Tassel Documentation

Table of Contents

Setup

If you already have Node installed, then you should uninstall it to avoid confusion in the shell. NVM can be used in place of your node.
1. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh|bash
2. nvm install 16 (with a v, not npm. May take a while.)

In case the above didn't work (if you saw GLIBC_2.28), try:
- nvm install 16

In the future, if you see GLIBC_2.28 error:
- nvm use 16

If you want more info about NVM: https://github.com/nvm-sh/nvm#installing-and-updating

Start up

Run **npm install** in the **main** directory and the **client** directory.

Run **npm start** in **main** directory; the backend api will run on localhost:5000.

Run **npm start** in the **client** directory; the React app will run on localhost:3000.
To clarify, you should have two terminals. One terminal will be in the main directory, the other terminal will be in the client directory. Run npm start on both terminals, order does not matter so long as they start at around the same time.

Deployment Instructions (Don't need to use until ready for deployment)
git push heroku main

To login, make sure to make a .env file in the main directory with the contents:

```
PGHOST = "acmatchmaker.c5qxtsrwvddh.us-east-1.rds.amazonaws.com"
PGPORT = "5432"
PGDATABASE = "ACMatchMaker"
PGUSER = "ACmm"
PGPASSWORD = "ACmmDev115b!"
PORT = 3001
```
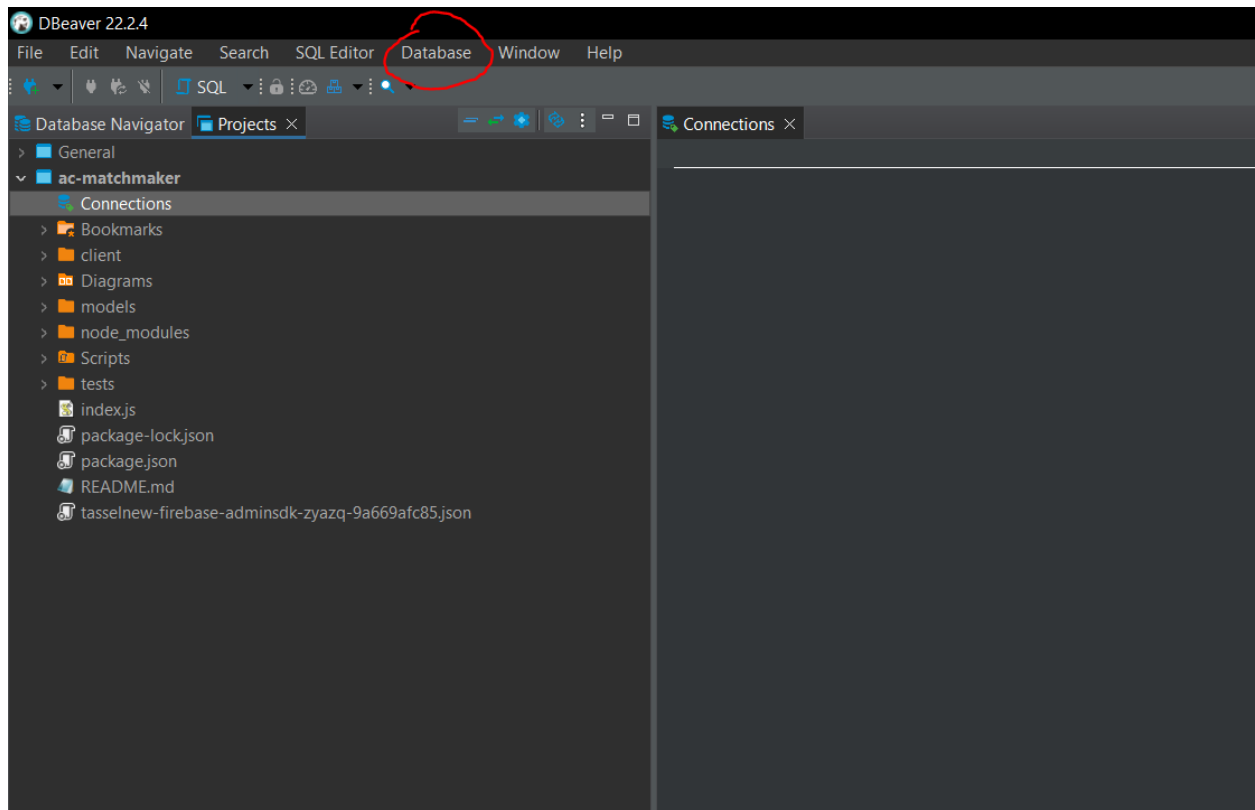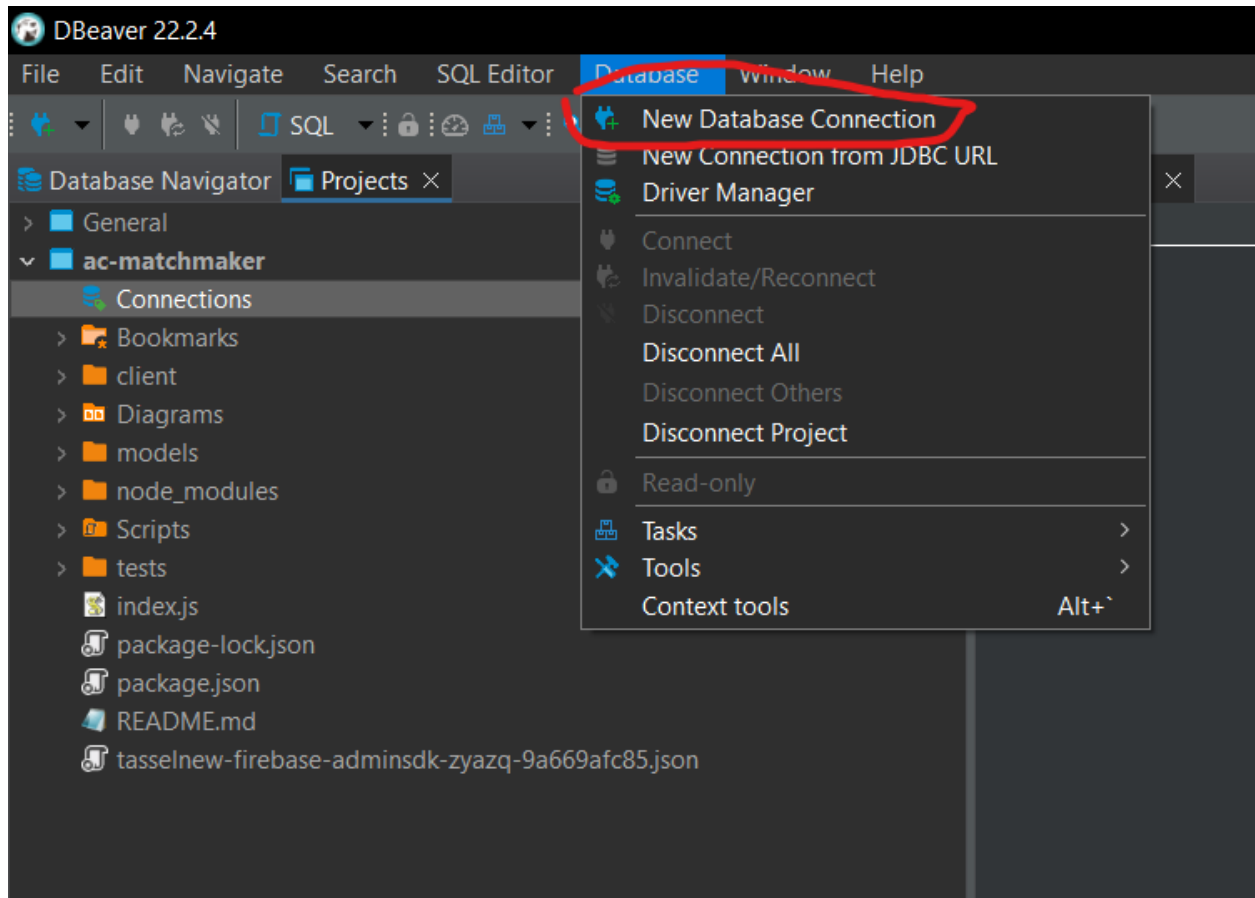
Wireframe

UI design:
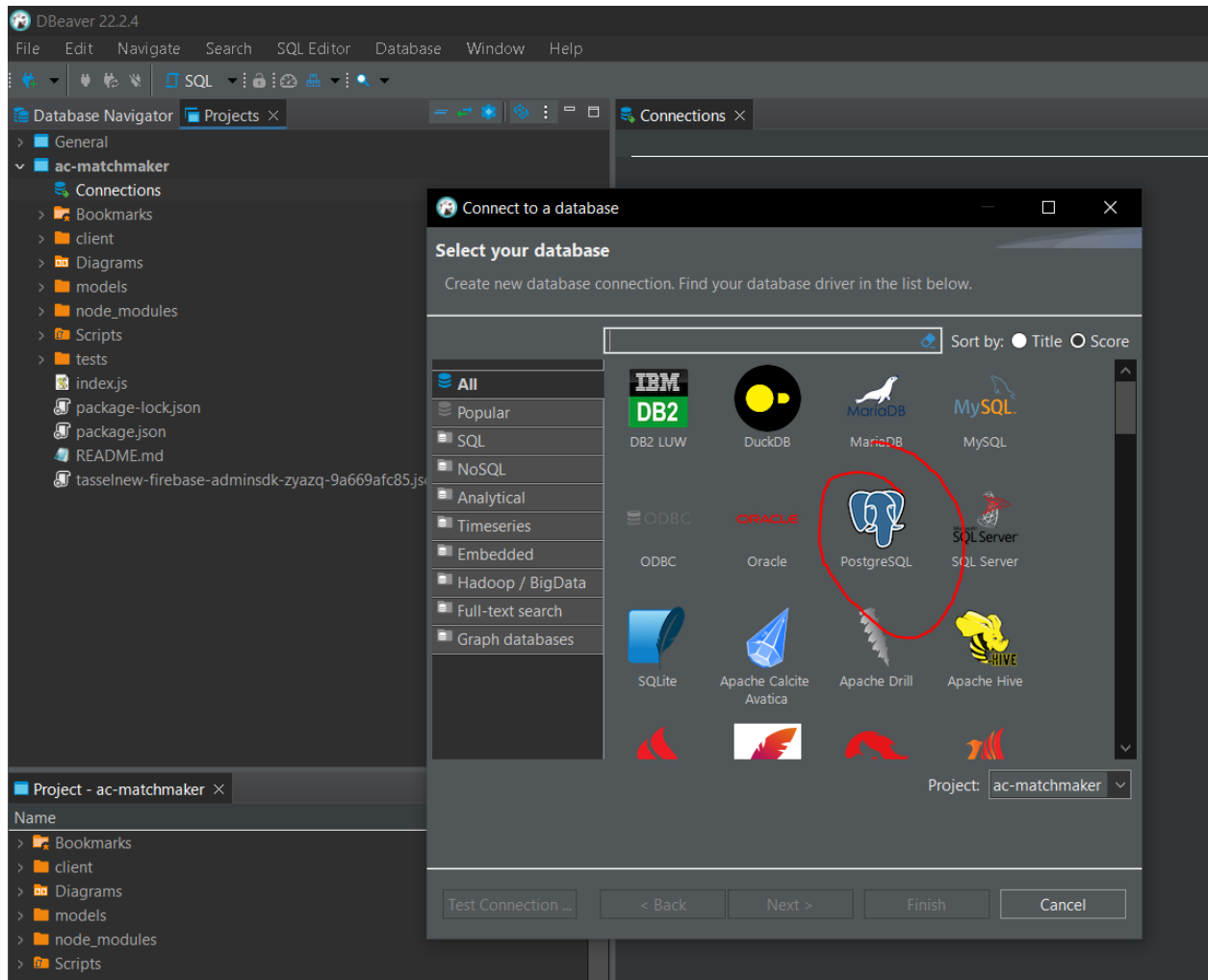
We hope this helps to understand the intended UI design.

