**1. Source Code**

**a. Data Scraping Script (scrape\_bus\_data.py)**

This script scrapes bus data from the Redbus website and stores it in a MySQL database.

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.common.exceptions import TimeoutException, NoSuchElementException

import time

import mysql.connector

import re

# MySQL database connection parameters

db\_config = {

'user': 'your\_username',

'password': 'your\_password',

'host': 'localhost',

'database': 'your\_database'

}

# Function to insert data into MySQL database

def insert\_bus\_data(bus\_data):

try:

conn = mysql.connector.connect(\*\*db\_config)

cursor = conn.cursor()

insert\_query = """

INSERT INTO bus\_routes (route\_name, route\_link, busname, bustype, departing\_time, duration, reaching\_time, star\_rating, price, seats\_available)

VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)

"""

for data in bus\_data:

cursor.execute(insert\_query, data)

conn.commit()

print("Data inserted successfully.")

except mysql.connector.Error as err:

print(f"Error: {err}")

finally:

cursor.close()

conn.close()

# Initialize WebDriver

driver = webdriver.Chrome()

# Initialize bus\_data

bus\_data = []

try:

driver.get("https://www.redbus.in/")

wait = WebDriverWait(driver, 20)

src\_input = wait.until(EC.element\_to\_be\_clickable((By.ID, "src")))

src\_input.send\_keys("Bangalore")

dest\_input = wait.until(EC.element\_to\_be\_clickable((By.ID, "dest")))

dest\_input.send\_keys("Chennai")

date\_picker\_button = WebDriverWait(driver, 10).until(EC.element\_to\_be\_clickable((By.ID, 'onwardCal')))

date\_picker\_button.click()

search\_button = wait.until(EC.element\_to\_be\_clickable((By.ID, "search\_button")))

search\_button.click()

def scroll\_to\_bottom(driver):

driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")

i = 0

while i < 12:

try:

wait.until(EC.presence\_of\_element\_located((By.CLASS\_NAME, "bus-items")))

bus\_items = driver.find\_elements(By.XPATH, '//ul[@class="bus-items"]/div/li')

for bus\_item in bus\_items:

try:

bus\_name = bus\_item.find\_element(By.CLASS\_NAME, "travels").text

bus\_type = bus\_item.find\_element(By.CLASS\_NAME, "bus-type").text

departure = bus\_item.find\_element(By.CLASS\_NAME, "dp-time").text

arrival = bus\_item.find\_element(By.CLASS\_NAME, "bp-time").text

price\_text = bus\_item.find\_element(By.CLASS\_NAME, "fare").text

available\_seats = bus\_item.find\_element(By.CLASS\_NAME, "seat-left").text

rating\_text = bus\_item.find\_element(By.CLASS\_NAME, "rating").text

price = re.sub(r'[^\d.]', '', price\_text)

price = float(price) if price else 0.0

rating = re.sub(r'[^\d.]', '', rating\_text)

star\_rating = float(rating) if rating else 0.0

available\_seats = int(re.sub(r'[^\d]', '', available\_seats)) if available\_seats else 0

route\_link = "N/A"

duration = "N/A"

departing\_time = departure

reaching\_time = arrival

bus\_data.append((route\_link, route\_link, bus\_name, bus\_type, departing\_time, duration, reaching\_time, star\_rating, price, available\_seats))

if len(bus\_data) > 20:

break

except NoSuchElementException as e:

print(f"Error finding element in bus item: {e}")

scroll\_to\_bottom(driver)

time.sleep(1)

i += 1

except Exception as e:

print(f"An error occurred: {e}")

finally:

driver.quit()

print("WebDriver closed.")

if bus\_data:

insert\_bus\_data(bus\_data)

else:

print("No bus data was scraped.")

**b. Streamlit Application (app.py)**

This script creates a Streamlit application for data filtering and visualization.

python

import streamlit as st

import mysql.connector

import pandas as pd

# Connect to MySQL

conn = mysql.connector.connect(

host="localhost",

user="your\_username",

password="your\_password",

database="your\_database"

)

# Query data from MySQL

def get\_data(query):

cursor = conn.cursor()

cursor.execute(query)

rows = cursor.fetchall()

columns = [i[0] for i in cursor.description]

cursor.close()

return pd.DataFrame(rows, columns=columns)

# Filter options

st.title("Redbus Schedule")

bustype = st.selectbox("Select Bus Type", ["All", "AC", "Non-AC"])

price\_range = st.slider("Price Range", 0, 5000, (100, 3000))

route = st.text\_input("Enter Route")

query = "SELECT \* FROM bus\_routes WHERE 1=1"

if bustype != "All":

query += f" AND bustype = '{bustype}'"

if route:

query += f" AND (busname LIKE '%{route}%' OR bustype LIKE '%{route}%')"

query += f" AND price BETWEEN {price\_range[0]} AND {price\_range[1]}"

df = get\_data(query)

st.dataframe(df)

conn.close()

**2. Documentation**

**a. Overview**

* **Data Scraping**: The scrape\_bus\_data.py script uses Selenium to scrape bus data from the Redbus website. It extracts details like bus name, type, departure and arrival times, price, rating, and available seats, and then stores this data in a MySQL database.
* **Streamlit Application**: The app.py script creates a web application using Streamlit. It allows users to filter bus data based on bus type, price range, and route, and displays the filtered results in a table.

