The Project Management System

The Project is all about developing an Application of Project Management System. This application helps the user to designate the tasks of different projects done by different team members. It also provide the report of all tasks with details of dead line of the project.

Submitted by:

- Asma Butt (2019-CS-16) 50% Leading
- Noor Fatima (2019-CS-18) 50% Leading
- Umar Razzaq (2019-CS-31)
- Kainat Jahan (2019-CS-06)
- Uswah Saeed (2019-CS-03)

Table Of Content

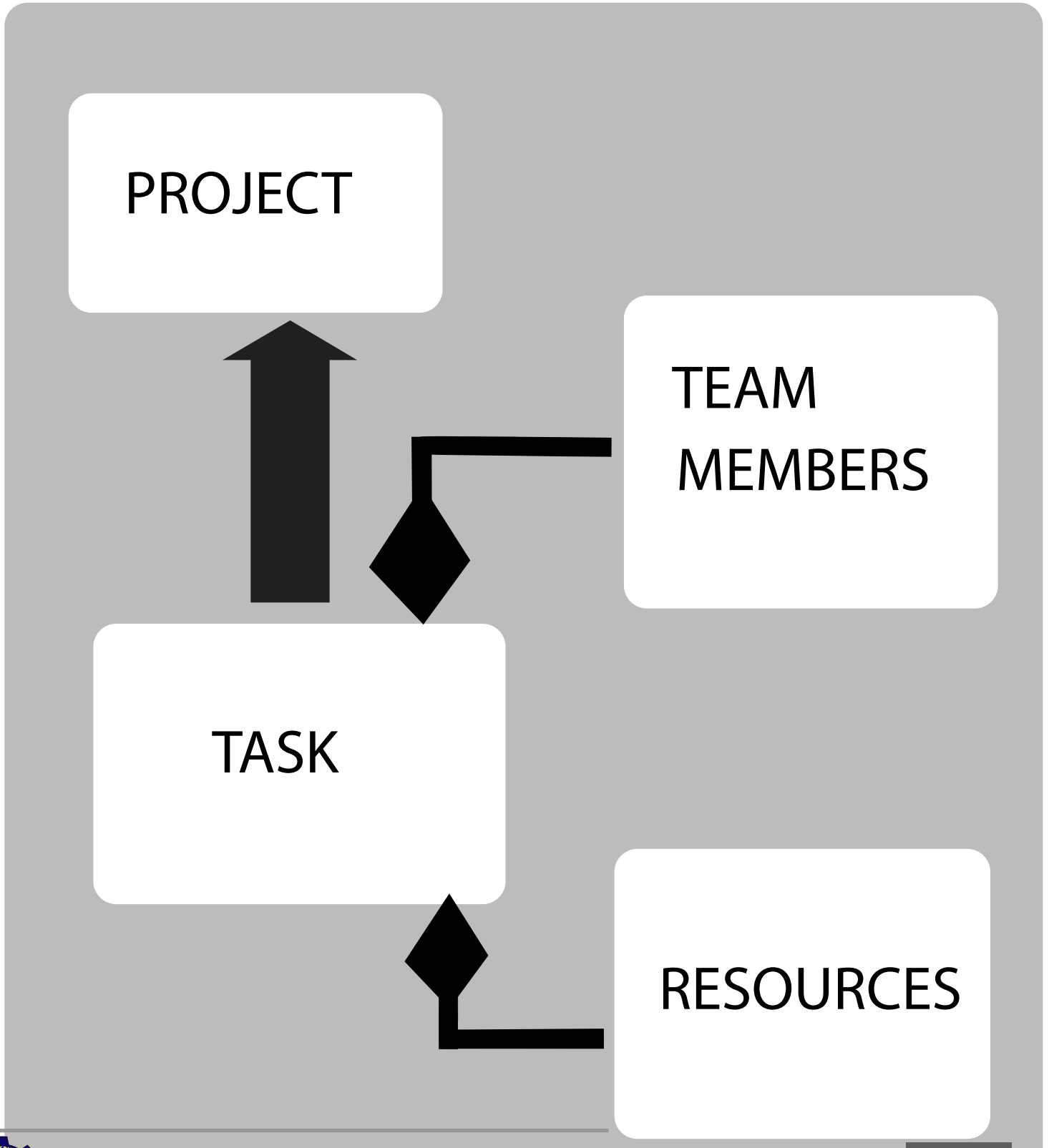
GOALS

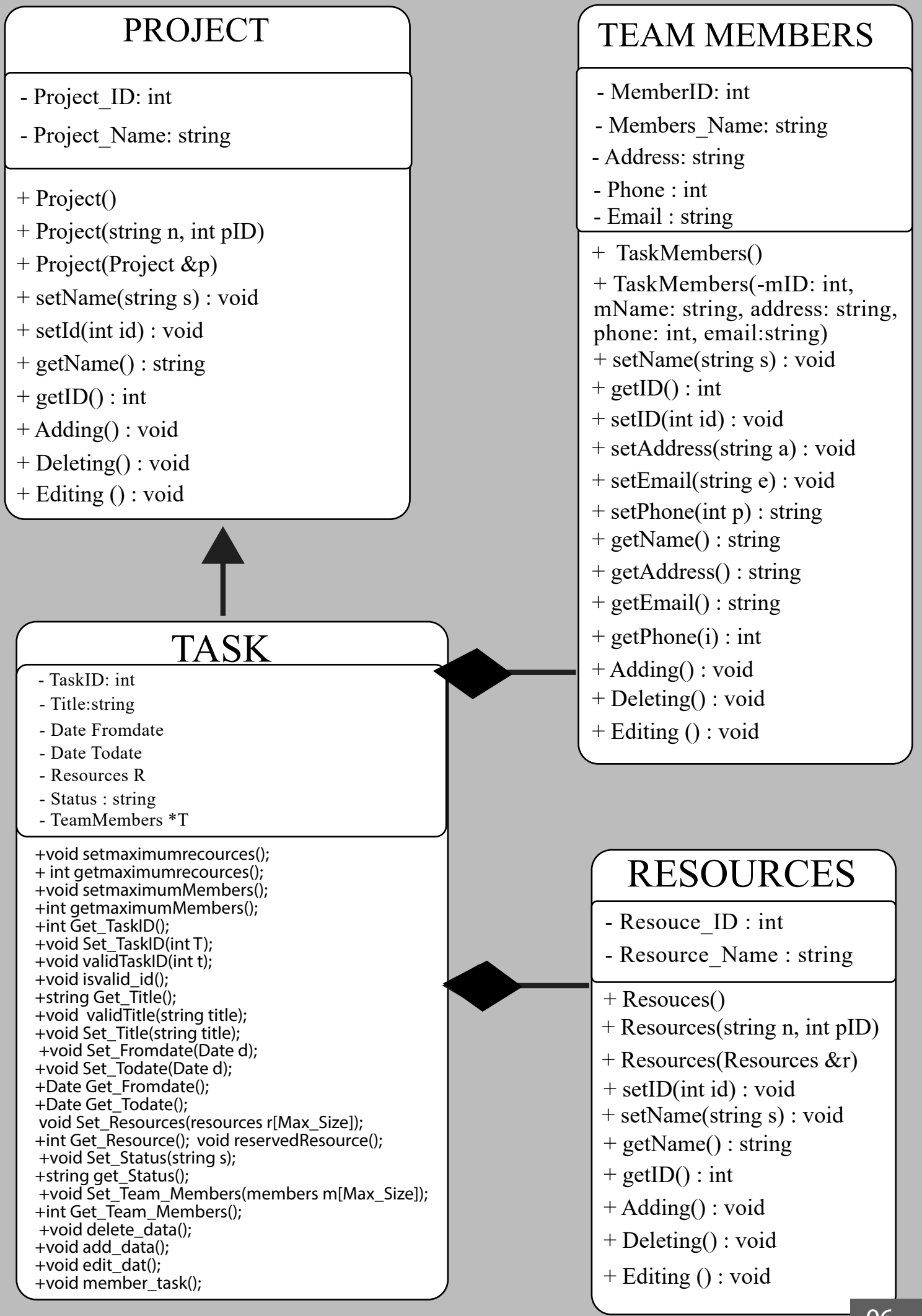
To cover all concepts of Object Oriented Programming. we, all team members, working on each and every point of OOP topics. In this application we learn from each other by collaborating

TOPICS

In this application we covered UML, Classes, Objects, Interfaces, Abstract Classes, Virtual functions, Friend Functions, Operator Overloading, Inheritance, Excess Visibility, Generic Funcions and Classes etc.
In addition we used the GUI Interface in this application of Project Management System.

UML DIAGRAM




PROJECT
- Project_ID: int
- Project_Name: string
+ Project()
+ Project(string n, int pID)
+ Project(Project &p)
+ setName(string s) : void
+ setId(int id) : void
+ getName() : string
+ getID() : int
+ Adding() : void
+ Deleting() : void
+ Editing () : void
TEAM MEMBERS
- MemberID: int
- Members_Name: string
- Address: string
- Phone : int
- Email : string
+ TaskMembers()
+ TaskMembers(-mID: int, mName: string, address: string, phone: int, email:string)
+ setName(string s) : void
+ getID() : int
+ setID(int id) : void
+ setAddress(string a) : void
+ setEmail(string e) : void
+ setPhone(int p) : string
+ getName() : string
+ getAddress() : string
+ getEmail() : string
+ getPhone(i) : int
+ Adding() : void
+ Deleting() : void
+ Editing () : void
TASK
- TaskID: int
- Title:string
- Date Fromdate
- Date Todate
- Resources R
- Status : string
- TeamMembers *T
+void setmaximumrecources();
+ int getmaximumrecources();
+void setmaximumMembers();
+int getmaximumMembers();
+int Get_TaskID();
+void Set_TaskID(int T);
+void validTaskID(int t);
+void isvalid_id();
+string Get_Title();
+void validTitle(string title);
+void Set_Title(string title);
+void Set_Fromdate(Date d);
+void Set_Todate(Date d);
+Date Get_Fromdate();
+Date Get_Todate();
void Set_Resources(resources r[Max_Size]);
+int Get_Resource(); void reservedResource();
+void Set_Status(string s);
+string get_Status();
+void Set_Team_Members(members m[Max_Size]);
+int Get_Team_Members();
+void delete_data();
+void add_data();
+void edit_dat();
+void member_task();
RESOURCES
- Resouce_ID : int
- Resource_Name : string
+ Resouces()
+ Resources(string n, int pID)
+ Resources(Resources &r)
+ setID(int id) : void
+ setName(string s) : void
+ getName() : string
+ getID() : int
+ Adding() : void
+ Deleting() : void
+ Editing () : void

WHY THIS APPLICATION!

Project Management System is used for storing and managening the details of different projects. It can be operated by admin only who will enter the details and credentials of client in order to keep every single detail managed appropriately, if we talk about manual working, it requires more man power and consumes more time, so it would not be regarded as efficient as it may result in documentation loss.

This application is very useful as it can be used in various different set ups like as a wedding planner, for storing student's project and making analysis and in many engineering fields as well.

In nutshell, this application is going to give client relief from hectic manual information storage procedure.

Dictionaries Used

#include is a way of including a standard or user-defined file in the program and is mostly written at the beginning of any C/C++ program. This directive is read by the preprocessor and orders it to insert the content of a user-defined or system header file into the following program.

IOSTREAM:
iostream stands for standard input-output stream. This header file contains definitions to objects like cin, cout, cerr etc.

IOMANIP:
Iomanip stands for input output manipulators. The methods declared in this files are used for manipulating streams. This file contains definitions of setw, setprecision, etc

FSTREAM:
This header file mainly describes the file stream. This header file is used to handle the data being read from a file as input or data being written into the file as output.

CCTYPE:
The C++ <cctype> header file declares a set of functions to classify (and transform) individual characters. For example, isupper() checks whether a character is uppercase or not.

CONIO.H:
The conio.h header file used in C programming language contains functions for console input/output. Some of its most commonly used functions are clrscr, getch, getche, kbhit etc. They can be used to clear screen, change color of text and background, move text, check whether a key is pressed or not and to perform other tasks.

STRING:
The string header provides functions for dealing with strings — null-terminated arrays of characters. This includes functions like strlen and strcpy etc.

WORKING & project highlights

Some Main Classes:

PROJECT

Project ID	Name
1	Term Project
2	Wedding Plans
3	…

thia class of project is storing the Names of the projects in correspondence with prject ID.

Class Data Members and Constructors:

```
class Project
{
    //Private Data Members//
protected:
    int Project_ID;

    string    Project_Name;
public:
    //Default Constructor//
    Project() {


        Project_Name = "\0";
        Project_ID = 0;
        cout << "default constructor of project class is called\n";

    }


    Project(int pID, string n)
    {

        Project_ID = pID;
        Project_Name = n;
        cout << "parametric constructor is called from project class\n";
    }
```

This class is using two data members Project_ID and Project_Name, and initializing them in default non-parameterized constructor further variables are used to assign their values in a parameterized constructor.

Setters & Getters

```
Project(int pID, string n)
{

    Project_ID = pID;
    Project_Name = n;
    cout << "parametric constructor is called from project class\n";
}

// Setter Functions//
void SetName(string s) {

    if (s == "\n" || s == " ")
    {
        cout << "do nothing";
    }
    else
    {

        Project_Name = s;

    }


}

void SetID(string i)
{

    int id;
    id = stoi(i);
    if (id == -1)
    {
        Project_ID = 1;
    }
    else
```

```
// Getter Functions//
string GetName()
{
    cout << "your project name outcome is:";

    cout << Project_Name;
    cout << "\n";
    return this->Project_Name;


}
int GetID() {

    cout << "your project id result:";
    cout << Project_ID;
    return (Project_ID);
}

// Public Function Decleration //
void AddProject();
void ReadProject();
string op, en;
void setoption(string s);
string getoption();
void setEname(string s);
string getEname();

void EditProject(string op, string name);
```

Writing & Reading Information

```
// Public Function Decleration //
void    Project::AddProject()
{
    ofstream myfile;
    myfile.open("adding.txt", ios::app);
    if (myfile.is_open())

    {



        int a = GetID();
        string s = GetName();



        myfile << a << s;

    }

    myfile.close();
}
```

```
void    Project::ReadProject()
{
    ifstream myfile_1;
    myfile_1.open("adding.txt", ios::in | ios::app | ios::ate);
    if (myfile_1.fail())
    {
        cout << "it is not opened\n";
    }
    else
    {
        char data;
        while (!myfile_1.eof())
        {
            data = myfile_1.get();
            cout << "Your record is Added :" << data << endl;
        }
        myfile_1.close();
    }
```

The functions AddProject and ReadProject writes and read information respectively, from the file created named as myFile and myFile_1.

Editing & Deleting Information

```
void    Project::DeleteProject(string op)
{
    int option = stoi(op);
    ofstream myfile_3;
    myfile_3.open("adding.txt", ios::app | ios::ate);
    if (myfile_3.is_open())

    {

        if (option == this->Project_ID)
        {

            for (int i = 0; i < 3; i++)
            {
                if (i != option)
                {
                    myfile_3 << Project_ID;
                    myfile_3 << Project_Name;

                }
            }
            myfile_3.close();

        }
    }
```

```
void  Project::EditProject(string op, string name)
{

    int option = stoi(op);

    ofstream myfile;
    myfile.open("adding_2.txt", ios::out | ios::app);
    if (myfile.is_open())

    {



        if (option == this->Project_ID)
        {
            SetName(name);
        }

        myfile << Project_ID << "\n";
        myfile << Project_Name << "\n";
    }

    myfile.close();
```

The functions EditProject and DeleteProject are used for Eidting and deleting information respect ively.

Member Functions declaration

```
    // Public Function Decleration //
    void AddProject();
    void ReadProject();
    string op, en;
    void setoption(string s);
    string getoption();
    void setEname(string s);
    string getEname();

    void EditProject(string op, string name);

    void DeleteProject(string op);

    void  DisplayProject();

};
```

The prototypes of all member_functions are declared in the class.

MEMBERS

Member ID	Name	Address	Phone	Email
1	Mohamed	…	…	…
2	Ahmed	…	…	…
3	Mostafa	…	…	…

This class is storing Names of persons along with their address, phone numbers, email addresses and assigning every single person a unique ID (Member ID).

Class Data Members and Constructors:

```
class  members
{

public:
    int member_id;
    string name;
    string address;
    int phone;
    string email;
    //MEMBERS//

    string op, E;
    //CONSTRUCTOR//
    members()
    {
        member_id = 0;
        name = "asma";
        address = "house";
        phone = 0;
        email = "asmabutt232@gmail.com ";
        cout << "default constructor of member class is called\n";

    }
    //parametric constructor//
    members(int m_id, string m_n, string addr, int ph, string email_1)
    {
        member_id = m_id;
        name = m_n;
        address = addr;
        phone = ph;
        email = email_1;
        cout << "parametric constructor of class memeber is called\n";
```

This class is having member_id, name, address, phone, email as DataMembers, initialized in default constructor and assigned their values to variables in parameterized constructor.

