PROJECT REPORT ON

SCHOOL MANAGEMENT SYSTEM

- **SIDDHARTHA KUMAR**

CONTENTS

S.No	Index	Page No
1	INTRODUCTION	
2	OBJECTIVE AND SCOPE OF THE PROJECT	
3	PROBLEM DEFINITION AND ANALYSIS	
4	HARDWARE AND SOFTWARE REQUIREMENTS	
5	SYSTEM DESIGN AND DEVELOPMENT	
6	SOURCE CODE	
7	WORKING OF THE SOFTWARE	
8	USER MANUAL (HOW TO INSTALL)	
9	FUTURE SCOPE OF THE PROJECT	
10	LIMITATIONS OF THE PROJECT	

1. INTRODUCTION

This software project is developed to automate the functionalities of a School. The purpose of the software project is to develop the Management Information System (MIS) to automate the record of the students, and their performance with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter- related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application Program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a school system.

During coding and design of the software Project, Python a powerful high level programming language is used. We have used Pandas and Matplotlib Libraries of Python for the same. As a back-end CSV file is used as per requirement of the CBSE curriculum of Informatics Practices Course.

Pandas is an open-source, Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays.

2. OBJECTIVE AND SCOPE OF PROJECT

The objective of the software project is to develop a computerized MIS to automate the functions of a school. This software project is also aimed to enhance the current record keeping system, which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

- To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environ1nent for MIS activities.
- The proposed system should maintain all the records and should generate the required reports and information ·when required.
- To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

In its current scope, the software enables user to retrieve and update the information from CSV file. This software does not require much training time of the users due to limited functionality and simplicity.

During the development of School Management System project, IDLE, a powerful, open source python default development environment is used with CSV as a database system.

3. PROBLEM DEFINITION AND ANALYSIS

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the What of a system, not How. The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- Identification and documentation of customer's or user's needs.
- Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- Evolution of needs.

After the analysis of the functioning of a School management system, the proposed System is expected to do the following: -

- To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for computerized School Management System.
- The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

4. HARDWARE AND SOFTWARE REQUIREMENTS

The Hardware used:

While developing the system, the used hardware is:

Laptop PC with Pentium IV processor or sometimes, Laptop PC with Core i3 (2.3 GHz) processor having 12 GB RAM.

The Software used:

Windows 7 and Windows 10 as Operating System
Python 3.7
Microsoft Excel for CSV files
Microsoft Word for Documentation

5. SYSTEM DESIGN AND DEVELOPMENT

Database Design:

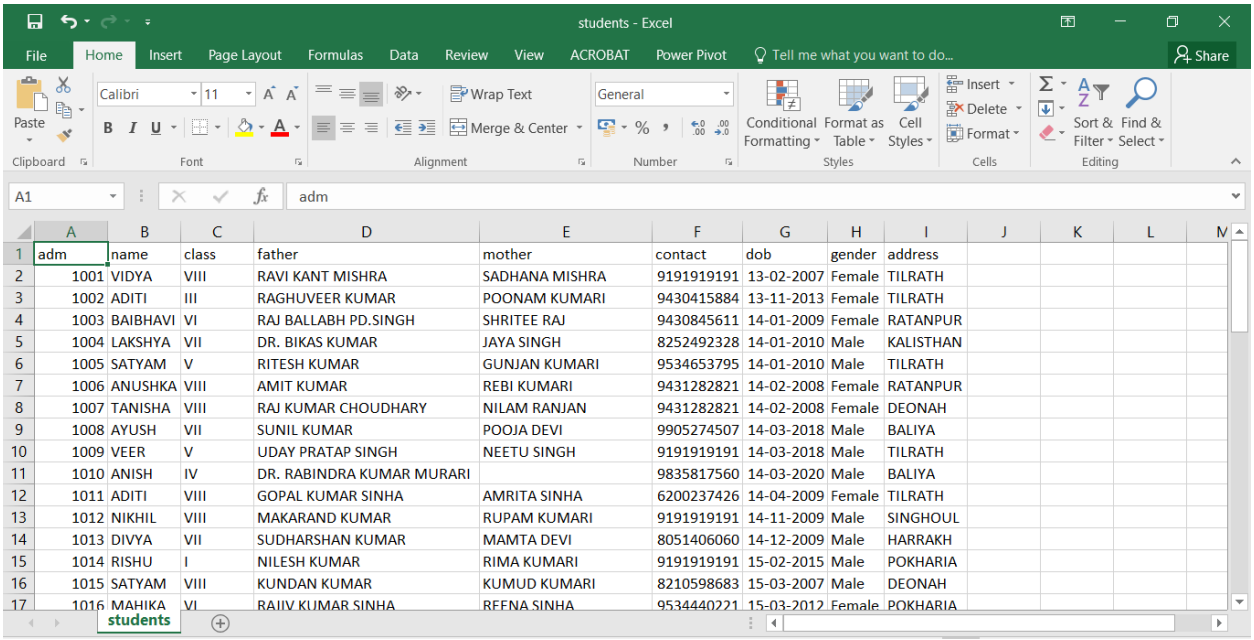
An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table struch1re and its relationship.

This software project maintains a three CSV Files names students, markstermi and markstermii.

Table Design:

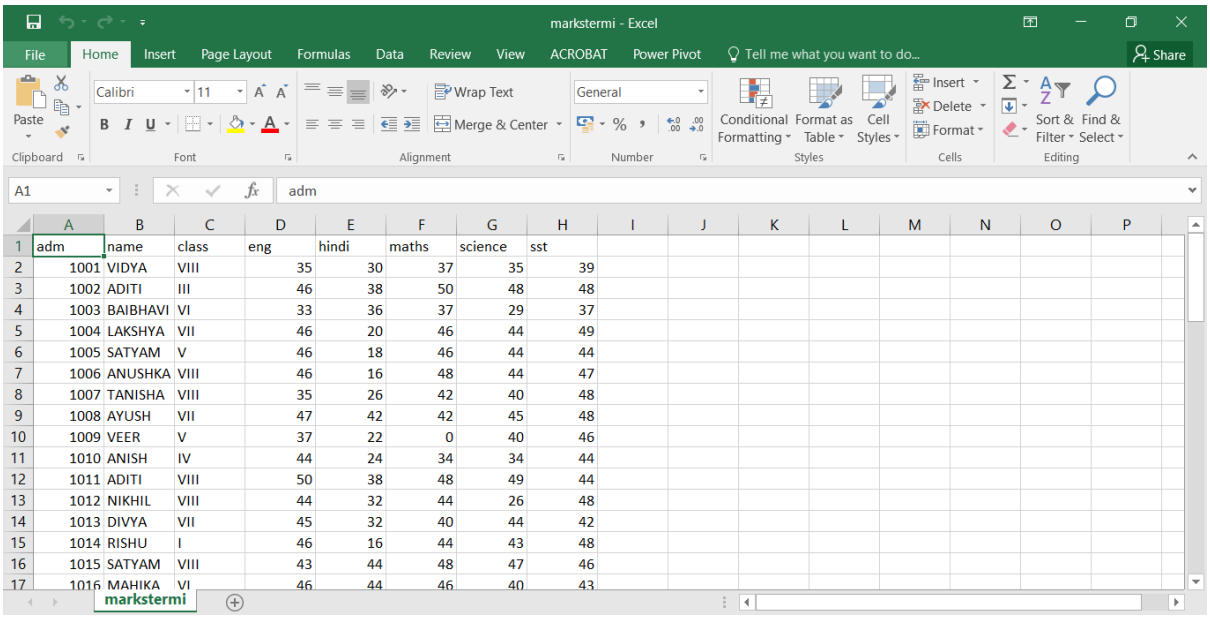
The database of School Management System contains 3 CSV tables. The tables are normalized to minimize the redundancies of data and enforcing the validation rules of the organization. The tables and their structure are given below.

students.csv



adm	name	class	father	mother	contact	dob	gender	address
1001	VIDYA	VIII	RAVI KANT MISHRA	SADHANA MISHRA	9191919191	13-02-2007	Female	TILRATH
1002	ADITI	III	RAGHUVEER KUMAR	POONAM KUMARI	9430415884	13-11-2013	Female	TILRATH
1003	BAIBHAVI	VI	RAJ BALLABH PD.SINGH	SHRITEE RAJ	9430845611	14-01-2009	Female	RATANPUR
1004	LAKSHYA	VII	DR. BIKAS KUMAR	JAYA SINGH	8252492328	14-01-2010	Male	KALISTHAN
1005	SATYAM	V	RITESH KUMAR	GUNJAN KUMARI	9534653795	14-01-2010	Male	TILRATH
1006	ANUSHKA	VIII	AMIT KUMAR	REBI KUMARI	9431282821	14-02-2008	Female	RATANPUR
1007	TANISHA	VIII	RAJ KUMAR CHOUDHARY	NILAM RANJAN	9431282821	14-02-2008	Female	DEONAH
1008	AYUSH	VII	SUNIL KUMAR	POOJA DEVI	9905274507	14-03-2018	Male	BALIYA
1009	VEER	V	UDAY PRATAP SINGH	NEETU SINGH	9191919191	14-03-2018	Male	TILRATH
1010	ANISH	IV	DR. RABINDRA KUMAR MURARI		9835817560	14-03-2020	Male	BALIYA
1011	ADITI	VIII	GOPAL KUMAR SINHA	AMRITA SINHA	6200237426	14-04-2009	Female	TILRATH
1012	NIKHIL	VIII	MAKARAND KUMAR	RUPAM KUMARI	9191919191	14-11-2009	Male	SINGHOUL
1013	DIVYA	VII	SUDHARSHAN KUMAR	MAMTA DEVI	8051406060	14-12-2009	Male	HARRAKH
1014	RISHU	I	NILESH KUMAR	RIMA KUMARI	9191919191	15-02-2015	Male	POKHARIA
1015	SATYAM	VIII	KUNDAN KUMAR	KUMUD KUMARI	8210598683	15-03-2007	Male	DEONAH
1016	MAHIKA	VI	RAJIV KUMAR SINHA	REENA SINHA	9534440221	15-03-2012	Female	POKHARIA

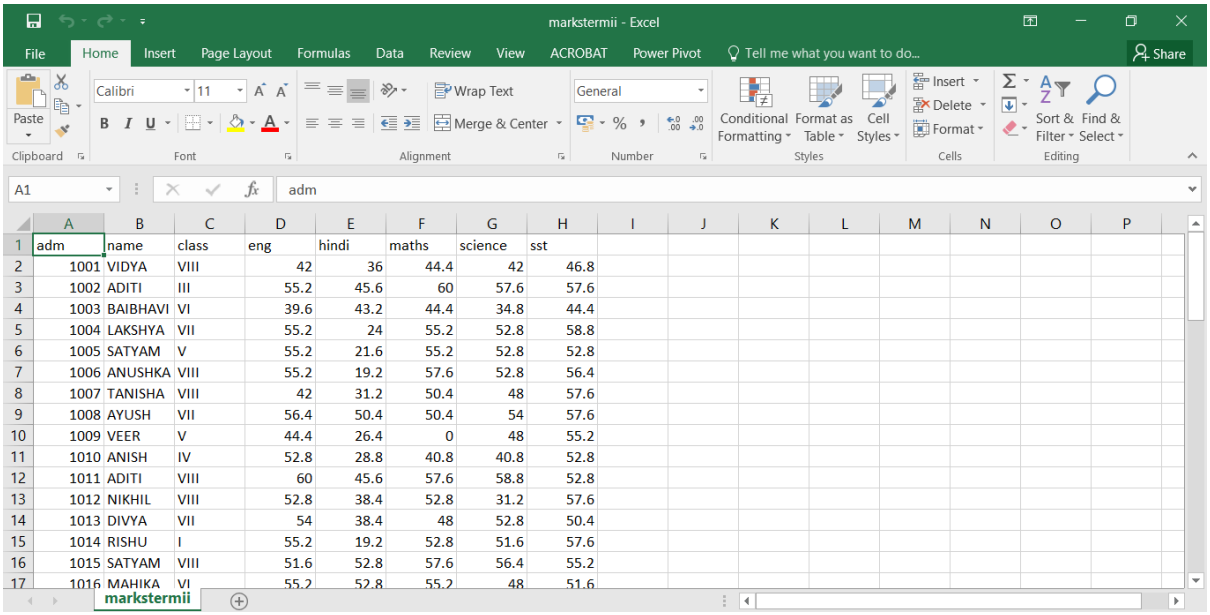
marksstermi.csv



marksstermi - Excel

adm	name	class	eng	hindi	maths	science	sst
1001	VIDYA	VIII	35	30	37	35	39
1002	ADITI	III	46	38	50	48	48
1003	BAIBHAVI	VI	33	36	37	29	37
1004	LAKSHYA	VII	46	20	46	44	49
1005	SATYAM	V	46	18	46	44	44
1006	ANUSHKA	VIII	46	16	48	44	47
1007	TANISHA	VIII	35	26	42	40	48
1008	AYUSH	VII	47	42	42	45	48
1009	VEER	V	37	22	0	40	46
1010	ANISH	IV	44	24	34	34	44
1011	ADITI	VIII	50	38	48	49	44
1012	NIKHIL	VIII	44	32	44	26	48
1013	DIVYA	VII	45	32	40	44	42
1014	RISHU	I	46	16	44	43	48
1015	SATYAM	VIII	43	44	48	47	46
1016	MAHIKA	VI	46	44	46	40	43

markstermii.csv



markstermii - Excel

adm	name	class	eng	hindi	maths	science	sst
1001	VIDYA	VIII	42	36	44.4	42	46.8
1002	ADITI	III	55.2	45.6	60	57.6	57.6
1003	BAIBHAVI	VI	39.6	43.2	44.4	34.8	44.4
1004	LAKSHYA	VII	55.2	24	55.2	52.8	58.8
1005	SATYAM	V	55.2	21.6	55.2	52.8	52.8
1006	ANUSHKA	VIII	55.2	19.2	57.6	52.8	56.4
1007	TANISHA	VIII	42	31.2	50.4	48	57.6
1008	AYUSH	VII	56.4	50.4	50.4	54	57.6
1009	VEER	V	44.4	26.4	0	48	55.2
1010	ANISH	IV	52.8	28.8	40.8	40.8	52.8
1011	ADITI	VIII	60	45.6	57.6	58.8	52.8
1012	NIKHIL	VIII	52.8	38.4	52.8	31.2	57.6
1013	DIVYA	VII	54	38.4	48	52.8	50.4
1014	RISHU	I	55.2	19.2	52.8	51.6	57.6
1015	SATYAM	VIII	51.6	52.8	57.6	56.4	55.2
1016	MAHIKA	VI	55.2	52.8	55.2	48	51.6

