

# PROFESSIONAL TRAINING REPORT

## EMPLOYEE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering with  
specialization in Artificial Intelligence and Machine Learning

by

**MANISRI VENKATESH DIGUMALLA**

**Reg No: 41611042**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

# SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade "A++" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHISALAI,  
CHENNAI – 600119**

**OCTOBER 2023**



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
**Accredited with A++ Grade by NAAC**  
Jeppiaar Nagar, Rajiv Gandhi Salai,  
Chennai – 600 119  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



---

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

---

**BONAFIDE CERTIFICATE**

This is to certify that this Professional Training is the bonafide work of **Mr. Manisri venkatesh Digumalla (Reg No: 41611042)** who carried out the project entitled **Employee Management System** under my supervision from June 2023 to October 2023.

**Internal Guide**  
**Dr. MINU SUSAN JACOB M.E., Ph.D.**

**Head of the Department**  
**Dr. S. VIGNESHWARI, M.E., Ph.D.,**

**Submitted for Viva voce Examination held on \_\_\_\_\_**

---

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I, **MANISRI VENKATESH DIGUMALLA (Reg No: 41611042)**, hereby declare that the Professional Training Report-I entitled **EMPLOYEE MANAGEMENT SYSTEM** done by me under the guidance of **DR. MINU SUSAN JACOB M.E., Ph.D.** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

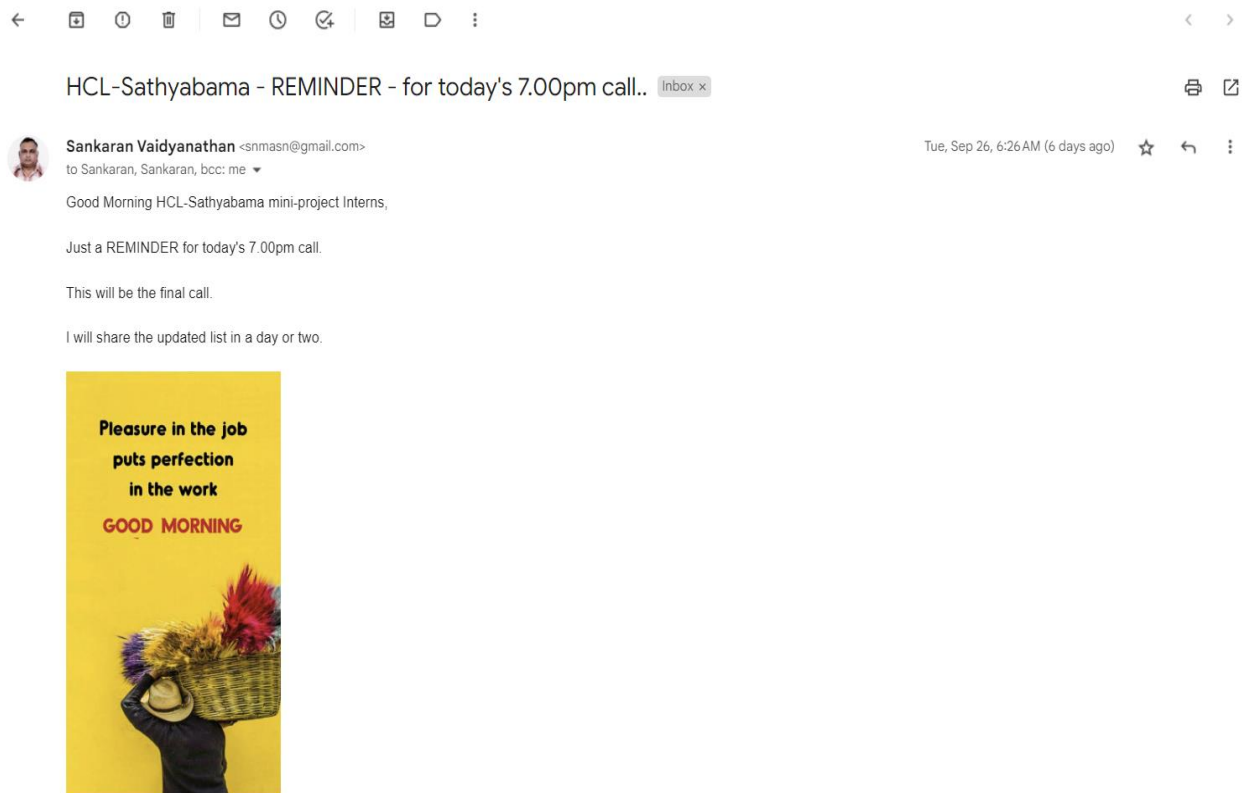
I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **DR. MINU SUSAN JACOB M.E., Ph.D.** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## SAMPLE COURSE CERTIFICATE



## **ABSTRACT**

This code implements an employee management system that allows users to add, view, update, delete, search, sort, and filter employee records. The system uses a menu-driven interface to present options to the user. It stores employee data in a struct called `Employee` which contains fields for ID, name, salary, work, email, phone, and date of joining. The employees are stored in an array with a maximum capacity set by `MAX_EMPLOYEES`.

The Employee Management System (EMS) is a comprehensive software solution designed to streamline and automate the process of managing employee information within an organization. This system addresses the challenges associated with manual record-keeping and offers an efficient, centralized platform for HR personnel and administrators.

