LIST OF COMPONENTS

- Accounts
    - *Create Account* → User will be able to go to a page where you can create an account. It will have you insert user information including username and password. It will only be valid if the username does not already exist. Once created, the user's profile will be added to the profile database.
    - *Login* → User can navigate to a login page where it will ask for the username and password. This component will need access to the profile database to authenticate the user.
    - *Logout* → This will log the user out of their account and go back to the site's home page.
- Navigation Between Pages
    - *Menu* → At either the top or bottom of the page, there will be a menu with possible pages the user can visit or things the user can do.
        - Possible entries:
            - Help Page
            - Login/Logout
            - Home Page
- Stories
    - Found on the /stories page, there will be a list of stories (user has not already added to) that the user can add a new line to. When they choose to do this they will only see the latest update to the story.
    - Any story that the user has already added to will be shown in the user's homepage (where you land directly after logging in)
    - On the user's homepage/stories page there will be a button saying "New Story" where the user will create a new story. It will ask for the name of the story and the first line before submitting.

TASKS

- Be able to log in/register (Kevin)
    - Getting and processing form data
    - Need to be able to compare the data provided via the forms to the data present in the database
    - Establish a session that doesn't disappear
- Database design (Emory)
    - Creating modules that add to each table
    - Being able to access information from each database

- The ability to view stories (Raymond)
    - Only show the latest update and author of the story
    - Disable ability to edit if a user has already written to it
- The ability to create stories (Kevin)
    - Adds a whole new table to the database
    - Requirements: a title and non-null text submission
    - Log the fact that a user created this and add to their landing page

- The ability to add to stories (Raymond)
    - Only a set amount of words allowed
    - Log the fact that a user edited this and add to their landing page

DATABASE LAYOUT DIAGRAM

Accounts table

Username (TEXT) [NOT NULL] [PRIMARY KEY]	Password (TEXT) [NOT NULL] ← probably don't want plaintext
"pollsmor"	"AadUYbrfV5mLKXp7Dp93pcUUMwkyyE"
"xXx_yolo420gamer69_xXx"	"1234"
"iLoveFortnite"	"vbucksaregreat"
"..."	"..."

Some notes:
- If we have time, we should implement some password restrictions. Maybe:
    - At least 8 characters long
    - Must contain both numbers and letters

Stories table

ID (TEXT) [NOT NULL] [PRIMARY KEY]	Title (TEXT) [NOT NULL]	Creator (TEXT) [NOT NULL]
"iplaypokego"	"I Play Pokemon Go Everyday"	"pollsmor"
"wattsim"	"Wattpad Simulator"	"iLoveFortnite"
"memes"	"Grand Tome of Dank Memes"	"iLoveFortnite"

Some notes:
- The ID is necessary so that the individual story tables are accessible, and ensures that the URL doesn't get too long.
- Everything here is NOT NULL besides the HTML code - the creator could've created an empty story if they wished to.
- If a user is trying to access a page that has already logged them editing it, they will be denied.
- The creator column really isn't *that* necessary but whatever. Nice to keep track.

Individual Story Table → name of table is the ID found in Stories Table

Text Update (TEXT) [NOT NULL]	User (TEXT) [NOT NULL]

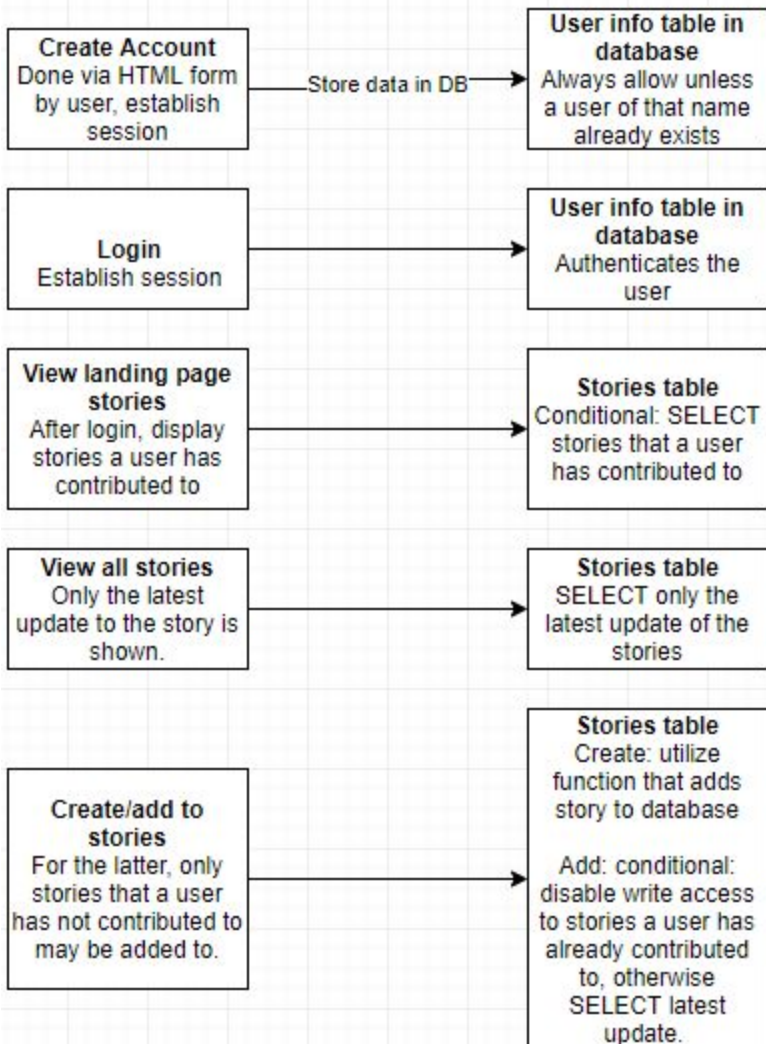| "Once upon a time"	"userKid299"
"Stuyvesant High School is"	"peterStuyvesant"

Some notes:
- Each story found in the Stories table will have their own table
- The name of the table is the ID
- The first line of the table is the initial line of the story and the user who created it
- The bottom line always contains the most recent update
- On the landing page route, this table is perused to see if the 2nd column contains the user's username.

Component Map

FRONT END (client)   **BACK END (server)**

Create Account
Done via HTML form by user, establish session
— Store data in DB →

User info table in database
Always allow unless a user of that name already exists

Login
Establish session →

User info table in database
Authenticates the user

View landing page stories
After login, display stories a user has contributed to →

Stories table
Conditional: SELECT stories that a user has contributed to

View all stories
Only the latest update to the story is shown. →

Stories table
SELECT only the latest update of the stories

Create/add to stories
For the latter, only stories that a user has not contributed to may be added to. →

Stories table
Create: utilize function that adds story to database

Add: conditional: disable write access to stories a user has already contributed to, otherwise SELECT latest update.

SITE MAP

https://drive.google.com/file/d/1J1izBrSZjEg1ZRYWbdt-4W-ad2F3l_Ac/view?usp=sharing