

**PROJECT REPORT**

A report submitted in partial fulfillment of the requirements for the



**Project**

School of Computer Science & Engineering By

# 

# Shabdshree Lohar

PRN No:22SC114281023 Roll No: 17

# Sakshi Chougule

PRN No:22SC114281024 Roll No: 18

# Rashi Ingale

PRN No:22SC114281025 Roll No: 19

# Suraj Karande

PRN No:22SC114281026 Roll No: 20

Program: BTech Class: SY BTech (Div A)

Under Supervision of

# Mrs.Veena Mali

Academic Year: 2023-2024



School of Computer Science & Engineering

**CERTIFICATE**

This is to certify that the “**Project Report**”

On

# “Dormitory administration”

submitted by

# Shabdshree Lohar

PRN No:22SC114281023 Roll No: 17

# Sakshi Chougule

PRN No:22SC114281024 Roll No: 18

# Rashi Ingale

PRN No:22SC114281025 Roll No: 19

# Suraj Karande

PRN No:22SC114281026 Roll No: 20

Program: BTech CSE Class: SY BTech(Div A)

is work done by him/her and submitted during the 2023 – 2024 academic year, in partial fulfillment of the **Project.**

**Sanjay Ghodawat University, Kolhapur**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mrs Veena Mali** | **Ms. Deepika Patil** | **Dr. Mrs.Deepika Patil** |  |
| **Project Guide** | **PBL Co-ordinator** | **Head, SOCSE &AIML** | **External** |



# DECLARATION

I the undersigned solemnly declare that the report of the project work entitled **“DORMITORY ADMINISTRATION”** which is carried out under the supervision of **Mrs. Veena Mali** I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree certificate in this University or any other University.

# Student Name:

# Shabdshree Lohar

PRN No:22SC114281023 Roll No: 17

# Sakshi Chaougule

PRN No:22SC114281024 Roll No: 18

# Rashi Ingale

PRN No:22SC114281025 Roll No: 19

# Suraj Karande

PRN No:22SC114281026 Roll No: 20

# Class: SY BTech (Div A)



# ACKNOWLEDGMENT

First, I would like to thank my Head of the Department **Dr. Deepika Patil** for constructive criticism throughout my project. I would like to thank PBL coordinator **Dr. Deepika Patil** and Department Project Guide **Mrs. Veena Mali** for support and advices to get and complete internship in above said organization. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am extremely grateful to my department staff members and friends who helped me in successful completion of this project.



# ABSTRACT

# Dormitory Administration System using C++ involves the development of a software system to manage hostel-related operations. This system would include functionalities such as managing student details, room allocation, fee payments, inventory tracking, staff management, and generating reports. Using C++, the system would be designed with classes, inheritance, and data structures to efficiently handle the tasks related to hostel management. The abstract might cover the program's architecture, essential classes (such as Student, Room, Hostel), and the functions required for different operations within the hostel. The system would provide a user-friendly interface for administrators, staff, and students to access and manage hostel-related tasks efficiently.

# TABLE OF CONTENT’S

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **Title** | **Page No.** |
| 1 | Introduction | 01-04 |
| 2. | Objective | 05 |
| 3 | System Requirements Specification (SRS) | 06 |
| 4 | Methodology | 07-08 |
| 5 | Implementation | 09-17 |
| 6 | Result | 18 |
| 7 | Conclusion And Future Scope | 19-20 |
| 8 | References | 21 |

# Introduction

## Introduction

## Dormitory administration is a c++ program utilizing a linked list and array is a modern and efficient approach to handle the allocation and tracking of rooms and guests within a hostel facility. In this system, each room is represented as a node in a linked list, and these linked lists are organized into an array corresponding to different floors or sections of the hostel. This method allows for dynamic management of rooms and guests, making it easier to add or remove rooms, assign or vacate them, and maintain a clear record of occupancy. It provides flexibility in handling various room types, pricing, and occupancy rules. Moreover, the use of a linked list array in dormitory administration simplifies tasks like check-ins, checkouts, and room transfers. It also aids in optimizing room allocation based on guest preferences, availability, and other factors. This technology-driven approach enhances efficiency, reduces errors, and enhances the overall guest experience in dormitory administration.

