

**WELCOME**

**SEATING ARRANGEMENT FOR  
EXAMINATIONS**

# **INTRODUCTION**

- >This C program is designed to manage the seating arrangement for students in any examination.
- >The program utilizes structures and dynamic memory allocation to handle student and room information.

# **AGENDA**

**1. Project Statement**

**2. Key Concepts**

**3. Code Structure**

**4. Codes**

**5. Output**

# **PROBLEM STATEMENT**

**->In the dynamic and fast-paced environment of educational institutions, managing seating arrangements for students poses a significant challenge. As the student body grows and classroom configurations vary, the manual assignment of seats becomes an intricate task, often leading to inefficiencies, bias, and logistical hurdles.**

# KEY CONCEPTS

->The data structures used in this project are :

1.Structures

2.Arrays

3.Linked list

## **Structures:**

->In programming,a structure is user defined data type.

->It allows us to group different variables of different data types under a single name.

->In C language,structures are declared as “struct” keyword.

## **Arrays:**

- >In a programming,an array is a data structure that allows you to store a collection of elements of the same type under a single name.
- >The elements in an array are stored in a contiguous memory location.
- >Each element is accessed by its index or position in the array

## **Linked list:**

- >A linked list is a linear data structure where elements are stored in nodes, and each node points to the next node in the sequence, forming a chain.
- > It allows for dynamic memory allocation and efficient insertion/removal of elements.

# Code Structure

- >In this code the user will be asked to enter the number of rooms and number of seats per the rooms.
- >After entering the rooms and seats per the room .Then it will give some choices,like
  - =>assign seats randomly
  - =>assign seats alphabatically
  - =>assign seats to the student based on the Roll Number
  - =>search for the student by name and roll number.
  - =>to generate the report
- >To run the the we need to have student data base in which there will be the students who are writing the examinations .
- >The output will also be displayed in the form of CSV file as well as terminal.

->Data base:

```
101 John Smith
102 Alice Johnson
103 Bob Williams
104 Emily Davis
105 David Brown
106 Sarah Lee
107 Michael Wilson
108 Lisa Anderson
109 James Martin
110 Jennifer Jackson
111 Pavan
112 Naresh
113 Nagesh
114 Naveen
115 Nikhil
116 Siva
117 Ram
118 Prasad
119 Abdul
120 Viraj
121 Mahendra
122 Mrunal
123 Thushar
124 Mithun
```



# CODE

->The different types of header files included are:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

->The different types structures used are:

```
typedef struct Student {  
    int roll_number;  
    char name[50];  
    int room_number;  
    int seat_number;  
    struct Student* next;  
} Student;  
  
typedef struct Room {  
    int room_number;  
    int seats_per_room;  
    struct Student* seating_list;  
} Room;
```

# ->main function

```
int main() {
    srand(time(NULL));

    initializeStudentsFromFile("student_dataset.txt");

    int n_rooms, seats_per_room;
    printf("Enter the number of rooms: ");
    scanf("%d", &n_rooms);
    printf("Enter the number of seats per room: ");
    scanf("%d", &seats_per_room);

    initializeRooms(n_rooms, seats_per_room);

    int choice;
    int roll_number;
    char name[50];

    do {
        printf("\n***WELCOME TO PESU SEATING ARRANGEMENT***\n");
        printf("-----\n");
        printf("1. Assign Seats Randomly\n");
        printf("2. Assign Seats Alphabetically\n");
        printf("3. Assign seats to the students based on Roll number\n");
        printf("4. Search for a student by Roll Number and Name\n");
        printf("5. Generate Total Report and Save to File\n");
        printf("6. Exit\n");
        printf("Enter your choice (1-5): ");
        scanf("%d", &choice);

        switch (choice) {
```

```

        switch (choice) {
            case 1:
                assignSeatsRandomly();
                break;

            case 2:
                assignSeatsAlphabetically();
                break;

            case 3:
                assignSeatsByRollNumber();
                break;

            case 4:
                printf("Enter the student's roll number: ");
                scanf("%d", &roll_number);
                printf("Enter the student's name: ");
                scanf(" %49[^\n]", name);
                displayStudentByRollAndName(roll_number, name);
                break;

            case 5:
                generateTotalReport();
                break;

            case 6:
                printf("Exiting the program. Goodbye!\n");
                break;

            default:
                printf("Invalid choice. Please select a valid option.\n");
        }
    } while (choice != 5);

    free(students);

    for (int i = 0; i < num_rooms; i++) {
        Student* current_student = rooms[i].seating_list;
        while (current_student != NULL) {
            Student* next_student = current_student->next;
            free(current_student);
            current_student = next_student;
        }
    }
    free(rooms);

    return 0;

