**Installation Instructions**

*Homebrew*

1. Open Terminal and type: ruby -e "$(curl -fsSL [https://raw.githubusercontent.com/Homebrew/install/master/install)"install/master/install](https://raw.githubusercontent.com/Homebrew/install/master/install)%22install/master/install))"
2. For more information about Homebrew, go to <http://brew.sh>

*MongoDB*

1. Open Terminal and type: brew install mongodb
2. For more information on MongoDB installation, go to <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-os-x/> and read the section “Install MongoDB with Homebrew”
3. To create the local data directory, open Terminal and type: mkdir -p /data/db
4. To check if your installation worked, open Terminal and type: mongodb

*Node.JS*

1. Go to <https://nodejs.org> and click the Install button to download the .dmg package
2. Open the .dmg file and finish the installation
3. To check if your installation worked, open Terminal and type: node
4. For more information on Node.js installation, go to <http://coolestguidesontheplanet.com/installing-node-js-on-osx-10-10-yosemite/>

**Accounts Setup**

*BitBucket*

1. Create a new BitBucket account at <https://bitbucket.org/account/signup/>
2. The repository can be found via browser at <https://bitbucket.org/kaitlynkwan/mps_proj> but you will be cloning it in the next step
3. To clone the repository, type: git clone <https://kaitlynkwan@bitbucket.org/kaitlynkwan/mps_proj.git>
4. Configure your git account by typing: git config --global user.name <name>
5. Then to add your email, type: git config --global user.email <email>
6. To read more about git repositories and how to add/edit files, read: <https://www.atlassian.com/git/tutorials/setting-up-a-repository> or <https://confluence.atlassian.com/display/BITBUCKET/Clone+Your+Git+Repo+and+Add+Source+Files>

*Heroku*

1. Create a new Heroku account at <https://signup.heroku.com/login> (If asked, development language should be Node.js)
2. To understand the backend (Node.js – JavaScript server), read <https://devcenter.heroku.com/categories/nodejs>
   1. Create a test application using this: <https://devcenter.heroku.com/articles/getting-started-with-nodejs#introduction>
   2. This will also help you install necessary dependencies such as the Heroku Toolbelt
3. Send your Heroku account email to me so I can add you as a collaborator on the project
4. To add the current heroku remote, open Terminal, change into the directory containing the project, and type heroku git:remote -a stark-crag-5229
   1. For more information on Heroku remotes, read <https://devcenter.heroku.com/articles/git#creating-a-heroku-remote>

**Running the Application Locally**

Start Application Locally

1. Open a Terminal window and change into the application directory
2. If you are starting for the first time, make sure all the Node.JS dependencies have been installed by typing: npm install
3. To start the mongo daemon, type mongod
4. Open a second Terminal window and again change into the application directory
5. To start the Node.JS server, type foreman start web
6. Open a browser window and navigate to: <http://localhost:5000/login/music> (or some other URL that begins with localhost:5000)

Deploy Application to Heroku

1. Open a Terminal window and change into the application directory
2. To add all files from the local directory, git add -A
3. To commit the files and get ready to push to the remote repository, git commit –m “Type a message here”
4. To push the committed files to BitBucket, git push origin master
5. To push the committed files to Heroku, git push heroku master

**Updating Database**

Uploading/Editing Questions

1. Open Terminal and connect to the database by typing: mongo ds061731.mongolab.com:61731/heroku\_app35837312 -u admin -p admin
2. In order to upload new versions of the questions, you first need to remove them from the database. To do this, you go into the “questions” collection, and use an identifier to select the questions you want to remove (the key-value information within the brackets). Some examples are:
   1. db.questions.remove({questionnaire: ‘policy’}) – removes all policy questions
   2. db.questions.remove({questionnaire: ‘music’}) – removes all music questions
   3. db.questions.remove({questionnaire: ‘music’, main: 0}) – removes all music questions that are not the main ones
   4. db.questions.remove({}) – removes all questions
   5. When you’re connected to the database, you can also perform a lot more commands to see the data in different ways. For example, you can find questions with certain specifications or conduct aggregations. To read up more on MongoDB commands, look at: <http://docs.mongodb.org/manual/reference/command/>
3. Then to upload the questionnaire with the updated questions, you import the excel CSV file into the database. Make sure the CSV was saved as a “Windows CSV” file and not just a regular or MS-DOS one, or it will not upload! Type: mongoimport -h ds061731.mongolab.com:61731 -d heroku\_app35837312 -c <collection> -u admin -p admin --file <file\_name> --type csv –headerline  
   Some examples are:
   1. mongoimport -h ds061731.mongolab.com:61731 -d heroku\_app35837312 -c questions -u admin -p admin --file questionnaire\_policy --type csv –headerline
   2. mongoimport -h ds061731.mongolab.com:61731 -d heroku\_app35837312 -c questions -u admin -p admin --file questionnaire\_music --type csv –headerline