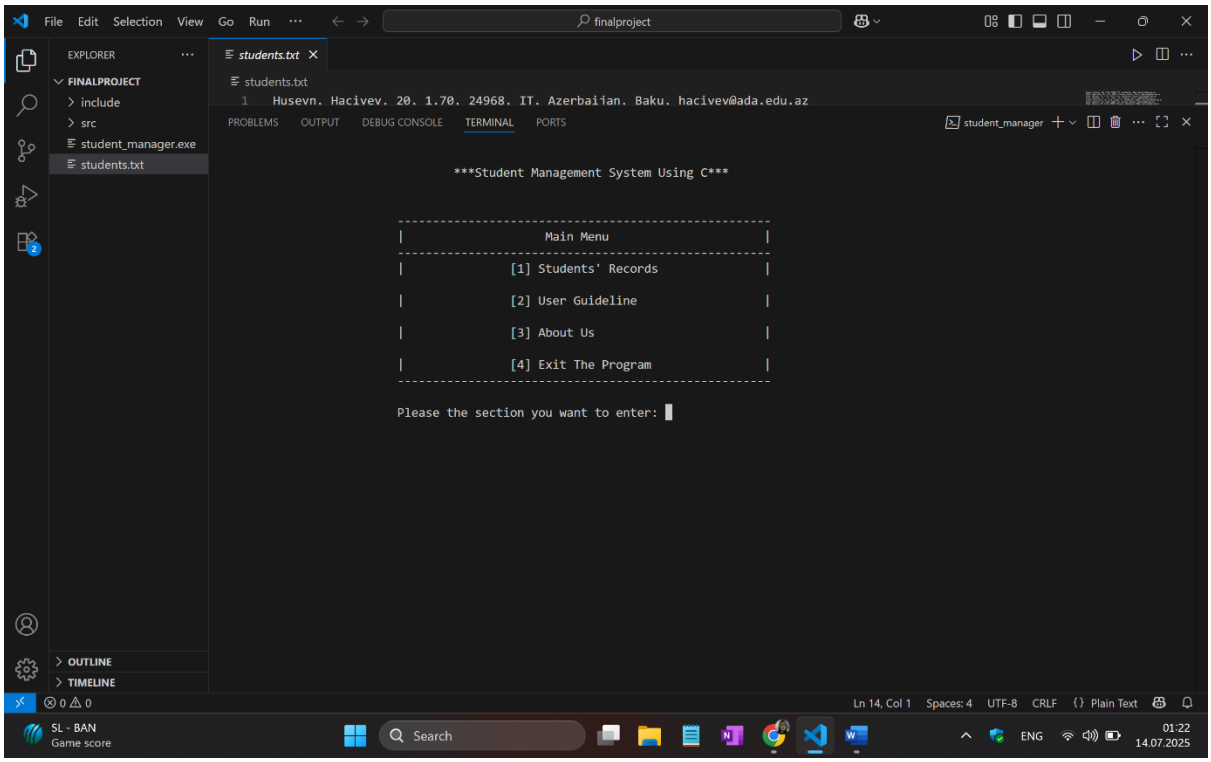
Final Project Guide Book

General Info:

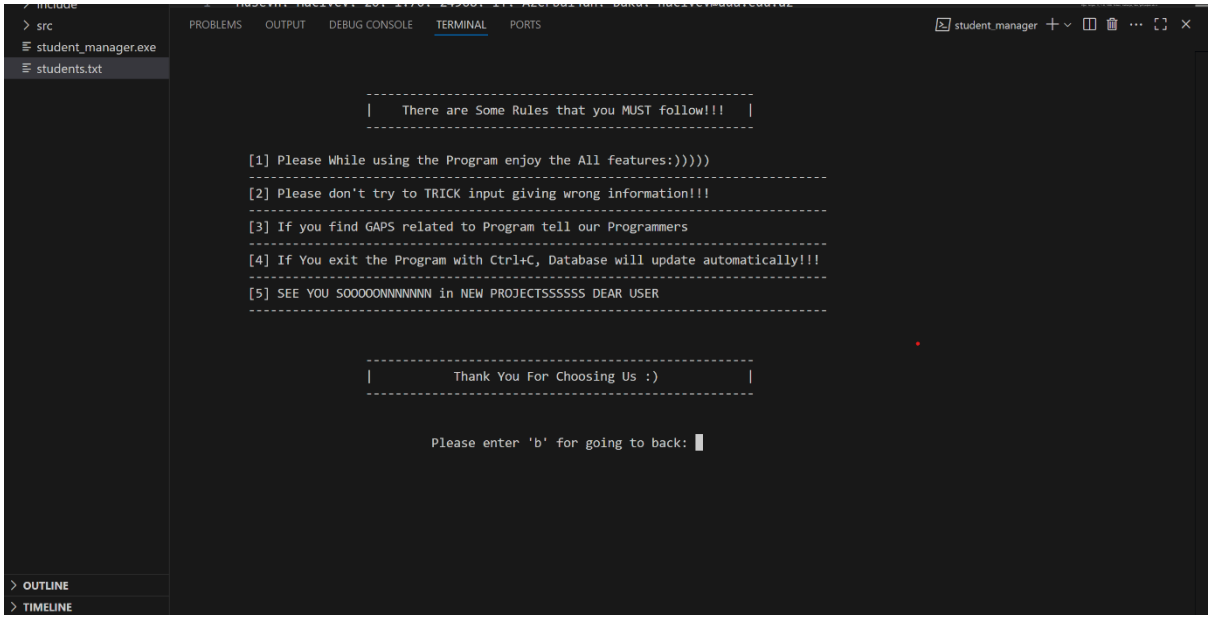
This program is a **Student Management System**, designed to help users manage student records efficiently. It allows adding, listing, editing, deleting, searching, filtering, and sorting student information. The interface is fully terminal-based, with clear formatting and simple navigation.

