

CODE ASSIST

Capstone Final Summary

Project

A community-based resource library of code samples in multiple programming languages to easily search, find, and learn code.

Alison Hall

alison@alisonkhall.com

December 18, 2016

TABLE OF CONTENTS

MAIN OBJECTIVE	3
STAGES.....	4
DISCOVERY.....	4
<i>Our Solution</i>	4
<i>Opportunities</i>	4
<i>Challenges</i>	5
<i>Sustainability</i>	5
<i>Uniqueness of Idea</i>	5
<i>Tech SWOT Analysis</i>	6
<i>Market Competition Research</i>	7
<i>Further Planning and Documentation</i>	8
<i>Project Management</i>	8
EARLY EXPLORATION	9
<i>Design</i>	9
<i>Early Challenges Encountered</i>	9
PROTOTYPE	10
<i>All MVP Prototype Features:</i>	10
<i>Early Exploration Role in Development</i>	10
<i>Further Difficulties Encountered</i>	11
FINAL POLISHED PROJECT	12
<i>All Completed Features:</i>	12
<i>Group Member Roles</i>	12
<i>Future Steps</i>	13
<i>Areas of Learning</i>	14
APPENDIX.....	15
COMMUNICATION	15
<i>Slack Channel</i>	15
<i>Trello Board</i>	15
<i>Google Calendar</i>	16
<i>Bitbucket Repository</i>	16
PLANNING	17
<i>Ideation Process</i>	17
<i>Pitch/Value Proposition Canvas (VPC)</i>	17
<i>Feature Map</i>	18
<i>Feature Map (at project completion)</i>	18
<i>Logical Data Model</i>	19
<i>Logical Data Model (at project completion)</i>	19
DESIGN.....	20
<i>Wireframe Examples</i>	20
<i>Mockup Examples</i>	21

<i>Style Guide</i>	22
DEVELOPMENT.....	23
<i>React Routes Code</i>	23
<i>Code to Add an Example</i>	23
<i>Git Commits</i>	25

MAIN OBJECTIVE

The idea for Code Assist came from continually searching the internet for programming aid, and often finding that I was repeatedly searching for the same term. As a student, learning multiple programming languages at once proved to be confusing, especially considering the very minor differences between JavaScript and jQuery. There wasn't a condensed place on the internet where someone could keep code notes, research solutions, learn and be able to discuss with other users.

I currently save code syntaxes and examples in OneNote as I learn a programming language, but I find it isn't as helpful as it should be. This is because it is hard to search and isn't a comprehensive list. I've found it easier to just search Google instead of looking at my notes. This project is partially for me to be able to use the notes that I have created and see them in a way that I can easily search and use them.

Code Assist is designed to fill that gap by creating a space where users can search, filter and connect with other programmers at different levels of expertise. It should make researching faster and easier, the results would be up to date, and the community would be able to help resolve any additional questions through the comments section below code snippets.

Code Assist targets coders, whether they are a working professional, student or self-learner. While Google helps users narrow down hours of reading books into minutes of reading websites, Code Assist would take it one step further and allow users to find those solutions quicker through the search bar and tagging features.

Code Assist would help its target market by reducing the time spent on the web. Productivity should increase by using Code Assist since resources are all on one website and a quick search away. Code Assist's community would always be updating code snippets and discussing best practices and solutions, making the content always be current.

In addition, after my internship at SapientNitro and seeing them use React.js on many of their projects, I wanted to learn more about React and build my skills and confidence in using it. This would make me more desirable to potential employers, and also was originally to show SapientNitro that I could go out and learn something on my own.

STAGES

DISCOVERY

OUR SOLUTION

We want to create *Code Assist*, an online resource library of code curated by a community of working professionals, aspiring coders, students and passionate self-learners. It will provide coders with something they were lacking, a community that exists to aid of researching code solutions and be open and social so that it is adaptive and academic.

Users will be able to create code snippets (or upload screenshots) to share with the community in a wide-range of coding languages. They will also be able to filter and search code snippets by categories, subcategories, difficulty level or programming language, allowing them to see exactly what they want.

Code Assist will be easily accessible. Users can either view the website as a guest or be able to join the community with an effortless sign up process. As a new user signing up, they will have the option of doing so with their email or by using one of their social media accounts (such as Gmail, Facebook or Twitter). *Code Assist*'s user interface (UI) will be user-friendly, making the creating, editing and saving of snippets simple.

Aside from *Code Assist* acting as a resource library, it can also be used as a learning tool. Beginners, intermediate, and advanced coders will come to the site to expand their current knowledge. As a user they'll be able to look up information either through the category list or by typing in the search bar. The filter system can then be used to narrow the results even further by selecting programming languages they wish to see. If two or more languages are selected, they'll be displayed at the same time, allowing for easy comparison of the two to limit reloading pages or having to go back to the search results.

Once code is displayed, an intermediate or advanced user can then teach themselves the differences between the two programming languages and learn how to transition their previous knowledge and apply it. Through the comments section, users will also be able to challenge and debate merits of techniques used in the snippet and offer different solutions.

As an extra feature for our users, they'll also have a place to store private code snippets and save snippets other users have written for easy access.

OPPORTUNITIES

Code Assist will be a website accessible via internet browser (both through your desktop and handheld device). Currently, there is no plan of a dedicated mobile application. However, if the demand for a mobile app appears, it would then be created. Through this app, it is expected that users will have the same functionality of the website. Transitioning from using the website through your mobile app and a desktop browser should be seamless with no limits on usability.

Users creating code snippets will have the option of typing out code, embedding code from other websites (e.g. CodePen, JSFiddle), or including screenshots that support their explanation. All types of media can be used in

tandem to give readers a clear understanding of the material presented. This will also allow users to create online coding tutorials to aid other users in their projects or learning.

During launch, *Code Assist* will be free to use, however, there is opportunity for monetization. A simple solution for monetization can include non-invasive advertising, however a sign-up cost is not suggested, as it will likely deter users from signing up and this will hurt community growth.

CHALLENGES

With *Code Assist*, the team is interested in learning new technologies and utilizing them in order to deliver the best experience for its users. There is a shell currently built in PHP, and our development team is working on coding this into a new language, react.js. Outside of the platform change, the team will also be incorporating various plugins in order to accomplish the list of features looking to be added.

During later phases, incorporation of ranked users will be introduced; a system where the community can up and downvote other users based on their knowledge. As users gain a higher up vote rating, they'll gain access to edit code snippets created by other users; this will help keep information and code snippets from containing deprecated solutions. It also allows the community to control who edits what, and allows them to have a say in who does so.

Ranked users may be a challenge for *Code Assist* as it only will function with the help of the community. In order for this to take flight, a strong and active community needs to be present.

SUSTAINABILITY

Code Assist will have a future, being kept alive by its community aspect. It'll prove valuable to different companies, such as GitHub, where *Code Assist* can incorporate Git commands to accomplish pushes. Schools will also benefit from *Code Assist*, as they can send students there to learn and get additional help. *Code Assist* will also stay competitive by incorporating new languages as they are developed and become widespread.