Setters & Getters

```
//SETTERS//
void set_ID(string me) {
    member_id = stoi(me);
}
void set_address(string a) {
    address = a;
}
void set_phone(string ph) {
    phone = stoi(ph);
}
void set_email(string e) {
    email = e;
}
```

```
//GETTERS//
int get_memberID()
{
    return this->member_id;


}
string  get_name()
{
    return name;
}
string get_address() {
    return  address;

}
int  get_phone()
{
    return this->phone;

}
string  get_email() {
    return this->email;
}
```

Setters are created to initiallize data members individually whereas Getters are created to return the values of data members set by user separately.

Writing & Reading Information

```
}

void take_input() {

    ofstream newFile;
    newFile.open("newFile.txt", ios::app);
    if (!newFile)
    {
        cout << "File not created successfully";


        int s = get_memberID();
        string n = get_name();
        string e = get_email();
        int p = get_phone();
        string a = get_address();


        newFile << s << setw(10) << n << setw(20) << a << setw(20) << p << setw(10) << e << e
        newFile.close();
    }
}
```

```
        //WRITE INTO FILE//
        //TAKE ALL INFORMATION FROM USER//

        cout << "Enter Member's ID: " << endl;
        cin >> member_id;

        cout << "Enter Member's Name: " << endl;
        cin.ignore();
        if (cin.peek() == '/n') {
            cin.ignore();
        }
        getline(cin,name[member_id]) ;

        cout << "Enter Member's address: " << endl;
        if (cin.peek() == '/n') {
            cin.ignore();
        }
```

```
        if (cin.peek() == '/n') {
            cin.ignore();
        }
        getline(cin,name[member_id]) ;

        cout << "Enter Member's address: " << endl;
        if (cin.peek() == '/n') {
            cin.ignore();
        }
        getline(cin,address[member_id]);

        cout << "Enter Member's phone number: " << endl;
        if (cin.peek() == '/n') {
            cin.ignore();
        }
        getline(cin,phone[member_id]);

        cout << "Enter Member's email: " << endl;
        if (cin.peek() == '/n') {
            cin.ignore();
        }
        getline(cin,email[member_id]);
```

```
//read from file//

void read_input() {

    ifstream myfile;
    myfile.open("newfile.txt", ios::in | ios::app);
    if (!myfile)
    {
        cout << "file not opened successfully";
    }
    else {
        cout << "member id" << setw(10) << "name" << setw(10) << "address" << setw(10) << "phone" << setw(10) << "email" << endl;
        myfile >> member_id >> setw(10) >> name[member_id] >> setw(20) >> address[member_id] >> setw(20) >> phone[member_id] >> setw(10) >> email[member_id];
        cout << endl;
        cout << member_id << setw(10) << name[member_id] << setw(20) << address[member_id] << setw(20) << phone[member_id] << setw(10) << email[member_id] << endl;
        myfile.close();
    }
}
```

The functions AddProject and ReadProject writes and read information respectively, from the file created named as myFile and myFile_1.

Editing & Deleting Information

```
void Edit_Input() {

    int id;
    char y;
    cout << "Enter your Member ID: ";
    cin >> id;
    cout << "Enter what you want to edit: (Name, address,phone,email)" << endl;
    cout << "press N for editing name" << endl;
    cout << "press A for editing adress" << endl;
    cout << "press P for editing phone" << endl;
    cout << "press E for editing email" << endl;
    cin >> y;

    if (y == 'N') {
        cout << "Write your name: ";
        if (cin.peek() == '\n') {
            cin.ignore();
        }
        getline(cin, name);

    }
    else if (y == 'A') {
        cout << "Write your address: ";
        if (cin.peek() == '\n') {
            cin.ignore();

        }
        getline(cin, address);

    }
    else if (y == 'P') {
        cout << "Write your phone number: ";
        if (cin.peek() == '\n') {
            cin.ignore();
        }
        cin >> phone;

    }
    else if (y == 'E') {
        cout << "Write your email: ";
        if (cin.peek() == '\n') {
            cin.ignore();
        }
        getline(cin, email);

    }
    else {
        cout << "Choose the given option correctly!";
    }
```

The functions EditProject and DeleteProject are used for Eidting and deleting information respect ively.

Validations

```
}
bool isValidate_ID(int member_id)
{
    if (!(isdigit(member_id)))
    {

        return false;
    }
    else
    {
        return true;
    }


}

bool isValidate_name(string n) {

    if (!(isalpha(n[4])))
    {

        return false;
    }
    else
    {
        return true;
    }
```

Validations are done in order to check whether the given information is syntactically correct or not,

RESOURCES

Resource ID	Name
1	Laptop 1
2	Printer 1
3	...

"Resource Class" is a base class. Two private Members are used.Resource_ID:int,Resource_Namme:string.Default Constructor and functions used are:SetName is used to input name of resource in private members.SetID is used to input the ID of resource.Get Name is used to output of name resource.GetID is used to output the Id of Resource.AddResource():it is used to add resource details like ID & Name. Edit (): it is used to update or edit the resource record that user want to. Delete Resource (): it is used to delete the resource that user wants to delete by entering the name of the resource.

Class Data Members and Constructors:

```
class resources
{
public:
    int id;
    string resource_name;
    string op, E;
    resources()
    {

        id = 0;
        resource_name = "laptop";
        cout << "default constructor is called\n";

    }
    resources(int id_1, string name)
    {

        id = id_1;
        resource_name = name;

    }
```

This class is having ID, resource_name as DataMembers, initialized in default constructor and assigned their values to variables in parameterized constructor.

Setters & Getters

```
}
resources(int id_1, string name)
{

    id = id_1;
    resource_name = name;
}

int get_id() {
    return  id;

}

string get_name() {
    return  resource_name;
}
void set_id(string r_id) {
    id = stoi(r_id);

}
void  setoption(string s) {
    op = s;
}
string getoption() {
    return op;
}
void  setE(string s)
{
    E = s;
}
string   getE() {
    return E;
}

void set_name(string name_1) {
    resource_name = name_1;
}
```

Setters are created to initiallize data members individually whereas Getters are created to return the values of data members set by user separately.

Writing & Reading Information

```
}

void add_data() {
    ofstream file_1;
    file_1.open("application.txt", ios::out | ios::app);
    if (file_1.is_open()) {
        cout << "file is not open\n";
        file_1 << endl;
        string x = get_name();
        int y = get_id();
        for (int i = 0; i < 2; i++)
        {
            file_1 << y << "\n" << x << endl;

        }
        file_1.close();
```

The functions AddProject and ReadProject writes and read information respectively, from the file created named as myFile and myFile_1.

Editing & Deleting Information

```
void display_data() {
    cout << "resource id:" << id;
    cout << endl;
    cout << "resources name: " << resource_name;
    cout << endl;


}
void edit_data(string o, string n) {
    ifstream myfile;
    int option = stoi(o);

    if (myfile.is_open())
    {

        cout << "file is opened successfully\n";
        if (option == this->id)
        {

            set_name(n);
        }

        myfile >> resource_name;


        //  cout << "Your record is edited:" << name;

    }
    myfile.close();
```

```
void delete_data(string n) {
    ifstream myfile_2;
    resources obj;
    int option = stoi(n);
    myfile_2.open("application.txt", ios::app);
    if (myfile_2.is_open())
    {
        cin >> option;

        if (option == this->id)
        {

            string name_2 = "**";

            obj.resource_name = name_2;

            for (int i = 0; i < 3; i++)
            {

                myfile_2 >> resource_name[i];
                myfile_2 >> id;
            }

        }
    }
    myfile_2.close();
    cout << "you file is deelted\n";
}
};
```

The functions EditProject and DeleteProject are used for Eidting and deleting information respect ively.

Validations

```
bool isvalidID() {

    if (isdigit(id))
    {
        //  cout << "Your id is correct\n";
        return true;
    }
    else
    {
        return false;
    }

}

bool isvalidname() {
    if (isalpha(resource_name == "0"))
    {
        cout << "your resource name is not valid try again\n";
        return false;
    }

    else {

        cout << "your resource name is valid\n";
        return true;
    }

}
```

Validations are done in order to check whether the given information is syntactically correct or not,

DATE

Class Data Members and Constructors:

```
class Date
{
public:
    int day;
    int month;
    int year;
    Date();
    Date(int d, int m, int y);
    void setday(string x);
    void setmonth(string h);

    void setyear(string n);
    int getday();
    int getmon();
    int getyear();
    bool Validate_Date();
};

Date::Date()
{
    this->day = 0;
    this->month = 0;
    year = 0;
}
Date::Date(int d, int m, int y)
{
    day = d;
    month = m;
    year = y;
}
```

This class is having day, month, year initialized in default constructor and assigned their values to variables in parameterized constructor.

