BODY FACTORY

AUTOMATED WORKOUT / FITNESS PLAN

PROJECT REPORT

GROUP MEMBERS:

❖ ZEESHAN	(20K-0361)
❖ ANAS ALI	(20K-0181)
❖ MUHAMMAD UMER	(20K-0225)

CONTENTS

BREIF DESCRIPTION

What is our project all about?

Our project is an automated workout/fitness plan which is made with C++ language. It utilizes all the rules and abilities of Objected Oriented Language. It is solely a consoled based application.

How does it works?

A user is asked to input name, age, gender, weight, height, contact number. Our program will calculate BMI. Then user will be asked about the level of fitness/workout plan he wants. After his selection of any of the option, he will be provided with complete diet, fitness and workout plan of whole specific period. Not only this, but user will also be provided with the video clip of each exercise, which will help him/her exercise efficiently.

Why we chose it?

In this rapidly developing world, we see our life becoming easier day by day. Just like that, fitness or workout is an essential part of our life. Every person wishes to have a perfect body, but complexity in finding an efficient plan has made its reach out of most of the people. To end this problem, we are bringing our project with an aim to provide an easy, efficient way to have your required fitness plan in your hands just within few clicks. People, also after having a plan, stuck in confusion that how to perform a specific exercise. Our attempts to provide a video clip for each exercise will be surely a great assistant for them. They won’t have to pay thousands to have a trainer.

Libraries

6 pre-defined libraries

- #include<iostream>
- #include<conio.h>
- #include<fstream>
- #include<string>
- #include <windows.h>
- #include<mmsystem.h>

1 user-defined library

- #include"project_required.h"
  - ➢ 6 functions were defined in the library

Concepts used:

Encapsulation:

Most of the functions were defined in separate .h files which provides the perfect encapsulation and security.
Data members were encapsulated with the help of access specifiers.

Inheritance

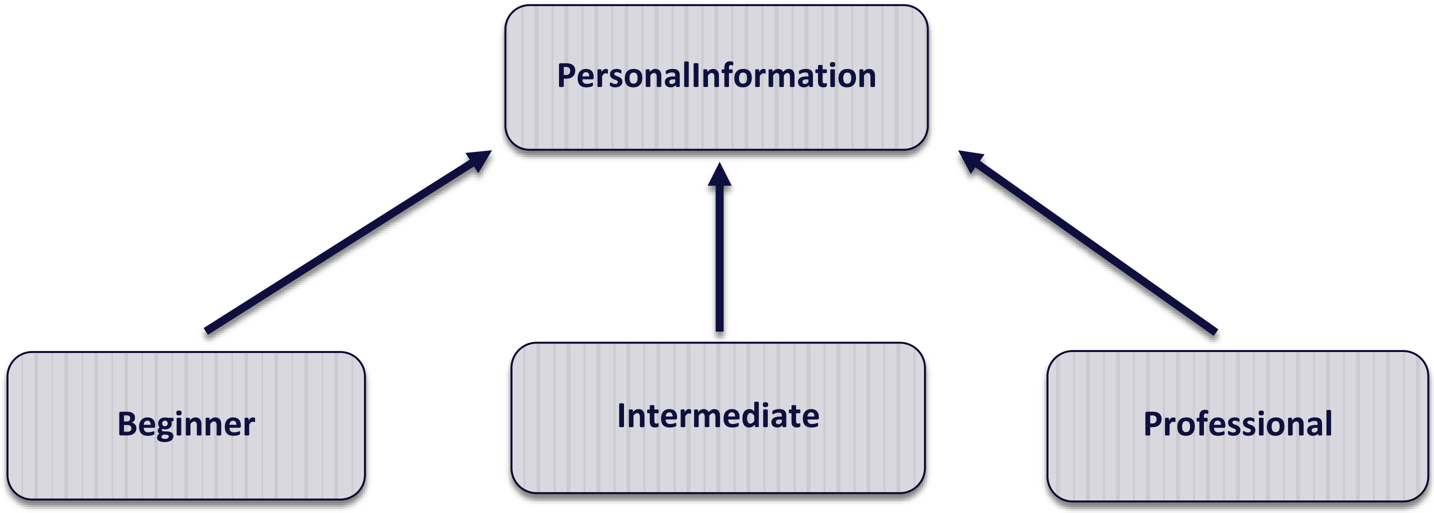
Hierarchal Inheritance is used to implement the concept of inheritance

PARENT CLASS

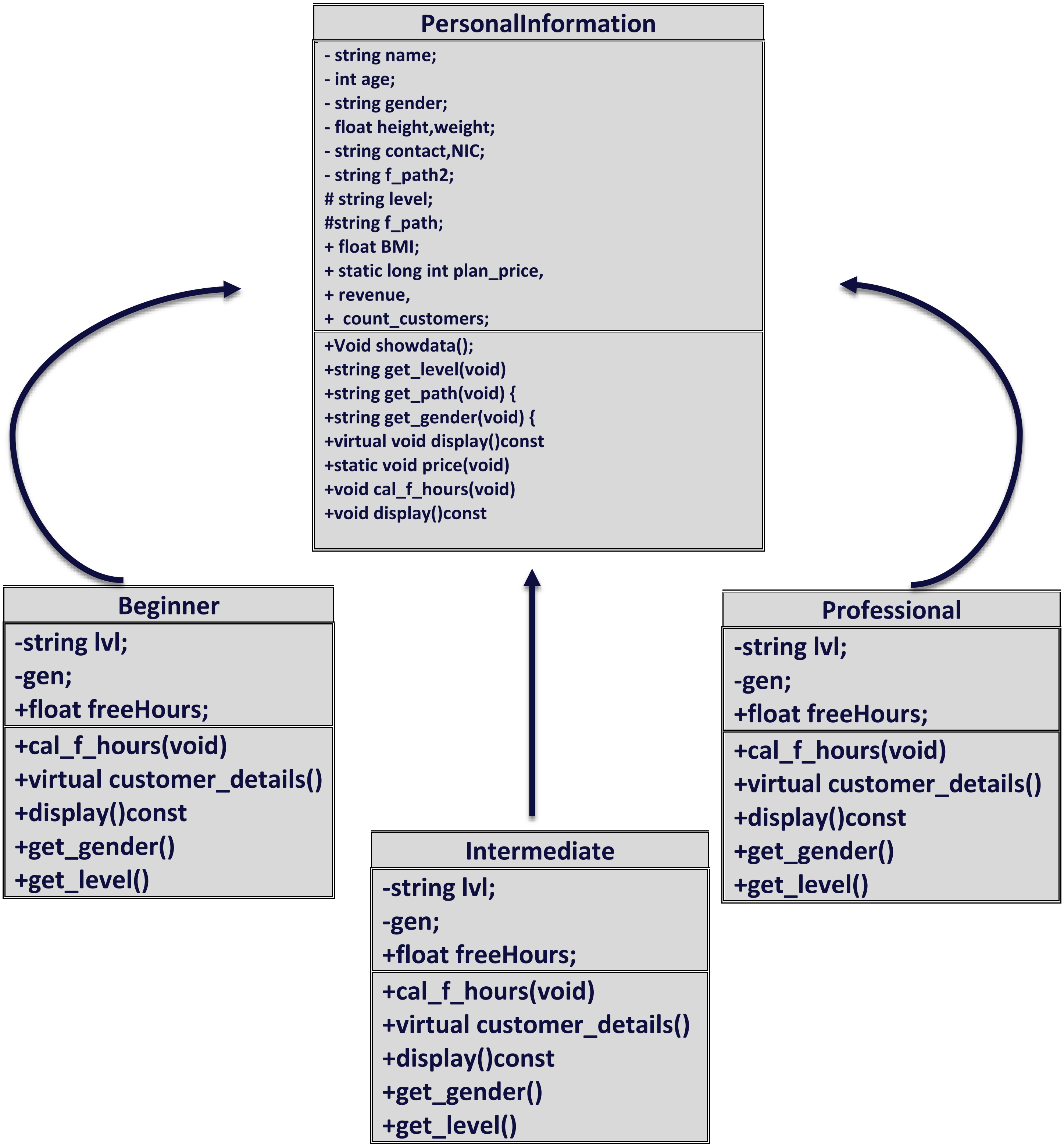
- PersonalInformation

CHILD CLASS

- Beginner
- Intermediate
- Professional



Class Diagram



PersonalInformation

- string name;
- int age;
- string gender;
- float height,weight;
- string contact,NIC;
- string f_path2;

