***Employe Management System***

My project is about ***Employe Management System***

You can add employes and their information and you can modify their records and delete if needed in a system that contain only 100 employees fixed with an array and a structure for the system to make it easier for the developer with a smooth interface for the user that’s easy to use

***The code***:

A computer screen with white text

Description automatically generated

The code starts with a define to define MAX as 100 when mentioned below.

And a struct employee to represent the structure of employee and group different variables under one name which in our case is **employee** and it’s used to represent complex data type.

A computer screen shot of a program code

Description automatically generated

Here we used void to define our function and the functions such as add\_record is declared with a void with no return value and in this case “add\_record” it updates the array employee and increases count but doesn’t need to return anything and so on

In view\_list it prints the list of employees to the console which is an output operation

A screen shot of a computer program

Description automatically generated

Here in modify\_record this function is used to modify the details of an existing employee in the employees array based on the employees ID how it works.

You input the employee ID and FOR loop iterates through the employees array to find matching ID and if found (found is set to 1) the user inputs the new name, age, and salary etc. and the employees array is updates and a success message is printed and loop is terminated using **break**

If employee not found (found remain 0) a message is printed indicating that’s no employee with the ID given

* Found Flag is used to check whether the employee ID exists in the array.
* The scanf statement allows the user to overwrite the existing data for the matched employee.

IN delete\_record

The purpose: this function is used to delete an employee record from the employees array based on the employee ID.

You input the employee ID and the FOR loop iterates through the employees array to find a match IF found a nested for loop shifts all subsequent employee records one position to the left, overwriting the current employee record.

The COUNT is decremented to reflect the deletion. A success message is printed, and the loop is terminated using BREAK.

If employee not found (found remains 0) a message is printed indicating that no employee with that ID

THE nested loop ensures that the array remains the same after deleting an employee by shifting to the left.

COUNT decrement reduces the total number of employee in the list.

And at the end the is if (! found) which means if not found

Print employee with ID %d not found.\n

The underlined %d means ID with h this int number and the \n means a new line the make the output more organized

A computer screen shot of a program code

Description automatically generated

Last thing is the Int main which is the main function of the program which serves the point for the employee management system.

Variable declaration int “choice” is declared to store the user menu selection.

The DO purpose to display a menu for the user to interact with the system with options such as:

* Add a new employee.
* View the list of employees.
* Modify employee records.
* Delete an employee record.
* Lastly, exit the program.

The user input is captured using the scanf and stored in “choice” variable.

The (SWITCH\_CASE) purpose

Based on the user input (choice) the program executes the corresponding function

For example, case1 calls add\_record function to add new employee.

Case2 calls the view\_list function to display the list of employees.

Case3 calls the modify\_record function to update an employee detail.

And so, on

But in case5 it prints a message and exit the system

In “default” it handles invalid inputs by printing an error message

Loop the menu “while” purpose to ensure the menu keeps displaying until user chooses to exit (choice==5)

Thw do-while loop executes the menu and waits for the user to input.

If the user selects 5 loop terminates and program exits

**Return 0**

Is used to indicate that the program has successfully executed.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated