Iteration: 4 (16/03 - 22/03)

Alex Taylor - amt22@aber.ac.uk

March 24, 2017

Version 0.4 (Draft)

Contents

1 Story: Admins can log into the website and run a session with a quiz **2**
1.1 Analysis - Breakdown of Tasks 2
1.2 Design 2
1.3 Implementation 4
1.4 Testing 4

1 Story: Admins can log into the website and run a session with a quiz

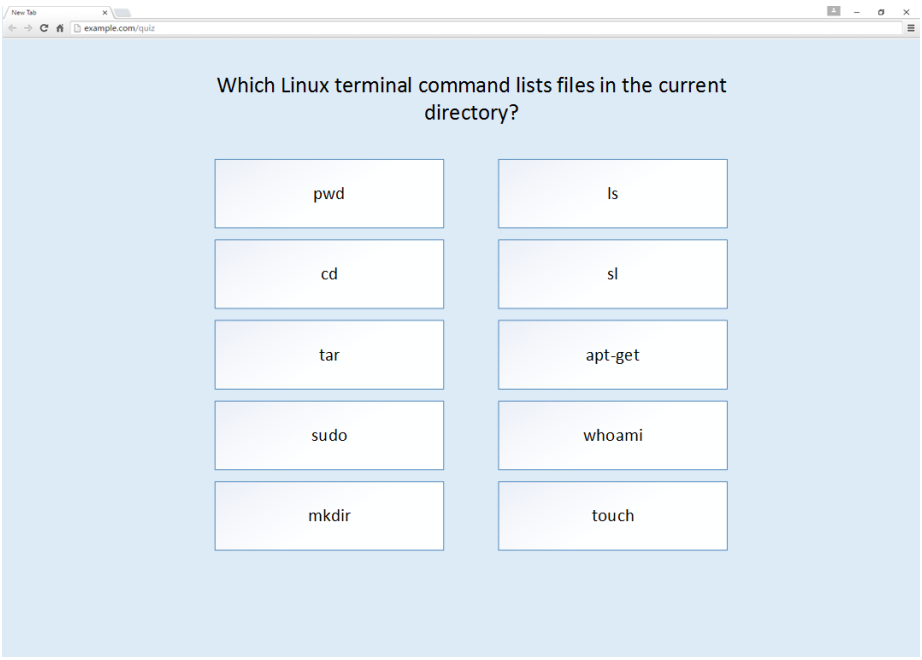
1.1 Analysis - Breakdown of Tasks

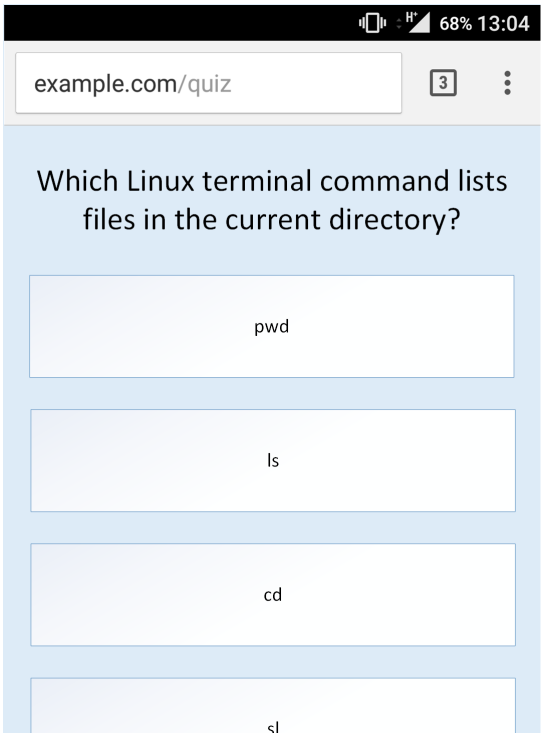
After doing some spike work last week, this task can now be approached and broken into several sub tasks:

- Set up config for Pusher
- Add event for broadcasting
- Write a command to trigger this event
- Add a button to quizzes to trigger this event
- Display this on the front end using the javascript which listens for Pusher events
- Add a session key box to front page
- Add session id to users
- Allows admins to change their id
- When admin clicks run, this id can be enetered into the key box to join a channel with the name of the id
- The user will be presented with the initial quiz page which will be default filled with the name and description of the quiz
- The admin will see the same page but with an "admin panel"
- This admin panel has a next and previous button for questions
- When these buttons are pressed, the question is sent to pusher
- These question are rendered as a form on the user end and admin end
- The user can submit the form

1.2 Design

Here are some mockups of the user end of the system (TODO: label and captio these images):


example.com/quiz
Which Linux terminal command lists files in the current directory?
pwd
ls
cd
sl
tar
apt-get
sudo
whoami
mkdir
touch


68% 13:04
example.com/quiz
3
Which Linux terminal command lists files in the current directory?
pwd
ls
cd
sl

1.3 Implementation

It began by configuring pusher, creating an event and writing some basic javscript to append to the front page. Some work was done on the Admin panel as well, making it only visible to logged in users, and adding the functionality fro previous and next questions. These are just AJAX requests to quiz controller actions which then trigger the event for WebSockets.

Early into the iteration a major flaw with using the WebSockets came up, whilst new content was easy to add to the page, if a new user joined the session late, they would see a blank page/ or the original content of the page that has not yet been removed with javascript. To remedy this, the current position in the quiz should be kept track of and the php on the page should load the question in that position. At the same time, the WebSockets will be running and updating the page from that point onwards in the session. I decided the best place for keeping track was within the session table, which now has appropriate columns for the position, quiz and if it is running.

1.4 Testing

No testing as the story is not yet complete.