# string level;
\#string f_path;
+ float BMI;
+ static long int plan_price,
+ revenue,
+ count_customers;

+Void showdata();
+string get_level(void)
+string get_path(void) {
+string get_gender(void) {
+virtual void display()const
+static void price(void)
+void cal_f_hours(void)
+void display()const

Beginner

-string lvl;
-gen;
+float freeHours;

+cal_f_hours(void)
+virtual customer_details()
+display()const
+get_gender()
+get_level()

Intermediate

-string lvl;
-gen;
+float freeHours;

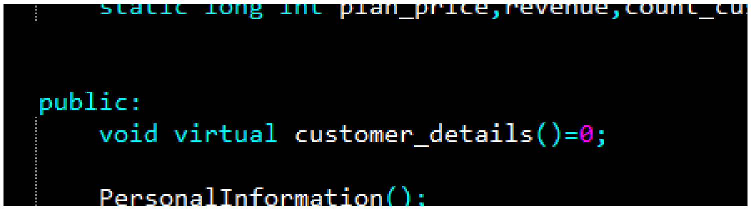
+cal_f_hours(void)
+virtual customer_details()
+display()const
+get_gender()
+get_level()

Professional

-string lvl;
-gen;
+float freeHours;

+cal_f_hours(void)
+virtual customer_details()
+display()const
+get_gender()
+get_level()

Abstract Class



```
        static long int plan_price,revenue,count_cu


    public:
        void virtual customer_details()=0;

        PersonalInformation();
```

Abstract class was used so that function declared in parent class should not be used, instead it should only be used in child classes, by equating the function to zero, in parent class.

Virtual Function

To implement the use of abstract class, virtual function was brought up.

Constructors

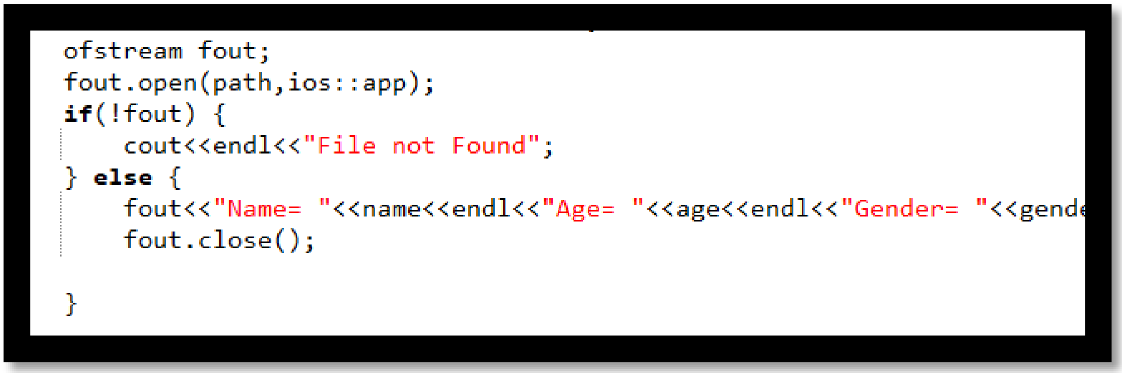
To initialize the variables in the class, parameterized constructors were used.

Polymorphism/ Function Overloading

To reduce the complexity of the code, we used the concept of polymorphism in the classes, by keeping the functions with same name but different arguments.

Filing

The concept of filing was implemented to store every single data of customer in our system, to keep a record.



```
ofstream fout;
fout.open(path,ios::app);
if(!fout) {
    cout<<endl<<"File not Found";
} else {
    fout<<"Name= "<<name<<endl<<"Age= "<<age<<endl<<"Gender= "<<gende
    fout.close();

}
```

Const

To keep the data safe and secure, we used the concept of const keyword. With its help we were able to keep the customer details secure, so that no one can change the data of customer, once entered.

Static

To keep some of the variables unchanged.

Abstraction

Abstraction was used to provide only essential details to the user and hide all the internal details.

Data members:

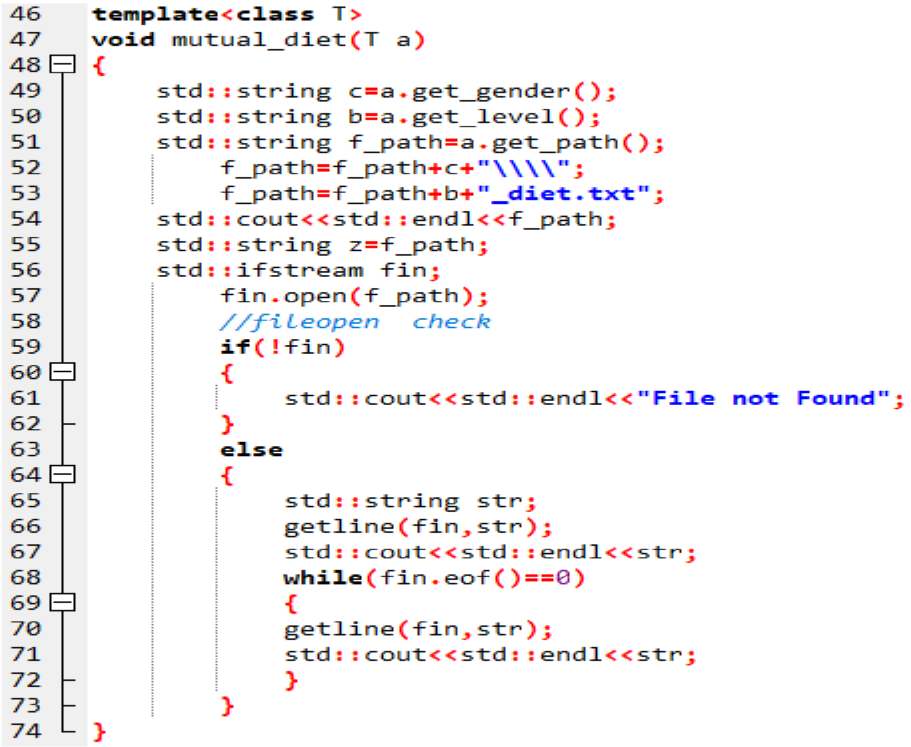
- Private:
  - string name;
  - int age;
  - string gender;
  - float height,weight;
  - string contact,NIC;
  - string f_path2;

- Protected:
  - string level;
  - string f_path;
- Public:
  - float BMI;
  - static long int plan_price,
  - revenue,
  - count_customers;

Member Functions:

- Void showdata();
- string get_level(void)
- string get_path(void) {
- string get_gender(void) {
- virtual void display()const
- static void price(void)
- void cal_f_hours(void)
- void display()const

TEMPLATE mutual_diet() :

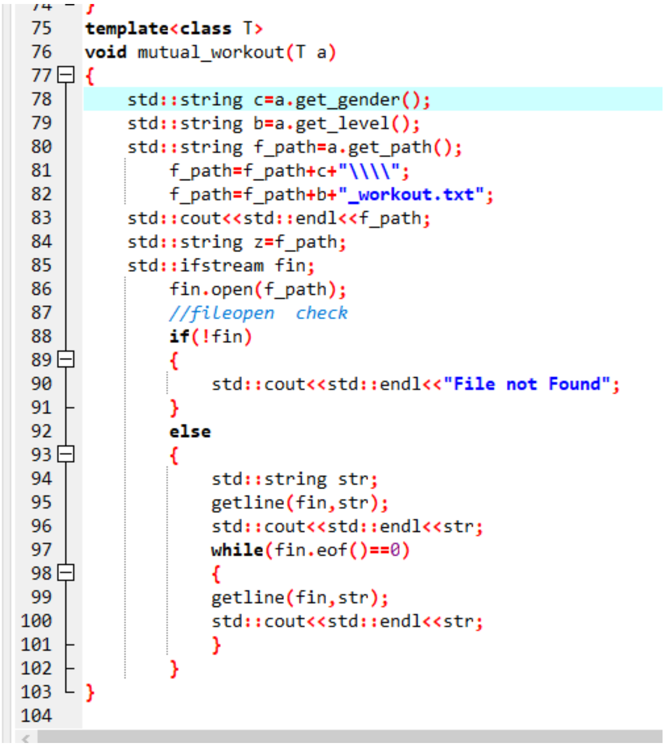


```
template<class T>
void mutual_diet(T a)
{
    std::string c=a.get_gender();
    std::string b=a.get_level();
    std::string f_path=a.get_path();
        f_path=f_path+c+"\\\\";
        f_path=f_path+b+"_diet.txt";
    std::cout<<std::endl<<f_path;
    std::string z=f_path;
    std::ifstream fin;
        fin.open(f_path);
        //fileopen  check
        if(!fin)
        {
            std::cout<<std::endl<<"File not Found";
        }
        else
        {
            std::string str;
            getline(fin,str);
            std::cout<<std::endl<<str;
            while(fin.eof()==0)
            {
            getline(fin,str);
            std::cout<<std::endl<<str;
            }
        }
}
```

In this function we are Just opening the file in which we have written the diet plan. This function was required in all classes to open the files according to the user's level and gender. So this is why we used template function. To get the path of a folder in which our diet plan file is saved ,We used a.get_level() and a.get_gender() fuction that will take the level and gender from an object out of (beginner,intermediate,professional) classes. We are concadenating string (string +string) and then finally, they join with f_path (i.e ,the path of a file in our .cpp file directory). By this f_path string we are opening our file.

