Task 2 Code

Prototype for the proposed digital system

The health advice group has commissioned our software development company to develop a digital system. The health advice group offers advice on how to deal with extreme weather temperatures, information on environmental health conditions and seasonal allergies as well as risk assessments for home environments.

The health advice group requirements are that the proposed solution:

* Weather forecasting to inform health decisions
* Access to a dashboard for monitoring air quality data
* Advice on how to deal with health matters affected by weather and environmental conditions.
* Personalised health advice based on location
* Accessibility features to support a wide range of user needs
* A personal health tracking tool
* Visual representations of data

APP – Window 1/Home Page

import webbrowser

from pathlib import Path

# !!! Important: Most of the code will be repeated and thus will not be commented on !!!

# from tkinter import \*

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage #Imports

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path(r"C:**\U**sers**\y**our-username**\O**neDrive\Documents**\G**ithub**\O**CC-Mock\build**\a**ssets**\f**rame0") #Change the path to wherever frame0 is located on your pc.

def relative\_to\_assets(path: str) -> Path:

    return ASSETS\_PATH / Path(path) #Returns the relevant path so assets can be used.

def callback\_page(url):

    webbrowser.open\_new(r'C:**\U**sers**\y**our-username**\O**neDrive\Documents**\G**ithub**\O**CC-Mock\build\Dashboard.py') #Opens up the Dashboard file.

username = ""

pw = ""

window = Tk()

window.geometry("862x519")

window.configure(bg = "#24A0FA")

canvas = Canvas( #Creates Canvas (Relevant to figma).

    window,

    bg = "#24A0FA",

    height = 519,

    width = 862,

    bd = 0,

    highlightthickness = 0,

    relief = "ridge"

)

canvas.place(x = 0, y = 0) #Places canvas.

canvas.create\_rectangle(

    430.9999999999999,

    0.0,

    861.9999999999999,

    519.0,

    fill="#F8F8F6",

    outline="")

button\_image\_1 = PhotoImage(

    file=relative\_to\_assets("button\_1.png")) #Gets the image for the button, will essentially be what the button will look like.

button\_1 = Button( #Creates button.

    image=button\_image\_1,

    borderwidth=0,

    highlightthickness=0,

    command=lambda: callback\_page (r'C:**\U**sers**\y**our-username**\O**neDrive\Documents**\G**ithub**\O**CC-Mock\build\Dashboard.py'), #Once the button is clicked it will open up this file. Make sure to change the path!!!

    relief="flat"

)

button\_1.place( #Places button.

    x=556.9999999999999,

    y=401.0,

    width=180.0,

    height=55.0

)

canvas.create\_text( #Creates some text.

    39.999999999999886,

    10.000000000000007,

    anchor="nw",

    text="Home Page",

    fill="#FFFFFF",

    font=("RubikRoman Bold", 32 \* -1)

)

canvas.create\_text(

    481.9999999999999,

    74.0,

    anchor="nw",

    text="Enter your details:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_rectangle( #Creates a rectangle.

    39.999999999999886,

    57.00000000000001,

    99.99999999999989,

    62.00000000000001,

    fill="#333333",

    outline="")

entry\_image\_1 = PhotoImage(

    file=relative\_to\_assets("entry\_1.png")) #Allows for this image to be used in the program.

entry\_bg\_1 = canvas.create\_image( #Creates image.

    650.4999999999999,

    167.5,

    image=entry\_image\_1

)

entry\_1 = Entry(

    bd=0,

    bg="#24A0FA",

    fg="#000716",

    highlightthickness=0

)

entry\_1.place( #Places image.

    x=489.9999999999999,

    y=137.0,

    width=321.0,

    height=59.0

)

entry\_image\_2 = PhotoImage(

    file=relative\_to\_assets("entry\_2.png"))

entry\_bg\_2 = canvas.create\_image(

    650.4999999999999,

    248.5,

    image=entry\_image\_2

)

entry\_2 = Entry(

    bd=0,

    bg="#24A0FA",

    fg="#000716",

    highlightthickness=0

)

entry\_2.place(

    x=489.9999999999999,

    y=218.0,

    width=321.0,

    height=59.0

)

canvas.create\_text(

    39.999999999999886,

    94.0,

    anchor="nw",

    text="In this app you will find weather forecasts, reports and advice!\n\n",

    fill="#FFFFFF",

    font=("RubikRoman Bold", 14 \* -1)

)

canvas.create\_text(

    39.999999999999886,

    174.0,

    anchor="nw",

    text="Try the tracking tool today!",

    fill="#FFFFFF",

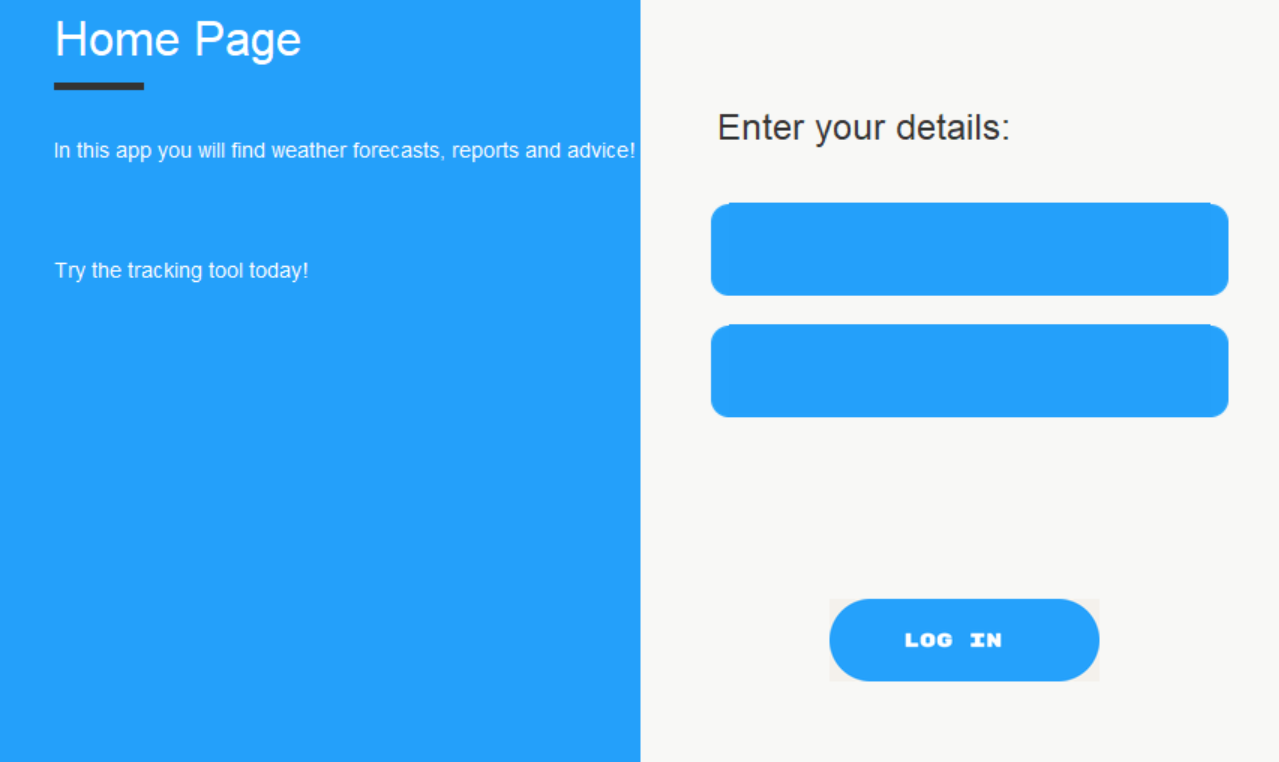
    font=("RubikRoman Bold", 14 \* -1)

)

window.resizable(False, False)

window.mainloop()

Output



This is the login screen; users can create and input usernames and passwords.

APP – Window 2/Dashboard

import webbrowser

from pathlib import Path

# !!! Important: Most of the code will be repeated and thus will not be commented on !!!

# from tkinter import \*

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage #Imports

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path(r"C:**\U**sers**\y**our-username**\O**neDrive\Documents**\G**ithub**\O**CC-Mock\build**\a**ssets**\f**rame1") #Change this path to wherever frame1 is located on your pc.

def relative\_to\_assets(path: str) -> Path:

    return ASSETS\_PATH / Path(path)

def callback\_page(url):

    webbrowser.open\_new(r'C:**\\**Users**\\**your-username**\\**OneDrive**\\**Documents**\\**Github**\\**OCC-Mock**\\**build**\\**Home-Page.py') #Will open the home-page file, will be relevant when clicking the buttons available. Make sure to change this path!!!

def callback\_web(url):

    webbrowser.open\_new(url) #Does the same action as line 16 but instead opens up your browser to direct to a relevant website.

window = Tk()

window.geometry("862x519") #Size of window.

window.configure(bg = "#FFFFFF")

canvas = Canvas( #Creates a canvas (all the code below is for the figma designs).

    window,

    bg = "#FFFFFF",

    height = 519,

    width = 862,

    bd = 0,

    highlightthickness = 0,

    relief = "ridge"

)

canvas.place(x = 0, y = 0)

canvas.create\_rectangle( #Creates a rectangle and places it in the relevant place.

    431.0,

    0.0,

    862.0,

    519.0,

    fill="#F8F8F6",

    outline="")

button\_image\_1 = PhotoImage(

    file=relative\_to\_assets("button\_1.png")) #This will be used in the python files as a functioning button, the image is what it will look like essentially.

button\_1 = Button( #Creates a button.

    image=button\_image\_1,

    borderwidth=0,

    highlightthickness=0,

    command=lambda: callback\_page (r'C:**\\**Users**\\**your-username**\\**OneDrive**\\**Documents**\\**Github**\\**OCC-Mock**\\**build**\\**Home-Page.py'), #This line means that if a the button is clicked it will direct to the other python file "Home-Page". Make sure to change this path!!!

    relief="flat"

)

button\_1.place( #Sets the place of the button.

    x=631.0,

    y=388.0,

    width=180.0,

    height=55.0

)

button\_image\_2 = PhotoImage(

    file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

    image=button\_image\_2,

    borderwidth=0,

    highlightthickness=0,

    command=lambda: callback\_web ('https://www.gov.uk/government/news/cold-health-alerts-issued-by-ukhsa-and-the-met-office'),

    relief="flat"

)

button\_2.place(

    x=631.0,

    y=457.0,

    width=180.0,

    height=55.0

)

canvas.create\_text( #Creates some text.

    107.0,

    5.0,

    anchor="nw",

    text="Weather Dashboard",

    fill="#333333",

    font=("RubikRoman Bold", 32 \* -1)

)

canvas.create\_text(

    21.0,

    211.0,

    anchor="nw",

    text="19 °C",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    19.0,

    246.0,

    anchor="nw",

    text="Monday:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    478.0,

    5.0,

    anchor="nw",

    text="Week:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    70.0,

    285.0,

    anchor="nw",

    text="Rain: 40%",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_rectangle(

    107.0,

    49.0,

    167.0,

    54.0,

    fill="#24A0FA",

    outline="")

image\_image\_1 = PhotoImage(

    file=relative\_to\_assets("image\_1.png")) #Allows the relevant image to be used in the program.

image\_1 = canvas.create\_image( #Creates the image.

    196.0,

    139.0,

    image=image\_image\_1

)

image\_image\_2 = PhotoImage(

    file=relative\_to\_assets("image\_2.png"))

image\_2 = canvas.create\_image(

    39.0,

    299.0,

    image=image\_image\_2

)

image\_image\_3 = PhotoImage(

    file=relative\_to\_assets("image\_3.png"))

image\_3 = canvas.create\_image(

    217.0,

    423.0,

    image=image\_image\_3

)

canvas.create\_rectangle(

    477.0,

    44.0,

    559.0,

    190.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_rectangle(

    606.0,

    44.0,

    688.0,

    190.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_rectangle(

    748.0,

    44.0,

    830.0,

    190.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_text(

    489.0,

    52.0,

    anchor="nw",

    text="Mon:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    622.0,

    54.0,

    anchor="nw",

    text="Tue:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    761.0,

    54.0,

    anchor="nw",

    text="Wed:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

image\_image\_4 = PhotoImage(

    file=relative\_to\_assets("image\_4.png"))

image\_4 = canvas.create\_image(

    517.0,

    131.0,

    image=image\_image\_4

)

image\_image\_5 = PhotoImage(

    file=relative\_to\_assets("image\_5.png"))

image\_5 = canvas.create\_image(

    647.0,

    132.0,

    image=image\_image\_5

)

image\_image\_6 = PhotoImage(

    file=relative\_to\_assets("image\_6.png"))

image\_6 = canvas.create\_image(

    788.0,

    129.0,

    image=image\_image\_6

)

canvas.create\_rectangle(

    478.0,

    223.0,

    560.0,

    369.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_rectangle(

    8.0,

    7.0,

    90.0,

    153.0,

    fill="#E5E5E5",

    outline="")

canvas.create\_rectangle(

    607.0,

    223.0,

    689.0,

    369.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_rectangle(

    472.0,

    382.0,

    554.0,

    510.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_rectangle(

    744.0,

    223.0,

    826.0,

    369.0,

    fill="#FFFFFF",

    outline="")

canvas.create\_text(

    489.0,

    232.0,

    anchor="nw",

    text="Thu:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    486.0,

    388.0,

    anchor="nw",

    text="Sun:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    627.0,

    232.0,

    anchor="nw",

    text="Fri:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

canvas.create\_text(

    762.0,

    232.0,

    anchor="nw",

    text="Sat:",

    fill="#333333",

    font=("RubikRoman Bold", 24 \* -1)

)

image\_image\_7 = PhotoImage(

    file=relative\_to\_assets("image\_7.png"))

image\_7 = canvas.create\_image(

    519.0,

    295.0,

    image=image\_image\_7

)

image\_image\_8 = PhotoImage(

    file=relative\_to\_assets("image\_8.png"))

image\_8 = canvas.create\_image(

    647.0,

    306.0,

    image=image\_image\_8

)

image\_image\_9 = PhotoImage(

    file=relative\_to\_assets("image\_9.png"))

image\_9 = canvas.create\_image(

    786.0,

    302.0,

    image=image\_image\_9

)

image\_image\_10 = PhotoImage(

    file=relative\_to\_assets("image\_10.png"))

image\_10 = canvas.create\_image(

    513.0,

    456.0,

    image=image\_image\_10

)

button\_image\_3 = PhotoImage( #Behaves the same way as the previous button/s.

    file=relative\_to\_assets("button\_3.png"))

button\_3 = Button(

    image=button\_image\_3,

    borderwidth=0,

    highlightthickness=0,

    command=lambda: callback\_page (r'C:**\\**Users**\\**your-username**\\**OneDrive**\\**Documents**\\**Github**\\**OCC-Mock**\\**build**\\**Home-Page.py'), #Make sure to change this path!!!

    relief="flat"

)

button\_3.place(

    x=17.0,

    y=11.0,

    width=62.0,

    height=57.0

)

button\_image\_4 = PhotoImage(

    file=relative\_to\_assets("button\_4.png"))

button\_4 = Button(

    image=button\_image\_4,

    borderwidth=0,

    highlightthickness=0,

    command=lambda: callback\_web ('https://www.metoffice.gov.uk/weather/warnings-and-advice/seasonal-advice/health-wellbeing/stay-well-in-winter/stay-well-in-winter'), #This behaves in the same way as the other actions but instead redirects you to a web page.

    relief="flat"

)

button\_4.place(

    x=16.0,

    y=85.0,

    width=63.0,

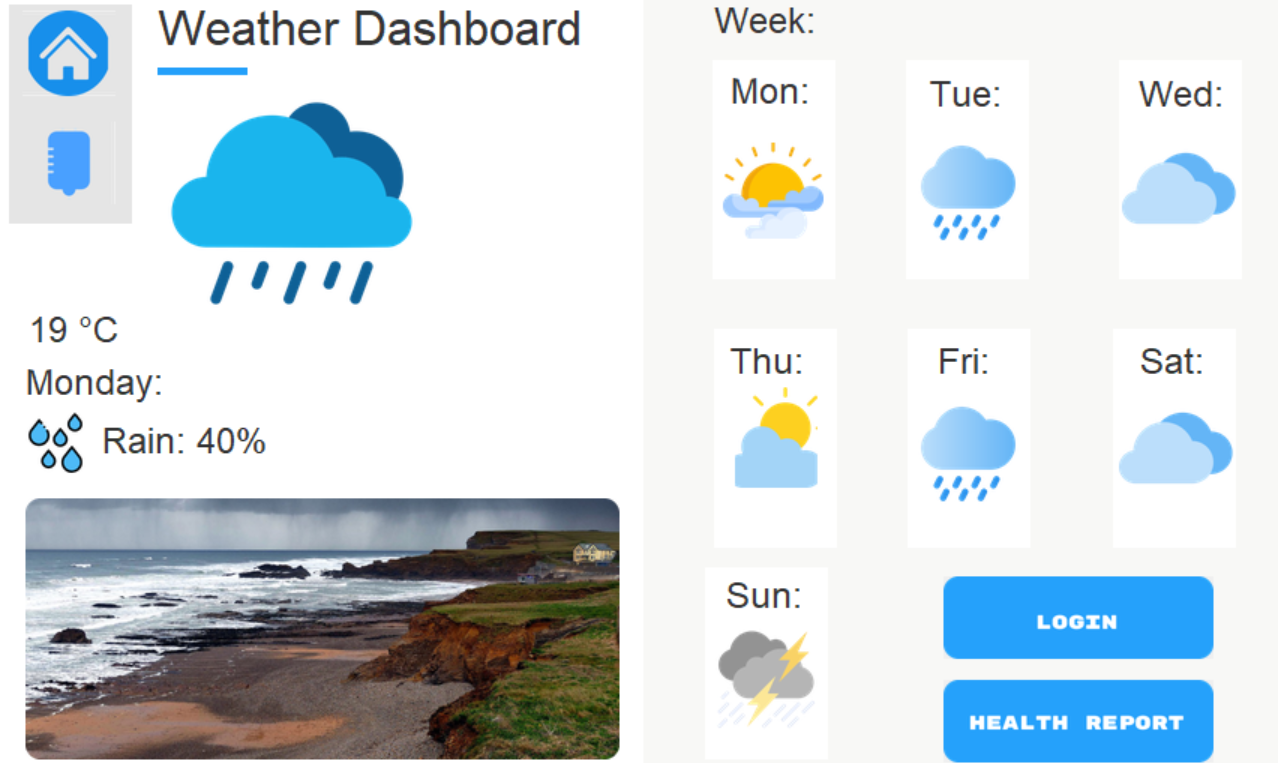
    height=55.0

)

window.resizable(False, False)

window.mainloop()

Output



This is the dashboard page and includes health reports and information on the current day.

APP – DB

import sqlite3

target\_name = "Fin"

conn = sqlite3.connect("users.db") #Connecting to db (Database)

cursor = conn.cursor() #Connect the cursor instance to use methods from sqlite such as fetching data from the result sets of queries.

def search\_data(id, name, city, age):

    cursor.execute('CREATE TABLE user(user\_id n(5), name char (30), city char (35), age decimal(7,2));') #Creates table along with columns.

    cursor.execute("INSERT INTO user VALUES (4, 'Findlay', 'Billingham', 1)")

    cursor.execute("""

                   INSERT INTO user(user\_id, name, city, age)

                   VALUES (?,?,?,?)

                   """, (id, name, city, age)) #Inserts new data (the parameters) into the table.

    rows = cursor.execute("SELECT user\_id, name, city, age FROM user").fetchall()

    search = cursor.execute(

        'SELECT user\_id, name, city, age FROM user WHERE name =?',(target\_name,), #Selects specific name stored.

        ).fetchall() #Selects the columns from the salesman table and fetches them all.

    conn.commit() #Commits to changes.

    print('Data entered...')

    conn.close() #Closes connection to DB.

    print(rows)

    print(search)

    if (conn):

        conn.close()

        print('\nDatabase closed...') #If the connection is closed it will print this message.

search\_data(4, 'Fin', 'Billingham', 1) #Inputs data through parameters to table.

Output

A screenshot of a computer

Description automatically generated

Given the time that was available for the project, I had to use SQLlite3 in python which allows for serverless databases to be created, saving a lot of time for the project to work on other aspects.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date used | Last change | Changes | Comments |
| Ver 1 | 31/01/2024 | 19/02/2024 | Colour scheme, added text boxes and inactive button. | This was one of the first version worked on, however it was decided to work on another version that would include more functionality. |
| Ver 2 | 31/01/2024 | 19/02/2024 | Font, text size. | A previous version with the new colour scheme that unfortunately messed up the text. |
| Ver 3 | 31/01/2024 | 19/02/2024 | Widgets, object placement. | This was a previous version where I was working on the dashboard, the conventions used have improved in the final build as well as includes more content. |
| Ver 4 | 31/01/2024 | 19/02/2024 | Colour scheme, no button functionality. | This was a version with the original colour scheme which was shortly changed after developing. |