

***SUBMITTED BY***

NAJAM

-

UL

-

HASSAN

(

945

-

2018)

M.SADIQ UL AMIN (828-2019)

WALEED AZHER KIDWAI

(

943

-

2018

)

***TEACHER NAME***

MOSHIN RAZA



**STUDENT**

**MANAGEMENT SYSTEM**

**SEMESTER**

**REPORT**

**2020**

**HAMDARD UNIVERSITY**

IN PERSUIT OF EXCELLENCE

# HAMDARD UNIVERSITY

## IN PERSUIT OF EXCELLENCE

**TABLE OF CONTENT**

1. Introduction........................................ 3

1. Objective.......................................... 4

1. Scope............................................ 5

1. Advantages & Disadvantages......................... 6

1. Features of Program.................................7-9

1. Code of Student Management System.................10-14

1. Conclusion......................................... 15

# HAMDARD UNIVERSITY

## IN PERSUIT OF EXCELLENCE

### INTRODUCTION

This is a Python Student Management System project. This system contains basic functions, which include Add students Detail like His Name, college name, contact number & Address and view student’s details.

The project is built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the details of students.

### OBJECTIVE

The main objective of the Student Management System aim to maintain the record of students, so that it helps teachers make instructional decisions and to obtain specific information that may assist in working with a student. It is the faster way to get information about the students. It improves data security and integrity. It provides an organized system of management of information. It improves personal efficiency and increases organizational control. The purpose of the project is to build an application program to reduce the manual work for managing the student’s information.

### SCOPE

The scope of this project is limited to the design and implementation of a Student Information Management System. The purpose of the Student Information Management system is to allow for storing information of large number of students.

