

PROJECT REPORT ON

STUDENT MANAGEMENT SYSTEM

INDEX

S.NO	TOPIC	PAGE NO
1.	Introduction	
2.	Code and Resources	
3.	Synopsis	
4.	Code	
5.	Output	

INTRODUCTION

This project “Student Information Management System” provides us a simple interface for maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project. Throughout the project the focus has been on presenting information in an easy and intelligible manner. The project is very useful for those who want to know about Student Information Management Systems and want to develop software based on the same concept. The project provides facilities like Add, Remove, Update, View of students thus reducing paperwork and automating the record generation process in an educational institution.

SYNOPSIS

Student Information Management System can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

Name of the Project: Student Management System

Objectives:

- ☐ registration of students
- ☐ Maintenance of student records
- ☐ Searching student records

Platform

Operating Systems: Microsoft Windows

Technologies Used:

- ☐ Front End: PYTHON CONSOL
- ☐ Web designing language: CORE PYTHON

(Back end): CSV

Software Requirements:

- ☐ JUPYTER NOTEBOOK

Hardware Requirements:

- ☐ Intel Pentium IV processor or equivalent or higher
- ☐ 512 MB Ram or Higher
- ☐ 20 GB HDD or Higher
- ☐ Network Connectivity

CODE

```
# Student Management System
# student.py

import csv

# Define global variables
student_fields = ['roll', 'name', 'age', 'email', 'phone']
student_database = 'students.csv'

def display_menu():
    print("-----")
    print(" Welcome to Student Management System")
    print("-----")
    print("1. Add New Student")
    print("2. View Students")
    print("3. Search Student")
    print("4. Update Student")
    print("5. Delete Student")
    print("6. Quit")

def add_student():
    print("-----")
    print("Add Student Information")
    print("-----")
    global student_fields
    global student_database
```

```

student_data = []
for field in student_fields:
    value = input("Enter " + field + ": ")
    student_data.append(value)

with open(student_database, "a") as f:
    writer = csv.writer(f)
    writer.writerows([student_data])

print("Data saved successfully")
input("Press any key to continue")
return

```

```

def view_students():
    global student_fields
    global student_database

    print("--- Student Records ---")

    with open(student_database, "r") as f:
        reader = csv.reader(f)
        for x in student_fields:
            print(x, end="\t |")
        print("\n-----")

    for row in reader:
        for item in row:
            print(item, end="\t |")

```

```
print("\n")
```

```
input("Press any key to continue")
```

```
def search_student():
```

```
    global student_fields
```

```
    global student_database
```

```
    print("--- Search Student ---")
```

```
    roll = input("Enter roll no. to search: ")
```

```
    with open(student_database, "r") as f:
```

```
        reader = csv.reader(f)
```

```
        for row in reader:
```

```
            if len(row) > 0:
```

```
                if roll == row[0]:
```

```
                    print("----- Student Found -----")
```

```
                    print("Roll: ", row[0])
```

```
                    print("Name: ", row[1])
```

```
                    print("Age: ", row[2])
```

```
                    print("Email: ", row[3])
```

```
                    print("Phone: ", row[4])
```

```
                    break
```

```
            else:
```

```
                print("Roll No. not found in our database")
```

```
    input("Press any key to continue")
```

```
def update_student():
```

```
    global student_fields
```


global student_database

print("--- Update Student ---")

roll = input("Enter roll no. to update: ")

index_student = None

updated_data = []

with open(student_database, "r") as f:

 reader = csv.reader(f)

 counter = 0

 for row in reader:

 if len(row) > 0:

 if roll == row[0]:

 index_student = counter

 print("Student Found: at index ",index_student)

 student_data = []

 for field in student_fields:

 value = input("Enter " + field + ": ")

 student_data.append(value)

 updated_data.append(student_data)

 else:

 updated_data.append(row)

 counter += 1

Check if the record is found or not

if index_student is not None:

 with open(student_database, "w") as f:

 writer = csv.writer(f)

 writer.writerows(updated_data)

