

CIRCUIT CONNECTION

Ashwanth & Kyle

GOALS

```
1 ## APP IDEA:
2
3 - Social Media for Web Developers (to create and customize their portfolios, showcase it, and see each others)
4
5 ### TECHNICAL GOALS:
6
7 - Use AngularJS partials to allow simultaneous web production.
8 - Maintain a good workflow in terms of branching, commits, issues and waffle as a daily tracker
9 - Github API (Social Authorization)
10 - Figure out Google Maps inclusion
11
12 ### FUNCTIONALITY GOALS:
13
14 - Portfolio information that links to your sites (add, edit, and delete it) - min functionality
15
16 ### User Stories:
17
18 MVP:
19
20 - User can log in/create account/log out.
21 - User can create a portfolio.
22 - User can look at other user's portfolios.
23 - User can edit own portfolio information.
24 - User can delete their own portfolio.
25 - Portfolio level styling.
26
```

MVP

- **MEAN stack** -
(We're very sorry, we had to use EJS for maps)
- **MVC**
- **One model with full CRUD** (USER model)
- **Authorisation** - Sign Up / Log In

USER model

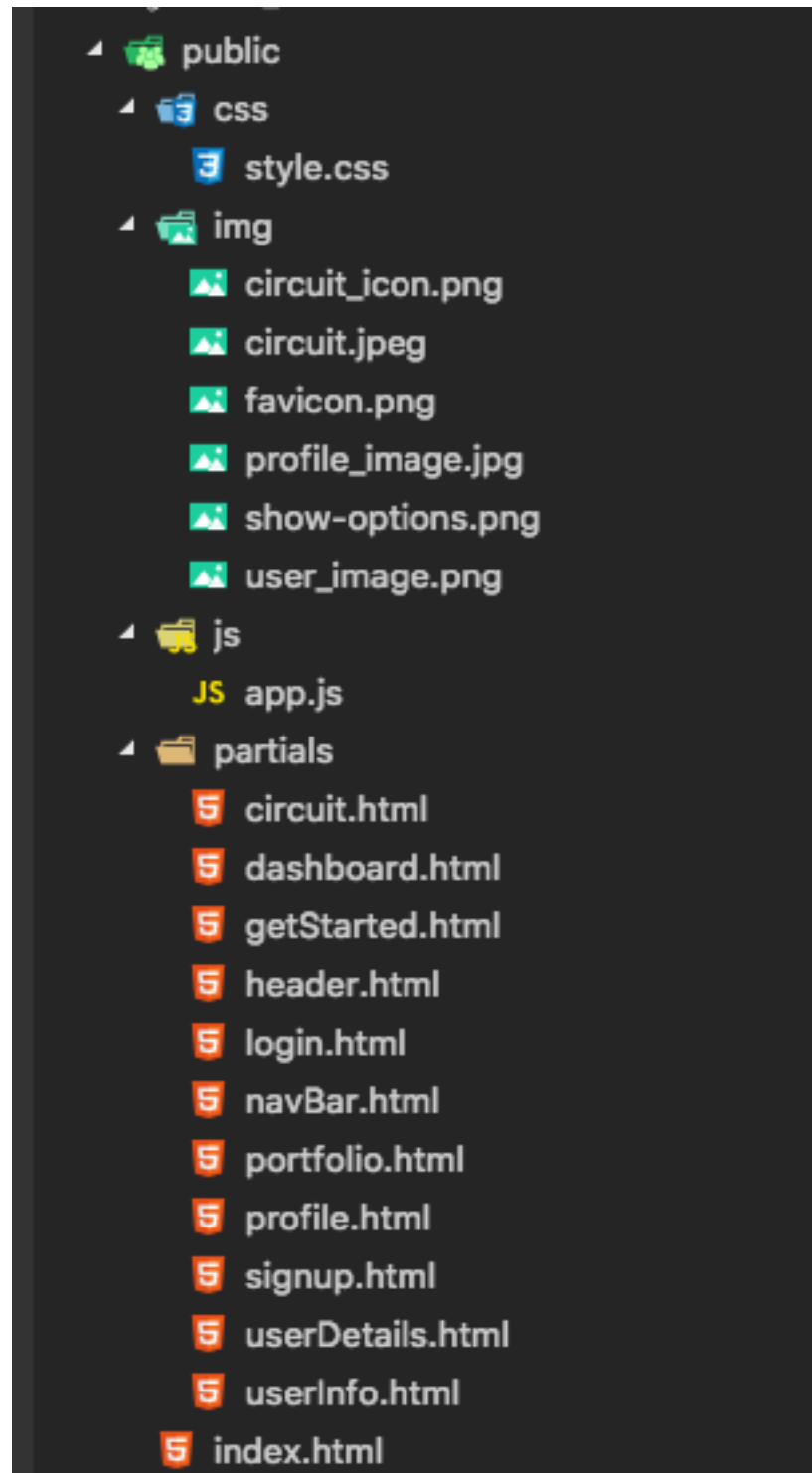
```
4  const userSchema = Schema({
5    username: { type: String, required: true },
6    password: { type: String, required: true },
7    comments: [String],
8    portfolioCustomizations: String,
9    profilePic: String,
10   followers: { type: Number, default: 0 },
11   projects: { type: Number, default: 0 },
12   circuitPoints: { type: Number, default: 0 },
13   github: String,
14   githubUserToken: { type: String, default: null },
15   name: String,
16   brandStatement: String,
17   proficiency: String,
18   skills: [String],
19   interests: [String],
20   location: String
21 });
22
```

