Shri Ramdeobaba College of Engineering and Management, Nagpur-13.

Department of Electronics Engineering

ENT355-4–Object Oriented Programming Lab

Odd Semester – 2022-23

REPORT ON GUI USING TKINTER

Graphical user interface of DALL-E the OPEN AI

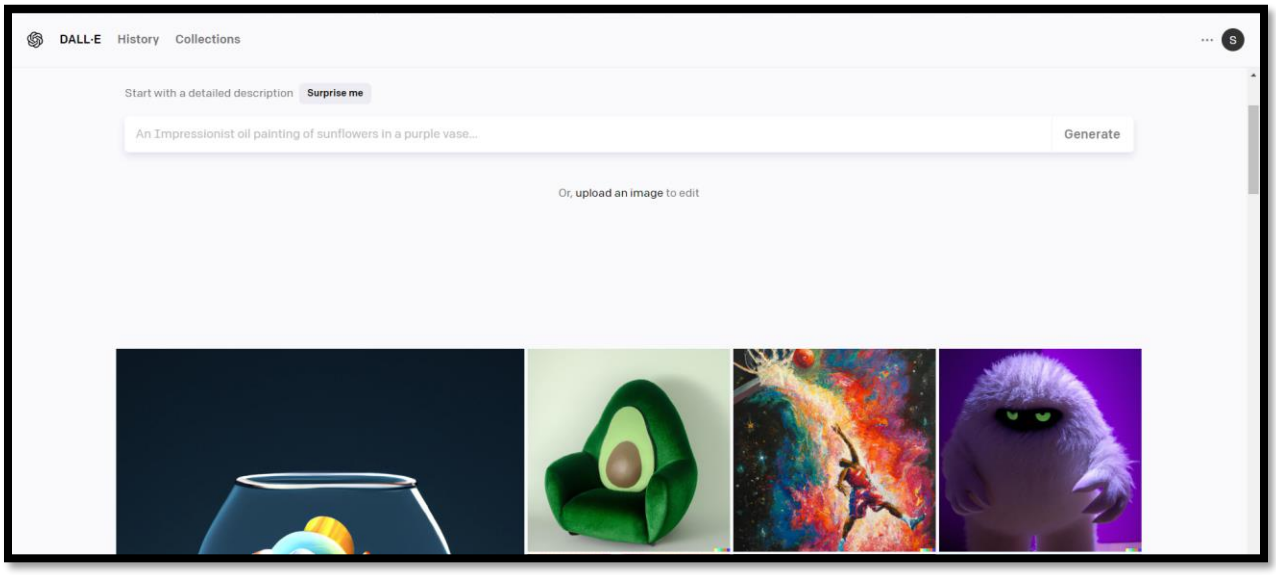
By Sejal Wasule

Roll no 48

Motivation and Reference interface:

Window 1:

Window 2:



Explanation for respective code snippets:

1. Creating first window and defining the parameters for it.

First, we import Tkinter, message box, dialogue and all the modules we need to make our GUI

We create and a root window or main window in which all the widgets will be contained.

We define the geometry of our window here geometry is (1000 x 600+300 + 200)

Then we configure the background and background color here I kept the resizability to be false so we are unable to maximize the window.

Code:

```
from tkinter import*
from tkinter import messagebox
from tkinter import filedialog
from tkinter.filedialog import askopenfile

root = Tk()
root.title("Welcome to open AI")
root.geometry("1000x1000+300+200")
root.configure(bg="#fff")
root.resizable(False,False)
```

2. Inserting image, heading, subheadings and menu buttons.

Here we added image and some menu buttons like follow on Instagram, View API Docs, Abouts, View Research. All the text is in teal color and font is Cambria.

Then we added a Frame of black colour and in that we put labels like heading (OPEN AI)and Subscript about Open AI. As background is black the text is white in colour for visibility.

