

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**A Project Report**

**On**

**My Hostel**

**Submitted By:**

Anmol Kumar Gupta (THA077BCT011)

Shailendra Rawal (THA077BCT041)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

**Under the Supervision of**

Er. Saroj Shakya

October, 2021

# COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering and author’s written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head

Department of Electronics and Computer Engineering

Thapathali Campus, Institute of Engineering

Lalitpur, Kathmandu

Nepal

# ACKNOWLEDGEMENT

We would like to express our gratitude towards our Department of Electronics and Computer Engineering for providing us with the opportunity to explore and implement the knowledge of programming in this project. Not just with the exploration and implementation of the programs the project was also helpful for us students to collaborate as a team and it served as a way to showcase our creativity. All the project members did a lot of work in learning and contributing towards the project. We tried our best to resolve all the problems we faced and in the journey we ended up learning more than ever. It was a pleasure doing this project and so we would like to acknowledge the equal contributions of every individuals for the project “**My Hostel”**.

We would also like to express our respect towards to the authors and various developers whose works we have been referenced in building this project. The articles served as a great source of knowledge and inspiration for all of us. The entire project wouldn’t have gone so smoothly without their help.

We would specially like to thank our lecturer Er. Saroj Shakya who was the first person we looked upon when we faced any problems. His encouragement and words drove us into this creating this wonderful project.

Finally, we would like to express our thankfulness to all the friends, teacher, authors and developers who have been involved with us in the project in one way or the other. We are extremely grateful for their support, cooperation, help, guidance and encouragement in doing this project.

Anmol Kumar Gupta (THA077BCT011)

Shailendra Rawal (THA077BCT041)

# ABSTRACT

In this digital world, each and every business wants to store their data in digital form rather than in written form. An organized and systematic solution is required for such business entities. Hostel is one of such business that also has its data and the owner of the hostel requires a proper record management system to keep record of its students in digital way. The information stored by hostel administration can be general student details, contact information, guardian details, etc. Looking into traditional way, these information are being created and stored in written form due to which it becomes a bit difficult to maintain, modify and store such records. Therefore, a computer based Data Management System would help the hostel administration to maintain their data and information in efficient way.

Considering this, we created a Database Management System which would help the hostel administration to easily store their data in computer. We are planning to utilize the powerful database management system, data retrival and data manipulation. We will provide more ease for managing the data than manually writing in the document. Proposed management system will eliminate all the manual intervetion and increase the speed of whole process.

*Keywords: Hostel, Students, Database, Record, Management, System, C Programming, File Handling, User-defined, Function*

# Table of Contents

[COPYRIGHT i](#_Toc86162601)

[ACKNOWLEDGEMENT ii](#_Toc86162602)

[ABSTRACT iii](#_Toc86162603)

[List of Figures vii](#_Toc86162605)

[List of Tables viii](#_Toc86162606)

[List of Abbreviations ix](#_Toc86162607)

[1. INTRODUCTION 1](#_Toc86162608)

[1.1 Background 1](#_Toc86162609)

[1.2 Motivation 1](#_Toc86162610)

[1.3 Objectives 1](#_Toc86162611)

[1.4 Scope 2](#_Toc86162612)

[2. LITERATURE REVIEW 3](#_Toc86162613)

[2.1 Student Record System in C: Mini C Project 3](#_Toc86162614)

[2.2 Hostel Management System 3](#_Toc86162615)

[3. METHODOLOGY 5](#_Toc86162616)

[3.1 Admin Login 5](#_Toc86162617)

[3.2 Main Menu 5](#_Toc86162618)

[3.2.1 Student Registration 6](#_Toc86162619)

[3.2.2 View All Student Details 6](#_Toc86162620)

[3.2.3 Search Student Details 7](#_Toc86162621)

[3.2.4 Students Access 7](#_Toc86162622)

[3.2.5 Update Student Details 7](#_Toc86162623)

[3.2.6 Delete Student Record 7](#_Toc86162624)

[3.2.7 Exit 8](#_Toc86162625)

[4. SYSTEM DESCRIPTION 9](#_Toc86162626)

[4.1 Admin Login 9](#_Toc86162627)

[4.2 Main Menu 9](#_Toc86162628)

[4.3 Student Registration 9](#_Toc86162629)

[4.4 Viewing All Students Details 10](#_Toc86162630)

[4.5 Searching Student Details 10](#_Toc86162631)

[4.6 Viewing Student Access 10](#_Toc86162632)

[4.7 Updating Student Details 11](#_Toc86162633)

[4.8 Deleting Student Details 11](#_Toc86162634)

[4.9 Exiting The Program 12](#_Toc86162635)

[5. RESULTS AND ANALYSIS 13](#_Toc86162636)

[5.1 Login Page 13](#_Toc86162637)

[5.2 Main Menu 13](#_Toc86162638)

[5.3 Student Registration 14](#_Toc86162639)

[5.4 All Student Details List 14](#_Toc86162640)

[5.5 Search Student Details 15](#_Toc86162641)

[5.5.1 Search By Student Name 15](#_Toc86162642)

[5.5.2 Search By Room No. 16](#_Toc86162643)

[5.6 Student Access 16](#_Toc86162644)

[5.7 Updating Student Details 17](#_Toc86162645)

[5.7.1 Updating Details By Id/Form No. 17](#_Toc86162646)

[5.7.2 Updating Details By Name 18](#_Toc86162647)

[5.8 Deleting Student Details 19](#_Toc86162648)

[5.8.1 Deleting Details By Id/Form No. 20](#_Toc86162649)

[5.8.2 Deleting Details By Name 20](#_Toc86162650)

[5.9 Exiting the Program 21](#_Toc86162651)

[6. CONCLUSION AND FUTURE ENHANCEMENT 22](#_Toc86162652)

[6.1 Conclusion 22](#_Toc86162653)

[6.2 Limitations 22](#_Toc86162654)

[6.3 Future Enhancement 22](#_Toc86162655)

[7. APPENDICES 23](#_Toc86162656)

[Appendix A: Gantt Chart 23](#_Toc86162657)

[References 24](#_Toc86162658)

# List of Figures

[Figure 3‑1 : Flowchart of Admin Login 5](#_Toc86007031)

[Figure 5‑1 : Login Page 13](#_Toc86007032)

[Figure 5‑2 : Main Menu Page 13](#_Toc86007033)

[Figure 5‑3 : Student Registration Page 14](#_Toc86007034)

[Figure 5‑4 : All Student Details Page 14](#_Toc86007035)

[Figure 5‑5 : Search Student Details First Page 15](#_Toc86007036)

[Figure 5‑6 : Search By Student Name 15](#_Toc86007037)

[Figure 5‑7 : Search By Student Room No. 16](#_Toc86007038)

[Figure 5‑8 : Student Access Page 16](#_Toc86007039)

[Figure 5‑9 : Updating Student Details First Page 17](#_Toc86007040)

[Figure 5‑10 : Updating Details By Id/Form No. 17](#_Toc86007041)

[Figure 5‑11 : Updated Details Page 18](#_Toc86007042)

[Figure 5‑12 : Updating Details By Name 18](#_Toc86007043)

[Figure 5‑13 : Updated Details Page 19](#_Toc86007044)

[Figure 5‑14 : Deleting Student Details First Page 19](#_Toc86007045)

[Figure 5‑15 : Deleting Student Details By Id/Form No. 20](#_Toc86007046)

[Figure 5‑16 : Deleting Student Details By Name 20](#_Toc86007047)

[Figure 5‑17 : Exiting the Program 21](#_Toc86007048)

# List of Tables

# List of Abbreviations

|  |  |
| --- | --- |
| DBMS | Database Management System |
| GCC | GNU Compiler Collection |
| ID | Identity |
| IDE | Integrated Development Environment |

# 1. INTRODUCTION

My Hostel is a program which would help the hostel administration to keep their student’s data and information in digital form rather than in handwritten documents. It would also assist them to modify and delete the students record as per the need.

