BCS 215: UNIX Operating Systems

CRN 10323 JL Intersession 2019

Final Project

Due Thursday, January 17, 11:59PM on Blackboard

This project is to acquaint you with fundamental UNIX commands and utilities when creating a multifunctional application. You will practice on managing shell variables, using shell operators, employing selection and repetition logic structures, creating a menu-based program, listing contents of a file, adding records to a file, searching patterns in a file, deleting records from a file, and using UNIX commands/utilities to manage cursor and output format, etc. The completion of the project will strengthen your knowledge and skills of utilizing UNIX commands/utilities and UNIX shell script programming language.

In this project you will build a menu-based application for tracking student information. The specifications of the project are as follows:

- (10 points) It is a menu-based program with a main screen offering the user options to view records, add new records, search for records, delete records, and quit the program, as shown in Figure 2. Use a case statement to determine what choice the user made, and perform the appropriate action. The only way to exit the application is by pressing "5" to quit the program. All other options should return to the main menu. Your program should handle the case in which the user doesn't enter anything, or the user enters an invalid option.
- (5 points) The student records are stored in a text file called `student_data`, as shown in Figure 1. The file contains information about a student's ID, last name, first name, program name, and total credits. Each record is stored on one line and has the above five fields separated by the colon ":" character. The initial data file contains the following records:

```
1001:Jones:Alice:Accounting:105
1320:Moore:Sarah:Mechanical Engineering:80
2100:Olson:Timothy:Computer Systems:66
1520:Mitchell:Barbara:Electrical Engineering:34
1800:Smith:Paul:Mathematics:87
2010:Whitman:Henry:Business Management:114
1480:Richard:Johnson:Computer Systems:90
1234:Willims:David:Economics:78
```

- (20 points) The **View Records** option should display a sub-menu, as shown in Figure 3, to let user choose how the records are sorted before they are displayed. The records can be sorted by last name or program name in alphabetic order, or sorted by credits in ascending order, or sorted by ID in ascending order by default. The records should be displayed in a neat fixed-length form as shown in Figure 4. The total number of records should also be displayed. The program returns to the main menu after records are displayed and the "Enter" key is pressed.
- (15 points) The **Add New Records** option should allow the user to enter the student ID, last name, first name, program name, and total credits. This information will be appended to the end

of the file `student_data`, all on one line, separated by the colon ":" character. The last name cannot be blank for the new record to be written into the data file. If a student ID has already existed in the data file, display an error message and let the user reenter the data (see Figure 5). The program should allow the user to add more records until a 'q' for quit is entered (see Figure 6, 7). The program returns to the main menu after the user enters a 'q'. Modifying the previous field when the user types in '-' or exiting if the user types in 'q' is optional.

- (5 points) The **Search for Records** option should prompt the user for a string pattern to search for in the data file and display all the records that contain that pattern (see Figure 8).
- (10 points) The **Delete Records** option should prompt the user for a pattern, display all the records that contain that pattern, ask the user to confirm yes or no to delete these records (see Figure 9, 10), and perform the appropriate actions.
- (8 points) The source code of the above four options (view, add, search, and delete) should be placed in four separate script files.
- (2 points) The **Quit** option will exit the application.
- (10 points) Make a shell script called `functions` that defines the following three functions:

  `sort_ID` to sort the records in the ascending order of student ID,
  `sort_lname` to sort the records in the alphabetic order of student last name,
  `sort_program` to sort the records in the alphabetic order of program name,
  `sort_credits` to sort the records in the ascending order of total credits.

  Use function(s) in the View Records module to sort the records according to user's choice.
- (5 points) The program should have necessary comments including, but not limited to, the script name, author's name, date information, and the purpose of the script.
- (10 points) Make a documentation file that have enough snapshots to show the states of your program execution, for example, the main menu, and each menu option, action/results, etc. It is highly suggested that you include snapshots similar to the ones in this document. Your snapshots should be legible and include the heading of the putty terminal window showing your username and host information.

Include your name, RAM ID, and the statement "**I certify that this submission is my own original work**" in your documentation.

Your project will be graded based on the completeness and correctness of your work according to the specifications of the project.

