Hooli XYZ

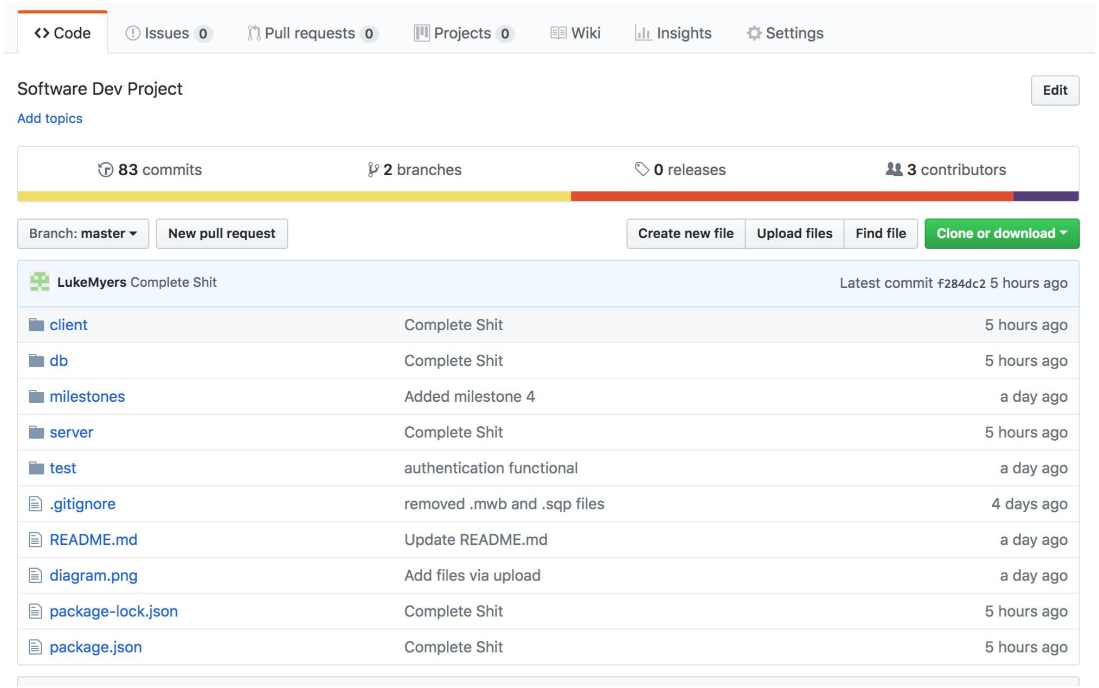
Ksenia Lepikhina, Peter Lindee, Andres Barrera, Noel Taterway, Wade Myers

About Hooli XYZ

- Goal: Create a CU specific “Github”
    - Create accounts with colorado.edu emails
    - Create private repos
    - Share repos with other students

VCS - Version Control System

- Ironically, chose to use Github



Software Dev Project

83 commits · 2 branches · 0 releases · 3 contributors

LukeMyers Complete Shit — Latest commit f284dc2 5 hours ago

client	Complete Shit	5 hours ago
db	Complete Shit	5 hours ago
milestones	Added milestone 4	a day ago
server	Complete Shit	5 hours ago
test	authentication functional	a day ago
.gitignore	removed .mwb and .sqp files	4 days ago
README.md	Update README.md	a day ago
diagram.png	Add files via upload	a day ago
package-lock.json	Complete Shit	5 hours ago
package.json	Complete Shit	5 hours ago

https://github.com/klepikhina/Hooli_XYZ

Benefits to Github

- Version control
- Branches allow for contributors to not step on each others toes
- Shared files
- Overall: A really useful too

★★★★★

Downside to Github

- Didn’t take advantage of branches

Git hell...