## 1.1 Background

Most of the students who come to study at another place by leaving their home like to stay in hostel. Hostel is considered as the second home for the students. Hostel administration has to keep records of their students. Keeping this thing in mind, we came with an idea to create a program to manage the records of the student so that it would help the hostel administration and improve their data management system. Tasks such as adding new student’s data, modifying existing student’s data, deleting student’s data, finding student’s data etc. are essential tasks for hostel administration. Our program would help the hostel administration to perform these tasks in efficient manner.

## 1.2 Motivation

Student record management is tedious process by manual way, since it involves work load time consumption. My Hostel (Student Database Management System) is a program developed for managing various activities in the hostel. For the past few years the numbers of hostels are increasing rapidly. With [Complete Management System](https://itsourcecode.com/free-projects/vb-net/complete-hotel-management-system/), we can easily manage the student details. [1]

## 1.3 Objectives

The main objectives of our project are listed below:

* To develop Hostel Students Data Management program and demonstrate it to the department faculties.
* To help hostel administration in maintaining their students record in organized way.

## 1.4 Scope

Our project can be very useful for hostel business entities. Since, our program is simple and easy to use, so even a non-technical person can also use our program to keep record of his/her hostel students. Owner of the hostel himself/herself can use our program and keep their students record as they are being carrying in handwritten document. There is no need of any specific person to be hired to run our program.

# 2. LITERATURE REVIEW

There are some programs which allows to perform tha tasks like adding the student info, view the added student, search the students, delete the added students, etc. Some of those programs are mentioned below:

## 2.1 Student Record System in C: Mini C Project

The mini-project “Student Record System Project in C” is a console application using the C programming language. In this console application, you can do basic Student Record tasks like adding the student info, view the added student, search the students, etc. This application is based on[file handling in C](https://aticleworld.com/file-handling-in-c/), where file-related function like [fopen](https://aticleworld.com/fopen-in-c/), [fread](https://aticleworld.com/fread-in-c/), [fwrite](https://aticleworld.com/fwrite-in-c/), etc are used. Good thing is that “Student Record System Project” is password-protected, so only authorized persons able to login to this application. The program is divided in different functions. Each function of the program extensively use in the [file handing function](https://aticleworld.com/file-handling-in-c/). [2]

## 2.2 Hostel Management System

Hostel Management System is based on a concept to maintain data of the students of a college living in different hostel premises. Before stepping into the main system a user has to pass through login system to get access, then only he/she can use the features of the system which includes Adding, Removing, Updating and Viewing student’s records of a particular hostel. Talking about the features of Hostel Management System, while entering student’s record he/she has to provide roll number, name, date of birth, age, city, phone number, email id, father’s name, father’s phone number, room number, and select hostel name. There are two methods to check a user’s account in detail i.e by roll number or by student’s name. He/she can view students of a particular hostel. The other listing record displays student’s name with roll number, address and contact detail. This system helps in easy data management of student in a hostel as it is not time-consuming. [3]

With our project, we have worked on similar techniques to build a new Student Record Management System mainly focused to hostel. Our project is user-friendly to the hostel administration and would help them in performing their record management task in efficient way.

# 3. METHODOLOGY

## 3.1 Admin Login

The program starts with an admin login page which asks for password. If user enters correct password then the program proceeds to the main menu. Otherwise, it asks to re-enter the password. Beep sound is played if the user enters incorrect password by using beep function*: Beep (600,500)*. Here 600 is the frequency of sound and 500 is the duration in mili-seconds for which the beep sound is played. [4]

Admin Login

Enter Password

check passwordd

False

True

Goto Main Menu

###### Figure 3‑ : Flowchart of Admin Login

## 3.2 Main Menu

After successful login, our program shows main menu of list of tasks admin can perform. Admin can do anything from the list by entering the corresponding index of the desired task to be performed. The list contains following actions which are described below.

1. [Student Registration](#_4.2.1_Student_Registration)
2. [View All Students](#_4.2.2_View_All)
3. [Search Student Details](#_4.2.3_Search_Student)
   1. Search by Name
   2. Search by Room No.
4. [Student Access](#_4.2.4_Student_Access)
5. [Modify Record](#_4.2.5_Update_Student)
6. [Delete record](#_4.2.6_Delete_Student)
7. [Exit](#_4.2.7_Exit)

The user can select required option by entering the corresponding index number. The above mentioned options and their working principle are elaborated below:

### 3.2.1 Student Registration

Student Registration adds the details of new student. The program asks the admin to enter following details of students:

* Form/Student No.
* Name
* Mobile No.
* School/College
* Guardian Name
* Guardian Mobile
* Date of Enrollment
* Room No

After entering every details, record of a new student is created with details entered by the admin. For creating and storing record of the students, use of structures and file handling will be done in our project.

### 3.2.2 View All Student Details

After admin select this action, the program shows details of every student entered during registration process in formatted view. The data stored during student registration is accessed and displayed on the console screen using file handling functions.

### 3.2.3 Search Student Details

This section helps admin to search student details using either room number or student name. First, the program reads all students data from file and then compares the room number or student name provided by the admin. If room number was given then, it shows the list of students with the provided room number. If student name was given then it shows the list of students whose name contains at least one substring same as user input name.

### 3.2.4 Students Access

Contact Details of a student is one of the frequently required data for hostel administration. Considering this in mind, our program contains a section for quickly accessing the contact details of every students. Admin can view the contact number of every students in list view through Student Access.

### 3.2.5 Update Student Details

After admin selects to update details of students, the program first asks for student name, then it shows the list of students whose name contains at least one substring same as user input name. Admin, then, have to provide serial number corresponding to the student whose details is needed to be updated. Then it asks for every update for every detail. If some detail is to be updated then new data should be entered otherwise left it blank. The program then stores all data from binary file and rewrite the file updating the student detail.

### 3.2.6 Delete Student Record

This section helps to delete complete details of a student. This is done by reading the datafile and storing the data in some temporary variable. The program asks the admin for student name whose record is to be deleted. Then it shows the admin a list of students with similar name and ask for serial number corresponding to the student whose record is to be deleted. Then it reads the whole datafile and stores the data in temporary variable temp and searches for student with the given name and writes the file with all student data except the student whose data is needed to be deleted.

### 3.2.7 Exit

The program can be terminated by performing this action. After working, the program needs to be exited. This option helps to exit the program without any error. All the data entered will be remained in the local drive of admin’s computer system and won’t be lost after terminating the program. It uses *exit(1)* to exit the program.

# 4. SYSTEM DESCRIPTION

## 4.1 Admin Login

The login page contains the greeting text “Hello!” and asks the admin to enter the password to access the main-menu. The password verification is done by using string handling function: *strcmp().* If the admin enters the same password as programmed in the system then he/she can be redirected to the main-menu page else the program asks to re-enter the correct password.

