

every time something is removed, put it back  
but in the mobile section.

Every time something is added, move it to  
the non-mobile section.

Have to set max-width: 801px

if your screen is 801px wide.

~~Screen has to be 799 px wide (or 800px)~~

max-width is 800px.

Screen is 800px wide

max-width at least 800px.

min-width at most 800px.

#-next {

} position: relative;

is necessary to prevent



Our rows

from not showing up

at the full width but instead  
showing up at less than (half, to be precise)

the actual screen width

The mobile styles should not get applied until 799px.  
Or lower, since both min-width and max-width  
are call-inclusive  
and we don't want them to overlap.

✓ What should the mobile cutoff be?  
I just changed it to 799px.

helpTheeDiv helpTheeDollarSign helpTheeScans  
card-ing-top helpTheeH3

To override React-Multi-Carousel, we have to use the  
important tag.

Currently the footer is only on the main page.  
Let's include it on the About Us page!

command ⌘ + shift + P or fn + F1

to open command palette

ext install vscode-kanban

Fn F1 Kanban: Open Board

→ No workspace found!

Settings:

settings.json:

Kanban: {  
  "openOnStartup": true  
}

Major League Hacking VS Code Extension!

option  $\text{Ctrl} + \text{Z}$  to toggle text wrap

(VSCode)

Step 1 npm install -g yo generator-code.

Step 2 yo code (It'll ask you a bunch of questions. Scroll with the arrow keys.)

Select New Extension (TypeScript)

Some extension name,  
some extension identifier,

leave blank!,

Yes, No, npm

\* Should I use  
typescript?

Then, cd some extension identifier  
code.

In extension.js, we can register new commands.

In package.json, we edit the keys:

activationEvents and contributed commands.

We can either do vscode.window.showInformationMessage or another VS Code API call, for example, vscode.window.showWarningMessage.

see user guide

How To Run? Fn + F5 to open a new  
Extension Development Host  
window.

Shift + Shift + P to get Command Palette;

In which we can do

Developer: Reload Window to get recent

The problem is not with the new login page.

goalbits.com/login?redirect=/goalbybit/

Did something change on pages/login.js?

## Goalby Debugging Part 2:

Login Page: Click

Choose an account: gladish.dean@gmail.com

login.js?3b69:69 "signed in successfully"  
login.js?3b69:70  
ax: { Ca: "|||", Es: ax { bt: "Dean", log there  
                  tR: "Gladish"  
                  } }  
we put a console  
This is the full  
data about the user's profile. FI: "https://lh5.  
                  K: "gladish.dean@gmail.com"  
                  S: "Dean Gladish"  
                  wR: "|||"}  
accessToken: "yA29  
googleId: "|||  
profileObj: { email: "gladish.dean@gmail.com"  
                  FamilyName: "Gladish"  
                  givenName: "Dean"  
                  googleId: "|||"  
                  imageUrl: "https://lh5.  
                  name: "Dean Gladish" }  
tokenId: "cyJ  
tokenIdObj:

tokenId: "cyJ  
tokenIdObj:

data apparently has accessToken and profileObj keys which are accessed directly by you  
What do these mean?

accessToken: {  
  access\_token: "ya29...  
  expires\_at: 1613881491872  
  expires\_in: 3599  
  first\_issued\_at: 1613877892872  
  id\_token: "eyJ...  
  idp\_id: "google"  
  login\_hint: "AJD  
  scope: "email profile openid"  
  https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email  
  session\_state:  
    extraQueryParams: { authuser: '0' }  
  token\_type: "Bearer" }

uc: identical to accessToken.

login.js?3669:90

{ email: "gladish.don@gmail.com" }

familyName: "Gladish"

givenName: "Don"

googleId: "111

actually imageUrl: "https://lh3.googleusercontent.com/Dean\_Gladish" data.profileObj.name: "Dean Gladish" — a subset of the original data object.

login.js?3669:97 signed up with google!

\*NON I AM ON localhost:3000/googlebiz.\*

index.js?3454:92 check out the next page!

{ approvedProfile: false }

authType: "email"

email: "gladish.don@gmail.com"

isCustomer: true

isAgent: false

name: "Dean Gladish"

profilePic: "https://c4.edn.nit" }

↳ data, profileObj  
↳ The data object passed by the onSuccess function.

↳ This occurs b/c we have a GoogleLogin — directly from API > onSuccess = response =>

↳ this.onSuccess(response) />

↳ react-google-login.

VSCODE - GitHub - project

TAB to autocomplete.

So we've got the

"Sign in with GitHub"  
button

1) Yuan Samuel into main from query-github.

2) aitik gupta from github-oauth

3) Shnill Shrestha from svelte-refact.

1) query-github has:

Sign in with GitHub Get started  
Sign in with GitHub

2) github-oauth has the same UI

out in the terminal (click on Building... !)

(!) Plugin svelte: `ext-vscod` is not defined.  
webviews/components/Sidebar.svelte

3) What does svelte-refact do?

Remember

gooley-frontend is the original,  
gooley-frontend-test, is my version.

① Replace the login page with  
its original code,  
and see if you can log in as  
gladish.lean@gmail.com.

★ ★ Several important things happen here:

asyn<sup>r</sup> response =>

```
var response = await axios.post(  
  process.env.API_URL + '/auth/login',  
  { user: { email: data.profileObj.email } },  
  { withCredentials: true })
```

Why are you storing this variable  
if you're not using it?

var reqBody = ...

```
var response2 = await axios.post(  
  process.env.API_URL + '/signup/saveUserToDatabase',  
  reqBody,  
  { withCredentials: true })
```

)  
Router.push('/profile#/')

end up in  
How do we ~~get to~~ index.js from here?

pages/profile#/index.js

on the asy<sup>r</sup> componentDidMount () {

```
try {  
  var response = await axios.get(  
    process.env.API_URL + '/user/getProfileData',  
    {  
      withCredentials: true  
    })  
  console.log(response.data.data)
```

Question: Where are we making requests, and what are the responses?

```
<GoogleLogin
  clientID='949...spd.apps.googleusercontent.com'
  buttonText='Login'
  onSuccess={response => this.onSuccess(response)}
  onFailure={response => console.log(response)}
  cookiePolicy='single-host-origin'
/>
```

Side Question: What is credential = firebase.auth.GoogleAuthProvider.credential(null, accessToken)?

In pages/login.js:

onSuccess function →

```
firebase
  .auth()
  .signInWithCredential(credential)
  .then(async response => {
```

① await axios.post(process.env.API\_URL + '/auth/login',
 { user: { email: data.profileObj.email },
 withCredentials: true })

② await axios.post(process.env.API\_URL + '/signup',
 { saveUserToDatabase },
 reqBody,
 { withCredentials: true })
 Router.push('/playbiz/')

③ await axios.get(process.env.API\_URL +  
    '/user/getProfileData',  
    {  
        withCredentials: true  
    })

④ await axios.get(process.env.API\_URL + '/getFilters',  
    { withCredentials: true })

When we do Router.push('/goalbybiz/'),  
what happens? (from pages/login.js)

(from pages/goalbybiz/index.js)

Router.push('/local?userId=' +  
          response.data.data.username)

What does this do? sends us to the URL at that username.

When we catch an error →

Router.push('/login?redirect=/goalbybiz/')

We can use GitHub to merge conflicts.

Link to the app should be on ~~qualex.com~~/~~qualibit.com~~.

For all kinds of experiences.

most tech companies have digital nomads working remotely.  
Through all their work, they have all kinds of access to  
different experiences - fast experience, in person, virtual,  
corporate.

beaches or mountains  
sauna

That's the future Miami —

San Diego for 3 months  
Qualex → live that lifestyle

Sunday night Jennifer Madden

There are arbitrage opportunities It happens

- git reset --hard [Commit ID]
- bundle update github-pages if you installed Bundler
- gem update github-pages. Otherwise

xcode-select --install

sudo gem install -n /usr/local/bin jekyll

from jekyllthemes.io/theme/agency-jekyll-theme  
git clone https://github.com/

y7Kim/agency-jekyll-theme.git

cd agency-jekyll-theme

jekyll serve

if your local machine is set up properly, then all you need to do is

In this particular case, we are simulating the `<Dropzone addMediaItem={this.addMediaItem}>` component.

How do we invoke this via our substitute with action?

`<MainDropzoneApp addMediaItem={this.addMediaItem}>`,

from `./ImageGallery.js` — change this filename, the main props

— which renders

`<Dropzone onDropOnOriginal={fnDrop}`

`accept={['image/*']}`

`addMediaItem={props.addMediaItem}/>`

In our case, we want

`<CoveringChanger`

// which naturally takes the props

`setCoverImg={this.setCoverImg}/>`

and

`<ProfileChanger setProfilePic={this.setProfilePic}/>`

Let's make some new files: `mainCoveringChanger` and `mainProfileChanger.js`!  
don't be shy!

What are we going to pass them?

`setCoverImg={this.setCoverImg} and`

`SetProfilePic={this.setProfilePic}.`

What does that mean?

It means that we're going to have to structure our internal component differently.

① Nafti Jones' phone number is 4.482+11.

Tonight Admin Dashboard: • show all fields ✓

- all reviews from submissions can't be saved if the customer can't be available in the Admin Dashboard.

Booking: • catch incorrect dates - don't accept dates in the past!

New Categories:

- I should be able to sign up as a user, and not get defaulted to Customer.

OTHER stuff that isn't mentioned on Trello:

- Admin should be able to delete users
- Admin should be able to unapprove users.

Customer → approvedProfile, ✓

b6, ✓ birthday ✓, category ✓,  
city ✓, completedProfile ✓, email ✓,  
name ✓, phone number ✓,  
customer ✓, date ✓, name ✓, phone number ✓,  
priceRange ✓, profilePic ✓, reviews ✓

[ ]

OBJ  
SocialMedia

"Facebook?"

STR

"Remember?" option?  
-for "if" -click in

STR

STR

deck: EmptyStreet

["First Name", "Last Name"]

comment

rate

reviewed

reviewEmail

starRating

number →

\* Sohan historically intended to send the review form here! Can it still be done?

react-redux npm i -D

redux

redux-devtools-extension

redux-logger

redux-thunk

Note.js

npx create-react-app my-app

Then

App.js

① import {connect}

mapDispatchToProps,

② connect() takes two arguments, mapStateToProps,

③ Add getData fn. Use json-generator.com to fill data.json.

index.js

④ Wrap the root component in a <Provider> so that we can access the store.

In order to mapStateToProps, you need Provider, which needs store.js.

or mapDispatchToProps

\* ⑤ Create store.js

Index.js ⑥ Import store.js and pass it to <Provider>

Alper Eduardo Silveira Santos - speaker

The store object itself has a very small API with only four methods:

① `store.dispatch(action)`

② `store.subscribe(listener)`

③ `store.getState()`

④ `replaceReducer(nextReducer)`

⑤ Changes are made with Pure Functions.

⑥ To store: Create a store with

`Redux.createStore()`

and pass in all reducers as arguments.

⑦ Don't mutate state, Copy It

⑧ Multiple Reducers

⑨ After a Dispatch, all reducers are called.

⑩ Immutable Data Structures

REdux Doc.

"The shape of the state is up to you: it can be a primitive, an array, an object, or even an Immutable.js data structure. The only important part is that you should not mutate the state object, but returning a new object if the state changes."

Q. How does immutable state help us?

⑪ `createStore()` has second arg: initial state.

It is cheap and easy to rehydrate old state.

→ We can time travel!

★ How do we connect React components to Redux in a more convenient way?  
React, redux, and react-redux! They are three separate modules on npm.

and the store makes it the new official source.

In Redux, the reducers are in a hierarchy which can have as many levels as needed.

★(Side note) Smart/dumb components.

aka container/presentational

• Smart components are in charge of the actions.  
Can pass action functions in via the props.

• no CSS styles      • no DOM      • arrange dumb components instead.

REACT-REDUX

How to connect

store <-----> views

? ✓

View layer binding.

If you're using React, this is react-redux. ★

3 concepts:

1. The Provider component

- wrapped around the component tree.
- So root component's children can hook up to the store using connect()

2. connect():

- the connect fn, provided by react-redux
- wraps a component so it can get state updates.
- Connect fn, specifies what parts of the state a component needs as properties.

Three Guiding Principles:

① "single source of truth"

② state is read-only: "The only way to mutate the state, is to emit an action, an object describing what happened."

③ "Actions" are dispatched to express an intent to change the state in the store.

### Admin Panel

- ① The artist data is from adminDashboard.js / 110. ✓  
(Can we access runtime variables as browsers of the website?)  
(If they are not easily logged?)

So, "Send review form" sends a review form to our email when we click "Send review form".

~~www.concise.com~~

Redux: a pattern that was inspired by Flux.

(Problems with Flux)

- ① The code for stores, statechange logic and the current state itself, are in the same object, preventing hot reloading.
- ② The state is rewritten with every action
- ③ No good places (extension points) for third party plug-ins to jump in.

So Redux adds:

• Action creators.

- Whenever you want to change the state of the application, you start off an action.
- Unlike Flux, action creators in Redux do not send the action to the dispatcher.

Instead, they return a formatted action object.

• The store.

- Takes care of holding on to the whole state tree.

• The reducers.

- When the store needs to know how an action changes the state, it asks the reducers.
- The root reducer takes charge and slices the state up based on the state objects keys.

• passes each slice of state to the reducer that knows how to handle it.

Each slice is copied - the state object isn't manipulated directly.

• all slices are combined into a new state object.

Reducers pass their copies back to the root reducer

root reducer

passes the copies together to form the updated state object

and then the root red. sends the new state object back to the state

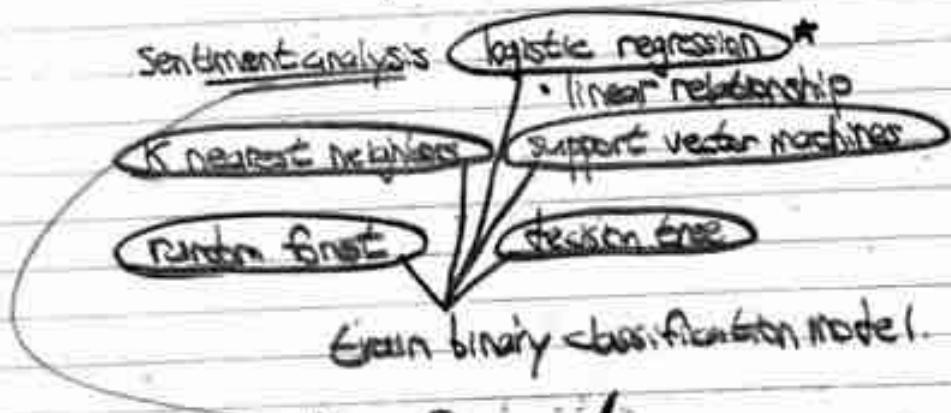
var response = await db  
collection('users')  
db('email')  
.update({ \$inc: { avgRating: 1 } },  
{ avgRating: avgRating })

Whenever we get all emails, we want to:

for each email

- for which an average star rating  
can be calculated

cosine similarity      Scikit with Python.



Naive Bayes ✓

Linear Regression

Support Vector Machines

Deep Learning

Confirm that the .csv is in fact being read properly,  
by comparing it to text directly entered in the  
terminal.

\*It may actually be working, but the 0s  
just indicate extremely small numbers.

- dataCollector
  - dataCollector
  - primaryUsage
  - privacy
  - privacy
  - improperAccess
  - improperAccess
  - personalInformation
    - personalInformation
    - usageInformation
    - usageInformation
  - DataUsage
    - DataUsage

means particular client -  
 I don't have to have the  
for the connection closer (basically)

Wherever I go,  
I find happiness!

A bunch of weird  
 stuff happens

and we see all things

INFINITE  
 LOOP

→ localhost:3000/goolybit →

Router.push('/login?redirect=/gooleybit/')

occurs in ~~both~~ areas: ~~these~~

pages/gooleybit/corporate.js — gooley-frontend

\* componentDidMount

pages/gooleybit/users.js — gooley-frontend

\* componentDidMount

pages/gooleybit/index.js — gooley-frontend

\* componentDidMount

★ SAME mediaItem.type is undefined error when  
 accessing mediaItem.type.includes('video'),  
except for pages/gooleybit/index.js

```
var response = await  
    .collection('users')  
    .doc(email)  
    .update({ name: 'John Doe' })
```

The pages in pages/login.js are what aggregate  
data.

We do cookies on the backend.

```
app.use(  
    session({
```

name: 'session',

secret: 'very secret',

saveUninitialized: false,

resave: true,

store: new FirebaseStore({

database: db

```
})
```

```
})
```

↑

four differences

This session middleware will assign an object to the Express  
request object named session (i.e. req.session) for each  
incoming request. The req.session object will store any  
data that we have assigned to it for any previous requests  
Made by the same client.

That means any middleware downstream of the session  
middleware will be able to access req.session to get and/or  
set client-specific data.

Earlier...

In our case, sessions will be represented via JavaScript objects  
living in the runtime of our server.

This means if our server restarts, our session state is wiped; it is  
just a JavaScript object in RAM (though it could be more

permanent if we set it up that way).

A session object will contain data about one particular client - something similar to sockets, except we don't have to have the connection open, and it can last after the connection closes (basically something we haven't done yet!).

$\Rightarrow$  the connection is a definite thing.

ant once the session object is made it doesn't have to be made again.

# Like Another World

In which

express-session handles all of the cookie reading and writing for us (convenient)

Focus (non-convenient)

- ① If a new request has no cookie, it makes a session, a new cookie, and updates the response to include information to set the cookie on the client.
- ② If a subsequent request has a cookie, it uses the embedded ID in that cookie to look up the server-side session object associated with that client. It's important to understand that nothing we add to `[req.session]` is ever sent back to the client. (Side note, some middleware about the cookie is made visible in `[req.session.cookie]`. If you want to inspect it at your leisure).

IF the admin login is successful, then set req.session.isLoggedIn = true. Initially, req.session.isLoggedIn = false. When the admin logs out, req.session.isLoggedIn = false.

- Buy on the website:
- When you edit a profile and add the "About Me", and when you format it so it includes paragraphs and new lines, and then hit Save, the formatting in the About Me does not show up ... so all the text is clumped together.

VIEW `localhost:3000/local?userId=carlos-suarez`

EDIT `localhost:3000/profile?true`

## The Forces That Lead To the Creation of Humanity:

The Piller text is properly spaced.

It comes from this.state.profileData.bio.

Whether or not response.data.userData.completedProfile is true or not, we still use

`profileData: response.userData`  
within `this.setState({})`.

So why not do it easier?

Within `pages/homepage.js/index.js` on apollo-frontend

In fact, we also set `loggedIn: true` in both cases,

⇒ it's coming straight off of the server.  
Where is the displayed text coming from?  
LINE 576 also off of `this.state.profileData.bio`

on `pages/local.js`

So, the profile data is also coming from  
`this.state.profileData.bio`,

which indicates that the data object is being  
read differently in each render, & how it's read

Dootky  
Questions for Sohan Sengar

that makes the difference.

So what is it about the placeholder attribute of Form Control or Form Group being rendered differently from the in-line block? This state.profileData.bios is rendered differently from

```
<Form.Group>
  <Form.Control>
    placeholder={this.state.profileData.bio}
```

```
</Form.Group> ? ✓
```

Kozaa  
{ this.state.profileData.bio.split("\n")  
.map(str => <p>{str}</p>); }  
=====

On the "Edit Profile" page, they should be able to edit the price filter values.

Where are these being rendered on the signup page?

How can we replicate this functionality?

<Form.Group controlId='priceRangePreference'>

```
</Form.Group>
```

Where is the data from the form extracted and submitted? ✓

<Form>

onSubmit={this.handleSubmit}

\*Sohan - what on Earth were you doing with the horizontal radios?  
Pages/dootky/biz/index.js

The first thing we do is extract e.target.elements, and what does this look like? checkboxes.

Searching for 'localPlayerId=' b/c this is where we are immediately directed there are only two locations in which this happens:

1. In componentDidMount when if this.props.query.true == ''

2. In submitEdits, which is definitely not what we're doing here.

\*Sohan, where does this.props.query come from?

Similarly, in this `subProfile` fn, where do we reference the `profilePic` object?

in the `subProfile` function,  
can write `this.changeProfilePic()`

With a componentDidMount, we definitely do perform the `this.setState({ profileData: response.data.pic })`

`this.state.profilePic` ultimately ends up in Ed: `changeProfilePic`.

Where is this variable ~~set~~? There is only one place:

In `setProfilePic`: `this.setState({ profilePic: [blob, file] })`

where is `setProfilePic` called? (`getProfilePic` is just a function which takes various parameters)

In this place ~~①~~ `ProfileChanger.setProfilePic = { this.setProfilePic }`

~~②~~ and ~~③~~ `<Form.Group controlId='exampleForm.Control1'>`  
`<Form.Label> Upload Profile Pic </Form.Label>`

~~④~~ When `this.state.completedProfile <ProfileChanger>`

`setProfilePic = [this.setProfilePic] </Form.Group>`

When `this.state.completedProfile` is true.

Similarly, where do we reference the `this.state...`

★. `priceRange` object? ~~and how does it get on state?~~ ★

```
iP (this.state.priceRange) {  
    data['priceRange'] = this.state.priceRange  
}
```

We are talking about the `data` object which is ~~posted~~ POSTed to `/user/updateProfile`

Ajax ~~and~~ /user/updateProfile,

★ Sohan, could you give me a rundown of what these do, respectively?

★ Why do we need to post the exact same query to both routes?

Of course, the key is not that the `data` object is being posted; the key is that the

`priceRange` attribute is being added to the `data` object.

Where is the code which for Ed's Profile being constructed

(how does the Edit Profile page construct and submit its data object?)

Remember: the render section is split up into two categories:

Incomplete Profile and  
"Tell us about yourself..."

Completed Profile  
"Edit Profile"

respectively.

In this section, there's a form called submitEdits.

Want: `if (form.priceRange.value) {  
 data['priceRange'] = form.priceRange.value  
}`

directly from the form

Then, naturally, when the /user/updateProfile POST API call is made, (the data object is included so) the .priceRange is included.

★ Note: Dot versus Bracket notation? ?

★ Note! /user/updateProfile is the only API call that we make in the submitEdits function.

★ So what does /user/corporateSignup do differently to initialize a new user for the first time?

The Time I worked With YAML

## Security

What do we want to do?

Use cookies to authenticate the admin panel.

Suppose the browser has a cookie, which server-side is associated with already being logged in.

Everything rendered is contingent on this.state.loggedIn

\* Can local state be changed through the browser?

componentDidMount: call this.loadAdminPanel,

Can we access the cookie on response? Yes

Nice process, env. API - URL call.

It's on response.data

Express session it does not persist across url  
instead of /admin/login

From the Frontend, make an API call to

/admin/loadDashboard with req.session.loggedIn=true,

Instead of doing req.session.cookie.login=false;

return res.status(200).json({ message: "Successfully logged in" })

on POST /admin/login,

\* We can't send data through GET.

So

different idea!

The cookie is not being accessed from the browser when we make an API call to admin/login.

It's not where we can.

## Session destroy ~~to be like~~

### Statement of the Problem

Issue ~~to destroy~~ cookie session

- ① ~~GET route has a persistent session ID.~~

~~Admin / but Dashboard & here, we are still receiving - Admin / but Dashboard & here, we are still receiving -~~

- ② ~~POST route generates a new session, ID and data~~

~~otherwise, it happens, every time we POST to~~

~~Admin / logout~~

~~The correct session id from the browser with this POST route.~~

### Solution

- ① Make a specific Express API route  
to GET the session.

```
const a = express()
```

```
a.use(session({
```

```
secret: '23456789456789456789456789456789' })
```

```
a.get('/', (req, res) => {  
    // req.session
```

```
})
```

- ② To access the same cookie data across different API routes:  
~~cookie-persistence~~ middleware  
is used to connect the cookie to the ~~different~~ client request objects with the help of different API routes.

- ③ // but npm modules

```
const express = require('express');
```

```
const uuid = require('uuid/v4')
```

```
const session = require('express-session')
```

```
const app = express();
```

1. ~~3 parameters~~ POST to ~~Dashboard~~

## Universal Unique ID

// configure session middleware

app.use(session({

genid: (req) => {

console.log('Inside the session middleware!')

\* return uuid() \*

},

secret: 'Keyboard cat',

resave: false,

saveUninitialized: true

})

same thing as before

// Homepage route at '/'

app.get('/', (req, res) => {

console.log('Called Homepage!')

console.log(req.sessionID)

res.send(`You hit home page! \n`)

// listen to server at port 3000

app.listen(3000, () => {

console.log('Listening on localhost:3000')

)

=====

FastText

• FastText

- 1. Build the model
- 2. Train / Test the model on new data
- 3. Invoke the script from the frontend website.

