

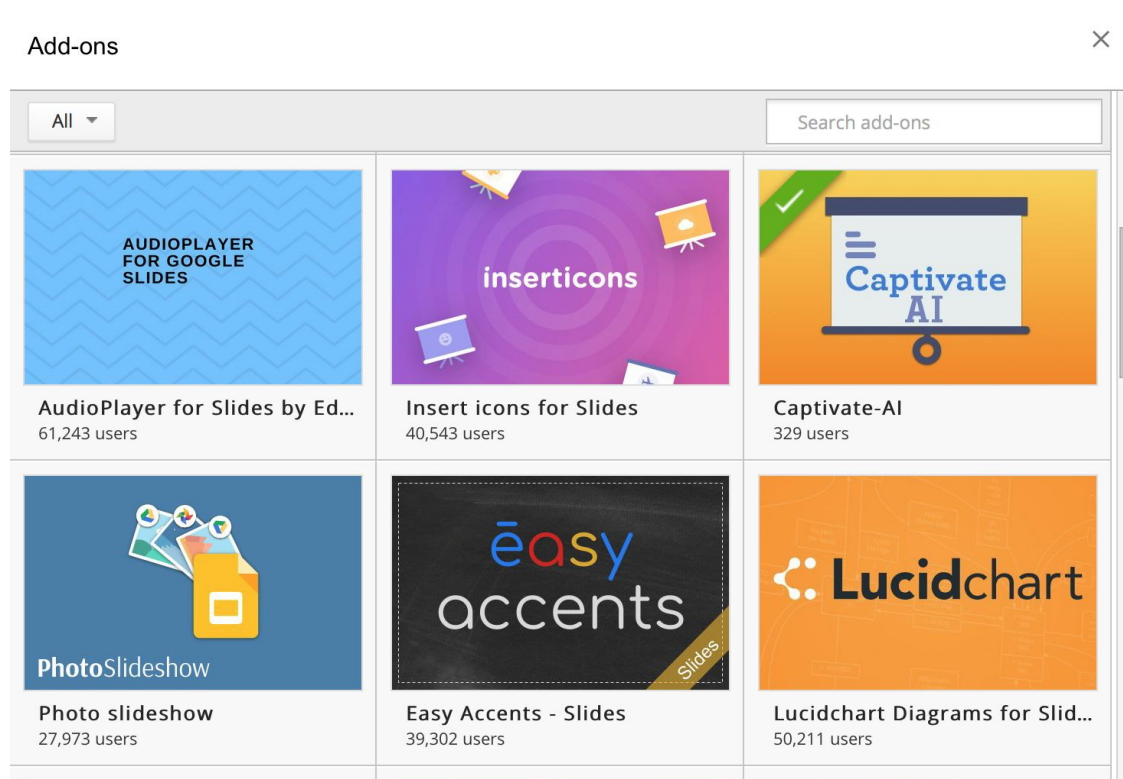
**Vikram Reddy, Nic Mon, Mudit Kakkar**  
**MIMS Capstone Project: Captivate AI**  
**Advisor: Steve Weber**  
<http://captivate-ai.us>

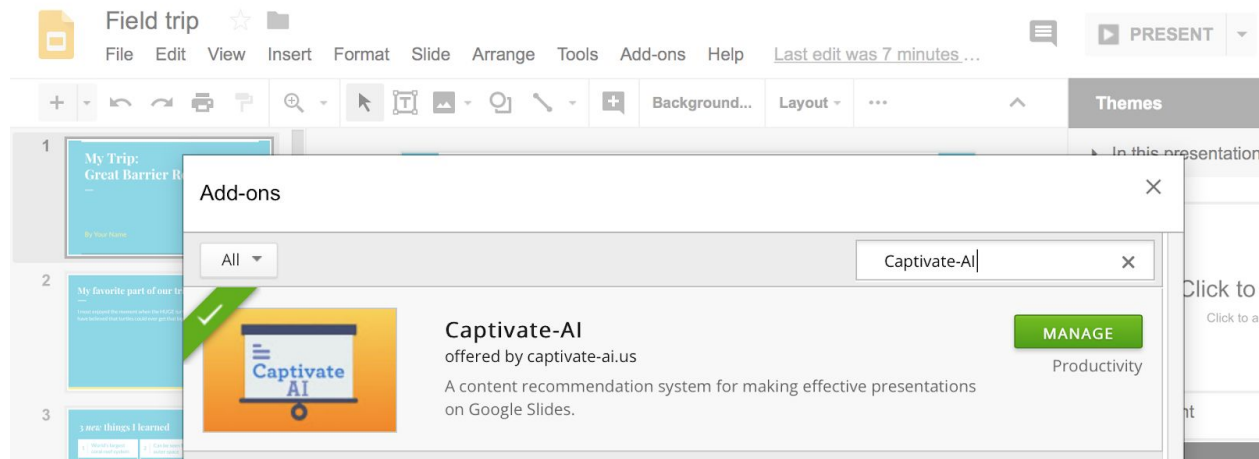
Thanks also to Prof. David Bamman and Steve Fadden for their advice on Captivate-AI's Machine Learning and UX/UI design components respectively.

