****

**Green University of Bangladesh**

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

**Faculty of Sciences and Engineering**

**Semester: (Spring, Year:2021), B.Sc. in CSE (Day/Eve)**

**Course Title: Operating System Lab**

**Course Code: CSE 310 Section: DA**

**Lab Project Name: Student Database System**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Jahid Hassan Santo | 201002463 |

**Submission Date : 16/05/2022**

**Course Teacher’s Name : Mr. Jahidul Islam**

|  |
| --- |
| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |

Table of Contents

[Chapter 1 3](#_Toc103636630)

[Introduction 3](#_Toc103636631)

[1.1 Introduction 3](#_Toc103636632)

[1.2 Design Goals/Objective 3](#_Toc103636633)

[Chapter 2 4](#_Toc103636634)

[Design/Development/Algorithms/Implementation of the Project 4](#_Toc103636635)

[2.1 Design and Development Software or Kit. 4](#_Toc103636636)

[2.2 Project Design/Algorithms. 5](#_Toc103636637)

[2.3 Project Implementation 6](#_Toc103636638)

[Chapter 3 12](#_Toc103636639)

[Performance Evaluation 12](#_Toc103636640)

[3.1 Simulation Environment/Simulation Procedure 12](#_Toc103636641)

[3.2 Results and Discussions 24](#_Toc103636642)

[3.2.1 Results 24](#_Toc103636643)

[3.2.2 Discussions 24](#_Toc103636644)

[Chapter 4 25](#_Toc103636645)

[Conclusion 25](#_Toc103636646)

[4.1 Introduction 25](#_Toc103636647)

[4.2 Practical Implications 25](#_Toc103636648)

[4.3 Scope of Future work 25](#_Toc103636649)

[References 26](#_Toc103636650)

# Chapter 1

Introduction

## Introduction

Using this database system, we understand that there should be a database process. In telecommunications, computing, and information architecture, a data bank or databank is a repository of information on one or more topics - a database - organized in such a way as to facilitate the retrieval of local or remote information and to process many continuous queries over a long period of time. In this database, we have tried to use a considerable amount of basic information that a student database system can maintain. Here we create four sections and take some valuable features. We also maintain user safety here.

## Design Goals/Objective

* Organize in a way that facilitates the retrieval of local or remote information and can process many continuous queries over a long period of time.
* To ensure user safety.
* Store student information and database information in a way that is easy to access.
* Keep track of student results and location.

# Chapter 2

Design/Development/Algorithms/Implementation of the Project

## Design and Development Software or Kit.

Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop.

Linux has been around since the mid-1990s and has since reached a user-base that spans the globe. Linux is actually everywhere: It’s in your phones, your thermostats, in your cars, refrigerators, Roku devices, and televisions. It also runs most of the Internet, all of the world’s top 500 supercomputers, and the world’s stock exchanges.

Linux is also distributed under an open-source license. Open source follows these key tenants:

* The freedom to run the program, for any purpose.
* The freedom to study how the program works, and change it to make it do what you wish.
* The freedom to redistribute copies so you can help your neighbor.
* The freedom to distribute copies of your modified versions to others.

These points are crucial to understanding the community that works together to create the Linux platform. Without a doubt, Linux is an operating system that is “by the people, for the people”. These tenants are also a main factor in why many people choose Linux. It’s about freedom and freedom of use and freedom of choice. You must have

* A PC or Laptop
* Linux Operating System.
* Minimum 1 GB of hard disk space
* 1024×768 or greater screen resolution.

Completing this project we have to install a new package named ‘Whiptail’. Below is instruction for install whiptail in Ubuntu.

Quick Install Instructions of whiptail on Ubuntu Server. It’s Super Easy! simply click on Copy button to copy the command and paste into your command line terminal using built-in APT package manager.

See below for quick step by step instructions of SSH commands, Copy/Paste to avoid miss-spelling or accidently installing a different package.

**Quick Install Steps:**

* Step 1

sudo apt-get update -y

* Step 2

sudo apt-get install -y whiptail

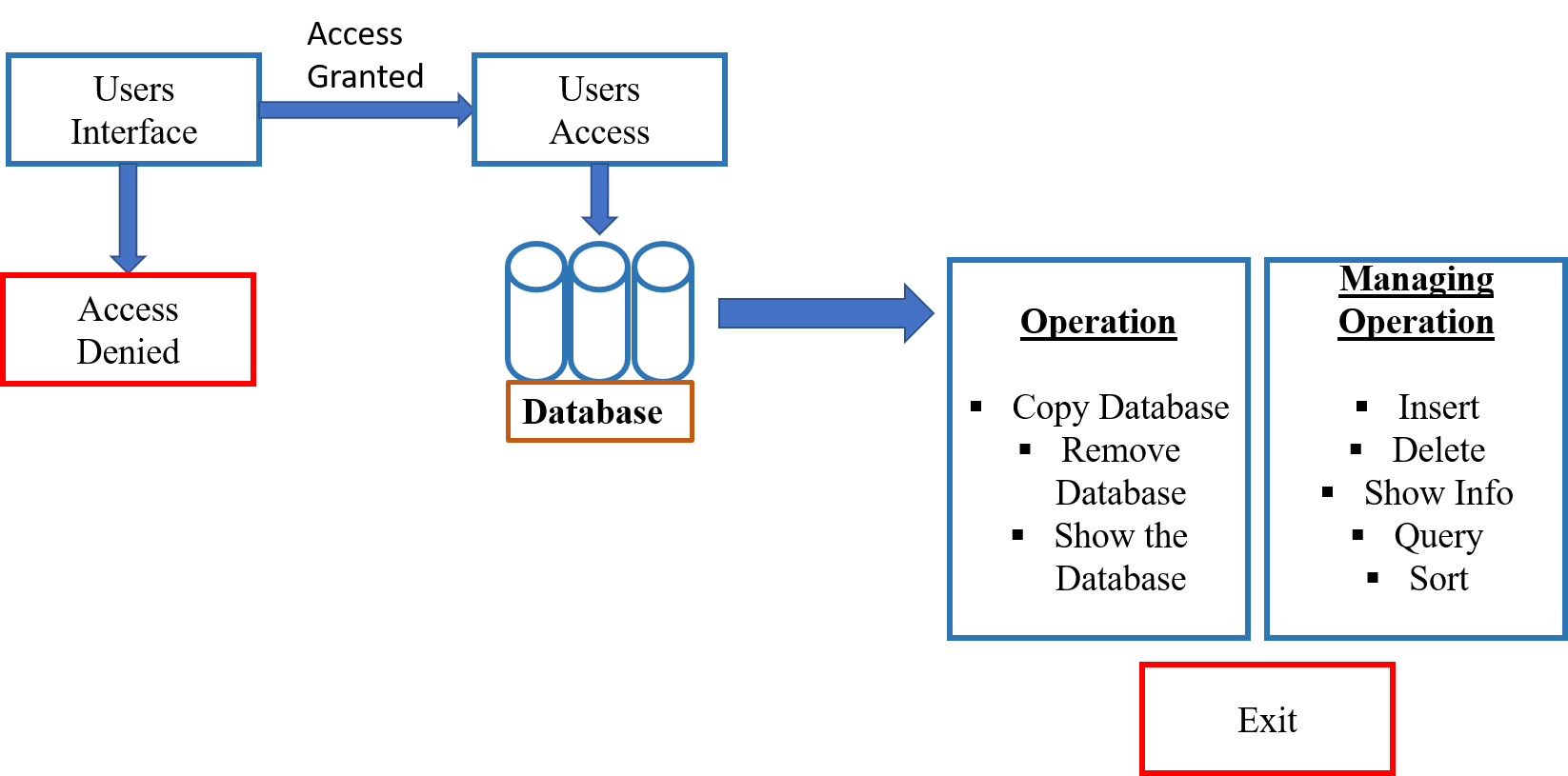
* Step 3

Check the system logs to confirm that there are no related errors. You can use ZoomAdmin to check the logs, manager servers, host multiple websites and apps on your servers and more. The apps run in docker containers, to learn more

see ZoomAdmin Features for list of features and demo videos. And you can start with the Free Plan.

Execute the commands above step by step. You can simply hit the copy button to copy the command and paste into the command line interface.

## Project Design/Algorithms.



# Chapter 3

Performance Evaluation

## Simulation Environment/Simulation Procedure

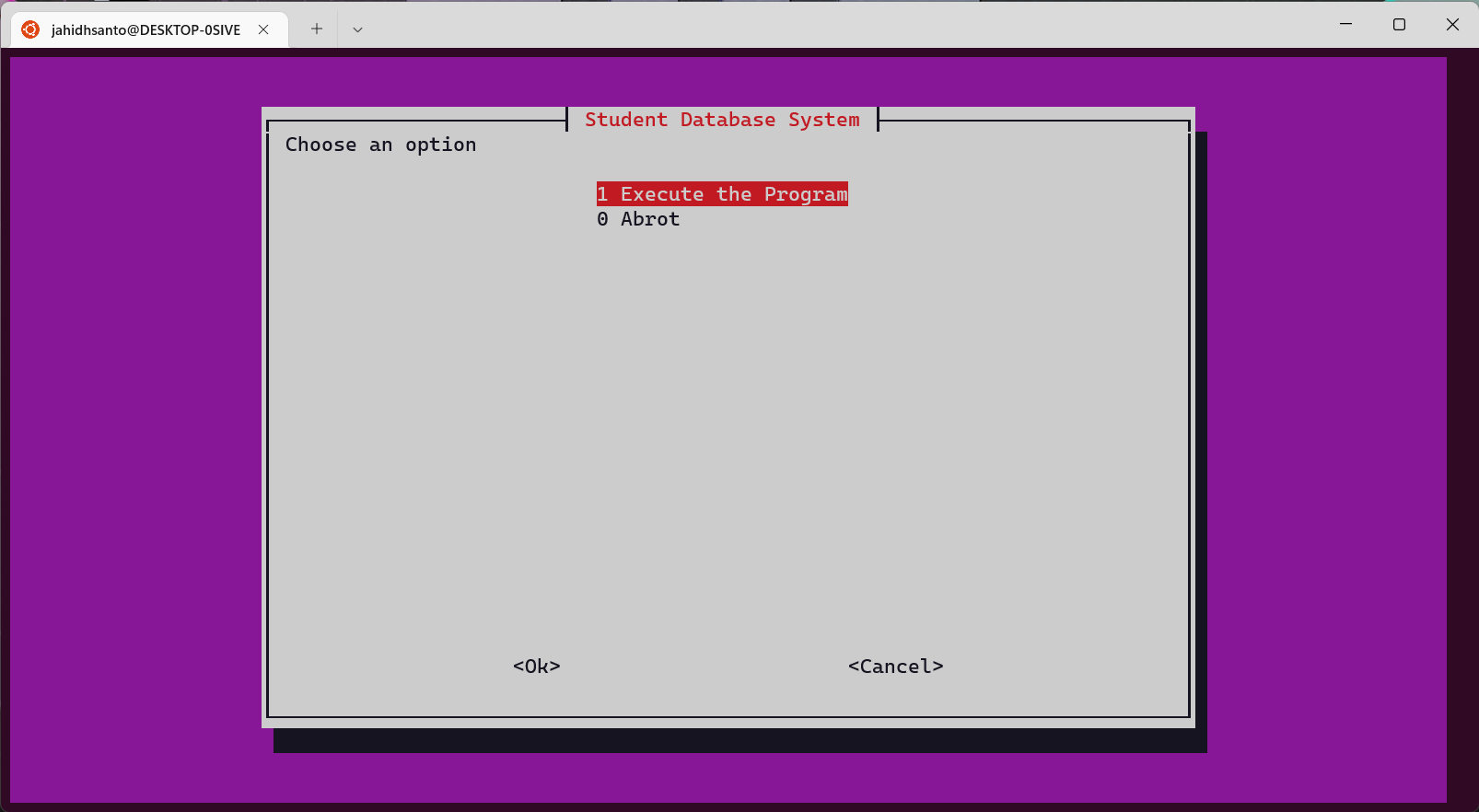
This is the user’s interface where two option is mention for users. ‘1’ for execute the program and ‘0’ for abort.

Figure 3.1.1: Users Interface

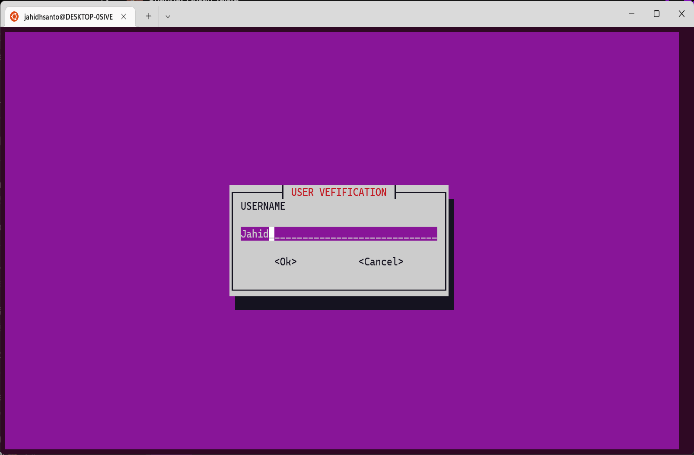
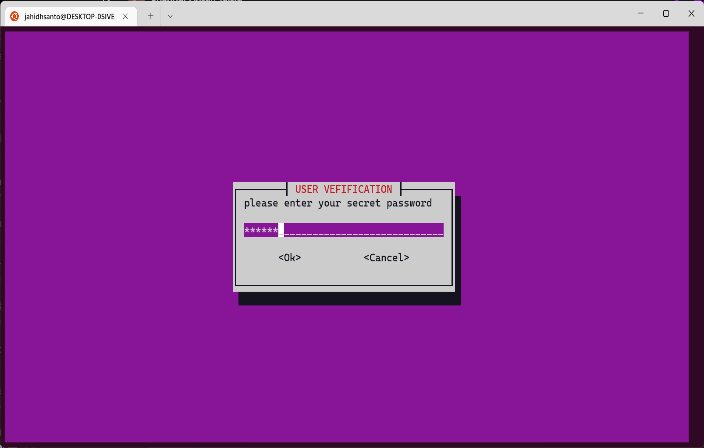
This is for users log in we have to input the user’s id and password.