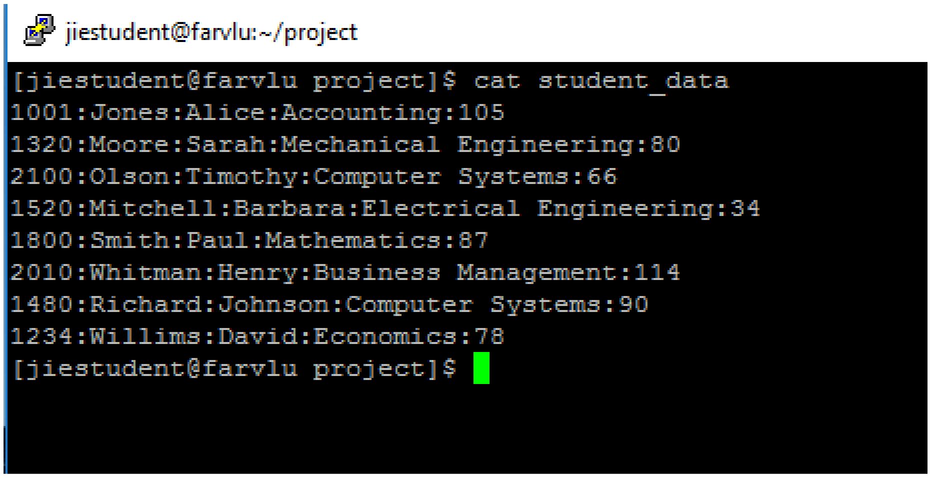
Hints: Be sure to complete and understand hands-on exercises of chapter 6 and 7 before working on the project.

Submission:
Submit a ZIP file that contains all your shell script files and the project documentation file into the *BCS215_Project_Submission* under the *Learning Modules > Module 4 > Course Project* folder on Blackboard before the deadline. **No late submission will be accepted.**

You can follow the instructions of Install-WinSCP to download and install WinSCP, an open source software that supports secure file transfer. With WinSCP you can transfer all you script files from the server to your local machine. Place your script files in the same folder as your documentation file and zip them into a ZIP file.

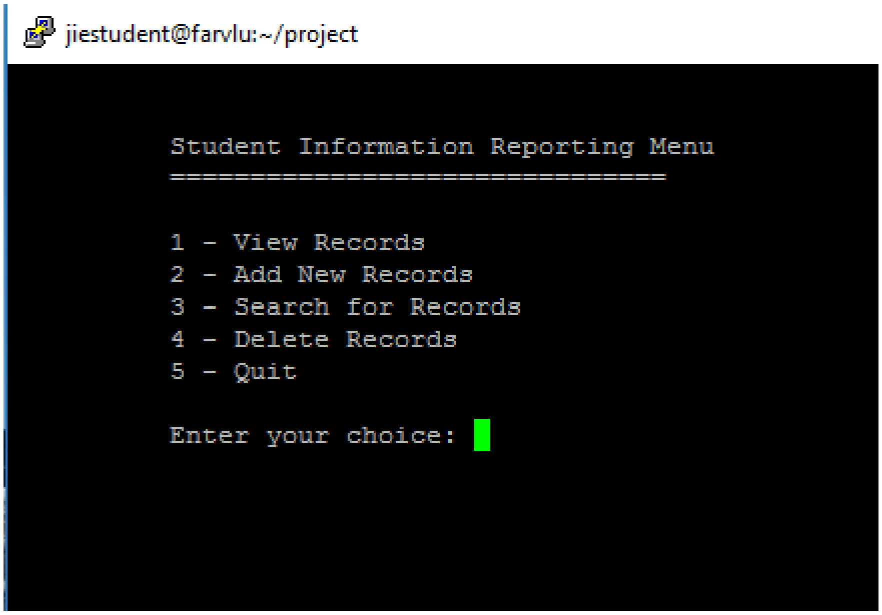
Snapshot Examples:

Figure 1: student_data file



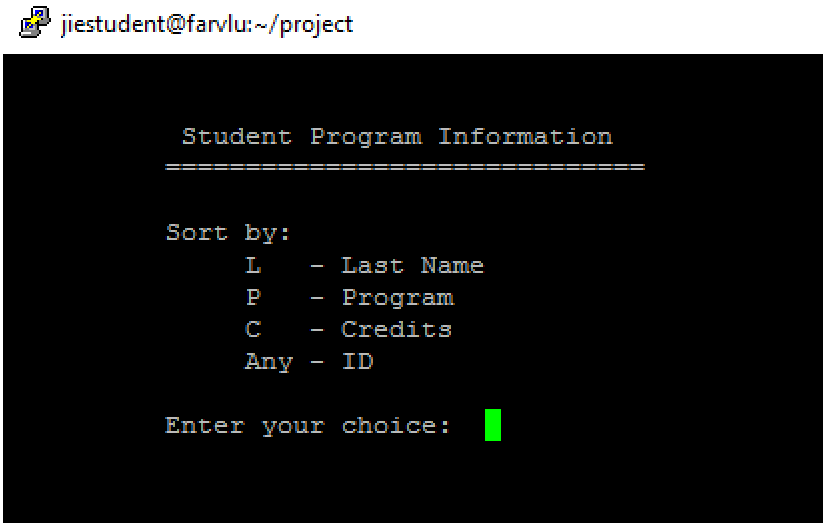
```
[jiestudent@farvlu project]$ cat student_data
1001:Jones:Alice:Accounting:105
1320:Moore:Sarah:Mechanical Engineering:80
2100:Olson:Timothy:Computer Systems:66
1520:Mitchell:Barbara:Electrical Engineering:34
1800:Smith:Paul:Mathematics:87
2010:Whitman:Henry:Business Management:114
1480:Richard:Johnson:Computer Systems:90
1234:Willims:David:Economics:78
[jiestudent@farvlu project]$
```

Figure 2: Main menu



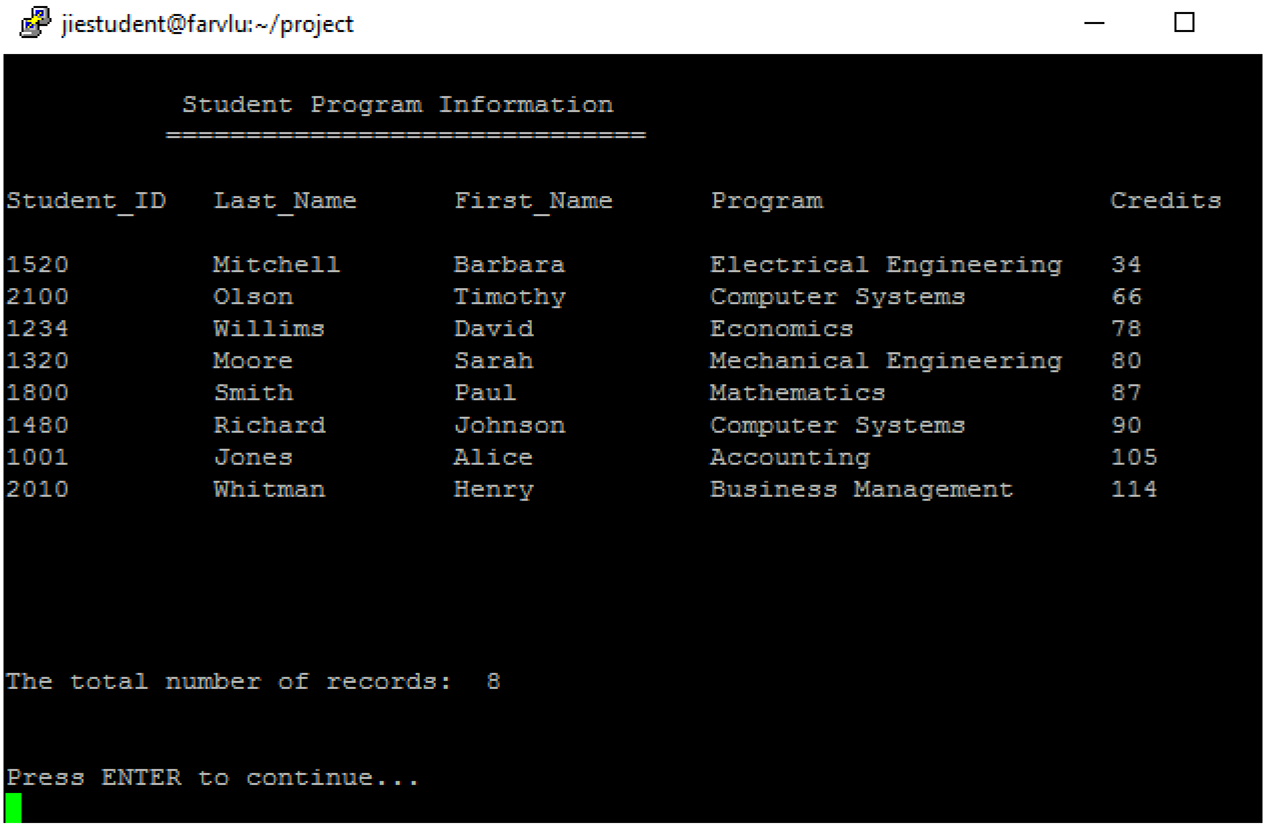
```


          Student Information Reporting Menu
          ===============================

          1 - View Records
          2 - Add New Records
          3 - Search for Records
          4 - Delete Records
          5 - Quit

          Enter your choice:
```

