**DBMS ASSIGNMENT**

**Question:**

Create an application for student contacts management application. The application enables a faculty user to login and search the details of a **Student** (Name, Address, Contact no, Email id, parent contact) with his register number. Create necessary tables and Forms.

Akash Ambashankar

CSE 4A

311018104005

**Data Schema:**

1. Accounts Table

|  |  |  |
| --- | --- | --- |
| **Name** | **Username** | **Password** |
| TEXT | TEXT PRIMARY KEY | TEXT |

1. Student Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Register\_Number** | **Address** | **Contact\_Number** | **Email\_Id** | **PCN** |
| TEXT | NUMBER PRIMARY KEY | TEXT | NUMBER | TEXT | NUM |

**Form Design:**

* + 1. Main

import PySimpleGUI as s

from Log\_In import login

from Sign\_Up import signup

col = "#343434"

col2 = "white"

layout = [

[s.Text("Student Contact Manager", background\_color=col, justification="left", text\_color=col2)],

[s.Text("", background\_color=col, text\_color=col2)],

[s.Button("Log In", button\_color=('white', 'black')), s.Button("Sign Up", button\_color=('white', 'black'))]

]

w = s.Window("Student Contact Manager", layout, background\_color=col, use\_default\_focus=False, element\_justification="center")

while True:

button, values = w.Read()

if button == "Log In":

w.Close()

login()

break

if button == "Sign Up":

w.Close()

signup()

break

if button is None:

w.Close()

break

* + 1. Log In

import PySimpleGUI as s

from Sign\_Up import signup

from back import \*

from Search import search\_page

def login():

col = "#343434"

col2 = "white"

login\_layout = [

[s.Text("Student Contact Manager", font='courier' '50', background\_color=col, justification="left", text\_color=col2)],

[s.Text("Username", background\_color=col, text\_color=col2), s.InputText("", key='username',size=(20,10))],

[s.Text("Password", background\_color=col, text\_color=col2), s.InputText("", key="password", password\_char='\*', size=(20,10))],

[s.Button("Login", bind\_return\_key=True, button\_color=('white', 'black'))],

[s.Text(" ", key='msg', background\_color=col)],

[s.Text("Don't have an account?",background\_color=col, text\_color=col2),s.Button("Sign Up",button\_color=('green',col),border\_width=0)]

]

w = s.Window("Login", login\_layout, resizable=1, background\_color=col, element\_justification="center")

while True:

button, values = w.Read()

if button == "Login":

if authenticate(values['username'], values['password']):

w.FindElement('msg').update("Logging in...", text\_color='green')

w.Close()

search\_page()

else:

if values["password"] == "":

w.FindElement('msg').update("Password required", text\_color='red')

else:

w.FindElement('msg').update("Username or password is incorrect", text\_color='red')

if button == "Sign Up":

w.Close()

signup()

if button is None:

w.Close()

break

* + 1. Sign Up

import PySimpleGUI as s

from back import \*

import Log\_In

col = "#343434"

col2 = "white"

def signup():

sign\_up\_layout = [[s.Text("Student Contact Manager", font='courier' '50', background\_color=col, justification="left", text\_color=col2)],

[s.Text("Enter Name ", background\_color=col,

text\_color=col2), s.InputText("", key='name', size=(20,10))],

[s.Text("Enter Username ", background\_color=col, text\_color=col2), s.InputText("", key='username', size=(20,10))],

[s.Text("Enter Password ", background\_color=col, text\_color=col2), s.InputText("", key="password", password\_char='\*', size=(20,10))],

[s.Text("Confirm Password", background\_color=col, text\_color=col2), s.InputText("", key="cpassword", password\_char='\*',size=(20,10))],

[s.Text(" ", key='msg', background\_color=col)],

[s.Button("Create Account", bind\_return\_key=True, button\_color=('white', 'black'))],

[s.Text("Already have an account?", background\_color=col, text\_color=col2),

s.Button("Log In", button\_color=('green', col), border\_width=0)]]

sign\_w = s.Window("Sign Up", sign\_up\_layout, resizable=1, background\_color=col)

while True:

s\_button, s\_values = sign\_w.Read()

if s\_button == "Create Account":

if s\_values['password'] == s\_values['cpassword']:

if len(s\_values['password']) > 7:

ifcreate\_acc(s\_values['name'],s\_values['username'], encrypt(s\_values['password'])):

sign\_w.FindElement('msg').update("Account Created", text\_color='green')

else:sign\_w.FindElement('msg').update("Username Taken", text\_color='red')

else:sign\_w.FindElement('msg').update("Password too short", text\_color='red')

else:sign\_w.FindElement('msg').update("Passwords do not match", text\_color='red')

if s\_button == "Log In":

sign\_w.Close()