Figure 3.1.2: Users Log In

This is the main page of our project or starting page.

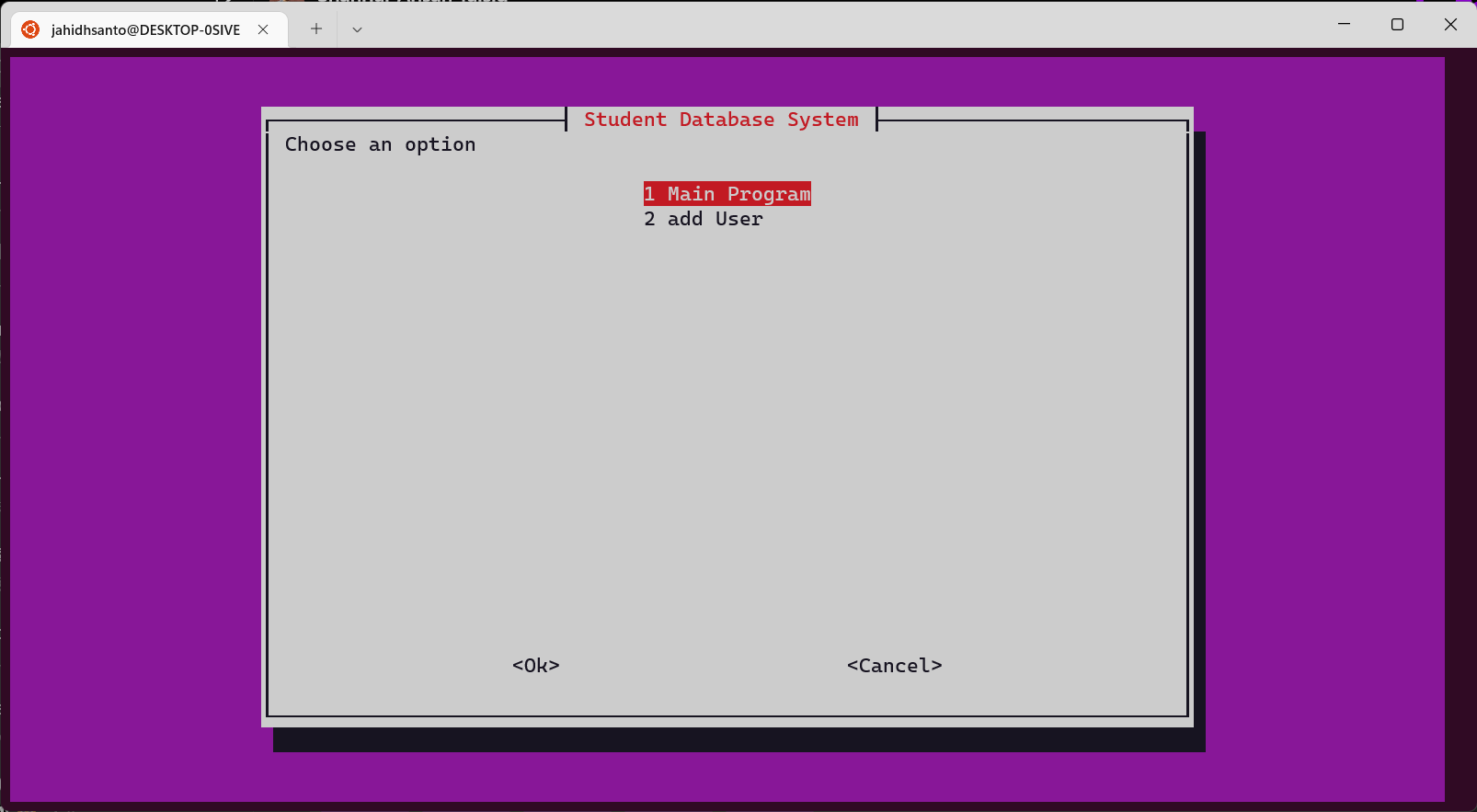


Figure 3.1.3: Entering into the Program

Here 6 option for every option different operation will be execute.

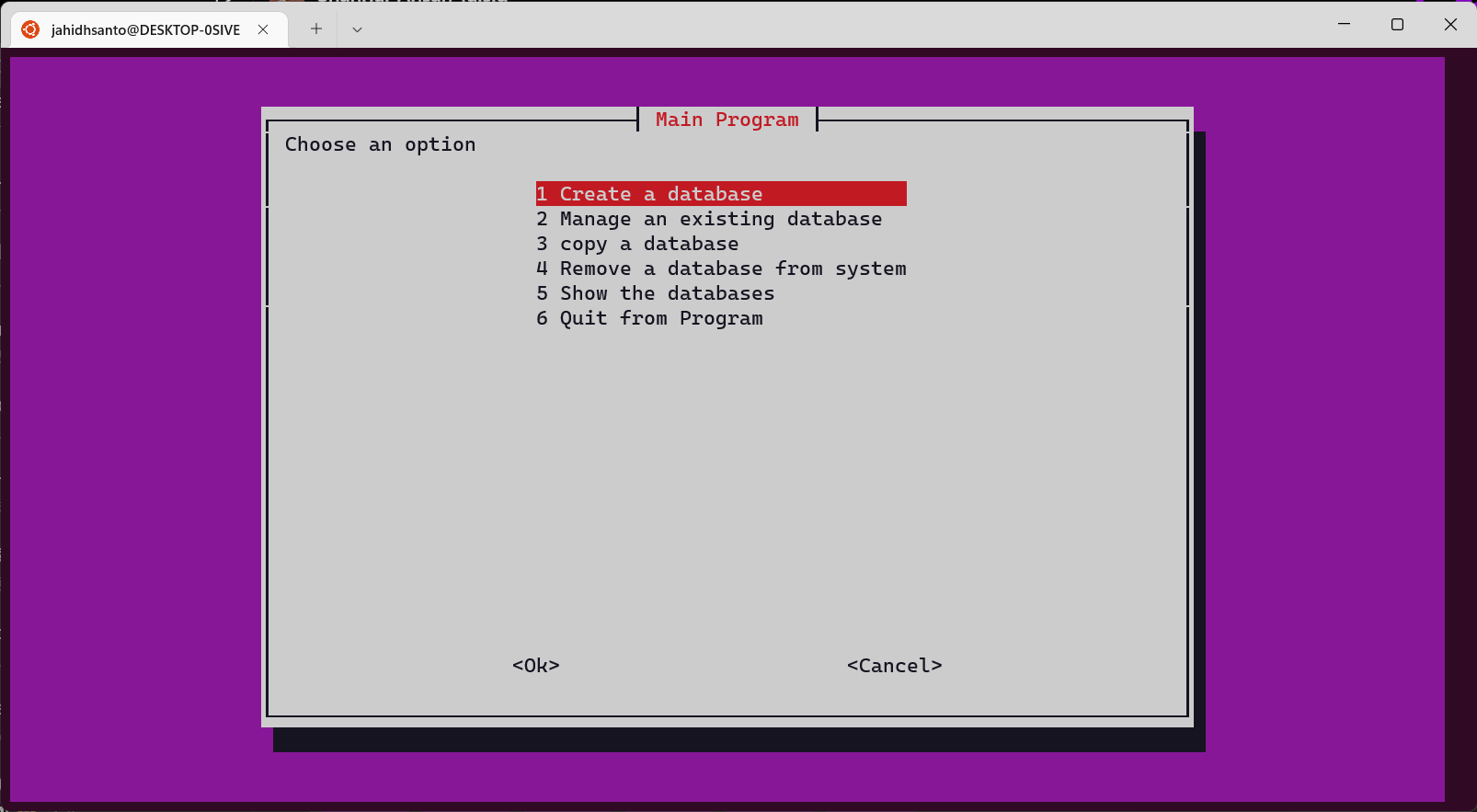


Figure3.1.4: Operation

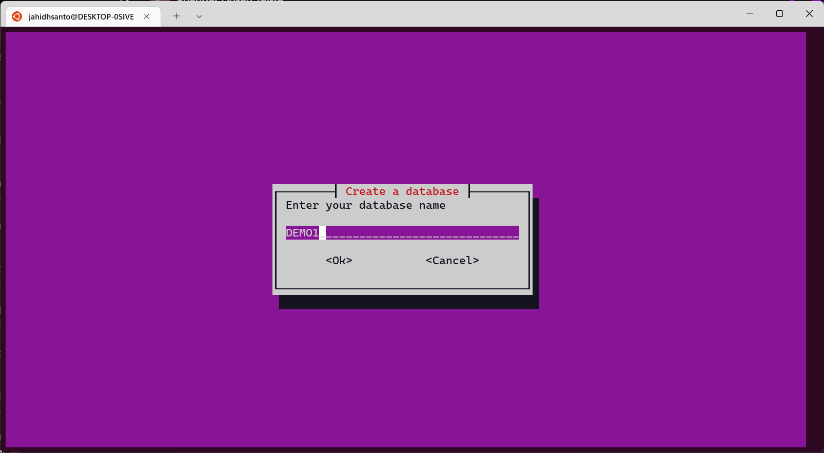
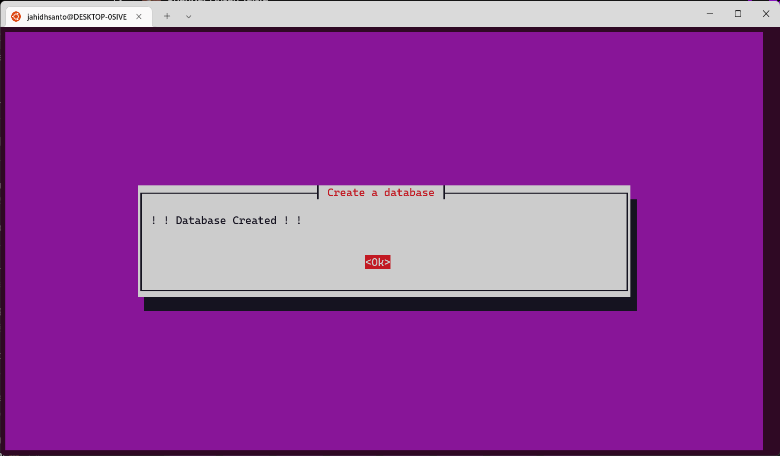
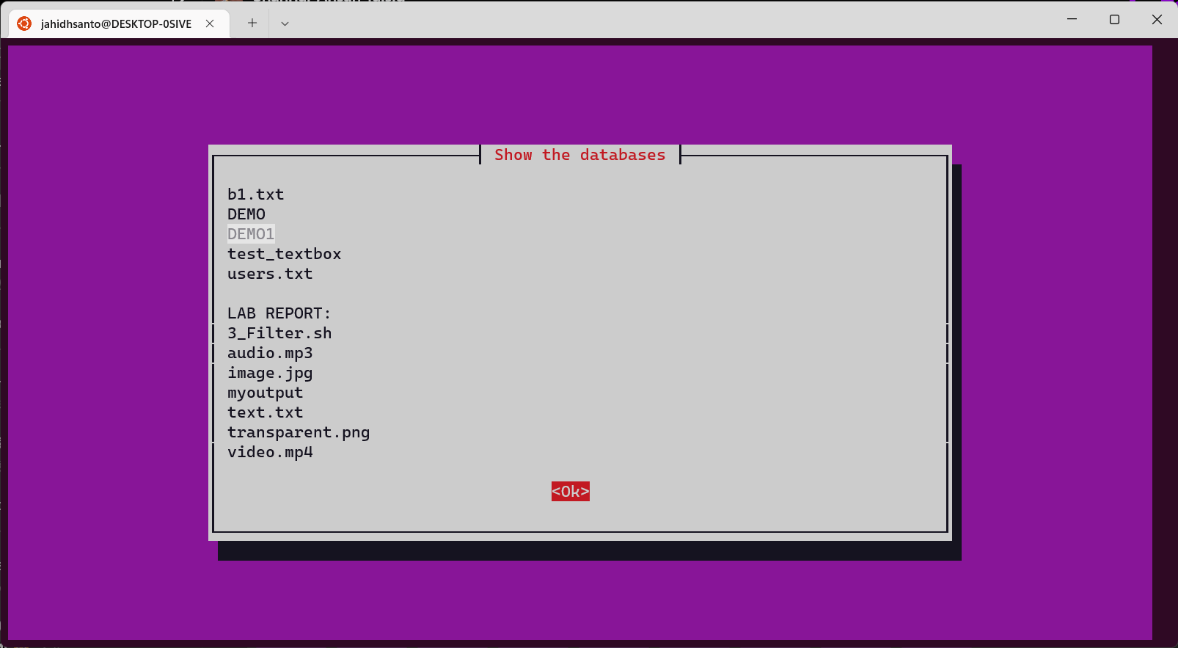
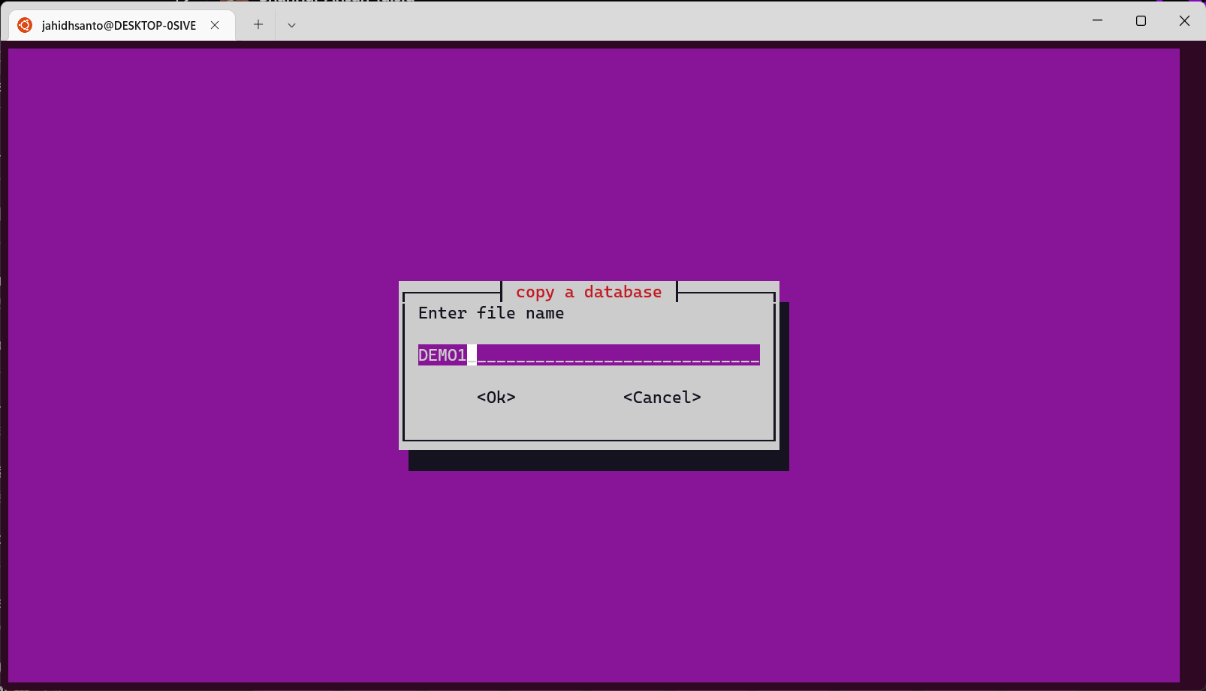
Here we create a database.

Figure 3.1.5: Create Database

Here we show the database are created



Figure3.1.6: Show the Database

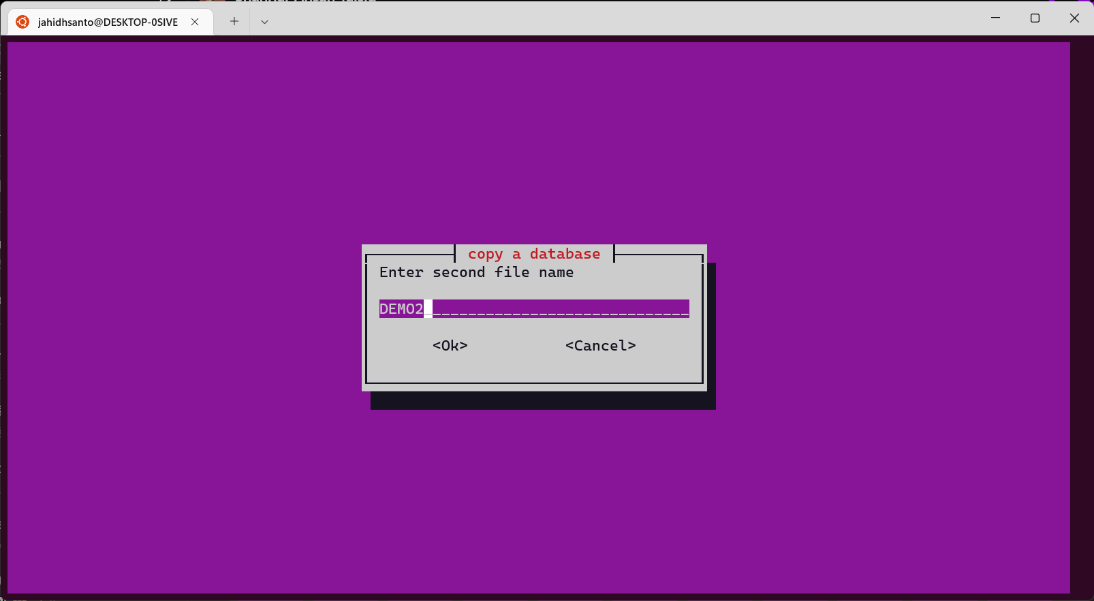


Figure 3.1.7: Copy a Database

Here we remove a database.

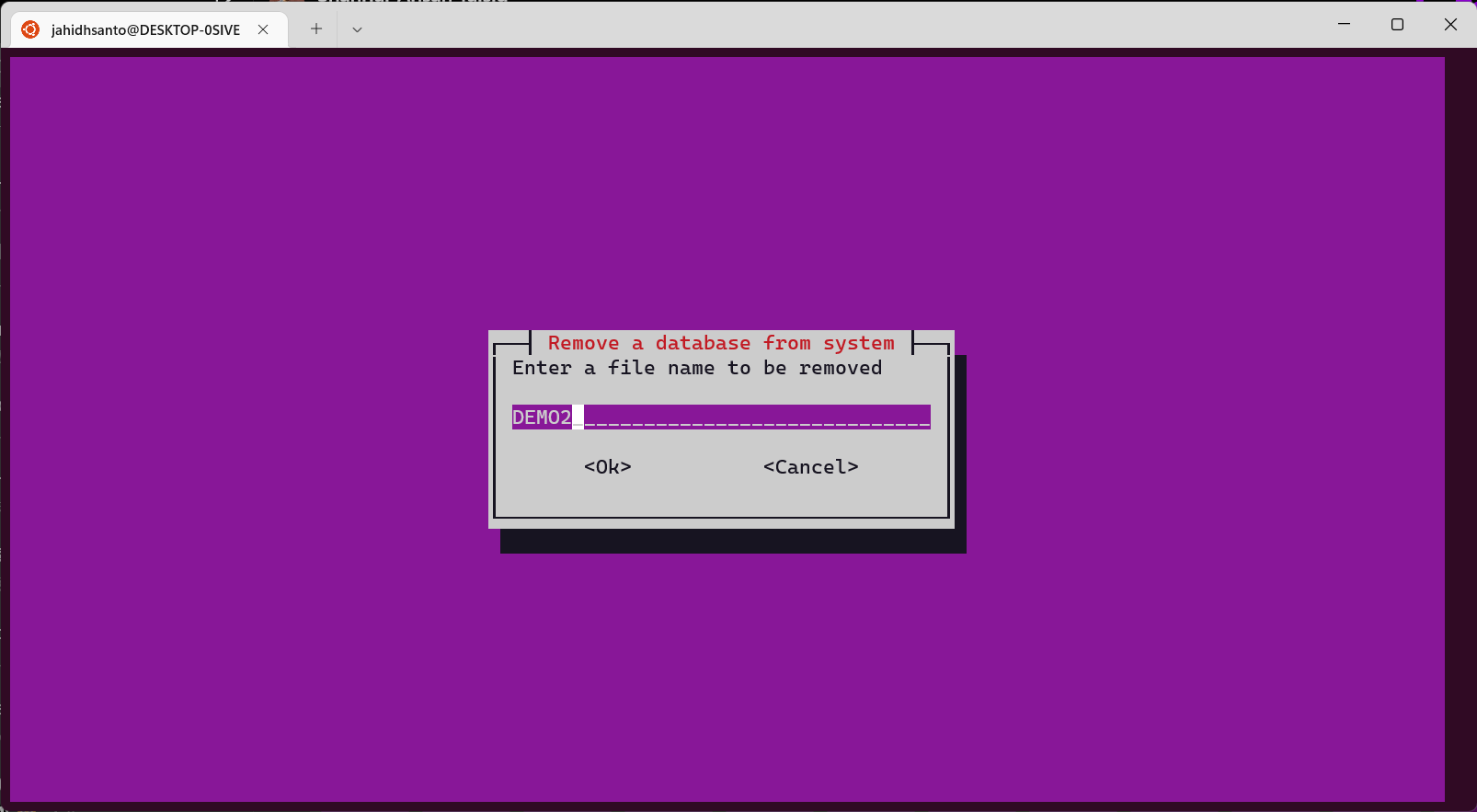
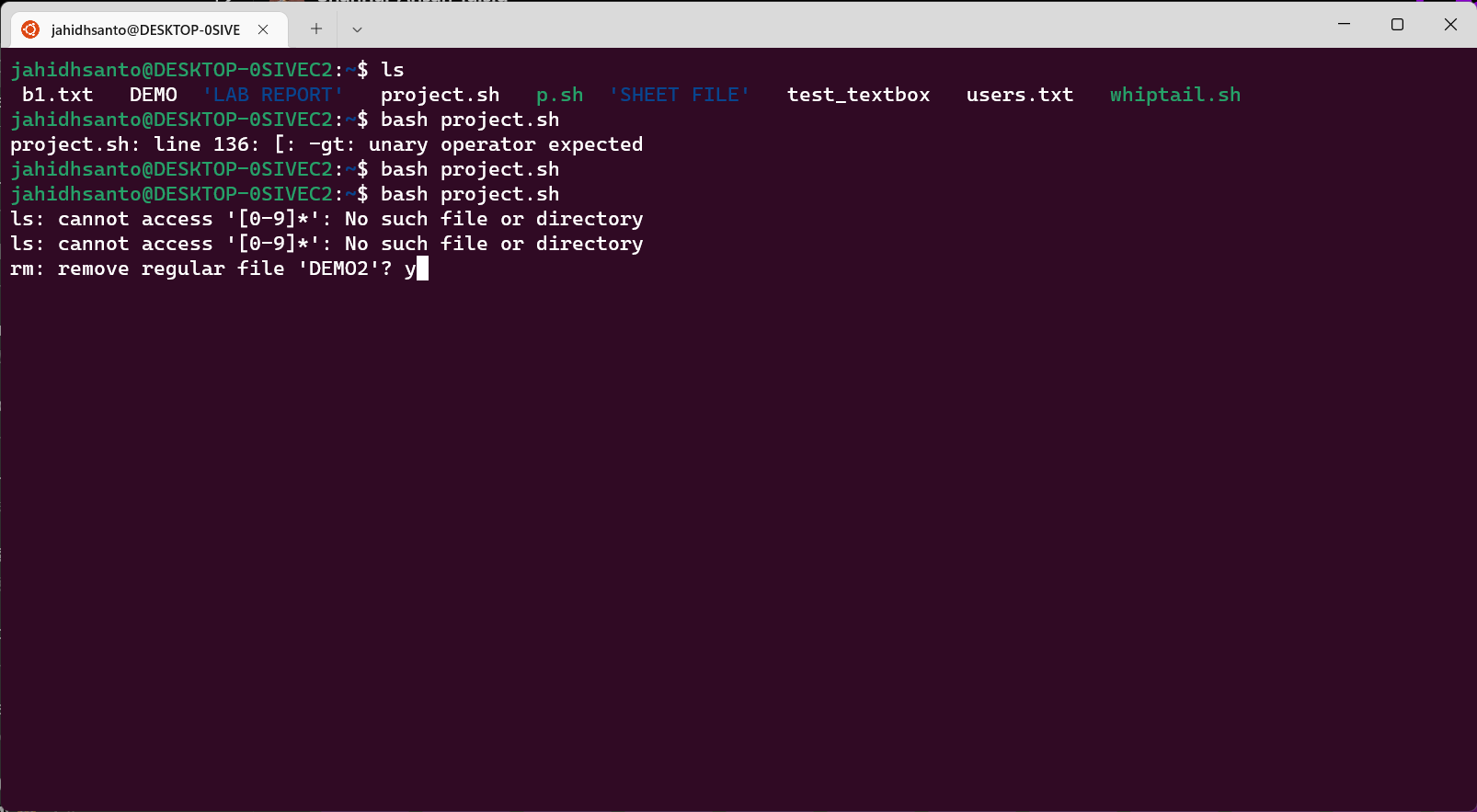