## Problem Definition:

A hostel management system using C++ typically involves the creation of a software solution that manages various aspects of a hostel, including room allocation, student information administrative tasks. It requires implementing functionalities such as:

1. Student Management: Storing and managing student information like name, ID, contact details, etc.

2. Room Allocation: Assigning and managing hostel rooms for students.

3. Administrative Tasks: Handling administrative activities such as staff management, inventory, maintenance requests, etc.

4. Reporting: Generating reports on room occupancy.

## Scope:

The scope of a hostel management system using C++ typically encompasses various areas:

1. Student Information Management: Capturing, storing, and managing student details, including personal information, contact details, academic information, and hostel-related data.

2. Room Allocation and Management: Assigning rooms to students, managing room availability, dealing with room change requests, and maintaining room-related data.

3. Admin and Staff Management: Allowing administrative staff to manage hostel-related tasks, track inventory, handle maintenance requests, and oversee day-to-day operations.

5. Reporting and Analytics: Providing reports on occupancy rates, fee collections, pending payments, and other relevant data for decision-making.

The scope might also extend to include security measures, automated communication systems, integration with other administrative systems, and scalability for accommodating potential future expansion or modifications in hostel infrastructure.

The implementation using C++ involves utilizing data structures, file handling, classes, and other language features to create a comprehensive and efficient hostel management system.

## Problem Identification:

Dormitory Administration System using C++ could address includes:

1. Manual Data Management: Difficulty in managing vast amounts of student data, room allocations, fee records, and administrative tasks using manual paperwork.

2. Inefficient Room Allocation: Difficulty in efficiently assigning rooms to students, tracking room occupancy, and managing room changes.

3. Limited Reporting and Analytics: Lack of a system to provide comprehensive reports and analytics on occupancy rates, fee collections, and other critical data for informed decision-making.

4.Administrative Tasks Overload: Overwhelming administrative tasks, including staff management, inventory tracking, and maintenance requests, leading to inefficiencies and errors.

Developing a Dormitory Administration System in C++ aims to solve these problems by automating processes, ensuring efficient data handling, enabling better room allocation and fee management, and providing tools for streamlined administrative tasks, reporting, and analytics.

# Objectives

1.Efficient Room Allocation: Ensure optimal allocation of rooms to guests, taking into account room availability, preferences, and booking details.

2. Streamlined Booking Process: Simplify the reservation process for guests, allowing online booking, check-in, and check-out.

3. Inventory Management: Keep track of available rooms, their types, and pricing, ensuring accurate information for bookings.