TEMPLATE mutual_workout():

In this function we are Just opening the file in which we have written the diet plan. This function was required in all classes to open the files according to the user's level and gender. So this is why we used template function. To get the path of a folder in which our diet plan file is saved ,We used a.get_level() and a.get_gender() fuction that will take the level and gender from an object out of (beginner,intermediate,professional) classes. We are concadenating string (string +string) and then finally, they join with f_path (i.e ,the path of a file in our .cpp file directory). By this f_path string we are opening our file.



```
}
template<class T>
void mutual_workout(T a)
{
    std::string c=a.get_gender();
    std::string b=a.get_level();
    std::string f_path=a.get_path();
        f_path=f_path+c+"\\\\";
        f_path=f_path+b+"_workout.txt";
    std::cout<<std::endl<<f_path;
    std::string z=f_path;
    std::ifstream fin;
        fin.open(f_path);
        //fileopen  check
        if(!fin)
        {
            std::cout<<std::endl<<"File not Found";
        }
        else
        {
            std::string str;
            getline(fin,str);
            std::cout<<std::endl<<str;
            while(fin.eof()==0)
            {
            getline(fin,str);
            std::cout<<std::endl<<str;
            }
        }
}
```

User-defined Functions Used:

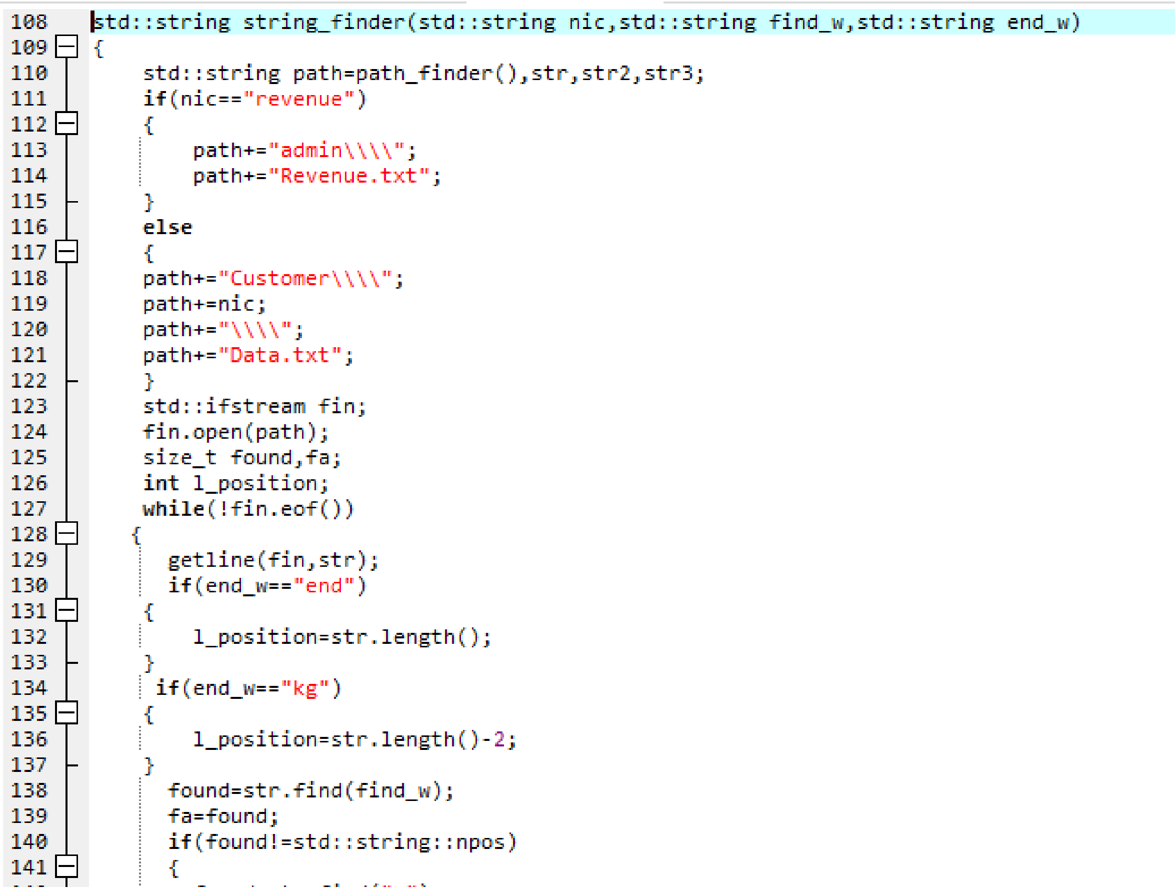
1. Substr function():

This is the built-in function of string in C++. It takes 2 arguments (1st position, Last position) and it saves the string occurs between these position in a separate variable. We used this buil in function in our string_finder() function that is discussed below.

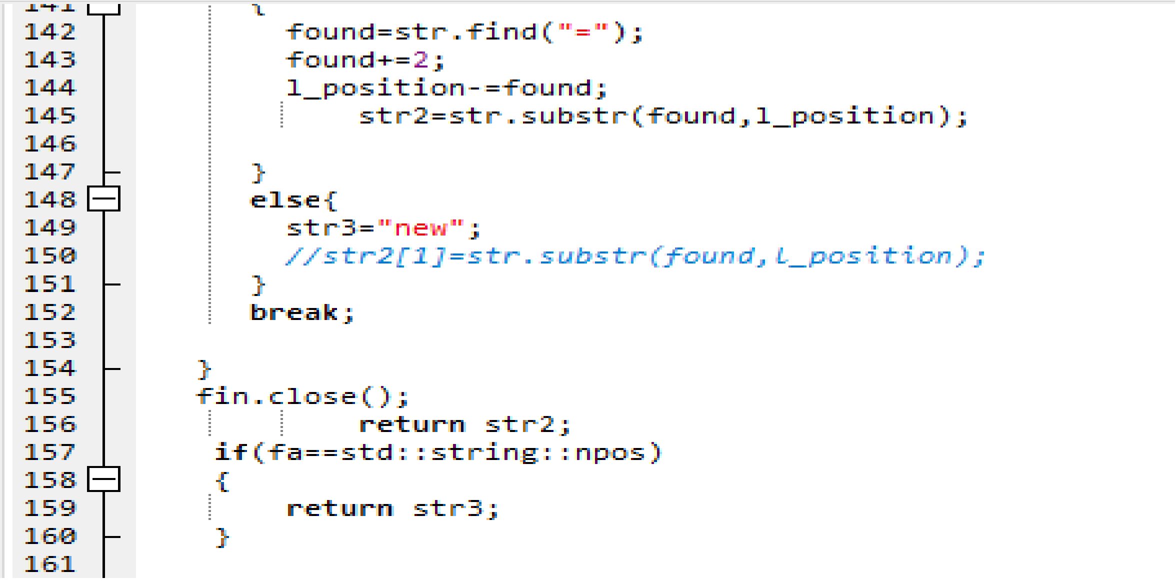
2. String_finder():

In this function We are finding a selective data from our string. We are taking customers name, gender and previous weight from a file . The function takes the NIC of a user, a word to be searched, and end word. From the NIC, the function is opening a customers file and then start reading from the beginning,

In if condition we are finding the position of our searched word that we passed in our function. Once we get the position then we are passing this in the substr(). That will give us the string between the searched word and last position. This function will return the string that is stored by substr() function.