Setters & Getters

```
void  Date::setday(string x)
{

    day = stoi(x);
}

void Date::setmonth(string h)
{

    month = stoi(h);
}

void Date::setyear(string n)
{
    year = stoi(n);
}
int Date::getday()
{

    return day;
}
int Date::getmon()
{

    return month;
}
```

Setters are created to initiallize data members individually whereas Getters are created to return the values of data members set by user separately.

DATE VALIDATION

```
}

bool Date::Validate_Date()
{
    if ((getday() >= 1) && (getday() <= 31))
    {
        if ((getmon() >= 1) && (getmon() <= 12))
        {
            if (((getmon() == 4) || (getmon() == 6) || (getmon() == 9) || (getmon() == 11)) && (getday() <= 30))
            {

                return true;

            }

            else if (((getmon() == 1) || (getmon() == 8) || (getmon() == 3) || (getmon() == 5) || (getmon() == 7) || (getmon() == 10) || (getmon() == 12)) && (getday() <= 31))
            {

                return true;

            }

            else if ((getmon() == 2) && (getday() <= 29))
            {

                return true;
            }

            else
            {

                return false;
            }
        }

        else
        {
```

TASK

Task ID	Project ID	Title	From Date	To Date	Member ID	Resources	Status
1	1	Design the class UML	10/1/2013	10/1/2013	[1, 2]	[1]	done
2	1	…	10/1/2013	10/1/2013	…	…	…
3	3	…	12/2/2013	12/2/2013	…	…	…
4	3	…	13/2/2013	13/2/2013	…	…	…

This class contains the main working of project. In this class the user enters the entire information about his/her task which includes the id of the task , title of the task ,the project id, the resources used for the task , the team members working in the class, the starting date called the "From Date", the deadline called the "To-Date", and the Status . The purpose of this class is to compile all the
entries and store the entries in a file, where the information can be deleted , edit or more information can be added according to the choice of the user.

Task Id: task id is the Identification number allotted to a task by the user.

Project Id : This is accessed by inheritance from "project" class. The user enters the Project id for the task .

Title : Given by its name this data member stores the title of the task.

From-Date: From-date is the date when the user starts working on the task . This is specified by the user.

To-Date : This data member stores the deadline of the task.

TeamMembers : This data member stores the ids of the team members working on a task . This is used as an array ,so that various tem members can be added. Composition from the members task is done here .So that there ids can be stored.

Resources: This data member stores the ids of the resources being used for a task . This is used as an array ,so that various resources can be added. Composition from the resouces task is done here .So that there ids can be stored.

Getters: Getters here are used to return the values of the datamembers.

Setters: Setters are being used to store the entries given by the user into the data members.

reservedResource() : This function will show the users which resources are reserved.

add_data(): This Function will store all the entries into a file called "task.txt" according to the pattern given in the description. delete_data():This Function will delete the task the user want to delete and store null entries in its place. edit_dat():This Function will edit the entries of the task chosen by the user and then will upload new entries in file

Class Data Members and Constructors:

```
//---------------------------------Default Constructor-----------------------------------------------------
Task::Task()
{

    TaskID = 0;
    Title = "\0";
    Status = "";
    maxc = 0;
    maxm = 0;
    Fromdate.setday(1);
    Fromdate.setmonth(7);
    Fromdate.setyear(2020);
    Fromdate.Validate_Date();

    Todate.setday(9);
    Todate.setmonth(12);
    Todate.setyear(2020);
    Todate.Validate_Date();

    for (int i = 0; i < getmaximumrecources(); i++)
    {
        Resource[i].id = {0};
    }
    for (int i = 0; i < getmaximumMembers(); i++)
    {
        Team_Members[i].member_id = {0};
    }
//  cout << "default constructor is called from task class\n";
}
```

```
//---------------------------------Parameterized Constructor-----------------------------------------------------//
Task::Task(int tID, int projectID, string title, Date fdate, Date tdate, resources  R[Max_Size], string status, members T[Max_Size])
{

    TaskID = tID;
    Title = title;
    Fromdate = fdate;
    Todate = tdate;
    Project_ID = projectID;
    for (int i = 0; i < getmaximumrecources(); i++)
    {
        Resource[i].id = R[i].id;
    }
    for (int i = 0; i < getmaximumMembers(); i++)
    {
        Team_Members[i].member_id = T[i].member_id;
    }
    Status = status;
    //cout << "Parametric constructor is called" << endl;
}
```

Task ID &Project ID

```
//---------------------------------Task ID-----------------------------------------------------//

void Task::validTaskID(int t)
{

    if (!(isdigit(t)))
    {
        t = -1;
    }

}

void Task::Set_TaskID(int T)
{
    validTaskID(T);
    Title = T;

}
int Task::Get_TaskID()
{
    return TaskID;
}
```

```
//---------------------------------Project ID-----------------------------------------------------

void Task::isvalid_id()
{

    if ((isdigit(Project_ID)))
    {
        cout << "Project id is valid\n";
    }
    else {

        cout << "Project id is not valid";

    }
}
```

From Date, TO Date

```
//---------------------------------To Date-----------------------------------------------------

void Task::Set_Todate(Date d)
{
    d.Validate_Date();
    Todate.setday((d.day));
    Todate.setmonth((d.month));
    Todate.setyear((d.year));


}

Date Task::Get_Todate()
{
    Todate.day;
    cout << "/";
    Todate.month;
    cout << "/";
    Todate.year;
    return Todate;
}
```

```
}
//---------------------------------From Date-----------------------------------------------------
void Task::Set_Fromdate(Date d1)
{
    d1.Validate_Date();

    Fromdate.setday((d1.day));
    Fromdate.setmonth((d1.month));
    Fromdate.setyear((d1.year));


}

Date Task::Get_Fromdate()
{
    Fromdate.day;
    cout << "/";
    Fromdate.month;
    cout << "/";
    Fromdate.year;
    return Fromdate;
}
```

Writing & Reading Information

```
void Task::add_data()
{

    ofstream add;
    add.open("task.txt", ios::app);
    if (!add.is_open())
    {
        cout << "your file is not openes successfuuly" << endl;
    }
    else {
        cout << "File is open successfully" << endl;
        cout << "Enter your TaskID!" << endl;


        cin >> TaskID;
        cout << "Enter the Project Id!" << endl;
        cin >> Project_ID;
        cout << "Enter your Task title!" << endl;
        getline(cin, Title);
        cout << "Enter ur from date " << endl;
        cout << "Day(from date):";
        Date obj;
        cin >> obj.day;
        cout << "MOnth(From date): ";
        cin >> obj.month;
        cout << "Year(From date): ";
        cin >> obj.year;
        Set_Fromdate(obj);
        //todate
        cout << "Enter ur todate date " << endl;
        cout << "Day(todate):";
        Date obj_1;
        cin >> obj_1.day;
        cout << "MOnth(todate): ";
        cin >> obj_1.month;
        cout << "Year(todate): ";
        cin >> obj_1.year;
        Set_Todate(obj_1);
```

```
        cout << "Enter the max team_memmbers for this task" << endl;
        int y;
        cin >> y;
        for (int i = 0; i < y; i++)
        {
            cout << "Enter team member id " << i + 1;
            int w;
            cin >> w;
            Team_Members[i].set_ID(to_string(w));


        }
        cout << "enter your maxi resources!" << endl;
        int  r_no;
        cin >> r_no;

        for (int i = 0; i < r_no; i++)
        {
            cout << "Resource no" << " " << i + 1;
            int x;
            cin >> x;
            Resource[i].set_id(to_string(x));

        }
        cout << "Enter your status for task id" << endl;
        cin >> Status;

        add << TaskID << ' ' << Project_ID << ' ' << Title << ' ' << Fromdate.day << '/' << Fromdate.month << '/' << Fromdate.year << ' ' << Todate.day << ' ' << Todate.m

        add << '[';
        for (int i = 0; i < y; i++)
        {
            add << Team_Members[i].member_id << ' ,';


        }
        add << ']' << ' ' << '[';
        for (int i = 0; i < r_no; i++)
        {
```

The functions AddProject and ReadProject writes and read information respectively, from the file created named as myFile and myFile_1.

