VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum 590014



*A Project Report On*

**“VARSITY ADMISSION TEST RESULT”**

*Submitted in partial fulfillment of the requirements for the completion of the course*

*Programming in C++*

In

Computer Science and Engineering

*By*

**ROMIL K BALAR** **SAGAR RUDAGI**

**1BM17CS081** **1BM17CS085**

**ROHAN J MAYYA SAIFUR RAHMAN**

**1BM17CS077 1BM17CS086**

*Under the guidance of*

Mr. Pradeep S

Designation , Department of CSE,

BMSCE

*Project carried out at*

BMS College of Engineering, Bangalore



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Bull Temple Road, Basavanagudi, Bangalore - 560019

**Department of Computer Science and Engineering**

**B.M.S College of Engineering**

**Bangalore-560019**



**CERTIFICATE**

This is to certify that the project work entitled **“VARSITY ADMISSION TEST RESULT”** carried out by **ROMIL K BALAR (1BM17CS081)**, **SAGAR RUDAGI (1BM17CS085)**, **ROHAN J MAYYA(1BM17CS077), SAIFUR RAHMAN(1BM17SC086)** bona fide students of BMSCollege of Engineering in partial fulfillment for the completion of the course “Programming with C++” in **Computer Science and Engineering** of BMS College of Engineering during the year **2018-2019** is found to be satisfactory.

Signature of the Guide Signature of the HOD

Name of the examiners Signature with date

1.

2.

**ACKNOWLEDGEMENT**

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude ***Pradeep S, Course instructor, BMS College of Engineering***for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

**ABSTRACT**

"Varsity admission test result” is the C++ project built by us which provides the merit list and also divides the students into their respective Courses based on the priority entered and the ranks of the test results. This summarizes the main functionality of the application.

To easy viewing by an administrator different files are created for each list and they can be handled separately after these lists are created by the application.

These lists can be viewed in a formatted manner from within the application through different display techniques and proper formatting.

Pure object- oriented programming was implemented instead of traditional arrays to provide better memory allocation and benefits of OOPS concepts such as modularity, abstraction and encapsulation. Each student is represented as a student.

**CONTENT**

|  |  |
| --- | --- |
| 1. | Overview of C++ |
| 2. | Introduction  -Problem Statement  -Objective |
| 3. | Requirement analysis  -Hardware requirements  -Software requirements |
| 4. | Implementation  -List of header files used  -List of functionalities added |
| 5. | Concepts used |
| 6. | Screenshots of Working project |
| 7. | Short comings and limitations |
| 8. | Bibliography |

**OVERVIEW OF C++**

C++ is a general purpose object oriented programming language developed by Bjarne Stroustrup. Initially it was called “C with Classes” and was renamed as “C++” in the year 1983 .

C++ based on the language C, combines both the high level ideas of classes with low level machine handling capacities of C making it a very robust and a reliable programming language. C++ is a strongly typed and a fast programming language.

C++ is a superset of C. Any program written in C is also a legal program in C++. C++ is standardised by ISO, which has released five versions of C++ standard and also currently working on the sixth version C++20.

## Features Of C++:

C++ fully supports the object-oriented aspect of programming which was the primary reasons for its invention by Bjarne Stroustrup. The four import features of C++ as an object-oriented programming language is

* + - * Data Encapsulation
      * Data Hiding
      * Inheritance
      * Polymorphism

To make sure the programs written in C++ are portable over a variety of domains it implements the ANSI (American National Standards Institute) standard. This standardisation is an attempt to make sure that all C++ programs are portable meaning a code written for Microsoft’s compiler will compile without giving any errors on a compiler on Mac, Unix or any other operating system.

All the major compilers designed for C++ supports the ANSI standard.

