Project Based Evaluation

Project Report

Semester-IV (Batch-2023)

Placemate

Supervised By:

Dr. Harmanjeet Singh

Submitted By:

Ronak Oberoi,2310991114

Yudhish Garg, 2310991168

Palak Garg,2310991341

Sanpreet,2310992282

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Abstract

Placemate is a terminal-based *Student Placement Management System* developed in Bash, designed to provide an intuitive, interactive, and lightweight interface for managing student placement records in educational institutions. It leverages the dialog utility to render a user-friendly text-based GUI, allowing seamless interaction through navigable menus and form elements directly in the Linux terminal environment.

The core of the system lies in its script student_mgmt.sh, which facilitates essential operations such as adding, modifying, viewing, and deleting student records. Additional functionalities include assigning companies to students, searching by roll number, and filtering records by branch or CGPA. This makes the system comprehensive in addressing placement management needs without the overhead of full-scale database systems or web servers.

All student records are stored in a structured students.txt file using a colon-separated format for easy parsing and editing. User interactions and changes are systematically logged in student_mgmt.log, ensuring traceability and accountability. The visual interface, as depicted in the attached screenshot, highlights a clear and organized menu system, simplifying navigation and minimizing the learning curve for non-technical users.

The project directory includes a README.md for quick setup and usage instructions, a detailed REPORT.pdf documenting the system's design and implementation, and testing scripts like test_dialog.sh for verifying UI components. Overall, **Placemate** stands as a practical and educational tool, demonstrating the capabilities of shell scripting in building functional administrative systems while promoting open-source principles and minimal resource dependency.

Table of Contents

1. Introduction

1.1 Background

In academic institutions, the placement process plays a crucial role in shaping the future of students. Managing student placement data such as academic performance, company allocation, and eligibility criteria often involves dealing with large volumes of information. Traditionally, this data is handled using spreadsheets or manual registers, which are prone to errors, difficult to maintain, and lack efficiency. As digital transformation continues to impact every sector, including education, there is a growing need for simple, robust, and accessible systems to handle placement-related operations. However, many existing software solutions are either too complex for small institutions or demand high system resources. To address this gap, a lightweight and terminal-based student placement management system named Placemate has been developed using Bash scripting.

1.2 Objective

The core objective of the Placemate project is to design and implement a command-line based student placement management system that is intuitive, resource-friendly, and suitable for use in Linux environments. Specific goals include:

- Allowing users to add, view, modify, and delete student records.
- Providing features to assign companies to eligible students.
- Enabling search operations based on student roll number.
- Offering filtering options by branch and CGPA to streamline record access.
- Ensuring that all data is stored and managed through a simple text file (students.txt), eliminating the need for external databases.

By achieving these objectives, the system provides a functional tool that can help automate and simplify the placement process for institutions or training centers.

1.3 Significance

The significance of the *Placemate* project lies in its pragmatic, educational, and technical value. Designed to run entirely within a Linux terminal environment, *Placemate* exemplifies how powerful and functional applications can be built using only native tools like Bash scripting, dialog for UI, and simple text files for data storage. This approach minimizes dependencies, allowing the system to run seamlessly even on older or resource-constrained hardware, making it ideal for use in academic institutions, training labs, or systems administration environments.

In a world increasingly dominated by complex web or mobile applications, *Placemate* reaffirms the relevance and efficiency of command-line tools. It highlights the utility of shell scripting in solving real-world problems without the overhead of setting up servers, databases, or frameworks. For many students and beginners in system-level programming or DevOps, this project offers a clear and hands-on introduction to file manipulation, user input handling, condition-based logic, and basic CRUD (Create, Read, Update, Delete) operations—all fundamental concepts in software development.

From an educational standpoint, *Placemate* serves as a practical case study in the use of dialog-based user interfaces within terminal environments. This not only helps learners develop logical problem-solving skills but also encourages best practices in shell scripting, such as modularity, input validation, error handling, and user feedback. Through features like logging operations, filtering data by criteria, and updating records interactively, students are exposed to essential scripting patterns that are directly applicable in real-world IT workflows.

Moreover, *Placemate* encourages system-level thinking—how scripts interact with the operating system, how processes are managed, and how user actions can be captured and logged. These insights are especially valuable for aspiring system administrators, DevOps engineers, and backend developers who often need to automate administrative tasks or build lightweight internal tools.

In summary, *Placemate* is not just a project for managing placement data; it is a thoughtfully designed tool that emphasizes simplicity, efficiency, and educational impact. It provides a stepping stone for deeper exploration into the world of Linux scripting and underscores how even modest tools can deliver powerful, user-friendly solutions.

2. Problem Definition and Requirements

2.1 Problem Statement

In most academic institutions, student placement is a critical process that determines the employment opportunities students can access after graduation. However, managing the data associated with student records—such as names, roll numbers, branches, CGPA scores, and assigned companies—often involves time-consuming and error-prone manual processes. Many institutions still rely on spreadsheets or handwritten notes, which makes updating, searching, and filtering student information tedious and inefficient.

While robust enterprise-level solutions do exist, they often require significant resources, training, and infrastructure that smaller institutions may not have. Furthermore, GUI-based solutions may not be suitable for environments where terminal usage is preferred or where systems have limited graphical capabilities.