Our product has been approved for public use by Google Apps Store, and can be downloaded in the Google Slides Add-ons Store. We would like to thank our Google Apps Store Advisor, Ivan Della Valle, for his efficiency and guidance throughout the process.

Captivate AI is a content recommendation system for helping our users make more effective presentations on Google Slides. It works as a Google Slides Add-on, integrating directly with the Google Slides app. The Captivate AI product has received positive feedback from our App Store advisor, who has said "[Captivate AI] is interesting, and it could be very useful for our users".

As of May 3rd we had 329 users:





To download the add-on, open up Google Slides. Then either create a new presentation or open an existing one. Click “Add-ons” on the top toolbar. Then click “Get Add-ons”. This will open up the Google Docs Add-ons Store. If “Captivate-AI” is not found on the front page, you can search for the product in the search bar. Click on the tile and click “Free +”. You are now ready to use the tool.

We built the app using an Agile Development Methodology. We held weekly scrum meetings beginning in October, 2017. From October to December, we discussed overarching design questions. Who were our users? What was the definition of a good presentation? What was a bad presentation? Can a presentation be thought of as storytelling? If so, what is underlying narrative of the story? How do users make presentations currently? What are their pain points and can some of those be solved through a software system? What types of presentations are there?

To answer these questions and more, we designed a user survey using Google Forms and shared it with students in UC Berkeley, professionals in the San Francisco Bay Area, and professionals in India.

Based on about 50 responses of the survey, we formulated user personas. There are many types of presentations and many categories of users. Users could be professors, lecturers, researchers, students, or industry professionals. If they are students, they could be presenting on History, Science, Math, Economics, Art, or hundreds of other subjects. Rather than make a generic tool to specifically help in each category, we decided to focus on a strategy that would cover 80% of all presentations: **goal-oriented delivery**.

### **Goal-Oriented Delivery:**

The goal of any presentation, we propose, is to convince an audience of something. A presenter has something to gain if she or he does well. The verb “to convince” can also be replaced with “to deliver a message to”, “to persuade”, “to spread awareness”, and “to sell something to”.

Similarly, an “audience” can be a lecture hall, a technical team meeting, an all-hands at a company, or a panel of investors. If the presentation goes well, the presenter might win something, achieve a personal goal, get a promotion, or get an investment. The lists for the substitution of each word can be endless, but we believe it is important for our users--the presenters--to *know* what their goals are. As such, we provide a questionnaire for our users that seeks to understand exactly that. The application is two-fold: answers to the questionnaire will guide our recommendations, but more importantly they will allow our users to think deeply about what they want to present about, and *why*. Throughout the tool, we try to empower our users to think about their goals.

### **The Brainstorm:**

Another feature that lets them explore their thoughts is the initial brainstorm session. We ask our users to write down the central theme of their presentation as well as a few bullet points of what they plan to talk about. Again this lets our users think about their upcoming task without the perceived stress of writing everything down perfectly. It is a win in terms of the Machine Learning ahead, too. We use the contents of their brainstorm as a data input into our content recommendation algorithms. Specifically, we use the brainstorm to generate images and quotes.

### **UI Design / UX Flow**

The user interacts with our tool through the sidebar.. The sidebar had tabs for Ideas, Fixes, and Checklist. The Ideas tab was meant for content that the user had not thought of before--just as new quotes or new fonts outside of their standard thematic typeface. The Fixes tab was meant to flag a notification if some of the user’s existing content had some aesthetic errors. Examples of errors include “too much text on a slide” or “using too many typefaces”. We cited Nancy Duarte’s *Slideology* for the gold standard of using a maximum of two fonts in a presentation. Although seemingly stringent for many of us, she believes that the simplicity brings a calm and elegance to viewers in the audience (Duarte 146).

