Quiz App

An app that lets you create quizzes and share them between friends. The creator of the quiz can view and share all the results at the end of the quiz.

Requirements:

- **users** can create **quizzes**
- *** users can make their quiz unlisted (make them private and not available on the home page, but if someone knows the quiz URL they can visit and take the quiz)
- users can share a link to a **single quiz**
- users can see a **list of public quizzes**
- users can see a list of public quizzes on the home page
- users can attempt a quiz
- users can see the results of their recent attempt
- users can share a link to the result of their attempt

MVD:

- Create public quizzes
- Creator can share links of created quizzes
- Users View list of public quizzes
- Users can attempt a quiz
- see results of quiz

- **Private quizzes separate from Public**
- **Share Link to results**

Extras

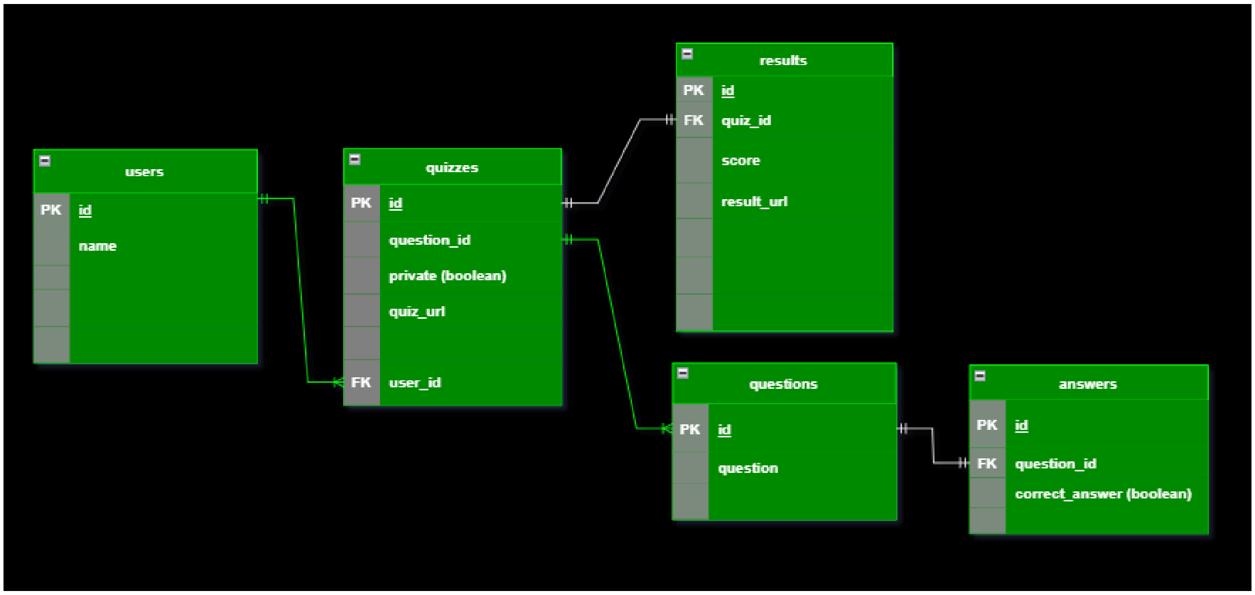
- **char counter for Questions, Answers.**
- **score**
- **robust results page**

User Stories: quiz_maker, quiz_taker [add more when you think of one]