6. SOURCE CODE

```
import pandas as pd
import matplotlib.pyplot as plt

path = "F:\Student_Management_System\students.csv"
term1path = "F:\Student_Management_System\markstermi.csv"
term2path = "F:\Student_Management_System\markstermii.csv"

def Menu():
    print("-----------------------------------------------------------------")
    print("\t\tWELCOME TO SCHOOL MANAGEMENT SYSTEM")
    print("-----------------------------------------------------------------")
    print("Please select from the below options : ")
    print("1. Show all Students Record")
    print("2. Search for a Student")
    print("3. Add a New Student ")
    print("4. Delete a Student ")
    print("5. Update any Student Details")
    print("6. Display Term I Result ")
    print("7. Display Term II Result ")
```

```
    print("8. Display / Input Marks of Term - I")
    print("9. Display / Input Marks of Term - II")
    print("10. Display Graphs")

def ShowAllRecord():
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    print("Displaying Record of All Students . . .")
    print()
    print(dfStu)



def ShowT1Result():
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
    print("Displaying Marks of Term - I ")
    print()
    print(dfTerm1)

def ShowT2Result():
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm',)
    print("Displaying Marks of Term - II ")
    print()
```

```
    print(dfTerm2)



def AddStudent():
    adm = int(input("Enter the Admission Number : "))
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    if adm in dfStu.index:
        print("Admission Number",adm,"already alloted to another student. . .")
        print("Try Again . . . ")
    else:
        name = input("Enter Student's Name : ") ; name = name.upper()
        Class = input("Enter Student's Class : ") ; Class = Class.upper()
        father = input("Enter Father's Name : ") ;  father = father.upper()
        mother = input("Enter Mother's Name : ") ; mother = mother.upper()


        con = True
        while (con==True):
            contact = int(input("Enter the Contact Number : "))
            if len(str(contact))!=10:
                print("Enter 10 digit contact number ...")
                print("Try Again . . .")
```

```
    else:
        con = False

con=True
while (con==True):
    DOB = input("Enter the Date of Birth (DD-MM-YYYY) :")
    if len(DOB) != 10:
        print("Enter Date of Birth as DD-MM-YYYY")
        print("Try Again . . .")
    else:
        con=False

gender = input("Gender (Male/Female) : "); gender = gender.upper()
address = input("Enter the Address : "); address = address.upper()

dfStu.at[adm,:] = [name,Class,father,mother,contact,DOB,gender,address]
dfStu.to_csv(path)

dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
dfTerm1.at[adm,:] = [name,Class,0,0,0,0,0]
dfTerm2.at[adm,:] = [name,Class,0,0,0,0,0]
dfTerm1.to_csv(term1path)
```

```
        dfTerm2.to_csv(term2path)
        print("Student Added Successfully. . .")



def StudentSearch():
    print("Press 1 for searching by Name")
    print("Press 2 for searching by Admission Number")
    ch = int(input("Enter your choice : "))
    if ch==1:
        name = input("Enter Student's Name : ") ; name = name.upper()
        dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
        df = dfStu.loc[dfStu["name"]==name]
        if df.empty:
            print("Student doesnot exist ...")
        else:
            print("Student details are as below : ")
            print(df)


    elif ch==2:
        adm = int(input("Enter the Admission Number : "))
        dfStu = pd.read_csv(path,index_col = 'adm',\
```

```
            usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
        if adm in dfStu.index:
            df = dfStu.loc[adm]
            print("Student Details are as below : \n")
            print(df)
        else:
            print("No Student Exists with the above mentioned Admission Number ...")
    else:
        print("Invalid Choice . . .")



def DeleteStudent():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfStu = pd.read_csv(path,index_col = 'adm',\
            usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    df = dfStu.loc[dfStu["name"]==name]
    if df.empty:
        print("Student doesnot exist ...")
    else:
        print("Student details are as follows : ")
        print(df[['name','class','father','mother']])
```

```
    adm = int(input("Enter the Admission Number to be deleted : "))
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
    dfStu = dfStu.drop(adm,axis=0)
    dfTerm1 = dfTerm1.drop(adm,axis=0)
    dfTerm2 = dfTerm2.drop(adm,axis=0)
    dfTerm1.to_csv(term1path)
    dfTerm2.to_csv(term2path)
    dfStu.to_csv(path)
    print("Student Deleted Successfully. . .")



def UpdateStudentDetails():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
    df = dfStu.loc[dfStu["name"]==name]
```

```
    if df.empty:
        print("Student doesnot exist ...")
    else:
        print("Student details are as follows : ")
        print(df)
    adm = int(input("Enter the Admission Number to be updated : "))
    print("Enter from below options :
\nName\nClass\nFather\nMother\nContact\nDOB\nGender\nAddress")
    change = input("Enter your choice : ") ; change =change.lower()

    if change == 'name':
        name = input("Enter Student's Updated Name : ") ; name = name.upper()
        dfStu.loc[adm,[change]] = name
        dfTerm1.loc[adm,[change]] = name
        dfTerm2.loc[adm,[change]] = name
    elif change == 'class':
        Class = input("Enter Student's Updated Class : ") ; Class = Class.upper()
        dfStu.loc[adm,[change]] = Class
        dfTerm1.loc[adm,[change]] = Class
        dfTerm2.loc[adm,[change]] = Class
    elif change =='father':
        father = input("Enter Father's Name : ") ;  father = father.upper()
        dfStu.loc[adm,[change]] = father
```

```
elif change == 'mother':
    mother = input("Enter Mother's Name : ") ; mother = mother.upper()
    dfStu.loc[adm,[change]] = mother

elif change == 'contact':
    con = True
    while (con==True):
        contact = int(input("Enter the Contact Number : "))
        if len(str(contact))!=10:
            print("Enter 10 digit contact number ...")
            print("Try Again . . .")
        else:
            dfStu.loc[adm,[change]] = contact
            con = False

elif change == 'dob':
    con=True
    while (con==True):
        DOB = input("Enter the Date of Birth (DD-MM-YYYY) :")
        if len(DOB) != 10:
            print("Enter Date of Birth as DD-MM-YYYY")
            print("Try Again . . .")
```

```
            else:
                con=False
                dfStu.loc[adm,[change]] = DOB

    elif change == 'gender':
        gender = input("Gender (Male/Female) : "); gender = gender.upper()
        dfStu.loc[adm,[change]] = gender

    elif change == 'address':
        address = input("Enter the Address : "); address = address.upper()
        dfStu.loc[adm,[change]] = address

    else:
        print("Wrong Choice . . .")

    dfTerm1.to_csv(term1path)
    dfTerm2.to_csv(term2path)
    dfStu.to_csv(path)
    print("Student Details Updated Sucessfully . . .")

def UpdateTerm1Marks():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
```

```
df = dfTerm1.loc[dfTerm1["name"]==name]
if df.empty:
    print("Student doesnot exist ...")
else:
    print("Student details are as follows : ")
    print(df[['name','class']])
    adm = int(input("Enter the Admission Number of the Student : "))
    df = df.loc[adm,:]
    print(df)
    ch = input("Do you want to update the marks (Y/N) : ") ; ch = ch.upper()
    if ch=='Y':
        name = df['name']
        Class = df['class']
        eng = float(input("Enter the marks in English : "))
        hindi = float(input("Enter the marks in Hindi : "))
        maths = float(input("Enter the marks in Maths : "))
        sci = float(input("Enter the marks in Science : "))
        sst = float(input("Enter the marks in SST : "))
        dfTerm1.loc[adm,:]=[name,Class,eng,hindi,maths,sci,sst]
        dfTerm1.to_csv(term1path)
        print("Marks added successfully . . .")
    else:
        print("Thanks . . .")
```

```
def UpdateTerm2Marks():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
    df = dfTerm2.loc[dfTerm2["name"]==name]
    if df.empty:
        print("Student doesnot exist ...")
    else:
        print("Student details are as follows : ")
        print(df[['name','class']])
        adm = int(input("Enter the Admission Number of the Student : "))
        df = df.loc[adm,:]
        print(df)
        ch = input("Do you want to update the marks (Y/N) : ") ; ch = ch.upper()
        if ch=='Y':
            name = df['name']
            Class = df['class']
            eng = float(input("Enter the marks in English : "))
            hindi = float(input("Enter the marks in Hindi : "))
            maths = float(input("Enter the marks in Maths : "))
            sci = float(input("Enter the marks in Science : "))
            sst = float(input("Enter the marks in SST : "))
```

```
            dfTerm2.loc[adm,:]=[name,Class,eng,hindi,maths,sci,sst]
            dfTerm2.to_csv(term2path)
            print("Marks added successfully . . .")
        else:
            print("Thanks . . .")


def ShowGraphs():
    print()
    print("Select from the below option . . . \n")
    print("1. Student's Location Graph ")
    print("2. Class Count Graph ")
    print("3. Gender Graph ")
    print("4. Term - I Class Wise Marks ")
    print("5. Term - II Class Wise Marks ")
    print("6. Individual Student Term - I Graph ")
    print("7. Individual Student Term - II Graph ")
    print("8. School Academic Graph ")
    op = int(input("Enter Your Choice : "))
    if op==1:
        StuLocGraph()
    elif op==2:
        StuClassCount()
```

```
    elif op==3:
        StuGender()
    elif op==4:
        ClassTerm1Graph()
    elif op==5:
        ClassTerm2Graph()
    elif op==6:
        StuTerm1Graph()
    elif op==7:
        StuTerm2Graph()
    elif op==8:
        SchAcadGraph()
    else:
        print("Invalid Choice . . . ")



def StuLocGraph():
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    df = dfStu['address'].value_counts()
    print(df)
    df.plot(kind='barh',color=['r','b','g','y','c','m'],width=0.9)
```

```
    plt.title("Student Address Graph")
    plt.xlabel("No. of Students ")
    plt.ylabel("Location")
    plt.show()


def StuClassCount():
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    df = dfStu['class'].value_counts()
    print(df)
    df.plot(kind='line',color=['b','r','y','g','m','c'],marker='D',linestyle='-.')
    plt.title("Class Count")
    plt.xlabel("Classes ")
    plt.ylabel("No. of students")
    plt.show()


def StuGender():
    dfStu = pd.read_csv(path,index_col = 'adm',\
                usecols =
['adm','name','class','father','mother','contact','dob','gender','address'])
    df = dfStu['gender'].value_counts()
    print(df)
```

```
    df.plot(kind='bar',color=['y','g'],width=0.9)
    plt.title("Gender Count")
    plt.xlabel("Gender ")
    plt.ylabel("No. of students")
    plt.grid(True)
    plt.show()

def ClassTerm1Graph():
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
    Class = input("Enter the class : ") ; Class = Class.upper()
    df = dfTerm1.loc[dfTerm1["class"]==Class]
    print(df)
    df.plot(kind='bar',x='name')
    plt.xlabel("Student's Name")
    plt.ylabel("Marks Scored out of 60")
    plt.title("Term - I Marks Analysis of Class "+Class)
    plt.show()

def ClassTerm2Graph():
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
    Class = input("Enter the class : ") ; Class = Class.upper()
    df = dfTerm2.loc[dfTerm2["class"]==Class]
    print(df)
```

```
    df.plot(kind='bar',x='name')
    plt.xlabel("Student's Name")
    plt.ylabel("Marks Scored out of 60")
    plt.title("Term - II Marks Analysis of Class "+Class)
    plt.show()

def StuTerm1Graph():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm')
    df = dfTerm1.loc[dfTerm1["name"]==name]
    if df.empty:
        print("Student doesnot exist ...")
    else:
        print("Student details are as follows : ")
        print(df[['name','class']])
        adm = int(input("Enter the Admission Number of the Student : "))
        df = df.loc[adm,['eng','hindi','maths','science','sst']]
        df.plot(kind='bar',color=['r','g','c','m','y'],width=0.8)
        plt.title("Term - I Marks Report of "+name)
        plt.xlabel("Subjects")
        plt.ylabel("Marks Scored out of 60")
        plt.show()
```

```
def StuTerm2Graph():
    name = input("Enter Student's Name : ") ; name = name.upper()
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm')
    df = dfTerm2.loc[dfTerm2["name"]==name]
    if df.empty:
        print("Student doesnot exist ...")
    else:
        print("Student details are as follows : ")
        print(df[['name','class']])
        adm = int(input("Enter the Admission Number of the Student : "))
        df = df.loc[adm,['eng','hindi','maths','science','sst']]
        df.plot(kind='bar',color=['r','g','c','m','y'],width=0.8)
        plt.title("Term - II Marks Report of "+name)
        plt.xlabel("Subjects")
        plt.ylabel("Marks Scored out of 60")
        plt.show()


def SchAcadGraph():
    dfTerm1 = pd.read_csv(term1path,index_col = 'adm',\
                          usecols = ['adm','eng','hindi','maths','science','sst'])
    dfTerm2 = pd.read_csv(term2path,index_col = 'adm',\
                          usecols = ['adm','eng','hindi','maths','science','sst'])
```

```
    df = dfTerm1+dfTerm2
    df2 = df.mean()

df2.plot(kind='bar',color=['r','g','c','m','y'],width=0.8,edgecolor='black',linestyle='--
')
    plt.xlabel("Subjects")
    plt.ylabel("Average")
    plt.title("School Academic Performance",fontsize=20)
    plt.show()



while True:
    print()
    print()
    Menu()
    ch = int(input("Enter your choice : "))
    if ch==1:
        ShowAllRecord()
    elif ch==2:
        StudentSearch()
    elif ch==3:
        AddStudent()
```

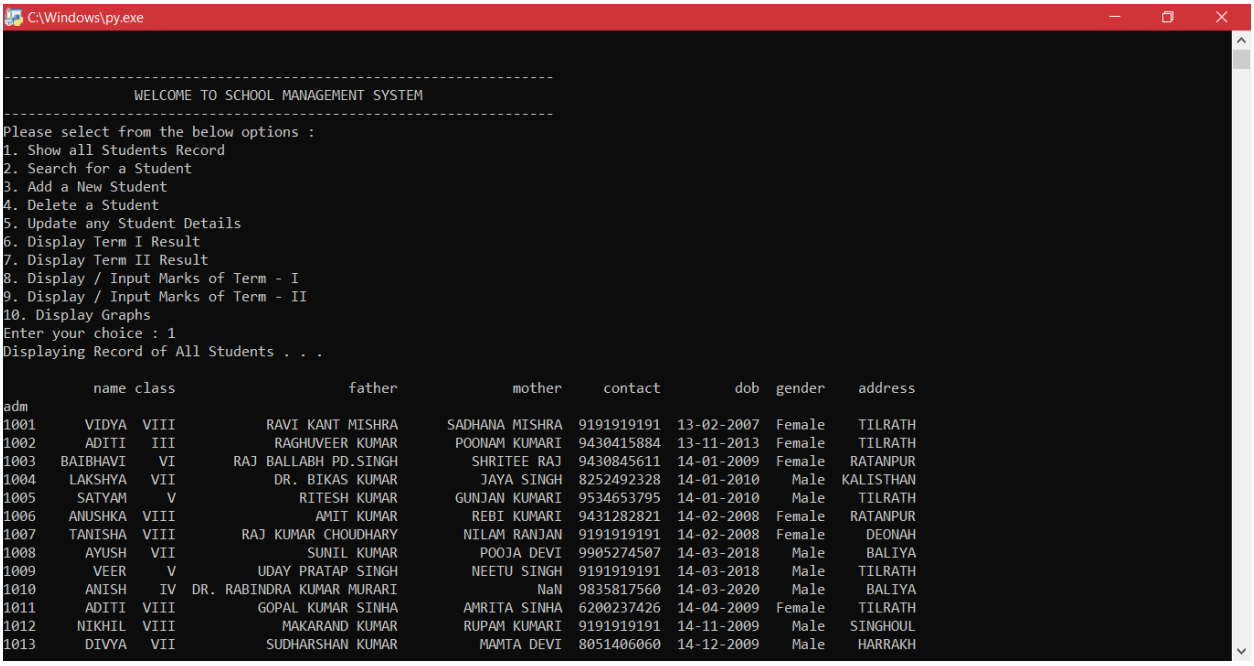
```
    elif ch==4:
        DeleteStudent()
    elif ch==5:
        UpdateStudentDetails()
    elif ch==6:
        ShowT1Result()
    elif ch==7:
        ShowT2Result()
    elif ch==8:
        UpdateTerm1Marks()
    elif ch==9:
        UpdateTerm2Marks()
    elif ch==10:
        ShowGraphs()
    else:
        print("Wrong Choice . . .")
        break


input()
```

7. WORKING OF THE SOFTWARE

1. Fetching record of all students:

User needs to press 1 after the start of the program to display all record of students stored in the database.