UNIQUENESS OF IDEA

- Consolidated library with multiple programming languages
- User curated, with rankings to determine editing permission
- Have a ranking for users to show their helpfulness to community
- Allow users to save private snippets, favourite/save snippets
- Rank the learning level (beginner, intermediate, advanced)
- Filter content by language or category
- Allow for snippets to be added in different formats (text, image, embedded widget)

TECH SWOT ANALYSIS

STRENGTHS

- Allowing community to govern itself should help it grow
- Community involved in creating the content means less effort on our part
- 3 months to complete the project
- 2 group members to split up the tasks
- Knowledgeable instructors we can go to for help
- Already having an introduction to react.js
- Access to official React video tutorials
- Developer/Design duo

WEAKNESSES

- Learning curve needed for react.js
- Learning curve needed for the plugins and features such as filtering, searching, user ranking, comments, social media login
- Learning curve for integrating the website with services such as AWS
- Building a community to support the project
- Figuring out the best way to curate user content for best quality
- Group work and splitting up of tasks

OPPORTUNITIES

- No collective website currently available, allowing project to grow
- To bring in other communities that may not have been present beforehand and give them a place to interact
- Possibility to include ads on the site to bring in money
- Future possibilities of paid memberships

THREATS

- New idea emerging on the internet where users may not want to adapt
- Idea and technology may not line up as hoped
- If the time to learn the technology takes too long
- The lack of resources slowing the learning process of react.js, etc.
- The community not helping us enough to build the database and content

MARKET COMPETITION RESEARCH

COMPETITOR 1: GOOGLE

Currently, when researching a problem that you are not sure how to solve, or you are learning a new coding language, users turn to Google. After Googling their issue, they're presented with multiple hits from the internet that may or may not help them solve their problem. While the Google search doesn't take much time, the act of going through the results of the search and finding a solution that works for them takes time.

COMPETITOR 2: W3SCHOOLS

W3Schools is an online web developers website. It includes written tutorials and references on various web development languages. The website has grown since 1998 when it was originally created, and is viewed millions of times in a month. However, when researching on the website, it displays a static solution and only shows one example. Code on W3Schools can be deprecated, and lack the updated resolutions to problems which could be a result of lacking a community, so there is no dialogue on how to use code, updates or troubleshooting.

COMPETITOR 3: MOZILLA DEVELOPER NETWORK (MDN)

MDN is an open-source website information database. This means that the original source code is made available to the community for suggested changes/updates, and when accepted, will be made public to the internet. It includes information on HTML, CSS, and APIs. It also has tutorials available for its users, and include working examples using CodePen and/or JSFiddle. However, they are only text-based and there is no community aspect to the website either. Users cannot leave feedback and have discussions on these posts, although information posts are updated regularly and are likely to be up-to-date information.

COMPETITOR 4: STACK OVERFLOW

Stack Overflow is a question and answer website for programmers. It greatly relies on the community to be active and aid in resolving other user's issues. Users visit the website to post their programming issues, including their code snippets, and then wait for the community to respond with possible solutions. While the community aspect here is desireable for *Code Assist*, it lacks a way to ensure all code snippets are relevant with the most current update. For example, when searching their website, or clicking a link from a Google result, users need to check the date of the post to ensure it isn't an old posting. This is often overlooked and leaves users assuming the solution is not outdated.

COMPETITOR 5: READING BOOKS

One of the oldest, but still relevant, ways of learning, is by reading a book. Whether by going to the library, or purchasing a physical or online copy, people will spend hours reading content from a book that they could find online. The problem of books though is that information can be outdated, and it is also only coming from one point of view. There are different ways of coming about solutions, and books are likely to only show one solution. Books often only contain a couple of examples, and there is no aid in troubleshooting if problems arise.

FURTHER PLANNING AND DOCUMENTATION

See Appendix: [Ideation Process](#)

See Appendix: [Pitch/Value Proposition Canvas](#)

See Appendix: [Feature Map](#)

See Appendix: [Logical Data Model](#)

PROJECT MANAGEMENT

The project was managed by first creating a Slack channel, and adding multiple plugins for other tools to post updates to it. Slack was to be used as the main hub and contact channel between the group members (See Appendix: [Slack Channel](#)). The other tools used were:

- Google Drive for sharing documents
- A Trello Board planning the different phases, what needed to go into each phase, and then put it into Trello to visualize and assign tasks to members (See Appendix: [Trello Board](#))
- A Google Calendar for classes, work periods, and due dates (See Appendix: [Google Calendar](#))
- A Bitbucket repository for the project code to allow all members to have access to it (See Appendix: [Bitbucket Repository](#))

EARLY EXPLORATION

DESIGN

Between Liana and I, we sketched out our ideas of what *Code Assist* should look like and what it should include. Liana then took those sketches to create the wireframes, and later the mockups and style guide.

See the Appendix for [Wireframes](#), [Mockups](#), and [Style Guide](#).

In between creating the wireframes and the mockups, I created a quick prototype on Marvel (which can be found at <https://marvelapp.com/1404732>). This allowed me to test what the user experience and the user flow would be like. It allowed me to discover that some of the navigation between categories and examples was unclear, and it was hard to tell where in the site I was. It also made me realize that we needed to create a 'Create Snippet' page in the mockups.

EARLY CHALLENGES ENCOUNTERED

1. Deciding on how to section the code components:
 - a. Applying the proper methodology for React
 - b. Planning for future components
2. Importing the test data into the props (This ended up not being needed. Instead, I found that passing the states onto child components makes them props, so importing data directly into props should not be done.)
3. Importing the test data into states:
 - a. Importing data as objects (At the time I was not able to do this. I have since learned how, but not gone back and changed the code. I temporarily restructured the test data to be imported as arrays instead.)
 - b. Importing data as arrays
4. Having the components load only after the data has finished loading (I solved this by using state variables to check if the data is loaded)
5. Reload the page when the state and data changes (This was solved by using `setState(...)` instead of the `state = "..."` statement. It made a difference because the `setState` function automatically will reload the component when the data changes.)
6. Passing the data between components (Solved by having a top component (Page.js) which loads the data and then passes the data to the subcomponents called by it)
7. Get jQuery UI working for the accordion and dropdown combobox (This was not completed. jQuery UI does not work well with React due to it being applied after the DOM is finished loading, but React is never truly finished, so jQuery UI isn't run properly. I later manually build the components' styling instead.)
8. Get .scss files compiled into the .css built file (I tried multiple solutions found online, but it was eventually solved by ignoring the suggested ExtractTextPlugin in the webpack file)
9. Add bootstrap through npm (Not done. This would have taken more time to try and load, so I just added the cdn to the html file.)
10. Add highlight.js through npm (Not done. This would have taken a lot more time to try and load, so I manually called the highlight function, but the styling wasn't how I wanted, so had to start again with Prism.js. Again, couldn't add through npm, so it was loaded manually.)

PROTOTYPE

At the point of the project where the prototype is finished, Code Assist has categories with code examples and syntaxes which can be viewed. It also has the functionality to be able to add categories and examples/syntaxes. Users can also select or deselect languages to filter the results shown.

ALL MVP PROTOTYPE FEATURES:

- setup React project
- filter examples based on language and category selection
- select and deselect languages
- list all the categories
- have a languages dropdown box
- view examples/syntaxes
- syntax highlighting of code based on language
- view comments
- max. 3 languages selected
- create a new syntax/example snippet
- data stored in cloud on Firebase
- add examples/syntaxes
- add categories

See Appendix: [Feature Map \(at project completion\)](#) to see the project's status at the MVP point.

EARLY EXPLORATION ROLE IN DEVELOPMENT

I used the Logical Data Model to create some test data to work with and load into the project at the start. I also used the Feature Map to help decide which features to concentrate on and give priority in development. The wireframes and mockups were used to determine which components went on which view, and the mockups also helped with coding the styles.

By planning ahead and starting early on sectioning the code into components, it made it easier to make changes to only parts of the project, as well as having the styling only affecting that one component.

To be able to add code examples, I needed to change over from loading the data from the test-data.json file and instead connecting to Firebase. This was because the test-data.json file wasn't supposed to be edited, and therefore wouldn't save any of the changes made to the data when adding an example. I ended up needing to rewrite the previous code I had done to load in the data.

FURTHER DIFFICULTIES ENCOUNTERED

1. Recursively adding data to the categories list with the proper hierarchy
2. Create the routing
 - a. determining which file the route data goes into
 - b. getting the pages to load different views (This was initially solved by having 'view' files for each view option. This meant a lot of repeated code and loading the test data multiple times.)
 - c. getting the pages to only reload the inner content instead of the header, sidebar, etc. (This has been partially done, but to fix this properly I need to use the Alt-Flux method of loading, using, and setting data.)
 - d. get the selected languages to be transferred when the page changes (This is partially done. It currently is saving the selected languages in Firebase, but it means that if multiple people are using the site at the same time, they would be changing each other's selected languages too. To properly solve this, I need to load the selected languages into local storage instead of Firebase.)
 - e. add react links where needed to change the React views

See Appendix: [React Routes Code](#) for the final Route code

3. Use Firebase
 - a. initial trouble connecting properly using the re-base npm plugin
 - b. had to recode sections to take into account the fact that Firebase doesn't keep any empty object as references
 - c. had trouble adding a new example with Firebase due to its non-existent empty objects

See Appendix: [Code to Add an Example](#) for the conditions to handle Firebase empty objects

FINAL POLISHED PROJECT

At the end of this project, Code Assist is a fully working database for code snippets. The main functionality of having categories with code examples and syntaxes in them are easily viewed. The examples/syntaxes shown are based on the category selected, as well as the programming language(s). Additional functionality includes the ability to add and delete categories, as well as adding and deleting examples/syntaxes.

ALL COMPLETED FEATURES:

- setup React project
- filter examples based on language and category selection
- select and deselect languages
- list all the categories
- have a languages dropdown box
- view examples/syntaxes
- syntax highlighting of code based on language
- view comments
- max. 3 languages selected
- create a new syntax/example snippet
- data stored in cloud on Firebase
- add examples/syntaxes
- edit examples/syntaxes
- add categories
- edit categories
- delete examples/syntaxes
- delete categories
- have accordions for the sidebar categories

GROUP MEMBER ROLES

LIANA:

Liana was designated the lead designer. We both worked together in the early parts of the project when fleshing out what we wanted included in the project (pitch idea, ideation process, tech SWOT, and the VPC). I tried to leave most of the design to Liana to split up the roles, but it ended up being more of a collaboration for what should be included in the design. Liana did fully make the wireframes and mockups herself. She had done a couple of hours of work in one class on coding the styling into the project before she stopped coming to classes and responding to emails.

ALISON (ME):

I was the lead developer and ended up being the overall project manager for this project. I came up with the idea for the project and worked on the documentation of the planning for the project (pitch idea, ideation process, tech SWOT, VPC, epics/user stories and features, a feature map, and a logical data model). I created the basis for keeping in contact with each other, sharing files, and a schedule (Google Drive folder, Slack channel, Google Calendar, Trello board, and a BitBucket repository with usage instructions). I had some input into the designs and created a prototype on Marvel based off of the wireframes. Meanwhile, after we had decided on our idea, I got started immediately on setting up the React project, integrating it with NPM and Webpack, and starting the coding and development. For the rest of the term, I did the rest of the development of the project.

FUTURE STEPS

- Add further features, add users and community aspect
- Get the project online
- Improve design and UX
- Use Alt-Flux to better pass data between components
- Use local storage for data such as which user is logged in, as well as which languages they have selected
- Save some of the React data as objects instead of arrays for faster searching and better memory usage
- Consider monetization
- Develop a strong user base
- Incorporate AWS for scalability

FUTURE FEATURES:

- Searching website
- Text copy content on home page, about page, etc.
- Allow embedded examples
- Allow images as examples
- Add comments and community (possibly using Disqus)
- Add how-to examples
- User registration and login
- Favourite/save examples
- Upvote/downvote examples and comments
- Use users' rank to determine editing permissions
- Have the language selection be autocompleted
- Have the search bar have autocomplete
- Prioritize search results
- Have lazy loading for when there are too many examples or search results

See Appendix: [Feature Map \(at project completion\)](#) to see its current status.

AREAS OF LEARNING

I've learned a lot during this project, and put a lot of effort into it. The skills I've learned include soft skills, design skills, and technical skills.

SOFT SKILLS:

- Attempt to compartmentalize user roles so that there isn't as much dependency on others
- Only assign tasks once the previous part is finished so that there isn't as much waiting for others to finish
- Not to stress and micromanage other partners' parts
- Confidence in my coding ability and level in preparation of graduating

DESIGN SKILLS:

- Emphasis on how simple is usually better than over-complicating something
- How the navigation and user flow needs to be really clear to users