Daily Tracker - waffle.io

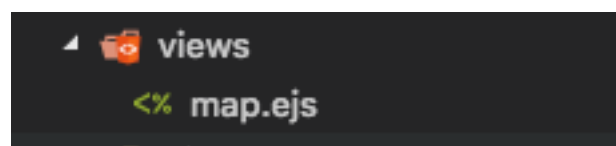
The screenshot displays a Kanban board with six columns: Inbox (0 items), Backlog (1 item), In Progress (1 item), Review (0 items), Bugs (1 item), and Done (36 items). Each column contains task cards with details such as ID, title, category, and status.

Column	Count	Item ID	Title	Category	Status
Inbox	0	-	-	-	-
Backlog	1	44	Add a basic Portfolio page	enhancement	Backlog
In Progress	1	45	Add a basic Circuit page	enhancement	In Progress
Review	0	-	-	-	-
Bugs	1	48	If you log in with an incorrect username/password the server breaks	Bugfix	Bugs
Done	36	43	Get OAuth working with GitHub - just the basics	enhancement	Done
Done	36	37	Add image option for profile	Bugfix	Done
Done	36	47	Adding basic google maps to site	enhancement	Done
Done	36	25	Add animating header	enhancement	Done
Done	36	6	User can delete their own profile.	MVP	Done
Done	36	39	Make nav bar switch when clicking on portfolio or circuit	Bugfix	Done
Done	36	42	Add Delete Button	MVP	Done
Done	36	32	ng-repeat on skills was breaking	-	Done

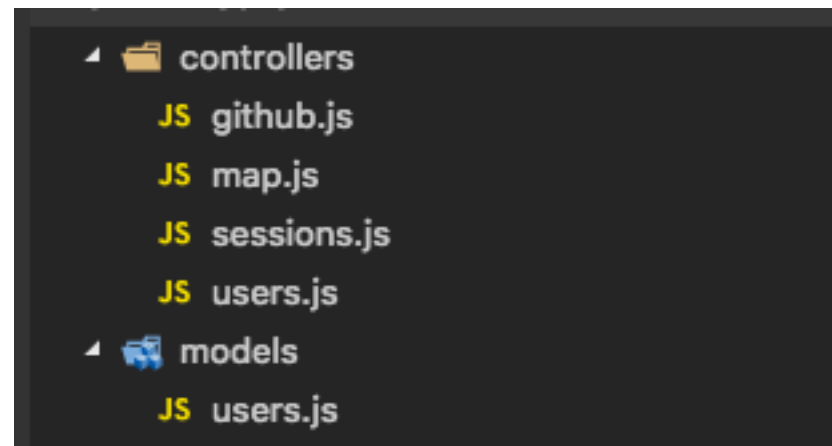
VIEWS



OOPS!!



