**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

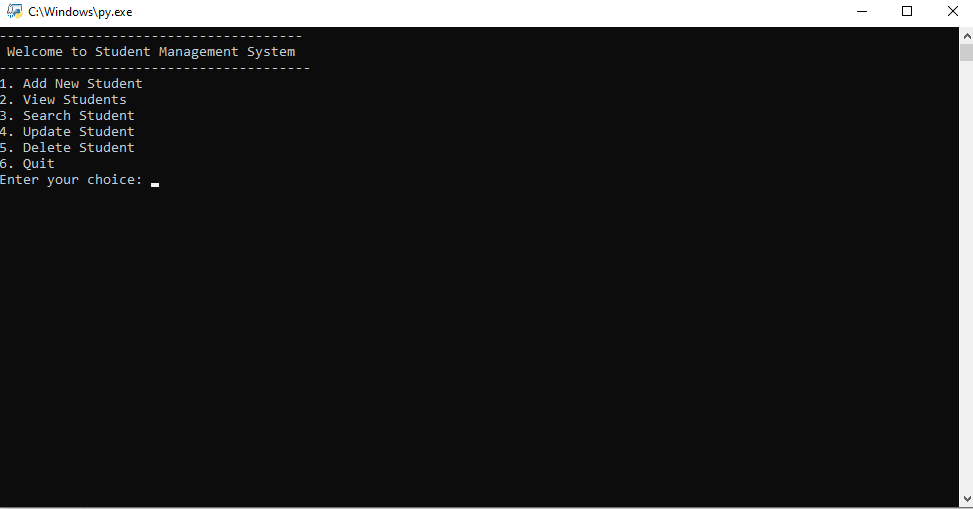
print("-------------------------------")

print(" Thank you for using our system ")

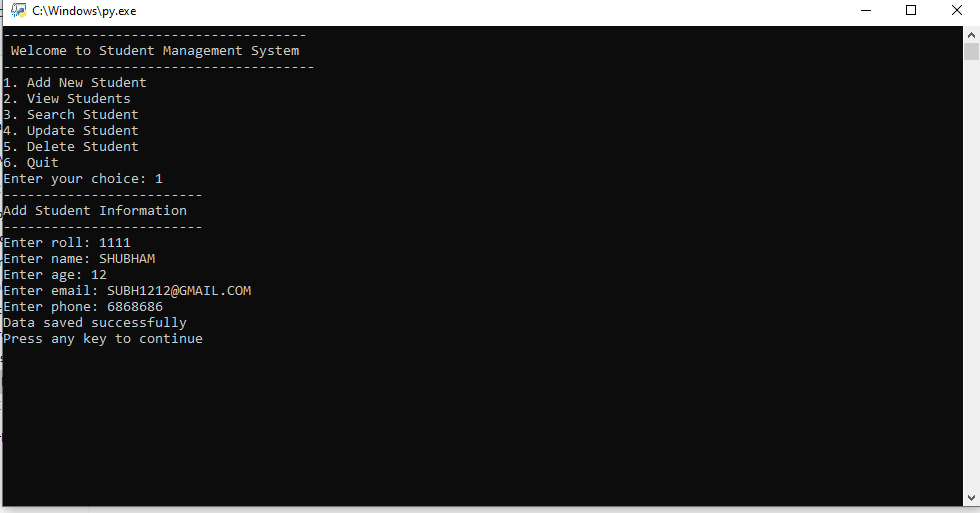
print("-------------------------------")