How to access the database

Download a PSQL client software application. After accessing the application and opening Tassel (AC matchmaker, Tassel's former name), form a connection with Tassel and use the information from .env to login.
Example with DBeaver:
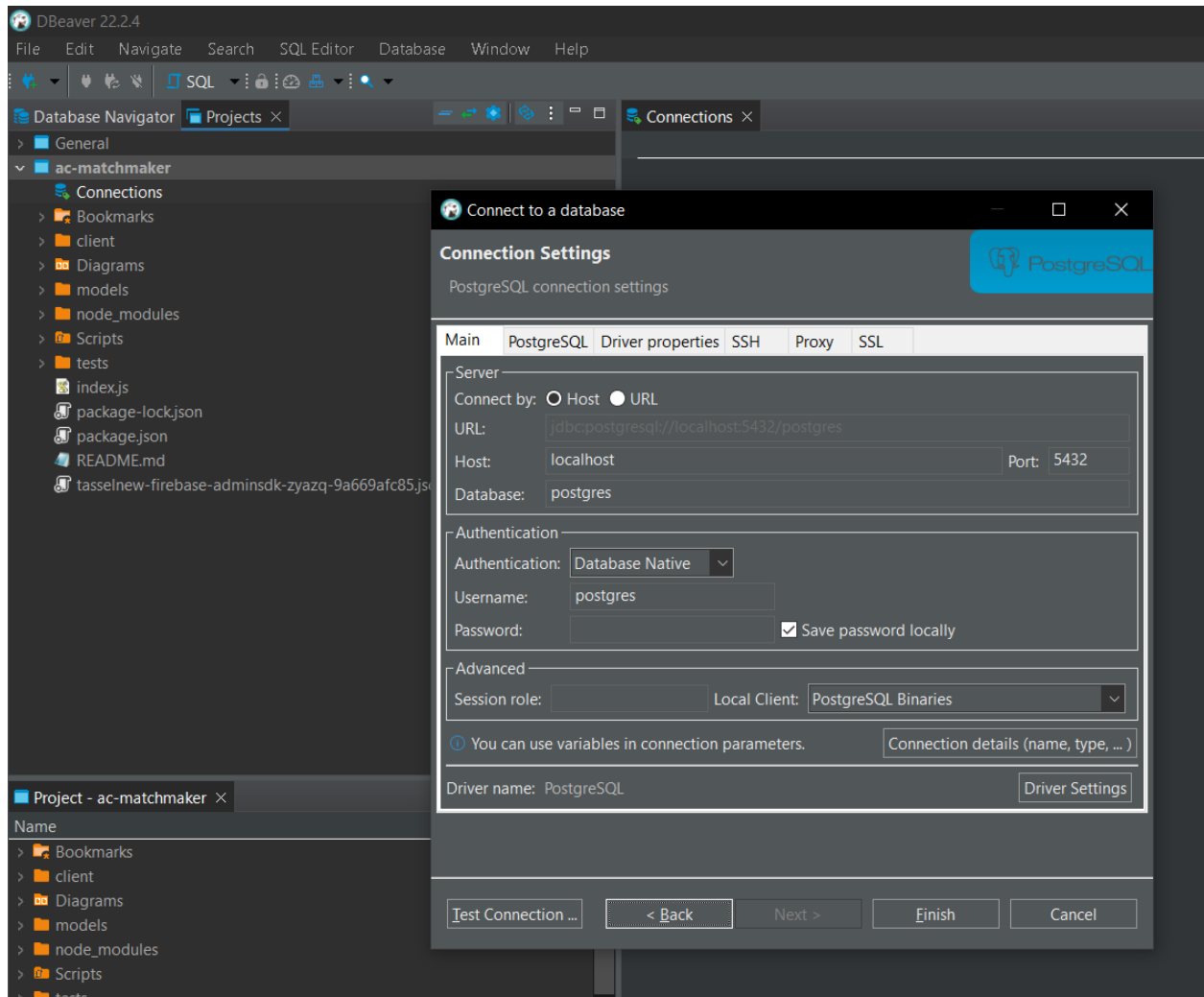
Input the .env info here ^.

To Be completed:

1. Admin Accounts
   [P2]As a website admin, I want to be able to approve or deny opportunities/users.
2. Opportunity events creation and customization
   [P2]As a host, I want to be able to create/modify an opportunity
   a. Set up a create button for an opportunity
   b. Set up edit button for an existing opportunity
   c. Modify existing opportunity data
   d. Display modified opportunity
3. Firebase

[P2]As a user, I want my account to be secured. (set up authentication with a third party)

   a. Authentication
      i. The current authentication is self made and uses JWT(JSON Web Tokens). The goal is to replace the existing authentication with Firebase.
   b. Notification
      i. Supposedly, the previous team was still developing the notification system using Firebase
   c. Connect Firebase with PSQL database ([Firebase PostgreSQL Integration: 2 Easy Methods (hevodata.com)](https://hevodata.com))(probably won't need to)
   d. Try to see if you can use firebase for the recommendation system too, otherwise, use O'reilly recommendation system course.