How to Use the Program:

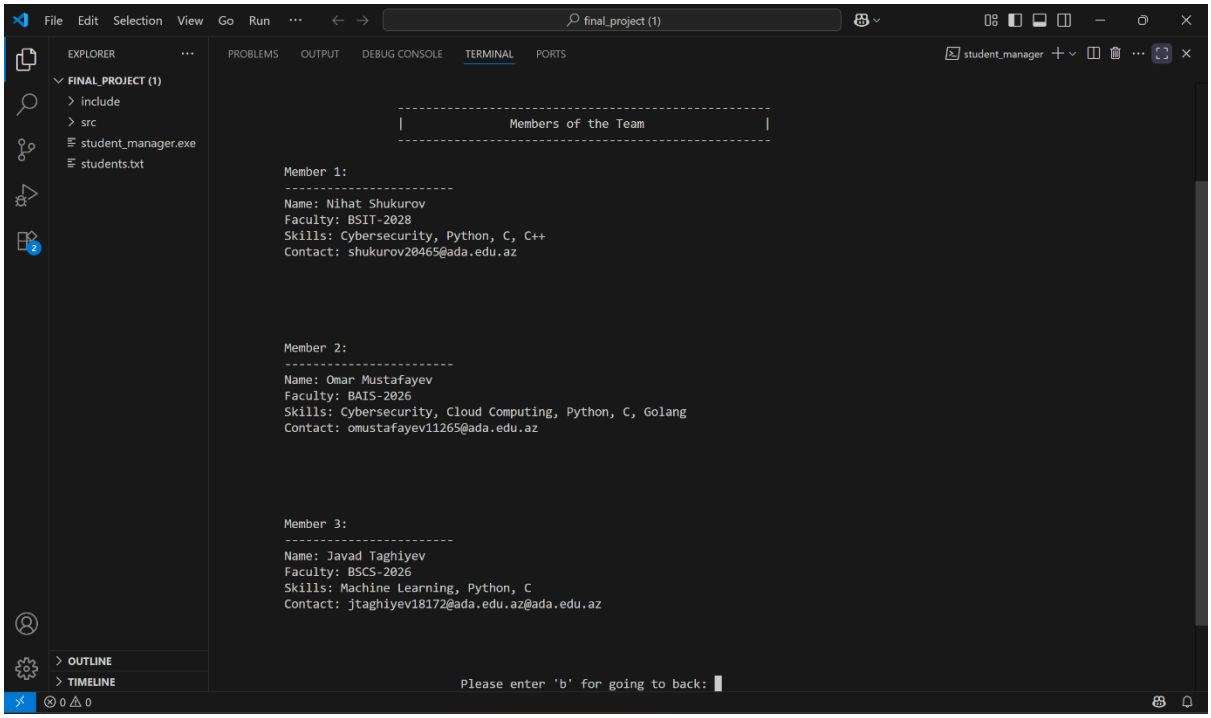
Hello dear user, after running “exe” file, you will see the **Main Menu.** This menu is starting of our program which everyhing starts here.



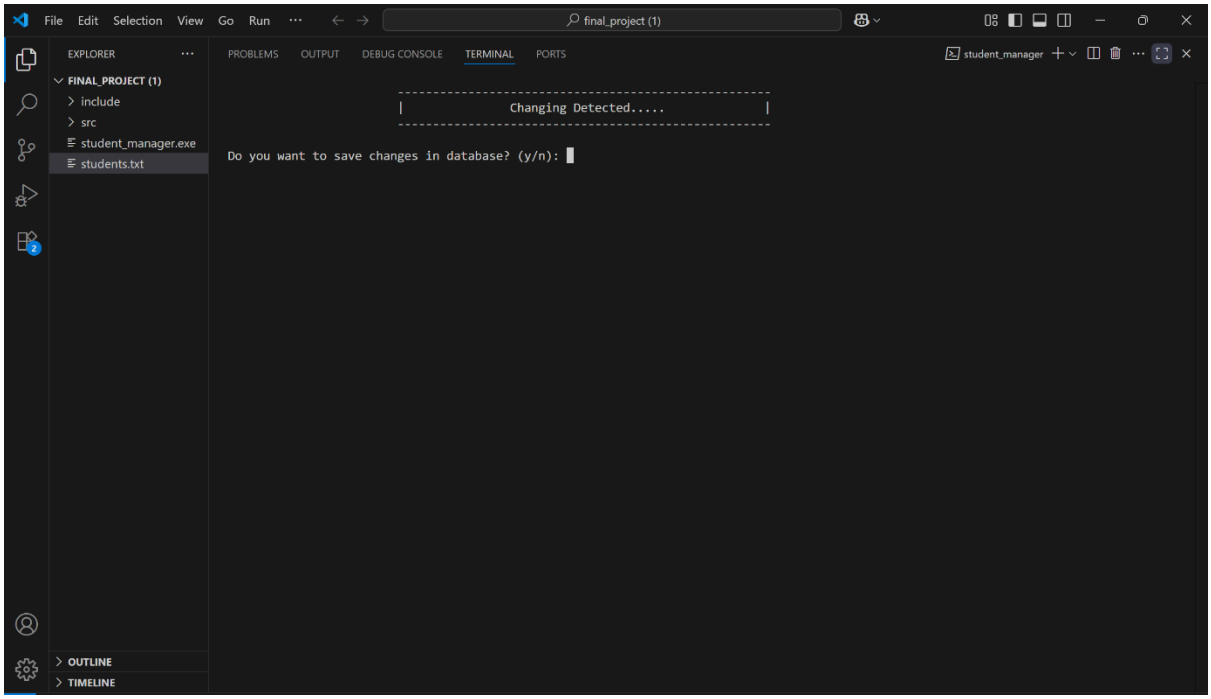
As you can see our interface has so clear and efficient design. We will talk about **First Section** later that is why l want to explain other sections’ job. **Second Section’s** job is introducing program’s rules to the user for better functionality. There are some rules you can read.



