**Reflection note on Jupyter Notebook Work**

**1. Objective**

The goal of this Jupyter notebook work was to set up a tracking database for the ZenSpace project (Phase 3) and then analyze user behavior through SQL queries and visualizations.

This helps answer key product questions like:

How often users engage daily (DAU)

Which features are most used (meditation, breathing, sound)

What content is most popular (meditation categories)

How effective reminders are (open rate)

Who the most consistent users are (streak leaders)

**2. What I Did**

Database Setup (Notebook A – ZenSpace\_DB\_Setup.ipynb):

Designed 3 tables: sessions, preferences, reminders.

Inserted dummy data (users with meditation, breathing, sound usage; preferences; reminders).

Ensured database integrity with primary keys and unique constraints.

Learned to handle IntegrityError by clearing tables or using INSERT OR REPLACE.

Analysis & Visualization (Notebook B – ZenSpace\_Analysis.ipynb):

Connected to the SQLite file (zenspace.db).

Created helper function run\_query() to run SQL and return Pandas DataFrames.

**Ran SQL queries for:**

Daily Active Users

Average session duration by feature

Popular meditation categories

Reminder effectiveness (open vs not opened)

Top active users (streak leaders)

Built charts with matplotlib/seaborn to make the results visual and easier to interpret.

**3. What I Learned**

Database design basics: defining tables, primary keys, avoiding duplicate entries.

SQL practice: writing SELECT, GROUP BY, and aggregate queries to extract insights.

Error handling: solved IntegrityError by using DELETE FROM before inserts or INSERT OR REPLACE.

Visualization skills: used bar charts and pie charts to show data clearly.

Workflow discipline: separated setup and analysis into two notebooks for cleaner structure.

**4. Challenges**

Accidentally re-inserting duplicate preference data caused errors. Solved it with table cleanup.

Making sure the analysis notebook used the correct DB file (zenspace.db).

Ensuring charts displayed properly inside Jupyter (needed plt.show() after each plot).