We try to offer suggestions that are contextual, relevant and non-intrusive. Through this, we aim to design a user experience similar to that of an ambient assistant: always present, but active only when you need it.

### **Features:**

The recommendations we give fall into two categories: Content and Aesthetics. Content-based suggestions include images and quotes. The images are guaranteed to be royalty-free and pre-approved for commercial use. Aesthetic suggestions include new typefaces, font size corrections, notifications for too much text on a slide, and replacing a text with a royalty-free image.

Here are the design screenshots:

First, here is the brainstorm session at the beginning of the User Experience:

**What do you plan to talk about?**

Tell us in a few words or sentences what you feel are going to be your biggest talking points. This helps speed up your presentation.

Enter words or sentences

Skip

Next

Then comes the user survey, asking her to fill out her goals:

## A few questions to help you get started

Answering these questions helps you create a more impactful presentation in less time

### 1. Who is your audience?

- ☐ Students
- ☐ Stakeholders
- ☐ Academia
- ☐ Other (please specify)

### 2. How much time do you have to present?

- ☐ 5 minutes
- ☐ 10-15 minutes
- ☐ 30 minutes
- ☐ Other (please specify)

### 3. What is your goal?

- ☐ To convince the audience
- ☐ To sell something
- ☐ To deliver a lecture(?)
- ☐ Other (please specify)

### 3. If this presentation succeeds, what would you gain?

- ☐ I would win a competition
- ☐ I might get investment for my product

Next, the sidebar appears. Below I display all three tabs:

# Captivate

45

to persuade stakeholders

MINUTES

GOAL

AUDIENCE

## Magic ideas

A|

Hide

Here are some typefaces that are a better fit for the content you are presenting.

Times New Roman

Athelas

Georgia

Didot

Batang

Garamond

?

“

Hide

Here are some quotes that can help make this slide more memorable

“Don't cry because it's over, smile because it happened.” — Dr. Seuss

“Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.” — Albert Einstein

?

💡

Ideas

✓✓✓

Checklist

✏️

Fixes

# Captivate

45

to persuade stakeholders

MINUTES

GOAL

AUDIENCE

## Checklist

🟢

Typography

✓

You are using more than 2 typefaces in your presentation.

It is recommended to keep the style and size limited to two each, for the entire deck

✓

“Don't cry because it's over, smile because it happened.” — Dr. Seuss


The quote seems like a good fit


🟢


Images


✓


Here are some recommended images that you can use to replace text on some slides














🟢

Alignment

✓

It is recommended that you left align all the text that is not a title or a heading

💡

Ideas

✓✓✓

Checklist

✏️

Fixes

# Captivate

## 45 to persuade stakeholders

MINUTES

GOAL

AUDIENCE



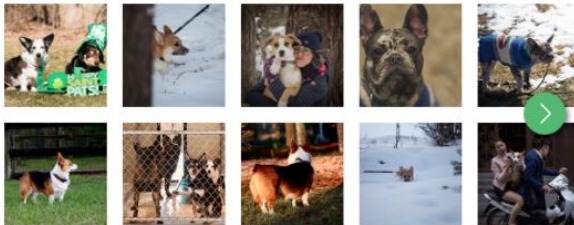
### Quick fixes



The information on this slide could be presented better. **Consider aligning the the content on your slide.**



There is too much text on this slide. **Consider selecting a picture you would like to replace it with.**



Ideas



Checklist

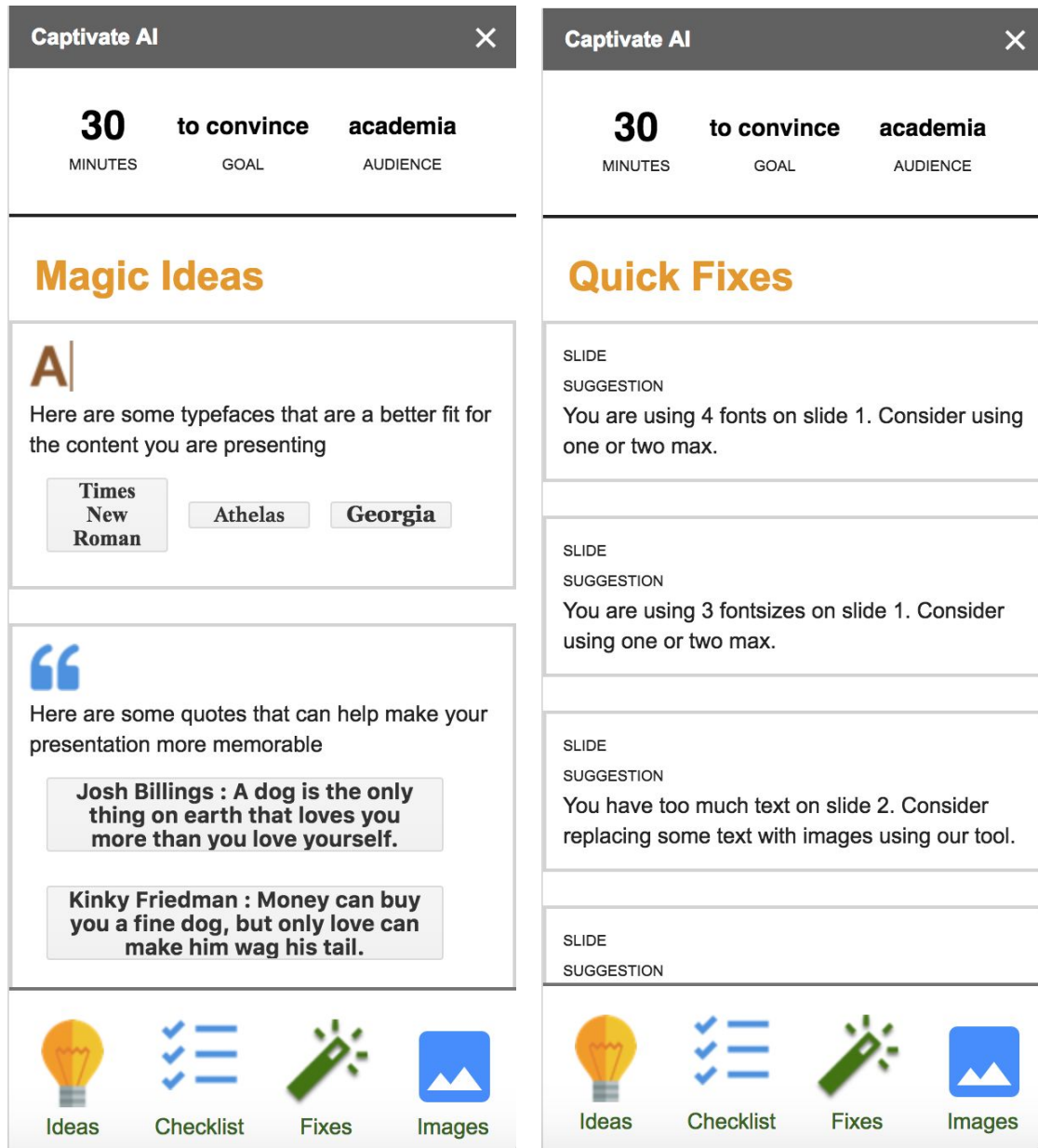


Fixes

## Design Translation into the Product:

And here is how we translated that design into a product:

The brainstorm and initial user survey screens remained the same. We added a separate sidebar tab for “Images”, which contains the image recommendations and the brainstorm to update. We based the bottom navbar on the Apple Music iPhone App’s navbar. Here are the sidebar tabs:





Captivate AI

30

to convince

academia

MINUTES

GOAL

AUDIENCE

Checklist

CLEAN SLIDE RATIO:


3 / 9 slides are clean.


Fonts used


Times New Roman, Comic Sans MS, Roboto, Caveat, Arial, Raleway, Lato.


Font sizes used

48, 36, 24, 12, 14, 9, 10.5, 30.

 Ideas

 Checklist

 Fixes

 Images

Captivate AI

30

to convince


academia

MINUTES

GOAL

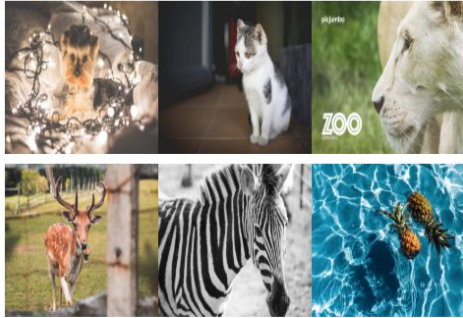
AUDIENCE


Brainstorm





At the Berkeley zoo, I saw many animals, including zebras, horses, and turtles.


SUBMIT



 Ideas

 Checklist

 Fixes

 Images

## **The Software System Architecture:**

We host the application on 3 server instances. The backend database server is hosted on a Google Cloud SQL instance. The server that hosts our in-house Natural Language API is Nic's own home-built server. The server runs 32 gigabytes of RAM and 16 gigabytes of graphics processing. Our Natural Language API that we built is a quote generator that discovers a list of quotes that are similar to the user's presentation. Lastly, our application server is housed on Google Cloud Platform. Google Cloud Platform talks to the Google Slides API, which allows Captivate AI to programmatically insert new slides, images and other content. We also use the Google Cloud Natural Language API to perform keyword extraction and rank the keywords by similarity. Finally, we use a Custom Image Search API that guarantees all generated images to be royalty-free and pre-approved for commercial use.

## **AI: Natural Language Processing**

We built our own Natural Language REST API that generates quotes on demand. The API makes use of Word2Vec, a library of neural networks that gives a dense representation for each word in the training set vocabulary. In our case, we trained Word2Vec against the entire Google News online library. This maps each word to an n-dimensional vector, the dense representation. The vector is then used to compute similarities between keywords in the user's presentation and categories of quotes. These "quote categories" tend to be single words, such as "Business", "Achievement", "Hardship", "Sports", etc. When the word vectors for one of these categories has a high similarity score many of the keywords in the presentation, the API will output the top 10 quotes in that category. On the front-end of the Captivate AI app, we lazy-load the recommended quotes.

## **Software Stack:**

We used a relational database management system for the backend, with JSON as the transactional messaging protocol, SQL as the query language, and a JDBC service to run the transactions from the client. The application itself was written in a mix of Google Apps Script (similar to native Javascript), JQuery, Materialize CSS, Materialize JS, HTML5, and CSS. We built our own REST API for the Quotes generator using Python Flask, a web framework.

We used behavior-driven design to organize the code in the front-end. We also had a widget-first design. Examples of widgets are the sidebar, popup dialogs, and modeless dialogs (an asynchronous version of a dialog). Each widget had a front-end component as well as access to the backend.

## **Development Design:**

The "server" folder housed our own database API as well as the consumption of all internal and external APIs. We communicated to these APIs via POST requests.

Every widget in the software has 4 files: widget.html, widget.js.html, widget.css.html, and widget.js. The word “widget” can be replaced with “sidebar”, “start\_survey”, “create\_template”, and “about”. These represent the 4 widgets we have in the application.

I will walk through each of file types and their use in the application. First, the widget.html file defines the UI of the widget itself. It imports external js and css libraries via Html Services. That is, in Google Apps Script, everything is either HTML or Google Apps Script. In order to get Javascript or CSS, it must first be encoding as widget.js.html and widget.css.html and then passed to widget.html via an Html Service. We built an internal import function, which calls these Html Services under the hood. Finally, the widget.js file houses Google Apps Script, which takes the widget.html UI and loads it into the appropriate Google Apps Script object. For example, sidebar.html is loaded into a Sidebar object via the showSidebar method in sidebar.js.

For all front-end calls to the backend, we used asynchronous design. This means that users could continue to use the product while the system made calls to the backend database and 3rd party APIs.

### **Database Design:**

We are able to support a multitude of different users. Our infrastructure also allows a each user to have many different presentations open with our tool. This is accomplished by using the presentation id as a primary key for the tables of record. Each widget has a corresponding table of record. For example, the initial survey we give users will record answers in the survey table. These answers are then distributed to the template and sidebar widgets via calls to read the database entries.

