STORING PEOPLE INFORMATION

A Mini Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

SAKETH – 1602-19-737-161

SRIDHAR – 1602-19-737-177

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, SAKETH and SRIDHAR bearing hall ticket numbers, 1602-19-737-161 and 1602-19-737-177, hereby declare that the project report entitled “STORING PEOPLE INFORMATION” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

SAKETH

1602-19-737-161

SRIDHAR

1602-19-737-177

(Faculty In-Charge) (Head.Dept IT)

Acknowledgements

The satisfaction that accompanies the successful completion of this project would not be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of this project.

We convey thanks to my project guide Mrs. Divya of Information technology Department for providing encouragement, constant support and guidance which was of a great help to complete this project successfully.

Last but not the least, we wish to thank our parents for financing our studies in this privileged Vasavi College of Engineering as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged

STORING PEOPLE INFORMATION :-

TEAM – 23

SAKETH – 161

SRIDHAR- 177

ABSTRACT

Our project is “ STORING PEOPLE INFORMATION SYSTEM” is based on the concept to store people census data. System that contains information about a group of n people. Information is entered,

a person being described by name , age , city etc..

this system makes easy to store records of each. The whole project is designed by different variables and strings have been used for the development of this project.

This main project is used to operate and understand by the user.

Data storage methods play a critical role in a company’s data .People give information about them and we store census data about people.

Table of contents

TOPIC PAGE NUMBER

INTRODUCTION

TECHNOLOGY

PROPOSED WORK

RESULTS

ADDITIONAL KNOWLEDGE

CONCLUSION

REFERENCE

INTRODUCTION

Storing people information with source code and output . Project is done using file handling, functions ,arrays and for loop.

Our project is “ STORING PEOPLE INFORMATION SYSTEM” is based on the concept to store people census data. System that contains information about a group of n people. Information is entered,

a person being described by name , age , city etc..

this system makes easy to store records of each. The whole project is designed by different variables and strings have been used for the development of this project. This main project is used to operate and understand by the user.

Data storage methods play a critical role in a company’s data .People give information about them and we store census data about people.

CLIENT SIDE:

1. FILE CREATION

2. DISPLAYING CONTENTS OF A FILE

3. ADDING INFORMATION

4. AGE AVERAGE CALCULATION

5. PERSON REMOVAL

Firstly, in user side, user gives inputs like name, age .then the file is created. the content of the file will be displayed by server . User adds details, if file exists the details are stored else the new file is created. User enters age and server calculates average age and displayed on it.

SERVER SIDE:

1. UPDATE CONTENT

2. SORT FILES

3. CORRECTNESS INFORMATION ANALYSIS

4. CALCULATE AVERAGE

5. DISPLAYS CONTENT

Server checks the correctness ,it checks the correctly entered or not. If no the data will be replaced and corrected.

SOFTWARE REQUIREMENTS

• OS – Windows 7 and above

• Editor – Any basic editor is preferable

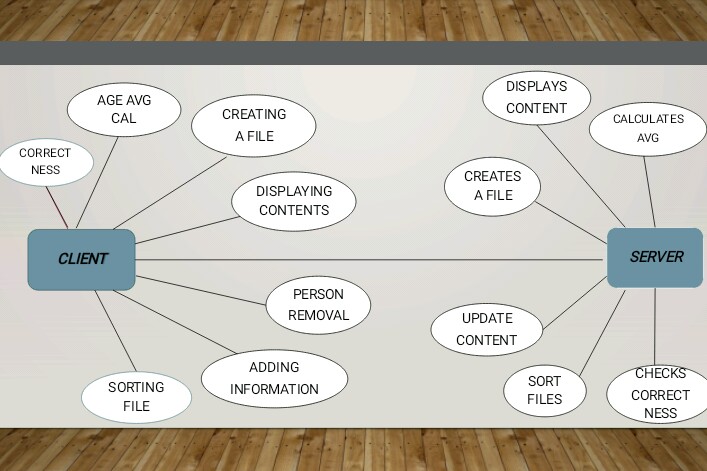
• C Compiler - Linux

b. HARDWARE REQUIREMENTS

• RAM – 512 MB

• INPUT DEVICES – Keyboard

• OUTPUT DEVICES – Monitor Software



#include <stdio.h>

#include <stdlib.h>

#include<string.h>

char fileName[20];

int createFile(){

FILE \* fptr = NULL;

int mode = 2;

printf("\n Enter name of file : ");

scanf(" %s", fileName);

fptr = fopen(fileName, "w");

if(fptr!=NULL)

fclose(fptr);

return 0;

