***Project Brief:***

A resource / website for students practicing acro-yoga.

Most students take acro yoga classes at a gym or they learn it from their friends, playing at parks, etc.

This website serves to provide logical guide for students to visualize via YouTube videos some of the poses that are needed for them to achieve mastery in a certain level. It is curated and separated out in different levels so that students are able to differentiate what is level 1, 2, 3.

The purpose of this is to help students not only gain mastery and confidence in their appropriate levels, but to develop strength in a format that is meant to level them up appropriately so that they can sustain their practice and not get hurt in the process.

(Some students jump into poses not understanding the prerequisite poses that were needed prior to executing more difficult washing machines)

***Aim of the website:***

Users:

1. Students who want to access the site to see the curated videos.

* Students would select from a list of levels of level. After that, a curated list of videos would be provided for them.

2.) Teachers or advanced practicioners who create an account, and uploads additional sites to the list of existing videos.

Question: Since I want to be the one to curate the site, wouldn’t I want to first see the videos uploaded before it gets placed into the DB on the site? For the sake of the project, should I allow users to just upload directly?

***Site would include:***

1. Home page: includes Login and New User buttons

* We want everyone to create an account to track our users for future analytics and also to send them newsletters or alerts

User table: (New User Sign up Page)

1. sequelize model:create --name User –-attributes first\_name:string,last\_name:string,email:string, user\_name:string,password:string
2. First\_name
3. Last\_name
4. email
5. user\_name
6. password
7. ~~address~~
8. ~~Yes / No to being on mailing list (called “list” in the table)~~
9. ~~Login Page:~~

~~UserName~~

~~Password~~

1. Already logged in or signed up-🡪 Selection page

* Level 1, Level 2, Level 3 🡪 opens on my site ( all of my pages will have the LogOut button)
* Resources to Acro Maps-🡪 opens in another tab (html attribute needs to be added on the a-tag) <a href …. target=”\_blank”)
* Resources to AcroPedia🡪 opens in another tab
* Resources to Acroyoga
* Fun Videos to watch 🡪 opens on my site
* Log Out button

1. Tables for Videos:

Level 1, Level 2, Level 3

* Level number
* Videos associated with that level
* List of pre-requesites or recommended positions

Sunday:

1. Take pictures 2-2:30
2. 3-5P
   1. Need to finish the modal for the User Login section
   2. Need to start and finish the 2nd page for users to select :
      1. Level 1, Level 2, Level 3, Fun Videos (these are in the Carousel example page.)
      2. Links to the resources should be one of those boxes as well.
      3. Total large pictures: 4
      4. Total smaller pictures 4-5
      5. Total small circle pictures – 6?
      6. When users click on the larger heading link (level 1, 2, 3), it will launch in the squre box next to link—or users can choose to expand the window

Tuesday:

1. download Heroku
2. Need to create a github respository
   1. Create a new repo in github.com
   2. Go into your folder in the terminal and type: git init
   3. Git status to check things out
   4. (be sure to create a file in your folder to gitignore the files that we don’t want to commit or push b/c these files might be too large—such as my images folder.)
   5. git add .
   6. git status
   7. git commit –m “Initial Commit”
   8. git status (we want to see a clean directory)
   9. git remote add origin https://…….(from github), this allows us to connect what we commited locally to the github
   10. git remote –v (allows us to see the remote and what is now available)
   11. git push –u origin master (this pushes our code to our github repo – from our local repo on our laptop.)
3. Create the DB, models, migrate the model
   1. DB for Users (see modal for list)
   2. DB for Login (username, password)
   3. DB for Level 1, videos, transition poses (may need to choose the levels from a drop down menu to guarantee db entry)
   4. DB for Level 2, videos, transition poses (may need to choose the levels from a drop down menu to guarantee db entry)
   5. DB for Level 3, videos, transition poses (may need to choose the levels from a drop down menu to guarantee db entry)
4. Work on the app.js, index.ejs, etc files
5. Work on linking up the resource tabs to the HTML page
6. Log out button needs to linked
7. Need to find all the videos to populate the DB
8. \*\*change th three columns…we do not want the link to show up on the table …we just want the user to click on the NAME of the pose.

Database modeling and migration—in the Terminal, these are the commands  
1.) npm init (sets up the json files)

1. sudo npm install –save express, ejs, pg, sequelize, body-parser, passport, bcrypt, connect-flash, ejs-locals, method-override, passport-local, cookie-session (12 modules installed)
2. sequelize init (this initializes the sequelize project and creates a config.json file)
   1. Then go to the config.json file and change it to the red items:

{

"development": {

"username": null

"password": null,

"database": "acroyoga\_project",

"host": "127.0.0.1",

"dialect": "postgres"

},

1. createdb acroyoga\_project
2. two models to create: Username and Videos
   1. Terminal: (User model)

sequelize model:create --name User –-attributes first\_name:string,last\_name:string,email:string, user\_name:string,password:string

(video model)

sequelize model:create –-name Video –-attributes level:string,url:string,words:string,pos\_name:string, userid:integer

1. (this is not the creation of a db tables yet)
2. Then – after the two models are created, go to the terminal and type:

a.) Sequelize db:migrate (this will create the tables in the DB!)

\*\*check to see if the DB is created with the correct tables.

(go to PSQL, \connect acroyoga\_project, then :

\dt

select \* from “Users”;

select \* from “Videos”;

1. SiteMap
2. Wireframes
3. Tables
4. Code/ HTML/CSS/ Bootstrap – front end
5. Node (NPM Init, NPM install all the modules (pg, sequelize, etc, etc), using Sequelize—create a DB, create models, migrate the DB, then put all these pieces into the front end.)
6. Create a power point presentation?

Poses for pictures:

1. Candlestick
2. Folded paschi/ thigh stand
3. Straddle throne, regular throne
4. Backbend
5. High flying whale
6. Threesome: downward dog, backbend x2
7. V-up/reverse plank
8. Straddle throne to thinker
9. Bird on hands (Arun/Melody)
10. Falling leaf, to tulip pose with hands on chin (for Myla)

Unfinished business:

1. Need to create error messages when users sign up with an incorrect name
2. Need to create error messages when the user is NOT in the system
3. Why does it not work on Heroku?

Mou (markdown editor for web developers)

Markdown

Macdown (is another one url: macdown.uranusjr.com)

Push the write up and save it as a readme.md

App.get (‘/videos/levels/:levels’)

Video.findAll(order\_by:level)

Buttons or as links

<a href : src =”<%lkjlkjlkjkjlkjk%>