****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Assignment NO – 03**

**Course Title: DS Section: D2**

**Submission Date : 11/11/2024**

**Course Teacher’s Name: Prof. Dr. Saiful Azad**

**Name: Rukonuzzaman Topu**

**ID : 232002280**

**Code:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct PropertySale {

int uid;

char address[50];

int zip;

int size;

int year;

int price;

struct PropertySale \*next;

};

struct SalesDatabase {

struct PropertySale \*head;

};

void printMainMenu();

void Sales(struct SalesDatabase \*db);

void Erase(struct SalesDatabase \*db);

void Search(struct SalesDatabase \*db);

void PrintDB(struct SalesDatabase \*db);

void GetZIP(struct SalesDatabase \*db);

void GetPrice(struct SalesDatabase \*db);

void AveragePrice(struct SalesDatabase \*db);

void SaleCount(struct SalesDatabase \*db);

void freeDatabase(struct SalesDatabase \*db);

struct PropertySale\* createNode();

int main() {

struct SalesDatabase db;

db.head = NULL;

int choice;

int pass;

printf("\nPASSWORD: ");

scanf("%d", &pass);

while(pass != 280) {

printf("Invalid Password!\n");

printf("\nPASSWORD: ");

scanf("%d", &pass);

}

printf("Login Successful!\n");

// Main menu loop

while(1){

printMainMenu();

printf("\n\nENTER YOUR CHOICE: ");

if (scanf("%d", &choice) != 1) {

printf("Invalid input. Please enter a number.\n");

while (getchar() != '\n');

continue;

}

printf("\n");

if (choice == 1) {

Sales(&db);

} else if (choice == 2) {

Erase(&db);

} else if (choice == 3) {

Search(&db);

} else if (choice == 4) {

PrintDB(&db);

} else if (choice == 5) {

GetZIP(&db);

} else if (choice == 6) {

GetPrice(&db);

} else if (choice == 7) {

AveragePrice(&db);

} else if (choice == 8 ) {

SaleCount(&db);

} else if (choice == 9) {

printf("Log Out Successful\n");

exit(0);

} else {

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

}

}

return 0;

}

// Main menu function

void printMainMenu(void) {

printf("\nMain Menu:\n");

printf("1. Insert new flat sale information\n");

printf("2. Delete a property from the system\n");

printf("3. Find a property from the system\n");

printf("4. See all properties\n");

printf("5. Get zip code\n");

printf("6. Get Price\n");

printf("7. Average prices of sales property\n");

printf("8. Total sales\n");

printf("9. Log Out\n");

printf("..........");

}

// Create a new node for a sale entry

struct PropertySale\* createNode() {

struct PropertySale\* newNode = (struct PropertySale\*)malloc(sizeof(struct PropertySale));

newNode->next = NULL;

return newNode;

}

// Function to insert a new sale info into the database

void Sales(struct SalesDatabase \*db) {

struct PropertySale \*newSale = createNode();

printf("Enter UID: ");

if (scanf("%d", &newSale->uid) != 1) {

printf("Invalid input for UID.\n");

free(newSale);

return;

}

printf("Enter Address: ");

scanf(" %[^\n]s", newSale->address);

printf("Enter ZIP code: ");

if (scanf("%d", &newSale->zip) != 1) {

printf("Invalid input for ZIP code.\n");

free(newSale);

return;

}

printf("Enter size: ");

if (scanf("%d", &newSale->size) != 1) {

printf("Invalid input for size.\n");

free(newSale);

return;

}

printf("Enter construction year: ");

if (scanf("%d", &newSale->year) != 1) {

printf("Invalid input for year.\n");

free(newSale);

return;

}

printf("Enter price: ");

if (scanf("%d", &newSale->price) != 1) {

printf("Invalid input for price.\n");

free(newSale);

return;

}

newSale->next = db->head;

db->head = newSale;

printf("Sale added successfully.\n");

}

// Function to delete a sold property info from the database

void Erase(struct SalesDatabase \*db) {

int uid;

printf("Enter the UID to delete: ");

if (scanf("%d", &uid) != 1) {

printf("Invalid input for UID.\n");

return;

}

struct PropertySale \*current;

struct PropertySale \*temp = db->head;

if(temp == NULL){

printf("No Sale to delete.\n");

return;

}

else if(db->head->uid == uid) {

current = db->head;

db->head = current->next;

free(current);

printf("Sale with UID %d deleted successfully.\n", uid);

return;

}

while (temp->next != NULL) {

if (temp->next->uid == uid) {

current = temp->next;

temp->next = current->next;

free(current);

printf("Sale with UID %d deleted successfully.\n", uid);

return;

}

temp = temp->next;

}

printf("Sale with UID %d not found.\n", uid);

}

// Function to search for a sold property in the database

void Search(struct SalesDatabase \*db) {

int uid;

printf("Enter the UID to search for: ");

if (scanf("%d", &uid) != 1) {

printf("Invalid input for UID.\n");

return;

}

struct PropertySale \*temp = db->head;

while (temp != NULL) {

if (temp->uid == uid) {

printf("Sale found: UID=%d, Address=%s, ZIP=%d, Size=%d, Year=%d, Price=%d\n",

temp->uid, temp->address, temp->zip,

temp->size, temp->year, temp->price);

return;