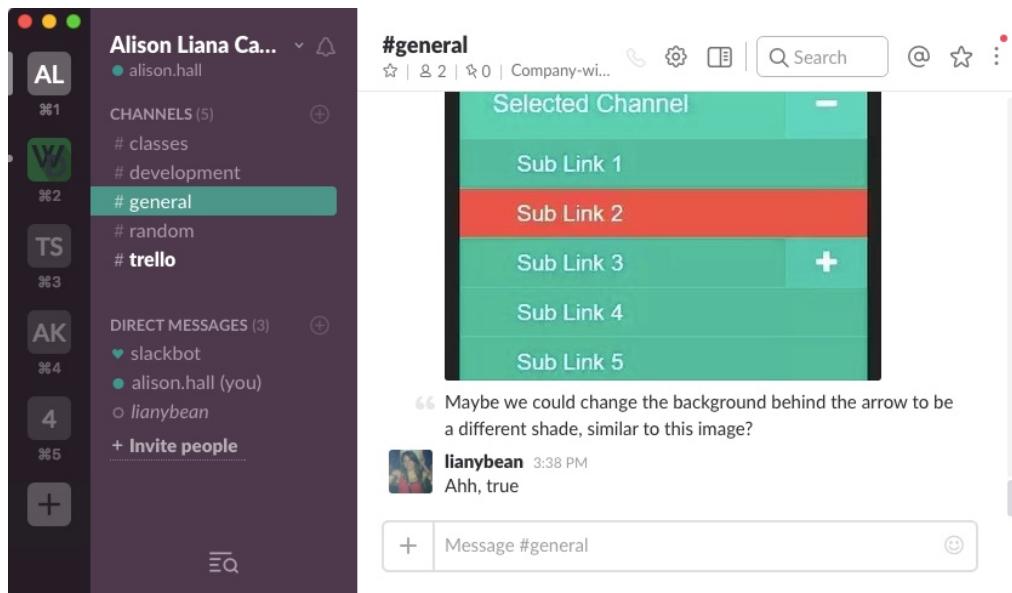
TECHNICAL SKILLS:

- Setting up a React.js project
- Incorporating React, Firebase, Webpack, ES6, and NPM in one project
- Using React.js
- How to route components in React
- How to have React only run parts of the code after the data has been passed to the component
- And much more miscellaneous coding skills

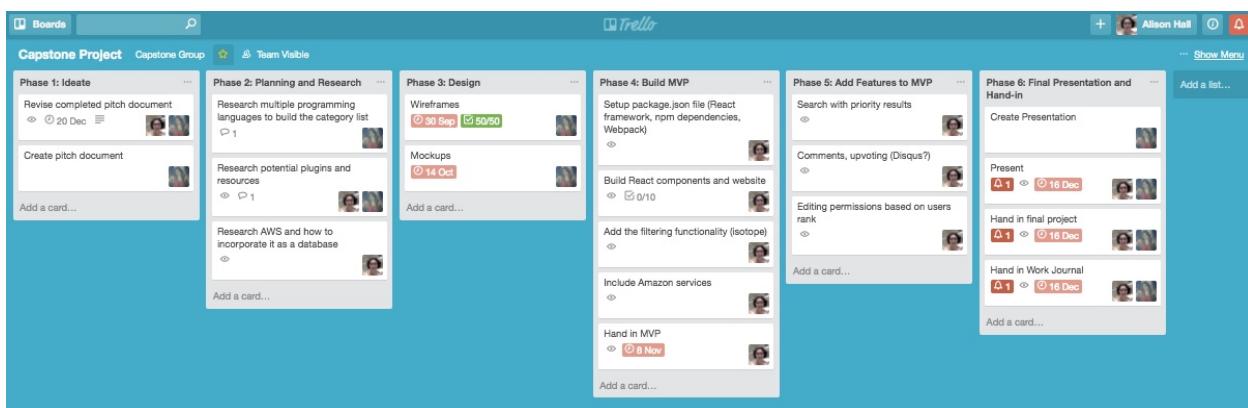
APPENDIX

COMMUNICATION

SLACK CHANNEL



TRELLO BOARD



GOOGLE CALENDAR

Google Calendar

Search Calendar

Calendar Today Thursday, Sep 1, 2016 Day Week Month 4 Days Agenda More Settings

CREATE

Expand All **Collapse All**

September 2016

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Tue Sep 6 12:35pm – 3:15pm **Capstone - Course Intro / Structure of a Pitch / Pitch to Class** - L209 - Humber College, North Campus

Wed Sep 7 11:40am – 2:20pm **Capstone - Pitch to Class / Lesson: Idea Math/Brainstorming** - L209 - Humber College, North Campus

Fri Sep 9 10:45am – 1:25pm **Capstone - Pitch to Class** - J132 - Humber College, North Campus

Tue Sep 13 12:35pm – 3:15pm **Capstone - Revising Idea – Instructor Support** - L209 - Humber College, North Campus

Wed Sep 14 11:40am – 2:20pm **Capstone - Revising Idea – Instructor Support** - L209 - Humber College, North Campus

Fri Sep 16 10:45am – 1:25pm **Capstone - Pitch Document Due** - J132 - Humber College, North Campus

Tue Sep 20 9:50am – 12:30pm **Capstone - Pitch Document Feedback** - L209 - Humber College, North Campus

Wed Sep 21 11:40am – 2:20pm **Capstone - Pitch Document Feedback / Lesson: Technology Documentation / How to choose technology** - L209 - Humber College, North Campus

Fri Sep 23 10:45am – 1:25pm **Capstone - Pitch Document Feedback /** - J132 - Humber College, North Campus

Tue Sep 27 12:35pm – 3:15pm **Capstone - In-class Instructor support** - L209 - Humber College, North Campus

Wed Sep 28 11:40am – 2:20pm **Capstone - In-class Instructor support** - L209 - Humber College, North Campus

Fri Sep 30 10:45am – 1:25pm **Capstone - Wireframe Due (10%) In-class Instructor support** - J132 - Humber College, North Campus

Tue Oct 4 12:35pm – 3:15pm **Capstone - Independent Study (no class)** - L209 - Humber College, North Campus

Wed Oct 5 11:40am – 2:20pm **Capstone - Independent Study (no class)** - L209 - Humber College, North Campus

Tue Oct 11 12:35pm – 3:15pm **Capstone - In-class Instructor support** - L209 - Humber College, North Campus

Wed Oct 12 11:40am – 2:20pm **Capstone - In-class Instructor support** - L209 - Humber College, North Campus

BITBUCKET REPOSITORY

Bitbucket Teams Projects Repositories Snippets Find a repository... ?

Alison Hall / WEBD 302 Capstone / Capstone Website Overview git@bitbucket.org:alison-hall/capstone

Last updated 2016-12-14 Language — Access level Admin (revoke)

13	0
Branches	Tags
0	2
Forks	Watchers

Edit README

Capstone Website

Getting Started

- 1) Open Terminal on your computer (or Terminal equivalent).
- 2) Navigate to the directory where you want this repository code to be on your local computer using the `cd foldername` command.
- 3) Enter the following 4 lines in Terminal:

```
git clone https://alisonhall@bitbucket.org/alison-hall/capstone-website.git
cd capstone-website
npm install
npm start
```

Make sure that you have the remote set up to connect your local files with the server repository. Do this by typing `git remote -v`. If a remote doesn't show up, you will need to use the following command (where 'origin' is the new

Recent activity

- 1 commit Pushed to alison-hall/capstone-website | 428c623 Improve styling of header and edi... Alison Hall · 4 days ago
- 1 commit Pushed to alison-hall/capstone-website | 5c83432 Redo sidebar categories and acc... Alison Hall · 4 days ago
- 1 commit Pushed to alison-hall/capstone-website | da6f5ad Change the ExampleCards and ... Alison Hall · 2016-12-09
- 1 commit Pushed to alison-hall/capstone-website