else:

```
print("Roll No. not found in our database")
```

```
input("Press any key to continue")
```

```
def delete_student():
```

```
    global student_fields
```

```
    global student_database
```

```
    print("--- Delete Student ---")
```

```
    roll = input("Enter roll no. to delete: ")
```

```
    student_found = False
```

```
    updated_data = []
```

```
    with open(student_database, "r") as f:
```

```
        reader = csv.reader(f)
```

```
        counter = 0
```

```
        for row in reader:
```

```
            if len(row) > 0:
```

```
                if roll != row[0]:
```

```
                    updated_data.append(row)
```

```
                    counter += 1
```

```
            else:
```

```
                student_found = True
```

```
if student_found is True:
```

```
    with open(student_database, "w") as f:
```

```
        writer = csv.writer(f)
```

```
        writer.writerows(updated_data)
```

```
        print("Roll no. ", roll, "deleted successfully")
```

```
else:
```

```
print("Roll No. not found in our database")

input("Press any key to continue")

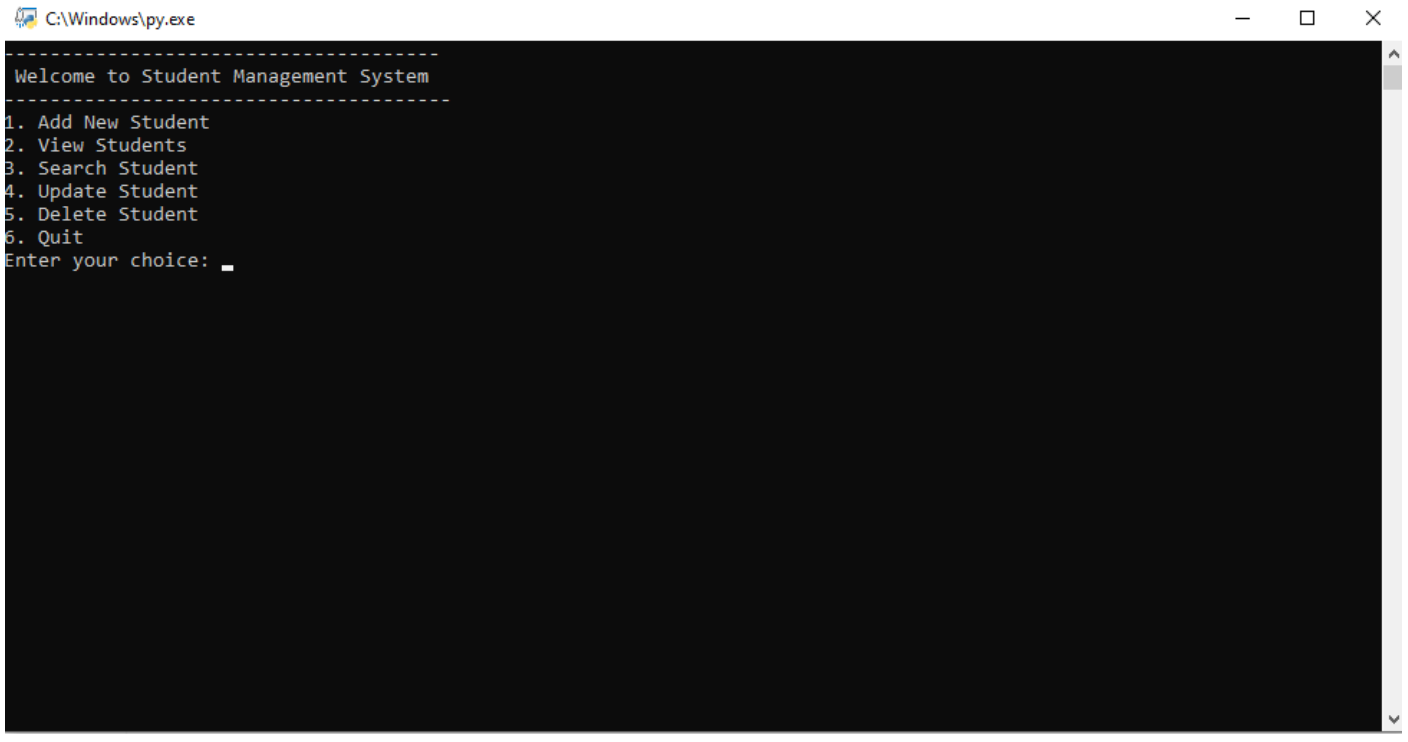
while True:
    display_menu()

    choice = input("Enter your choice: ")
    if choice == '1':
        add_student()
    elif choice == '2':
        view_students()
    elif choice == '3':
        search_student()
    elif choice == '4':
        update_student()
    elif choice == '5':
        delete_student()
    else:
        break

print("-----")
print(" Thank you for using our system ")
print("-----")
```