Figure 3: View Records -- Menu



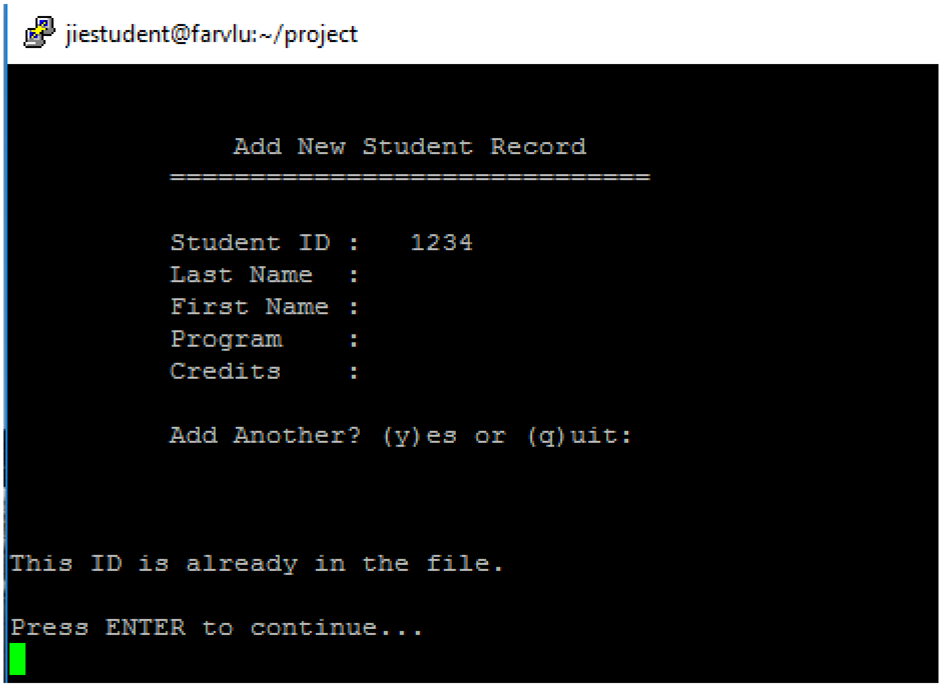
```
           Student Program Information
          ==============================

          Sort by:
               L   - Last Name
               P   - Program
               C   - Credits
               Any - ID

          Enter your choice:
```

Figure 4: View Records (The data are sorted on Credits)