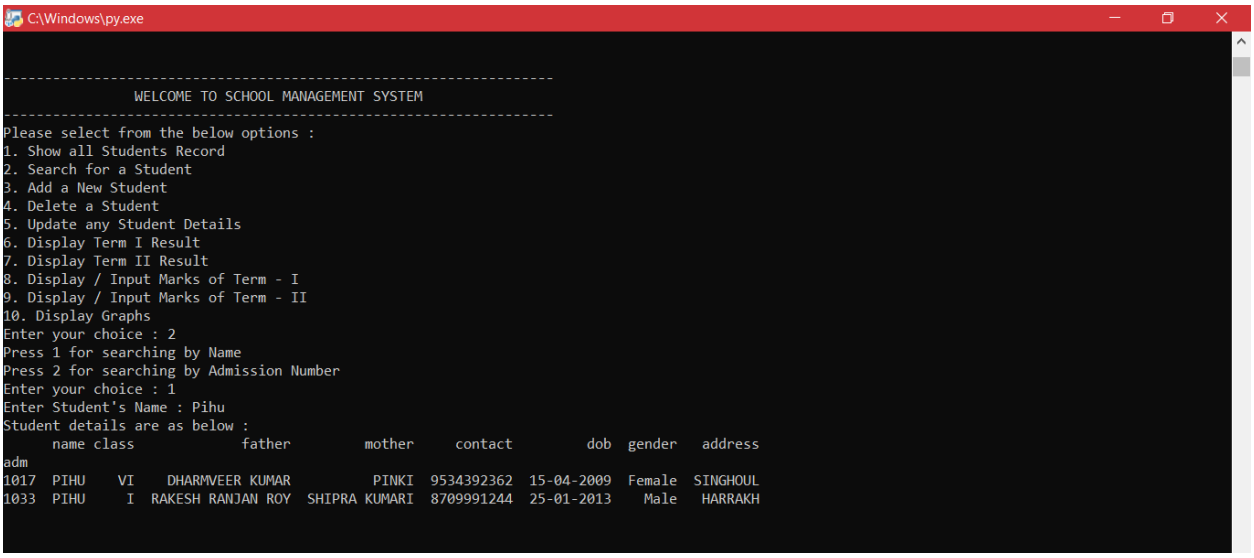


adm	name	class	father	mother	contact	dob	gender	address
1001	VIDYA	VIII	RAVI KANT MISHRA	SADHANA MISHRA	9191919191	13-02-2007	Female	TILRATH
1002	ADITI	III	RAGHUVEER KUMAR	POONAM KUMARI	9430415884	13-11-2013	Female	TILRATH
1003	BAIBHAVI	VI	RAJ BALLABH PD.SINGH	SHRITEE RAJ	9430845611	14-01-2009	Female	RATANPUR
1004	LAKSHYA	VII	DR. BIKAS KUMAR	JAYA SINGH	8252492328	14-01-2010	Male	KALISTHAN
1005	SATYAM	V	RITESH KUMAR	GUNJAN KUMARI	9534653795	14-01-2010	Male	TILRATH
1006	ANUSHKA	VIII	AMIT KUMAR	REBI KUMARI	9431282821	14-02-2008	Female	RATANPUR
1007	TANISHA	VIII	RAJ KUMAR CHOUDHARY	NILAM RANJAN	9191919191	14-02-2008	Female	DEONAH
1008	AYUSH	VII	SUNIL KUMAR	POOJA DEVI	9905274507	14-03-2018	Male	BALIYA
1009	VEER	V	UDAY PRATAP SINGH	NEETU SINGH	9191919191	14-03-2018	Male	TILRATH
1010	ANISH	IV	DR. RABINDRA KUMAR MURARI	NaN	9835817560	14-03-2020	Male	BALIYA
1011	ADITI	VIII	GOPAL KUMAR SINHA	AMRITA SINHA	6200237426	14-04-2009	Female	TILRATH
1012	NIKHIL	VIII	MAKARAND KUMAR	RUPAM KUMARI	9191919191	14-11-2009	Male	SINGHOUL
1013	DIVYA	VII	SUDHARSHAN KUMAR	MAMTA DEVI	8051406060	14-12-2009	Male	HARRAKH

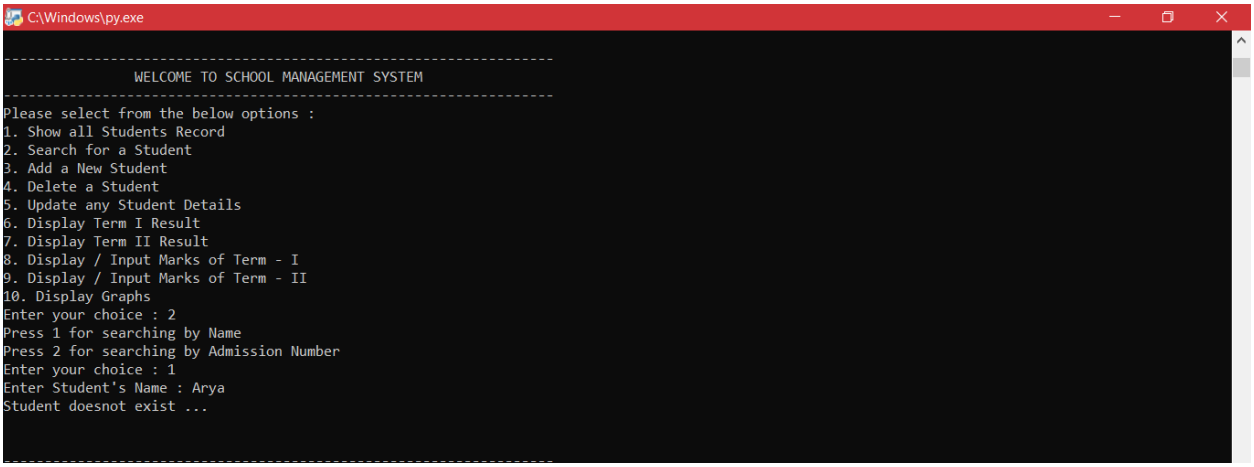
2. Fetching details of a student:

a. By Name

User needs to press 2 to search for a student and then press 1 to search by name.

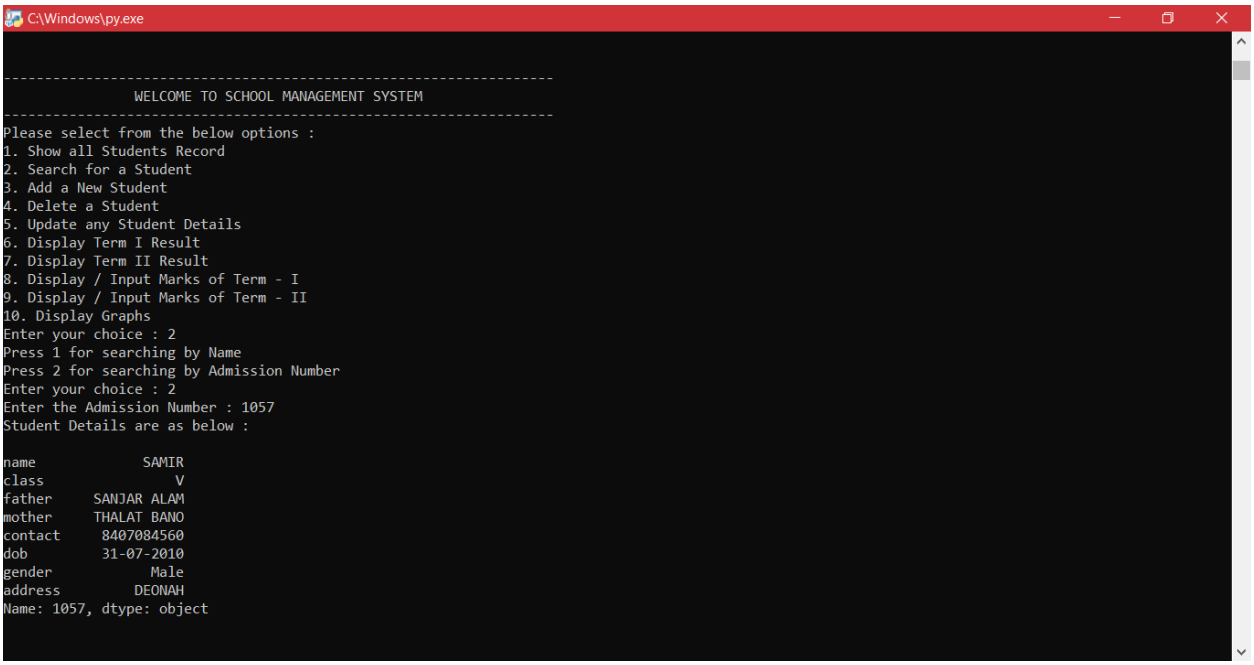


If name not found in the database, system prints student doesnot exist…

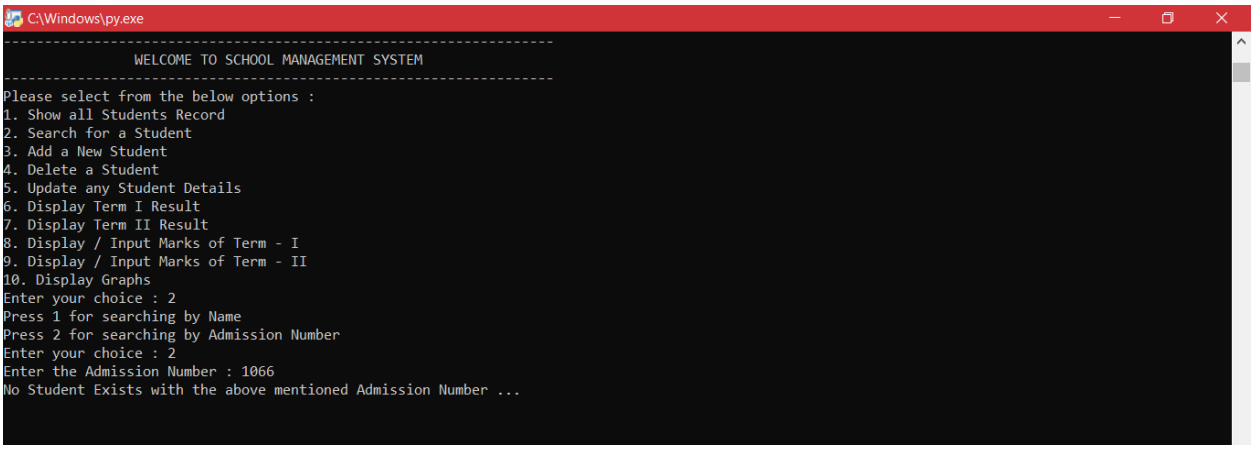


b. By Admission Number

User needs to press 2 to search for a student and then press 2 to search by Admission Number.

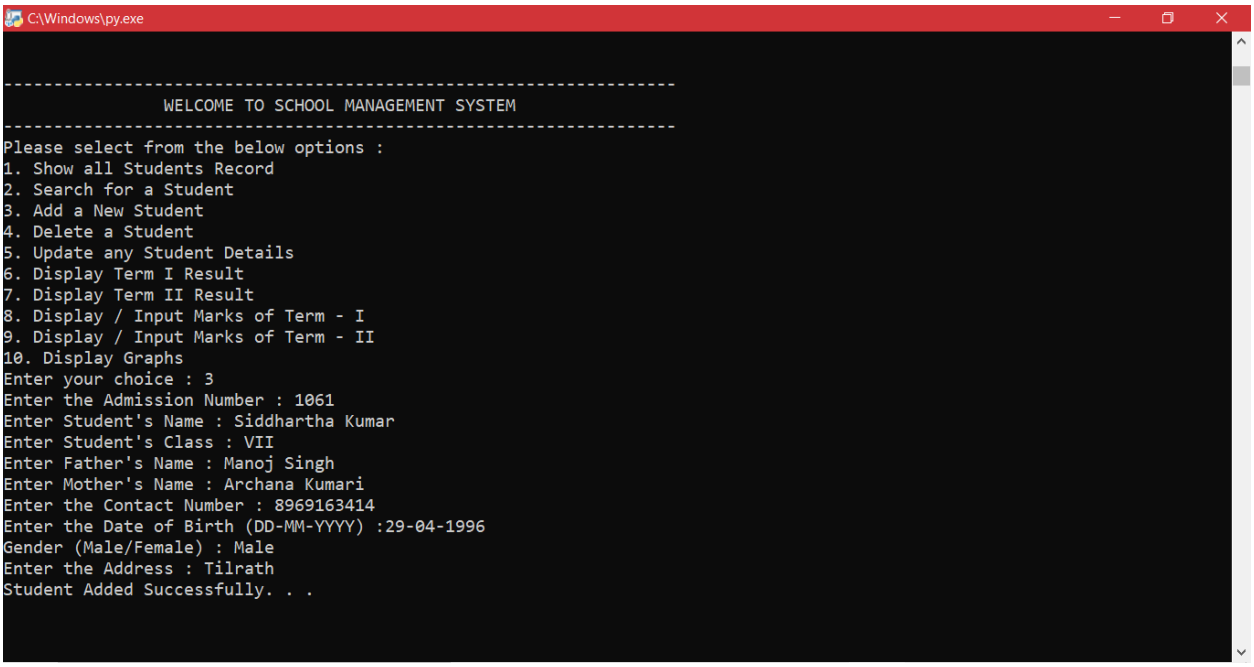


If Admission Number not found in the database, system prints No Student exists with the above mentioned admission number.

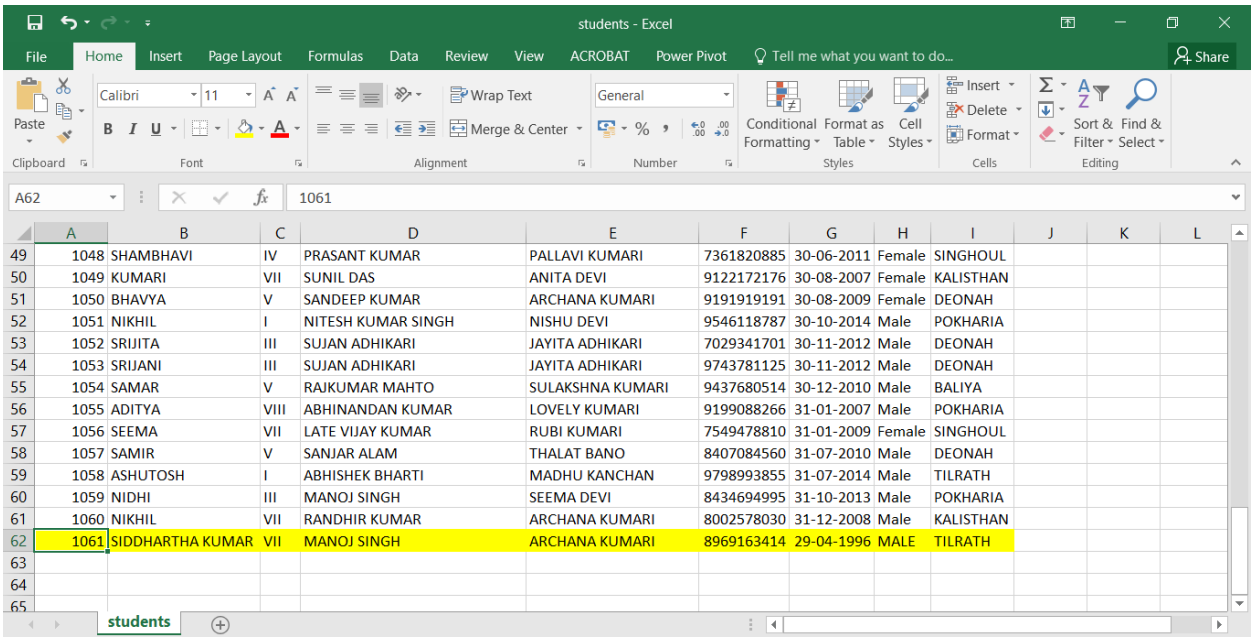


3. To add a student:

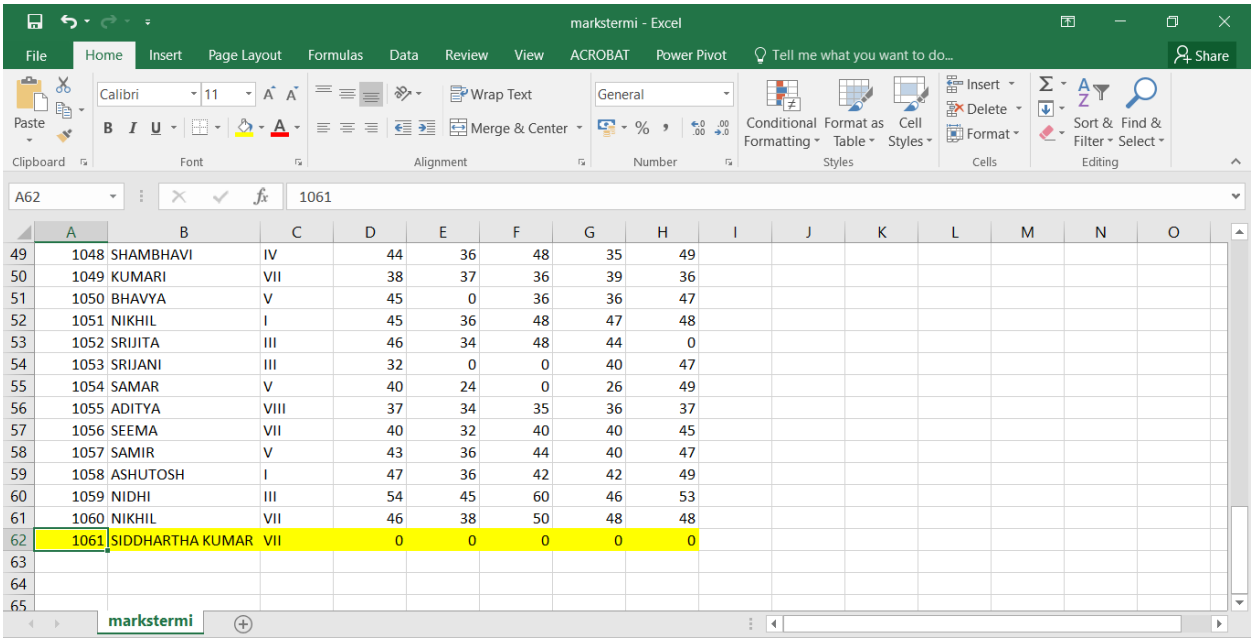
User needs to press 3 to add a student to the database. User needs to enter the admission number first.



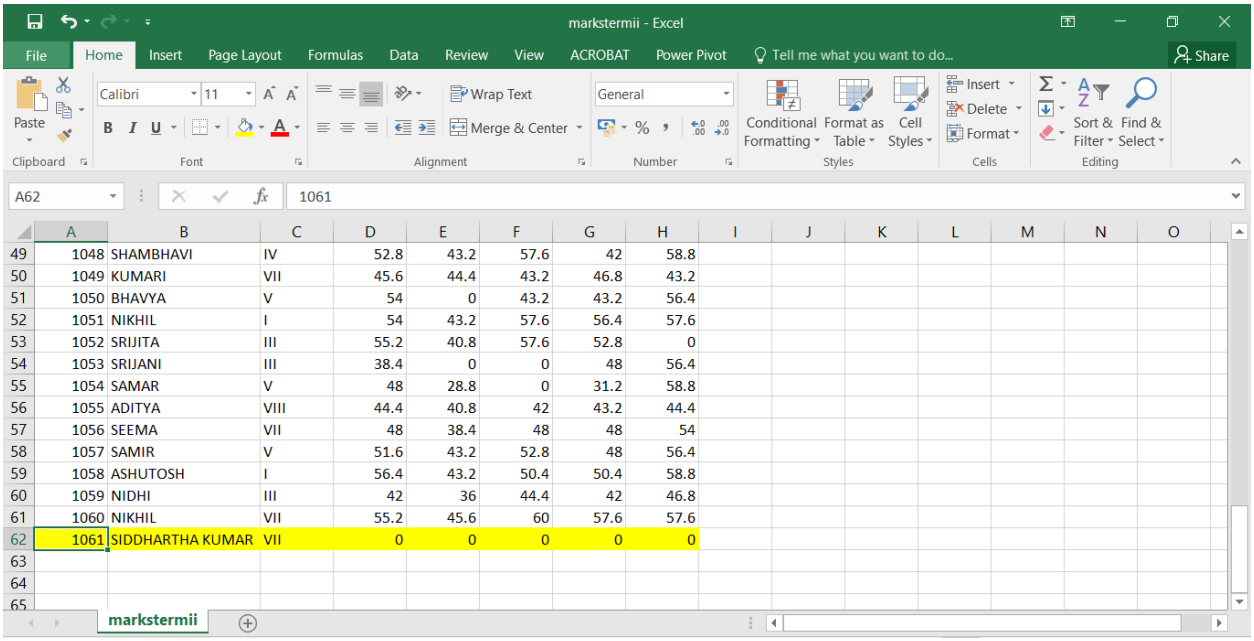
Adding a new student updates the students.csv table and inserts adm no, name and class to both the markstermi.csv and markstermii.csv files with default marks as 0 in all subjects.



	A	B	C	D	E	F	G	H	I	J	K	L
49	1048	SHAMBHAVI	IV	PRASANT KUMAR	PALLAVI KUMARI	7361820885	30-06-2011	Female	SINGHOUL			
50	1049	KUMARI	VII	SUNIL DAS	ANITA DEVI	9122172176	30-08-2007	Female	KALISTHAN			
51	1050	BHAVYA	V	SANDEEP KUMAR	ARCHANA KUMARI	9191919191	30-08-2009	Female	DEONAH			
52	1051	NIKHIL	I	NITESH KUMAR SINGH	NISHU DEVI	9546118787	30-10-2014	Male	POKHARIA			
53	1052	SRIJITA	III	SUJAN ADHIKARI	JAYITA ADHIKARI	7029341701	30-11-2012	Male	DEONAH			
54	1053	SRIJANI	III	SUJAN ADHIKARI	JAYITA ADHIKARI	9743781125	30-11-2012	Male	DEONAH			
55	1054	SAMAR	V	RAJKUMAR MAHTO	SULAKSHNA KUMARI	9437680514	30-12-2010	Male	BALIYA			
56	1055	ADITYA	VIII	ABHINANDAN KUMAR	LOVELY KUMARI	9199088266	31-01-2007	Male	POKHARIA			
57	1056	SEEMA	VII	LATE VIJAY KUMAR	RUBI KUMARI	7549478810	31-01-2009	Female	SINGHOUL			
58	1057	SAMIR	V	SANJAR ALAM	THALAT BANO	8407084560	31-07-2010	Male	DEONAH			
59	1058	ASHUTOSH	I	ABHISHEK BHARTI	MADHU KANCHAN	9798993855	31-07-2014	Male	TILRATH			
60	1059	NIDHI	III	MANOJ SINGH	SEEMA DEVI	8434694995	31-10-2013	Male	POKHARIA			
61	1060	NIKHIL	VII	RANDHIR KUMAR	ARCHANA KUMARI	8002578030	31-12-2008	Male	KALISTHAN			
62	1061	SIDDHARTHA KUMAR	VII	MANOJ SINGH	ARCHANA KUMARI	8969163414	29-04-1996	MALE	TILRATH			
63												
64												
65												

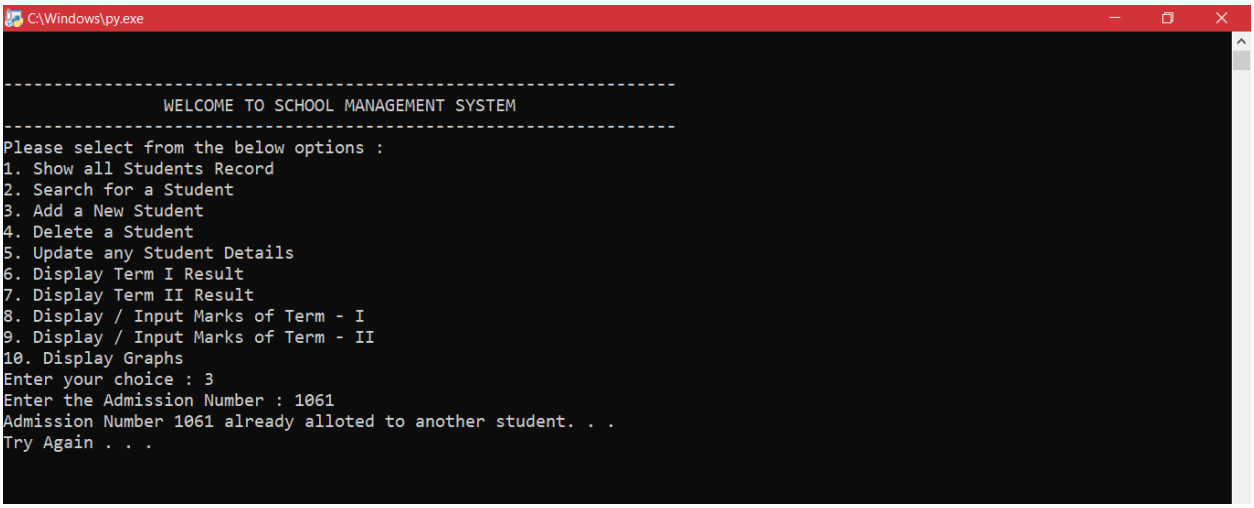


	A	B	C	D	E	F	G	H
49	1048	SHAMBHAVI	IV	44	36	48	35	49
50	1049	KUMARI	VII	38	37	36	39	36
51	1050	BHAVYA	V	45	0	36	36	47
52	1051	NIKHIL	I	45	36	48	47	48
53	1052	SRIJITA	III	46	34	48	44	0
54	1053	SRIJANI	III	32	0	0	40	47
55	1054	SAMAR	V	40	24	0	26	49
56	1055	ADITYA	VIII	37	34	35	36	37
57	1056	SEEMA	VII	40	32	40	40	45
58	1057	SAMIR	V	43	36	44	40	47
59	1058	ASHUTOSH	I	47	36	42	42	49
60	1059	NIDHI	III	54	45	60	46	53
61	1060	NIKHIL	VII	46	38	50	48	48
62	1061	SIDDHARTHA KUMAR	VII	0	0	0	0	0
63								
64								
65								

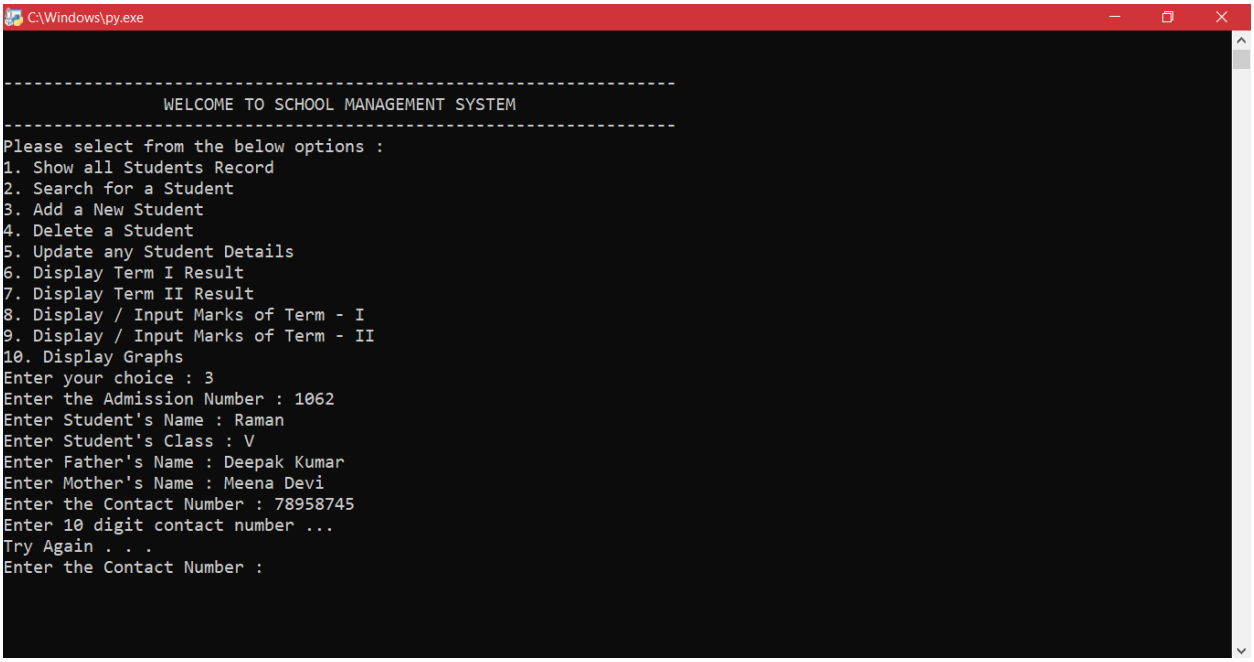


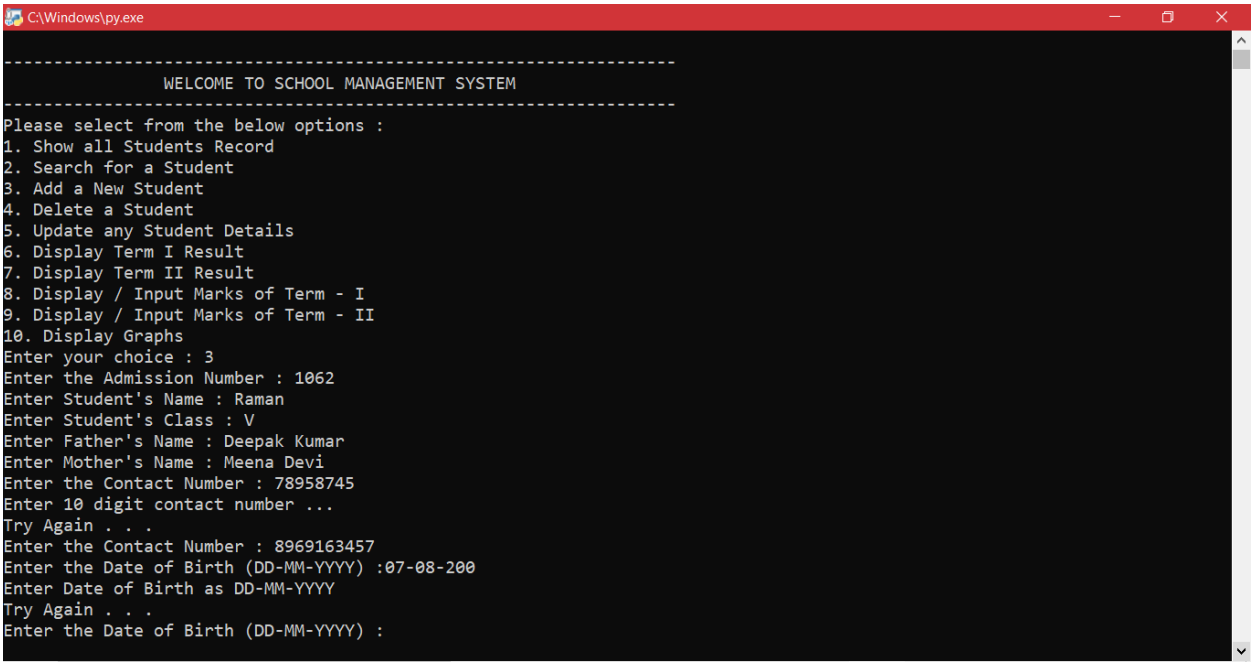
	A	B	C	D	E	F	G	H
49	1048	SHAMBHAVI	IV	52.8	43.2	57.6	42	58.8
50	1049	KUMARI	VII	45.6	44.4	43.2	46.8	43.2
51	1050	BHAVYA	V	54	0	43.2	43.2	56.4
52	1051	NIKHIL	I	54	43.2	57.6	56.4	57.6
53	1052	SRIJITA	III	55.2	40.8	57.6	52.8	0
54	1053	SRIJANI	III	38.4	0	0	48	56.4
55	1054	SAMAR	V	48	28.8	0	31.2	58.8
56	1055	ADITYA	VIII	44.4	40.8	42	43.2	44.4
57	1056	SEEMA	VII	48	38.4	48	48	54
58	1057	SAMIR	V	51.6	43.2	52.8	48	56.4
59	1058	ASHUTOSH	I	56.4	43.2	50.4	50.4	58.8
60	1059	NIDHI	III	42	36	44.4	42	46.8
61	1060	NIKHIL	VII	55.2	45.6	60	57.6	57.6
62	1061	SIDDHARTHA KUMAR	VII	0	0	0	0	0
63								
64								
65								

If the admission number is already allotted to another student system gives a warning message, Admission Number […] already allotted to another student. . .



The system also checks for integrity for contact number and date of birth. If contact number is found not equal to 10 digits, the system prompts to enter the contact number again and it's the same as in date of birth.

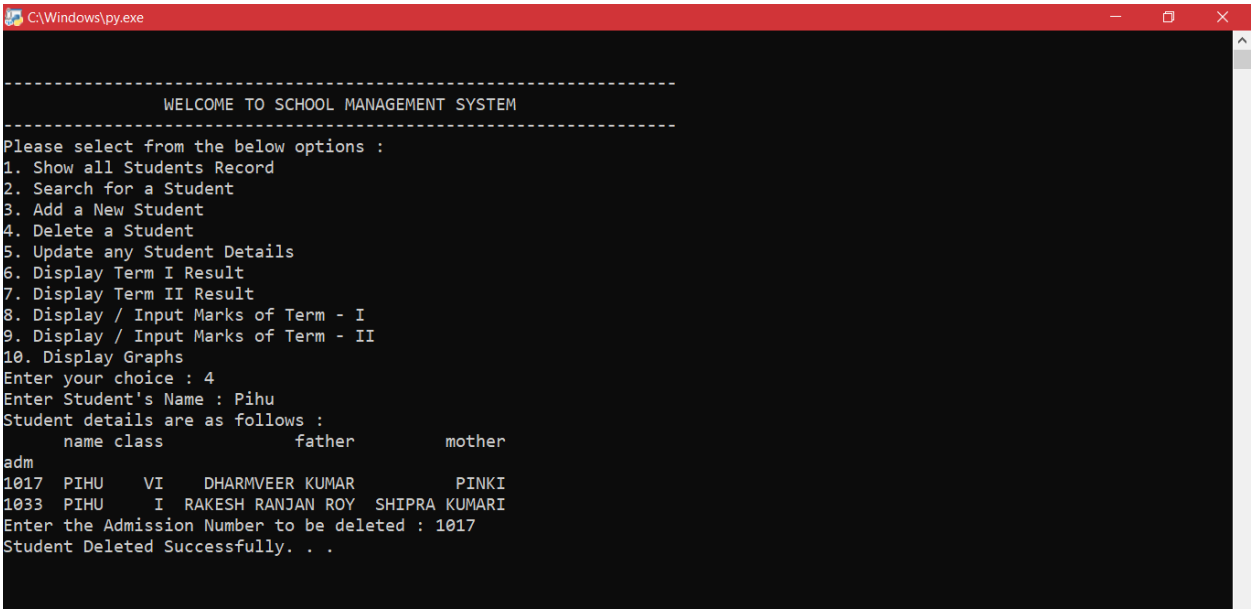



C:\Windows\py.exe
WELCOME TO SCHOOL MANAGEMENT SYSTEM
Please select from the below options :
1. Show all Students Record
2. Search for a Student
3. Add a New Student
4. Delete a Student
5. Update any Student Details
6. Display Term I Result
7. Display Term II Result
8. Display / Input Marks of Term - I
9. Display / Input Marks of Term - II
10. Display Graphs
Enter your choice : 3
Enter the Admission Number : 1062
Enter Student's Name : Raman
Enter Student's Class : V
Enter Father's Name : Deepak Kumar
Enter Mother's Name : Meena Devi
Enter the Contact Number : 78958745
Enter 10 digit contact number ...
Try Again . . .
Enter the Contact Number : 8969163457
Enter the Date of Birth (DD-MM-YYYY) :07-08-200
Enter Date of Birth as DD-MM-YYYY
Try Again . . .
Enter the Date of Birth (DD-MM-YYYY) :

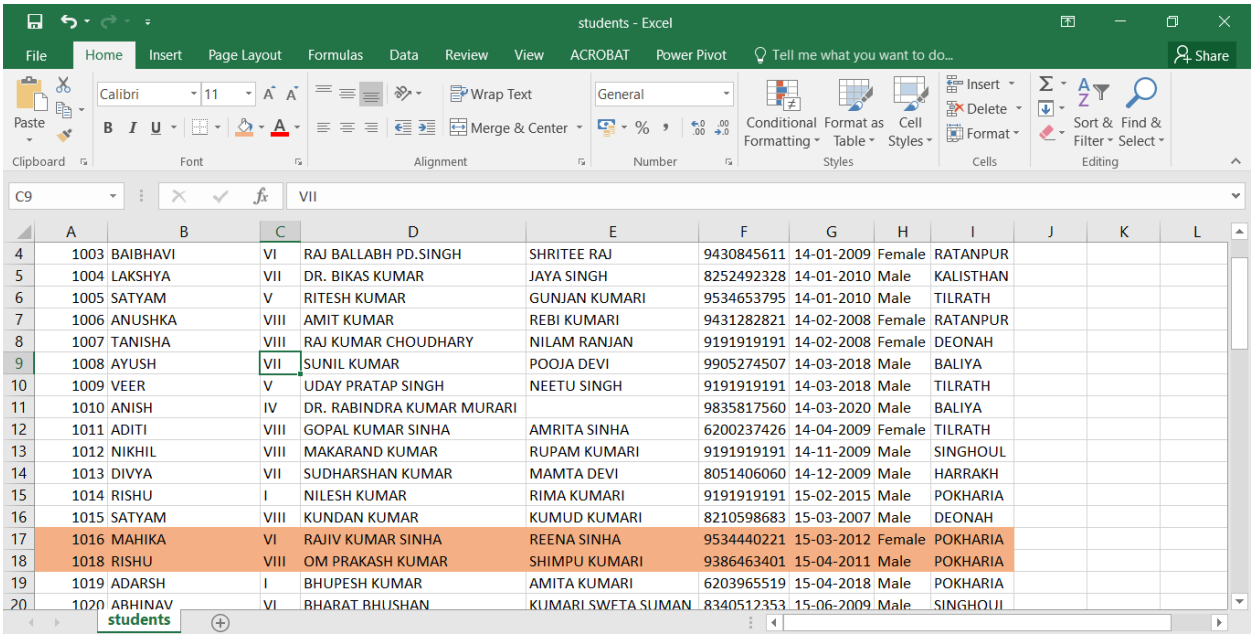
4. Delete a student

User needs to press 4 to delete a student. The system first asks for name to be entered. It shows the list of students with the matching names. The user then needs to enter the correct admission number of the child to delete. If the name is not found in the database, system prints no student found.

The data is deleted from all students.csv, markstermi.csv and markstermii.csv files.

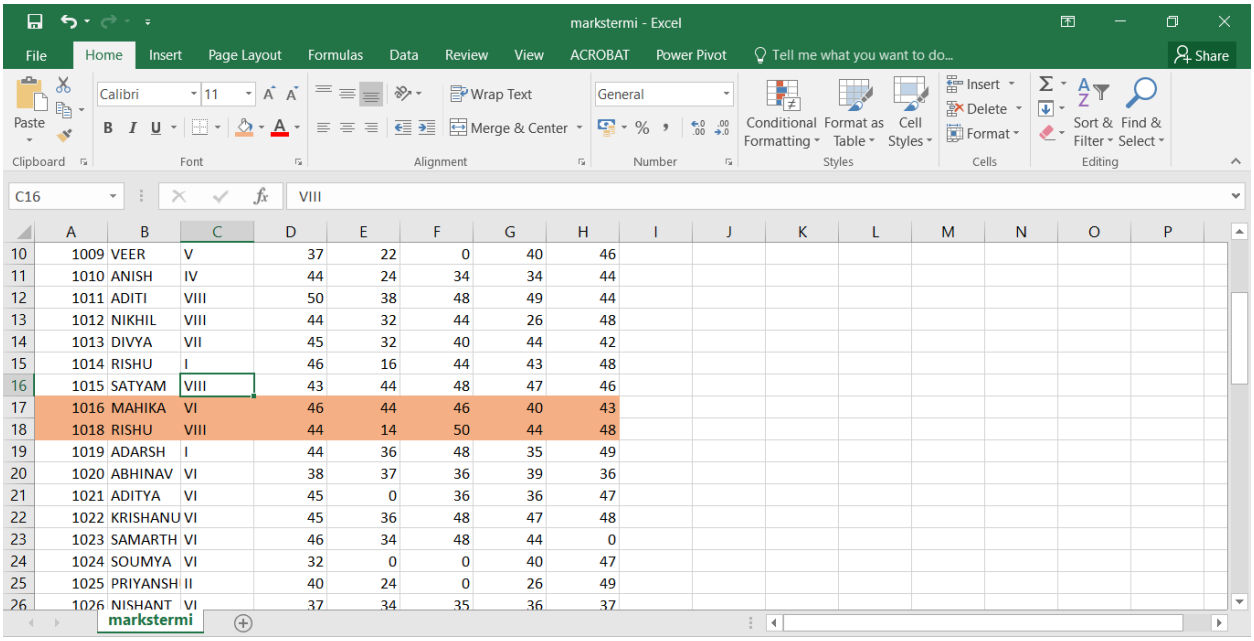


students.csv



	A	B	C	D	E	F	G	H	I	J	K	L
4	1003	BAIBHAVI	VI	RAJ BALLABH PD.SINGH	SHRITEE RAJ	9430845611	14-01-2009	Female	RATANPUR			
5	1004	LAKSHYA	VII	DR. BIKAS KUMAR	JAYA SINGH	8252492328	14-01-2010	Male	KALISTHAN			
6	1005	SATYAM	V	RITESH KUMAR	GUNJAN KUMARI	9534653795	14-01-2010	Male	TILRATH			
7	1006	ANUSHKA	VIII	AMIT KUMAR	REBI KUMARI	9431282821	14-02-2008	Female	RATANPUR			
8	1007	TANISHA	VIII	RAJ KUMAR CHOUDHARY	NILAM RANJAN	9191919191	14-02-2008	Female	DEONAH			
9	1008	AYUSH	VII	SUNIL KUMAR	POOJA DEVI	9905274507	14-03-2018	Male	BALIYA			
10	1009	VEER	V	UDAY PRATAP SINGH	NEETU SINGH	9191919191	14-03-2018	Male	TILRATH			
11	1010	ANISH	IV	DR. RABINDRA KUMAR MURARI		9835817560	14-03-2020	Male	BALIYA			
12	1011	ADITI	VIII	GOPAL KUMAR SINHA	AMRITA SINHA	6200237426	14-04-2009	Female	TILRATH			
13	1012	NIKHIL	VIII	MAKARAND KUMAR	RUPAM KUMARI	9191919191	14-11-2009	Male	SINGHOUL			
14	1013	DIVYA	VII	SUDHARSHAN KUMAR	MAMTA DEVI	8051406060	14-12-2009	Male	HARRAKH			
15	1014	RISHU	I	NILESH KUMAR	RIMA KUMARI	9191919191	15-02-2015	Male	POKHARIA			
16	1015	SATYAM	VIII	KUNDAN KUMAR	KUMUD KUMARI	8210598683	15-03-2007	Male	DEONAH			
17	1016	MAHIKA	VI	RAJIV KUMAR SINHA	REENA SINHA	9534440221	15-03-2012	Female	POKHARIA			
18	1018	RISHU	VIII	OM PRAKASH KUMAR	SHIMPU KUMARI	9386463401	15-04-2011	Male	POKHARIA			
19	1019	ADARSH	I	BHUPESH KUMAR	AMITA KUMARI	6203965519	15-04-2018	Male	POKHARIA			
20	1020	ABHINAV	VI	BHARAT BHUSHAN	KUMARI SWETA SUMAN	8340512353	15-06-2009	Male	SINGHOUL			

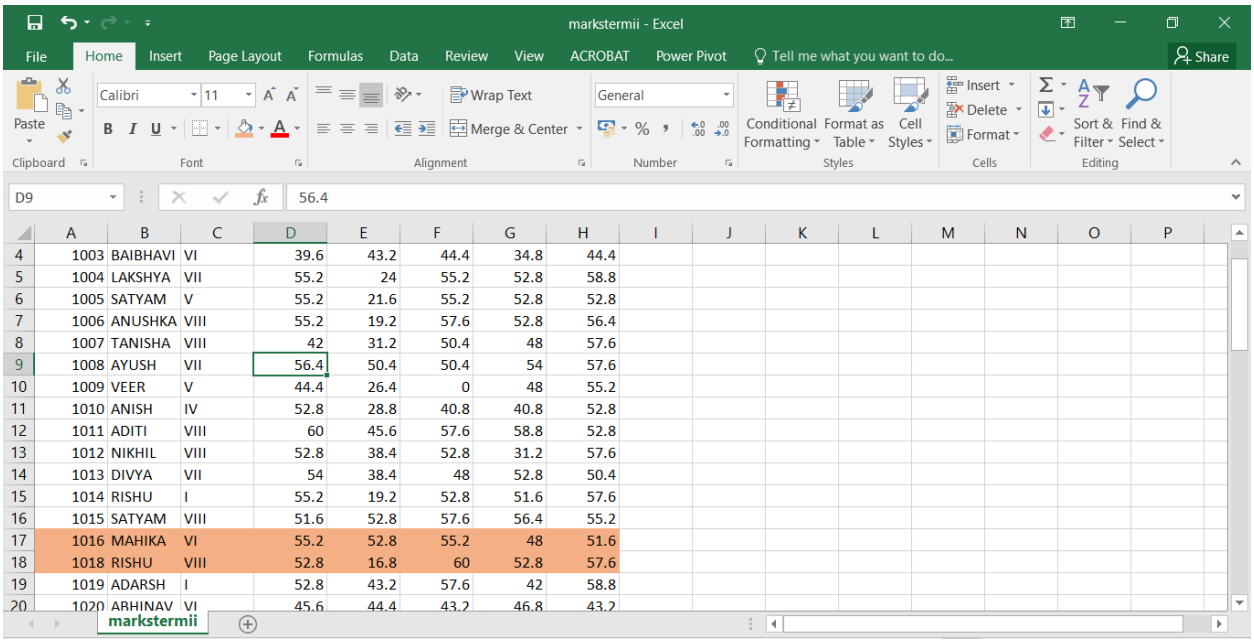
marksstermi.csv



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
10	1009	VEER	V	37	22	0	40	46								
11	1010	ANISH	IV	44	24	34	34	44								
12	1011	ADITI	VIII	50	38	48	49	44								
13	1012	NIKHIL	VIII	44	32	44	26	48								
14	1013	DIVYA	VII	45	32	40	44	42								
15	1014	RISHU	I	46	16	44	43	48								
16	1015	SATYAM	VIII	43	44	48	47	46								
17	1016	MAHIKA	VI	46	44	46	40	43								
18	1018	RISHU	VIII	44	14	50	44	48								
19	1019	ADARSH	I	44	36	48	35	49								
20	1020	ABHINAV	VI	38	37	36	39	36								
21	1021	ADITYA	VI	45	0	36	36	47								
22	1022	KRISHANU	VI	45	36	48	47	48								
23	1023	SAMARTH	VI	46	34	48	44	0								
24	1024	SOUMYA	VI	32	0	0	40	47								
25	1025	PRIYANSH	II	40	24	0	26	49								
26	1026	NISHANT	VI	37	34	35	36	37								

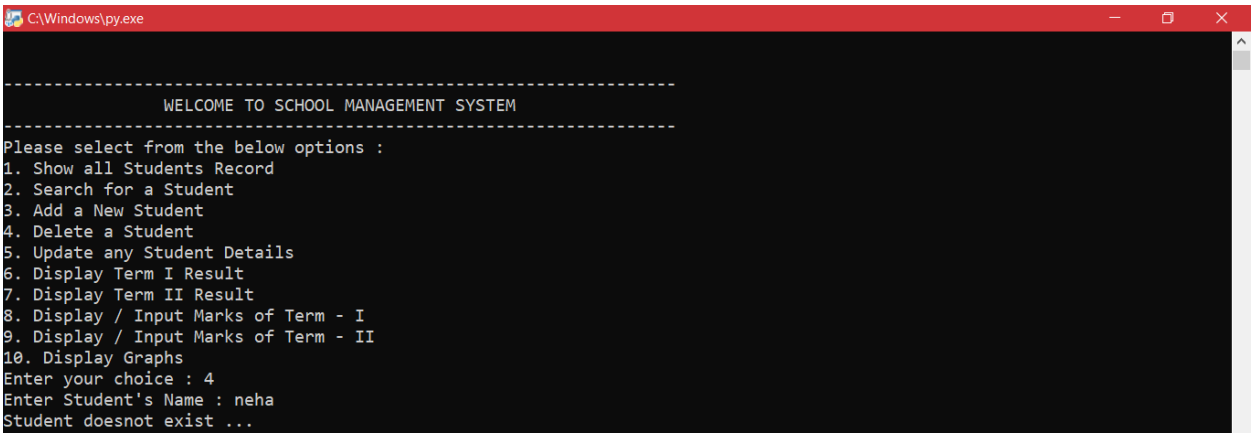
marksstermi

marksstermii.csv



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
4	1003	BAIBHAVI	VI	39.6	43.2	44.4	34.8	44.4								
5	1004	LAKSHYA	VII	55.2	24	55.2	52.8	58.8								
6	1005	SATYAM	V	55.2	21.6	55.2	52.8	52.8								
7	1006	ANUSHKA	VIII	55.2	19.2	57.6	52.8	56.4								
8	1007	TANISHA	VIII	42	31.2	50.4	48	57.6								
9	1008	AYUSH	VII	56.4	50.4	50.4	54	57.6								
10	1009	VEER	V	44.4	26.4	0	48	55.2								
11	1010	ANISH	IV	52.8	28.8	40.8	40.8	52.8								
12	1011	ADITI	VIII	60	45.6	57.6	58.8	52.8								
13	1012	NIKHIL	VIII	52.8	38.4	52.8	31.2	57.6								
14	1013	DIVYA	VII	54	38.4	48	52.8	50.4								
15	1014	RISHU	I	55.2	19.2	52.8	51.6	57.6								
16	1015	SATYAM	VIII	51.6	52.8	57.6	56.4	55.2								
17	1016	MAHIKA	VI	55.2	52.8	55.2	48	51.6								
18	1018	RISHU	VIII	52.8	16.8	60	52.8	57.6								
19	1019	ADARSH	I	52.8	43.2	57.6	42	58.8								
20	1020	ABHINAV	VI	45.6	44.4	43.2	46.8	43.2								