}

void displayFileContents(){

FILE \*fptr = fopen(fileName, "r");

char c;

long int count = 0;

printf("\n---- File Content ---- \n\n");

while ((c = getc(fptr)) != EOF){

putchar(c); count++;

}

if(count==0)

printf("\n Warning: File is empty!!");

else

fclose(fptr);

}

void addInfo(){

char name[50];

unsigned char age;

getchar();

printf("\n\n Enter Name : ");

gets(name);

FILE\* fptr = fopen(fileName, "r");

char line[500];

char alreadyExist = 0;

while (fgets(line, sizeof(line), fptr)) {

if(strstr(line, name))

alreadyExist = 1;

}

fclose(fptr);

if(alreadyExist==1)

{

printf("\n Record already exist !!");

printf(" %s ", line);

}

else{

fptr = fopen(fileName, "a");

fputs(name, fptr);

printf("\n Enter Age : ");

scanf("%d",&age);

char buffage[5];

char temp[5];

strcpy(temp, " | ");

itoa(age, buffage, 10);

strcat(temp, buffage);

strcat(temp, "\n");

fputs(temp, fptr);

fclose(fptr);

}

}

void calcualteAvgAge(){

FILE\* fptr = fopen(fileName, "r");

char line[500];

char \*ptr = NULL;

float avg = 0;

char alreadyExist = 0;

int count =0;

while (fgets(line, sizeof(line), fptr)) {

ptr = strtok(line, "|");

ptr = strtok(NULL, "|");

avg += atoi(ptr);

count++;

}

if(count>0){

avg = avg/count;

printf("\nAverage Age is : %.2f",avg) ;

}

else{

printf("\n No data to calcualte avg age.");

}

}

void sortFileContents(){

FILE\* fptr = fopen(fileName, "r");

char line[500];

float tempArr[500];

char \*ptr = NULL;

int i=0, count=0;

while (fgets(line, sizeof(line), fptr)) {

count++;

ptr = strtok(line, "|");

ptr = strtok(NULL, "|");

tempArr[i++] = atoi(ptr);

}

int order=0;

printf("\n\n Press 1 for Assending order ");

printf("\n Press 2 for Desending order : ");

scanf("%d",&order);

float temp =0;

int j;

for(i=0; i<count; i++){

for(j=i+1; j<count;j++){

if(order==2){

if(tempArr[j] > tempArr[i])

{ temp=tempArr[j];

tempArr[j] = tempArr[i];

tempArr[i] = temp;

}

}

else{

if(tempArr[i] > tempArr[j])

{

temp=tempArr[j];

tempArr[j] = tempArr[i];

tempArr[i] = temp;

}

}

}

}

for(i=0; i<count; i++){

rewind(fptr);

while (fgets(line, sizeof(line), fptr)) {

float temp=0;

char tempLine[500];

memset(tempLine, 0, sizeof(tempLine));

strcpy(tempLine, line);

ptr = strtok(line, "|");

ptr = strtok(NULL, "|");

temp = atoi(ptr);

if(temp == tempArr[i]){

printf("\n %s", tempLine);

break;

}

}

}

}

void removePerson(char \*nameptr){

printf("\n --- Remove Person from Records ---");

char name[50];

memset(name, 0, sizeof(name));

if(nameptr==NULL){

printf("\n\n Enter Name : ");

getchar(); gets(name);

}

else

strcpy(name, nameptr);

FILE\* fptr = fopen(fileName, "r");

if(fptr==NULL){

printf("\n Error: No record exist");

return;

}

char temp[]="temp";

FILE\* fptrTmp = fopen(temp, "w");

if(fptrTmp==NULL){

printf("\n Error: Unable to perform this operation");

return;

