HW 5 - Flask

In this lab we'll be implementing a simple survey web app using Flask.

The general concepts we'll be covering are:

- Routing
- Templating
- Passing Parameters
- Sessions

Requirements

Creating a login page

1. Open the 'login.html' template in your text editor
    a. Make this file an extension of the 'base.html' template
    b. Make sure the HTML in this file is replacing the block called 'content' in the 'base.html' template
    c. Modify the form attributes so the form would send a post request to the view function /login endpoint
    d. Add input and labels that would can be linked to the view function. Hint: name attribute

Creating a session

2. Open the 'views.py' file in your text editor
    a. Find the route for '/login'
        i. Make sure in the decorator you specify that this function takes get/posts methods
    b. Within the 'login()' function, if the request method is POST
        i. Set the session's 'username' key to whatever the user entered for their username
        ii. Also set the session's 'email' key to whatever the user entered for their email
    c. Now if you go to /index route
        i. Check the user is already in session, if yes
        ii. Direct user to take the survey at survey.html. You have to pass in the username variable

Survey Form

3. Open the 'survey.html' template in your text editor
    a. Make this file an extension of the 'base.html' template
    b. The HTML in this file for the form with the id of 'logout' is currently replacing the entire contents of the <header> tag in the 'base.html' template. Using the super() function, pull in the original contents of the <header> tag from the 'base.html' template
    c. Make sure the HTML in this file from the <h1> and down is replacing the block called 'content' in the 'base.html' template

Creating an Ajax request with the user's form inputs from 'survey.html'

4. Open the 'interaction.js' script in your text editor
    a. Notice that the button from the 'survey.html' template with the id 'submit-survey' has a click event binding
    b. Also notice that there are variable declarations for each of the form input fields
    c. Create a $.post() Ajax request within this 'click' event handler
        i. The url for this Ajax request should point to 'submit-survey'
        ii. The data parameter of this Ajax request should be an object whose key-value pairs correspond to the variables for each form input field
        iii. The success function for this Ajax request should set the innerHTML of document.body.parentNode to the response data object. Hint: Jquery $("html")

Rendering the survey results via Flask

5. Go back to the 'views.py' file in your text editor
    a. Find the route for '/submit-survey'
    b. You'll notice that an empty object has been assigned to the variable name 'surveyResponse'
    c. You'll also notice that 'fe-before' and 'fe-after' keys in the 'surveyResponse' object have been assigned values that correspond to values in the data object we passed in from step 5 above.
        i. Assign the keys 'color', 'food', and 'vacation' for the 'surveyResponse' object to corresponding values from the passed-in data object in a similar fashion
        ii. Pass in both the username and survey responses to the result.html
6. You should now be able to do the following:
    a. Log into this simple survey web app with a username and email
    b. Take the survey

    c. See the survey results displayed
    d. Email yourself the survey results object

7. **Extra Credit (1 point)**
    a. Display a conditional message on results.html depending on the survey responses.
    b. Example, if the user improved, display one message. If not, display another./