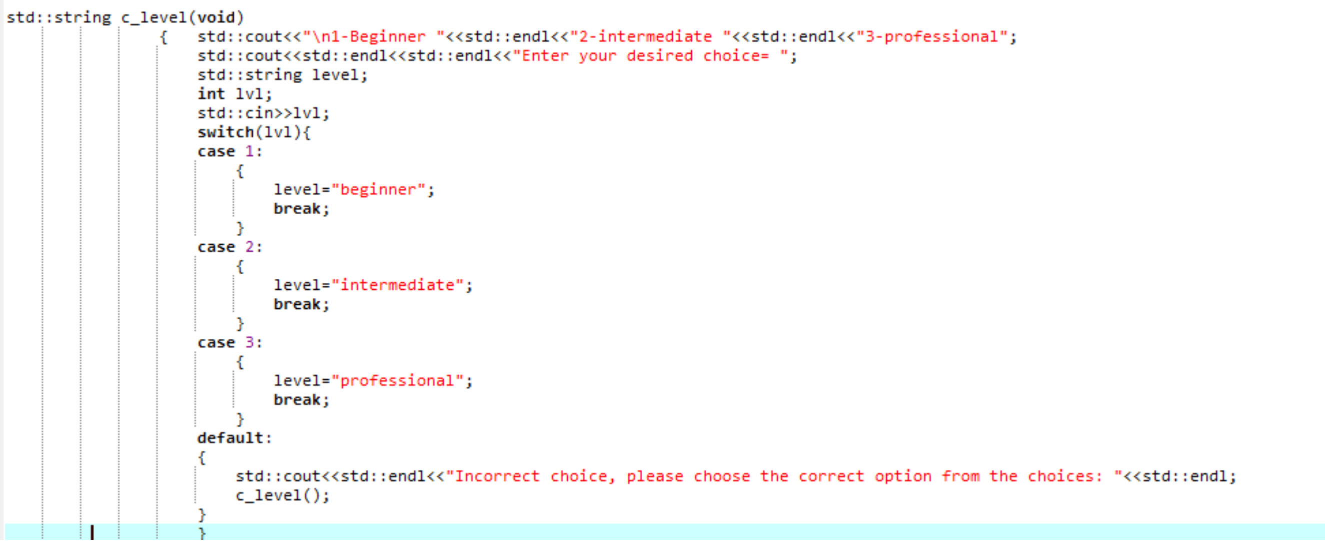
```
std::string string_finder(std::string nic,std::string find_w,std::string end_w)
{
    std::string path=path_finder(),str,str2,str3;
    if(nic=="revenue")
    {
        path+="admin\\\\";
        path+="Revenue.txt";
    }
    else
    {
    path+="Customer\\\\";
    path+=nic;
    path+="\\\\";
    path+="Data.txt";
    }
    std::ifstream fin;
    fin.open(path);
    size_t found,fa;
    int l_position;
    while(!fin.eof())
   {
      getline(fin,str);
      if(end_w=="end")
    {
        l_position=str.length();
    }
     if(end_w=="kg")
    {
        l_position=str.length()-2;
    }
      found=str.find(find_w);
      fa=found;
      if(found!=std::string::npos)
      {
```



```
          found=str.find("=");
          found+=2;
          l_position-=found;
              str2=str.substr(found,l_position);

        }
        else{
          str3="new";
          //str2[1]=str.substr(found,l_position);
        }
        break;

     }
     fin.close();
              return str2;
      if(fa==std::string::npos)
      {
          return str3;
      }

```

3. string c_level(void):

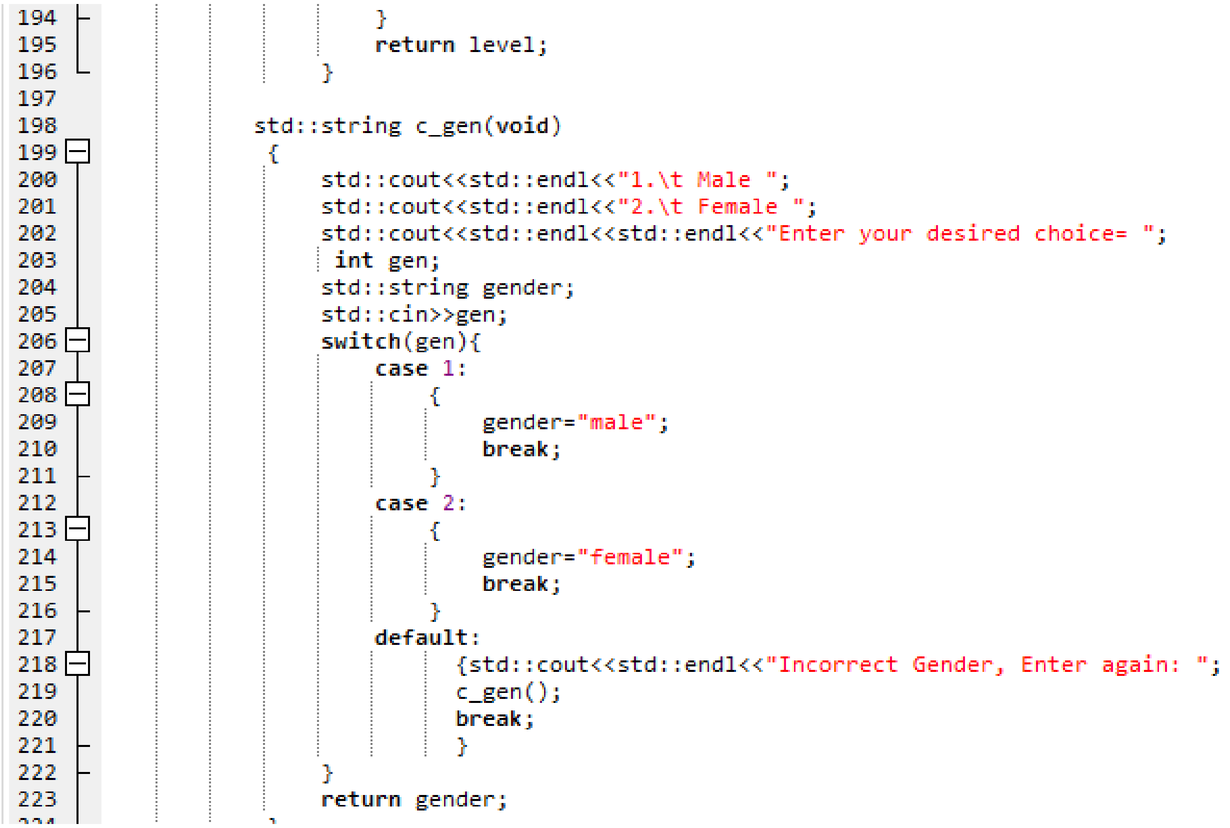
This function is just taking the choice of a level (beginner,intermediate,professional) from a user. The main purpose to create this function is that if a user enters any other number than 1,2 and 3 then the function will recurse and again ask the user to enter right level from the choices. This function will return the level to our main().



```
std::string c_level(void)
                {   std::cout<<"\n1-Beginner "<<std::endl<<"2-intermediate "<<std::endl<<"3-professional";
                    std::cout<<std::endl<<std::endl<<"Enter your desired choice= ";
                    std::string level;
                    int lvl;
                    std::cin>>lvl;
                    switch(lvl){
                    case 1:
                        {
                            level="beginner";
                            break;
                        }
                    case 2:
                        {
                            level="intermediate";
                            break;
                        }
                    case 3:
                        {
                            level="professional";
                            break;
                        }
                    default:
                    {
                        std::cout<<std::endl<<"Incorrect choice, please choose the correct option from the choices: "<<std::endl;
                        c_level();
                    }
                    }
```

4. string c_gen(void):