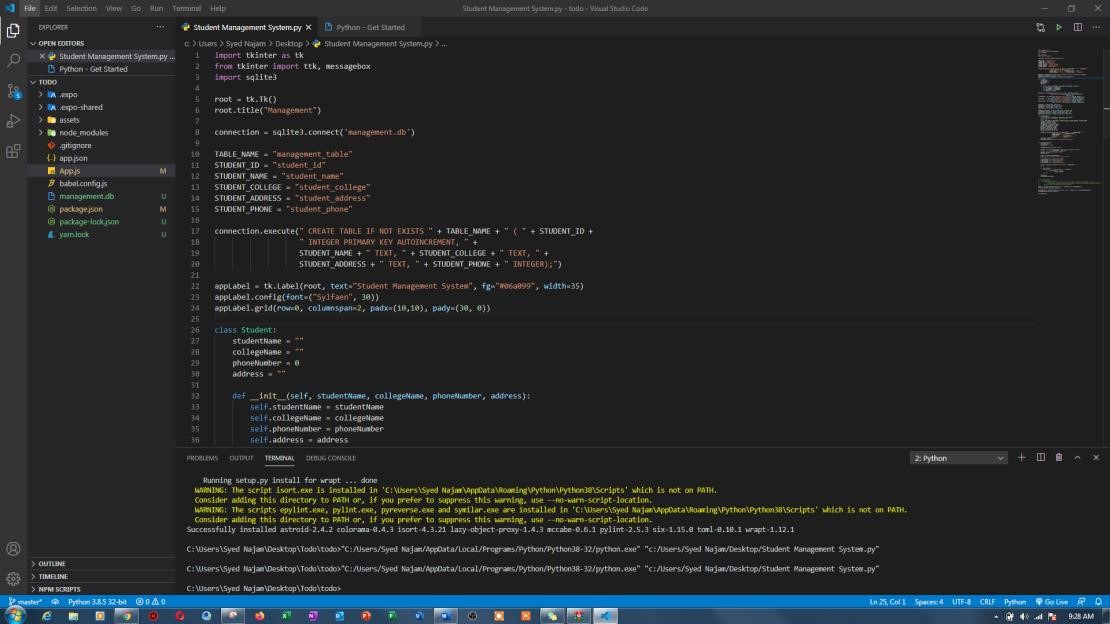
### AGVANTAGES & DISADVANTAGES

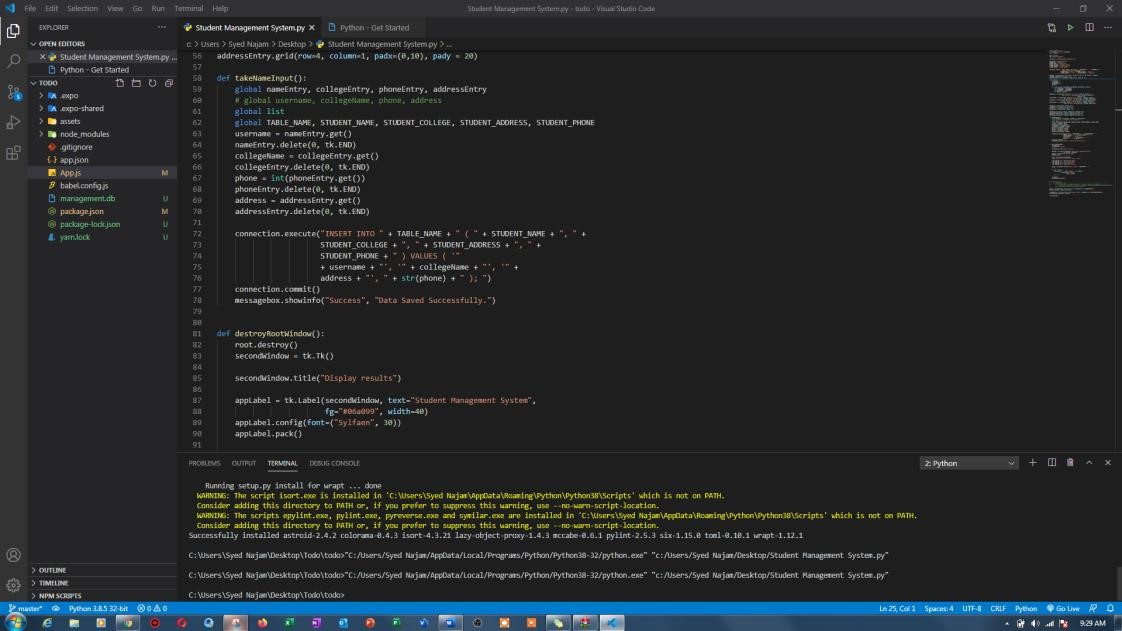
* **ADVANTAGES:** 
  + Improves the General Comfort of Staffs
  + Better Communication
  + Easy Access to All
  + Complete Tracking of the Students
  + Helps Build a Strong Alumni
  + Improves the General Comfort of Staffs

* **DISADVANTAGES:** 
  + There is no search option to find the record of specific student.
  + Only, people who are accustomed to regular use of smartphones or computers can operate this software.
  + The risk of data mishandling might be occur