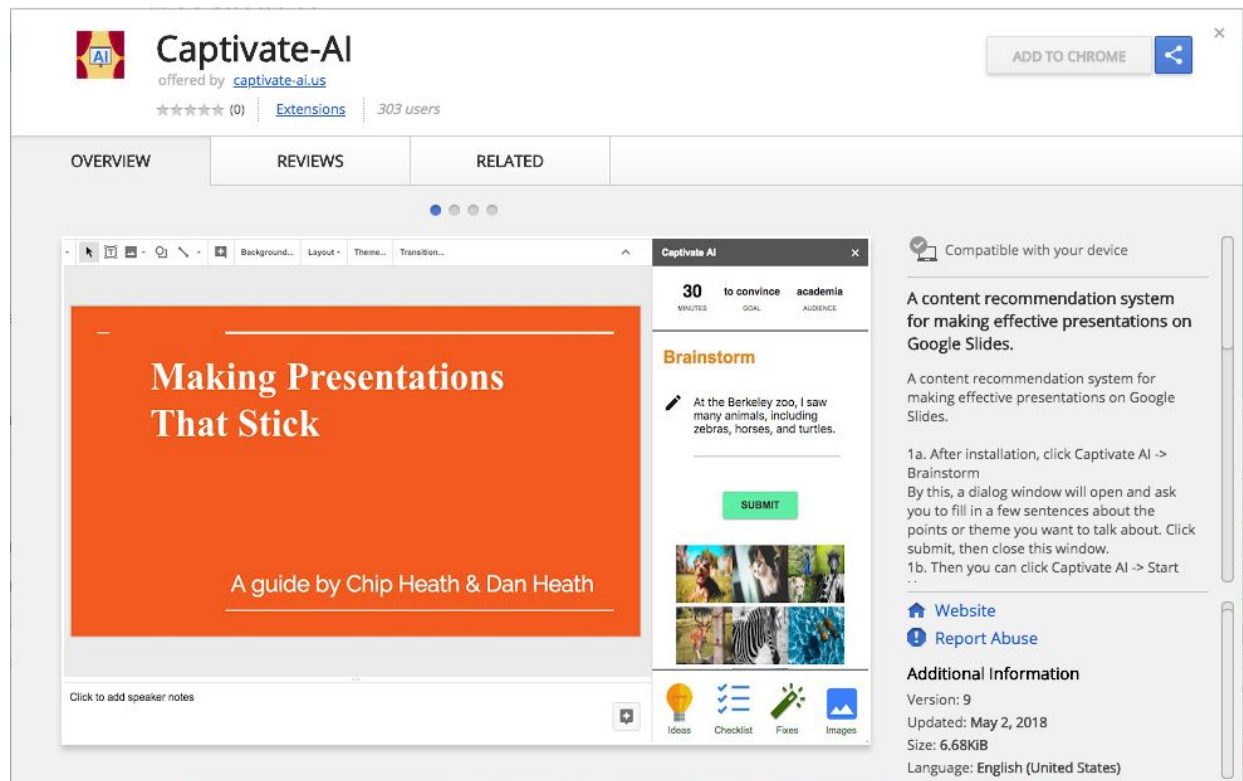
### **Testing and Integration:**

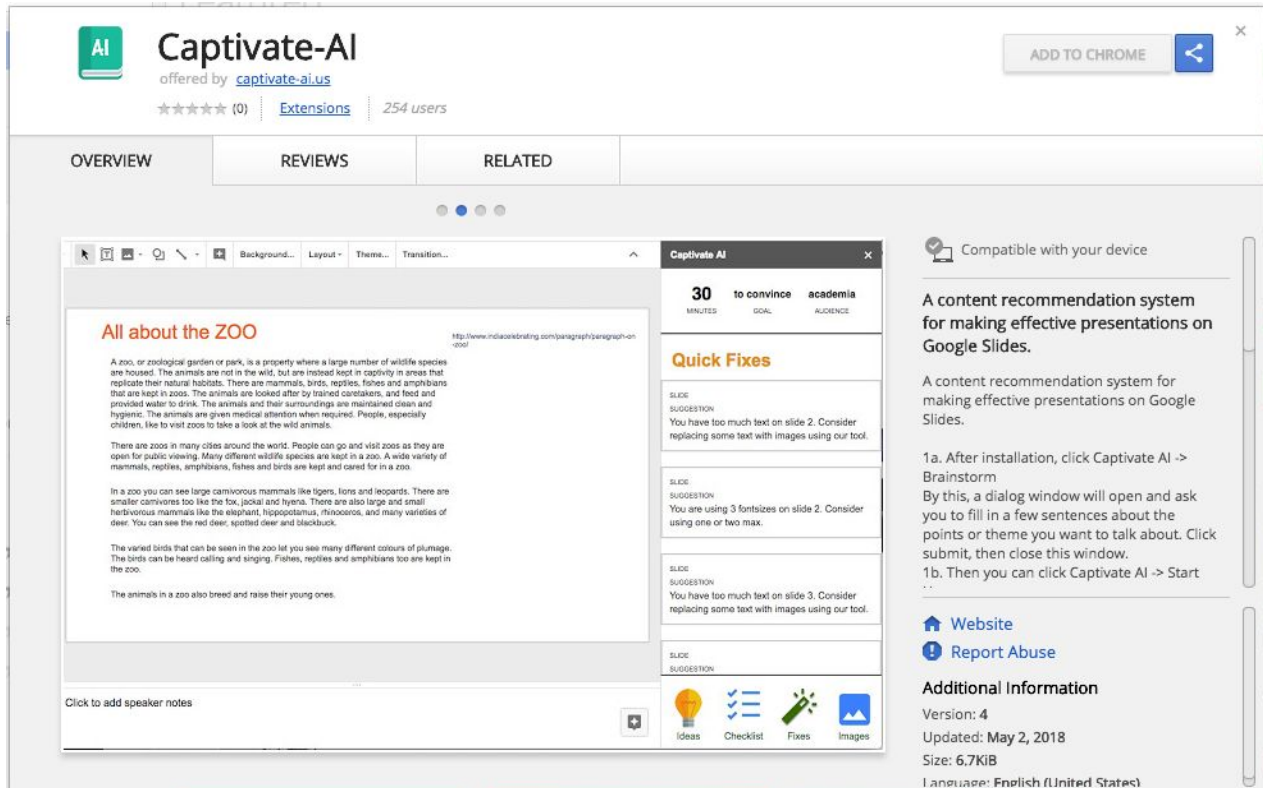
We utilize GitHub and Git Flow to integrate the various features and design work. We also integrated our code with clasp, a Google command line tool, which lets us push code to the Google Add-on testing environment. So our Development and Testing flow is as follows: branch a new feature, clasp push to Google testing environment, test integration and previous functionality. And if everything passes, git push branch to master, and git merge with master.

