Bruin Connect

UCLA-Based "Connections" App

Demo Available

https://roaring-cobbler-2b9645.netlify.app/

Our Three Distinct Features

Questions

Users can answer an ever-increasing list of questions, and our app will calculate and display other users with similar responses to you.

Swipe

When users are viewing other users with similar responses, they can either swipe left or right. A left swipe represents a pass, and a right swipe represents a desire to connect.

View Profile

If two users have "swiped right" (indicated that they want to connect with each other), then they can view each other's profiles and get their contact information to start a conversation.

Pages

Signup/Login	Login page, uses UCLA email to authenticate.
Profile	Users' page displaying their name, image, and a short description
Questions	Gives the user questions to answer in order to help find them the best matches
Swipes	Displays unswiped users with most similar question responses to you using the matching algorithm
Connections	Shows the user's matches, with the ability to search the matches

Login Page

Custom client-side form validation and server-side validation to allow only UCLA students.

Supabase — third party Identity Manager

Authentication using OAuth 2.0 PCKE flow with email and password

E-Mail

Postgres row-level security for complex authorization logic

Login

UCLA Email *

Password *

SUBMIT

Sign up
Reset password

Sign Up

UCLA Email *

Password *
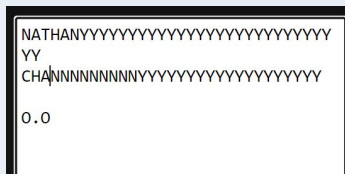
Re-enter Password *

SUBMIT

Login
Reset Password

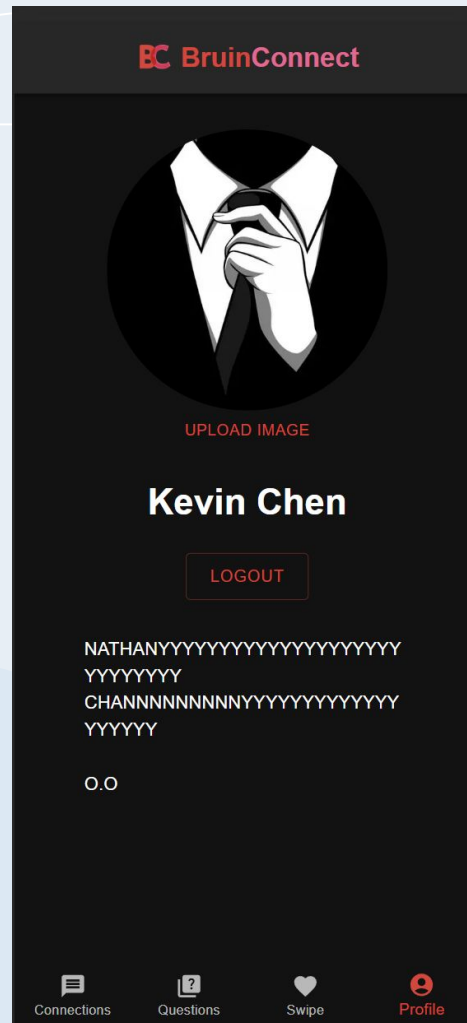# Profile Page

Displaying user's own profile page:

Profile Picture

Option to upload image that replaces current profile picture

Logout Button

Editable Description that is viewable by all of your matches

NATHANYYYYYYYYYYYYYYYYYYYYYY
YY
CHANNNNNNNNYYYYYYYYYYYYYYYYYYY

O.O

Tap to edit description

BruinConnect

UPLOAD IMAGE

Kevin Chen

LOGOUT

NATHANYYYYYYYYYYYYYYYYYYYYY
YYYYYYYY
CHANNNNNNNNYYYYYYYYYYYYYY
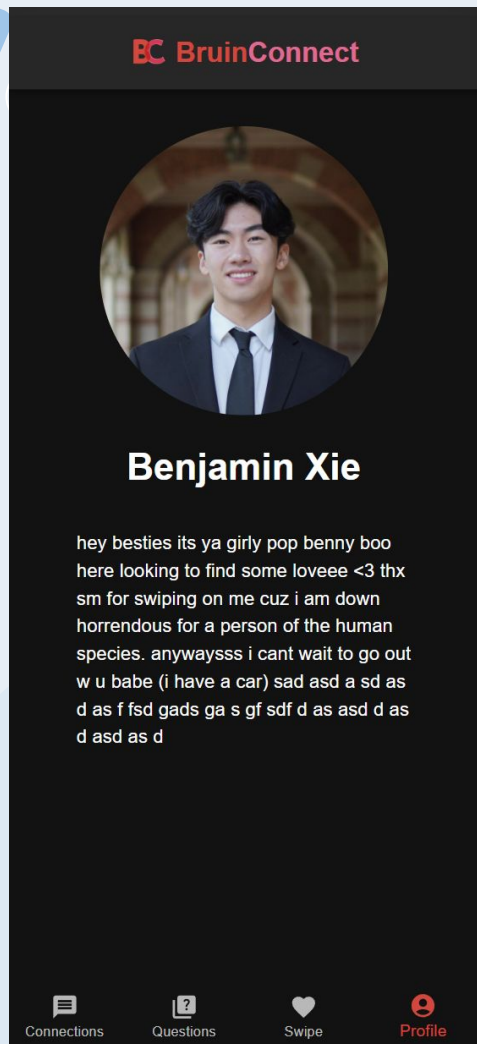YYYYYY

O.O

Connections   Questions   Swipe   Profile

Others' Profile Page

Edit description and Upload Photo features disabled for everyone else

Default message for no descriptions

BruinConnect

Benjamin Xie

hey besties its ya girly pop benny boo here looking to find some loveee <3 thx sm for swiping on me cuz i am down horrendous for a person of the human species. anywaysss i cant wait to go out w u babe (i have a car) sad asd a sd as d as f fsd gads ga s gf sdf d as asd d as d asd as d

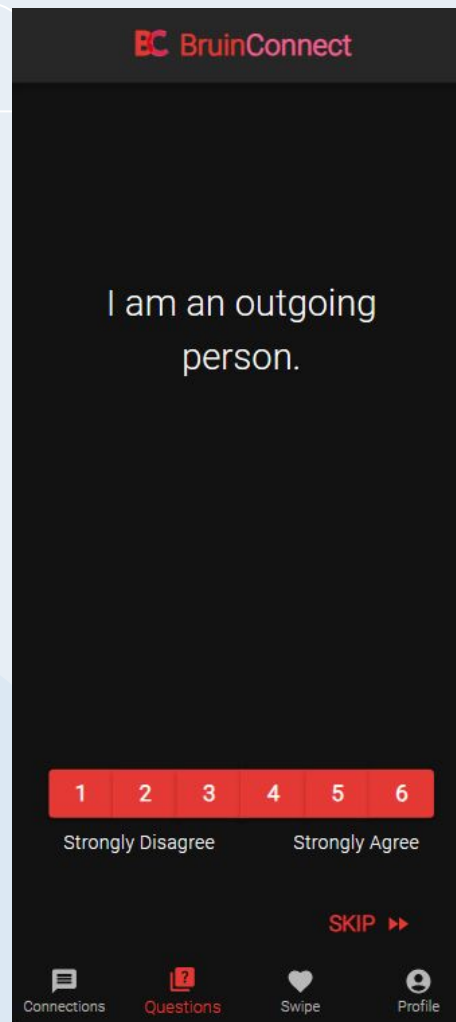Connections   Questions   Swipe   Profile

Question Page

Displays UCLA-related questions for users to answer.