PLANNING

IDEATION PROCESS

Ideation process

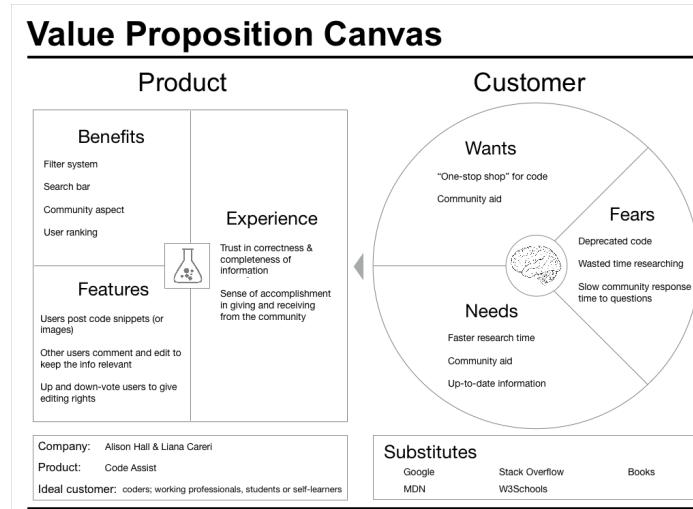
Symbol	Definition	Applied
X	Idea	Code Compare - organize snippets of code in different programming languages
X ^d	Take current idea and make it even crazier/better/push it to the next level	Add a community aspect Incorporate the ranking of users and their permissions Autocomplete tags and more when creating a code snippet
X + Y	Take idea and combine with another separate idea	Code database + community involvement
X !	Find all the problems the solution fixes	Easily look up resources Library of code Learning a new language
X	Treat the idea as the problem and find all the solutions	Idea/Problem: Need to easily find code examples: Solutions: Search Google, read a book, browse MDN
X++	Add an adjective to your idea	Faster (to find needed code), Open (community aspect), Social (community aspect), Adaptive (editable by community), Academic (learning code), Applied (shows examples of code), Educational (learning code), Self-educated (learning code), Well-rounded (multiple languages)
X	Do the exact opposite of the idea	Multiple plain lists of code for each language

- What are you trying to do? Articulate your objectives using absolutely no jargon.
 - Create a community-based resource library of code samples in multiple programming languages to easily search, find, and learn code.
- How is it done today, and what are the limits of current practice?
 - Search Google: Too many results, no curation
 - Visit multiple resource library websites: Takes a long time, unknown credibility, unknown results

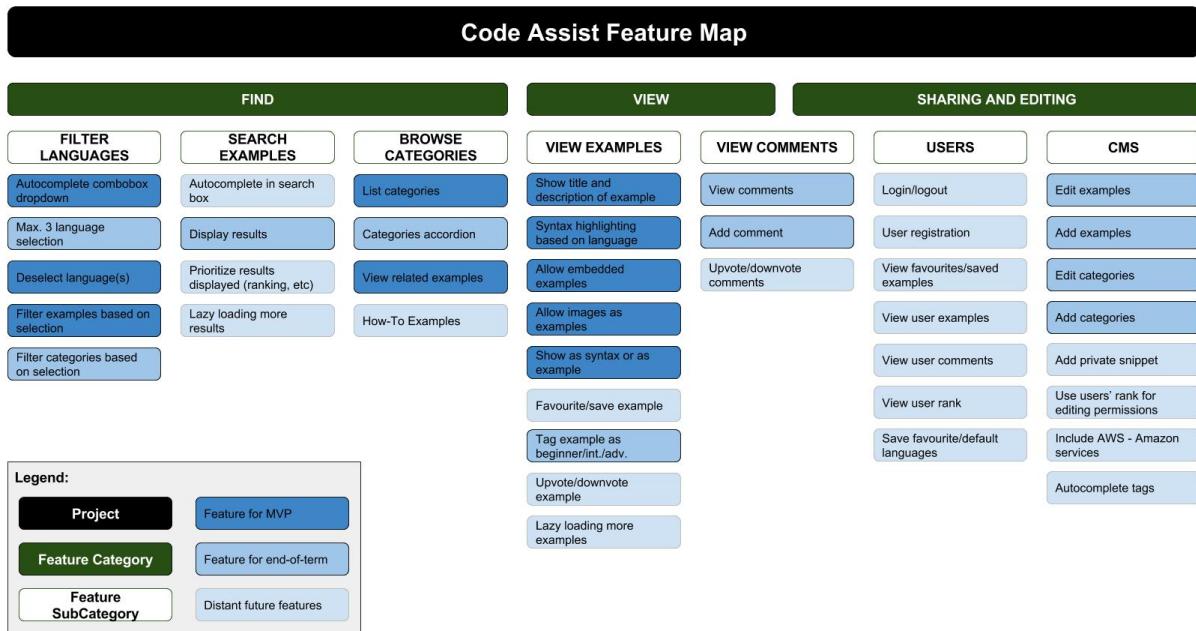
- Compare examples online to find best practices: Takes a long time, differences of opinions, deprecated information
◦ Search engines: Cost/money, time to watch them, inappropriate to personal learning style
 - Read a book on the language: Possibly deprecated information, time to find appropriate book, time to read book, only one opinion/view point
 - Take a course: Cost/money, time, relevancy
 - Consult with other people: Their knowledge incorrect, finding person to ask, wasting their time, only one opinion/view point
- What's the goal? What does success look like? What will it be successful?
 - Consolidated library with multiple programming languages
 - User curated, with rankings to determine editing permission
 - Have a ranking for users to show their helpfulness to community
 - Allow users to save private snippets, favourite/save snippets
 - Rank the learning level (beginner, intermediate, advanced)
 - Filter content by language or category
 - Allow for snippets to be added in different formats (text, image, embedded widget)
 - Who's going to use it?
 - Me, Students
 - Coding community
 - Working professionals
 - Aspiring coders
 - If you're successful, what difference will it make?
 - Faster learning curve
 - Increase in productivity
 - Relevant/current information
 - What are the risks and the payoffs?
 - Risks:
 - React
 - Filtering (isotope)
 - Searching
 - User ranking
 - Comments
 - Social media login
 - Payoffs:
 - Learn a new framework
 - Learn plugins for standard website components
 - Learn how to build a community
 - Something for the portfolio
 - Something to showcase to potential employers
 - How much will it cost?
 - \$0
 - Time

- Future growth: marketing, employees, lawyers, server costs, domain costs
- How long will it take?
 - Time:
 - A month for the MVP
 - A month for adding main features
 - Two months for adding extra features
 - Infinite time for upkeep
 - Phases:
 -
- What are the midterm and final "exams" to check for success?
 - Midterm:
 - MVP usability
 - Final:
 - Longevity
 - Having a community
 - Expanding supported languages

