from kivy.app import App

from kivy.uix.screenmanager import ScreenManager, Screen

from kivy.uix.floatlayout import FloatLayout

from kivy.core.window import Window

from kivy.lang import Builder

from kivy.uix.image import Image

from kivy.uix.button import ButtonBehavior, Button

from kivy.uix.label import Label

from kivy.uix.widget import Widget

from kivy.uix.checkbox import CheckBox

from kivy.uix.textinput import TextInput

from kivy.uix.dropdown import DropDown

from kivy.uix.popup import Popup

import sqlite3

import random

db = sqlite3.connect("RAW\_recipes.db") # The DataSet

curs = db.cursor() # Cursor for execution

info\_list = open("info\_list.txt", "r").read().split("|") # List as memory

favourite\_list = info\_list[0].split(",")[:-1] # Contains FavouriteRecipes

fridge\_list = info\_list[1].split(",")[:-1] # Contains FridgeIngredients

shopping\_list = info\_list[2].split(",")[:-1] # Contains ShoppingLists

sm = ScreenManager()

sm.transition.direction = 'down'

logo = """

Label:

text: "CookHub"

font\_size: root.height//1.4

color: (224/255,10/255,20/255,1)

canvas.before:

Color:

rgba: (1,1,1,1)

RoundedRectangle:

pos: self.pos

size: self.size

radius: [17,17,17,17]

size\_hint: 0.8,0.1

pos\_hint: {"x":0.1, "top":0.9}

""" # Will present on each page.

height = Window.height

class ShowingList(FloatLayout):

f\_s = (Window.width + Window.height)//115 # Static font\_size for dynamic layout

t\_s = (Window.width//7, None) # Static text\_size for dynamic layout

def \_\_init\_\_(self, recipe\_name="None"):

super(ShowingList, self).\_\_init\_\_()

self.list\_logo = Builder.load\_string("""

Label:

text:"I need to buy:"

size\_hint: (0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.95}

font\_size:(root.width+root.height)//12

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

""")

self.add\_widget(self.list\_logo)

self.to\_buy\_list = Builder.load\_string("""

ScrollView:

size\_hint: (0.9, 0.6)

pos\_hint: {"x": 0.05, "top": 0.81}

s\_gl: s\_gl

s\_gl1: s\_gl1

s\_gl2: s\_gl2

GridLayout:

id: s\_gl

cols: 2

size\_hint\_y: None

spacing: root.width//25, 0

GridLayout:

id: s\_gl1

spacing: 0, root.width//15

cols: 1

size\_hint: 0.2, None

GridLayout:

id: s\_gl2

spacing: 0, root.width//15

cols: 1

size\_hint: 0.8, None

""")

self.add\_widget(self.to\_buy\_list)

required\_ingredients = str(curs.execute(f'SELECT ingredients FROM '

f'CookHub WHERE '

f'name="{recipe\_name.lower()}"').fetchall())

required\_ingredients = required\_ingredients[4:-5].split(",")

z = []

for i in required\_ingredients:

i = i.strip(" ' ").capitalize()

if i not in fridge\_list:

z.append(i)

self.to\_buy\_list.s\_gl.height = len(z)\*50

self.to\_buy\_list.s\_gl1.height = len(z)\*50

self.to\_buy\_list.s\_gl2.height = len(z)\*50

for i in sorted(z):

if\_bought = CheckBox(id=f"{i}", color=(1, 1, 120/255, 1))

self.to\_buy\_list.s\_gl1.add\_widget(if\_bought)

lists\_btn = Button(text=f"{i}",

font\_size=self.f\_s, text\_size=self.t\_s,

background\_color=(63/255, 135/255, 232/255, 1))

self.to\_buy\_list.s\_gl2.add\_widget(lists\_btn)

# self.add\_widget(self.close)

class AddingIngredients(FloatLayout): # A popup content for VirtualFridge

def \_\_init\_\_(self):

super(AddingIngredients, self).\_\_init\_\_()

self.ingredients\_input = TextInput(padding=(10, 10),

size\_hint=(0.8, 0.5), pos\_hint={"x": .1, "top": .8}) # List of addings

self.add\_widget(self.ingredients\_input)

self.dd = DropDown(max\_height=Window.height\*0.4) # Will Display All Possible Ingredients

self.mainbutton = Button(text="Select Ingredients:",

size\_hint=(.9, .1), pos\_hint={"x": .05, "top": .25},

background\_normal="selection.png")

self.mainbutton.bind(on\_release=self.dd.open)

self.add\_widget(self.mainbutton)

self.submition = Button(text="Add Ingredients", # Press to expand Fridge

size\_hint=(0.9, 0.1), pos\_hint={"x": .05, "top": 0.13},

background\_color=(179/255, 220/255, 111/255, 1),

on\_press=self.add\_to\_fridge)