Questions include but are not limited to topics like:

- UCLA Dining
- Hobbies
- Personality
- Favorite campus spots

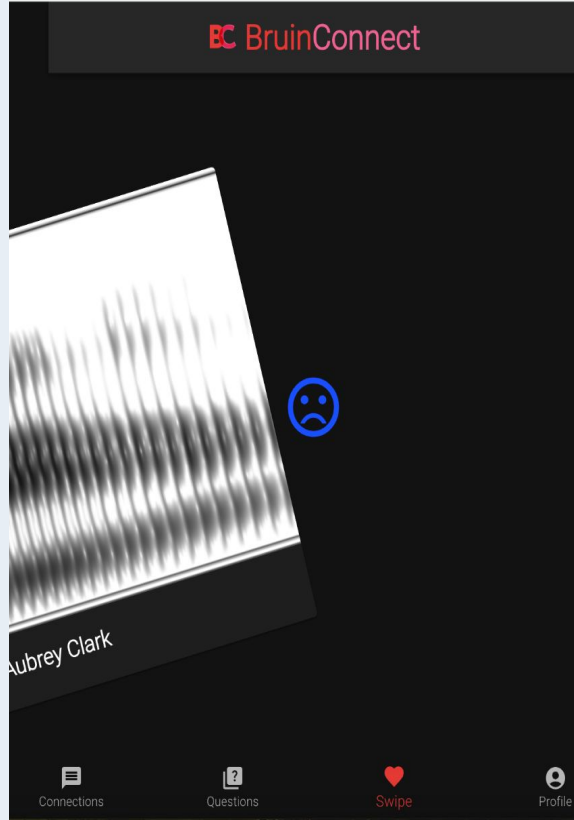Answers to questions are then used in a matching algorithm to determine users to display on swipe page
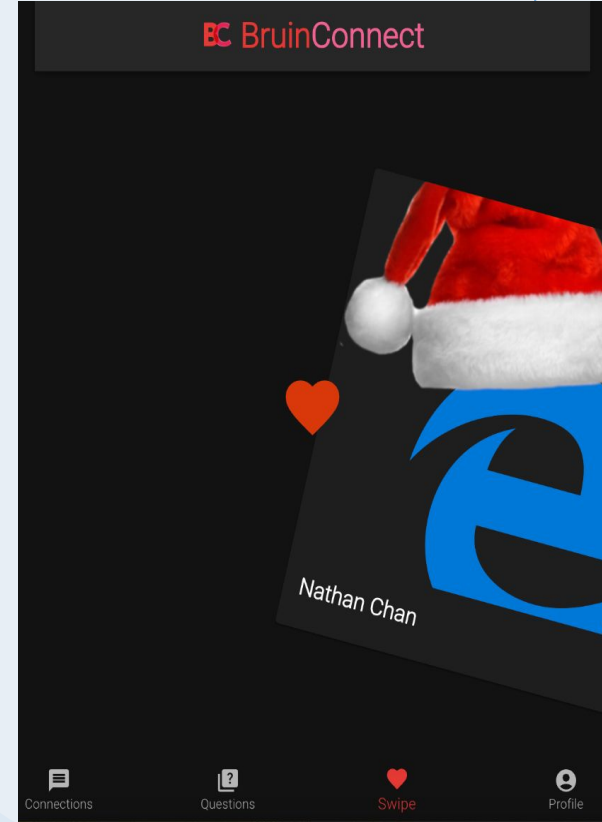
Swipes Page

Lists other users in order depending on their "compatibility," which is calculated from the cosine similarity of their response vectors.

The user can swipe right on the profile if they like them, and swipe left if they don't like them.