}

temp = temp->next;

}

printf("Sale with UID %d not found.\n", uid);

}

// Function to print the sales database

void PrintDB(struct SalesDatabase \*db) {

if (db->head == NULL) {

printf("No sales available.\n");

return;

}

struct PropertySale \*temp = db->head;

while (temp != NULL) {

printf("UID: %d, Address: %s, ZIP: %d, Size: %d, Year: %d, Price: %d\n",

temp->uid, temp->address, temp->zip,

temp->size, temp->year, temp->price);

temp = temp->next;

}

}

// Function to get zip code for a property

void GetZIP(struct SalesDatabase \*db) {

int uid;

printf("Enter the UID: ");

if (scanf("%d", &uid) != 1) {

printf("Invalid input for UID.\n");

return;

}

struct PropertySale \*temp = db->head;

while (temp != NULL) {

if (temp->uid == uid) {

printf("Zip Code: %d\n", temp->zip);

return;

}

temp = temp->next;

}

printf("Zip code with UID %d not found.\n", uid);

}

// Function to get price of a property

void GetPrice(struct SalesDatabase \*db) {

int uid;

printf("Enter the UID: ");

if (scanf("%d", &uid) != 1) {

printf("Invalid input for UID.\n");

return;

}

struct PropertySale \*temp = db->head;

while (temp != NULL) {

if (temp->uid == uid) {

printf("Price: %d\n", temp->price);

return;

}

temp = temp->next;

}

printf("Price with UID %d not found.\n", uid);

}

// Function to calculate average price of sales

void AveragePrice(struct SalesDatabase \*db) {

if (db->head == NULL) {

printf("Average price of sales: 0.00\n");

return;

}

int count = 0, totalPrice = 0;

struct PropertySale \*temp = db->head;

while (temp != NULL) {

totalPrice += temp->price;

count++;

temp = temp->next;

}

float avg = (float)totalPrice / count;

printf("Average price of sales: %.2f\n", avg);

}

// Function to count total sales in the database

void SaleCount(struct SalesDatabase \*db) {

int count = 0;

struct PropertySale \*temp = db->head;

while (temp != NULL) {

count++;

temp = temp->next;

}

printf("Total sales: %d\n", count);

}

// Function to free all nodes in the database

void freeDatabase(struct SalesDatabase \*db) {

struct PropertySale \*temp;

while (db->head != NULL) {

temp = db->head;

db->head = db->head->next;

free(temp);

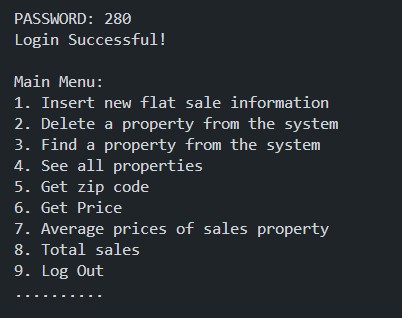
}

printf("Database memory freed.\n");

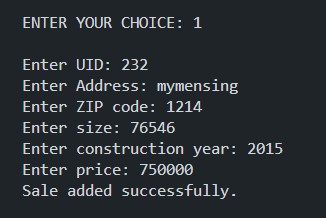
}

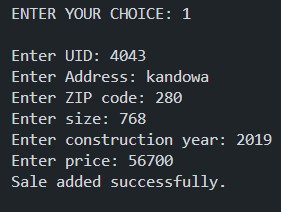
**Output:**

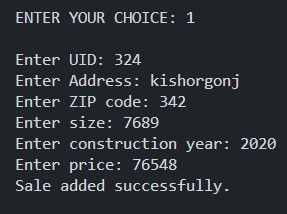
Step 1:



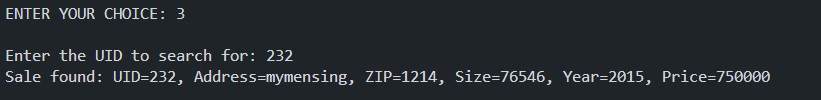
Step 2:



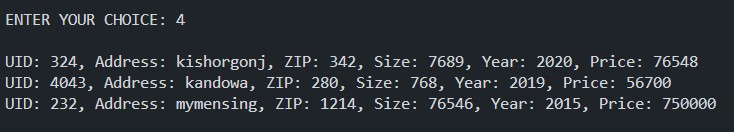




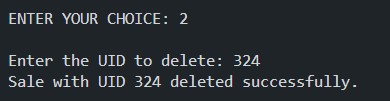
Step 3:

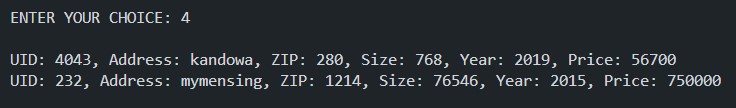


Step 4:

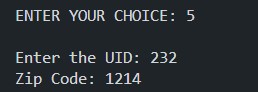
****

Step 5:

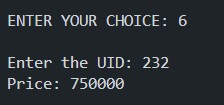




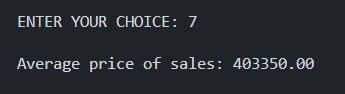
Step 6:



Step 7:



Step 8:



Step 9:



Step 10:

