Roll NO:27

NAME : Mali Anjali Prakash

DIV: FYMCA-A

Assignment no -3

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Car {

    char modelName[50];

    char manufacturer[50];

    float price;

    float engineCapacity;

    struct Car\* next;

    struct Car\* prev;

};

typedef struct Car Car;

Car\* head = NULL;

void insertCar();

void deleteCar();

void updatePrice();

void listCarsInPriceRange();

void listCarDetails();

void displayCars();

Car\* createCarNode();

int main() {

    int choice;

    while (1) {

        printf("\nMenu:\n");

        printf("1. Insert a new car model\n");

        printf("2. Delete a car model\n");

        printf("3. Update the price of a car model\n");

        printf("4. List car models within a price range\n");

        printf("5. List details of a car model\n");

        printf("6. Display all cars\n");

        printf("7. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                insertCar();

                break;

            case 2:

                deleteCar();

                break;

            case 3:

                updatePrice();

                break;

            case 4:

                listCarsInPriceRange();

                break;

            case 5:

                listCarDetails();

                break;

            case 6:

                displayCars();

                break;

            case 7:

                exit(0);

            default:

                printf("Invalid choice, please try again.\n");

        }

    }

    return 0;

}

Car\* createCarNode() {

    Car\* newCar = (Car\*)malloc(sizeof(Car));

    printf("Enter model name: ");

    getchar();  // to consume newline left by previous input

    fgets(newCar->modelName, sizeof(newCar->modelName), stdin);

    newCar->modelName[strcspn(newCar->modelName, "\n")] = 0;  // removing newline character

    printf("Enter manufacturer name: ");

    fgets(newCar->manufacturer, sizeof(newCar->manufacturer), stdin);

    newCar->manufacturer[strcspn(newCar->manufacturer, "\n")] = 0;

    printf("Enter price: ");

    scanf("%f", &newCar->price);

    printf("Enter engine capacity: ");

    scanf("%f", &newCar->engineCapacity);

    newCar->next = NULL;

    newCar->prev = NULL;

    return newCar;

}

void insertCar() {

    Car\* newCar = createCarNode();

    if (head == NULL) {

        head = newCar;

    } else {

        Car\* temp = head;

        while (temp->next != NULL) {

            temp = temp->next;

        }

        temp->next = newCar;

        newCar->prev = temp;

    }

    printf("Car model added successfully.\n");

}

void deleteCar() {

    char modelName[50];

    printf("Enter the model name of the car to delete: ");

    getchar();  // to consume newline left by previous input

    fgets(modelName, sizeof(modelName), stdin);

    modelName[strcspn(modelName, "\n")] = 0;

    Car\* temp = head;

    while (temp != NULL) {

        if (strcmp(temp->modelName, modelName) == 0) {

            if (temp->prev != NULL) {

                temp->prev->next = temp->next;

            } else {

                head = temp->next;

            }

            if (temp->next != NULL) {

                temp->next->prev = temp->prev;

            }

            free(temp);

            printf("Car model deleted successfully.\n");

            return;

        }

        temp = temp->next;

    }

    printf("Car model not found.\n");

}

void updatePrice() {

    char modelName[50];

    printf("Enter the model name of the car to update price: ");

    getchar();  // to consume newline left by previous input

    fgets(modelName, sizeof(modelName), stdin);

    modelName[strcspn(modelName, "\n")] = 0;

    Car\* temp = head;

    while (temp != NULL) {

        if (strcmp(temp->modelName, modelName) == 0) {

            printf("Enter new price for %s: ", temp->modelName);

            scanf("%f", &temp->price);

            printf("Price updated successfully.\n");

            return;

        }

        temp = temp->next;

    }

    printf("Car model not found.\n");

}

void listCarsInPriceRange() {

    float minPrice, maxPrice;

    printf("Enter the minimum price: ");

    scanf("%f", &minPrice);

    printf("Enter the maximum price: ");

    scanf("%f", &maxPrice);

    Car\* temp = head;

    int found = 0;

    while (temp != NULL) {

        if (temp->price >= minPrice && temp->price <= maxPrice) {

            printf("\nModel: %s\n", temp->modelName);

            printf("Manufacturer: %s\n", temp->manufacturer);

            printf("Price: %.2f\n", temp->price);

            printf("Engine Capacity: %.2f\n", temp->engineCapacity);

            found = 1;

        }

        temp = temp->next;

    }

    if (!found) {

        printf("No cars found in the given price range.\n");

    }

}

void listCarDetails() {

    char modelName[50];

    printf("Enter the model name of the car to view details: ");

    getchar();  // to consume newline left by previous input

    fgets(modelName, sizeof(modelName), stdin);

    modelName[strcspn(modelName, "\n")] = 0;

    Car\* temp = head;

    while (temp != NULL) {

        if (strcmp(temp->modelName, modelName) == 0) {

            printf("\nModel: %s\n", temp->modelName);

            printf("Manufacturer: %s\n", temp->manufacturer);

            printf("Price: %.2f\n", temp->price);

            printf("Engine Capacity: %.2f\n", temp->engineCapacity);

            return;

        }

        temp = temp->next;

    }

    printf("Car model not found.\n");

}

void displayCars() {

    if (head == NULL) {

        printf("No cars to display.\n");

        return;

    }

    Car\* temp = head;

    while (temp != NULL) {

        printf("\nModel: %s\n", temp->modelName);

        printf("Manufacturer: %s\n", temp->manufacturer);

        printf("Price: %.2f\n", temp->price);

        printf("Engine Capacity: %.2f\n", temp->engineCapacity);

        temp = temp->next;

    }

}