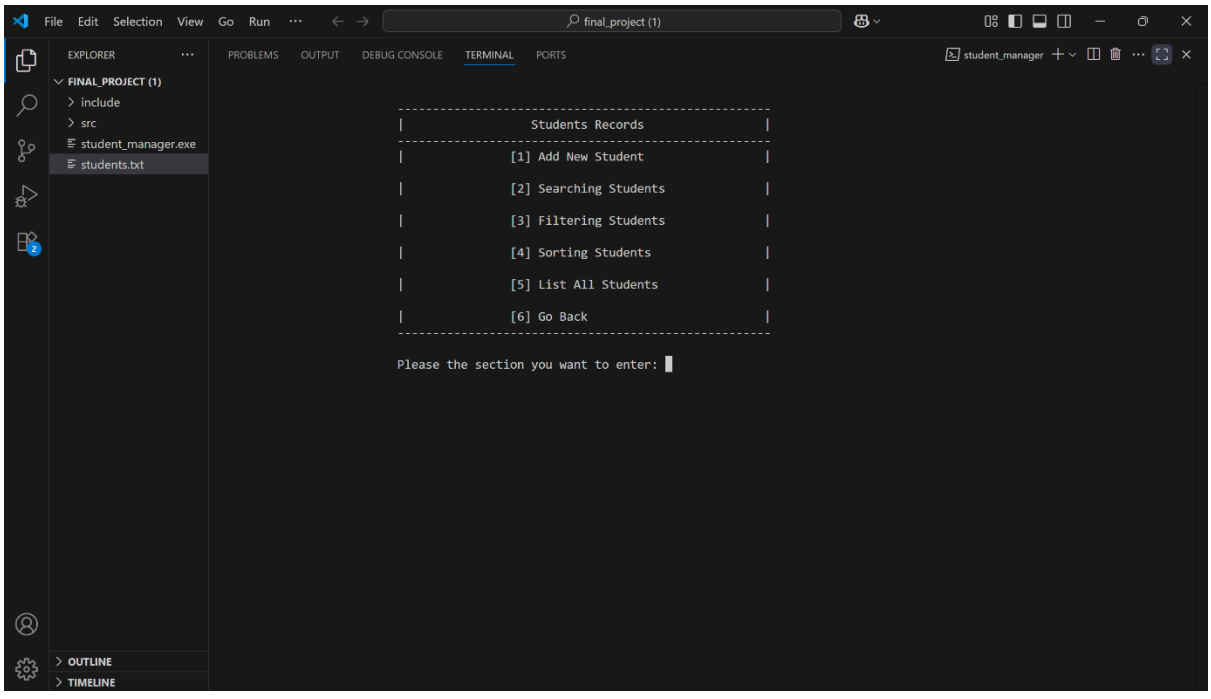
Next we have **Third Section** which creates connetion between developer and user. Everyone knows that in every program has some bugs that users discover while using the program. We think that we can share our contact information with users who can share issues, bugs with us for preventing these problems.