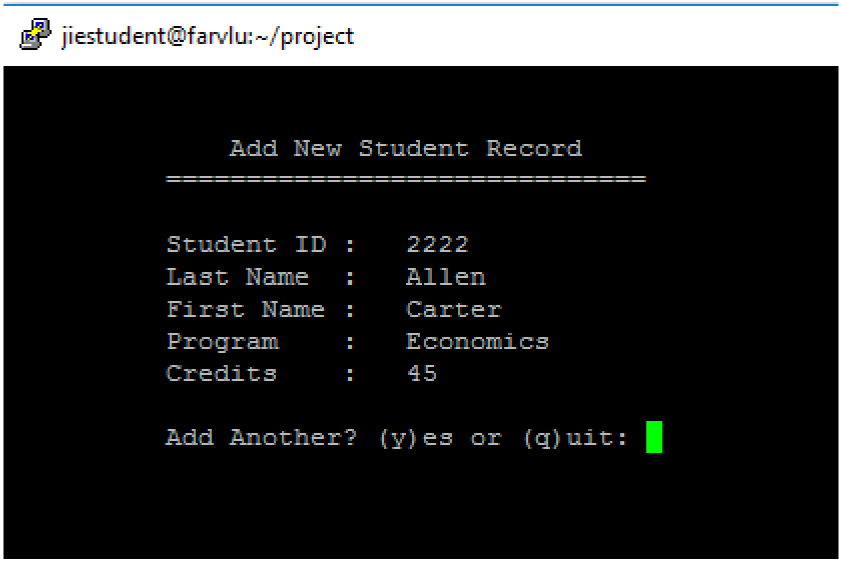
```
           Student Program Information
          ==============================

Student_ID   Last_Name      First_Name      Program                  Credits

1520         Mitchell       Barbara         Electrical Engineering   34
2100         Olson          Timothy         Computer Systems         66
1234         Willims        David           Economics                78
1320         Moore          Sarah           Mechanical Engineering   80
1800         Smith          Paul            Mathematics              87
1480         Richard        Johnson         Computer Systems         90
1001         Jones          Alice           Accounting               105
2010         Whitman        Henry           Business Management      114




The total number of records:  8


Press ENTER to continue...
```

Figure 5: Add Records – handling duplicated student ID



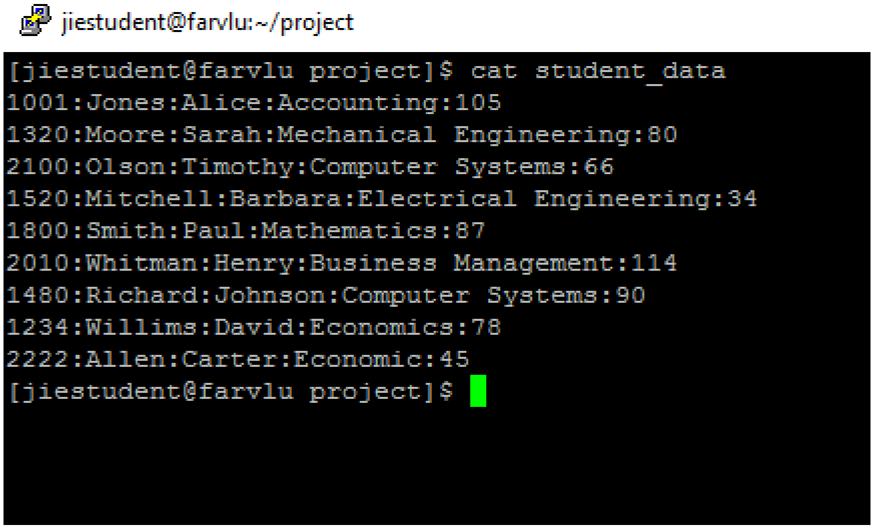
```
                Add New Student Record
          ==============================

          Student ID :    1234
          Last Name  :
          First Name :
          Program    :
          Credits    :

          Add Another? (y)es or (q)uit:



This ID is already in the file.

Press ENTER to continue...
```

