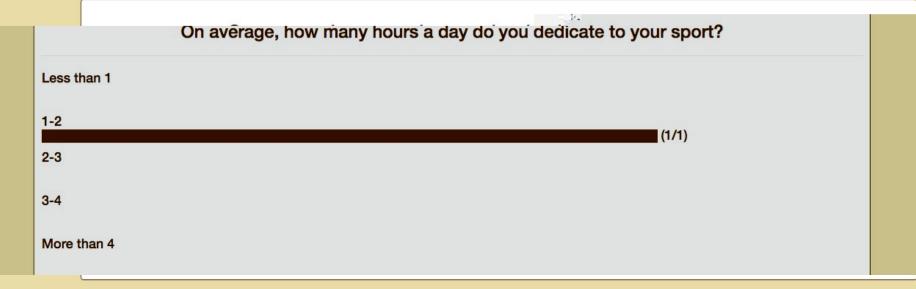
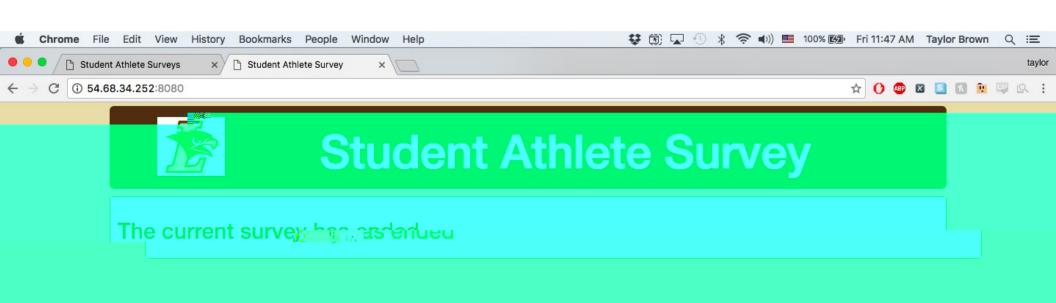


