32 Project Outline

Overview:

Team members: Laura Wilson, Camillo Saueressig, Grace Bramley-Simmons, Daniel Park

Strengths & Weaknesses:

	Strengths	*Weaknesses*
Laura Wilson	Front end, ui/ux, graphics	Non-visual backend, algos
Camillo Saueressig	Algos, generics	UX, CSS
Grace Bramley-Simmons	JavaScript, web dev	Testing, QA
Daniel Park	Algorithms and debugging	Front-End Languages

Project Ideas:

1. **Game that teaches users about data structures.**

   Problem: There aren't resources available for learning about data structures in a fun, visual, and entertaining way.

   Solution: Games that illustrate how data structures work.

   Critical features
   1) Lobby or world that users can navigate through freely and select which minigame they would like to play.
   2) Individual Games
      a) They illustrate 1+ data structures.
      b) Backend validates moves and keeps track of score and level.
      c) Frontend visually communicates if moves are valid and allows user to manipulate nodes, etc to construct or search structure.

   Possible Additions: Progress saving, open world, multiplayer, level up, items

   Challenges:
   ● Making games that are both informative and fun.
   ● Making the games extendable such that it would be easy to add more.

   Question: Can we code in this React?

2. **Handwritten document to Latex/Digital document converter**

Problem: CSCI 0220 – Introduction to Discrete Structures and Probability

Solution: Freeform converter/formatter website that can transform pictures of written work to digital representations.

Critical features
1) Image analysis to Latex/digital output, with formatting preserved.
2) Word count manipulator that allows you to extend or shorten chunks of text to be the desired word length.
3) Ability to upload images and download/export the formatted result.

Challenges:
1) Reproducing handwritten text/formulas digitally with high accuracy
2) Recognizing variable features in handwritten text, e.g. a fraction bar can be quite short or really long, 'x' can represent x or times.
3) Doing image/NL processing in java. Might have to make an api in python.

3. **Collaborative Real-time Meeting/Brainstorming Space**

Problem: No application to host live meetings in a shared, freeform, and visual workspace at the same time. Applicable to situations like brainstorming sessions, online tutoring sessions, etc.

Solution: A collaborative environment that hosts audio calls, while also providing a space where all participants can brainstorm or illustrate their ideas through adding images, text, and drawings. A speech-to-text feature would record the conversation of the meeting, and there would be a feature that analyzes workspace content and offers related content from the web to add to the workspace, thereby furthering the brainstorming process.

Critical Features:
- Brainstorming space (freeform canvas), which would allow for drawing tools, text, mind maps, import files, etc.
    - Multi-user MS Paint.
- Audio chat (Skype API) - w/ speech to text
    - Displayable and downloadable transcript of meeting notes
- Google Explore feature, receive suggestions for additional media based on transcript or workspace content and be able to search google within the app.

Challenges:
- Integrating Google and Skype features seamlessly
- Storing content in the backend so that it's accessible by multiple users
- Finding a natural language processing module for chat
- Saving state between sessions.
- Picking up on relevant text from canvas