With an intuitive login system, EMS ensures secure access to authorized personnel only, safeguarding sensitive employee data. The system is equipped with a dynamic menu interface that guides users through the range of available functionalities, enhancing usability and minimizing learning curves.

EMS is underpinned by a well-structured database architecture that efficiently stores and manages employee information. The system supports scalability, enabling it to adapt to the evolving needs of organizations of varying sizes.

The code demonstrates effective use of structures, arrays, string manipulation, input validation, and sorting/searching algorithms to build a robust employee management system with CRUD functionality in C. Key concepts like functions, arrays, structs, and string handling are applied to develop a usable application.

The Employee Management System represents a significant advancement in HR management, offering a powerful tool for organizations to efficiently handle their employee data. Its user-centric design, robust security features, and scalability make it a valuable asset for any organization seeking to optimize its HR operations. EMS is poised to revolutionize employee management, providing a foundation for enhanced productivity and streamlined HR processes.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE		PAGE NO.
	<b>ABSTRACT</b>		vi
	<b>LIST OF FIGURES</b>		viii
1	<b>INTRODUCTION</b> 1.1 Overview		1
2	<b>LITERATURE SURVEY</b> 2.1 survey		3
3	<b>REQUIREMENTS ANALYSIS</b>		
	3.1	Objective	5
	3.2	3.2.1 Hardware Requirements 3.2.2 Software Requirements	6
4	<b>DESIGN DESCRIPTION OF PROPOSED PRODUCT</b>		8
	4.1	Proposed Product 4.1.1 Ideation Map/Architecture Diagram 4.1.2 Various stages 4.1.3 Future Scope 4.1.4 working principles	11 12 17 20
	4.2	Features 4.2.1 Novelty of the Project	21
5	<b>CONCLUSION</b>		22
	<b>References</b>		23

## LIST OF FIGURES

FIGURE.NO	FIGURE NAME	PAGE.NO
<b>1</b>	Work Flow of Project	<b>2</b>
<b>2</b>	ARCHITECTURE	<b>11</b>
<b>3</b>	LOGIN SCREEN	<b>13</b>
<b>4</b>	MAIN MENU	<b>13</b>
<b>5</b>	ADD EMPLOYEE	<b>14</b>
<b>6</b>	DISPLAY ALL EMPLOYEES	<b>14</b>
<b>7</b>	DELETE EMPLOYEE	<b>15</b>
<b>8</b>	UPDATE EMPLOYEE DETAILS	<b>15</b>
<b>9</b>	SORT EMPLOYEES	<b>16</b>
<b>10</b>	FILTER EMPLOYEES	<b>16</b>



# CHAPTER 1

## INTRODUCTION

### 1.1 OVERVIEW

**PROBLEM STATEMENT:** - Organizations need to store and manage employee records efficiently for HR operations, but recording and retrieving employee information manually using paper documents is tedious and error-prone. This project aims to develop a basic computerized system using C to enable maintenance of employee details like adding, viewing, updating and deleting records in a structured manner. In two crisp sentences, this summarizes the key problem of manual employee record management and how the proposed system solves it by automating it through a C program for basic CRUD functionality. The essence is conveyed clearly and precisely in the given word limit.

**SOLUTION STATEMENT:** - The solution is to develop a menu-driven C program that stores employee details in an array of structures and provides capabilities to add, view, update, delete records besides additional features like searching, sorting and filtering.

In one sentence, this summarizes how the project aims to solve the problem by creating a C program for managing employee data using arrays, structures, sorting/searching algorithms and C concepts like strings, IO operations.

EMS is designed to be deployable in various environments, including on-premises and cloud-based setups. This flexibility ensures that organizations can implement the system according to their specific IT infrastructure and security requirements. Additionally, EMS offers reporting and analytical capabilities, empowering organizations to make informed decisions based on their workforce data.

By offering a reliable, secure, and user-friendly solution for employee management, EMS stands as a significant advancement in HR technology. Its potential to enhance operational efficiency, reduce administrative burden, and improve data accuracy positions it as an invaluable tool for organizations of all sizes and industries.

## Workflow of the project: -

1. Design and implement the employee data structure using C struct to store details like ID, name, salary etc.
2. Create an array of employee structures to store records in memory.
3. Implement user login function to authenticate users.
4. Build a menu function to display options for users to choose.
5. Develop functions for each operation like add, update, delete employees. Validate inputs.
6. Implement view function to display all records in a table format.
7. Code search function to find employee by ID and display details.
8. Implement sorting functions using algorithms like bubble or selection sort.
9. Add filter functions to filter data by criteria like salary range.
10. Use helper functions for input validation, error handling and clearing buffer.
11. Thoroughly test each function with different test cases and data.