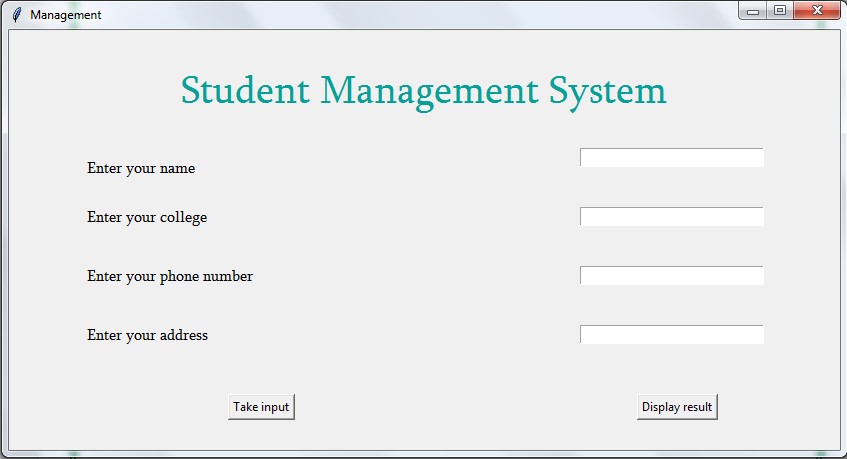
### FEATURES OF PROGRAM

* **SOURCE CODE:**

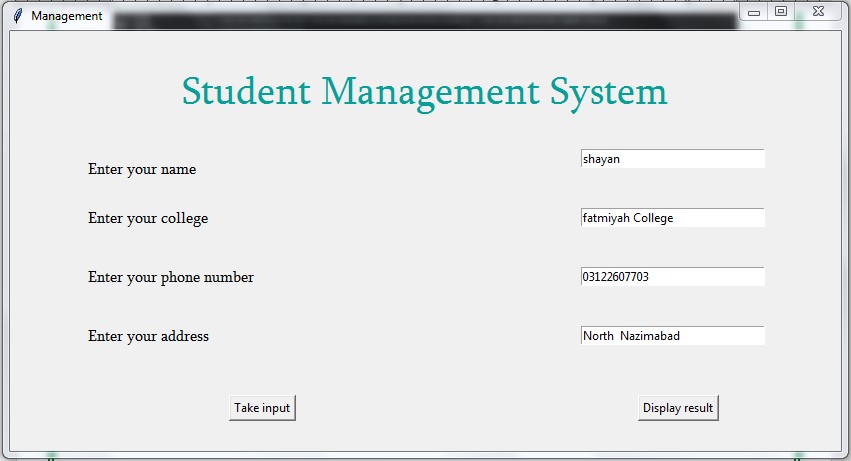




* **INTERFACE:**



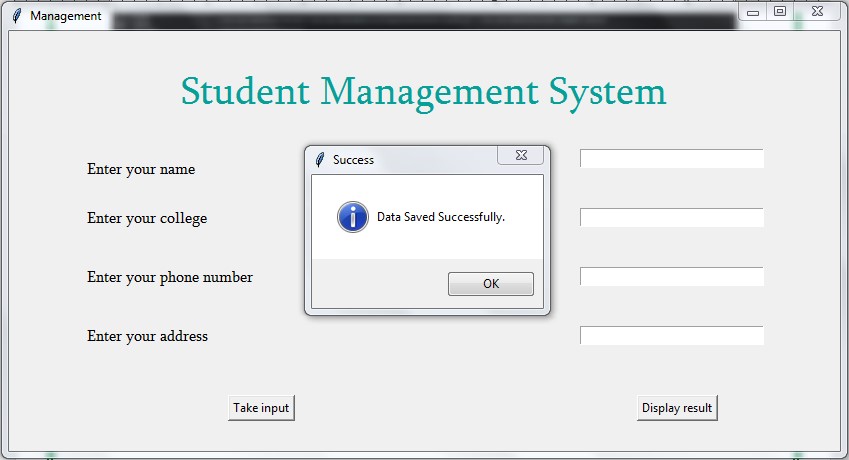
* **INSERT DETAILS:**



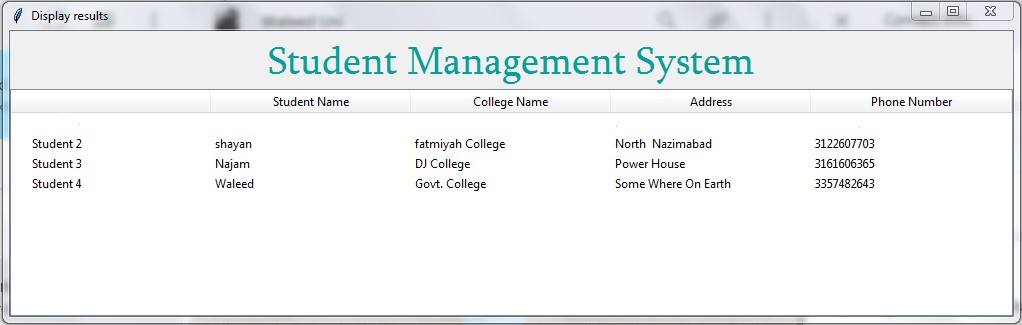


**DATA INSERT SUCESSFULLY**

**:**



* **VIEW RECORD:**



# HAMDARD UNIVERSITY

## IN PERSUIT OF EXCELLENCE

**CODE OF STUDENT MANAGEMENT SYSTEM**

import tkinter as tk from tkinter import ttk, messagebox import sqlite3

root = tk.Tk() root.title("Management")

connection = sqlite3.connect('management.db')

TABLE\_NAME = "management\_table"

STUDENT\_ID = "student\_id"

STUDENT\_NAME = "student\_name"

STUDENT\_COLLEGE = "student\_college"

STUDENT\_ADDRESS = "student\_address"

STUDENT\_PHONE = "student\_phone"

connection.execute(" CREATE TABLE IF NOT EXISTS " + TABLE\_NAME + " ( " + STUDENT\_ID +

