Assignment 3

Note:

The output of this exercise should be a single Flask app with multiple routes. There are 15 subtasks which are all worth one point. The maximum possible score is 13, meaning that there is a tolerance of two points for small mistakes.

Part 1: Getting started

1. Create your own flask app. Return a welcome message when the user opens it
2. Make the app run on port 80
3. Create a second route called "/about". Render a html page with some basic information about yourself or any other subject. Use multiple html elements, but in any case include: a hyperlink, a heading, a small paragraph of text, and an image
4. Create a third route called "/weather/*location*" where location is a city name that can entered as a variable. Display a short sentence on that page describing the weather of the specified location, using three attributes from the weather API (e.g. one could be the current temperature)

Part 2: Hometown route

1. Create a fourth route called "/myhometown". Create an information page about your hometown using various html attributes. Specifically include an ordered or unordered list as well as a html table. (Hint: we haven't covered these in class, take a look at the w3schools.com page for some good explanations)
2. Back in your weather route, create a redirect to "/hometown" where users get redirected to your hometown page when they enter its name. Instead of manually entering the link use the Flask's *url_for* function
3. In your index page (welcome page), now add a list with the names of your website's subpages and make them clickable links to get to these pages. Make sure that your weather route already has an example filled in.

   Hint: the *url_for* function can and should be used in the html file to direct to other pages (read about it here)
4. *Manually download* an image of your hometown (or another city if you prefer), add it to your flask project and display it on your myhometown route. (Hint: In flask, files that don't change, like images, are generally saved in a subfolder called 'static')

Part 3: Dynamically rendering images

1. Unsplash is a website with over 2 million royalty free images. They also have an API. Register as a developer at https://unsplash.com/developers and get your own access key and secret key. To do this you will have to submit a short description of the application you will be building. Write a few sentences, but don't worry, this will not be manually checked for approval.
2. In a separate file (not your flask app, to make testing easier) write a function that:
    a. Takes the argument "search_query" as an input argument
    b. Searches for the *search_query* term on the Unsplash API and returns the image URL for the first result of the search (careful: the JSON file returned is quite big and contains some lists, you will have to do some subscripting)
3. In the same separate file, write a second function that
    a. takes the inputs "img_url", "folder_prefix" and "filename".
    b. Downloads the image and saves it in the specified folder with the selected file name
4. In your flask app, create a new route that is called "/homescreen/*name*/*location*", which renders a template that:
    a. Renders an image of the location they are currently at: You can do this either by passing a link to the template or by passing the filename from a file saved in your static folder. Use the function(s) you created in the previous subtask, either by copy pasting them into the flask app or by importing them.
    b. Displays (a) the name of the user, (b) the weather of the city they are currently in and (c) the current time. For name and weather, use the values entered in the URL (e.g. name="Anna", location="Zurich")

Part 4: Custom Homescreen based on user input in a form

1. Create your own form class (similar as in class) that takes the inputs ‘Name’, ‘Location’ and ‘Main goal of the day’
2. Create a new route called “/custom-homescreen” and render your form that you just created. On the same page, also include a small text that changes when the user successfully submits the form, using the values they entered, e.g. “[name] from [location] says that his/her main goal for today is [goal]”
3. Change your route such that instead of reloading the page with the changing small text, the user is directed to your “/homescreen/…” route with the inputs being adjusted from your form. You may have to adjust your homescreen route to also include the content of the *main goal* field.