}

char line[500];

int count =0;

while (fgets(line, sizeof(line), fptr)) {

if(strstr(line, name)){

count++;

printf("\n---debug: name matched line is : %s",line);

memset(line, 0, sizeof(line));

continue;

}

else{

fputs(line, fptrTmp);

printf("\n putting : %s in new file", line);

}

memset(line, 0, sizeof(line));

}

fclose(fptrTmp);

fclose(fptr);

if(count>0){

printf("fileName : %s removed ", fileName);

if(remove(fileName) == 0)

printf("\n File %s is deleted successfully", fileName);

else

printf("\n Error while deleting file %s ", fileName);

getchar(); getchar();

rename(temp, fileName);

printf("\n Success ! Reocrd found & deleted");

}

else{

printf("\n Error: Record not found");

}

remove(temp);

}

void correctInfo(){

FILE\* fptr = fopen(fileName, "r");

if(fptr==NULL){

printf("\n Error: No record exist");

return;

}

char temp[]="temp1";

char line[500];

FILE\* fptrTmp = fopen(temp, "w");

while (fgets(line, sizeof(line), fptr)) {

fputs(line, fptrTmp);

memset(line, 0, sizeof(line));

}

fclose(fptrTmp);

fclose(fptr);

fptrTmp = fopen(temp, "r");

float tempArr[500];

char \*ptr = NULL;

char name[50];

memset(name, 0, sizeof(name));

int i=0, count=0;

char choice='\0';

memset(line, 0, sizeof(line));

while (fgets(line, sizeof(line), fptr)) {

count++;

printf("\n %s ", line);

printf("\n Do u want to update this record? y/n :");

scanf(" %c",&choice);

if(choice=='y'){

ptr = strtok(line, "|");

strcpy(name, ptr);

removePerson(name);

printf("\n\n Please enter new values ");

addInfo();

break;

}

}

//close(fptr);

}

void menu(){

char choice = -1;

while(choice!=8){

printf("\n\n\n #### Main Menu ####\n\n");

printf("\n 1. File creation");

printf("\n 2. Display the contents of the file");

printf("\n 3. Add Info to file");

printf("\n 4. Age average calcualtion");

printf("\n 5. Sort file contents");

printf("\n 6. Person removal");

printf("\n 7. Correctness information analysis");

printf("\n 8. End program");

printf("\n\n Enter your choice : ");

scanf("%d", &choice);

switch(choice){

case 1:

if(createFile() == 0)

printf("\n Success : File created successfully!");

else

printf("\n Error: Can't create file'");

break;

case 2:

displayFileContents();

break;

case 3:

addInfo();

break;

case 4:

calcualteAvgAge();

break;

case 5:

sortFileContents();

break;

case 6:

removePerson(NULL);

break;

case 7:

correctInfo();

break;

case 8: printf("\n thanks for using the application \n ");

default: printf("\n Error: Invalid Choice! Enter again");

getchar();

