

COMP 10222 - Emerging Web Technologies

Lab #6 - Grade App with Vue.js

Due date: Friday December 4th at 11:59pm

Worth: 14% of total grade

Marks: 100 marks total

Learning objectives

- Use a component-based front-end framework to create a web application

Requirements

Create an application using Vue.js which allows an instructor to manage and analyze a list of grades associated with courses. The functionality should be implemented using Vue.js in the expected ways (e.g. using templates, directives, Vue instance data, computed properties, etc.).

The focus of the application should be a list of grades, with each grade having an associated course name.

The instructor should be able to do the following:

- **Create a grade.** After clicking on a create button, a modal dialog box should appear. The instructor should be able to enter a new grade between 0-100 and a course name (e.g. math/95, english/67). If the course name already exists in the list of grades or the grade is not between 0-100, the instructor should be presented with an instructive error message and asked to fix their

grade entry. If the data entered is OK, the modal dialog box should disappear and the grade and course name should appear in a list.

- **Delete a grade** The instructor should be able to click a delete icon next to each grade in the list to delete the grade from the list.
- **Edit a grade.** The instructor should be able to click an edit icon next to each grade in the list, and the same modal dialog used to create the grade item should appear. The model dialog should allow the instructor to edit the grade and course name the same as if they were creating it for the first time.
- **Sort grades.** By clicking arrow icons at the top of the list, the instructor should be able to sort the grades entered by grade (ascending/descending) or course name (ascending/descending).
- **Filter courses.** An input text box at the top of the page should allow the instructor to type in a course filter. As the instructor types in each individual letter, the list of grades should be filtered to contain only those grades with course names containing the filter text. So for example if the instructor has typed in "History" than "American History" and "Canadian History" would appear in the grade list, but "Math" and "Science" would not. If the instructor returns the input text box to a blank state, the entire unfiltered list should appear again.
- **Filter honor grades.** An "honor grades" button at the top of the page should have an on/off state, and it should initially be off. When the instructor clicks the button, the button itself should switch to a noticeably "on" state (e.g. make it green), and the grade list should only contain grades in which the grade is ≥ 80 . When the instructor clicks the button when it is in an "on" state, it should revert back to the "off" state, and the list of grades should no longer be filtered for grades ≥ 80 .
- **Filter fail grades.** A "fail grades" button at the top of the page should have an on/off state, and it should initially be off. When the instructor clicks the button, the button itself should switch to a noticeably "on" state (e.g. make it green), and the grade list should only contain grades in which the grade is < 50 . When the instructor clicks the button when it is in an "on" state, it should revert back to the "off" state, and the list of grades should no longer be filtered for grades < 50 .

Note that the effects of sorting and/or filtering grades should be cumulative. For example, if a user sorts grades by course name ascending, they could also filter for fail grades, and filter by entering a course filter, and all of these should together impact how the list is displayed (and how much of the list is displayed). This goes for any other combination of filters and any sort option.

At the bottom of the page there should be three values: min, max and average. The values should each have a label so that the instructor knows what is what. The values should be the min, max and average of whatever grades currently appear in the list (i.e. the values should take into account any filtering that is currently occurring).

Style the application to a professional quality. Just changing the background and font color isn't enough... grades in the list, icons and data should all be done to your best ability. If you want to use styles you find online to help you, that is OK so long as you cite them. Use some kind of nice looking transition or animation for deleting grades from the list (see: <https://vuejs.org/v2/guide/transitions.html#CSS-Animations>).

For this lab, just include Vue.js and any other libraries you use with script tags (i.e. no build tools or pre-compilation of templates). We will go over build tools before we're done with Vue.js, but it doesn't matter for our purposes with this lab and I can see your source this way.

Save your main file as index.html. This application does not require a back-end.

Submission instructions

When you have completed the lab, put your source files in a folder called **lab6.zip** and submit it in the Dropbox.

Statement of Authorship

Copying others people work and claiming it as your own is a serious breach of ethics. If copied work is found, all parties involved will receive a failing grade as per departmental policy. Group work is encouraged but it is expected that you

do your own work. If you do, you'll learn the material and feel better for it. Since all work submitted is assumed to be your own original work, you must include the following "Statement of Authorship" in every program file you submit for grading:

"StAuth10065: I John Doe, 123456 certify that this material is my original work. No other person's work has been used without due acknowledgement. I have not made my work available to anyone else."

- Use the exact text above. No substitutions.
- Replace John Doe with your name and the number 123456 with your student ID.
- Place this text as one line as close to the top of each document as possible.
- Include StAuth10065: at the beginning of each statement.
- Failure to include this statement will cause your work to be ineligible for grading.

Marking scheme

| | |
|--------------|-------------|
| List | 10 |
| Create | 10 |
| Edit | 10 |
| Delete | 10 |
| Sort | 10 |
| Filter | 20 |
| Max/Min/Avg | 10 |
| Style | 20 |
| Total | /100 |

Labs submitted late will receive a 15% penalty per day. Labs more than 2 days late will receive a grade of zero. For example, a lab submitted at 11:01pm on the due date that was marked 80/100 would receive a mark of $80 - 15 = 65/100$. A lab submitted at 11:01pm the day after the due date that was marked 65/100 would receive a mark of $65 - 30 = 35/100$. A lab submitted at 11:01pm two days after the due date would receive a mark of zero.