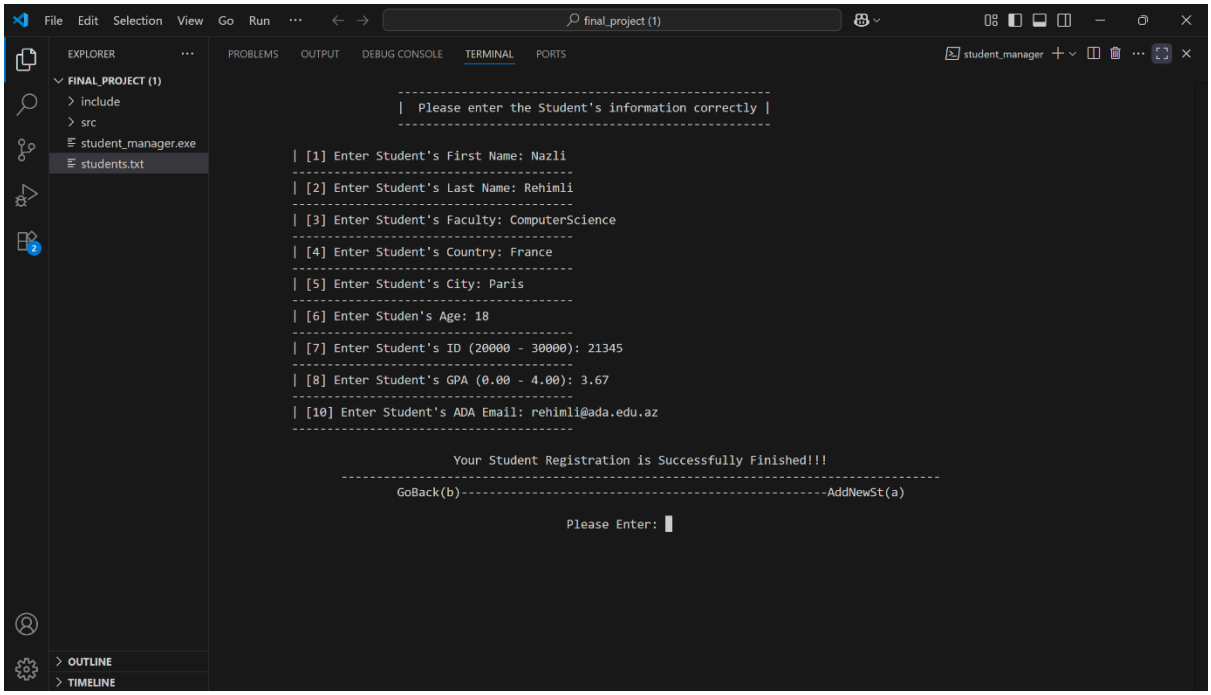
Fourth Section is used for saving the changes in database, when user using program, “adding”, “editing”, and “deleting” functions can change the students’ information. That is why we added this option, if user wants to save changes or not, he or she can use this function for clearify the circumstance. if function detects the diferent database, it asks to user save or not, both option it exits the program.



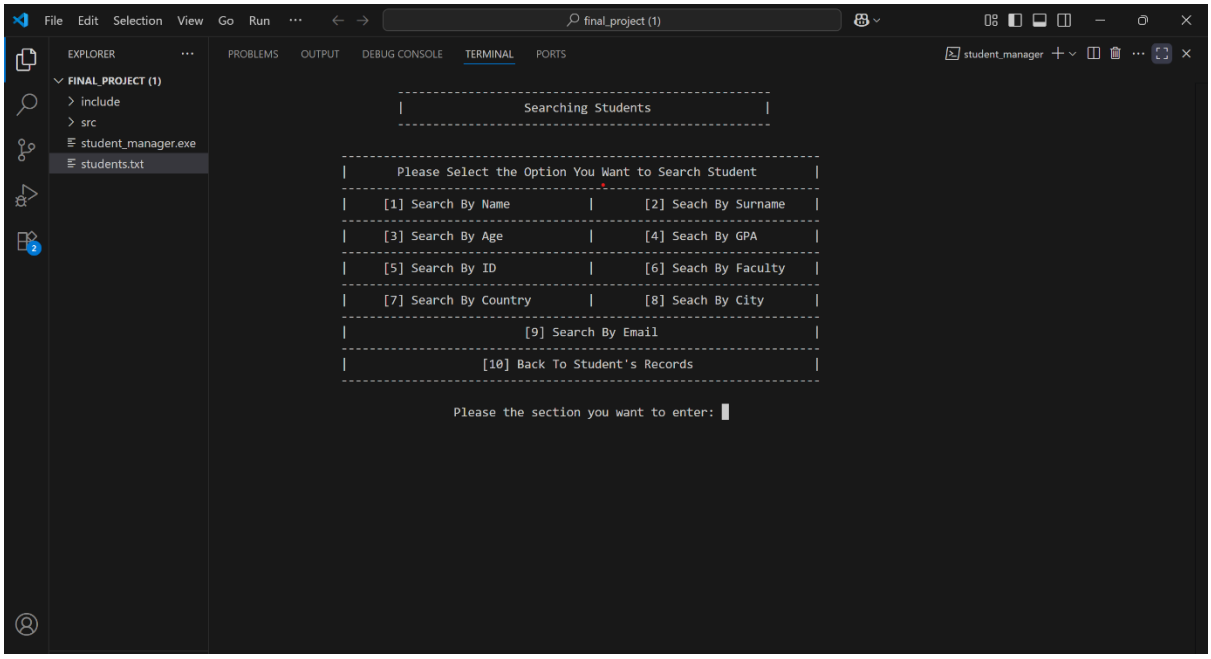
We said that we will explain the **First Section** deeply. When users entered it, they will see 6 Sections like that.

