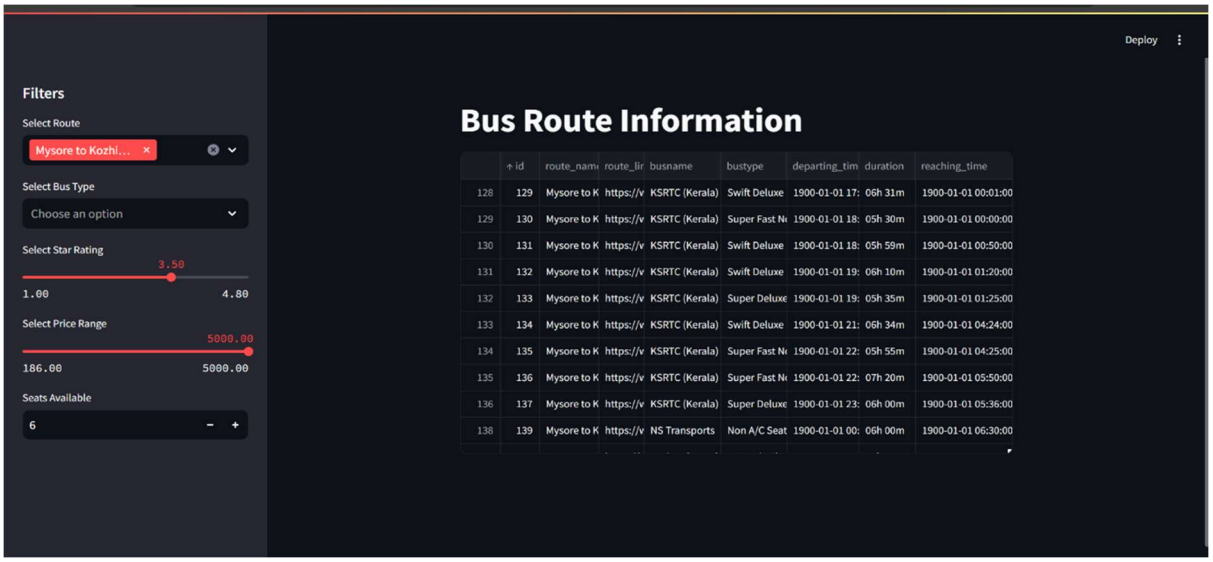

Image 3:

Image 4:

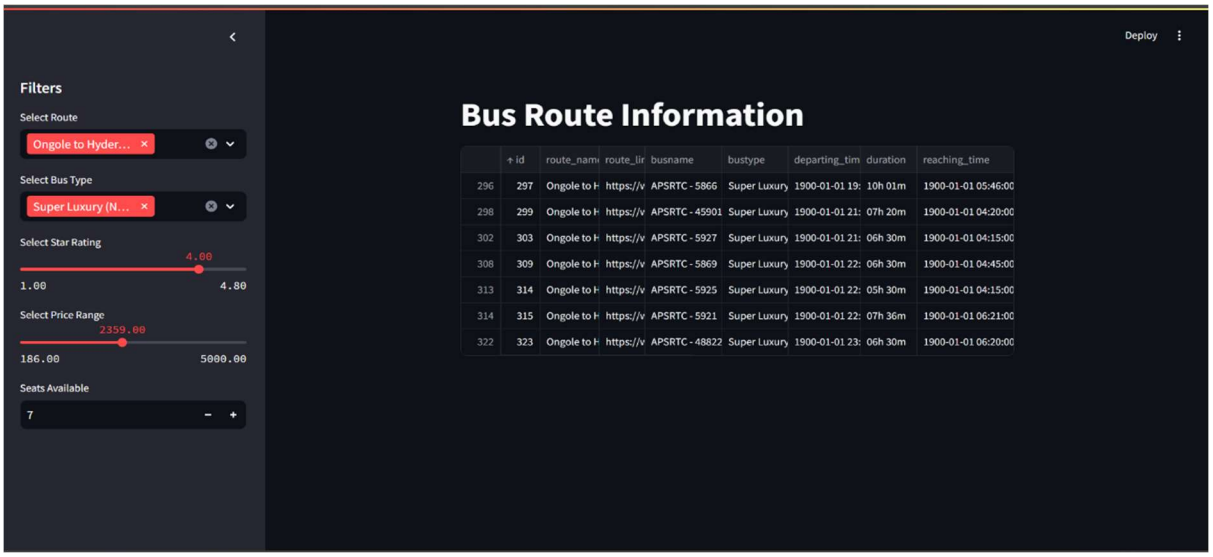

Image 5:

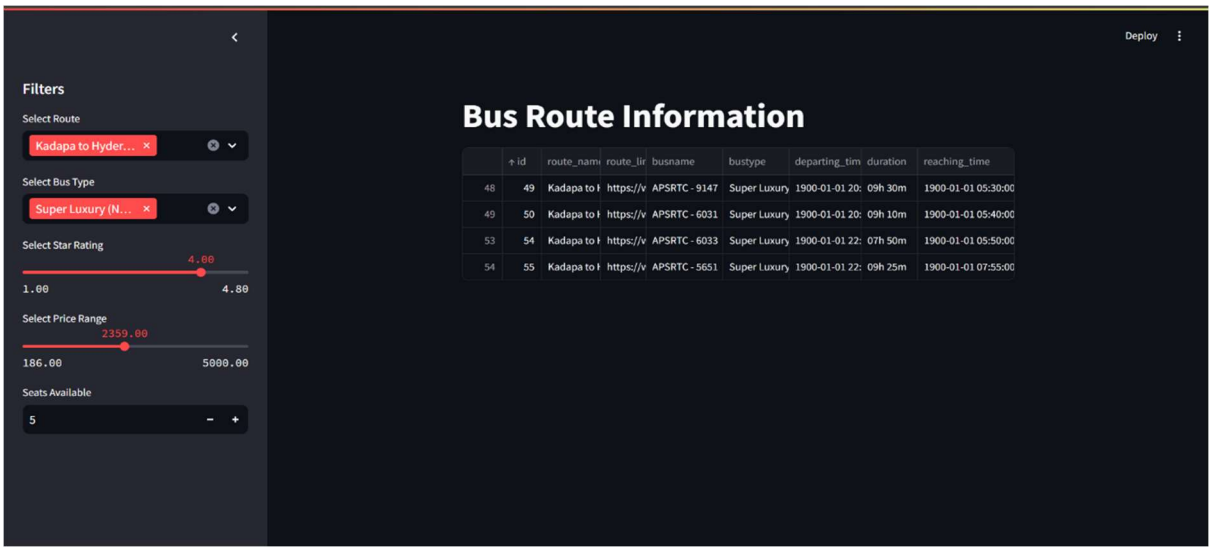

Image 6:

Image 7:


Filters
Select Route
Kadapa to Bang...
Select Bus Type
Super Luxury (N...
Select Star Rating
4.00
1.00
4.80
Select Price Range
2359.00
186.00
5000.00
Seats Available
6
Deploy
Bus Route Information
id route_nam route_lir busname bustype departing_tim duration reaching_time
169 170 Kadapa to E https://v APSRTC - 6207 Super Luxury 1900-01-01 22: 05h 56m 1900-01-01 04:26:00
171 172 Kadapa to E https://v APSRTC - 5227 Super Luxury 1900-01-01 22: 06h 45m 1900-01-01 05:30:00
172 173 Kadapa to E https://v APSRTC - 6065 Super Luxury 1900-01-01 22: 05h 50m 1900-01-01 04:35:00
176 177 Kadapa to E https://v APSRTC - 48483 Super Luxury 1900-01-01 23: 06h 45m 1900-01-01 06:00:00
178 179 Kadapa to E https://v APSRTC - 9254 Super Luxury 1900-01-01 23: 06h 00m 1900-01-01 05:30:00
179 180 Kadapa to E https://v APSRTC - 48516 Super Luxury 1900-01-01 23: 07h 00m 1900-01-01 06:30:00
182 183 Kadapa to E https://v APSRTC - 5977 Super Luxury 1900-01-01 23: 06h 40m 1900-01-01 06:30:00