



Project Submitted to the
CENTRAL BOARD OF SECONDARY EDUCATION

New Delhi.

Submitted by

HARITHA D

In the academic year

2023-2024

Under the guidance of

Mr. Swaminathan, M.Sc., B.Ed.,

Department of Computer Science



THAMARAI INTERNATIONAL SCHOOL

THANJAVUR

INDEX PAGE

Serial no	Description	Page
1	Bonafide Certificate	2
2	Acknowledgement	3
3	Python overview	4
4	Features of python	5
5	Advantages and installation of python	6
6	MySQL overview	8
7	Installation of MySQL	9
8	Hardware & Software requirements	10
9	Abstract of the project	11
10	Modules used and their purposes	12
11	Source code	13
12	Output screen	22
13	Bibliography	27

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"STUDENT MANAGEMENT SYSTEM"** is a bonafide work carried out by **HARITHA D** of Class XII (Computer Science Stream) Thamarai International School, Affiliated to the Central Board of Secondary Education, New Delhi during the academic year 2023-2024 in partial fulfilment of the requirements of the award of class XII Computer Science project work.

Teacher In-charge

Senior Principal

Submitted to All India Senior Secondary Examination held on_____ at
Thamarai International School, Thanjavur.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mentioning the names of those persons who helped to make it possible.

I take this opportunity to express my gratitude in few words and respect to all those who helped me in the completion of this project. It is my humble pleasure to acknowledge my deep senses of gratitude to my respected computer science teacher **Mr. Swaminathan** for his valuable support, constant help and guidance at each and every stage, without which this project would not have come forth.

I also register my sense of gratitude to our respected chairman **Mr. T. Venkatesan**, and respected Senior Principal **Mrs. Jayashree Badrinath** for their immense encouragement that made this project successful. I would also like to thank my friends and my family for encouraging me during the course of this project.

Last, but not the least, I would like to thank CBSE for giving us the opportunity to undertake this project.

OVERVIEW OF PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. the key advantages of learning Python are:

1. Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
2. Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
3. Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Characteristics of Python:

1. It supports functional and structured programming methods as well as OOP.
2. It can be used as a scripting language or can be compiled to byte-code for building large applications.
3. It provides very high-level dynamic data types and supports dynamic type checking.
4. It supports automatic garbage collection.
5. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python Features:

1. Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2. Easy-to-maintain – Python's source code is fairly easy-to-maintain.
3. A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
4. Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
5. Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
6. Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
7. Databases – Python provides interfaces to all major commercial databases.
8. GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
9. Scalable – Python provides a better structure and support for large programs than shell scripting.
10. Apart from the above-mentioned features, Python has a big list of good features, few are listed below –
11. It supports functional and structured programming methods as well as OOP.
12. It can be used as a scripting language or can be compiled to byte-code for building large applications.
13. It provides very high-level dynamic data types and supports dynamic type checking.
14. It supports automatic garbage collection.
15. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

ADVANTAGES OF PYTHON:

1. Python is very easy-to-use
2. Enhance Productivity
3. Interpreted programming Language
4. Support Open-source
5. Simply portable
6. Highly compatible with many other programming languages

INSTALLATION OF PYTHON:

1. Go to the official Python download page for Windows. Find a stable Python 3 release.
2. Click the appropriate link for your system to download the executable file: Windows installer (64-bit) or Windows installer (32-bit).
3. After the installer is downloaded, double-click the .exe file, for example python-3.10.10-amd64.exe, to run the Python installer.
4. Select the Install launcher for all users checkbox, which enables all users of the computer to access the Python launcher application.
5. Select the Add python.exe to PATH checkbox, which enables users to launch Python from the command line.
6. Then click Install Now and go to Step 4 - Verify the Python Installation. To install other optional and advanced features, click Customize installation and continue.
7. Click Next.
8. The Advanced Options dialog displays.
9. Select the options that suit your requirements:
 - a. Install for all users: recommended if you're not the only user on this computer

- b. Associate files with Python: recommended, because this option associates all the Python file types with the launcher or editor
- c. Create shortcuts for installed applications: recommended to enable shortcuts for Python applications
- d. Add Python to environment variables: recommended to enable launching Python
- e. Precompile standard library: not required, it might slow down the installation
- f. Download debugging symbols and Download debug binaries: recommended only if you plan to create C or C++ extensions

MYSQL OVERVIEW

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

More about MySQL:

- MySQL is a database management system.
- MySQL databases are relational.
- MySQL software is Open Source.
- The MySQL Database Server is very fast, reliable, scalable, and easy to use.
- MySQL Server works in client/server or embedded systems.
- A large amount of contributed MySQL software is available.