## Advantages of Using C++:

C++ is a robust improvement over an already incredible programming language C. C++ offers a verity of excellent features making it an incredible programming language to code with. C++ has a lot of advantages. Few of them are:

* + - * Use of classes makes it an incredibly flexible programming language
      * A large pool of functions are available through the C++ standard library
      * Easy portability due to the implementation of the ANSI Standard
      * One can easy move from programming with C to programming with C++ due to the same program structure.

## Applications Of C++:

* Operating systems like Microsoft Windows, MacOS X or Linux are all programmed in C++. Since it a strongly typed and a fast programming language it is an ideal choice to program operating systems in C++
* Since graphical applications require very fast rendering, C++ helps with reducing the latency.
* Databases like Postgres and MySQL the two of the most widely used database applications are written in C and C++
* Since C++ is a low level programming language Many compilers written for other languages use

C++ as the backend programming language.

**INTRODUCTION**

**PROBLEM STATEMENT:**

VARSITY ADMISSION TEST RESULT:

We were provided with a system which only read from an external file and initialised values into various variables through multidimension arrays for each parameter. This did provide working menus for each function but didn’t really provide many real-world functionalities and was missing real life exams such as CET, COMEDK, etc.

**ISSUES**: No real use for the basic program and lack of readability in program for future developers who look to work on the application.

**METHOD**: Converting multidimensional array concept to modern object-oriented programming to provide readability, efficiency and for providing easier means of updating code.

Adding missing real-world applications and adding complex sorting and insertion algorithms to create separate lists based on student rank and preference.

**OBJECTIVE**

Every educational institution has admissions and this process of admission has a lot of caveats and restrictions which need to be considered while creating the merit list, which millions of students are eagerly waiting and looking forward to, so that they can solidify their place in their institution of choice, along with their subject of choice. This makes an application which needs to perform this task very important and any human error should be avoided as far as possible, hence automation of this task would remove any discrepancies.

We have made this application taking into consideration all the conditions and restrictions on the number of seats available in each course and the number of students which can be taken from each entrance exam, and providing a pre-defined quota for management which remains defined and clear.

**SYSTEM REQUIREMENTS**

## Hardware Requirements

* More than 128 Mega Bytes of RAM
* 10 Mega Bytes of Hard Disk Storage

## Software Requirements

* Microsoft Window 7 or Higher
* Mac OS X
* Any C++ integrated development environment

Implementation

# List of Header Files Used

## IOSTREAM:

This header file is used to make use of the member functions cin and cout. These two functions facilitate to take input from the user and output onto the console. We have made use of the STD namespace to bring the STD keyword to the program namespace or else we would have to use the “std::” keyword for using the cin and cout functions everywhere.

## STRING.H:

This header file is used to make use all the strings functionality. We have made use of the string functions strcmp() to compare two strings which are passed as the parameter to the function. This function returns

* A value 0 if both the strings are same.
* A value greater than 0 if the first matching character in the left string has a greater ASCII value
* A value lesser than 0 if the first matching character in the left string has a lesser ASCII value

We have also made use of the function strcpy(). Two strings are passed as parameters to the function. It copies the contents of the right string and puts in the left string.

## IOMANIP:

IOMANIP is a library that is used to manipulate the output of C++ program. Using C++, header providing parametric manipulators. This library is used to make use of the function setw() which was helpful in spacing the columns of the table for displaying the information of the lists or the records.

## FSTREAM:

Input/output stream class to operate on files. Input/output stream class to operate on files.

* [**ofstream**](http://www.cplusplus.com/ofstream)**:** Stream class to write on files
* [**ifstream**](http://www.cplusplus.com/ifstream)**:** Stream class to read from files

We have extensively used both these classes in file handling viz., writing to a file or reading from a file.

**LIST OF FUNCTIONALITIES ADDED**

The basic aim of the program is to help the institution in the admission process. The program contains a basic Master Text File which contains the details of the students who enrolled for the institution. The text file contains the information of the student as follows:

1. Application Number
2. Name
3. Priority 1
4. Priority 2
5. Priority 3
6. CET Rank
7. COMEDK Rank
8. Management Flag

The program creates couple of text files to segregate the student according to means they wish to enrol. They are:

1. Application Number
2. Name
3. Priority 1
4. Priority 2
5. Priority 3
6. CET/COMEDK Rank

NOTE: The Management List contains only the first five options.