Placemate aims to solve this issue by offering a terminal-based, dialog-driven student placement system that performs essential operations like adding, modifying, deleting, viewing, and filtering student data. It provides a lightweight and easily deployable solution using Bash scripting and text files for data storage.

2.2 Software Requirements

The software component of Placemate is designed to work seamlessly in a Linux-based environment. It is developed using shell scripting, particularly Bash, and utilizes tools available in most Linux distributions by default.

Software requirements include:

1. **Operating System**: Any Linux distribution (Ubuntu, Fedora, etc.)
2. **Shell**: Bash (Bourne Again SHell)
3. **Utilities**:
    a `dialog`: Used to create text-based graphical interfaces
    b `awk`, `grep`, `sed`: Used for text parsing and filtering
    c `chmod`, `touch`, `cat`: For file handling and permission control
4. **Text Editor**: Any basic editor like `vim`, `nano`, or `gedit` (for manual editing if needed)
5. **Terminal Emulator**: GNOME Terminal, Konsole, or any default Linux terminal

2.3 Hardware Requirements

Placemate is a lightweight system that has minimal hardware requirements. It can run on most machines that support a basic Linux OS installation.

Minimum hardware requirements:

- **Processor**: 1 GHz or faster (Intel/AMD)
- **RAM**: 512 MB or more
- **Storage**: At least 10 MB of free disk space (for scripts, logs, and student records)
- **Display**: Standard resolution compatible with terminal text UI
- **Input Devices**: Keyboard (for terminal interaction)

2.4 Data Set

The data set in Placemate is managed through a plain text file named students.txt. Each student record typically includes the following fields:

- Roll Number
- Name
- Branch
- CGPA
- Assigned Company (if any)

These records are stored in a structured format, usually delimited by a specific character (e.g., comma or colon) for easier parsing. The file serves as both the input and persistent storage for the application.

3. Proposed Design / Methodology

3.1 Overview Of Methodology

The Placemate project adopts a modular and menu-driven design, making it both user-friendly and easy to extend. The methodology behind its development centers around simplicity, portability, and functionality in a Linux environment using shell scripting. The program flow is defined to be intuitive so users can interact with it efficiently even with minimal technical training.

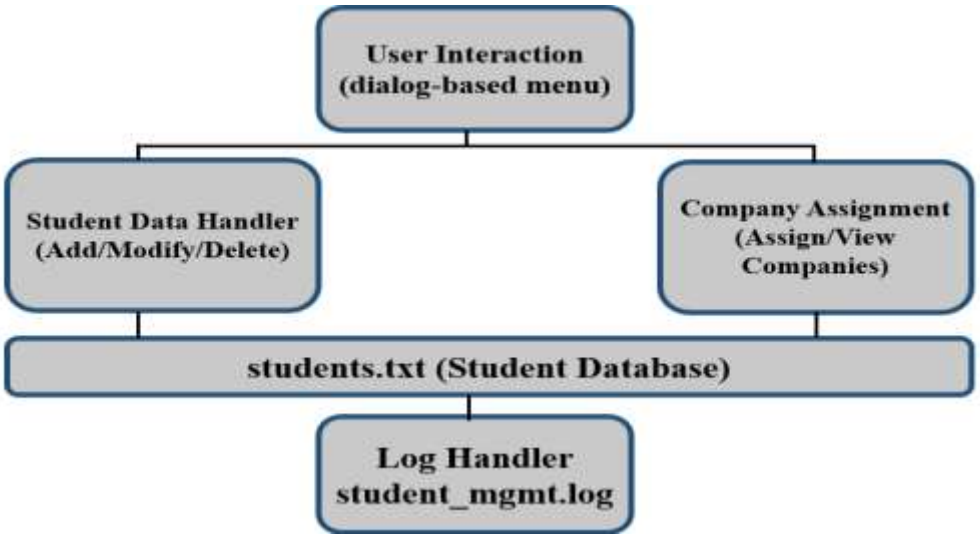
The overall design can be divided into the following core modules:

- User Interface Design (Dialog Boxes)
- File-based Data Management
- Menu-driven Execution Flow
- Logging and Error Handling

Each component of the system has been designed to work seamlessly in the terminal, leveraging dialog for the interface and plain text files for storage and processing.

3.2 Schematic Diagram / Architecture

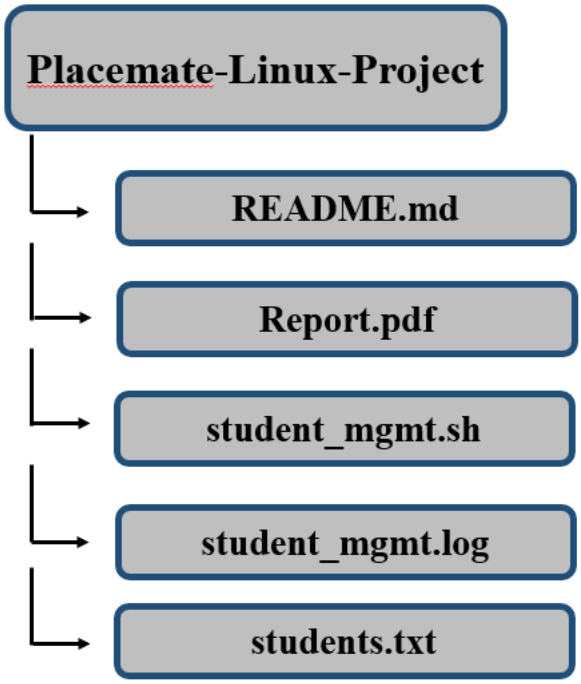
Here is a high-level schematic representation of the system architecture:



3.2.1 Schematic Diagram of the Placemate

3.3 File Structure

Below is the actual file structure of the project:



3.3.1 File Structure

Explanation of Key Files:

- **student_mgmt.sh**: This is the core script that presents the user with a menu using the dialog utility. It contains all the functions for adding, deleting, modifying, searching, and filtering student records.
- **students.txt**: This is the central data file. Each line contains a student's record with fields separated by delimiters. It is designed to be human-readable and easily parsed using shell utilities like awk, cut, or IFS.
- **student_mgmt.log**: Every operation (add, delete, modify, etc.) is logged here along with a timestamp, ensuring accountability and providing a trail of changes for audit or debugging purposes.

3.4 Menu-Driven Approach

Placemate uses a terminal-based GUI created using dialog, which allows users to interact with menus, message boxes, and input fields directly in the terminal. Upon running the student_mgmt.sh script, users are greeted with a main menu displaying various operations such as:

1. Add Student Record
2. Delete Student Record
3. Modify Student Record
4. View All Student Records
5. Assign Company to Student
6. Search Student by Roll No
7. Filter By Branch
8. Filter By CGPA

Each menu item triggers a specific function in the script, which performs the necessary file operations and provides appropriate feedback to the user.

3.5 Module Breakdown

a. Add/Modify/Delete Student
These functions allow users to enter or update student details via input boxes. The script validates each input (e.g., checking for duplicates or invalid CGPA) before writing to the file.

b. View All Records
Reads the students.txt file and presents data in a readable format using dialog's textbox or message box.

c. Assign Company
This feature appends or updates the company name for a given student by roll number. It uses a temporary file for safe modification and replaces the original file upon confirmation.

d. Search and Filter
Users can search for students using roll numbers or filter records by branch or CGPA. awk and grep are heavily used here to match and extract relevant data.

e. Logging
Each operation writes a line to student_mgmt.log using the format:
[Timestamp] [Operation] [Details]
This is useful for debugging, tracking usage, or recovering from failures.

3.6 Data Flow And Processing

Each function follows a simple yet consistent data flow:

1. **Input Capture**: via dialog input boxes
2. **Validation**: checks for completeness, format, and conflicts
3. **Processing**: read/write to students.txt
4. **Confirmation**: feedback dialog or error message
5. **Logging**: entry in student_mgmt.log

This ensures that data integrity is maintained and users are always informed of the action outcomes.

4. Result

The *Placemate* project successfully implements a command-line-based Student Placement Management System using native Linux tools. It allows users to manage student records efficiently through a dialog-based interface, supporting operations like adding, modifying, deleting, and filtering data. The system ensures data persistence using text files and performs reliably during execution, with no major errors or performance issues. Overall, the project meets its objectives by offering a lightweight, user-friendly, and educational tool suitable for placement management in Linux environments.