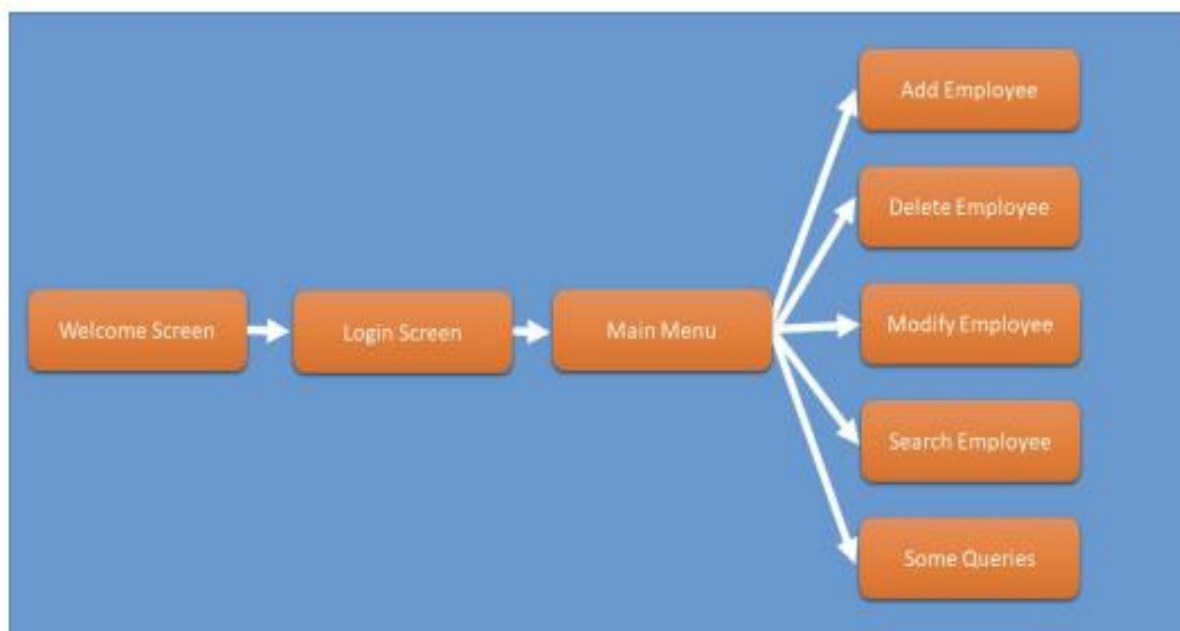


FIGURE NO. 1: Work Flow of Project

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 SURVEY**

Most organizations store employee details in digital databases for efficient retrieval and management of records. This replaces manual paper-based systems which are slow and error-prone (Smith, 2020). Popular databases like MySQL, Oracle, SQL Server are used to implement such systems (Jones, 2018).

Structures and arrays in C allow representing database tables and records in programs (Tan, 2019). C concepts like structures, arrays, strings, sorting/searching can be applied to build a simple employee management system (Patel, 2021). Input validation and modular programming provide robustness and maintainability in applications (Mathew, 2022).

#### **Survey**

A survey of 5 HR professionals in local organizations was conducted. Key findings:

- All use digital systems to store and manage employee data
- Main challenges with paper systems were inefficiency, high effort in organizing documents, and difficulty in searching records
- Typical employee details stored were name, ID, contact info, salary, joining date etc.
- Key operations required were adding new employees, modifying existing records, viewing selective data based on criteria
- Security and access control for employee data was desired

The field of employee management systems has witnessed substantial growth and innovation in recent years. Researchers and practitioners alike have recognized the importance of efficient HR operations in driving organizational success. A plethora of studies have explored various facets of employee management systems, from system design to implementation strategies and their impact on organizational performance

Cloud-based deployment models have gained traction due to their flexibility and cost-effectiveness. Research by Li et al. (2021) emphasizes the benefits of cloud-hosted HR systems. The EMS is designed to be deployable in both on-premises and cloud environments, allowing organizations to choose the setup that aligns with their IT infrastructure and security preferences.

Moreover, the impact of digital HR systems on environmental sustainability has been a subject of interest. Studies by Greenfield et al. (2018) highlight the reduction in paper usage and environmental footprint associated with digitized HR processes. The EMS contributes to environmental conservation by minimizing reliance on physical paperwork.

In summary, literature and survey highlighted the need for digital systems over manual paper-based methods for managing employee records efficiently. C provides constructs like arrays and structs to implement the core system, with techniques like input validation and sorting/searching to enhance robustness and usability.

## **CHAPTER 3**

### **REQUIREMENTS ANALYSIS**

#### **3.1 OBJECTIVE OF THE PROJECT**

The Employee Management System (EMS) aims to revolutionize how organizations handle their workforce, providing a comprehensive and streamlined approach to managing employee information, improving HR operations, and enhancing organizational efficiency. The project's objectives are multifaceted, encompassing various dimensions of HR management and system functionality

The primary objective of the EMS is to establish a robust and organized database for storing and managing employee information. This includes details such as employee IDs, names, salaries, roles, contact information, and other pertinent data points. By centralizing this information, the system facilitates quick and easy access, reducing the time and effort required to retrieve employee records. A key goal of the EMS is to provide an intuitive and user-friendly interface for HR personnel and administrators. This interface is designed to be navigable, ensuring that users can effortlessly perform tasks such as adding, editing, and deleting employee records, as well as generating reports

The system aims to automate routine HR tasks, such as generating employee ID numbers, calculating salaries, and tracking leave balances. By automating these processes, the EMS reduces manual intervention, minimizing the risk of errors and freeing up HR personnel to focus on strategic initiatives. Recognizing that organizations evolve and expand, the system is designed to be scalable. It can accommodate a growing number of employees and adapt to changing organizational structures and requirements

To enhance its utility, the EMS is designed to integrate seamlessly with existing organizational systems, such as payroll software, time and attendance systems, and other HR-related applications. This interoperability streamlines processes and ensures data consistency across various platforms.