Program consists of three classes which contains all the parent functions. The **3** **classes** are as follows:

1. **Student** Class: This class basically takes five variables.

* **Application no -** Application number of the student who applied to the institution.
* **Name -** Name of the student.
* **CET Rank -** Rank the student secured in CET.
* **COMEDK Rank -** Rank the student secured in COMEDK.
* **Management Flag –** Holds 0 or 1 for non-management or management respectively.
* **Priority[3]** - Holds the numerical priority of the three subjects(CSE,ISE,ME).

The class contains some **functions** as stated with their functionality:

* **getDetails()** : This function asks the user for the details of the student and then appends it to Master List ‘File.txt’.
* **addStudent()** : This function is used to call the **getDetails()** function**.**
* **printStudentDetails()** : This function as the name states displays the information of the specified student.

1. **MeritList** Class : This class contains no variables but contains the most important functions. They are as stated:

* **makeMeritList()** : This function takes array of objects of Student Class and appends each record to the merit list.
* **printList()** : This function takes a record in specific and displays its details.
* **addToCETList()** : This function takes a specific record as parameter and adds it to the CET List if the CET rank exists in the record.
* **addToCOMEDKList()** : This function takes a specific record as parameter and adds it to the COMEDK List if the COMEDK rank exists in the record.
* **addToManagementList()** : This function takes a specific record as parameter and adds it to the Management List if the management flag is 1.

1. **Branch** Class : This class basically takes 3 variables.

* **Student List** - Holds the records of students.
* **Quota Limit** - Holds the limit intake of student for each quota.
* **Quota Count** – Holds the intake count of student for each quota.

Program consists of some non-member functions which are a primary part of setting up the files. They are as follows:

1. **countTotalLines()** : Counts the total number of records in the Master File ‘File.txt’.

2. **appendToFile()** : Takes a student record, file name(CETList.txt,COMEDKList.txt) as parameters and creates a file if it doesn’t exist and appends the student to the file.

3. **sortCETList()** : Takes a student array of objects and sorts them according to the CET rank. After sorting, it calls the **appendToFile()** for each student record.

4. **sortCOMEDKList()** : Takes a student array of objects and sorts them according to the COMEDK rank. After sorting, it calls the **appendToFile()** for each student record.

5. **searchArray()** : Takes a student record as a parameter and searches it in the student list of the branch class and return 1 if found else -1.

6. **tryBranch()** : Takes quota, branch and the student record as the parameter and checks whether the record already exists in the student list based on a constraint. If it doesn’t exist, it appends it to the student list. And increases the quota count in the end.

7. **assignBranches()** : Takes quota, student records array and the array count as parameters. It sets the branches to the student records and appends to the student file array by calling the **tryBranch()** and passing the necessary parameters.

8. **THE MAIN FUNCTION** : Main functions is where the user gets to choose what he wants to do by means of a **switch** case. The cases are as follows:

1. **Make merit list** : Creates a merit list.
2. **Add Student** : Adds a student to the Master File, ‘File.txt’.
3. **View merit list** : Displays the merit list.
4. **View student details** : Displays the information of a specific student.
5. **View CET list** : Displays the CET list with the student with their respective information.
6. **View COMEDK** **list** : Displays the COMEDK list with the student with their respective information.
7. **View Management list** : Displays the Management list with the student with their respective information.
8. **View CSE list** : Creates a list of student selected for CSE branch and displays the whole list.
9. **View ISE list** : Creates a list of student selected for ISE branch and displays the whole list.

10.**View** **ME list** : Creates a list of student selected for ME branch and displays the whole list.