## 4.2 Main Menu

The main menu page prints all the tasks that can be performed by the admin along with corresponding index numbers for choosing a particular task to perform. The program takes index number as input from the admin and then uses *switch case* to perform various tasks using different user-defined functions created in the program which are as follows:

*case 1: reg\_stu()* for registering student details

*case 2: show\_stu()* for showing all student details

*case 3: search()* for searching student details

*case 4: stu\_access()* for viewing students’ contact details

*case 5: update()* for updating existing student details

*case 6: del()* for deleting existing student details

*case 7: exit(1)* for quiting the program

## 4.3 Student Registration

The student registration process is carried out by the user defined function: *void reg\_stu().* The registration details of the students is taken and stored in a structure named *STU stu.* The *gets()* and *scanf()* functions are used to take input from the admin. The program uses the library function *time(&now)* to fetch system time and store in date of enrollment. After taking all the inputs from the admin, the program displays the details of the registered student. All the details entered are stored in *data.bin* file using file handling functions. The details are further sorted by the program using user-defined function: *sort().*

## 4.4 Viewing All Students Details

All the details entered during registration process are displayed under this section. The details of all the students are displayed in the tabular form. This task is performed by user-defined function: *show\_stu().* The details are read from the *data.bin* file and displayed using file-handling function.

## 4.5 Searching Student Details

This task is performed by user-defined function: *search().* Admin can search the details either by student name or room no. If admin selects to search by student name then the function *search\_name(char name[])* is called and the name entered by the admin is passed as the argument. The entered name is searched in the *data.bin* file using file-handling function and if match is found then a list of matched student details is displayed. Similarly, if admin selects to search by student room no. then the function *search\_room(int room)* is called and the room no. entered by the admin is passed as the argument. The entered room no. is searched in the *data.bin* file using file-handling function and if match is found then a list of matched student details is displayed.

If match is not found for entered Name or Room No. then a sorry message is displayed saying *“No Student Found with (entered) Name or Room No.”*

## 4.6 Viewing Student Access

The contact information of all the students are displayed under this section. This task is performed by the user-defined function *stu\_access().* The details like student’s name and mobile number are read from the *data.bin* file using file-handling function and displayed on the screen in tabular form.

## 4.7 Updating Student Details

The details of existing student can be updated/modified under this section and this task is performed by user-defined function: *update().* The admin is asked whether he/she knows the ID or Form No. of the student whose details are to be updated. If he/she knows the student’s ID or Form No. then further the program asks to enter the ID or Form No. The program then searches for the entered ID number in *data.bin* file and if ID number is found then the function: *update\_id(int id)* is called and the entered ID number is passed as argument. The *update\_id()* functions helps the admin to update the details of student. The admin can choose which details to update and which to not. The updated details are then stored in *data.bin* file.

If the admin doesn’t know the ID or Form No. then he/she is asked to enter the name of the student whose details are to be updated/modified. Then program searches for the entered name in *data.bin* file and if match is found then the details are displayed and then the admin is asked to enter the ID number of that student then *update\_id(int id)* is called and the entered ID number is passed as argument.

If match is not found for entered ID/Form No. or Name then a sorry message is displayed saying *“No Student Found with (entered) Id or Name.”*

## 4.8 Deleting Student Details

The details of existing student can be deleted under this section and this task is performed by user-defined function: *del().* The admin is asked whether he/she knows the ID or Form No. of the student whose details are to be deleted. If he/she knows the student’s ID or Form No. then further the program asks to enter the ID or Form No. The program then searches for the entered ID number in *data.bin* file and if ID number is found then the function: *del\_id(int id)* is called and the entered ID number is passed as argument. The *del\_id()* functions deletes the details of student with the given ID number from the *data.bin* file.

If the admin doesn’t know the ID or Form No. then he/she is asked to enter the name of the student whose details are to be deleted. Then program searches for the entered name in *data.bin* file and if match is found then the details are displayed and then the admin is asked to enter the ID number of that student then *del\_id(int id)* is called and the entered ID number is passed as argument.

If match is not found for entered ID/Form No. or Name then a sorry message is displayed saying *“No Student Found with (entered) Id or Name.”*

## 4.9 Exiting The Program

The program is terminated using the function: *exit(1).* The program also displays exit message *“\*\*\*Thank You For Using My Hostel\*\*\*”.*

