**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**LAB REPORT**

(ACADEMIC YEAR 2021-22)

COURSE NAME: Python Programming

COURSE CODE:

DEPARTMENT: CSE

FACULTY NAME: Dr. Soni Sweta

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**LAB REPORT**

(ACADEMIC YEAR 2021-22)

**SUBMITTED BY**

STUDENT NAME: Shubham Pawar

ENROLLMENT NUMBER: A70405220148

CLASS: Div A

SEMESTER: 4

DATE OF SUBMISSION:

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**CERTIFICATE OF SUBMISSION**

Student Name: Shubham Balaso Pawar

Class: Div A Semester: 4

Enrolment Number: A70405220148

This is certified to be the bonafide work of student in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Laboratory during the academic year 2021-22.

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

**Department Coordinate HoI**

{Department of \_\_\_\_\_\_\_\_} ASET & AIT, AUM

ASET, AUM

**Date:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Stamp**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Description | Date | Page No. | Grade |
| 1 | Write a Python script to check whether a given key already exists in a dictionary | 10/02 |  |  |
| 2 | Write a Python program to get the key, value, and item in a dictionary | 17/02 |  |  |
| 3 | Write a Python program to combine two lists into a dictionary | 24/02 |  |  |
| 4 | Create a list and tuple and apply all the operations on them | 03/03 |  |  |
| 5 | Create a function with variable length of arguments | 03/03 |  |  |
| 6 | Create a function which return multiple values from a function | 10/03 |  |  |
| 7 | Create a function with positional and default argument | 10/03 |  |  |
| 8 | Write a python program to handle multiple error with one except statement. | 31/03 |  |  |
| 9 | Write a python program to depict else clause with try-except | 31/03 |  |  |
| 10 | Write a Python program to read last n lines of a file. | 07/04 |  |  |
| 11 | Write a Python program to copy the contents of a file to another file | 07/04 |  |  |
| 12 | Creation of array using Numpy module and perform many operations on it. | 21/04 |  |  |
| 13 | Creation of series using Pandas module and perform many operations on it | 21/04 |  |  |
| 14 | Creation of ND array using Pandas module and perform many data manipulation operations on it using IRIS dataset. | 28/04 |  |  |
| 15 | Create any two GUI applications using Tkinter | 05/05 |  |  |

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 1**

**Write a Python script to check whether a given key already exists in a dictionary**

Student Name: Shubham Balaso Pawar

Class: Div A Semester: 4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

def is\_key\_present(x):

if x in d:

print('Key is present in the dictionary')

else:

print('Key is not present in the dictionary')

is\_key\_present(int(input("Enter the key to check :")))

Graphical user interface, application

Description automatically generated

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 2**

Write a Python program to get the key, value, and item in a dictionary

Student Name: Shubham Balaso Pawar

Class: Div A Semester: 4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

def dictKey(test\_dict):

# Printing dictionary

print ("Original dictionary is : " + str(test\_dict))

# using in operator to

# get key and value

print ("Dict key-value are : ")

for i in test\_dict :

print(i, test\_dict[i])

test\_dict = {"Shubham":98,"Anurag":97,"Abhishek":99}

dictKey(test\_dict)

Text

Description automatically generated with medium confidence

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 3**

Write a Python program to combine two lists into a dictionary

Student Name: Shubham Balaso Pawar

Class: Div A Semester: 4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

# here keys list goes as the keys to the dictionary and

# values list goes as the values to the keys

keys = ['Shubham', 'Anurag', 'Abhishek']

values = [98,97,99]

dictionary = dict(zip(keys, values))

print(dictionary)

**Text

Description automatically generated with medium confidence**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 4**

Create a list and tuple and apply all the operations on them

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

## list

print("#####List Operations#####")

# list with mixed datatypes

li = ["Ab","CD",89,55,12]

#accessing the elements

print(li[1])

print(li[-1]) # negative indexing

# slicing

print(li[1:4])

print(li[::-1])

# changing values

li[1]="Neel"

print(li)

## Tuple

print("#####Tuple Operations#####")

# tuple with mixed datatypes

my\_tuple = (1, "Hello", 3.4)

print(my\_tuple)

print(type(my\_tuple))

