ACCOUNT MANAGEMENT SYSTEM

CS 1010 - MINOR PROJECT REPORT

*Submitted by*

**NIKITA MISHRA (RA1611003010377)**

***for the course***

**CS 1010 – Computer Skills (Python)**

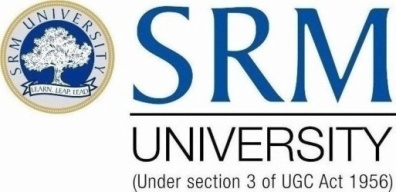
***in partial fulfillment for the award of the degree***

***of***

**Bachelor of Technology**

*in*

# COMPUTER SCIENCE AND ENGINEERING



SRM UNIVERSITY

KATTANKULATHUR

OCTOBER 2018

SRM UNIVERSITY

KATTANKULATHUR

**BONAFIDE CERTIFICATE**

Certified that this project report **“ACCOUNT MANAGEMENT SYSTEM”** is the bonafide work of “**NIKITA MISHRA (RA1611003010377)”** who carried out the project work as part of their course **CS1010 – Computer Skills (Python)**.

**SIGNATURE SIGNATURE**

**Course Instructor** **Dr. B. Amutha**

**HEAD OF THE DEPARTMENT Computer Science & Engg**

**INTERNAL EXAMINER**

**Department of Computer Science and Engineering**



**SRM Institute of Science & Technology**

**Own Work\* Declaration Form**

*This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assessments you submit – work will not be marked unless this is done*

*To be completed by the student for all assessments*

**Degree / Course :** B.Tech / 15IT322E - Python Programming

CS 1010 Computer Skills (Python)

**Student Name : Nikita Mishra**

**Registration Number : RA1611003010377**

**Title of Work : Account Management System**

I / We hereby certify that this assessment complies with the University’s Rules and Regulations relating to Academic misconduct and plagiarism\*\*, as listed in the University Website, Regulations, and the Education Committee guidelines.

*I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that*

*I / We have met the following conditions:*

Clearly referenced / listed all sources as appropriate

Referenced and put in inverted commas all quoted text (from books, web, etc)

Given the sources of all pictures, data etc. that are not my own

Not made any use of the report(s) or essay(s) of any other student(s) either past or present

Not sought or used the help of any external professional academic agencies for the work

Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)

Complied with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies

and regulations.

**DECLARATION:**

I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work**,** except where indicated by referencing, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

**Please Note:** If you are still unsure about what plagiarism is or need advice on how to avoid it, you can

1. Consult your programme handbook

2. Refer the University website or

3. Contact any one of the following for assistance:

Your Faculty, Course Co-ordinator, Project Supervisor, Faculty Advisor, Head of the Department.

\* The University's degrees and other academic awards are given in recognition of a student's **personal achievement**. All work submitted by students for assessment is accepted on the understanding that it is the student's own effort.

\*\* Plagiarism is defined as the submission or presentation of work, in any form, which is not one's own, without **acknowledgement of the sources (even if it’s the author’s previous work)**.The incorporation of material without formal and proper acknowledgement (even with no deliberate intent to cheat) can constitute plagiarism.

**ABSTRACT:**

* In this project we made a python module which deals with storing given data of various variables into a set and gets stored and can be modified at any given time.
* The program takes in the name of the client, the username or email Id , a separate password which will be given for easy segregation ,A base category of the client ,and the date of the transaction of the following given data.

**INTRODUCTION:**

Dealing with data these days has become the need of the hour. With heavy influx of the data from the side of the user as well as the client, efficient storing of data is quite essential. Therefore, we decided to make an account management system, that stores various data fields and gives us the capability to manipulate the data according to our wish. We can perform functions like search on the data to navigate easily. The GUI makes the data easy to use.

**RELATED WORK:**

Before beginning with this project we tried to look for some related works on the internet so that we could come up with something new and interesting. Following were some works we looked upon:

<https://code-projects.org/account-management-system-in-python-with-source-code/>

<https://github.com/Martialhimanshu/Account-management-system>

**INDIVIDUAL CONTRIBUTION:**

The following was my contribution for making this project:

1. Pre-project research and analysis

2. Working with SQL and understanding its implementation of SQL in python.

3. Code Debugging

**ALGORITHM**

The code basically uses implementation of SQL to the fullest. SQL provides us various DDL (data definition language), DML (data manipulation language). Search options are in the form of queries or subqueries.