# 5. RESULTS AND ANALYSIS

## 5.1 Login Page



###### Figure 5‑ : Login Page

## 5.2 Main Menu



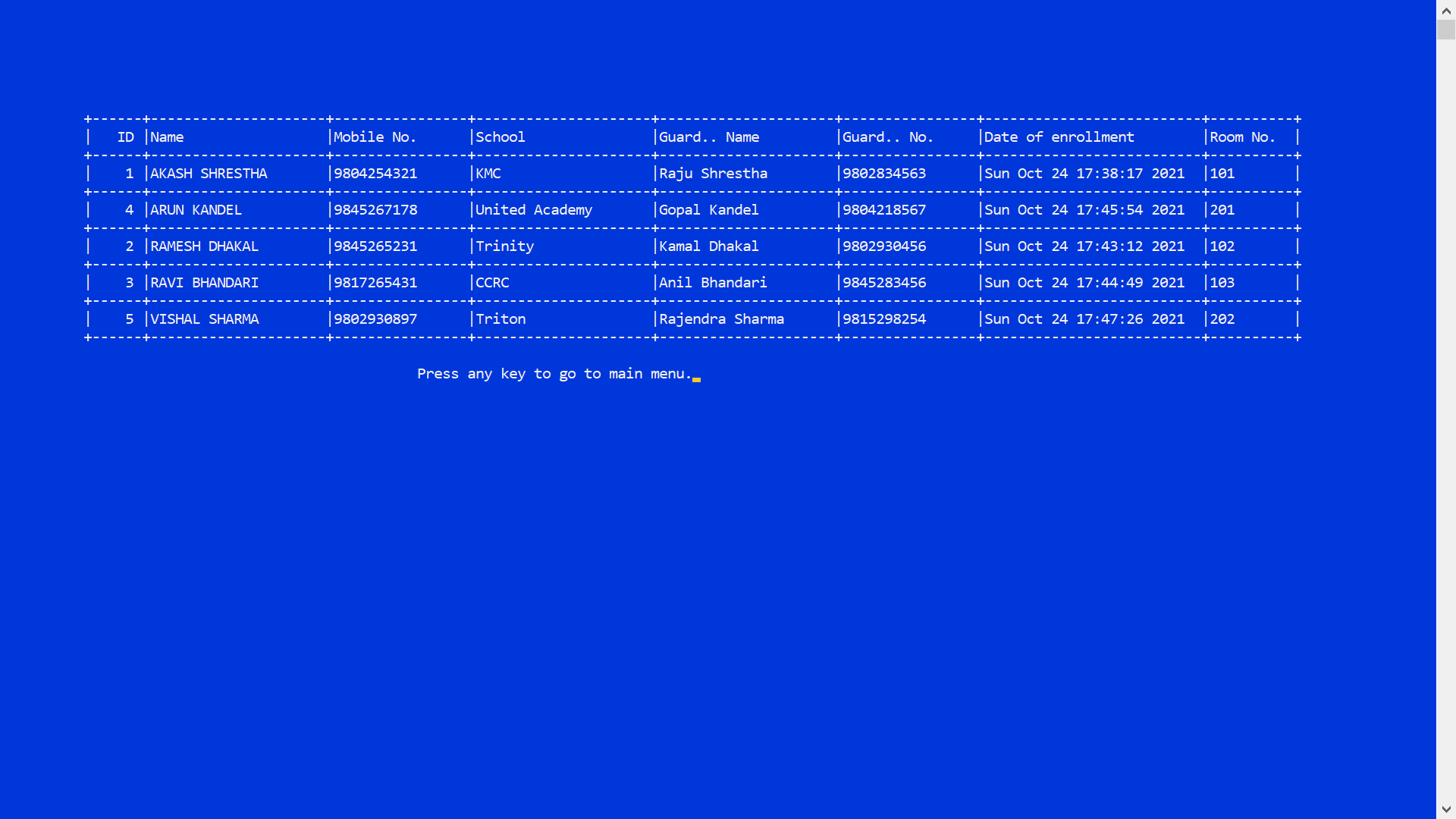
###### Figure 5‑ : Main Menu Page

## 5.3 Student Registration



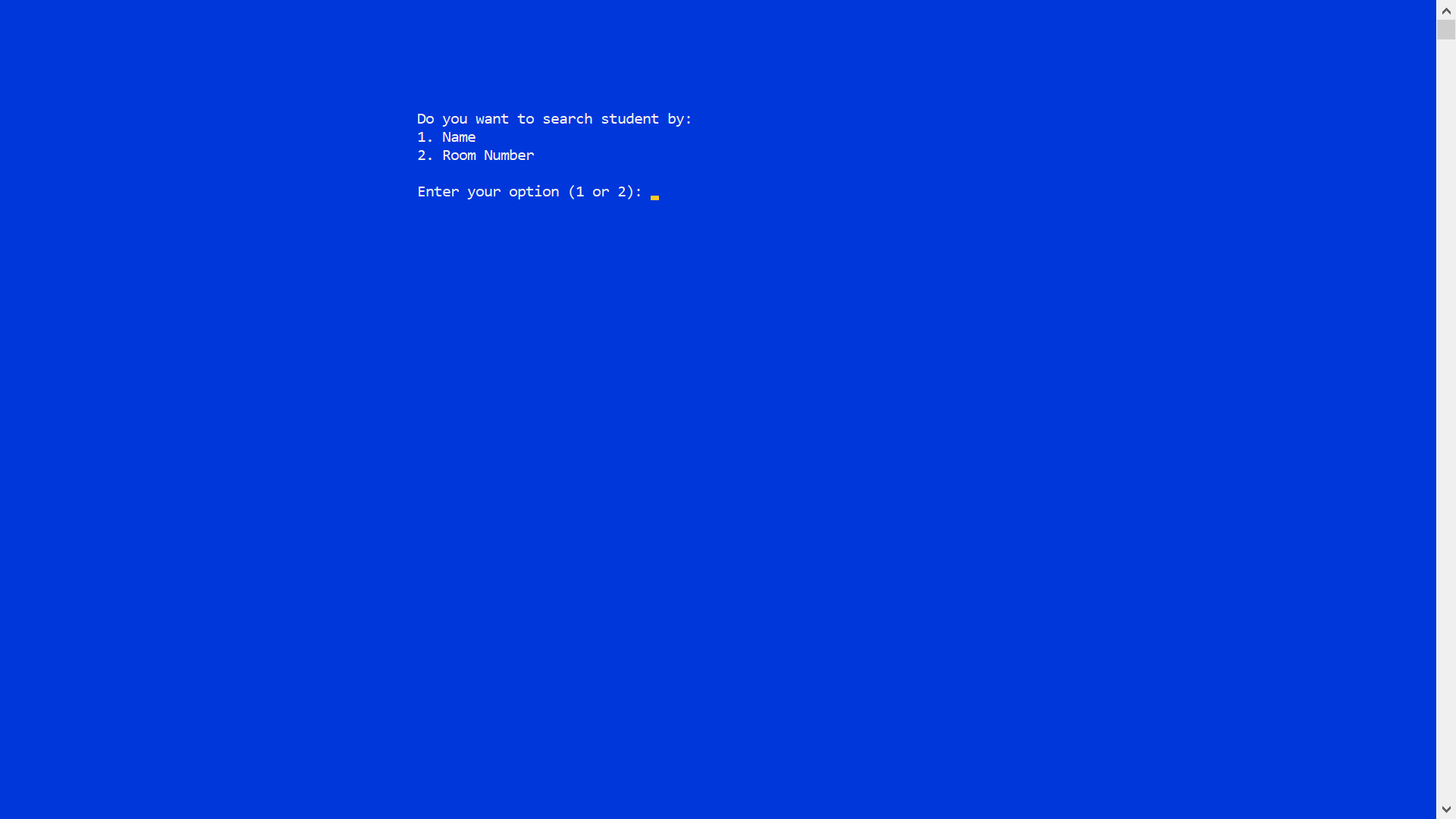
###### Figure 5‑ : Student Registration Page

## 5.4 All Student Details List



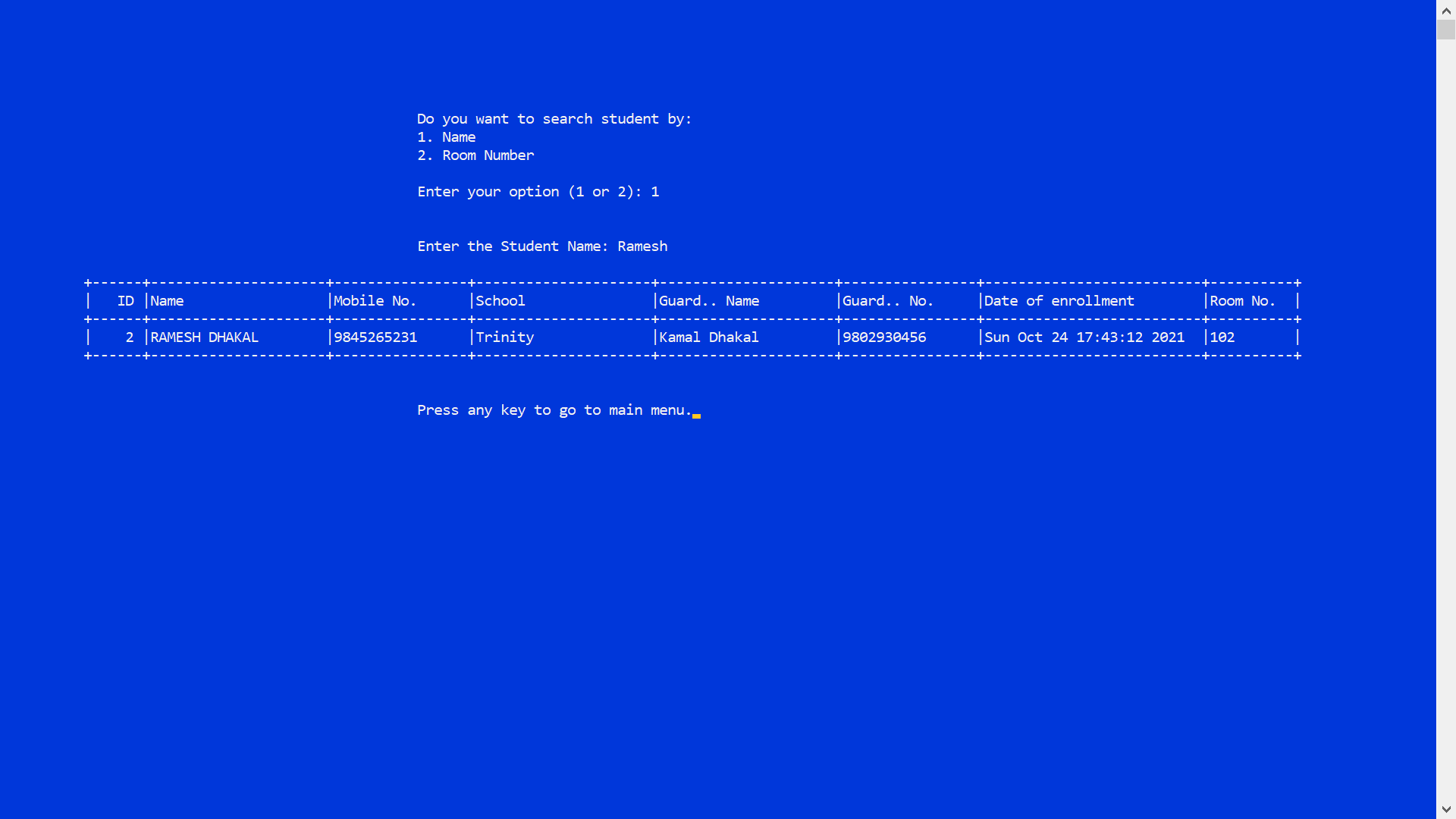
###### Figure 5‑ : All Student Details Page