Figure 3.1.8: Removed Database



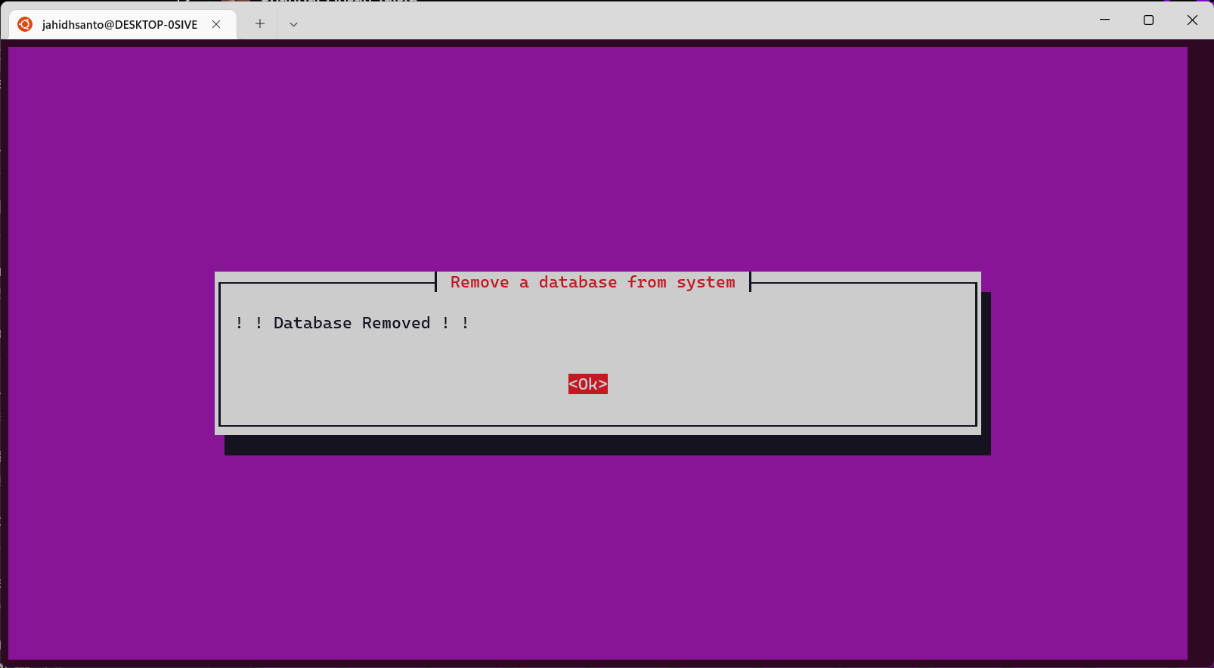


Figure 3.1.9: Confirmation of Removed Database.

Here option 2 for manage the database.

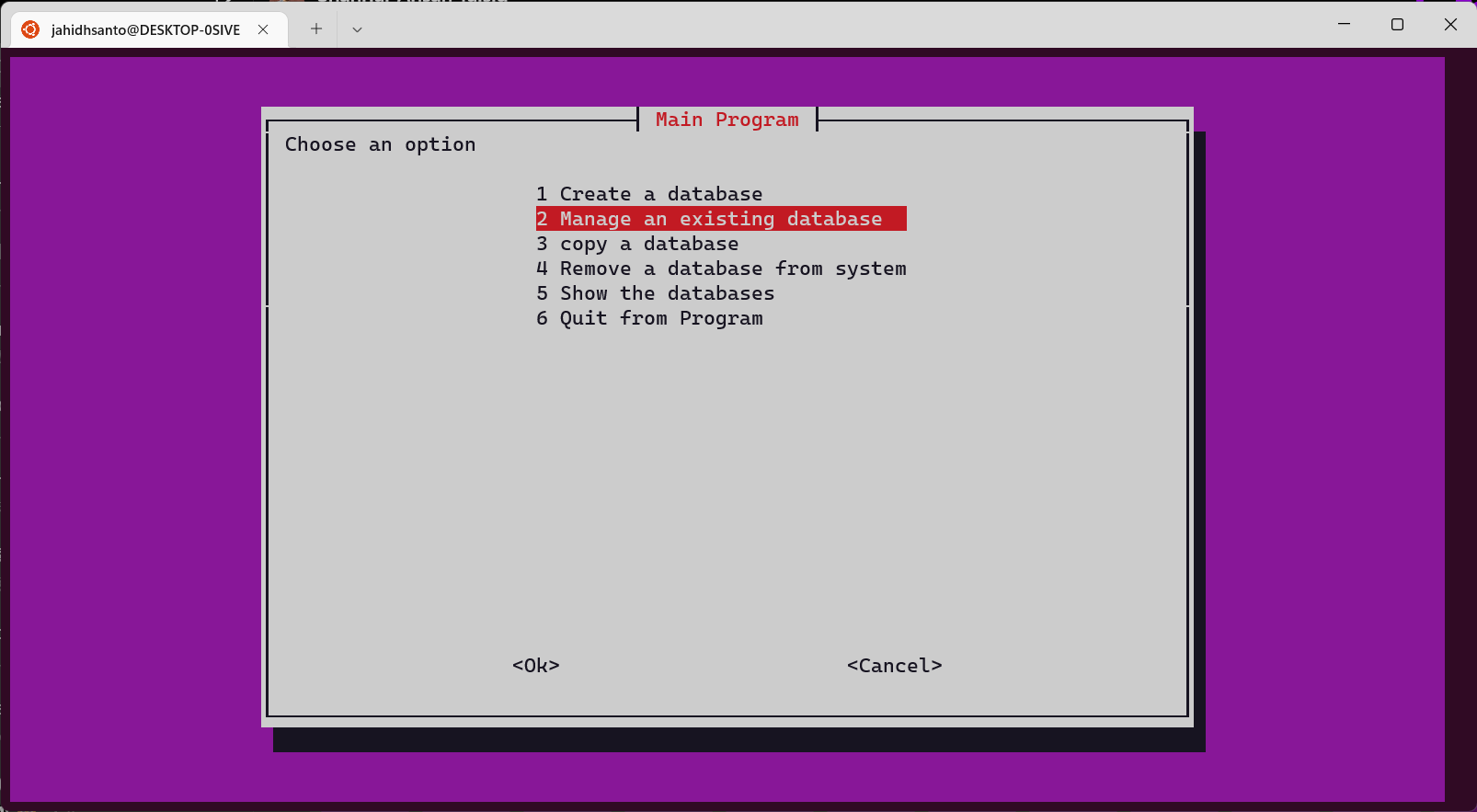


Figure 3.1.10 : Managing Database

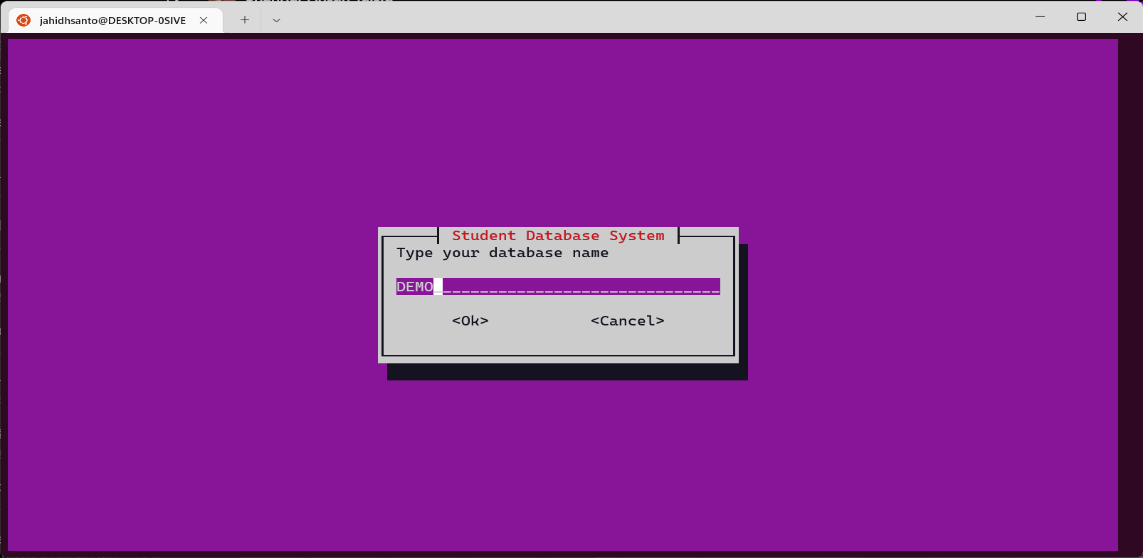


Figure 3.1.11 : Entering into a Database

There are some options for managing the data base 1 for insertion 2 for Delete 3 for show info

4 for query like sort, search etc.