If two people both swipe right on each other, they both get added to each others' connections, which can be seen on the connection page.
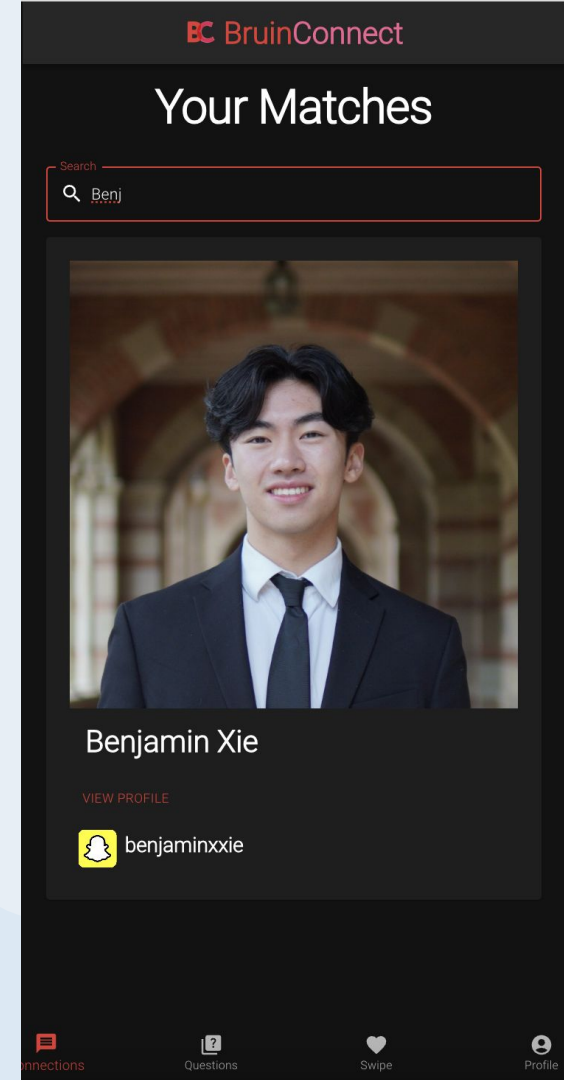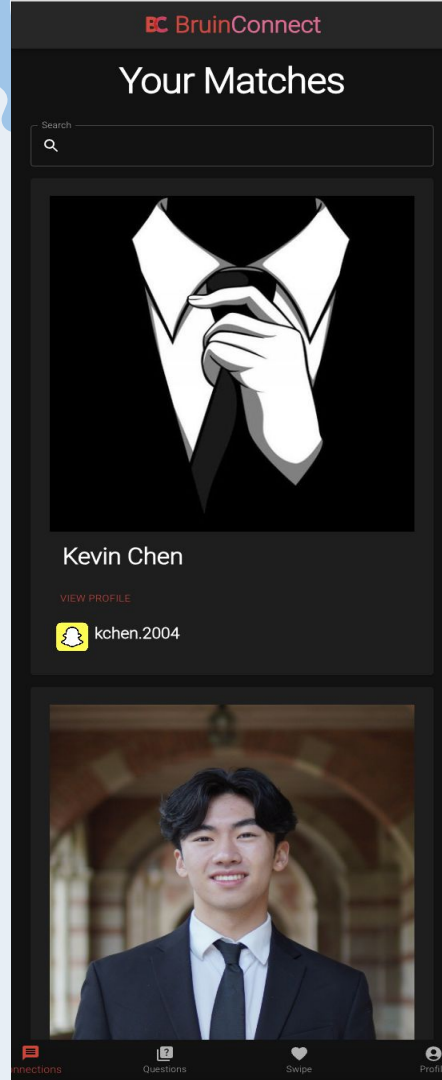

Swipe left

Swipe right

Connections Page

Displays users who swipe right on you and you swipe right on.

Each user displayed has:

- Profile Image
- Snapchat Username
- View Profile Redirect Button

The page also includes a filter/search bar that actively filters for both the match's name and Snapchat Username.

Challenges

Client vs. Server Rendering

Next.js uses React Server-Side Components (SSC) and React client-side components.

Since Next.js will render on both the client and server, nondeterministic variables have to be synced with React effectful functions.

Authentication Storage

Supabase provides separate libraries for client and server that we had to sync using custom authentication storage using cookies.

Next.js streams React SSCs, so they cannot write cookie headers. To get around this, we wrote custom middleware to do cookie write operations.

Hosting

When hosting, the server-side URL parameters were incorrect, giving the wrong port for HTTPS and not giving access to certain environment variables. Thus, the code needed to be extended to account for Netlify's built in environment variables while still being able to work locally.

Tech Stack

Languages

- HTML (JSX)
- CSS
- JS
- (Postgres) SQL

Front End

- React
- Material UI
- Emotion CSS

Back End

- Node.js
- Next.js
- Supabase - OAuth