**Mawlana Bhasani Science and Technology University Santosh,Tangail.**

**A Report on -**

***Student Database System***

**A project based on c programming language named “STUDENT DATABASE SYSTEM” is submitted to the department of ICT**

**SUPERVISED BY**

**MD ABIR HOSSAIN**

**LECTURER**

**DEPT OF ICT,**

**MBSTU.**

**SUBMITTED BY**

**SHARNA DAS**

**ID : IT-15007**

**SESSION : 2014-15**

**SHAKERA KHANDAKAR**

**ID : IT-15008**

**SESSION : 2014-15**

**ACKNOWLEDGEMENT :**

We are grateful to have the opportunity to express our heartiest and most sincere gratitude to a number of peoples who help us in courses of this project.

This project was performed under the supervision of Md. Abir Hossain, Lecturer, Department of Information and communication Technology (ICT), Mawlana Bhasani Science and Technology University, Santosh, Tangail. First and foremost, we wish to concede our profound indebtedness to him for guidance, valuable suggestions, encouragement and cordial cooperation.

We are also grateful to the rest of the teachers of our respective Department, Department of Information & Communication Technology for helping us in many ways.

Heartiest thanks to my Department of the University, Information & Communication Technology for providing me such an opportunity for the accomplishment of our project smoothly.

Lot of thanks to those who were the respondents of our project work & soulfully helped us a lot for the collection of data for our project.

Finally, we would like to express our gratitude and thankfulness to the development staff and our friends, who have inspired us in this project.

**PREFACE**

Special Thanks

We humbly give our special thanks to our beloved sir Md. Abir Hossain who helps us throughout the whole period in order to give shape our idea of the project .He gives his knowledge ,time , expertise and effort to help us in our project .He also gives us valuable suggestions to develop our idea of the project .We also give our gratitude to our dear sir to help us in our critical phase for succeeding in our project.

We finally thank some of our friends for their help and support.

**ABSTRACTION:**

The aim of the project is to create and manage the Student Database System of our department more efficiently. This project includes investigation , analysis ,design and development of the basic system of **“STUDENT DATABASE SYSTEM”.**

# **STUDENT DATABASE SYSTEM :**

The objective of our project is to create an easy and efficient system for the students to store and get the information about them through the student database system.

This system enables the students and also the teachers and the office workers to store valuable information about students such as their name, id, father’s name, mother’s name, phone number, address and the semester they are completing.

This system also has features of searching the information of the students based on their ids’ .Also one can delete their information and update their current information stored in the system. Because the system to be password protected data are secured than collecting them on random notebook. Comparing the way department use to store the students information, this “Student Database System” is more efficient.

**ADVANTAGES :**

* It’s more technical and efficient and can be developed further.
* Due to the system to store information no data can be lost.
* One can easily collect desirable information or can update or can delete data.
* Due to password word protection information is secured to the respective office worker.

CONTENTS

PREFACE

CHAPTER 1 INTRODUCTION………………………….. 06-07

1.1 Manual

1.2 Background

1.3 Project description

1.4 Main Objectives

CHAPTER 2 DESIGN AND IMPLEMENTATION ………08-24

2.1 Algorithm

2.1.1 Main Screen

2.1.2 Menu

2.1.3 Add

2.1.4 Search

2.1.5 Delete

2.1.6 Update

2.1.7 Log in and Log out

2.2 Implementation

2.2.1 C programming Language

2.2.2 Mouse programming

2.2.3 Graphics programming

2.3 Isnum,isalpha,isspace

2.4 File

CHAPTER 3 ANALYSIS THE RESULT OF STUDENT DATA BAGE SYSTEM…………………………………..25-30

**CHAPTER 4 CONCLUSION…………………………………….31**

**CHAPTER 5 SOURCE CODE…………………………………..31-51**

CHAPTER 1

* 1. **MANUAL**

More and more people who work in the computer industry find that their livelihood either already depends in some way on the database computing or it is progressing in that direction.

Basically, in every sector of our life we need to store records of huge volume .In order to express the systems man first various signs and insignias to preserve the information. They would keep accounting by using pens and notebook. Through the passage of time and men’s knowledge and thinking, they evolved letters and figures and they started using pen and papers. With the advancement of modern civilization the arena of men’s work got expanded and so also his busy life to keep more information in lesser effort and longer time the invention and use of machines and equipment got into his brain. Man produced computer system. Although man could save time, energy though calculators and could save labor and brain, yet man was not satisfied .Because of these machines cannot work even a moment without the help of a man.

In a big data processing Manual System, in one-side men’s intervention is resulting in a defective operation and on the other hand it kills times. To solve this problem, men started inventing the machine processing and through Automatic Systems .In this process,

In order to get the required result, some necessary intervention are to be recorded chronologically in the language of the machines and then the programming has to started

When the machine complete the work through stages. Computer plays a vital rule in the modem world in operating the Automatic System, News and views of mind can now be transmitted and stored in a moment’s time. We can have data processing now unbelievable speed.

**1.2 BACKGROUND**

We are now living in a new world. The name of this is “Information Age”. This new information world owns a close relationship with computer, Which is one of the most important parts of the technology. So computer should be used in all educational be equipment firstly and should be used in all educational purposes. To computerize an important issue like Tabulated System is a very useful task to be done with the current compatibility.

Currently the Hospital Management system of a university is not only time consuming but also laborious and including corruption. With the growing technological improvement the manual process of casting and counting vote are not beneficial. To keep up with the information

Revaluation of the world, computerizing Tabulated System is a must .We can think of Tabulated System as a database for counting and casting vote.

Though with the object oriented programming ,programming can achieve a tremendous success in making man’s life easy and advanced and comfortable.We use c programming as our programming language to complete our project. We tried to pull off our project to make the student database at primary stage i.e. to store various information. To keep pace with the modern world we tried to develop the so called student data base system using pen and paper into an advanced one by using c language. C language provides some excellent features.

These excellent characteristics provide all the facilities to develop software like ‘**Student Database System**’.

**1.3** **Project description**

The main goal of our project is to computerize the manual **Student Database System.**

We have done our project thinking about the complexity of storing information in paper wise. Our project is a efficient to solve the problem in some sort.

This project is to system a to include information, search information ,to delete and update information in a computerized way. Providing a password protection, th system is well secured in changing or deleting information without authorization. We can also collect the information as a printed paper and store them properly and more precisely.

**Main Objectives:**

The main objective of our project is to create an easy and efficient system for the students to store and get the information about them through the student database system.

This system enables the students and also the teachers and the office workers to store valuable information about students such as their name, id, father’s name, mother’s name, phone number, address and the semester they are completing. It’s more technical and efficient and can be developed further. Due to the system to store information no data can be lost. One can easily collect desirable information or can update or can delete data. Due to password word protection information is secured to the respective office worker. So the basic reason to create such a project is to give shape of the idea to make our department’s student database system more practical and advanced and efficient.

**CHAPTER 2**

**DEISGN TREE :**

MENU SCREEN

DETAILS ABOUT DEVELOPER

SHUT DOWN

ABOUT

CLOSE

MENU

DELETE

SEARCH

UPDATE

ADD

BACK TOMENU

TAKE INFORMATION

BACK TO MENU

BACK TO MENU

BACK TO MENU

IF LOGGED IN

LOGGED IN

IF ID MATCH THEN PRINT DETAILS FROM THE FILE CREATED IN ADD

IF ID MATCHES , REMOVE CORRESPONDING FILE FROM THE DIRECTORY

UPDATE THE CORREESDONDING FILE

**Figure : Design Tree of Student Database System(Working Features)**

To use the delete and update botton, one needs to log in first. The log in algorithm is given bellow:

LOG IN

TAKE PASSWORD

IF FALSE BACK TO MENU

IF CORRECT SET THE LOGGIN VARIABLE TO 1

There is a variable called login. If login is equal to 1 then it means that the user is logged in and if 0 then the person is logged out.