Code:

```
img = PhotoImage(file="aiart.png")
Label(root,image=img,bg = "white").place(x=10,y=100)

blg = Button(root,width=13,text="BLOG",border = 0,bg="white",fg
="Teal",font=("Calibri",14,"bold”)
blg.place(x=10,y=00)
```

```
follw = Button(root,width=20,text="FOLLOW ON INSTAGRAM",border = 0,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
follw.place(x=150,y=00)

view = Button(root,width=15,text="VIEW API DOCS",border = 0,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
view.place(x=390,y=00)

abt= Button(root,width=10,text="ABOUT",border = 0,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
abt.place(x=560,y=00)

Add = Button(root,width=17,text="VIEW RESEARCH",border = 0,bg="white",fg  ="Teal",
font=("Calibri",14,"bold"))
Add.place(x=680,y=00)

frame = Frame(root,width=500,height=500,bg="black")
frame.place(x=480,y=70)

heading =
Label(frame,text="OPEN.AI",fg="#fff",bg="black",font=("Cambria",70,"bold"))
heading.place(x=50,y=5)

subheading = Label(frame,text="OPEN.AI is a new system that \n can create realistic
images\n and art from a description \n in natural language.",fg="#fff",bg="black",
font=("Cambria",25,"bold"))
subheading.place(x=30,y=120)
```

3. Creating frame and user input Login details on it.

Add the Label texts as Login and put username , password as Entry on the black frame. The username and password are given a command to redirect to a function signin using anouther Button named Enter(With text explore your imagination on it)

Added the label text as “don’t have an account?” on screen background .

The Button named signup is created and the signup command is attached to it.

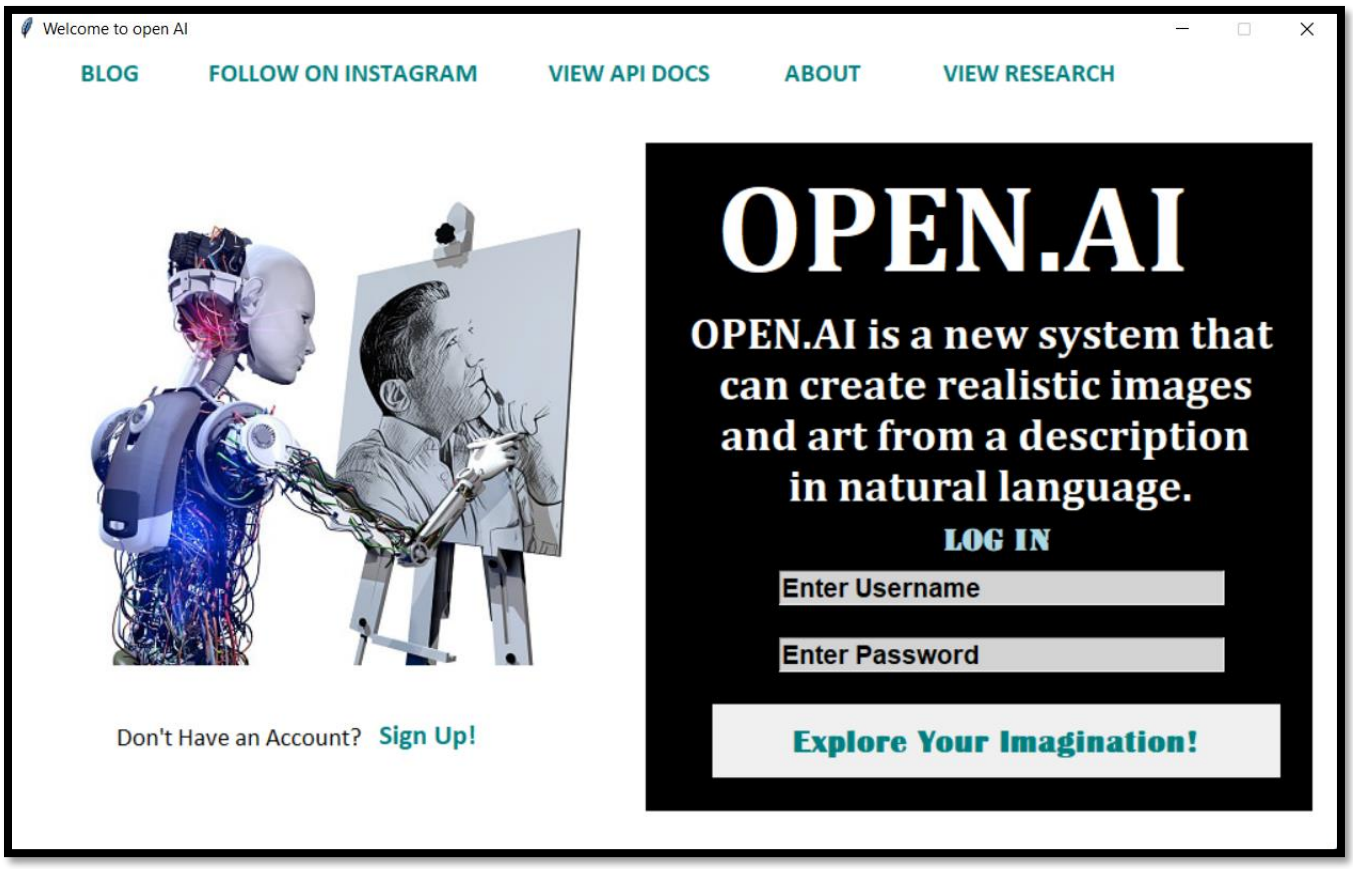
Also user and password input has functions like focus on clicking and focus out when clicked somewhere else.