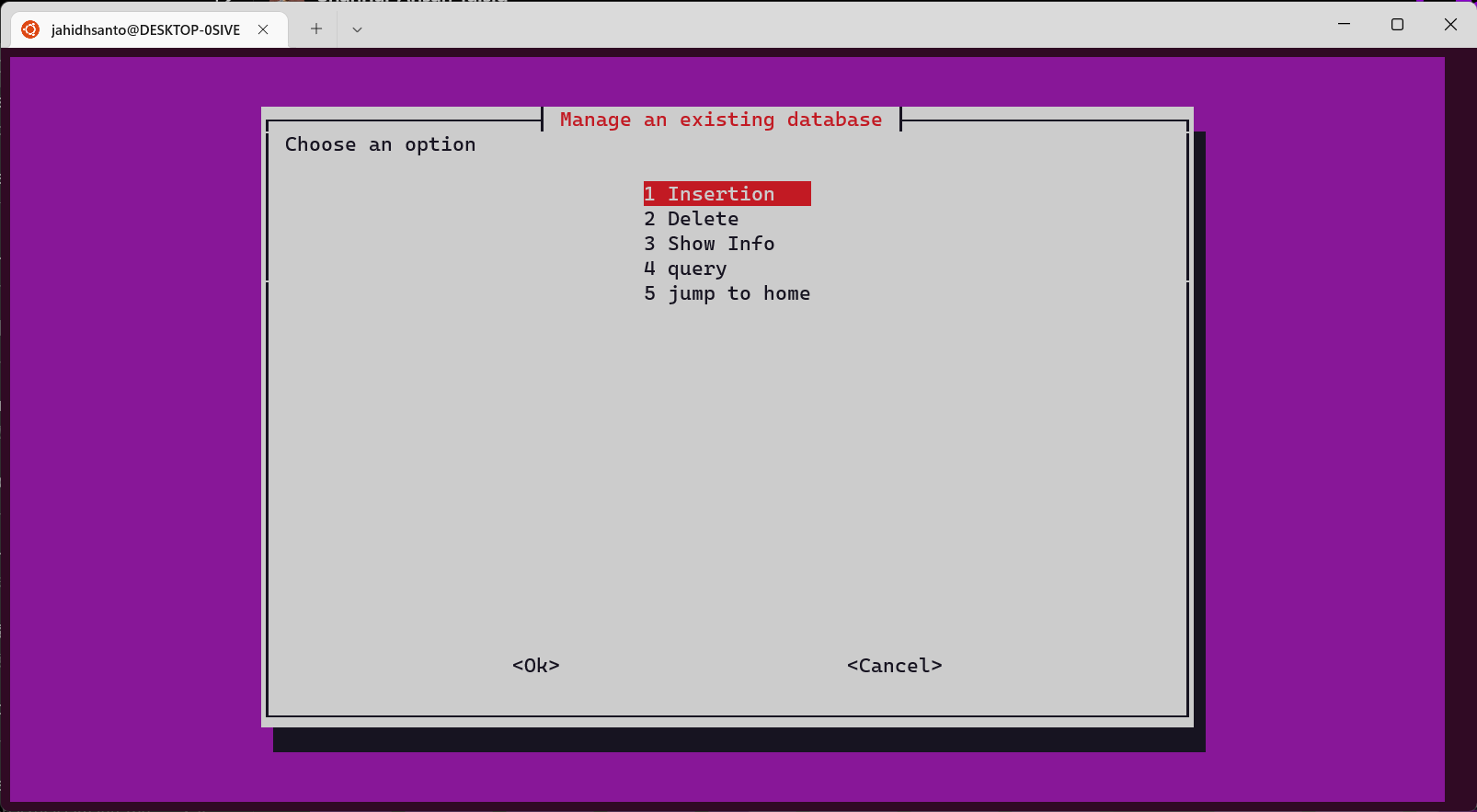
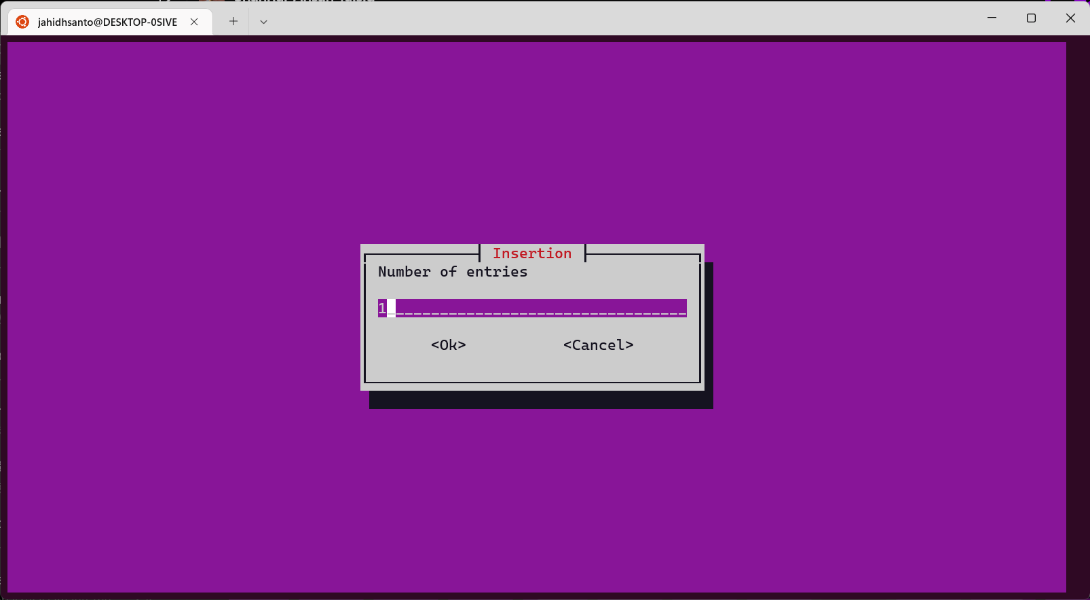
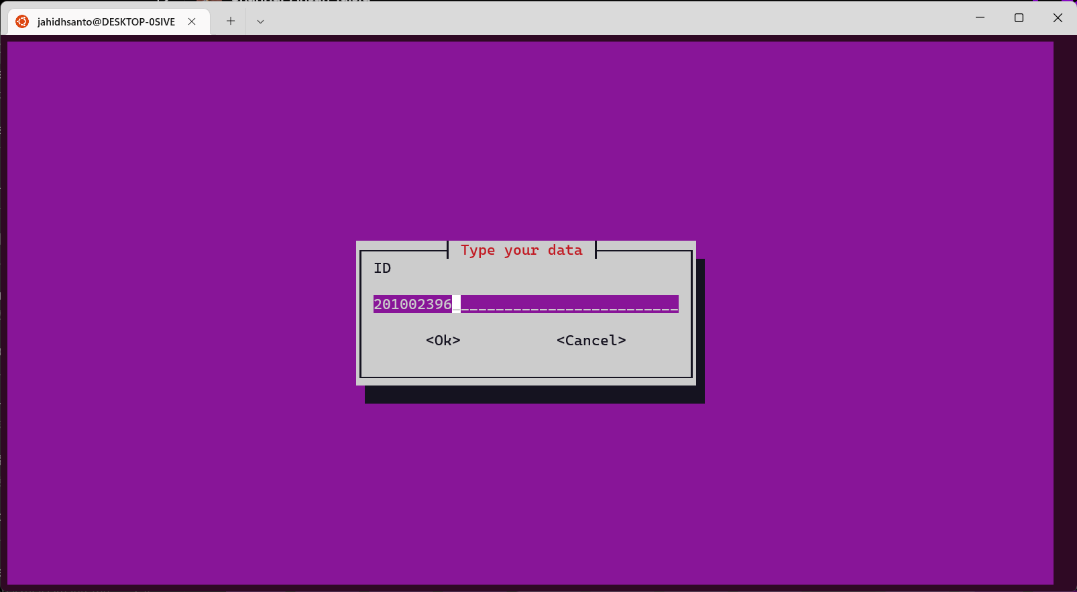
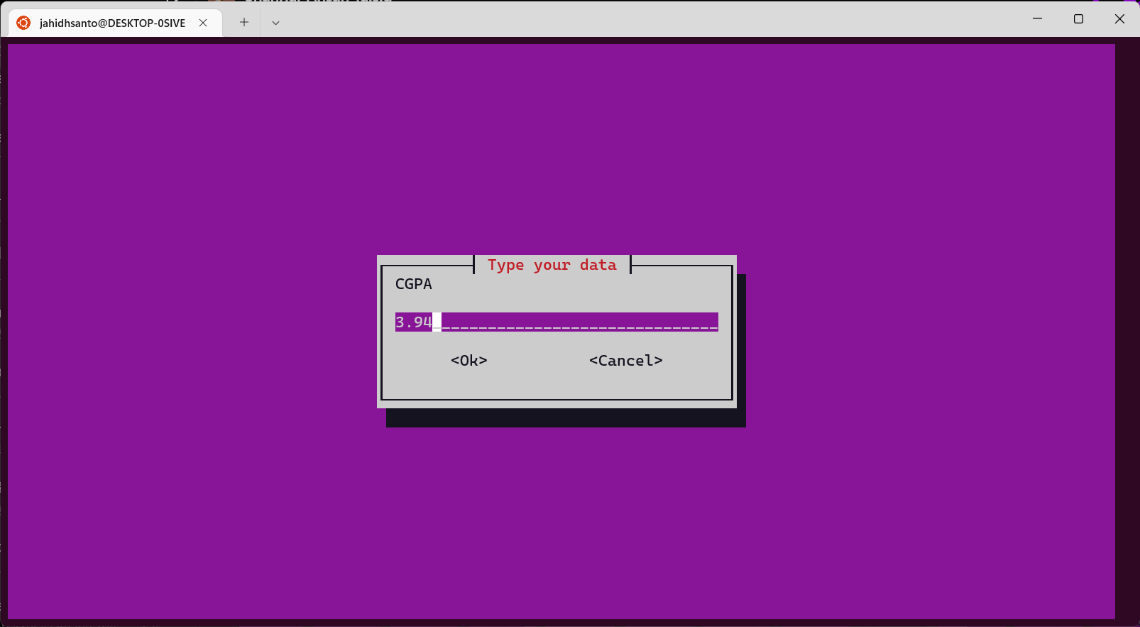


Figure 3.1.12 : Continue with insertion operation







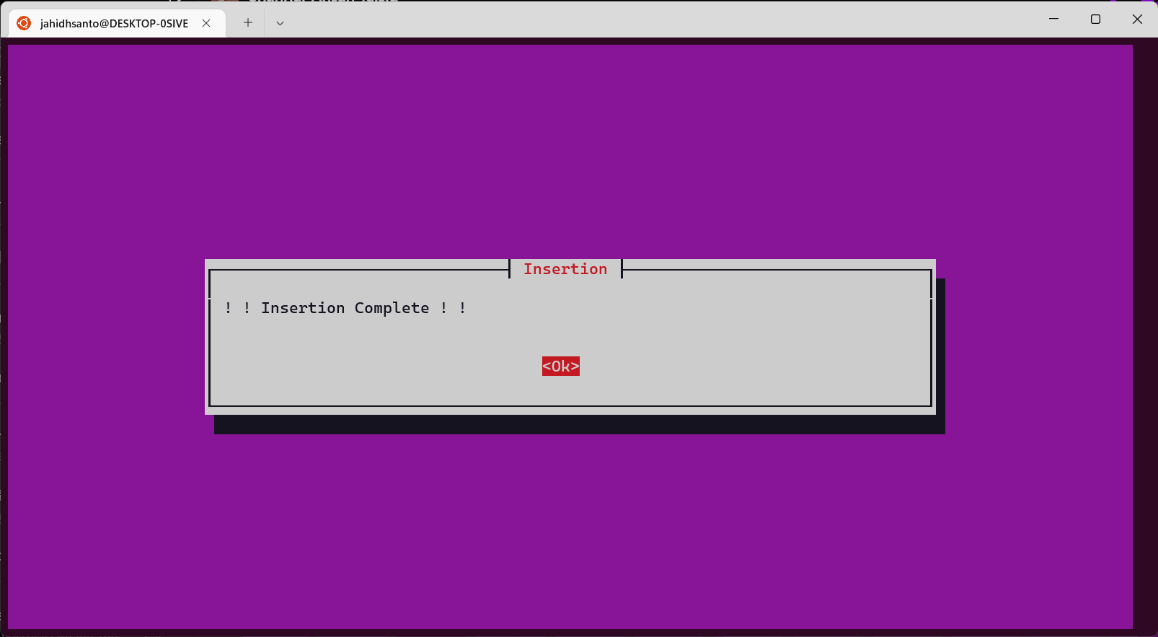


Figure 3.1.13 : Insert Data



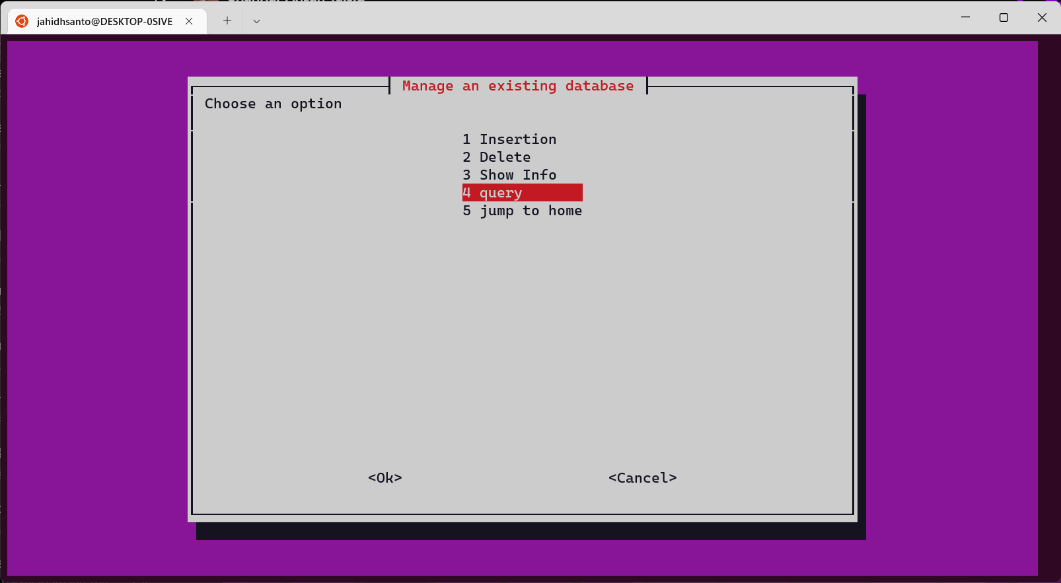
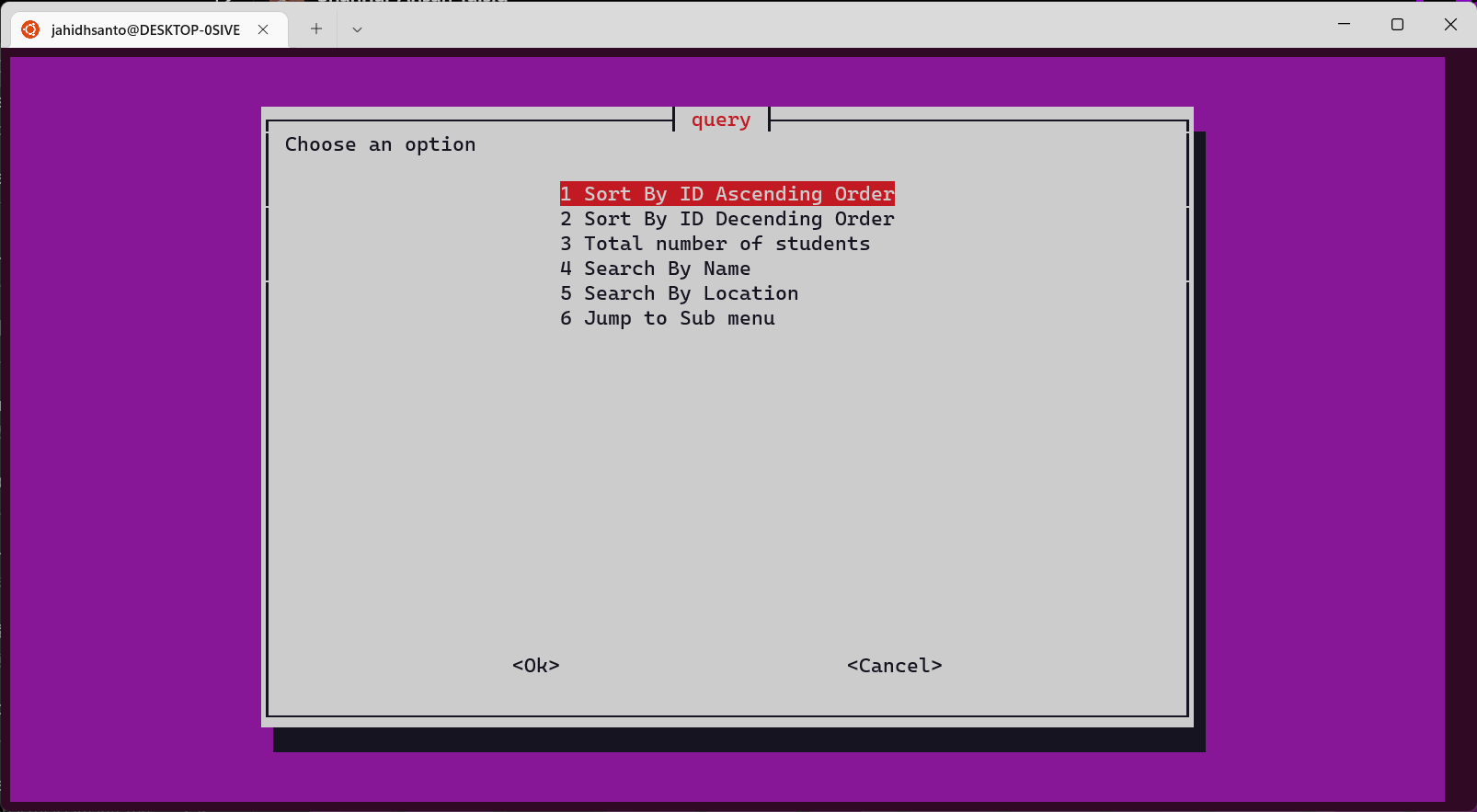
Figure 3.1.14 : Show all the data of Database  