LOG OUT

LOG IN=0

MENU

**Figure : Log In and Log Out feature.**

**IMPLEMENTATION :**

To implement our idea of the project in the real life we used three sectors to make this project happen. We distributed this project into many catagories. We mainly use three features to give the real shape of our project. These are :

* C programming Language.
* Mouse programing.
* Graphics programing.

**C programming Language:**

**About C programming language :**

Nowadays C programming language is a buzzing word in software programming sector. Now most of the new commers and beginners use this language as a learning method .C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972.In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard.

The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C. C has now become a widely used professional language for various reasons. These are :

* Easy to learn
* Structured language
* It produces efficient programs
* It can handle low-level activities
* It can be compiled on a variety of computer platforms

**Facts about C :**

* C was invented to write an operating system called UNIX.
* C is a successor of B language which was introduced around the early 1970s.
* The language was formalized in 1988 by the American National Standard Institute (ANSI).
* The UNIX OS was totally written in C.
* Today C is the most widely used and popular System Programming Language.
* Most of the state-of-the-art software have been implemented using C.
* Today's most popular Linux OS and RDBMS MySQL have been written in C.

**Why use C?**

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C might be −

* Operating Systems
* Language Compilers
* Assemblers
* Text Editors
* Print Spoolers
* Network Drivers
* Modern Programs
* Databases
* Language Interpreters
* Utilities

A C program basically consists of the following parts and they are very much easy and understandable. They are:

* Preprocessor Commands
* Functions
* Variables
* Statements & Expressions
* Comments

## The C Compiler :

The source code written in source file is the human readable source for your program. It needs to be "compiled", into machine language so that your CPU can actually execute the program as per the instructions given.The compiler compiles the source codes into final executable programs. The most frequently used and free available compiler is the GNU C/C++ compiler, otherwise you can have compilers either from HP or Solaris if you have the respective operating systems.

The following section explains how to install GNU C/C++ compiler on various OS. We keep mentioning C/C++ together because GNU gcc compiler works for both C and C++ programming languages.

**Header files of C Language:**

A header file is a file with extension **.h** which contains C function declarations and macro definitions to be shared between several source files. There are two types of header files: the files that the programmer writes and the files that comes with the compiler. One request to use a header file in a program by including it with the C preprocessing directive **#include**, like we have seen inclusion of **stdio.h** header file, which comes along with the compiler.

Including a header file is equal to copying the content of the header file but we do not do it because it will be error-prone and it is not a good idea to copy the content of a header file in the source files, especially if we have multiple source files in a program.

**Mouse Programming:**

In GUI's like Windows, mouse is very important for user interaction. But in DOS, mouse will come in to picture only in some of the programs. For writing DOS games or graphics programs, one can add mouse functionality to the code.

  Mouse can be used in text mode as well as in graphics mode. Usually it is used in graphics mode. Hence we must first change over to graphics mode. In our program the function initgraph() is responsible for switching the mode from text to graphics .DETECT is a macro defined in 'graphics.h'. It requests initgraph() to automatically determine which graphics driver to load in order to switch to the highest resolution graphics mode. The initgraph() function takes three parameters, the graphics driver, the graphics mode and the path to the driver file.

           Once the driver has been loaded, initgraph() sets up the numeric values of the graphics mode chosen in the variables gd and gm respectively. Here we are assuming that the driver files are in the directory 'c:\tc\bgi'. Hence the path passed to initgraph() is 'c:\tc\bgi'.The various mouse functions can be accessed by setting up the AX register with different values (service number) and issuing interrupt number 51. The functions are listed bellow

|  |  |  |
| --- | --- | --- |
| 51 | 0 | Reset mouse and get status  Call with AX = 0  Returns:  AX = FFFFh If mouse support is available                Ax = 0 If mouse support is not available |
| 51 | 1 | Show mouse pointer   Call with AX = 1    Returns: Nothing |
| 51 | 2 | Hide mouse pointer  Call with AX = 2   Returns: Nothing |
| 51 | 3 | Get mouse position and button status   Call with AX = 3   Returns: BX = mouse button status  Bit   Significance   0     button not pressed   1     left button is pressed    2     right button is pressed   3     center button is pressed         CX = x coordinate  DX = y coordinate |
| 51 | 4 | Set mouse pointer position    Call with AX = 4  CX = x coordinate  DX = y coordinate   Returns: Nothing |
| 51 | 7 | Set horizontal limits for pointer  Call with AX = 7  CX = minimum x coordinate  DX = maximum x coordinate   Returns: Nothing |
| 51 | 8 | Set vertical limits for pointer  Call with AX = 8  CX = minimum y coordinate  DX = maximum y coordinate   Returns: Nothing |

* callmouse( )
* mouseposi( )
* mousehide( )
* setposi( )
* Horizontal ( )
* Vertical( )

     callmouse()  :-  In this function AX is set to "1". When this function is called in main() it displays the mouse pointer. The position of the pointer can be changed by using the mouse.

     mousehide() :-  In this function AX is set to "2".When this function is called in main() it hides the mouse pointer. This function is useful while drawing  figures, first the mouse pointer is kept hidden, then the figure is been drawn and again the mouse pointer is been called.

     mouseposi() :-  In this function AX is set to "3". This function returns the position of the mouse pointer. It contains three parameters,they are xpos,ypos,click. xpos and ypos returns the position of x co-ordinate and y co-ordinate respectively. Click is the integer variable which returns the values 1,2,3 corresponding to the button pressed on the mouse and 0 for buttons being not pressed. If any key is pressed kbhit returns nonzero integer; if not it returns zero.

      setposi() :-      In this function AX is set to "4". This function sets the mouse pointer to specific position . CX is been loaded by x co-ordinate of the mouse pointer and DX is been loaded with the y co-ordinate of the mouse pointer.

 Horizontal() :- In this function AX is set to "7". Its sets the horizontal barrier for the pointer which restricts the mouse pointer to pass that limit. CX is been loaded with the minimum x co-ordinate and Dx is been loaded with the maximum x co-ordinate.

     Vertical() :-     In this function AX is set to "8".Its sets the vertical barrier for the pointer which restricts the mouse pointer to pass that limit. CX is been loaded with the minimum y co-ordinate and Dx is been loaded with the maximum y co-ordinate.

C Graphics Programming :

Graphics programming contains lots of fundamental graphics program like drawing of various geometrical shapes(rectangle, circle eclipse etc), use of mathematical function in drawing curves, coloring an object with different colors and patterns and simple animation programs like jumping ball and moving cars.An overview of computer graphics and it's fundamentals.

The first step in any graphics program is to initialize the graphics drivers on the computer using**initgraph** method of graphics.h library.

***void initgraph(int \*graphicsDriver, int \*graphicsMode, char \*driverDirectoryPath);***

It initializes the graphics system by loading the passed graphics driver then changing the system into graphics mode. It also resets or initializes all graphics settings like color, palette, current position etc, to their default values. Below is the description of input parameters of initgraph function.

* **GraphicsDriver** : It is a pointer to an integer specifying the graphics driver to be used. It tells the compiler that what graphics driver to use or to automatically detect the drive. In all our programs we will use **DETECT** macro of graphics.h library that instruct compiler for auto detection of graphics driver.
* **GraphicsMode** : It is a pointer to an integer that specifies the graphics mode to be used. If \*graphdriver is set to DETECT, then initgraph sets \*graphmode to the highest resolution available for the detected driver.
* **DriverDirectoryPath** : It specifies the directory path where graphics driver files (BGI files) are located. If directory path is not provided, then it will seach for driver files in current working directory directory. In all our sample graphics programs, you have to change path of BGI directory accordingly where you turbo C compiler is installed.