**b. Running the Scripts**

1. **Data Scraping**:
   * Install necessary libraries: selenium, mysql-connector-python, and pandas.
   * Download and set up ChromeDriver.
   * Update db\_config in scrape\_bus\_data.py with your MySQL credentials.
   * Run the script:

bash

python scrape\_bus\_data.py

1. **Streamlit Application**:
   * Install Streamlit and required libraries: streamlit, mysql-connector-python, and pandas.
   * Update MySQL credentials in app.py.
   * Run the Streamlit app:

bash

streamlit run app.py

**c. Data Collection**

* **Data Source**: Redbus website.
* **Data Collected**: Bus name, type, departure time, arrival time, price, available seats, and star rating.

**3. Database Schema**

**a. SQL Script to Create the Database and Table**

sql

-- Create database

CREATE DATABASE redbus\_db;

-- Use database

USE DATABASE redbus\_db;

-- Use the database

USE redbus\_db;

-- Create table

CREATE TABLE bus\_routes (

id INT AUTO\_INCREMENT PRIMARY KEY,

route\_name TEXT,

route\_link TEXT,

busname TEXT,

bustype TEXT,

departing\_time TIME,

duration TEXT,

reaching\_time TIME,

star\_rating FLOAT,

price DECIMAL(10, 2),

seats\_available INT

);

**b. SQL Script to Populate the Database**

This step is not applicable directly as data is inserted via the scraping script. However, if needed, you could insert sample data manually:

sql

INSERT INTO bus\_routes (route\_name, route\_link, busname, bustype, departing\_time, duration, reaching\_time, star\_rating, price, seats\_available)

VALUES

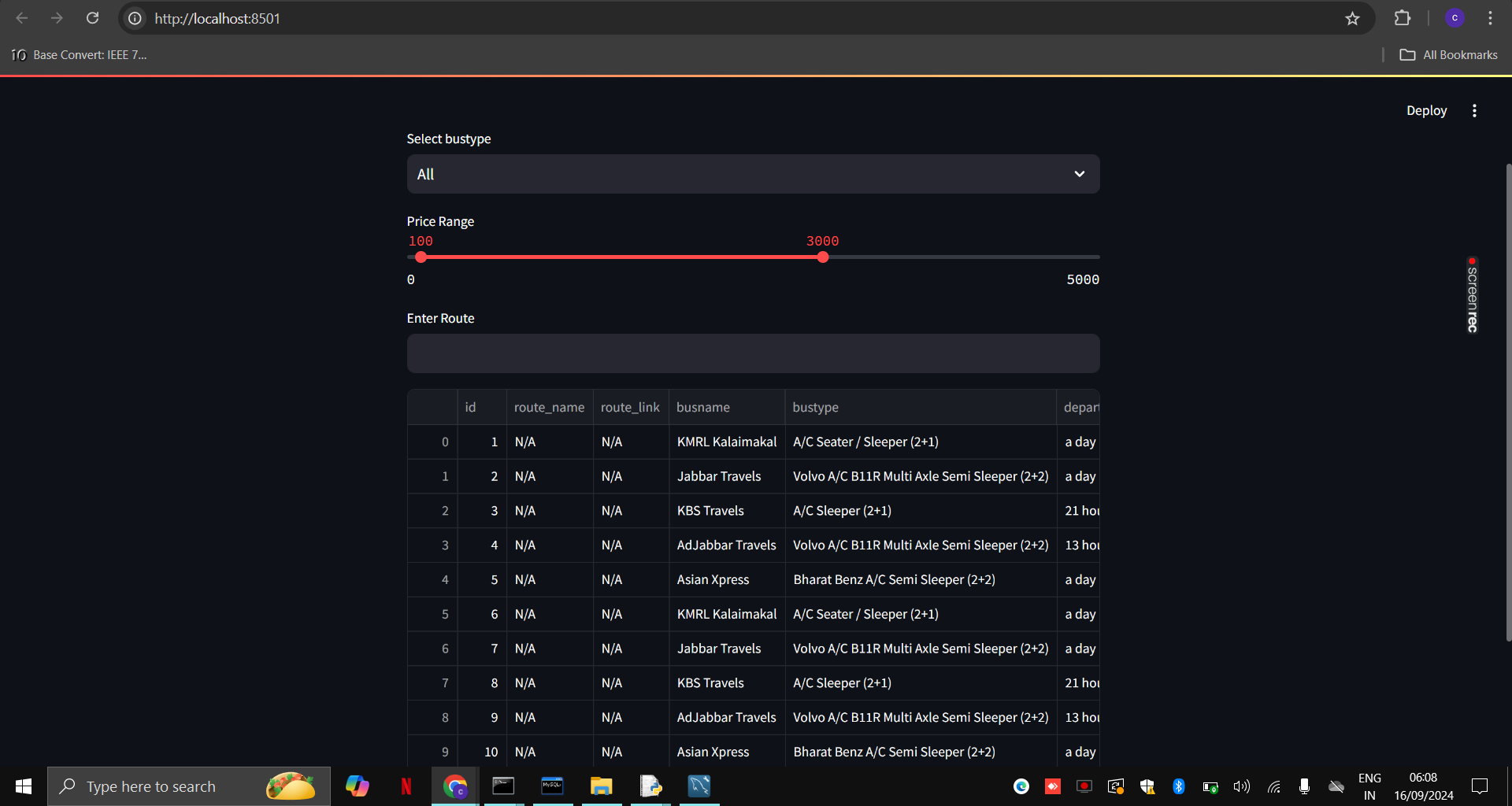
('Route 1', 'link1', 'Bus A', 'AC', '10:00:00', '5h', '15:00:00', 4.5, 1500.00, 20),

('Route 2', 'link2', 'Bus B', 'Non-AC', '12:00:00', '4h', '16:00:00', 3.5, 800.00, 30);

**4. Application Using Streamlit**

**a. Screenshots**

* **Screenshot 1**: Streamlit Application Home Page



* **Screenshot 2**: Filtered Results (e.g., filtering by AC buses and a price range)