Log\_In.login()

break

if s\_button is None:

sign\_w.Close()

break

* + 1. Search

student = []

col = "#343434"

col2 = "white"

layout = [

[s.Text("Student Contact Manager", font='courier' '50', background\_color=col, justification="left", text\_color=col2), s.Button("Log Out", button\_color=("green", col), border\_width=0)],

[s.Text("Enter Register Number", background\_color=col, text\_color=col2)],

[s.Text(" ", key="msg", background\_color=col, text\_color=col2)],

[s.InputText("", key="reg"), s.Button("Search", button\_color=("white", "black"), bind\_return\_key=True)],

[s.Listbox(student, size=(50, 6), key='box', background\_color="white")],

[]

]

w = s.Window("Student Contact Manager", layout, background\_color=col, use\_default\_focus=False)

while True:

button, values = w.Read()

if button == "Search":

try:

details = search(values["reg"])

w.FindElement("reg").update("")

w.FindElement("box").update(details)

w.FindElement("msg").update("")

except:

w.FindElement("msg").update("Invalid Number", text\_color="red")

if button == "Log Out":

w.Close()

Log\_In.login()

break

if button is None:

w.Close()

break

* + 1. Other Functions

from passlib.hash import sha256\_crypt

def encrypt(text):

return sha256\_crypt.encrypt(text)

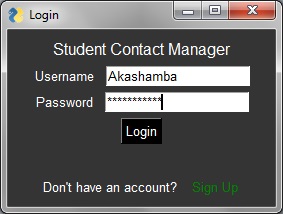
def verify(encrypted, entry):

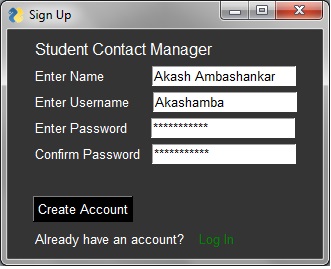
return sha256\_crypt.verify(entry, encrypted)

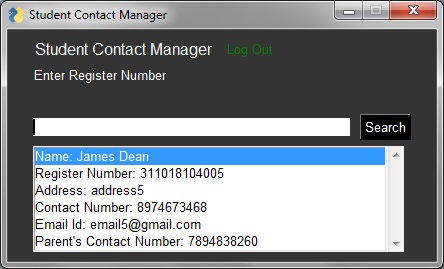
def clean(list):

return list[0][0]

Screenshots:







**Source Code:**

import sqlite3

db = sqlite3.connect('contacts\_db')

c = db.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS Accounts(Name TEXT, Username TEXT PRIMARY KEY,Password TEXT)''')

c.execute('''CREATE TABLE IF NOT EXISTS Student(Name TEXT, Register\_Number INTEGER PRIMARY KEY, Address TEXT, Contact\_Number INTEGER,

Email\_ID TEXT, Parent\_Contact\_Number INTEGER)''')

db.commit()

def create\_acc(name, username, password):

try:

c.execute('''INSERT into Accounts(Name, Username, Password) VALUES(?,?,?) ''', (name, username, password))

db.commit()

return True

except:

return False

def delete\_acc(username):

c.execute('''delete from Accounts where Username=?''', (username,))

db.commit()

def authenticate(username, password):

try:

c.execute('''select Username from Accounts where Username=?''', (username,))

x = clean(c.fetchall())

c.execute('''select Password from Accounts where Username=?''', (username,))

y = clean(c.fetchall())

db.commit()

if username == x and verify(y, password):

return True

except:

return False

def search(reg):

details = []

columns = ["Name:", "Register Number:", "Address:", "Contact Number:", "Email Id:", "Parent's Contact Number:"]

c.execute('''select \* from Student where Register\_Number=?''', (reg,))

x = c.fetchall()

for i in range(len(x[0])):

details.append(str(columns[i]) +" "+ str(x[0][i]))

return details