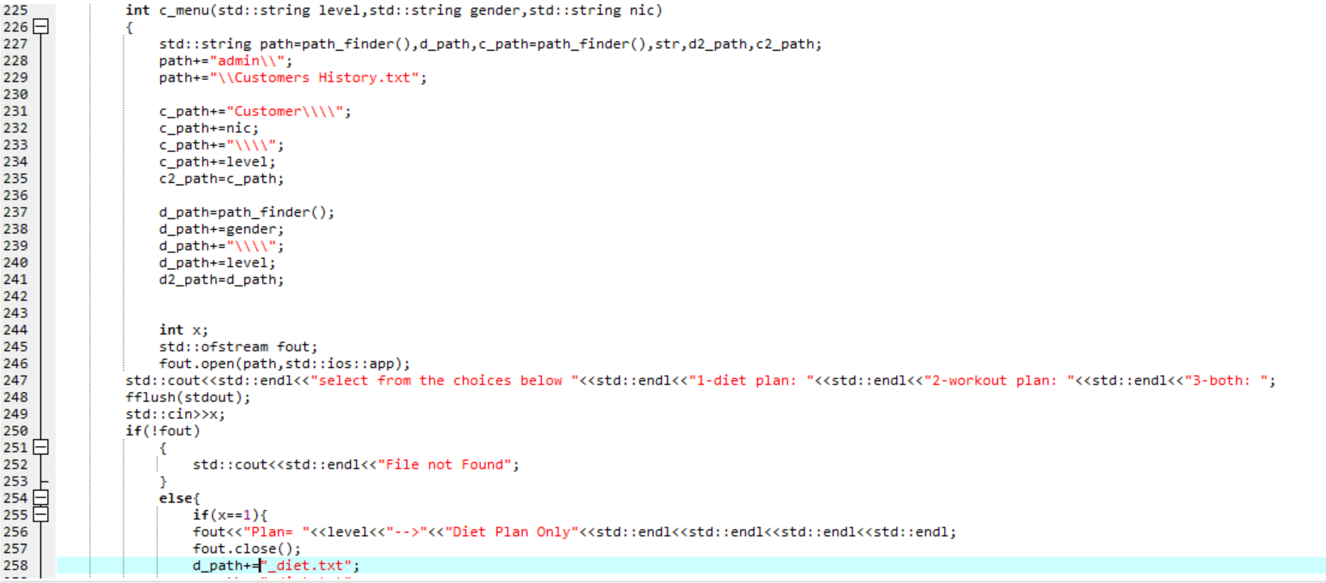
This function is just taking the choice of a gender (Male, Female) from a user. The main purpose to create this function is that if a user enters any other number than 1 and 2 then the function will recurse and again ask the user to enter right gender from the choices. This function will return the gender to our main().



```
            }
            return level;
        }

    std::string c_gen(void)
     {
        std::cout<<std::endl<<"1.\t Male ";
        std::cout<<std::endl<<"2.\t Female ";
        std::cout<<std::endl<<std::endl<<"Enter your desired choice= ";
         int gen;
        std::string gender;
        std::cin>>gen;
        switch(gen){
            case 1:
                {
                    gender="male";
                    break;
                }
            case 2:
                {
                    gender="female";
                    break;
                }
            default:
                  {std::cout<<std::endl<<"Incorrect Gender, Enter again: ";
                  c_gen();
                  break;
                  }
        }
        return gender;
```

5. C_menu():

This function will ask the user to select which program He/She want. If user enters the wrong choice then the function will recurse itself and ask the user to enter the correct choice from the menu. This function will also copy a file of a plan that user takes into the folder of that particular user.

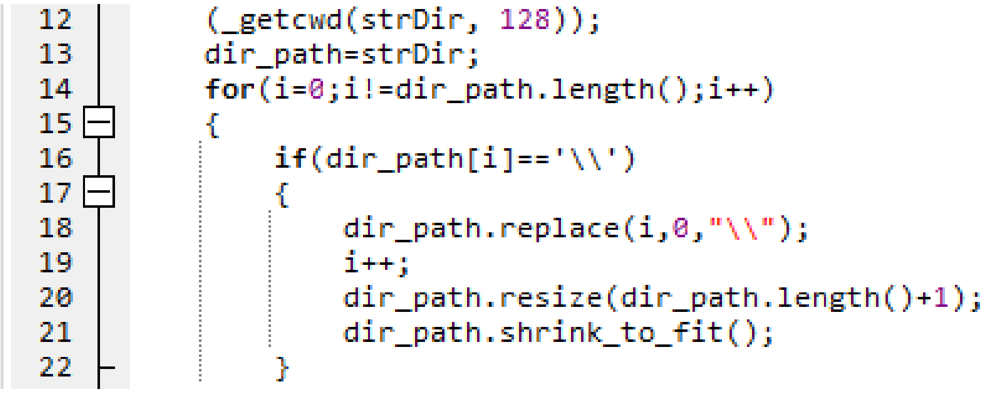


```
int c_menu(std::string level,std::string gender,std::string nic)
{
    std::string path=path_finder(),d_path,c_path=path_finder(),str,d2_path,c2_path;
    path+="admin\\";
    path+="\\Customers History.txt";

    c_path+="Customer\\\\";
    c_path+=nic;
    c_path+="\\\\";
    c_path+=level;
    c2_path=c_path;

    d_path=path_finder();
    d_path+=gender;
    d_path+="\\\\";
    d_path+=level;
    d2_path=d_path;


    int x;
    std::ofstream fout;
    fout.open(path,std::ios::app);
std::cout<<std::endl<<"select from the choices below "<<std::endl<<"1-diet plan: "<<std::endl<<"2-workout plan: "<<std::endl<<"3-both: ";
fflush(stdout);
std::cin>>x;
if(!fout)
    {
        std::cout<<std::endl<<"File not Found";
    }
    else{
        if(x==1){
        fout<<"Plan= "<<level<<"-->"<<"Diet Plan Only"<<std::endl<<std::endl<<std::endl<<std::endl;
        fout.close();
        d_path+="_diet.txt";
```

Pre-defined Functions Used:

1. Str.replace()

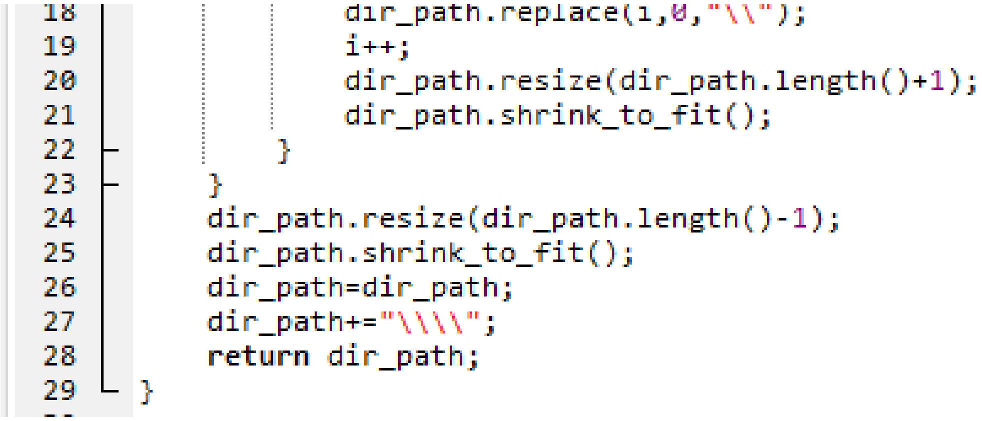
To replace one string with another.



```
(_getcwd(strDir, 128));
dir_path=strDir;
for(i=0;i!=dir_path.length();i++)
{
    if(dir_path[i]=='\\')
    {
        dir_path.replace(i,0,"\\");
        i++;
        dir_path.resize(dir_path.length()+1);
        dir_path.shrink_to_fit();
    }
```

2. Str.resize()

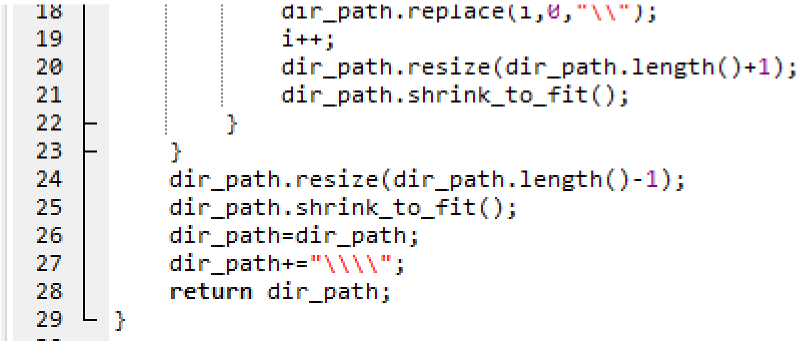
To get the size of a particular string.



```
            dir_path.replace(i,0,"\\");
            i++;
            dir_path.resize(dir_path.length()+1);
            dir_path.shrink_to_fit();
        }
    }
    dir_path.resize(dir_path.length()-1);
    dir_path.shrink_to_fit();
    dir_path=dir_path;
    dir_path+="\\\\";
    return dir_path;
}
```