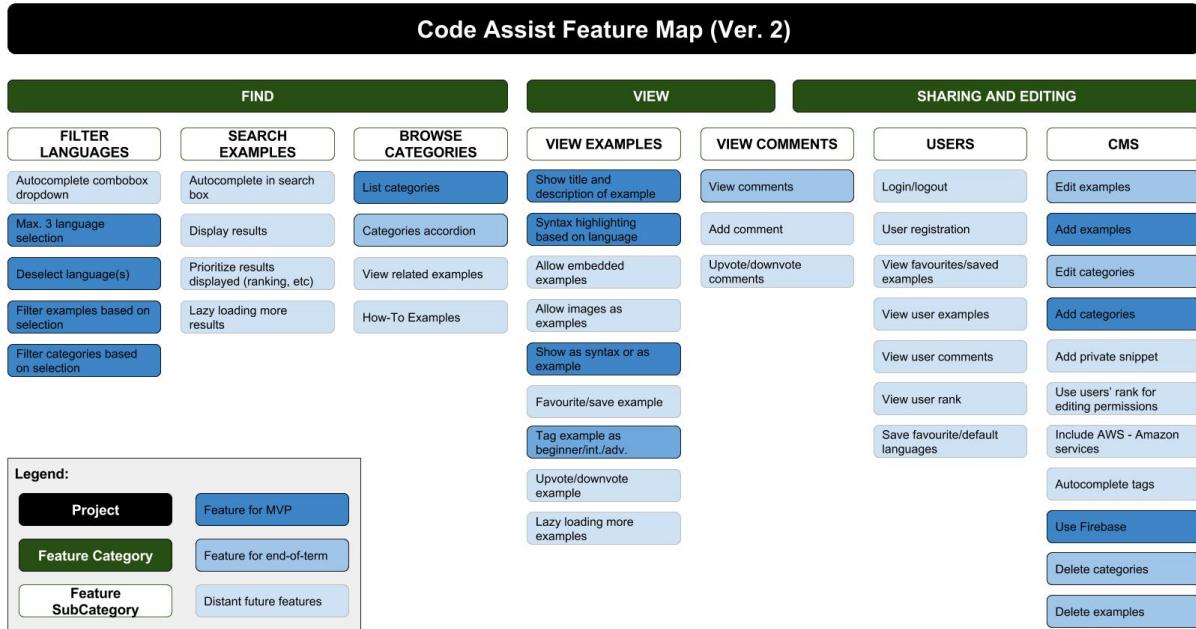
PITCH/VALUE PROPOSITION CANVAS (VPC)



FEATURE MAP

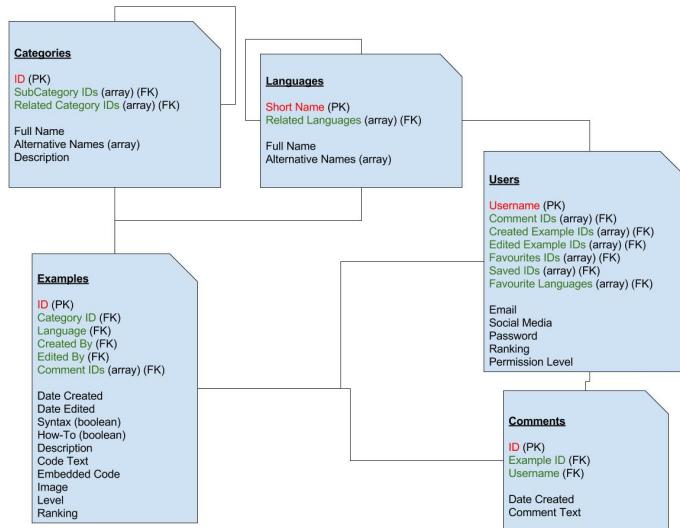


FEATURE MAP (AT PROJECT COMPLETION)



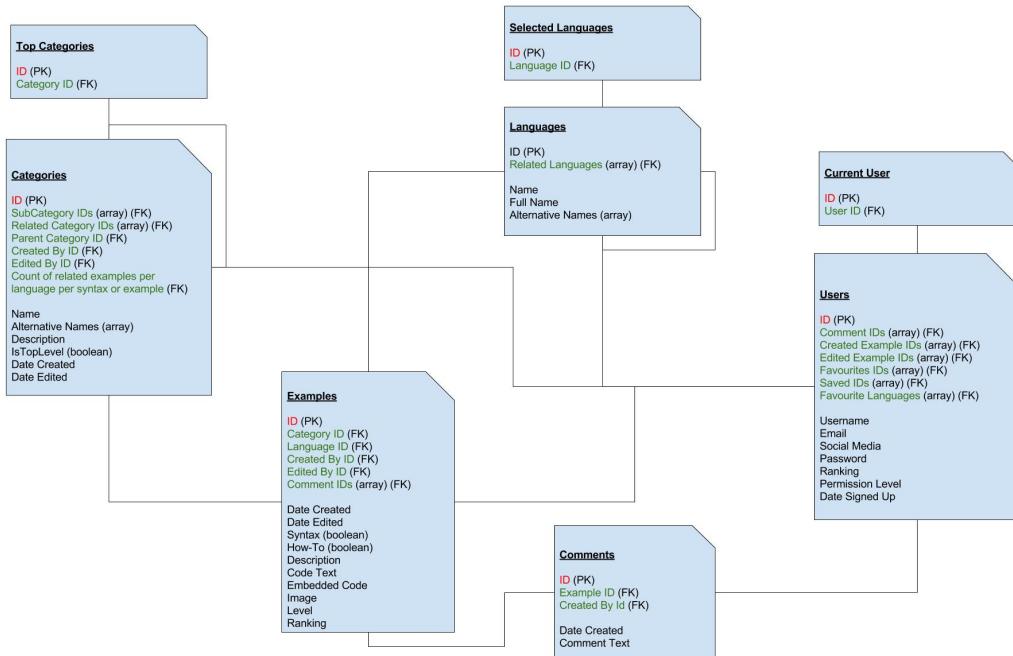
LOGICAL DATA MODEL

Logical Data Model



LOGICAL DATA MODEL (AT PROJECT COMPLETION)

Logical Data Model Ver.2



DESIGN

WIREFRAME EXAMPLES

Get Started!

Default message about searching, using filters, how to navigate.

Try it now!

Type your search here

Function Argument

Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

Examples

Rank #	Last edit:	Username	Language
Date 00,0000 Level: Intermediate			
<p>Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat...</p> <p>Syntax:</p> <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"></pre> <p>#### of comments</p>			

Rank #	Last edit:	Username	Language
Date 00,0000 Level: Intermediate			
<p>Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat...</p> <p>Syntax:</p> <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"></pre> <p>#### of comments</p>			

Move to Next Category

MOCKUP EXAMPLES

Get Started!

Default message about searching, using filters, how to navigate.

Try it now!

Type your search here

[legal](#) | [terms & conditions](#) | [about us](#)

copyright 2016

Function Argument

Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

[Expand](#)

Date 0,0000
Level Beginner

Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat...