**For Colors in C Graphics Programming**

There are 16 colors declared in C Graphics. We use colors to set the current drawing color, change the color of background, change the color of text, to color a closed shape etc. To specify a color, we can either use color constants like setcolor(RED), or their corresponding integer codes like setcolor(4). Below is the color code in increasing order.

| **COLOR MACRO** | **INTEGER VALUE** |
| --- | --- |
| BLACK | 0 |
| BLUE | 1 |
| GREEN | 2 |
| CYAN | 3 |
| RED | 4 |
| MAGENTA | 5 |
| BROWN | 6 |
| LIGHTGRAY | 7 |
| DARKGRAY | 8 |
| LIGHTBLUE | 9 |
| LIGHTGREEN | 10 |
| LIGHTCYAN | 11 |
| LIGHTRED | 12 |
| LIGHTMAGENTA | 13 |
| YELLOW | 14 |
| WHITE | 15 |

At the end of our graphics program, we have to unloads the graphics drivers and sets the screen back to text mode by calling **closegraph** function. Here is our first C Graphics program to draw a straight line on screen.

/\* C graphics program to draw a line \*/

#include<graphics.h>

#include<conio.h>

int main() {

int gd = DETECT, gm;

/\* initialization of graphic mode \*/

initgraph(&gd, &gm, "C:\\TC\\BGI");

line(100,100,200, 200);

getch();

closegraph();

return 0;

}

In this program **initgraph** function auto detects an appropriate graphics driver and sets graphics mode maximum possible screen resolution. Then **line** function draws a straight line from coordinate (100, 100) to (200, 200). Then we added a call to getch function to avoid instant termination of program as it waits for user to press any key. At last, we unloads the graphics drivers and sets the screen back to text mode by calling **closegraph** function.

We use this knowledge in our project as the result front page such as the color and the rectangular box designing.

Isnum,isalpha,isspace:  
The C library function **void isdigit(int c)** checks if the passed character is a decimal digit character.

Decimal digits are (numbers) − 0 1 2 3 4 5 6 7 8 9.

Declaration

Following is the declaration for isdigit() function.

int isdigit(int c);

Parameters

* **c** − This is the character to be checked.

Return Value

This function returns non-zero value if c is a digit, else it returns 0.

In C programming, isalpha() function checks whether a character is an alphabet(a to z and A-Z) or not. If a character passed to isalpha() is an alphabet, it returns non-zero integer if not it returns 0.

**Defined in Header file:** [<ctype.h>](http://www.programiz.com/c-programming/library-function/ctype.h)

## Function Prototype of isalpha()

int isalpha(int argument);

Function isalpha() takes single argument in form of an integer and returns integer value. Even though, isalpha() takes integer as an argument, character is passed to isalpha() function. When character is passed as an argument, corresponding [ASCII value](http://www.programiz.com/ascii-character-codes) of that character is passed instead of that character itself.

In C programming, isspace() checks whether a character is white-space character or not. If a character passed to isspace() is white-space character, it returns non-zero integer if not it returns 0.

White-space characters in C programming are:

| Character | Description |
| --- | --- |
| ' ' | Space |
| '\n' | Newline |
| '\t' | horizontal tab |
| '\v' | vertical tab |
| '\f' | form feed |
| '\r' | Carraige return |

Defined in Header file: [<ctype.h>](http://www.programiz.com/c-programming/library-function/ctype.h)

## Function Prototype of isspace()

int isspace(int argument):

Function isspace() takes single argument in form of an integer and returns integer value. Even though, isspace() takes integer as an argument, generally character is passed to isspace() function. When character is passed as an argument, corresponding [ASCII value](http://www.programiz.com/ascii-character-codes) of that character is passed instead of that character itself.

These functions implementation my code:

int isnum(const char \*s)

{

while (\*s)

{

if (!isalpha(\*s) && !isspace(\*s)) return 1;

\*s++;

}

return 0;

}

Graphics function:

In a C program, first step is to initialize the graphics drivers on the computer. This is done using the initgraph method provided in graphics.h library. In the next few pages we will discuss graphics.h library in details. Important functions in graphic.h library will be discussed in details and samples programs will be provided to show the power of C programming language.

We will restrict our discussion on Graphics in C Language to 16 bit C programming and MS DOS environment and 640×480 VGA monitor.

Graphics mode Initialization

First of all we have to call the initgraph function that will initialize the graphics mode on the computer. initigraph has the following prototype.

|  |  |
| --- | --- |
| 1 | void initgraph(int far \*graphdriver, int far \*graphmode, char far \*pathtodriver); |

Initgraph initializes the graphics system by loading the graphics driver from disk (or validating a registered driver) then putting the system into graphics mode. Initgraph also resets all graphics settings (color, palette, current position, viewport, etc.) to their defaults, then resets graphresult to 0.

\*graphdriver

Integer that specifies the graphics driver to be used. You can give graphdriver a value using a constant of the graphics\_drivers enumeration type whcih is listed in graphics.h. Normally we use value as “0” (requests auto-detect). Other values are 1 to 10 and description of each enumeration type is [listed here](http://www.cs.colorado.edu/~main/bgi/doc/detectgraph.html).

\*graphmode

Integer that specifies the initial graphics mode (unless \*graphdriver = DETECT). If \*graphdriver = DETECT, initgraph sets \*graphmode to the highest resolution available for the detected driver. You can give \*graphmode a value using a constant of the graphics\_modes enumeration type and description of each enumeration type is [listed here](http://www.cs.colorado.edu/~main/bgi/doc/detectgraph.html).

\*pathtodriver

Specifies the directory path where initgraph looks for graphics drivers (\*.BGI) first.

1. If they’re not there, initgraph looks in the current directory.
2. If pathtodriver is null, the driver files must be in the current directory.

\*graphdriver and \*graphmode must be set to valid graphics\_drivers and graphics\_mode values or you’ll get unpredictable results. (The exception is graphdriver = DETECT.)

After a call to initgraph, \*graphdriver is set to the current graphics driver, and \*graphmode is set to the current graphics mode. You can tell initgraph to use a particular graphics driver and mode, or to auto detect the attached video adapter at run time and pick the corresponding driver. If you tell initgraph to auto detect, it calls detectgraph to select a graphics driver and mode.

Normally, initgraph loads a graphics driver by allocating memory for the driver (through \_graphgetmem), then loading the appropriate .BGI file from disk. As an alternative to this dynamic loading scheme, you can link a graphics driver file (or several of them) directly into your executable program file.

Here is some examples of the functions with instance of our project :

Graphics in my code:

// DataBase();;

int gd = DETECT , gm ;

initgraph(&gd,&gm,"c:/tc/bgi");

int x , y , c , f = 0;

while(!kbhit())

{

cleardevice();

settextstyle(10, HORIZ\_DIR,1);

outtextxy(80,30,"Welcome to MBSTU Student DataBase");

settextstyle(0, HORIZ\_DIR, 1);

rectangle(200,100,400,150);

//settextstyle(DEFAULT\_FONT, DEFAULT\_FONT, 1);

outtextxy(280,120,"Menue");

rectangle(200,200,400,250);

outtextxy(280,220,"Close");

rectangle(200,300,400,350);

outtextxy(280,320,"About");

//rectangle(200,400,400,450);

//outtextxy(280,420,"UPDATE");

cm();

}

File in C:

A **file** represents a sequence of bytes on the disk where a group of related data is stored. File is created for permanent storage of data. It is a ready made structure.

In C language, we use a structure **pointer of file type** to declare a file.