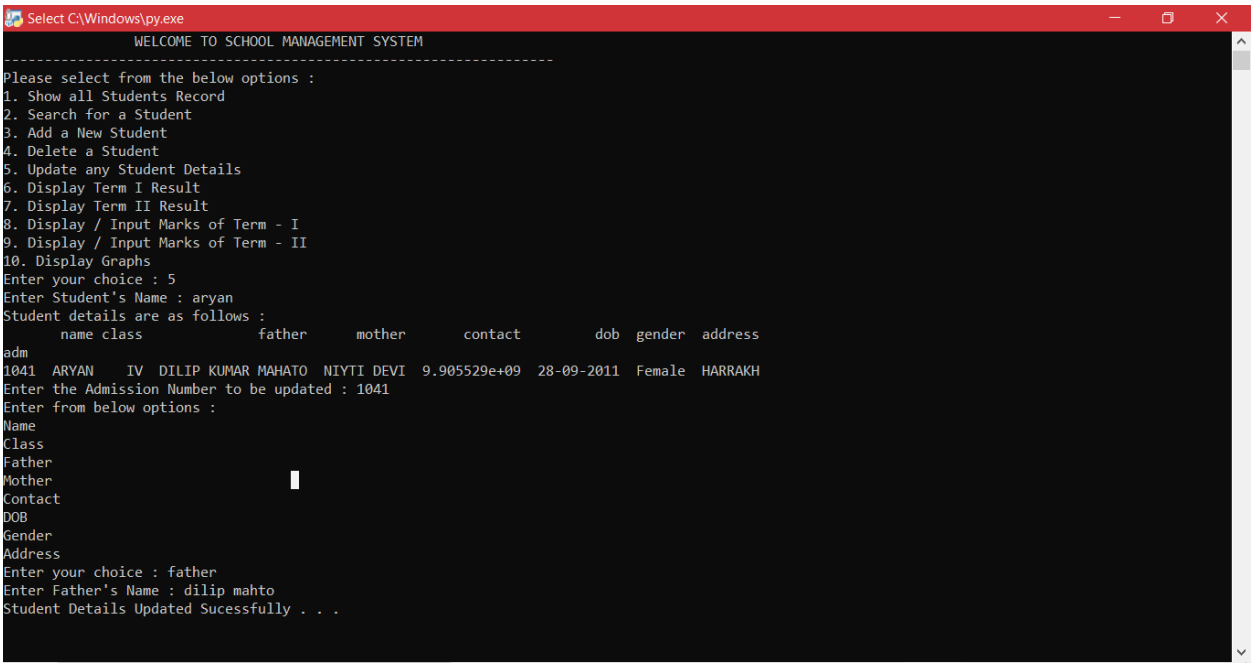
marksstermii

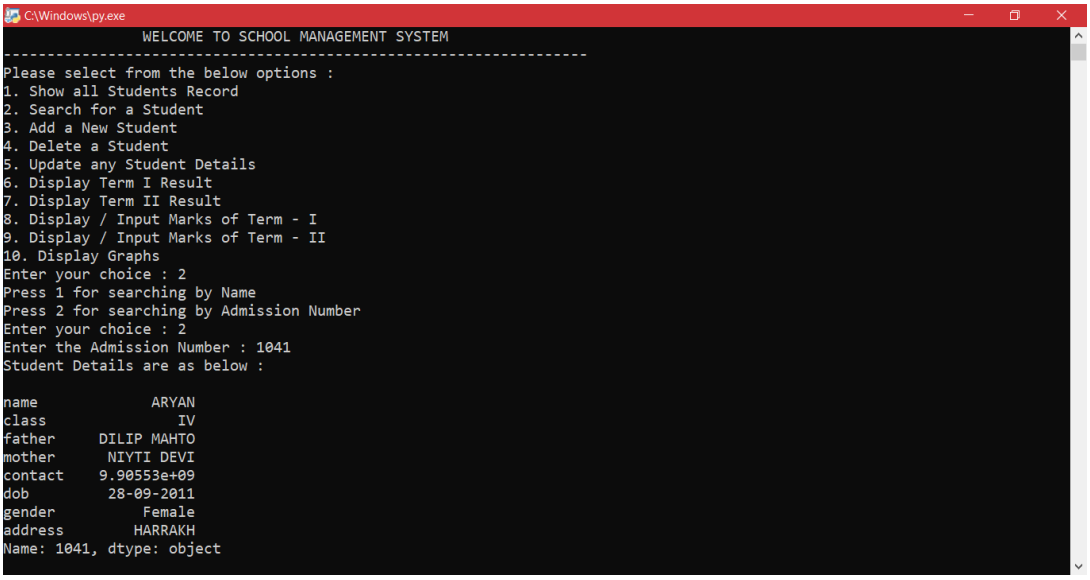

C:\Windows\py.exe
WELCOME TO SCHOOL MANAGEMENT SYSTEM
Please select from the below options :
1. Show all Students Record
2. Search for a Student
3. Add a New Student
4. Delete a Student
5. Update any Student Details
6. Display Term I Result
7. Display Term II Result
8. Display / Input Marks of Term - I
9. Display / Input Marks of Term - II
10. Display Graphs
Enter your choice : 4
Enter Student's Name : neha
Student doesnot exist ...

5. Update a student details:

User needs to press 5 to update a student record in database. The system first asks for name to be entered. It shows the list of students with the matching names. The user then needs to enter the correct admission number of the child to update.

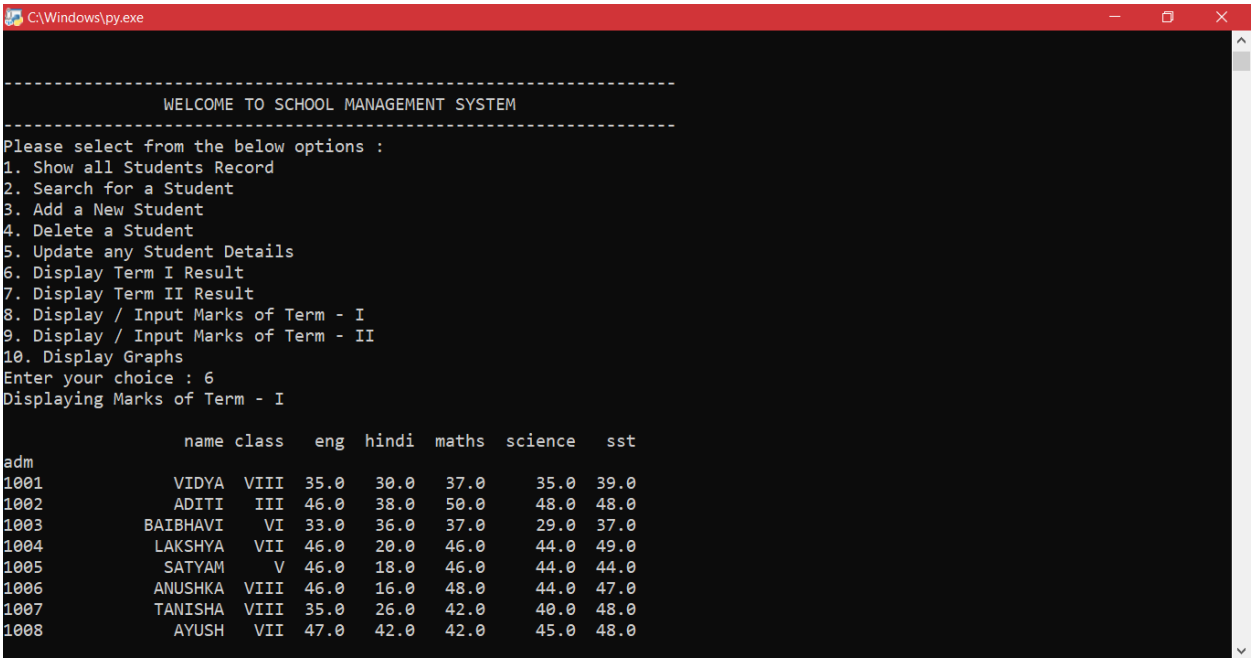
Then the user needs to select from a list of options to be entered for updating like name, class, father name, mother name, etc. If the name is not found in the database, system prints no student found.





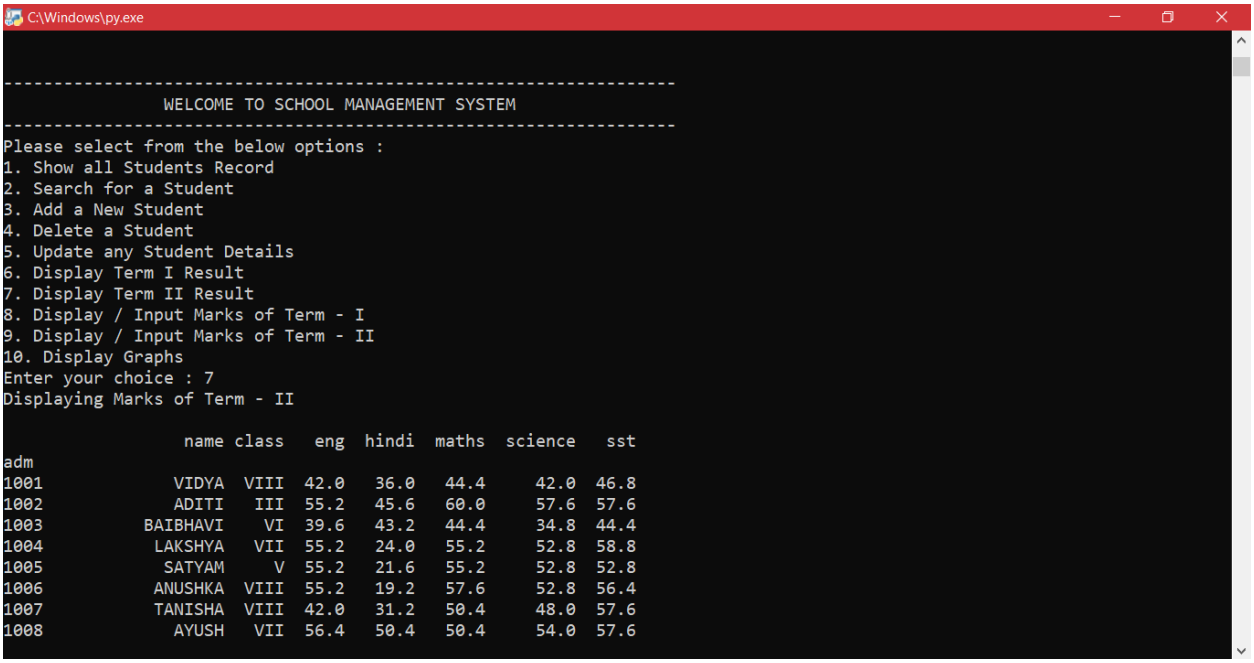
6. Display Term-I Marks:

User needs to press 6 to display Term-I Marks of entire school.



7. Display Term-II Marks:

User needs to press 6 to display Term-II Marks of entire school.

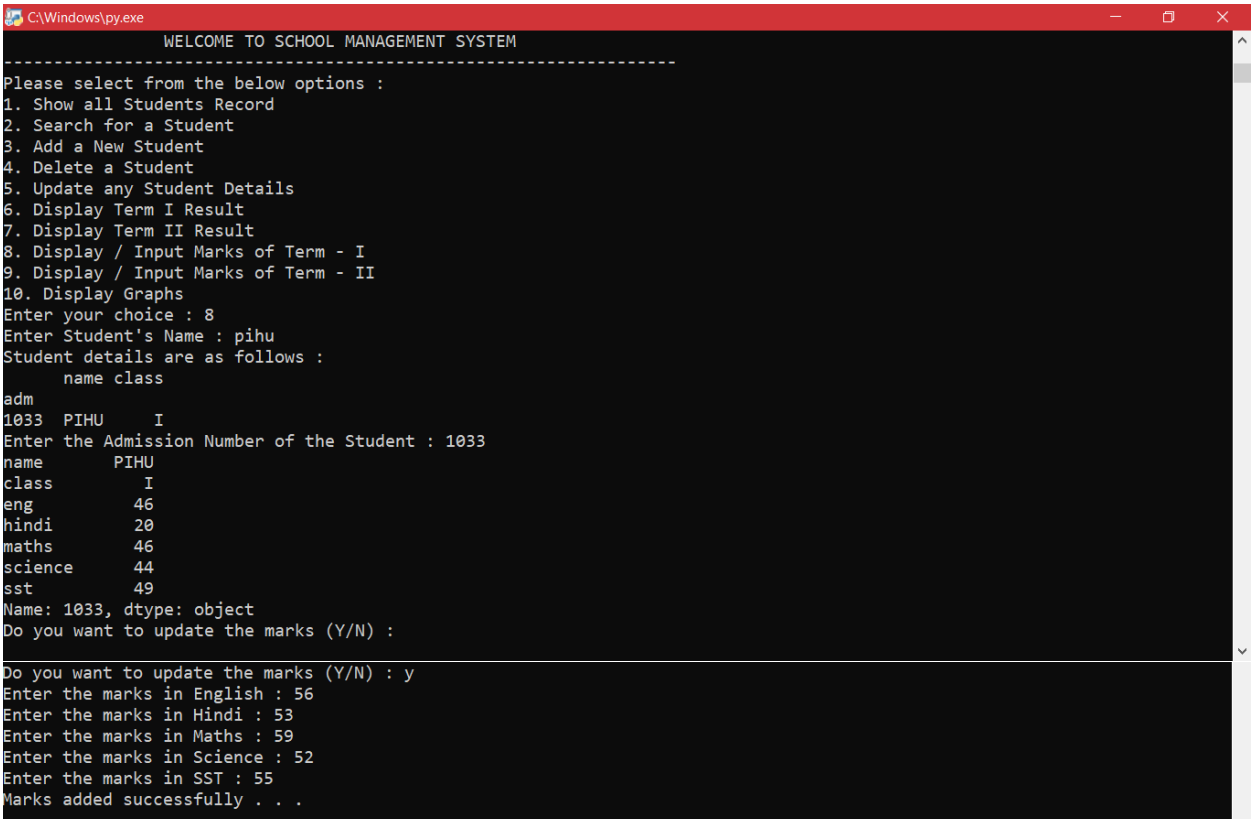


8. Display/ Input Marks of Term-I:

In this module, we can either view the marks of a particular student or we can update or enter the fresh marks of the student.

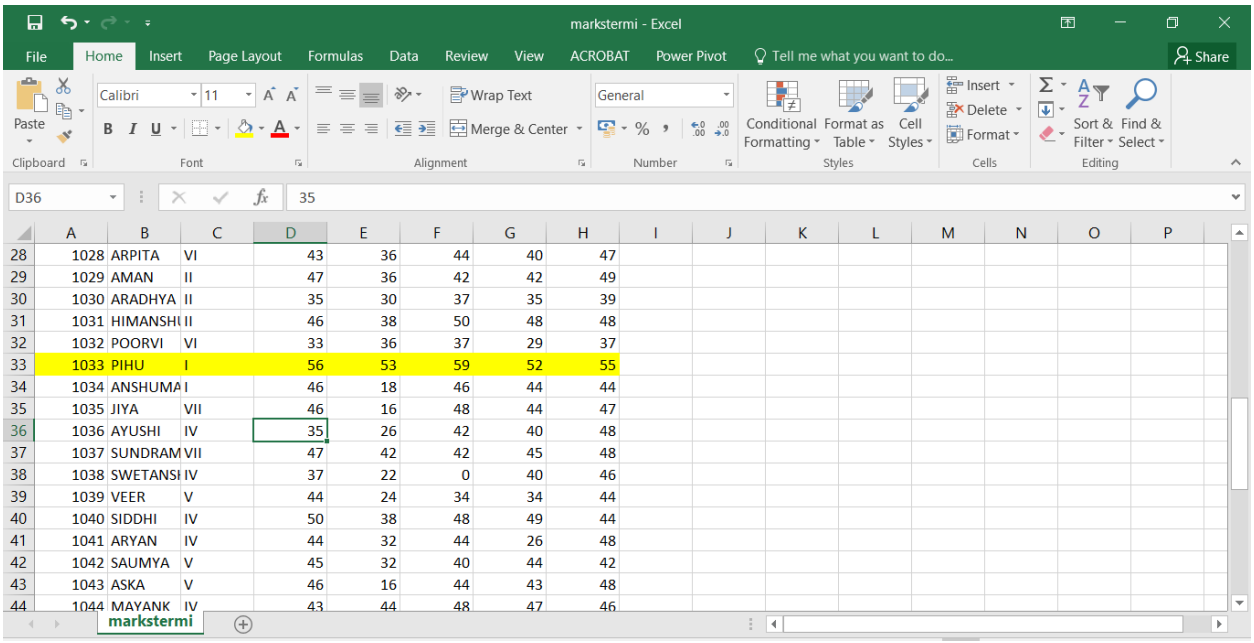
To do this user needs to first press 8 and then type the student name. The system will display the list of matching names from students.csv file and then the user needs to enter the admission number for which he/she wants to display the marks.