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

[Expand](#)

Examples

Rank # Last edit: [Username](#) Language Level: Beginner

Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat...

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

of comments [Expand](#)

Rank # Last edit: [Username](#) Language

Date 0,0000
Level Beginner

Description: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat...

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

of comments [Expand](#)

[Move to Next Category](#)

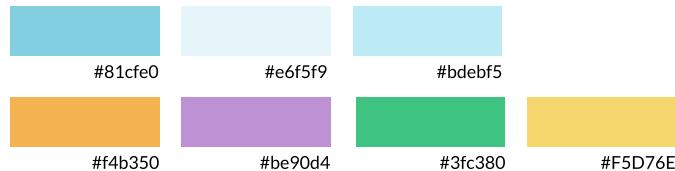
[legal](#) | [terms & conditions](#) | [about us](#)

copyright 2016

STYLE GUIDE

CODE ASSIST STYLE GUIDE

01 COLORS



02 TYPOGRAPHY

HEADING 1 - UNICA ONE - 2.25em (LOGO ONLY)

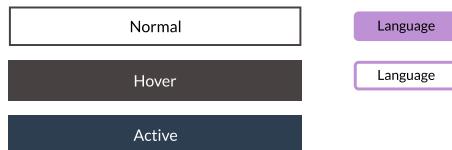
Heading 2 - Lato - Bold - 1.875em

Heading 3 - Lato - Bold - 1.125em

Body text - Lato - Regular - 1.063em

 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

03 BUTTONS



04 ICONOGRAPHY



05 INPUTS



DEVELOPMENT

REACT ROUTES CODE

```

24 const routes = (
25   <Router history={browserHistory}>
26     <Route path="/" component={HomeView} />
27     <Route path="/home" component={HomeView} />
28     <Route path="/about" component={AboutView} />
29     <Route path="/login" component={LoginView} />
30     <Route path="/create/category" component={CreateCategoryView} />
31     <Route path="/create/snippet" component={CreateSnippetView} />
32     <Route path="/user/:userId" component={UserSettingsView} />
33     <Route path="/search/:searchTerm" component={SearchResultsView} />
34     <Route path="/category/:categoryId" component={CategoryView} />
35     <Route path="/example/:exampleId" component={OpenedExampleCardView} />
36     <Route path="/" component={Error404} />
37   </Router>
38 );
39
40 ReactDOM.render(
41   routes,
42   document.getElementById('react-page')
43 );

```

CODE TO ADD AN EXAMPLE

```

616 addExample(object, index, category, language, isSyntax) {
617   console.log("Add example to " + category);
618   console.log(object, index, category, language, isSyntax);
619   var allExamples = this.state.allExamples;
620   var allCategories = this.state.allCategories;
621
622   var type = (isSyntax) ? 'syntaxes' : 'examples';
623   var numTypes = 0;
624
625
626   if(allCategories[category].count) {
627     if(allCategories[category].count[language]) {
628       if(allCategories[category].count[language][type]) {
629         numTypes = allCategories[category].count[language][type].length + 1;
630         allCategories[category].count[language][type][numTypes] =
631           index;
632       } else {
633         var syntaxes = (allCategories[category]["count"][language][
634           "syntaxes"]) ? allCategories[category]["count"][language][
635             "syntaxes"] : [];
636         var examples = (allCategories[category]["count"][language][
637           "examples"]) ? allCategories[category]["count"][language][
638             "examples"] : [];
639
640         if(type == 'syntaxes') {
641           allCategories[category]["count"][language] = {
642             // ["syntaxes": syntaxes,
643             // ["examples": examples,
644             // [type]: {
645               // [0]: index
646             // },
647           }
648         } else if (type == 'examples') {
649           allCategories[category]["count"][language] = {
650             // ["examples": examples,
651             // ["syntaxes": syntaxes,
652             // [type]: {
653               // [0]: index
654             // },
655           }
656         } else {
657           console.log("ERROR: unknown type in allCategories[
658             category][count][language]");
659         }
660       }
661     }
662   }
663 }

```

```

656     } else {
657       var totalNumLanguages = this.state.allLanguages.length;
658
659       var countObjects = [];
660
661       for (var i = 0; i < this.state.allLanguages.length; i++) {
662         var languageObject = (allCategories[category]["count"][i]) ?
663           allCategories[category]["count"][i] : {};
664
665         if (language == i) {
666           countObjects.push({
667             [type]: {
668               [0]: index
669             }
670           });
671         } else {
672           countObjects.push(languageObject);
673         }
674       }
675
676       allCategories[category]["count"] = { ...countObjects };
677     }
678   } else {
679     var id = allCategories[category].id;
680     var key = (allCategories[category].key) ? allCategories[category].key
681       : '';
682     var subCategoryIDs = (allCategories[category].subCategoryIDs) ?
683       allCategories[category].subCategoryIDs : [];
684     var relatedCategories = (allCategories[category].relatedCategories) ?
685       allCategories[category].relatedCategories : [];
686     var isTopLevel = (allCategories[category].isTopLevel) ? allCategories
687       [category].isTopLevel : null;
688     var name = (allCategories[category].name) ? allCategories[category].
689       name : '';
690     var alternativeNames = (allCategories[category].alternativeNames) ?
691       allCategories[category].alternativeNames : [];
692     var description = (allCategories[category].description) ?
693       allCategories[category].description : '';
694     var dateCreated = (allCategories[category].dateCreated) ?
695       allCategories[category].dateCreated : '';
696     var dateEdited = (allCategories[category].dateEdited) ? allCategories
697       [category].dateEdited : '';
698     var editedBy = (allCategories[category].editedBy) ? allCategories[
699       category].editedBy : '';
700     var createdBy = (allCategories[category].createdBy) ? allCategories[
701       category].createdBy : '';
702     var parentCategory = (allCategories[category].parentCategory) ?
703       allCategories[category].parentCategory : '';
704
705     allCategories[category] = {
706       ["count": {
707         [language]: {
708           [type]: {
709             [0]: index
710           }
711         },
712         id,
713         key,
714         subCategoryIDs,
715         relatedCategories,
716         isTopLevel,
717         name,
718         alternativeNames,
719         description,
720         dateCreated,
721         dateEdited,
722         editedBy,
723         createdBy,
724         parentCategory
725       }
726     }
727
728     allExamples[index] = object;
729
730     this.setState({ allCategories: allCategories, allExamples: allExamples })
731   ;
732 }