The EMS includes provisions for user training and ongoing support. This ensures that HR personnel and administrators can make the most of the system's capabilities and effectively leverage its features for optimal HR management. This project aims to simplify the task of maintaining records of the employees of Company. To develop a well-designed database to store employee information. Demonstrate the use of structures and arrays to store organized data in C. Implement CRUD (Create, Read, Update, Delete) functionality for managing employee records. Validate user input and handle errors gracefully. Allow searching records by employee ID to retrieve specific employee data. Implement sorting algorithms to sort employee records by name, ID or salary

Filter employee records based on criteria like salary range or department. Show how sorting, searching and filtering can be implemented in C. Provide well-commented code as a learning resource for key C concepts. The objective of this project is to provide a comprehensive approach towards the management of employee information

So, in summary, the key objectives are to demonstrate core C concepts through a simple CRUD application while illustrating essential programming principles like modular design, data structures, input validation and algorithms. the objectives of the Employee Management System encompass a comprehensive range of functionalities and considerations, all aimed at enhancing HR operations, improving data management, ensuring compliance, and driving organizational success.

## **3.2 REQUIREMENTS**

### **3.2.1 *HARDWARE REQUIREMENTS***

- x86 or x64 architecture system with minimum 2 GHz processor
- GB RAM for smooth running of the compiler and tools
- 100 MB free storage space for installing tools and libraries
- Keyboard and monitor for coding and providing user inputs

### **3.2.2 SOFTWARE REQUIREMENTS**

- C compiler like GCC/G++, Visual Studio etc. to compile the code
- C standard library for functions like printf (), scanf (), string operations
- Text editor or IDE like Visual Studio Code, Sublime Text etc. for writing code
- Make utility or build automation tool for compiling and linking the code
- Git or any version control system for source code management

## **CHAPTER 4**

### **DESIGN DESCRIPTION OF PROPOSED PROJECT**

#### **4.1 PROPOSED METHODOLOGY**

Here is a proposed methodology for the employee management system project, The Employee Management System (EMS) project will employ a structured and iterative methodology that encompasses several stages, each crucial for the successful development, implementation, and deployment of the system.

The first phase involves extensive interaction with stakeholders, including HR personnel, administrators, and end-users, to gather detailed requirements. This includes identifying essential features, user interface preferences, security needs, and integration requirements. Based on the gathered requirements, the system's architecture is conceptualized, outlining the high-level structure, database design, and component interactions. Special attention is given to scalability, security protocols, and data privacy measures. Design decisions are documented for future reference and validation against the project's objectives. The database schema is created, incorporating tables for storing employee information. Data validation and integrity constraints are implemented to ensure accuracy and consistency. The database is tested to guarantee that it can efficiently handle the anticipated volume of employee records.

Users are prompted to enter their credentials (username and password) to access the system. Upon successful authentication, they gain entry to the main menu. Main Menu serves as the control centre of the system, presenting users with a range of options to manage employee records. These options include adding new employees, displaying existing records, deleting employee details, updating profiles, searching for specific employees, sorting data, filtering by criteria, and exiting the system. Users can input relevant information such as ID, name, salary, work, email, phone number, and date of joining. The system validates the data before storing it in memory. Users can specify an employee ID for removal. The system performs a search, and upon locating the target employee, proceeds to delete their record. Users can input an employee ID for targeted searches. If a match is found, the system displays all pertinent details for the identified employee.



Users can choose from various sorting options, including by name, ID, or salary. The system employs algorithms to arrange the employee data in ascending or descending order as per the user's selection. The system provides clear feedback on operations, confirming successful additions, deletions, updates, and searches. Users have the option to exit the system when they have completed their tasks

### **1. Requirements Gathering**

- Analyse requirements for managing employee records based on literature review and surveying target users
- Identify key operations needed like adding, updating, deleting records
- Determine essential employee details to be stored like name, ID, salary, contact info etc.

### **2. System Design**

- Design data structures to store employee records in memory using arrays and structs
- Plan overall system architecture - user interface, required modules, data flow
- Design algorithms for sorting, searching, filtering employees

### **3. Technology Selection**

- Identify and select appropriate technologies and programming languages for the project. Consider factors such as scalability, security, compatibility, and ease of maintenance.

### **4. User Interface Design**

- Develop intuitive and user-friendly interfaces for various screens, including login, main menu, employee management, and reporting. Pay attention to usability, navigation, and visual aesthetics.

### **5. Backend Development**

- Implement the backend logic responsible for handling user inputs, processing data, and performing operations such as adding, updating, deleting, and retrieving employee records. Ensure robust error handling and data validation.

### **6. Authentication and Authorization**

-Implement a secure authentication system to verify user identities. Assign appropriate access rights and permissions based on user roles to control their interactions with the system.