- fastText is an open-source library, developed by the Facebook AI Research lab.
- Its main focus is on achieving scalable solutions for the tasks of text classification and representation while processing large datasets quickly and accurately.
- fastText stacks impressively in both accuracy and training and testing times against previously published state-of-the-art models.

It achieves this computational efficiency and accuracy by employing 2 methods to address classification and training word representations of text.

### I. Hierarchical Softmax

- A Softmax function is often used as an activation function to output the probability of a given input to belong to K classes in multi-class classification problems.
- Hierarchical Softmax proves to be very efficient when:
  - There are a large # of categories and
  - There is a class imbalance present in the data.
  - Here, the classes are arranged in a tree distribution, instead of a flat, list-like structure.
  - Huffman coding tree
    - DFS: shorter trees represent more frequently occurring classes and longer trees for rarer, more infrequent classes.

Probability (that a given text belongs to a class) is explored via a DFS along the nodes across the different branches.

- For data where there are a huge number of classes, this will result in a highly reduced order of complexity, thereby speeding up the classification process significantly compared to traditional models.

## 2. Word n-grams

- BECAUSE using only a bag-of-words representation leaves out crucial sequential information.

**AT THE SAME TIME**

- Taking word order into account will end up being computationally expensive for large datasets.

SO AS A HAPPY MEDIUM

FastText incorporates a bag-of-n-grams representation along with word vectors to preserve some information about the surrounding words appearing near each word.

- This representation is very useful for classification applications, as the contextual meaning of a couple of different words strung together also results in a particular sentiment echoed by that piece of text.

## Install fastText

FORMAT: fastText expects the category first, with the prefix `--label--` before each category, and then the input text, like so,

```
--label-- positive I really enjoyed this
restaurant. Won't love to visit again.
```

CODE: this is NLP preprocessing techniques to:

- remove unwanted symbols
- remove unwanted punctuation
- convert text to lower case
- add the prefix '`--label--`' to each row in the category column.

## • AND THEN :

- save our datframe as GIF files  
→ train and test our model.
- To improve the performance of our model the word Ngrams parameter is set to 2.  
In other words, the model is being trained on bigrams instead of considering individual words.

## ★ TESTING ★

There are 2 methods to test our model, both of them being slightly different.

### The predict method:

- Used to predict the most likely label for a given input.
- CHOOSE an observation which belongs to the category "HQ" and see if the model correctly predicts the category of "HQ" with a probability of 95% or greater.

### The test method:

- To evaluate our classifier on the entire test dataset (15000 samples), which yields values for precision and recall.

# correct labels among the labels predicted by  $F_{pred}$   $\rightarrow$  K labels

# labels that successfully were predicted, among all the real labels.

★ if you don't specify the value of the parameter  $K$ , the model will by default predict only 1 class it thinks the given input question belongs to.

FastText is much, much faster than training neural networks  
on multi-dimensional word vectors, and it can also  
also achieve good performance on the test set.

ft-data-prep.py train.csv valid.csv

New files: ft-data-prep.py  
test.txt  
train.csv  
train.txt  
valid.csv

Need to import csv  
import fasttext  
these are generated by the model

# predicting on a single input  
print(model.predict(ds.iloc[2, 0]))

row 3 column 1 (question)  
(column 2 is the category)

.iloc [0] first row  
[1] second row  
[-1] last row

[:, 0] first column of data frame

[:, 1] second column of data

[:, -1] last column

[0:5] first five rows

[:, 0:2] first two columns with all rows

[0, 3, 6, 24], [0, 5, 6] 1st, 4th, 7th, 25th row +  
1st 6th 7th columns.

[0:5, 5:8] first 5 rows and 5th, 6th, 7th columns.

Task: Want: Principle, Sentence  
Query Segment  
Body Sentence

dataPrivacy, sentence text

- Need: to combine Query column with the Label column using a hyphen.

Need to re-export to CSV in Numbers

Once the data matches up  
try out different labels!

The train-test split should be 50:40

Qooley-Webapp: /admin/dashboard  
/admin/logout  
redesign start

Qooley-Frontend:

ERROS: login as Alex Joyce pages/qooleybiz/index.js line 277,  
which comes from componentDidMount.

★ TypeError: Cannot read property 'covering' of undefined  
• Looping still occurring when I'm trying to log in or  
response.data.data.covering FROM Alex Joyce.

```
var response = await axios.get(  
  process.env.API_URL + '/user/getProfileData',  
  { withCredentials: true } )
```

Alex Joyce maybe it's only happening for the magicians.

- Price Range has to show up on View Profile AND  
on Edit Profile.

not showing up on Database  
submit form is on line 1565.

Get sign up as well

Want:  
① Default cover image not just on ~~view~~  
~~not logged in user page but~~ Edit Profile page,  
~~but also on View Profile page~~  
rob@robReime.com

② External Links should show up on Edit Profile page.

How do I get the default cover images to show up on  
the View Profile page?  
using rob@mbferre.com as an example.

### External Links

this.state.profileData.externalLinks

★ Why are there a bunch of unused variables on this.state?

Common Errors: Cannot read property .externalLinks of null.

So it's a regular array located on this.state.profileData.  
externalLinks.

★ Don't we want the ability to remove tags?

pages/gosleybiez/index.js — gosley-frontend  
renders

if this.state.loggedIn  
if this.state.completedProfile

pages/local.js — when you're actually logged in.

Q "Why should you bark me?"

- 1) View Profile
- 2) Profile Page. Are we logged in? No.
- 3) Edit Profile

Over the user/updateProfile API route really need none of the middle text?

pages/gallery/index.js ← where we were putting things in the data object. This is in the submitEdit function.

AND  
pages/gallery/index.js is where we have added setstate invocations within componentDidMount,  
the component mounted ×1 → problem is not with setstate (...) .  
the component updated! ×5 → the rendered HTML keeps loading, and reloading, and reloading...

## WHAT DOES THIS MEAN? ✓

now importing { default as ImageGallery } from ./imageGallery.js instead of MainProposedApp.

The ImageGallery component is the culprit.

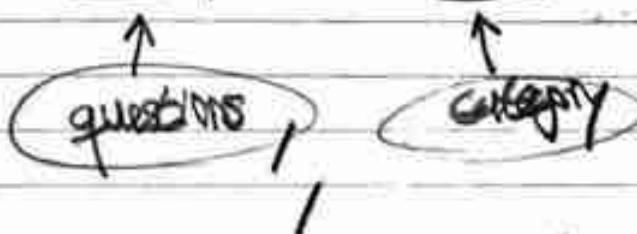
Which one?  
The last one, to which we are passing  
currImages={this.state.showSomeDeams}

which loads 52 times!

The original model look like this:

for StackOverflow  
Questions

We have Body and Y



the geomentry function, composition did Mount scope

the `createTree` function, the `createBstTree` function,

the `triangle` function, ... and this is where things start getting off track,

We also have `giving` function which gives us the promise? Why does the `fi` need to be synchronous? or, what is the lifecycle of a promise?

We also have `(async)` function `estimatePoseOnImage` which takes in an `imageElement` which awaits the `(Object)` what it gets and where is it from? and we also have a `getCamera()` function.

and we also have the function update() which checks the canPlayVideo variable, the clock, the isLagged variable, and furthermore executes the play.

and then abruptly ends.

Back in component) DeMount

the updateAtoms() function ~~modifies the position of the cube~~,  
and modifies the position of the update.

Back in crossroads business.

much in ComponentDidMount, the function `checkOverlap()`? What about it? It takes ~~10~~ arguments and instead checks collisions and removes trees. And we also have the `onComponentDidMount`.

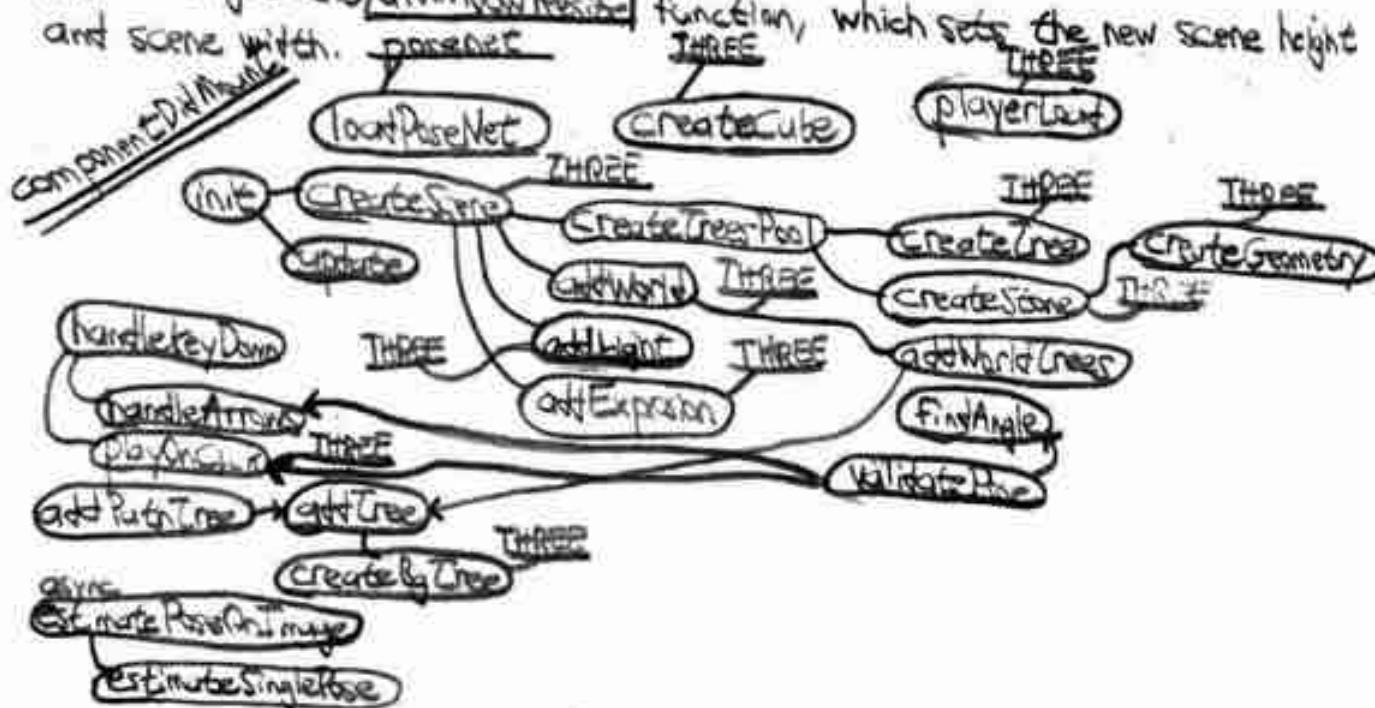
And we also have the Explorers which ~~detects~~ collisions and removes errors.

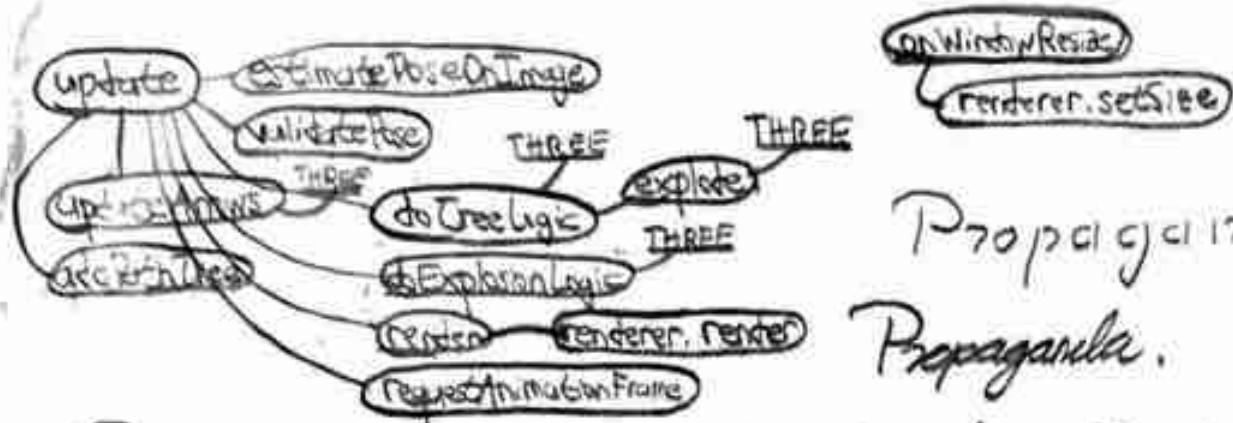
among other things, the `explode`, which similarly sets particles' position, particle geometry, and visibility.