**PROGRAM**

SQL CODE:

import sqlite3

def create():

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("CREATE TABLE IF NOT EXISTS account(id INTEGER PRIMARY KEY,name TEXT,user TEXT, password TEXT,category TEXT,cdate TEXT)")

con.commit()

con.close()

def viewall():

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("SELECT \* FROM account")

rows = cur.fetchall()

con.close()

return rows

def search(name="",user="",password="",category=""):

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("SELECT \* FROM account WHERE name=? OR user=? OR password=? OR category=?",(name,user,password,category))

rows = cur.fetchall()

con.close()

return rows

def add(name,user,password,category,cdate):

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("INSERT INTO account VALUES(NULL,?,?,?,?,?)",(name,user,password,category,cdate))

con.commit()

con.close()

def update(id,name,user,password,category,cdate):

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("UPDATE account SET name=?,user=?,password=?,category=?,cdate=? WHERE id=?",(name,user,password,category,cdate,id))

con.commit()

con.close()

def delete(id):

con = sqlite3.connect("aledger.db")

cur = con.cursor()

cur.execute("DELETE FROM account WHERE id=?",(id,))

con.commit()

con.close()

create()

#print(search(category="social"))

PYTHON CODE

from tkinter import \*

import ledger\_bk

window = Tk()

window.title("Account Ledger")

def view\_command():

lb.delete(0,END)

for row in ledger\_bk.viewall():

lb.insert(END,row)

def search\_command():

lb.delete(0,END)

for row in ledger\_bk.search(name=name.get(),user=user.get(),password=password.get(),category=category.get()):

lb.insert(END,row)

def add\_command():

ledger\_bk.add(name.get(),user.get(),password.get(),category.get(),cdate.get())

lb.delete(0,END)

lb.insert(END,name.get(),user.get(),password.get(),category.get(),cdate.get())

def get\_selected\_row(event):

try:

global selected\_tuple

index=lb.curselection()[0]

selected\_tuple = lb.get(index)

e1.delete(0,END)

e1.insert(END,selected\_tuple[1])

e2.delete(0,END)

e2.insert(END,selected\_tuple[2])

e3.delete(0,END)

e3.insert(END,selected\_tuple[3])

e4.delete(0,END)

e4.insert(END,selected\_tuple[4])

e5.delete(0,END)

e5.insert(END,selected\_tuple[5])

except IndexError:

pass

def update\_command():

ledger\_bk.update(selected\_tuple[0],name.get(),user.get(),password.get(),category.get(),cdate.get())

view\_command()

def delete\_command():

ledger\_bk.delete(selected\_tuple[0])

view\_command()

#lb.delete(END,get\_selected\_row.selected\_tuple)

def clear\_command():

lb.delete(0,END)

e1.delete(0,END)

e2.delete(0,END)

e3.delete(0,END)

e4.delete(0,END)

e5.delete(0,END)

l1 = Label(window,text="Name")

l1.grid(row=0,column=0,columnspan=2)

l2 = Label(window,text="Username/Email")

l2.grid(row=1,column=0,columnspan=2)

l3 = Label(window,text="Password")

l3.grid(row=2,column=0,columnspan=2)

l4 = Label(window,text="Category")

l4.grid(row=3,column=0,columnspan=2)

l5 = Label(window,text="Date")

l5.grid(row=4,column=0,columnspan=2)

name=StringVar()

e1 = Entry(window,textvariable=name,width=50)

e1.grid(row=0,column=0,columnspan=10)

user=StringVar()

e2 = Entry(window,textvariable=user,width=50)

e2.grid(row=1,column=0,columnspan=10)

password=StringVar()

e3 = Entry(window,textvariable=password,width=50)

e3.grid(row=2,column=0,columnspan=10)

category=StringVar()

e4 = Entry(window,textvariable=category,width=50)

e4.grid(row=3,column=0,columnspan=10)

cdate=StringVar()

e5 = Entry(window,textvariable=cdate,width=50)

e5.grid(row=4,column=0,columnspan=10)

b1 = Button(window,text="Add",width=12,command=add\_command)

b1.grid(row=5,column=0)

b2 = Button(window,text="Update",width=12,command=update\_command)

b2.grid(row=5,column=1)

b3 = Button(window,text="Search",width=12,command=search\_command)

b3.grid(row=5,column=2)

b4 = Button(window,text="View All",width=12,command=view\_command)

b4.grid(row=5,column=3)

b5 = Button(window,text="Delete",width=12,command=delete\_command)

b5.grid(row=5,column=4)

b6 = Button(window,text="Cancel",width=12,command=window.destroy)

b6.grid(row=5,column=5)

b7 = Button(window,text="Clear All",width=12,command=clear\_command)

b7.grid(row=0,column=5)

lb=Listbox(window,height=20,width=94)

lb.grid(row=6,column=0,columnspan=6)

sb=Scrollbar(window)

sb.grid(row=6,column=6,rowspan=6)

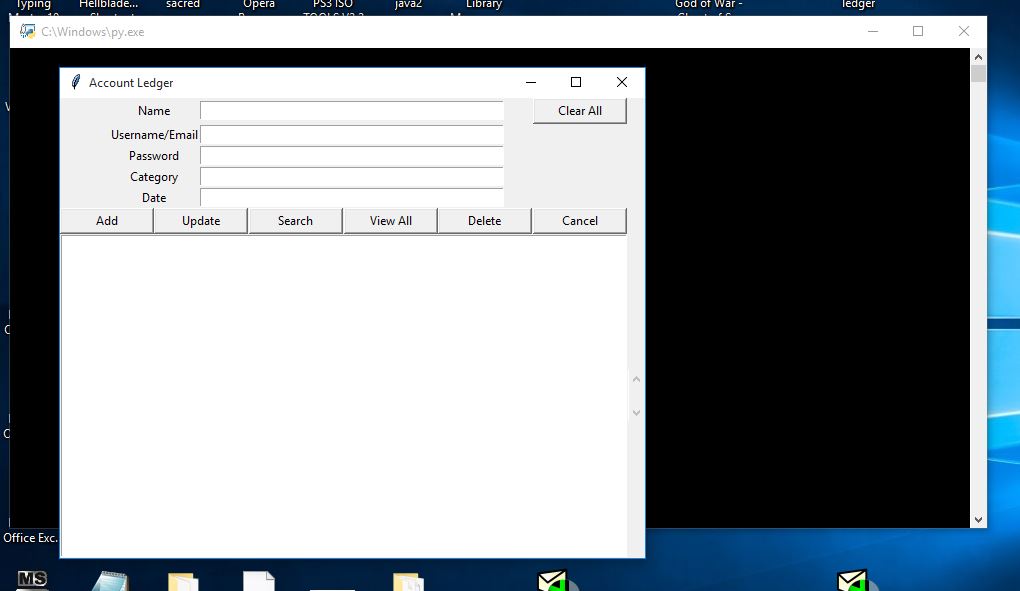
lb.configure(yscrollcommand=sb.set)

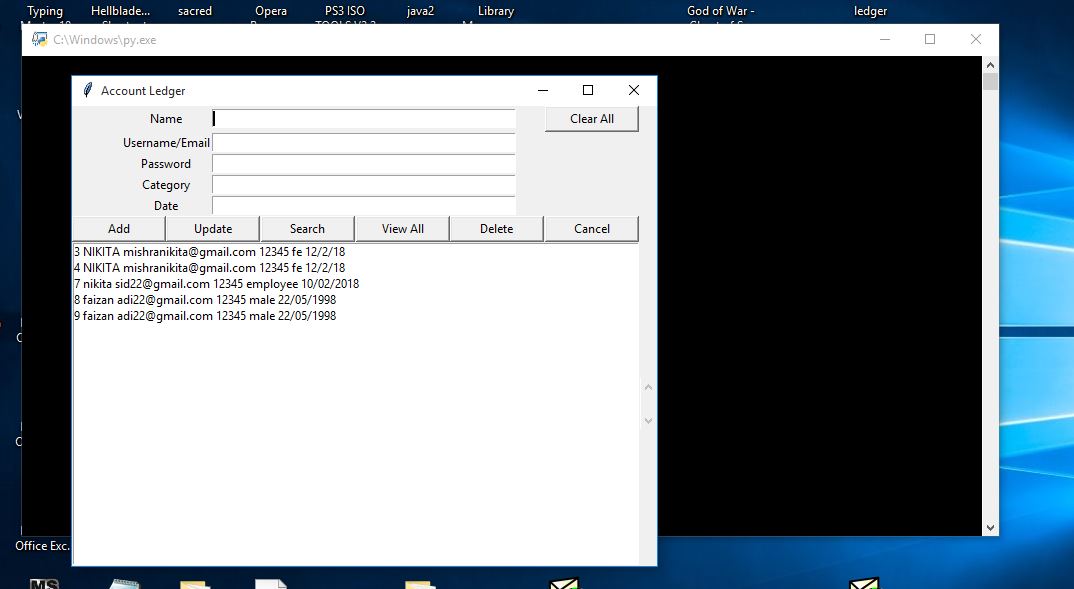
sb.configure(command=lb.yview)

