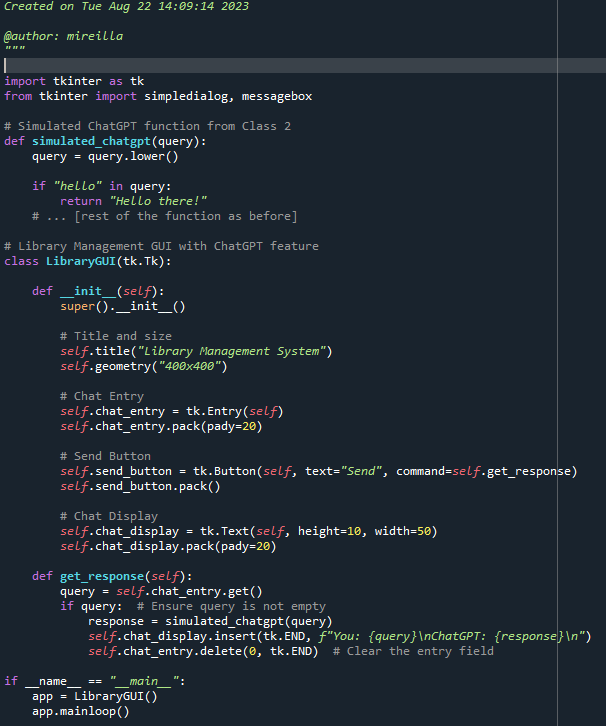
# Integrating Simulated ChatGPT with a GUI (tkinter)

## **Modifying the Library Management System to incorporate the simulated ChatGPT**

Let’s assume the Library Management System GUI primarily consists of features like borrowing and returning books. We’ll now incorporate a chat feature where users can ask questions and get responses from our simulated ChatGPT.

Here's a basic example:



**Exercise 1**: **Problem**: Add a "Clear Chat" button to the Library Management GUI that clears the chat display when clicked.

**Exercise 2**: **Problem**: Currently, the simulated ChatGPT only responds with static answers. Modify the function to provide a list of available books in the library when the user asks, "What books are available?"

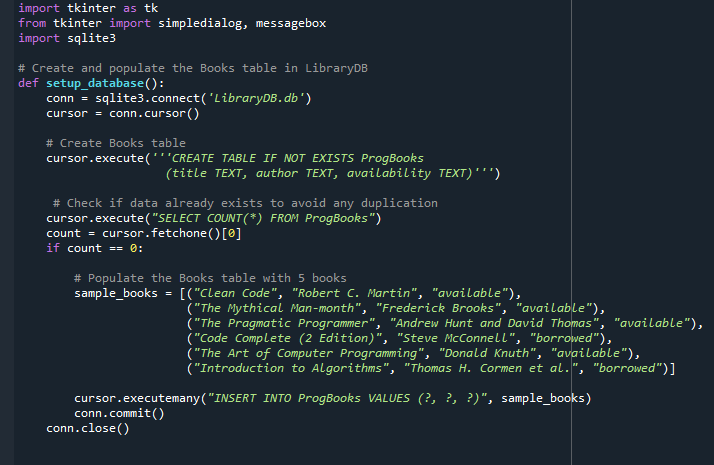
# Fetching Data from 'LibraryDB' and incorporating it in ChatGPT

## **Fetching data from 'LibraryDB' based on user queries**

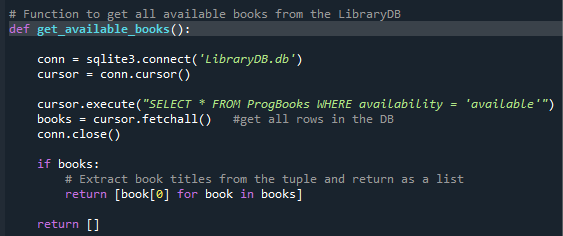
We are updating the code from the previous task.

Let’s assume we have your SQLite database named **LibraryDB** with a table **ProgBooks** that contains the columns ‘**title’**, ‘**author’**, and ‘**availability’**. We’ll integrate this with our simulated ChatGPT function.

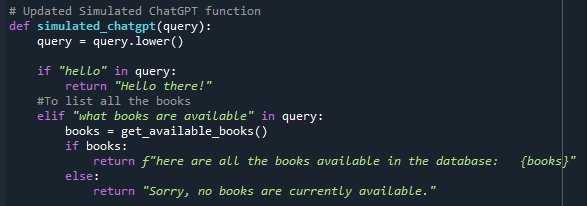
1. Ensure that the ProgBooks table exists and is populated with 6 book entries:



1. Write a function to extract all books that are “available”.



1. Update Simulated ChatGPT function:



**Exercise 1**: **Problem**: Enhance the simulated ChatGPT to respond with the author of a book when asked, "Who wrote [book\_name]?"

**Exercise 2**: **Problem**: Add functionality to the simulated ChatGPT to return the number of available books when asked, "How many books are available?"