Luca de Alfaro @ UCSC

Search this site

**Welcome!**

**Archived Projects**

**Classes**

- CMPS 121 - Mobile Applications
- CMPS 183: Web Applications
  - CMPS 183 environment installation instructions
  - CMPS 183 Fall 2012
  - CMPS 183 Fall 2013
  - CMPS 183 Fall 2015
  - CMPS 183 Fall 2016
    - Homework 1
    - Homework 2
    - **Homework 3**
    - Homework 4
    - Lecture 1
    - Lecture 10
    - Lecture 11
    - Lecture 12
    - Lecture 13
    - Lecture 14
    - Lecture 15
    - Lecture 16
    - Lecture 17
    - Lecture 18
    - Lecture 2
    - Lecture 3
    - Lecture 4
    - Lecture 5
    - Lecture 6
    - Lecture 7
    - Lecture 8
    - Lecture 9
  - CMPS 183 Winter 2015

Classes > CMPS 183: Web Applications > CMPS 183 Fall 2016 >

# Homework 3

**Due: Friday November 4, 11pm**

**Assignment site:** https://www.crowdgrader.org/crowdgrader/venues/view_venue/

To submit, proceed as follows.

- First *pack* the web site. To this end, navigate to http://127.0.0.1:8000/admin/site (assuming you are serving the page from port 8000, the default), and for the Start application, click on Manage > Pack All. This will let you download a file called web2py.app.start.w2p. **Make sure your file name does not contain dashes '-'; underscores '_' are ok.**
- Then, go to this CrowdGrader assignment: https://www.crowdgrader.org/crowdgrader/venues/ and choose the link to submit. Upload the file web2py.app.start.w2p as an attachment.

Sample solution by Ethan Vadai

## The Assignment

For this homework assignment, you have to produce an app that is very similar to what you produced for Homework 2, except that:

- **Single page:** The app must be implemented as a single page app: the user should be able to do everything (except login) from the default/index page, without ever having to load a new page.
- **vue.js:** The app should use vue.js as its javascript framework.
- **Adding posts:** If the user is logged in, at the top of the page, there should be a button to add a new post. The text of the button should be exactly "Add new post". When pressed, you should display a textarea (a kind of multi-line input area) where the user can write a new post. When the user clicks on a "Post" button below this area, the post has to be:
  - added to the top of the feed
  - stored in the database (so that if the user reloads the page, the post is still there)
  - the text area on top must disappear, and the button to add a new post should be displayed again.
  - Please ensure that the textarea for entering a new post is initially empty.
  - The textarea to add a new post, and its 'Post' button, should be in a div with id add-new-post. The post button should say "Post" and have class post-button.

Web2py logging

Why web2py?

CMPS 276 - Software Engineering

CMPS 290G

How to host a simple app on Google Appengine

Raspberry PI Boot Camp

**Collaborators**

**Contact Information**

**Creating and using a git repository**

**Interested in UCSC?**

**Is de Alfaro under A or D?**

**Office Hours Appointments**

**Past Projects**

**Publications**

**Publications (by topic)**

**Resources**

How to create a simple app on appengine

**SlugIOT**

**Talks**

**TestDocs**

**The Wikipedia Authorship Project**

**External Links**

Resume

Luca's personal home page

CrowdGrader

UCSC School of Engineering

UC Santa Cruz

G+1 0

- There should be also a Cancel button that enables the user to choose not to add a message after clicking on "Add new post".
- **Displaying posts:** Posts should be displayed as in homework 2, except that you should display initially only the four most recently created posts. If there are more than four posts in the database, you should show a "Load more" button below the four posts. The load-more button should have id load-more. Whenever the button is pressed, you load four more posts (or as many as there are left in the database). The button to "load more" should be present whenever there are more posts than are currently displayed. Each post should be displayed in a div of class post-content; this div should enclose the post text, AND all the buttons that enable to edit and delete that post.
- If the user is logged in, each post by the user should have a button to delete it (use `<i class="fa fa-trash-o"></i>`) and a button to edit it (use icon `<i class="fa fa-pencil-o"></i>`. Use exactly these icons and not other ones.
- **Deleting posts:** If the user presses the trash icon `<i class="fa fa-trash-o"></i>` to delete a post, the post should be deleted both from the feed, and from the database.
- **Editing posts:** If the user presses the edit icon `<i class="fa fa-pencil-o"></i>`, the text of the post should be replaced *in place* by a textarea that enables to edit it, with below, two buttons, "Post", "Cancel". The buttons should say exactly this, and they should be inside the div of class post-content mentioned above. The Post button should have a class post-edit-button, and the cancel button should have a class cancel-button.
    - If the user presses "Post", then the new text should replace the old one, both in the post, and in the database.
    - If the user presses "Cancel", the edit should be abandoned.
    - Please ensure that the editing area contains initially the old value of the post.

Notes:

- Do not tamper with web2py top bar and login / logout mechanism, as it will be used in automated grading.

## Hints

In this assignment, you will often need to make one thing appear in place for another -- for instance, a <p> with the text in it or a <textarea> with the same editable text in it. To do this, define a javascript variable for vue:

```
self.vue = new Vue({
    el: "#vuediv",
    delimiters: ['${', '}'],
    unsafeDelimiters: ['!{', '}'],

    data: {
          b: false
     }
});
```

and then, use Vue.js ability to display certain elements depending on the boolean status of b:

```
<p v-if="b"> ${my_text_variable}</p>
<textarea v-if="!b" v-model="my_text_variable"></textarea>
```

## Starter Code

The starter code is in the hw3 branch of our starter app.

web2py.app.lectures-3... Luca de Alfaro, Nov 14, v.1

### Comments

You do not have permission to add comments.

Sign in | Report Abuse | Print Page | Powered By Google Sites