3. Str.shrink_to_fit()

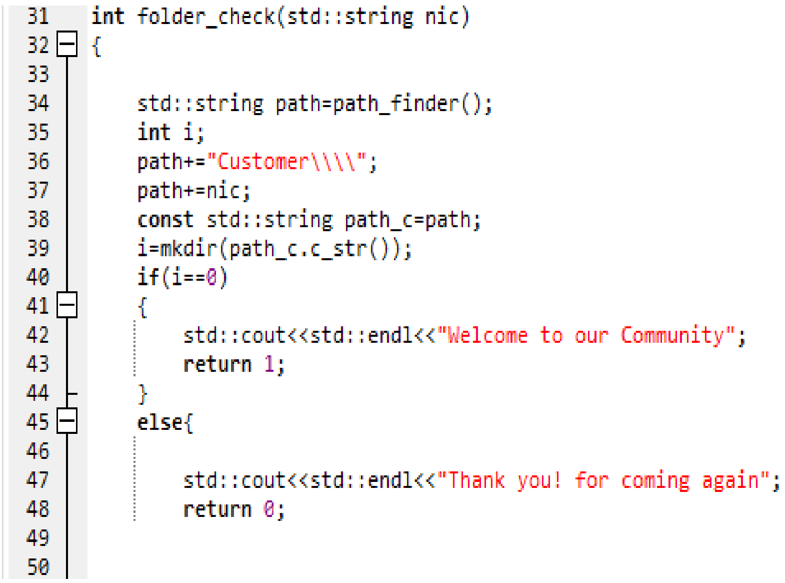
This was used to request the string to reduce its capacity to fit its size.



```
            dir_path.replace(i,0,"\\");
            i++;
            dir_path.resize(dir_path.length()+1);
            dir_path.shrink_to_fit();
        }
    }
    dir_path.resize(dir_path.length()-1);
    dir_path.shrink_to_fit();
    dir_path=dir_path;
    dir_path+="\\\\";
    return dir_path;
}
```

4. Mkdir()

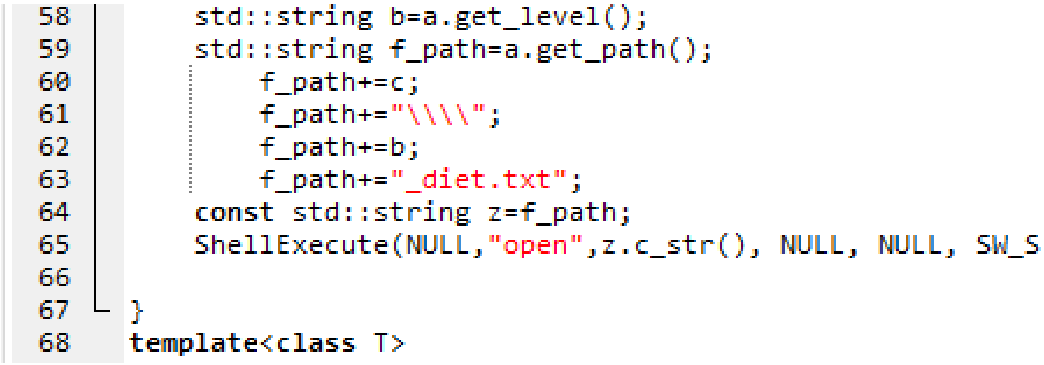
This function was used to create a directory specified by a pathname.



```
int folder_check(std::string nic)
{

    std::string path=path_finder();
    int i;
    path+="Customer\\\\";
    path+=nic;
    const std::string path_c=path;
    i=mkdir(path_c.c_str());
    if(i==0)
    {
        std::cout<<std::endl<<"Welcome to our Community";
        return 1;
    }
    else{

        std::cout<<std::endl<<"Thank you! for coming again";
        return 0;


```

5. Shell execute()

This function was used to open videos and other files.



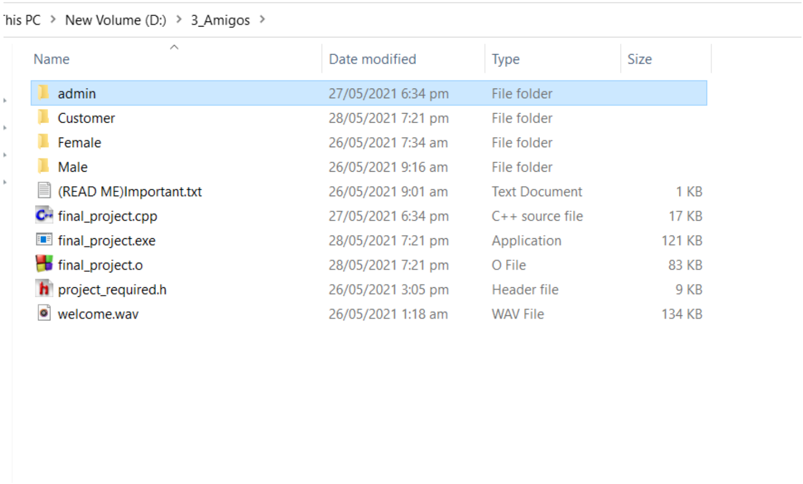
```
    std::string b=a.get_level();
    std::string f_path=a.get_path();
        f_path+=c;
        f_path+="\\\\";
        f_path+=b;
        f_path+="_diet.txt";
    const std::string z=f_path;
    ShellExecute(NULL,"open",z.c_str(), NULL, NULL, SW_S

}
template<class T>
```

FEATURES:

1. **BMI calculator**
2. **Provides fitness and workout plan**
3. **Provides authentic diet plan for every level**
4. **Applicable for male and female both**
5. **Applicable for every level; beginner, intermediate and professional**
6. **Provides video clips for every single exercise in a specific plan, separate for both male and female.**
7. **Free hours calculator**
8. **Separate data file for every single customer including customer history.**
9. **Total revenue generator.**
10. **All the data is being saved in files. It does not vanishes with the ending of the program.**
11. **Total plan sell counter.**

Results:

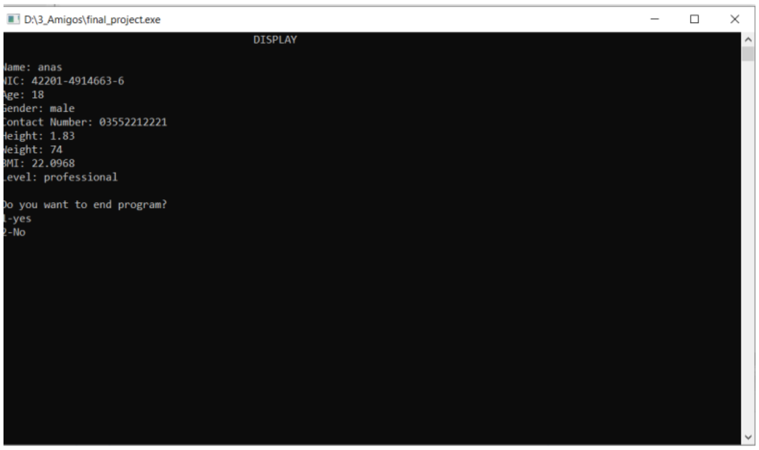
DIRECTORY



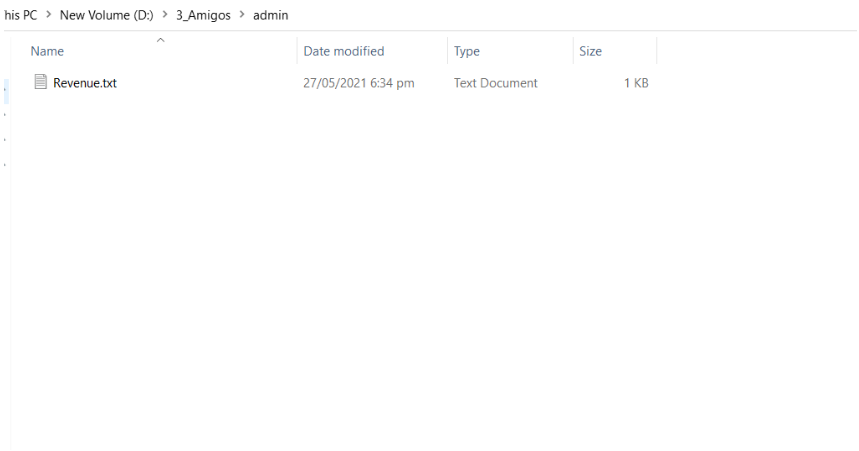
his PC > New Volume (D:) > 3_Amigos >

Name	Date modified	Type	Size
admin	27/05/2021 6:34 pm	File folder	
Customer	28/05/2021 7:21 pm	File folder	
Female	26/05/2021 7:34 am	File folder	
Male	26/05/2021 9:16 am	File folder	
(READ ME)Important.txt	26/05/2021 9:01 am	Text Document	1 KB
final_project.cpp	27/05/2021 6:34 pm	C++ source file	17 KB
final_project.exe	28/05/2021 7:21 pm	Application	121 KB
final_project.o	28/05/2021 7:21 pm	O File	83 KB
project_required.h	26/05/2021 3:05 pm	Header file	9 KB
welcome.wav	26/05/2021 1:18 am	WAV File	134 KB