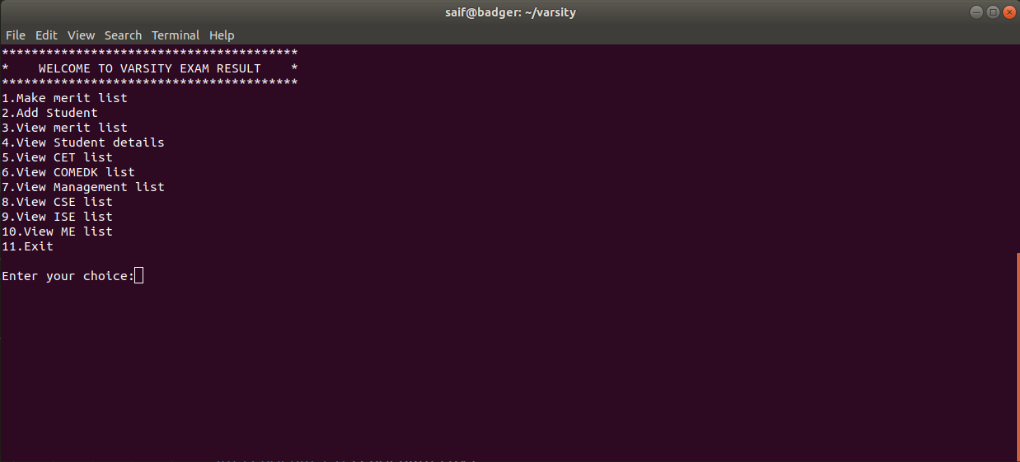
**CONCEPTS USED**

The execution of this program inculcates many concepts of the subject. They include:

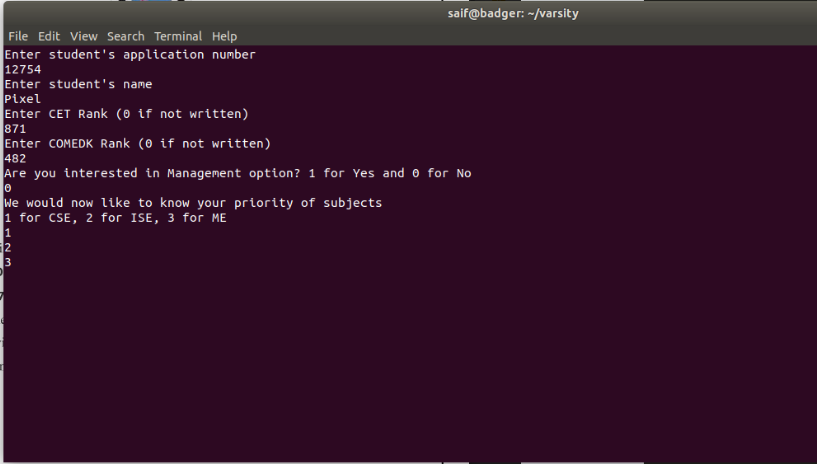
1. File handling (File.txt, CETList.txt, COMEDKList.txt, ManagementList.txt)
2. Classes and Instances
3. Inheritance
4. Basic loops and statements
5. Global variables and functions
6. Arrays
7. Functions