C provides a number of functions that helps to perform basic file operations. Following are the functions,

|  |  |
| --- | --- |
| **Function** | **Description** |
| fopen() | create a new file or open a existing file |
| fclose() | closes a file |
| getc() | reads a character from a file |
| putc() | writes a character to a file |
| fscanf() | reads a set of data from a file |
| fprintf() | writes a set of data to a file |
| getw() | reads a integer from a file |
| putw() | writes a integer to a file |
| fseek() | set the position to desire point |
| ftell() | gives current position in the file |
| rewind() | set the position to the begining point |

\*fp = FILE **\*fopen**(const char *\*filename*, const char *\*mode*);

Here **filename** is the name of the file to be opened and **mode** specifies the purpose of opening the file. Mode can be of following types,

**\*fp** is the FILE pointer (FILE \*fp), which will hold the reference to the opened(or created) file.

|  |  |
| --- | --- |
| **mode** | **Description** |
| r | opens a text file in reading mode |
| w | opens or create a text file in writing mode. |
| a | opens a text file in append mode |
| r+ | opens a text file in both reading and writing mode |
| w+ | opens a text file in both reading and writing mode |
| a+ | opens a text file in both reading and writing mode |
| rb | opens a binary file in reading mode |
| wb | opens or create a binary file in writing mode |
| ab | opens a binary file in append mode |
| rb+ | opens a binary file in both reading and writing mode |
| wb+ | opens a binary file in both reading and writing mode |
| ab+ | opens a binary file in both reading and writing mode |

#### Closing a File

The fclose() function is used to close an already opened file.

Here fclose() function closes the file and returns **zero** on success, or **EOF** if there is an error in closing the file. This **EOF** is a constant defined in the header file **stdio.h**.

File use in my code:

/////////////////

if( gets(file\_name))

flag = 1 ;

if(!fopen(file\_name,"r") && flag==1)

if(gets(file\_name)) flag = 1 ;

// if file is open

if(fopen(file\_name,"r") && flag ==1){

// removing the file from DATABASE\_DIRECTORY

cleardevice();

std tempo ; // temporary structure

char tem[100] ;// temporary string

fp = fopen(file\_name,"r");

fscanf(fp,"%s\n",tem); // id

fscanf(fp,"%s\n",tem); // name

fscanf(fp,"%[^\n]\n",tem); // fathers name

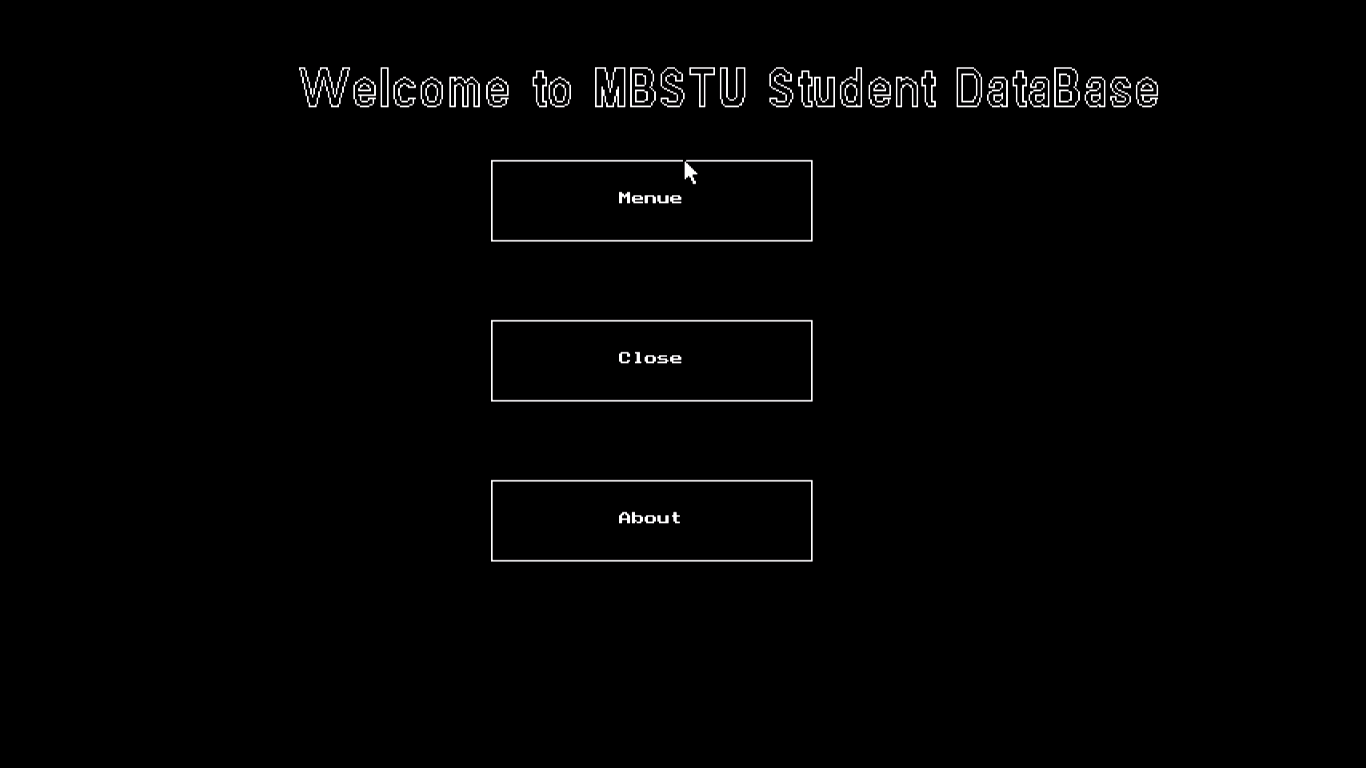
//printf("%s\n",tem);