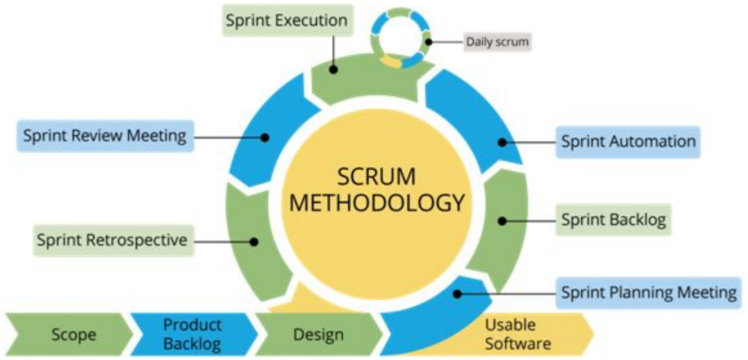
Methodologies

- Agile
  - Preference: continuous design ☺ vs design ☹ upfront
  - Target audience: CU Students
  - Customer interaction key for Agile

★★★★★

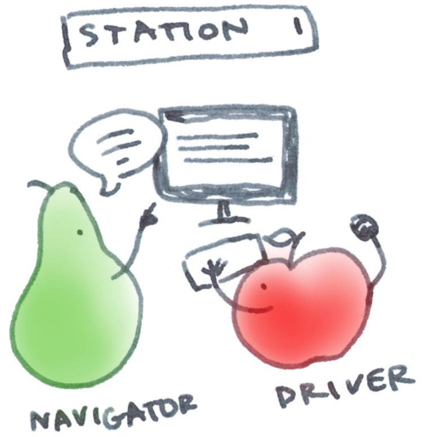
Methodologies

- Strategy:
  - Split up tasks (front end, back end, database) but allow each team member to move between teams as needed
  - Agile encourages a creative and self-organized environment
- Scrum
  - Scrum meetings occurred twice a week for approximately two hours



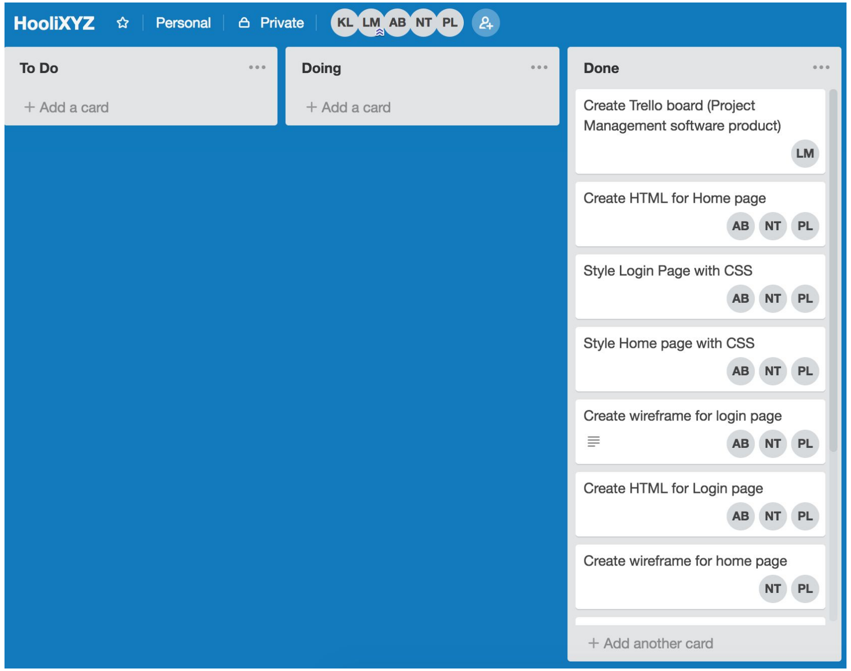
Methodologies

- Pair programming
  - Collaboration for creating HTML pages: Peter, Andres, Noel
  - Benefits: fewer mistakes, easier to keep going (moral support), shared best practices



Project Tracking

- Trello
  - Beneficial for assigning tasks to individuals
  - Beneficial for organizing a to do list and keeping track of what is being done
  - Can easily check things off of the list
  - No calendar