4. Guest Information: Maintain a database of guest details, including contact information, stay history, and preferences

# System Requirements Specification

## Software Requirement:

* Microsoft Visual Studio Code
* Dev C++

## Hardware Requirement

## Computer or laptop

## Intel (R)core (TM)

## RAM -1 GB Minimum

* 500 GB HDD
* i3 processor

# Methodology

## Algorithm:

## Step 1: Start

## Step 2: Choose the options for the room service below ( book / cancel/check)

## Step 3: Display current status of room

## Step 4: Search for the room according how many people you are

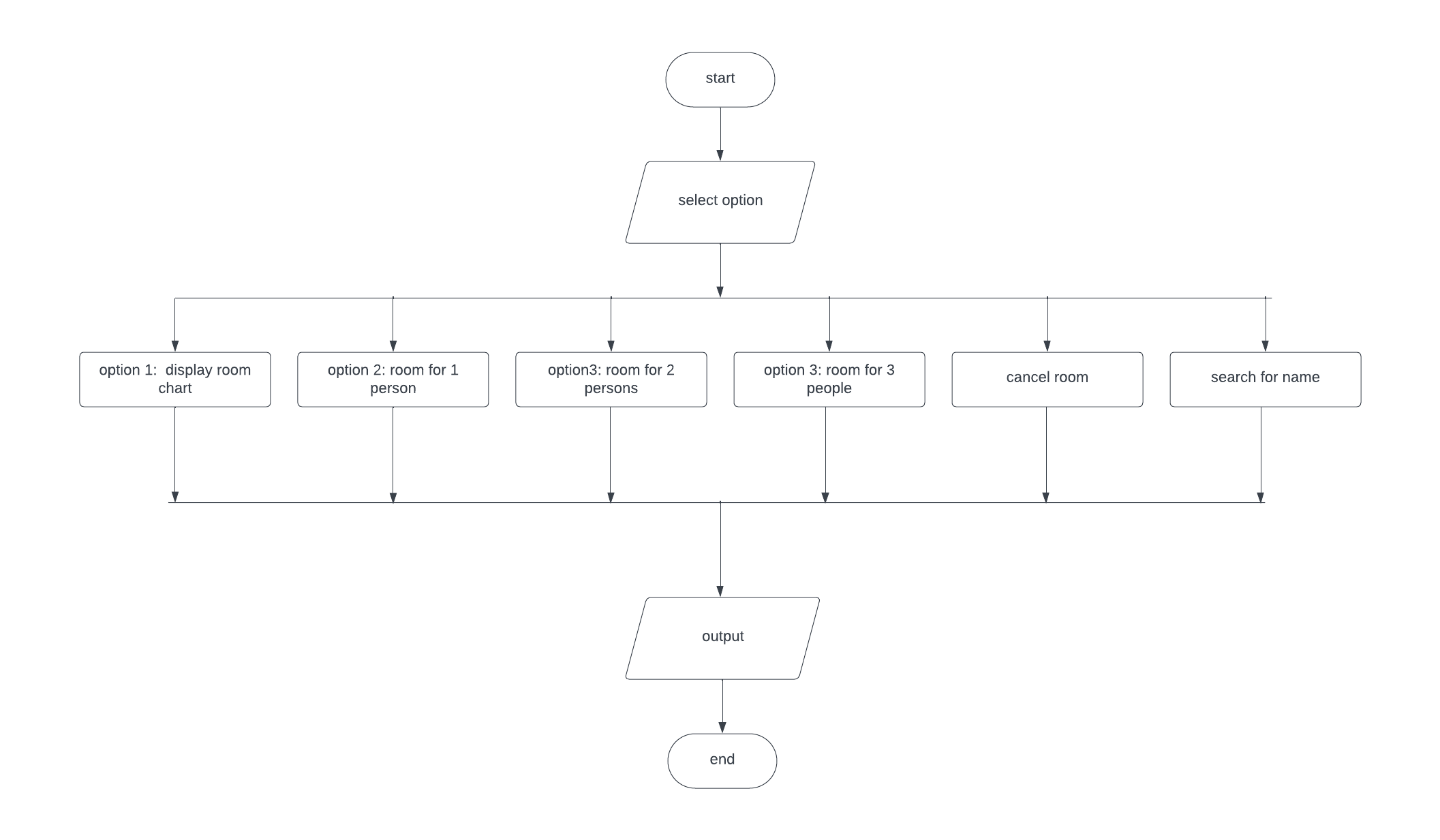
## Step 5: Displays vaccant rooms Step 6 : Book your room

## Step 7: Enter your name

## Step 8: Your room is booked

## Step 9: End

* **Flow Diagram(Flow Chart):**



# Implementation

Creating a hostel management system in C++ involves various components like student information, room allocation, fee management, and more. Below is a simplified overview of what such a system might include:

Hostel Management System in C++:

1. Student Information:

Create a Student class with attributes like name, ID, contact details, etc. Implement methods to add, update, delete, and display student information.

2. Room Allocation:

Design a Room class with room number, capacity, availability, etc. Manage room allocation to students and track available rooms.

3. Fee Management:

Maintain fee details, payment records, and due dates for each student. Implement functionalities for fee payment and overdue fee tracking.

4. Staff Information:

Store details of hostel staff (wardens, maintenance, etc.) if required.

5. Menu System:

Allow management of the hostel mess/menu if applicable.

