



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Semester: (Summer, Year: 2025), B.Sc. in CSE (Day)

Nebula-Navigator

Course Title: Operating System Lab

Course Code: CSE 310

Section: 223_D2

Students Details

Name	ID
Md.Shajalal	223002088
Md.Hridoy Mia	223902010

Submission Date: 08 - 26 - 2025

Course Teacher's Name:Umme Habiba

For teachers use only:

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	2
1.1	Overview	2
1.2	Motivation	2
1.3	Problem Definition	2
1.3.1	Problem Statement	2
1.3.2	Complex Engineering Problem	3
1.4	Design Goals/Objectives	3
1.5	Application	4
1.6	Legal Document Signing	4
2	Design/Development/Implementation of the Project	5
2.1	Introduction	5
2.2	Project Details	5
2.3	Implementation	6
2.3.1	Tools and libraries	6
2.4	Algorithms	6
2.4.1	Code:	8
3	Performance Evaluation	14
3.0.1	Simulation Environment / Simulation Procedure	14
3.0.2	Development Environment	14
3.0.3	Results Analysis/Testing	14
4	Conclusion	17
4.1	Discussion	17
4.2	Limitations	17

Chapter 1

Introduction

1.1 Overview

Nebula Navigator is a Bash-based Galaxy Star Record System. It allows users to log in securely and manage star information by adding, viewing, searching, editing, and deleting records. The system also lets users check star distances and sort stars by distance.

All data is stored in text files, and Bash commands like `grep`, `awk`, and `sort` are used for processing. The terminal interface includes colorful menus and ASCII formatting, making it interactive and user-friendly.

This project shows how Bash scripting can create a menu-driven record management system with basic authentication, file handling, and data processing.

1.2 Motivation

The motivation for developing Nebula Navigator is to easily manage information about stars. This project helps users store, view, search, edit, and delete star records in a simple text-based system.

It also allows learners to practice Bash scripting, including file handling, menus, and basic authentication, while creating a fun and interactive star record system.

1.3 Problem Definition

1.3.1 Problem Statement

Managing and organizing information about stars manually can be time-consuming and prone to errors. There is a need for a simple, interactive system that allows users to store, retrieve, edit, and delete star records easily.

Most available astronomical databases are complex and require advanced knowledge to use. A lightweight, terminal-based system like Nebula Navigator can help students and enthusiasts learn about stars while practicing programming skills in Bash.

1.3.2 Complex Engineering Problem

This project can solve some complex engineering problem.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributes	Explain how to address
P1: Depth of knowledge required	To complete the Nebula Navigator project, the developer needs basic knowledge of Bash scripting, including loops, conditions, and functions. They should also understand how to work with text files and use commands like grep, awk, and sort. A basic understanding of CRUD operations and the ability to navigate a terminal environment are also required.
P2: Range of conflicting requirements	The main stakeholders for the Nebula Navigator project are students and astronomy enthusiasts who want a system that is both educational and easy to use. Developers must balance these user needs with technical constraints, ensuring the system is secure, functional, and lightweight. Conflicts may arise between usability, security, and functionality, requiring careful design to meet both user expectations and system limitations.
P3: Depth of analysis required	—
P4: Familiarity of issues	—
P5: Extent of applicable codes	—
P6: Extent of stakeholder involvement and conflicting requirements	The main stakeholders of the Nebula Navigator project are students and astronomy enthusiasts who want an easy-to-use system to manage star records. The developers need to ensure that the system is secure, functional, and lightweight. Conflicts can arise between usability, security, and features, and the system must be designed to balance these requirements effectively.
P7: Interdependence	—

1.4 Design Goals/Objectives

The main objective of the Nebula Navigator project is to create a simple and interactive system for managing information about stars. The system is designed to allow users to log in securely and perform operations such as adding new stars, viewing existing records, searching for specific stars, editing information, and deleting records. Another goal is to provide features for checking and comparing star distances in a clear and organized way. The project also aims to demonstrate how Bash scripting can be used for

file handling, text processing, and creating menu-driven interfaces. Overall, the design focuses on usability, functionality, and educational value for students and astronomy enthusiasts.

