# Task Explanation Document

## Task 1:

Kindly check out the notebook. Everything is documented there for this task step by step.

## Task 2: Database Organization

1. **Organization**
    a. **Users** Table:
        i. user_id: Primary key providing a unique identifier for each user.
        ii. user_name: Stores the name of each user.
        iii. user_email: Stores the email address of each user.
    b. **Posts** Table:
        i. post_id: Primary key, providing a unique identifier for each post.
        ii. user_id: Foreign key referencing the user_id in the users table, establishing a relationship between a post and its creator.
        iii. post_title: Stores the title of each post.
        iv. post_text: Stores the actual content of each post.

2. **Structure**
    a. This database structure follows a relational model, suitable for applications with structured and interrelated data.
    b. The use of primary keys (user_id and post_id) ensures uniqueness and data integrity.
    c. Relationships (posts relationship in users and user_id foreign key in posts) allow for efficient querying and retrieval of related data.

3. **Reasoning for structure**
    a. Have kept it as simple as possible while having basic relationship constraints.
    b. Follows best practices in database design, including the use of primary keys and foreign keys to establish relationships.
    c. The one-to-many relationship between users and posts allows for flexibility in handling multiple posts per user.

## Task 3: Code Organization & Data Processing

1. **Code Organization**
    a. To keep things simple, the code has two primary functions:
        i. Read from a Google Sheets file
        ii. Add a new row to the file

b. The main function contains initialization of gspread for opening a Google Sheets file and a user interaction loop.
c. read_sheet: Retrieves data from a Google Sheet and converts it into a DataFrame.
d. add_row_to_sheet: Appends a new row of values to the Google Sheet.
e. print_option_screen: Prints a user menu for interaction.
f. prompt_for_values: Prompts the user for values for each column.

**2. Data Processing Approaches**

a. Reading Data: Utilizes gspread to fetch data from the Google Sheet, then converts it into a DataFrame using Pandas.
b. Adding a New Row: Uses gspread to append a new row of values to the Google Sheet. This method is specific to the Google Sheets API.
c. Handling User Input: Utilizes a loop to continuously prompt the user for their choice. Processes the choice accordingly, enabling multiple actions without restarting the script.
d. Data Validation: Ensures that the user provides an integer for the 'id' column, preventing input errors.
e. Error Handling: If the user attempts to input a non-integer value for the 'id' column, it displays an error message and continues to prompt for valid input.