```

GIT COMMITS

Commit	Author	Date
[origin/master] [master] Minor styling improvements.	f77ba44 Alison Hall <alison... Dec 18, 2016, 1...	
Improve styling of header and edit/delete buttons.	428c623 Alison Hall <alison... Dec 14, 2016, 2...	
Redo sidebar categories and accordions.	5c83432 Alison Hall <alison... Dec 14, 2016, 2...	
Change the ExampleCards and OpenExampleCard components to include the Sy...	da6f5ad Alison Hall <alison... Dec 9, 2016, 1:1...	
Finish adding functionality to edit an example. Minor fix of 'isActive' check.	9d87cbe Alison Hall <alison... Dec 7, 2016, 12...	
Add functionality to delete and restore an example. Starting to add functionality...	4eff060 Alison Hall <alison... Dec 7, 2016, 11...	
Added ability to delete and edit categories by adding the following functions and...	6bf0c84 Alison Hall <alison... Dec 2, 2016, 4:0...	
Create the CategoryInfo component. Have it also handle the form to edit the category.	846ec9f Alison Hall <alison... Dec 2, 2016, 4:0...	
Start creating the CategoryInfo component, and set up the Examples component...	f87fa06 Alison Hall <alison... Dec 2, 2016, 4:0...	
Minor changes to test data, as well as backups of Firebase database	498a4e6 Alison Hall <alison... Dec 2, 2016, 4:0...	
More minor comment fixes	720c77f Alison Hall <alison... Dec 2, 2016, 4:0...	
Fix the number of comments dependencies.	f88f7dc Alison Hall <alison... Dec 2, 2016, 4:0...	
Minor fixes on CreateCategory and CreateSnippet components.	7a00f40 Alison Hall <alison... Dec 2, 2016, 4:0...	
Temporarily just plainly list all of the categories in the sidebar.	f5e949a Alison Hall <alison... Dec 2, 2016, 4:0...	
[github-prod] Change reference locations.	e1b13bc Alison Hall <alison... Dec 2, 2016, 2:1...	
First commit	730c88a Alison Hall <alison... Dec 2, 2016, 2:1...	
[prod] Add prod folder, copy files to it.	247047f Alison Hall <alison... Dec 2, 2016, 2:0...	
[origin/feature-add-snippet] [feature-add-snippet] Add the timestamp to the Cr...	3dba840 Alison Hall <alison... Nov 16, 2016, 5...	
Merge branch 'feature-add-category'	19c384f Alison Hall <alison... Nov 16, 2016, 5...	
[origin/feature-add-category] [feature-add-category] Install moment.js using np...	33eb076 Alison Hall <alison... Nov 16, 2016, 5...	
Change test-data to have the dates be a proper timestamp. Have the categories...	93282bd Alison Hall <alison... Nov 16, 2016, 5...	
Clean up the unneeded console.log's, and fix the way an example recreates the s...	4b5ed35 Alison Hall <alison... Nov 16, 2016, 4...	
Edit the CreateSnippet component to update the component state when an input...	7b81be1 Alison Hall <alison... Nov 16, 2016, 4...	
Add the functionality to the Page component to add the category to the state data.	20eb2bd Alison Hall <alison... Nov 16, 2016, 4...	
Create the CreateCategory component.	57e81ee Alison Hall <alison... Nov 16, 2016, 4...	
Setup for the CreateCategory component. Change the router for the path to crea...	7a7171c Alison Hall <alison... Nov 16, 2016, 4...	
Create the add category button and style it.	658c549 Alison Hall <alison... Nov 16, 2016, 4...	
[origin/styling] [styling] Styling changes for language selected, the create sni...	5c88992 Alison Hall <alison... Nov 16, 2016, 1...	
[github-version] Merge branch 'master' of github.com:alisonhall/codeassist	3bf5eb0 Alison Hall <alison... Nov 15, 2016, 1...	
Delete CNAME	546b956 Alison Hall <n010... Nov 15, 2016, 1...	
Create CNAME	9391c39 Alison Hall <n010... Nov 15, 2016, 1...	
Delete CNAME	63c06d3 Alison Hall <n010... Nov 15, 2016, 1...	
Create CNAME	cca2f40 Alison Hall <n010... Nov 14, 2016, 1...	
Delete CNAME	0877f63 Alison Hall <n010... Nov 10, 2016, 6...	
Create CNAME	70dcf79 Alison Hall <n010... Nov 9, 2016, 5:1...	
Merge branch 'styling'	cabb653 Alison Hall <alison... Nov 15, 2016, 1...	
Include all files locally that are downloaded from a CDN, to be used if without an...	d82f2db Alison Hall <alison... Nov 15, 2016, 1...	
Change styling and data shown of the open cards to be slightly different for the...	a1b0e04 Alison Hall <alison... Nov 15, 2016, 1...	
[origin/rebuild-firebase-2] [rebuild-firebase-2] Get the OpenExampleCard and...	545454b Alison Hall <alison... Nov 15, 2016, 1...	
Change the way the page connects to Firebase using re-base, by first downloadi...	9e5f5e3 Alison Hall <alison... Nov 15, 2016, 1...	
Get the Create Snippet code working with Firebase again	abb8047 Alison Hall <alison... Nov 8, 2016, 4:2...	
Make changes to code to account for Firebase not having placeholders for empt...	95d60a5 Alison Hall <alison... Nov 8, 2016, 4:1...	
Connect states with Firebase data	39a8111 Alison Hall <alison... Nov 8, 2016, 4:1...	
Create functionality to add a new example/snippet to the examples array. Use th...	5ef9650 Alison Hall <alison... Nov 3, 2016, 12...	
Temporarily change the EditMenu button to go straight to the CreateSnippet view.	d831f81 Alison Hall <alison... Nov 3, 2016, 12...	
[origin/liana] Fix problem with indexing of accordion categories	8629465 Alison Hall <alison... Nov 2, 2016, 2:2...	
Create additional test data for 'Functions'. Improve styling.	418ead4 Alison Hall <alison... Nov 2, 2016, 2:2...	
Fix the calling of examples vs. syntaxes. Create the EditMenu component.	2674058 Alison Hall <alison... Nov 2, 2016, 2:2...	
[origin/feature-languages] [feature-languages] Fix reference to which languag...	e36bb5f Alison Hall <alison... Nov 2, 2016, 9:0...	
[origin/feature-example-cards] [feature-example-cards] Split the NoContent co...	8d0882b Alison Hall <alison... Nov 2, 2016, 8:4...	
Add delete button and functionality to selected languages.	0ba2c58 Alison Hall <alison... Nov 1, 2016, 4:2...	
[origin/feature-accordion] [feature-accordion] Add functionality and styling to...	e549d69 Alison Hall <alison... Nov 1, 2016, 3:5...	
Fix language bar width	9ae680c Alison Hall <alison... Nov 1, 2016, 3:5...	
Merge branch 'master' into feature-example-cards	71d21c5 Alison Hall <alison... Nov 1, 2016, 3:0...	
Fix: refer to CategoriesSidebar instead of Categories	d3085fd Alison Hall <alison... Nov 1, 2016, 12...	
Only have examples/syntaxes of the selected language(s) shown. Show these in...	fac36df Alison Hall <alison... Nov 1, 2016, 3:0...	
Change name of Categories component to CategoriesSidebar	f08fac9 Alison Hall <alison... Oct 27, 2016, 9:...	
Change relation to files in index.html to always refer to base folder	231dc16 Alison Hall <alison... Oct 26, 2016, 6:...	