Figure 3.1.15 : Continue with Query Operation  


Sort the data Ascending Order

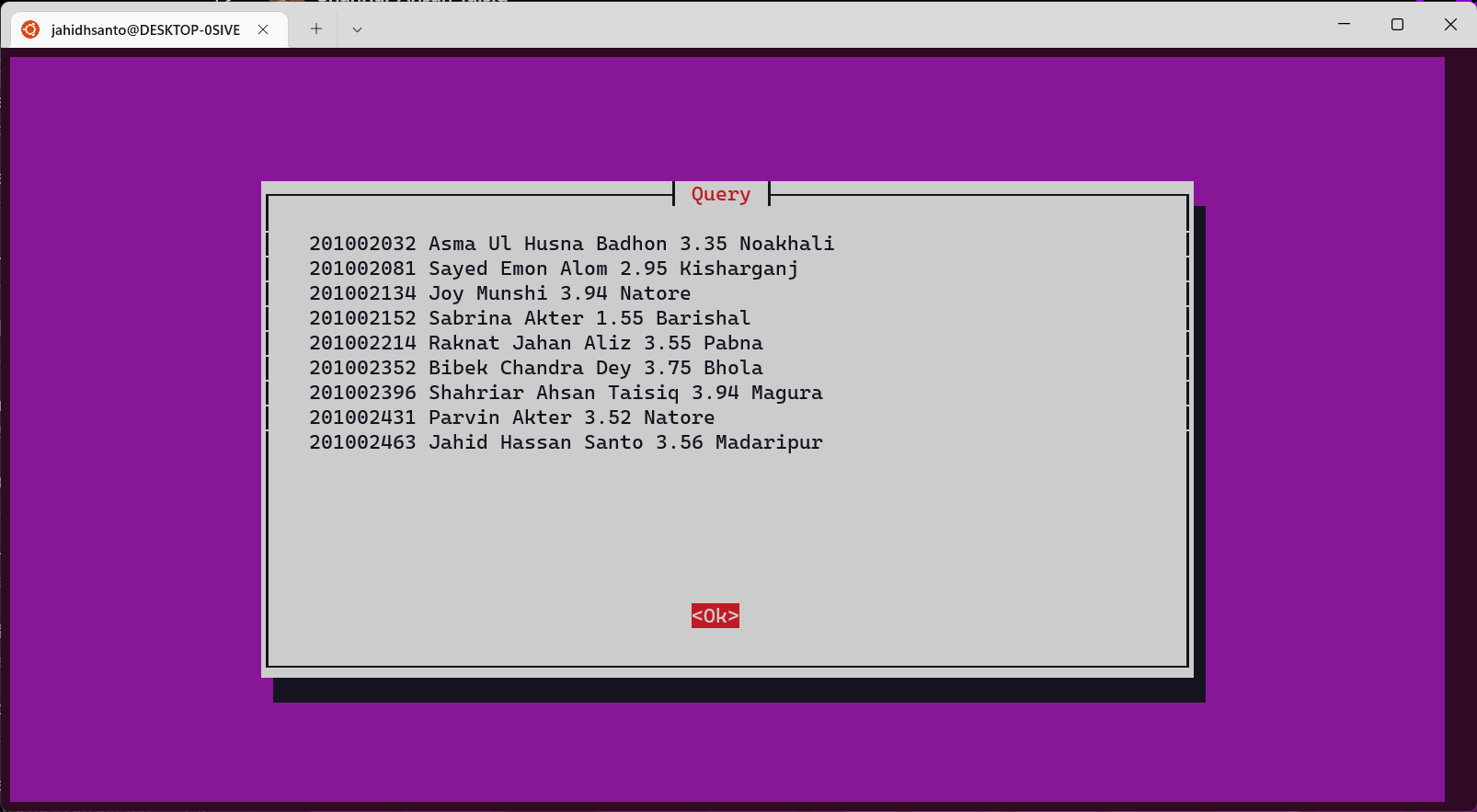
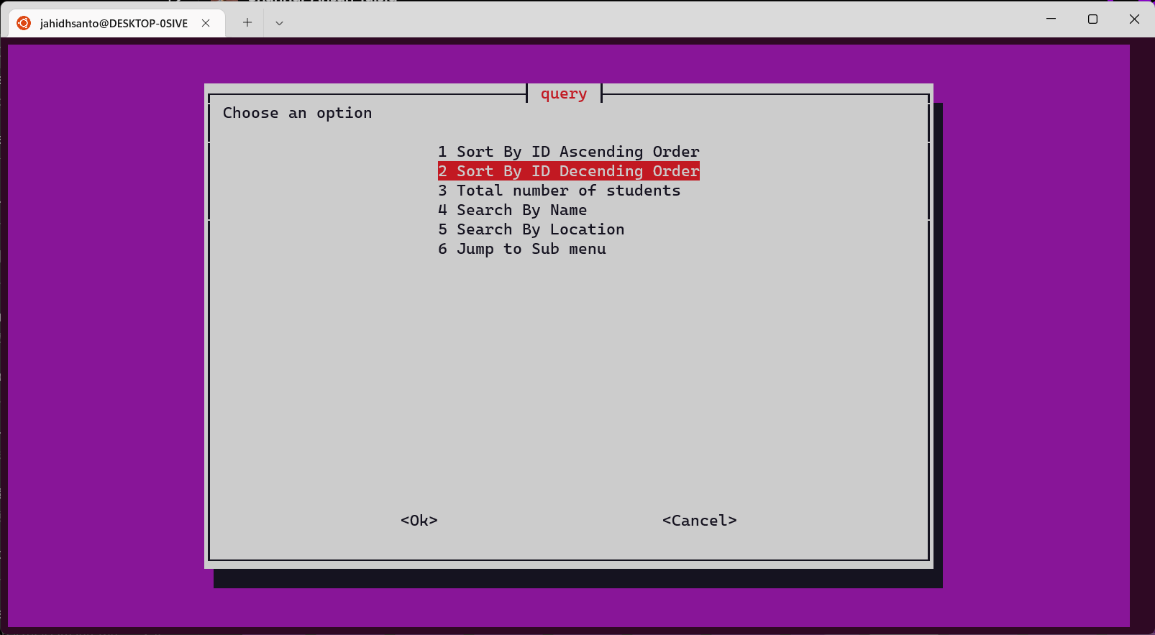


Figure 3.1.16 : Sort the data Ascending order

Sort the data Descending Order.



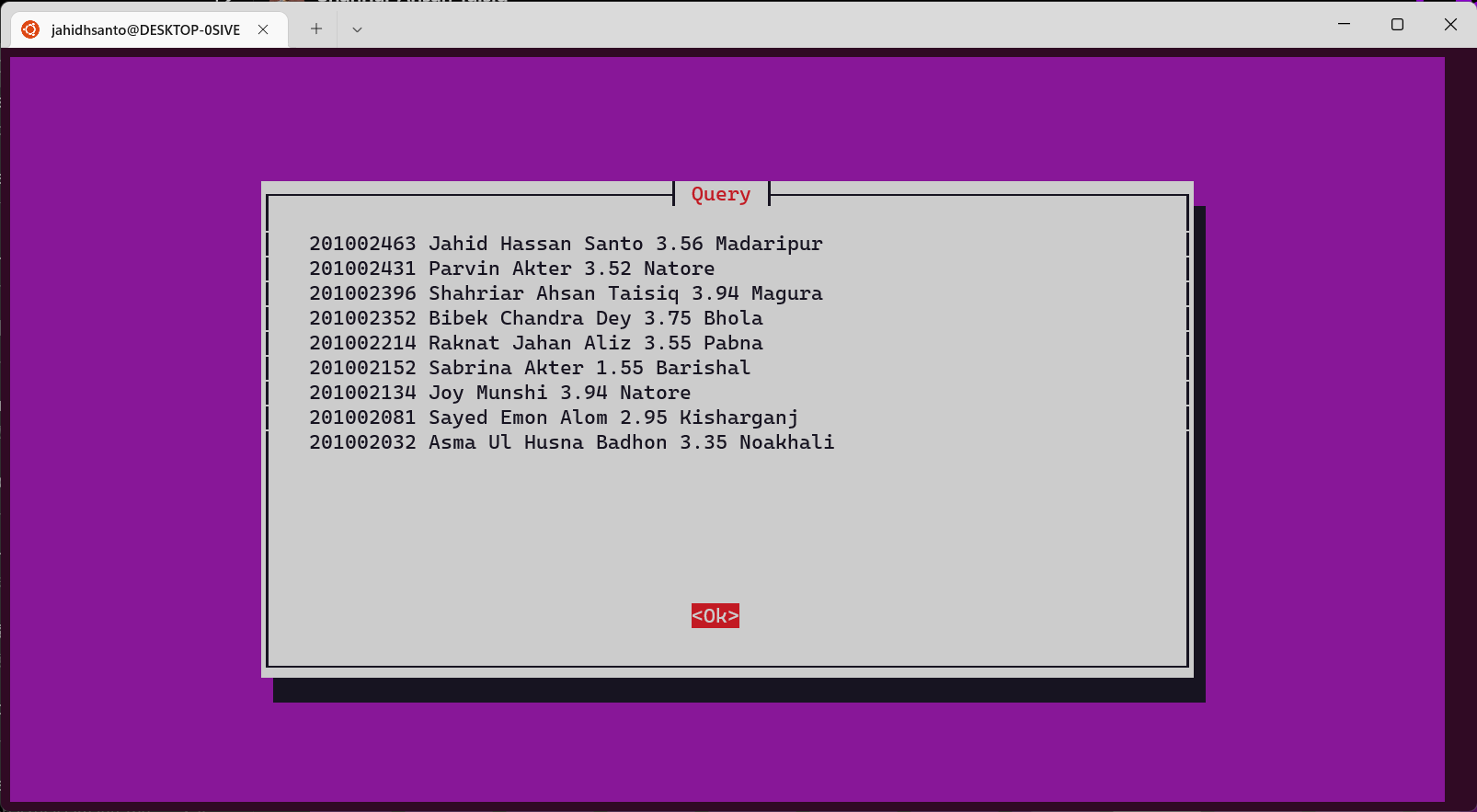
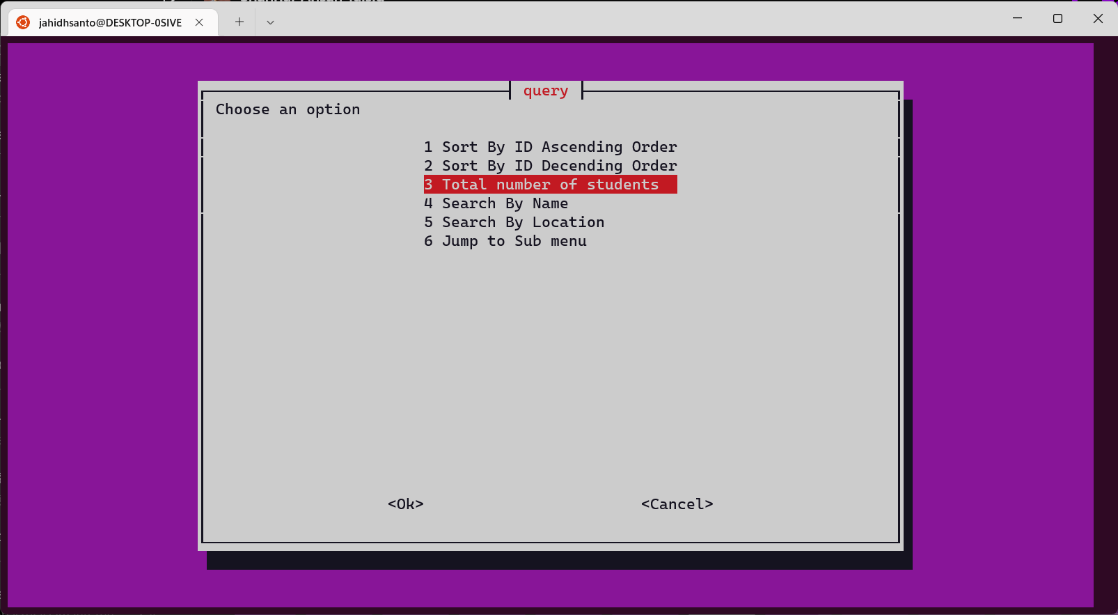


Figure 3.1.17 : Sort the data Descending Order

Count the number of students we inserted.