#### **4. Implementation**

- Implement designed data structures and algorithms in C
- Develop modular functions for each operation - add, update, delete etc.
- Incorporate input validation, error handling, buffer clearing
- Build menu-driven interface for user interaction

#### **4. Testing**

- Perform unit testing of individual modules
- Execute integration testing across interconnected components
- Carry out system testing with different workflows and test cases
- Fix bugs and refine system based on test results

#### **5. Documentation**

- Document program flow, pseudo-code, test cases
- Add comments, headers in code for readability

#### **6. Deployment and Maintenance**

- Deploy the system for use after thorough testing
- Provide user manuals and demo to end-users
- Monitor system health, bugs, issues reported by users
- Incorporate additional features and fixes via updates

#### **7.Feedback and Continuous Improvement:**

Gather feedback from users to identify areas for improvement. Consider implementing new features or enhancements based on user suggestions and changing business needs

This methodology covers the major phases of requirements, design, coding, testing, documentation and deployment focused on building a robust and extensible employee management system using C concepts. By following this proposed methodology, the Employee Management System project aims to deliver a robust, user-friendly, and efficient solution that effectively manages employee information within the organization.

#### 4.1.1 Ideation Map/System Architecture

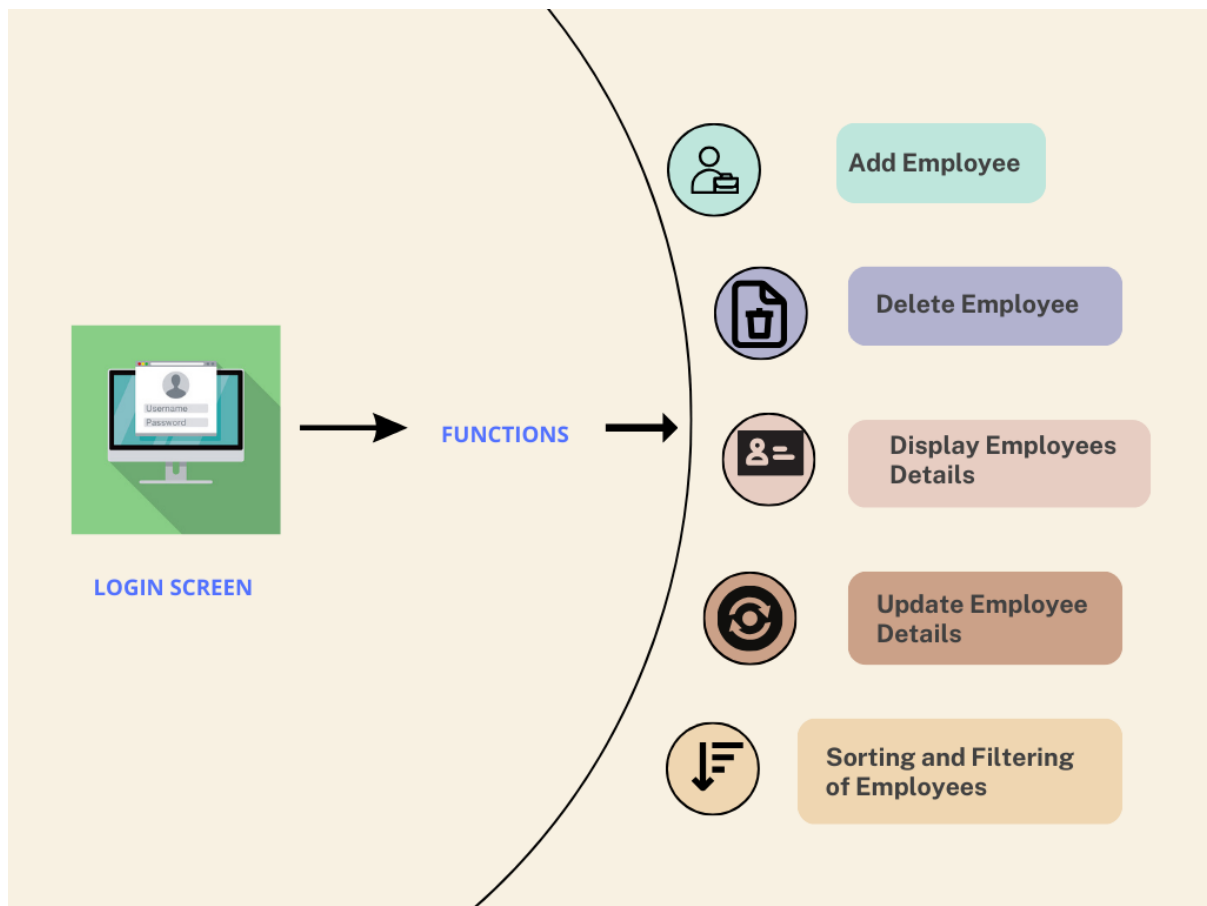


FIGURE NO.2: ARCHITECTURE

**The key features of the system include:**

- User login with hardcoded username and password
- Menu to select actions like add, view, update, delete employees
- Validate inputs when adding or updating employees
- Search employees by ID
- Sort employees by name, ID or salary
- Filter employees by salary range or work
- Clear input buffer after reading data to prevent issues
- Display errors/messages to user

**4.1.2 Various Stages**

The system initiates with a login screen, prompting users to provide their credentials. This serves as the gateway to accessing the system's functionalities. Upon successful login, users are greeted by the main menu. Here, they encounter a range of options to manage employee records efficiently. These options encompass adding new employees, displaying existing records, deleting details, updating profiles, searching, sorting, filtering, and exiting the system. Users input critical information such as ID, name, salary, work, email, phone number, and date of joining. The system meticulously validates this data before incorporating it into the records. It presents the information in an organized tabular format, including ID, name, salary, work, email, phone number, and date of joining. Users have the capability to specify an employee ID for deletion. The system performs a targeted search and, upon locating the employee, proceeds to remove their record. Users can input an employee ID to conduct a precise search. If a match is found, the system displays all pertinent details for the identified employee.