SCREEN SHOTS & OUTPUT

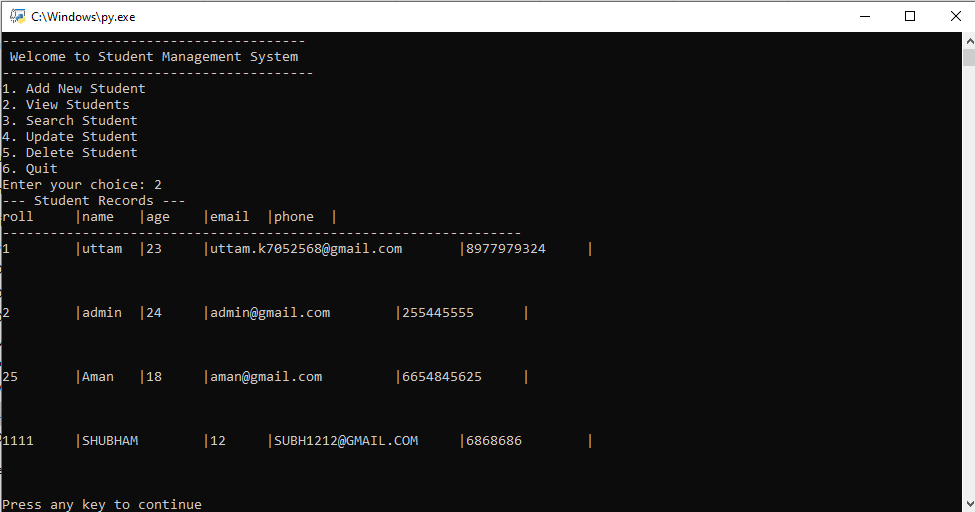
MAIN SCREEN



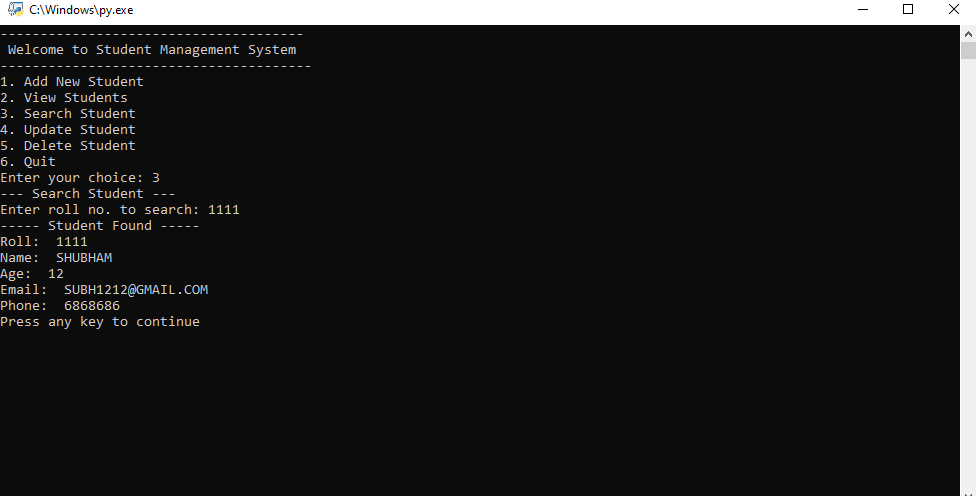
STUDENT REGISTER



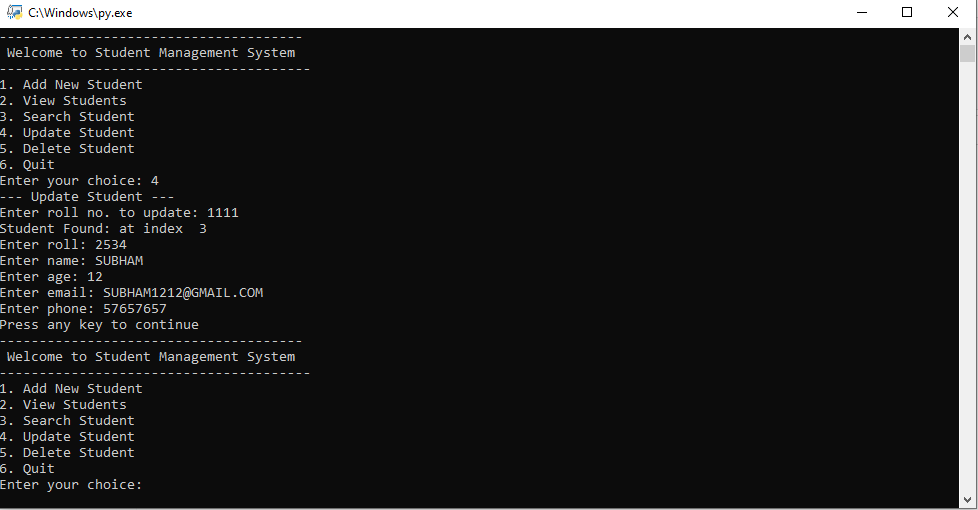
VIEW STUDENTS



SEARCH STUDENT



UPDATE STUDENT



DELETE STUDENT