# Creating a tuple having one element

tup = "hello",

print(type(tup))

# Accessing the elements in tuple

print(my\_tuple[1])

print(my\_tuple[-1])

# Accessing tuple elements using slicing

my\_tup = ('p','r','o','g','r','a','m','i','z')

print(my\_tup[1:5])

print(my\_tup[:6])

# deleting a tuple

del(my\_tup)

try:

print(my\_tup)

except NameError:

print("Tuple Not Found")

Text, letter

Description automatically generated

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 5**

Create a function with variable length of arguments

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

def student(firstname, lastname ='Sharma', standard ='10th'):

print(firstname, lastname, 'studies in', standard, 'Standard')

# 1 positional argument

student('Rohit')

# 2 positional arguments

student('Rohit', 'Singh')

student('Rohit', '7th')

# 3 positional arguments

student('Rohit', 'Singh', '7th')

Text

Description automatically generated

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 6**

**Create a function which return multiple values from a function**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

# We can return the multiple values by separating be commas #and is stored as tuple

def returnMultiple():

return "Shubham","Abhishek"

result = returnMultiple()

print(result)

print(type(result))

A picture containing graphical user interface

Description automatically generated

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 7**

**Create a function with positional and default argument**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

# default arguments

print("Default argument ==================")

def greet(name, msg="Good morning!"):

# if msg provided then prints that

# else prints the default msg

print("Hello", name + ', ' + msg)

greet("Shubham")

greet("Abhishek","Have a nice day")

# positional Argument

print("position argument ==================")

def student(firstname, lastname ='Sharma', standard ='10th'):

print(firstname, lastname, 'studies in', standard, 'Standard')

# 1 positional argument

student('Rohit')

# 2 positional arguments

student('Rohit', 'Singh')

student('Rohit', '7th')

# 3 positional arguments

student('Rohit', 'Singh', '7th')

**Text, letter

Description automatically generated**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 8**

Write a python program to handle multiple error with one except statement.

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

try:

# print(3/0)

name = 'SHubham'

name += 5

except (TypeError , ZeroDivisionError, NameError) as error:

print(error)



**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 9**

Write a python program to depict else clause with try-except

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

def divide(x, y):

'''This code is to run else clause'''

try:

result = x / y

except ZeroDivisionError:

print("Cannot divide by zero ")

else:

print("Your answer is :", result)

# Look at parameters and note the working of Program

divide(3, 0)

divide(3, 2)

****

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 10**

**Write a Python program to read last n lines of a file.**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

def read\_lastnlines(fname,n):

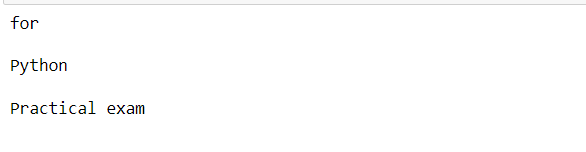
'''Reading last n lines of file'''

with open('D:\\read.txt') as f:

for line in (f.readlines() [-n:]):

print(line)

read\_lastnlines('states.txt',3)

****

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 11**

**Write a Python program to copy the contents of a file to another file**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

with open('D:\\read.txt','r') as firstfile, open('D:\\two.txt','w') as secondfile:

for line in firstfile:

secondfile.write(line)

**Text

Description automatically generated**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 12**

**Creation of array using Numpy module and perform many operations on it.**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

import numpy as np

# Create 1D Array

a = np.array([1,2,3])

print(a)

# dimension

a.ndim

# Create 2D Array [3x3]

a = np.array([[1,2,3],[4,5,6],[7,8,9]])

print(a)

# returns the shape of the array

a.shape

np.eye(3,4)

np.diag([1,2,3,4])

np.diag(a)

a = np.array([[1,2],[3,4],[5,6]])

# Print the sum of all hte elements in the array

a.sum()

**A picture containing shape

Description automatically generated**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 13**

**Creation of series using Pandas module and perform many operations on it**

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

import pandas as pd

#Creating Series

s1 = pd.Series([1,2,3,4,5])

s1

# Create a dictionary

dic = pd.Series([1,2,3,4,5], index=['a','b','c','d','e'])