So when the user clicks on entry labels of user and password the text which already exists dissappears and user does’nt have to delete all the already existing text and can directly enter the input.

Code:

```
login = Label(frame,text="Log in",fg="Light blue",bg="black",
font=("STENCIL",18,"bold"))
login.place(x=220,y=280)

def on_enter(e):
    user.delete("0","end")

def on_leave(e):
    name = user.get()
    if name=="":
        user.insert(0,"Username")
user = Entry(frame,width =30,fg = "black", border =1, bg="Light gray",font =("Comic
Sans",14,"bold"))
user.place(x=100,y=320)
user.insert(0,"Enter Username")
user.bind("<FocusIn>",on_enter)
user.bind("<FocusOut>",on_leave)

def on_enter(e):
    passw.delete("0","end")

def on_leave(e):
    code = passw.get()
    if code=="":
        passw.insert(0,"Password")


passw = Entry(frame, width =30,fg = "black", border =1, bg="Light gray",font
=("Comic Sans",14,"bold"))
passw.place(x=100,y=370)
passw.insert(0,"Enter Password")
passw.bind("<FocusIn>",on_enter)
passw.bind("<FocusOut>",on_leave)

Enter= Button(frame,width=30,pady=7,text="Explore Your Imagination!",fg
="Teal",border = 0,font=("Britannic Bold",18,"bold"),command=signin)
Enter.place(x=50,y=420)

Noacc = Label(root,text="Don't Have an Account?",fg="Black",bg="white", font
=("Calibri",14 ))
Noacc.place(x=80,y=500)

sign_up = Button(root,width=8,text="Sign Up!",border = 0,bg="white",fg
="Teal",font=("Calibri",16,"bold"),command=signup)
sign_up.place(x=270,y=492)

root.mainloop()  #goes in the end
```

- **Output : First window**



4. Creating a signup function to create anouther window.

Here we create a signup function in which we create a Toplevel window i.e child window of root window. We title new window as “Sign Up!”. We set up its geometry as (600*600+300+200) and configure it with a white background.We keep resizability as false so that we cannot maximise the window.

We are keeping this window very simple . We add labels like Create Account(Heading), Name, Email and create password.