****

**“Introduction of C++Programming”**

**C**++ is a powerful and versatile programming language that originated as an extension of the C programming language. Developed by Bjarne Stroustrup in the early 1980s, C++ is known for its efficiency, flexibility, and high performance. It's widely used in various domains such as system software, application software, game development, embedded systems, and more.

Key Aspects of C++:

1.Object-Oriented Programming (OOP):C++ supports the key principles of OOP, including classes, objects, inheritance, polymorphism, and encapsulation. This paradigm enables developers to organize and structure their code more efficiently.

2. Efficiency and Performance: C++ allows low-level memory manipulation, providing greater control over system resources. This makes it well-suited for performance-critical applications like games or operating systems.

3. Standard Template Library (STL): The STL offers a collection of classes and functions for common programming tasks, like data structures, algorithms, and I/O operations. It includes containers (vectors, lists, maps), algorithms, and iterators, enhancing code reusability and efficiency.

4. Portability: C++ code can be compiled and executed on various platforms, offering portability across different operating systems and hardware.

5. Wide Applicability: It's used in a variety of fields, from developing system software (like operating systems) to creating high-performance applications, games, and even in embedded systems.

**Advantages of learning C++**:

Learning C++ offers numerous advantages due to its versatility, performance, and wide range of applications:

1. Performance and Efficiency:

Speed: C++ allows direct manipulation of hardware resources, enabling efficient code execution.

Low-level Manipulation: Provides control over memory allocation and management, making it highly efficient.

2. Versatility:

Multi-paradigm Language: Supports procedural, object-oriented, and generic programming, offering a flexible approach to problem-solving.

Large Standard Library: The Standard Template Library (STL) includes containers, algorithms, and utilities for various programming tasks.

3. Wide Applicability:

System Software: Used in developing operating systems, device drivers, and other system software.

Application Software: Employed in building high-performance applications, games, and complex software systems.

Embedded Systems: Commonly used in embedded systems and IoT devices where resource efficiency is crucial.

4. Career Opportunities:

In-Demand Skill: C++ remains a sought-after language in industries like game development, finance, high-performance computing, and more.

\*Foundation for Learning Other Languages:\* Understanding C++ provides a strong foundation for learning other languages due to its comprehensive nature.

5. Critical Thinking and Problem-Solving:

Memory Management: Learning C++ sharpens understanding of memory allocation, pointers, and resource management, promoting a deeper understanding of computer architecture.

Optimized Solutions: Challenges in C++ often require critical thinking and efficient problem-solving, fostering valuable skills applicable across different domains.

## Source Code:

#include <bits/stdc++.h>

using namespace std;

string name\_save[10][3];

class Node{

public:

int room\_capacity = 1;

int fill\_room = 0;

char name[3][10];

Node \*next;

Node \*previous;

};

class hostel{

Node \*head[3];

Node \*created\_node;

public:

hostel()

{

for (int i = 0; i < 3; i++)

head[i] = NULL;

}

void create();

void book(int);

void cancel(int);

void display();

void search(string);

};

void hostel :: create(){

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 9; j++)

{

Node \*newnode = new Node;

newnode->next = NULL;

newnode->previous = NULL;

if (head[i] == NULL)

{

head[i] = newnode;

newnode -> room\_capacity = 1;

}

else

{

Node \*created\_node = head[i];

while (created\_node -> next != NULL)

{

created\_node = created\_node -> next;

}

if (j == 3 || j == 5 || j == 7 || j == 8)

{

created\_node -> room\_capacity = 3;

}

if (j == 2 || j == 4 || j == 6)

{

created\_node -> room\_capacity = 2;

}

created\_node -> next = newnode;

newnode -> previous = created\_node;

}

}

}

}

void hostel :: book(int people){

int floor, room;

cout << "Enter the floor number: ";

cin >> floor;

if (floor < 0 || floor > 4){

cout << "Invalid floor number : " << floor << endl;