fclose(fp);}

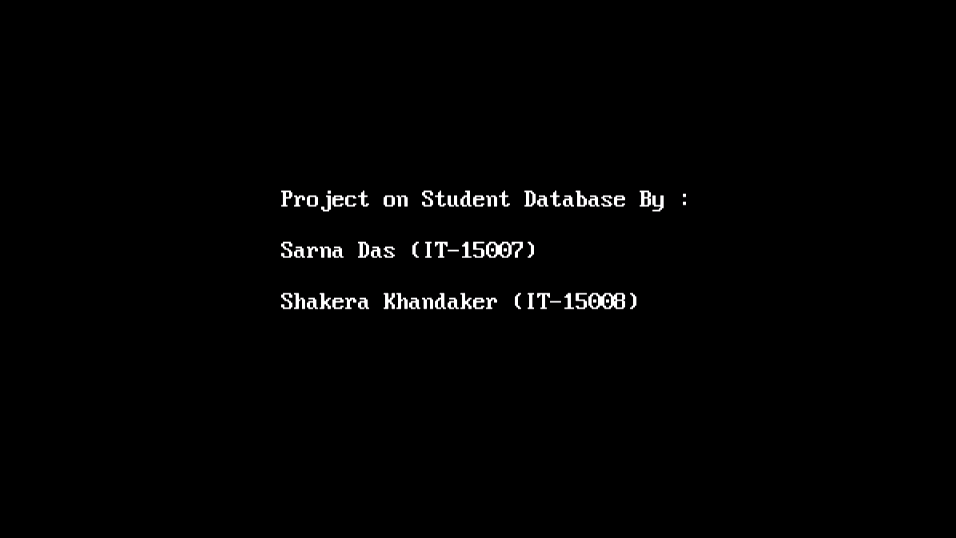
CHAPTER 3

**OUTPUT :**

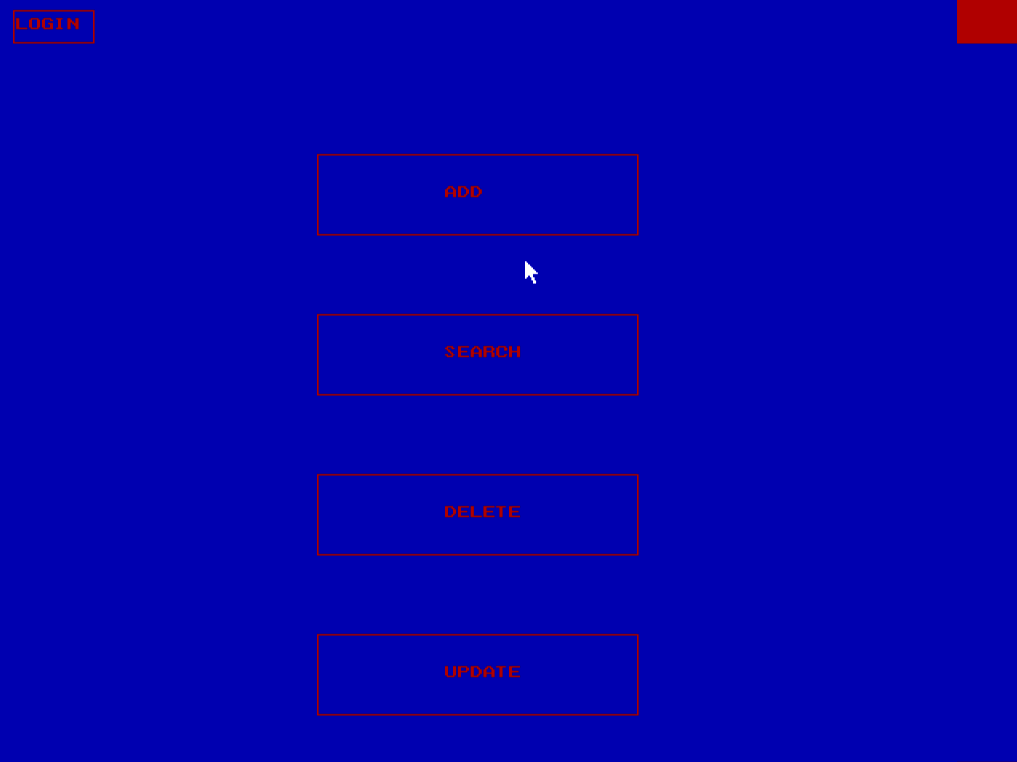
When we run our program , the MENU SCREEN is like



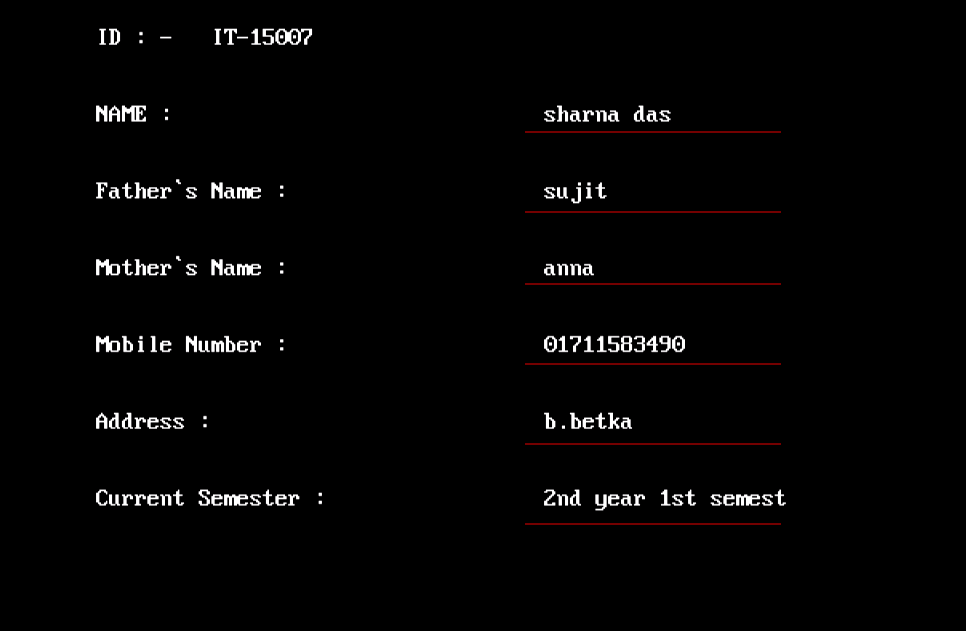
Then if anyone click the close button then the program will close and if anyone press About button then it will appear like



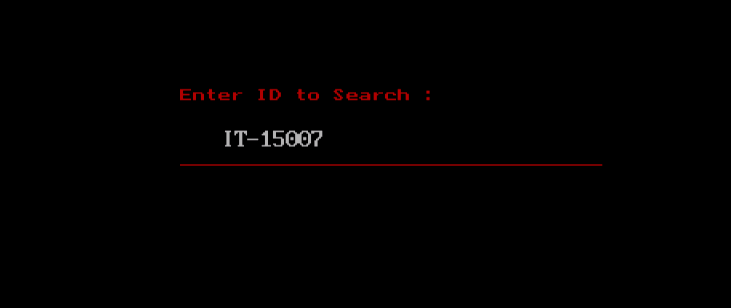
Again if anyone press the menu button the MENU will appear on the screen like



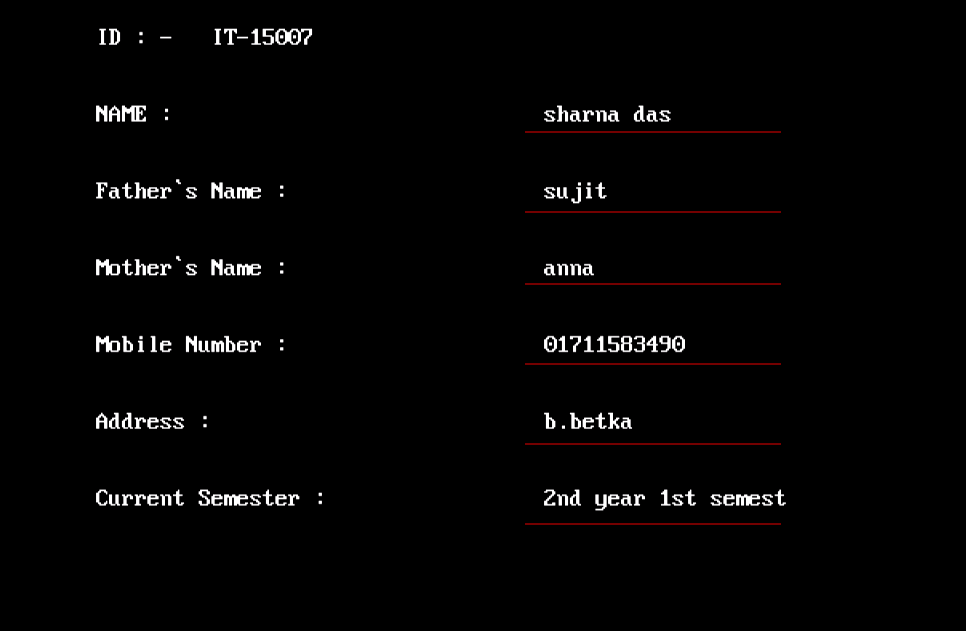
Now this four buttons work as four doors to perform four application. If anyone press ADD then that person can add information .



