



আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম  
الجامعة الإسلامية العالمية شيتاغونغ  
International Islamic University Chittagong

Department of  
**Computer Science and Engineering**

**DS Final Project**

Course Code : CSE - 2322  
Course Title : Data Structure Lab  
Section : 3AM

Team Name : Team Console  
Name of Team Leader : Gazi Shaharabi Anwar Tuhin  
Team Leader ID : C231035  
Name of Team Member 1 : SM Muntasir Sarwar  
Team Leader ID : C231037  
Name of Team Member 2 : Adnan Mahmud Alvee  
Team Leader ID : C231022

Date of Submission : 25/06/2024  
Submitted To : Mohammed Shamsul Alam

Remark

## **Introduction**

The Employee Management System (EMS) is a console-based application designed to help users manage employee information efficiently. This system supports functionalities like adding, viewing, updating, deleting, and searching employee records. Additionally, it includes user authentication, allowing multiple users to maintain their own sets of employee data.

## **Features**

1. **User Authentication:**
  - Sign Up: Allows new users to create an account.
  - Log In: Allows existing users to access their accounts.
  - User credentials are stored in a `users.txt` file, and each user has a separate file to store employee data.
2. **Employee Data Management:**
  - Insert Data: Add new employee details.
  - Show Data: Display all employee records.
  - Search Data: Search for an employee by ID.
  - Update Data: Modify existing employee records.
  - Delete Data: Remove specific or all employee records.
3. **File Handling:**
  - Employee data is persisted in text files.
  - Utilizes file streams for reading and writing data.
4. **User Interface:**
  - Console-based interface with clear options.
  - Visual feedback through changing console colors and delay effects.

## **Implementation**

### **Functions Used - Short Description**

1. `insertEmp(const emp& employee) :`
  - Inserts a new employee record into the linked list.
2. `saveEmpDataToFile() :`
  - Saves all employee data from the linked list to the user's file.
3. `loadEmpDataFromFile() :`
  - Loads employee data from the user's file into the linked list.
4. `signUp() :`
  - Registers a new user by saving their credentials and creating a new file for their employee data.
5. `login() :`
  - Authenticates an existing user and loads their employee data.
6. `empdata() :`
  - Collects and inserts new employee data from the user.
7. `show() :`
  - Displays all employee records.
8. `Search() :`
  - Searches for an employee by ID and displays their information.

9. **update()** :
  - Updates an existing employee's information based on their ID.
10. **del()** :
  - Deletes specific or all employee records.
11. **feature()** :
  - Displays the list of features with a delay effect.
12. **sleepMilliseconds(int milliseconds)** :
  - Introduces a delay in the program execution.

### **Future Plan/Possible Extension**

1. **Enhanced Error Handling:**
  - Implement comprehensive error handling to cover more edge cases and provide better user feedback.
2. **Input Validation:**
  - Ensure that inputs like employee IDs are unique and follow a specific format.
3. **Graphical User Interface (GUI):**
  - The transition from a console-based interface to a GUI for a better user experience.
4. **Database Integration:**
  - Integrate with a database system like MySQL or SQLite for more efficient data management and retrieval.
5. **Concurrent Access Handling:**
  - Allow multiple users to access and modify data simultaneously with proper synchronization.
6. **Reporting and Analytics:**
  - Add features to generate reports and perform analytics on employee data.
7. **Data Encryption:**
  - Encrypt user credentials and employee data files to enhance security.

### **Screen Shorts :**

#### **Welcome Page :**

```
----- Employee Management System -----  
  
Press 1 to Sign Up  
Press 2 to Log In  
  
Enter your choice:
```

### Menu Page :

```
Press 1 to enter data
Press 2 to show data
Press 3 to search data
Press 4 to update data
Press 5 to delete data
Press 6 to show feature
Press 7 to logout
```

Enter your choice:

### Enter data page :

How many employees data do you want to enter??

2

Enter data of employee 1

Enter employee name: Mun

Enter id: 1

Enter address: Muradpur

Enter contact: 01845742105

Enter salary: 14000

Enter data of employee 2

Enter employee name: Tuhin

Enter id: 2

Enter address: Kumira

Enter contact: 01847201596

Enter salary: 15000

Press 1 to enter data

Press 2 to show data

Press 3 to search data

Press 4 to update data

Press 5 to delete data

Press 6 to show feature

Press 7 to logout

Enter your choice: 2\_

**Show data page :**

Data of employee

Name: tuni

ID: 1

Address: df

Contact: 454

Salary: 12213

Data of employee

Name:

ID:

Address:

Contact: 454

Salary: 12213

Data of employee

Name: Mun

ID: 1

Address: Muradpur

Contact: 1845742105

Salary: 14000

Data of employee

Name: Tuhin

ID: 2

Address: Kumira

Contact: 1847201596

Salary: 15000

Press 1 to enter data

Press 2 to show data

Press 3 to search data

Press 4 to update data

Press 5 to delete data

Press 6 to show feature

Press 7 to logout

Enter your choice:

### Search data page :

```
Enter id of employee which you want to search
1
Data of employee
Name: tuni
ID: 1
Address: df
Contact: 454
Salary: 12213

Press 1 to enter data
Press 2 to show data
Press 3 to search data
Press 4 to update data
Press 5 to delete data
Press 6 to show feature
Press 7 to logout

Enter your choice: █
```

### Update data page :

```
Enter id of employee which you want to update
1
Old data of employee
Name: tuhin
ID: 1
Address: Kumira
Contact: 1845721036
Salary: 14500

Enter new data
Enter employee name: Alvi
Enter id: 1
Enter address: Rangamati
Enter contact: 01875412369
Enter salary: 14800
```

### Delete data page :

```
Press 1 to delete specific record
Press 2 to delete full record
1
Enter id of employee which you want to delete
1
```

```
Press 1 to delete specific record
Press 2 to delete full record
1
Enter id of employee which you want to delete
1
Your required record is deleted

Press 1 to enter data
Press 2 to show data
Press 3 to search data
Press 4 to update data
Press 5 to delete data
Press 6 to show feature
Press 7 to logout

Enter your choice:
```

## Feature page :

C:\Users\Shaharabi\Desktop\Code\LL\_Col.exe

---- Features ----

User Authentication:

Supports user signup (signup()) and login (login()).  
User credentials are stored in users.txt with each user having a separate file (username.txt) to store employee data.

Employee Data Management:

Insert Data (empdata()):  
Allows users to enter employee details such as name, ID, address, contact, and salary. Uses insertion sort to maintain employees sorted by ID.

Show Data (show()):

Displays all entered employee data if available.

Search Data (search()):

Allows searching for an employee by ID using binary search.

Update Data (update()):

Enables updating employee details based on ID. After updating, the data is re-sorted to maintain order.

Delete Data (del()):

Provides options to delete specific employee records by ID or delete all records.

File Handling:

Employee data is stored in text files (username.txt).  
Uses file streams (ifstream and ofstream) for reading from and writing to files.

User Interface:

Uses console-based interface with options displayed using cout.  
Clear screen (system("CLS")) to manage screen display.

How It Works:

User Management:  
Users can sign up with a new username and password.  
Existing users can log in to access their specific employee data file (username.txt).

Data Management:

Users can add, view, search, update, and delete employee records.  
Data operations ensure that the employee records remain sorted by ID for efficient searching and management.

File Operations:

Upon login, the system loads.

File Operations:

Upon login, the system loads existing employee data from the user's file (username.txt).  
Changes (additions, updates, deletions) to employee records are saved back to the file to maintain persistence.

Error Handling:

Includes basic error handling for file operations and user inputs to ensure smooth functioning of the program.

Limitations and Considerations:

The system assumes a single user per file (username.txt), limiting concurrent access.  
Error handling is minimal and may need enhancement for robustness.  
Input validations (e.g., for ID uniqueness) are not explicitly handled in the provided code.  
Overall, this Employee Management System provides essential functionalities for managing employee data through a console interface, leveraging file storage for persistence across sessions.

Press 1 to enter data  
Press 2 to show data  
Press 3 to search data  
Press 4 to update data  
Press 5 to delete data  
Press 6 to show feature  
Press 7 to logout

Enter your choice:

Screenshot

Search - Gaudi's birthday

Screenshot



## Sign-up page :

```
----- Employee Management System -----

Press 1 to Sign Up
Press 2 to Log In

Enter your choice: 1

----- Signup -----
Enter new username: Arafat
Enter new password: arafat

Your new id is creating please wait.....

Your id created successfully

Press 1 to Sign Up
Press 2 to Log In

Enter your choice: █
```

## Login with new data :

```
Press 1 to Sign Up
Press 2 to Log In

Enter your choice: 2

LOGIN
Enter username: Arafat
Enter password: arafat
```

## Log out page :

```
Process returned 0 (0x0)   execution time : 1828.635 s
Press any key to continue.
```

## Conclusion :

The Employee Management System provides essential functionalities for managing employee data through a console-based interface. It supports multiple users with individual data management and ensures data persistence through file handling. The future enhancements aim to improve user experience, security, and scalability. The current implementation serves as a robust foundation for further development and expansion.