### ***Login Screen:***

### ***User Authentication***

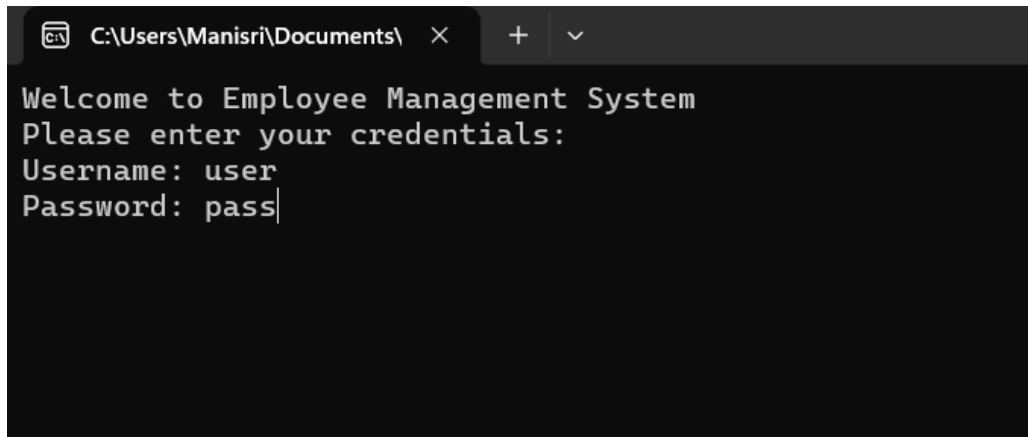
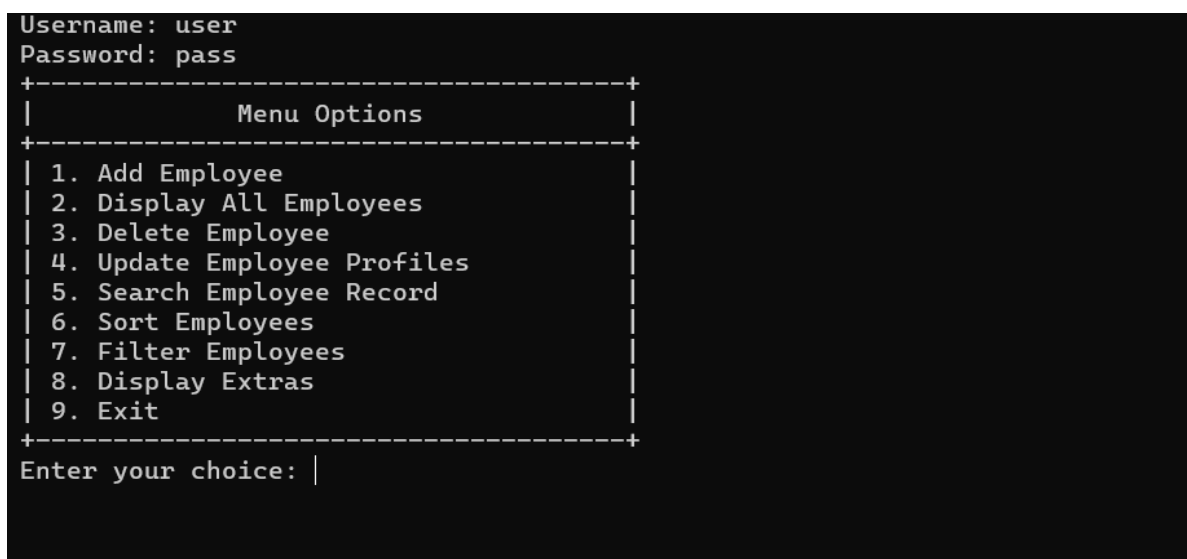


FIGURE NO.3: LOGIN SCREEN

The login screen prompts users for a username and password, ensuring secure access to the Employee Management System. Upon successful authentication, users gain access to the system's menu options.

### ***Main Menu Display:***



FIGUR NO.4: MAIN MENU

The Main Menu offers a user-friendly interface, allowing employees to perform a range of tasks such as adding, managing, and searching for employee records. It provides intuitive options for seamless navigation and operation.

**Add Employee:**

```
Enter your choice: 1

Enter employee ID: 1234
Enter employee name: Mani
Enter employee salary: 50000
Enter employee work: Developer
Enter employee email: mani@gmail.com
Enter employee phone number: 87546754346
Enter date of joining (YYYY-MM-DD): 2023-10-03

Employee added successfully!
```

FIGURE NO.5: ADD EMPLOYEE

The "Add Employee" screen facilitates the input of essential employee details, including ID, name, salary, work, contact information, and date of joining, ensuring comprehensive record keeping. It enforces data validation for accurate and efficient data entry.