Reasons for popularity:

- MySQL is an open-source database, so you don't have to pay a single penny to use it.
- MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages.
- MySQL is customizable because it is an open-source database, and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.
- MySQL is quicker than other databases, so it can work well even with the large data set.
- MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL is very friendly with PHP, the most popular language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

INSTALLATION OF MYSQL

1. Download MySQL Installer from <https://dev.mysql.com/downloads/installer/> and execute it.
2. Determine the setup type to use for the initial installation of MySQL products. For example:
 - a. Developer Default: Provides a setup type that includes the selected version of MySQL Server and other MySQL tools related to MySQL development, such as MySQL Workbench.
 - b. Server Only: Provides a setup for the selected version of MySQL Server without other products.
 - c. Custom: Enables you to select any version of MySQL Server and other MySQL products.
3. Install the server instance (and products) and then begin the server configuration by following the onscreen instructions. For more information about each individual step, see Section 2.3.3.3.1, "MySQL Server Configuration with MySQL Installer".

SOFTWARE & HARDWARE REQUIREMENTS

HARDWARE:

1. Processor – Pentium(R) G2030@ 3.70 GHz
2. RAM: 4 GB
3. System Architecture: 32 or 64 bit Operating system
4. HDD: 500 GB

SOFTWARE:

1. Operating system: Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 11
2. IDE: Python 3.6 or higher
3. Database: MySQL Connector/Python 8.0 with MySQL Server 8.0, 5.7, and 5.6

ABSTRACT

The Student Management System is a simple project developed using python, which is inexpensive and easy to use. It can handle all the details of student such as name, roll no, contact, e-mail id, D.O.B. etc. It has features like adding records, updating records, deleting records and searching records etc. It is time saving, user friendly.

MODULES USED AND THEIR PURPOSE

- 1. Tkinter:** This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc.
- 2. MYSQL:** MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

SOURCE CODE

```
from tkinter import *
from tkinter import ttk
##import my sql
from tkinter import messagebox
import mysql.connector as m
root=Tk()
root.title("Student Management System-abi and haritha")
root.geometry("1366x768")
title=Label(root,text="Student Mangement System",font=("COURIER
NEW",44,"bold"),bg="medium violet red",fg="white")
title.pack(side=TOP,fill=X)
con=m.connect(host='localhost',user='root',password='root',database='stm')
cur=con.cursor()
cur.execute('CREATE DATABASE STM;')
cur.execute('USE STM;')
cur.execute("""CREATE TABLE IF NOT EXISTS STUDENTS(
ROLL_NO INT PRIMARY KEY,
NAME CHAR(50),
EMAIL_ID char(50),
GENDER char(10),
CONTACT BIGINT,
DOB date,
ADDRESS char(90)
);""")
IMG=PhotoImage(file='z://GRADE XII TKINTER PROJECTS/IMAGE-STUDENT.png')
w=Label(root,image=IMG)
```

```
w.place(x=70,y=2)
```

```
***** All variable *****
```

```
def add_student():
```

```
    if roll_no_var.get()=="" or name_var.get()=="":
```

```
        messagebox.showerror("Error" ,"All Fields are required!")
```

```
    else:
```

```
        con=m.connect(host='localhost',user='root',password='root',database='stm')
```

```
        cur=con.cursor()
```

```
        qry=("insert into students values('{0}','{1}','{2}','{3}','{4}','{5}','{6}');". format(
```

```
        roll_no_var.get(),name_var.get(),email_var.get(),gender_var.get(),
```

```
        contact_var.get,dob_var.get(),txt_addr.get('1.0',END)))
```

```
        cur.execute(qry)
```

```
        con.commit()
```

```
        messagebox.showinfo("Success","Record has been inserted")
```

```
        fetch_data()
```

```
        clear()
```

```
        con.close()
```

```
def fetch_data():
```

```
    con=m.connect(host='localhost',user='root',password='root',database='stm')
```

```
    cur=con.cursor()
```

```
    cur.execute("select * from students;")
```

```
    rows=cur.fetchall()
```

```
    if len(rows)!=0:
```

```
        Student_table.delete(*Student_table.get_children())
```

```
        for row in rows:
```

```
            Student_table.insert("",END,values=row)
```

```
            con.commit()
```

```

else:
    for item in Student_table.get_children():
        Student_table.delete(item)
    con.close()
def clear():
    roll_no_var.set("")
    name_var.set("")
    email_var.set("")
    contact_var.set("")
    gender_var.set("")
    dob_var.set("")
    txt_addr.delete("1.0",END)
def get_cursor(ev):
    cursr_row=Student_table.focus()
    contents=Student_table.item(cursr_row)
    row=contents['values']
    roll_no_var.set(row[0])
    name_var.set(row[1])
    email_var.set(row[2])
    contact_var.set(row[4])
    gender_var.set(row[3])
    dob_var.set(row[5])
    txt_addr.delete("1.0",END)
    txt_addr.insert(END,row[6])
def update_data():
    con=m.connect(host='localhost',user='root',password='root',database='stm')
    cur=con.cursor()

```



```

cur.execute("update students set  name='{1}',email_id='{2}',gender='{3}',contact=
'{4}',dob='{5}',address='{6}' where roll_no='{0}' ;" .format(roll_no_var.get(),
name_var.get(),email_var.get(),gender_var.get(),contact_var.get(),dob_var.get(),
txt_addr.get('1.0',END)))
con.commit()

messagebox.showinfo("Success","Record has been updated")

fetch_data()

clear()

con.close()

def delete_data():

    con=m.connect(host='localhost',user='root',password='root',database='stm')

    cur=con.cursor()

    cur.execute("delete from students where roll_no={};".format(roll_no_var.get()))

    con.commit()

    messagebox.showinfo("Success","Record has been deleted")

    con.close()

    fetch_data()

    clear()

def search_data():

    con=m.connect(host='localhost',user='root',password='root',database='stm')

    cur=con.cursor()

    cur.execute("select * from students where "+str(search_by.get())+" LIKE '%" +
str(search_txt.get())+"%'")

    rows=cur.fetchall()

    if len(rows)!=0:

        Student_table.delete(*Student_table.get_children())

        for row in rows:

```

```

        Student_table.insert("",END,values=row)

        con.commit()

    con.close()

roll_no_var=StringVar()
name_var=StringVar()
email_var=StringVar()
gender_var=StringVar()
contact_var=StringVar()
dob_var=StringVar()
search_by=StringVar()
search_txt=StringVar()

#*****Manage Frame*****

Manage_frame=Frame(root,bd=4,relief=RIDGE,bg="blue4")
Manage_frame.place(x=20,y=100,width=500,height=580)
m_title=Label(Manage_frame,text="Manage Student", bg="blue4", fg
="white",font=("times new roman",25,"bold"))
m_title.grid(row=0,columnspan=2,pady=15)

lbl_roll=Label(Manage_frame,text="Roll no.",bg="blue4",fg="white",font=("times
new roman",15,"bold"))
lbl_roll.grid(row=1,column=0,pady=10,padx=25,sticky="w")

txt_roll=Entry(Manage_frame,textvariable=roll_no_var,font=("times new
roman",10,"bold"),bd=5,relief=GROOVE)
txt_roll.grid(row=1,column=1,pady=10,padx=20,sticky="w")

lbl_name=Label(Manage_frame,text="Name",bg="blue4",fg="white",font=("times

