**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

** Belgaum, Karnataka– 575006**

# FILE STRUCTURES LABORATORY WITH MINI PROJECT

(VI SEMESTER)

# MINI PROJECT REPORT

ON

“EMPLYOEE MANAGEMENT”

***Submitted by***

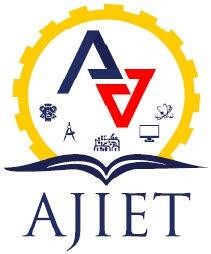
SINCHANA B R 4JK19IS057

SUKANYA N SHET 4JK19IS059

Under the guidance of

**Mr. Arul N Dr. Naganna Chetty**

**(Assistant Professor, Dept. of ISE) ( Professor, Dept. of ISE)**



DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

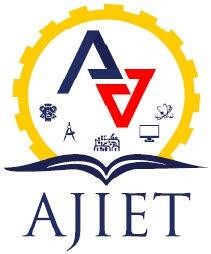
A J INSTITUTE OF ENGINEERING AND TECHNOLOGY

Kottara Chowki, Mangalore -575006, Karnataka.

2021-2022

# **A J INSTITUTE OF ENGINEERING AND TECHNOLOGY**

# **Kottara Chowki, Mangaluru – 575006**

 **DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**CERTIFICATE**

*This is to certify that the Mini project entitled* ***“EMPLOYEE MANAGEMENT”*** *is a bonafide work carried out by*

*SINCHANA B R 4JK19IS037*

*SUKANAYA N SHET 4JK19IS038*

*Students of sixth semester B.E. Information Science & Engineering, and submitted as a part of the course FILE STRUCTURES LABORATORY with Mini Project (18ISL67), during the academic year 2021-2022*.

**Mr Arul N Dr Naganna Chetty Dr Nagesh H R**

**Project Guide Project Guide Head of the Department**

# 

# **Name of the Examiners Signature with Date**

**1.** **1.**

**2. 2.**

**ACKNOWLEDGEMENT**

We dedicate this page to acknowledge and thank those responsible for the shaping of the project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

We sincerely thank the Managementof **A J INSTITUTE OF ENGINEERING AND TECHNOLOGY** a unit of **LAXMI MEMORIAL EDUCATION TRUST** for all their support.

We are extremely thankful to our **Principal, Dr Shantharama Rai C** for his support and encouragement.

We owe our profound gratitude to **Dr Nagesh H R, Head of the Department**, Information Science and Engineering, whose kind consent and guidance helped us to complete this work successfully.

We sincerely thank **Mr Arul N, Assistant Professor**, and **Dr Naganna Chetty, Professor**, Information Science and Engineering, for the guidance and valuable suggestions which helped us to fulfil the experiments prescribed by the university.

We would like to thank all our Information Science and Engineering staff members who have always been with us extending their support, precious suggestions, guidance and encouragement through the project.

We also like to extend thanks to our friends and family members for their continuous support.

TABLE OF CONTENT

**CHAPTER NO. DESCRIPTION PAGE NO**

Acknowledgement 4

Abstract 6

1. **INTRODUCTION**   **7**

1.1 Introduction to File Structure 7

**2 ALGORITHM** **7**

**3 REQUIREMENT SPECIFICATION 11**

2.1 Software Requirements 11

2.3 Hardware Requirements 11

**4 OBJECTIVES OF THE PROJECT 12**

Objective of project 12

Scope and Importance 13

**5 IMPLEMENTATION 14**

Description about Implementation

**6 RESULTS 15**

Pseudo Codes of the modules 15-29

Output 30-34

**7 CONCLUSION 35**

**8 REFERENCES 36**

**ABSTRACT**

Employee Management System Project is essential software designed to keep track of employee information in any company. It stores data such as their employees’ personal information. The goal of the “Employee Management System” is to create a work center scheduling system. Scheduling is a technology that makes the process of informing activities and notifications in the company where it is implemented simply and even online.

The employee management system project gives managers a better idea of their employees and helps them plan and manage their work hours in order to cut costs and boost productivity. It gives appropriate directions and supervision for employees. It also secures and manages information that is important to the employees including personal and work-related information.

**CHAPTER 1**

**INTRODUCTION**

## 1.1 INTRODUCTION TO FILE STRUCTURE

**File structure** is the organization of data in secondary storage device in such a way that it minimizes the access time and the storage space. A file structure is a combination of representations for data in files. It is also a collection of operations for accessing the data. It enables applications to read, write, and modify data. File structure may also help to find the data that matches certain criteria. The main goal of developing file structure is to minimize the number of trips to the disk in order to get desired information. It ideally corresponds to getting what we need in one disk access or getting it with as little disk access as possible.

A good file structure should:

* Fast access to a great capacity.
* Reduce the number of disk accesses
* Manage growth by splitting these collections

***Records and its types:***

They’re the collection of fields, possibly of different data types, typically in fixed number of sequences.

The fields are also called as members. For example, a date could be sorted as a record containing a numeric year field, a Month field represented as a string, and a numeric day of month field. A record is distinguished from arrays by the fact that the number of fields is typically fixed, each field has a name, and that each field may have a different types. A record type is a data type that describes such values and variables. The Definition includes specifying the data type of each field and identifiers by which it can be accessed.

***Why we need File Structure?***

As we know without proper structure of organizing is ordinary system, it will generate some types of problems. So avoiding this kind of problems we go for File Structure.

The File structure means its tell about how the system will stored and access the record from memory and also its tell how the disk are performed its tasks and also how to speed up the execution of transferring data.

Data processing from a computer science prospective:

* + storage of data
  + organization of data
  + access to data

This will be built on your on your knowledge of Data Structures. Some of the important concepts used:

***Field Structures:***

* Fixed Length Fields: The method to organize fields is by limiting the maximum size of each field. The advantage in this method is that since the size of each field is fixed, the entire field can be read at once.
* Length Indicator Fields: The length of each field is specified as a prefix to actual data
* Delimited Fields: Any special character which is not a part of actual data can be used as separator.

***Record Structures:***

* Fixed Length Record Structure: Each record is stored in fixed size. The size can be determined by adding the maximum space occupied by each field and some space reserved for the header data.
* Fixed Field Count: The number of the fields in each record is fixed.
* Index Structure for Records: An index is a collection of key field and reference field.
* Key Field: Key Field is a member of record which can uniquely identify the record.
* Reference Field: Reference Field contains the value that points to the address of the corresponding record in the file.

A successful file structure organizes your data and code with the goal of repeatability, making it easier for you and your collaborators to revisit revise and develop the project. File structures are not structured entities, but rather build a framework that communicates the function and purpose of elements within a project by separating concerns into a hierarchy of folders and using consistent, chronological, and descriptive names. A file structure needs to be predefined format in such a way that an operating system understands, it has an exclusively defined structure, which is based on its type. Three types of files structures in operating systems are: a text file: it’s a series of characters that is organized in lines, an object file: it’s a series of bytes that is organized into blocks, a source file: it’s a series of function and processes.

**CHAPTER 2**

# **REQUIREMENT SPECIFICATION**

### **HARDWARE REQUIREMENTS**

* + - Processor – AMD Ryzen 5
    - RAM - Minimum 4GB
    - Hard Disk - 500GB
    - Mouse - Standard Mouse
    - Processor speed - 2.50GHz

### **SOFTWARE REQUIREMENTS**

* + - Operating System - Windows
    - IDE - Code Blocks
    - Compiler - GCC compiler
    - Language used - C++

**CHAPTER 3**

# **OBJECTIVE OF THE PROJECT**

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

Paperless: To make existing system paperless and save lots of bunching logs of files on the shelf which makes the later on access of the record not at all easy task and overhead to peoples. Automatic: Making the existing system fully automatic which will save lots of human resources work. As the current system is all human resource work is needed to maintain and keep the record and details of every employee under and organization to keep track of every employee in staff working in an organization.

**SCOPE AND IMPORTANCE**

Though the Human Resource Management includes the management of various aspects of human resource, the project will be focused on the development of the Employee details

**CHAPTER 4**

**IMPLEMENTATION**

Implementation is defined as specific set of activities designed to put into practice an activity or program of known dimensions. Implementation processes are purposeful and are described in sufficient details such that independent can detect the presence and strength of the “specific set of activities” related to implementation.

C++ was designed with an orientation toward [systems programming](https://en.wikipedia.org/wiki/Systems_programming) and [embedded](https://en.wikipedia.org/wiki/Embedded_software), resource-constrained software and large systems, with [performance](https://en.wikipedia.org/wiki/Performance_(software)), efficiency, and flexibility of use as its design highlights

C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including [desktop applications](https://en.wikipedia.org/wiki/Application_software), [video games](https://en.wikipedia.org/wiki/Video_game_development), [servers](https://en.wikipedia.org/wiki/Server_(computing)) (e.g. [e-commerce](https://en.wikipedia.org/wiki/E-commerce), [web search](https://en.wikipedia.org/wiki/Web_search_engine), or [databases](https://en.wikipedia.org/wiki/Database)), and performance-critical applications (e.g. [telephone switches](https://en.wikipedia.org/wiki/Telephone_switches) or [space probes](https://en.wikipedia.org/wiki/Space_probes)).

C++ is a general purpose coding language which has been used for game programming, software engineering, data structures, developing browsers, operating systems, applications, and more for over forty years.

C++ is known as a “low-level” programming language that is the language provides little abstraction from the architecture the computer uses, and maintains similar structure to the processors structure. It maintains the simplicity and directness of this “low-level” function.

As such, its performance and memory are efficient, as there isn’t a lot of intermediary between the code written and what it takes for the computer to understand that code.

This means that C++ runs well and efficiently. That’s the part of the reason why it’s still in use, and why that use is expanding and not contracting.

**CHAPTER 5**

**Pseudo Codes of the modules**

#include<iostream>

#include<cstring>

#include<cstdlib>

#include<iomanip>

#include<windows.h>

*//#include <ctime>*

*//#include <dos.h>*

#include<dos.h>

#include<conio.h>

#include<cstdio>

#define max 20

using namespace std;

struct employee

{

 char name[20];

 long int code;

 char designation[20];

 int exp;

 int age;

};

int num;

employee emp[max],tempemp[max],sortemp[max],sortemp1[max];

int main()

{

 system("cls");

 void build();

 void list();

 void insert();

 void deletes();

 void edit();

 void search();

 void sort();

 char option;

 void menu();

 menu();

 while((option=cin.get())!='q')

 {

  switch(option)

  {

   case 'b':

         build();

         break;

   case 'l':

         list();

         break;

   case 'i':

         insert();

         break;

   case 'd':

         deletes();

         break;

   case 'e':

       edit();

        break;

   case 's':

        search();

        break;

   case 'n':

         sort();

         break;

  }

   menu();

  }

  return 0;

 }

 void menu()

 {

  system("cls");

*// highvideo();*

cout<<"          ";

printf("\n\*\*\*\*\*  Employees Management System 1.0 \*\*\*\*\* ");

*//normvideo();*

cout<<endl;

cout<<"             ";

cout<<"\n\t\t Press  b---->Built The Employee Table ";

cout<<"             ";

cout<<"\n\t\t Press  l---->List The Employee Table  ";

cout<<"             ";

cout<<"\n\t\t Press  i---->Insert New Entry        ";

cout<<"             ";

cout<<"\n\t\t Press  d---->Delete An Entry         ";

cout<<"             ";

cout<<"\n\t\t Press  e---->Edit An Entry           ";

cout<<"             ";

cout<<"\n\t\t Press  s---->Search Arecord          ";

cout<<"             ";

cout<<"\n\t\t Press  n---->Sort The Table          ";

cout<<"             ";

cout<<"\n\t\t Press  q---------->Quit Program              ";

cout<<"             ";

cout<<"\n\n \t\t Select Your Option Please ====> ";

}

void build()

{

 system("cls");

*// highvideo();*

 printf("Build The Table");

 cout<<endl;

*//normvideo();*

 cout<<"maximum number of entries  -----  >  20"<<endl;

 cout<<"how many do you want    ----->";

 cin>>num;

 cout<<"Enter The Following Items"<<endl;

 for(int i=0;i<=num-1;i++)

 {

  cout<<" Name  ";

  cin>>emp[i].name;

  cout<<"Code  ";

  cin>>emp[i].code;

  cout<<"Designation  ";

  cin>>emp[i].designation;

  cout<<"Years of Experience  ";

  cin>>emp[i].exp;

  cout<<"Age  ";

  cin>>emp[i].age;

 }

  cout<<"going to main menu";

 Sleep(500);

}

void  list()

{

 system("cls");

*// highvideo();*

 printf("       \*\*\*\*\*\*\*\*List The Table\*\*\*\*\*\*\*\*");

 cout<<endl;

*//normvideo();*

 cout<<"     Name     Code     Designation     Years(EXP)     Age "<<endl;

 cout<<"    ------------------------------------------------------"<<endl;

 for(int i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<emp[i].name;

  cout<<setw(6)<<emp[i].code;

  cout<<setw(15)<<emp[i].designation;

  cout<<setw(10)<<emp[i].exp;

  cout<<setw(15)<<emp[i].age;

  cout<<endl;

 }

  cout<<"going to main menu";

 getch();

  }

  void insert()

  {

  system("cls");

  int i=num;

  num+=1;

*// highvideo();*

  printf("Insert New Record");

  cout<<endl;

*//normvideo();*

  cout<<"Enter The Following Items"<<endl;

  cout<<"Name  ";

  cin>>emp[i].name;

  cout<<"Code  ";

  cin>>emp[i].code;

  cout<<"Designation  ";

  cin>>emp[i].designation;

  cout<<"Years of Experience  ";

  cin>>emp[i].exp;

  cout<<"Age  ";

  cin>>emp[i].age;

  cout<<endl<<endl;

  cout<<"going to main menu";

 Sleep(500);

  }

  void deletes()

  {

   system("cls");

*// highvideo();*

   int code;

   int check;

   printf("Delete An Entry");

*//normvideo();*

   cout<<endl;

   cout<<"Enter An JobCode To Delete That Entry  ";

   cin>>code;

   int i;

   for(i=0;i<=num-1;i++)

   {

    if(emp[i].code==code)

    {

      check=i;

    }

   }

   for(i=0;i<=num-1;i++)

   {

    if(i==check)

    {

    continue;

    }

    else

    {

    if(i>check)

    {

     tempemp[i-1]=emp[i];

    }

    else

    {

     tempemp[i]=emp[i];

    }

     }

   }

  num--;

  for(i=0;i<=num-1;i++)

  {

   emp[i]=tempemp[i];

  }

 }

void edit()

{

 system("cls");

 int jobcode;

*// highvideo();*

 printf("          Edit An Entry           ");

 cout<<endl;

 cout<<endl;

 int i;

 void editmenu();

 void editname(int);

 void editcode(int);

 void editdes(int);

 void editexp(int);

 void editage(int);

 char option;

*//normvideo();*

 cout<<"Enter An jobcode To Edit An Entry----  ";

 cin>>jobcode;

  editmenu();

 for(i=0;i<=num-1;i++)

   {

    if(emp[i].code==jobcode)

    {

while((option=cin.get())!='q')

{

      switch(option)

      {

       case 'n':

            editname(i);

            break;

       case 'c':

            editcode(i);

            break;

       case 'd':

            editdes(i);

            break;

       case 'e':

            editexp(i);

            break;

       case 'a':

           editage(i);

           break;

     }

   editmenu();

    }

  }

  }

  }

  void editmenu()

  {

   system("cls");

   cout<<"        What Do You Want To edit";

   cout<<"          n--------->Name ";

   cout<<"          c--------->Code ";

   cout<<"          d--------->Designation";

   cout<<"          e--------->Experience ";

   cout<<"          a--------->Age        ";

   cout<<"              q----->QUIT                            ";

   cout<<"   Options Please ---->>>  ";

  }

  void editname(int i)

  {

     cout<<"Enter New Name----->  ";

     cin>>emp[i].name;

  }

  void editcode(int i)

  {

   cout<<"Enter New Job Code----->  ";

   cin>>emp[i].code;

  }

  void editdes(int i)

  {

   cout<<"enter new designation----->  ";

   cin>>emp[i].designation;

  }

  void editexp(int i)

  {

   cout<<"Enter new Years of Experience";

   cin>>emp[i].exp;

  }

  void editage(int i)

  {

   cout<<"Enter new Age ";

   cin>>emp[i].age;

  }

void search()

{

 system("cls");

*// highvideo();*

  printf("Welcome To Search Of Employee Database ");

*//normvideo();*

  cout<<endl;

  cout<<endl;

  int jobcode;

  cout<<"You Can Search Only By Jobcode Of An Employee";

  cout<<"Enter Code Of An Employee                    ";

 cin>>jobcode;

 for(int i=0;i<=num-1;i++)

   {

    if(emp[i].code==jobcode)

    {

    cout<<"     Name     Code     Designation     Years(EXP)     Age ";

 cout<<"     ------------------------------------------------------                                  ";

  cout<<setw(13)<<emp[i].name;

  cout<<setw(6)<<emp[i].code;

  cout<<setw(15)<<emp[i].designation;

  cout<<setw(10)<<emp[i].exp;

  cout<<setw(15)<<emp[i].age;

  cout<<endl;

 }

  }

    cout<<"going to main menu";

 getch();

}

void sort()

{

 system("cls");

*// highvideo();*

 printf("Sort The Databse By JobCode");

*//normvideo();*

 void sortmenu();

 void sortname();

 void sortcode();

 void sortdes();

 void sortexp();

 char option;

 void sortage();

 cout<<endl;

 cout<<endl;

 sortmenu();

 while((option=cin.get())!='q')

 {

  switch(option)

  {

   case 'n':

          sortname();

          break;

   case 'c':

          sortcode();

          break;

   case 'd':

          sortdes();

          break;

   case 'e':

          sortexp();

          break;

   case 'a':

          sortage();

          break;

   }

   sortmenu();

  }

 }

 void sortmenu()

 {

    system("cls");

   cout<<"          What Do You Want To edit";

   cout<<"          n--------->Name         ";

   cout<<"          c--------->Code         ";

   cout<<"          d--------->Designation  ";

   cout<<"          e--------->Experience   ";

   cout<<"          a--------->Age          ";

   cout<<"                               q----->QUIT            ";

   cout<<"   Options Please ---->>>  ";  }

void sortname()

{

 system("cls");

 int i,j;

 struct employee temp[max];

 for(i=0;i<=num-1;i++)

 {

  sortemp1[i]=emp[i];

 }

 for(i=0;i<=num-1;i++)

  {

   for(j=0;j<=num-1;j++)

   {

    if(strcmp(sortemp1[i].name,sortemp1[j].name)<=0)

    {

     temp[i]=sortemp1[i];

     sortemp1[i]=sortemp1[j];

     sortemp1[j]=temp[i];

    }

   }

 }

 for( i=0;i<=num-1;i++)

   {

    cout<<"     Name     Code     Designation     Years(EXP)     Age ";

 cout<<"     ------------------------------------------------------                                  ";

 for( i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<sortemp1[i].name;

  cout<<setw(6)<<sortemp1[i].code;

  cout<<setw(15)<<sortemp1[i].designation;

  cout<<setw(10)<<sortemp1[i].exp;

  cout<<setw(15)<<sortemp1[i].age;

  cout<<endl;

 }

  cout<<"Press Any Key To Go Back";

 getch();

} }

void sortcode()

{

 system("cls");

 int i,j;

 struct employee temp[max];

 for(i=0;i<=num-1;i++)

 {

  sortemp1[i]=emp[i];

 }

 for(i=0;i<=num-1;i++)

  {

   for(j=0;j<=num-1;j++)

   {

    if(sortemp1[i].code<sortemp1[j].code)

    {

     temp[i]=sortemp1[i];

     sortemp1[i]=sortemp1[j];

     sortemp1[j]=temp[i];

    }

   }

 }

 for( i=0;i<=num-1;i++)

   {

    cout<<"     Name     Code     Designation     Years(EXP)     Age ";

 cout<<"     ------------------------------------------------------                                  ";

 for( i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<sortemp1[i].name;

  cout<<setw(6)<<sortemp1[i].code;

  cout<<setw(15)<<sortemp1[i].designation;

  cout<<setw(10)<<sortemp1[i].exp;

  cout<<setw(15)<<sortemp1[i].age;

  cout<<endl;

 }

  cout<<"Press Any Key To Go Back";

 getch();

} }

void sortdes()

{

 system("cls");

 int i,j;

 struct employee temp[max];

 for(i=0;i<=num-1;i++)

 {

  sortemp1[i]=emp[i];

 }

 for(i=0;i<=num-1;i++)

  {

   for(j=0;j<=num-1;j++)

   {

    if(strcmp(sortemp1[i].designation,sortemp1[j].designation)<=0)

    {

     temp[i]=sortemp1[i];

     sortemp1[i]=sortemp1[j];

     sortemp1[j]=temp[i];

    }

   }

 }

 for( i=0;i<=num-1;i++)

   {

    cout<<"     Name     Code     Designation     Years(EXP)     Age";

 cout<<"     ------------------------------------------------------                                 ";

 for( i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<sortemp1[i].name;

  cout<<setw(6)<<sortemp1[i].code;

  cout<<setw(15)<<sortemp1[i].designation;

  cout<<setw(10)<<sortemp1[i].exp;

  cout<<setw(15)<<sortemp1[i].age;

  cout<<endl;

 }

  cout<<"Press Any Key To Go Back";

 getch();

} }

void sortage()

{

 system("cls");

 int i,j;

 struct employee temp[max];

 for(i=0;i<=num-1;i++)

 {

  sortemp1[i]=emp[i];

 }

 for(i=0;i<=num-1;i++)

  {

   for(j=0;j<=num-1;j++)

   {

    if(sortemp1[i].age<sortemp1[j].age)

    {

     temp[i]=sortemp1[i];

     sortemp1[i]=sortemp1[j];

     sortemp1[j]=temp[i];

    }

   }

 }

 for( i=0;i<=num-1;i++)

   {

    cout<<"     Name     Code     Designation     Years(EXP)     Age";

 cout<<"     ------------------------------------------------------                                 ";

 for( i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<sortemp1[i].name;

  cout<<setw(6)<<sortemp1[i].code;

  cout<<setw(15)<<sortemp1[i].designation;

  cout<<setw(10)<<sortemp1[i].exp;

  cout<<setw(15)<<sortemp1[i].age;

  cout<<endl;

 }

  cout<<"Press Any Key To Go Back";

 getch();

} }

void sortexp()

{

 system("cls");

 int i,j;

 struct employee temp[max];

 for(i=0;i<=num-1;i++)

 {

  sortemp1[i]=emp[i];

 }

 for(i=0;i<=num-1;i++)

  {

   for(j=0;j<=num-1;j++)

   {

    if(sortemp1[i].exp<sortemp1[j].exp)

    {

     temp[i]=sortemp1[i];

     sortemp1[i]=sortemp1[j];

     sortemp1[j]=temp[i];

    }

   }

 }

 for( i=0;i<=num-1;i++)

   {

    cout<<"     Name     Code     Designation     Years(EXP)     Age ";

 cout<<"  ------------------------------------------------------ ";

 for( i=0;i<=num-1;i++)

 {

  cout<<setw(13)<<sortemp1[i].name;

  cout<<setw(6)<<sortemp1[i].code;

  cout<<setw(15)<<sortemp1[i].designation;

  cout<<setw(10)<<sortemp1[i].exp;

  cout<<setw(15)<<sortemp1[i].age;

  cout<<endl;

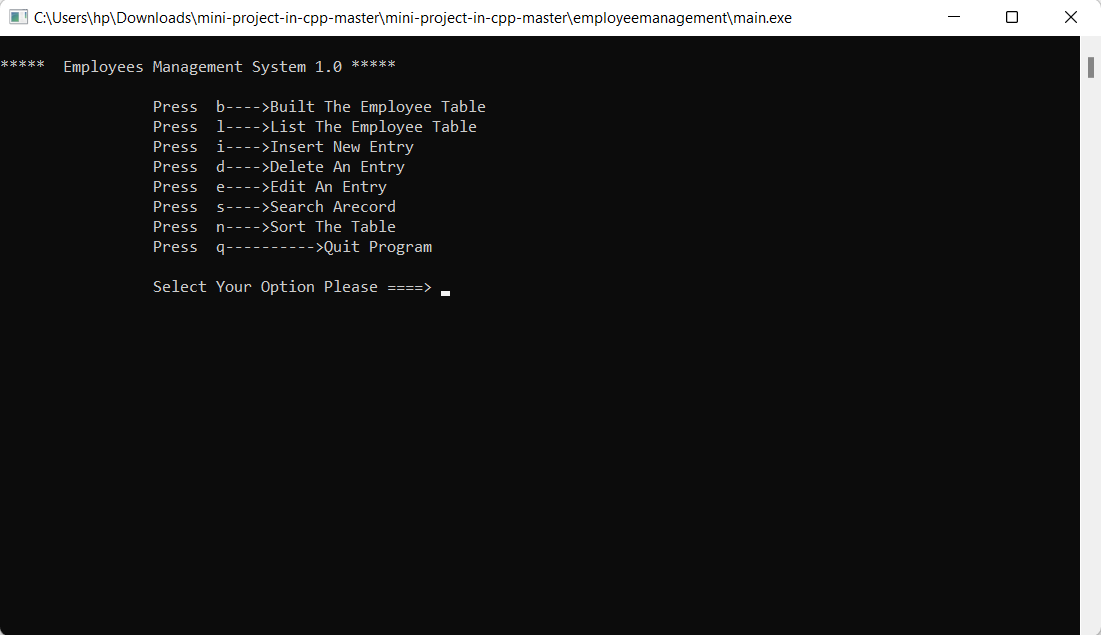
 }

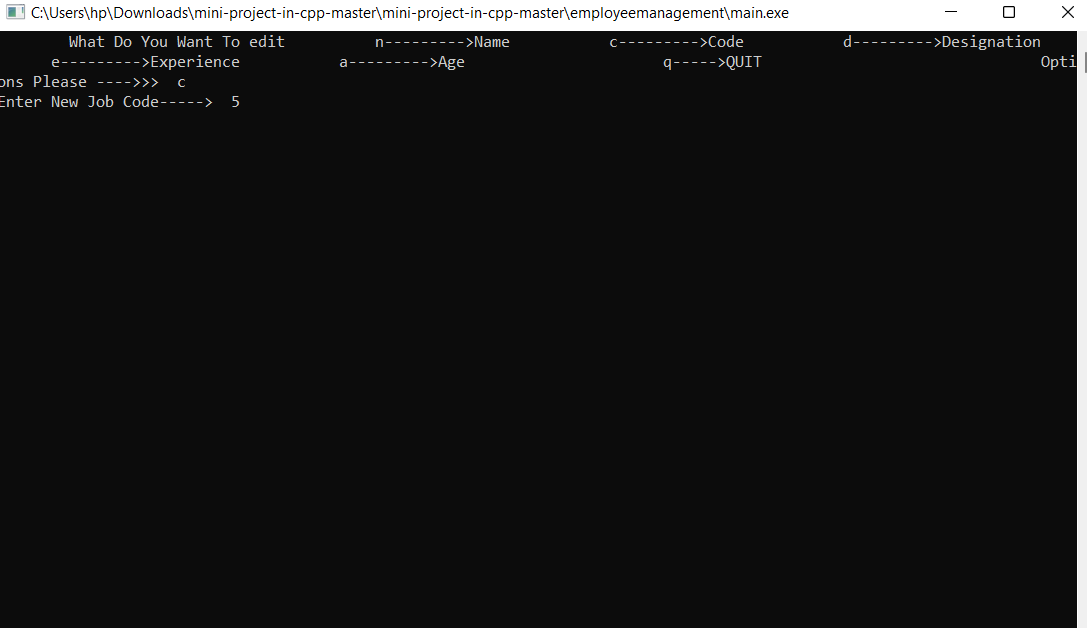
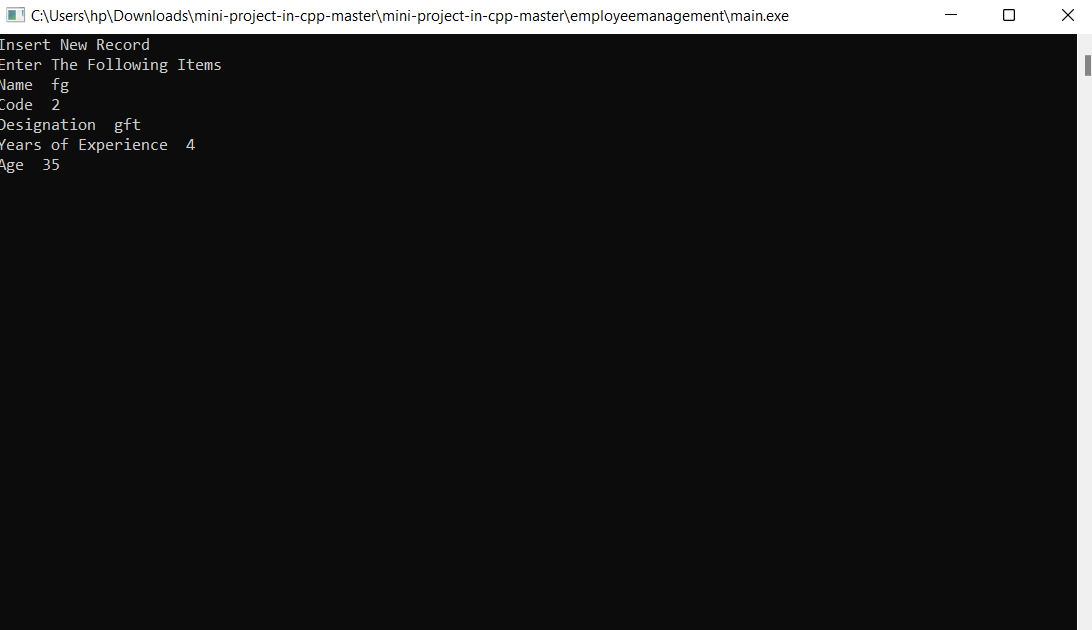