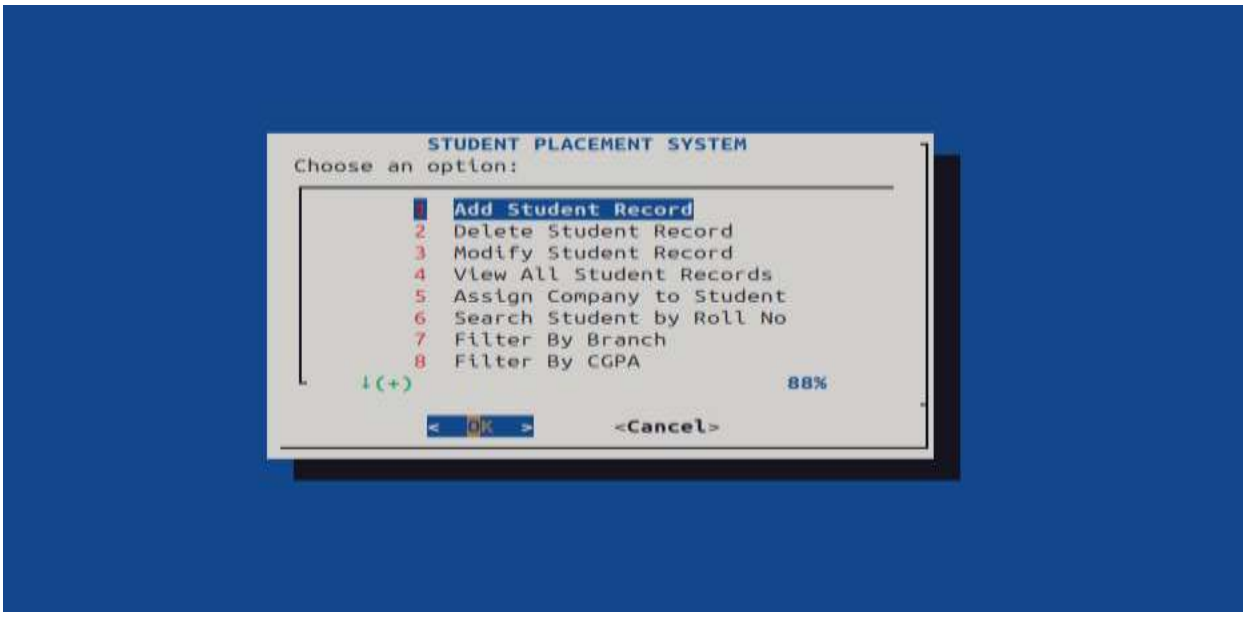


Fig. 4.1 Main Menu Screen

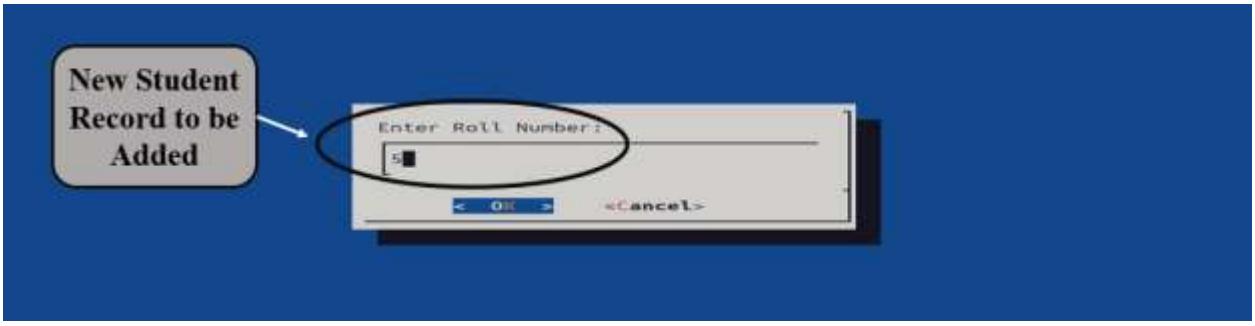


Fig. 4.2 Add New Student

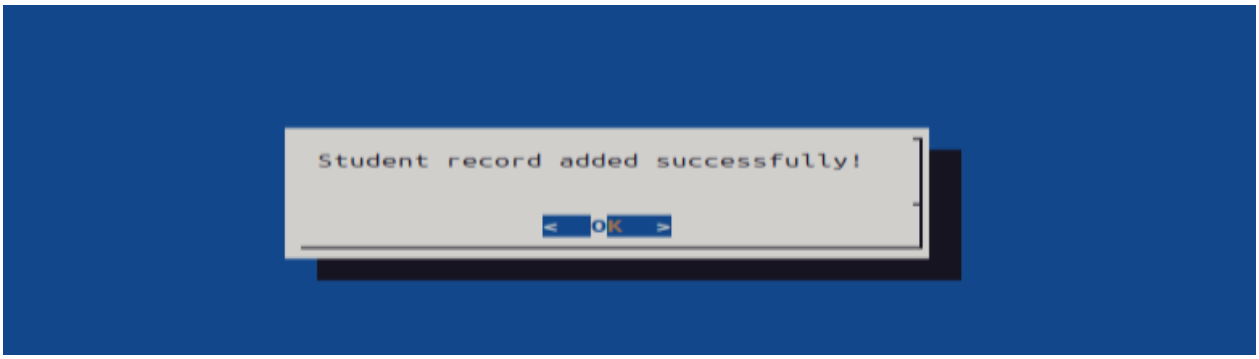


Fig. 4.3 New Student Record Added