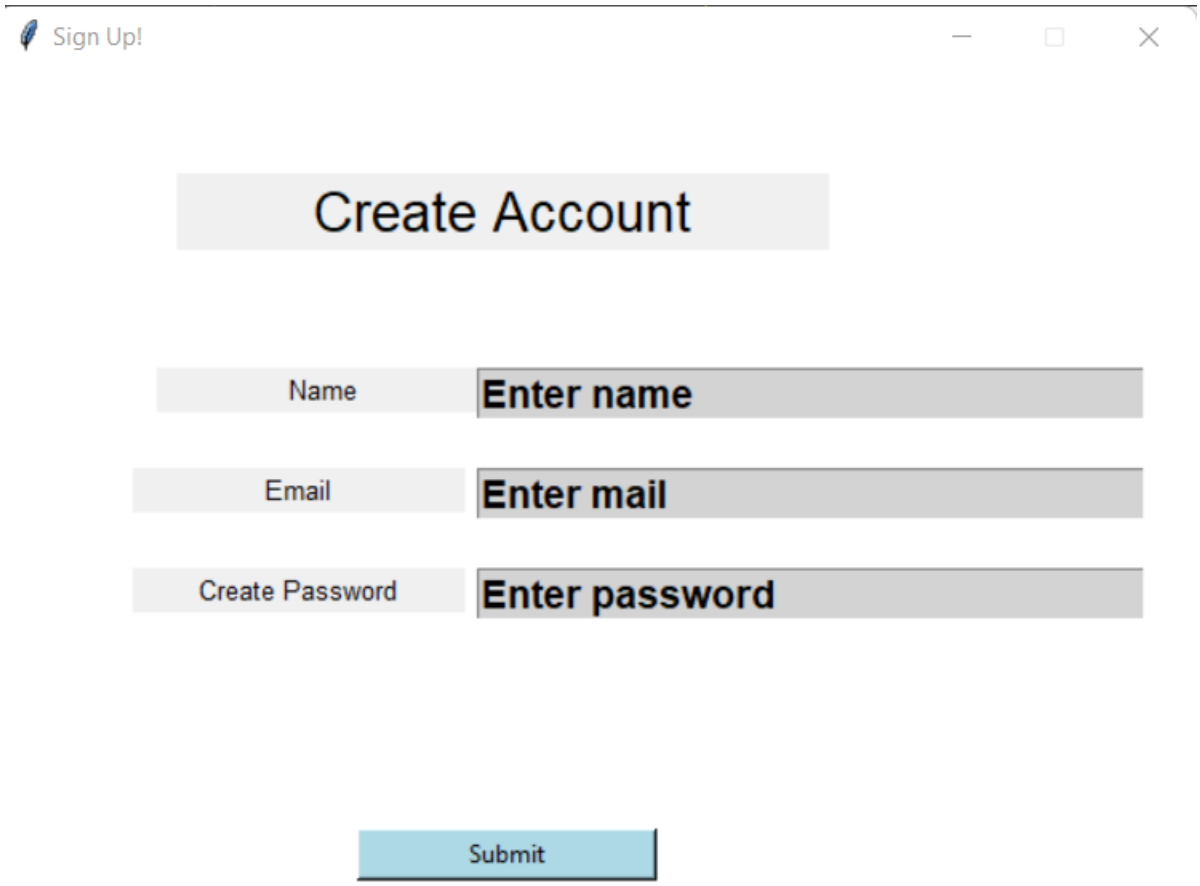
In front of them we create the entry boxes so user can enter their details (Name, Email and create passwords).

We create “Enter ” button so user can enter the information and enter a new window. So to the enter button we have given a command for summoning new window i.e signup function.

Code:

```
def signup():
    window = Toplevel(root)
    window.title("Sign Up!")
    window.geometry('600x600+300+200')
    window.configure(background = "white")
    window.resizable(False,False)


    labl_0 = Label(window, text="Create Account",width=20,font=("bold", 20))
    labl_0.place(x=90,y=53)

    name = Label(window, text="Name",width=20,font=("bold", 10))
    name.place(x=80,y=150)
    ent_name = Entry(window,width =30,fg = "black", border =1, bg="Light gray",font
=("Comic Sans",14,"bold"))
    ent_name.place(x=240,y=150)
    ent_name.insert(0,"Enter name")

    mail = Label(window, text="Email",width=20,font=("bold", 10))
    mail.place(x=68,y=200)
    ent_mail = Entry(window,width =30,fg = "black", border =1, bg="Light gray",font
=("Comic Sans",14,"bold"))
    ent_mail.place(x=240,y=200)
    ent_mail.insert(0,"Enter mail")

    newpas = Label(window, text="Create Password",width=20,font=("bold", 10))
    newpas.place(x=68,y=250)
    ent_newpas = Entry(window,width =30,fg = "black", border =1, bg="Light
gray",font =("Comic Sans",14,"bold"))
    ent_newpas.place(x=240,y=250)
    ent_newpas.insert(0,"Enter password")

    Button(window, text='Submit',width=20,bg='Light
Blue',fg='black',command=newwindow ).place(x=180,y=380)

    window.mainloop()
```