Editing & Deleting Information

```
void Task::edit_dat()
{
    fstream edit;
    edit.open("task.txt", ios::app, ios::in);
    if (!edit.is_open())
    {

        cout << "your file is not openes successfuuly" << endl;
    }
    else {


        cout << "File is open successfully" << endl;
        cout << "enter your task id to edit record:" << endl;
        int id_3;
        cin >> id_3;
        if (id_3 == TaskID)
        {
            cout << "Change the record in this sequence Project ID" << endl;
            cin >> Project_ID;
            cout << "Enter your Task title!" << endl;
            getline(cin, Title);
            cout << "Enter ur from date " << endl;
            cout << "Day(from date):";
            Date obj;
            cin >> obj.day;
            cout << "MOnth(From date): ";
            cin >> obj.month;
            cout << "Year(From date): ";
            cin >> obj.year;
            Set_Fromdate(obj);
            cout << "Enter ur todate date " << endl;
            cout << "Day(todate):";
            //todate
            Date obj_1;
            cin >> obj_1.day;
            cout << "MOnth(todate): ";
            cin >> obj_1.month;
            cout << "Year(todate): ";
```

```
void Task::delete_data()
{

    fstream myfile;
    myfile.open("task.txt", ios::app, ios::in);
    if (!myfile.is_open())
    {

        cout << "file is not open:" << endl;

    }
    else {



        cout << "file is open " << endl;
        cout << "plz enter ur taskk id to deleet the record" << endl;
        int id_4;
        cin >> id_4;
        if (TaskID == id_4)
        {
            Project_ID = 000;
            Title = "****";
            Status = "0000";
            Date obj;
            obj.day = 000;
            obj.month = 000;
            obj.year = 000;
            myfile << Project_ID << Title << Status << obj.day << obj.month << obj.year;
            cout << "Yor record is deleted" << endl;

        }
    }
    myfile.close();
}
```

The functions EditProject and DeleteProject are used for Eidting and deleting information respect ively.

Main Function

```
int main()
{
    int main()
    {
        int option;
        cout << "**************************************LOGIN IN PROJECT MANAGEMENT SYSTEM**************************************\n";
        cout << "\t\t1)login\n";
        cout << "\t\t2)Register\n";
        cout << "Enter your option\n";
        cin >> option;

        switch (option) {
        case 1:
        {
            string user_name;
            string password;
            ofstream myfile;
            cout << "Enter the username : ";
            cin.clear();
            cin >> user_name;
            cin.clear();
            cout << "Enter the password : \n";
            cin.ignore();

            getline(cin, password);
            cout << "\n";
            myfile.open("application.txt", ios::app);
            myfile << user_name << ' ' << password << "\n";
            cin.clear();
            cout << "YOU ARE REGIDTERED SUCCESSFULLY\n";
            break;
        }
        case 2:
        {
            string username;
            string password_1;
            cout << "\t\tENTER YOUR PASSWORD TO LOGIN IN THE PROJECT MANGEMENT SYSTEM\n";
            cout << "\t\t USERNAME : ";
            cin >> username;
            cout << "\t\tPassword : ";
            getline(cin, password_1);
            if (username == "a" || username == "z" || username == "A" || username == "Z")
                if (password_1 == "0" || password_1 == "9" || password_1 == "a" || "z")
                    cout << "Your PASSWORS AND USERNAME IS VALID\n";
            cout << "You ARE LOGIN SUCCESSFULLY\n";
            break;
```

The file handling of classes Resources and Project in class Project the member functions are void AddProject(),void ReadProject(),void EditProject(string op, string name); void DeleteProject(string op),void DisplayProject();

These are the declarations of member functions that is defined in Project.cpp file.Similarly the member functions of class Resources are void add_data(),

void edit_data(string o, string n), void delete_data(string n),void display_data().

These are the declaration of add ,edit,deleted and display member functions in a class Resources.

I make the main function of the whole Project and make interfaces of login for Class project, TeamMembers , Resources , Date ,Task ,for print report and define Task.

```
Project obj;
cin.clear();
cout << "\t\t**********************WELCOME TO PROJECT MANAGEMENT SYSTEM APPLICATION**********************\n";
cout << "\t\t      *****MENU*****\n";
cout << "\t\t      Enter 1 for Adding the Project ID,PROJECT Name\n ";
cout << "\t\t      Enter 2 for Editing the Project ID,PROJECT Name\n ";
cout << "\t\t      Enter 3 for Deleting the Project ID,PROJECT Name\n ";
cout << "\t\t      Enter 4 for Displaying the Project ID,PROJECT Name\n ";
cin.clear();
char p_n[20] = "Project Management";
Project obj1(1, p_n);
cin.clear();
obj1.GetID();
cin.clear();
obj1.SetID(1);
obj1.GetName();
cin.clear();
obj1.SetName(p_n);
cin.clear();
int n;
cout << "Enter your option\n";
cin >> n;
cin.clear();
switch (n)
{
case 1:

    obj1.AddProject();
    break;
case 2:
    obj1.EditProject();
    break;
case 3:
    obj1.DeleteProject();
    break;
case 4:
    obj1.DisplayProject();
    break;
case 5:
    obj1.DisplayProject();
default:
    cout << "nothing\n";

}
cin.clear();
```

CODE EXECUTION AT CONSOLE

```
                *******MENU*********
                 Enter 1 for Adding the Member ID,PROJECT Name
                 Enter 2 for Editing the MemeberID,PROJECT Name
                 Enter 3 for Deleting the Member ID,PROJECT Name
 Enter your Member Id: 1
nter your Name: Ali
nter your Email Address: asmabutt232gmail.com
nter your Addresss :house#1
nter your Phone Number :122345

              ***Enter your option***

ile created successfully
our record is added
Aliasmabutt232gmail.com122345
              **********WELCOME TO PROJECT MANAGEMENT SYSTEM APPLICATION**********
              *****MENU*****
              Enter 1 for Adding the Resource ID,Resource Name
              Enter 2 for Editing the Resource ID,Resource Name
              Enter 3 for Deleting the Resource ID,Resource Name
              Enter 4 for Displaying
              ***Enter your option***

nter your resources Id:1
 Enter your resource name: Laptop
our resource name is valid
ile is opened suucessfully
aptop1
              **********WELCOME TO PROJECT MANAGEMENT SYSTEM APPLICATION**********
              *****MENU*****
              Enter 1 for Adding the Project ID,PROJECT Name
              Enter 2 for Editing the Project ID,PROJECT Name
              Enter 3 for Deleting the Project ID,PROJECT Name
              ***Enter your option***
nter your option
```

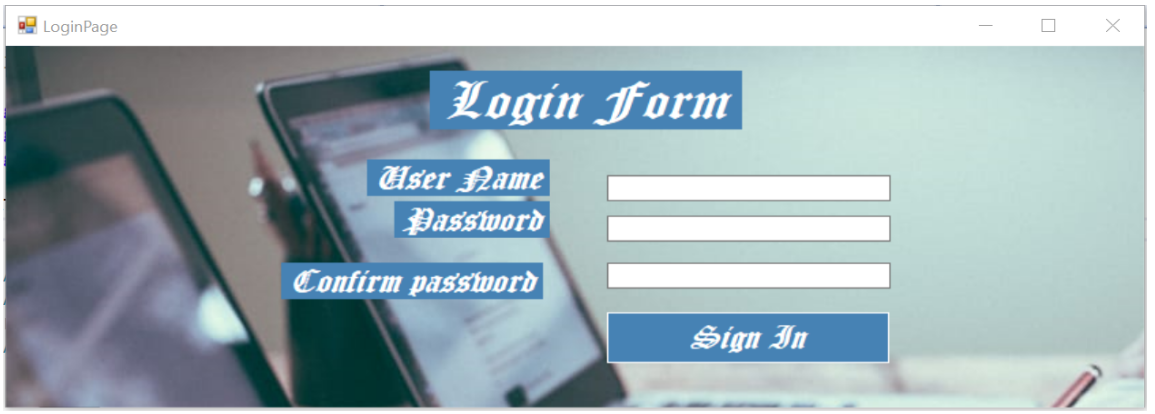
```
C:\Users\HP i5\source\repos\Project100\Debug\Project100.exe
n main**********LOGIN IN PROJECT MANAGEMENT SYSTEM**********
***
               1)login
               2)Register
nter your option
1
nter the username : Asma
nter the password : 1234

OU ARE REGIDTERED SUCCESSFULLY
              **********WELCOME TO PROJECT MANAGEMENT SYSTEM APPLICATION**********
                   *****MENU*****
                   Enter 1 for Adding the Project ID,PROJECT Name
                   Enter 2 for Editing the Project ID,PROJECT Name
                   Enter 3 for Deleting the Project ID,PROJECT Name
                   Enter 4 for Displaying the Project ID,PROJECT Name
 Enter your option
```

```
YOU ARE REGIDTERED SUCCESSFULLY
          **********WELCOME TO PROJECT MANAGEMENT SYSTEM APPLICATION**********
               *****MENU*****
               Enter 1 for Adding the Project ID,PROJECT Name
               Enter 2 for Editing the Project ID,PROJECT Name
               Enter 3 for Deleting the Project ID,PROJECT Name
               Enter 4 for Displaying the Project ID,PROJECT Name
 Enter your option
1
Enter your Project ID:1

Enter your Project name: Wedding
1Wedding
Your record is Added
```