If you search your info with your ID then



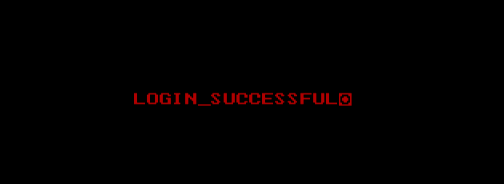
You can see the information you entered like



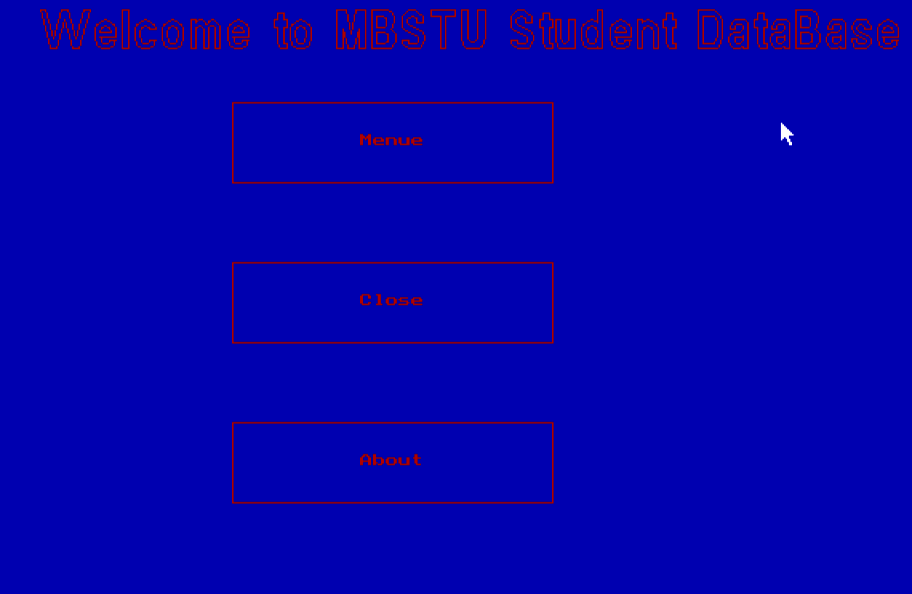
If anyone wishes to delete or update his/hers info then he/she have to log in first with a classified password like







Then finally anyone closes the cross button it will appear like



Then by pressing esc or the the close button one can exit from the program.

**CHAPTER 4**

**CONCLUSION :**

**FUTURE WORK :**

This project includes features like data storing ,searching ,updating and deleting but this project had another aspect to include image of every student as the part of their information. Due to the lack of knowledge in other language we were unable to attach image as the data .But it is our aim to create a developed version of this “Student Database System” including the image feature and other higher version of sectors we use in the system by using higher level programming language.

**REFERENCE :**

To complete our project we get help from online and other sources. Such as :Cpp.com(for some of the string functions)(n.d).Cora.com(for some file oriented programming)(n.d).Stack overflow(https://www.google.com.bd/?gws\_rd=cr&ei=q1vpVKtBqbL6ATTiIHwDA#q=stack+overflow)(04,2016)(its very usable for programming oriented problems).Graphics-in-c-lab-manual-chapter-rab-nawaz-jadoon.pdf.(03,2016)<http://www.techcrashcourse.com/2015/08/c-graphics-programming-tutorial.html> and <http://www.techcrashcourse.com/2015/08/c-program-draw-rectangle-bar-graphics.html>for rectangular design.And also some special help from our fellow friend.

**CHAPTER 5**

**SOURCE CODE:**

/\* project : student data base \*/

#include<stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<ctype.h> //for atoi() / itoa()

#include<conio.h>

#include<graphics.h>

#include<string.h>

#define PASSWORD 54321

struct student

{

char id[10];

char name[20];

char mobile\_number[13];

char fathers\_name[20];

char mothers\_name[20] ;

char address[20];

char semester[50];

} ;

typedef struct student std ;

union REGS i , o ;

void loadingAni();

// to check if a string is number of not ref stack overflow

int isnum(const char \*s)

{

while (\*s)

{

if (!isalpha(\*s) && !isspace(\*s)) return 1;

\*s++;

}

return 0;

}

// call mouse

void cm() ;

// mouse position

void mp(int &x , int &y ,int & c ) ;

void DataBase();

void WelcomePage();

/\*

void main()

{

//DataBase();

WelcomePage();

}

\*/

void main()

{

// DataBase();;

int gd = DETECT , gm ;

initgraph(&gd,&gm,"c:/tc/bgi");

int x , y , c , f = 0;

while(!kbhit())

{

cleardevice();

settextstyle(10, HORIZ\_DIR,1);

outtextxy(80,30,"Welcome to MBSTU Student DataBase");

settextstyle(0, HORIZ\_DIR, 1);

rectangle(200,100,400,150);

//settextstyle(DEFAULT\_FONT, DEFAULT\_FONT, 1);

outtextxy(280,120,"Menue");

rectangle(200,200,400,250);

outtextxy(280,220,"Close");

rectangle(200,300,400,350);

outtextxy(280,320,"About");

//rectangle(200,400,400,450);

//outtextxy(280,420,"UPDATE");

cm();

while(!kbhit())

{

gotoxy(5,5);

//printf("x = %d y = %d ",x,y);

mp(x,y,c);

// menue

if(c == 1 && x >200 && y >100 && x <400 && y <150)

{

c = 0 ;

cleardevice();

delay(200);

loadingAni();

DataBase();

break ;

}

// close

if(c == 1 && x >200 && y >200 && x <400 && y <250)

{

cleardevice();

closegraph();

return ;

}

// about

if(c == 1 && x >200 && y >300 && x <400 && y <350)

{

cleardevice();

gotoxy(20,10);

printf("Projectct on Student Database By : ") ;

gotoxy(20,12);

printf("Sarna Das (IT-15007)");

gotoxy(20,14);

printf("Shakera Khandaker (IT-15008)\n");

getch();

break ;

}

// if(c == 1 && x > && y > && x < && y <){

// }

}

}

// return ;

}

void DataBase()

{

// mouse vars

int x , y , c ;

int out = 0 ;

FILE \*fp ,\*fp2, \*fp3;

char file\_name[20] ;

int esc = 0 ;

int login = 0 ;

int pass = 0 ;

int t = 0 ; // temporary variable

int flag = 0 ;

// declaring graphics driver

//and graphics mode

int gd = DETECT, gm ;

int need = 0 ;

// setting the graphics

initgraph(&gd,&gm,"c:/tc/bgi");

while(out==0)

{

// gui

// prime buttons

cleardevice();

// setbkcolor(BLACK);

cm();

setbkcolor(BLUE);

setcolor(RED);

if(login==0)

{

rectangle(10,10,60,30);

outtextxy(12,15,"LOGIN");

}

else

{

rectangle(10,10,60,30);

outtextxy(12,15,"LOG OUT");

}

//verride

/\*rectangle(200,50,400,100);

outtextxy(280,70,"ADD");

rectangle(200,150,400,200);

outtextxy(280,170,"SEARCH");

rectangle(200,250,400,300);

outtextxy(280,270,"DELETE");

rectangle(200,350,400,400);

outtextxy(280,370,"UPDATE");

\*/

// the title

settextstyle(10, HORIZ\_DIR,1);

//outtextxy(80,30,"Welcome to MBSTU Student DataBase");

rectangle(200,100,400,150);

settextstyle(0, HORIZ\_DIR, 1);

//settextstyle(DEFAULT\_FONT, DEFAULT\_FONT, 1);

outtextxy(280,120,"ADD");

rectangle(200,200,400,250);

outtextxy(280,220,"SEARCH");

rectangle(200,300,400,350);

outtextxy(280,320,"DELETE");

rectangle(200,400,400,450);

outtextxy(280,420,"UPDATE");

// exit button

setfillstyle(SOLID\_FILL,RED);

bar(600,0,650,30);

textcolor(BLACK);

outtextxy(605,0,"X");

/\*

rectangle(600,10,625,20);

outtextxy(610,12,"X");

\*/

//setcolor(BLACK);

// background

while(1)

{

mp(x,y,c);

//logout

if(login==1)

{

if(c==1 && x>=10 && x<=60 && y>=10 && y<=30 )

{

setbkcolor(BLACK);

cleardevice();