self.add\_widget(self.submition)

unique\_ingredients = curs.execute("SELECT ingredients FROM UNIQUES").fetchall()

for i in sorted(unique\_ingredients[0][0].strip("[]").split(",")):

i = i.strip(" ' ")

if i[0] == " ":

i = i[2:]

btn = Button(text=f"{i.capitalize()}", text\_size=(self.width, None),

size\_hint\_y=None,

on\_release=lambda buttn: self.dd.select(buttn.text),

background\_color=(12/255, 63/255, 82/255, 1))

self.dd.add\_widget(btn)

self.dd.bind(on\_select=lambda intance, x: self.add\_to\_input(x))

def add\_to\_input(self, x): # Chosen ingredients will be Added to INgredientsInput

try:

if self.ingredients\_input.text[-1] == ",":

self.ingredients\_input.text += f"{x},"

else:

self.ingredients\_input.text += f",{x},"

except:

self.ingredients\_input.text += f"{x},"

def add\_to\_fridge(self, action): # Ingredients From IngredientsInput getting added to Fridge

try:

new\_ingredients = self.ingredients\_input.text.split(",")

for i in new\_ingredients:

if i != "":

fridge\_list.append(i)

with open("info\_list.txt", "w") as file:

for i in favourite\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

virtualfridge\_screen.ingredients\_listing()

self.ingredients\_input.text = ""

virtualfridge\_screen.popup\_window.dismiss()

except:

pass

Builder.load\_string(""" # Easier to add via loading

<AddingIngredients>:

Label:

text:"Add Ingredients:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.95}

font\_size:(root.width+root.height)//30

color: (1,1,1,1)

canvas.before:

Color:

rgba: (45/255, 105/255, 200/255, 1)

Rectangle:

size: self.size

pos: self.pos

""")

class ImageButton(ButtonBehavior, Image):

def \_\_init\_\_(self, src="add.png", \*\*kwargs):

super(ImageButton, self).\_\_init\_\_(\*\*kwargs)

self.source = src

if src == "delete.png":

self.on\_press = self.delete\_ingredients

if src == "delete2.png":

self.on\_press = self.delete\_shoppinglists

def delete\_ingredients(self): # Will remove Chosen Ingredient From Virtual Fridge

fridge\_list.remove(self.id)

with open("info\_list.txt", "w") as file:

for i in favourite\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

virtualfridge\_screen.ingredients\_listing()

def delete\_shoppinglists(self): # Will remove Chosen ShoppingList

shopping\_list.remove(self.id)

with open("info\_list.txt", "w") as file:

for i in shopping\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

shoppinglist\_screen.shopping\_listing()

class IconButton(ButtonBehavior, Image): # Pressing will switch Pages

def \_\_init\_\_(self, src, dest="Home"):

super(IconButton, self).\_\_init\_\_()

self.source = src

# self.size = Window.width\*0.1, 140

self.size\_hint = 0.1, None

self.allow\_stretch = True

self.keep\_ratio = False

self.dest = dest

def on\_press(self):

sm.transition.direction = "up"

sm.current = self.dest

class ButtonPanel(Widget):

def \_\_init\_\_(self, direct="down"):

super(ButtonPanel, self).\_\_init\_\_()

sm.transition.direction = direct

self.icons\_box = Builder.load\_string(""" # Buttons' Holder

GridLayout:

size\_hint: 1,None

height: {}

rows:1

""".format(height\*0.12))

self.icons\_box.add\_widget(IconButton(src="home.png"))

self.icons\_box.add\_widget(IconButton(src="search.png",

dest="Search"))

self.icons\_box.add\_widget(IconButton(src="virtualfridge.png",

dest="VF"))

self.icons\_box.add\_widget(IconButton(src="shoppinglist.png",

dest="SL"))

self.icons\_box.add\_widget(IconButton(src="favourite.png", dest="F"))

class RecipeDetails(Screen, FloatLayout): # Shows Info of chosen recipe

if\_favourite\_btn = Button()

if\_favourite\_cb = CheckBox()

create\_shoppingist = Button()

listing = ""

def \_\_init\_\_(self, recipe\_name, minutes, tags, nutrition, n\_steps,

steps, description, ingredients, n\_ingredients, \*\*kwargs):

super(RecipeDetails, self).\_\_init\_\_(\*\*kwargs)

self.logo = Builder.load\_string(logo)

self.logo.pos\_hint = {"x": .1, "top": .98}

self.add\_widget(self.logo)

self.create\_shoppingist = Button(text="Create Shopping List",

size\_hint=(0.7, 0.07),

pos\_hint={"top": 0.86, "x": 0.15},

font\_size=(Window.width+Window.height)//55,

on\_press=self.change\_shoppinglist,

background\_color=(179/255, 220/255, 111/255, 1))

