Design:

The code is a Python script that uses the tkinter library to create a graphical user interface (GUI) that allows users to display and browse images. The GUI has a button called "show_image_as_button()" that displays all the images in a specific directory, and when the user clicks on any of these images, a new frame is created, which contains the images of that friend's friends. The images are displayed as buttons and labels, and the user can remove the images by clicking on an "X" button.

The script has defined some global variables, such as the path of the image directory, the size of the labelFrame, image size, label width, and height, etc.

The GUI has three main functions: show_image_as_button(), show_friends_of_friend(), remove_button(), addNewFriend(), and deleteNewFriend().

Implementation:

Import all required libraries such as tkinter, pillow, and os.
Initialize some global variables such as path, filedir, friend_friend_image_path_list, friend_friend_labelFrame_list, frame_row, image_extension, label_width, label_height, and image_size.
Create a tkinter window and set its title and size.
Define the show_image_as_button() function, which creates a new LabelFrame to hold the images of friends. The function first checks whether the gallery is already displayed or not, and if it is displayed, a messagebox appears. Otherwise, it reads all the image files from the image directory and sorts them alphabetically. It then creates a button for each image and adds it to the LabelFrame. The button shows the friend's name, and when clicked, it shows that friend's friends in a new frame. Each friend's friend image is displayed as a label, and a remove button is added to the label, which can be used to remove the image from the frame.
Define the show_friends_of_friend() function, which creates a new frame and adds all the images of that friend's friends to the frame as labels. It also adds a remove button to each label, which can be used to remove the image from the frame.
Define the remove_button() function, which is called when the user clicks on the remove button. It removes the label from the frame.

Evaluation:

The code is well-structured, and the functions are well-defined. The use of global variables is minimized, and the code is easy to read and understand. The user interface is simple and easy to use. The code can be further improved by adding error handling and input validation.