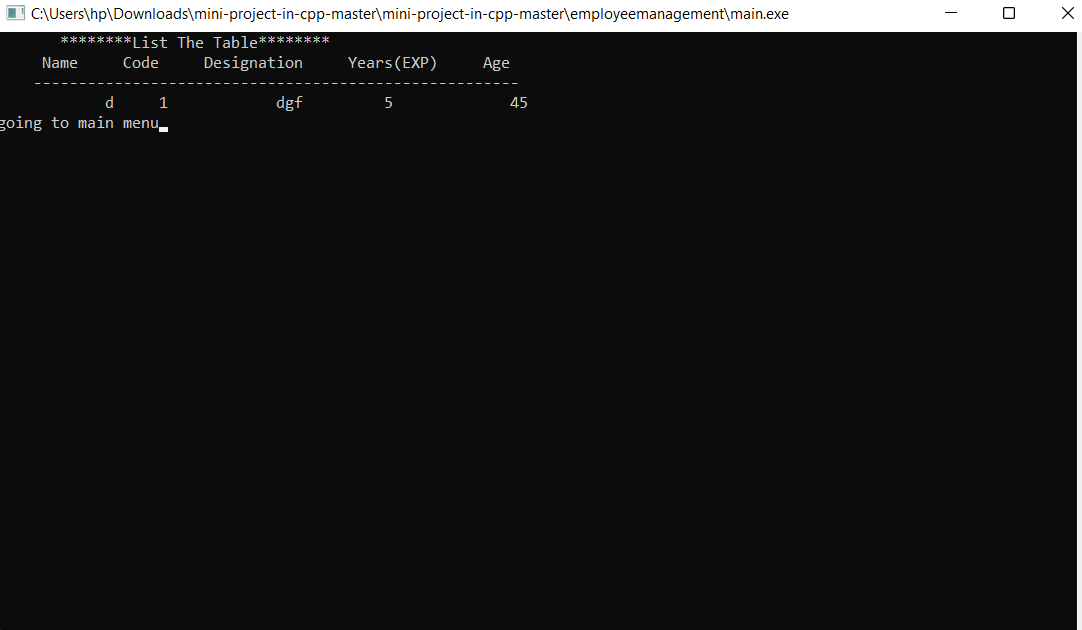
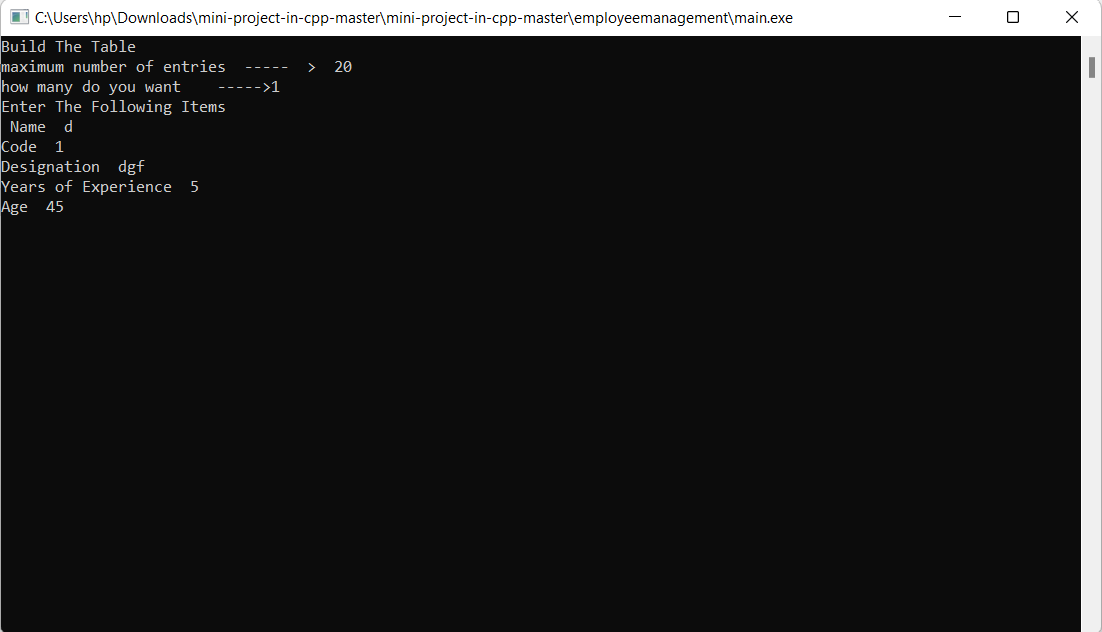
  cout<<"Press Any Key To Go Back";