}

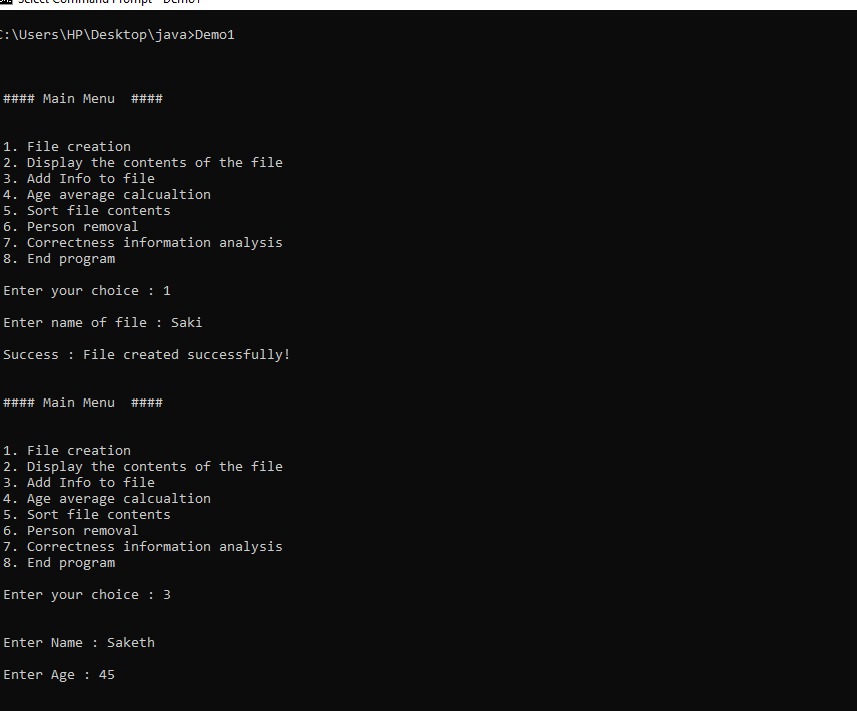
}

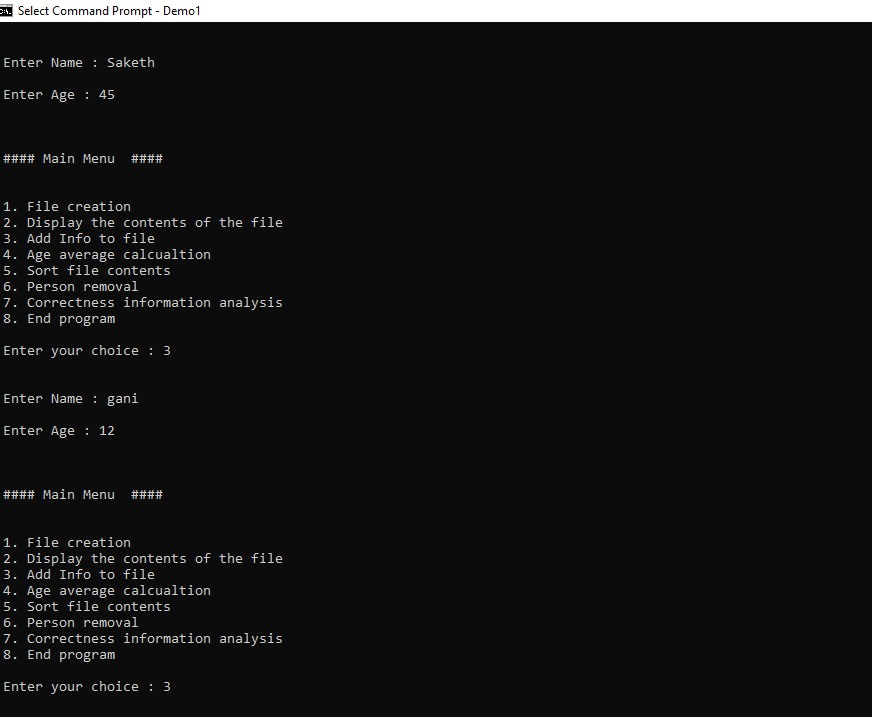
}