In the same module, system prints a statement, do you want to update the marks, then by pressing 'Y' for Yes and 'N' for No. the system does accordingly. If the user presses 'Y' then the system asks for marks in various subjects and stores it in the markstermi.csv file.



```
C:\Windows\py.exe
                WELCOME TO SCHOOL MANAGEMENT SYSTEM
--------------------------------------------------------------------
Please select from the below options :
1. Show all Students Record
2. Search for a Student
3. Add a New Student
4. Delete a Student
5. Update any Student Details
6. Display Term I Result
7. Display Term II Result
8. Display / Input Marks of Term - I
9. Display / Input Marks of Term - II
10. Display Graphs
Enter your choice : 8
Enter Student's Name : pihu
Student details are as follows :
      name class
adm
1033  PIHU     I
Enter the Admission Number of the Student : 1033
name       PIHU
class         I
eng          56
hindi        53
maths        59
science      52
sst          55
Name: 1033, dtype: object
```

markstermi.csv



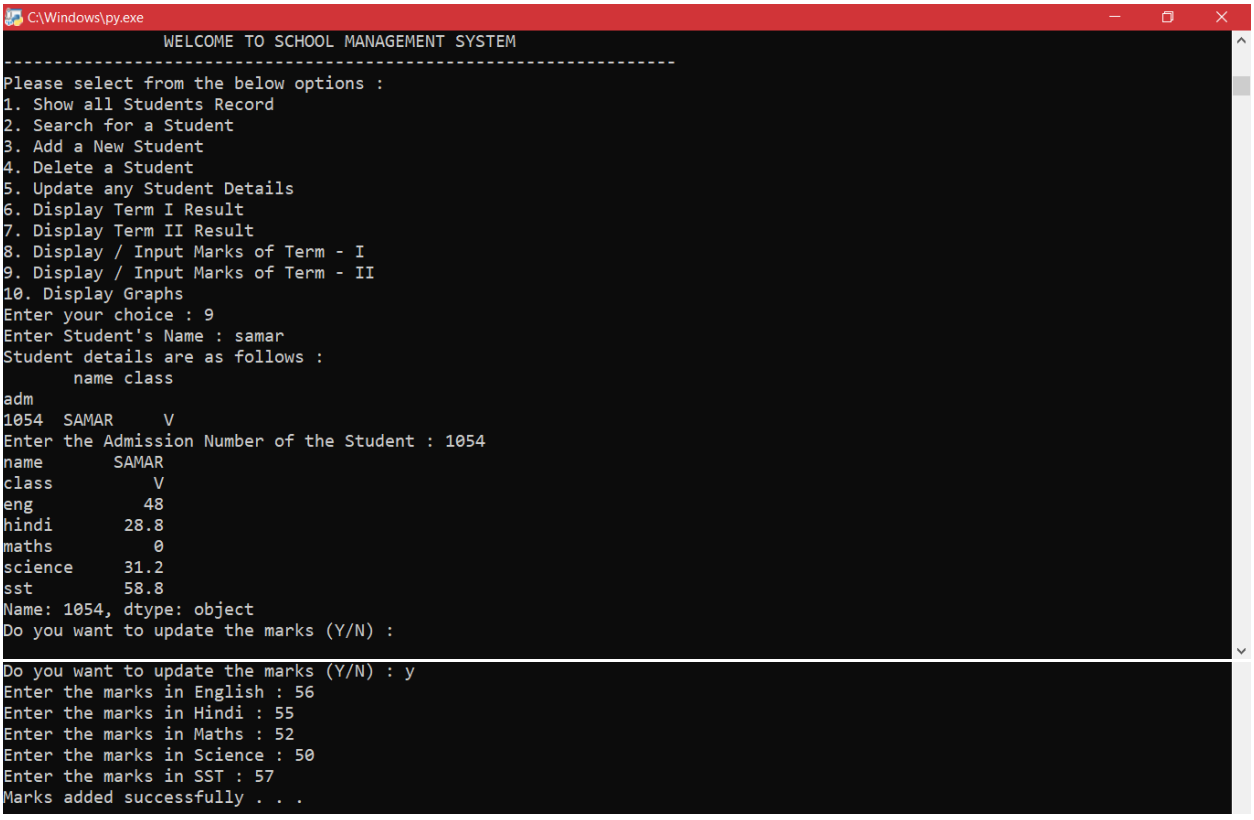
	A	B	C	D	E	F	G	H
28	1028	ARPITA	VI	43	36	44	40	47
29	1029	AMAN	II	47	36	42	42	49
30	1030	ARADHYA	II	35	30	37	35	39
31	1031	HIMANSHI	II	46	38	50	48	48
32	1032	POORVI	VI	33	36	37	29	37
33	1033	PIHU	I	56	53	59	52	55
34	1034	ANSHUMA	I	46	18	46	44	44
35	1035	JIYA	VII	46	16	48	44	47
36	1036	AYUSHI	IV	35	26	42	40	48
37	1037	SUNDRAM	VII	47	42	42	45	48
38	1038	SWETANSI	IV	37	22	0	40	46
39	1039	VEER	V	44	24	34	34	44
40	1040	SIDDHI	IV	50	38	48	49	44
41	1041	ARYAN	IV	44	32	44	26	48
42	1042	SAUMYA	V	45	32	40	44	42
43	1043	ASKA	V	46	16	44	43	48
44	1044	MAYANK	IV	43	44	48	47	46

9. Display/ Input Marks of Term-II:

In this module, we can either view the marks of a particular student or we can update or enter the fresh marks of the student.

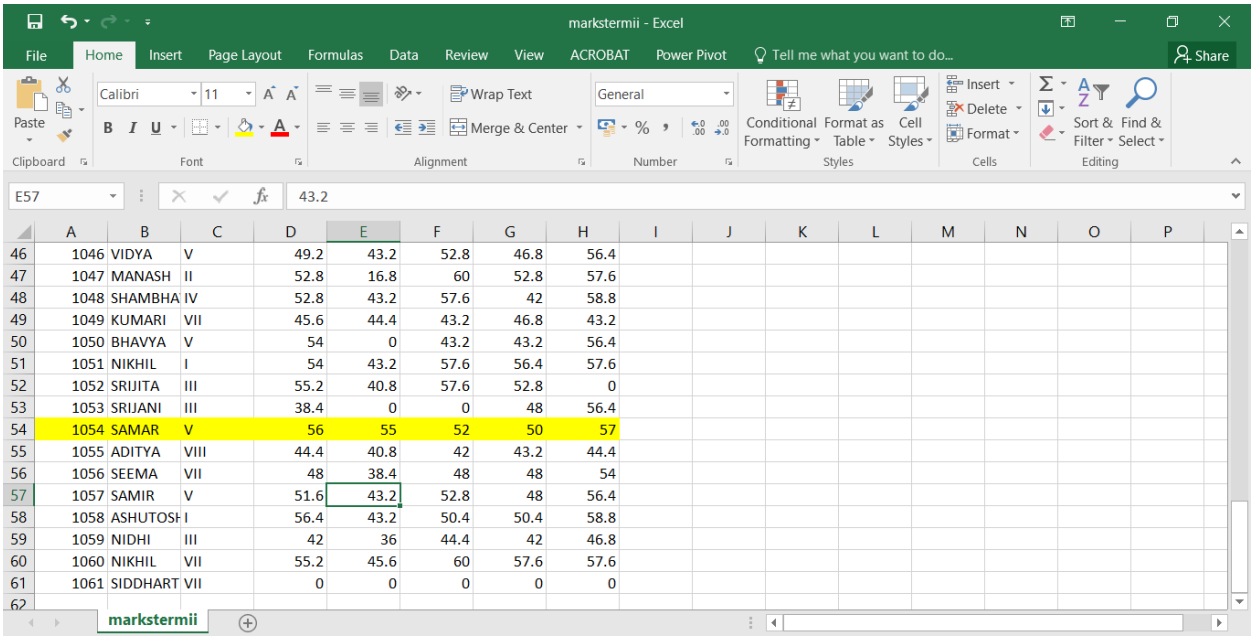
To do this user needs to first press 9 and then type the student name. The system will display the list of matching names from students.csv file and then the user needs to enter the admission number for which he/she wants to display the marks.

In the same module, system prints a statement, do you want to update the marks, then by pressing 'Y' for Yes and 'N' for No. the system does accordingly. If the user presses 'Y' then the system asks for marks in various subjects and stores it in the markstermii.csv file.



```
C:\Windows\py.exe
                WELCOME TO SCHOOL MANAGEMENT SYSTEM
----------------------------------------------------------------------
Please select from the below options :
1. Show all Students Record
2. Search for a Student
3. Add a New Student
4. Delete a Student
5. Update any Student Details
6. Display Term I Result
7. Display Term II Result
8. Display / Input Marks of Term - I
9. Display / Input Marks of Term - II
10. Display Graphs
Enter your choice : 9
Enter Student's Name : samar
Student details are as follows :
       name class
adm
1054  SAMAR     V
Enter the Admission Number of the Student : 1054
name        SAMAR
class           V
eng            56
hindi          55
maths          52
science        50
sst            57
Name: 1054, dtype: object
```

markstermii.csv



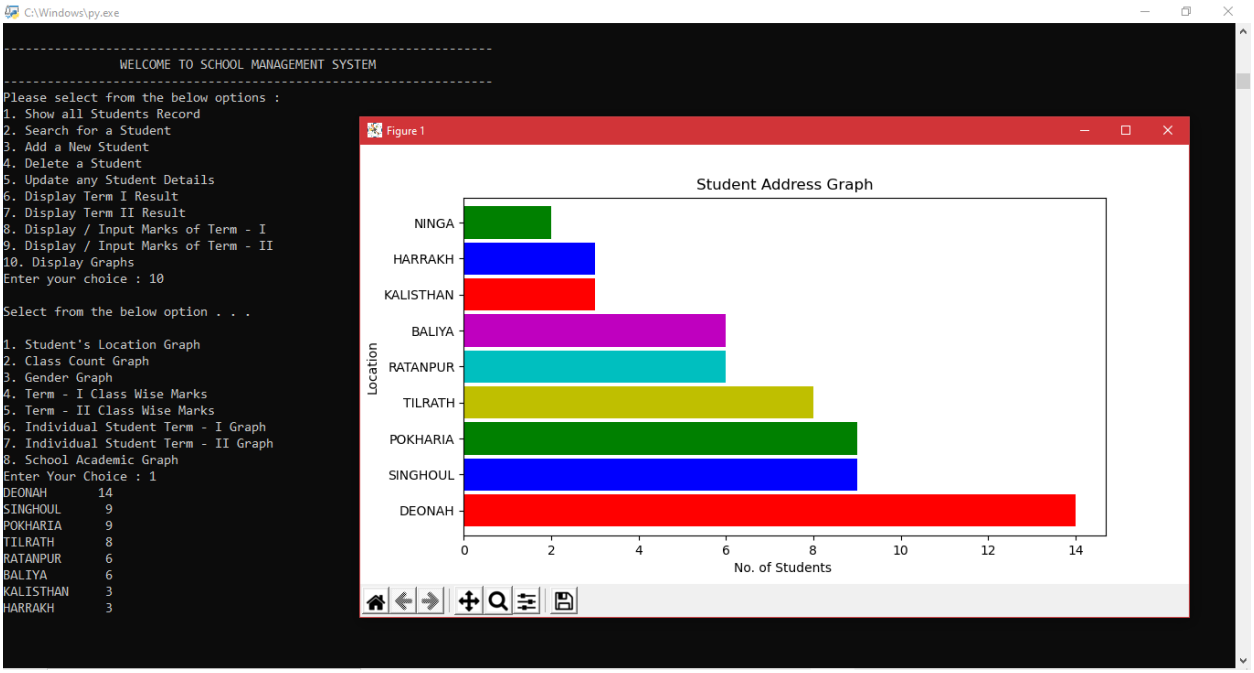
	A	B	C	D	E	F	G	H
46	1046	VIDYA	V	49.2	43.2	52.8	46.8	56.4
47	1047	MANASH	II	52.8	16.8	60	52.8	57.6
48	1048	SHAMBHA	IV	52.8	43.2	57.6	42	58.8
49	1049	KUMARI	VII	45.6	44.4	43.2	46.8	43.2
50	1050	BHAVYA	V	54	0	43.2	43.2	56.4
51	1051	NIKHIL	I	54	43.2	57.6	56.4	57.6
52	1052	SRIJITA	III	55.2	40.8	57.6	52.8	0
53	1053	SRIJANI	III	38.4	0	0	48	56.4
54	1054	SAMAR	V	56	55	52	50	57
55	1055	ADITYA	VIII	44.4	40.8	42	43.2	44.4
56	1056	SEEMA	VII	48	38.4	48	48	54
57	1057	SAMIR	V	51.6	43.2	52.8	48	56.4
58	1058	ASHUTOSH	I	56.4	43.2	50.4	50.4	58.8
59	1059	NIDHI	III	42	36	44.4	42	46.8
60	1060	NIKHIL	VII	55.2	45.6	60	57.6	57.6
61	1061	SIDDHART	VII	0	0	0	0	0
62								

10. Display Graphs:

Using this module, user can view graphical representation of the data stored in the database. This module has 8 different graphs to choose from.

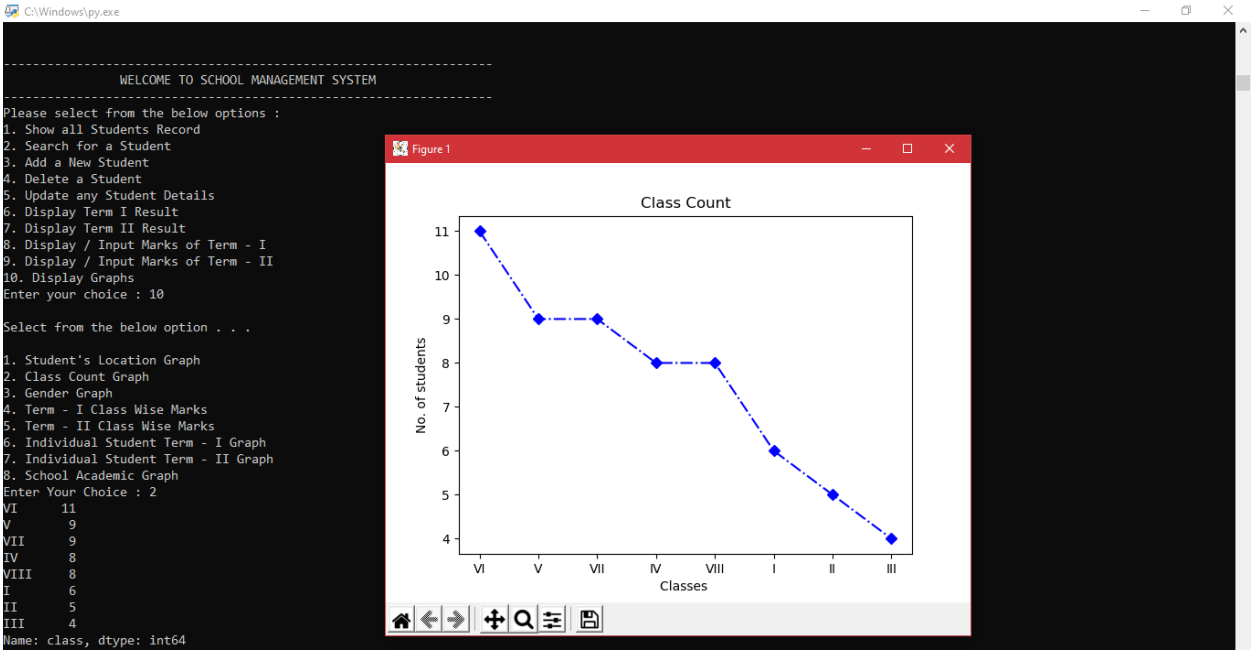
i. <u>Student Location Graph:</u>

This Graph shows the number of students coming from various places. User needs to press 10 and then press 1 to view this graph.



ii. Class Count Graph:

This Graph shows the number of students class wise. User needs to press 10 and then press 2 to view this graph.