And we've also got the `render()` function, which I assume is called elsewhere, and which calls the `renderer.render(scene, camera)` ~~function~~.

And we've got a What did this do?

And we've got the `onWindowResized` function, which sets the new scene height `THESE THREE`





Propaganda

Propaganda.

(git remote add upstream <https://github.com/MLH-Fellowship/PoseRunner.git>  
 git clone <https://github.com/gladish/PoseRunner.git>

Does the terminal actually not care about spelling when deciding to cd into a git repository? cd (poserunner)

poserunner  
Poserunner  
PoseRunner!

The async componentDidMount only runs when you click "Run w/o Poses!"

OR "Run with Poses!" → "RUN!"

that ⇒ the Game component only mounts in those 2 scenarios.

OR when you refresh the page while playing.

In the createGeometry fn, what are the values in

let texArray = [youtube, insta, unclctd, deadline];

?

Very nice examples in the very first section, 'this keyword', of 4. What is "this"?

Misconceptions of "this" 1 - this does not refer to the function itself

```
function whatIsThis() { /* the object logged is NOT the function object
    console.log(this);
} * whatIsThis, which demonstrates this does not refer
   * to itself. */
```

whatIsThis() → Window { ... } 2 - this is not a function's scope.

this is determined by the execution context, or how a function is called.

It has similarities to scope, but it is not possible for this to refer to a function's scope.

Execution context is the environment in which the JS code is executed; the value of this variables, objects, and functions JS code has access to at a particular time.

Fundamentally, scope is function-based while context is object-based. In other words, scope pertains to the variable access of a function when it is invoked and is unique to each invocation. Context is always the value of the this keyword which is a reference to the object that "owns" the currently executing code.

When you invoke or call a function, the function creates its own Execution Context.

The execution context is a record of information of code executing, such as where it's called (where the code is in the call stack), how the function was invoked, its scope chain, the value of this, and more!

\* One of the most controversial updates in ES2015 is the introduction of the class keyword and the related syntax that goes with it. JavaScript is not a class-based language by nature - objects inherit functionality via the [[Prototype]] chain.

Constructor ↠ to Class: How to create a parent-child relationship using ES2015.

ES2015 added "syntactic sugar" which hides some of the core fundamentals of the language.

(?) (1) refactoring the constructor function (2) creating the Prototype Chain to "link" the Parent and Child. COMPARE TO ES2015 subclass refactor with extends:

achieving the parent-child relationship between GithubUser (Parent) and GithubAdminUser (Child) with ES2015 syntax.

Topics: • class • class Subclass extends ParentClass • add a constructor method to a class • use the super() function in a constructor method of a Subclass.

Self-Quiz • What is the first argument to .call()? this. • After the first argument does .call() accept additional arguments? yes. • In the "First Example" from the code snippet, what is the value of this (it is passed to .call() as the first argument)? Window { ... } / global object. • Why is an object literal passed to .call() in the second example? to explicitly set the value of this. • What would happen if repeatIntro() was invoked without using .call()? It assumes the first argument is the name and uses the default value of this.

The `this` keyword has no static value. It's defined within the scope of a function.  
`this` can refer:

- The global object
- An instance of a constructor function
- An object a method is attached to
- A specific object

#0 Default value rule: By default, the "this" Keyword references the global object for the current runtime environment (Node.js, Web Browser, etc).

#1 `console.log(this)` → `Window { ... }` in Chrome  
Default binding rule:  

```
function whatIsThis() {
  console.log(this);
}
```

#2 `whatIsThis() → Window { ... }`

Implicit/METHOD Invocation Rule: When a method uses the `this` keyword, `this` references the object the method is called on. This can be identified by locating the ~~set~~ operator.  
`const pizza = {`  
 `toppings: ['Cheese', 'Ham'],`  
 `getToppings: function() {`  
 `return "Toppings: - " + this.toppings;`  
`}`  
`console.log(pizza.getToppings()); → Toppings: - Cheese, Ham`

#4 "new" Binding Rule: whenever you invoke a function with the `new` keyword, under the hood, the JavaScript interpreter will create a brand new object for you and call it `this`. \*If we aren't using `(call(), apply())` or `bind()`, our `this` will be our global object.  
`function Person(name) {`  
 `this.name = name;`  
 `console.log(this);`  
`const patrick = new Person("Patrick");`  
`console.log(patrick); → Person {name: "Patrick"}`  
In all the examples, we will **ALWAYS** use function declarations, not arrow functions.

Wd 9/14: Not the default internal prototype for our object  
add a field to say, `Person`, `way` papers from the internal prototype

constructor functions - prototype object  
constructor functions as arguments and returns an empty object

function newObj()

Wd 23: 71

On the Immersive Experience Signup Page,  
adds activityCategory, activityNotes, and activityImages.

### immersiveActivity

city, dateEnd, dateStart, displayName, fullOutInformation  
ImmersiveActivityDuration, nameThisActivity, state, street, zipCode

✓ NameThisActivity

✓ activityCategory

✓ activityImages

✓ fullOutInformation

✓ dateStart dateEnd

✓ immersiveActivityDuration

✓ activityNotes

✓ state city street zipCode or "Virtual"

call this in

call

^ -  
Local Data

email

Name

profilePic }

When a new experience is created, push to the existing  
activities collection on Firebase, instead of pushing to the  
existing user document by updating the user at the email address.

POST form object to

/user/updateProfile which  
updates var data = await db  
.collection('users')  
.doc(user.email)  
.update(newData)

API Route!

var data = await db  
.collection('activities')  
.add({

activityDuration: req.body.immersiveActivityDuration : req.body.activityDurations

activityRange: [req.body.dateStart, meetingLocation]

bookings: [req.body.dateEnd] name: req.body.nameThisActivity,

categories: req.body.activityCategories: req.body.activityNotes,

city: req.body.city

price

description: req.body.fullOutInformation

draft:

status

localData: localData,

req.body.street + "," + city + "," + state + zipCode or

Virtual

On pages/goolyby/corporate.js,

adds `MediaUpload` array to `this.state`, `this.addMediaItem` function, which is given to the `ImageGallery` component, which updates the Firebase storage tab at the provided email address, with the `addMediaItem` mediaItem is the activities key, creates `activityImages` is an array

```
[{"cancelled": false, "height": 828, "type": "image"}, {"uri": "file:///var/mobile...", "width": 829}]
```

### <p> Activity Images and Videos

```
<ImageGallery addMediaItem={this.addMediaItem}>
```

global scope

```
import { default as ImageGallery } from './imageGallery.js'
```

```
addMediaItem = async (blob, file) => {
  var mediatrons = this.state.mediaUpload
  mediatrons.push([blob, file])
  this.setState({ mediaUpload: mediatrons })
```

```
this.state = { mediaUpload: [] }
```

```
if (this.state.mediaUpload.length > 0) {
  await this.uploadPhotos()
}
```

```
uploadPhotos = async () => {
```

```
  var uploadTrons = this.state.mediaUpload
```

```
  var files = {}
```

```
  this.deletePhotos()
```

```
  for (let [blob, file] of uploadTrons) {
```

```
    var uploadRef = (storageRef) child (
```

```
      `${this.state.profileData.email}/profileShowcase,
```

```
      `/ ${file.name}`
```

```
)
```

```
  files[file.name] = file
```

```
  await uploadRef.put(blob)
```

import { as Firebase  
from 'firebase'  
import 'firebase/storage'

storage tab on

Firebase

global scope

```
const storageRef = firebase
```

```
storage().ref()
```

function which it inherits. The addMedia function captures the Media on this.state and tries to upload to Firebase storage at the provided email address. Provides Firebase Credentials.

Creates Email address returned by server.

var fileRef = firebase

.storage()

.ref(`\${this.state.profileData.email}/profileShowcase`)

fileRef.putAll().then((snapshot) => {

var images = snapshot.items

var showcaseUrls = {}

var counter = 0

for (var imageRef of images) {

var imageUrl = await imageRef.getDownloadURL()

var file = files[imageRef.name]

showcaseUrls[counter] = {}

showcaseUrls[counter]['url'] = imageUrl

showcaseUrls[counter]['type'] = file.type

counter += 1

}

var res = await axios.post(`

process.env.API\_URL + '/user/updateProfile',

{ showcase: showcaseUrls },

{ withCredentials: true }

)

Console.log

firebase.initializeApp(firebaseConfig)

on the backend, creates /user/updateImmersiveExperience  
POST route which updates the activities collection on Firebase.

includes email from req.session as the response user, which to update the Firebase  
Signup Flow Storage tab ~~at email/immersiveShowcase~~

ImmersiveExperienceShowcase/file.name!

fetching URLs of Immersive Activity and displaying them  
as React elements in React Carousel!

Activity Description

Start Date / End Date

Duration

Notes

Styles Address includes requestUser function for Report  
Booking Form!

fetches images using the email addresses from this prop. contact query  
displays activity title and activity

Traditional functions keep variables in scope!

① How would you categorize your activity?

- Arts & Entertainment
- Food & Drink
- Professional Training
- Health & Wellness
- Educational
- History & Culture
- Volunteer Experiences
- Outdoor Activities

TODD Fix "sometimes" rendering

immersive experience image!

- What does this state look  
like at different points?

turn it into a dropdown menu!

(localhost:3001/signup → localhost:3001/goodbyebiz  
→ localhost:3001/goodbyebiz/corporate)

<Form.Control name='activityCategory' type='

<div class="form-group">

<label for="usr">Name:</label>

<input type="text" class="form-control" id="usr">

</div>

<div class="form-group">

<label for="sel1">Select list:</label>

<select class="form-control" id="sel1">

- profile picture links to the expert profile
- never multi-select with + button

<Form.Select aria-label="Default select example">

<option>Open this select menu</option>

<option value="1">One </option>

<option value="2">

</Form.Select> Turn

onSelect={e => this.setState({immissibilityduration: e})}

and as={ActivityDurationToggle}

and as={ActivityDurationArrow}

and smthng="15 minutes" into categories

<Dropdown onselect={e => this.setState({concerningcategory: e})}>

<div class="dropdown-content-menu">

<Dropdown.Toggle as={ActivityCategoryToggle}

this state immisibilitycategory

this state immisibilitycategory : 'Activity Category'

</Dropdown.Toggle>

</div>

<Dropdown.Menu as={ActivityCategoryMenu}>

<Dropdown.Item smthng="Arts & Entertainment">

Arts & Entertainment

</Dropdown.Item>

<Dropdown.Divider> \* </Dropdown.Divider>

<Dropdown.Item> cancel </Dropdown.Item>.

</Dropdown.Menu>

</Dropdown>

Virtual Only / In-person should be a toggle button to be  
singularly concise; if the toggle button is checked as "Virtual",  
then a meeting location need not be required.

"The maximum number of people in the group" textbox

Mon Sep 13 2021

Want: to show Experience cards, one per experience, for each artist!

localhost:3000/local?userId=max-shanker

• filter by artist email!

See how the POST /local/getLocalProfile route connects userIdName ID into email by referencing off the database!

Returns email from the API route

response.data.data.email

this.state.impressiveExperiences.map((el) =>  
[this.state.email])

Displays unapproved experts!

pages/local.js

if (this.state.profileData.approvedProfile == null)

this.state.impressiveExperiences.map((el) => [this.state.email])

Display each image from the same impressive experience

Creates /unreviewedEmail API route, to connect/activity with:

fetches artist data on /activity page

Displays expert name, category, profile picture relevant to expert page

On artist page /local

, creates social media icons even if there are no reviews.

Creates experience images carousel!

## Experience Feature



- Plus button

- The button should be present so that we can add new experiences.

- Make the expert experiences images smaller thinner!

- Why is it taking up the whole width of the page?  
div class="name = "expertCarousel"

- justify-content: center;

- alignItems: center;

- padding-top: 20;

- padding-bottom: 10;

- padding-right: 10%;

- padding-left: 10%;

② What can cause data to be loaded in a different order in React... elements on this.state not being there some of the time?

such as this state immutability ExperienceImmutability

email12308902@gmail.com

Why can't we change the API URL during runtime?

```
<Carousel slidesPerPage={2} arrows={true} centered infinite>
  {this.state.showcases}
</Carousel>
```

this.state.showcaseItems []

```
setShowcaseItems(url) [
```

```
  this.setState([showcaseItems: [...this.state.showcaseItems, url]]);
```

on page/activity is,

to keep runtime flowing in order, puts showcase construction and state setting within the await firebase.storage().ref... call.  
makes the imageRef getDownloadURL asynchronous to allow setShowcaseItems calls to be asynchronous

... promise response image you ...  
Instead of being rendered when the arrow buttons are clicked for the

on /goolephiz/corporate, removes background color and header color to match Svelte Project Page.  
Removes cover image and adjusts height

first time, the value of this.state.showcase is allowed to be completely populated by the time of rendering.