self.if\_favourite\_btn = Button(text="Select As Favourite",

size\_hint=(0.7, 0.07),

pos\_hint={"top": 0.78, "x": 0.15},

font\_size=(Window.width+Window.height)//55,

background\_color=(179/255, 220/255, 111/255, 1))

self.add\_widget(self.create\_shoppingist)

self.if\_favourite\_btn.on\_press = self.change\_favourite

self.add\_widget(self.if\_favourite\_btn)

self.if\_favourite\_cb = CheckBox(size\_hint=(0.15, 0.3),

pos\_hint={"top": 0.9, "x": 0},

color=(233/255, 169/255, 63/255, 1))

self.add\_widget(self.if\_favourite\_cb)

if recipe\_name in favourite\_list:

self.if\_favourite\_cb.active = True

self.if\_favourite\_btn.text = "Remove from Favourite"

self.recipe\_name = recipe\_name

self.minutes = minutes

self.tags = tags

self.nutrition = nutrition[1:-1].split(",")

self.n\_steps = n\_steps

self.steps = steps

self.description = description

self.ingredients = ingredients

self.n\_ingredients = n\_ingredients

self.all\_details = Builder.load\_string(""" # Will display all information

GridLayout:

cols:1

sv:sv

size\_hint: 1, 0.7

pos\_hint: {0}

ScrollView:

id: sv

gl: gl

l:l

l2: l2

size\_hint: 1, 1

GridLayout:

id: gl

spacing: 0, root.height\*0.03

height: root.height\*3

size\_hint\_y: None

cols: 1

GridLayout:

cols: 2

spacing: root.width\*0.03, 0

GridLayout:

spacing: 0, root.height\*0.04

size\_hint\_y: 0.2

cols: 1

Label:

text: "{2}"

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//35

size\_hint: 0.5, 0.2

text\_size: root.width//2, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

Label:

text: "{3}"

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//40

size\_hint: 0.5, 0.2

text\_size: root.width//2, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

Label:

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//40

text: "{4}"

size\_hint: 0.5, 0.2

text\_size: root.width//2, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

Label:

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//40

text: "{5}"

size\_hint: 0.5, 0.2

text\_size: root.width//2, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

GridLayout:

size\_hint: (0.9, 0.55)

cols: 1

ScrollView:

Label:

size\_hint\_y: None

height: root.height\*3

text: "{6}"

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//40

text\_size: root.width//2, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

GridLayout:

rows:4

spacing: 0, root.height\*0.03

ScrollView:

Label:

height: root.height//1.1

size\_hint\_y: None

padding\_x: root.width\*0.05

text: "Ingredients:{1}"

font\_size: (root.width+root.height)//35

text\_size: root.width, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

ScrollView:

Label:

size\_hint\_y: None

padding\_x: root.width\*0.05

height: root.height//1.1

text: "Tags: {7}"

font\_size: (root.width+root.height)//35

text\_size: root.width, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

ScrollView:

Label:

id: l

size\_hint\_y: None

padding\_x: root.width\*0.05

height: root.height//1.1

font\_size: (root.width+root.height)//35

text\_size: root.width, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

ScrollView:

Label:

id: l2

height: root.height\*2

size\_hint\_y: None

padding\_x: root.width\*0.05

font\_size: (root.width+root.height)//35

text\_size: root.width, None

canvas.before:

Color:

rgba: (52/255,72/255,96/255,1)

Rectangle:

size: self.size

pos: self.pos

""".format({"x": 0, "top": 0.69}, self.ingredients[1:-1],

f"{self.recipe\_name}", f"Takes {self.minutes} minutes!",

f"Number of ingredients: {self.n\_ingredients}!",

f"Number of steps: {self.n\_steps}",

str(self.description.capitalize()), self.tags[1:-1]))

self.all\_details.add\_widget(ButtonPanel().icons\_box)

z = "Steps:"

j = 1

for i in list(self.steps[1:-1].split("',")):

i = i.strip("'")

i = i.strip(" '")

z += f'\n{j}){i.capitalize()}'

j += 1

self.all\_details.sv.l2.text = z

self.all\_details.sv.l.text = f"""

Nutrition:

Total Fat(PDV):{self.nutrition[0]}

Sugar(PDV):{self.nutrition[1]}

Sodium(PDV):{self.nutrition[2]}

Protein(PDV):{self.nutrition[3]}

Saturated Fat(PDV):{self.nutrition[4]}

Carbohydrates(PDV):{self.nutrition[5]}"""

self.add\_widget(self.all\_details)

def change\_favourite(self): # Will either add or remove from favourites:

if "Select" in self.if\_favourite\_btn.text:

self.if\_favourite\_cb.active = True

self.if\_favourite\_btn.text = "Remove from Favourite"

if self.recipe\_name not in favourite\_list:

favourite\_list.append(self.recipe\_name)

with open("info\_list.txt", "w") as file:

for i in favourite\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

else:

try:

favourite\_list.remove(self.recipe\_name)

finally:

with open("info\_list.txt", "w") as file:

for i in favourite\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

self.if\_favourite\_cb.active = False

self.if\_favourite\_btn.text = "Select As Favourite"

favourite\_screen.listing()

def change\_shoppinglist(self, action):

shopping\_list.append(self.recipe\_name)

shoppinglist\_screen.shopping\_listing()

with open("info\_list.txt", "w") as file:

for i in favourite\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in fridge\_list:

print(f"{i}", end=",", file=file)

print("|", end="", file=file)

for i in shopping\_list:

print(f"{i}", end=",", file=file)

Builder.load\_string("""

<RecipeButton>:

random\_color: (1,1,1,1)

canvas.before:

Color:

rgba: self.random\_color

Rectangle:

size: self.size

pos: self.pos

""") # Necessitated due to background

class RecipeButton(ButtonBehavior, Label): # Creates Button redirecting to recipe details

def \_\_init\_\_(self, rcp, text\_size=(Window.width\*0.07, None),

font\_size=Window.width//60):

super(RecipeButton, self).\_\_init\_\_()

self.text = rcp

self.text\_size = text\_size

self.font\_size = font\_size

colors\_choice = [

(154/255, 88/255, 188/255, 1),

(52/255, 205/255, 99/255, 1),

(27/255, 160/255, 131/255, 1),

(211/255, 85/255, 0/255, 1),

(152/255, 164/255, 164/255, 1),

(235/255, 112/255, 94/255, 1),

(39/255, 127/255, 191/255, 1)]

self.random\_color = random.choice(colors\_choice)

def on\_press(self):

recipe\_details = curs.execute(

f'SELECT \* FROM CookHub where name'

f'="{self.text.lower()}"').fetchall()[0]

try:

sm.remove\_widget(sm.get\_screen("RD"))

except:

pass

finally:

sm.add\_widget(RecipeDetails(name="RD", recipe\_name=self.text,

minutes=recipe\_details[1], tags=recipe\_details[2],

nutrition=recipe\_details[3], n\_steps=recipe\_details[4],

steps=recipe\_details[5], description=recipe\_details[6],

ingredients=recipe\_details[7], n\_ingredients=recipe\_details[8]))

sm.current = "RD"

Builder.load\_string("""

<ShowingList>:

close: close

Button:

id: close

text:"Done!" # Press to close Popup

size\_hint:(0.9, 0.1)

pos\_hint:{"x": .05, "top": 0.15}

background\_color:(179/255, 220/255, 111/255, 1)

""")

Builder.load\_string("""

<SmoothButton@Button>:

background\_color: (0,0,0,0)

canvas.before:

Color:

rgba: (10/255, 101/255, 131/255, 1)

RoundedRectangle:

pos: self.pos

size: self.size

radius: [15,15,15,15]

<Search>:

by\_name: by\_name

by\_tag: by\_tag

by\_ingredient: by\_ingredient

Label:

text:"Search By:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.75}

font\_size:(root.width+root.height)//30

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

SmoothButton:

id: by\_name

text: "By Name"

size\_hint: 0.8, 0.1

pos\_hint: {"x":0.1, "top":0.6}

SmoothButton:

id: by\_tag

text: "By Tag"

size\_hint: 0.8, 0.1

pos\_hint: {"x":0.1, "top":0.47}

SmoothButton:

id: by\_ingredient

text: "By Ingredient"

size\_hint: 0.8, 0.1

pos\_hint: {"x":0.1, "top":0.34}

""")

class ByIngredient(FloatLayout):

f\_s = (Window.width + Window.height)//100 # Static font\_size for dynamic layout

t\_s = (Window.width//6, None) # Static text\_size for dynamic layout

id\_s = []

name = [] # Container for matching names, preventing duplicates

def \_\_init\_\_(self):

super(ByIngredient, self).\_\_init\_\_()

self.add\_widget(Builder.load\_string("""

Label:

text:"Enter Ingredient:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":.97}

font\_size:(root.width+root.height)//12

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

"""))

self.search\_output = Builder.load\_string("""