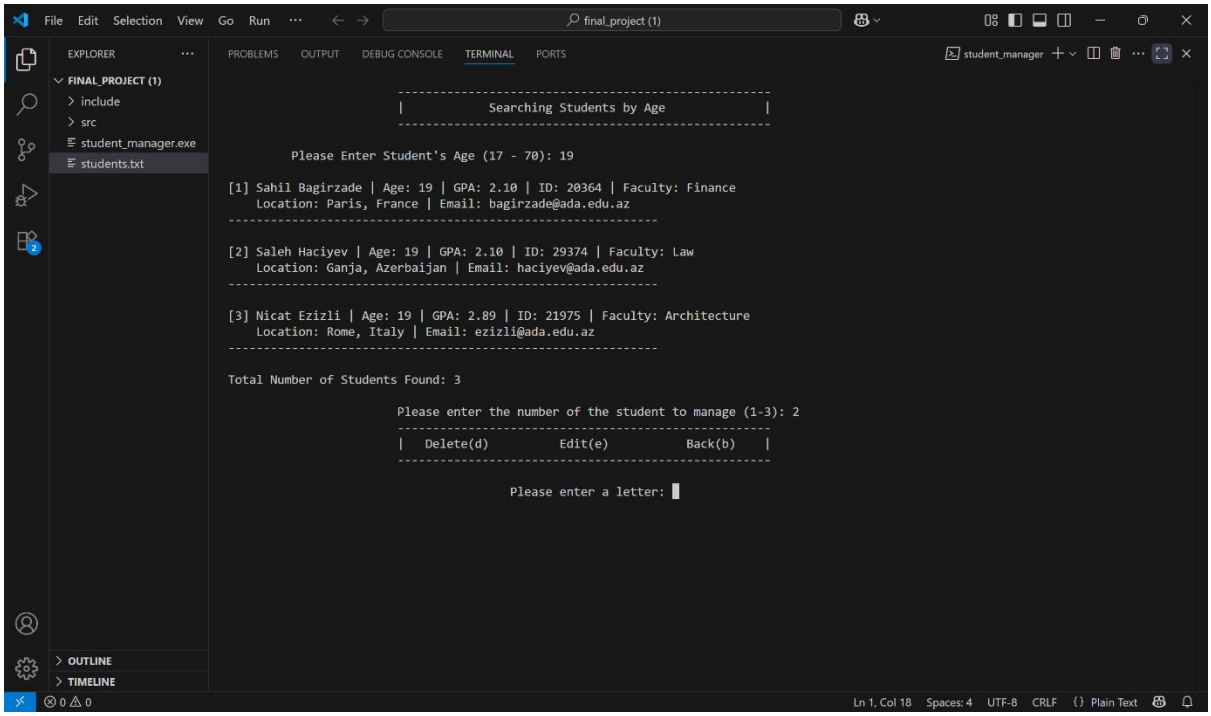


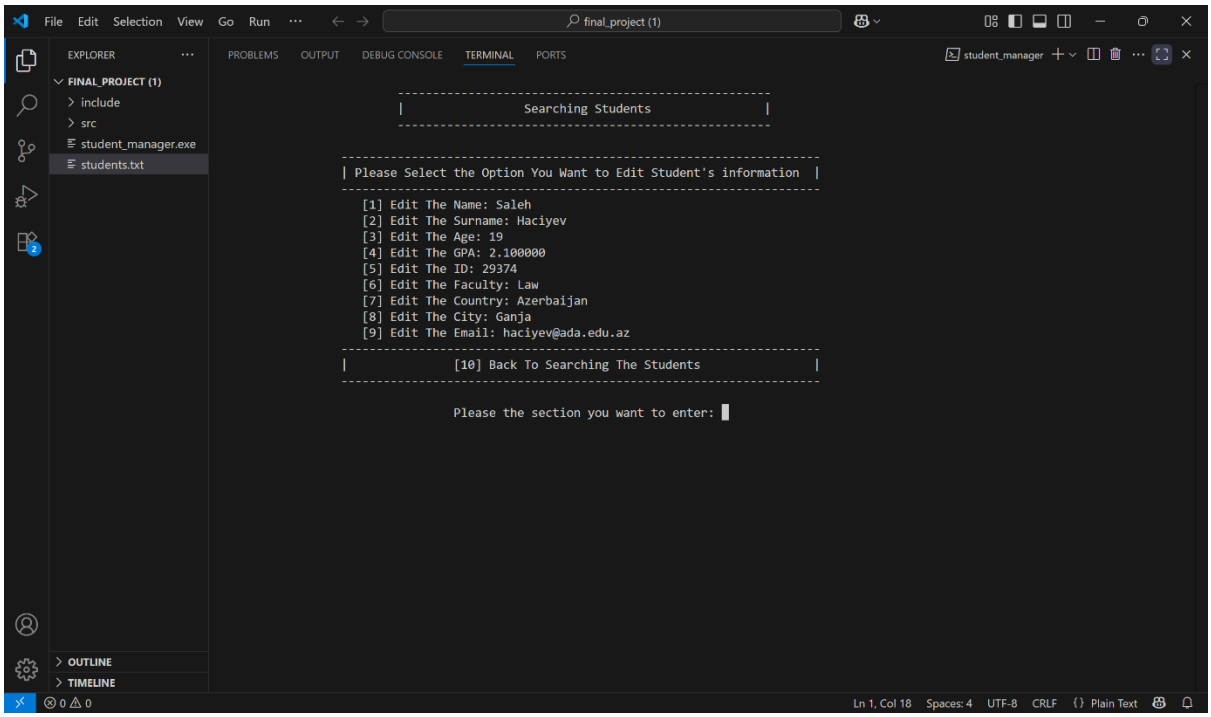
I want to start with **First Section of Students' Records**. This section is used for creating a new student identity for increasing database information. There are 9 contact variables, users can enter like name, surname, id, age, gpa, faculty, country, city, and email address. One of the most widespread issues about adding student information is preventing wrong input errors like adding numbers instead of letter or space. That is why in every section users can not enter invalid value for tricking the program. After addding everything program asks for exiting or contuneing to create students. Our program contains dynamic memory allocation feature, so in every command database updates itself.