*Only after the promise is resolved, the state has not been filled yet.*

In both cases the immersive experience image gallery gets updated continuously; without a promise, render() happens sooner than this.state.showcase is populated.

- The immersive experience image card for email12308902@email.com

on localhost:3000

is not rendering the sample image of the about immersive experience being displayed.

Creating HTML as in

<Card1.html

src = {immersiveExperienceFirstId ?

immersiveExperienceFirstId :

'http://image.unsplash.com/photos-15....'

}

Within update(): -

state.immersiveExperienceFirstId should be available when render calls for it.

This is one of the traits of async functions — their return values are guaranteed to be converted to promises.

An async function is a function that knows how to expect the possibility of the await keyword being used to invoke asynchronous code.

learn.fullstackacademy.com password is ZueXVeZ!KunxVN7

username is gladish.lean@gmail.com

{this.state.immersiveExperimentItems})

converts brainhub/smt-carousel into a carousel with

slidesPerPage received as the number of images in this.state.

immersiveExperimentItems, as a carousel without arrows which is not centered, not infinite, and not dragable.

- ✓ Bubblesort compares adjacent elements at the first level, iterating completely through a list until it makes one full pass without any swaps.
- ✓ Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hand. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.
- ✓ For-in versus For-of loop on properties of an array: iterating through non-array objects.
- ✓ How to write test specs, such as checking whether `map` is called twice
- ✓ Sorting and concatenating objects into a single string.
- ✓ Generating a grid.
  - What happens in the nested loop of a sort function?
  - The distinction between variables and properties in the context of the `createCalculator` function
  - Underlining the difference between `collection.hasOwnProperty(key)` and `collection[key]`
  - arrays are non-primitive; what does that mean?
  - When are parameters turned into arguments?
  - When the function is called.
- ✓ ESLint: What do the comments mean, and what kind of errors do they prevent?
- ✓ Default behavior of `push()`
- ✓ Demonic property name requires bracket notation.
- ✓ Dot bracket notation
- ✓ How to add to Array.prototype
- ✓ Template literals to combine line breaks in long return statement
- ✓ String interpolation in PostgreSQL with Template literals
- ✓ GitHub Keychain on macOS
- ✓ Order of operations within `Array.prototype.reduce`
- ✓ Flatten array of various dimensions within elements
- ✓ Methods inside/outside the constructor

- ✓ Syntax from the TestFirstReview solutions
- ✓ Deep copy and `Array.prototype.slice()` index rules
- ✓ Features of ES6
- ✓ Difference between `(num++)` and `(num+1)`?
- ✓ Bringing functions into the global scope.
- ✓ ~~Define~~ type of `...args` to a Function
- ✓ Check how many times a function has been called, from the global scope
- ✓ Guessing Game Sharing and Questions
- ✓ Refactoring if statements and a question about lookup maps
- ✓ TestFirstReview - rotator
- ✓ LC 42 Trapping Rain Water
  - [www.githubusercontent.com/glaelkel/LeetCode-Problems/](https://www.githubusercontent.com/glaelkel/LeetCode-Problems/)
- ✓ Approximating Coronavirus Cases in 2019
- ✓ Shuffle a Given Array using Fisher-Yates Shuffle Algorithm Interleave a String of Two Other Strings, Generate All Permutations of a Given String
- ✓ Burst Balloons
- ✓ LeetCode 9. Decode Ways
- ✓ LeetCode 786. Ranking Points
- ✓ LeetCode 76. Minimum Window Substring
  - [leetcode.com/submissions/detail/424529130/](https://leetcode.com/submissions/detail/424529130/)
- ✓ LeetCode 68 Text Justification
- ✓ Chegg first 10 Questions <sup>② ③</sup> last 10 questions
- ✓ <sup>④</sup> next 10 questions <sup>What is a CORS? → Since Next.js supports "cors"</sup>
- ✓ next 10 questions → what are the deployment services...
- ✓ How hard would it be to get a ...
- ✓ Best database-hosting service for an e-commerce website AWS

Gooley Login ~~aaaaabc@zzzzz~~ email.com  
password

- gooley-frontend-test-2.herokuapp.com/login
- GET https://gooley-webapp-test-2.herokuapp.com/user/isLoggedIn  
400 (Bad Request)
- • GET https://gooley-webapp-test-2.herokuapp.com/user/getProfileData 400 (Bad Request)
- infinite redirect loop error message: Error: Request failed with status code 400
- GET https://gooley-webapp-test-2.herokuapp.com/logout  
400 (Bad Request)
- (Uncaught (in promise) Error: Request failed with status code 400)

this is where the gooley-frontend-test-2.herokuapp.com server reaches its first error message!

```
localhost:3000/login      if (!isAuthenticated) {  
    checking for log in      res.status(401)  
    user undefined           }  
    false                    }  
not logged in : (
```

★ By adding return res.status(204).end();  
the Error [ERR\_HTTP\_HEADERS\_SENT]: Cannot set headers  
after they are sent to the client error, which actually means  
you have already sent response to the client after a request  
is made, but you're trying to send response again back to  
the client, can be resolved.

will return response and not run the rest of the code after  
just once so that we don't call res.end() more than once  
which will throw an error.

this is only necessary for shortcircuiting the function.  
we would not recommend to return the [end()] call  
itself, but rather after it.

the user is defined for `/user/isLoggedIn`,  
`/auth/setLogin`,  
and then getting profile data  
undefined  
not logged in : (

`GET /user/getProfileData`  
which means that the user is not on req.session.user.  
`GET /user/getProfileData` is called within  
pages/local.js and pages/gooleybiz/index.js.

/auth/setLogin takes the user from req.body.user  
and sets req.session.user = { email: user@mail };  
is this session formation not successful?  
on localhost:3000/gooleybiz?true  
is it possible (that setUser; on the next line after  
res.status(201).end() within our app use middleware  
prevents the execution of our app use (session) name: 'session'...  
middleware which is responsible for loading the session?

the server is, in essence, a combination of middleware  
which consists of segmented series of checkpoints which  
behave as functions which are executed when getuser  
is reached.

login works on localhost and not on the test server.  
On localhost:3000/gooleybiz?true, which denotes the  
Edit Profile page

(without this parameter) redirects to localhost:3000/local?  
user\_id=dearayaulish26  
this page

async componentDidMount() {

- Move the Immersive Experience editing feature to the Edit Profile page

localhost:3001/local?userId=clearaylealist26

```
let experiencesCollection = await axios.get(`process.env.API_URL + '/experience/search/all', { withCredentials: true, })
```

let immersiveActivities = experiencesCollection.data.experiences

let immersiveExperienceImageUrls = []

for (let i = 0; i < immersiveActivities.length; i++) {

    immersiveExperienceImageUrls[i][immersiveActivities[i].email] = []

    await firebase.storage().ref(`\$[immersiveActivities[i].email]/immersiveExperienceThumbnail`).listAll().then(function (res) {

        res.items.forEach(function (imageRef) {

            imageRef.getDownloadURL().then(function (url) {

                immersiveExperienceImageUrls[i][immersiveActivities[i][email]] = [...immersiveExperienceImageUrls[i][immersiveActivities[i][email]], url]

            })

        })

    })

}

this.setState({ immersiveExperienceImageUrls, })

let immersiveExperiences = []

If (this.state.impressiveExperienceImageUrls[this.state.email]) {

    this.setState({ immersiveExperiences })

    handle

only the experience... → over immersive experience image carousel!

Some resource load requests were throttled while the tab was in background, and no request was sent from the queue in the last 1 minute. This means previously requested in-flight requests haven't received any response from servers. See <https://www.chromestatus.com/feature/5523160148197776> for more details

```
submitForm = async e => {
    await axios.post(`process.env.API_URL + '/auth/setLogin'`)
    app.post('/auth/setLogin', (req, res) => {
        console.log(req.body)
        var user = req.body.user
        console.log(user)
        req.session.user = { email: user.email }
        res.status(200).json({ success: true })
    })
    app.get('/user/getProfileData', async (req, res) => {
        console.log('getting profile data')
        var user = req.session.user
        console.log(user)
    })
}
```

- ✓ • Separate First Name and Last Name true on /signup and /goolebiz
- ✓ • Det the street number as part of the address from the Google Maps API
- ✓ • Center display on experience page
- ✓ • Put the experiences in the column panel
- ✓ • On Edit Profile page: Show existing experience card; everything that you will start filling the Edit Profile page from left to right
  - experience cards should show up on the Profile Page
  - The Plus button should show up to the right of the last experience card
- ✓ • If there are experiences linked to this profile, fetch those experiences as cards along the line of the main page cards. The same card from the main page should show

upon the expert logged-in profile.

- because we want them to be able to edit their experience as well. Right now there's no way to edit the experiences.
  - ✓ • display the experience cards on the expert profiles, so they can edit them.
  - have a plus button so they can create any number of new experiences.

On pages/goolebig/index.js, which is the signup page, last time we removed display:none

~~also it back—not reusername because displayName  
is different from First Name and Last Name.~~

adds back Upload Images and Videos because these are for the student/expert who can create multiple or impressive experiences each of which also carries its own URL and a Video file to go with it.

seen Upload Images and Videos feature specific to each experience, on the next page goobybit/corporate. changes displayNames from textArea to input.

- <Form.Group> refactor: displayName → displayFirstRow  
    <Form.Label> DisplayName </Form.Label> displayLastRow  
    <Form.Control>

Required = [false]

`name = 'displayname'`

`as='input'`

`rows="1"`

classmate

placeholder="Display Name"

→ Signup Page format

with name = displayFirstName

display Last Name

8

```
placeholder={this.state.profileData.displayName}
```

```
• if (form.displayName && form.displayName.value) {  
    data["displayName"] = form.displayName.value
```

`/user/updateProfile` - Firebase does not have any restrictions on the names of the data fields which are submitted to the API route!

On pages/activity.js

font-size of titles is 1.7rem, font-weight is 500,  
text-align is center and spaces are preserved between  
titles. Height of the div enclosing the Request Booking  
div is 'auto' so that the Request Booking div does  
not overlap with the activity description divs.

pages/index.js

```
  &#123; this.state.carouselDisplay['Immersive Experiences']  
    && this.state.carouselDisplay['Immersive Experiences'] }  
this.state.carouselDisplay is mentioned in functions sortByPrice,  
priceSelect, clearFilter, refresh, and is instantiated in the  
declaration of this.state.
```

```
refresh: this.setState({ carouselDisplay: displayRows,  
var displayRows = {}  
displayRows['Immersive Experiences'] = []  
displayRows['Immersive Experiences'].push(  
  <h1 className='categoryTitle'>  
    { 'Immersive Experiences'}  
  </h1>  
)
```

carousel  
carousel.push <a href={`/activity/`...  
which requires email address of the host,  
immersive activity name,  
handleButtonClick.

impressiveActivityListWithInformation

... and src={impressiveExperienceFirstUrl}

let impressiveExperienceFirstUrl = this.state.impressiveExperience  
(imaginelets [impressiveActivities[i].email][0])

addImmersiveExperienceImageCarouselToTheEditProfilePage. Changes background color and border of  
react-carousel-item to 'none', and only handles impressive  
experiences for the current user.

`import { withRouter } from "next/router"`  
`export default withRouter(Activity)`  
changes route to `localhost:3000/activity?email=aaaaab@1.40mail.com`  
this prop. router.query takes time to load, & experience = the 1.20 second  
so don't put it at the beginning of (gated) componentDidMount!  
it can be put at the end of componentDidMount.

- ✓ • put some space between the cards
- ✓ • To be able to see the whole title, make the font slightly smaller so it fits in two lines; Should automatically go to the next line if it doesn't fit in one line! on the View Profile page
- ✓ • The plus button always goes after the last experience
- ✓ • make Immersive Experience title larger and give it space.  
• On geolevibz? true, the edit profile. When you click a card, the same geolevibz/corporate page should show up with the information filled in for editing.  
People are going to make typos!
- Just use the local names for the display names for the profiles, we should display the name of the experience for the immersive experience; get rid of the display name because it's repetitive and they're going to be confused.
- Immersive Experience Should be deletable
- Experiences should be deleted only when the artist is approved: Matches the pattern that only approved artists are returned from the server.