" INTEGER PRIMARY KEY AUTOINCREMENT, " +

STUDENT\_NAME + " TEXT, " + STUDENT\_COLLEGE + " TEXT, " +

STUDENT\_ADDRESS + " TEXT, " + STUDENT\_PHONE + " INTEGER);")

appLabel = tk.Label(root, text="Student Management System", fg="#06a099", width=35) appLabel.config(font=("Sylfaen", 30)) appLabel.grid(row=0, columnspan=2, padx=(10,10), pady=(30, 0))

class Student:

studentName = "" collegeName = "" phoneNumber = 0 address = ""

def \_\_init\_\_(self, studentName, collegeName, phoneNumber, address):

self.studentName = studentName self.collegeName = collegeName self.phoneNumber = phoneNumber self.address = address

nameLabel = tk.Label(root, text="Enter your name", width=40, anchor='w', font=("Sylfaen", 12)).grid(row=1, column=0, padx=(10,0),

pady=(30, 0))

collegeLabel = tk.Label(root, text="Enter your college", width=40, anchor='w', font=("Sylfaen", 12)).grid(row=2, column=0, padx=(10,0)) phoneLabel = tk.Label(root, text="Enter your phone number", width=40, anchor='w', font=("Sylfaen", 12)).grid(row=3, column=0, padx=(10,0)) addressLabel = tk.Label(root, text="Enter your address", width=40, anchor='w', font=("Sylfaen", 12)).grid(row=4, column=0, padx=(10,0))

nameEntry = tk.Entry(root, width = 30) collegeEntry = tk.Entry(root, width = 30) phoneEntry = tk.Entry(root, width = 30) addressEntry = tk.Entry(root, width = 30)

nameEntry.grid(row=1, column=1, padx=(0,10), pady=(30, 20)) collegeEntry.grid(row=2, column=1, padx=(0,10), pady = 20) phoneEntry.grid(row=3, column=1, padx=(0,10), pady = 20) addressEntry.grid(row=4, column=1, padx=(0,10), pady = 20)

def takeNameInput():

global nameEntry, collegeEntry, phoneEntry, addressEntry

# global username, collegeName, phone, address

global list

global TABLE\_NAME, STUDENT\_NAME, STUDENT\_COLLEGE, STUDENT\_ADDRESS, STUDENT\_PHONE

username = nameEntry.get() nameEntry.delete(0, tk.END) collegeName = collegeEntry.get() collegeEntry.delete(0, tk.END) phone = int(phoneEntry.get()) phoneEntry.delete(0, tk.END) address = addressEntry.get() addressEntry.delete(0, tk.END)

connection.execute("INSERT INTO " + TABLE\_NAME + " ( " + STUDENT\_NAME + ", " +

STUDENT\_COLLEGE + ", " + STUDENT\_ADDRESS + ", " +

STUDENT\_PHONE + " ) VALUES ( '"

+ username + "', '" + collegeName + "', '" + address + "', " + str(phone) + " ); ") connection.commit() messagebox.showinfo("Success", "Data Saved Successfully.")

def destroyRootWindow():

root.destroy() secondWindow = tk.Tk()

secondWindow.title("Display results")

appLabel = tk.Label(secondWindow, text="Student Management System",

fg="#06a099", width=40) appLabel.config(font=("Sylfaen", 30)) appLabel.pack()

tree = ttk.Treeview(secondWindow) tree["columns"] = ("one", "two", "three", "four")

tree.heading("one", text="Student Name") tree.heading("two", text="College Name") tree.heading("three", text="Address") tree.heading("four", text="Phone Number")

cursor = connection.execute("SELECT \* FROM " + TABLE\_NAME + " ;")

i = 0

for row in cursor:

tree.insert('', i, text="Student " + str(row[0]), values=(row[1], row[2], row[3], row[4])) i = i + 1

tree.pack() secondWindow.mainloop()

# def printDetails():

# for singleItem in list:

# print("Student name is: %s\nCollege name is: %s\nPhone number is: %d\nAddress is: %s" %

# (singleItem.studentName, singleItem.collegeName, singleItem.phoneNumber, singleItem.address)) # print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

button = tk.Button(root, text="Take input", command=lambda :takeNameInput()) button.grid(row=5, column=0, pady=30)

displayButton = tk.Button(root, text="Display result", command=lambda :destroyRootWindow()) displayButton.grid(row=5, column=1)

root.mainloop()

# HAMDARD UNIVERSITY

## IN PERSUIT OF EXCELLENCE

### CONCLUSION

At the beginning of this project, we want to make a Student Management System and at the end of this project, we made this. After preparing the papers, we got the answers of the following questions:

* What is Student Management System?
* Is it possible to run the whole management system into Computer?
* What are the main responsibilities of a Student Management System?
* How will you handle large groups of Students Record?