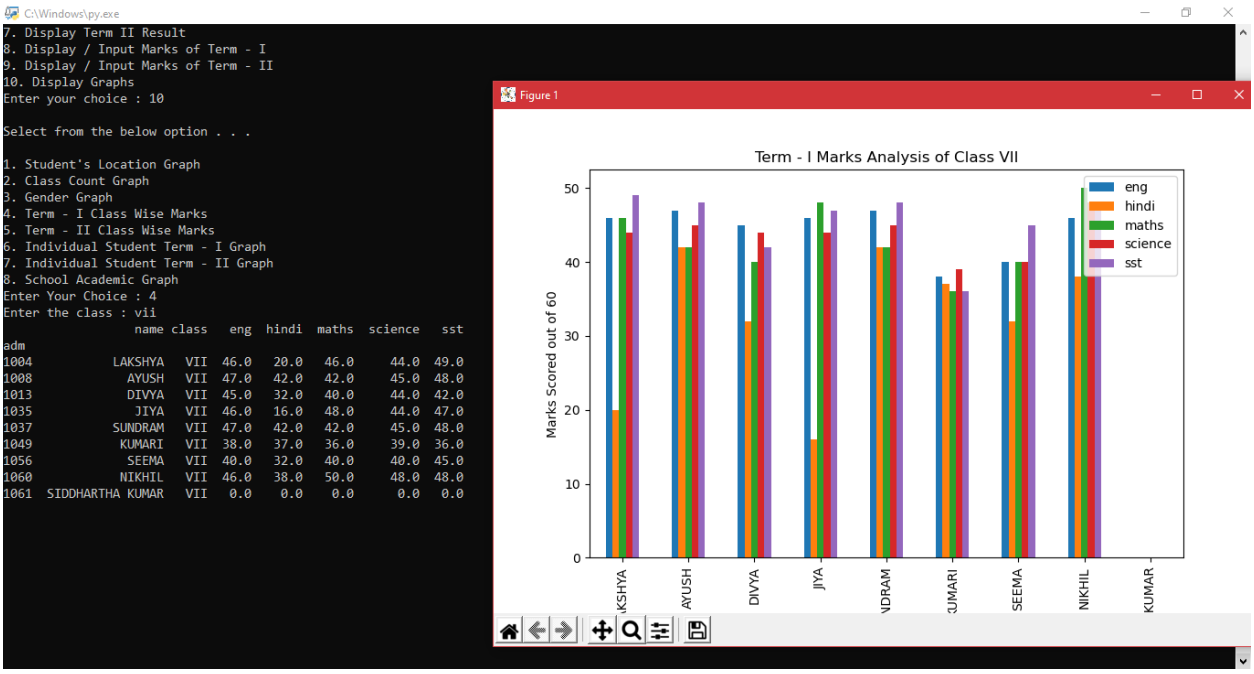


iii. Gender Graph:

This Graph shows the male and female count of whole school. User needs to press 10 and then press 3 to view this graph.

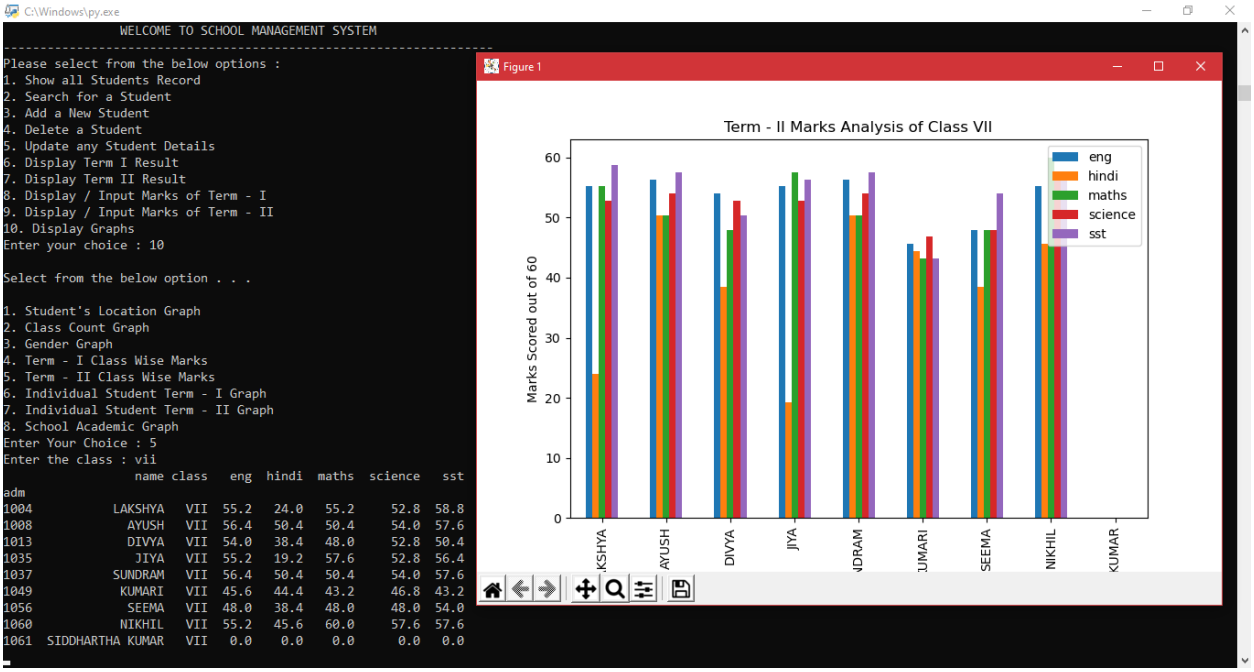
iv. Term I Class Wise Marks:

This Graph shows the class wise marks scored by the students in Term I. User needs to press 10 and then press 4 to view this graph. Then the user needs to enter the class for which he/she wants to see the graph. The data is obtained from the marks stored in markstermi.csv file.



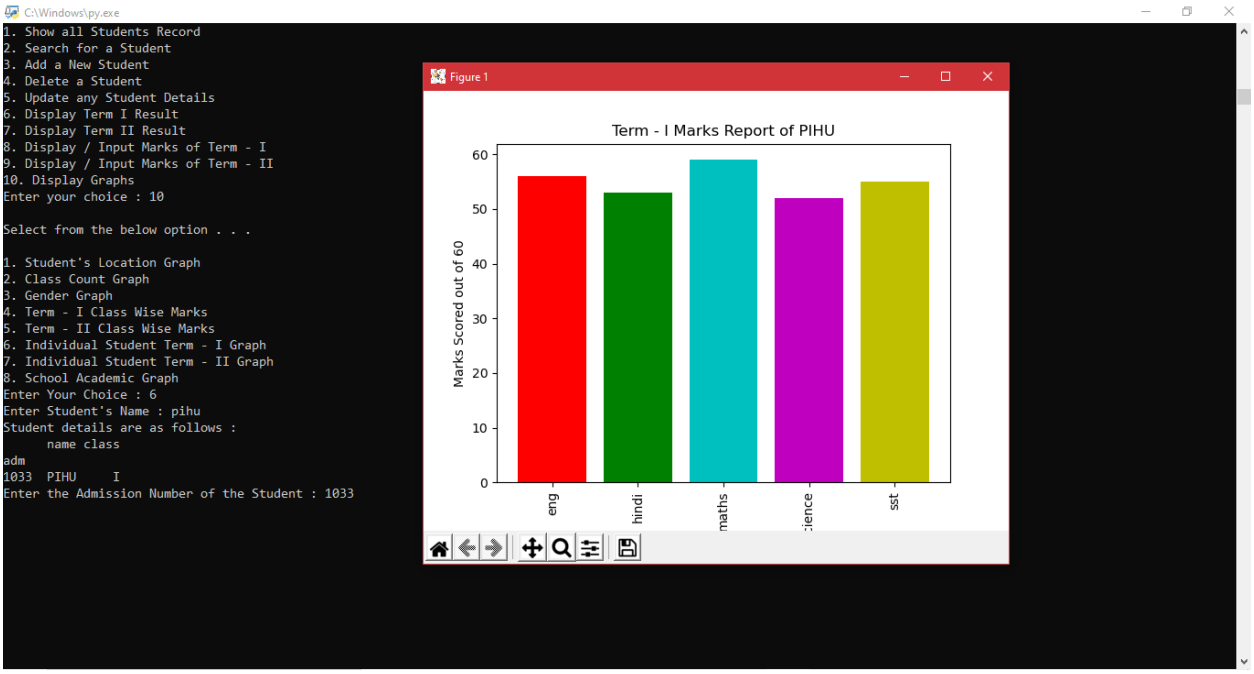
v. Term II Class Wise Marks:

This Graph shows the class wise marks scored by the students in Term II. User needs to press 10 and then press 5 to view this graph. Then the user needs to enter the class for which he/she wants to see the graph. The data is obtained from the marks stored in markstermii.csv file



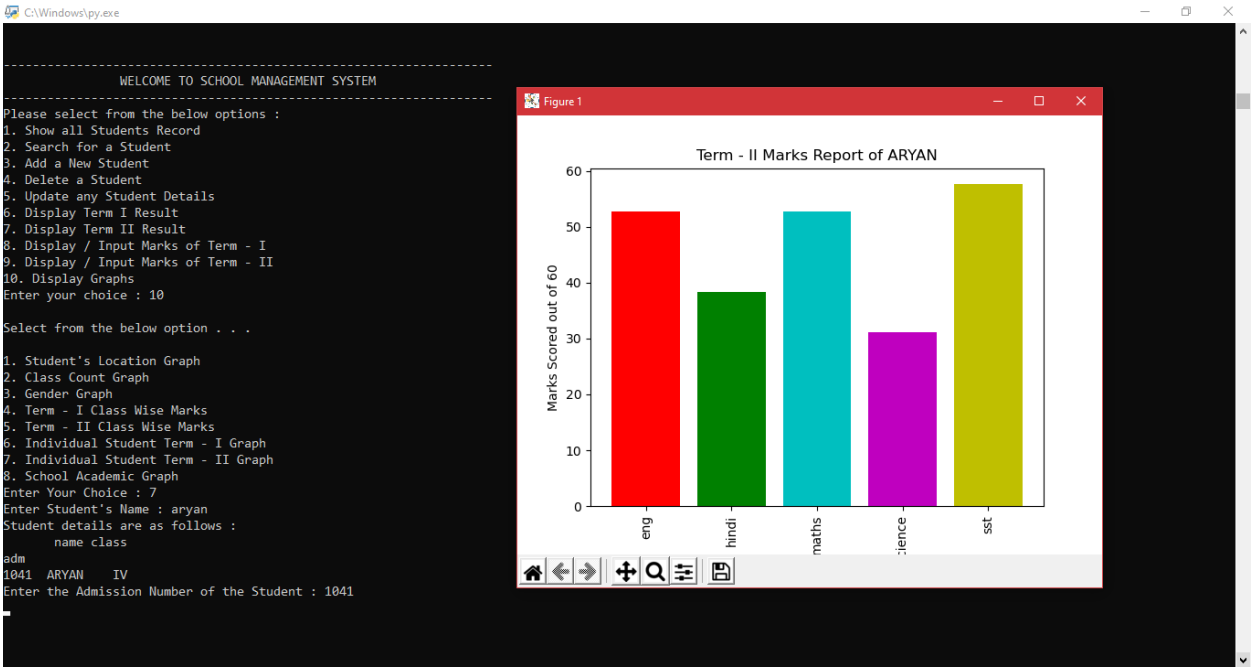
vi. Individual Student Term-I Graph:

This Graph shows the marks scored by a particular student in Term-I. User needs to press 10 and then press 6 to view this graph. User needs to type the student name. The system will display the list of matching names from markstermi.csv file and then the user needs to enter the admission number for which he/she wants to display the graph.



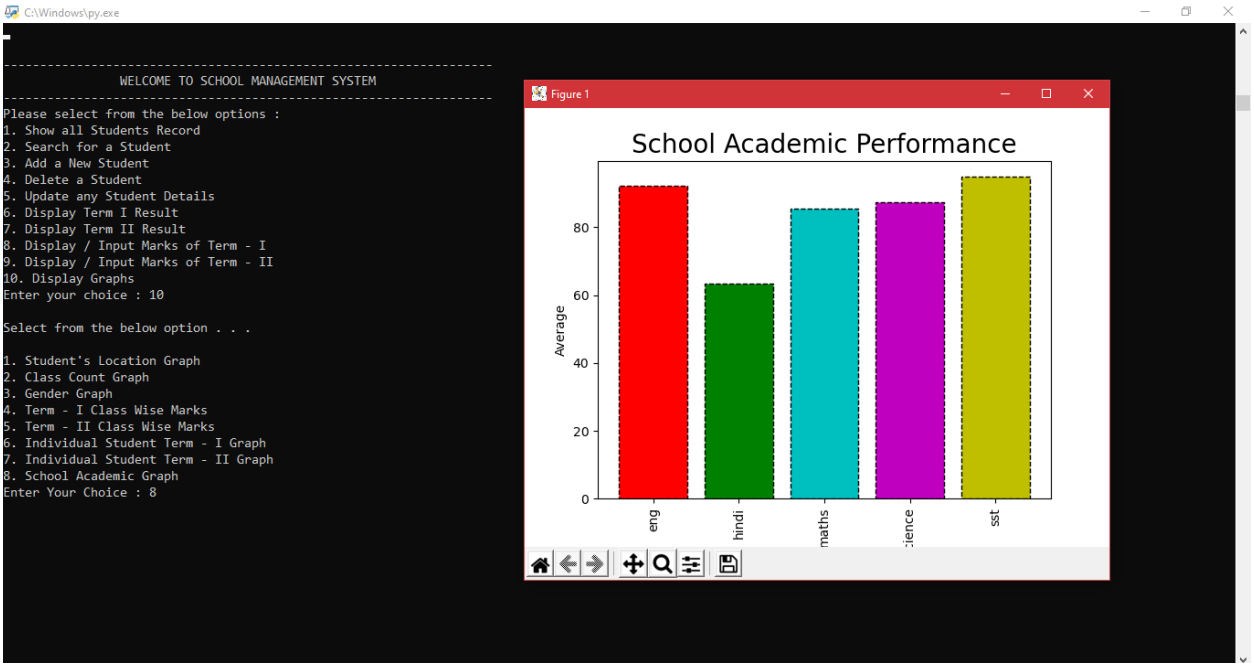
vii. Individual Student Term-II Graph:

This Graph shows the marks scored by a particular student in Term-II. User needs to press 10 and then press 7 to view this graph. User needs to type the student name. The system will display the list of matching names from markstermi.csv file and then the user needs to enter the admission number for which he/she wants to display the graph.



viii. School Academic Graph:

This Graph shows the Subject Wise Analysis of the whole school. User needs to press 10 and then press 8 to view this graph. The data is obtained by a combination of markstermi.csv and markstermii.csv files.



8. USER MANUAL (HOW TO INSTALL)

How to install Software:

Hardware Requirement·

- Intel Pentium/Celeron or similar processor based PC
- 128 MB RAM and 4GB HOD space (for Database) is desirable.
- Standard I/O devices like Keyboard and Mouse etc.

Software Requirement-

- Windows 7/8/10 OS is desirable.
- Python version 3 is required.
- Pandas and Matplotlib Libraries of Python must be installed using **pip install pandas** and **pip install matplotlib** commands.
- Microsoft Excel must be installed so as to read csv files.

Database Installation

The software project is distributed with a sample csv files named students.csv, markstermi.csv and markstermii.csv.

User needs to open the **sms.py** file using Python **IDLE** and change:

- **path** variable to the location where students.csv file is saved
- **term1path** variable to the location where markstermi.csv file is saved
- **term2path** variable to the location where markstermi.csv file is saved

9. FUTURE SCOPE OF THE PROJECT

We can convert the above software into a desktop application using Tkinter or a web-application using Django for a better user interface.

In this software, we can have further modules updated like:

- Fees Payment
- Transport Facility
- Enhanced Exam Report Analysis
- A login module to enhance the security feature
- Student and Administration Menu

10. LIMITATIONS OF THE PROJECT

The major drawback of the project is its command-line interface. We can convert it into a desktop application using Tkinter or a web-application using Django.

The project has following limitations:

1. Field validation is not present in most of the field, so sometimes the software crashes.
2. CSV file is used as a database software which is more complex to handle than MySQL database.