DISPLAY DATA

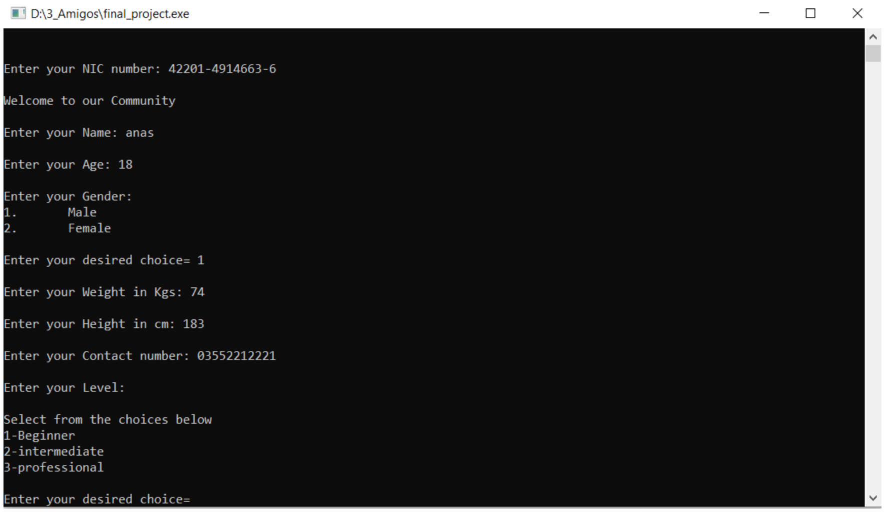


```
                                        DISPLAY

Name: anas
NIC: 42201-4914663-6
Age: 18
Gender: male
Contact Number: 03552212221
Height: 1.83
Weight: 74
BMI: 22.0968
Level: professional

Do you want to end program?
1-yes
2-No
```



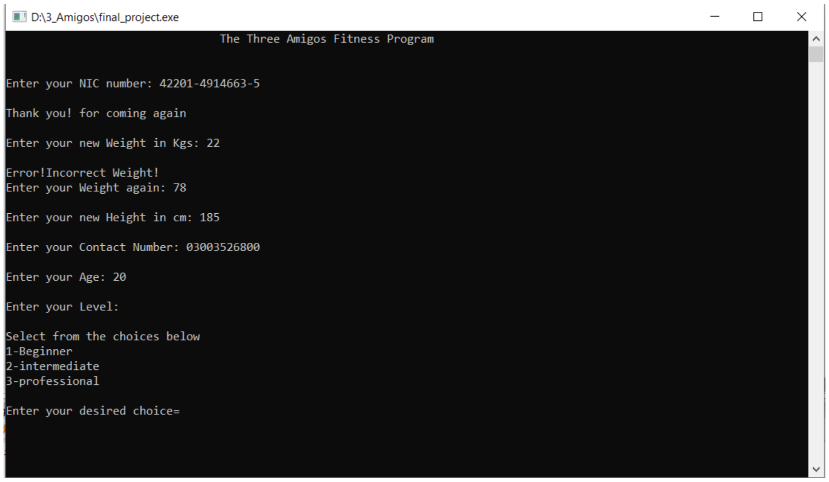
Name	Date modified	Type	Size
Revenue.txt	27/05/2021 6:34 pm	Text Document	1 KB

FIRST COMMER



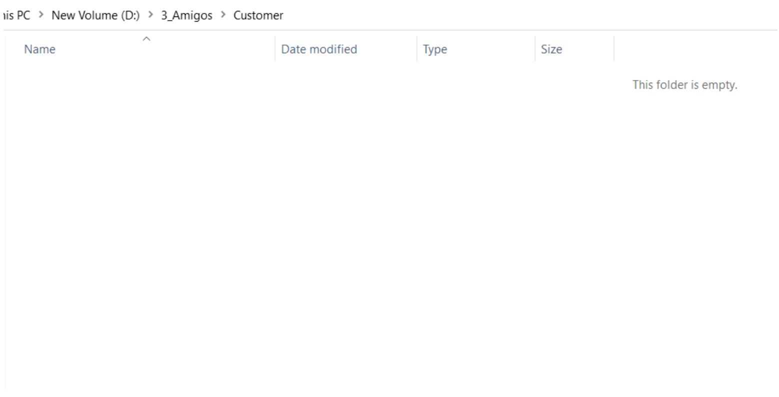
```
Enter your NIC number: 42201-4914663-6

Welcome to our Community

Enter your Name: anas

Enter your Age: 18

Enter your Gender:
1.       Male
2.       Female

Enter your desired choice= 1

Enter your Weight in Kgs: 74

Enter your Height in cm: 183

Enter your Contact number: 03552212221

Enter your Level:

Select from the choices below
1-Beginner
2-intermediate
3-professional

Enter your desired choice=
```

SECOND COMMER



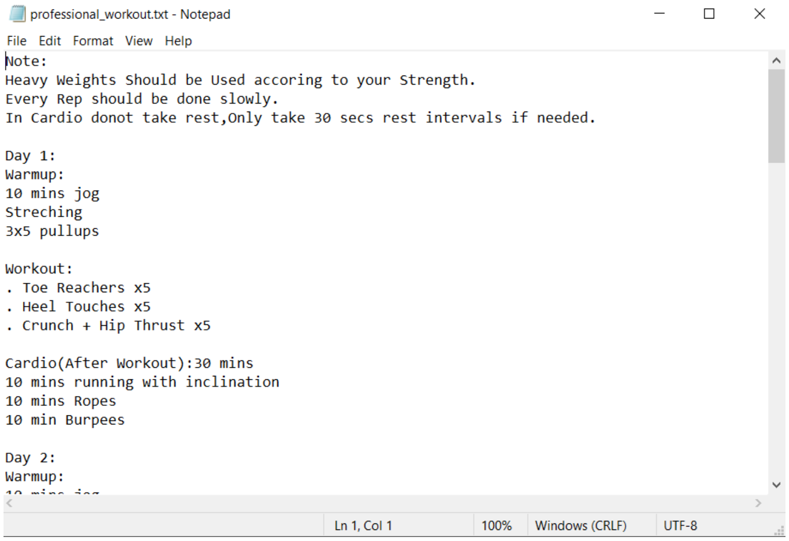
```
                                The Three Amigos Fitness Program


Enter your NIC number: 42201-4914663-5

Thank you! for coming again

Enter your new Weight in Kgs: 22

Error!Incorrect Weight!
Enter your Weight again: 78

Enter your new Height in cm: 185

Enter your Contact Number: 03003526800

Enter your Age: 20

Enter your Level:

Select from the choices below
1-Beginner
2-intermediate
3-professional

Enter your desired choice=
```

EMPTY CUSTOMER



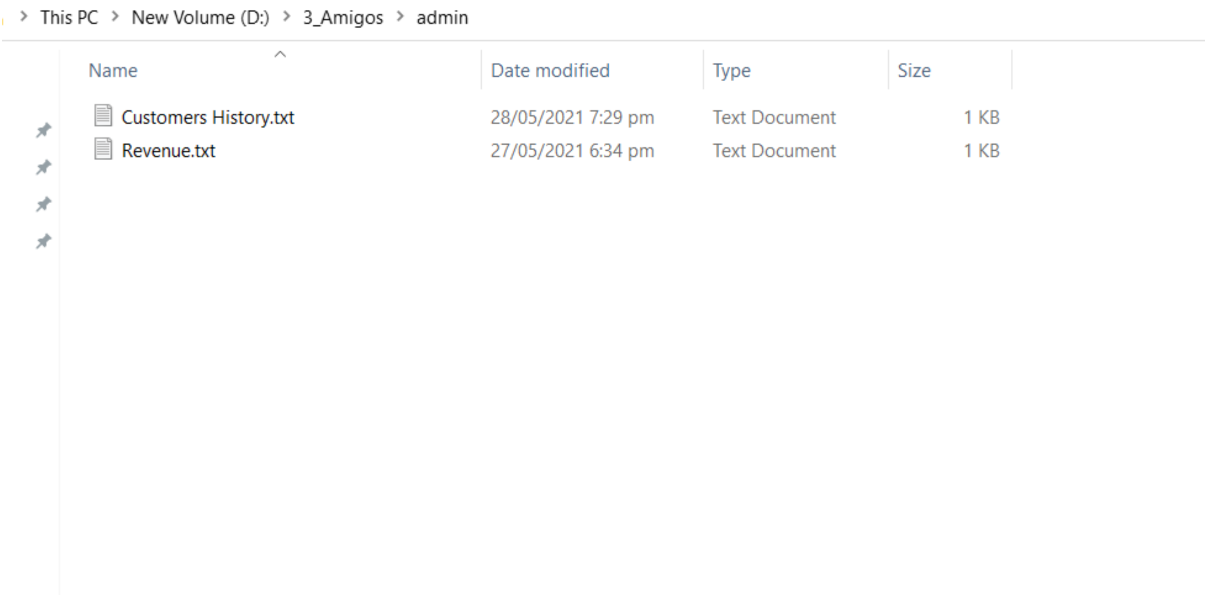
This folder is empty.