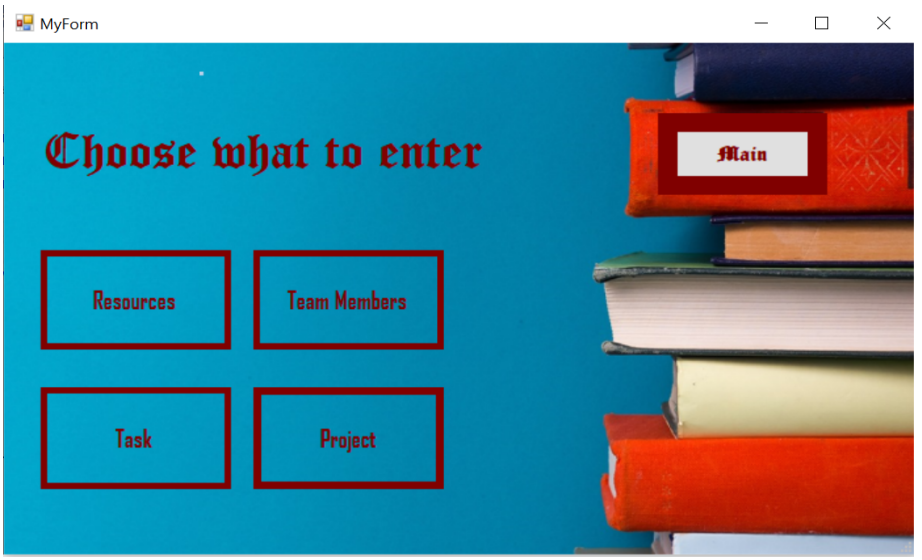
```
                *******MENU*********
                 Enter 1 for Adding the Member ID,PROJECT Name
                 Enter 2 for Editing the MemeberID,PROJECT Name
                 Enter 3 for Deleting the Member ID,PROJECT Name
 Enter your Member Id: 1
Enter your Name: ali
Enter your Email Address: asmabutt2322gmail.com
Enter your Addresss :houeno1
Enter your Phone Number :0300466

              ***Enter your option***
1
File created successfully
Your record is added
1aliasmabutt2322gmail.com300466
```