MVC



CONTROLLERS

MODEL

ANGULAR PARTIALS & ROUTING

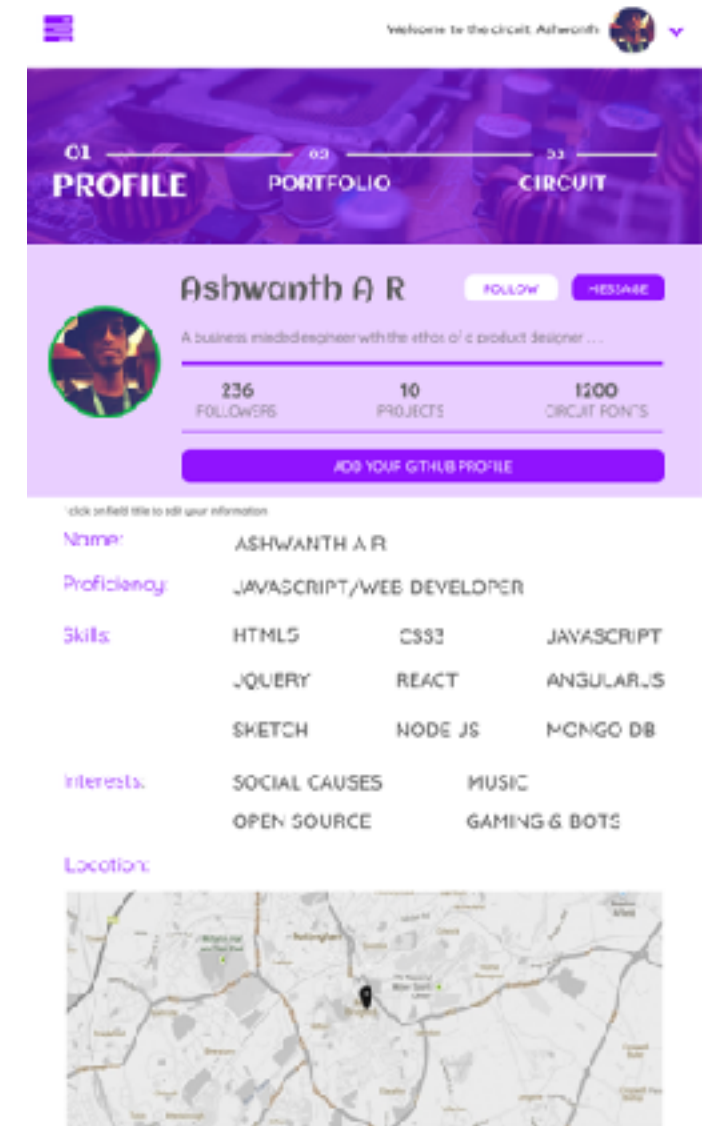
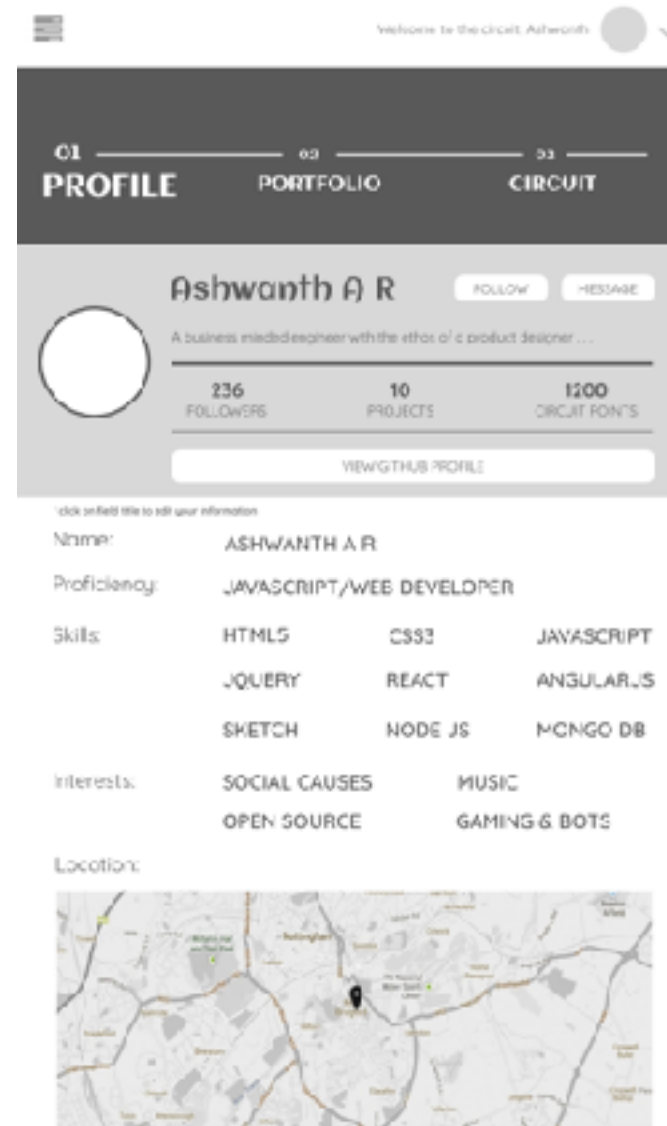
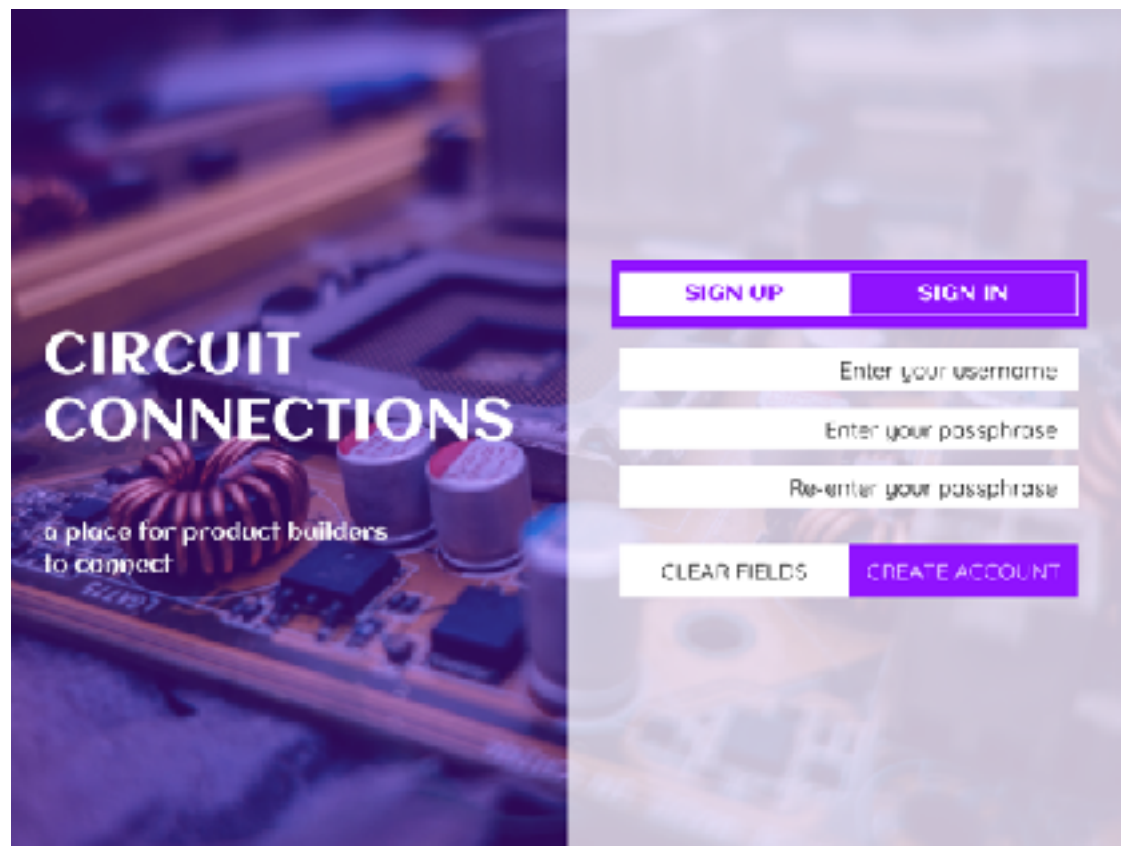
```
// ----- Partials Logic -----
this.includePath = "partials/signup.html";

this.pagePath = "partials/getStarted.html";

this.tabPath = "partials/profile.html";

//=====
// FUNCTION FOR MOVING BETWEEN PAGES ON APP
//=====
this.changePagePath = path => {
  this.pagePath = `partials/${path}.html`;
};
```

Portfolio Quality Styling



Making good plans and progress . . .