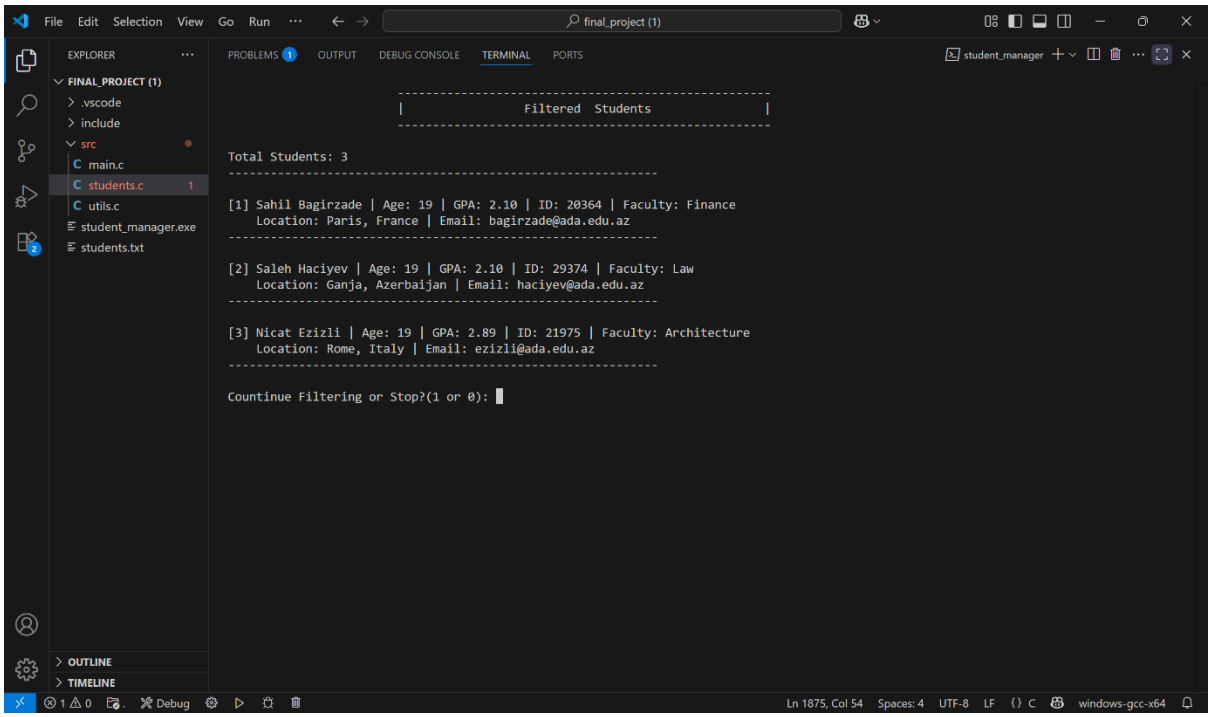
Second Section features are exciting more than previous section. Users can search students' information in every contact, and after finding students can alter every information in program as you see from screenshots.