## 5.5 Search Student Details



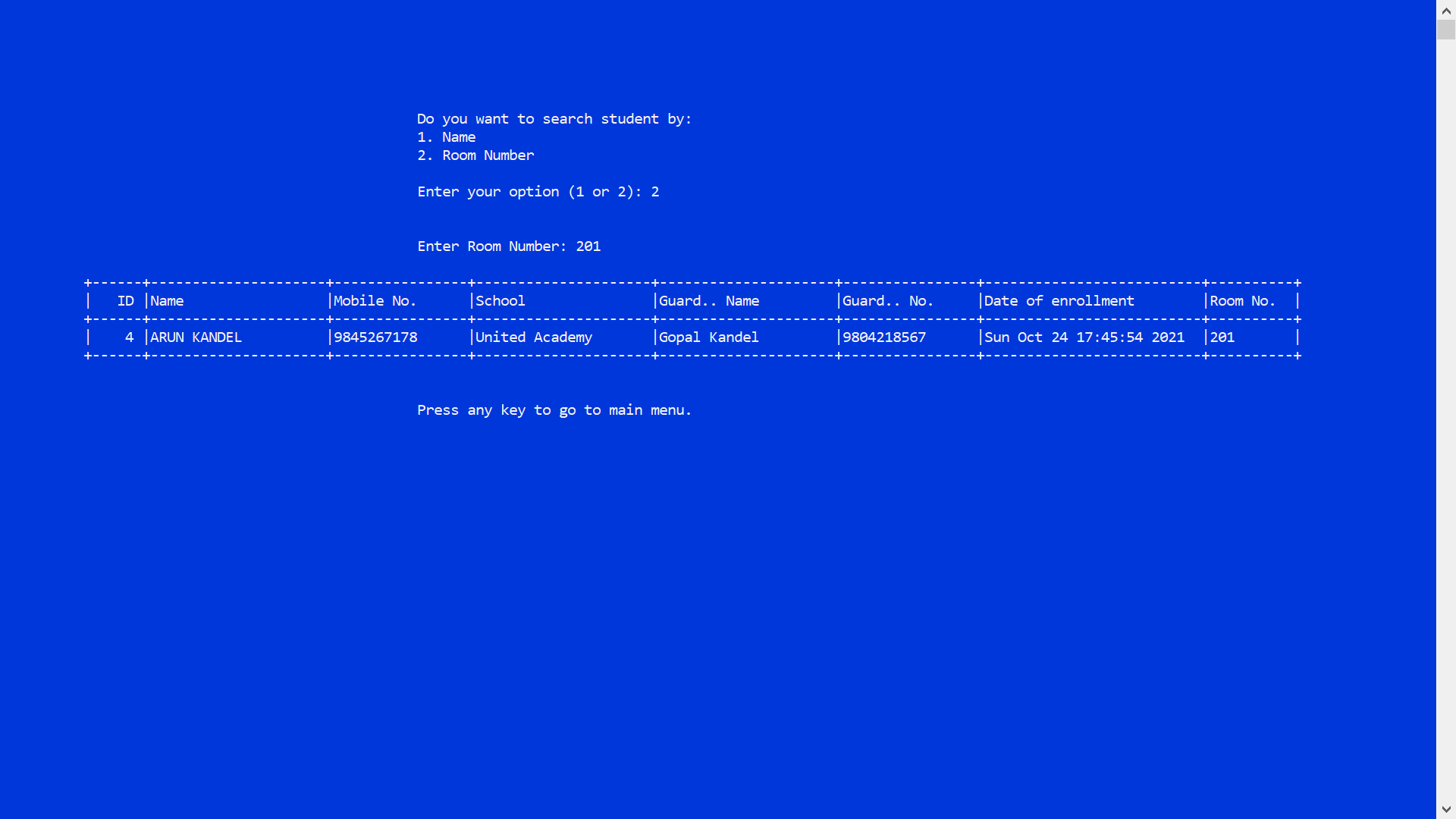
###### Figure 5‑ : Search Student Details First Page

### 5.5.1 Search By Student Name



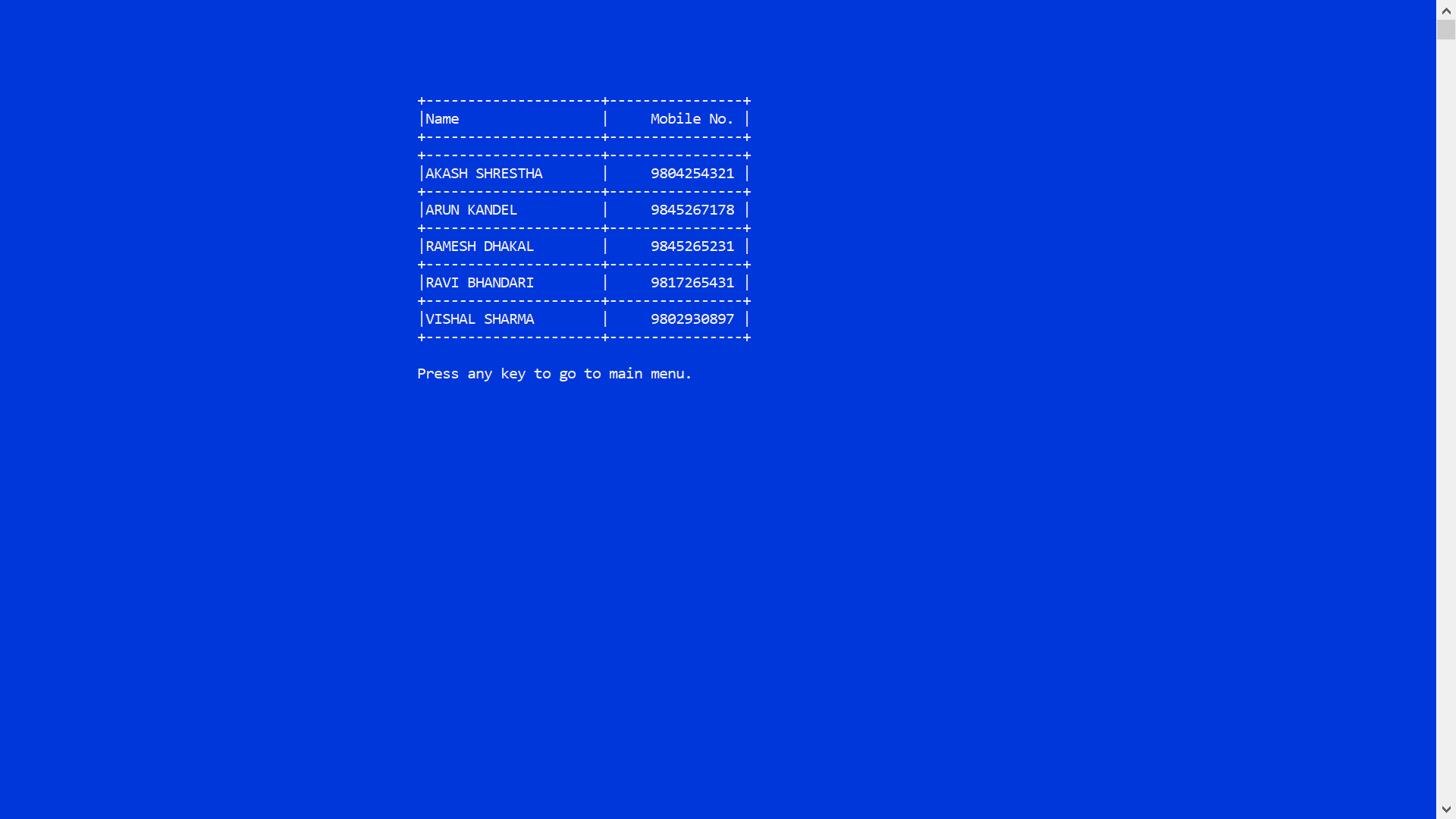
###### Figure 5‑ : Search By Student Name