**SUNDAY
AFTERNOON -
MVP DONE**

**SMOOTH SAILING
SO FAR . . .**





Oops . . .

We spoke too soon



OAUTH HAD OTHER PLANS FOR US

ANGULAR JS HATES US - MAPS

AJAX REFUSES TO MAKE REQUESTS

THE ROOKIE MISTAKE

THANK GOD FOR MATT AND DAN!!

OAUTH - THE PROCESS

You can list your application in the [GitHub Marketplace](#) so that other users can discover it.

[List this application in the Marketplace](#)

0 users

Client ID
6d59e7b9f70ef40e6417

Client Secret
eb0992a2e191aef63e7cf46a5a9079b89f2e359d

[Revoke all user tokens](#) [Reset client secret](#)

Application name
CircuitConnection
Something users will recognize and trust

Homepage URL
<https://circuit-connection.herokuapp.com>
The full URL to your application homepage

Application description
Application description is optional
This is displayed to all users of your application

Authorization callback URL
<https://circuit-connection.herokuapp.com/github/callback>
Your application's callback URL. Read our [OAuth documentation](#) for more information.

CLIENT ID, CLIENT SECRET AND CALLBACK URL

STEP 1 - CREATING THE APPLICATION IN GITHUB DEVELOPER SETTINGS

OAUTH - THE PROCESS



STEP 2 - USER CLICKS THE GET DATA FROM GITHUB BUTTON

```
// .....
this.getGithubData = () => {
  const authorize_url = "https://github.com/login/oauth/authorize";
  const redirect_uri = "http://localhost:3000/github/callback";
  const encoded_redirect_uri = encodeURIComponent(redirect_uri);
  console.log(encoded_redirect_uri);
  const client_id = "806f2c024e556b9cd021";
  $http({
    method: "GET",
    url: `/github/${this.currentUser._id}`
  }).then(
    response => {
      console.log(response);
      if (response.data === "User Token Not Found") {
        console.log(
          `should redirect to authorize url using window location`
        );
        $window.location.href = `${authorize_url}?scope=user%3Aemail&client_id=${client_id}&redirect_uri=${encoded_redirect_u
      } else {
        console.log(response);
      }
    },
    err => {
      console.log(err);
    }
  );
};
};
```