1.5 Application

The Nebula Navigator project can be used by students, hobbyists, and astronomy enthusiasts to manage and explore information about stars in a structured way. It allows users to store, view, search, edit, and delete star records, making it easier to organize astronomical data. The system can also be used as an educational tool to learn about different types of stars, their distances, and characteristics. Additionally, it demonstrates practical applications of Bash scripting, file handling, and menu-driven program design, which can be useful for beginners in programming and data management.

1.6 Legal Document Signing

The Nebula Navigator project, being a student academic project, requires a formal acknowledgment and signing of legal documents by the developers and the supervising faculty. This ensures that the work submitted is original, authentic, and does not infringe on intellectual property rights. By signing the document, the students confirm that all code, design, and content in the project are their own work, and they agree to abide by the university's academic integrity and submission guidelines. It also serves as a formal record for the approval and acceptance of the project by the institution.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The Nebula Navigator project is a Bash-based Galaxy Star Record System designed to manage information about stars in a simple and organized way. The system allows users to log in securely and perform tasks such as adding new stars, viewing records, searching for stars, editing details, deleting records, and checking star distances.

It uses text files to store data and relies on Bash commands like `grep`, `awk`, and `sort` for data management. The project features a menu-driven interface with colored text and ASCII formatting to make it interactive and easy to use. The system demonstrates how Bash scripting can be applied for file handling, CRUD operations, and terminal-based program design.

2.2 Project Details

The Nebula Navigator project is a Bash-based Galaxy Star Record System that allows users to manage information about stars. Users can log in securely and perform tasks such as adding, viewing, searching, editing, and deleting star records, as well as checking star distances.

Each star record contains an ID, Name, Type, Distance, and Description, and all data is stored in text files. The system uses Bash commands like `grep`, `awk`, and `sort` for data management. A menu-driven interface with colored text and ASCII formatting makes the system interactive and easy to use.

This project demonstrates how Bash scripting can be used for file handling, data processing, and terminal-based program design.

2.3 Implementation

2.3.1 Tools and libraries

The Nebula Navigator project is implemented using Bash scripting, which provides a lightweight and efficient way to manage text-based data. The main tools and libraries used include:

- **Bash Shell:** For writing the script, handling user input, and implementing the menu-driven interface.
- **Text Files (data.txt and users.txt):** Used for storing star records and user credentials.
- **Bash Commands:** Commands such as `grep`, `awk`, `sed`, `sort`, and `wc` are used for searching, editing, sorting, and managing data.
- **Terminal Utilities:** ANSI escape codes are used to add colors and formatting to the interface for better user experience.

No external libraries are required, making the project fully portable on any system with a Bash shell. This setup demonstrates how core Bash tools can be effectively used to implement a structured record management system.

2.4 Algorithms

The Nebula Navigator project is designed as a menu-driven system that performs several operations on star records. The key features and algorithms include:

1. **Login Authentication:** The system checks the username and password entered by the user against the `users.txt` file. Only valid users with correct credentials can access the main menu.
2. **Add New Star:** The system generates a unique ID for each new star and stores its Name, Type, Distance, and Description in the `data.txt` file.
3. **View All Stars:** Uses the `sort` and `column` commands to display all stored stars in a neat tabular format.
4. **Search Star by Name:** The system uses `awk` to find and display star records that match the exact name input by the user.
5. **Edit Star Information:** Reads the star record by ID, allows the user to update details, and rewrites the file with the updated information.
6. **Delete Star:** Confirms deletion by asking for the user's password and then removes the star record from `data.txt` using `grep -v`.
7. **Distance Check:** Displays the distance of a specific star and sorts all stars by distance for easy comparison.