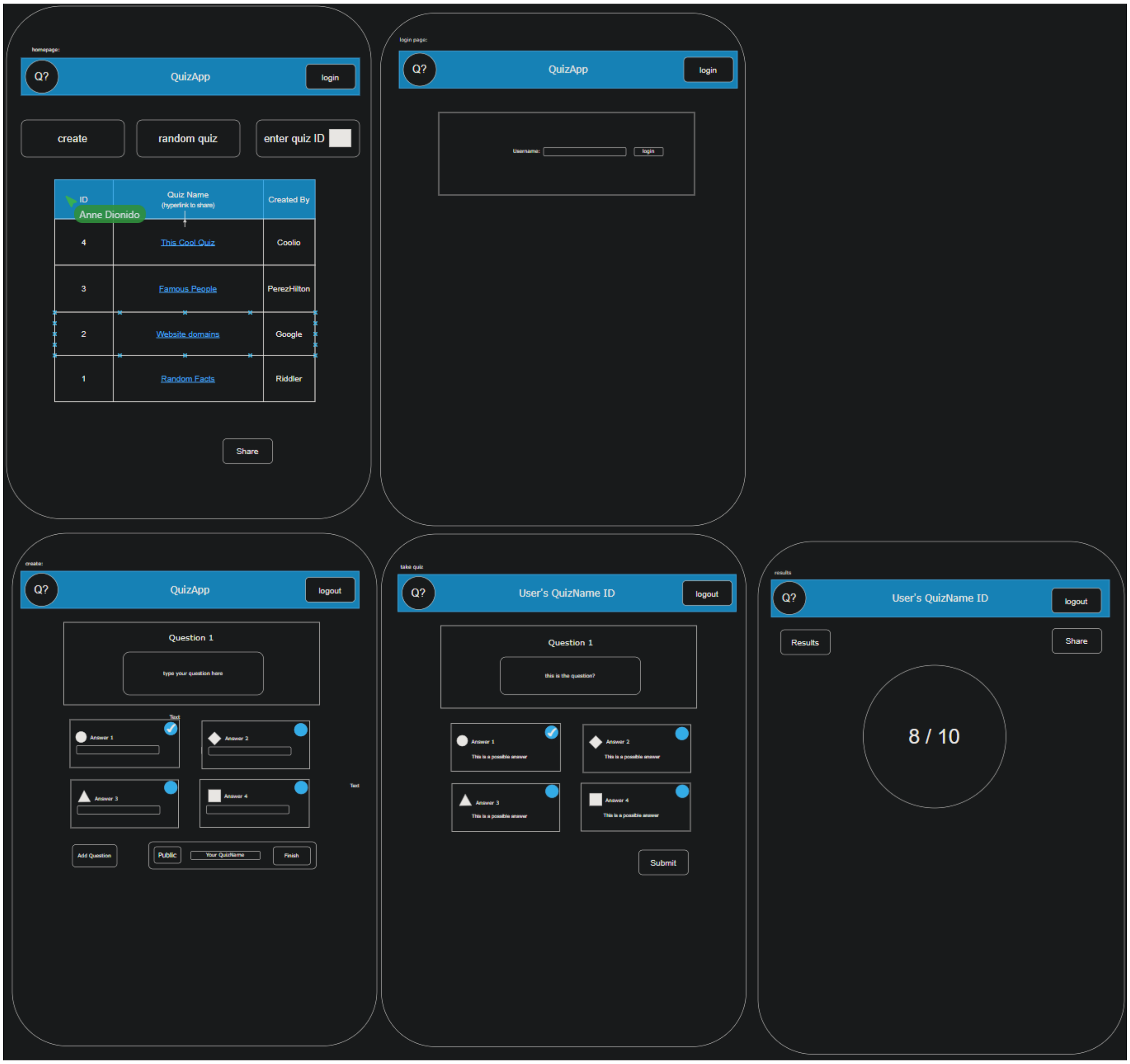
As a quiz taker, I want to take quizzes to test my knowledge

As a quiz maker I want to be able to create and share quizzes to test knowledge

ERD: Rick



Wireframes: Rick & Sandeep & Anne
https://app.diagrams.net/#G1Y89en1LWQeCoLFy3m9weybHicmVMBYhV



ID	Quiz Name (hyperlink to share)	Created By
4	This Cool Quiz	Coolio
3	Famous People	PerezHilton
2	Website domains	Google
1	Random Facts	Riddler

User Login:

```
// do this instead
app.get('/login/:id', (req, res) => {
  // using encrypted cookies
  req.session.user_id = req.params.id;

  // or using plain-text cookies
  res.cookie('user_id', req.params.id);

  // send the user somewhere
  res.redirect('/');
});
```

Fork project skeleton via Github: Rick

Stack Requirements
Your projects must use:

ES6 for server-side (NodeJS) code
NodeJS
Express
RESTful routes
One or more CSS or UI "framework"s:
jQuery
A CSS preprocessor such as SASS, Stylus, or PostCSS for styling -- or CSS Custom properties and no CSS preprocessor
PostgreSQL and pg (with promises) for DBMS
git for version control

Splitting work:
Start vertical (divided tasks) no conflicts with files
Pair programming to work on trouble areas and get through difficult pages.