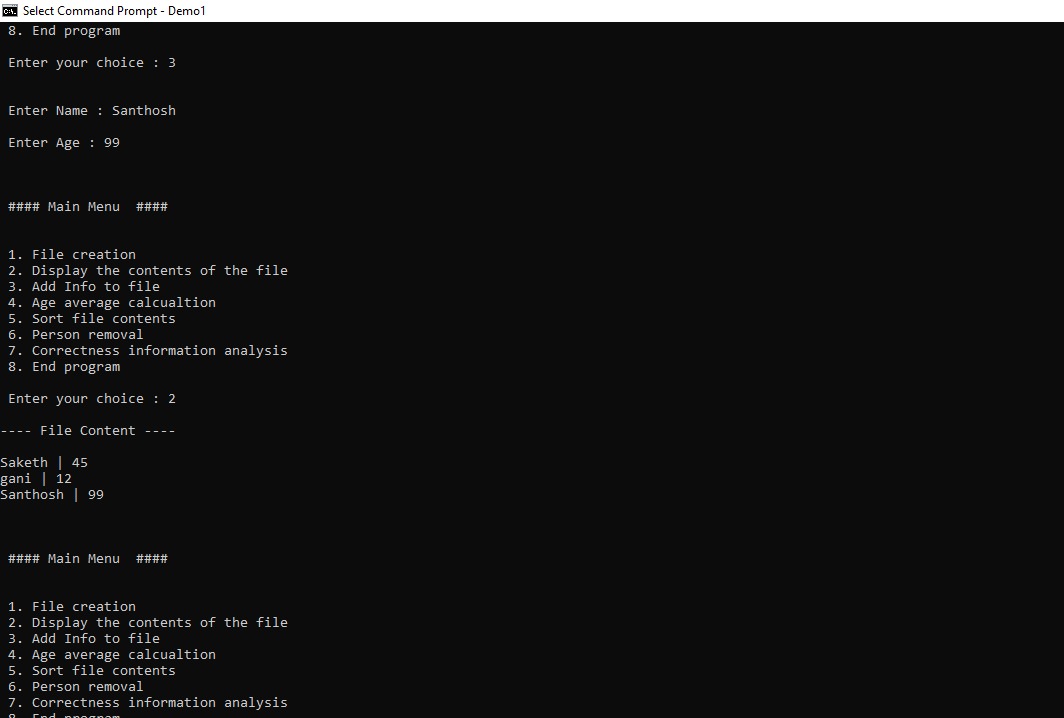
int main(int argc, char \*argv[]) {

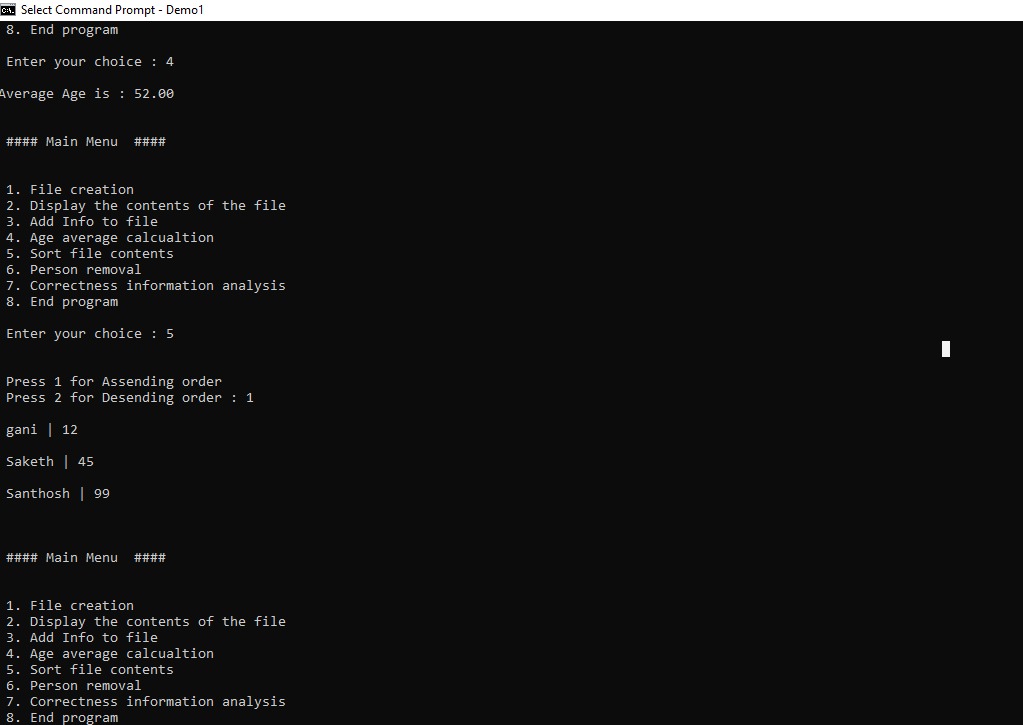
menu();

