INTRODUCTION:

What is EMS?

- Employee Management System is a distributed application, developed to maintain the details of employees working in any organization.
- The EMS has been developed to override the problems prevailing in the practicing manual system.
- It maintains the information about the personal and official details of the employees.

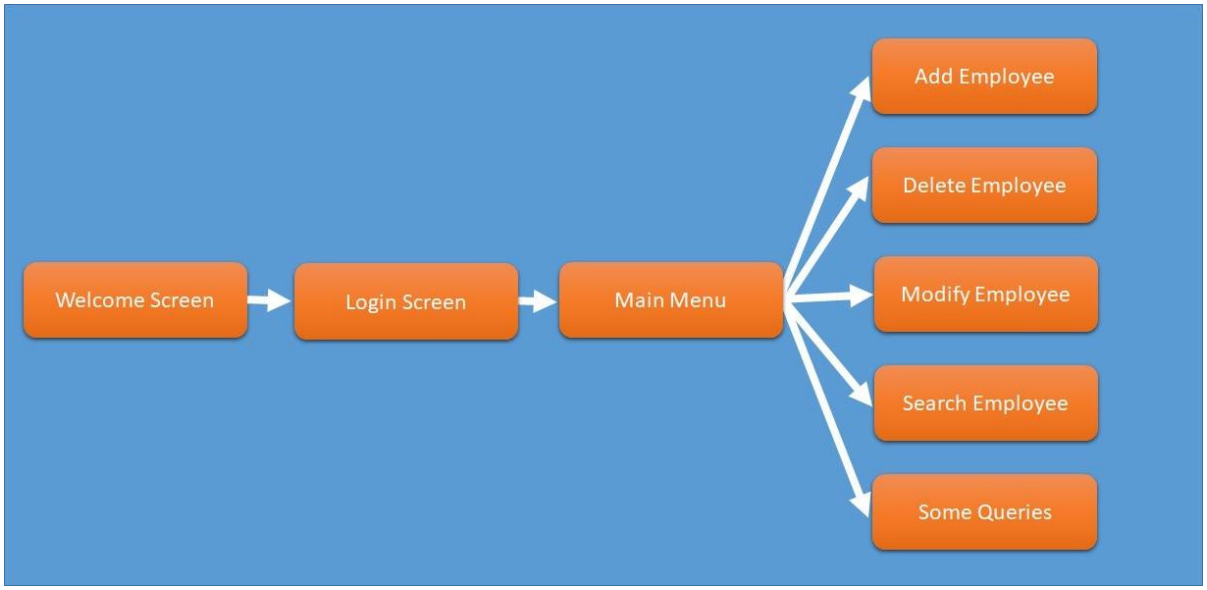
OBJECTIVES OF OUR PROJECT:

- This project aims to simplify the task of maintaining records of the employees of Company.
- To develop an well-designed database to store employee information.
- Provides full functional reports to management of Company.
- The objective of this project is to provide a comprehensive approach towards the management of employee information.

FEATURES :

- Proper Login Screen (Invisible Password Typing).
- Password Protected.
- Encrypted File (Binary).
- Easily Add, Delete, Modify Records.
- Various Essential Queries.

BASIC STRUCTURE :



FUNCTIONS AND CODE:

- Header files:
  - #include<stdio.h>
  - #include<string.h>
  - #include<stdlib.h>
  - #include<math.h>