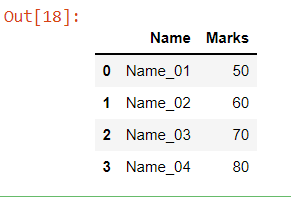
dic

# Create a dataframe

df = pd.DataFrame({"Name":['Name\_01', 'Name\_02', 'Name\_03', 'Name\_04'],

"Marks":[50, 60, 70, 80]})

df

****

df.head(1)

Graphical user interface, application

Description automatically generated

df.tail(1)

Logo

Description automatically generated with medium confidence

df.shape



df.describe()

Graphical user interface, table

Description automatically generated with medium confidence

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 14**

Creation of ND array using Pandas module and perform many data manipulation operations on it using IRIS dataset.

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

s2 = pd.read\_csv('IRIS.csv')

s2.iloc[0:3,0:2]

s2.iloc[30:40,3:]

s2['petal\_width']

s2.loc[0:30, ('petal\_width')]

d1 = s2.drop(['species'], axis=1)

d1

d2 = s2.drop([1,2,3], axis=0)

d2

**A screenshot of a computer screen

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with low confidence**

**Text

Description automatically generated**

**A picture containing text

Description automatically generated**

**A screen shot of a computer

Description automatically generated with low confidence**

**A screen shot of a computer

Description automatically generated with low confidence**

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

A picture containing text, clipart

Description automatically generated

**(Academic Year 2021-22)**

**LAB 15**

Create any two GUI applications using Tkinter

Student Name: Shubham Balaso Pawar

Class: Div A Semester:4

Enrolment Number: A70405220148

**Faculty In-charge**

{Department of \_\_\_\_\_\_\_\_}

ASET, AUM

from tkinter import \*

import mysql.connector

# from tkinter import messagebox

con = mysql.connector.connect(host='localhost', user='root', password='Sage', database='tkinter')

cur = con.cursor(buffered=True)

form = Tk()

# Window configuration

form.title("Registration")

form.geometry("700x700")

#functions

def close():

form.destroy()

def insert(fn,ln,ag,pn,em,ur,pa):

cur.execute('insert into registration values ("{}","{}",{},{},"{}","{}","{}")'.format(fn,ln,ag,pn,em,ur,pa))

print("Insert succcessful")

con.commit()

def clear():

print("text fields to clear")

# Clear the field

fname\_entry.delete(0,END)

lname\_entry.delete(0,END)

age\_entry.delete(0,END)

phone\_entry.delete(0,END)

email\_entry.delete(0,END)

username\_entry.delete(0,END)

password\_entry.delete(0,END)

output\_label.config(text="Fields cleared",fg="green")

def addUser():

firstName = fname.get()

lastName = lname.get()

userAge = age.get()

phoneNumber = phone.get()

emailid = email.get()

username = user.get()

password = passw.get()

# print("Firstname :",firstName)

# print("LAst Name :",lastName)

# print("Age :",userAge)

# print("Phone : ",phoneNumber)

# print("Email :",emailid)

# print("Username :",username)

# print("Password :",password)

print("to be checked ......")

# Checking for the input values

if firstName=="" or lastName=="" or userAge=="" or phoneNumber=="" or emailid=="" or username=="" or password=="" :

output\_label.config(text="Missing values",fg="red")

print("Missing values , Renter the values")

return

elif len(userAge) != 2:

output\_label.config(text="Invalid Age",fg="red")

age\_entry.delete(0, END)

print("Invalid Age")

return

elif len(phoneNumber) != 10:

output\_label.config(text="Phone Number invalid",fg="red")

phone\_entry.delete(0, END)

print("Invalid Phone Number")

return

elif checkUser(username):

output\_label.config(text="Username already taken \nplease Enter another one",fg="red")

username\_entry.delete(0,END)

print("Duplicate value for username found")

else:

print("values correct \n\nInserting values")

insert(firstName,lastName,int(userAge),phoneNumber,emailid,username,password)

clear()

output\_label.config(text="Register successful")

def checkUser(username):

'''

:param username:

:return:

true if duplicate value is found

false if duplicate value is not found and is ready to enter

'''

cur.execute('select \* from registration;')

for i in cur:

if i[5] == username:

return True

return False

#functions end

#

frame1 = Frame(form)

frame1.place(x=0,y=5,width=700)

main\_label = Label(frame1,text="Registration Form",font=('arial',18)).pack()

# frame 2 for main box

frame2 = Frame(form,width=600,height=600,highlightbackground="black", highlightthickness=10)

frame2.place(x=50,y=50)

# frame 2 widgets

#first name

fname\_label = Label(frame2,text="First Name", font=('Arial', 16))

fname\_label.place(x=70,y=20)

fname = StringVar()

fname\_entry = Entry(frame2,textvariable=fname,font=('Arial', 16))

fname\_entry.place(x=270,y=20)

#last Name

lname\_label = Label(frame2,text="Last Name", font=('Arial', 16))

lname\_label.place(x=70,y=80)

lname = StringVar()

lname\_entry = Entry(frame2,textvariable=lname,font=('Arial', 16))

lname\_entry.place(x=270,y=80)

#age

age\_label = Label(frame2,text="Age", font=('Arial', 16))

age\_label.place(x=70,y=140)

age = StringVar()

age\_entry = Entry(frame2,textvariable=age,font=('Arial', 16))

age\_entry.place(x=270,y=140)

#phone

phone\_label =Label(frame2,text="Phone Number", font=('Arial', 16))

phone\_label.place(x=70,y=200)

phone = StringVar()

phone\_entry = Entry(frame2,textvariable=phone,font=('Arial', 16))

phone\_entry.place(x=270,y=200)

#email

email\_label = Label(frame2,text="Email", font=('Arial', 16))

email\_label.place(x=70,y=260)

email = StringVar()

email\_entry = Entry(frame2,textvariable=email,font=('Arial', 16))

email\_entry.place(x=270,y=260)

#username

user\_label = Label(frame2,text="Username", font=('Arial', 16))

user\_label.place(x=70,y=320)

user = StringVar()

username\_entry = Entry(frame2,textvariable=user,font=('Arial', 16))

username\_entry.place(x=270,y=320)

#password

pass\_label = Label(frame2,text="Password", font=('Arial', 16))

pass\_label.place(x=70,y=380)

passw = StringVar()

password\_entry = Entry(frame2,textvariable=passw,font=('Arial', 16),show="x")

password\_entry.place(x=270,y=380)

#Button

register\_button = Button(frame2,text="Register",width=20,height=2,command=addUser,bg="green")

register\_button.place(x=200,y=440)

close\_button = Button(frame2,text="Close",width=20,height=2,command=close,bg="red")

close\_button.place(x=400,y=440)

clear\_button = Button(frame2,text="Clear",width=15,height=2,command=clear,bg="blue")

clear\_button.place(x=450,y=520)

# output label

output\_label = Label(frame2,text="")

output\_label.place(x=70,y=520)

form.mainloop()

Graphical user interface

Description automatically generated

from tkinter import \*

import mysql.connector

#Functions

def submit():

fname = fname\_f.get()

lname=lname\_f.get()

mname = mname\_f.get()

email=email\_f.get()

dob=dob\_f.get()

phone=phone\_f.get()

if acceptance(fname,mname,lname,email,dob,phone):

if addToCSV(fname,mname,lname,email,phone,dob):

out = "Name - {} {} {} \nEmailId - {}\nPhone Number - {}\t\tDOB - {}".format(fname,mname,lname,email,phone,dob)

print(out)

st = "Student Details - \n"

final\_label.config(text=st+out,fg="white")

else:

final\_label.config("Error try after some time")

return True

def addToCSV(fname,mname,lname,email,phone,dob):

f = open("D:\\shubham\\Academic\\2 SY\\4rd-sem\\Subjects\\PP\\PP-lab\\file\_handling\\student.csv", "a")

f.write("\n,{},{},{},{},{},{}".format(fname,mname,lname,email,phone,dob))

f.close()

return True

def addToBase(fname,mname,lname,email,phone,dob):

con = mysql.connector.connect(host='localhost', user='root', password='Sage', database='tkinter')

cur = con.cursor(buffered=True)

cur.execute('insert into admission (stfname,stmname,stlname,stemail,stnumber,stdob)values ("{}","{}","{}","{}","{}","{}")'.format(fname,mname,lname,email,phone,dob))

con.commit()

print("Added successful")

return True

def acceptance(fn,mn,ln,email,db,pn):