`localhost:3000/local?userId = bangladesh26`

It takes a while to set variable values on this state, so use the `immersiveExperienceImage` variable itself instead of the state `immersiveExperienceImage`.

Why is the URL slug putting an = sign in front of each query key? Because there are key: value pairs; this prop. router.query is an object.

`aaaaab@1.40mail.com password`

`<br />`

`<h1 style={{text-align: 'center'}}>Immersive Experience </h1>`  
`<br />`

Object Keys (this.props.router.query)

so the route is this.props.router.query

{ email: 'aaacabc@email.com',

experience: "The best name you've ever had",  
edit: 'true' }

destructive:

let { email, experience, edit } = this.props.router.query  
const setShowcaseItem = this.setShowcaseItem

fetches images at email for firebase immersive Experience Showcase  
storage bucket and experiences data filtered from  
/experiences/search/all.

- The required format for Form.Control value property is "yyyy-MM-dd".
- Sets the date and adds initial value property on the JSX on Form.Control.
- Sets the duration, transfers it from immersiveActivity[0] fetched from /experiences/search/all to this.state.immersiveActivity.duration from within componentDidMount.

||| within JSX allows us to do short circuit evaluation

to execute the code even if the this.state.immersiveActivity[0] Category prerequisite for .category doesn't yet exist!

- Run this.deletePhotos; button onClick = {e => console.log}  
requires e to be stated, parameters are based on  
this.props.router.query email and before that, edit == 'true'.  
makes the button conditionally on edit == 'true'  
styling Delete Photos button based on Submit button  
input element.

on /googlebiz/corporate/edit mode,  
Populates Form elements sent from e.target.elements,  
the react-bootstrap library.  
cleans up console logs.

The Immersive Experiences are pushed to the carousel on line 426 of pages/index.js : if (impressiveActivities[i])

line 312, this.state.allKittiesWithExperts

validEmail = cleangmail28 has cosaamail29109@email.com

approvedProfile = true so city button Boise City, Idaho  
should show up as a city button.

The variables currentKitties and this.state.allKittiesWithExperts  
are arrays which contain Boise City, Idaho.

\* this.state.allKittiesWithExperts.includes(city.split(' ') [0])  
is not true when city is 'Boise City, Idaho' so the list of  
city buttons is not containing of Boise City, Idaho.

.cells second || or required condition:

(( this.state.allKittiesWithExperts.includes(city)  
to the pushing of city buttons!

.pushes to validEmail based on allKitties array which  
contains all approved users and not unapproved users,  
to conditionally push to carousel if validEmail.includes  
(impressiveActivities[i].email) .

Walkthrough of Immersive Experiences

On the main page, add some space between the price filter  
and comedy ; put a trick underneath the price filter regardless  
of what the category is

Classify the use of additional pictures on signup ; upload  
pictures and images of yourself. Make it an optional  
field.

When submitting an experience, display a pop-up message  
or even a new page...

If it's immersive experiences, then only the experience shows  
up and not the expert

Fix Immersive Experiences View mode to follow along with  
Edit mode formatting

Full justify the experience description and experience notes,  
because there's going to be more information for those, if it

- Full justify if it takes up the whole line or more!
- Test new experience creation and existing experience edits locally.
- turn form, e.g. target.clients, into a reusable variable from within sweet.
- On /goolebyr/ true sets a href to target = "- blank" to open modal, including e.preventDefault() to prevent redircting to /local/world = + this state.profiledata.username.
- on /goolebyr/corporate, change value to placeholder to prepopulate form fields. on /goolebyr/corporate, wraps uploadPhotos in try/catch block to prevent file type validation error, checks for the existence of file. Someone could handle attachment formats from page/local.js experience controller

- ✓ What if we put the title of the activity at the bottom and let it have two lines?

```

gooleby.com/index.js
await db.collection('experiences')
  .doc(`${req.body.email}+${experienceName}`)
  .delete()
app.delete? const app = express()

```

app.delete(path, callback [, callback ...]) similar to  
 app.post(path, callback [, callback ...]) so structure  
 Request body as seen on the POST  
 app.post('/activity/submittActivity',  
 look at req.session.user)

```

app.post('/experience/DeleteExperience', async (req, res) => {
  var user = req.session.user
  if (user.email === req.body.ImpressiveExperienceEmail)
    /* Make a delete request to the database
    * collection for experiences. */

```

## Web Developer position at RENCI

complete in a relatively short amount of time

demonstrate fundamental web development skills and knowledge  
required for this position.

decompose, develop the application completely within this repository  
user interface design, user experience, responsiveness, error-handling,  
code hygiene, documentation, and git usage.

use npx create-react-app. To contribute, other developers should  
have a good code editor like VSCode, iterm, and be familiar  
with installing Git on macOS via Xcode Command Line Tools.

The great thing about Create React App is that it performs npm  
install dependency. We can host on localhost! after heroku auth:login,  
clone heroku create, and put it in root folder.

add question addition problem, input box, refactors functional component to  
class component, includes mouseLeave function with this binding,  
#->, -->, /->%2F, .->, space->%20, +->%2B, =->%3D, I->%5B, U

retrieved by encodeURIComponent('expression')). add planning and documentation.

instead of using event listeners and the DOM, integrates axios with  
method chaining within which the axios GET request is made for the  
axios slice, which does error catching and handling to submit and  
return messages from the api/mathjs.org API, instead of modifying  
DOM elements when events' listeners are triggered on them, uses axios.get  
method chaining to handle server response in a .then/.catch/.finally block  
which works with the promised response, and by the way the user and  
the soon-to-be event handler will work with user input!

installs React Bootstrap to include React Bootstrap and Button from  
'react-bootstrap/Button', styling button with type=submit to follow  
React Bootstrap guidelines, modeling button after react-bootstrap.github.io/  
components/buttons. `useState(defaultValue)` is on the componentDidMount term  
into `async onSubmit(e) { ... }` for which there's an event handler written in JSX!

Optimizes submit button for mobile view @media only screen and (max-width: 600px) { ... }

and modifies button with moving CSS background, displays previous user  
input and API responses from Mathjs with the selfResponse helper function

transferring from this. includes non-typing equality check! `:placeholder is colored :)`

## preflight Requests

```
Des.Handlers('Access-Control-Allow-Methods', "GET, POST, PUT, DELETE")
app.post(app.delete) /experience/deleteExperience,
async (req, res) => {
    var user = req.session.user
    if (user.email === req.body.impressiveExperienceEmail) {
        try {
            await db.collection('experiences')
                .doc(user.email + " " + req.body.impressiveExperienceName)
                .delete()
        } catch (err) {
            return res.status(500).json({ error: 'success' })
        }
    }
    catch (err) {
        console.log(err)
        return res.status(400).json({ error: err })
    }
}
.on pages/goolebig/index
.line number 422: next to <h3...
<button onClick={e => {
    e.preventDefault()
    this.deleteActivity(impressiveActivities[i].email
        + " " + impressiveActivities[i].impressiveActivity
        .nameThisActivity)
    ...
}}
style={{'fontSize': '2em', 'borderRadius': '1em'}}>
    </button>
    this.deleteExperience = this.deleteExperience.bind(this)
    async deleteExperience(experienceEmailAndName) {
        var response = await axios.delete(
            process.env.API_URL + '/experience/deleteExperience',
            { withCredentials: true, headers: { experienceEmailAndName, } }
        ).then((res) => { console.log(res) })
    }
}
```

Getting calendar unless logged in, <Form control name='start'> and <Form control name='end'> to display calendars by default without clicking on the input. mandatory to add at least one image to the experiences on the experience page

response data  
data  
changes

git stash --patch test(han)email.com todo:

- ✓ Look at the design for scheduling & the password is just for the backend.
- ✓ When you upload to Firebase Storage Bucket, create a ~~not for firebase~~ resized image. Generate thumbnails as soon as they're being uploaded: they're compressed and resized so that the fetching is much faster.
- ✓ On the frontend, render experiences which are current even if they don't have a start date or end date.
- ✓ There should be a way to deploy without installing node modules, because that step takes a lot of time; npm client alternatives like ied might solve this problem altogether, but in the meantime it would be interesting to test hashing package.json and skipping the installs with empty files. ~~--dry-run~~ git push test --dry-run.
- ✓ The fields on ~~gooleby~~ ~~will be required~~ except for start date and end date.
- ✓ import HttpsRedirect from 'react-https-redirect';

local/getLocalProfile /user/getProfile Data

```
• class HttpsApp extends React.Component {  
  constructor() {  
    super();  
    this.state = {  
      user: null,  
      error: null  
    };  
  }  
  
  componentDidMount() {  
    const user = JSON.parse(localStorage.getItem('user'));  
    if (user) {  
      this.setState({ user });  
    }  
  }  
  
  handleLogout() {  
    localStorage.removeItem('user');  
    this.setState({ user: null });  
  }  
  
  handleLogin(user) {  
    localStorage.setItem('user', JSON.stringify(user));  
    this.setState({ user });  
  }  
  
  render() {  
    return (  
      <div>  
        <HttpsWithRedirect>  
          <App />  
          <HttpsRedirect />  
        </HttpsWithRedirect>  
      </div>  
    );  
  }  
}
```

- ✓ \* On the booking Request form website: specify how many attendees. pages/customBookings.js
- ① Videos - when the expert uploads a video, it does not show up - we need to figure out what thumbnail will get displayed for a video.

- ② We need to compress images because half the images aren't loading half the time. The images are showing up on the profile, just not on the card; the card should fetch the compressed images, and the profile fetch the high-resolution one.

test(han)email.com

todo:

✓ Look at the design for scheduling & the password is just for the backend.

✓ When you upload to Firebase Storage Bucket, create a ~~not for firebase~~ resized image. Generate thumbnails as soon as they're being uploaded: they're compressed and resized so that the fetching is much faster.

✓ On the frontend, render experiences which are current even if they don't have a start date or end date.

✓ There should be a way to deploy without installing node modules, because that step takes a lot of time; npm client alternatives like ied might solve this problem altogether, but in the meantime it would be interesting to test hashing package.json and skipping the installs with empty files. ~~--dry-run~~ git push test --dry-run.

✓ The fields on ~~gooleby~~ ~~will be required~~ except for start date and end date.

✓ import HttpsRedirect from 'react-https-redirect';

local/getLocalProfile /user/getProfile Data

```
• class HttpsApp extends React.Component {  
  constructor() {  
    super();  
    this.state = {  
      user: null,  
      error: null  
    };  
  }  
  
  componentDidMount() {  
    const user = JSON.parse(localStorage.getItem('user'));  
    if (user) {  
      this.setState({ user });  
    }  
  }  
  
  handleLogout() {  
    localStorage.removeItem('user');  
    this.setState({ user: null });  
  }  
  
  handleLogin(user) {  
    localStorage.setItem('user', JSON.stringify(user));  
    this.setState({ user });  
  }  
  
  render() {  
    return (  
      <div>  
        <HttpsWithRedirect>  
          <App />  
          <HttpsRedirect />  
        </HttpsWithRedirect>  
      </div>  
    );  
  }  
}
```

✓ On the activity page if there is no address and they select in-person, then say "Address to be determined without specifying".

✓ Add a field to experiences: max number of attendees allowed.

- ✓ \* On the booking Request form website: specify how many attendees. pages/customBookings.js

① Videos - when the expert uploads a video, it does not show up - we need to figure out what thumbnail will get displayed for a video.

② We need to compress images because half the images aren't loading half the time. The images are showing up on the profile, just not on the card; the card should fetch the compressed images, and the profile fetch the high-resolution one.