}

else{

created\_node = head[floor - 1];

cout << "Enter the room number: ";

cin >> room;

if (room < 0 || room >=10){

cout << "Invalid room number: " << room << endl;

}

else{

int i = 1;

while (i < room){

created\_node = created\_node -> next;

i++;

}

if (created\_node -> room\_capacity >= people){

cout << "Room is Available! You can Apply for the Room!" << endl;

int count = 0;

while(created\_node -> fill\_room - 1 <= created\_node -> room\_capacity){

cout << "Enter Name - " << created\_node -> fill\_room + 1 << ": ";

cin >> created\_node -> name[created\_node -> fill\_room];

cout << "Re-Enter Name - " << created\_node-> fill\_room + 1 << ": ";

cin >> name\_save[room-1][floor-1];

count++;

created\_node -> fill\_room++;

cout<<"Room Booked";

if(count >= people){

break;

}

}

created\_node -> room\_capacity = created\_node -> room\_capacity - people;

}

else{

cout << "Room of Your Choice is Currently Not Available" << endl;

}

}

}

}

void hostel :: cancel(int check){

char namecheck[10];

int flag = 0;

int room, i = 1;

if (check < 0 || check > 4)

{

cout << "Invalid Floor No.: " << check << endl;

}

else

{

cout << "Enter Room Number: ";

cin >> room;

if (room < 0 || room > 10)

{

cout << "Invalid Room number: " << room << endl;

}

else

{

cout << "Enter the name to be deleted: ";

cin >> namecheck;

string ncheck = namecheck;

search(ncheck);

created\_node = head[check - 1];

while (i < room)

{

created\_node = created\_node -> next;

i++;

}

i = 0;

while (i < 3)

{

if (!strcmp(namecheck, created\_node -> name[i]))

{

flag = 1;

break;

i = 0;

}

else

i++;

}

if (flag == 1 && created\_node -> fill\_room != 0)

{

cout << "Record deleted : " << created\_node -> name[i] << endl;

created\_node -> name[i][0] = 'A';

created\_node -> name[i][1] = '\0';

created\_node -> fill\_room--;

created\_node -> room\_capacity++;

}

else

cout << "\nRecord Not Avaiable" << endl;

}

}

}

void hostel :: display(){

int j = 0, k = 0, l = 0;

for (int i = 0; i < 60; i++){

cout << "--";

}

cout << endl;

for (int i = 1; i < 4; i++){

cout << " Floor number : " << i << " \t\t\t";

}

cout << endl;

for (int i = 0; i < 60; i++){

cout << "--";

}

created\_node = head[j];

Node \*secondfloor\_node = head[j + 1];

Node \*thirdfloor\_node = head[j + 2];

cout << endl;

while (created\_node != NULL){

if (created\_node->fill\_room != created\_node->room\_capacity && created\_node->room\_capacity != 0){

j++;

cout << " Room no: " << j;

cout << ":-- Vacant Room --: " << created\_node -> room\_capacity;

}

else{

j++;

cout << " Room no: " << j;

cout << " Already Booked by: "<< name\_save[j-1][0];

}

if (secondfloor\_node->fill\_room != secondfloor\_node->room\_capacity && secondfloor\_node->room\_capacity != 0){

k++;

cout << "\t Room no: " << k;

cout << ":-- Vacant Room --: "<< secondfloor\_node->room\_capacity;

}

else{

k++;

cout << "\t Room no: " << k;

cout << "\t Already Booked by: "<< name\_save[k-1][1];

}

if(thirdfloor\_node->fill\_room != thirdfloor\_node->room\_capacity && thirdfloor\_node->room\_capacity != 0){

l++;

cout << "\t Room no: " << l;

cout << ":-- Vacant Room --: "<< thirdfloor\_node->room\_capacity << "| ";

}

else{

l++;

cout << " Room no: " << l;

cout << "\t Already Booked by: "<< name\_save[l-1][2];

}

cout << " " << endl;

cout << endl;

created\_node = created\_node->next;

secondfloor\_node = secondfloor\_node->next;

thirdfloor\_node = thirdfloor\_node->next;