### **Deployment Process to Google Apps Store:**

Deployment to the Google Apps Store has several requirements. For user experience and onboarding, they require a step by step how to use section, a help section, and an email for support and to request to fix issues. For Product and Company information they require a registered domain as well as Google Verification that we own this domain. Our site can be found at <http://captivate-ai.us>. Lastly, for all promotional ads, the install screen, and the marketplace views, they require about 10 different images, including screenshots of the product.

This is how the product looks on the Chrome Web Store page:





## Feedback from Google:

Our advisor, Ivan, then gave constructive feedback on how to pass Google’s security and privacy verification check as well as Google’s User Experience requirements. The latter requirements were security measures that constrained our initial user designs. In particular, Google Add-ons such as our own are not allowed to open up dialogs, sidebars, or any other widget, without the user explicitly clicking an “open” button. After further design and development iterations, we were able to strike a balance between in-app security and a fluid user interface.

## User-Centric Privacy Policy:

In response to Ivan’s feedback, we drafted a Privacy Policy and hosted it on our site: <http://captivate-ai.us/privacy-policy>. The privacy policy begins by giving our users a high-level description of the policy, in plain English. It then dives into the legal terminology. We hope that this small addition of the easy-to-digest summary will set a precedent for future applications to follow.

On April 26, 2018, we were verified for our privacy measures on all our features by the Google Cloud Platform and API Safety Team. We were also granted OAuth tokens for all the scopes we used.

### **Further Iterations (Design Constraints):**

Because the app requires that a user explicitly click on an “open widget” button before a widget opens, we had to redesign our UI and UX a bit. Specifically, the onboarding screens were no longer valid. We had designed them to pop open as soon as the user installs the app, so that she can immediately learn how to use the app. Instead, we provide the onboarding process in text-form as a post-install tip (per Google App guidelines). If the user needs more help, she can click the About Captivate AI section for a full list of instructions, our website, and support email.

### **Successful Deployment in App Store:**

After these changes, our product was approved and entered the App Store on May 2nd, 2018.

### **UX Research:**

We conducted 8 usability tests, including one with Berkeley Lecturer and Director, Analytics UX Research at Salesforce, Steve Fadden. From our research, we found that users liked certain features, disliked other features, and wanted to see some things that weren't there. For example, all 8 participants immediately said they expected a template to be generated when they completed the user survey.

What they liked:

They liked the concept of the brainstorm and user survey at the beginning, which feeds into the sidebar and the content suggestions. They also like that the magic ideas may give them new ideas that they may or may not use, but that they have as options. They liked that the checklist gives them a score of how well they are reaching their goals.

