Technical Document: Modernized Car Database Management System

1.) Overview:

This document outlines the design and implementation of a Simple Car Database Management System using tools like Python, and tkinter with tkk a more polished graphical user interface (GUI).

2.) System Components:

- **Database:** SQLite is used to manage car records (make, model, and year).
- **GUI:** The graphical interface is built using tikinter and enhanced with ttk to give it a modern look and feel.
- **Key Features:**
  - Add and delete car records
  - Display car records in a table format using tt.Treeview.
  - Filtering and sorting of data.
  - Responsive layout with enhanced padding, themes, and custom fonts.

3.) Technology Stack:

Programming Language: Python
Database: SQLite
GUI Framework: tkinter with ttk for styling
Optional Libraries: PIL for image Handling (Icons)

4.) Design Features:

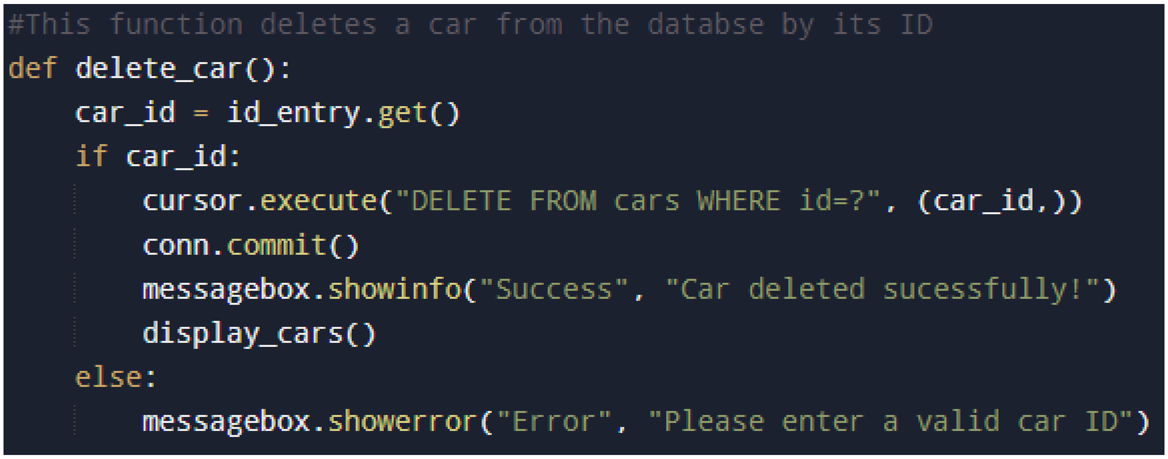
- **Themed Widget:** ttk widgets replace standard tkinter widgets for a modern interface.
- **Treeview for Data Display:** ttk.Treeview is used to display car records in a table format, supporting sort and column customization.
- **Responsive Layout:** The GUI layout is configured with grids for better alignment and resizing.
- **Status Bar:** Displays real-time feedback to users.

5.) Key Functions:

Add Car: inserts new car records into the database.
Delete Car: removes car records based on ID.
Display Cars: Fetches and displays all cars records in the Treeview.
Filter Cars: Filters displayed cars based on user input.

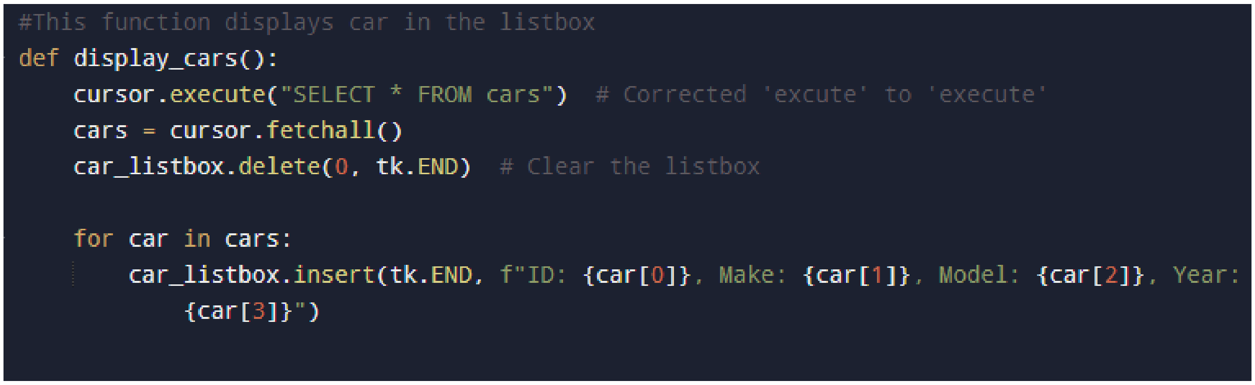
6.) Code Snippets

- **Function to Delete a Car:**



```
#This function deletes a car from the databse by its ID
def delete_car():
    car_id = id_entry.get()
    if car_id:
        cursor.execute("DELETE FROM cars WHERE id=?", (car_id,))
        conn.commit()
        messagebox.showinfo("Success", "Car deleted sucessfully!")
        display_cars()
    else:
        messagebox.showerror("Error", "Please enter a valid car ID")
```

- **Function to Display Cars:**



```
#This function displays car in the listbox
def display_cars():
    cursor.execute("SELECT * FROM cars")  # Corrected 'excute' to 'execute'
    cars = cursor.fetchall()
    car_listbox.delete(0, tk.END)  # Clear the listbox

    for car in cars:
        car_listbox.insert(tk.END, f"ID: {car[0]}, Make: {car[1]}, Model: {car[2]}, Year:
            {car[3]}")
```