These features are implemented using loops, conditional statements, functions, and text processing commands in Bash. The system emphasizes simplicity, interactivity, and structured data management.

2.4.1 Code:

```

1  #!/bin/bash
2
3
4  USERS_FILE="./data/users.txt"
5  DATA_FILE="./data/data.txt"
6
7  CURRENT_USER=""
8
9  RED="\e[31m"
10 GREEN="\e[32m"
11 YELLOW="\e[33m"
12 BLUE="\e[34m"
13 MAGENTA="\e[35m"
14 CYAN="\e[36m"
15 WHITE="\e[37m"
16 RESET="\e[0m"
17
18 TAB="$(printf '\t\t\t\t\t\t')"
19 LINE="$(printf '%.0s' {1..40})"
20
21 TERM_WIDTH=$(tput cols)
22 SCREEN_WIDTH=$(( term_width * 70 / 100 ))
23
24 header() {
25     clear
26     echo -e ""
27     echo -e "\t\t\t\t\t\t${CYAN}*****"
28     echo -e "\t\t\t\t\t\t${CYAN}**\t\t\t\t\t\tNebula Navigator\t\t\t\t\t\t${RESET}"
29     echo -e "\t\t\t\t\t\t${CYAN}**\t\t\t\t\t\tA GALAXY STAR RECORD SYSTEM\t\t\t\t\t\t*"
30     echo -e "\t\t\t\t\t\t${CYAN}**\t\t\t\t\t\tDeveloped by: Md.Shajalal and Hridoy Mia\t\t\t\t\t\t*"
31     echo -e "\t\t\t\t\t\t${CYAN}**\t\t\t\t\t\tGreen University Bangladesh\t\t\t\t\t\t*"
32     echo -e "\t\t\t\t\t\t${CYAN}*****"
33 }
34
35 login() {
36     attempts=0
37     while [ $attempts -lt 3 ]; do
38         header
39         echo -e "${TAB}${YELLOW}LOGIN${RESET}"
40         read -p "${TAB}Username: " username
41         read -s -p "${TAB>Password: " password
42         echo
43         if grep -q "^$username|$password$" "$USERS_FILE"; then
44             echo -e "${TAB}${GREEN>Login successful!${RESET}"
45             CURRENT_USER="$username"
46             sleep 1
47             mainMenu
48         else

```

```

49         echo -e "${TAB}${RED}Invalid credentials!${RESET}"
50         ((attempts++))
51         sleep 1
52     fi
53 done
54 echo -e "${TAB}${RED}Too many failed attempts. Exiting.${RESET}"
55 exit 1
56
57 }
58
59 menu() {
60     header
61     echo -e "${TAB}${YELLOW}[1]${RESET} Add New Star"
62     echo -e "${TAB}${CYAN}[2]${RESET} View All Stars"
63     echo -e "${TAB}${MAGENTA}[3]${RESET} Search Star by Name"
64     echo -e "${TAB}${BLUE}[4]${RESET} Edit Star Info"
65     echo -e "${TAB}${RED}[5]${RESET} Delete Star"
66     echo -e "${TAB}${GREEN}[6]${RESET} Show Distance of a Star"
67     echo -e "${TAB}${MAGENTA}[7]${RESET} Logout"
68     echo -e "${TAB}${WHITE}[8]${RESET} Exit"
69     echo "${TAB}${LINE}"
70     echo -n "${TAB}Choose an option [1-8]: "
71 }
72
73 add_star() {
74     header
75     echo -e "${TAB}\t\t\t${YELLOW}ADD NEW STAR${RESET}"
76     id=$((wc -l < "$DATA_FILE" + 1))
77     printf -v "${TAB} id "%03d" $id
78     read -p "${TAB}Star Name: " name
79     read -p "${TAB}Star Type: " type
80     read -p "${TAB}Distance (in ly): " distance
81     read -p "${TAB}Short Description: " desc
82     echo "$id|$name|$type|$distance|$desc" >> "$DATA_FILE"
83     echo -e "${TAB}${GREEN}Star added successfully!${RESET}"
84     read -p "${TAB}Press Enter to return to menu..."
85 }
86
87 view_stars() {
88     header
89
90     echo -e "${TAB}ALL STARS RECORDED"
91     echo -e "${LINE}${LINE}${LINE}"
92     echo -e "${WHITE}ID${RESET} \t\t${WHITE}NAME${RESET} \t\t\t ${WHITE}TYPE${RESET} \t\t\t "
93     echo -e "${LINE}${LINE}${LINE}"
94
95     if [ -s "$DATA_FILE" ]; then
96         sort -t '|' -k1n "$DATA_FILE" | column -s '|' -t
97     else
98         echo -e "${RED}No records found.${RESET}"
99     fi

```

```

100
101     echo -e "${LINE}${LINE}${LINE}"
102     read -p "Press Enter to return to menu..."
103 }
104
105 search_star() {
106     header
107     echo -e "${TAB}${YELLOW}SEARCH STAR${RESET}"
108     read -p " Enter star name (exact): " key
109     key=$(echo "$key" | xargs)
110
111     result=$(awk -F'|' -v k="$key" 'BEGIN{IGNORECASE=1} tolower($2) == tolower(k)' "$DATA_FILE")
112
113     if [ -n "$result" ]; then
114         echo -e " ${LINE}${LINE}${LINE}"
115         echo -e "\n ${GREEN}Matching Result:${RESET}\n"
116         echo -e " ${LINE}${LINE}${LINE}"
117
118         echo "$result" | while IFS='|' read -r id name type distance desc; do
119             echo " ID:          $id"
120             echo " Name:          $name"
121             echo " Type:          $type"
122             echo " Distance:      $distance ly"
123             echo " Description:   $desc"
124             echo -e "${LINE}${LINE}${LINE}"
125         done
126     else
127         echo -e " ${RED}Star not found!${RESET}"
128     fi
129
130     echo
131     read -p " Press Enter to return to menu..."
132 }
133
134 edit_star() {
135     header
136     echo "EDIT STAR INFO"
137     read -p "Enter Star ID to edit: " id
138     line=$(grep "^$id|" "$DATA_FILE")
139     if [ "$line" ]; then
140         echo "Old Data:"
141         echo "$line" | column -s '|' -t
142         grep -v "^$id|" "$DATA_FILE" > temp.txt
143         read -p "New Name: " name
144         read -p "New Type: " type
145         read -p "New Distance (ly): " distance
146         read -p "New Description: " desc
147         echo "$id|$name|$type|$distance|$desc" >> temp.txt
148         mv temp.txt "$DATA_FILE"
149         echo -e "${GREEN}Star updated successfully!${RESET}"
150     else

```

```

151     echo -e "${RED}Star ID not found!${RESET}"
152 fi
153 read -p "Press Enter to return to menu..."
154 }
155
156 delete_star() {
157     header
158     echo "DELETE STAR"
159     read -p "Enter Star ID to delete: " id
160
161     record=$(grep "^$id|" "$DATA_FILE")
162     if [ -n "$record" ]; then
163         echo "Star found:"
164         echo -e "${LINE}${LINE}${LINE}"
165         echo "$record"
166         echo -e "${LINE}${LINE}${LINE}"
167
168         read -p "Are you sure you want to delete this star? (y/n): " confirm
169         if [[ "$confirm" =~ ^[Yy]$ ]]; then
170             echo -e -n "${RED}Enter your password to confirm: ${RED}"
171             read password
172             echo
173
174             if grep -q "^$CURRENT_USER|$password$" "$USERS_FILE"; then
175                 grep -v "^$id|" "$DATA_FILE" > temp.txt && mv temp.txt "$DATA_FILE"
176                 echo -e "${GREEN}Star deleted successfully!${RESET}"
177             else
178                 echo -e "${RED}Incorrect password! Deletion aborted.${RESET}"
179             fi
180         else
181             echo "Deletion cancelled."
182         fi
183     else
184         echo -e "${RED}Star ID not found!${RESET}"
185     fi
186
187     read -p "Press Enter to return to menu..."
188 }
189
190 distance_star() {
191     header
192     echo "DISTANCE CHECK"
193     read -p "Enter Star Name: " name
194
195     match=$(grep -i "$name" "$DATA_FILE")
196
197     if [ "$match" ]; then
198         echo "Matching star(s):"
199         echo "$match" | awk -F "|" '{print "ID: "$1" | Name: "$2" | Distance: "$4" " "ly"}'
200
201         echo