TEXT FILE OPEN


professional_workout.txt - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

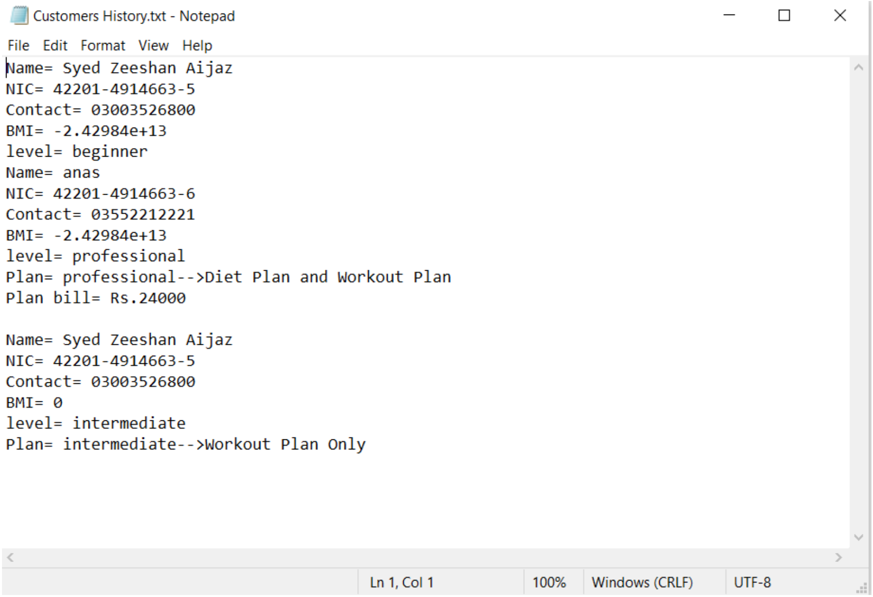
```
Note:
Heavy Weights Should be Used accoring to your Strength.
Every Rep should be done slowly.
In Cardio donot take rest,Only take 30 secs rest intervals if needed.

Day 1:
Warmup:
10 mins jog
Streching
3x5 pullups

Workout:
. Toe Reachers x5
. Heel Touches x5
. Crunch + Hip Thrust x5

Cardio(After Workout):30 mins
10 mins running with inclination
10 mins Ropes
10 min Burpees

Day 2:
Warmup:
10 mins jog
```

ADMIN DATA

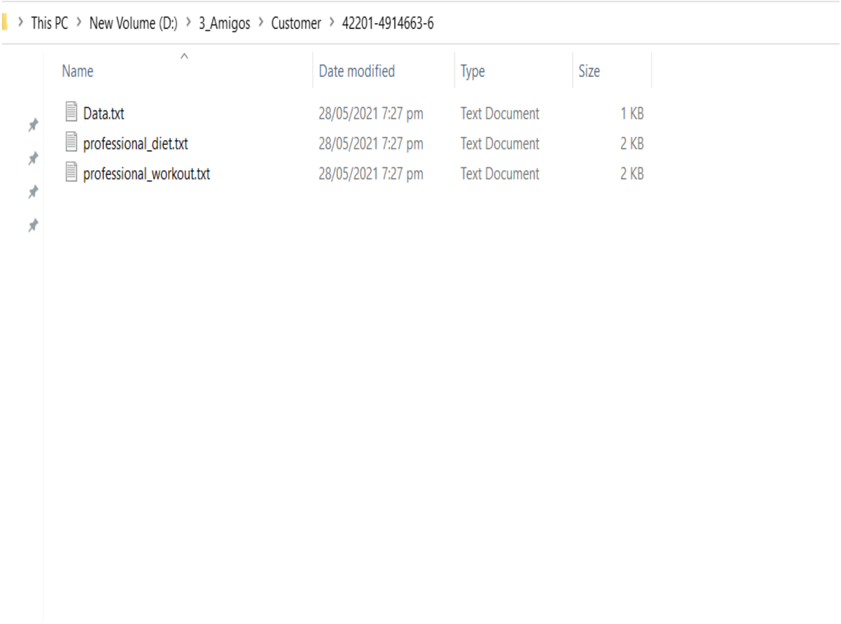


Name	Date modified	Type	Size
Customers History.txt	28/05/2021 7:29 pm	Text Document	1 KB
Revenue.txt	27/05/2021 6:34 pm	Text Document	1 KB

CUSTOMER HISTORY



```
Name= Syed Zeeshan Aijaz
NIC= 42201-4914663-5
Contact= 03003526800
BMI= -2.42984e+13
level= beginner
Name= anas
NIC= 42201-4914663-6
Contact= 03552212221
BMI= -2.42984e+13
level= professional
Plan= professional-->Diet Plan and Workout Plan
Plan bill= Rs.24000

Name= Syed Zeeshan Aijaz
NIC= 42201-4914663-5
Contact= 03003526800
BMI= 0
level= intermediate
Plan= intermediate-->Workout Plan Only
```

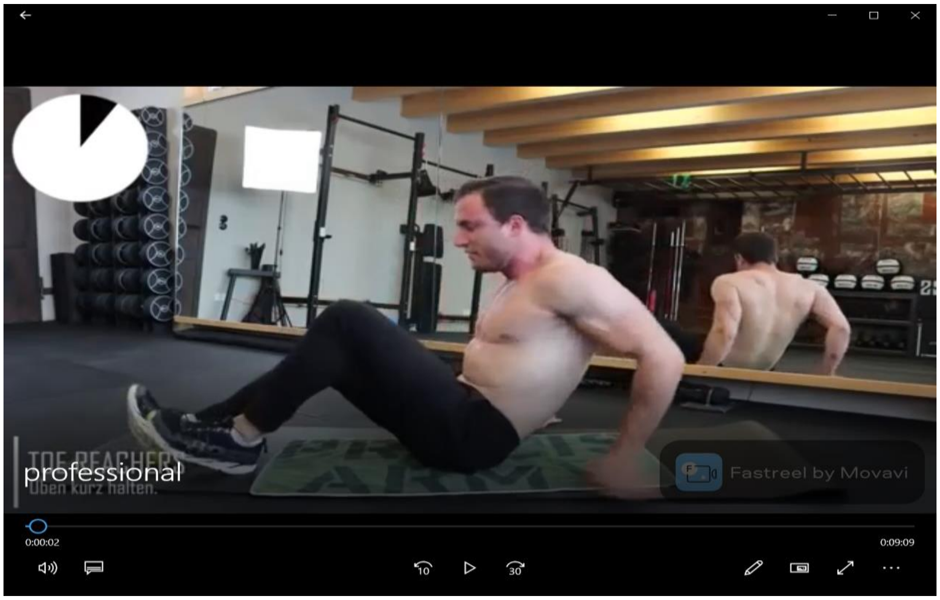

› This PC › New Volume (D:) › 3_Amigos › Customer › 42201-4914663-6

Name	Date modified	Type	Size
Data.txt	28/05/2021 7:27 pm	Text Document	1 KB
professional_diet.txt	28/05/2021 7:27 pm	Text Document	2 KB
professional_workout.txt	28/05/2021 7:27 pm	Text Document	2 KB

LOADING

```
Please wait ████████████_
```

VIDEO OP ENING FOR EVERY EXERCISE



THANKYOU