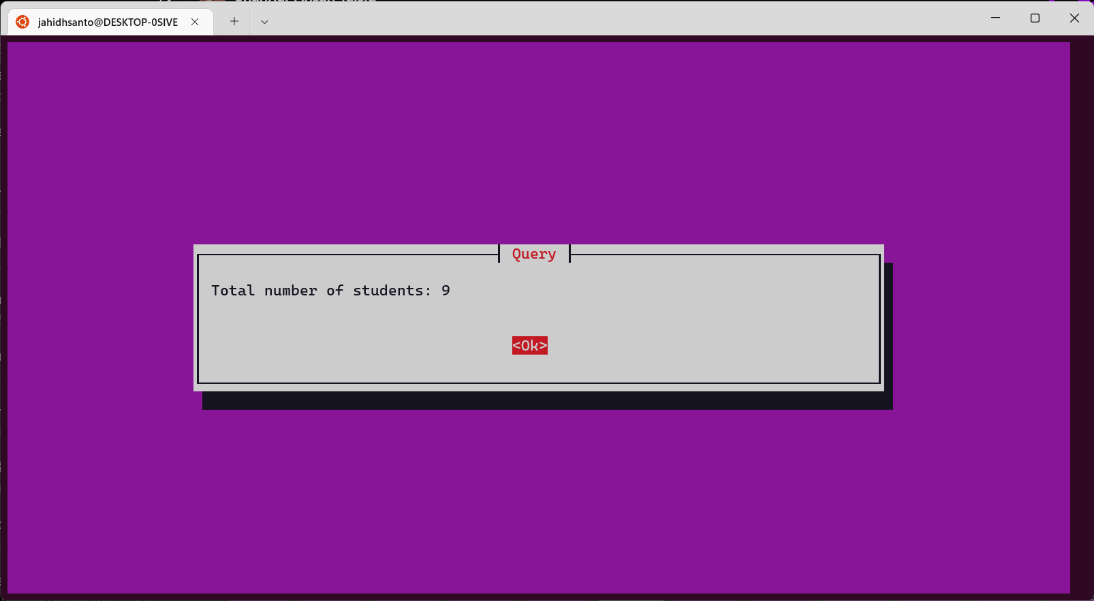
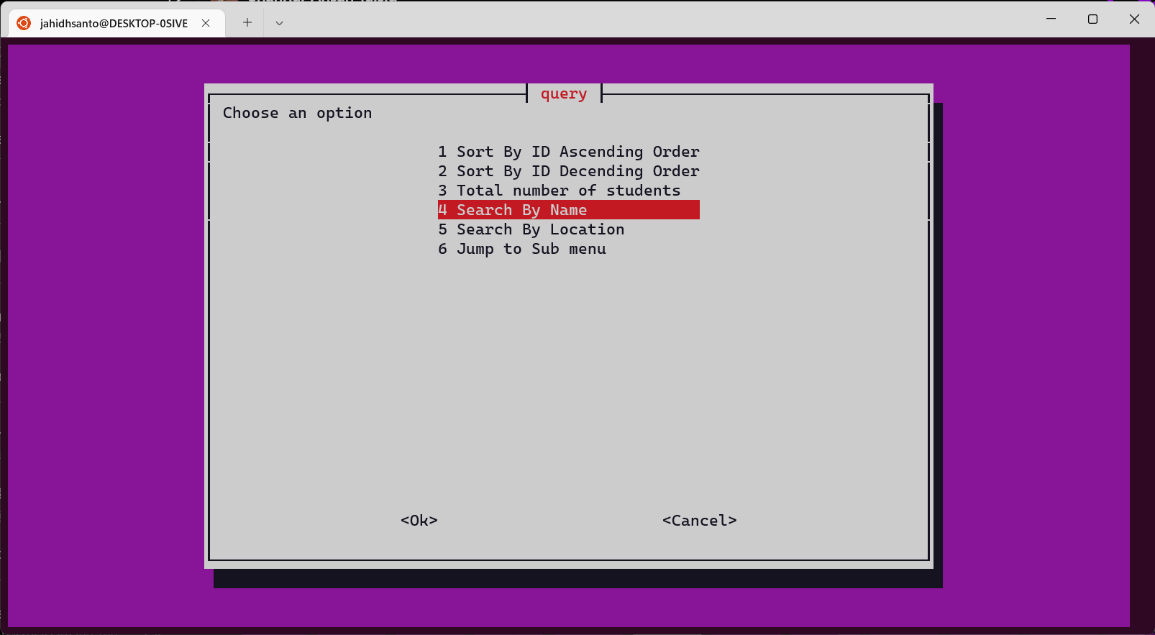
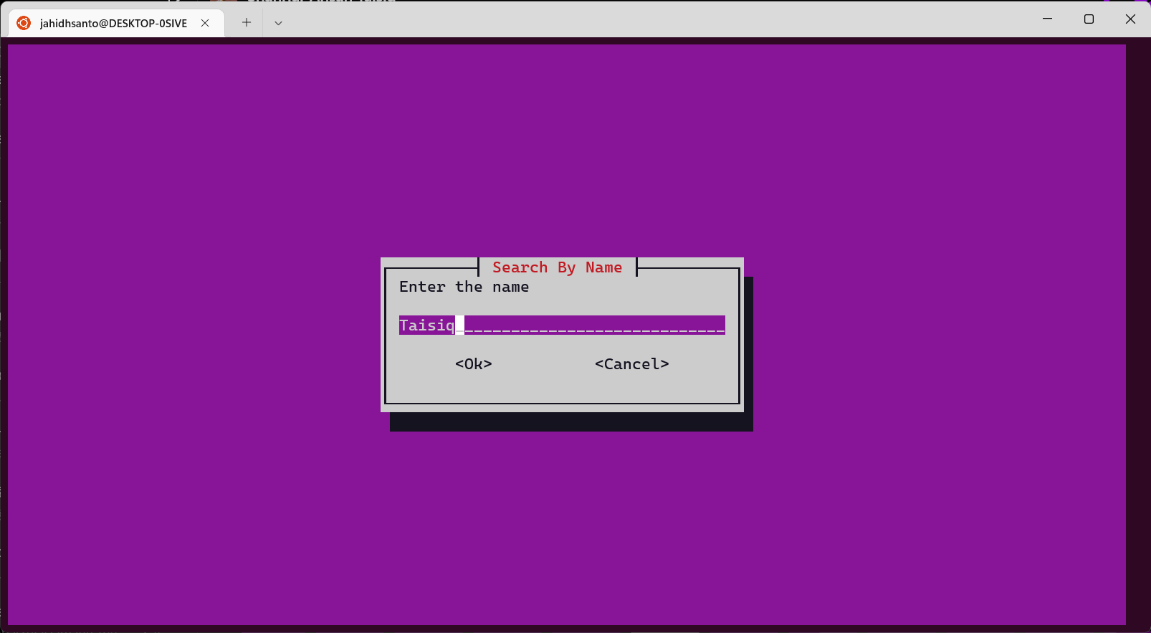


Figure 3.1.18 : Count the Total Number of Student

Search by the student name and Search by the Location of Student.





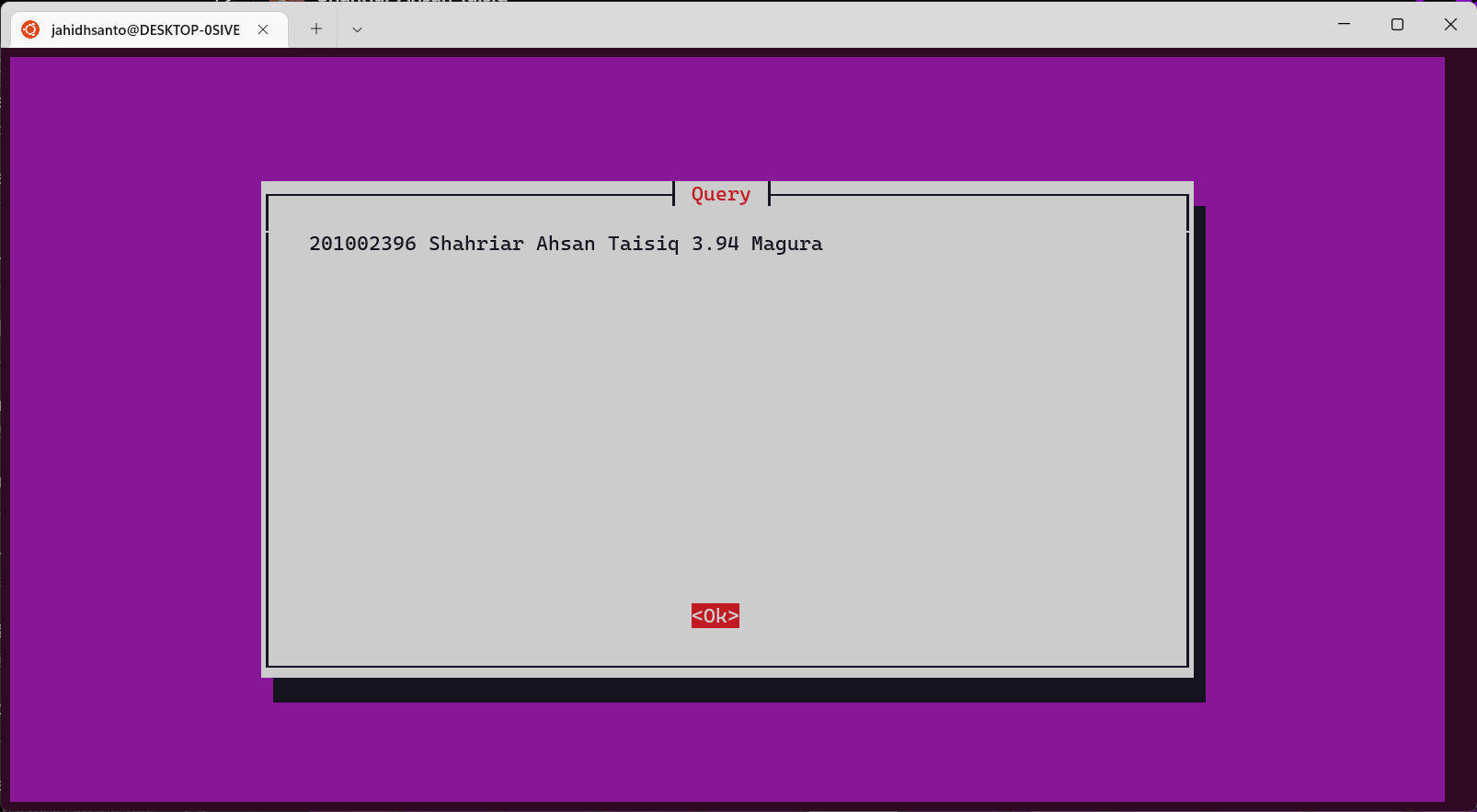
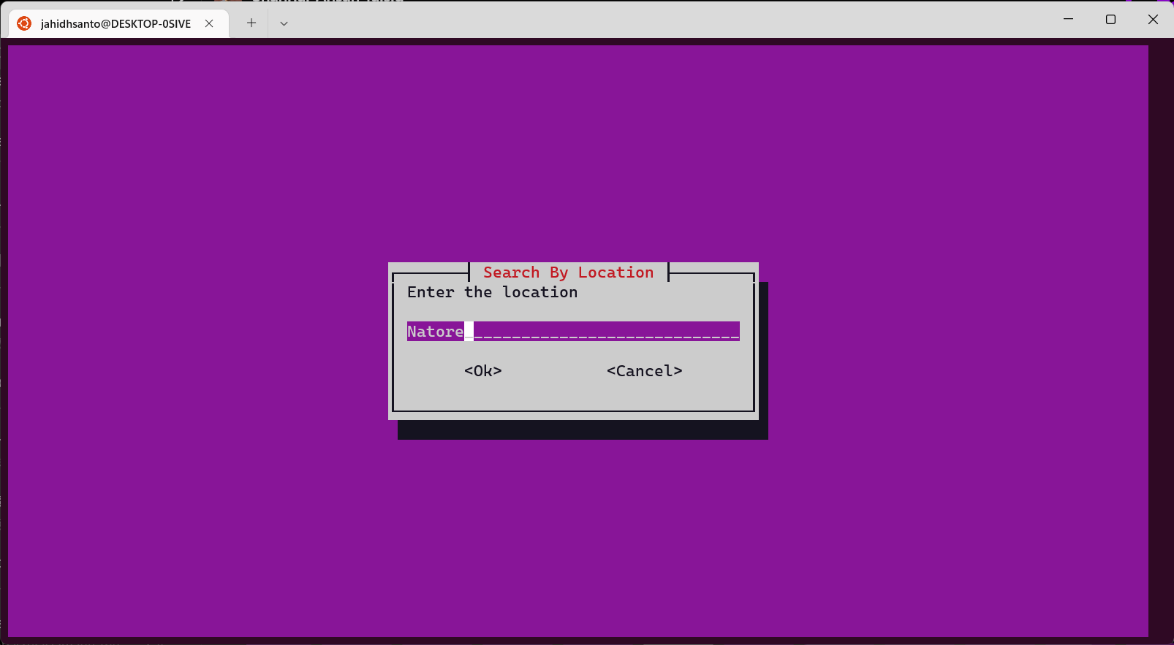


