1 Homework 4

Due: May 5, by 11:59p.

To be turned in as a PDF using the upload link for Homework 4 on mycourses.

Important Reminder: As per the course *Academic Honesty Statement*, cheating of any kind will minimally result in your letter grade for the entire course being reduced by one level.

1. Bill is a non-technical proprietor for a small business. He hires a consultant to develop a web site where customers can place orders. The consultant successfully delivers a slick looking web site based on express.js and sets up the express.js server using a web hosting company. The web site works well and orders are pouring in.

   (a) Unfortunately, after a few days of successful operation John gets a call from the web hosting company informing him that the memory footprint of the express.js server has increased tremendously.

   (b) John has no idea what that means and places a call to the consultant requesting assistance.

   (c) Before the consultant can get back to him, John receives another call from the web hosting company informing him that his server crashed and they have restarted it.

   (d) The web site is working well with the server having a small memory footprint. However, John starts receiving complaints from customers saying that they have lost the contents of their shopping carts.

   Give a possible reason for this sequence of events. *7-points*

2. Assume that you have the following data structure for representing course grades:

```
  { //course-info
   rubricCategories: {
    hw: { title: ’Homework’, },
prj: { title: ’Project’, },
quiz: { title: ’Quiz’, },
exam: { title: ’Exam’, }
   },
   rubrics: {
    ...
    hw4: {
 title: ’Homework 4’,
 category: ’hw’,
 dateAssigned: ’2020-04-28’,
  dateDue: ’2020-05-05’,
  ...
        },
...
prj4: {
  title: ’Project 4’,
  category: ’prj’,
  dateAssigned: ’2020-04-21’,
  dateDue: ’2020-05-05’,
  ...
},
...
      }, //rubrics
      students: {
        ’b123456’: {
  name: ’John Doe’,
  class: ’senior’,
  ...
},
...
        ’b234567’: {
  name: ’Mary Doe’,
  class: ’graduate’,
  ...
},
...

      } //students,
      grades: {
        hw4: {
  b123456: { points: 92, ... },
  b234567: { points: 87, ... },
  ...
},
        prj4: {
  b123456: { points: 97, ... },
  b234567: { points: 90, ... },
  ...
},

      } //grades,
    } //course-info
```

Assume that you need to display a page which displays the grades for an individual student. The page should contain the following information:

- The student name and id (B-number).

- The student's grades, organized by rubric category and by rubric id within each rubric. There should also be an average displayed for each rubric category.
- An average of all the grades recorded for that student (the details of how the average is computed are unimportant).
- The rank of the student in the class.

How would you adapt the above data so that this page can be rendered using a mustache template. The details of the template and the page formatting are not required. What is required are the details of the data which will be mixed into a suitable mustache template. *20-points*

3. Given an HTML page displaying the grades for a student, how would you organize the display such that it is easy to write javascript / jquery expressions for:

   (a) Adding a class `hightlight` to the element on the page which contains the student name.

   (b) Returning a jquery list containing all the elements on the page which contain homework grades.

   (c) Calculating the average of all the homework grades contained on the page.

   (d) Adding a `hightlight` class to the element on the page which contains the grade for the first homework.

   (e) Identifying the element on the page which contains the grade for `prj4`.

   What is required for this question is showing the HTML structure, `class` and `id` attributes such that it is possible to write the above expressions. You **must** specify the expressions which will work with your structure. *20-points*

4. Instead of rendering an HTML page on the server using mustache for question 2, you decide to:

   - Build a web service which returns **all** the data specified for question 1.
   - Renders the grades for the specified student using reactjs.

   Discuss the details of this design. Specifically:

   (a) The advantages and disadvantages of this approach. Hint: there is a major security problem with this approach, but it is easily remedied.

   (b) Give the details of the design for the react component which you will use to render the grades for the specified student. The details should include:

- The properties for the component.
- How will the component access the web service data.
- A description of the massaging which should be done on the data in order to have the `render()` function contain simple jsx.

The details of how the student grades are formatted are not required. *20-points*

5. Discuss which HTML control you would use for each of the following form controls:

   (a) A control which allows the user to provide the percentage of components which are defective.

   (b) A control which allows the user to select one-or-more of the courses which are currently being offered by the CS department.

   (c) A control which allows the user to enter a customer-service complaint.

   (d) A control which allows the user to enter in a BU B-Number.

   (e) A control which allows the user to select their favorite dessert from a predefined list of desserts.

   (f) A control which allows a US user to enter in a telephone number (which may be an international phone number).

   Your answers should be set up to maximally constrain the user's input to legal values and minimize the opportunity for error. You should use at most a single HTML control for each of the above and should provide the actual HTML code for each control. *18-points*

6. Discuss the validity of the following statements. What is more important than whether you ultimately classify the statement as **true** or **false** is your justification for arriving at your conclusion. *15-points*

   (a) The `props` of a Reactjs component can be mutated to send information to its containing component.

   (b) Syntactically correct HTML must be well-formed; i.e. all tags must be properly nested.

   (c) Syntactically correct XML must be well-formed; i.e. all tags must be properly nested.

   (d) Syntactically correct JSX must be well-formed; i.e. all tags must be properly nested.

   (e) If a button is clicked, it is possible to set things up so that the button never sees the click event.