```

```

new roman",15,"bold"))

lbl_name.grid(row=2,column=0,pady=10,padx=20,sticky="w")

txt_name=Entry(Manage_frame,textvariable=name_var,font=("times new
roman",10,"bold"),bd=5,relief=GROOVE)

txt_name.grid(row=2,column=1,pady=10,padx=20,sticky="w")


lbl_email=Label(Manage_frame,text="Email",bg="blue4",fg="white",font=("times
new roman",15,"bold"))

lbl_email.grid(row=3,column=0,pady=10,padx=20,sticky="w")


txt_email=Entry(Manage_frame,textvariable=email_var,font=("times new
roman",10,"bold"),bd=5,relief=GROOVE)

txt_email.grid(row=3,column=1,pady=10,padx=20,sticky="w")


lbl_gender=Label(Manage_frame,text="Gender",bg="blue4",fg="white",font=("times
new roman",15,"bold"))

lbl_gender.grid(row=4,column=0,pady=10,padx=20,sticky="w")

combo_gender=ttk.Combobox(Manage_frame,textvariable=gender_var,font=("times
new roman",9,"bold"),state="readonly")

combo_gender['values']=("Male","Female","Other")

combo_gender.grid(row=4,column=1,pady=10,padx=20,sticky="w")

lbl_contact=Label(Manage_frame,text="Contact",bg="blue4",fg="white",font=("times
new roman",15,"bold"))

lbl_contact.grid(row=5,column=0,pady=10,padx=20,sticky="w")

txt_contact=Entry(Manage_frame,textvariable=contact_var,font=("times new
roman",10,"bold"),bd=5,relief=GROOVE)

txt_contact.grid(row=5,column=1,pady=10,padx=20,sticky="w")

```

```
lbl_dob=Label(Manage_frame,text="D.O.B.",bg="blue4",fg="white",font=("times new roman",15,"bold"))
```

```
lbl_dob.grid(row=6,column=0,pady=10,padx=20,sticky="w")
```

```
txt_dob=Entry(Manage_frame,textvariable=dob_var,font=("times new roman",10,"bold"),bd=5,relief=GROOVE)
```

```
txt_dob.grid(row=6,column=1,pady=10,padx=20,sticky="w")
```

```
lbl_addr=Label(Manage_frame,text="Address",bg="blue4",fg="white",font=("times new roman",15,"bold"))
```

```
lbl_addr.grid(row=7,column=0,pady=10,padx=20,sticky="w")
```

```
txt_addr=Text(Manage_frame,width=19,height=3)
```

```
txt_addr.grid(row=7,column=1,pady=10,padx=20,sticky="w")
```

```
*****button Frame*****
```

```
btn_frame=Frame(Manage_frame,bd=4,relief=RIDGE,bg="blue4")
```

```
btn_frame.place(x=10,y=500,width=450)
```

```
addbutton=Button(btn_frame,text="Add",width=10,command=add_student).grid(row=0,column=0,padx=10,pady=10)
```

```
updatebutton=Button(btn_frame,text="Update",width=10,command=update_data).grid(row=0,column=1,padx=10,pady=10)
```

```
deletebutton=Button(btn_frame,text="Delete",width=10,command=delete_data).grid(row=0,column=2,padx=10,pady=10)
```

```
clearbutton=Button(btn_frame,text="Clear",width=10,command=clear).grid(row=0,c
```

```
*****Detail Frame*****
```

```

Detail_frame=Frame(root,bd=4,relief=RIDGE,bg="blue4")
Detail_frame.place(x=530,y=100,width=790,height=580)

lbl_search=Label(Detail_frame,text="Search By",bg="blue4",fg="white",font=("times
new roman",15,"bold"))
lbl_search.grid(row=0,column=0,pady=10,padx=20,sticky="w")

combo_search=ttk.Combobox(Detail_frame,width=10,textvariable=search_by,font=("t
imes new roman",10,"bold"),state="readonly")
combo_search['values']=("roll_no","name","contact")
combo_search.grid(row=0,column=1,pady=10,padx=20,sticky="w")
txt_search=Entry(Detail_frame,textvariable=search_txt,font=("times new
roman",10,"bold"),bd=4,relief=GROOVE)
txt_search.grid(row=0,column=2,pady=10,padx=20,sticky="w")
searchbtn=Button(Detail_frame,text="Search",width=10,pady=3,command=search_d
ata).grid(row=0,column=3,padx=10,pady=10)
showbtn=Button(Detail_frame,text="Show
All",width=10,pady=3,command=fetch_data).grid(row=0,column=4,padx=10,pady=1
0)

#*****Table Frame*****
Table_frame=Frame(Detail_frame,bd=4,relief=RIDGE,bg="ghost white")
Table_frame.place(x=10,y=70,width=760,height=500)
scroll_x=Scrollbar(Table_frame,orient=HORIZONTAL)
scroll_y=Scrollbar(Table_frame,orient=VERTICAL)
Student_table=ttk.Treeview(Table_frame,columns=("roll","name","email","gender","co
ntact","dob","address"),xscrollcommand=scroll_x.set,yscrollcommand=scroll_y.set)
scroll_x.pack(side=BOTTOM,fill=X)