Output: Second window:



5. Creating a function to allow a certain username and password to enter new window.

We create the signin function using if else which will only take user to the next window if user inputs username as "Sejal" and password as "2002".

If user inputs any username or password other than above mentioned then a message box will appear which will inform user that username or password is invalid.

Code:

```
def signin():
    username = user.get()
    password= passw.get()

    if username == "Sejal" and password =="2002":
        newwindow()

    elif username!="Sejal" or password !="2002":
```

```
        messagebox.showerror("Invalid","Invalid username or password")
```

6. Creating a signup function to create anouther window.

We create a new function as newwindow(). In it we create We des root window or main window in which all the widgets will be contained.

We define the geometry of our window here geometry is (1000 x 600+300 + 200)

Then we configure the background and background color here I kept the resizability to be false so we are unable to maximize the window.

Code:

```
def newwindow():
    root.destroy()
    screen=Tk()
    screen.title("OPEN.AI-EXPLORE YOUR IMAGINATION!")
    screen.geometry("925x500+300+200")
    screen.config(bg="white")
```

7. Creating a menu bar in new window.

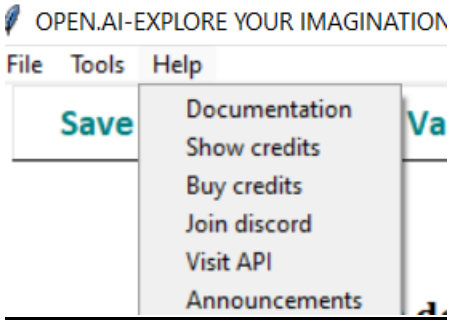
Creating a menubar with file, tool and more options in which open in pop up list of options. Here we kept tearoff 0 so there is no dotted line to separate the menu.

Code:

```
# Main Menu
    mainmenu = Menu(screen)

# Menubar
    filemenu = Menu(mainmenu, tearoff = 0)
    filemenu.add_command(label = "Open", command = filedialog.askopenfilename )
    filemenu.add_command(label = "save", command = filedialog.asksaveasfilename)
    filemenu.add_separator()
    filemenu.add_command(label = "Exit", command = screen.destroy)
    mainmenu.add_cascade(label="File", menu=filemenu)

    option = Menu(mainmenu, tearoff = 0)
    option.add_command(label = "Favourites" )
    option.add_command(label = "History")
    option.add_command(label = "Collections")
    option.add_command(label = "Settings")
    mainmenu.add_cascade(label="Tools", menu=option)
```

```
    More = Menu(mainmenu, tearoff = 0)
    More.add_command(label = "Documentation")
    More.add_command(label = "Show credits")
    More.add_command(label = "Buy credits")
    More.add_command(label = "Join discord")
    More.add_command(label = "Visit API")
    More.add_command(label = "Announcements")
    mainmenu.add_cascade(label = "Help", menu = More)

    screen.config(menu = mainmenu)
```

Output:



8. Creating a signup function to create anouther window.

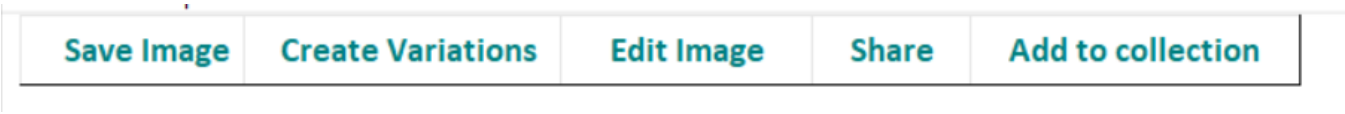
Creating couple of buttons for options like save, create variations, Edit Image, Share and add Collection.

