App4: Photo Searcher

Outside packages:

# kivy \*\* # GUI tool:

## backbone layout # .py file

# To start with, set up your main app

from kivy.app import App

from kivy.screenmanager import ScreenManager, Screen

from kivy.lang import Builder # create .kv file in kv language

Builder.load\_file( ‘XX.kv’ )

# instead of making widgets in the current .py file, you write in .kv file and import

class \*\* your screen\*\* (Screen): # if you want more screens, add more classes

# functions in your screen

class RootWidget (ScreenManager): # screen manager

pass

class MainApp(App):

def build (self):

return RootWidget # what it returns is what you see on the screen

# run your app

MainApp().run()

## frontend.kv # .kv file sample

# no ‘=’, only ‘:’ in .kv file

<FirstScreen>: -> indicates the screen

GridLayout:

cols: 1 -> each column 1 widgets

padding: 10 -> adding space between left and right

spacing: 10 -> adding space between each widgets

Image:

id: img

size\_hint\_y: 0.8 -> proportion of each column

TextInput:

size\_hint\_y: 0.1

Button:

text: 'Search Image'

size\_hint\_y: 0.1

on\_press: root.search\_image()

-> function when button is clicked

<RootWidget>: -> indicates ScreenManager

FirstScreen:

id: first\_screen

name: 'first\_screen'

# wikipedia # online search tool

import wikipedia

page= wikipedia.page( ‘\* search bar \*’ ) # type in what you want to search

page.images # return a list of urls of the images of your search

page.summary # return a str of the defination of that search

# dir(page) in the console to see more

# requests # request HTTP web tool

import requests

req= requests.get( \* URL\*) # req is a requests model class

req.content # if url is an img, it will be a str of bytes

with open( ‘XX.jpg’, ‘wb’) as file:

file.write(req.content) # convert bytes into img and save in the file.