

Eyal:

- Strengths:
 - Organization/being on top of stuff
 - Really enjoy these projects/coding
 - Very motivated → will work a lot to see things through
 - Team communication
- Weaknesses:
 - Code can get messy (getting better!)
 - Not the fastest coder
 - Front end, but willing to improve a lot

Ben:

- Strengths:
 - Ok with doing nothing but coding for a week or two
 - Detail-oriented
 - Like to start early
- Weaknesses:
 - Not a lot of programming experience outside of 32
 - Don't know how to do front-end stuff
 -

ShalanStrengths:

- Time management: Start things early and try to pace things out
- Assessing group needs and making sure everyone is involved/heard
- Enjoy integrating code and getting big parts to fit and work together
- Seeing the global view of a project
- finding bugs

Weakness:

- CSS: Not very familiar with flex box/spacial formatting
- Takes longer to fully understand algorithms sometimes
- Can get stressed out about a project and start doing other people's shares

NateStrengths:

- Time management, able to plan things out well, like to start early
- Can code for long stretches of time
- Communication
- Debugging

Weaknesses

- Sometimes need time to fully understand the project/algorithm/implementation before I begin coding
- Frontend

Ideas:

1. Every year, colleges generate extreme amounts of trash. A lot of the stuff can be reused by students. Website would be a marketplace for college students to sell and buy and rent used textbooks/furniture/storage space.
 - a. Features:
 - i. Can search for items you need
 - ii. Searching for storage people are willing to give up for a small fee
 - iii. Special feature for selling items at the end of the year(most trash generated at this point)
 1. Issue with storing items over the summer
 - iv. Sale Day: People bring listed items to location. (big garage sell)
 1. Beginning, middle, end of school year
 - v. Does the company take a cut of \$ items?
 - vi. Eco Friendly Solution to college trash
 - b. Challenges:
 - i. need to store stuff for a summer?
 - ii. guarantee quality of listings?
 - iii. delivery of merchandise
 - iv. 3rd party moving/delivery companies
 - v. payment
 - vi. Searching algorithms
 1. Filtering by price, item type, rent or buy..., search by location/school
2. An application that tells you all the awesome dishes you can make by just providing the food you have in your fridge to save people time and money
 - a. Features:
 - i. Can enter ingredients and return recipes you can make
 - ii. Build up preferences based on previous recipes
 - iii. Determine user's preferences based on previous dishes made and show dishes they are more likely to like
 - iv. Companies can pay to have their recipes boosted in the search algorithm
 - v. Autocomplete recipes, eg you can make X with just 1 more tomato!
 1. Weighting system for recipes based off of how many extra ingredients you need, 100% match with your ingredients
 - b. Challenges:
 - i. There are existing competitors (though this could be seen as an advantage)
 1. Can make it more user friendly?
 2. We can make it more intelligent
 - ii. Getting the data/recipes (from recipe websites?)

- iii. Figuring out the algorithm we would want to use
 - c. Why do we like this?
 - i. Involves a lot of creativity and could implement a ton of cool features
 - ii. very open ended yet with a clear goal.
- 3. BandManager: App that helps bands form and organize themselves. Each user has a profile with equipment owned/skills/description/what they're looking for. Users can create a group that represents a band and seek new band members. Bands can also look for equipment/venue they need.
 - a. Features:
 - i. Band component
 - 1. Calendar/reminders
 - 2. Have a "manager" who handles important stuff
 - 3. Information about upcoming gigs, stuff like that and reminders.
 - 4. Display things needed for shows, rehearsals, buying/selling
 - 5. Recruitment / band profile including skills, prev performances, etc.
Band can be either "closed" or "open" to indicate whether they are looking for new members
 - 6. WhenToMeet function?
 - ii. Individual component (required)
 - 1. Profile
 - Could list skills, favorite genres, etc..
 - 2. Sign up for jam seshs, band applications
 - 3. See upcoming gigs
 - 4. Have a feed
 - b. Challenges:
 - i. Determine what to show a user in the feed(public/private events)
 - ii. Searching algorithms:
 - 1. Location
 - 2. Skills needed
 - 3. Genre
 - 4. What goes into a person's feed(filtering info that applies to person)
 - c. Why do we like this?
 - i. Seems useful, addresses problems some of us have/have had before
 - ii. A ton of functionality to implement, could be really good at the end