So I’m not really sure how to document this formally, so instead I’ll just present it as a stream of consciousness, like the same things I would say if I was describing it out loud.

My idea for the external game validation site is to build a simple (i.e. no fancy theming, think 90’s internet or Craigslist) MVC app (that’s model-view-controller, I can go into more detail on what exactly that means if you guys would like) tied to a database with some automatically generated tables.

The exact contents of the tables are a mystery to me because its auto-generated fluff but I know that they store the email address and the hash of the corresponding password.

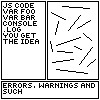
We can then set up a controller (again, lmk if you need elaboration) in which each action is locked behind authentication, i.e. you can’t access any of it unless you are logged in.

We won’t have to worry about which admins have access to what; if they are logged in, they are admin. Admin is admin.

We can use the actions on this single controller to implement the functionality of the game testing sandbox.

My idea for the layout of the view that will be controlled by this controller is something like the “try it now” editors that many programming how-to websites have. Think W3Schools.

That’s a textfield box on the left for entering user-submitted javascript and an HTML5 canvas on the right for rendering that javascript output to.



Here is a simple mockup

I am not an artist

We could populate the lower panel with information fetch from a javascript rating bot/service like Tyler was talking about. We probably don’t need to get a full javascript linter running on the client or server side, because if a user submits broken code, it’s not an admins job to fix it, just throw it out and notify the submitter that their code was bad.

Alternatively, instead of a text box to paste user-submitted js into, we could populate the upper-leftmost panel with a list of submissions. Clicking on the submission could load the script and render it to the upper-rightmost panel. This would be a bit more complex and would likely require us to make a new table in the database and a corresponding model for submissions.

However, I think we should keep it as simple as possible and not get too ambitious with it, at least not yet.

As for the question of securely linking to the Game Validation site from Bucstop, a simple hyperlink embedded somewhere in the header or footer of BucStop should suffice. If we want user-submitted code to be sent directly to an inbox of sorts on the external site, however, that would be a bit trickier.