**SCREENSHOTS**

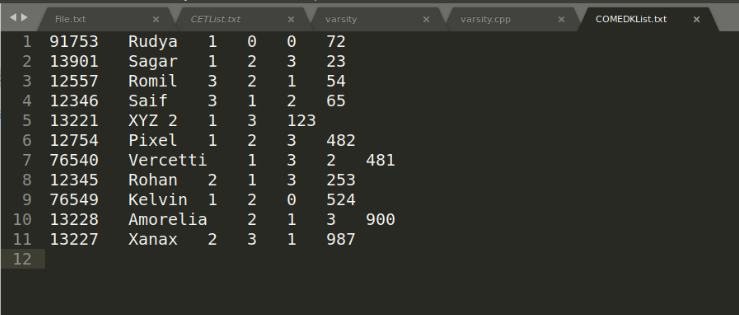
1. Welcoming screen

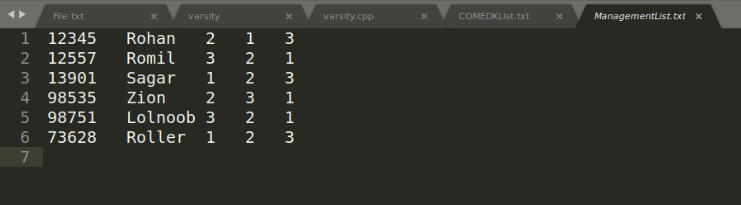


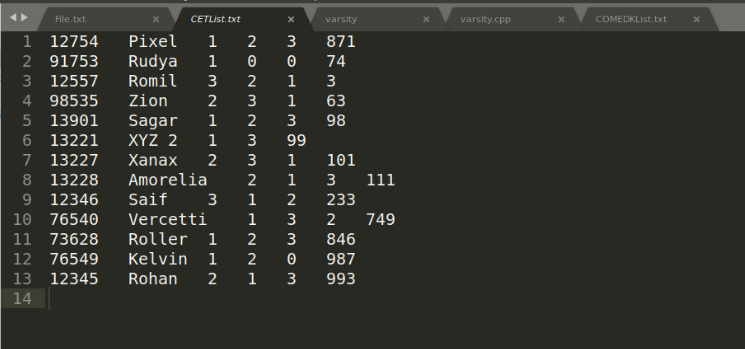
1. Student Details Insertion



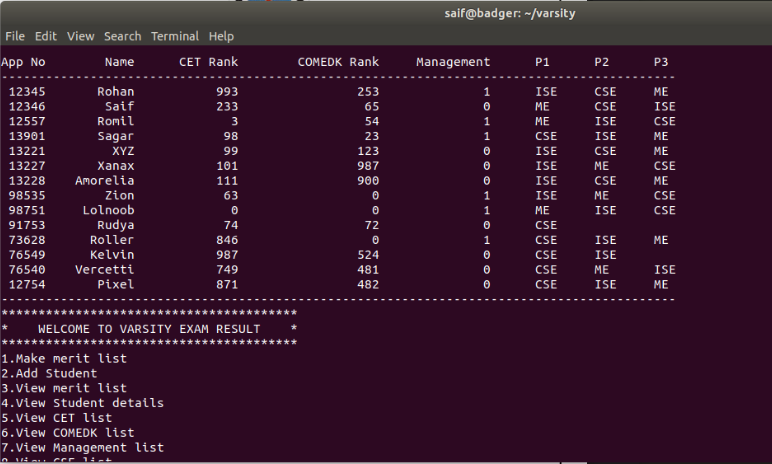
1. The three QUOTA Lists: COMEDK List, Management List, CET List respectively