*Note 1: Webpack is already in Tassel project. No need to install
*Note 2, recommended resources for learning firebase:
   ● [Add Firebase to your JavaScript project (google.com)](https://google.com)
   ● [Get Started with Firebase Authentication on Websites (google.com)](https://google.com)
   ● ▶ React Firebase Authentication Tutorial | Firebase 9 Tutorial
*Note 3: I recommend looking into:
   ● client/src/app/pages/
      ○ Login.js (has some of the authentication happens and login button)
      ○ Signup.js (has some of the authentication happens and signup button)
      ○ MyProfile.js
   ● client/src/app/util/
      ○ AuthContext.js
   ● client/src/app/components/
      ○ NavBarLoggenIn.js (to get the logout button, line 280)

4. Recommendation System
   [P4]As a user, I want to be prompted with recommended events based on friends and/or my profile. As an event organizer, I want to be prompted with recommended volunteers based on their profiles and/or friends.
      ● [Library Access (oreilly.com)](https://oreilly.com)
      ● Add tags to user's profiles and opportunities
5. Search
   [P3]As a host, I want to search for volunteers and invite them to my opportunity.
   [P3] As a volunteer, I want to search for opportunities, and request to join.
6. Collaboration
   [P4] As a host, I want to be able to co-host with another host for an opportunity.
7. Send first and last name from the signup page to the profile

[P1] As a user, I want my name to be on my profile so that I can be properly identified.
8. Database
   Still need to update some of the input fields in the database

Debugging

- Use developer tools to access the console in the browser.
- Note: When you first open dev tools, you'll be on element first, you have to switch to console using the tabs at the top.
- You can also use networks to see if data was successfully sent.

System and unit test report

Instead of unit tests, we did functional integration testing.

Working prototype known problem(s)

None known besides what needs **to be completed.**

Things accomplished

- Creating an account and logging in.
- New update profile webpage to add new information on profiles. *Doesn't include all profile fields
- Created new documentation and coding practices for future collaboration

Sprint Board

https://trello.com/invite/b/2gB4E2eZ/ATTI4573b5ab43202d95fe71c6316f92dce27199C750/trello-agile-sprint-board-template

Backlog User Stories

[P2]As a user, I want my account to be secured. (set up authentication with a third party)

[P2]As a website admin, I want to be able to approve or deny opportunities/users.

[P3]As a host, I want to search for volunteers and invite them to my opportunity.

[P3] As a volunteer, I want to search for opportunities, and request to join.

[P4] As a host, I want to be able to co-host with another host for an opportunity.

[P4]As a user, I want to be prompted with recommended events based on friends and/or my profile. As an event organizer, I want to be prompted with recommended volunteers based on their profiles and/or friends.

+ Add a card

User Stories

[P2]As a user, I want to have access to my profile to view and edit/customize

[P2]As a host, I want to be able to create/modify an opportunity

+ Add a card

Tasks to start

Set up edit button for an existing opportunity

Modify existing opportunity data
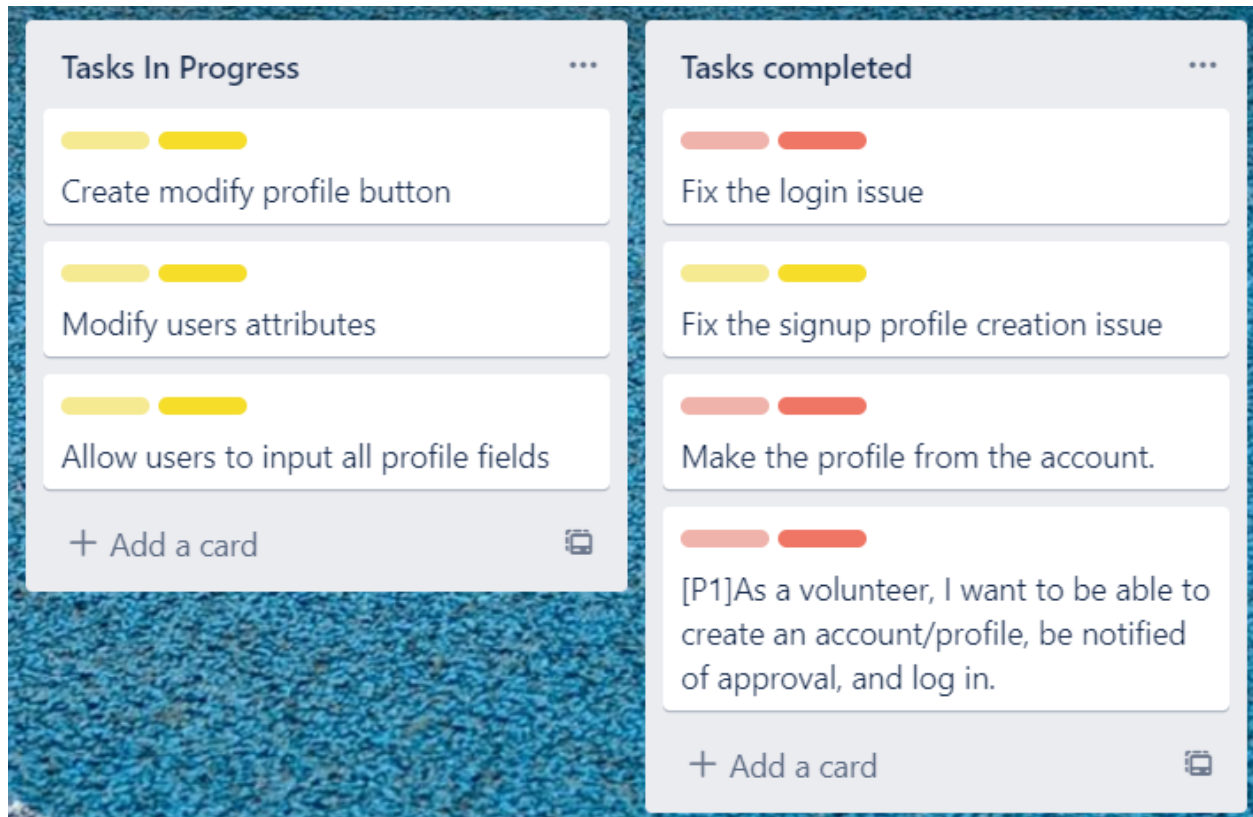
Display modified opportunity