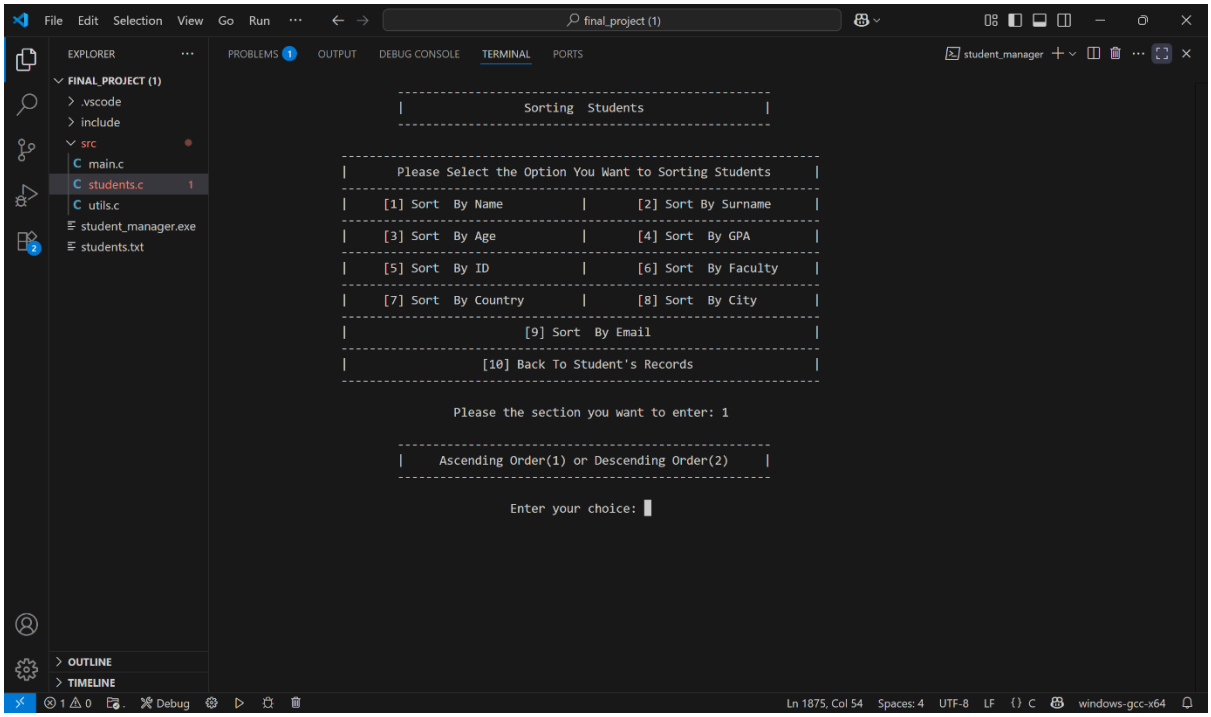

File Edit Selection View Go Run
final_project (1)
EXPLORER
FINAL_PROJECT (1)
include
src
student_manager.exe
students.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
student_manager
Searching Students by Age
Please Enter Student's Age (17 - 70): 19
[1] Sahil Bagirzade | Age: 19 | GPA: 2.10 | ID: 20364 | Faculty: Finance
Location: Paris, France | Email: bagirzade@ada.edu.az
[2] Saleh Haciyev | Age: 19 | GPA: 2.10 | ID: 29374 | Faculty: Law
Location: Ganja, Azerbaijan | Email: haciyev@ada.edu.az
[3] Nicat Ezizli | Age: 19 | GPA: 2.89 | ID: 21975 | Faculty: Architecture
Location: Rome, Italy | Email: ezizli@ada.edu.az
Total Number of Students Found: 3
Please enter the number of the student to manage (1-3): 2
| Delete(d) Edit(e) Back(b) |
Please enter a letter:
OUTLINE
TIMELINE
Ln 1, Col 18 Spaces: 4 UTF-8 CRLF Plain Text

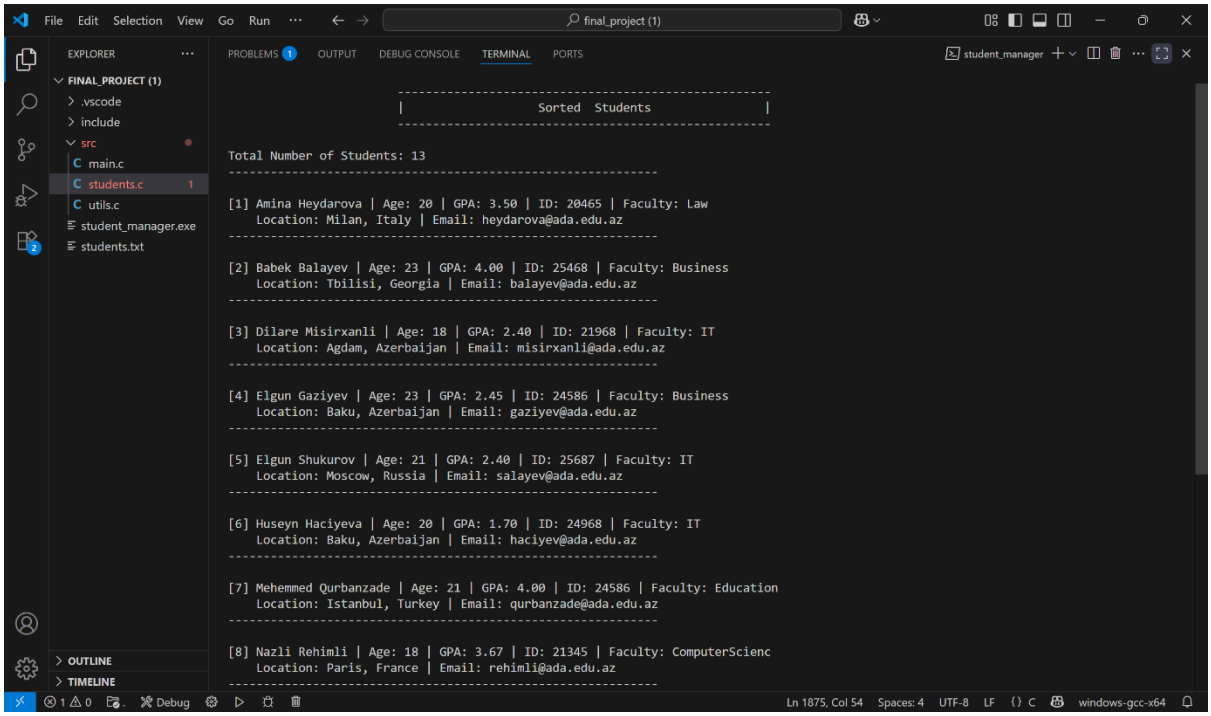

File Edit Selection View Go Run
final_project (1)
EXPLORER
FINAL_PROJECT (1)
include
src
student_manager.exe
students.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
student_manager
Searching Students
Please Select the Option You Want to Edit Student's information
[1] Edit The Name: Saleh
[2] Edit The Surname: Haciyev
[3] Edit The Age: 19
[4] Edit The GPA: 2.100000
[5] Edit The ID: 29374
[6] Edit The Faculty: Law
[7] Edit The Country: Azerbaijan
[8] Edit The City: Ganja
[9] Edit The Email: haciyev@ada.edu.az
[10] Back To Searching The Students
Please the section you want to enter:
OUTLINE
TIMELINE
Ln 1, Col 18 Spaces: 4 UTF-8 CRLF Plain Text