//outtextxy(100,100,"LOGGED\_OUT");

esc = -1 ;

//getch();

break ;

}

}

// dew to unusual error

if(esc==-1)

{

esc = 0 ;

login = 0 ; // logged out

break ;

}

// login

if(login==0)

{

if(c==1 && x>=10 && x<=60 && y>=10 && y<=30 )

{

setbkcolor(BLACK);

clrscr();

cleardevice();

outtextxy(230,200,"ENTER\_PASS\_WORD : ");

outtextxy(230,240,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(30,15);

//outtextxy(100,100,"ENTER\_PASS\_WORD : ");

scanf("%d",&pass);

if(pass==PASSWORD)

{

cleardevice();

login = 1 ; // logged in

outtextxy(230,200,"LOGIN\_SUCCESSFUL\n");

getch();

break ;

}

else

{

cleardevice();

outtextxy(100,100,"WRONG\_PASSWORD");

getch();

break ;

}

}

}

//update

if(c==1 && x>=200 && x<=400 && y>=400 && y<=450 )

{

setbkcolor(BLACK);

if(login==0)

{

clrscr();

cleardevice();

outtextxy(230,200,"You Have To Log In First ");

getch();

break ;

}

file\_name[0] = '\0' ;

delay(30) ;

cleardevice();

outtextxy(230,200,"Enter ID to Update : ");

outtextxy(230,240,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(30,15);

if( gets(file\_name))

flag = 1 ;

if(!fopen(file\_name,"r") && flag==1)

{

clrscr();

cleardevice();

outtextxy(230,200,"ID\_DOES\_NOT\_EXIST\_ON\_DATABASE\n");

getch();

flag = 0 ;

break ;

}

else

{

clrscr();

cleardevice();

std temp ;

printf("%25sFill up details : \n","");

// own name

gotoxy(5,5);

printf("NAME : ");

outtextxy(300,75,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// gotoxy(13,5);

// scanf("%s",ch[20]);

gotoxy(5,8);

printf("Father`s Name : ");

outtextxy(300,125,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// scanf("%s",ch[20]);

gotoxy(5,11);

printf("Mother`s Name :");

outtextxy(300,170,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,14);

printf("Mobile Number :");

outtextxy(300,220,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,17);

printf("Address : ");

outtextxy(300,270,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,20);

printf("Current Semester :");

outtextxy(300,320,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// scanning part

// name

gotoxy(40,5);

// gets(file\_name); // for none logic not found

gets(temp.name);

while(isnum(temp.name) || strlen(temp.name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,5);

if( strlen(temp.name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,5);

printf(" ");

gotoxy(40,5);

gets(temp.name);

}

// if input is number then

strcat(temp.name,"\n");

// changing colout if pressed enter

setcolor(GREEN);

outtextxy(300,75,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// faters name

gotoxy(40,8);

gets(temp.fathers\_name);

while(isnum(temp.fathers\_name) || strlen(temp.fathers\_name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,8);

if( strlen(temp.fathers\_name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,8);

printf(" ");

gotoxy(40,8);

gets(temp.fathers\_name);

}

strcat(temp.fathers\_name,"\n");

setcolor(GREEN);

outtextxy(300,125,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// mothers\_name

gotoxy(40,11);

gets(temp.mothers\_name);

while(isnum(temp.mothers\_name) || strlen(temp.mothers\_name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,11);

if( strlen(temp.mothers\_name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,11);

printf(" ");

gotoxy(40,11);

gets(temp.mothers\_name);

}

strcat(temp.mothers\_name,"\n");

setcolor(GREEN);

outtextxy(300,170,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//mobile\_number

gotoxy(40,14);

gets(temp.mobile\_number);

// in case mobile number is too short

while(1){

if(isnum(temp.mobile\_number) && strlen(temp.mobile\_number)==11){

break ;

}else{

gotoxy(40,14);

printf("Has To Be 11 Digits");

getch();

gotoxy(40,14);

printf(" ");

gotoxy(40,14);

gets(temp.mobile\_number);

}

}

strcat(temp.mobile\_number,"\n");

setcolor(GREEN);

outtextxy(300,220,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// address

gotoxy(40,17);

gets(temp.address);

strcat(temp.address,"\n");

setcolor(GREEN);

outtextxy(300,270,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//semestr

gotoxy(40,20);

gets(temp.semester);

strcat(temp.semester,"\n");

setcolor(GREEN);

outtextxy(300,320,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// writing the collected data to file

// strcat(file\_name);

fp = fopen(file\_name,"w");

fprintf(fp,temp.id);

fprintf(fp,temp.name);

fprintf(fp,temp.fathers\_name);

fprintf(fp,temp.mothers\_name);

fprintf(fp,temp.mobile\_number);

fprintf(fp,temp.address);

fprintf(fp,temp.semester);

gotoxy(40,25);

printf("PRESS\_ENTER\_TO\_SAVE");

getch();

fclose(fp) ;

break ;

}

}

//delete button

if(c==1 && x>=200 && x<=400 && y>=300 && y<=350 && login==0) // exeption not logged in

{

clrscr();

cleardevice();

setbkcolor(BLACK);

outtextxy(230,200,"Log In To Delet \n");

getch();

break ;

}

else if(c==1 && x>=200 && x<=400 && y>=300 && y<=350 && login==1)

{

cleardevice() ;

delay(200);

setbkcolor(BLACK);

cleardevice();

file\_name[0] = '\0' ;

delay(30) ;

outtextxy(230,200,"Enter ID to Delet : ");

outtextxy(230,240,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(30,15);

flag = 0 ;

if(gets(file\_name)) flag = 1 ;

// if file is open

if(fopen(file\_name,"r") && flag ==1 )

{

// removing the file from DATABASE\_DIRECTORY

cleardevice();

std tempo ; // temporary structure

char tem[100] ;// temporary string

fp = fopen(file\_name,"r");

fscanf(fp,"%s\n",tem); // id

fscanf(fp,"%s\n",tem); // name

fscanf(fp,"%[^\n]\n",tem); // fathers name

//printf("%s\n",tem);

fclose(fp);

outtextxy(230,200,"REMOVING\_FILE\_PLEASE\_WAIT....\n");

delay(5000);

remove(file\_name);

remove(tem); // fathers name file removed

cleardevice();

outtextxy(230,200,"DATA\_HAS\_BEEN\_FORMATTED\n");

getch();

break ;

// flag = 0 ;

}

else // file is not open

{

cleardevice();

outtextxy(230,200,"ID\_DOES\_NOT\_EXIST\_ON\_DATABASE\n");

if(need==0){

need = 1 ;

}else

getch();

break ;

}

}

// search button

if(c==1 && x>=200 && x<=400 && y>=200 && y<=250 )

{

setbkcolor(BLACK);

clrscr();

cleardevice();

outtextxy(230,200,"Enter ID to Search : ");

outtextxy(230,240,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(30,15);

gets(file\_name);

//scanf("%s",file\_name);

if(fopen(file\_name,"r")==0)

{

gotoxy(5,5);

clrscr();

cleardevice();

outtextxy(30,15,"NO\_ENTRY\_FOUND\_IN\_DATABASE\n");

getch();

break ;

}

else

{

cleardevice();

fp = fopen(file\_name,"r");

char t[20] ;

fgets(t,10,fp);

gotoxy(5,2);

printf("ID : - %s \n",t);

gotoxy(5,5);

printf("NAME : ");

outtextxy(300,75,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// gotoxy(13,5);

// scanf("%s",ch[20]);

gotoxy(5,8);

printf("Father`s Name : ");

outtextxy(300,125,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// scanf("%s",ch[20]);

gotoxy(5,11);

printf("Mother`s Name :");

outtextxy(300,170,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,14);

printf("Mobile Number :");

outtextxy(300,220,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,17);

printf("Address : ");

outtextxy(300,270,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,20);

printf("Current Semester :");

outtextxy(300,320,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

////

// name

gotoxy(40,5);

fgets(t,20,fp);

printf("%s",t);