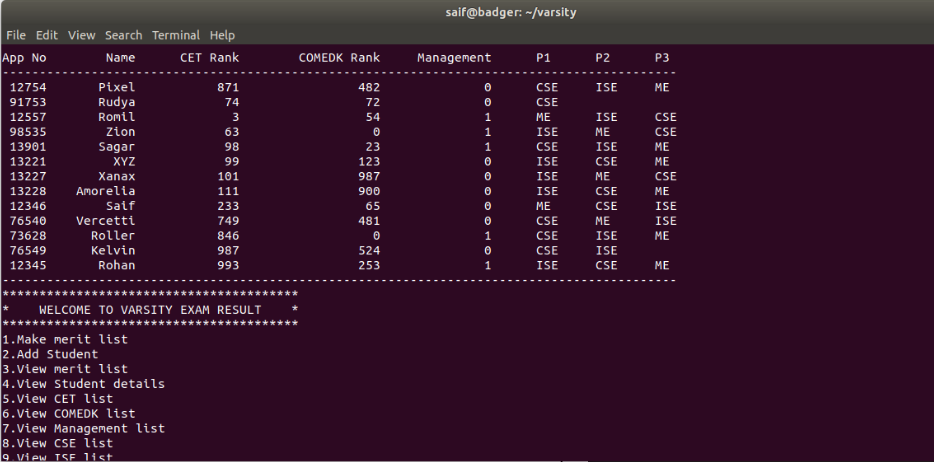




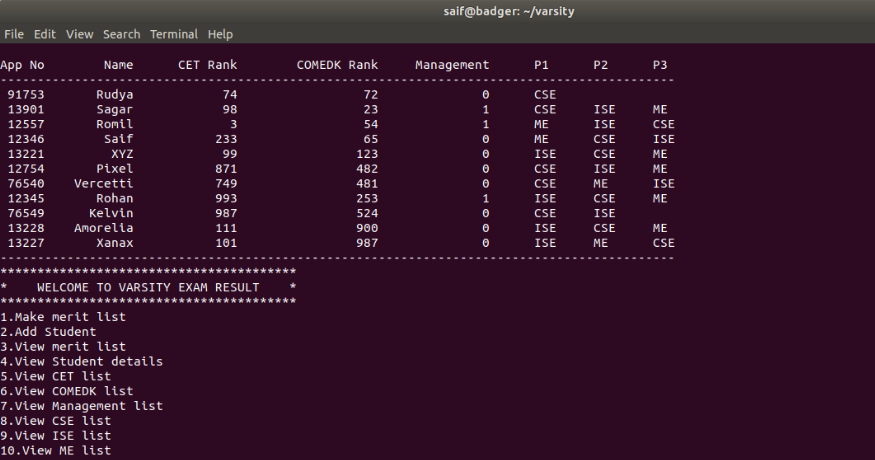
1. Merit List



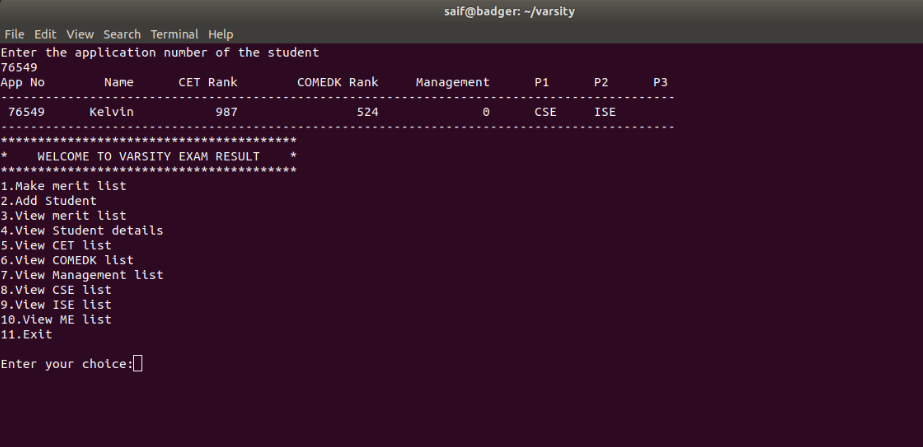
1. CET List



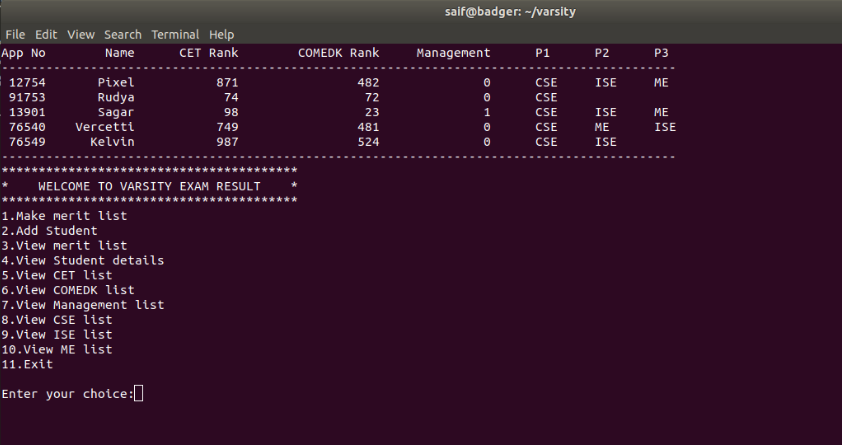
1. COMEDK List



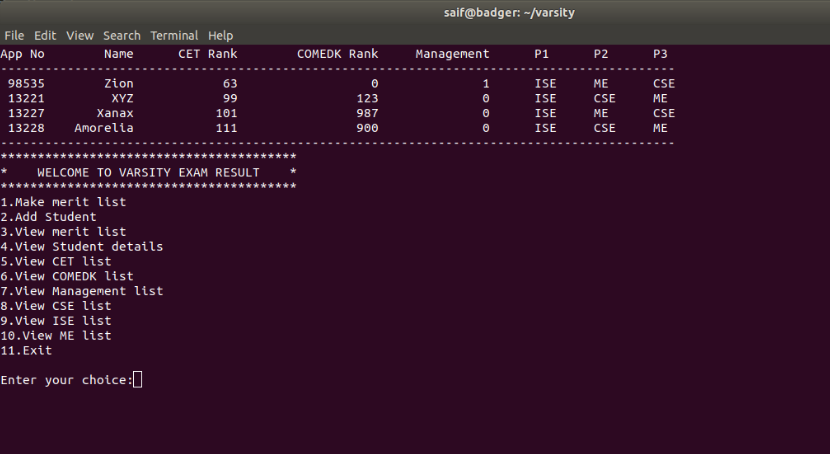
1. Specific Student Details



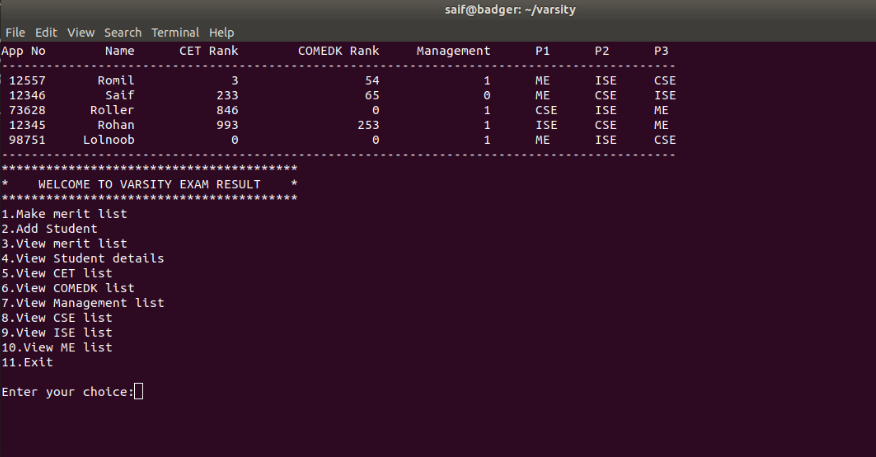
1. CSE List



1. ISE List



10.ME List



SHORTCOMINGS/ LIMITATION

Right now, the only limitation is the application of college side cut offs for each course and the implementation of multiple rounds which are usually implemented at the time of admission so as to allow students to alter their priorities.

**BIBLIOGRAPHY**

* For details on the C++ language: <https://en.wikipedia.org/wiki/C%2B%2B>
* Static variable reference: <https://www.geeksforgeeks.org/static-keyword-cpp/>
* For C++ reference: <https://www.tutorialspoint.com/cplusplus/>
* For C++ resource: <http://www.cplusplus.com>
* For general C++ OOP related queries:

https://cboard.cprogramming.com/cplusplus-programming/131854-help-cin-using-classes.html