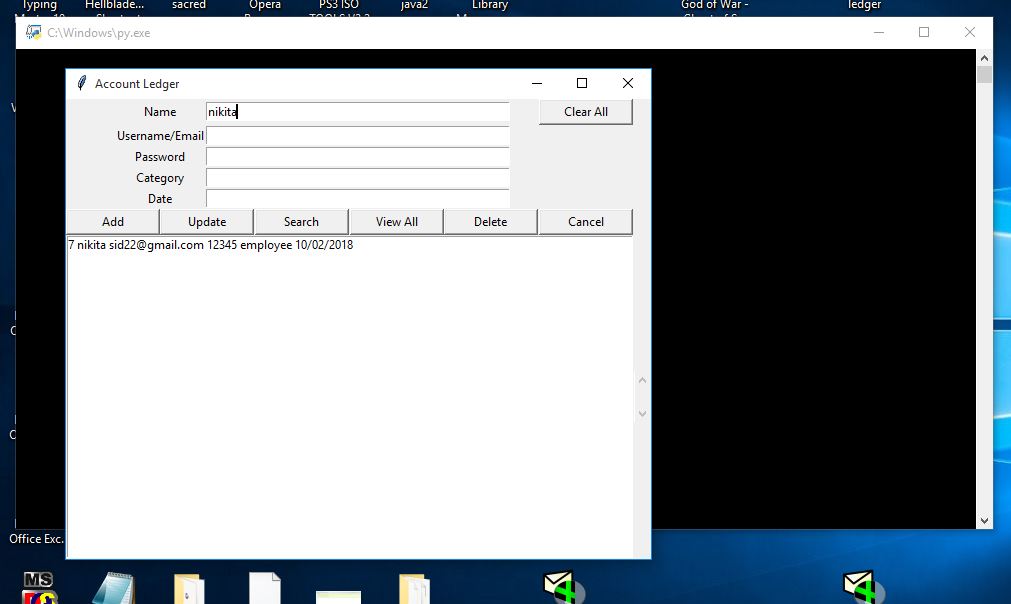
lb.bind('<<ListboxSelect>>',get\_selected\_row)

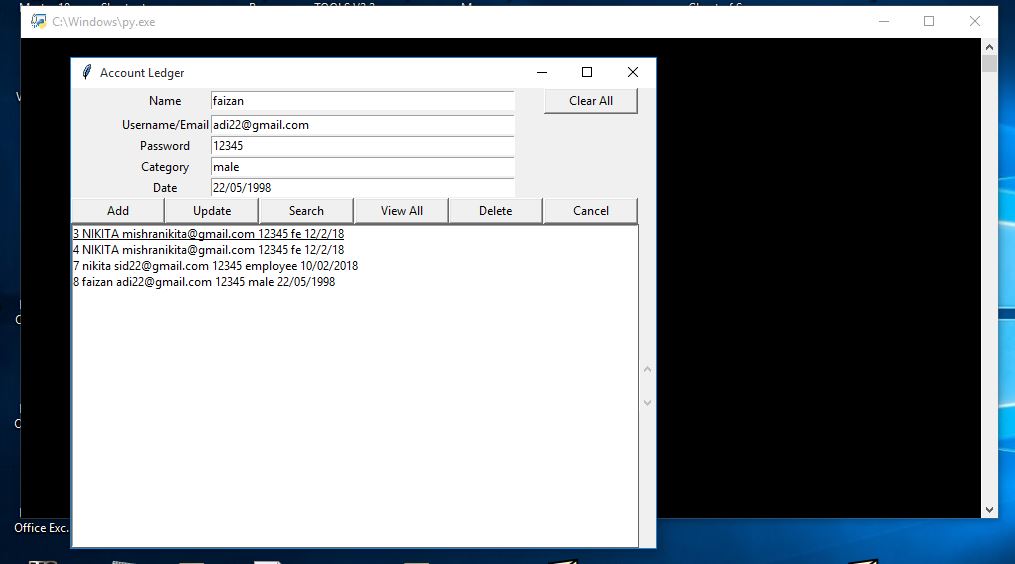
window.mainloop()

**RESULTS**

****

****

****

****

**CONCLUSION**

We came across various difficulties while making the project, like solving the errors and linking SQL to python. But we a lot in the process. The knowledge gained will definitely help us in future. We will carry this information in future

**REFERENCES**

These are the following works I referred to

<https://code-projects.org/account-management-system-in-python-with-source-code/>

<https://github.com/Martialhimanshu/Account-management-system>

<https://www.codester.com/items/4605/easy-client-management-using-python>

<https://www.tutorialspoint.com/python/python_gui_programming.htm>