What they didn't like:

They didn't like the alignment function. Most people, when asked “where would you go to arrange text”, would think about it and go to the Google native taskbar. When pointed to the alignment section of our tool, they all commented on how they would rather use Google's alignment feature. They cited trust and capability as reasons they would prefer to use Google.

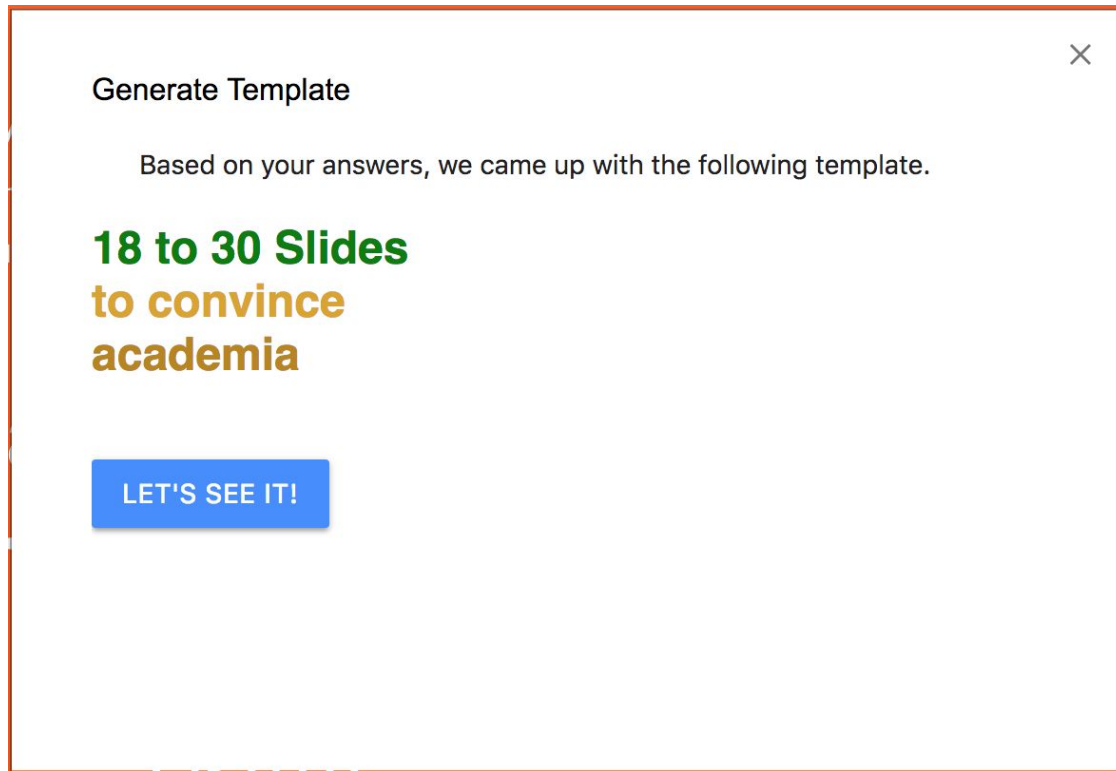
What they thought was missing:

As I mentioned, they unanimously wanted and expected a template after the user survey. The users differed, however, in what they envisioned the template to be. One person wanted it to shape the tone and imagery of the presentation (happy, pleasant, serious, funny, etc.). We had decided early on that this would be a technical improbability to implement, since emotions and are subjective and users themselves may not be able to define an appropriate emotion for a presentation. The majority of users wanted the AI to apply some subset of colors of theme to the template.

### **New Features from Research:**

They spoke, we listened. All user tests expected there to be a template after the user survey. So we built one. After the user survey, users have an option to generate a template. Some features were hard to let go of: the alignment tool was something we all thought would be a hit. Though after speaking with the users, we realized that their instinctual response would be to look for alignment in the Google native taskbar, rather than in our tool. Thus, we removed the alignment feature from our product.

Here is the template generation dialog as it appears in our product:



### Continuous Improvement / Future Work:

We know that improvement of our product does not stop with our 50 user surveys or 8 user interviews. Improvement is continuous. We are continuing to envision new features and new ways to ensure our users to have an enjoyable and intuitive experience. Some of these future features include: narrative-based suggestions. Examples of narrative-based suggestions are deleting an unnecessary slide, recommending a slide that is similar to the storyline, and embedding a starter story into the template (beginning, middle, climax, and end).

We hope you enjoy reviewing our product as much as we enjoyed building it!