③ When the user goes back to edit their experience from "Edit Profile" mode, the previous entered information is just squeezed out and displayed as a placeholder, but it's not actually there... so they still need to type out information they already entered... can we change this so that the information remains in the fields just like for the profile?

```
<form>
```

```
  <div> <input type="date" /> </div>
```

webshim: The polyfilling, capability based loading

JavaScript Library, npm install webshim

```
webshim.setOptions('forms-ext', {
```

```
  replaceall: 'auto', // if 'auto' input widgets are only
```

```
  types: 'date', // replaced in desktop browsers...
```

```
  date: {} // form-ext options can convert to
```

```
  startView: 2, // date-date-start-date, 2 for
```

```
  inlinePicker: true, // month view (the default) or month!
```

```
  classes: 'date-inputs'
```

```
});
```

```
webshim.setOptions('forms', {
```

```
  lazyCustomMessages: true // loads the messages only
```

```
}); // after window onload and therefore has no
```

```
webshim.polyfill // negative impact on the network performance.
```

```
('forms forms-ext'); // A polyfill is a piece of code
```

```
// usually JavaScript on the Web level to provide modern
```

```
// functionality on older browsers that do not natively support it.
```

```
$().function() {
```

```
  $('.format-date').each(function () {
```

```
    var $display = $('#date-display', this);
```

```
    $(this).on('change', function (e) {
```

```
      var localizedDate = webshim.format.date($prop({
```

```
        e.target, 'value'));
```

- ```

    $display.html(localizedData);
  });
});
};

• Create a [cloud-function] where the image can be
  compressed after [Upload] and replace the original
  image. You will need to watch for [storage] upload
  to trigger your function.
github.com/firebase/functions-samples/tree/main/generateThumbnail. See file functions/index.js for the
  thumbnail generation code. exports generateThumbnail
★ https://console.cloud.google.com/iam-admin/iam?
★ project = goolebybz has App Engine default service
  account and the Service Account Token Creator
  assigned to the member goolebybz@appspot.gservice
  account.com, which allows the app to create signed
  public URLs to the images.
• On pages/index.js, line 437 var [dayEnd: dayEnd,
  dayStart: dayStart] = immersiveActivity[].
  immersiveActivity
if (!!(dayStart || dayEnd)) moment().isAfter(dayStart)
  || moment().isBefore(dayEnd))
|| (dayStart || !dayEnd) && moment.isAfter
  (dayStart || dayEnd) && moment.isBefore
|| (!dayStart || !dayEnd)
• On pages/gooleby/corporate.js line 39
  this.state.meetingLocation is either 'Virtual' or 'Line'.
• components/dropzone gets props.addMeetingLink which from
  goolebyz/corporate filters the blob and file, which are
  used with firebase.storage().ref()
  (which expert uploads files? And what how does the
  data flow?

```

ganley-frontend-test-1.herokuapp.com/login  
aaaaaaaaaaaa@aaemail.com password

/user/getProfileData 400 (Bad Request)  
app.post('/auth/setLogin')

(req.session.user = { email: user.email })

commenting out

if (req.method == 'OPTIONS') &&

req.rawHeaders.includes('Origin')) &&

req.rawHeaders.includes('Access-Control-Request-Method')) {

res.status(204).end();

return;

returns status 204

req.method is  
OPTIONS  
req.method is  
POST

which is supposed to handle preflight requests,  
allows auth/setLogin to acquire user object  
on req.session.

① then POST /auth/setLogin

② then GET /user/getProfileData

③ "req.method is GET"

"getting profile data"

on /user/getProfileData, the req.session  
does not include the user object.

• 1, 2 or 3 (the middleware) resets the req.session.

In Heroku deployment, skip installing node dependencies:

npm install --unsafe-perm --userconfig \$build\_dir/.npmrc 2>&1

--unsafe-perm: option to run npm as the root account.

--userconfig: per-user configuration file (defaults to \$HOME/.npmrc);

configurable via CLI option --userconfig or environment variable  
\$NPM\_CONFIG\_USERCONFIG).

However, we create a file called .npmrc in your project folder  
with the following contents - proxy=. This will cause npm install  
to not modify your existing node\_modules directory. Note that  
this change will also apply to the npm prune command that Heroku

- ② How to get push test --dry-run and  
 • Sets form control value = this.state.endDate ? moment(this.state.endDateFull);  
 • maintains continuity of date types (Date) and 2021-10-28  
 • decreases size of 'Show Preview' text between this.state.startDate and this.state.endDateFull
- ③ When to use React in-line styles? Date and startDateFull runs, but Ul will NOT apply to the span selected command.  
 git push test --dry-run

```
<Form Control name='dateStart' type='date'  

  value={
```

this.state.startDate ? this.state.startDate : 'Date'

}

onchange={e => this.setState({startDate: e.target.value})}

/

• Currently, the form.dateStart.value is the value being submitted.

- moves activity duration dropdown menu to center below choosing the start date and end date calendar + preview, along with the question "How long is this activity going to be?" question.
- Want: Toggle in-line preview and calendar.
- converts onchange to a proper function, on Form Control and DatePicker.

To go between 2021-10-21 and

Thu Oct 21 2021 12:26:34 GMT-0400  
 (Eastern Daylight Time)

startDate = moment(startDate).format().slice(0, 10)

```
<input type="checkbox" id="showPreview" name="showPreview"  

  value="showPreview">
```

<label for="showPreview"> Show Preview </label>

{e => console.log(e.target.value)} onchange

this.state.showPreview

- moves styles from preview div on pages/googlebiz/corporate.js to styles/globals.css

④ How to escape div from parent div style?

1. Manually compensate for parent's positioning

2. Dynamically move the child with JavaScript.

What is the opposite of display: none?

Is there an opposite of display: none?

todo:

- ✓ put a price field in the Experiences page
- ✓ Flag the experiences that are virtual with a "Virtual" sticker - purple background - color #5f0455 - top right corner of the experience card - like the sticker on the first page.
- ✓ phone version - text should be shrunk down - same ratio between headers and text - smaller font so it's easier to navigate.
- optimize the cell phone experience
- Look at web browser simulators
  - ① How might one define @media queries in-line within `index.js`?
  - installs npm i react-currency-input-field
  - imports and uses `<CurrencyInput>`
- ② In React Bootstrap Forms, how to add to `form.activityPrice.value` without a `Form.Control` component?  
which sets `activityPrice` on this state.

How to copy all associated styles of one eliv like `Form.Control` to another eliv?

experiencePrice and activityMeetAttendee

- adds value to `form.activityPlanAttendee`
- conditionally `unless` on `./pages/activity.js`!
- only ~~if true~~ pushes to the Immersive Experiences image carousel only if `approvesEmails`. includes (`immersiveActivities[i].email`), on `./pages/index.js`.
- Shifting style from the virtual label on the first page to `./pages/index.js` top page between the virtual label and the experiences' 'virtual' flag, styles the "Virtual" sticker in the top right of the immersive experience card when `immersiveActivities[i].immersiveActivity` is `virtual`.

The modifications needed for the test server are:

On `app.post('/local/getLocalProfile')`

`.doc('testUser@email.com')` instead of `.doc(email)`

`app.get('/users/getProfileData')`: comment out the `if(user)`/`else` statement, assume there is a user,

and `.doc('testUser@email.com')` instead of `.doc(users.email)`

`localhost:3000/activity` links to `localhost:3000/local?userId=undefined`. On `.pages/activity.js`, this is `let href = '/local?userId=' + this.state.userId;`, from `this.setState({userId: userId + data.userId})`, from this is `kiana.dance24@gmail.com`

`var userId = await axios.post`

`process.env.API_URL + '/usernameByEmail'`,

`{ emailAddress: email }, { withCredentials: true, } )`  
contains receipt of `req.body.emailAddress` is `Kiana.dance24@gmail.com`.

`app.post('/usernameByEmail', async (req, res) => {`

`var data = await db`

`.collection('users')`

`.doc(req.body.emailAddress)`

`.get()`

`res.status(200).json({ userId: data['-fieldsProto'].`

`username.stringValue })`

3). With the example `aidanlewtt00@gmail.com`,

which has the username: "aidanlewtt",

adds the username: "Kiana.dance24" to the database `goolebybig/users/Kiana.dance24@gmail.com`. The value returned from the server is `userId: { ... data: userId: "Kiana.dance24" }`, which is sent to `/local/getLocalProfile`.

It appears that `collection.username/aidanlewtt:1` is a document object `{ email: "aidanlewtt00@gmail.com" }`. Adding the `Kiana.dance24` username collection document, consisting of the object

`{ email: "Kiana.dance24@gmail.com" }` results in the localhost

`:3000/activity?userId=kiana...` URL leading all our experiences/expert data, correctly.

★ If you don't specify an app name, Heroku will generate a random one for you.

Heroku login · heroku create

https://hidden-seas-60510.herokuapp.com/ |

https://git.heroku.com/hidden-seas-60510.git

~/create-react-app-blog (master ✘) npm start

~/create-react-app-blog/mysanityblog (master ✘) sanity start  
sanity.co/guides/build-your-first-blog-using-react

→ github.com/sanity-co/create-react-app-blog

· Removing yarn.lock and git pushing to heroku:master  
results in an Application error.

sanity.co/answers/hi-can-someone-help-me-deploying

-sanity-#160693|366108200

· run sanity build on deploy and then use the express.static  
to point your index there.

· sanity deploy makes https://hidden-seas-60510.sanity.studio/

· sanity build creates the mysanityblog folder

package.json:

"scripts": { "start": "react-scripts start" }

What does react-scripts start do?

from create-react-app: touch ./server/index.js

npm install express --save

follow the guide on expressjs.com/en/guide/using-middleware.html  
to generate Application-level middleware

const express = require('express');

const app = express();

const port = process.env.PORT || 5000;

app.listen(port, () => console.log(`Listening on port  
\${port}`));

app.get('/express-backend', (req, res) => {

res.send(`Your Express Backend is  
Connected to React!`);

});

node . /server/index.js  
in ~ /create-react-app-blog (master) node . /server/index.js  
Listening on port 5000  
localhost:5000 /express-backend

{  
  "express": "YOUR EXPRESS BACKEND IS CONNECTED  
  TO REACT  
}

npm run build  
npm install -g serve  
serve -s build

heroku run bash  
express is for the backend API routes.

Deploy using Vercel app: vercel.com/dashboard  
new → import repo

vercel.com/new/server:

Develop Run react-scripts start to run your project locally

Preview Push to any Git branch other than master to preview changes

Ship Push to master to ship changes to production

What are the frameworks for Vercel app deployment?

Deployment takes a minute and thirty seconds

- hidden-beaches-605f0.vercel.app adds to sanity.io/manage/personal/project/7your3tm/api# cors-origins
- hidden-beaches-605f0.sanity.studio/deck  
npx tailwindcss init

• Pictures

- ✓ • For "Price Range Preference", did we say "Price per attendee"? — or let's just say Pricing — because it could also be per hour — free text field with placeholder text "How does your pricing work?"

- ✓ • We don't need city of residence for venue; a venue else where somewhere

side by side

- ✓ • But for the location field, we can break it into "Street Address" and "City" dropdown so it can still show up under the city from the sort feature on our front page.
- ✓ • Add some space between the "Venue" header and the first field at the top of the page
- ✓ • On the Venue page, remove the links to "Become a Qooly Expert" or "Already an Expert?". We can just have the "About Us" page there.

/pages/activity.js

- 1) artistImage 2) artistName 3) artistCategory  
4) showcase 5) title 6) category 7) experienceDescription  
8) start date & end date 9) duration 10) notes 11) activityLocation and activityMaxAttendees · activityPrice  
12) this.state.meetingUpload 13) this.state.nameThisEntity 14) this.state &  
15) this.state.activityCategory 16) this.state.infoCatInformation  
18) this.state.startDate, this.state.endDate 19) this.state.  
impressionsActivityDuration 10) this.state.activityNotes,  
this.state.activityMaxAttendees this.state.activityPrice  
this.state.meetingLocation  
is "Live" or "Virtual". this.state.street

· city

· state

· zipCode

## Questions & Answers from Notebook 4

- In React.js, what are data formats and storage for which JavaScript can parse?

JSON ~~JSON~~ is schema-less, text-based representation of structured data. XML

- In Express.js, req.headers.origin the distinction or make between directories:

- The root argument specifies the root directory from which to serve static assets next()
- The function determines the file to serve by combining req.url with the provided root directory.

FIREBASE Set CORS configuration on a bucket by specifying information, such as HTTP methods and originating domains, that identify the types of requests it will accept.

gsutil, XML API, JSON API, or command-line tool

The client libraries for Cloud Storage to set CORS configuration on a bucket.

Google APIs Select your cloud function

→ add member → allUsers → Cloud Functions  
see a remark "Allow unauthenticated" Trustee  
on the list of your cloud functions.