Discord to arrange meetings (hangouts/zoom/etc)

Header.
QuizApp ‘logo’ left←distance between→ login/logout with name

Main body - 1. Create a Quiz 2. Take a random quiz 3. Take a Specific (Quiz ID required)

1. create
   (minimum of 3 questions)
   (check mark to mark the correct answer)
   (add question button at the bottom-left corner)
   (name your quiz & finish button at the bottom-right corner)

2. Take a Random Quiz

3. [quiz id] take quiz (specific)
   (take a specific quiz, the quiz ID would need to be provided to access the quiz)

EJS is required. Utilize multiple .ejs files, one for each page.

header:	partials/_header.ejs - Rick
homepage:	index.ejs Rick
create_quiz:	create_quiz.ejs -Sandeep, Anne - changes pushed
take_quiz:	take_quiz.ejs Rick, Anne, Sandeep - changes pushed
results:	results.ejs Anne - Initial commit

functionality required to implement:
_ starting database for quizzes - Rick (complete)
_ scss to apply to all ejs pages   - Anne (complete)
_

db files: (quizzes - Rick)
✔create **schema**/quizzes.sql
✔create **seeds**/quizzes.sql
✔create **routes**/quizzes.js
✔create **routes**/quizzes-api.js
✔create **queries**/quizzes.js

^do the same for rest of ERD (quizzes, questions, answers, results)

✔questions: schema, seeds, routes, routes/..-api, queries - Rick
✔answers: schema, seeds, routes, routes/..-api, queries - Rick
✔results: schema, seeds, routes, routes/..-api, queries - Anne

we will need to expand our **queries to cover specific requests from each page.

mentor suggested step. however unclear of implimentation/use :
in server.js
app.get('/quizzes' to call the route quizzes.js to return json data to
res.render('index')

general FYI on res/req flows:
request flow: browser /index -> routes/quizzes.js -> queries/quizzes.js
response flow: routes/quizzes.js render the index with the data from database -> browser

Updates to “Create”   -Anne (complete)
—
separate file to two sections.
Quiz Title __________
Your Name _________
make private? ______
[Next]

Add another ejs. (create_quiz2)
for question details.
—
^for create1/2 next button submission on form instead of

reverted to single ejs by Sandeep after feedback from Mentor

order of events users example:
public/script/users.js AJAX call ‘/api/users’ which is
routes/users-api.js)
invoke func usersQueries.getUsers() from **db/queries/**users.js
then routes/users-api.js (line 15) is responding with json file back to the AJAX call in public/script/users.js which creates $var as jquery selector for element of id **users** from **users.ejs** (`ul id="users"></ul>`)

Get your branch up to date with the latest changes in the master branch:

```
git checkout my-branch
git fetch
git checkout master
git merge
git checkout my-branch
git merge master
git push origin my-branch
```

Get my-branch into master

```
git checkout master
git pull  origin master
git checkout my-branch
git merge master
git push origin my-branch
git checkout master
git merge my-branch
git push origin master
```

Mentor Questions

how to get the question to display on the take page?

- ☑ ~~Without cookies will the site retain info we have entered when creating the quiz (looks like it stays in current db as long as server session is active? / db is not reset)~~
- ☑ ~~How to access quiz we've created when clicking take quiz~~
- ☑ ~~How to specify the quiz on the questions SELECT? (use ejs <% %> reference quiz id?)~~
- ☑ ~~how to specify the answers on the app.js answer select. (order by answers.id?)~~

To do list:

- ☑ ~~When clicking create quiz, redirects to create quiz~~
- ☐ When clicking random quiz, redirects to random quiz
- ☑ ~~Apply hyperlinks for public quizzes~~
- ☑ ~~Created quiz, when hitting submit, redirects to home page~~
- ☑ ~~When clicking take quiz, quiz should show either hard coded or created information~~
- ☑ ~~SCSS on create~~
- ☑ ~~SCSS on take~~
- ☑ ~~SCSS on results~~
- ☑ ~~SCSS on index~~
- ☑ ~~When we hit submit on take quiz, redirects to results~~
- ☐ Results page needs share button for url
- ☐ Once quiz is created, should be able to share url
- ☐

[redacted]

```
<!-- <div class="answer-results">
       <% for (let answer of attempt) { %>
         <span class="question-results"> Question: <%= answer.question %></span>
         <span class="input-answer"> Your Answer: <%= answer.answer %></span>
         <% if (answer.iscorrect===false) { %>
           <span class="false">WRONG</span>
           <br>
           <% } else { %>
             <span class="true">CORRECT</span>
           <br>
         <% } %>
       <% } %>
     </div> -->
```

- □ — Which project you chose and why
- □ — What each of you did, individually
- □ — Show us what you built (demo the app and show us the features)
- □ — What challenges you had and how you overcame them
- □ — What you learned about project workflow
- □ — What you learned and/or what you’ll do differently in final projects