```

```

202     echo "Sorting all stars by distance..."
203     echo -e "${LINE}${LINE}${LINE}"
204
205     sort -t "|" -k4 -n "$DATA_FILE" | awk -F "|" '{print "ID: "$1" | Name: "$2" | Dis
206 else
207     echo -e "${RED}Star not found!${RESET}"
208 fi
209
210     read -p "Press Enter to return to menu..."
211 }
212
213 signUpMenu() {
214     header
215     echo "${TAB}Login Menu"
216     echo "${TAB}${LINE}"
217     echo -e "${TAB}${WHITE}[1]${RESET} Login"
218     echo -e "${TAB}${WHITE}[2]${RESET} Exit"
219     echo "${TAB}${LINE}"
220     echo -n "${TAB}Choose an option [1-2]: "
221
222     read choice
223     case $choice in
224         1) login ;;
225         2) exit ;;
226         *) echo -e "${TAB}${RED}Invalid input${RESET}"; sleep 1 ;;
227     esac
228 }
229
230 mainMenu() {
231     while true; do
232         header
233         menu
234         read choice
235         case $choice in
236             1) add_star ;;
237             2) view_stars ;;
238             3) search_star ;;
239             4) edit_star ;;
240             5) delete_star ;;
241             6) distance_star ;;
242             7) signUpMenu ;;
243             8) echo -e "${TAB}${GREEN}Thanks for using Nebula Navigator${RESET}"; exit ;;
244             *) echo -e "${TAB}${RED}Invalid input${RESET}"; sleep 1 ;;
245         esac
246     done
247 }
248
249 function main() {
250     while true; do
251         signUpMenu
252     done

```

253
254
255
256
257
258

}

main

Chapter 3

Performance Evaluation

3.0.1 Simulation Environment / Simulation Procedure

The Nebula Navigator project runs in a Linux or Unix terminal using Bash scripting. All star records and user credentials are stored in text files. The user starts the program by running the script and logs in with a valid username and password. After logging in, the main menu allows the user to add, view, search, edit, or delete star records, and check star distances. All actions are performed using Bash commands, demonstrating an interactive terminal-based system.

3.0.2 Development Environment

The Nebula Navigator project was developed using Bash scripting in a Linux terminal environment. A text editor, such as VS Code or Nano, was used to write and edit the script. All data is stored in plain text files (data.txt and users.txt) for simplicity and easy access. The system uses standard Bash commands like grep, awk, sed, and sort for processing and managing data. No external libraries or frameworks are required, making the project lightweight and portable.

3.0.3 Results Analysis/Testing

```
Lab/Project/Nebula-Navigator/src

*****
**                               **
**      Nebula Navigator         **
**      A GALAXY STAR RECORD SYSTEM      **
**      Developed by: Md.Shajalal        **
**      Green University Bangladesh      **
**                               **
*****

Login Menu
=====
[1] Login
[2] Exit
=====
Choose an option [1-2]: |
```

1.