Figure 6: Add Records – adding a record



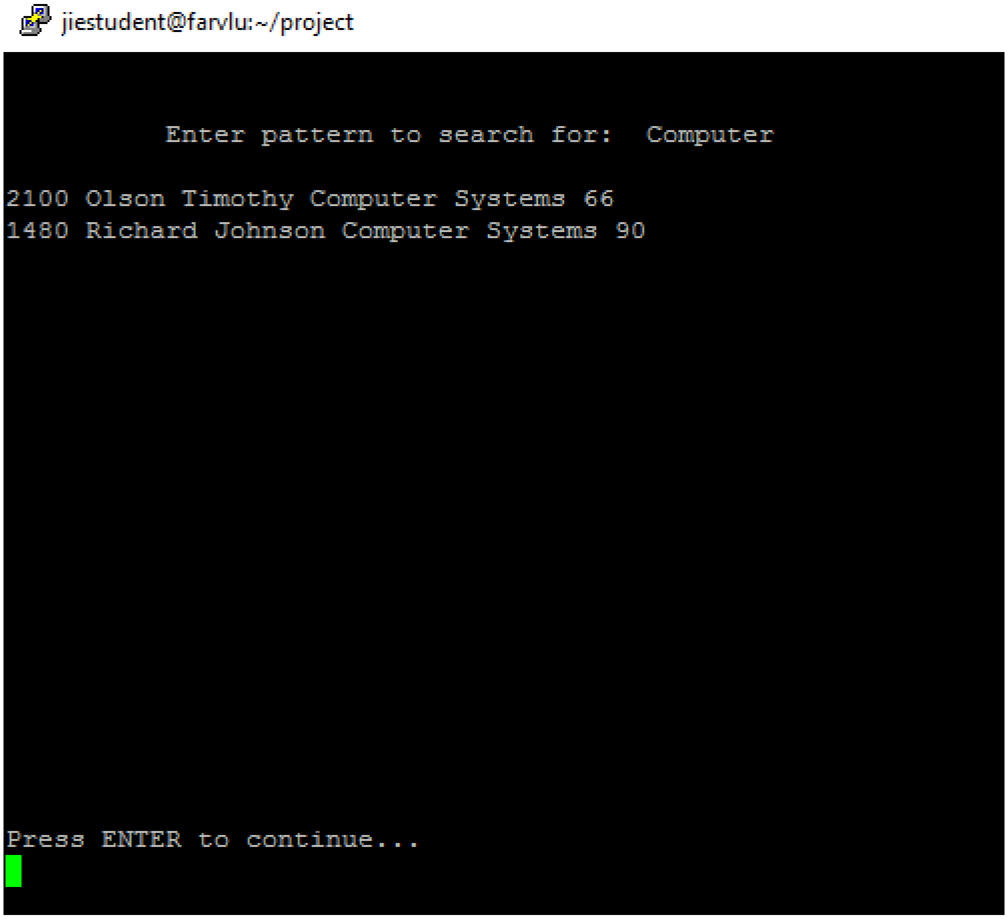
```
                Add New Student Record
          ==============================

          Student ID :    2222
          Last Name  :    Allen
          First Name :    Carter
          Program    :    Economics
          Credits    :    45

          Add Another? (y)es or (q)uit:
```