Figure 3.1.19 : Search by Name



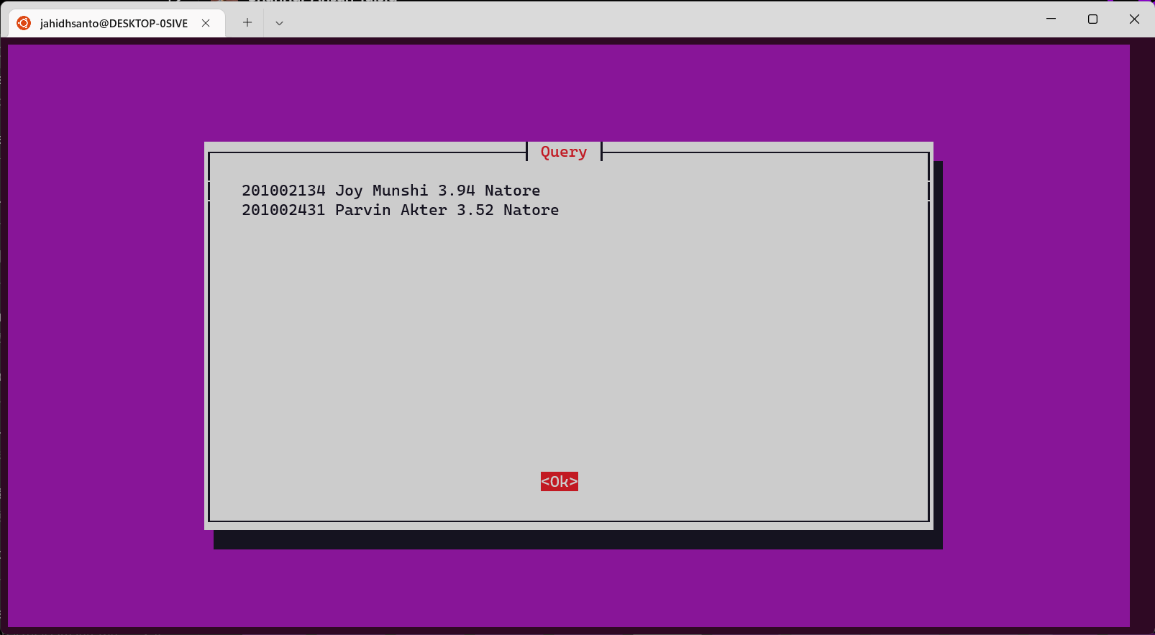


Figure 3.1.20 : Search by Location

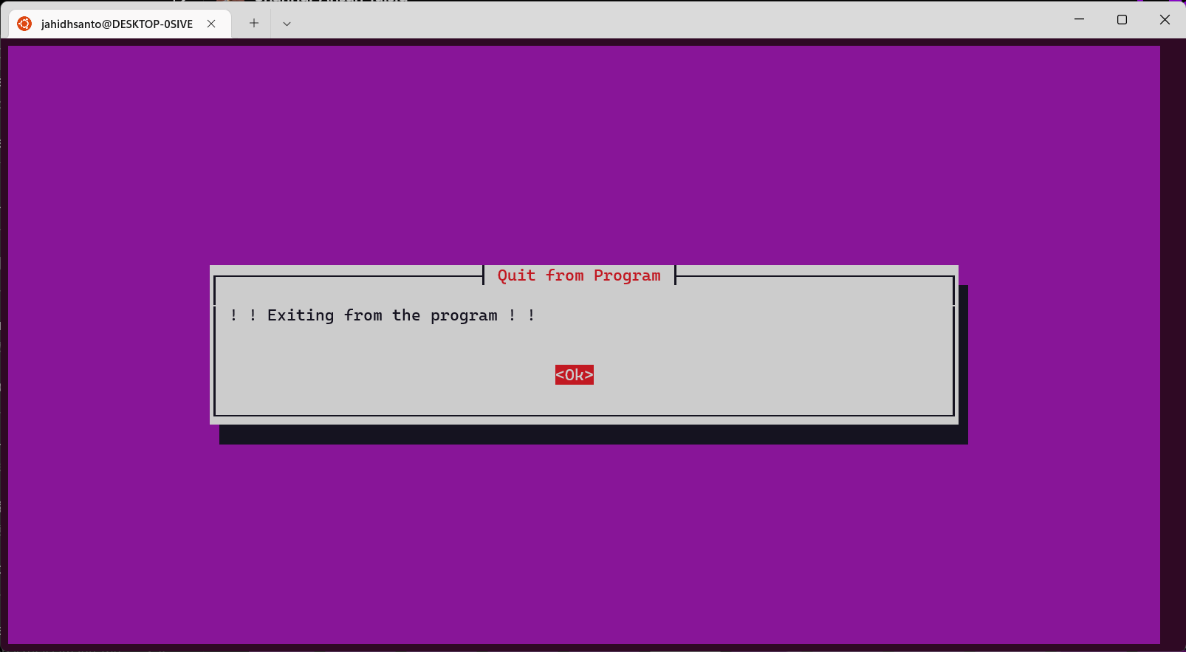


Figure 3.1.21 : Quit the Program

## Project Implementation

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234 | Menu\_Loop ()  {  select=100  while [ $select -gt 0 ]  do  select=$(whiptail --title "Main Program" --menu "Choose an option" 25 78 10 \  "1" "Create a database" \  "2" "Manage an existing database" \  "3" "copy a database" \  "4" "Remove a database from system" \  "5" "Show the databases" \  "6" "Quit from Program" 3>&1 1>&2 2>&3)  case "$select" in  1)  name=$(whiptail --inputbox "Enter your database name" 8 39 --title "Create a database" 3>&1 1>&2 2>&3)  echo>$name  whiptail --title "Create a database" --msgbox "! ! Database Created ! !" 8 78  ;;  2)  db=$(whiptail --inputbox "Type your database name" 8 39 --title "Student Database System" 3>&1 1>&2 2>&3)  menu=100  while [ $menu -gt 0 ]  do  menu=$(whiptail --title "Manage an existing database" --menu "Choose an option" 25 78 5 \  "1" "Insertion" \  "2" "Delete" \  "3" "Show Info" \  "4" "query" \  "5" "jump to home" 3>&1 1>&2 2>&3)  case "$menu" in  1)  num=$(whiptail --inputbox "Number of entries" 8 39 --title "Insertion" 3>&1 1>&2 2>&3)  for(( i=0; i<num; i++ ))  do  id=$(whiptail --inputbox "ID" 8 39 --title "Type your data" 3>&1 1>&2 2>&3)  nam=$(whiptail --inputbox "Name" 8 39 --title "Type your data" 3>&1 1>&2 2>&3)  gpa=$(whiptail --inputbox "CGPA" 8 39 --title "Type your data" 3>&1 1>&2 2>&3)  loc=$(whiptail --inputbox "Location" 8 39 --title "Type your data" 3>&1 1>&2 2>&3)  var=" $id $nam $gpa $loc "  echo " $var " >>$db  whiptail --title "Insertion" --msgbox "! ! Insertion Complete ! !" 8 78  done  ;;  2)  ln=$(whiptail --inputbox "Enter the id number" 8 39 --title "Delete" 3>&1 1>&2 2>&3)  whiptail --title "ln" --msgbox "$ln" 30 78  grep -v " $ln " $db  ;;  3)  #cat $db  whiptail --title "Example Dialog" --msgbox "$(cat $db)" 30 78  ;;  4)  que=100  while [ $que -gt 0 ]  do  que=$(whiptail --title "query" --menu "Choose an option" 25 78 6 \  "1" "Sort By ID Ascending Order" \  "2" "Sort By ID Decending Order" \  "3" "Total number of students" \  "4" "Search By Name" \  "5" "Search By Location" \  "6" "Jump to Sub menu" 3>&1 1>&2 2>&3)  case "$que" in  1)  whiptail --title "Query" --msgbox "$(sort $db)" 20 78  ;;  2)  whiptail --title "Query" --msgbox "$(sort -r $db)" 20 78  ;;  3)  line=`wc -l < $db | tr -d ' ' `  whiptail --title "Query" --msgbox "Total number of students: $line " 8 78  ;;  4)  en=$(whiptail --inputbox "Enter the name" 8 39 --title "Search By Name" 3>&1 1>&2 2>&3)  whiptail --title "Query" --msgbox "$(grep $en $db)" 20 78  ;;  5)  el=$(whiptail --inputbox "Enter the location" 8 39 --title "Search By Location" 3>&1 1>&2 2>&3)  whiptail --title "Query" --msgbox "$(grep $el $db)" 20 78  ;;  6)  que=0  ;;  esac  done  ;;  5)  menu=0  ;;  esac  done  ;;  3)  file=$(whiptail --inputbox "Enter file name" 8 39 --title "copy a database" 3>&1 1>&2 2>&3)  file1=$(whiptail --inputbox "Enter second file name" 8 39 --title "copy a database" 3>&1 1>&2 2>&3)  if [ -f $file ]  then  cp $file $file1  whiptail --title "copy a database" --msgbox "! ! File Copied ! !" 8 78  else  whiptail --title "copy a database" --msgbox "! ! File does not exist ! !" 8 78  fi  ;;  4)  file=$(whiptail --inputbox "Enter a file name to be removed" 8 39 --title "Remove a database from system" 3>&1 1>&2 2>&3)  if [ -f $file ]  then  rm -i $file  whiptail --title "Remove a database from system" --msgbox "! ! Database Removed ! !" 8 78  else  whiptail --title "Remove a database from system" --msgbox "! ! $file does not exist ! !" 8 78  fi  ;;  5)  whiptail --title "Show the databases" --msgbox "$(ls {\*[!.sh],[0-9]\*})" 20 78  ;;  6)  whiptail --title "Quit from Program" --msgbox "! ! Exiting from the program ! !" 8 78  exit  ;;  esac  done  }  count=`wc -l < users.txt | tr -d ' '`  arr=( Jahid Taisiq Santo Shahriar )  control=1  while [ $control -gt 0 ]  do  control=$(whiptail --title "Student Database System" --menu "Choose an option" 25 78 5 \  "1" "Execute the Program" \  "0" "Abrot" 3>&1 1>&2 2>&3)  case "$control" in  1)  name=$(whiptail --inputbox "USERNAME" 8 39 --title "USER VEFIFICATION" 3>&1 1>&2 2>&3)  for (( i=0; i<count; i++ ))  do  if [[ $name == "${arr[$i]}" ]]  then  pass="access"  new\_pass=$(whiptail --passwordbox "please enter your secret password" 8 39 --title "USER VEFIFICATION" 3>&1 1>&2 2>&3)  if [[ $new\_pass == $pass ]]  then  {  for ((i = 0 ; i <= 100 ; i+=5)); do  sleep 0.1  echo $i  done  } | whiptail --gauge "Please wait while we are sleeping..." 6 50 0  whiptail --title "USER VEFIFICATION" --msgbox "Access Granted" 8 78  entry=$(whiptail --title "Student Database System" --menu "Choose an option" 25 78 5 \  "1" "Main Program" \  "2" "add User" 3>&1 1>&2 2>&3)  case "$entry" in  1)  Menu\_Loop  ;;  2)  n=$(whiptail --inputbox "How many User You Want to Add?" 8 39 --title "Add Temporary Users" 3>&1 1>&2 2>&3)  new\_count=$(($count+$n))  for(( i=$count; i<$new\_count; i++ ))  do  m=$(whiptail --inputbox "USERNAME" 8 39 --title "USER VEFIFICATION" 3>&1 1>&2 2>&3)  echo "$m" >> users.txt  arr=(${arr[0]} "$m")  echo ${arr[0]}  done  ;;  esac  else  whiptail --title "Student Database System" --msgbox "Wrong Password" 8 78  fi  break;  else  elim=$(($count-1))  if [[ $i == $elim ]]  then  {  for ((i = 0 ; i <= 100 ; i+=5)); do  sleep 0.1  echo $i  done  } | whiptail --gauge "Please wait while we are sleeping..." 6 50 0  whiptail --title "Student Database System" --msgbox "Access Denied" 8 78  fi  fi  done  ;;  2)  control=0;;  esac  done |

## Results and Discussions

### Results

* We can organize in a way that facilitates local or remote information retrieval.
* User privacy kept well managed.
* Able to process many continual queries over a long period of time.
* Low cost and reliable database.
* Facilitates temporary users to access the database.

### Discussions

* + - The delete operation does not get executed properly.
    - Limited amount of student data can be stored.
    - The database can store student result & location details.
    - This database can be applied to any organizational site in terms of its privacy maintain protocol.

# Chapter 4

# Conclusion

## 4.1 Introduction

Using Ubuntu and Shell Scripting, we need to create a database system that can store, access, and link system data according to search operations (queries). In this system, we have tried to use the basic features that are required to maintain a student database system. Here we create 4 sections that have some valuable features. We maintain privacy and user access here. We used filters and wildcards to search for information throughout the system. Database manipulation is possible as well, the information in that database.

## 4.2 Practical Implications

* The database can store information about students' results as well as their location. Thus, it can be used as an organizational database.
* We can create a user-friendly database system with a webpage or website attached to the database.
* In terms of privacy management, this database can be applied to any organizational site.

## 4.3 Scope of Future work

* + - We can make a user-friendly database system with a webpage or website attached to the database.
    - A lot of fields must be included to make the system more reliant.
    - The deletion process should be performed in a way that is actually erasing information from a particular database.

# References

1. <https://projectsgeek.com/2011/04/student-database-using-shell-programming.html>
2. <http://www.dailyfreecode.com/code/shell-script-perform-database-1657.>