```

```
scroll_y.pack(side=RIGHT,fill=Y)
scroll_x.config(command=Student_table.xview)
scroll_y.config(command=Student_table.yview)
Student_table.heading("roll",text="Roll No")
Student_table.heading("name",text="Name")
Student_table.heading("email",text="Email Id")
Student_table.heading("gender",text="Gender")
Student_table.heading("contact",text="Contact")
Student_table.heading("dob",text="D.O.B.")
Student_table.heading("address",text="Address")
Student_table['show']='headings'
Student_table.column("roll",width=60)
Student_table.column("name",width=110)
Student_table.column("email",width=110)
Student_table.column("gender",width=110)
Student_table.column("contact",width=110)
Student_table.column("dob",width=110)
Student_table.column("address",width=110)
Student_table.pack(fill=BOTH,expand=1)
Student_table.bind("<ButtonRelease-1>",get_cursor)
fetch_data()
```

OUTPUT

MAIN WINDOW

Student Management System-abi and haritha

Student Mangement System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
---------	------	----------	--------	---------	--------	---------

Activate Windows
Go to Settings to activate Windows

Type here to search

17:53
13-09-2023

ADDING RECORD

Student Management System-abi and haritha

Student Mangement System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
---------	------	----------	--------	---------	--------	---------

Success
Record has been inserted

Activate Windows
Go to Settings to activate Windows

Type here to search

17:56
13-09-2023

RECORD ADDED

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR

Activate Windows
Go to Settings to activate Windows

UPDATING RECORD

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHAD	harithad@gmail.com	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA	teenagarwal@gmail.com	Female	7589642368	2006-05-06	MUMBAI
9605	RAJUL CLAUDIA	Rajul@gmail.com	Male	8888856825	1996-01-16	DELHI

Success
Record has been updated
OK

Activate Windows
Go to Settings to activate Windows

UPDATED RECORD

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHAD	harithad@gmail.com	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA AGARWAL	teenaagarwal@gmail.com	Female	75896423	2006-05-06	MUMBAI
9685	RAGUL SHARMA	Ragul@gmail.com	Male	8888856825	1996-01-16	DELHI

Activate Windows
Go to Settings to activate Windows

DELETING RECORD

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHAD	harithad@gmail.com	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA AGARWAL	teenaagarwal@gmail.com	Female	75896423	2006-05-06	MUMBAI
9685	RAGUL SHARMA	Ragul@gmail.com	Male	8888856825	1996-01-16	DELHI

Success

Record has been deleted

Activate Windows
Go to Settings to activate Windows

DELETED RECORD

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHA.D	harithad@gmail.cor	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA AGARWAL	teenaagarwal@gma	Female	75896423	2006-05-06	MUMBAI

Activate Windows
Go to Settings to activate Windows

SEARCHING RECORDS

Student Management System-abi and haritha

Student Management System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAMIT	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHA.D	harithad@gmail.cor	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA AGARWAL	teenaagarwal@gma	Female	75896423	2006-05-06	MUMBAI

Activate Windows
Go to Settings to activate Windows

SEARCHED RECORD

Student Management System-abi and haritha

Student Mangement System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI

Activate Windows
Go to Settings to activate Windows

VIEWING RECORD

Student Management System-abi and haritha

Student Mangement System

Manage Student

Roll no.

Name

Email

Gender

Contact

D.O.B.

Address

Search By

Roll No	Name	Email Id	Gender	Contact	D.O.B.	Address
123	SREE ABIRAAAMI.T	sreeabi123@gmail.c	Female	9876543210	2000-05-05	THANJAVUR
655	FASEEHA NAVREEN	fasi234@gmail.com	Female	9999955555	2000-01-01	CHENNAI
3962	HARITHA.D	harithad@gmail.com	Female	6239587410	2007-05-23	CHENNAI
8967	TEENA AGARWAL	teenaagarwal@gma	Female	75896423	2006-05-06	MUMBAI

Activate Windows
Go to Settings to activate Windows

BIBLIOGRAPHY

1. Computer science with Python - Class XI & XII by: Sumita Arora
2. Website : <https://www.stackoverflow.com>