SHREE SANATAN DHARM EDUCATION CENTRE

Affiliated to Central Board of Secondary Education, Affiliation No.: 2130245

COMPUTER SCIENCE(083)

PROJECT

A PROGRAMMER'S PERSPECTIVE

MADE BY: SHANTANU GUPTA

CLASS: XII-A

ROLL NO.: 23669764

2021-22

ARIMS

Army Relational Information Management System

CERTIFICATE

This is to certify that Cadet SHANTANU GUPTA has successfully completed the project Work entitled Army Records Information Management System (ARIMS) in the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in Shree Sanatan Dharm Education Center.

Internal Examiner

External Examiner

Principal

ACKNOWLEDGEMENT

I would like to convey my heartfelt thanks to
Ms. Shobha Das
my school Principal for providing us with necessary resources and
Mr. Dinesh Maurya
my Computer Science Teacher,
who gave valuable suggestions and guidelines for completion of my project.
My project has been a success only because of his guidelines.

SHANTANU GUPTA
XII-A
Board Roll No : 23669764

INDEX

Project on Army Records Information Management System (ARIMS)

INTRODUCTION

Military affairs are chaotic in nature. Statistics, occurrences, and the organization of materials and equipment are difficult to manage. Battlefield commanders require the ability to view large amounts of information in a simplistic manner, particularly as ongoing situations become more complex and more units, personnel, and equipment enter the battle space. The Army requires that battle space material logistics be displayed by hierarchy. This system is designed to connect with a large database of personnel, equipment items, vehicles, and consumables. These entities are organized by military hierarchal structure which can be traversed at a glance, giving commanders a visual summary.

OBJECTIVES OF THE PROJECT

Primarily, the goal of the ARIMS system is to allow for quick data access in a highly visual manner. The basis for the design was to create a system which can make a tremendous volume of information seem simple. This was accomplished by displaying this huge data in tabular format and displaying the limited and only the required data. The interface is also user friendly allowing the user full control over the tremendous data.

EXISTING SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today’s merciless competition where not to wise saying “to err is human” no longer valid, it’s outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

PROPOSED SYSTEM

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

I.OPERATING SYSTEM	:	WINDOWS 7 AND ABOVE
II. PROCESSOR	:	PENTIUM(ANY) OR AMD ATHALON(3800+- 4200+ DUAL CORE)
III. MOTHERBOARD	:	1.845 OR 915,995 FOR PENTIUM
		Or

K9MM-V VIA K8M800+8237R PLUS CHIPSET FOR AMD ATHALON

IV. RAM : 512 MB+

V. Hard disk : SATA 40 GB OR ABOVE

VI. CD/DVD r/w multi drive combo: (If back up required)

VIII. MONITOR : 14.1 or 15 -17 inch

IX. Key board and mouse

SOFTWARE REQUIREMENTS

Windows OS

Python 3 or above

SET UP

After you get the file “first_page.zip” extract this file to the desired location. After extraction you should have a file named “first_page.py” and a folder named “sprites”. Check the following files in the sprites folder.

- kisspng-army-day-indian-army-desktop-wallpaper-image-anti-terrorism-amp-counter-ied-solutions-5b6c7accf21ee0.8001570415338359809917.png
- artillery.png
- missiles.png
- nurse.png
- soldier.png
- soldier1.png
- tank.png
- worker.png

After you have checked. You are good to go just open “first_page.py” in python IDLE and run it.

Code

```
import sys
import os
import pickle
from datetime import date
from tkinter import *
from tkinter import messagebox,ttk
from PIL import Image,ImageTk
sample_data = []
data = 0
lg_t = 0
c = 0
doctors = 0
soldiers = 0
workers = 0
today_data = 0
table_ids = ()
table_tids = ()
date_ = str(date.today().strftime('%d%b%Y'))
main_window = Tk()
main_window.title('ARIMS')
icon = PhotoImage(file='sprites\\soldier1.png')
main_window.iconphoto(1, icon)
def ex(event=0):
    if messagebox.askokcancel('Exit','Do you really want to exit ?'):
        password_window.destroy()
        main_window.destroy()
        sys.exit()
def set_up():
    global data
    def cancel_fun(event=0):
        setup_window.destroy()
        password_window.destroy()
        main_window.destroy()
    def ok_fun(event=0):
        global data
        if entry_newp.get() == entry_conp.get():
            file = open('pwd.dat', 'wb')
            pickle.dump(entry_conp.get(),file)
            file.close()
            data = entry_conp.get()
            setup_window.destroy()
            messagebox.showinfo('Password applied','Password set up succesful!!!')
            password_window.deiconify()
        else:
            messagebox.showerror('Error',"Password don't match")
            setup_window.deiconify()
            entry_newp.delete(0,END)
            entry_conp.delete(0,END)
    setup_window = Toplevel(main_window, bg='#2f2f2f', padx=10)
    setup_window.title('Set password')
    setup_window.geometry('600x300+250+100')
    setup_window.protocol('WM_DELETE_WINDOW', ex)
    setup_window.resizable(0, 0)
    lbl_setup = Label(setup_window, text='Firstly let us set a password for ARIMS',
font=('roboto thin', 20), bg='#2f2f2f', fg='light yellow')
    lbl_newp = Label(setup_window, text='Password:', font=('roboto thin', 15),
bg='#2f2f2f', fg='light yellow')
    entry_newp = Entry(setup_window, font=('roboto light', 10), bg='#333333', fg='white',
show='*')
    lbl_conp = Label(setup_window, text='Confirm Password:', font=('roboto thin', 15),
bg='#2f2f2f', fg='light yellow')
    entry_conp = Entry(setup_window, font=('roboto light', 10), bg='#333333',
fg='white',show='*')
```

```
    entry_conp.bind('<KeyPress-Return>', ok_fun)
    ok_btn = Button(setup_window, font=('roboto light', 10), bg='#5f5f5f', fg='white',
text='OK', padx=30, command=ok_fun)
    ok_btn.bind('<KeyPress-Return>', ok_fun)
    cancel_btn = Button(setup_window, font=('roboto light', 10), bg='#5f5f5f',
fg='white', text='Cancel', padx=20,command=cancel_fun)
    cancel_btn.bind('<KeyPress-Return>', cancel_fun)
    lbl_setup.pack(fill = X)
    lbl_newp.place(x=20, y=90)
    entry_newp.place(x=20, y=120)
    lbl_conp.place(x=20, y=160)
    entry_conp.place(x=20, y=190)
    ok_btn.place(x=20, y=250)
    cancel_btn.place(x=110, y=250)
if not os.path.exists('pwd.dat'):
    data = 0
    set_up()
else:
    file = open('pwd.dat','rb')
    data = pickle.load(file)
    file.close()
if os.path.exists(date_+'.dat'):
    today_file = open(date_+'.dat','rb')
    today_data = pickle.load(today_file)
    today_file.close()
    c = len(today_data)
if os.path.exists('main.dat'):
    main_file = open('main.dat','rb')
    main_data = pickle.load(main_file)
    main_file.close()
    main_len = len(main_data) -1
    for rec in main_data:
        sample_data.append([rec[0],rec[10]])
        if rec[-2] == 'Soldier' :
            soldiers += 1
        if rec[-2] == 'Doctor' :
            doctors += 1
        if rec[-2] == 'Engineer' :
            workers += 1
else:
    main_data = []
    main_len = 0
    sample_data = []
if os.path.exists('gun.dat'):
    gun_file = open('gun.dat','rb')
    gun_data = pickle.load(gun_file)
    last_date = pickle.load(gun_file)
    gun_file.close()
    gun_counter = len(gun_data) - 1
else:
    gun_data = []
    gun_counter = 0
    last_date = ''
if os.path.exists('Panzer.dat'):
    t_file = open('Panzer.dat','rb')
    t_data = pickle.load(t_file)
    lt_date = pickle.load(t_file)
    t_counter = len(t_data) - 1
    t_file.close()
    tank = 0
    missile = 0
    artillery = 0
    for rec in t_data:
        if rec[1] == 'Tank':
            tank += 1
        elif rec[1] == 'Missile Truck':
            missile += 1
        else:
            artillery += 1
else:
    t_data = []
    t_counter = 0
    lt_date = ''
    tank = 0
    missile = 0
    artillery = 0
def clear_all():
    global main_data,main_len
    entry_fname.delete(0, END)
    entry_lname.delete(0, END)
    entry_faname.delete(0, END)
    entry_moname.delete(0, END)
    entry_height.delete(0, END)
    entry_weight.delete(0, END)
    entry_ydob.delete(0, END)
    entry_mdob.delete(0, END)
    entry_ddob.delete(0, END)
    combo_blood.delete(0, END)
    entry_home.delete(0, END)
    combo_religion.delete(0, END)
    entry_solderid.delete(0, END)
    entry_Lcity.delete(0, END)
    entry_Was.delete(0, END)
    entry_ydoj.delete(0, END)
    entry_mdoj.delete(0, END)
    entry_ddoj.delete(0, END)
    entry_rank.delete(0, END)
def fill_all(length):
    global main_data
    name = main_data[length][0].split()
    DOB = main_data[length][1].split('-')
    DOJ = main_data[length][11].split('-')
    entry_fname.insert(0, name[0])
    entry_lname.insert(0, name[1])
    entry_faname.insert(0, main_data[length][2])
    entry_moname.insert(0, main_data[length][3])
    if main_data[length][4] == 'male':
        radio_male.select()
    else:
        radio_female.select()
    entry_height.insert(0, main_data[length][5])
    entry_weight.insert(0, main_data[length][6])
    entry_ydob.insert(0, DOB[2])
    entry_mdob.insert(0, DOB[1])
    entry_ddob.insert(0, DOB[0])
    combo_blood.insert(0, main_data[length][9])
    entry_home.insert(0, main_data[length][7])
    combo_religion.insert(0, main_data[length][8])
    entry_solderid.insert(0, main_data[length][10])
    entry_Lcity.insert(0, main_data[length][12])
    entry_Was.insert(0, main_data[length][13])
    entry_ydoj.insert(0, DOJ[2])
    entry_mdoj.insert(0, DOJ[1])
    entry_ddoj.insert(0, DOJ[0])
    entry_rank.insert(0, main_data[length][14])
def get_all():
    name = str(entry_fname.get()) + ' ' + str(entry_lname.get())
    DOB = date(int(entry_ydob.get()), int(entry_mdob.get()),
int(entry_ddob.get())).strftime('%d-%m-%Y')
    faname = str(entry_faname.get())
    moname = str(entry_moname.get())
    gender1 = gender.get()
    height = str(entry_height.get())
    weight = str(entry_weight.get())
    home = str(entry_home.get())
    rel = str(combo_religion.get())
    blood = str(combo_blood.get())
    soldierid = str(entry_solderid.get())
    DOJ = date(int(entry_ydoj.get()), int(entry_mdoj.get()),
int(entry_ddoj.get())).strftime('%d-%m-%Y')
    Lcity = str(entry_Lcity.get())
    Was = str(entry_Was.get())
    rank = str(entry_rank.get())
    ele = [name, DOB, faname, moname, gender1, height, weight, home, rel, blood,
soldierid, DOJ, Lcity, Was, rank]
    return ele
def showFrame(frame):
    global name,solid
    frame.tkraise()
    if sample_data:
        name.set(sample_data[0][0])
        solid.set(sample_data[0][1])
def password(event=0):
    global entry_password, st, password_window, file, main_window,lg_t
    st = entry_password.get()
    if st != data:
        entry_password.delete(0, END)
        messagebox.showwarning('Warning', 'Enter correct password')
        password_window.deiconify()
        lg_t += 1
    else:
        password_window.destroy()
        main_window.deiconify()
        frame1.tkraise()
    if lg_t >= 3:
        password_window.destroy()
        main_window.destroy()
        sys.exit()
def pre_nxt_fun(e):
    global table,c
    if not e:
        try:
            c += 1
            name.set(sample_data[c][0])
            solid.set(str(sample_data[c][1]))
        except:
            IndexError
            messagebox.showinfo('Information','End of records')
            c -= 1
    else:
        try:
            c -= 1
            if not c<0 :
                name.set(sample_data[c][0])
                solid.set(str(sample_data[c][1]))
            else:
                raise IndexError
        except:
            IndexError
            messagebox.showinfo('Information','End of records')
            c += 1
def add_fun():
    try:

table.insert(parent='',iid=c,index=END,values=(name.get(),solid.get(),present.get(),statu
s.get()))
    except:
        TclError
        messagebox.showinfo('info','Record already exists!!!')
def update_fun():
    i_id = int(table.selection()[0])
    table.set(i_id,column='name',value=name.get())
    table.set(i_id,column='soldierid',value=solid.get())
    table.set(i_id,column='present',value=present.get())
    table.set(i_id,column='status',value=status.get())
def save_fun(dat):
    global table
    today_data1 = []
    for i in range(len(table.get_children())):
        today_data1.append(table.item(i)["values"])
    today_file1 = open(dat+'.dat','wb')
    pickle.dump(today_data1,today_file1)
    today_file1.close()
    messagebox.showinfo('Save','Data was saved')
def search_fun():
    global search,entry_name,c
    searched = search.get()
    search.set('Search by')
    btn_search.config(state = 'disabled')
    c = len(table.get_children())
    if searched == 'Name':
        searched_name = entry_name.get()
        for i in range(len(table.get_children())):
            if table.item(i)['values'][0] != searched_name:
                table.detach(i)
            else:
                table.reattach(i,parent='',index = END)
    elif searched == 'Soildier ID':
        searched_id = entry_soldierID.get()
        for i in range(len(table.get_children())):
            if str(table.item(i)['values'][1]) != searched_id:
                table.detach(i)
            else:
                table.reattach(i,parent='',index = END)
    elif searched == 'Presentees' :
        for i in range(len(table.get_children())):
            if str(table.item(i)['values'][2]) != 'yes':
                table.detach(i)
            else:
                table.reattach(i,parent='',index = END)
    elif searched == 'Healthy' :
        for i in range(len(table.get_children())):
            if str(table.item(i)['values'][3]) != 'healthy':
                table.detach(i)
            else:
                table.reattach(i,parent='',index = END)
    else:
        btn_search.config(state='normal')
        messagebox.showinfo('info','Please select option')
def done_fun():
    entry_name.delete(0, END)
    entry_soldierID.delete(0, END)
    btn_search.config(state='normal')
    btn_done.config(state = "disabled")
    for i in range(c):
        table.reattach(i, parent='', index=END)
def dates():
    global date_,Date,c
    def cancel_fun(event=0):
        date_window.destroy()
    def ok_fun(event=0):
        global date_,Date,c
        try:
            day = int(entry_date.get())
            month = int(spin_month.get())
            year = int(entry_year.get())
            user_date = date(year,month,day).strftime('%d%b%Y')
            if os.path.exists(user_date + '.dat'):
                date_ = user_date
                Date.set('Date of file: '+date_)
                user_file = open(user_date + '.dat', 'rb')
                user_data = pickle.load(user_file)
                c = len(user_data)
                user_file.close()
                for item in table.get_children():
                    table.delete(item)
                for ele in range(len(user_data)):
                    table.insert(parent='',iid=ele,index=END,values=user_data[ele])
                date_window.destroy()
            elif user_date == date.today().strftime('%d%b%Y'):
                date_ = user_date
                date_window.destroy()
                Date.set('Date of file: '+date_)
                btn_pre.config(state="normal")
                btn_nxt.config(state='normal')
                btn_add.config(state='normal')
                for i_id in table.get_children():
                    table.delete(int(i_id))
            else:
                messagebox.showinfo('info', 'File does not exists')
                date_window.deiconify()
        except:
            ValueError
            messagebox.showerror('Error','Enter valid date')
            date_window.deiconify()
    #========DATE WINDOW widgets
    date_window = Toplevel(main_window, bg='#2f2f2f', padx=10, takefocus=True)
    date_window.title('Enter date')
    date_window.geometry('300x300+400+200')
    date_window.resizable(0, 0)
    lbl_day = Label(date_window,text = 'Day:', font=('roboto thin',15), bg='#2f2f2f', fg
= 'light yellow')
    entry_date = Entry(date_window,font=('roboto light',10), bg='#5f5f5f', fg = 'white')
    lbl_month = Label(date_window,text = 'Month:', font=('roboto thin',15), bg='#2f2f2f',
fg = 'light yellow')
    spin_month = Spinbox(date_window,from_ = 1,to = 12,font=('roboto light',10),
bg='#5f5f5f', fg = 'white',buttonbackground = 'dark grey')
    lbl_year = Label(date_window,text = 'Year:', font=('roboto thin',15), bg='#2f2f2f',
fg = 'light yellow')
    entry_year = Entry(date_window,font=('roboto light',10), bg='#5f5f5f', fg = 'white')
    ok_btn = Button(date_window,font=('roboto light',10), bg='#5f5f5f', fg = 'white',text
= 'OK',padx = 30,command = ok_fun)
    ok_btn.bind('<KeyPress-Return>',ok_fun)
    cancel_btn = Button(date_window,font=('roboto light',10), command =
cancel_fun,bg='#5f5f5f', fg = 'white',text = 'Cancel',padx = 20)
```

```
    cancel_btn.bind('<KeyPress-Return>',cancel_fun)
    #======Date Window
    lbl_day.place(x = 20,y = 20)
    entry_date.place(x = 20 ,y= 50)
    lbl_month.place(x = 20,y = 90)
    spin_month.place(x = 20 ,y= 120)
    lbl_year.place(x = 20 ,y= 160)
    entry_year.place(x = 20 ,y= 190)
    ok_btn.place(x=20,y= 250)
    cancel_btn.place(x = 110 ,y = 250)
    date_window.deiconify()
def show(event):
    global btn_update
    btn_update.config(state = NORMAL)
    i = int(table.selection()[0])
    insert_values = table.item(i)['values']
    name.set(insert_values[0])
    solid.set(insert_values[1])
    present.set(insert_values[2])
    status.set(insert_values[3])
def show_gun(event):
    parent = str(event.widget.master.master)
    if int(parent[-1]) == 3:
        btn_update3_1.config(state = 'normal')
        i = int(table3_1.selection()[0])
        insert_values = table3_1.item(i)['values']
        gun_name.set(insert_values[0])
        gun_type.set(insert_values[1])
        gun_instk.set(insert_values[2])
        gun_deploy.set(insert_values[3])
        gun_ammo.set(insert_values[4])
    if int(parent[-1]) == 4:
        btn_update3_2_2.config(state = 'normal')
        i = int(table3_2.selection()[0])
        insert_values = table3_2.item(i)['values']
        t_number.set(insert_values[0])
        t_type.set(insert_values[1])
        t_pilot.set(insert_values[2])
        t_deploy.set(insert_values[3])
        t_ammo.set(insert_values[4])
def combo_fun():
    entry_name.delete(0, END)
    entry_soldierID.delete(0, END)
    btn_done.config(state = NORMAL)
def new_fun():
    global main_len
    main_len = len(main_data)
    clear_all()
def add1_fun():
    global main_data,main_len,soldiers,doctors,workers
    try:
        # ele = [name, DOB, faname, moname, gender1, height, weight, home, rel, blood,
soldierid, DOJ, Lcity, Was, rank]
        ele = get_all()
        if not ele[-5] in [x[-5] for x in main_data]:
            if ele[-2] == 'Soldier' :
                soldiers += 1
            if ele[-2] == 'Doctor' :
                doctors += 1
            if ele[-2] == 'Engineer' :
                workers += 1
            no_of_soldiers.set('Number of Soldiers: ' + str(soldiers))
            no_of_doctors.set('Number of Doctors: ' + str(doctors))
            no_of_workers.set('Number of Engineers: ' + str(workers))
            main_data.append(ele)
            main_len += 1
            combo_pilot3_2_1.config(values=[x[0] for x in main_data if x[-2] != 'Doctor'
and x[-2] != 'Engineer'])
            messagebox.showinfo('Info', 'Record added')
            clear_all()
        else:
            messagebox.showinfo('Info','Record alredy exists')
    except:
        ValueError
        messagebox.showinfo("Info",'Enter valid values')
def save1_fun():
    global main_data,sample_data
    main_file = open('main.dat','wb')
    pickle.dump(main_data,main_file)
    main_file.close()
    sample_data = []
    messagebox.showinfo('Save','Data was saved')
    for rec in main_data:
        sample_data.append([rec[0],rec[10]])
def prev_fun():
    global main_data, main_len
    try:
        main_len -= 1
        if main_len < 0:
            main_len = 0
            messagebox.showinfo('EOF', 'No more values to display')
        else:
            clear_all()
            fill_all(main_len)
    except:
        IndexError
        messagebox.showinfo('EOF', 'No more values to display')
        main_len += 1
def nxt1_fun():
    global main_data, main_len
    try:
        main_len += 1
        clear_all()
        fill_all(main_len)
    except:
        IndexError
        messagebox.showinfo('EOF', 'No more values to display')
        main_len -= 1
        fill_all(main_len)
def update1_fun():
    global main_data,main_len
    ele = get_all()
    main_data[main_len] = ele
def del_fun():
    global main_data, main_len,workers,soldiers,doctors
    choice = messagebox.askokcancel('Confirm ?','Really delete this record ?')
    if choice:
        if main_data[main_len][-2] == 'Soldier' :
            soldiers -= 1
        if main_data[main_len][-2] == 'Doctor' :
            doctors -= 1
        if main_data[main_len][-2] == 'Engineer' :
            workers -= 1
        no_of_soldiers.set('Number of Soldiers: ' + str(soldiers))
        no_of_doctors.set('Number of Doctors: ' + str(doctors))
        no_of_workers.set('Number of Engineers: ' + str(workers))
        main_data.pop(main_len)
        main_len -= 1
        combo_pilot3_2_1.config(values=[x[0] for x in main_data if x[-2] != 'Doctor'])
        clear_all()
        fill_all(main_len)
def cng_pwd():
    global data
    def check(event):
        old_pwd = entry_oldp.get()
        if old_pwd == data:
            entry_newp.config(state='normal')
            entry_conp.config(state='normal')
        else:
            messagebox.showwarning('Warning','Wrong Password')
            entry_oldp.delete(0,END)
            cng_window.deiconify()
    def cancel_fun(event=0):
        cng_window.destroy()
    def ok_fun(event=0):
        global data
        old_pwd = entry_oldp.get()
        if entry_newp['state'] == 'normal':
            if entry_newp.get() == entry_conp.get():
                file = open('pwd.dat', 'wb')
                pickle.dump(entry_conp.get(),file)
                file.close()
                data = entry_conp.get()
                cng_window.destroy()
                messagebox.showinfo('Password Changed','Password changed succesfully!!!')
            else:
                messagebox.showerror('Error',"Password don't match")
                cng_window.deiconify()
                entry_newp.delete(0,END)
                entry_conp.delete(0,END)
        elif old_pwd:
            if old_pwd == data:
                entry_newp.config(state='normal')
                entry_conp.config(state='normal')
            else:
                messagebox.showwarning('Warning', 'Wrong Password')
                entry_oldp.delete(0, END)
                cng_window.deiconify()
        else:
            messagebox.showwarning('Warning','Enter old password first')
    cng_window = Toplevel(main_window, bg='#2f2f2f', padx=10)
    cng_window.title('Change password')
    cng_window.geometry('300x300+450+100')
    cng_window.resizable(0, 0)
    cng_window.deiconify()
    lbl_oldp = Label(cng_window, text='Old Password:', font=('roboto thin', 15),
bg='#2f2f2f', fg='light yellow')
    entry_oldp = Entry(cng_window, font=('roboto light', 10), bg='#333333', fg='white')
    entry_oldp.bind('<KeyPress-Return>',check)
    lbl_newp = Label(cng_window, text='New password:', font=('roboto thin', 15),
bg='#2f2f2f', fg='light yellow')
    entry_newp = Entry(cng_window, font=('roboto light', 10), bg='#333333',
fg='white',state = "disabled",disabledbackground = "#5f5f5f",show = '*')
    lbl_conp = Label(cng_window, text='Confirm Password:', font=('roboto thin', 15),
bg='#2f2f2f', fg='light yellow')
    entry_conp = Entry(cng_window, font=('roboto light', 10), bg='#333333',
fg='white',state = "disabled",disabledbackground = "#5f5f5f",show = '*')
    ok_btn = Button(cng_window, font=('roboto light', 10), bg='#5f5f5f', fg='white',
text='OK', padx=30,command = ok_fun)
```

```
    ok_btn.bind('<KeyPress-Return>',ok_fun)
    cancel_btn = Button(cng_window, font=('roboto light', 10), bg='#5f5f5f',
fg='white',text='Cancel', padx=20,command = cancel_fun)
    cancel_btn.bind('<KeyPress-Return>',cancel_fun)
    # ======Change Password Window
    lbl_oldp.place(x=20, y=20)
    entry_oldp.place(x=20, y=50)
    lbl_newp.place(x=20, y=90)
    entry_newp.place(x=20, y=120)
    lbl_conp.place(x=20, y=160)
    entry_conp.place(x=20, y=190)
    ok_btn.place(x=20, y=250)
    cancel_btn.place(x=110, y=250)
def reset_t():
    t_number.set('')
    t_type.set('')
    t_deploy.set('No')
    t_pilot.set('')
    t_ammo.set(0)
def reset_gun():
    gun_name.set('')
    gun_type.set('')
    gun_instk.set(0)
    gun_deploy.set(0)
    gun_ammo.set(0)
def add_gun(e):
    if not e:
        try:
            global gun_data,gun_counter
            Gun = gun_name.get()
            Type = gun_type.get()
            InStk = gun_instk.get()
            Deployed = gun_deploy.get()
            Ammo = gun_ammo.get()
            ele = [Gun, Type, InStk, Deployed, Ammo]
            if '' in ele:
                messagebox.showerror('Error', 'Enter valid values !!')
            else:
                gun_data.append(ele)
                gun_counter += 1
                table3_1.insert(parent='',iid=gun_counter,index=END,values = ele)
                reset_gun()
        except:
            TclError
            messagebox.showerror('Error','Enter valid values !!')
    else:
        try:
            global t_counter,t_data,tank,missile,artillery
            ele = [Num,Type,Pilot,Deploy,Ammo] =
[t_number.get(),t_type.get(),t_pilot.get(),t_deploy.get(),t_ammo.get()]
            if '' in ele:
                messagebox.showerror('Error', 'Enter valid values !!')
            else:
                table3_2.insert(parent='', iid=t_counter+1, index=END, values = ele)
                t_data.append(ele)
                t_counter += 1
                reset_t()
                if Type == 'Tank':
                    tank += 1
                    no_tank.set('Tanks:'+str(tank))
                if Type == 'Missile Truck':
                    missile += 1
                    no_missile.set('Missile Trucks:'+str(missile))
                if Type == 'Artillery':
                    artillery += 1
                    no_artillery.set('Artilleries:'+str(artillery))
        except:
            TclError
            messagebox.showerror('Error','Enter valid values !!')
def save_gun(e):
    if not e:
        global gun_data
        gun_data=[]
        last_saved.set('Last saved:'+date.today().strftime('%d/%b/%Y'))
        for item in table3_1.get_children():
            gun_data.append(table3_1.item(item)['values'])
        gun_file = open('gun.dat','wb')
        pickle.dump(gun_data,gun_file)
        pickle.dump(last_saved.get(),gun_file)
        gun_file.close()
        messagebox.showinfo('Save','Data was saved')
    else:
        global t_data
        t_data = []
        last_saved.set('Last saved:'+date.today().strftime('%d/%b/%Y'))
        for item in table3_2.get_children():
            t_data.append(table3_2.item(item)['values'])
        file = open('Panzer.dat','wb')
        pickle.dump(t_data,file)
        pickle.dump(last_saved.get(),file)
        file.close()
        messagebox.showinfo('Save', 'Data was saved')
def update_gun(e):
    if not e:
        try:
            i_id = int(table3_1.selection()[0])
            # ('name', 'type', 'instk', 'deploy', 'ammo')
            table3_1.set(i_id, column='name', value=gun_name.get())
            table3_1.set(i_id, column='type', value=gun_type.get())
            table3_1.set(i_id, column='instk', value=gun_instk.get())
            table3_1.set(i_id, column='deploy', value=gun_deploy.get())
            table3_1.set(i_id, column='ammo', value=gun_ammo.get())
            reset_gun()
        except:
            TclError
            messagebox.showerror('Error','Enter Valid values')
    else:
        try:
            i_id = int(table3_2.selection()[0])
            # ('name', 'type', 'instk', 'deploy', 'ammo')
            table3_2.set(i_id, column='number', value=t_number.get())
            table3_2.set(i_id, column='type', value=t_type.get())
            table3_2.set(i_id, column='commander', value=t_pilot.get())
            table3_2.set(i_id, column='deploy', value=t_deploy.get())
            table3_2.set(i_id, column='ammo', value=t_ammo.get())
            reset_t()
        except:
            TclError
            messagebox.showerror('Error','Enter Valid values')
def combo_gun_fun(e):
    if e:
        reset_gun()
        btn_search3_1.config(state = NORMAL)
    else:
        reset_t()
        btn_search3_2_2.config(state = NORMAL)
```

```
def search_gun(e):
    global table_ids,table_tids
    if e:
        btn_search3_1.config(state = DISABLED)
        btn_see_all3_1.config(state=NORMAL)
        table_ids = table3_1.get_children()
        if gun_search.get() == 'Name':
            for i in (table_ids):
                if table3_1.item(i)['values'][0] != gun_name.get():
                    table3_1.detach(i)
                else:
                    table3_1.reattach(i,parent='',index=END)
        if gun_search.get() == 'Type':
            for i in (table_ids):
                if table3_1.item(i)['values'][1] != gun_type.get():
                    table3_1.detach(i)
                else:
                    table3_1.reattach(i,parent='',index=END)
    else:
        btn_search3_2_2.config(state = DISABLED)
        btn_see_all3_2_2.config(state=NORMAL)
        table_tids = table3_2.get_children()
        if t_search.get() == 'Number':
            for i in table_tids:
                if table3_2.item(i)['values'][0] != t_number.get():
                    table3_2.detach(i)
                else:
                    table3_2.reattach(i,parent="",index=END)
        elif t_search.get() == 'Type':
            for i in table_tids:
                if table3_2.item(i)['values'][1] != t_type.get():
                    table3_2.detach(i)
                else:
                    table3_2.reattach(i,parent="",index=END)
        elif t_search.get() == 'Commander':
            for i in table_tids:
                if table3_2.item(i)['values'][2] != t_pilot.get():
                    table3_2.detach(i)
                else:
                    table3_2.reattach(i,parent="",index=END)
        elif t_search.get() == 'Deployed':
            for i in table_tids:
                if table3_2.item(i)['values'][3] != 'Yes':
                    table3_2.detach(i)
                else:
                    table3_2.reattach(i,parent="",index=END)
        else:
            pass
def see_all_gun(e):
    if e:
        for i in table_ids:
            table3_1.reattach(i, parent='', index=END)
        gun_search.set('search by')
        btn_see_all3_1.config(state = DISABLED)
        btn_search3_1.config(state = NORMAL)
    else:
        for i in table_tids:
            table3_2.reattach(i,parent='',index=END)
        btn_see_all3_2_2.config(state=DISABLED)
        btn_search3_2_2.config(state = NORMAL)
def del_gun(e):
    if not e:
        global gun_counter
        if messagebox.askokcancel('Confirm ?','Really delete this record ?'):
            i_id = int(table3_1.selection()[0])
            table3_1.delete(i_id)
            reset_gun()
            gun_counter -= 1
    else:
        global t_counter,tank,missile,artillery
        if messagebox.askokcancel('Confirm ?', 'Really delete this record ?'):
            i_id = int(table3_2.selection()[0])
            Type = (table3_2.item(i_id)['values'][1])
            table3_2.delete(i_id)
            reset_t()
            if Type == 'Tank':
                tank -= 1
                no_tank.set('Tanks:' + str(tank))
            if Type == 'Missile Truck':
                missile -= 1
                no_missile.set('Missile Trucks:' + str(missile))
            if Type == 'Artillery':
                artillery -= 1
                no_artillery.set('Artilleries:' + str(artillery))
password_window = Toplevel()
password_window.title('ARIMS')
#=========password window widgets
army_img = Image.open('sprites\\kisspng-army-day-indian-army-desktop-wallpaper-image-
anti-terrorism-amp-counter-ied-solutions-5b6c7accf21ee0.8001570415338359809917.png')
army_img_re = army_img.resize((150,150))
army_logo = ImageTk.PhotoImage(army_img_re)
lbl_wlcm = Label(password_window, text='Welcome !', font=('roboto thin',40), bg='dark
green', fg = 'light yellow')
lbl_army = Label(password_window,text='Indian Army', font=('roboto thin',20),
bg='#2f2f2f', fg = 'light yellow',image = army_logo,compound='bottom')
lbl_passwd = Label(password_window, text = 'Enter password:', font = ('roboto thin',20),
bg='#2f2f2f', fg = 'light yellow')
entry_password = Entry(password_window, font=('roboto light', 10, 'underline'),
bg='#2f2f2f', fg='light blue', width = 15, bd = 2, relief = RIDGE, insertbackground
='light blue', insertborderwidth = 5, justify = CENTER, show ='#')
entry_password.bind("<KeyPress-Return>", password)
submit_btn = Button(password_window, text='submit', font=('roboto thin',20), fg='light
yellow', command=password, bg='#262626', bd=0, activebackground = '#262626',
activeforeground = 'light yellow')
exit_btn = Button(password_window, text='exit', font=('roboto thin',20), fg='light
yellow', command = ex, bg = 'dark green', bd=0, activebackground = 'dark green',
activeforeground = 'light yellow')
if not data:
    password_window.withdraw()
s = ttk.Style()
s.theme_use("default")
#=======Variables
name = StringVar()
solid = StringVar()
present = StringVar()
status = StringVar()
search = StringVar()
gun_search = StringVar()
t_search = StringVar()
Date = StringVar()
gender = StringVar()
gun_name = StringVar()
gun_type = StringVar()
gun_instk = IntVar()
gun_deploy = IntVar()
gun_ammo = IntVar()
```

```
t_number = StringVar()
t_type = StringVar()
t_pilot = StringVar()
t_deploy = StringVar()
t_deploy.set('Yes')
t_ammo = IntVar()
t_search.set('search by')
no_of_soldiers = StringVar()
no_of_doctors = StringVar()
no_of_workers = StringVar()
no_tank = StringVar()
no_artillery = StringVar()
no_missile = StringVar()
last_saved = StringVar()
lt_saved = StringVar()
if last_date:
    last_saved.set(last_date)
if lt_date:
    lt_saved.set(lt_date)
no_of_soldiers.set('Number of Soldiers: '+str(soldiers))
no_of_doctors.set('Number of Doctors: '+str(doctors))
no_of_workers.set('Number of Engineers: '+str(workers))
no_tank.set('Tanks:'+str(tank))
no_artillery.set('Artilleries:'+str(artillery))
no_missile.set('Missile Trucks:'+str(missile))
gender.set('male')
Date.set('Date of file: '+date_)
search.set('Search by')
gun_search.set('search by')
if sample_data:
    name.set(sample_data[0][0])
    solid.set(sample_data[0][1])
present.set('yes')
status.set("healthy")
#======weopen menu
weopon_menu = Menu(main_window, tearoff = 0)
weopon_menu.add_command(label = 'Hand held weopons',command =lambda :showFrame(frame3_1))
weopon_menu.add_separator()
weopon_menu.add_command(label = 'Heavy transports',command =lambda :showFrame(frame3_2))
#=======more menu
more_menu = Menu(main_window, tearoff = 0)
more_menu.add_command(label = 'Change password',command = cng_pwd)
more_menu.add_separator()
more_menu.add_command(label = 'Exit',command = ex)
#========main menu
main_menu = Menu(main_window)
main_menu.add_command(label = 'Daily record',command = lambda :showFrame(frame1))
main_menu.add_command(label = 'Individual record',command =lambda :showFrame(frame2))
main_menu.add_cascade(label = 'Weoponary',menu = weopon_menu)
main_menu.add_cascade(label = 'More',menu = more_menu)
#=========MAIN WINDOW WIDGETS
frame1 = Frame(main_window,bg = "#303030" )
frame2 = Frame(main_window,bg = "#303030", padx = 8,pady = 8)
frame3_1 = Frame(main_window,bg = "#303030" )
frame3_2 = Frame(main_window,bg = "#303030" )
frame1_1 = Frame(frame1,bg = "#252525")
frame1_2 = Frame(frame1,bg = "#202020")
frame1_3 = Frame(frame1,bg = '#303030',bd=3, relief = RAISED)
#=========FRAME_1_1_WIDGETS
lbl_name = Label(frame1_1,text = "Name:", font = ('roboto thin',20),bg = "#252525",fg = '#B8CF69')
entry_name = Entry(frame1_1,bg = '#353535',width = 20,fg = '#B8CF69', font = ('roboto light',15),textvariable = name)
```

```
lbl_soldierID = Label(frame1_1,text = "Soldier ID:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_soldierID = Entry(frame1_1,bg = '#353535',width = 20,fg = '#B8CF69', font =
('roboto light',15),textvariable = solid)
check_present = Checkbutton(frame1_1,text = "Present", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69',selectcolor='#2f2f2f',variable = present,onvalue = "yes",
offvalue = 'no')
lbl_status = Label(frame1_1,text = "Status:", font = ('roboto thin',20),bg = "#252525",fg
= '#B8CF69')
radio_healthy = Radiobutton(frame1_1,text = "healthy", font = ('roboto thin',15),bg =
"#252525",fg = '#B8CF69',variable = status,value = "healthy",selectcolor = 'black')
radio_unhealthy = Radiobutton(frame1_1,text = "unhealthy", font = ('roboto thin',15),bg =
"#252525",fg = '#B8CF69',variable = status,value = "unhealthy",selectcolor = 'black')
#=========frame1_2 widgets
btn_pre = Button(frame1_2,text = 'Prev', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2, command = lambda: pre_nxt_fun(1))
btn_nxt = Button(frame1_2,text = 'Next', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2, command = lambda: pre_nxt_fun(0))
btn_add = Button(frame1_2,text = 'Add', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,command = add_fun)
btn_update = Button(frame1_2,text = 'Update', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2, command = update_fun, state = "disabled")
btn_save = Button(frame1_2,text = 'Save', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,command =lambda :save_fun(date_))
btn_date = Button(frame1_2,text = 'Date search', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,command = dates)
btn_search = Button(frame1_2,text = 'Search', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,command = search_fun)
combo_search = ttk.Combobox(frame1_2,values = ('Name','Soildier
ID','Presentees','Healthy'), font = ('roboto thin',20),width = 8,textvariable =
search,postcommand = combo_fun,state = 'readonly')
btn_done = Button(frame1_2,text = 'See all', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,state = "disabled",command = done_fun)
#========frame1_3 widgets
lbl_date = Label(frame1_3,textvariable = Date, font = ('roboto thin',10),bg =
'#303030',fg='light yellow',anchor = W)
scrol_y = Scrollbar(frame1_3,orient = "vertical")
table = ttk.Treeview(frame1_3,columns =
('name','soldierid','present','status'),yscrollcommand=scrol_y.set)
scrol_y.config(command = table.yview)
table.heading('name',text = 'Name')
table.heading('soldierid',text = 'Soldier ID')
table.heading('present',text = 'Present')
table.heading('status',text = 'Status')
table.column("#0",width = 0,stretch = 0)
table.column('name',width = 100,anchor = W)
table.column('soldierid',width = 100,anchor = CENTER)
table.column('present',width = 100,anchor = CENTER)
table.column('status',width = 100,anchor = CENTER)
if today_data:
    for i in range(len(today_data)):
        table.insert(parent='',iid=i,index=END,values=today_data[i])
        btn_pre.config(state="disabled")
        btn_nxt.config(state="disabled")
        btn_add.config(state="disabled")
table.bind("<<TreeviewSelect>>",show)
#===========FRAME 2 widgets
frame2_1 = LabelFrame(frame2,text = 'Personal details',bd = 4,relief = RIDGE,labelanchor
= NW,bg = '#2f2f2f', font = ('roboto thin',20),fg='light yellow',padx = 8,pady = 8)
lbl_fname = Label(frame2_1,text = "First Name:", font = ('roboto thin',13),bg =
"#2f2f2f",fg = '#B8CF69')
entry_fname = Entry(frame2_1,bg = '#2f2f2f',width = 20,fg = 'white', font = ('roboto
light',13))
```

```
lbl_lname = Label(frame2_1,text = "Last Name:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
entry_lname = Entry(frame2_1,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
frm_dob = LabelFrame(frame2_1,text = 'Date of Birth',bd = 2,relief = GROOVE,labelanchor = NW,bg = '#2f2f2f', pady = 8,padx = 8,font = ('roboto thin',13),fg='white',height = 85,width = 250)
lbl_ddob = Label(frm_dob,text = "Day:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_mdob = Label(frm_dob,text = "Month:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_ydob = Label(frm_dob,text = "Year:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
entry_ddob = Entry(frm_dob,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto light',13))
entry_mdob = Entry(frm_dob,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto light',13))
entry_ydob = Entry(frm_dob,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto light',13))
frm_gender = LabelFrame(frame2_1, text='Gender', bd=2, relief=GROOVE, labelanchor=NW, bg='#2f2f2f', pady=8, padx=8,font=('roboto thin', 13), fg='white', height=85, width=250)
radio_male = Radiobutton(frm_gender, text='Male', font=('roboto thin', 13), bg="#2f2f2f", fg='#B8CF69', variable=gender,value='male', selectcolor='black')
radio_female = Radiobutton(frm_gender, text='Female', font=('roboto thin', 13), bg="#2f2f2f", fg='#B8CF69',variable=gender, value='female', selectcolor = 'black')
combo_religion = ttk.Combobox(frame2_1,font = ('roboto light',13),values = ('Hindu','Muslim','Sikh','Christian'),width = 8)
lbl_religion = Label(frame2_1,text = "Religion:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_height = Label(frame2_1,text = "Height(ft):", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_weight = Label(frame2_1,text = "Weight(Kg):", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
entry_faname = Entry(frame2_1,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
entry_moname = Entry(frame2_1,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
entry_height = Entry(frame2_1,bg = '#2f2f2f',width = 20,fg = 'white', font = ('roboto light',13))
entry_weight = Entry(frame2_1,bg = '#2f2f2f',width = 20,fg = 'white', font = ('roboto light',13))
lbl_faname = Label(frame2_1,text = "Father's name:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_moname = Label(frame2_1,text = "Mother's name:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
combo_blood = ttk.Combobox(frame2_1,font = ('roboto light',13),values = ('A+','A','B','B+','AB','O','O+'),width = 4)
lbl_blood = Label(frame2_1,text = "Blood type:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_home = Label(frame2_1,text = "Hometown:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
entry_home = Entry(frame2_1,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
#=========frame2_2
frame2_2 = LabelFrame(frame2,text = 'Professional details',bd = 4,relief = RIDGE,labelanchor = NW,bg = '#2f2f2f', font = ('roboto thin',20),fg='light yellow',padx = 8,pady = 12)
lbl_solderid = Label(frame2_2,text = "Soldier ID:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
lbl_Lcity = Label(frame2_2,text = "Last city:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
entry_solderid = Entry(frame2_2,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
frm_doj = LabelFrame(frame2_2,text = 'Date of joining',bd = 2,relief = GROOVE,labelanchor = NW,bg = '#2f2f2f', pady = 8,padx = 8,font = ('roboto thin',13),fg='white',height = 85,width = 250)
lbl_ddoj = Label(frm_doj,text = "Day:", font = ('roboto thin',13),bg = "#2f2f2f",fg = '#B8CF69')
```

```
lbl_mdoj = Label(frm_doj,text = "Month:", font = ('roboto thin',13),bg = "#2f2f2f",fg =
'#B8CF69')
lbl_ydoj = Label(frm_doj,text = "Year:", font = ('roboto thin',13),bg = "#2f2f2f",fg =
'#B8CF69')
entry_ddoj = Entry(frm_doj,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto
light',13))
entry_mdoj = Entry(frm_doj,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto
light',13))
entry_ydoj = Entry(frm_doj,bg = '#2f2f2f',width = 10,fg = 'white', font = ('roboto
light',13))
lbl_rank = Label(frame2_2,text = "Rank:", font = ('roboto thin',13),bg = "#2f2f2f",fg =
'#B8CF69')
lbl_Was = Label(frame2_2,text = "Working as:", font = ('roboto thin',13),bg =
"#2f2f2f",fg = '#B8CF69')
entry_Lcity = Entry(frame2_2,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
entry_Was = ttk.Combobox(frame2_2, font = ('roboto light',13),values =
('Doctor','Engineer','Soldier'),width=18)
entry_rank = Entry(frame2_2,bg = '#2f2f2f',fg = 'white', font = ('roboto light',13))
#==========frame2 widgets
frame2_3 =  LabelFrame(frame2,bd = 4,relief = RIDGE,bg = '#2f2f2f',text =
'Menu',labelanchor = NW, font = ('roboto thin',20),fg='light yellow')
#==========frame2_3 widgets
btn_prev = Button(frame2_3,text = 'Previous', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=10,command = prev_fun)
btn_nxt1 = Button(frame2_3,text = 'Next', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=25,command = nxt1_fun)
btn_new = Button(frame2_3,text = 'New', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=30,command = new_fun)
btn_add1 = Button(frame2_3,text = 'Add', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=30,command=add1_fun)
btn_save1 = Button(frame2_3,text = 'Save', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=25,command = save1_fun)
btn_update1 = Button(frame2_3,text = 'Update', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=20,command = update1_fun)
btn_del = Button(frame2_3,text = 'Delete', font = ('roboto thin',20),bg =
'#303030',fg='light yellow',bd = 2,padx=20,command = del_fun)
if main_data:
    fill_all(main_len)
#==========frame3_1 widgets
frame3_1_1 = Frame(frame3_1,bg = "#252525")
frame3_1_2 = Frame(frame3_1,bg = "#40433B")
frame3_1_3 = Frame(frame3_1,bg = "#ffffff",bd = 3,relief = RAISED)
frame3_1_4 = Frame(frame3_1,bg = "#535353")
#==========frame3_1_1 widgets
lbl_name3_1 = Label(frame3_1_1,text = "Name:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_name3_1 = Entry(frame3_1_1,bg = '#353535',width = 20,fg = '#B8CF69', font =
('roboto light',15),textvariable = gun_name)
lbl_type = Label(frame3_1_1,text = "Type:", font = ('roboto thin',20),bg = "#252525",fg =
'#B8CF69')
entry_type = Entry(frame3_1_1,bg = '#353535',width = 20,fg = '#B8CF69', font = ('roboto
light',15),textvariable = gun_type)
lbl_instk = Label(frame3_1_1,text = "In stock:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_instk = Entry(frame3_1_1,bg = '#353535',width = 10,fg = '#B8CF69', font = ('roboto
light',15),textvariable = gun_instk)
lbl_deploy = Label(frame3_1_1,text = "Deployed:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_deploy = Entry(frame3_1_1,bg = '#353535',width = 10,fg = '#B8CF69', font = ('roboto
light',15),textvariable = gun_deploy)
lbl_ammo = Label(frame3_1_1,text = "Ammo:", font = ('roboto thin',20),bg = "#252525",fg =
'#B8CF69')
entry_ammo = Entry(frame3_1_1,bg = '#353535',width = 10,fg = '#B8CF69', font = ('roboto
light',15),textvariable = gun_ammo)
#==========frame3_1_2 widgets
btn_add3_1 = Button(frame3_1_2,text = 'Add', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=30,command = lambda: add_gun(0))
btn_save3_1 = Button(frame3_1_2,text = 'Save', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=25,command = lambda :save_gun(0))
btn_update3_1 = Button(frame3_1_2,text = 'Update', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda:
update_gun(0))
btn_search3_1 = Button(frame3_1_2,text = 'Search', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda
:search_gun(1))
btn_del3_1 = Button(frame3_1_2,text = 'Delete', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,command = lambda :del_gun(0))
btn_see_all3_1 = Button(frame3_1_2,text = 'See all', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda:
see_all_gun(1))
combo_gun = ttk.Combobox(frame3_1_2,font = ('roboto light',13),values =
('Name','Type'),width = 8,state = 'readonly',textvariable=gun_search,postcommand=lambda
:combo_gun_fun(1))
#==========frame3_1_3 widgets
lbl_last_saved = Label(frame3_1_3,textvariable = last_saved, font = ('roboto thin',10),bg
= '#303030',fg='light yellow',anchor = W)
scrol_y3_1 = Scrollbar(frame3_1_3,orient = "vertical")
table3_1 = ttk.Treeview(frame3_1_3,columns =
('name','type','instk','deploy','ammo'),yscrollcommand=scrol_y3_1.set)
scrol_y3_1.config(command = table3_1.yview)
table3_1.heading('name',text = 'Name')
table3_1.heading('type',text = 'Type')
table3_1.heading('instk',text = 'In stock')
table3_1.heading('deploy',text = 'Deployed')
table3_1.heading('ammo',text = 'Ammo')
table3_1.column("#0",width = 0,stretch = 0)
table3_1.column('name',width = 100,anchor = W)
table3_1.column('type',width = 100,anchor = W)
table3_1.column('instk',width = 75,anchor = CENTER)
table3_1.column('deploy',width = 75,anchor = CENTER)
table3_1.column('ammo',width = 75,anchor = CENTER)
table3_1.bind('<<TreeviewSelect>>',show_gun)
if gun_data:
    for g in range(len(gun_data)):
        table3_1.insert(parent='',iid=g,index=END,values=gun_data[g])
#==========frame3_1_4 widgets
img_soldier = Image.open('sprites\\soldier.png')
resi_soldier_img = img_soldier.resize((200,200))
act_solid = ImageTk.PhotoImage(resi_soldier_img)
lbl_no_soldiers = Label(frame3_1_4,image = act_solid,bg =
"#535353",compound='top',textvariable=no_of_soldiers,font = ('roboto thin',15),fg='light
yellow')
img_doctor = Image.open('sprites\\nurse.png')
resi_doctor_img = img_doctor.resize((200,200))
act_doct = ImageTk.PhotoImage(resi_doctor_img)
lbl_no_doctors = Label(frame3_1_4,image = act_doct,bg =
"#535353",compound='top',textvariable=no_of_doctors,font = ('roboto thin',15),fg='light
yellow')
img_worker = Image.open('sprites\\worker.png')
resi_worker_img = img_worker.resize((200,200))
act_work = ImageTk.PhotoImage(resi_worker_img)
lbl_no_workers = Label(frame3_1_4,image = act_work,bg =
"#535353",compound='top',textvariable=no_of_workers,font = ('roboto thin',15),fg='light
yellow')
#==========frame3_2 widgets
frame3_2_1 = Frame(frame3_2,bg = "#252525")
```

```
frame3_2_2 = Frame(frame3_2,bg = "#40433B")
frame3_2_3 = Frame(frame3_2,bg = "#ffffff",bd = 3,relief = RAISED)
frame3_2_4 = Frame(frame3_2,bg = "#535353")
#==========frame3_2_1 widgets
lbl_name3_2_1 = Label(frame3_2_1,text = "Number:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_name3_2_1 = Entry(frame3_2_1,bg = '#353535',width = 20,fg = '#B8CF69', font =
('roboto light',15), textvariable = t_number)
lbl_type3_2_1 = Label(frame3_2_1,text = "Type:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
combo_type3_2_1 = ttk.Combobox(frame3_2_1,width = 20, font = ('roboto light',15), values
= ['Tank', 'Artillery', 'Missile Truck'],state = 'readonly', textvariable = t_type)
lbl_pilot3_2_1 = Label(frame3_2_1,text = "Commander:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
combo_pilot3_2_1 = ttk.Combobox(frame3_2_1,width = 20,font = ('roboto thin',15),values =
([x[0] for x in main_data if x[-2] != 'Doctor' and x[-2] != 'Engineer']+['None']),
textvariable = t_pilot,state = 'readonly')
check_deploy3_2_1 = Checkbutton(frame3_2_1,text = "Deployed", font = ('roboto
thin',20),bg = "#252525",selectcolor='#2f2f2f',fg = '#B8CF69',activebackground="#252525",
variable = t_deploy, onvalue = 'Yes', offvalue = 'No')
lbl_ammo3_2_1 = Label(frame3_2_1,text = "Ammo:", font = ('roboto thin',20),bg =
"#252525",fg = '#B8CF69')
entry_ammo3_2_1 = Entry(frame3_2_1,bg = '#353535',width = 10,fg = '#B8CF69', font =
('roboto light',15), textvariable = t_ammo)
#==========frame3_2_2 widgets
btn_add3_2_2 = Button(frame3_2_2,text = 'Add', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=30,command = lambda: add_gun(1))
btn_save3_2_2 = Button(frame3_2_2,text = 'Save', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=25,command = lambda: save_gun(1))
btn_update3_2_2 = Button(frame3_2_2,text = 'Update', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda:
update_gun(1))
btn_del3_2_2 = Button(frame3_2_2,text = 'Delete', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,command = lambda :del_gun(1))
btn_search3_2_2 = Button(frame3_2_2,text = 'Search', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda:
search_gun(0))
combo_t = ttk.Combobox(frame3_2_2,font = ('roboto light',13),values =
('Number','Type','Commander','Deployed'),width = 12,state = 'readonly',textvariable =
t_search, postcommand = lambda :combo_gun_fun(0))
btn_see_all3_2_2 = Button(frame3_2_2,text = 'See all', font = ('roboto thin',15),bg =
'#40433B',fg='light yellow',bd = 0,padx=20,state = 'disabled',command = lambda:
see_all_gun(0))
#==========frame3_2_3 widgets
scrol_y3_2=Scrollbar(frame3_2_3,orient=VERTICAL)
table3_2 = ttk.Treeview(frame3_2_3,columns =
('number','type','commander','deploy','ammo'),yscrollcommand=scrol_y3_2.set)
scrol_y3_2.config(command = table3_2.yview)
table3_2.heading('number',text = 'Number')
table3_2.heading('type',text = 'Type')
table3_2.heading('commander',text = 'Commander')
table3_2.heading('deploy',text = 'Deployed')
table3_2.heading('ammo',text = 'Ammo')
table3_2.column("#0",width = 0,stretch = 0)
table3_2.column('number',width = 100,anchor = W)
table3_2.column('type',width = 100,anchor = W)
table3_2.column('commander',width = 100,anchor = W)
table3_2.column('deploy',width = 100,anchor = CENTER)
table3_2.column('ammo',width = 100,anchor = CENTER)
table3_2.bind('<<TreeviewSelect>>',show_gun)
if t_data:
    for i in range(len(t_data)):
        table3_2.insert(parent='',index=END,iid=i,values=t_data[i])
```

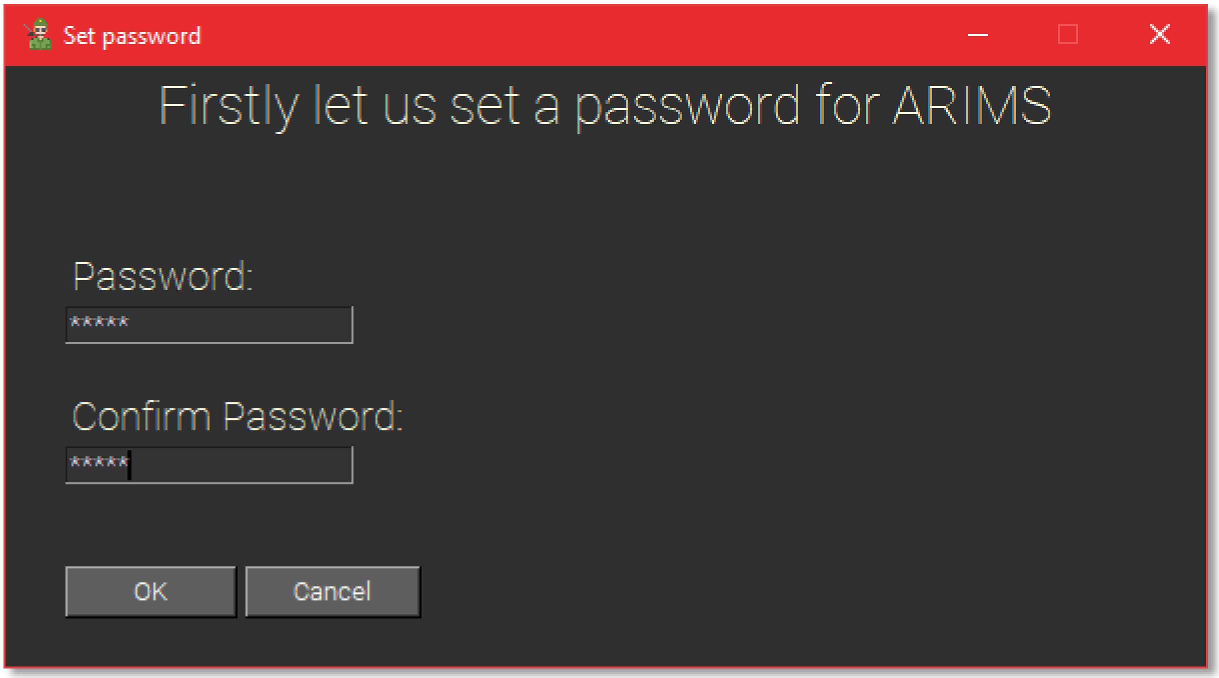
```
lbl_ltd = Label(frame3_2_3,textvariable = lt_saved,font = ('roboto thin',10),bg =
'#303030',fg='light yellow',anchor = W)
#=========frame3_2_4 widgets
img_tank = Image.open('sprites\\tank.png')
re_tank = img_tank.resize((200,200))
act_tank = ImageTk.PhotoImage(re_tank)
lbl_tank = Label(frame3_2_4,image = act_tank,bg = "#535353",compound =
TOP,textvariable=no_tank,font = ('roboto thin',15),fg='light yellow')
img_missiles = Image.open('sprites\\missiles.png')
re_missiles = img_missiles.resize((200,200))
act_missiles = ImageTk.PhotoImage(re_missiles)
lbl_missiles = Label(frame3_2_4,image = act_missiles,bg = "#535353",compound =
TOP,textvariable=no_missile,font = ('roboto thin',15),fg='light yellow')
img_artillery = Image.open('sprites\\artillery.png')
re_artillery = img_artillery.resize((200,200))
act_artillery = ImageTk.PhotoImage(re_artillery)
lbl_artillery = Label(frame3_2_4,image = act_artillery,bg = "#535353",compound =
TOP,textvariable=no_artillery,font = ('roboto thin',15),fg='light yellow')

#=========PASSWORD WINDOW
lbl_wlcm.place(x = 0,y=0,relwidth=1)
lbl_army.place(x = 0,y = 80,relwidth = 1)
lbl_passwd.place(x=100,y=280)
exit_btn.place(x=650,y=0)
entry_password.place(x=300, y=290, width = 200)
submit_btn.pack(side = RIGHT,padx=20)
#=========MAIN WINDOW
#=========frame_1
frame1.place(x=0,y=0,height = 560,width = 1120)
frame1_1.place(x=0,y=0,height = 450 , width = 300)
frame1_2.place(x=0,y=450, height = 110, relwidth = 1)
frame1_3.place(x=300,y=0,height = 450 , width = 820)
#========frame1_1
lbl_name.place(x=25,y=25)
entry_name.place(x=15, y = 65)
lbl_soldierID.place(x=25,y=125)
entry_soldierID.place(x=15,y=165)
check_present.place(x=65,y=225)
lbl_status.place(x=25,y=285)
radio_healthy.place(x=25,y=335)
radio_unhealthy.place(x=25,y=375)
#========frame1_2
btn_pre.grid(row = 1,column = 0,pady = 25,padx = 21)
btn_nxt.grid(row = 1,column = 1,pady = 25)
btn_add.grid(row = 1,column = 2,pady = 25,padx = 21)
btn_update.grid(row = 1,column = 3,pady = 25)
btn_save.grid(row = 1,column = 4,pady = 25,padx = 21)
btn_date.grid(row = 1,column = 5,pady = 25)
btn_search.grid(row = 1,column = 6,pady = 25,padx = 21)
combo_search.grid(row = 1, column = 7)
btn_done.grid(row = 1,column = 8,pady = 25,padx = 21)
#========frame1_3
scrol_y.pack(side = RIGHT,fill = Y)
lbl_date.pack(side = BOTTOM,fill = X)
table.pack(fill = BOTH,expand = 1)
#========frame 2
frame2.place(x=0,y=0,height = 560,width = 1120)
#========frame 2_1
frame2_1.grid(row=0,column=0,padx = 8)
lbl_fname.grid(row = 0,column = 0,padx = 8,pady = 8)
entry_fname.grid(row = 0,column = 1)
lbl_lname.grid(row = 1,column = 0)
entry_lname.grid(row = 1,column = 1,padx = 8,pady = 8)
```

```
frm_dob.grid(row = 2,column = 0,padx = 8,columnspan = 2,pady = 4,sticky = W,rowspan =2)
lbl_ddob.grid(row=0,column=0,padx = 4,sticky = W)
lbl_mdob.grid(row=0,column=1,padx = 4,sticky = W)
lbl_ydob.grid(row=0,column=2,padx = 4,sticky = W)
entry_ddob.grid(row=1,column=0,padx = 4,sticky = W)
entry_mdob.grid(row=1,column=1,padx = 4,sticky = W)
entry_ydob.grid(row=1,column=2,padx = 4,sticky = W)
lbl_faname.grid(row = 0,column = 2,padx = 8,pady = 8,sticky = W)
entry_faname.grid(row = 0,column = 3,pady = 8)
lbl_moname.grid(row = 1,column = 2,padx = 8,sticky = W)
entry_moname.grid(row = 1,column = 3,padx = 8,sticky = W)
lbl_height.grid(row = 2,column = 2,padx = 8,sticky = W)
entry_height.grid(row = 2,column = 3,padx = 8,sticky = W)
lbl_weight.grid(row = 3,column = 2,padx = 8,sticky = W)
entry_weight.grid(row = 3,column = 3,padx = 8,sticky = W)
frm_gender.grid(row = 4,column =0 ,padx = 8,pady = 8,sticky = W,columnspan = 2)
radio_male.grid(row = 5,column =0 ,padx = 8,sticky = W)
radio_female.grid(row = 5,column =1 ,padx = 8,sticky = W)
lbl_blood.grid(row = 4,column =2,padx = 8,sticky = W)
combo_blood.grid(row = 4,column =3,sticky = W)
lbl_religion.grid(row = 5,column =0,padx = 8)
combo_religion.grid(row = 5,column =1,padx = 4,sticky = W)
lbl_home.grid(row = 5,column =2,padx = 8,sticky = W)
entry_home.grid(row = 5,column = 3,padx = 8,sticky = W,pady = 10)
#========frame 2_2
frame2_2.grid(row =0,column = 1,sticky = NW,padx = 8)
lbl_solderid.grid(row = 0,column = 0,padx = 8,pady = 8)
entry_solderid.grid(row = 0,column = 1,padx = 8,pady = 8)
frm_doj.grid(row = 1,column=0,columnspan=2,pady = 8)
lbl_ddoj.grid(row=0,column=0,padx = 4,sticky = W)
lbl_mdoj.grid(row=0,column=1,padx = 4,sticky = W)
lbl_ydoj.grid(row=0,column=2,padx = 4,sticky = W)
entry_ddoj.grid(row=1,column=0,padx = 4,sticky = W)
entry_mdoj.grid(row=1,column=1,padx = 4,sticky = W)
entry_ydoj.grid(row=1,column=2,padx = 4,sticky = W)
lbl_Lcity.grid(row = 2,column = 0,padx = 8,pady = 8)
entry_Lcity.grid(row = 2,column = 1,padx = 8,pady = 8)
lbl_Was.grid(row = 3,column = 0,padx = 8,pady = 8)
entry_Was.grid(row = 3,column = 1,padx = 8,pady = 8)
lbl_rank.grid(row = 4,column = 0,padx = 8,pady = 8)
entry_rank.grid(row = 4,column = 1,padx = 8,pady = 23)
#========frame2_3
frame2_3.place(x=8,y=375,height = 160,width = 1070)
btn_prev.grid(row = 0,column = 0,padx = 8,pady = 25)
btn_nxt1.grid(row = 0,column = 1,padx = 8,pady = 8)
btn_new.grid(row = 0,column = 2,padx = 8,pady = 8)
btn_add1.grid(row = 0,column = 3,padx = 8,pady = 8)
btn_save1.grid(row = 0,column = 4,padx = 8,pady = 8)
btn_update1.grid(row = 0,column = 5,padx = 8,pady = 8)
btn_del.grid(row = 0,column = 6,padx = 8,pady = 8)
#=======frame3_1
frame3_1.place(x=0,y=0,height = 560,width = 1120)
frame3_1_1.place(x=0,y=0,height = 560, width = 300)
frame3_1_2.place(x=300,y=0, height = 50, width = 820)
frame3_1_3.place(x=300,y=50,height = 255, width = 820)
frame3_1_4.place(x=300,y=305,height = 255, width = 820)
#=======frame3_1_1
lbl_name3_1.place(x=25,y=25)
entry_name3_1.place(x=15, y = 65)
lbl_type.place(x=25,y=125)
entry_type.place(x=15,y=165)
lbl_instk.place(x=25,y=225)
entry_instk.place(x=15, y=265)
```

```
lbl_deploy.place(x=25,y=325)
entry_deploy.place(x=15,y=365)
lbl_ammo.place(x=25,y=425)
entry_ammo.place(x=15,y=465)
#=======frame3_1_2
btn_add3_1.grid(row = 0,column = 2,padx = 1,pady = 5)
btn_save3_1.grid(row = 0,column = 3,padx = 1,pady = 5)
btn_update3_1.grid(row = 0,column = 4,padx = 1,pady =5)
btn_search3_1.grid(row = 0,column = 5,padx = 1,pady = 5)
btn_del3_1.grid(row = 0,column = 6,padx = 1,pady = 5)
btn_see_all3_1.grid(row = 0,column = 7,padx = 1,pady =5)
combo_gun.grid(row = 0,column = 8,padx = 4,pady =5)
#=======frame3_1_3
lbl_last_saved.pack(side = "bottom",fill = X)
scrol_y3_1.pack(side = RIGHT,fill = Y)
table3_1.pack(fill = BOTH,expand = 1)
#=======frame3_1_4
lbl_no_soldiers.grid(row = 0,column=1,padx=40,pady=10)
lbl_no_doctors.grid(row = 0,column=2,padx=20,pady=10)
lbl_no_workers.grid(row = 0,column=3,padx=20,pady=10)
#=======frame3_2
frame3_2.place(x=0,y=0,height = 560,width = 1120)
frame3_2_1.place(x=0,y=0,height = 560 , width = 300)
frame3_2_2.place(x=300,y=0, height = 50, width = 820)
frame3_2_3.place(x=300,y=50,height = 255 , width = 820)
frame3_2_4.place(x=300,y=305,height = 255 , width = 820)
#=======frame3_2_1
lbl_name3_2_1.place(x=25,y=25)
entry_name3_2_1.place(x=15, y = 65)
lbl_type3_2_1.place(x=25,y=125)
combo_type3_2_1.place(x=15,y=165)
lbl_pilot3_2_1.place(x=25,y=225)
combo_pilot3_2_1.place(x=15,y=265)
check_deploy3_2_1.place(x=25,y=345)
lbl_ammo3_2_1.place(x=25,y=425)
entry_ammo3_2_1.place(x=15,y=465)
#=======frame3_2_2
btn_add3_2_2.grid(row = 0,column = 2,padx = 1,pady = 5)
btn_save3_2_2.grid(row = 0,column = 3,padx = 1,pady = 5)
btn_update3_2_2.grid(row = 0,column = 4,padx = 1,pady =5)
btn_del3_2_2.grid(row = 0,column = 5,padx = 1,pady = 5)
btn_search3_2_2.grid(row = 0,column = 6,padx = 1,pady = 5)
combo_t.grid(row = 0,column = 7,padx = 7,pady =5)
btn_see_all3_2_2.grid(row = 0,column = 8,padx = 1,pady =5)
#=======frame3_2_3
lbl_ltd.pack(fill=X,side=BOTTOM)
scrol_y3_2.pack(side = RIGHT,fill = Y)
table3_2.pack(fill = BOTH, expand=1)
#=======frame3_2_4
lbl_tank.grid(row = 0,column = 0,padx=40,pady=10)
lbl_artillery.grid(row = 0,column = 1,padx=20,pady=10)
lbl_missiles.grid(row = 0,column = 2,padx=20,pady=10)
#=======password window settings
password_window.resizable(False,False)
password_window.geometry('720x520+300+50')
password_window.config(bg='#2f2f2f',pady = 20)
password_window.bind('<KeyPress-Escape>',ex)
password_window.protocol('WM_DELETE_WINDOW',ex)
#=======main window settings
main_window.resizable(False,False)
main_window.config(bg='#2f2f2f',menu = main_menu)
main_window.geometry('1120x560+50+50')
main_window.withdraw()
main_window.bind('<KeyPress-Escape>',ex)
main_window.protocol('WM_DELETE_WINDOW',ex)
#========main loop
password_window.mainloop()
main_window.mainloop()
```

Output



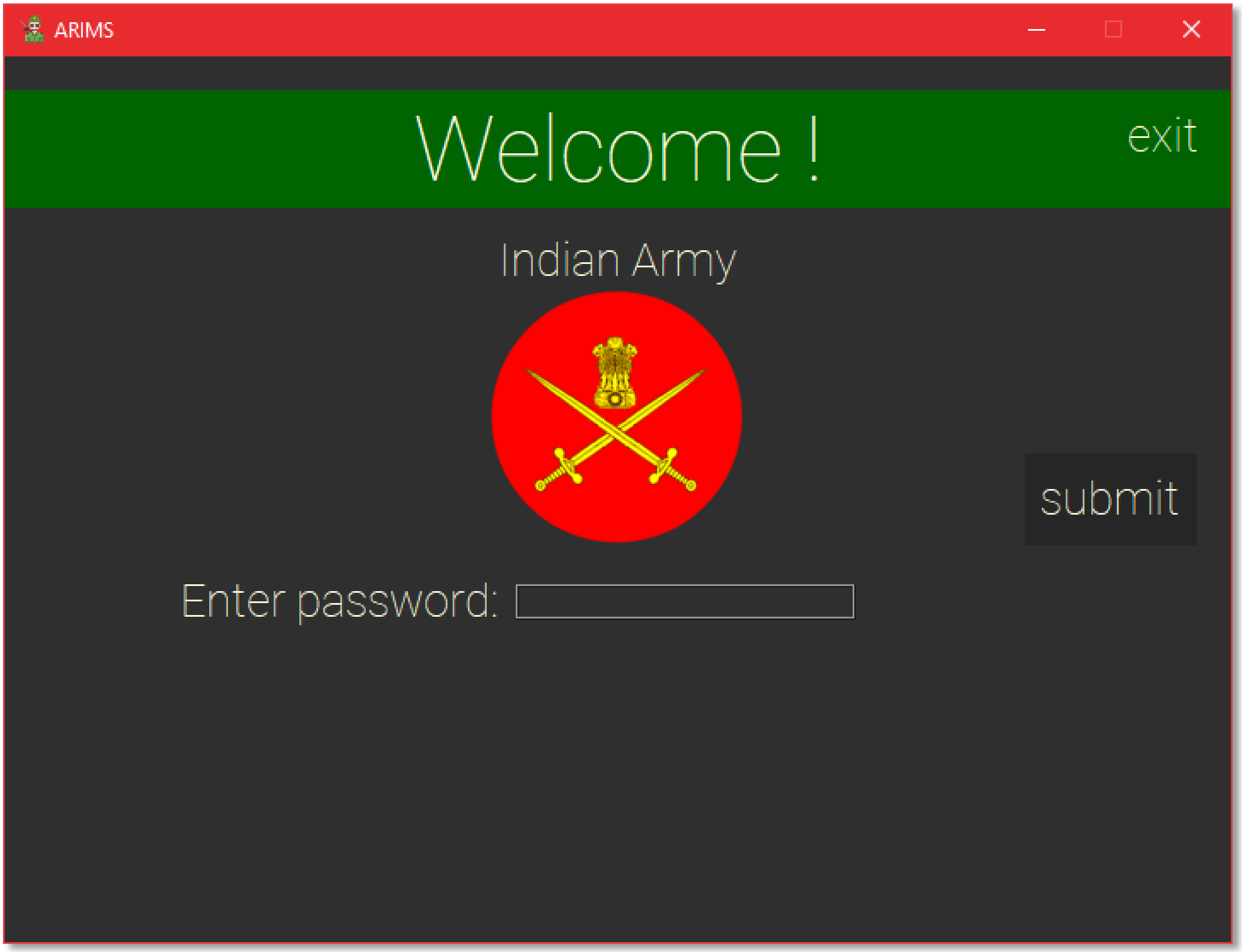
Firstly let us set a password for ARIMS

Password:
*****

Confirm Password:
*****

Set up window

To set password

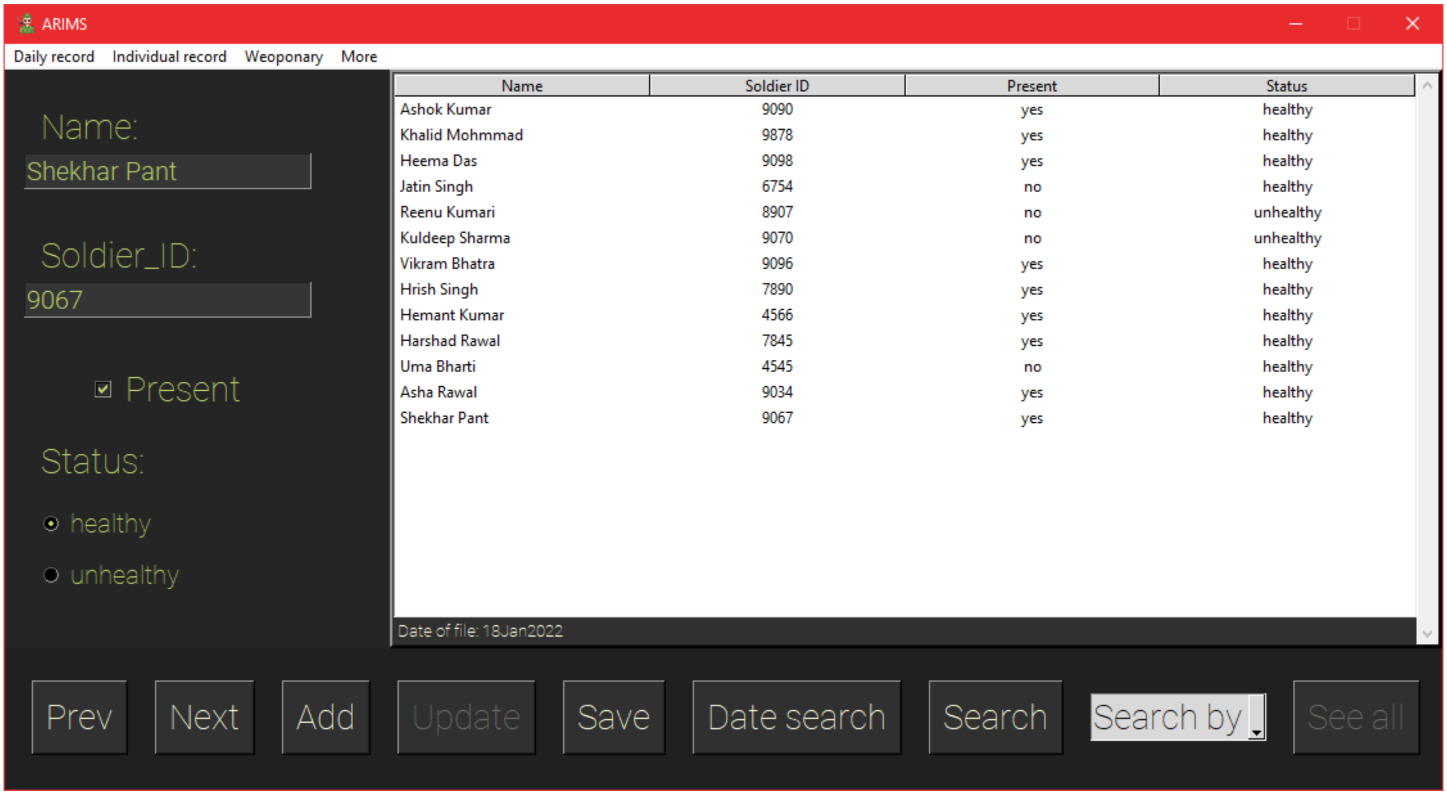


Welcome !

Indian Army

Enter password:

Log in window: for password



Name: Shekhar Pant

Soldier_ID: 9067

☑ Present

Status:

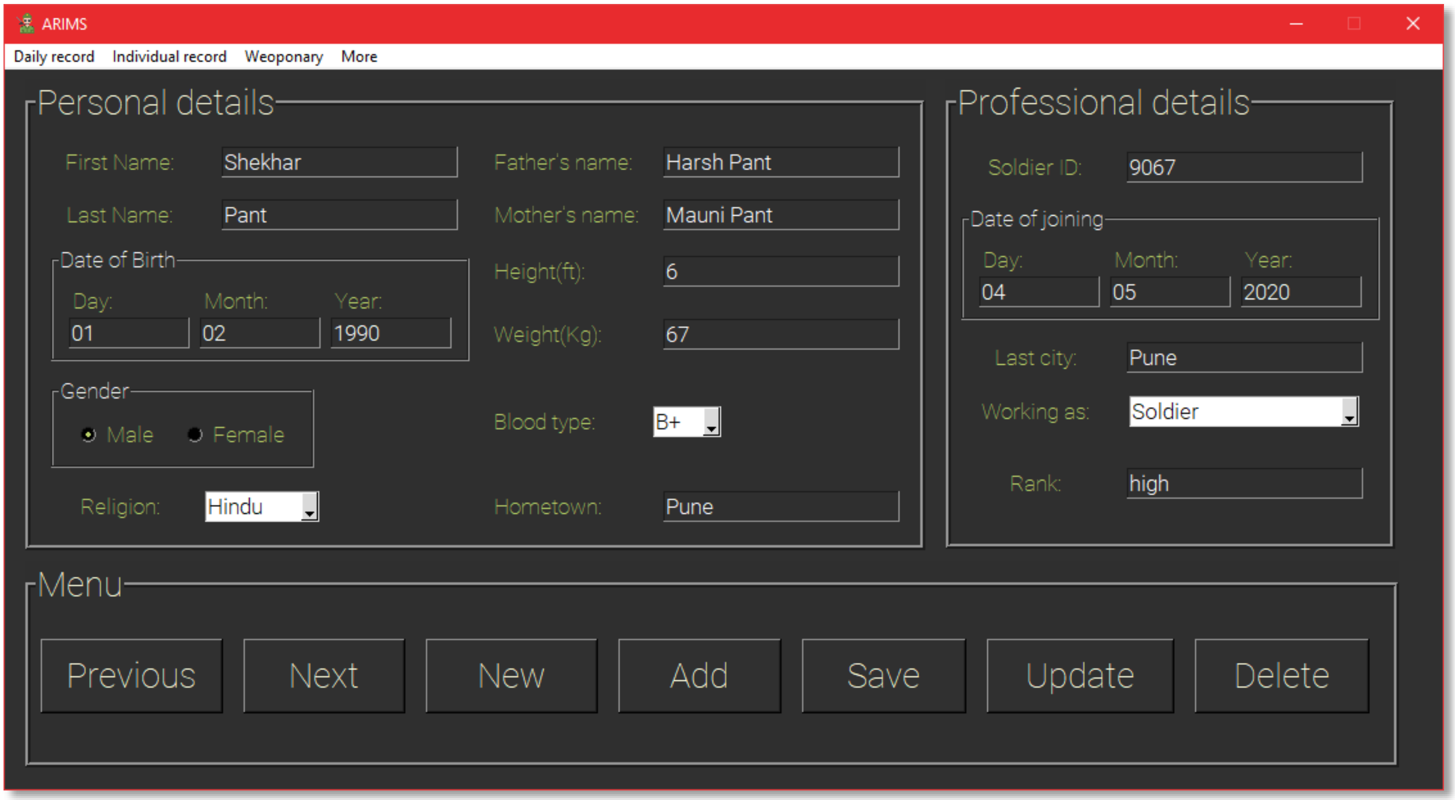
● healthy

○ unhealthy

Name	Soldier ID	Present	Status
Ashok Kumar	9090	yes	healthy
Khalid Mohmmad	9878	yes	healthy
Heema Das	9098	yes	healthy
Jatin Singh	6754	no	healthy
Reenu Kumari	8907	no	unhealthy
Kuldeep Sharma	9070	no	unhealthy
Vikram Bhatra	9096	yes	healthy
Hrish Singh	7890	yes	healthy
Hemant Kumar	4566	yes	healthy
Harshad Rawal	7845	yes	healthy
Uma Bharti	4545	no	healthy
Asha Rawal	9034	yes	healthy
Shekhar Pant	9067	yes	healthy

Date of file: 18Jan2022

Daily record window: for managing daily records of soldiers.



Personal details

First Name: Shekhar

Last Name: Pant

Date of Birth: Day: 01 Month: 02 Year: 1990

Gender: ● Male ○ Female

Religion: Hindu

Father's name: Harsh Pant

Mother's name: Mauni Pant

Height(ft): 6

Weight(Kg): 67

Blood type: B+

Hometown: Pune

Professional details

Soldier ID: 9067

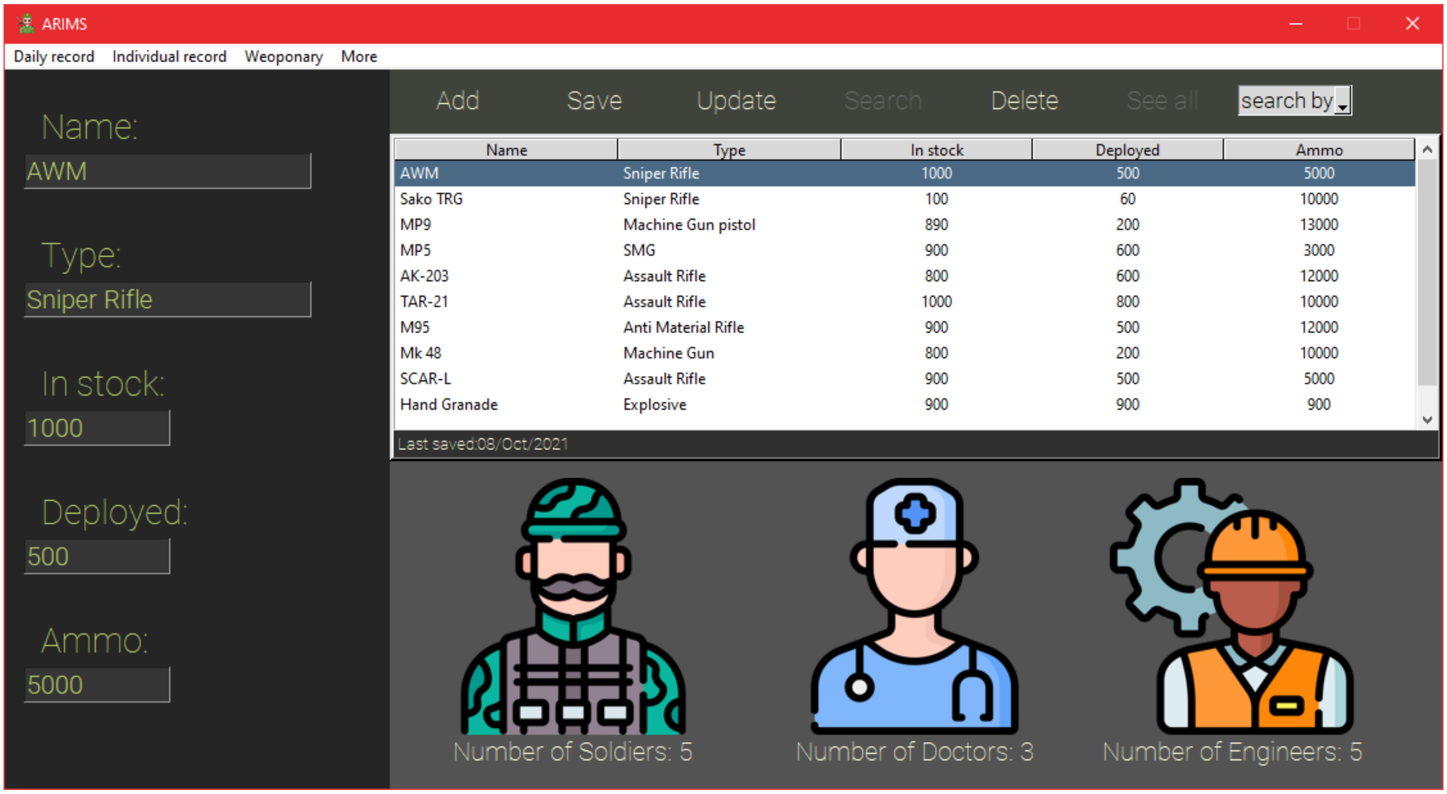
Date of joining: Day: 04 Month: 05 Year: 2020

Last city: Pune

Working as: Soldier

Rank: high

Individual record window: for managing individual records of soldiers.



Name:
AWM

Type:
Sniper Rifle

In stock:
1000

Deployed:
500

Ammo:
5000

Name	Type	In stock	Deployed	Ammo
AWM	Sniper Rifle	1000	500	5000
Sako TRG	Sniper Rifle	100	60	10000
MP9	Machine Gun pistol	890	200	13000
MP5	SMG	900	600	3000
AK-203	Assault Rifle	800	600	12000
TAR-21	Assault Rifle	1000	800	10000
M95	Anti Material Rifle	900	500	12000
Mk 48	Machine Gun	800	200	10000
SCAR-L	Assault Rifle	900	500	5000
Hand Granade	Explosive	900	900	900

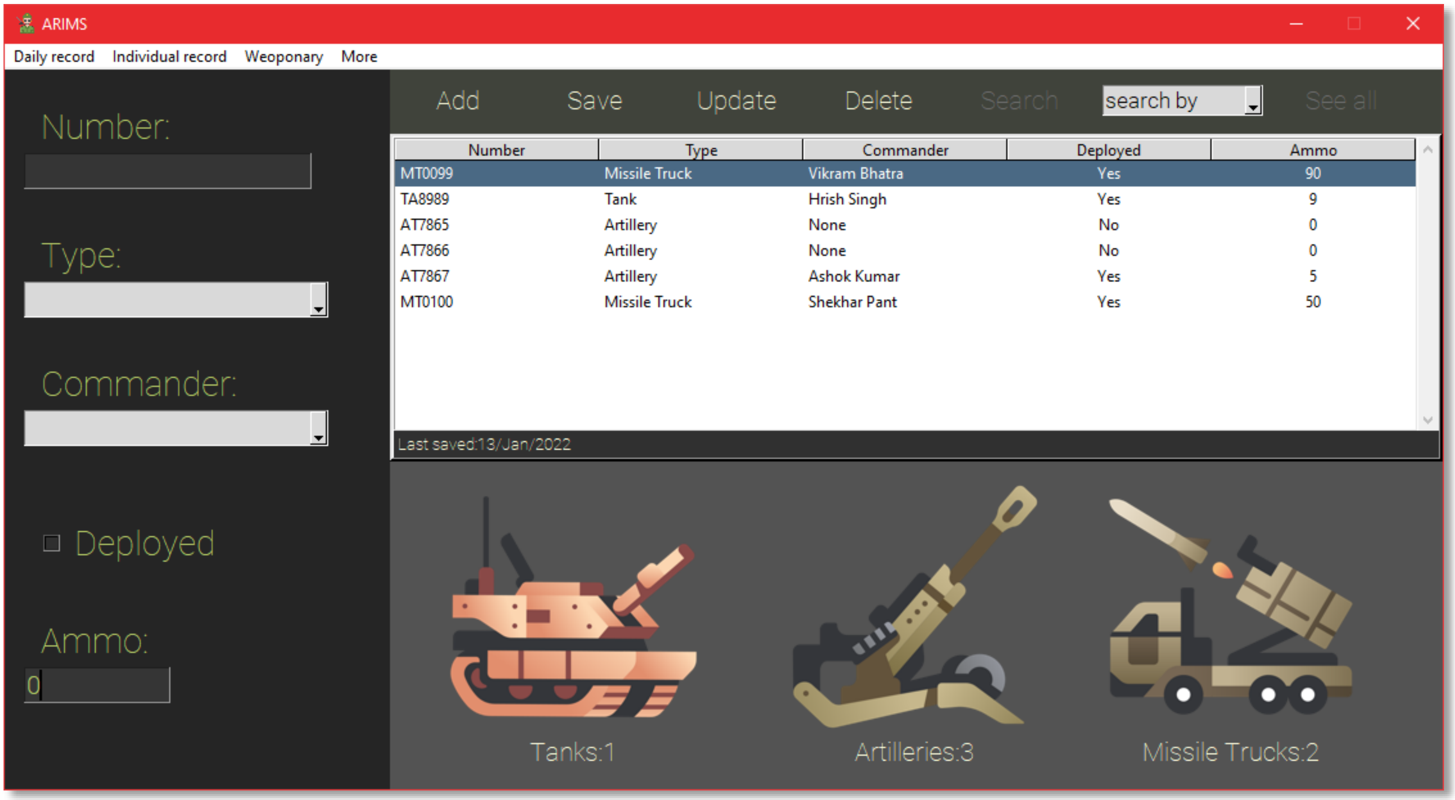
Last saved:08/Oct/2021

Number of Soldiers: 5

Number of Doctors: 3

Number of Engineers: 5

Weoponary record window: for managing weoponary



Number:

Type:

Commander:

□ Deployed

Ammo:
0

Number	Type	Commander	Deployed	Ammo
MT0099	Missile Truck	Vikram Bhatra	Yes	90
TA8989	Tank	Hrish Singh	Yes	9
AT7865	Artillery	None	No	0
AT7866	Artillery	None	No	0
AT7867	Artillery	Ashok Kumar	Yes	5
MT0100	Missile Truck	Shekhar Pant	Yes	50

Last saved:13/Jan/2022

Tanks:1

Artilleries:3

Missile Trucks:2

Transport management window: for managing vehicles.

BIBLIOGRAPHY

Sources from internet

- Stackoverflow.com
- Codemy.com

Books

- Computer Science with Python (Preeti Arora)