Step 5: Streamlit Dashboard Development

This step involved designing a fully functional **Streamlit dashboard** to visualize insights derived from the SQL database (`Asteroid_Data.db`).

🚀Technologies Used:

- **Streamlit**: For building the web UI.
- **SQLite3**: Backend database integration.
- **Pandas**: Data handling and visualization.

Setup & Initialization

- Installed required libraries:

```
pip install streamlit streamlit_option_menu
```

- Created main dashboard script using `%%writefile nasa_project.py`

- Configured wide-layout Streamlit UI:

```
st.set_page_config(layout='wide')
```

🏫Query-Based Exploration

- Sidebar navigation created using `st.sidebar.selectbox()` to select from **20 pre-written SQL queries**:

- 15 mandatory queries (approach counts, velocity, hazard checks, etc.)

- 5 bonus queries (deep insights like top 5 closest, orbiting bodies, etc.)

- Each query returns a filtered view of asteroid data from `asteroids` and `close_approach` tables.

- Results displayed using `st.dataframe()` for interactive scrolling, sorting, and search.

Advanced Filtering UI

A second section of the dashboard allowed users to:

- Filter by **date**, **velocity**, **distance (AU, LD, KM)**
- Set **min/max estimated diameter** range
- Choose hazard status: Both / Yes / No

These filters dynamically generate an SQL query which is executed and shown in a live table.

Example filter snippet:

```
selected_date = st.date_input("Select Close Approach Date", datetime(2000, 1,
1))
min_velocity = st.slider("Min Velocity", 0.0, 100000.0, 0.0)
...
```

The final query uses all slider inputs to filter and return only the relevant asteroid approaches.

🚚Launching in Google Colab (Streamlit + LocalTunnel)

Instructions to launch the app inside Google Colab:

```
!wget -q -O - ipv4.icanhazip.com
!streamlit run nasa_project.py & npx localtunnel --port 8501
```

This produces a secure and shareable **Streamlit web app link**, making it accessible without local setup.

Final Outcome: Interactive NEO Tracking Portal

- Users can run **predefined queries** or apply **custom filters**.
- Dashboard enables **visual SQL-based exploration** of asteroid approach patterns.
- Clean and minimal design using wide layout for better readability.
- Successfully deployed through **Google Colab + LocalTunnel**.

🚢Project Conclusion

This NASA NEO Tracking Project is a complete data pipeline that:

1. Extracted raw data from NASA's API
2. Cleaned and structured it into an SQL database

3. Performed advanced query-based analysis
4. Delivered insights through a powerful Streamlit dashboard

It provides a solid foundation for:

- Real-time space object tracking
- Hazard monitoring
- Educational or scientific dashboard development

Future extensions may include:

- Real-time data updates
- Asteroid visualizations via plots or maps
- Alert system for hazardous NEOs

✅**End of Project Summary**