Fig. 4.4 All Student Records

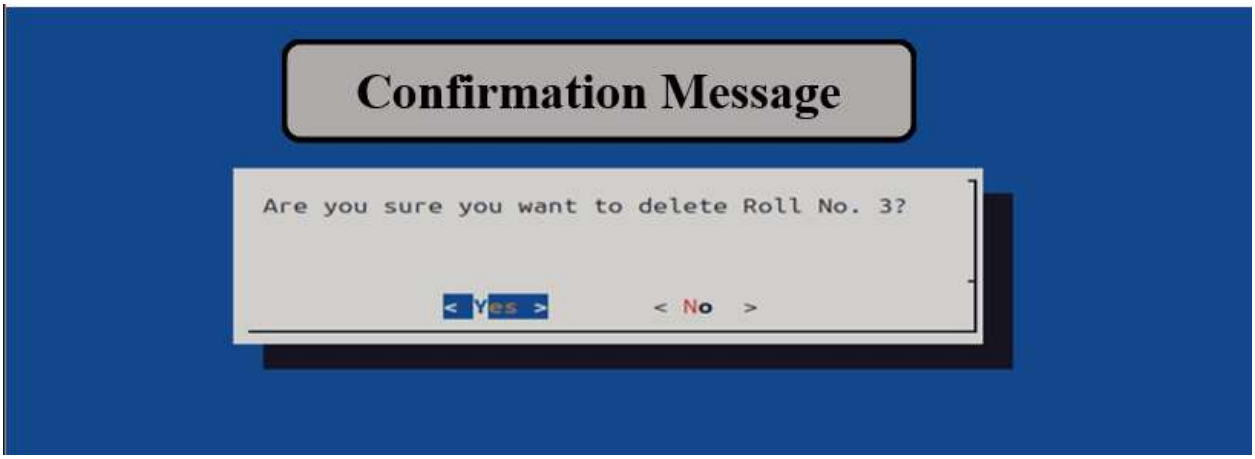


Fig. 4.5 Confirmation before Deletion

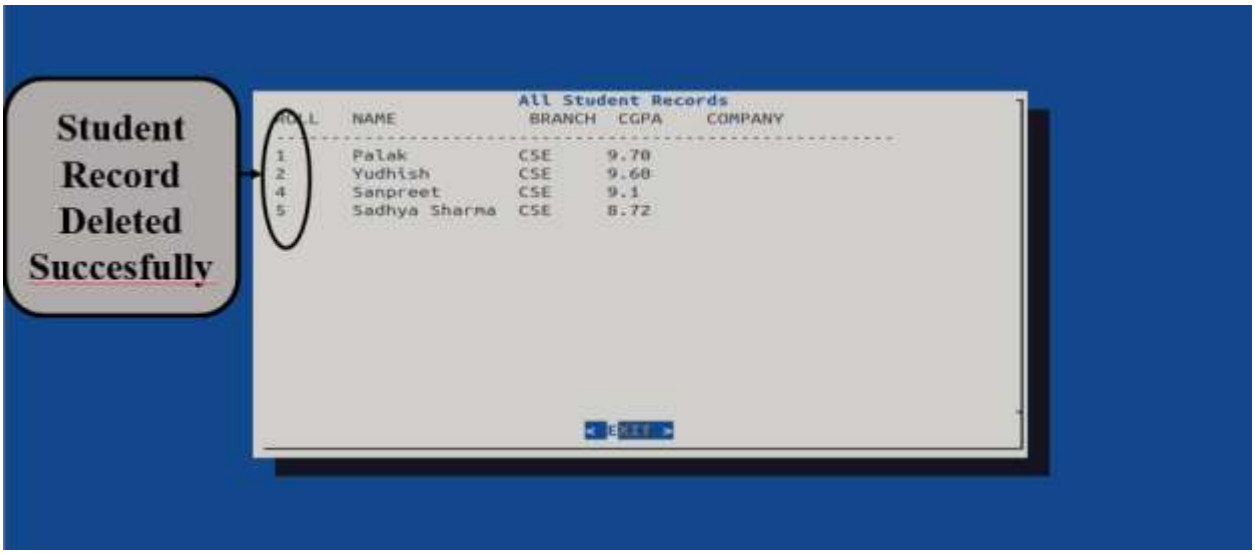


Fig. 4.6 Deletion success