Code:

```
#Widgets
    download = Button(screen,width=13,text="Save Image",border = 1,bg="white",fg
="Teal",font=("Calibri",14,"bold"),command = filedialog.asksaveasfilename)
    download.pack(expand=True)
    download.place(x=10,y=00)

    vartn = Button(screen,width=17,text="Create Variations",border =
1,bg="white",fg ="Teal",font=("Calibri",14,"bold"))
    vartn.pack(expand=True)
    vartn.place(x=130,y=00)

    edit = Button(screen,width=13,text="Edit Image",border = 1,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
    edit.pack(expand=True)
    edit.place(x=300,y=00)

    share= Button(screen,width=8,text="Share",border = 1,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
    share.pack(expand=True)
    share.place(x=435,y=00)
```

```
    Add  = Button(screen,width=17,text="Add to collection",border = 1,bg="white",fg
="Teal",font=("Calibri",14,"bold"))
    Add.pack(expand=True)
    Add.place(x=520,y=00)
```

Output:



9. Adding features like entry inputbar and Generate bar.

First we create a function a which will show a dialogue box that image is generating.

Then we add a label asking user to give detailed description of the image they want.

We create a frame in which create an entry box with some example text and generate button which will call the generate function created earlier.

Also text entry input has functions like focus on clicking and focus out when clicked somewhere else.

So when the user clicks on entry labels the text which already exists dissappears and user does'nt have to delete all the already existing text and can directly enter the input.

Code:

```
def generating():
        messagebox.showinfo("Generating ......")

    a=Label(screen, text="Start with a detailed
description",bg="white",font=("Cambria",15,"bold"))
    a.pack(expand=True)
    a.place(x=100,y=100)

    surprise = Button(screen,width=10,text="Surprise me",border = 1,bg="white",fg
="Teal",font=("Calibri",16,"bold"),command= generating)
    surprise.pack(expand=True)
    surprise.place(x=450,y=90)

    frm = Frame(screen,width=1000,height=100,bg="Light Blue")
    frm.pack()
    frm.place(x=100,y=170)

    def on_enter(e):
        Input.delete("0","end")
```

```
    def on_leave(e):
        D = Input.get()
        if D=="":
            Input.insert(0,"Username")

    Input = Entry(frm,width =100,fg = "black", border =0, bg="Light Blue",font
=("Comic Sans",12))
    Input.pack()
    Input.place(x=70,y=30)
    Input.insert(0,"Enter Detailed Discription(Eg:An Avocado shaped Chair)")
    Input.bind("<FocusIn>",on_enter)
    Input.bind("<FocusOut>",on_leave)

    generate = Button(frm,width=10,text="Generate",border = 1,bg="Black",fg
="white",font=("Calibri",16,"bold"),command=generating)
    generate.pack(expand=True)
    generate.place(x=860,y=40)

    upload = Button(screen,width=30,text="Upload an Image you want to edit",border
= 0,bg="white",fg ="Teal",font=("Calibri",16,"bold"),command =
filedialog.askopenfilename)
    upload.pack(expand=True)
    upload.place(x=450,y=270)
```

10. Creating other frame to contain history and few other buttons.

We create a frame of background teal on which we create 3 buttons named clear all, logout (Exits window) and recent button.

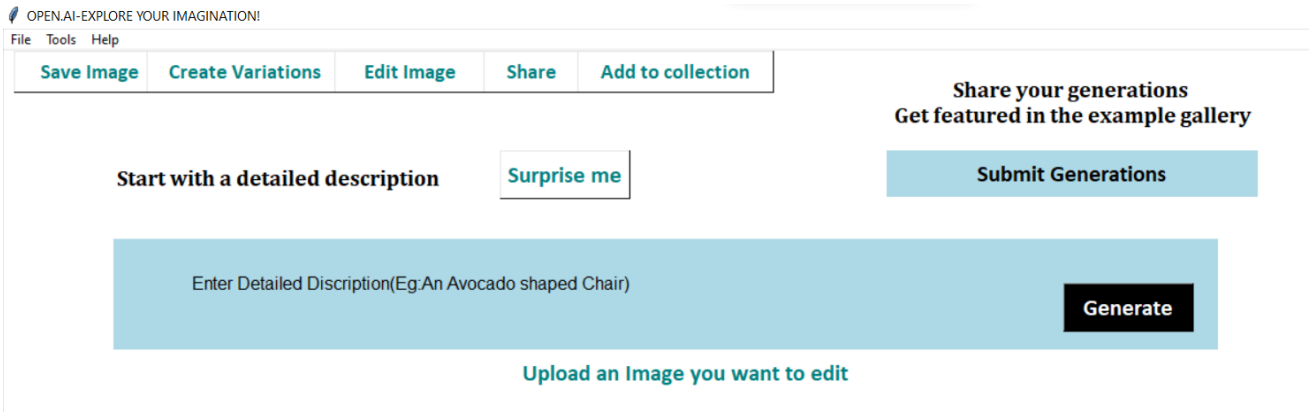
We add a Label asking a user sharing their generations for featuring in the library and a button will be also created to submit these generations. Submit button when clicked will open your files and ask you to select file.

Code:

```
    frame1 = Frame(screen,width=400,height=800,bg="Teal")
    frame1.place(x=1200,y=0)

    recent = Button(frame1,width=8,text="Recent",border = 0,bg="black",fg
="white",font=("Calibri",16))
    recent.pack(expand=True)
    recent.place(x=50,y=20)

    clr = Button(frame1,width=8,text="Clear all",border = 0,bg="black",fg
="white",font=("Calibri",16))
    clr.pack(expand=True)
```

```
    clr.place(x=200,y=20)

    logout = Button(frame1,width=8,text="Logout",border = 0,bg="black",fg
="white",font=("Calibri",16),command=screen.destroy)
    logout.pack(expand=True)
    logout.place(x=150,y=700)

    b=Label(screen, text="Share your generations\n Get featured in the example
gallery",bg="white",font=("Cambria",15,"bold"))
    b.pack(expand=True)
    b.place(x=800,y=20)

    subgen = Button(screen,width=30,text="Submit Generations",border = 0,bg="Light
Blue",fg ="black",font=("Calibri",16,"bold"),command = filedialog.askopenfilename)
    subgen.pack(expand=True)
    subgen.place(x=800,y=90)
```

Output:



11. Creating a gallery and adding pictures.

We will create a label for naming gallery and then we are adding a few images in gallery and also creating them in such a way that they act as buttons. And then we add lables below those pictures asking user to try and click these pictures.