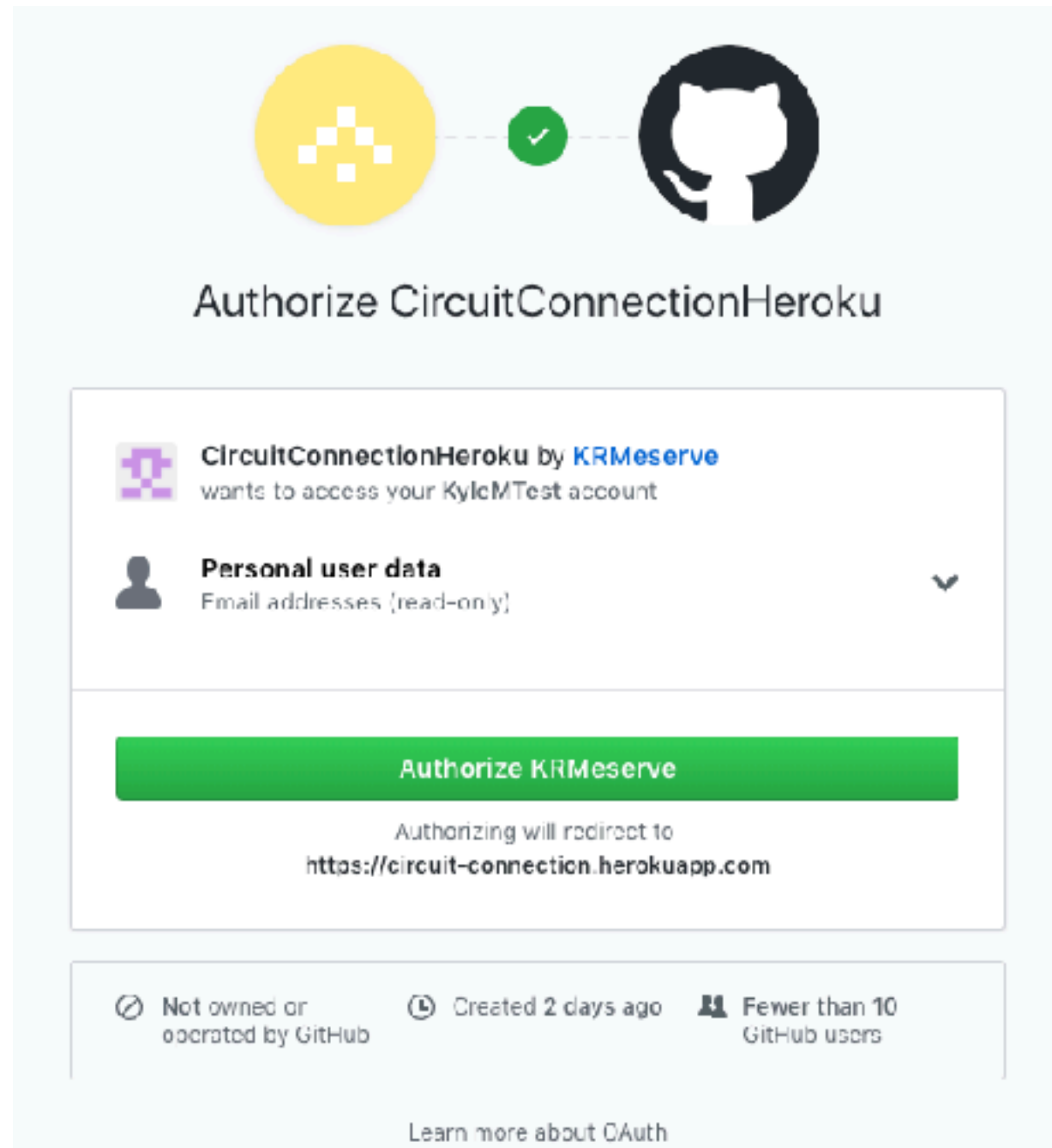
OAUTH - THE PROCESS

```
//=====
// RECEIVE DATA FROM GET REQUEST
//=====
router.get("/:id", (req, res) => {
  const { id } = req.params;
  console.log(id);
  User.findById(id, (err, foundUser) => {
    if (err) {
      console.log(err);
    } else {
      console.log(foundUser);
      if (foundUser.githubUserToken !== null) {
        console.log(`user token was already there`);
        access_token = foundUser.githubUserToken;
        getAndDisplayData(res, req);
      } else {
        console.log(`user token is not found`);
        res.json("User Token Not Found");
      }
    }
  });
});
```

STEP 3 - CHECKS IF ACCESS TOKEN ALREADY EXISTS FOR USER

We didn't end up saving token because we got it working so late

OAUTH - THE PROCESS



STEP 4 - GITHUB SHOWS AUTHORISATION FORM AND USER AUTHORIZES

OAUTH - THE PROCESS

```
54 //=====
55 // AFTER 1st TIME AUTHORIZATION, GITHUB
56 // REDIRECTS TO THIS URL
57 //=====
58 router.get("/callback", (req, res) => {
59   console.log(`entering callback route`);
60   const auth_code = req.query.code;
61   console.log(auth_code);
62   console.log({
63     client_id: client_id,
64     client_secret: client_secret,
65     code: auth_code,
66     redirect_uri: redirect_uri
67   });
68   request( ...
90 );
91 });
92
```

STEP 5 - GITHUB SENDS BACK AN AUTH CODE AS QUERY PARAMETER TO CALLBACK URL WHICH WAS SENT EARLIER AS REDIRECT URL

OAUTH - THE PROCESS

```
request(  
  {  
    uri: token_url,  
    method: "POST",  
    form: {  
      client_id: client_id,  
      client_secret: client_secret,  
      code: auth_code,  
      redirect_uri: redirect_uri  
    }  
  },  
  function(err, response, body) {  
    //=====  
    // RETRIEVE ACCESS TOKEN FROM BODY  
    //=====  
    access_token = body.split("&")[0].split("=")[1];  
    console.log(access_token);  
    //=====  
    // LATER: NEED TO SAVE ACCESS TOKEN IN DB FOR EACH USER  
    //=====  
    getAndDisplayData(res, req);  
  }  
);
```