}

for (int i = 0; i < 60; i++){

cout << "--";

}

cout << endl;

}

void hostel :: search(string key){

bool found = 0;

int i\_position;

int j\_position;

for(int i = 0; i < 10; i++){

for(int j = 0; j < 3; j++){

if(name\_save[i][j] == key){

found = 1;

i\_position = i;

j\_position = j;

break;

}

}

}

if(found)

cout << "Name Found at: \nRoom no " << i\_position + 1 << "\nFloor no. " << j\_position + 1<< endl;

else

cout << "Name not found "<< endl;

}

int main(){

hostel management;

int choice, floorcheck;

char ch;

string namecheck;

management.create();

do{

cout<<"\*\*\*\*\*\*\*\*\*\*\*DORMITORY ADMINISTRATION\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

cout << "1. Display Current Status of Rooms" << endl;

cout << "2. Book a Room for 1 person." << endl;

cout << "3. Book a Room for 2 person." << endl;

cout << "4. Book a Room for 3 person." << endl;

cout << "5. Cancel a Room" << endl;

cout << "6. Search a Name" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice){

case 1:

management.display();

break;

case 2:

management.book(1);

break;

case 3:

management.book(2);

break;

case 4:

management.book(3);

break;

case 5:

cout << "Enter Floor no.: ";

cin >> floorcheck;

management.cancel(floorcheck);

break;

case 6:

cout << "Enter name: ";

cin >> namecheck;

management.search(namecheck);

break;

default:

cout << "Invalid Choice" << endl;

}

cout << "Do You want to Continue (Y/N): ";

cin >> ch;

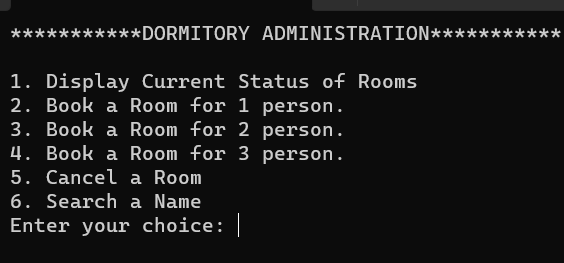
cout << endl;

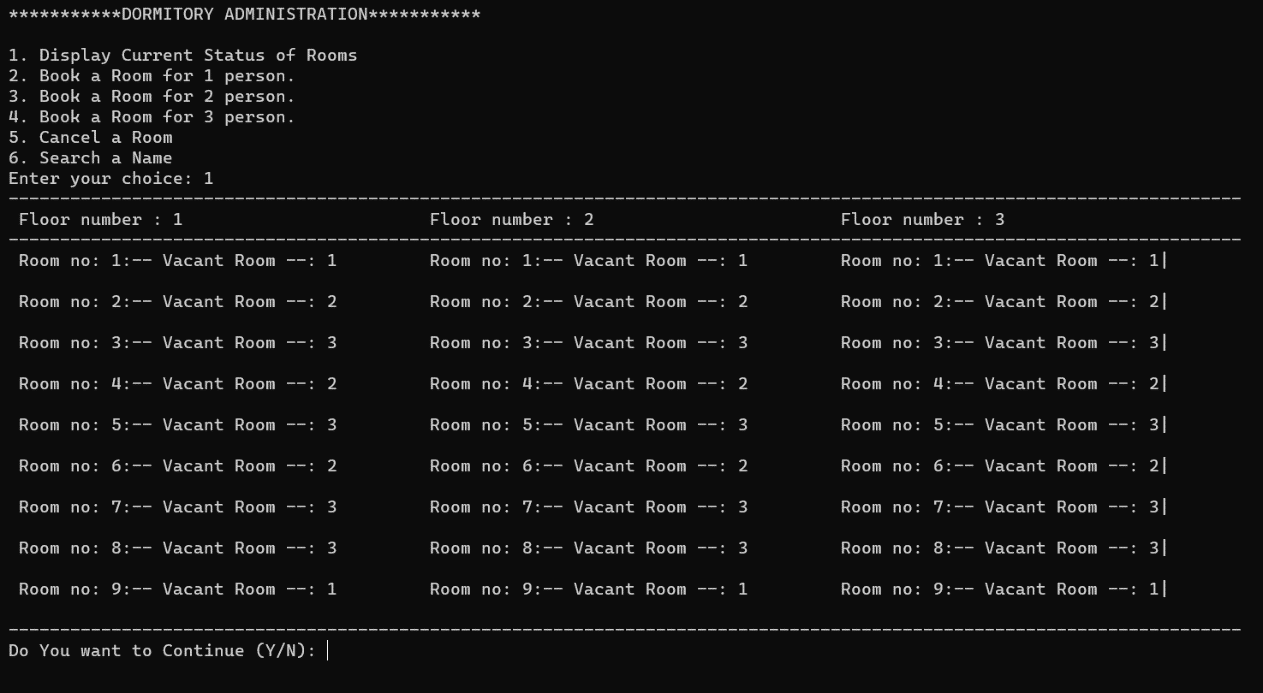
}while(ch == 'Y' || ch == 'y');