'''

:param fn: first name

:param mn: middle name

:param ln: last name

:param email: email ID

:param db: date of birth

:param pn: phone number

:return:

True if everything passes the basic condition

False if does not pass the given condition

'''

if fn=="" or mn=="" or email=="" or ln =="" or db=="" or pn=="":

print("Cannot be empty")

final\_label.config(text="No field can be empty",fg="red")

return False

elif len(pn)!=10:

print("Wrong Phone number")

final\_label.config(text="Phone number invalid",fg="red")

return False

elif int(db[0:4])>2022 or int(db[4:6])>12 or int(db[6:])>31:

print("Error in Date of birth")

final\_label.config(text="Error in date of birth",fg="red")

return False

return True

def clear():

'''

:return: true after clearing the entry fields in the form

'''

fname\_entry.delete(0,END)

mname\_entry.delete(0,END)

lname\_entry.delete(0,END)

email\_entry.delete(0,END)

phone\_entry.delete(0,END)

dob\_entry.delete(0,END)

final\_label.config(text="")

return True

# functions end

form = Tk()

form.geometry('1200x600')

form.title("Admission Form")

form.config(bg="grey")

frame1 = Frame(form,bg="pink",width=1200,height=50).place(x=0,y=10)

main\_label = Label(frame1,text="Basic Information of student",bg="pink",font=("arial",16))

main\_label.place(x=400,y=16)

frame2 = Frame(form,bg="light yellow",width=1100,height=400,highlightbackground="black", highlightthickness=5).place(x=50,y=80)

# Frame 2 widgets

fname\_label = Label(frame2,text="First name",bg="light yellow",font=("arial",15))

fname\_label.place(x=70,y=100)

fname\_f = StringVar()

fname\_entry = Entry(frame2,textvariable=fname\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

fname\_entry.place(x=230,y=100)

mname\_label=Label(frame2,text="Middle name",bg="light yellow",font=("arial",15))

mname\_label.place(x=70,y=170)

mname\_f = StringVar()

mname\_entry = Entry(frame2,textvariable=mname\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

mname\_entry.place(x=230,y=170)

lname\_label = Label(frame2,text="Last name",bg="light yellow",font=("arial",15))

lname\_label.place(x=70,y=240)

lname\_f = StringVar()

lname\_entry = Entry(frame2,textvariable=lname\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

lname\_entry.place(x=230,y=240)

email\_label = Label(frame2,text="EmailId",bg="light yellow",font=("arial",15))

email\_label.place(x=620,y=100)

email\_f = StringVar()

email\_entry = Entry(frame2,textvariable=email\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

email\_entry.place(x=820,y=100)

phone\_label = Label(frame2,text="Phone Number",bg="light yellow",font=("arial",15))

phone\_label.place(x=620,y=170)

phone\_f = StringVar()

phone\_entry = Entry(frame2,textvariable=phone\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

phone\_entry.place(x=820,y=170)

dob\_label = Label(frame2,text="Date Of Birth",bg="light yellow",font=("arial",15))

dob\_label.place(x=620,y=240)

dob\_format\_label = Label(frame2,text="YYYYMMDD",bg="light yellow",fg="red",font=("arial",10))

dob\_format\_label.place(x=640,y=270)

dob\_f = StringVar()

dob\_entry = Entry(frame2,textvariable=dob\_f,font=("arial",15),highlightbackground="black", highlightthickness=3)

dob\_entry.place(x=820,y=240)

# Buttons

submit\_button = Button(frame2,text="Submit",font=("corbel",17),height=2,width=10,command=submit)

submit\_button.place(x=300,y=350)

clear\_button = Button(frame2,text="Clear",font=("corbel",17),height=2,width=10,command=clear)

clear\_button.place(x=800,y=350)

#final label

final\_label = Label(form,text="",bg="grey",font=("corbel",14))

final\_label.place(x=40,y=490)

form.mainloop()

Graphical user interface, website

Description automatically generated