### 5.5.2 Search By Room No.



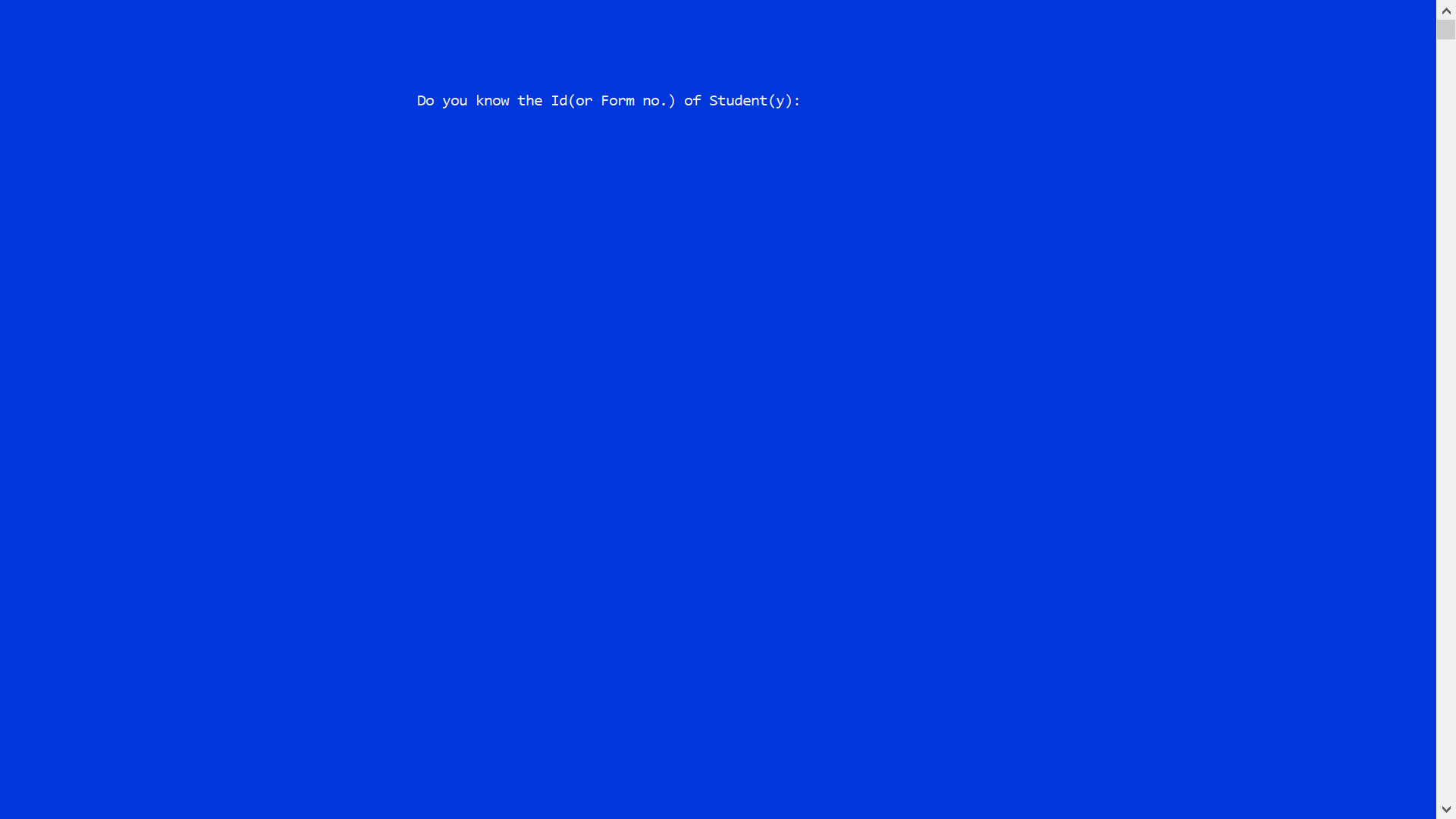
###### Figure 5‑ : Search By Student Room No.