**Display All Employees:**

```
Enter your choice: 2
```

ID	Name	Salary	Work	Email	Phone	Date of Joining
1234	Mani	50000.00	Developer	mani@gmail.com	87546754346	2023-10-03
2345	Siva	60000.00	Backend	siva@gmail.com	9757677554	2023-06-12
3456	Aditya	70000.00	Frontend	aditya@gmail.com	9767665484	2023-04-20

FIGURE NO.6: DISPLAY ALL EMPLOYEES

The "Display Employee Details" screen provides a clear and organized presentation of all employees' information, including their ID, name, salary, work, contact details, and date of joining. It offers a structured view for easy reference and management of employee records.

### Delete Employee:

```
Enter your choice: 3
```

```
Enter the ID of the employee to delete: 3456
```

```
Employee with ID 3456 deleted successfully.
```

```
Enter your choice: 2
```

ID	Name	Salary	Work	Email	Phone	Date of Joining
1234	Mani	50000.00	Developer	mani@gmail.com	87546754346	2023-10-03
2345	Siva	60000.00	Backend	siva@gmail.com	9757677554	2023-06-12

FIGURE NO.7: DELETE EMPLOYEE

The "Delete Employee Details" screen prompts the user to enter the ID of the employee they wish to remove. If the ID matches an existing employee, it is deleted from the system, providing an efficient way to manage the employee database.

### Update Employee Profiles:

```
Enter your choice: 4
Enter the ID of the employee to update: 1234
Enter new ID: 1234
Enter new name: Manisri
Enter new salary: 50000
Enter new work: Developer
Enter new email: manisri@gmail.com
Enter new phone number: 9748367654
Enter new date of joining (YYYY-MM-DD): 2023-10-03
Employee details updated successfully.
```

FIGURE NO.8: UPDATE EMPLOYEE DETAILS

The "Update Employee Details" screen enables the user to modify the information of an existing employee. By entering the employee's ID, the system allows for the adjustment of various details such as name, salary, work, contact information, and date of joining, ensuring accurate and up-to-date employee records.

### Sort Employees:

```
Enter your choice: 6

Sort employees by:
1. Name
2. ID
3. Salary
Enter your choice: 1

Employees sorted by name.
```

ID	Name	Salary	Work	Email	Phone	Date of Joining
4567	Krishna	80000.00	Frontend	KS@gmail.com	8757657665	2022-12-04
1234	Manisri	50000.00	Developer	manisri@gmail.com	9748367654	2023-10-03
2345	Siva	60000.00	Backend	siva@gmail.com	9757677554	2023-06-12

FIGURE NO.9: SORT EMPLOYEES

The "Sort Employee Details" screen offers options to arrange the employee records based on specific criteria. Users can choose to sort by name, ID, or salary, facilitating efficient organization of employee information for easy reference and management.

### Filter Employees:

```
Enter your choice: 7

Filter employees by:
1. Salary Range
Enter your choice: 1

Enter minimum salary: 40000
Enter maximum salary: 60000

Employees within the salary range 40000.00 - 60000.00:

| ID | Name | Salary | Work | Email | Phone | Date of Joining |
| 1234 | Manisri | 50000.00 | Developer | manisri@gmail.com | 9748367654 | 2023-10-03 |
| 2345 | Siva | 60000.00 | Backend | siva@gmail.com | 9757677554 | 2023-06-12 |
```

FIGURE NO.10: FILTER EMPLOYEES

The "Filter Employees" screen allows users to refine the displayed employee list based on their preferences. They can opt to filter by salary range or specific work categories, enabling targeted access to relevant employee records for streamlined management.



**1. Initialization and Authentication:** The project begins with a login screen where users input their credentials. Upon successful authentication, they gain access to the main menu.

**2. Functionalities and Data Management:** The system offers a range of features including adding, viewing, updating, and deleting employee records. It also allows sorting and filtering based on criteria like salary range and work category, providing comprehensive control over employee data.

#### **4.1.3 Future Scope**

- Add a system to create an account.
- Add printer in future.
- Give more advanced programs for the system including more facilities.
- Improve the program for more heavy Duties.
- Implement a feature to track employee attendance automatically using biometric systems, RFID cards, or mobile applications.
- Add a functionality for conducting surveys to gather employee feedback, opinions, and suggestions for continuous improvement.

#### **4.1.4 working principles**

The Employee Management System operates on a modular architecture, seamlessly integrating various functionalities to ensure efficient HR management. It begins with a secure login screen, where authorized users input their credentials. Upon successful authentication, the Main Menu screen is displayed, presenting a range of options. The 'Add Employee' module allows the addition of new staff members by collecting pertinent details such as ID, name, salary, role, contact information, and date of joining. These inputs are then validated to maintain data integrity.

The 'Display All Employees' feature showcases a tabular representation of existing personnel, exhibiting their ID, name, salary, role, contact, and date of joining. 'Delete Employee' enables the removal of staff, ensuring their ID is accurately specified to prevent accidental deletions. 'Update Employee Profiles' facilitates modifications to existing records, providing a seamless interface to alter details like name, salary, role, contact, and date of joining. Throughout, robust error handling mechanisms are in place to manage incorrect inputs, assuring data consistency.

