PARKING MANAGEMENT SYSTEM DOCUMENTATION

Introduction

This project is a simple Parking Management System implemented in C. The system allows users to record and manage entries for different types of vehicles, including two-wheelers, three-wheelers, four-wheelers, and six-wheelers. Users can perform operations such as adding new entries, displaying the current status, deleting entries by providing vehicle numbers, or resetting the entire list. The data is saved to and loaded from a file (`data.txt`) to persist information across program executions.

Features

1. **Add Entries:**
   - Users can add entries for each type of vehicle, providing vehicle numbers.
   - Two-Wheelers: ₹50 each
   - Three-Wheelers: ₹100 each
   - Four-Wheelers: ₹150 each
   - Six-Wheelers: ₹200 each
2. **Display Status:**
   - Shows the number of each type of vehicle along with their respective vehicle numbers.
   - Displays the total number of vehicles.
   - Shows the total gain amount.
3. **Delete Entries:**
   - Users can delete entries by providing vehicle numbers or reset the entire list.
   - Deletion affects the count and total gain amount.
4. **Data Persistence:**
   - Data is saved to a file (`data.txt`) before program exit.
   - Data is loaded from the file at the beginning of each program run.

Functions

- `main()`:
  - Entry point of the program.
  - Calls `LoadData()` to load existing data.

    - Enters an infinite loop to continuously present the user with a menu.
- `Menu()`:
    - Displays the main menu and prompts the user for a choice.
    - Returns the user's choice.
- `ShowDetails()`:
    - Displays the current status, including the number of each type of vehicle, their respective vehicle numbers, the total number of vehicles, and the total gain amount.
- `DeleteMenu()`:
    - Displays the delete menu and prompts the user for a choice.
    - Returns the user's choice.
- `Delete()`:
    - Performs deletion based on the user's choice from the delete menu.
    - Users can delete entries by providing vehicle numbers or reset the entire list.
    - Resets counts and amount for the deleted entry or the entire list.
    - Calls `SaveData()` to save data after deletion.
- `two_wheeler()`, `three_wheeler()`, `four_wheeler()`, `six_wheeler()`:
    - Functions to add entries for each type of vehicle, prompting users to input vehicle numbers.
    - Increment counts and update the total gain amount.
    - Calls `SaveData()` to save data after the entry operation.
- `SaveData()`:
    - Saves current data (counts, vehicle numbers, and amount) to the file (`data.txt`).
- `LoadData()`:
    - Loads data (counts, vehicle numbers, and amount) from the file (`data.txt`) at the beginning of the program.

File Handling

- Data is saved to and loaded from a file named `data.txt`.
- The file contains space-separated values representing counts of each type of vehicle, their respective vehicle numbers, the total amount, and the total count.

Usage

1. Clone the repository from GitHub.
2. Compile the C code using a suitable compiler (`gcc` recommended).
3. Run the compiled executable file.
4. Follow the on-screen menu to interact with the Parking Management System.

Conclusion

This simple Parking Management System provides a basic framework for managing entries of different types of vehicles. Users can add new entries, view the current status, delete entries by providing vehicle numbers or reset the entire list. The data is saved to a file for persistence across program runs.

-