

Minh Nguyen

Assignment 2

<https://trying-992.appspot.com/>

This site is an example of an accounts database for gamers. Upon loading up the page, you are asked to enter in 5 different data types for your account. A username (text input), sex (radio button), Operating System used (select dropdown), Types of game preferred (check boxes), and a description of yourself (text area input). These 5 different form field types are then saved into a view page where you can see what you have entered. You are then able to edit this information you entered via an edit button that is supplied on the views page.

Test plan:

- Allow users to enter in blank data. Any data entered will be viewable in view page after submitting form.
- User chooses no genre of games preferred : view page should not display any games preferred
- User chooses all game genre : view page should fill up all 4 game genre
- description of user can be empty: view page should display nothing
- description of user should have one string: view page should display a string
- description of user should display multiple strings with spaces: view page should display the same content
- clicking edit page should fill up with same information as view page
- clicking update on edit page should go to view page with the information that was on edit page
- checkboxes should be checked if input data page checks in (genre) for edit page
- Radio boxes should be filled in on edit page if input page choose said radio box (sex)
- operating system drop down menu should be selected in edit page if user chose it in main menu
- updating user name in edit should change username accordingly to user input

- updating sex in edit menu should change upon clicking update
- updating OS should change depending on choice of OS in update options
- If user unchecks all of genre type in edit, then update should remove all of genre games in view mode
- text area input for edit page should display the same content that was in view page

Result of test plan:

All of the test passed except for the last case with the text area input. If a user enters in a string without white spaces in the text area then in edit page it will display that string. But when user enters in strings with spaces, then only the first string is saved and populated in the edit page. The reason for this test failing is due to a design flaw in the view.html page. I did not iterate through the string correctly when clicking the "edit" button on the view page to go to the edit page. When the edit page's post request is active, it does not read the "text" correctly because when submitting from the view page to the edit page, information was lost.

Templating:

I used jinja2 since I did my web app in Python on GAE. Templating was used here to render the html for all 3 of my pages, input, view, and edit pages. With the use of templating, I was able to cut down on writing the actual HTML code in python and makes for a much more organized program. Passing variables via jinja2 through post requests is simple and straightforward. I was able to display any variable I want by giving the variable a name and a value, then passing to the html page where the html has jinja2 code in it to display the variables.

This extra level of abstraction makes code look clean as well as efficient when needing to change code depending if it is in the html for the front end, or the python for the back end.

Changes:

If I had to do this program over again, I would make use of the NDB datastores that is in GAE. For now, I'm simply passing local variables around and saving it only in that instance of the app. I would like to implement a more permanent accounts page and that requires using datastores to save information. I would also imagine it would give me a head start on the next assignment as well since I

would then know how to set up the NDB to my program already. As of now, I would still have to learn it for the next assignment.

Also, I would give myself more time to work on this web app next time around as I would like to add some CSS to it. That is not important but it would be something I would consider doing next time around. If I had given myself more time to work on it, I would have been able to fix my one bug I had in the test plan