## 5.6 Student Access



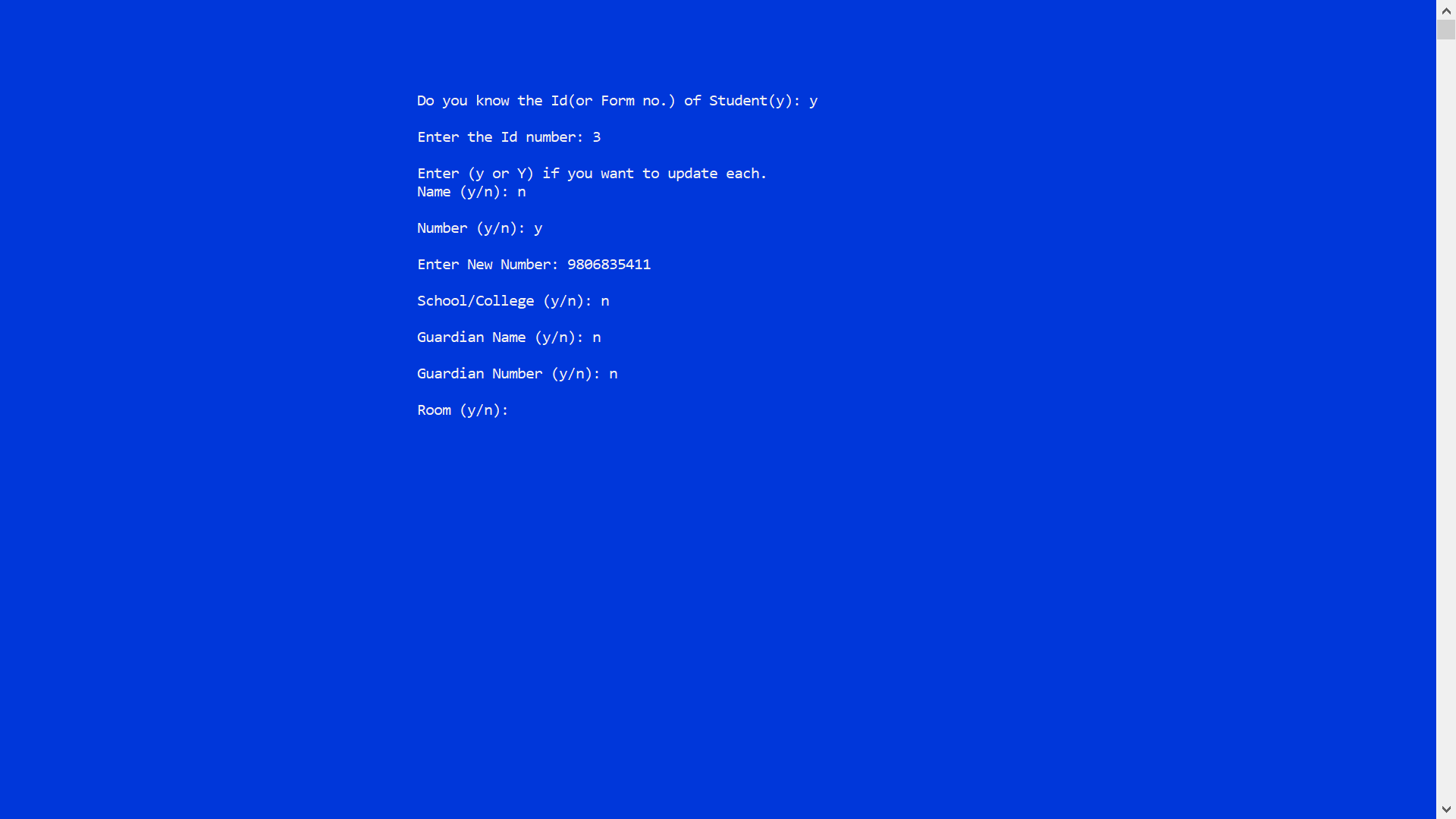
###### Figure 5‑ : Student Access Page

## 5.7 Updating Student Details



###### Figure 5‑ : Updating Student Details First Page

### 5.7.1 Updating Details By Id/Form No.

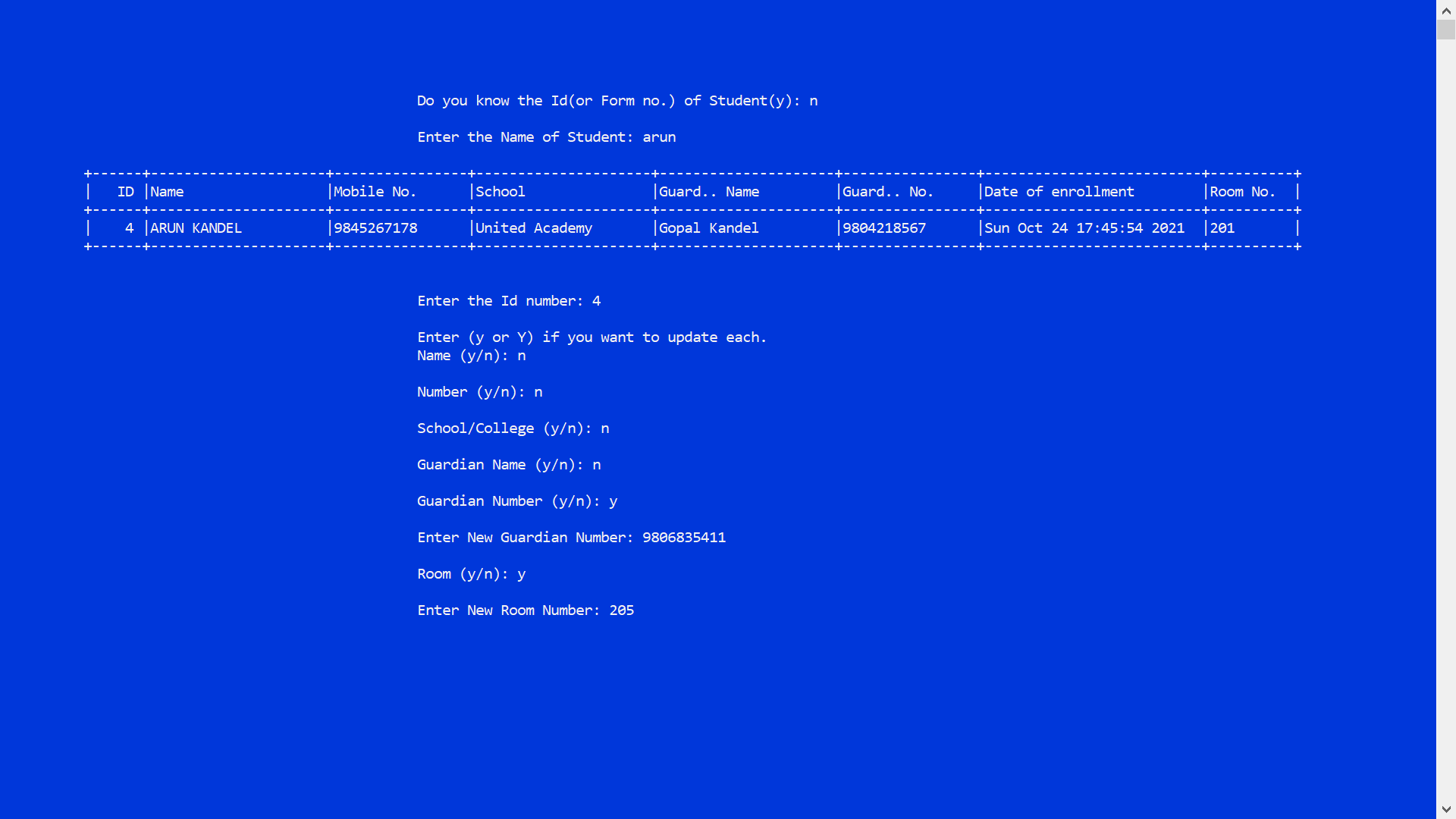


###### Figure 5‑ : Updating Details By Id/Form No.



###### Figure 5‑ : Updated Details Page

### 5.7.2 Updating Details By Name

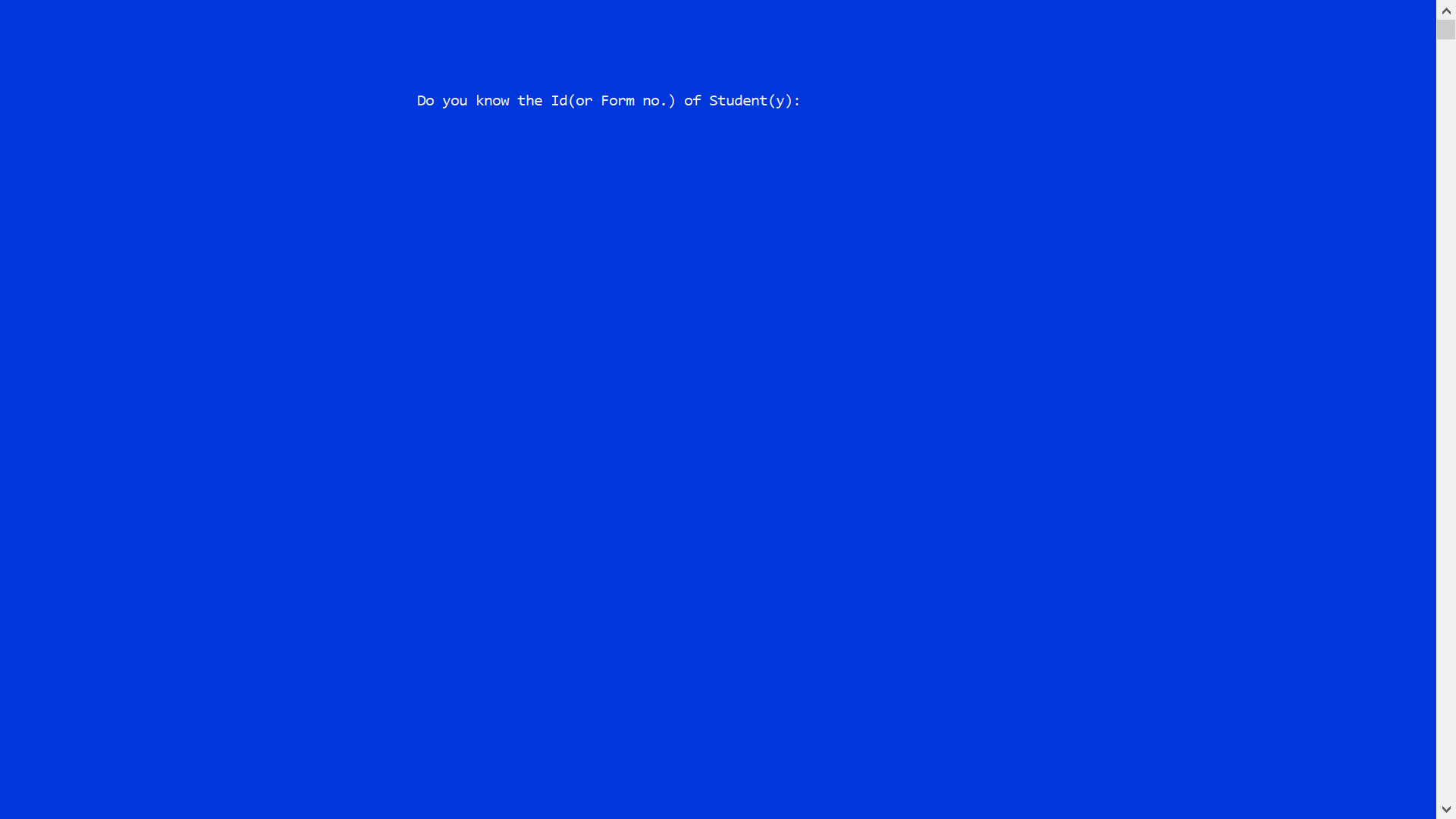


###### Figure 5‑ : Updating Details By Name



###### Figure 5‑ : Updated Details Page

## 5.8 Deleting Student Details



###### Figure 5‑ : Deleting Student Details First Page

### 5.8.1 Deleting Details By Id/Form No.



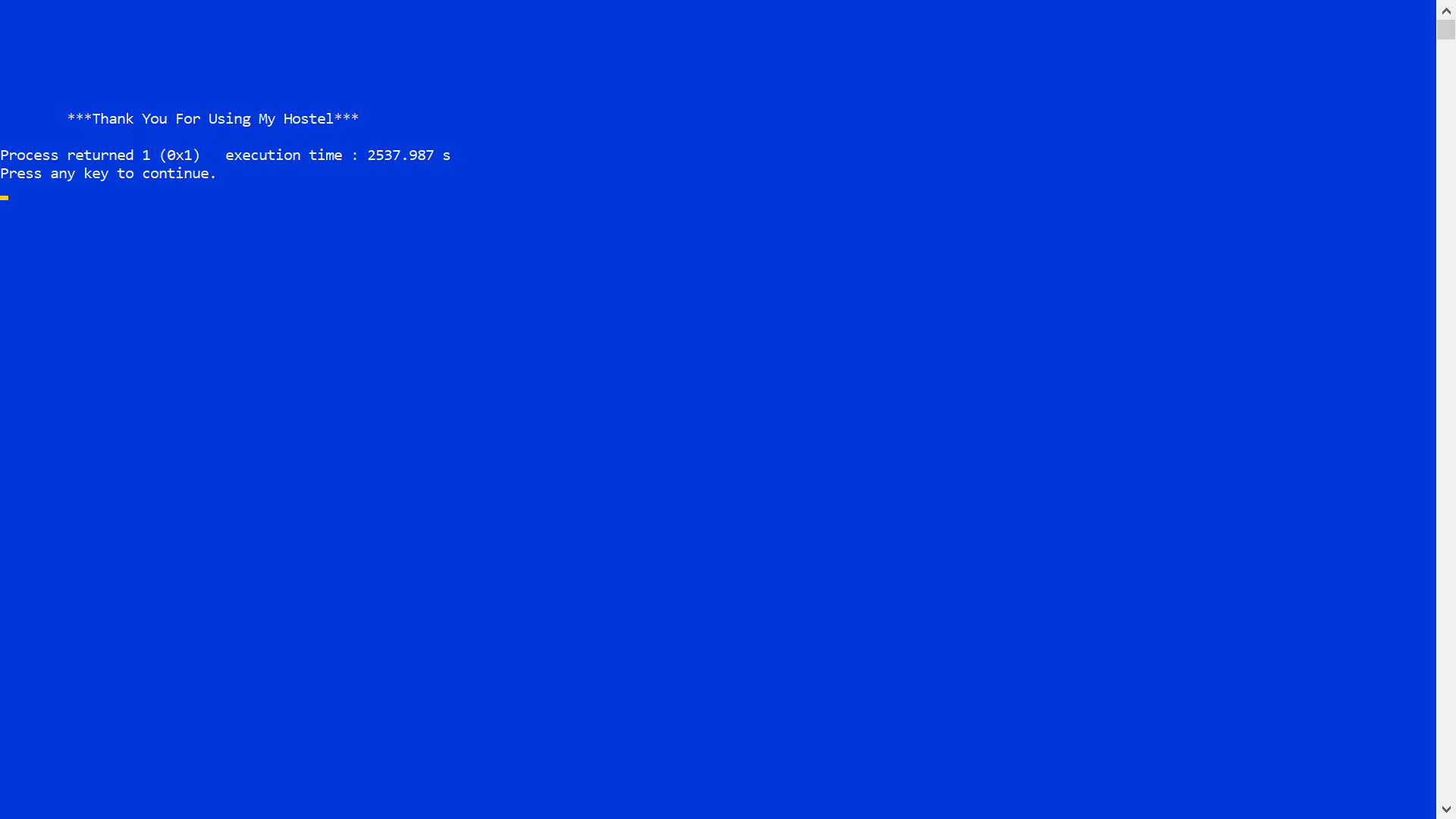
###### Figure 5‑ : Deleting Details By Id/Form No.

### 5.8.2 Deleting Details By Name



###### Figure 5‑ : Deleting Details By Name

## 5.9 Exiting the Program



###### Figure 5‑ : Exiting the Program

# 6. CONCLUSION AND FUTURE ENHANCEMENT

## 6.1 Conclusion

My Hostel is a program developed to handle data and records of students and it is based on Database Management System. This program would help the hostel administration to keep their student’s data and information in digital form rather than in handwritten documents. It would also assist them to modify and delete the students record as per the need.

## 6.2 Limitations

* Our program currently doesn’t support the feature to change the password of the program.
* Our program can only perform the basic data managing tasks of the hostel administration.