HooliXYZ ☆ Personal Private KL LM AB NT PL

To Do
+ Add a card

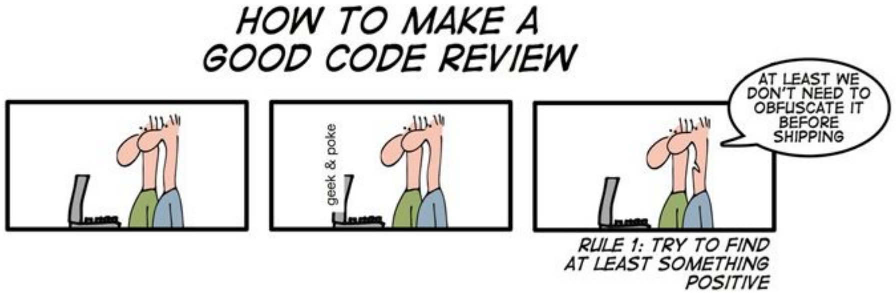
Doing
+ Add a card

Done
- Create Trello board (Project Management software product) — LM
- Create HTML for Home page — AB NT PL
- Style Login Page with CSS — AB NT PL
- Style Home page with CSS — AB NT PL
- Create wireframe for login page — AB NT PL
- Create HTML for Login page — AB NT PL
- Create wireframe for home page — NT PL

+ Add another card

Code Reviews

- Valuable for catching mistakes
- Mildly stressful but beneficial for limiting buggy code


HOW TO MAKE A GOOD CODE REVIEW
geek & poke
AT LEAST WE DON'T NEED TO OBFUSCATE IT BEFORE SHIPPING
RULE 1: TRY TO FIND AT LEAST SOMETHING POSITIVE

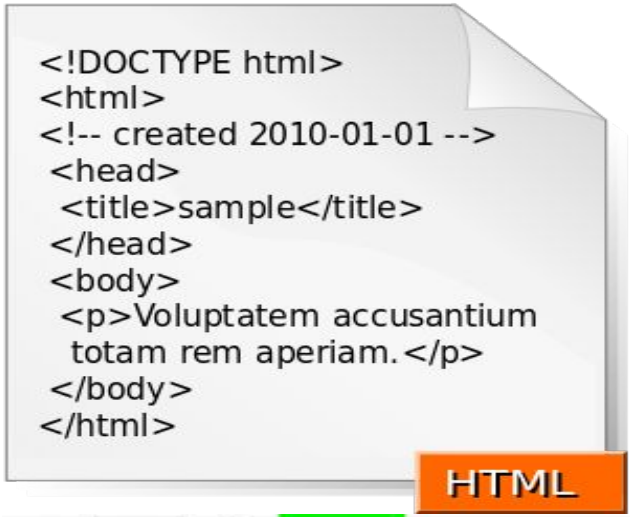
Slack

- Communication tool
- Effective for rapid communication
- Allows for group messaging as well as direct messaging
- Allows for a notification to be generated when needed
    - (i.e. “@here” sends all users in a channel a notification)

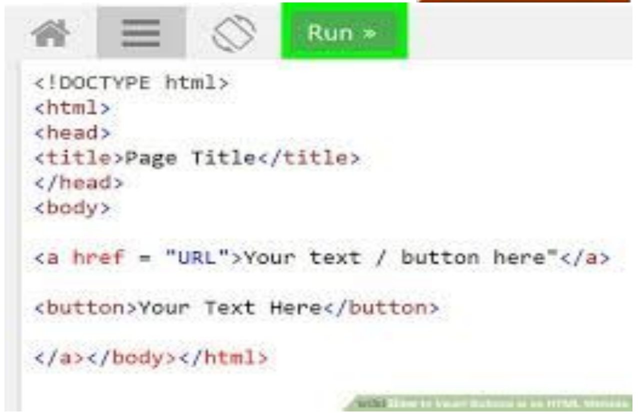
★★★★★

Front End

- Why our design?
  - Clean
  - Simple color scheme
  - Aesthetic appeal
- Impact on user
  - Makes the user feel like their files/ personally identifiable information (username, password, email address) are secure



```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
 <head>
  <title>sample</title>
 </head>
 <body>
  <p>Voluptatem accusantium
   totam rem aperiam.</p>
 </body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<a href = "URL">Your text / button here"</a>

<button>Your Text Here</button>

</a></body></html>
```