```

->user defined functions:

->To initialize the students from the data base or file:

```
void initializeStudentsFromFile(const char* student_dataset) {
    FILE* file = fopen(student_dataset, "r");
    if (file == NULL) {
        printf("Failed to open the student data file.\n");
        exit(1);
    }

    int student_count = 0;
    char line[100];
    while (fgets(line, sizeof(line), file) != NULL) {
        student_count++;
    }

    students = (Student*)malloc(student_count * sizeof(Student));
    num_students = student_count;

    rewind(file);

    for (int i = 0; i < num_students; i++) {
        if (fgets(line, sizeof(line), file) == NULL) {
            printf("Error reading student data from the file.\n");
            exit(1);
        }
        sscanf(line, "%d %49[^\n]", &students[i].roll_number, students[i].name);
        students[i].room_number = 0;
        students[i].seat_number = 0;
    }

    fclose(file);
}
```

->user defined functions:

->To initialize the rooms:

```
void initializeRooms(int n_rooms, int seats_per_room) {  
    num_rooms = n_rooms;  
    rooms = (Room*)malloc(num_rooms * sizeof(Room));  
  
    for (int i = 0; i < num_rooms; i++) {  
        rooms[i].room_number = i + 1;  
        rooms[i].seats_per_room = seats_per_room;  
        rooms[i].seating_list = (Student*)malloc(seats_per_room * sizeof(Student));  
    }  
}
```

->To know whether the seat is occupied:

```
int isSeatOccupied(Room room, int seat_number) {  
    return room.seating_list[seat_number - 1].roll_number != 0;  
}
```

->To compare the names of the student:

```
Student* students = NULL;  
int num_students = 0;  
Room* rooms = NULL;  
int num_rooms = 0;  
  
int compareStudentsByName(const void* a, const void* b) {  
    return strcmp(((Student*)a)->name, ((Student*)b)->name);  
}
```

->To compare the name of the student by Roll Number:

```
int compareStudentsByRoll(const void* a, const void* b) {  
    return ((Student*)a)->roll_number - ((Student*)b)->roll_number;  
}
```

->To assign seats randomly:

```
void assignSeatsRandomly() {
    printf("Assigning seats to students randomly...\n");

    // Shuffle the students randomly
    for (int i = num_students - 1; i > 0; i--) {
        int j = rand() % (i + 1);
        // Swap students[i] and students[j]
        Student temp = students[i];
        students[i] = students[j];
        students[j] = temp;
    }

    // Assign seats based on the shuffled order
    int student_index = 0;
    for (int room = 0; room < num_rooms; room++) {
        rooms[room].seating_list = NULL; // Reset the seating list for the room
        Student* last_student = NULL;
        for (int seat = 1; seat <= rooms[room].seats_per_room; seat++) {
            if (student_index < num_students) {
                students[student_index].room_number = room + 1;
                students[student_index].seat_number = seat;
                students[student_index].next = NULL;

                if (last_student != NULL) {
                    last_student->next = &students[student_index];
                } else {
                    rooms[room].seating_list = &students[student_index];
                }

                last_student = &students[student_index];
                student_index++;
            } else {
                break;
            }
        }
        if (student_index >= num_students) {
            break;
        }
    }
}
```



