Module 1 Homework: survey server

Due Wednesday October 10 @11:15 AM
CSCI 5117, Fall 2018

Your task is to create a small dynamic flask app for collecting survey responses, and to host it at heroku.

- Get your repo here: https://classroom.github.com/a/1CiUNFwJ
- This task is to be done individually
- The README file that comes with your git repository contains fields for you to fill out

Task: Host a Custom Survey

Your app is designed to collect survey responses. It has the following requirements:

App Design

Your survey app must have the following URLs/pages:

- / -- the root page describes the survey and asks the user to consent to participate. It has two buttons at the bottom: "consent" (go to /survey) and "decline" (go to /decline).
- /survey -- asks the user a few questions, then a "next" button (go to /thanks). The input types must include:
  - text input -- this field is "required" and has a minimum length of 3 characters. The user cannot proceed without filling it out -- use html5 validation.
  - a group of 3 or more radio buttons
  - select box with 3 or more options
  - checkbox
  - finally, there must be one "conditional" field of type textarea that appears or disappears depending on the state of the checkbox input.
- /decline -- a page that says "thanks anyway" or something like that
- /thanks -- says thank you to the user for completing the survey

The pages above must share a template with a header that is consistent throughout.

The app must be styled using https://purecss.io/ (with additional custom CSS, if you wish). The survey form fields should be styled according to https://purecss.io/forms/.

The app must have one more URL:

- /api/results -- This URL is an API endpoint that returns a JSON representation of all the survey responses that you have collected to date (see below for data storage requirements). By default, it should return all responses (natural order by primary key is fine). This endpoint should support the following query parameter:
  - ?reverse=true -- if the user includes this query parameter the results are instead ordered by most recent first

Data Storage

Your postgres database only needs a single table to store survey responses. It should include a field for each survey question, as well as an auto-incrementing primary key (e.g., `` `id SERIAL PRIMARY KEY` ``) and a timestamp.

The json representation of survey responses should include the id and timestamp fields, along with all of the responses. You may use python or postgres to encode the json, it's up to you.

Challenge Task

Completing the previous steps correctly will earn you an A- (90%). If you wish to push yourself to earn the final 10%, complete this section.

Add one more page to complete the challenge task:

- /admin/summary -- this web page shows an aggregated summary of the survey results:
    - text answers are shown individually
    - for each choice/checkbox question: one summary chart (e.g., bar chart)
    - one additional time series chart that shows the number of survey responses per day
    - the charts must be rendered with http://www.chartjs.org

Technical Requirements and Restrictions

- The app must be hosted on heroku.
- The server must be written in flask, using jinja for templating.
- The server must use heroku's postgres database for data storage.
- The client must use Pure CSS for styling.
- The client may use vanilla javascript or jquery.