* Our program can be viewed in fullscreen only and no other tasks can be performed, in order to come out of the fullscreen program ALT + Esc is to be pressed.

## 6.3 Future Enhancement

* Our program can be further modified to support the feature to change the password to enter the main-menu.
* Our program can be further modified to provide facility to hostel administration to keep records of their staff members as well.
* Our program can be further modified to keep hostel payment and due records of the students.
* Our program can be further modified to keep records of students under different block buildings of hostel (if any).
* Our program can be further modfied to give better user experience to the user.

# 7. APPENDICES

## Appendix A: Gantt Chart

###### Table : Gantt Chart with Project Activities and Timeline

# References

|  |  |
| --- | --- |
| [1] | H. M. S. D. Design, "itsourcecode.com," [Online]. Available: https://itsourcecode.com/free-projects/database-design-projects/database-design-for-hostel-management-system/. [Accessed 25 July 2021]. |
| [2] | Almendra. [Online]. Available: https://aticleworld.com/student-record-system-project-in-c/. [Accessed 26 July 2021]. |
| [3] | F. Ros. [Online]. Available: https://code-projects.org/hostel-management-system-in-c-programming-with-source-code/. [Accessed 26 July 2021]. |
| [4] | "Geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/beep-function-in-c-with-examples/. [Accessed 28 July 2021]. |