**STEP 6 - FROM CALLBACK URL WE MAKE A REQUEST FOR ACCESS TOKEN
USING THE AUTH CODE THAT WE JUST GOT**

OAUTH - THE PROCESS

```
9 //=====
0 // GET AND DISPLAY DATA
1 //=====
2 const getAndDisplayData = (res, req) => {
3   console.log(
4     "access token was found and we're inside get and display data fn"
5   );
6   request(
7     {
8       uri: `${api_url}?access_token=${access_token}`,
9       method: "GET",
10      headers: {
11        "User-Agent": "CircuitConnection"
12      }
13    },
14    function(err, response, body) {
15      // const image = body.avatar_url;
16      // console.log(body.avatar_url);
17      // res.json(JSON.parse(body));
18      req.session.github = JSON.parse(body);
19      res.redirect("/");
20    }
21  );
22 };
```

STEP 7 - ONCE WE HAVE ACCESS TOKEN WE MAKE A GET REQUEST TO THE API URL AND SAVE THIS DATA TO SESSION OBJECT AND REDIRECT HOME

OAUTH - THE PROCESS

```
this.goToSession = () => {  
  $http({  
    method: "GET",  
    url: "/sessions"  
  }).then(response => {  
    console.log(response);  
    if (response.data.currentUser) {  
      this.pagePath = "partials/dashboard.html";  
      console.log(response.data);  
      this.currentUser = response.data.currentUser;  
    }  
  })  
}
```

STEP 8 - WE RETRIEVE USER FROM SESSION AND AUTOMATICALLY ROUTE TO DASHBOARD SO THAT THE USER DOESNT HAVE TO LOG IN AGAIN

OAUTH - THE PROCESS

```
this.goToSession = () => {  
  $http({  
    method: "GET",  
    url: "/sessions"  
  }).then(response => {  
    console.log(response);  
    if (response.data.currentUser) {  
      this.pagePath = "partials/dashboard.html";  
      console.log(response.data);  
      this.currentUser = response.data.currentUser;  
    }  
  })  
}
```

STEP 8 - WE RETRIEVE USER FROM SESSION AND AUTOMATICALLY ROUTE TO DASHBOARD SO THAT THE USER DOESNT HAVE TO LOG IN AGAIN

OAUTH - THE PROCESS

```
this.image = response.data.github.avatar_url;
this.followers = response.data.github.followers;
this.projects = response.data.github.public_repos;
const { following, public_gists } = response.data.github;
this.circuitPoints =
  this.followers * 10 +
  this.projects * 5 +
  following +
  public_gists * 3;
$http({
  method: "PUT",
  url: "/users/" + this.currentUser._id,
  data: {
    profilePic: this.image,
    followers: this.followers,
    projects: this.projects,
    circuitPoints: this.circuitPoints
  }
}).then(
  response => {
    console.log(response);
  },
  error => {
    console.log(error);
  }
);
}
```

**STEP 9 - WE RETRIEVE THE DATA STORED IN GITHUB PROPERTY
IN SESSION OBJECT**

ANGULAR JS HATES US - MAPS

Location: CLICK TO SET LOCATION

SUBMIT

SHOW MAP


```
// ----- GOOGLE MAPS STUFF -----  
this.goToMap = () => {  
  console.log("entering function goToMap");  
  $window.location.href = "/map";  
};
```


```
router.get('/', (req, res) => {  
  console.log('got to map router');  
  const latitude = req.query.lat;  
  const longitude = req.query.lon;  
  res.render('map.ejs', {  
    latitude: latitude,  
    longitude: longitude,  
    apiKey: process.env.MAP_API_KEY  
  });  
});
```


**WE HAD ISSUES WITH SCRIPTS
RUNNING ON LOAD**


**SO WE HAD TO USE EJS TO
CALL A SEPARATE PAGE
TO SHOW MAPS**


ANGULAR JS HATES US - MAPS

 Google Maps

 Overview

 APIs

 Metrics


 Support

← Maps JavaScript API ■ DISABLE

Metrics Quotas Credentials

Use one of these credentials to access this API, or create new credentials by visiting [Credentials in the API Manager](#).