- **Main Funtion:**

Code:

```
int main()
{
 FILE * fp;
Employee e;
int option;
char another;

if((fp=fopen("employeeInfo.txt","rb+"))==NULL)
{
    if((fp=fopen("employeeInfo.txt","wb+"))==NULL)
       {
           printf("can't open file");
           return 0;
       }
}
char username[10],password[10];
printHead();
welcome();
printHead();
printf("\n\t\t\t\t  Login Screen");
printf("\n\t\t\t      Enter Your Credential");
printf("\n\n\n\t\tUsername: ");
scanf("%s",username);
printf("\n\t\tPassword: ");
int i;
```

```
i=0;
     do
     {
          password[i] = getch();
          if(password[i] == 13 )
          {
               break;
          }
          else if(password[i]==8 && i>0)
          {
               printf("%c%c%c",8,32,8);
               i--;
          }
          else
          {
               printf("*");
               i++;
          }
     }while(password[i]!=13);
     password[i] = '\0';


if(((strcasecmp(username,"admin"))==0)&&((strcasecmp(passw
ord,"pass")==0)))
{
    while(1)
{
    printHead();
    printf("\n\t\t\t\tMain Menu");
    printf("\n\n\n");
```

```
    printf("\n\n\t\t\t1. Add Employee");
    printf("\n\n\t\t\t2. Delete Employee");
    printf("\n\n\t\t\t3. Modify Employee");
    printf("\n\n\t\t\t4. Display Employee List");
    printf("\n\n\t\t\t5. Search Record");
    printf("\n\n\t\t\t6. Display Basic Info");
    printf("\n\n\t\t\t7. Display Basic Contact Info");
    printf("\n\n\t\t\t8. List of Male Employee");
    printf("\n\n\t\t\t9. List of Female Employee");
    printf("\n\n\t\t\t10. List of Employee From Dhaka");
    printf("\n\n\t\t\t11. List of Employee From Others
District");
    printf("\n\n\t\t\t12. List of Employee of Main Branch");
    printf("\n\n\t\t\t13. List of Employee of Others
Branch");
    printf("\n\n\t\t\t0. EXIT");


    printf("\n\n\t\tEnter Your Option :--> ");
    scanf("%d",&option);


    switch(option)
    {
        case 0: return 1;
                break;
        case 1: add(fp);
                break;
        case 2: fp=del(fp);
```

```
        break;
case 3: modify(fp);
        break;
case 4: displayList(fp);
        break;
case 5: searchRecord(fp);
        break;
case 6: displaybasic(fp);
        break;
case 7: basiccontact(fp);
        break;
case 8: maleemp(fp);
        break;
case 9: femaleemp(fp);
        break;
case 10: frmdhaka(fp);
        break;
case 11: frmors(fp);
        break;
case 12: mainbr(fp);
        break;
case 13: otherbr(fp);
        break;
default: printf("\n\t\tYou Pressed wrong key");
          printf("\n\t\tProgram terminated");
          getch();
          exit(0);
```

```
    }
}
}
else {
    printf("\n\t\tLogin Failed");
}
return 1;
}
```

- **Funtion For Add Employee Record:**

Code:

```
void add(FILE * fp)
{
printHead();

printf("\n\t\t\t\Add Employee");
char another='y';
Employee e;

fseek(fp,0,SEEK_END);
while(another=='y'||another=='Y')
{
    printf("\n\n\t\tEnter ID number: ");
    scanf("%d",&e.id);

    printf("\n\n\t\tEnter Full Name of Employee: ");
    fflush(stdin);
```

```
    fgets(e.name,100,stdin); //fgets takes an extra \n
character as input
    e.name[strlen(e.name)-1]='\0';



    printf("\n\n\t\tEnter Designation: ");
    fflush(stdin);
    fgets(e.desgn,10,stdin); //fgets takes an extra \n
character as input
    e.desgn[strlen(e.desgn)-1]='\0';



    printf("\n\n\t\tEnter Gender: ");
    fflush(stdin);
    fgets(e.gender,10,stdin); //fgets takes an extra \n
character as input
    e.gender[strlen(e.gender)-1]='\0';

    printf("\n\n\t\tEnter Branch: ");
    fflush(stdin);
    fgets(e.branch,50,stdin);
    e.branch[strlen(e.branch)-1]='\0';



    printf("\n\n\t\tEnter Salary: ");
    scanf("%f",&e.sal);

    printf("\n\n\t\tEnter Present Address: ");
    fflush(stdin);
```

```
    fgets(e.psaddr,200,stdin);
    e.psaddr[strlen(e.psaddr)-1]='\0';

    printf("\n\n\t\tEnter Permanant Address: ");
    fflush(stdin);
    fgets(e.prtaddr,200,stdin);
    e.prtaddr[strlen(e.prtaddr)-1]='\0';

    printf("\n\n\t\tEnter Phone: ");
    fflush(stdin);
    fgets(e.phone,50,stdin);
    e.phone[strlen(e.phone)-1]='\0';

    printf("\n\n\t\tEnter E-mail: ");
    fflush(stdin);
    fgets(e.mail,20,stdin);
    e.mail[strlen(e.mail)-1]='\0';

    fwrite(&e,sizeof(e),1,fp);

    printf("\n\n\t\tWant to enter another employee info
(Y/N)\t");
    fflush(stdin);
    another=getchar();
}
}
```

➢ Function For Delete Employee Record:

Code:

```
FILE * del(FILE * fp)
{
 printHead();
printf("\n\t\t\t\Delete Employee");
Employee e;
int flag=0,tempid,siz=sizeof(e);
FILE *ft;

if((ft=fopen("temp.txt","wb+"))==NULL)
{
    printf("\n\n\t\t\t\\t!!! ERROR !!!\n\t\t");
    system("pause");
     return fp;
}

printf("\n\n\tEnter ID number of Employee to Delete the
Record");
printf("\n\n\t\t\tID No. : ");
scanf("%d",&tempid);

rewind(fp);



while((fread(&e,siz,1,fp))==1)
{
    if(e.id==tempid)
```

```
    { flag=1;
    printf("\n\tRecord Deleted for");

printf("\n\n\t\t%s\n\n\t\t%s\n\n\t\t%d\n\t",e.name,e.branch,e.id);
    continue;
    }

    fwrite(&e,siz,1,ft);
}



fclose(fp);
fclose(ft);

remove("employeeInfo.txt");
rename("temp.txt","employeeInfo.txt");

if((fp=fopen("employeeInfo.txt","rb+"))==NULL)
{
    printf("ERROR");
    return  NULL;
}

if(flag==0) printf("\n\n\t\t!!!! ERROR RECORD NOT FOUND \n\t");

printChar('-',65);
printf("\n\t");
```

```
system("pause");
return fp;
}
```

- **Function For Modify Employee Record:**

Code:

```
void modify(FILE * fp)
{
printHead();
printf("\n\t\t\t\Modify Employee");
Employee e;
int i,flag=0,tempid,siz=sizeof(e);
float sal;

printf("\n\n\tEnter ID Number of Employee to Modify the
Record : ");
scanf("%d",&tempid);

rewind(fp);

while((fread(&e,siz,1,fp))==1)
{
    if(e.id==tempid)
        {flag=1;
        break;
        }
}
```

```
if(flag==1)
    {
    fseek(fp,-siz,SEEK_CUR);
    printf("\n\n\t\tRecord Found");
    printf("\n\n\t\tEnter New Data for the Record");

     printf("\n\n\t\tEnter ID number: ");
    scanf("%d",&e.id);

    printf("\n\n\t\tEnter Full Name of Employee: ");
    fflush(stdin);
    fgets(e.name,100,stdin); //fgets takes an extra \n
character as input
    e.name[strlen(e.name)-1]='\0';



    printf("\n\n\t\tEnter Designation: ");
    fflush(stdin);
    fgets(e.desgn,10,stdin); //fgets takes an extra \n
character as input
    e.desgn[strlen(e.desgn)-1]='\0';



    printf("\n\n\t\tEnter Gender: ");
    fflush(stdin);
    fgets(e.gender,10,stdin); //fgets takes an extra \n
character as input
    e.gender[strlen(e.gender)-1]='\0';
```

```
printf("\n\n\t\tEnter Branch: ");
fflush(stdin);
fgets(e.branch,50,stdin);
e.branch[strlen(e.branch)-1]='\0';


printf("\n\n\t\tEnter Salary: ");
scanf("%f",&e.sal);

printf("\n\n\t\tEnter Present Address: ");
fflush(stdin);
fgets(e.psaddr,200,stdin);
e.psaddr[strlen(e.psaddr)-1]='\0';

printf("\n\n\t\tEnter Permanant Address: ");
fflush(stdin);
fgets(e.prtaddr,200,stdin);
e.prtaddr[strlen(e.prtaddr)-1]='\0';

printf("\n\n\t\tEnter Phone: ");
fflush(stdin);
fgets(e.phone,50,stdin);
e.phone[strlen(e.phone)-1]='\0';

printf("\n\n\t\tEnter E-mail: ");
fflush(stdin);
fgets(e.mail,20,stdin);
```

```
    e.mail[strlen(e.mail)-1]='\0';

    fwrite(&e,sizeof(e),1,fp);
}

else printf("\n\n\t!!!! ERROR !!!! RECORD NOT FOUND");

printf("\n\n\t");
system("pause");

}
```