Server

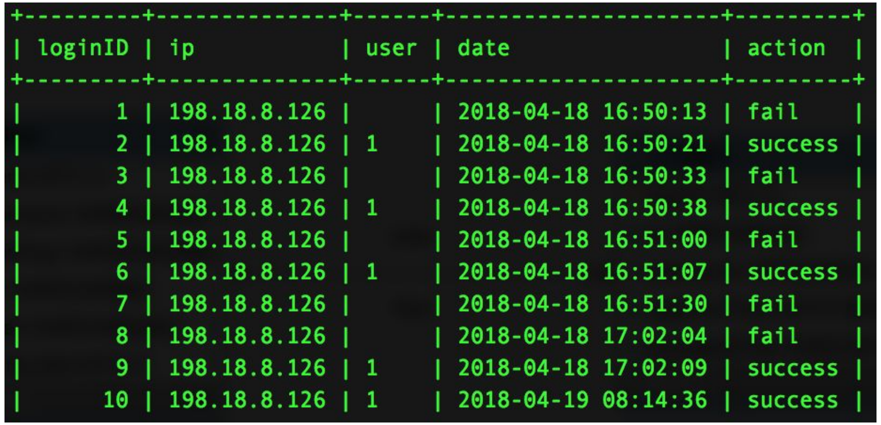
- Built with Express
- Page Routing
- Database REST API
- Asynchronous (Promise Based)

Database

- MySQL
  - Created the Hooli XYZ database
  - Plan: Folders, Files, Users, Whitelist, Login, UserFolder
  - Current: Files, Users

Plan for Database

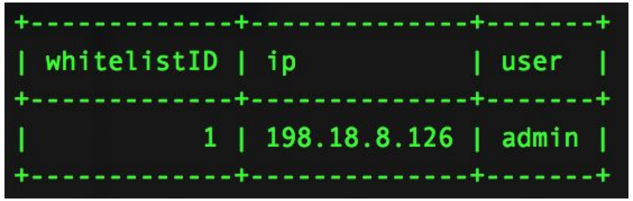
- Login Table:
  - Keep track of who tried to login
  - From which IP
  - When
  - Whether or not they were successful



```
+---------+--------------+------+---------------------+---------+
| loginID | ip           | user | date                | action  |
+---------+--------------+------+---------------------+---------+
|       1 | 198.18.8.126 |      | 2018-04-18 16:50:13 | fail    |
|       2 | 198.18.8.126 | 1    | 2018-04-18 16:50:21 | success |
|       3 | 198.18.8.126 |      | 2018-04-18 16:50:33 | fail    |
|       4 | 198.18.8.126 | 1    | 2018-04-18 16:50:38 | success |
|       5 | 198.18.8.126 |      | 2018-04-18 16:51:00 | fail    |
|       6 | 198.18.8.126 | 1    | 2018-04-18 16:51:07 | success |
|       7 | 198.18.8.126 |      | 2018-04-18 16:51:30 | fail    |
|       8 | 198.18.8.126 |      | 2018-04-18 17:02:04 | fail    |
|       9 | 198.18.8.126 | 1    | 2018-04-18 17:02:09 | success |
|      10 | 198.18.8.126 | 1    | 2018-04-19 08:14:36 | success |
```

Plan for Database

- Whitelist Table
  - Keep track of who is whitelisted
  - Prevents DDOS attacks by blocking users who were not able to login (after 5 attempts) and are not on the whitelist



```
+-------------+--------------+-------+
| whitelistID | ip           | user  |
+-------------+--------------+-------+
|           1 | 198.18.8.126 | admin |
+-------------+--------------+-------+
```