API keys

Name	Creation date ▾	Restrictions	Key
 API key 1	Oct 30, 2018	None	AlzaSyCKRNX1MabH8ONqKrBBPy

**FAIRLY STRAIGHTFORWARD TO FOLLOW DOCUMENTATION
CUZ GOOGLE DOCUMENTS THEIR STUFF WELL**

Welcome to the circuit, Kyle



DELETE ACCOUNT

LOG OUT

```
404  /*===== KEYFRAMES =====*/
405  @keyframes flip {
406    0% {transform: rotate(0deg);}
407    100% {transform: rotate(180deg);}
408  }
409  @keyframes flipReset {
410    0% {transform: rotate(180deg);}
411    100% {transform: rotate(0deg);}
412  }
413  @keyframes hiddenBoxAnimationAppear {
414    0% {
415      transform: translate(0, -50px);
416    }
417    100% {
418      transform: translate(0, 0);
419    }
420  }
421  @keyframes hiddenBoxAnimationDisappear {
422    0% {
423      transform: translate(0, 0px);
424    }
425    100% {
426      transform: translate(0, -100px);
427    }
428  }
429
```

```
this.removeHiddenBox = () => {
  console.log("running");
  $("#hiddenLogOutBox").addClass("hiddenBox");
};

// ----- Function to hide/show logout button on profile screen -----
this.logoutHidden = () => {
  this.hideLogout = !this.hideLogout;
  if (this.hideLogout) {
    $("#show-options-image").addClass("flipped");
    $("#show-options-image").removeClass("flippedReset");
    $("#hiddenLogOutBox").removeClass("hiddenBox");
    $("#hiddenLogOutBox").addClass("hiddenBoxAnimationAppear");
    $("#hiddenLogOutBox").removeClass("hiddenBoxAnimationDisappear");
  } else {
    $("#hiddenLogOutBox").removeClass("hiddenBoxAnimationAppear");
    $("#hiddenLogOutBox").addClass("hiddenBoxAnimationDisappear");
    $("#show-options-image").removeClass("flipped");
    $("#show-options-image").addClass("flippedReset");
    setTimeout(this.removeHiddenBox, 150);
  }
};
```

Brand Statement:

A business minded engineer with the ethos of a product designer

Proficiency:

Javascript / Web Developer

```
379 //=====
380 // Keep track of profile field edit states
381 //=====
382 this.nameEdit = false;
383 this.brandEdit = false;
384 this.proficiencyEdit = false;
```

```
455 this.toggleProficiencyEdit = () => {
456   this.proficiencyEdit = !this.proficiencyEdit;
457   if (this.proficiencyEdit) {
458     if (this.proficiency === "CLICK HERE TO ADD PROFICIENCY") {
459       this.proficiency = "";
460     }
461     $((() => {
462       $("#proficiencyEditField").focus();
463     }));
464   } else {
465     console.log(`need to post proficiency`);
466     $http({
467       method: "PUT",
468       url: "/users/" + this.currentUser._id,
469       data: {
470         proficiency: this.proficiency
471       }
472     }).then(
473       response => {
474         console.log(response);
475       },
476       error => {
477         console.log(error);
478       }
479     );
480   }
481 };
```

```
9 </div class= height40 ></div>
10 <div class="dFlex flexRow alignCenter">
11   <div class="flex1 fWeight300 fPurple">Brand Statement:</div>
12   <div ng-if="!ctrl.brandEdit" class="flex2 hoverGreen cPointer pad10" ng-click="ctrl.toggleBrandEdit()">{{ctrl.brandStatement}}</div>
13   <input ng-if="ctrl.brandEdit" type="text" class="flex2 hoverGreen cPointer pad10 fQuicksand fSize2" ng-click="ctrl.toggleBrandEdit()"
14     value="" id="brandEditField" ng-model="ctrl.brandStatement" />
15 </div>
```