->To assign the students alphanatically:

```
void assignSeatsAlphabetically() {
    printf("Assigning seats to students alphabetically...\n");

    qsort(students, num_students, sizeof(Student), compareStudentsByName);

    // Assign seats based on the sorted order
    int student_index = 0;
    for (int room = 0; room < num_rooms; room++) {
        rooms[room].seating_list = NULL; // Reset the seating list for the room
        Student* last_student = NULL;
        for (int seat = 1; seat <= rooms[room].seats_per_room; seat++) {
            if (student_index < num_students) {
                students[student_index].room_number = room + 1;
                students[student_index].seat_number = seat;
                students[student_index].next = NULL;

                if (last_student != NULL) {
                    last_student->next = &students[student_index];
                } else {
                    rooms[room].seating_list = &students[student_index];
                }

                last_student = &students[student_index];
                student_index++;
            } else {
                break;
            }
        }
        if (student_index >= num_students) {
            break;
        }
    }
}
```

->To display student by name and roll number:

```
void displayStudentByRollAndName(int roll_number, const char* name) {  
    for (int i = 0; i < num_students; i++) {  
        if (students[i].roll_number == roll_number && strcmp(students[i].name, name) == 0) {  
            printf("Student found: Roll Number: %d, Name: %s, Room: %d, Seat: %d\n",  
                students[i].roll_number, students[i].name, students[i].room_number, students[i].seat_number);  
            return;  
        }  
    }  
    printf("Student not found with Roll Number: %d and Name: %s\n", roll_number, name);  
}
```

->To generate the total report:

```
void generateTotalReport() {
    printf("Total Seating Arrangement Report:\n");
    for (int room = 0; room < num_rooms; room++) {
        printf("Room %d:\n", rooms[room].room_number);
        for (int seat = 1; seat <= rooms[room].seats_per_room; seat++) {
            int student_index = (room * rooms[room].seats_per_room) + seat - 1;
            if (students[student_index].room_number == room + 1 && students[student_index].seat_number == seat) {
                printf("Seat Number %d: Occupied by Student with Roll Number %d, Name: %s\n",
                    seat, students[student_index].roll_number, students[student_index].name);
            } else {
                printf("Seat Number %d: Unoccupied\n", seat);
            }
        }
    }

    FILE* report_file = fopen("seating_report.csv", "w");
    if (report_file == NULL) {
        printf("Failed to open the report file for writing.\n");
    } else {
        fprintf(report_file, "Total Seating Arrangement Report:\n");
        for (int room = 0; room < num_rooms; room++) {
            fprintf(report_file, "Room %d:\n", rooms[room].room_number);
            for (int seat = 1; seat <= rooms[room].seats_per_room; seat++) {
                int student_index = (room * rooms[room].seats_per_room) + seat - 1;
                if (students[student_index].room_number == room + 1 && students[student_index].seat_number == seat) {
                    fprintf(report_file, "Seat Number %d: Occupied by Student with Roll Number %d, Name: %s\n",
                        seat, students[student_index].roll_number, students[student_index].name);
                } else {
                    fprintf(report_file, "Seat Number %d: Unoccupied\n", seat);
                }
            }
        }
        fclose(report_file);
        printf("The report has been saved to 'seating_report.csv'.\n");
    }
}
```

->To assign seats to the students based on Roll Number:

```
void assignSeatsByRollNumber() {
    printf("Assigning seats to students based on roll number...\n");

    // Sort students based on roll number
    qsort(students, num_students, sizeof(Student), compareStudentsByRoll);

    // Assign seats based on the sorted order
    int student_index = 0;
    for (int room = 0; room < num_rooms; room++) {
        rooms[room].seating_list = NULL; // Reset the seating list for the room
        Student* last_student = NULL;
        for (int seat = 1; seat <= rooms[room].seats_per_room; seat++) {
            if (student_index < num_students) {
                students[student_index].room_number = room + 1;
                students[student_index].seat_number = seat;
                students[student_index].next = NULL;

                if (last_student != NULL) {
                    last_student->next = &students[student_index];
                } else {
                    rooms[room].seating_list = &students[student_index];
                }

                last_student = &students[student_index];
                student_index++;
            } else {
                break;
            }
        }
        if (student_index >= num_students) {
            break;
        }
    }
}
```

# OUTPUTS:

->when we run the program and entered the number of rooms and number of seats per room.

```
Enter the number of rooms: 2
Enter the number of seats per room: 15

***WELCOME TO PESU SEATING ARRANGEMENT***
-----
1. Assign Seats Randomly
2. Assign Seats Alphabetically
3. Assign seats to the students based on Roll number
4. Search for a student by Roll Number and Name
5. Generate Total Report and Save to File
6. Exit
```