- **Function For Search Employee Record:**

Code:

```
void searchRecord(FILE *fp)
{printHead();
printf("\n\t\t\t\Search Employee");
int tempid,flag,siz,i;
Employee e;
char another='y';

siz=sizeof(e);

while(another=='y'||another=='Y')
{
printf("\n\n\tEnter ID Number of Employee to search the
record : ");
scanf("%d",&tempid);
```

```
rewind(fp);

while((fread(&e,siz,1,fp))==1)
{
    if(e.id==tempid)
        {flag=1;
        break;
        }
}

if(flag==1)
    {
    printf("\n\t\tNAME : %s",e.name);
        printf("\n\n\t\tID : %d",e.id);
        printf("\n\n\t\tDESIGNATION : %s",e.desgn);
        printf("\n\n\t\tBRANCH : %s",e.branch);
        printf("\n\n\t\tSALARY: %.2f",e.sal);
        printf("\n\n\t\tPRESENT ADDRESS : %s",e.psaddr);
        printf("\n\n\t\tPERMANANT ADDRESS :
%s",e.prtaddr);
        printf("\n\n\t\tPHONE : %s",e.phone);
        printf("\n\n\t\tE-MAIL : %s",e.mail);

        printChar('=',65);

}
else printf("\n\n\t\t!!!! ERROR RECORD NOT FOUND !!!!");
```

```
printf("\n\n\t\tWant to enter another search (Y/N)");
fflush(stdin);
another=getchar();
}
}
```

- **Function For Display List:**

Code:

```
void displayList(FILE * fp)
{   printHead();
printf("\n\t\t\t\List  of Employees");
    Employee e;
    int i,siz=sizeof(e);

    rewind(fp);

    while((fread(&e,siz,1,fp))==1)
    {
        printf("\n\n\t\tID : %d",e.id);
        printf("\n\n\t\tNAME : %s",e.name);
        printf("\n\n\t\tDESIGNATION : %s",e.desgn);
        printf("\n\n\t\tGENDER : %s",e.gender);
        printf("\n\n\t\tBRANCH : %s",e.branch);
        printf("\n\n\t\tSALARY : %.2f",e.sal);
        printf("\n\n\t\tPRESENT ADDRESS : %s",e.psaddr);
        printf("\n\n\t\tPERMANANT ADDRESS :
%s",e.prtaddr);
```

```
            printf("\n\n\t\tPHONE : %s",e.phone);
            printf("\n\n\t\tE-MAIL : %s\n\n\t",e.mail);
            printChar('=',65);
        }
        printf("\n\n\n\t");
        printf("\n\n\t");
        system("pause");
}
```

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.

CONCLUSION:

Our project is strongly preferable for a proper management of Employees of an Organization. The security in our project is a very good point to be noted. The project is password

protected, so that authorities can feel secure. We included several user friendly programs, features and operations in details in it. This Project is so user friendly. So, anyone can use this without any complexities. Finally, the system is implemented and tasted according to test cases.