2.

```
*****
**                               **
**           Nebula Navigator           **
**           A GALAXY STAR RECORD SYSTEM       **
**           Developed by: Md.Shajalal         **
**           Green University Bangladesh       **
**                               **
*****
LOGIN
Username: n
Password: |
```

3.

```
*****
**                               **
**           Nebula Navigator           **
**           A GALAXY STAR RECORD SYSTEM       **
**           Developed by: Md.Shajalal         **
**           Green University Bangladesh       **
**                               **
*****
[1] Add New Star
[2] View All Stars
[3] Search Star by Name
[4] Edit Star Info
[5] Delete Star
[6] Show Distance of a Star
[7] Logout
[8] Exit
=====
Choose an option [1-8]: |
```

4.

```
*****
**                               **
**           Nebula Navigator           **
**           A GALAXY STAR RECORD SYSTEM       **
**           Developed by: Md.Shajalal         **
**           Green University Bangladesh       **
**                               **
*****
                ADD NEW STAR
Star Name: Md.Shajalal
Star Type: 123
Distance (in ly): 34
Short Description: seif
Star added successfully!
Press Enter to return to menu...|
```



```

*****
**                                     Nebula Navigator                                     **
**                                     A GALAXY STAR RECORD SYSTEM                         **
**                                     Developed by: Md.Shajalal                         **
**                                     Green University Bangladesh                         **
*****
ALL STARS RECORDED
=====
ID | NAME | TYPE | DISTANCE(Light Years) | DESCRIPTION
=====
001 Proxima Centauri Red Dwarf 4.24 Closest star
002 Alpha Centauri A Main Sequence 4.37 Sun-like star
003 Alpha Centauri B Main Sequence 4.37 Companion star
004 Barnard's Star Red Dwarf 5.96 Nearby dwarf
005 Wolf 359 Red Dwarf 7.86 Very faint
006 Luyten 726-8 A Red Dwarf 8.73 Binary star
007 Luyten 726-8 B Red Dwarf 8.73 Companion star
008 Sirius A Main Sequence 8.6 Brightest star
009 Sirius B White Dwarf 8.6 Companion star
010 Ross 154 Red Dwarf 9.68 Nearby dwarf
011 Ross 248 Red Dwarf 10.3 Faint star
012 Epsilon Eridani K-type Main Sequence 10.5 Nearby system
013 Lacaille 9352 Red Dwarf 10.7 Nearby dwarf
014 Ross 128 Red Dwarf 11 Nearby dwarf
015 EZ Aquarii A Red Dwarf 11.3 Triple system
016 EZ Aquarii B Red Dwarf 11.3 Companion star
017 EZ Aquarii C Red Dwarf 11.3 Companion star
018 Procyon A Main Sequence 11.4 Bright star
019 Procyon B White Dwarf 11.4 Companion star
020 61 Cygni A K-type Main Sequence 11.4 Binary star
021 nazmul big 12 lightweight
022 Hh sas 12 mjg
023 sojib cse 1 q
024 Md.Shajalal cse 1 ire
025 Md.Shajalal 123 34 seif
=====
Press Enter to return to menu...|

```

5.

Chapter 4

Conclusion

4.1 Discussion

The Nebula Navigator project successfully demonstrates a terminal-based star record management system using Bash scripting. It allows users to securely log in and perform operations such as adding, viewing, searching, editing, and deleting star records. The system is lightweight, interactive, and easy to use, making it suitable for students and astronomy enthusiasts. Using Bash commands for data processing shows the practical application of scripting for file handling and menu-driven programs. Overall, the project highlights how simple tools can be used to build functional and educational software, and it can be expanded in the future with more advanced features.

4.2 Limitations

The Nebula Navigator project is limited to a terminal-based interface, which may not be as visually appealing as a graphical application. It relies on text files for data storage, which can become slow if the number of star records grows very large. The system requires users to have basic knowledge of terminal operations to interact effectively. Security is basic and only uses username and password authentication, without advanced protection features. Additionally, the project does not include integration with external astronomical databases, limiting the amount of available star data.