We can talk about **Third Section** which contains filtering students multiple features like name, surname, age, etc. in one attempt. After finishing the filtering students, users can start again when choose the 0 for stoping and starting with new array of database. İf users choose 1, program remembers 3 students and starts filtering them until user finish.

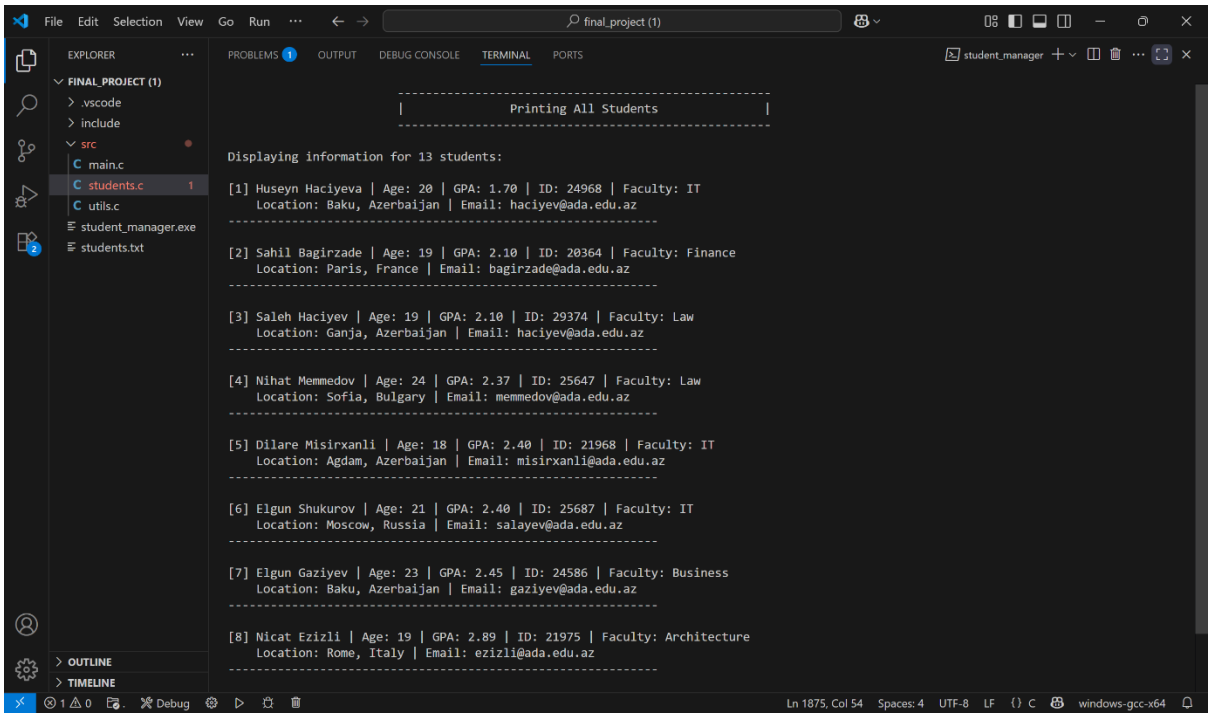


Furthmore, I can demostrate **Fourth Section,** doing sorting students again every contact information with giving choice like Ascending or Descending. İ don't want to talk about much, you can see from screenshots.





İn **Fifth Section** has not much functionality, it is just printing all students who stored in database.



Contribution Table And Introducing The Team Members:

Member1: Nihat Shukurov

Member2: Omar Mustafayev

Member3: Javad Taghiyev

After Nuraddin teacher introducting the Final Project, we have discussed how can we do that the most professional standards for getting full point. We have reseached, gathered information, analysed many examples for starting with best knowledge to Project. When we created the plan and splitting every responsibility among the members, we were ready for starting. After long nights and hardworking we could do it excellent job.

"**Dream it, Plan it, Do it"**

Contribution Table

Name of The Member	Responsibiliy of Member	Countribution Percentage(%)
Nihat Shukurov	Creating Plan of Project and Splitting Responsibilities, Adding,Filtering, Sorting Function, Supervising whole process, Creating connection among functions, Writing Report	**33.3%**
Omar Mustafayev	Writing structures and logic, Seperating .h and .c files, Writing Exit, List_All_Students Functions, Debugging Bugs,	**33.3%**
Javad Taghiyev	Writing code structures and logic,, Interface of Program, Writing Searching Function (with edit and delete)	**33.3%**

Thank You For Your Attention!