Issues

- ☑ ~~Log in page - sign up and sign in (Jesus)~~
- ☑ ~~Home page (Justin)~~
  - ☑ ~~View random word of the day (Justin)~~
  - ☑ ~~View definition for word of the day (Justin)~~
  - ☑ ~~View example for word of the day (Justin)~~
  - ☑ ~~View synonyms for the word of the day (Justin)~~
  - ☑ ~~View antonyms for the word of the day (Justin)~~
  - □ View rhymes for the word of the day (Justin)
- ☑ ~~Home page pt. 2 (Roy)~~
  - ☑ ~~Add new post (poetry post) (Roy)~~
    - ☑ ~~Edit your post (Roy)~~
    - ☑ ~~Delete post (Roy)~~
  - ☑ ~~Text box~~
  - ☑ ~~Posts of the Day~~
  - ☑ ~~Your Word of the Day~~
- □ Home page pt. 3 (Shannyn)
  - ☑ ~~See other posts on that same word of the day (Shannyn)~~
    - □ Add thumbs up and thumbs down (Shannyn)
- ☑ ~~Viewing User Profiles Page (By Clicking on User) (Shannyn)~~
  - ☑ ~~View other users' posts (Shannyn)~~
  - ☑ ~~User Bio (Shannyn)~~
- ☑ ~~Search for a word page (Jesus)~~
  - ☑ ~~Display all posts with the searched word (Jesus)~~
- ☑ ~~Search for a user page (Justin)~~
  - ☑ ~~Click on a user profile to see their bio + all their posts (Justin)~~
    - ☑ ~~On the home page, click on a user to view their profile/posts (Justin)~~
- ☑ ~~Previous words of the day (Shannyn)~~
- ☑ ~~Add style (Jesus)~~
- ☑ ~~Add the API to the project (Shannyn)~~
- ☑ ~~Create a database for users(Jesus)~~
- ☑ ~~Time reset (daily random word) (Roy)~~
- □ Favoriting Posts
- ☑ ~~Edit Bio~~

Brainstorm

- Add a new post
- Below there would be a list of all the previous ones
- View user profiles and see their postings (Different tab? Or Something similar)
    - Maybe clicking their name becomes a hyperlink that sends the user to that page of their profile and postings
- Incorporate when the day of the word resets
    - Might need an API for that
- History of the previous words of the day and their posts associated with it (MAYBE ~)

- Potentially using the random feature from ‘wordsapi’’ to get random words
- Definitions
    - The meaning of the word, including its part of speech.
- Synonyms
    - Words that can be interchanged for the original word in the same context.
- Antonyms
    - Words that have the opposite context of the original word.
- Examples
    - Example sentences using the word.

- Rhymes (not as a big priority)
    - Could be useful
    - Or the poster can just make the post individually

- We do not need to host it
- Database
    - Create locally (expo sqlite)

- Profile pictures
    - Premade images
    - Or user uploads them

Database categories:

User ID	username	bio	# of posts	image

Post id	Selected word	title	description	Author Id (username)

User

- UserId PK AUTOINCREMENT
- Username Varchar(225) unique
- Password Varchar(225) unique
- Email Varchar(255)
- Bio VARCHAR(225)

Posts

- PostId PK
- UserId FK
- Date data
- text_quote Varchar(1000)

API's

- https://dictionaryapi.dev/
- https://www.wordsapi.com/docs/#rhymes

- If it has a finite amount of api calls
    - Designate a separate key for presenting
-

Git

- Assign remote repository to a local repository
- Be able to pull a remote repository to your local machine
- Make commits

By Week 2, the app must have been started

- Api set up and some info pages

# Week 2 For Iteration Report

- Have small issues to push
- Styling
- Nav bar
-

Must haves (non-negotiable)

- MUST allow users to log in and see something specific to them
    - Their list of words
    - Their posts
- MUST pull from a web based REST API
    - See here for a list
    - Import Yeti
- MUST use React Native/Expo
- MUST use local storage
    - https://github.com/andpor/react-native-sqlite-storage
    - https://www.youtube.com/watch?v=GoDERit8mVo
    - 100% inflexible on this. I don't want data stored on Firebase.
- MUST use Git and Github
    - Each week I will check for a pull request
    - The individual contributions to the github repo will be a major part of the evaluation of this project.
        - This means everyone MUST contribute. I don't want to hear "I didn't do anything because…"

Minimum Viable Product (MVP)

- □ Authenticate users
    - □ Look in to OAuth though it is not required
    - □ It will be required for project 2
- □ Fetch data from an API
- □ Store information locally
- □ Display some information
- □ (optional ish) Create users

Documentation

- Readme
    - How the code works, how to run it, what it's supposed to do, how to install it, what it is, what it does
    - For code: this is what this does, etc…
        - Comments in the code should explain the why (why it's doing that)
        - and be clear what it does

- Like if you’re setting a float value to temperature, but we don’t know what unit it is (why and how it is in Celsius instead of Fahrenheit)

Unit Tests

- Database operations
  - INSERTS, UPDATE, DELETE
    - Testing these operations to see if they actually work
- Minimum maximum words for post
- format/number/type conversions (like date conversion)
- Navigation testing?
- Testing to see if API entries exist
  - If you’re getting a JSON response, you should be able to get a response and that field should still exist (check if you can run API calls and make sure the field you expect are there)
  - Run the test on a known good word and see what is to be expected

- Make sure to test with curl

# Coverage
Npx jest –coverage