// faters name

gotoxy(40,8);

fgets(t,20,fp);

printf("%s",t);

// mothers\_name

gotoxy(40,11);

fgets(t,20,fp);

printf("%s",t);

//mobile\_number

gotoxy(40,14);

fgets(t,20,fp);

printf("%s",t);

// address

gotoxy(40,17);

fgets(t,20,fp);

printf("%s",t);

//semestr

gotoxy(40,20);

fgets(t,20,fp);

printf("%s",t);

getch();

break ;

}

}

// add button

if(c==1 && x>=200 && x<=400 && y>=100 && y<=150 )

{

//clrscr();

cleardevice();

setbkcolor(BLACK);

std temp ;

outtextxy(230,200,"Enter ID to Add : ");

outtextxy(230,240,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(30,15);

//scanf("%s",temp.id);

gets(temp.id);

while(strlen(temp.id)<4){

gotoxy(30,15);

printf("Invalid!!!");

getch();

gotoxy(30,15);

printf(" ");

gotoxy(30,15);

gets(temp.id);

}

char t[20];

//t[0] = '\0' ;

strcpy(t,temp.id);

strcat(temp.id,"\n");

// id will be considered as file name hear

// scanf("%s",file\_name); // file name is student id hear

//fp = fopen(temp.id,"w");

clrscr();

cleardevice();

if(fopen(t,"r")!=0) // previously exists

{

cleardevice();

outtextxy(230,200,"ID Already Exists\n");

getch();

break ;

}

else

{

printf("%25sFill up details : \n","");

// own name

gotoxy(5,5);

printf("NAME : ");

outtextxy(300,75,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,8);

printf("Father`s Name : ");

outtextxy(300,125,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,11);

printf("Mother`s Name :");

outtextxy(300,170,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,14);

printf("Mobile Number :");

outtextxy(300,220,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,17);

printf("Address : ");

outtextxy(300,270,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

gotoxy(5,20);

printf("Current Semester :");

outtextxy(300,320,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// scanning part

// name

gotoxy(40,5);

//gets(file\_name); // for none logic not found

gets(temp.name);

while(isnum(temp.name) || strlen(temp.name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,5);

if( strlen(temp.name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,5);

printf(" ");

gotoxy(40,5);

gets(temp.name);

}

strcat(temp.name,"\n");

// changing colout if pressed enter

setcolor(GREEN);

outtextxy(300,75,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// faters name

gotoxy(40,8);

gets(temp.fathers\_name);

while(isnum(temp.fathers\_name) || strlen(temp.fathers\_name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,8);

if( strlen(temp.fathers\_name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,8);

printf(" ");

gotoxy(40,8);

gets(temp.fathers\_name);

}

strcat(temp.fathers\_name,"\n");

setcolor(GREEN);

outtextxy(300,125,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// mothers\_name

gotoxy(40,11);

gets(temp.mothers\_name);

while(isnum(temp.mothers\_name) || strlen(temp.mothers\_name)<=3 ){ // if string is empty or string is only number will have to enter again

gotoxy(40,11);

if( strlen(temp.mothers\_name)<=3){

printf("Too Shorrt ");

}else{

printf("Enter charecter only ");

}

getch();

gotoxy(40,11);

printf(" ");

gotoxy(40,11);

gets(temp.mothers\_name);

}

strcat(temp.mothers\_name,"\n");

setcolor(GREEN);

outtextxy(300,170,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//mobile\_number

gotoxy(40,14);

gets(temp.mobile\_number);

// in case mobile number is too short

while(1){

if(isnum(temp.mobile\_number) && strlen(temp.mobile\_number)==11){

break ;

}else{

gotoxy(40,14);

printf("Has To Be 11 Digits");

gotoxy(40,14);

getch();

gotoxy(40,14);

printf(" ");

gotoxy(40,14);

gets(temp.mobile\_number);

}

}

strcat(temp.mobile\_number,"\n");

setcolor(GREEN);

outtextxy(300,220,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// address

gotoxy(40,17);

gets(temp.address);

while(strlen(temp.address)<=4){

gotoxy(40,17);

printf(" ");

getch();

gotoxy(40,17);

gets(temp.address) ;

}

strcat(temp.address,"\n");

setcolor(GREEN);

outtextxy(300,270,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//semestr

gotoxy(40,20);

while(1){

gotoxy(40,20);

gets(temp.semester);

if(atoi(temp.semester)>0 && atoi(temp.semester)<=8){

break ;

}else{

gotoxy(40,20);

printf("Invalid !!");

getch();

gotoxy(40,20);

printf(" ");

}

}

strcat(temp.semester,"\n");

setcolor(GREEN);

outtextxy(300,320,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

// writing the collected data to file

// strcat(file\_name);

// as id file

fp = fopen(t,"w");

fprintf(fp,temp.id);

fprintf(fp,temp.name);

fprintf(fp,temp.fathers\_name);

fprintf(fp,temp.mothers\_name);

fprintf(fp,temp.mobile\_number);

fprintf(fp,temp.address);

fprintf(fp,temp.semester);

fclose(fp) ;

// append mode

// as fathers name

char t2[100] ;

strcpy(t2,temp.fathers\_name);

t2[strlen(temp.fathers\_name)-1] = '\0';

fp3 = fopen(t2,"w");

//fprintf(fp,temp.id) ;

fprintf(fp3,temp.id);

fprintf(fp3,temp.name);

fprintf(fp3,temp.fathers\_name);

fprintf(fp3,temp.mothers\_name);

fprintf(fp3,temp.mobile\_number);

fprintf(fp3,temp.address);

fprintf(fp3,temp.semester);

fclose(fp3) ;

// as fathers name

/\*

// as name

fp = fopen(temp.name,"a+");

fprintf(fp,"\n\n");

fprintf(fp,temp.id) ;

fprintf(fp,temp.name);

fprintf(fp,temp.fathers\_name);

fprintf(fp,temp.mothers\_name);

fprintf(fp,temp.mobile\_number);

fprintf(fp,temp.address);

fprintf(fp,temp.semester);

fclose(fp) ;

// end

\*/

gotoxy(40,25);

printf("PRESS\_ENTER\_TO\_SAVE");

getch();

break ;

}

}

//exit button

/\*if(c == 1 && x>600 && y>10 && x<625 && y<20) {

cleardevice();

delay(100);

return ;

}

\*/

if(c==1 && x>=600 && x<=650 && y>=0 && y<=30 )

{

out = -1 ;

break ;

}

}

}

// background

}

void cm()

{

i.x.ax = 1 ;

int86(0x33,&i,&o);

}

void mp(int &x ,int &y ,int &c )

{

i.x.ax = 3 ;

int86(0x33,&i,&o);

c = o.x.bx ;

x = o.x.cx ;

y = o.x.dx ;

}

void loadingAni()

{

int x,y,i;

int g=DETECT, d;

initgraph(&g, &d,"C:\\TC\\BGI");

cleardevice();

x=getmaxx()/2;

y=getmaxy()/2;

settextstyle(TRIPLEX\_FONT,HORIZ\_DIR,3);

setbkcolor(rand());

setcolor(4);

int c=150;

for(int e=0;e<15;e++)

{

for(int i=50;i<100;i++)

{

settextstyle(SMALL\_FONT, HORIZ\_DIR, 10);

outtextxy(x,y,"Loading...");

settextstyle(SMALL\_FONT, HORIZ\_DIR, 5);

outtextxy(x+5,y+5,"project.. ");

setcolor(3);

circle(x,y,i);

circle(x,y,c);

c--;

cleardevice();

}

}

setcolor(6);

closegraph();

}