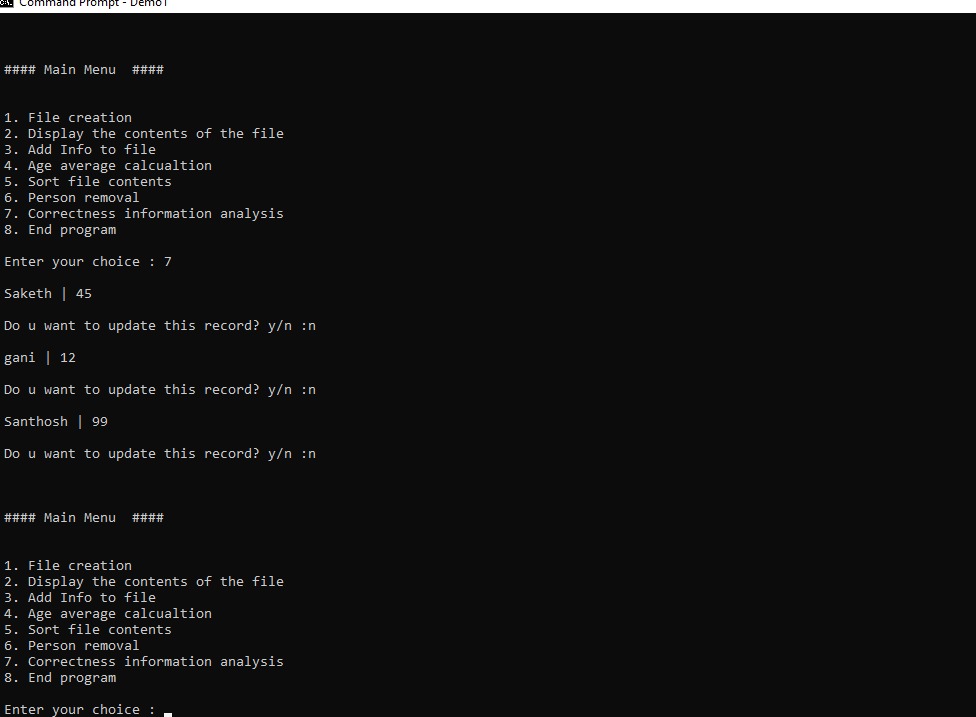
return 0;

} 









|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC01 | | | Use case ID :  **FC-01** |
| **Test case Title :** Tries to create a new file | | |
| **Test case description** : Tries to create a new file. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The system is going to create a new file. | New file will be created. | New file will be created. | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC02 | | | Use case ID :  **FC-02** |
| **Test case title** : Enters the details in new file. | | |
| **Test case description** : Details will be stored in new file. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| Details of the people will be going to storeed in new file | The information of people is going to stored in the file. | The information of people is going to stored in the file. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC03 | | | Use case ID :  **FC-03** |
| **Test case title** : system displays the information stored in files. | | |
| **Test case description** : The system displays the information. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| Admin asks to display the file. | The system displays the information stored in the file. | The system displays the information stored in the file. | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC04 | | | Use case ID :  **FC-04** |
| **Test case title** : Going to calculate the average age of people. | | |
| **Test case description** :It will calculate the average age. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| Admins asks to calculate the average age. | The system should display the average age of people in the file. | The system should display the average age of people in the file. | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC05 | | | Use case ID :  **FC-05** |
| **Test case title** :It will sort the files | | |
| **Test case description** : It will sort the files. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| It will going to sort the names in files by age either in ascending order or descending order. | The system should display the names by age,like either in ascending order or in descending order. | The system displays the names in either in ascending order or in descending order by their age. | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC06 | | | Use case ID :  **FC-06** |
| **Test case title** : It will the edit the file. | | |
| **Test case description** : It is going to edit the file. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| It is going to edit the file if we want to make any changes to our file. | The system displays the file after editing . | The system displays the file after editing . | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Test case ID** : TC07 | | | Use case ID :  **FC-07** |
| **Test case title** : It will asks to remove any details from the file. | | |
| **Test case description** : It will asks to remove any details from the file. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| It is going to remove any details that we don’t want to have in our file. | The system displays the new file with details that haven’t removed from old file. | The system displays the new file with details that haven’t removed from old file. | |

**What is the additional knowledge gained as a result of implementing this miniproject apart from the syllabus covered in the course programming for problem solving?**

This project really helped us to know the various real world application of c programming and know the essence of this language . These project made me to build my programming skills and I learnt many applications in C language.

CONCLUSION :-

I hope this C mini project on storing people information will serve you this project is an undertaking by one or more people to develop and create people census data

We have covered technologies for storing data,assessing data, processing data and publishing data on the web. we have looked at the advantages and disadvantages of different storage options ,we have looked at how to design data storage efficiently