->when we entered the choice as “1”:

```
Enter your choice (1-5): 1
Assigning seats to students randomly...

***WELCOME TO PESU SEATING ARRANGEMENT***
-----
1. Assign Seats Randomly
2. Assign Seats Alphabetically
3. Assign seats to the students based on Roll number
4. Search for a student by Roll Number and Name
5. Generate Total Report and Save to File
6. Exit
```



->When we entered the choice as “4”:

```
Enter your choice (1-5): 4
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 2: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 3: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 4: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 5: Occupied by Student with Roll Number 111, Name: Pavan Kalyan
Seat Number 6: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 7: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 8: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 9: Occupied by Student with Roll Number 115, Name: Nikhil
Seat Number 10: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 11: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 12: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 13: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 14: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 15: Occupied by Student with Roll Number 112, Name: Naresh
Room 2:
Seat Number 1: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 2: Occupied by Student with Roll Number 120, Name: Virat
Seat Number 3: Occupied by Student with Roll Number 121, Name: Mahendra Singh Dhoni
Seat Number 4: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 5: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 6: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 7: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 8: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 9: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
The report has been saved to 'seating_report.csv'.

***WELCOME TO PESU SEATING ARRANGEMENT***
-----
1. Assign Seats Randomly
2. Assign Seats Alphabetically
3. Search for a student by Roll Number and Name
4. Generate Total Report and Save to File
5. Exit
Enter your choice (1-5): █
```

->And the same output is also stored in the form of CSV file:

```
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 2: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 3: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 4: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 5: Occupied by Student with Roll Number 111, Name: Pavan Kalyan
Seat Number 6: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 7: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 8: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 9: Occupied by Student with Roll Number 115, Name: Nikhil
Seat Number 10: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 11: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 12: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 13: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 14: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 15: Occupied by Student with Roll Number 112, Name: Naresh
Room 2:
Seat Number 1: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 2: Occupied by Student with Roll Number 120, Name: Virat
Seat Number 3: Occupied by Student with Roll Number 121, Name: Mahendra Singh Dhoni
Seat Number 4: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 5: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 6: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 7: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 8: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 9: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
```



->In the first choice if we chose “2”:

```
Enter your choice (1-5): 2
Assigning seats to students alphabetically...

***WELCOME TO PESU SEATING ARRANGEMENT***
-----
1. Assign Seats Randomly
2. Assign Seats Alphabetically
3. Assign seats to the students based on Roll number
4. Search for a student by Roll Number and Name
5. Generate Total Report and Save to File
6. Exit
```

->If we enter know the “4”:

```
Enter your choice (1-5): 4
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 2: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 3: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 4: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 5: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 6: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 7: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 8: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 9: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 10: Occupied by Student with Roll Number 121, Name: Mahendra Singh Dhoni
Seat Number 11: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 12: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 13: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 14: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 15: Occupied by Student with Roll Number 112, Name: Naresh
Room 2:
Seat Number 1: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 2: Occupied by Student with Roll Number 115, Name: Nikhil
Seat Number 3: Occupied by Student with Roll Number 111, Name: Pavan Kalyan
Seat Number 4: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 5: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 6: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 7: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 8: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 9: Occupied by Student with Roll Number 120, Name: Virat
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
The report has been saved to 'seating_report.csv'.
```

->The output will also saved in the CSV file:

```
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 2: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 3: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 4: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 5: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 6: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 7: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 8: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 9: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 10: Occupied by Student with Roll Number 121, Name: Mahendra Singh Dhoni
Seat Number 11: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 12: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 13: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 14: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 15: Occupied by Student with Roll Number 112, Name: Naresh
Room 2:
Seat Number 1: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 2: Occupied by Student with Roll Number 115, Name: Nikhil
Seat Number 3: Occupied by Student with Roll Number 111, Name: Pavan Kalyan
Seat Number 4: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 5: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 6: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 7: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 8: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 9: Occupied by Student with Roll Number 120, Name: Virat
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
```