ScrollView:

size\_hint: 0.9, .46

pos\_hint: {"x":0.05, "top": .6}

gl:gl

GridLayout:

id: gl

cols: 1

spacing: 0, root.width//25

size\_hint\_y: None

""")

self.add\_widget(self.search\_output)

self.search\_input = TextInput(padding=(5, 5), size\_hint=(0.69, 0.1),

pos\_hint={"x": .05, "top": .84}, multiline=False)

self.do\_search = ImageButton(src="search2.png", size\_hint=(.2, .2),

pos\_hint={"x": .75, "top": .89})

self.do\_search.on\_press = self.find\_by\_ingredient

self.add\_widget(self.search\_input)

self.add\_widget(self.do\_search)

self.close = Button(text="Close", size\_hint=(0.8, 0.1),

pos\_hint={"x": .1, "top": .1},

background\_color=(179/255, 220/255, 111/255, 1))

self.add\_widget(self.close)

self.dd = DropDown(max\_height=Window.height//3)

self.mainbutton = Button(text="Select Ingredient:", size\_hint=(0.9, 0.1),

pos\_hint={"x": .05, "top": .72},

font\_size=self.f\_s,

background\_normal="selection.png")

self.mainbutton.bind(on\_release=self.dd.open)

self.add\_widget(self.mainbutton)

ingredient = curs.execute("SELECT ingredients from Uniques").fetchall()

btn = Button(text="1 From VirtualFridge", text\_size=(self.width, None),

size\_hint\_y=None, height=Window.height//15,

on\_release=lambda buttn: self.change\_input(buttn.text),

background\_color=(12/255, 63/255, 82/255, 1),)

self.dd.add\_widget(btn)

for i in sorted(ingredient[0][0][1:-1].split(",")):

i = i.strip(" ' ").capitalize()

btn = Button(text=f"{i}", text\_size=(self.width, None),

size\_hint\_y=None, height=Window.height//15,

on\_release=lambda buttn: self.change\_input(buttn.text),

background\_color=(12/255, 63/255, 82/255, 1),)

self.dd.add\_widget(btn)

def change\_input(self, x):

self.search\_input.text += f"{x},"

def find\_by\_ingredient(self):

self.name = []

ingredients = curs.execute("SELECT ingredients FROM CookHub").fetchall()

if "1 From VirtualFridge" in self.search\_input.text:

for f in fridge\_list:

self.search\_input.text = self.search\_input.text + f"{f},"

for i in ingredients:

for j in self.search\_input.text.split(",")[:-1]:

if j.lower() in str(i):

try:

x = curs.execute(f'SELECT name FROM'

f' CookHub where ingredients="{str(i)[2:-3]}"').fetchall()

if x[0][0] not in self.name:

self.name.append(x[0][0])

except:

continue

try:

for i in self.id\_s:

self.search\_output.gl.remove\_widget(i)

except:

pass

self.search\_output.gl.height = 0

for i in self.name:

btn = RecipeButton(rcp=i.capitalize(),

text\_size=self.t\_s, font\_size=self.f\_s)

self.id\_s.append(btn)

self.search\_output.gl.add\_widget(btn)

self.search\_output.gl.height += 50

class ByTag(FloatLayout):

f\_s = (Window.width + Window.height)//100 # Static font\_size for dynamic layout

t\_s = (Window.width//6, None) # Static text\_size for dynamic layout

id\_s = []

name = [] # Container for matching names, preventing duplicates

def \_\_init\_\_(self):

super(ByTag, self).\_\_init\_\_()

self.add\_widget(Builder.load\_string("""

Label:

text:"Enter Tag:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":.97}

font\_size:(root.width+root.height)//12

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

"""))

self.search\_output = Builder.load\_string("""

ScrollView:

size\_hint: 0.9, .46

pos\_hint: {"x":0.05, "top": .6}

gl:gl

GridLayout:

id: gl

cols: 1

spacing: 0, root.width//25

size\_hint\_y: None

""")

self.add\_widget(self.search\_output)

self.search\_input = TextInput(padding=(5, 5), size\_hint=(0.69, 0.1),

pos\_hint={"x": .05, "top": .84}, multiline=False)

self.do\_search = ImageButton(src="search2.png", size\_hint=(.2, .2),

pos\_hint={"x": .75, "top": .89})

self.do\_search.on\_press = self.find\_by\_tag

self.add\_widget(self.search\_input)

self.add\_widget(self.do\_search)

self.close = Button(text="Close", size\_hint=(0.8, 0.1),

pos\_hint={"x": .1, "top": .1},

background\_color=(179/255, 220/255, 111/255, 1))

self.add\_widget(self.close)

self.dd = DropDown(max\_height=Window.height//3)

self.mainbutton = Button(text="Select Tags:", size\_hint=(0.9, 0.1),

pos\_hint={"x": .05, "top": .72},

background\_normal="selection.png")

self.mainbutton.bind(on\_release=self.dd.open)

self.add\_widget(self.mainbutton)

tag = curs.execute("SELECT tags from Uniques").fetchall()

for i in sorted(tag[0][0][1:-1].split(",")):

i = i.strip(" ' ").capitalize()

btn = Button(text=f"{i}", text\_size=(self.width, None),

size\_hint\_y=None, height=Window.height//15,

on\_release=lambda buttn: self.change\_input(buttn.text),

background\_color=(12/255, 63/255, 82/255, 1),)

self.dd.add\_widget(btn)

def change\_input(self, x):

self.search\_input.text += f"{x},"

def find\_by\_tag(self):

self.name = []

tags = curs.execute("SELECT tags FROM CookHub").fetchall()

for i in tags:

for j in self.search\_input.text.split(",")[:-1]:

if j.lower() in str(i):

try:

x = curs.execute('SELECT name FROM'

f' CookHub where tags="{str(i)[2:-3]}"').fetchall()

if x[0][0] not in self.name:

self.name.append(x[0][0])

except:

continue

try:

for i in self.id\_s:

self.search\_output.gl.remove\_widget(i)

except:

pass

self.search\_output.gl.height = 0

for i in self.name:

btn = RecipeButton(rcp=i.capitalize(),

text\_size=self.t\_s, font\_size=self.f\_s)

self.id\_s.append(btn)

self.search\_output.gl.add\_widget(btn)

self.search\_output.gl.height += 50

class ByName(FloatLayout):

f\_s = (Window.width + Window.height)//100 # Static font\_size for dynamic layout

t\_s = (Window.width//6, None) # Static text\_size for dynamic layout

id\_s = []

def \_\_init\_\_(self):

super(ByName, self).\_\_init\_\_()

self.add\_widget(Builder.load\_string("""

Label:

text:"Enter Name:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":.97}

font\_size:(root.width+root.height)//12

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

"""))

self.search\_output = Builder.load\_string("""

ScrollView:

size\_hint: 0.9, .6

pos\_hint: {"x":0.05, "top": .72}

gl:gl

GridLayout:

id: gl

cols: 1

spacing: 0, root.width//25

size\_hint\_y: None

""")

self.add\_widget(self.search\_output)

self.search\_input = TextInput(padding=(5, 5), size\_hint=(0.69, 0.1),

pos\_hint={"x": .05, "top": .84}, multiline=False)

self.do\_search = ImageButton(src="search2.png", size\_hint=(.2, .2),

pos\_hint={"x": .75, "top": .89})

self.do\_search.on\_press = self.find\_by\_name

self.add\_widget(self.search\_input)

self.add\_widget(self.do\_search)

self.close = Button(text="Close", size\_hint=(0.8, 0.1),

pos\_hint={"x": .1, "top": .1},

background\_color=(179/255, 220/255, 111/255, 1))

self.add\_widget(self.close)

def find\_by\_name(self):

name = curs.execute("SELECT name from CookHub").fetchall()

try:

for i in self.id\_s:

self.search\_output.gl.remove\_widget(i)

except:

pass

self.search\_output.gl.height = 0

for i in name:

for j in self.search\_input.text.split(" "):

if j.lower() in i[0] and j != "" and j != " ":

btn = RecipeButton(rcp=i[0].capitalize(),

text\_size=self.t\_s, font\_size=self.f\_s)

self.id\_s.append(btn)

self.search\_output.gl.add\_widget(btn)

self.search\_output.gl.height += 50

class Search(Screen, FloatLayout):

def \_\_init\_\_(self, \*\*kwargs):

super(Search, self).\_\_init\_\_(\*\*kwargs)

self.add\_widget(ButtonPanel().icons\_box)

self.add\_widget(Builder.load\_string(logo))

show1 = ByName()

self.by\_name\_popup = Popup(title="Search by Name",

content=show1,

title\_color=(224/255, 10/255, 20/255, 1),

size\_hint=(.9, .9),

background="bg.png")

show1.close.on\_press = self.by\_name\_popup.dismiss

self.by\_name.on\_press = self.by\_name\_popup.open

show2 = ByTag()

self.by\_tag\_popup = Popup(title="Search by Tag",

content=show2,

title\_color=(224/255, 10/255, 20/255, 1),

size\_hint=(.9, .9),

background="bg.png")

show2.close.on\_press = self.by\_tag\_popup.dismiss

self.by\_tag.on\_press = self.by\_tag\_popup.open

show3 = ByIngredient()

self.by\_ingredient\_popup = Popup(title="Search by Ingredient",

content=show3,

title\_color=(224/255, 10/255, 20/255, 1),

size\_hint=(.9, .9),

background="bg.png")

show3.close.on\_press = self.by\_ingredient\_popup.dismiss

self.by\_ingredient.on\_press = self.by\_ingredient\_popup.open

sm.add\_widget(Search(name="Search"))

class ShoppingList(Screen, FloatLayout): # ShoppingList Page

id\_s = []

popup\_window = Popup(title="Buy Ingredients",

title\_color=(224/255, 10/255, 20/255, 1),

size\_hint=(0.9, 0.8),

background="bg.png")

def \_\_init\_\_(self, \*\*kwargs):

super(ShoppingList, self).\_\_init\_\_(\*\*kwargs)

self.add\_widget(Builder.load\_string(logo))

self.ShoppingLayout = Builder.load\_string("""

FloatLayout:

gl: gl

s\_gl: s\_gl

s\_gl1: s\_gl1

s\_gl2: s\_gl2

size\_hint: 1, 0.8

pos\_hint: {"x":0, "top":0.8}

Label:

text:"My Shopping Lists:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.98}

font\_size:(root.width+root.height)//30

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

GridLayout:

id: gl

cols: 1

FloatLayout:

size\_hint: 1,1

ScrollView:

size\_hint: 0.9, 0.7

pos\_hint: {"x":0.05, "top":0.83}

canvas.before:

GridLayout:

id: s\_gl

cols: 2

size\_hint\_y: None

spacing: root.width//25, 0

GridLayout:

id: s\_gl1

size\_hint: 0.2, None

spacing: 0, root.width//15

cols: 1

GridLayout:

id: s\_gl2

size\_hint: 0.8, None

spacing: 0, root.width//15

cols: 1

""")

self.f\_s = (Window.width + Window.height)//100 # Static font\_size for dynamic layout

self.t\_s = (Window.width//6, None) # Static text\_size for dynamic layout

self.shopping\_listing()

self.ShoppingLayout.gl.add\_widget(ButtonPanel().icons\_box)

self.add\_widget(self.ShoppingLayout)

def shopping\_listing(self):

self.show = ShowingList()

self.popup\_window.content = self.show

self.ShoppingLayout.s\_gl.height = len(shopping\_list)\*50

self.ShoppingLayout.s\_gl1.height = len(shopping\_list)\*50

self.ShoppingLayout.s\_gl2.height = len(shopping\_list)\*50

for i in self.id\_s:

self.ShoppingLayout.s\_gl1.remove\_widget(i[0])

self.ShoppingLayout.s\_gl2.remove\_widget(i[1])

self.id\_s = []

for i in sorted(shopping\_list):

delete\_btn = ImageButton(src="delete2.png", id=f"{i}")

self.ShoppingLayout.s\_gl1.add\_widget(delete\_btn)

lists\_btn = Button(text=f"{i}",

font\_size=self.f\_s, text\_size=self.t\_s,

background\_color=(92/255, 146/255, 48/255, 1),

on\_press=lambda btn: self.popup\_open(btn.text))

self.id\_s.append([delete\_btn, lists\_btn])

self.ShoppingLayout.s\_gl2.add\_widget(lists\_btn)

def popup\_open(self, x):

show = ShowingList(x)

self.popup\_window.content = show

show.close.on\_press = self.popup\_window.dismiss

self.popup\_window.open()

shoppinglist\_screen = ShoppingList(name="SL")

sm.add\_widget(shoppinglist\_screen)

class VirtualFridge(Screen, FloatLayout): # Virtual Fridge Page

id\_s = []

def \_\_init\_\_(self, \*\*kwargs):

super(VirtualFridge, self).\_\_init\_\_(\*\*kwargs)

self.add\_widget(Builder.load\_string(logo))

self.FridgeLayout = Builder.load\_string("""

FloatLayout:

gl: gl

ai: ai

s\_gl: s\_gl

s\_gl1: s\_gl1

s\_gl2: s\_gl2

size\_hint: 1, 0.8

pos\_hint: {"x":0, "top":0.8}

ImageButton:

id: ai

source: "add.png"

size\_hint: (.2, .4)

pos\_hint:{"top": 0.47, "x":0.05}

Label:

text: "Add your ingredients"

font\_size: (root.width+root.height)//45

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

RoundedRectangle:

pos: self.pos

size: self.size

radius: [17,17,17,17]

size\_hint: (0.65, 0.11)

pos\_hint:{"top": 0.325, "x":0.28}

Label:

text:"My Ingredients:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.98}

font\_size:(root.width+root.height)//30

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

GridLayout:

id: gl

cols: 1

FloatLayout:

size\_hint: 1,1

ScrollView:

size\_hint: 0.9, 0.6

pos\_hint: {"x":0.05, "top":0.83}

canvas.before:

GridLayout:

id: s\_gl

cols: 2

size\_hint\_y: None

spacing: root.width//25, 0

GridLayout:

id: s\_gl1

size\_hint: 0.2, None

spacing: 0, root.width//15

cols: 1

GridLayout:

id: s\_gl2

size\_hint: 0.8, None

spacing: 0, root.width//15

cols: 1

""")

self.f\_s = (Window.width + Window.height)//100 # Static font\_size for dynamic layout

self.t\_s = (Window.width//6, None) # Static text\_size for dynamic layout

self.ingredients\_listing()

self.FridgeLayout.gl.add\_widget(ButtonPanel().icons\_box)

self.add\_widget(self.FridgeLayout)

self.show = AddingIngredients()

self.popup\_window = Popup(title="Add Ingredients",

title\_color=(224/255, 10/255, 20/255, 1),

content=self.show, size\_hint=(0.9, 0.7),

background="bg.png")

self.FridgeLayout.ai.on\_press = self.popup\_window.open

def ingredients\_listing(self):

self.FridgeLayout.s\_gl.height = len(fridge\_list)\*50

self.FridgeLayout.s\_gl1.height = len(fridge\_list)\*50

self.FridgeLayout.s\_gl2.height = len(fridge\_list)\*50

for i in self.id\_s:

self.FridgeLayout.s\_gl1.remove\_widget(i[0])

self.FridgeLayout.s\_gl2.remove\_widget(i[1])

self.id\_s = []

for i in sorted(fridge\_list):

delete\_btn = ImageButton(src="delete.png", id=f"{i}")

self.FridgeLayout.s\_gl1.add\_widget(delete\_btn)

ingredient\_btn = Button(text=f"{i}",

font\_size=self.f\_s, text\_size=self.t\_s,

background\_color=(92/255, 146/255, 48/255, 1))

self.id\_s.append([delete\_btn, ingredient\_btn])

self.FridgeLayout.s\_gl2.add\_widget(ingredient\_btn)

virtualfridge\_screen = VirtualFridge(name="VF")

sm.add\_widget(virtualfridge\_screen)

class Favourite(Screen, FloatLayout): # Displays favourite list

ids = []

def \_\_init\_\_(self, \*\*kwargs):

super(Favourite, self).\_\_init\_\_(\*\*kwargs)

self.add\_widget(Builder.load\_string(logo))

self.add\_widget(ButtonPanel().icons\_box)

self.add\_widget(Builder.load\_string("""

Label:

text:"Your Favourites:"

size\_hint:(0.9, 0.1)

pos\_hint: {"x":0.05, "top":0.77}

font\_size:(root.width+root.height)//10

color: (0.8, 0.1, 0.4, 1)

canvas.before:

Color:

rgba: (179/255, 220/255, 111/255, 1)

Rectangle:

size: self.size

pos: self.pos

"""))

self.favourite\_listing = Builder.load\_string("""

ScrollView:

gl: gl

size\_hint: 0.9, 0.4

pos\_hint: {"x": 0.05, "top": 0.63}

GridLayout:

id: gl

size\_hint\_y: None

cols: 1

spacing: 0, root.height \* 0.03

""")

self.add\_widget(self.favourite\_listing)

self.t\_s = (self.width\*1.5, None)

self.f\_s = Window.width//60

self.listing()

def listing(self):

try:

for i in self.ids:

self.favourite\_listing.gl.remove\_widget(i)

self.ids = []

except:

pass

self.favourite\_listing.gl.height = len(favourite\_list)\*Window.height\*0.08

for i in range(len(favourite\_list)):

self.ids.append(RecipeButton(

rcp=favourite\_list[i], text\_size=self.t\_s, font\_size=self.f\_s))

self.favourite\_listing.gl.add\_widget(self.ids[i])

favourite\_screen = Favourite(name="F")

sm.add\_widget(favourite\_screen)

class Home(Screen, FloatLayout): # Home Page

def \_\_init\_\_(self, \*\*kwargs):

super(Home, self).\_\_init\_\_(\*\*kwargs)

self.add\_widget(Builder.load\_string(logo))

self.add\_widget(ButtonPanel().icons\_box)

name = curs.execute("SELECT name from CookHub").fetchall()

curs.execute("SELECT max(rowid) from CookHub")

n\_rows = int(curs.fetchall()[0][0])

self.recommendation\_list = Builder.load\_string("""

ScrollView:

gl: gl

size\_hint: (0.9, 0.55)

pos\_hint: {}

bar\_width: root.width//50

bar\_color: (1,1,1,1)

GridLayout:

size\_hint\_y: None

spacing: root.width\*0.03, root.height\*0.04

id: gl

cols: 3

height: {}

""".format({"x": 0.05, "top": 0.75}, (n\_rows/3+1)\*110))

for i in range(n\_rows):

self.recommendation\_list.gl.add\_widget(RecipeButton(rcp=name[i][0].capitalize()))

self.add\_widget(self.recommendation\_list)

sm.add\_widget(Home(name="Home"))

class CookHubApp(App): # Initializing Class

Window.size = (230, 470)

Window.clearcolor = (233/255, 169/255, 63/255, 1)

def build(self):

sm.current = "Home"

return sm

if \_\_name\_\_ == "\_\_main\_\_":

CookHubApp().run()