Plan for Database

- Folder Table:
  - Keep track of folders that belong to users
- User Table:
  - Keep track of users who have accounts on Hooli XYZ
- File Table:
  - Keep track of files in each folder
- UserFolder:
  - Resolve the many-many relationship between users and folders

Current

- Currently have the User table and Files table implemented and in use
- User authentication is enabled (check user table)

Testing Tool

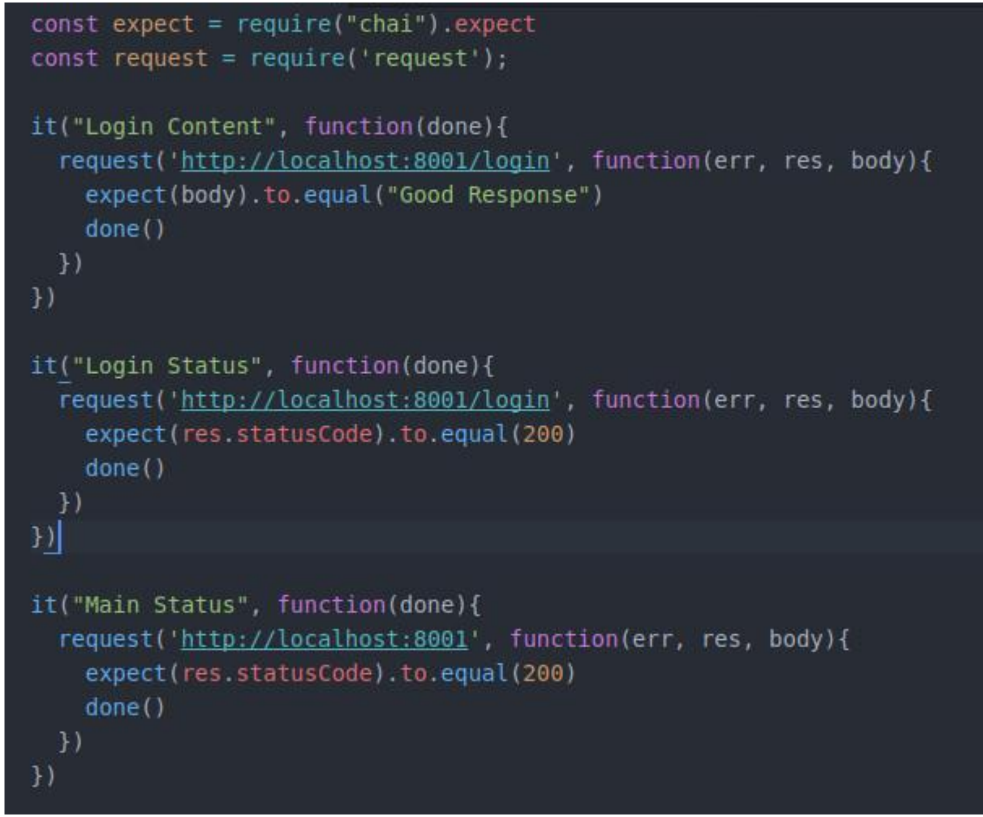
- Mocha
  - Node Testing Framework
  - async testing, accurate reporting
  - Not Python
- Chai
  - Javascript Assertion Library
  - Paired with Mocha

