**Name : Tanishq Thuse**

**Branch : SY-CS(AI)**

**Div : B**

**Roll No. : 60**

**Subject : ADS Assignment-2**

**Title : Student DataBase**

**Q1) Student database using array**

Code :

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Define a structure to represent a Student

typedef struct {

int id;

char name[100];

int age;

} Student;

// Function prototypes

void addStudent(Student students[], int \*count);

void viewStudents(const Student students[], int count);

void deleteStudent(Student students[], int \*count);

int main() {

Student students[100];

int count = 0;

int choice;

while (1) {

printf("Student Database Menu:\n");

printf("1. Add Student\n");

printf("2. View Students\n");

printf("3. Delete Student\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addStudent(students, &count);

break;

case 2:

viewStudents(students, count);

break;

case 3:

deleteStudent(students, &count);

break;

case 4:

exit(0);

default:

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

}

}

return 0;

}

// Function to add a student

void addStudent(Student students[], int \*count) {

if (\*count >= 100) {

printf("Database is full. Cannot add more students.\n");

return;

}

Student newStudent;

printf("Enter student Roll No.: ");

scanf("%d", &newStudent.id);

printf("Enter student name: ");

scanf("%s", newStudent.name);

printf("Enter student age: ");

scanf("%d", &newStudent.age);

students[\*count] = newStudent;

(\*count)++;

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

}

// Function to view all students

void viewStudents(const Student students[], int count) {

if (count == 0) {

printf("No students in the database.\n");

return;

}

printf("Student Records:\n");

for (int i = 0; i < count; i++) {

printf("ID: %d, Name: %s, Age: %d\n", students[i].id, students[i].name, students[i].age);

}

}

// Function to delete a student

void deleteStudent(Student students[], int \*count) {

if (\*count == 0) {

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

return;

}

int id;

printf("Enter student ID to delete: ");

scanf("%d", &id);

int index = -1;

for (int i = 0; i < \*count; i++) {

if (students[i].id == id) {

index = i;

break;

}

}

if (index == -1) {

printf("Student not found.\n");

return;

}

for (int i = index; i < \*count - 1; i++) {

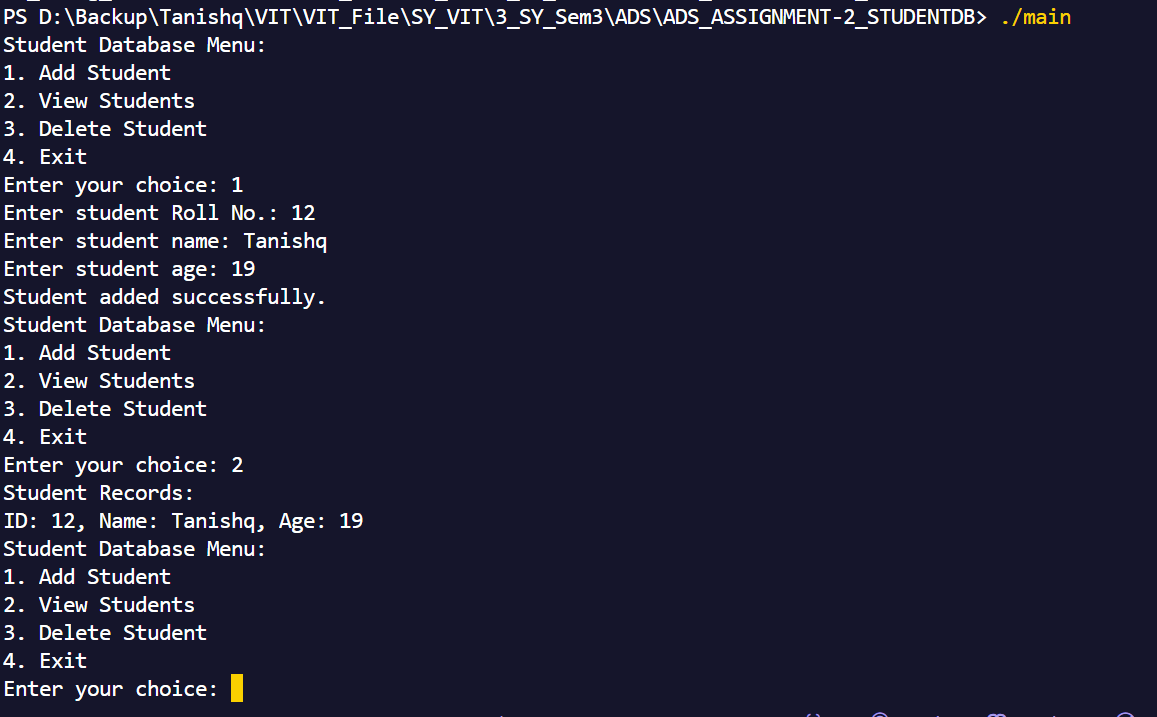
students[i] = students[i + 1];

}

(\*count)--;

printf("Student deleted successfully.\n");

}



**Q2) Student Database using LinkedList**

Code :

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Define a structure to represent a Student

typedef struct Student {

int rollNo;

char name[100];

int age;

struct Student\* next;

} Student;

// Function prototypes

void addStudent(Student\*\* head);

void viewStudents(const Student\* head);

void deleteStudent(Student\*\* head);

int main() {

Student\* head = NULL;

int choice;

while (1) {

printf("Student Database Menu:\n");

printf("1. Add Student\n");

printf("2. View Students\n");

printf("3. Delete Student\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addStudent(&head);

break;

case 2:

viewStudents(head);

break;

case 3:

deleteStudent(&head);

break;

case 4:

// Free the allocated memory before exiting

while (head != NULL) {

Student\* temp = head;

head = head->next;

free(temp);

}

exit(0);

default:

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

}

}

return 0;

}

// Function to add a student

void addStudent(Student\*\* head) {

Student\* newStudent = (Student\*)malloc(sizeof(Student));

if (newStudent == NULL) {

printf("Memory allocation failed. Cannot add student.\n");

return;

}

printf("Enter student roll number: ");

scanf("%d", &newStudent->rollNo);

printf("Enter student name: ");

scanf("%s", newStudent->name);

printf("Enter student age: ");

scanf("%d", &newStudent->age);

newStudent->next = \*head;

\*head = newStudent;

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

}

// Function to view all students

void viewStudents(const Student\* head) {

if (head == NULL) {

printf("No students in the database.\n");

return;

}

printf("Student Records:\n");

const Student\* current = head;

while (current != NULL) {

printf("Roll No: %d, Name: %s, Age: %d\n", current->rollNo, current->name, current->age);

current = current->next;

}

}

// Function to delete a student

void deleteStudent(Student\*\* head) {

if (\*head == NULL) {

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

return;

}

int rollNo;

printf("Enter student roll number to delete: ");

scanf("%d", &rollNo);

Student\* current = \*head;

Student\* previous = NULL;

while (current != NULL && current->rollNo != rollNo) {

previous = current;

current = current->next;

}

if (current == NULL) {

printf("Student not found.\n");

return;

}

if (previous == NULL) {

\*head = current->next;

} else {

previous->next = current->next;

}

free(current);

printf("Student deleted successfully.\n");

}

**Output :**