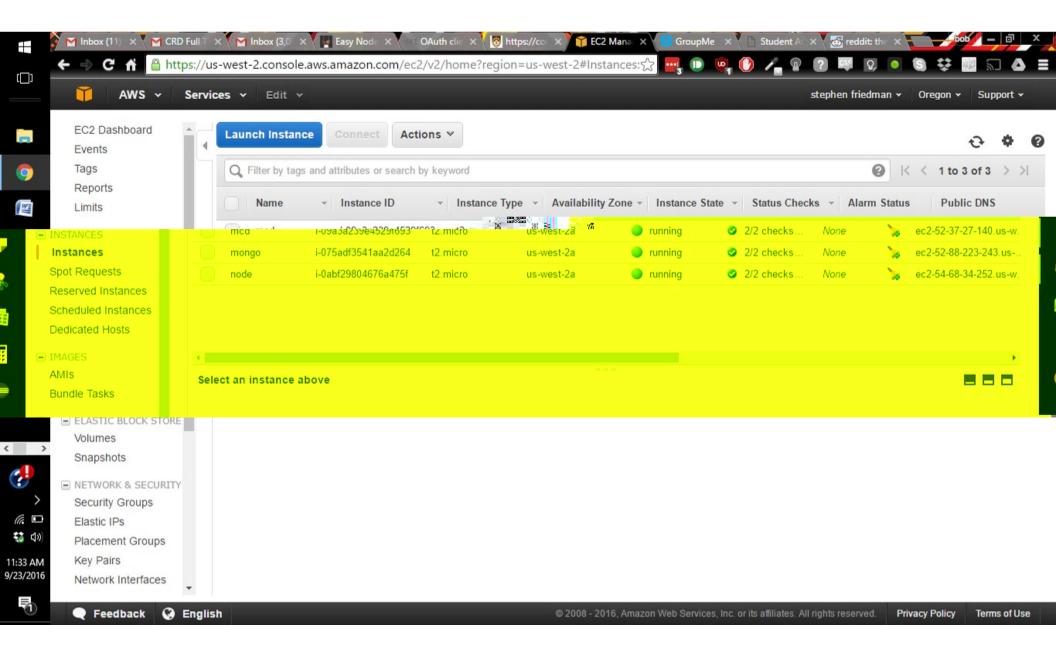


test4









Group: Taylor Brown; Stephen Friedman

- 1) One major benefit of using mlab since they are a SaaS (software as a Service) the user does not need to worry about updating since the provider takes care of all the backend issues. Also, this is usually far easier to setup than a VM with mongodb on it allowing for rapid development with little planning. The user, however, does not know who else they are sharing resource with on the mongodb SaaS leading to performance possible being less than expected. Also, it is understood that since the user does not know the security around their information on the SaaS there may be holes that the provider does not know about. If the user is using their own VM with mongodb they will have a full view of the possible security holes.
- 2) Since we are using mongodb it is possible that Eventual Consistency will become an issue. Mongodb is a nosql database and is document based, this means the data is replicated on multiple servers, the clients can access any of the servers to retrieve the data. This means that a user writes a piece of data to one of the servers, that has not been copied to the rest. Then another user reads the most up-to-date copy of this data from this server. However, this client also access another server and receives an old copy. This means that a bug could form in the app that allow a user to access data that is old when is means to access something that is up to date. Leading to the totals on the survey becoming messed up and incorrect.
- 3) The app does a good job of storing the config within the environment that it is configuring. Since the app has broken up the configuration of the different pieces into the respective spots in the structure of the app is follows the requirement to have a strict separation of config from the code. None of the configuration information is hardcoded into the app code, it is always calling on the separate configuration files that are needed

- to run it. It also does not have just one configuration file but many to break the whole configuration up into the specific pieces that are being configured.
- 4) The app does a very poor job in codebase revision control. An app should be using a tracking system that allows for separation between developing code and production code. All of these should be connect to a codebase that they then build off of. Since the app has all the code stored on the node VM with no connection to gitHub or another version control system it means any update to the code will have an immediate effect on the app. This is very poor design for an app as it can lead to it not working at all times and leads itself to possible errors.
- 5) The bug is in fact not solvable. In the event where a user has for example survey question #7 open, and the admin closes that question while the user still has the question open, the user will be able to fill out their selection and submit their answer. However, this is not a problem because of MongoDB utilizing eventual consistency. This means that even though the user was able to submit their answer after the admin had closed the question, the database will not will not store their answer because it is able to recognize that the survey question was closed before the survey answer was submitted.
- 6) It is in fact possible to prevent a user from taking a survey twice while keeping their anonymity. This answer follows the assumption that the survey taker would need to receive a link to the survey in order to complete it. Upon the admin sending out the survey to the user, the admin should create a unique key that a user must enter upon the start of the survey. The admin shall store this key in the database upon sending out the key and the link to the survey taker. In the database, the key is not associated with any user information, just the key itself should be stored. When the user goes to take the survey and enters in the provided key to begin the guestions, the database should check

that the key exists. If the key exists in the database the survey should begin and the key should be deleted from the database. If the key is invalid, the survey should not commence. So, after a user uses their key to take the survey, they will not be able to reuse the original key that was provided to them. Thus, a user will only to be able to take a survey once while their answers remain anonymous.

- 7) The way the application works now is that the questions are pulled from the database each time the user takes on the next survey question. Doing so requires pulling from the database an equal amount of times as there are survey questions. Rather than pulling each question each time the user loads a new survey question, all survey questions could be loaded into memcached upon the admin starting the survey. This would lessen the amount of times that data would have to be pulled from the database. Thus, when a user navigates to each question in the survey, they would be pulling the questions out of memcahced as opposed to the expensive computation of pulling from the database.
- 8) Heroku is inferior to the AWS VMs that we have created for a few reasons. Mainly, the developer can add whatever software system they are most comfortable in using to the system. They hard not beholden to the offerings of Heroku. This means that the user has greater control over making the system work to the specific needs of the app versis trying to force or hack what heroku gives them to work with. With AWS you can also break up the caching of the app, with Heroku this is not possible as you do not have this low level access. The AWS also allow for the developer to choose what hardware to use, enabling greater fine tuning to the needs of the app and the software that it uses.
- 9) One reason using Heroku would be better is if there was a plan for the app to be scalable. It is far easier to scale the app in heroku than it is with AWS. Also, for the app heroku would force the use of git, or another similar system, allowing for much better use

of codebase revision control which this app is lacking as of now. This is due to how heroku deploys, the developer has to push into the git repo that Heroku provides.