GRAPHICAL USER INTERFACE

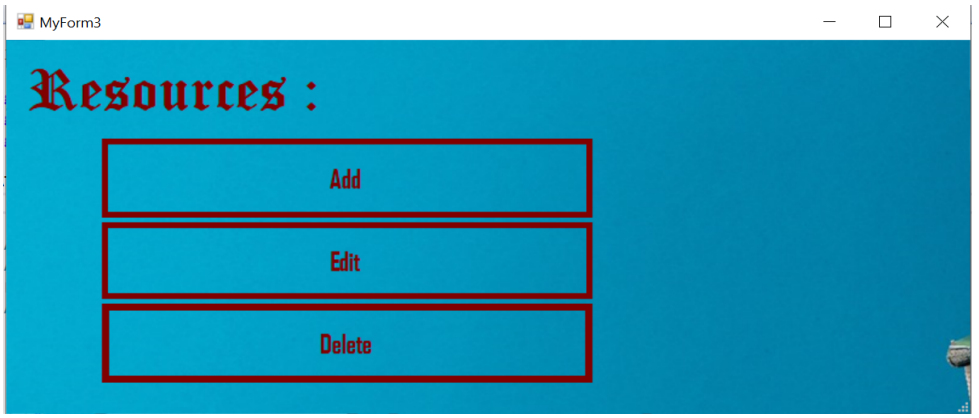
LOGIN PAGE



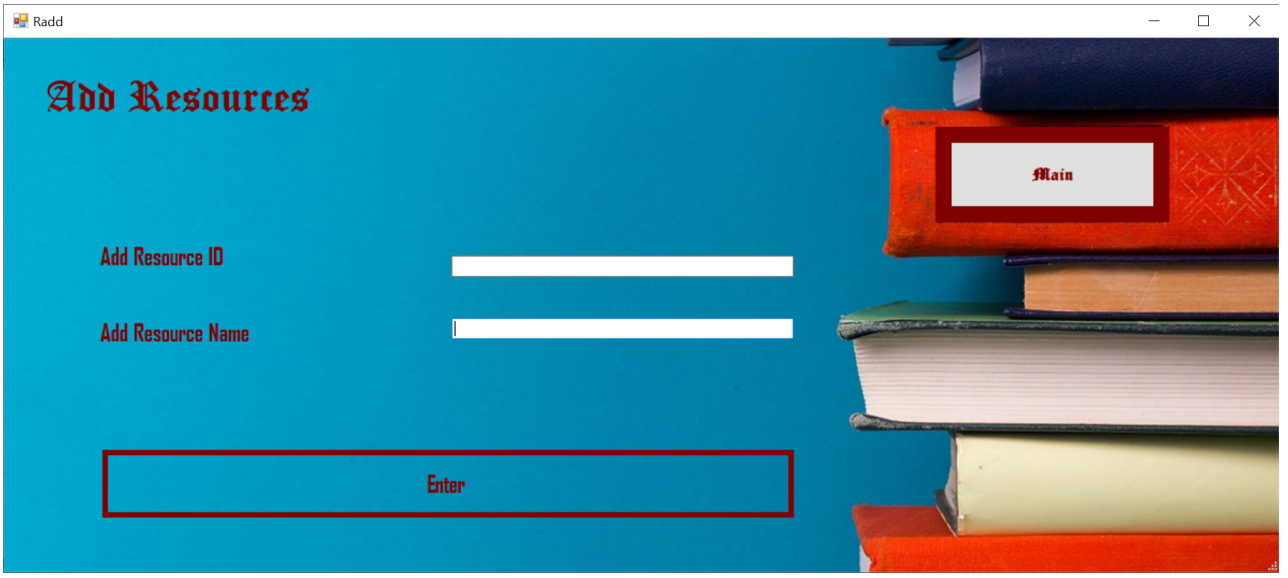
OPTIONS MENU



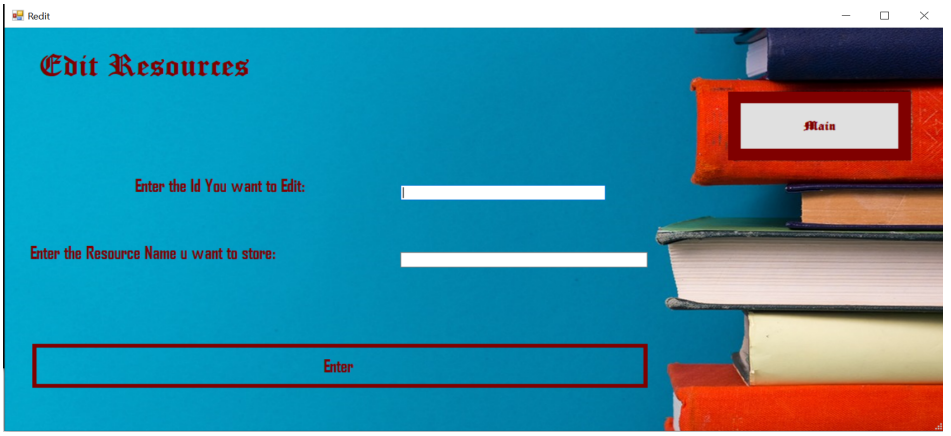
Resource Menu



Add Resouces

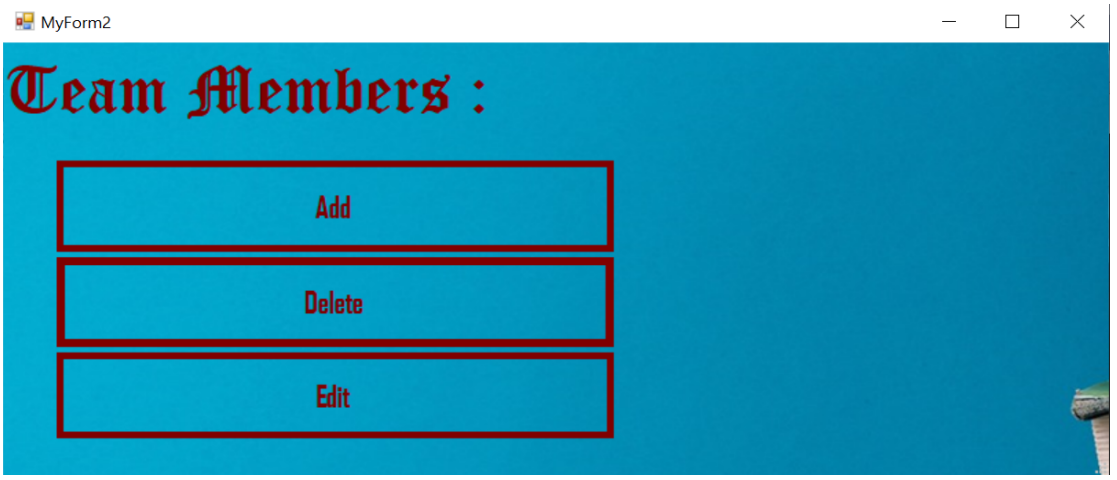


Edit Resources

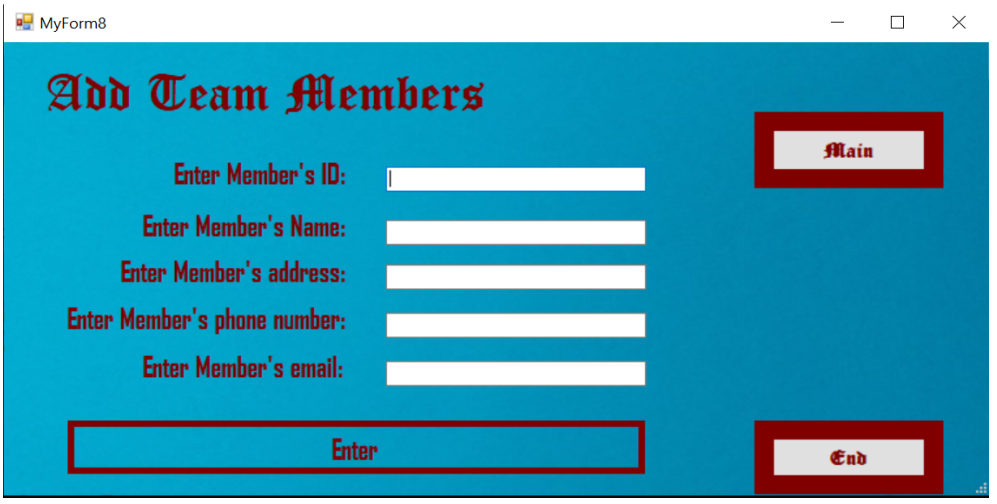


Delete Resources

Team Members Menu



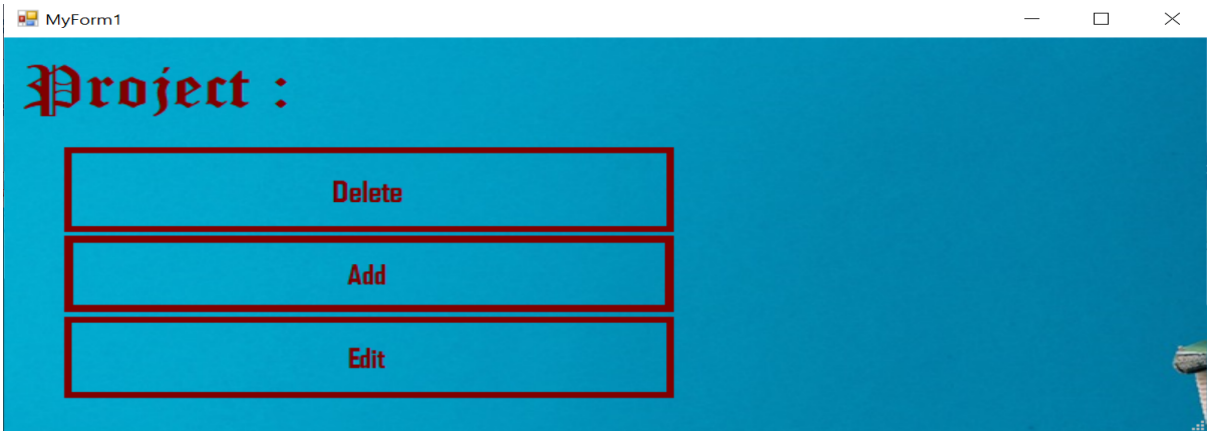
Add Team Members



Edit Team Members

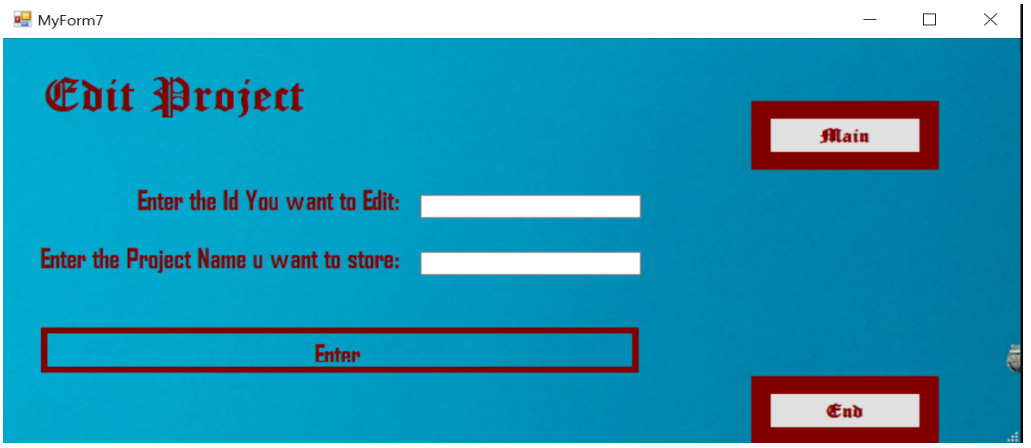
Delete Team Members

Project Menu

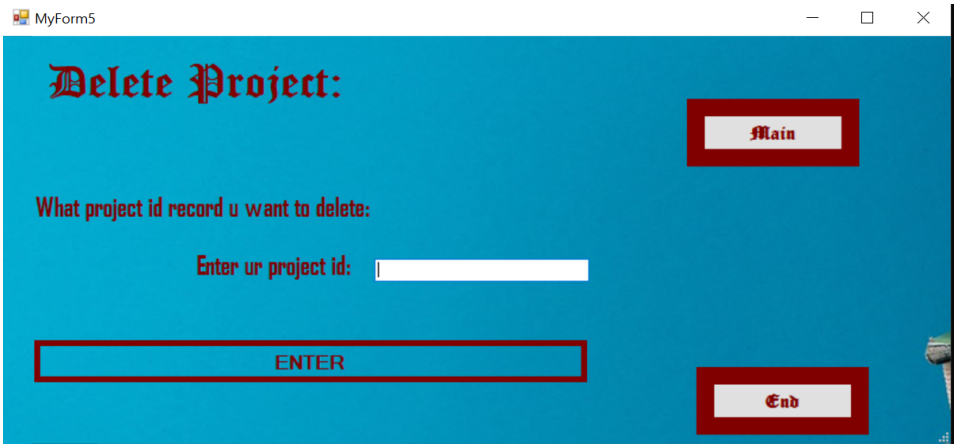


Add Project

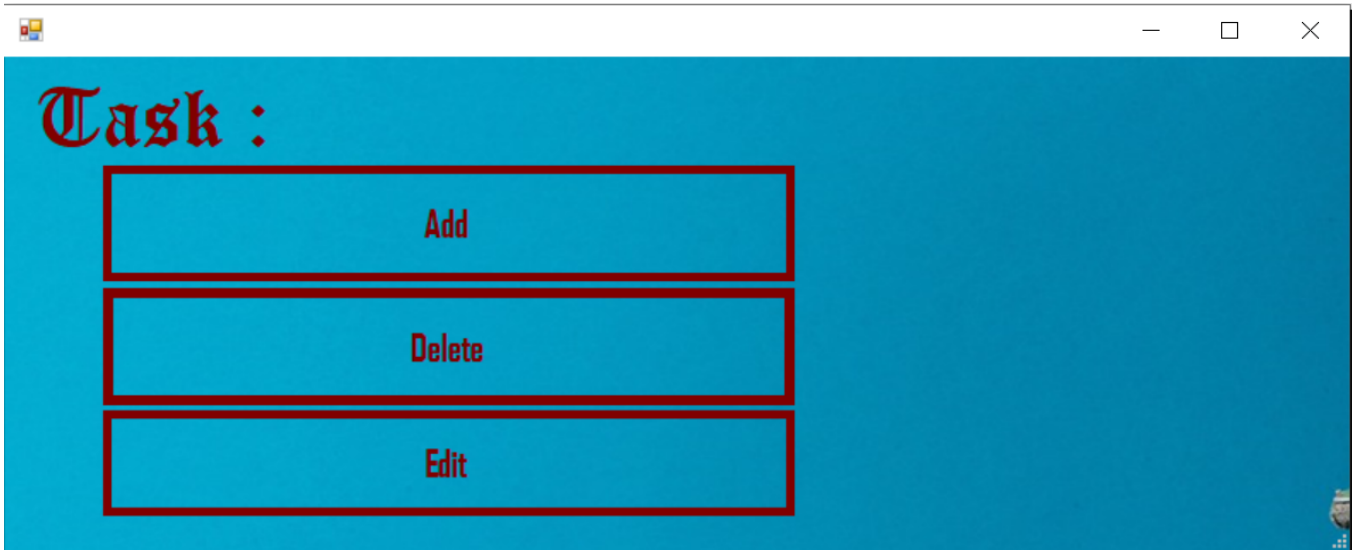
Edit Project



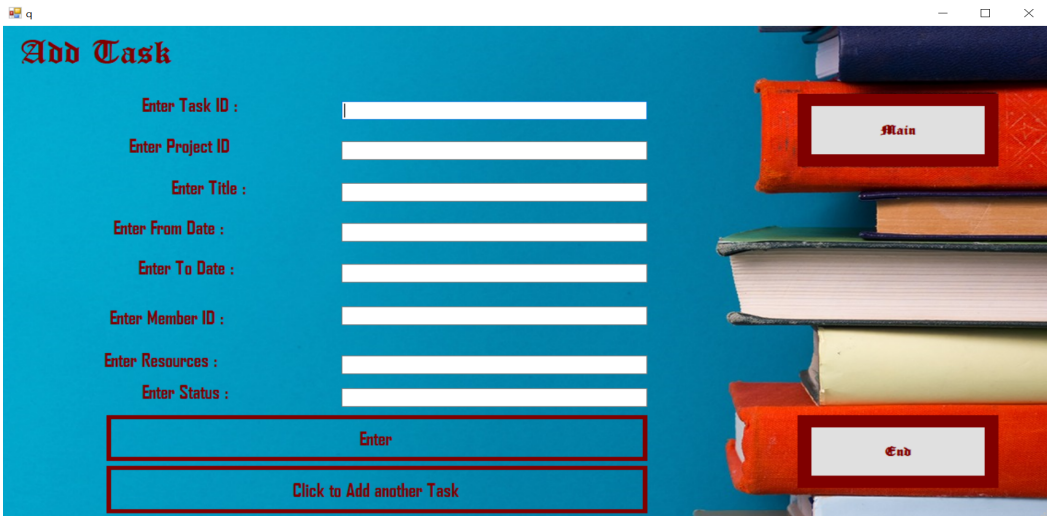
Delete Project



Task Menu

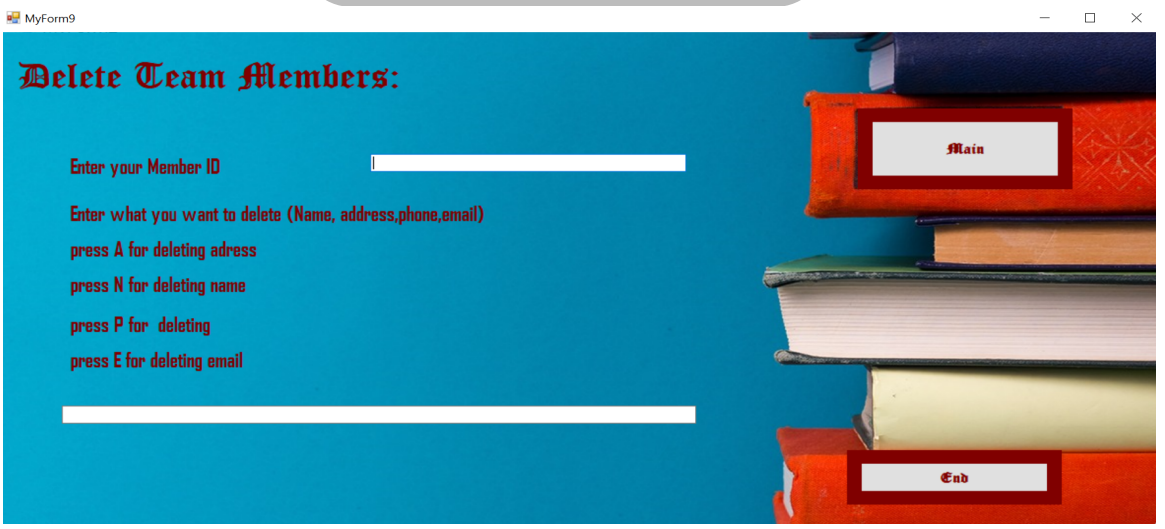


Add Task



Edit Task

Delete Task



Group Contribution Analysis

Asma Butt:

Explain the marking specifications of the project
Made the "Main Function"
Exceptions and error handling of project
File handling of classes "project" and "Resources"
Made GUI of Login Page

Noor Fatima:

Made UML of project
Explain the project to the team Members
Made classes "Task" and "Date" and also did their File Handling
Made GUI of entire project except the Login Page
Final Compilation of files
Errors and Exceptions Handling.

Kainat Jahan:

Made the class "Resources".
Good co-operation with team Members

Umar Razzaq:

Made the class "Project"
Made the documentation of class Project
Good co-operation with team Members

Uswah Saeed:

Made the class "Team Members"
Did file Handling of "Team Members" class
Compiled and Formatted whole Documentation
Co-operated with team members

Team Member's Reviews on learning and outcomes

Asma Butt:

I am able to understand the real world Project and its management.Now I have enough knowledge to lead any Project. Moreover, I will be able to understand the whole project concepts. I clear my more concepts about file handling I am able to identify different compiler exceptions,syntax errors and able to resolve these errors. Plus I learn the more concepts of Oops. I learn about Gui winform and make login form. Thanks to sir Usman Asghar .

Noor Fatima:

Firstly this Project helped me clear my concepts of file handling . Moreover i learned how to covert a class into headers and cpps . I also was able to undertsand and work on errors of the entire project and was also able to deal with all the exceptions , and Debug them. After completing the coding parts . I started learning GUI. Made the forms of all the classes. I did some changes in classes to integrate them with GUI And did the integration of Gui with classes as much as i could . At the end i was able to compile the project . I have tried to give my best efforts . But still im learning to be more accurate

Kainat Jahan:

As a group member, I worked on the "Resource Class".This is my individual contribution. In Project "Project Management System" I learn many concepts of OOP and apply different techniques in project with that's I was unfamiliar.I have learnt many new things during this project. I understand the importance of C++ coding . By this project I understand the clear picture of OOP and its use in Computer Science world and I have learnt that how OOP is used in daily problems. It was practical implementation of our study.
At the end i am grateful of my teachers,due to them now we are be able to solve programming problems.

Umar Razzaq:

In the individual contribution, as a group member, I worked on the "Project Class". In Project "Project Management System" I learn many new concepts of OOP and apply different techniques in project with that's I was unfamiliar. I understand the importance of C++ coding and also know the OOP Application in real life. By this project I understand the clear picture of OOP and its use in Computer Science world. It was practical implementation of our study.
Finally, I want to thanks our Professor's that we will be able to solve programming problems.

Uswah Saeed:

I was assigned the task of making class of "Team Members", before starting working on it I thoroughly revised all concepts of OOP, as I was facing much difficulty in implementation, I consulted books and Youtube videos, after that I started and completed the given task including file Handeling of my class within two days, futher I was assigned Documentation of Project, Iwhich I did on Illustrator.