SCREEN SHOTS & OUTPUT

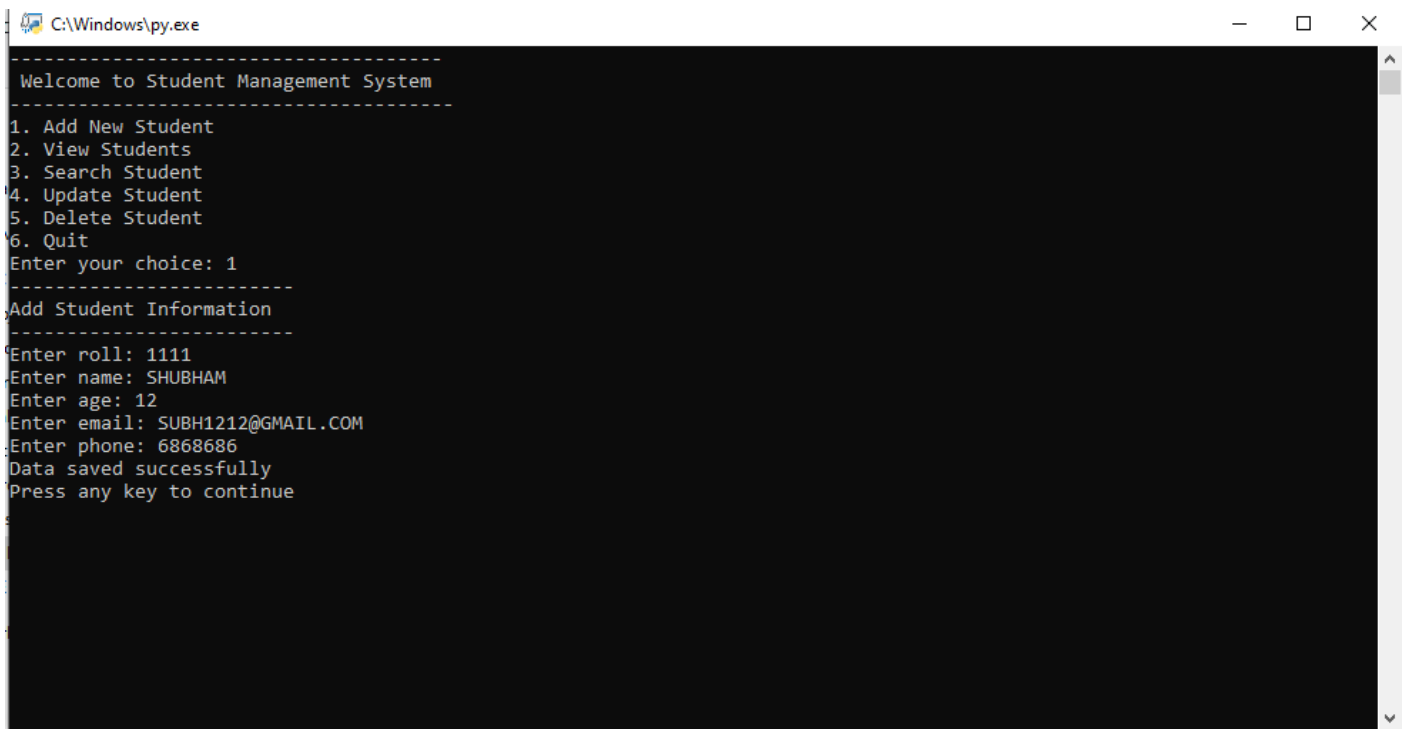
MAIN SCREEN



A screenshot of a Windows command prompt window titled "C:\Windows\py.exe". The window displays the "Welcome to Student Management System" message, followed by a list of six options: 1. Add New Student, 2. View Students, 3. Search Student, 4. Update Student, 5. Delete Student, and 6. Quit. The prompt "Enter your choice: " is shown with a cursor at the end.

```
C:\Windows\py.exe
-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: _
```

STUDENT REGISTER



A screenshot of a Windows command prompt window titled "C:\Windows\py.exe". The window displays the "Welcome to Student Management System" message, followed by the same list of six options as the main screen. The user has entered "1" for "Add New Student". The prompt "Enter your choice: 1" is shown. The user has then entered "1111" for "Enter roll:", "SHUBHAM" for "Enter name:", "12" for "Enter age:", and "SUBH1212@GMAIL.COM" for "Enter email:". The prompt "Enter phone:" is shown with a cursor at the end. The message "Data saved successfully" is displayed, followed by "Press any key to continue".

```
C:\Windows\py.exe
-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: 1
-----
Add Student Information
-----
Enter roll: 1111
Enter name: SHUBHAM
Enter age: 12
Enter email: SUBH1212@GMAIL.COM
Enter phone:
Data saved successfully
Press any key to continue
```

VIEW STUDENTS

```
C:\Windows\py.exe

-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: 2
--- Student Records ---
roll    |name  |age  |email      |phone  |
-----|-----|-----|-----|-----|
1       |uttam |23   |uttam.k7052568@gmail.com |8977979324 |
2       |admin |24   |admin@gmail.com |255445555 |
25      |Aman  |18   |aman@gmail.com  |6654845625 |
1111    |SHUBHAM |12   |SUBH1212@GMAIL.COM |6868686 |
Press any key to continue
```

SEARCH STUDENT

```
C:\Windows\py.exe

-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: 3
--- Search Student ---
Enter roll no. to search: 1111
----- Student Found -----
Roll: 1111
Name: SHUBHAM
Age: 12
Email: SUBH1212@GMAIL.COM
Phone: 6868686
Press any key to continue
```

UPDATE STUDENT

```
C:\Windows\py.exe

-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: 4
--- Update Student ---
Enter roll no. to update: 1111
Student Found: at index 3
Enter roll: 2534
Enter name: SUBHAM
Enter age: 12
Enter email: SUBHAM1212@GMAIL.COM
Enter phone: 57657657
Press any key to continue

-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice:
```

DELETE STUDENT

```
C:\Windows\py.exe

-----
Welcome to Student Management System
-----
1. Add New Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Quit
Enter your choice: 5
--- Delete Student ---
Enter roll no. to delete: 2
Roll no. 2 deleted successfully
Press any key to continue.
```