->If we entered the choice as “3”:

```
Enter your choice (1-5): 3
Assigning seats to students based on roll number...

***WELCOME TO PESU SEATING ARRANGEMENT***
-----
1. Assign Seats Randomly
2. Assign Seats Alphabetically
3. Assign seats to the students based on Roll number
4. Search for a student by Roll Number and Name
5. Generate Total Report and Save to File
6. Exit
```



->know we enter the “5” and the output will be like:

```
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 2: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 3: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 4: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 5: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 6: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 7: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 8: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 9: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 10: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 11: Occupied by Student with Roll Number 111, Name: Pavan
Seat Number 12: Occupied by Student with Roll Number 112, Name: Naresh
Seat Number 13: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 14: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 15: Occupied by Student with Roll Number 115, Name: Nikhil
Room 2:
Seat Number 1: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 2: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 3: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 4: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 5: Occupied by Student with Roll Number 120, Name: Viraj
Seat Number 6: Occupied by Student with Roll Number 121, Name: Mahendra
Seat Number 7: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 8: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 9: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
The report has been saved to 'seating_report.csv'.
```

->The report will also be saved in our CSV file:

```
Total Seating Arrangement Report:
Room 1:
Seat Number 1: Occupied by Student with Roll Number 101, Name: John Smith
Seat Number 2: Occupied by Student with Roll Number 102, Name: Alice Johnson
Seat Number 3: Occupied by Student with Roll Number 103, Name: Bob Williams
Seat Number 4: Occupied by Student with Roll Number 104, Name: Emily Davis
Seat Number 5: Occupied by Student with Roll Number 105, Name: David Brown
Seat Number 6: Occupied by Student with Roll Number 106, Name: Sarah Lee
Seat Number 7: Occupied by Student with Roll Number 107, Name: Michael Wilson
Seat Number 8: Occupied by Student with Roll Number 108, Name: Lisa Anderson
Seat Number 9: Occupied by Student with Roll Number 109, Name: James Martin
Seat Number 10: Occupied by Student with Roll Number 110, Name: Jennifer Jackson
Seat Number 11: Occupied by Student with Roll Number 111, Name: Pavan
Seat Number 12: Occupied by Student with Roll Number 112, Name: Naresh
Seat Number 13: Occupied by Student with Roll Number 113, Name: Nagesh
Seat Number 14: Occupied by Student with Roll Number 114, Name: Naveen
Seat Number 15: Occupied by Student with Roll Number 115, Name: Nikhil
Room 2:
Seat Number 1: Occupied by Student with Roll Number 116, Name: Siva
Seat Number 2: Occupied by Student with Roll Number 117, Name: Ram
Seat Number 3: Occupied by Student with Roll Number 118, Name: Prasad
Seat Number 4: Occupied by Student with Roll Number 119, Name: Abdul
Seat Number 5: Occupied by Student with Roll Number 120, Name: Viraj
Seat Number 6: Occupied by Student with Roll Number 121, Name: Mahendra
Seat Number 7: Occupied by Student with Roll Number 122, Name: Mrunal
Seat Number 8: Occupied by Student with Roll Number 123, Name: Thushar
Seat Number 9: Occupied by Student with Roll Number 124, Name: Mithun
Seat Number 10: Unoccupied
Seat Number 11: Unoccupied
Seat Number 12: Unoccupied
Seat Number 13: Unoccupied
Seat Number 14: Unoccupied
Seat Number 15: Unoccupied
```

->If we entered the choice as “3”:

->when we entered the choice as “3” then it will ask the user to enter the Roll number and Name.

```
Enter your choice (1-5): 3
Enter the student's roll number: 101
Enter the student's name: John Smith
Student found: Roll Number: 101, Name: John Smith, Room: 1, Seat: 8
```

->if we entered the wrong details it will display as the student not found with particular roll number and name.

->The project done by :

- |                     |                   |
|---------------------|-------------------|
| 1.M.SIVA RAM PRASAD | SRN:PES2UG22CS324 |
| 2.NISCHITH          | SRN:PES2UG22CS370 |
| 3.NARESH P V        | SRN:PES2UG22CS340 |