mocha chai

★★★★★

Testing Tool

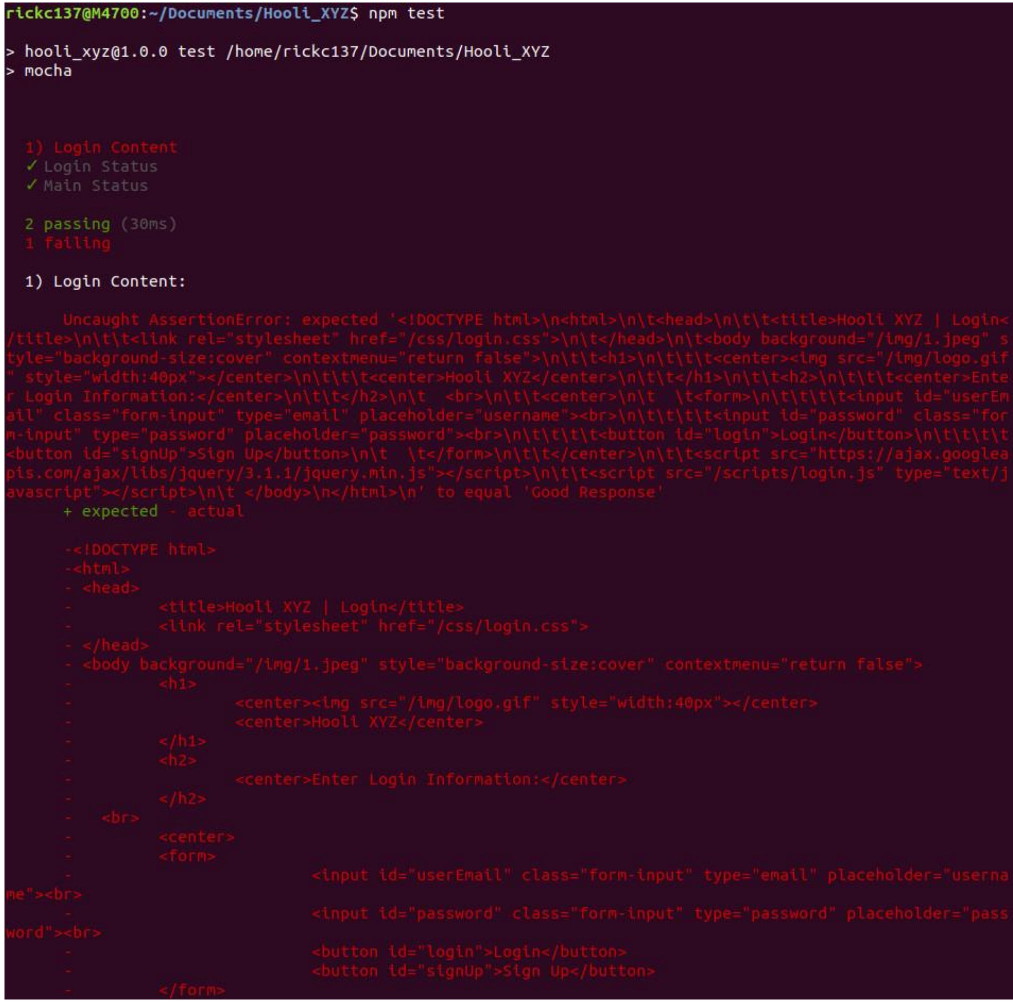
- Tests:
  - 1. Login Route Response
  - 2. Login Route Status
  - 3. Index Route Status



```
const expect = require("chai").expect
const request = require('request');

it("Login Content", function(done){
  request('http://localhost:8001/login', function(err, res, body){
    expect(body).to.equal("Good Response")
    done()
  })
})

it("Login Status", function(done){
  request('http://localhost:8001/login', function(err, res, body){
    expect(res.statusCode).to.equal(200)
    done()
  })
})

it("Main Status", function(done){
  request('http://localhost:8001', function(err, res, body){
    expect(res.statusCode).to.equal(200)
    done()
  })
})
```