Figure 7: `student_data` file after adding a new record



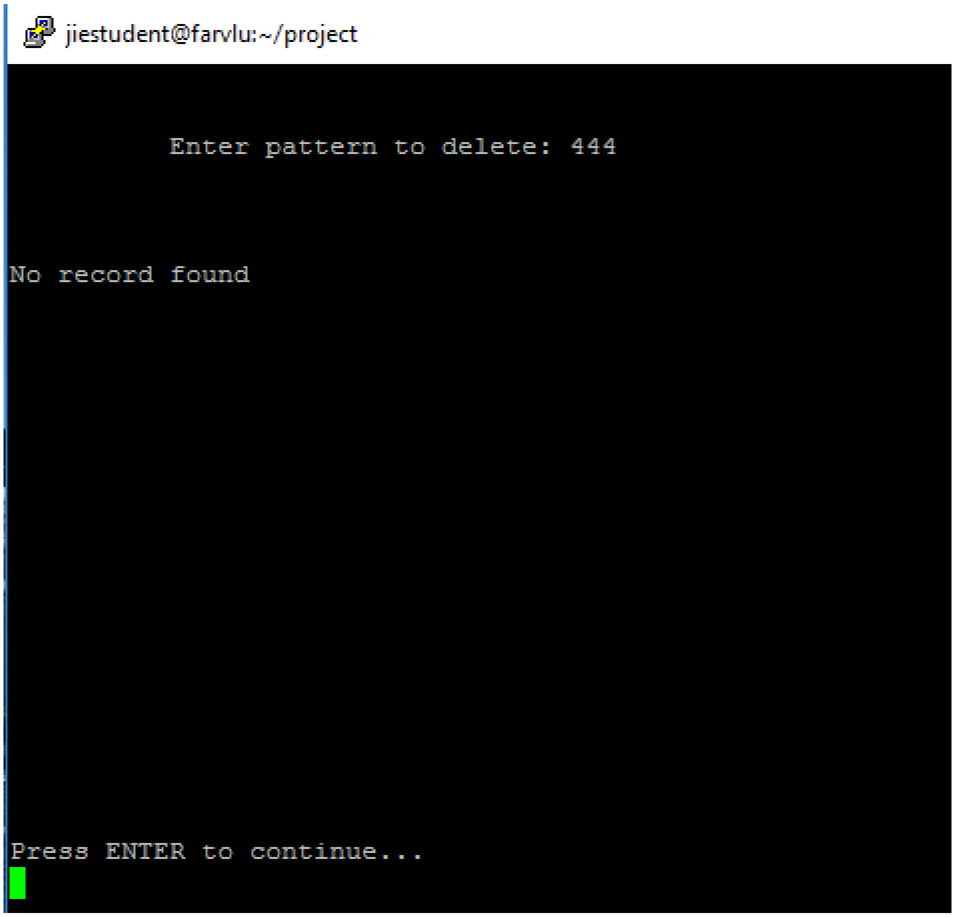
```
[jiestudent@farvlu project]$ cat student_data
1001:Jones:Alice:Accounting:105
1320:Moore:Sarah:Mechanical Engineering:80
2100:Olson:Timothy:Computer Systems:66
1520:Mitchell:Barbara:Electrical Engineering:34
1800:Smith:Paul:Mathematics:87
2010:Whitman:Henry:Business Management:114
1480:Richard:Johnson:Computer Systems:90
1234:Willims:David:Economics:78
2222:Allen:Carter:Economic:45
[jiestudent@farvlu project]$
```

Figure 8: Search for Records



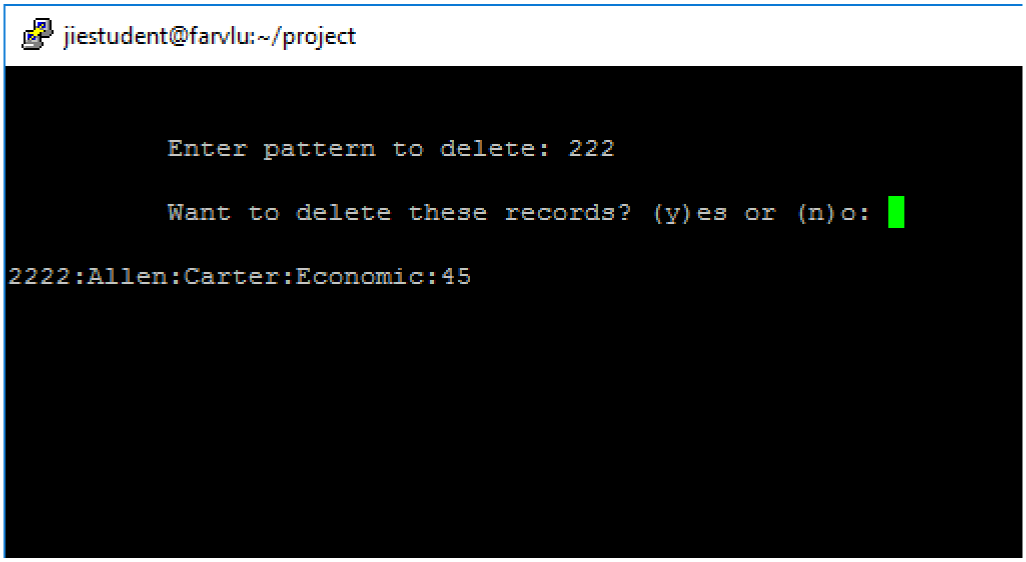
```


          Enter pattern to search for:  Computer

2100 Olson Timothy Computer Systems 66
1480 Richard Johnson Computer Systems 90


















Press ENTER to continue...
```

Figure 9: Delete Records (No record found)



```
          Enter pattern to delete: 444


No record found









Press ENTER to continue...
```

Figure 10: Delete Records



```
          Enter pattern to delete: 222

          Want to delete these records? (y)es or (n)o:

2222:Allen:Carter:Economic:45
```