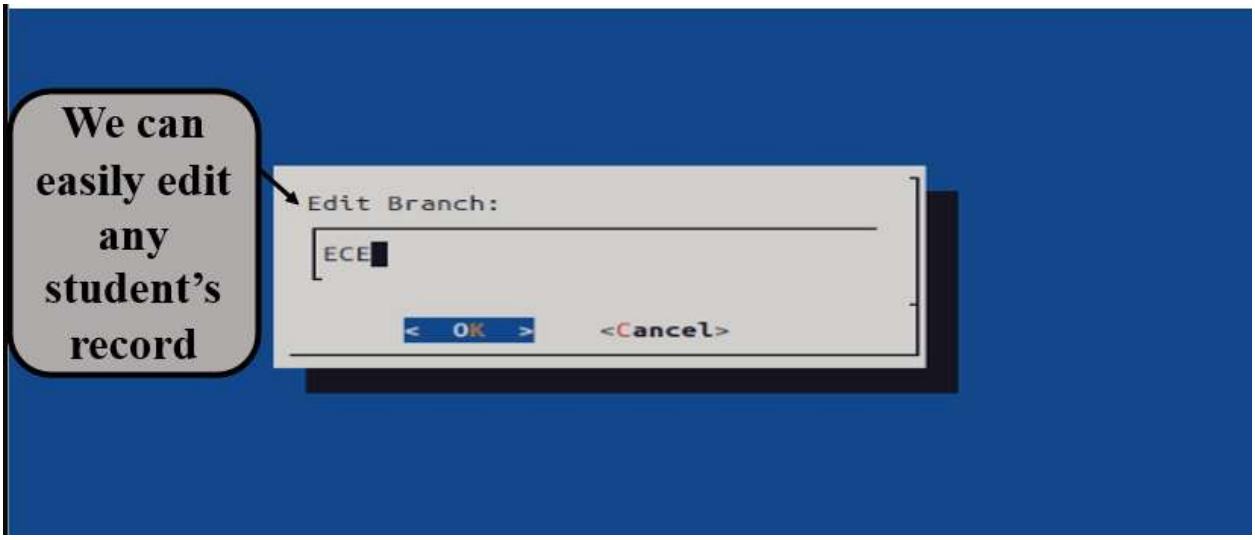


Fig. 4.7 Edition of Student Record

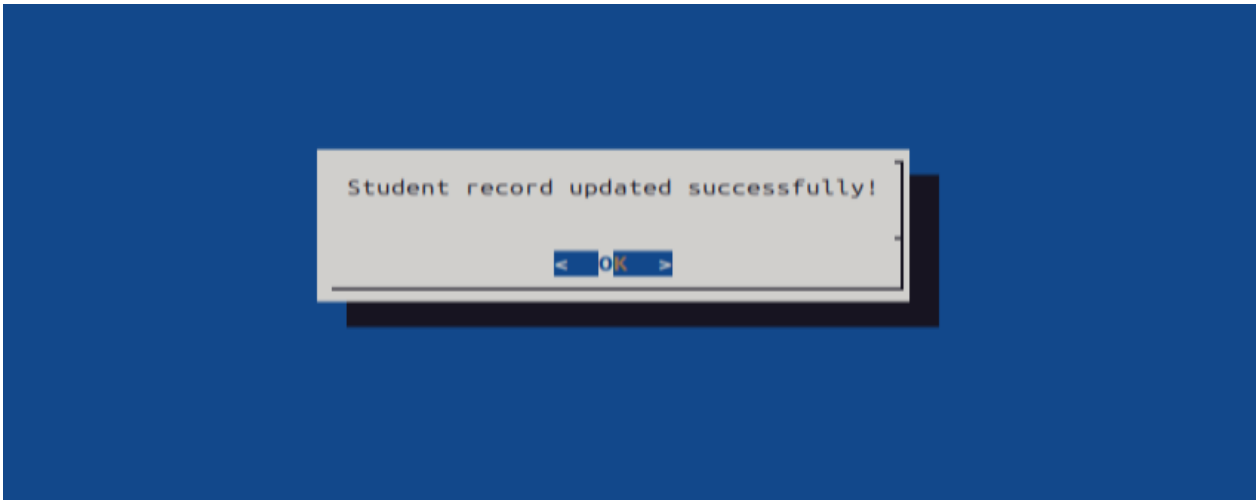


Fig. 4.8 Updation Success

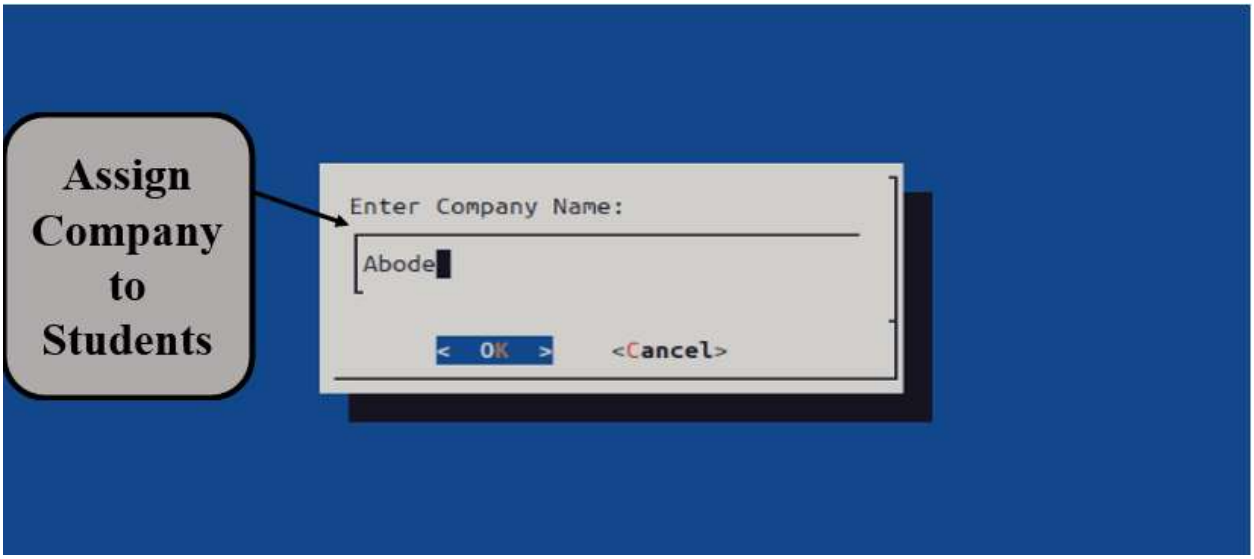


Fig. 4.9 Assigning Company to Student