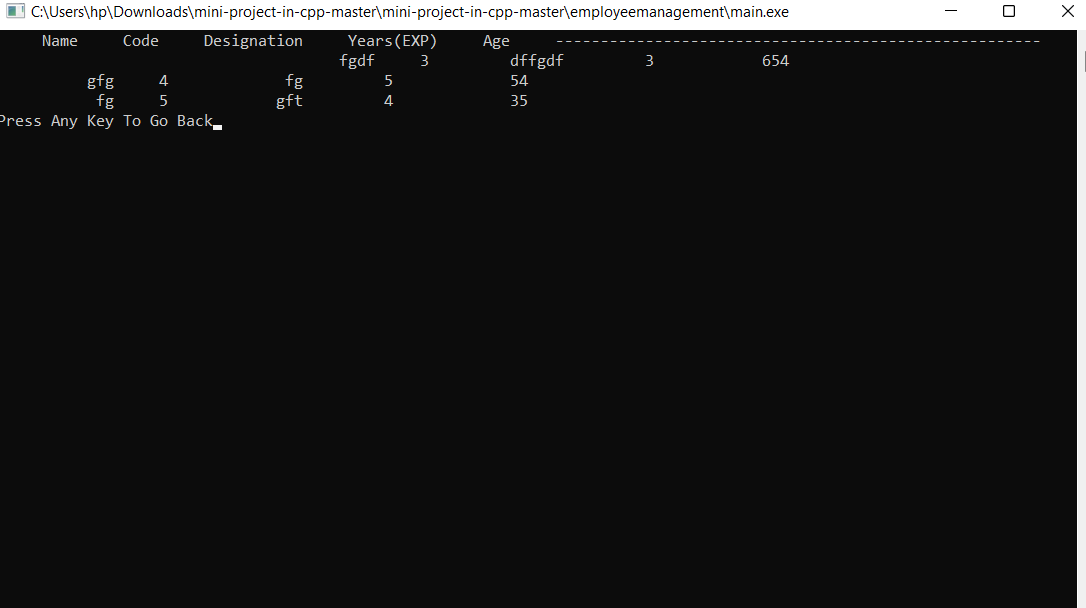
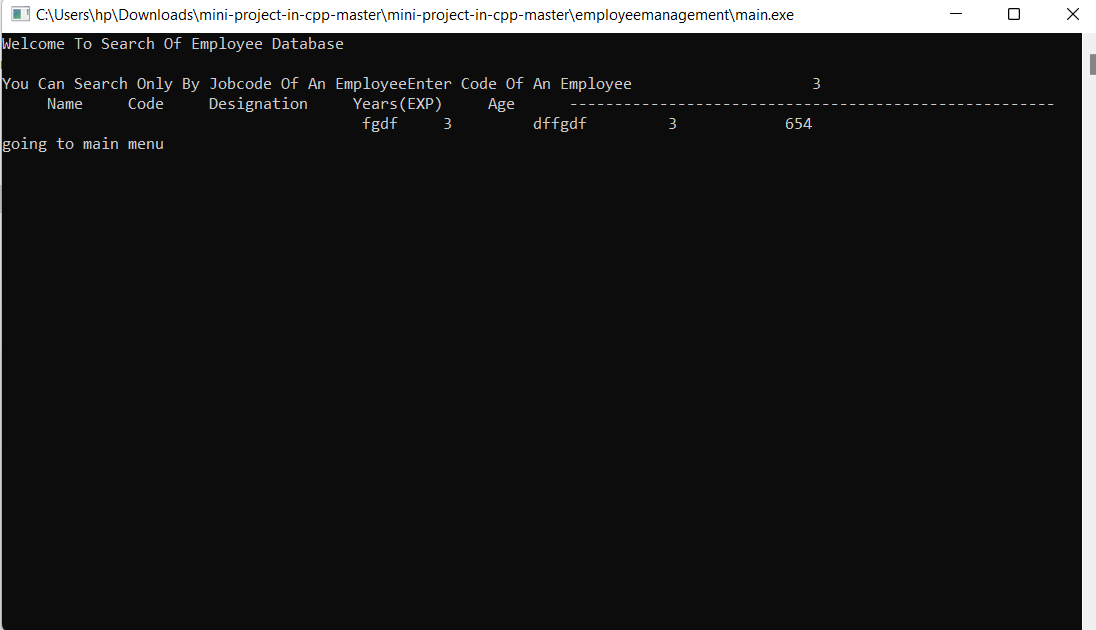
 getch();

} }

**OUTPUT:**

****

****

****

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

CONCLUSION

In this report, an information system's development has been presented. It was emphasized on the basic steps, consequently taken during the project's development course as a particular attention was turned to the basic operative functions performed upon the data into the database.

FUTURE ENHANCEMENT

As a future work, some additional stuff could be implemented and integrated into the application code making it much more reliable and flexible; especially what concerns apay-roll module, for instance.

Apparently, the role of such systems is basic and essential within each company that wants to keep a really good control and record concerning its personnel data, functionality and performance on all levels in its structure. Every organization, in now a days, has the necessity of managing its staff on a really good level as the staff has definitely the greatest merit of building up a company as such as it is. The well managed employee means giving the appropriate financial award-ness and all Hp kind of benefits as such as they have been deserved. That's why the development of such systems is not just a programming business - a lot of people are ordinarily involved insuch projects and one of the basic requirements is the reliability of the

System, especially what concerns the storage of data and all of the operations that will be performed upon it.

**CHAPTER 7**

**REFERNCES:**

* [https://projectsgeek.com](https://projectsgeek.com/)
* [https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)
* <https://youtu.be/wnb4laGrXug>
* <https://youtu.be/Ma0P9T4nTDA>