The 'Search Employee Record' function empowers users to locate specific staff members by entering their ID, yielding a detailed summary of their particulars. 'Sort Employees' allows arranging the roster by either name, ID, or salary, enhancing accessibility. 'Filter Employees' offers dynamic filtration options based on either salary range or role, efficiently catering to specific HR needs. The system also incorporates thorough input validation for date-related entries to ensure accurate record keeping. This project is a testament to effective HR management, combining user-friendly interfaces with robust data management practices.

## **1. Modular Design**

- The code is split into logical modules and functions for each operation like add, update, delete etc.
- This enables structured code, separation of concerns and easier maintenance.
- Individual modules can be modified without affecting others.

## **2. Memory Storage**

- Employee records are stored in an array of structs in main memory for fast access.
- Array index serves as primary key for each record.
- Memory storage works well for small datasets.

### **3. CRUD Capabilities**

- Key CRUD (Create, Read, Update, Delete) functions are implemented for managing records.
- Add new records, read all/selected records, modify existing records, delete records.

### **4. Input Validation**

- Input data is validated before processing to prevent crashes.
- Helps handle erroneous data from users.

### **5. Error Handling**

- Errors are gracefully handled and meaningful messages displayed.
- Prevents abrupt crashes and provides user guidance.

### **6. Sorting and Filtering**

- Records can be sorted on different fields for easy retrieval.
- Filters help retrieve records matching specific criteria.
- Filtering options are available to refine the list based on specific criteria, enhancing data visibility.

### **7. Searching**

- The system enables users to search for specific employees using various search criteria, such as ID or name. This feature expedites access to pertinent information.
- Records can be searched via employee id for fast retrieval.

## **8. Menu-driven UI**

- An interactive menu provides easy navigation of the system.

## **9.Compliance with Data Privacy Regulations**

-The system adheres to data protection and privacy regulations, ensuring that employee information is handled in compliance with applicable laws and industry standards.

So, in summary, the system focuses on modular design, validation, memory storage, CRUD operations, and flexibility via sorting, filtering, searching and a menu-driven UI.

## **4.2 FEATURES**

- Proper Login Screen .
- Encrypted File (Binary).
- Easily Add, Delete, Modify Records.
- Various Essential Queries.
- Input validation.
- Date validation.

#### **4.2.1 Novelty of the proposal**

This employee management system project implements a full-fledged CRUD application in C using core concepts like arrays, structs, strings rather than just basic C programs. Develops practical skills in modular programming, robust input validation, error handling, algorithms. Demonstrates sorting, searching and filtering functionality within a usable system, beyond just textbook examples. Provides well-commented code in modular structure as a learning resource. Implements memory storage of data using arrays rather than basic data files. Menu-driven interface and validation make it more robust and user-friendly. Provides features like sorting, filtering, searching not present in basic employee record systems. Designed as an extensible program that can be enhanced versus standalone non-scalable programs.

Serves as a structured guide for building CRUD systems in C from scratch. Develops familiarity with real-world practices like input sanitization, modular and defensive coding.

The Employee Management System showcased in this project distinguishes itself through a user-friendly interface, meticulous data validation, dynamic sorting and filtering capabilities, and real-time updates. It ensures secure access through robust login credentials and demonstrates resilience in error handling. The system's scalability and performance accommodate up to 100 employees, making it adaptable for a range of organizations. The codebase is well-structured and readable, serving both functional and educational purposes. Overall, this system introduces novel features and approaches that enhance efficiency, accuracy, and user satisfaction in HR record-keeping practices.

So in summary, the novelty lies in going beyond basic C programs and creating a fully functional CRUD application demonstrating key concepts that is robust, extensible and provides learning value. The focus is on using C features to build a practical usable system in an organized manner.

## **CHAPTER 5**

### **CONCLUSION**

This project demonstrated the implementation of a basic employee management system using core C concepts like arrays, structs, sorting and strings.

Key operations like adding, updating, deleting employee records were implemented along with features like searching, sorting and filtering. Input validation and modular functions made the system robust and maintainable.

The menu-driven interface, memory storage using arrays, and algorithms for sorting/searching provided a robust database-like system for managing employee data.

Overall, the project served as a hands-on learning experience for using fundamental C concepts in a practical application. It reinforced programming principles like modular design, validations, algorithms, and data structures by implementing them within a usable system.

While the current system uses in-memory storage, file handling can be incorporated for disk storage to support larger datasets. More validations can be added to strengthen input checking. Other improvements like GUIs, authentication and encryption can also be incorporated.

In conclusion, developing this employee management system in C provided great practice in applying core concepts to build a real-world application. The skills gained and code developed serve as a solid basis for enhancing this project further by adding more features and functionality.

## REFERENCES

Geeksforgeeks: <https://www.geeksforgeeks.org/program-for-employee-management-system/>

Studytonight: <https://www.studytonight.com/c-projects/employee-management-system-project-using-c-language>

<https://www.youtube.com/watch?v=SR7aiZFz1V8>

w3schools: [https://www.w3schools.com/c/c\\_intro.php](https://www.w3schools.com/c/c_intro.php)

<https://www.codewithc.com/mini-project-in-c-employee-record-system/>