Notes on Private Wall Assignment 4/10/19

Note: removed this from messages.html deicding that we always have the logged id in a cookie,so don’t need to pass it in:

<input type="hidden" id="" name="sendersID" value="{{the\_users['sender\_id']}}"> <!--Need sender's id here so /messageSend route knows how to record sender in database. /wall route populates it, and if this form's submit is clicked, it goes back-->

-----

Checklist:

* Routes outlined
* Database created
  + Table and field names?
* From yellow sticky
  + Include all validations for registration & login
    - First name and last name should be at least 2 characters
    - Email address should be valid
    - Passwords should match
    - Password should be at least 7 char
* I don’t see it stated, but:
  + Check db for existing email address before writing/registering
* The root route should display a template with the login and registration forms
* Validate the registration input
* If registration is invalid, error messages should be displayed on the reglogin page
* Validate the login input
* If login is invalid, display an error message on the index page
  + Same reglog.html page as used for collecting information
* If login is valid, store the user in session and then redirect to the success page
  + In Amy’s video, same success.html page with exact same display as registration success
* Add a functioning logout button to the success page that clears session
  + See snippets page for how to clear the cookie
* After logging out, ensure you cannot reach the success page
* From yellow sticky
  + Logout: After logging out, the user shouldn’t be able to reach the main page…or only logged in users can reach Wall One page
* From yellow sticky
  + Viewing Messages
    - The logged in user should be able to see messages sent to them on the left
    - Messages should display the name of the user who wrote the message and the contents of the message
  + Sending Messages
    - The logged in user should be able to send messages to all other registered users on the right
    - Apply validation to messages: content should be at least 5 char long
  + Deleting messages
    - The delete button should delete the associated message

Populating the message page:

* Need the first name for the top

Need Send side forms to have a dictionary of all users except the logged in user and then:

>>>>Action: learn how to get a select result that has the needed below but also does not have the logged in ID record included

-----need first name of each and logged in id

>>>On messages received, need to collect:

select all messages order by sender's first name, desc where sender ID not equal to logged in id

collect the sender's first name of each and

the message ID so it can be deleted

the content of each message

also grab the time message created...in case of sensei bonus

Ninja Bonus: Display the number of messages for the logged in user

try to get a SQL satement that counts # in message table where received id =logged in

----or maybe have Python count the items in the list which should be count of records or dictionaries

>>>On message to send

select all users including fname and id.....do order by fname desc do where not equal to logged in so it won't have one for sending to oneself

Ninja Bonus

Display the number of messages the logged in user has sent

try to get a SQL satement that counts # in message table where sender id = logged in,

----or maybe have Python count the items in the list which should be count of records or dictionaries

Sensei Bonus

* For each message, display the time since it was posted
  + Did display that time at which it was posted
* List the users in alpha order
* Do not allow someone to remove a message that does not belong to them. If they try, have your app display; Message ## does not belong to you..your are bad
  + Your ip address xxxx has been recorded
  + Next time you attempt an activity like this we’ll log you off automatically and report your ip address to our site administrator

More checklist:

* use anchor tag for deletion with message id in url
  + confirmed working
* Michael suggests using an image in the anchor tag
* From M video:
  + Show all users (except the logged in user) and allow the logged in user to send a message to any other user in the system

**nina\_t**

[4:21 PM](https://dojo.news/coding-dojo/pl/a1bj4ymmufnzimohg5jiqzty1h)

is anyone working on private wall yet? I wanted to see how your database was designed

[4:21 PM](https://dojo.news/coding-dojo/pl/d9j3uc1y3tr3mqwns54bewmi6c)

I have two tables but before I forward engineer this baby I want to do a sanity check

**mike\_murray**

[4:21 PM](https://dojo.news/coding-dojo/pl/mbsn4j5ra7fit8h5eamm55potr)

i'l literally looking at that right now..

**kent\_h**

[4:25 PM](https://dojo.news/coding-dojo/pl/9cs9ekdy87djtrkjmfmqkwbdwr)

I will look at its db design, now

I would guess at least a users table and a messages table

The below might indicate using more than two tables...because having more than two relationships that are not self joings would mean more than 2 tables:

Include multiple one-to-many relationships

**kent\_h**

[4:38 PM](https://dojo.news/coding-dojo/pl/b3c5pxdjdjgh8bcphhuho7iwae)

In the video Michael talks about how many forms we have, but of course there is only one action per html page, so any of those send buttons really do the same thing....Amazon uses that by scattering "buy now" around their pages

**kent\_h**

[4:46 PM](https://dojo.news/coding-dojo/pl/pqkuduwb37gu3gumssw7r895ph)

I think in the real world it would be better to have one text area and dropdown to pick the recipient. THere are only four send buttons on the wireframe....what if there were 20 or more users...would the page just grow to have that many text areas?

I think the below is related to Michael's idea of having the message ID pass in the URL and thus when bad user figures that out he can try to delete any message:

do not allow someone to remove a message that doesn't belong to them. If someone tries to remove a message that doesn't belong to them, have your app display the following:

because the entire message displays on the "Wall One" page, it would seem it also could cause the page to grow excessively long...unliss they are all inside of a div that has overflow with scroll bars...if that works

Still thinking about table count...

I do think Michael might want more than two tables, but I can't imagine what they would be....users...the regular stuff specific to a user...if we are tracking their postal address, then that would be another table, but I don't think so. The message table has a list of messages. Each has a sender and a receiver and content and date created...I don't know what else. If deleting a message just hides it forever, then another field to show messages are archived.

**kent\_h**

[5:11 PM](https://dojo.news/coding-dojo/pl/ms1yrtzanjd5djyb4qo798dzio)

Multiple forms with only one page send...how to know who is the recipient...

this page discusses: <https://stackoverflow.com/questions/547821/two-submit-buttons-in-one-form>

Stack Overflow

[**Two submit buttons in one form**](https://stackoverflow.com/questions/547821/two-submit-buttons-in-one-form)

I have two submit buttons in a form. How do I determine which one was hit serverside?



I hope users are smart enough to not populate two text boxes then click a third submit

New Messages

**kent\_h**

[5:27 PM](https://dojo.news/coding-dojo/pl/wuxe5xqiijya5f778dftrekmur)

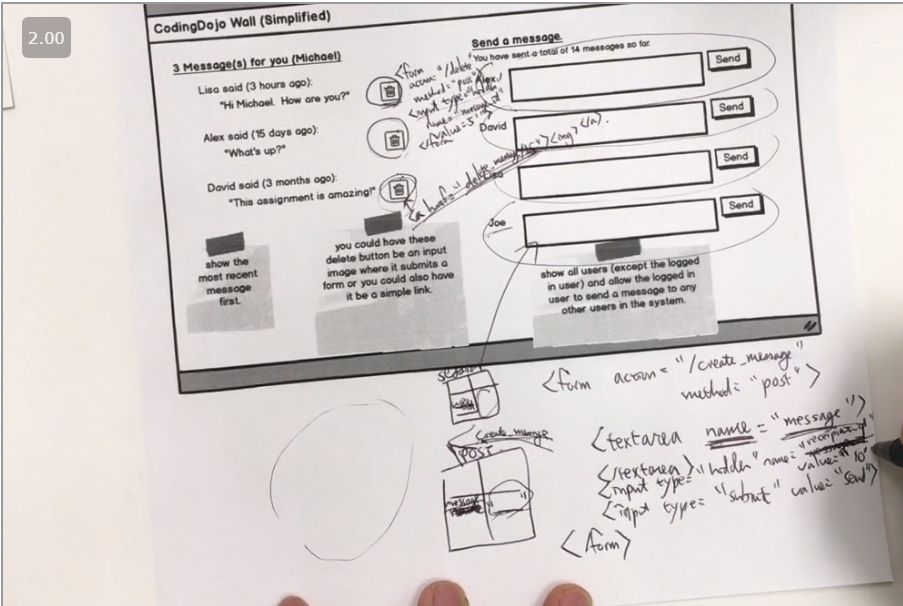
On the submit...Jan's answer on this page seems to match Michael's. Of course we get to iterate each form dynamically

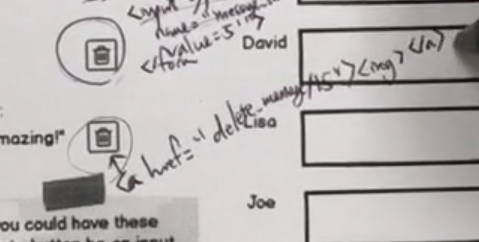
<form method="post" action=""> <input type="hidden" name="product\_id" value="123"> <button type="submit" name="action" value="add\_to\_cart">Add to Cart</button> </form>

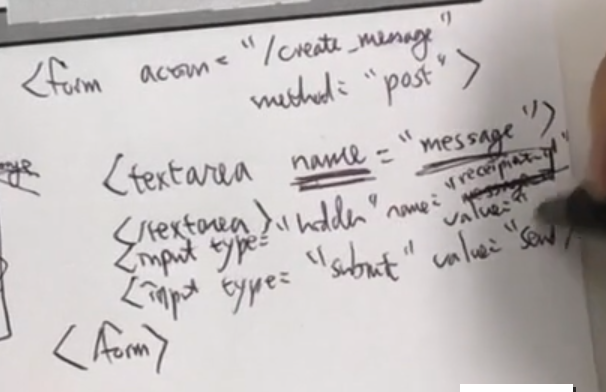
<https://stackoverflow.com/questions/8712398/multiple-forms-or-multiple-submits-in-a-page>

Stack Overflow

Michael’s video

* Use anchor tags for the deletes
* Use below for message submission:
* 





-=======