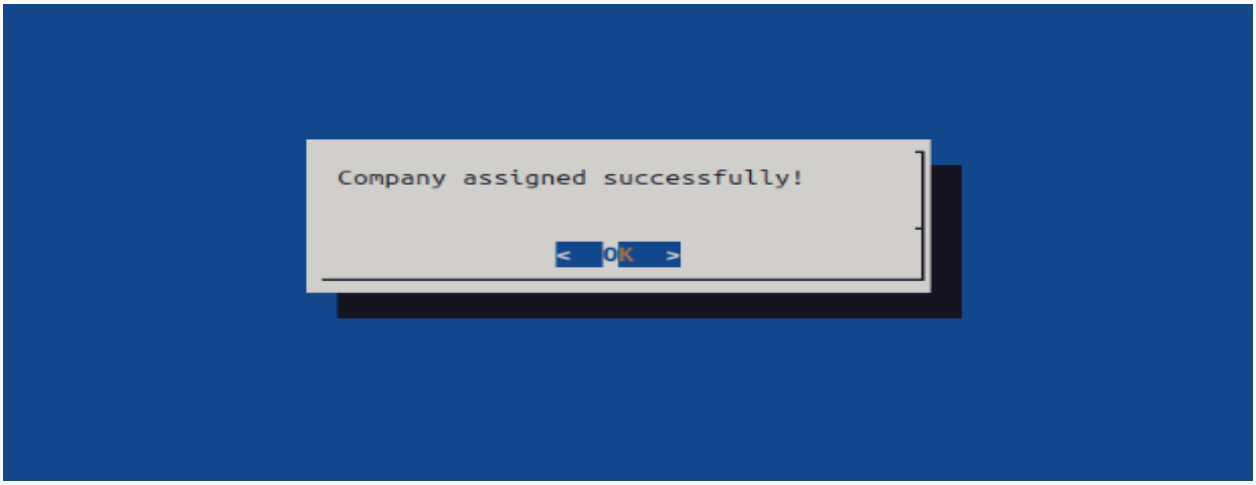


Fig. 4.10 Company Assigned Successfully

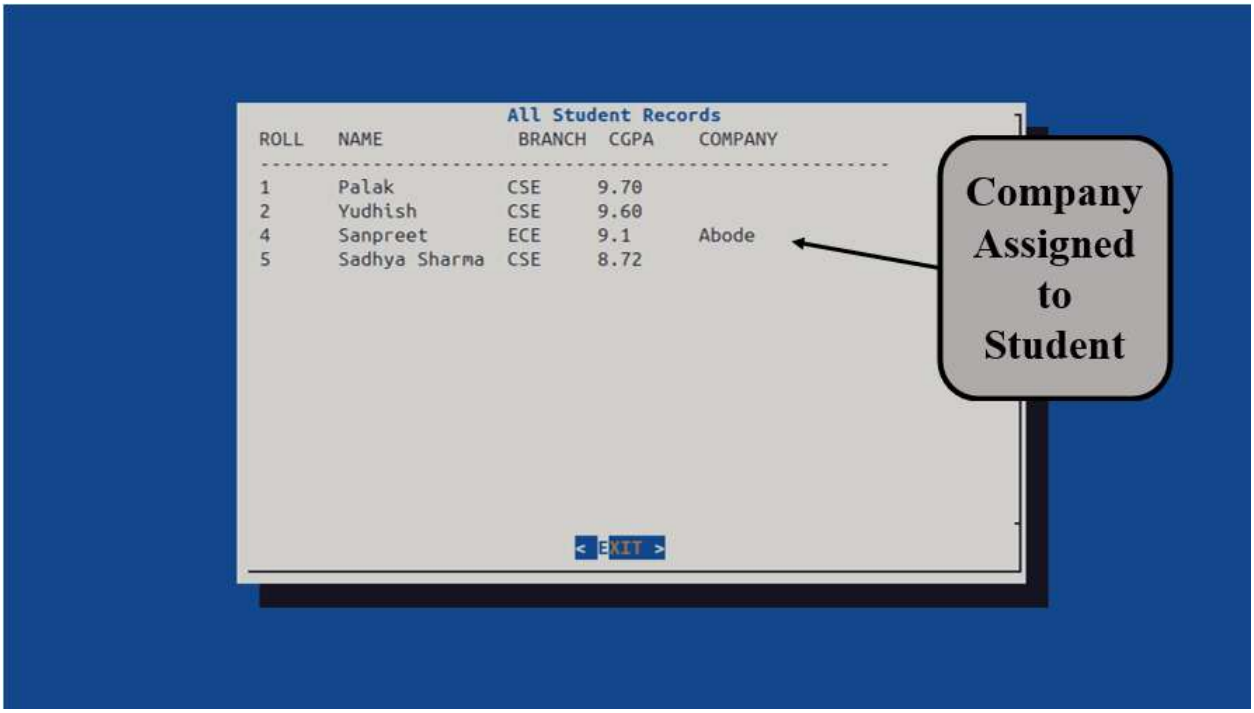


Fig. 4.11 Company Also Added in Records

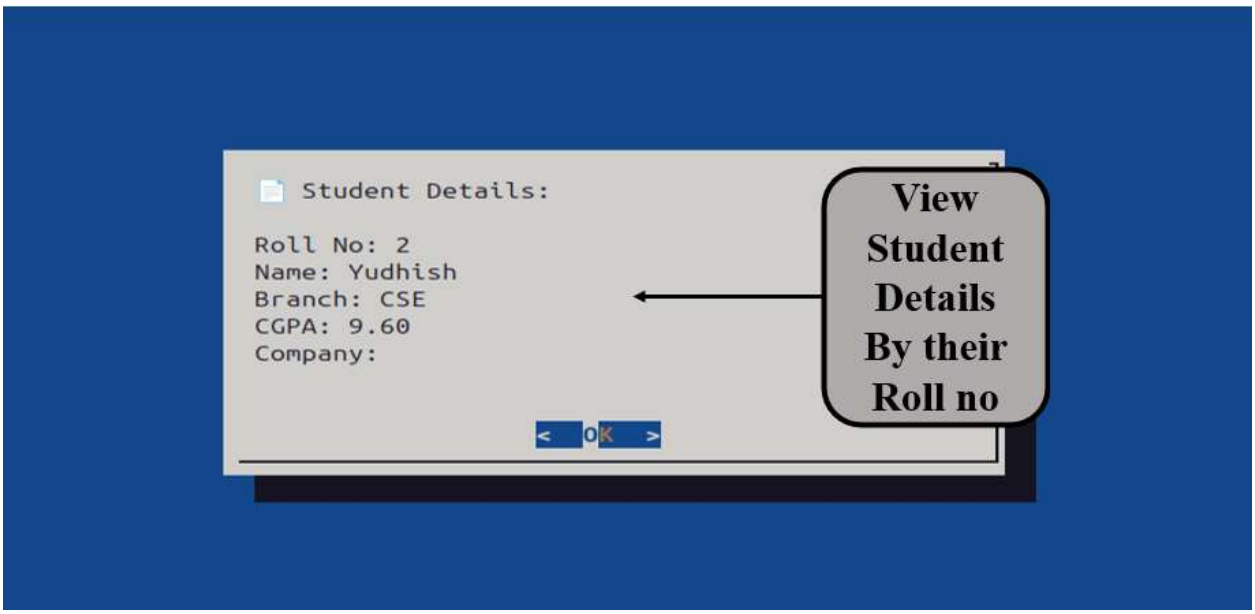