Testing Results

1. Login Response: Failed
2. Login Status: Passed
3. Index Status: Passed



```
rickc137@M4700:~/Documents/Hooli_XYZ$ npm test

> hooli_xyz@1.0.0 test /home/rickc137/Documents/Hooli_XYZ
> mocha



  1) Login Content
  ✓ Login Status
  ✓ Main Status

  2 passing (30ms)
  1 failing

  1) Login Content:

      Uncaught AssertionError: expected '<!DOCTYPE html>\n<html>\n\t<head>\n\t\t<title>Hooli XYZ | Login<
/title>\n\t\t<link rel="stylesheet" href="/css/login.css">\n\t</head>\n\t<body background="/img/1.jpeg" s
tyle="background-size:cover" contextmenu="return false">\n\t\t<h1>\n\t\t\t<center><img src="/img/logo.gif
" style="width:40px"></center>\n\t\t\t<center>Hooli XYZ</center>\n\t\t</h1>\n\t\t<h2>\n\t\t\t<center>Ente
r Login Information:</center>\n\t\t</h2>\n\t  <br>\n\t\t<center>\n\t  \t<form>\n\t\t\t\t<input id="userEm
ail" class="form-input" type="email" placeholder="username"><br>\n\t\t\t\t<input id="password" class="for
m-input" type="password" placeholder="password"><br>\n\t\t\t\t<button id="login">Login</button>\n\t\t\t\t
<button id="signUp">Sign Up</button>\n\t  \t</form>\n\t\t</center>\n\t\t<script src="https://ajax.googlea
pis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>\n\t\t<script src="/scripts/login.js" type="text/j
avascript"></script>\n\t </body>\n</html>\n' to equal 'Good Response'
      + expected - actual

      -<!DOCTYPE html>
      -<html>
      - <head>
      -         <title>Hooli XYZ | Login</title>
      -         <link rel="stylesheet" href="/css/login.css">
      - </head>
      - <body background="/img/1.jpeg" style="background-size:cover" contextmenu="return false">
      -         <h1>
      -                 <center><img src="/img/logo.gif" style="width:40px"></center>
      -                 <center>Hooli XYZ</center>
      -         </h1>
      -         <h2>
      -                 <center>Enter Login Information:</center>
      -         </h2>
      -   <br>
      -         <center>
      -         <form>
      -                         <input id="userEmail" class="form-input" type="email" placeholder="userna
me"><br>
      -                         <input id="password" class="form-input" type="password" placeholder="pass
word"><br>
      -                         <button id="login">Login</button>
      -                         <button id="signUp">Sign Up</button>
      -         </form>
```

Deployment Environment

- Product tested on:
  - MacBook Pro 2015
    - Mac OS High Sierra version 10.13.5
    - Google Chrome Version 67.0.3396.99
  - Dell M4700
    - Ubuntu 17.01
    - Google Chrome (v8)

Development Environment

- HTML/Bootstrap
  - Front end
- Javascript/jQuery
  - Front end, Middle-layer
- Node.js
  - Server
- MySQL
  - Database

CSS { }

★★★★★

HTML

★★★★★

JavaScript

★★★★★

node JS

★★★★★

MySQL

★★★★★

Challenges Encountered

- Having the CSS not show up on our login/home page
- Integrating the database with the front end and the back end
- Learning to use Slack as a collaboration tool

Overcoming the Challenges

- Having the CSS not show up on our login/home page
  - Significant help from Abhijit Suresh
  - Add “app.use” statements to configure relative paths
    - Helps node js server understand where to look for corresponding folders

Overcoming the Challenges

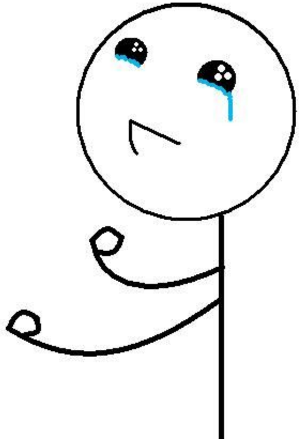
- Integrating the database with the front end and the back end
  - Still struggling with this
  - Learning how to run queries from MySQL through Node and have it display on an HTML page is a challenge
  - With more time, Hooli XYZ wouldn't have had any problems resolving this problem

Overcoming the Challenges

- Learning to use Slack as a collaboration tool
  - Discussed the importance of the tool in industry work environments
  - Agreed that this was the most effective tool for our project

How Challenges Affected Original Plan

- Had to cut Folder table
- Had no real use for Login table since we could not check authentication with the database
- Had no way of adding to the Whitelist table



Summary

- Our final product allows users to login and see all files
- If a user attempts to login but does not have an account associated with that email or that username does not exist, the user is redirected to the “create an account” page
- A download button and upload button are present

Demo