Add data to a csv table (13 points) (needs to be broken down)

[FE] Set up the "Create an opportunity" page (3 points)

Post opportunity to model (? points)

New backend function for opportunities

+ Add a card

Tasks In Progress ...

Create modify profile button

Modify users attributes

Allow users to input all profile fields

+ Add a card

Tasks completed ...

Fix the login issue

Fix the signup profile creation issue

Make the profile from the account.

[P1]As a volunteer, I want to be able to create an account/profile, be notified of approval, and log in.

+ Add a card

Style guide

The client code uses React styling which can be found on [the React website](#)

1. Naming
   a. Use Camel case for variable and function names
2. Declaration
   a. Initialize and declare within the same line
      int x = 5;   // Good
      int x;   // Bad
      x = 5;

3. Spacing and Alignment
   a. Spaces around operators
   b. Use 2 spaces to indent code
   c. Use a new line for the trailing arguments if line length > 100
   d. Space after //
   e. Space between elements in a list
      const cars = ["Volvo", "Saab", "Fiat"];
4. Complex statements
   a. Put the opening bracket at the end of the first line.

b. Use one space before the opening bracket.
c. Put the closing bracket on a new line, without leading spaces.
d. Do not end a complex statement with a semicolon.

```
function toCelsius(fahrenheit) {
  return (5 / 9) * (fahrenheit - 32);
}
```

5. Object rules
   a. Place the opening bracket on the same line as the object name.
   b. Use colon plus one space between each property and its value.
   c. Use quotes around string values, not around numeric values.
   d. Do not add a comma after the last property-value pair.
   e. Place the closing bracket on a new line, without leading spaces.
   f. Always end an object definition with a semicolon.

```
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
```

6. Comments
   a. Place comments above the function
   Note: In the comment, explain both how its functionality ties to high-level goal and what the function does.

Definition of done

1. Checked into the repository
   a. The code should be pushed and merged into the main branch. All merge conflicts should be resolved.
   b. Brief commit message describing what changes have been made to the code base.
2. Reviewed for standards compliance
   a. All compile errors have been removed or documented
   b. Code is checked by the coder and other team members to ensure consistency in style and functionality
3. Code is reviewed by team member
   a. Team member is walked through what has been changed and how the new change works in the project

          b.   Team member approves the style and edits made
   4. External API documented
          a.   API is documented along with any external processing API used to retrieve data
   5. Documentation for code change and build
          a.   Any changes in the code and build system are documented and shared within the group discord so everyone can review
   6. Non-functional test pass
          a.   Test on user interface and usability passed
          b.   New lag isn't introduced into the system or the generation of shapes

Review

When we initially joined the project, we planned on building a recommendation engine and incentive features. After gaining access to the application, we realized the state of the current project was not where we needed it to be. Thus, we had to shift our focus to adding basic features. Since that didn't go as smoothly as planned, another aspect we focused on was documenting as well as we could. The current Tassel still needs development in some basic functionalities. It is recommended that future team members have knowledge of JS, React, Node.js, HTML, and/or PSQL. With this, we hope our documentation helps.