Fig. 4.12 View Details of Any Student Record

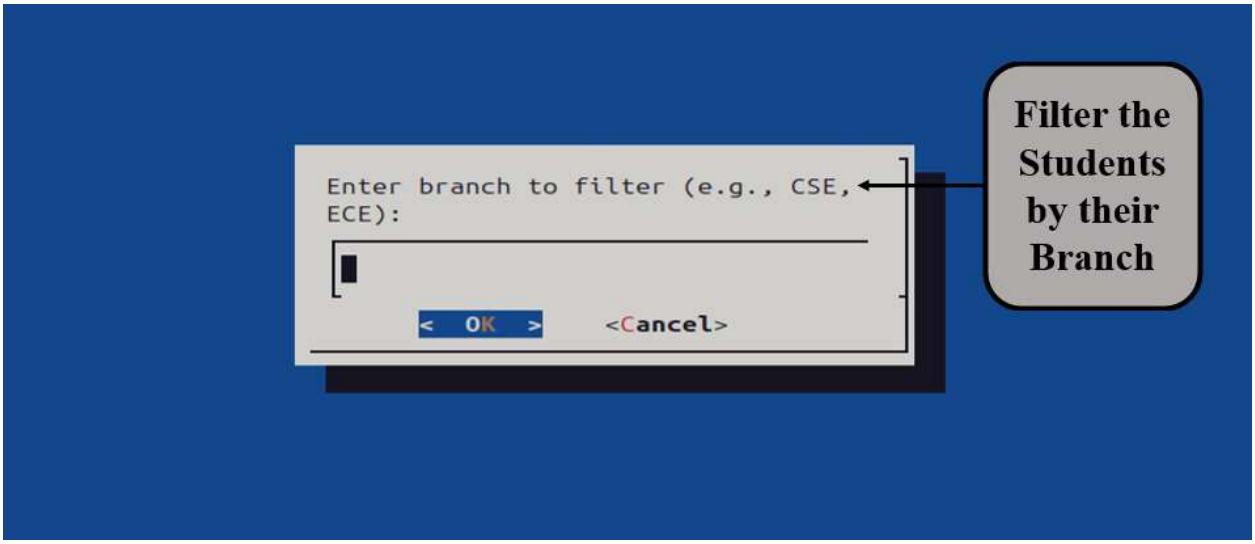


Fig. 4.13 Filter Student Records by Branch

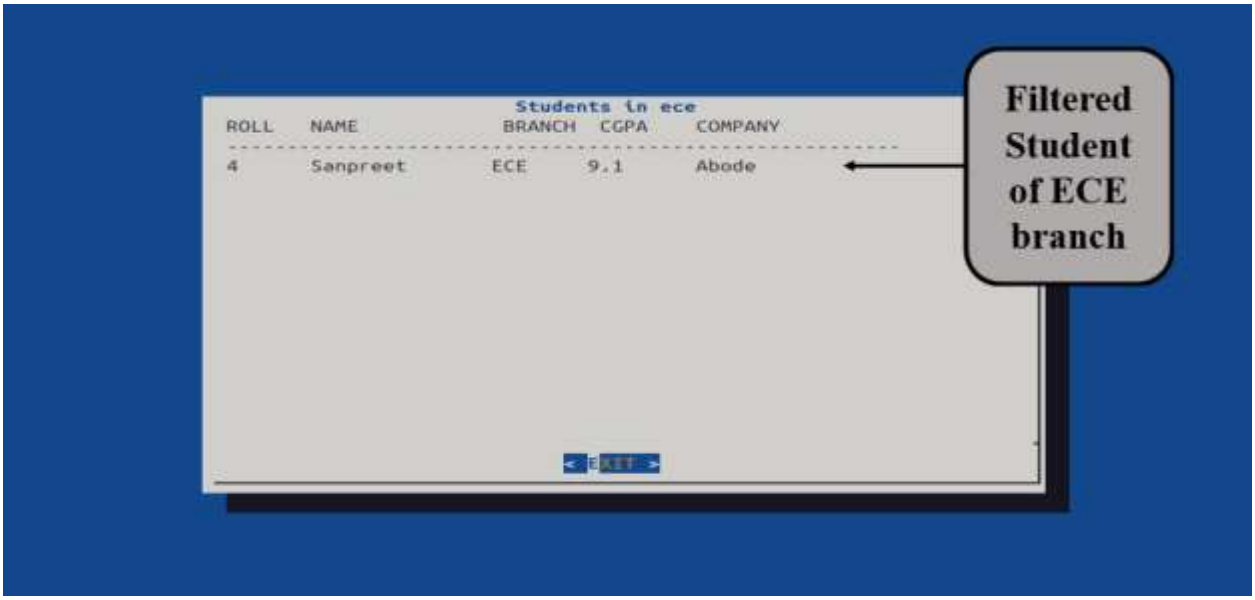


Fig. 4.14 Filtered Records