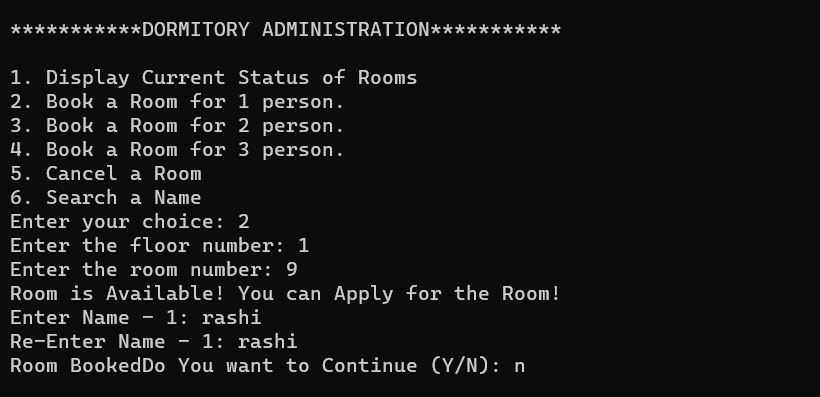
return 0;

}

# Result







# Conclusion & Future Scope

## Future Scope:

## Creating a dormitory administration system using C++ can offer various opportunities in the future, such as:

## 1. Industry Relevance: Hostel management systems are widely used in educational institutions, corporate settings, and hospitality. Developing this system in C++ can provide a solid understanding of system development that's still relevant in the industry.

## 2. Software Development Skills: Building a hostel management system allows you to enhance your skills in C++ programming, data structures, algorithms, and system design, which are valuable skills in software development.

## 3. Career Prospects: Knowledge in C++ and system development can open doors for roles in software development, system architecture, database management, and more.

## 4. Entrepreneurial Opportunities: You could potentially market and sell your hostel management system or use it as a foundation for other software ventures.

## 5. Enhanced Problem-solving Abilities: Developing such a system can improve your problem-solving skills and logical thinking, valuable for tackling complex challenges in various domains.

## Ensure you keep your system adaptable and scalable, incorporating robust features, security measures, and user-friendly interfaces to stay competitive in the evolving tech landscape

## Conclusion:

In conclusion, developing a hostel management system using C++ offers a plethora of advantages. It not only provides hands-on experience in programming, data handling, and system design but also opens doors to a wide array of career opportunities in software development and related fields. The project serves as a platform to sharpen problem-solving skills and offers prospects for entrepreneurial endeavors, whether through marketing the system or using it as a foundation for other software solutions. To maximize its potential, ensure the system is robust, adaptable, scalable, and user-friendly to remain competitive in the ever-evolving technological landscapes

# References

## BOOKS

## 1.The C++ Programming Language

## - BJIARNE STROUSTRUP

## 2. Effective C++

## - SCOTT MEYERS

## LINKS:

## <https://www.lucidchart.com>

## <https://www.geeksforgeeks.org/c-plus-plus/>