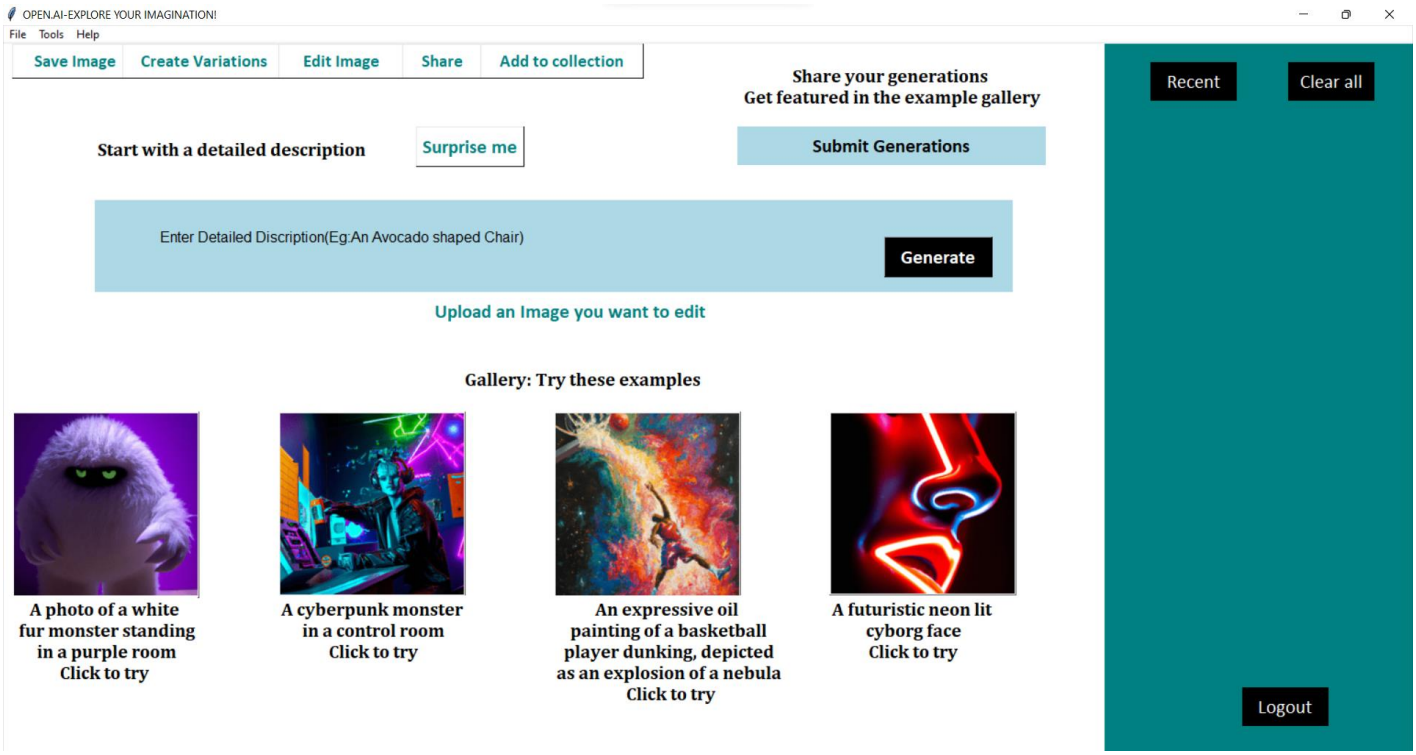
Code:

```
gal=Label(screen, text="Gallery: Try these
examples",bg="white",font=("Cambria",15,"bold"))
    gal.pack(expand=True)
    gal.place(x=500,y=350)

    Img_btn1 = PhotoImage(file = "furry monster.png")
    img1 = Button(screen,image = Img_btn1,borderwidth = 1 )
    img1.pack()
    img1.place(x = 10 ,y=400)
```

```
    img11=Label(screen, text="A photo of a white\n fur monster standing\n in a
purple room\nClick to try",bg="white",font=("Cambria",15,"bold"))
    img11.pack(expand=True)
    img11.place(x=10,y=600)

    Img_btn2 = PhotoImage(file = "roboNeon .png")
    img2 = Button(screen,image = Img_btn2,borderwidth = 1 )
    img2.pack()
    img2.place(x = 300 ,y=400)
    img22=Label(screen, text="A cyberpunk monster\n in a control room\n Click to
try",bg="white",font=("Cambria",15,"bold"))
    img22.pack(expand=True)
    img22.place(x=300,y=600)

    Img_btn3 = PhotoImage(file = "Supernova.png")
    img3 = Button(screen,image = Img_btn3,borderwidth = 1 )
    img3.pack()
    img3.place(x = 600 ,y=400)
    img33=Label(screen, text="An expressive oil \npainting of a basketball\nplayer
dunking, depicted\nas an explosion of a nebula\n Click to
try",bg="white",font=("Cambria",15,"bold"))
    img33.pack(expand=True)
    img33.place(x=600,y=600)

    Img_btn4 = PhotoImage(file = "neonlights.png")
    img4 = Button(screen,image = Img_btn4,borderwidth = 1 )
    img4.pack()
    img4.place(x =900 ,y=400)
    img44=Label(screen, text="A futuristic neon lit \n cyborg face \nClick to
try",bg="white",font=("Cambria",15,"bold"))
    img44.pack(expand=True)
    img44.place(x=900,y=600)


    screen.mainloop()
```

Output:

Final window Output:



Final code is submitted as a separate file.

Reference website link: https://labs.openai.com/e/ZhBaom4oO9C92TzSXlkQmKBP