What features were completed?

Front-end	Back-end
- Sign in page - Sign up page/form - Profile page - Flash cards + animation - Decks page - Make new deck - Quiz - CSS and JS also working with pages	- Created all tables for - User - Saved lists - Public lists - Word accuracy - Sessions - List improvements - Implemented backend node docker containers - Got all pages running on node - Implemented login & sign-up system through node - Implemented session system to ensure only logged-in users can access backend pages - Implemented password salting & hashing

What worked during the demo? – Demo lasted 12 minutes

- The database is able to be spun up on a local host
  - Insert statements, docker container
- "My decks" displays properly
- Login page and sign up page display no way to go back accidentally after sign up/sign in
- Account page allows users to see their stats and decks
- Flashcards are able to be navigated through by the user
- What issues were faced either during the development or during the demo?
  - At the time of demo the quiz page was incomplete and just a html skeleton, since then it has been updated to take in input and quiz users.
- What were the suggestions offered by the TA?
  - Rather than using a specific path, create a named value. Folder is still created, but it is away from the repo, minimizing clutter (Docker)

- ○ Use pg-promise to connect scripts to postgresql
- ○ Arrows in the diagram go both ways >:(
- ● Individual contributions by each team member since last milestone
  - ○ Beckett
    - ■ Created docker containers for node and DB
    - ■ Implemented get functions for front-end pages on node
    - ■ Implemented sign-in/sign-up functionality
  - ○ Joe
    - ■ Modified database files and their docker container.
    - ■ Implemented password hashing/salting algorithm for sign-up page.
    - ■ Also added comparison functionality for when users log in.
  - ○ Max
    - ■ Created flashcard animation: cards move left and right, cards flip back and forth, words are pulled from a JSON object.
    - ■ Worked on quiz interface. https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-016-04/commit/08917bdadd458f717618e9cc8cf96e5ba2642fc5
  - ○ Amanda
    - ■ Created login/signup page, profile page, made sure html pages were connected and working correctly
  - ○ Matthew
    - ■ Helped to set up the database files and test their functionality
  - ○ Heather
    - ■ Worked on deck page and cards page
    - ■ Updated nav bar some link adjustments/added a profile icon
    - ■ Created a build new deck page with form

Front End/Browser

HTML/CSS

Front End JS/ Bootstrap

Merriam-Webster API.

- Query Definitions of Words
- Query Translations
- Query audio files of pronunciation
- Query parts of speech
- Query conjugations

NodeJS backend

Account login / sign up.

Quiz Stats

Manually create sets.

Generate set with API.

Store and query lists.

User stats.

REST API

pg-promise

Database (PostgreSQL):

- User Data Table
- Table of Saved Lists
- Table of Public Lists
- Word Accuracy Stats Table
- List Word Accuracy Stats Table
- Session Stats Table
- List Improvement Stats Table

google-cloud Node.JS Client Libraries

Possible Google API.

Text-to-speech translation of card.