- HTTPS SSL Certificate.
- Generate a CA certificate
  - Sign SSL certificates
  - Create a Node.js Express Server
  - Test in a browser and API client!
  - Import CA Certificate to the browser
  - Configure and test the API client

zoomJSlib zoomJSAVLib jsmedialibzoom

For ZoomMtg, check source: zoom.us

NPM Start script is default, not for script

- npm start without run keyword
- npm run dev

npm init creates package.json  
devDependencies modules are only required during development

node.js babel ESLint Chai Mocha Enzyme

Beautiful Axios Logging Messages

Google Analytics

Google Ads

Google Marketing Platform

<!-- Global site tag (gtag.js) - Google Analytics -->  
<script async src="https://www.googletagmanager.com/gtag/

/js?id=GA-MEASUREMENT-ID"></script>

<script>  
window.dataLayer = window.dataLayer || [];  
function gtag() { window.dataLayer.push(arguments); }  
gtag('js', new Date());

gtag('config', 'GA-MEASUREMENT-ID');

</script>

event: activity + element

Record user interactions React GA.event()

specific interaction video podcast form button

that don't trigger a change in URL:

[category: 'User', action: 'Created an Account']

[category: 'Social', action: 'Rated an App', value: 3]

[category: 'Editing', action: 'Deleted Component']

[category: 'Promotion', label: 'Game Widget',

action: 'Displayed Promotional Widget'

label: 'Homepage Thing',

interaction: true

ignore

non ignore

package.json whitelist files you want to include property

{"files": ["index.js"]}

- is used to specify which files should be ignored by Git from being tracked

Different classes each range of HTTP three-digit response HTTP status code.

installed

How do React Apps know to look for index.js?

1. package.json

2. public/index.html

3. Also, this file has a line of code `<div id="root"></div>`

This line is highly important since all the application components are loaded into this div.

4. src/index.js: This is the javascript file corresponding to index.html

this has the common code `ReactDOM.render(<App/>,  
(document.getElementById('root')));`

5. as we know App Component has to be loaded into an html element with id root.

- the configuration specifies that Webpack uses src/index.js as an "entry point".
- Git outperforms SCM solutions like Subversion, CVS, Perforce, and ClearCase.

cheap local branching, convenient staging areas, and numerous workflows

```
import pdfjsWorker from "pdfjs-dist/build/pdf-worker.entry";  
<Worker workerId={pdfjsWorker}>
```

`fileId={uri}`

`defaultScale={SpecialZoomLevel.PageFit}`

`plugins={[`

`defaultLayoutPluginInstance`

`]}`

`>`

`</Worker>`

The term "web worker" refers to a browser feature

worker → Can POST messages to an event handler specified by the JavaScript code that created it

\* can also use XMLHttpRequest to perform I/O (though the responseXML and channel attributes are always null) or fetch information (with no such restrictions).

## Explore our Services and Packages Mobile View

• There are certain CSS features, like `@media` queries, that must be defined in a declaration block in a stylesheet.

— Stackoverflow Question #28405444

- The font becoming thinner and more defined by transferring in-line styles into `.exploreOurExpertsMobileScreen`, `.boostEngagement`, ..., specifying the width and height of the `servicesAndPackages` class such that the parent `div`, which inherits from its constituent CSS classes.

Since 165px is the computed margin-left from `.servicesAndPackagesFullDiv`, adds `margin-left: 0;` to `.servicesAndPackagesMobileVersion` in order to get mobile view to sit to the left side of the page, `align-items` is to the center which allows the columns to have space on both sides.

- `/pages/index.js`: class becomes `className`, and `.servicesAndPackages`: hover adds a solid border. `.servicesAndPackagesFullDiv` `.nav-link` has `padding: 0`, and when they hover over `(thatWeSpecializeIn)` `onMouseOver` display an alert "This will take care of your (on-boarding, engagement, holiday/birthday party) events."

- For categories Onboarding, Team Building, Engagement, and Perks & Rewards, `onClick` initiates element booleans on this state which are rendering criteria via short-circuit & & evaluation. Makes all text white within the `(thatWeSpecializeIn)` flex container, rendering more information `onClick`.

Want: Be able to see all origins when accessing <http://www.gosley.com>.

What do we have currently?

```
app.use(function(req, res, next) {  
    res.header('Access-Control-Allow-Origin',)  
        ('http://localhost:3000')}
```

What do we want?

We want to have

```
app.use(function(req, res, next) {  
    var allowedDomains = ['http://localhost:3001',  
        'http://localhost:8080'];  
  
    var origin = req.headers.origin;  
    if (allowedDomains.indexOf(origin) > -1) {  
        res.setHeader('Access-Control-Allow-Origin', origin);  
    }  
})
```

what this means is that,

(given HTTP headers let the client and the server  
pass additional information with an HTTP request or response.)

\* If API URL is HTTP then

"This site can't provide a secure connection"

\* ?https=true & in package.json? ✓

however, the local computer cannot provide a secure connection.

SSL

Want:

\* app.use Is it just a shortcut?

for app.use function (req, res, next) { ✓

① So, does the middleware last each and every time  
you make a request? ✓

\* The middleware runs each time a new request comes in ✓

TRENDZ → undo Shift+Z → redo  
up to 15 steps!

What in Earth do we want?

- Feedback comment box ★ Posts to the database ★
- New Field: To the Create Profile and Edit Profile pages —

"Display Name" should be an optional field.

IF This field is filled out, then the

name of the service provider

on their profile ~~and~~ on their

card should change to the value entered here.

→ DisplayName currently does not exist on the database,

- We need to ★ POST to the DataBase ★

- Price Range field display:

This would require changes in the following files:

DB: priceRanges.

Where is the Edit Profile page? and Symp Edit page?  
pages/greedybit/index.js.

Which just indicates:

pages/greedybit/index.js : submitProfile fn governs  
signing up, while  
submitEdits fn governs  
editing profile.

Currently, want: fourth option:

options = [  
  { label: '<200', value: 0 },  
  { label: '200-400', value: 1 },  
  { label: '400-600', value: 2 },  
  { label: '600+', value: 3 }]

... { label: 'Don't Display', value: 4 }, ] }]

on pages/gsolleybie/index.js specifically  
this.setState([

priceRange: e.value ]), which is pulled off the

this.state.priceRange

in the functions: submitProfile and submitEdits.

And what ever this end up meaning?

\* I've got to add the fourth option, \*

and

I don't have to do much else, except for

How do you POST things to the database? How in Earth do you  
POST things to the database?

Obviously, when we click submitEdits or when we click submitProfile.  
So then, on pages/gsolleybie/index.js, do we have a separate route for  
/user/corporateSignup? yes we do. Why is that?

a POST route.

```
var response = await axios.post(  
  process.env.API_URL + '/user/updateProfile',
```

data,

{ withCredentials: true }

\* Does it all depend on just this data object? \*

Yes.

How is the POST request deconstructed  
(once the request hits the API router?)

gsolley-webapp

```
var data = await db  
  .collection('users')
```

.doc(user.email)

.update(newData)

newSession.profileData = newData

Sohm just hit a home run!

Time to find out!

\* Does Firestore have an ORM -  
a way to validate key-value pairs? \*

Can we just post anything we want? I don't know.



what we can

of objects can we POST to the database? ✓

pages/qaoleybz/index.js: submitProfile function and submitEdits function:

Change the data object:

Add a displayName field:

→ `data["displayName"] = "Temporary testing text"`

Put this on the submitEdits function, hardcode it.

I guess "name" is defined earlier, on the initial signup page?

Yes

→ but we don't want to have to hardcode the text:

get it directly from the form:

```
if (form.displayName.value) {  
  data['displayName'] = form.displayName.value.  
}
```

How do we get displayName on the form?

and what is this?

? What is this?

```
<Form.Group controlId='exampleForm.ControlInput4'> Sensitivity,  
<Form.Label>Display Name</Form.Label> what does this do?
```

<Form.Control

required={false} ✓

name='displayName' ✓

as='textArea' ✓

rows='3' ← see what this one does!

placeholder='What do you want to call yourself?' ✓

\* So they can make it as long as  
they want it to be \*

</Form.Group>

\* ? Kanisha, where do you want the display name to be? \*

Great! Now that the displayName is located on the database,

On the View Profile page, we can render the displayName if it

exists in the response which the server sends?

Yes, displayName is being passed to the database.

- On View Profile (pages/local.js), how can we access the displayName property?  
↳ the first "Why should you trust me?"

So on → pages/local.js — from where does

this.props.query.userId come?!

It was on componentDidMount.

- The displayName — is it being returned from the server?

Yes, specifically on response.state.data.displayName.

Interestingly, the response we are getting is outside of the try/catch block,

that is, variables declared within a try/catch block are accessible outside of that try/catch block.



Then we can immediately put the response.state.data.displayName on local state, and then after getting it on local state, we can do the following:

{ this.state.displayName ? this.state.displayName : this.state.profileData.name }

So on the View Profile page I've got a displayName entry both as a Title and then beneath the picture.

Still on pages/local.js.

There seems to be two more locations in which this.state.profileData.name is mentioned, and what are these places for?

These are definitely for the Edit Profile page.

So on pages/local.js — the profile names have all been filled in.

yes, definitely want to figure out why gladish.earn@gmail.com is inaccessible.

(#4 refers to subtitle on official page.) Ah, but we can't actually

(#3 refers to main title on official page.) get it naturally if we aren't

When the official page is loaded, at what are we looking? Logged in!

From where is the information coming?

They are being fetched from /local/getLocalProfile, in fact a POST route,

and to that effect let's rather whiteout the displayName localData.displayName.

And yes, the display name task is completely done, except for one main thing—  
what is it?

It is the fact that oh! the carts and the placemats.

Personalizing is important

Want to look at Display Name Form Label.

Need: to do this, will need

placeholder={[this.state.profileData.displayName]}. ✓

We are on `localhost:3000` `sslKeyFile`? `true`

Which registers as pages/100 web/index.js

→ Great the state has been updated

→ makes the displayName ~~task~~ complete. →

When will Don't Display become a thing? later

Oh! and the card. Is it very good?

What Does That Mean? It means that we need to know where the name data on the card comes from.

```
this.setState carouselDisplay
```

displayRow5

userObject → user.displayName or user.name

The name of the service provider is now showing up on their card!

★ Remember that `response.data.data.priceRange == 0` is a falsey value!

Signup <https://pages.googlebitz/index.js> and Edit

[View Profile](#) (~~logged in now!~~) [pages/local.js](#)

Next task: On `pages/local.js`,

get the display name to show up whether or not the viewer is logged in as that artist!

This will ~~use~~ still getting the user ID on local state.

But, if all — ~~where~~, on `pages/local.js`, would we find the name that is being displayed? A: within the second half of this.state.profileData.name.

Want to check for `this.state.profileData.name`

{  
  `this.state.profileData.displayName` ~~out~~  
  `this.state.profileData.name`...}

Want: to see where the

{  
  `(this.state.profileData && this.state.profileData.displayName)`  
  ?

{  
  `this.state.profileData.displayName : this.state.profileData.name`  
  statement breaks down. That's the profile page when not logged in.

By contrast, when we are logged in we have

{  
  `this.state.displayName ? this.state.displayName`  
  : `this.state.profileData.name`

\*git diff\*

3 3 3

Next Time! Tasks:

OAuth Google Signup

On the Our Customers section, the images are getting clipped off!  
Where are the images being rendered?

background-size: 100%; →

background-size: contain;

for the classnames thinkchips, ..., redshelf shown on `pages/index.js` starting at about line 1045.

for `.wustburnUniversity`, background-size: 100% 100%; →

background-size: contain;

except .orgSSPlaces.com, b/c otherwise SSPLACES.com would be  too small.

When there are gods, for whom are they going to look?

When the truth is revealed, be reincarnated well. What you don't notice...

The JavaScript origin in the request,

<https://www.gstatic.com>,

does not match the ones authorized for the OAuth client.

Visit

<https://console.developers.google.com/apis/credentials/oauthclient...>  
project=94934631743

to update the authorized JavaScript origins.

[Learn more](#)

Set a redirect URI

\* Triffin paradox:

the conflict of economic interests that arises between short-term domestic and long-term international objectives for countries whose currencies serve as global reserve currencies.

A reserve currency (or anchor currency) is a foreign currency that is held in significant quantities by central banks or other monetary authorities as part of their foreign exchange reserves.

A central bank is a financial institution given privileged control over the production and distribution of money and credit for a nation or a group of nations.

Foreign exchange reserves are cash and other reserve assets such as gold held by a central bank or other monetary authority that are primarily available to balance payments of the country, influence the foreign exchange rate of its currency, and to maintain confidence in financial markets.

- How many people went to our website on a daily basis ← dashboard on our admin portal.
- Build a dashboard to display number that updates every time someone clicks.
  - 1) Total # of clicks → how many people went to our website on a daily basis.
    - Filter by month.
    - Compare between months.

What kind of data?

① Data Format?

② Where do we store the data? ✓

Search Engine Optimization Headers:

1) Use Headers to Give Details -

Your H1 presents the theme your page is about, similarly as a title enlightens a person about a book.

The H2s are similar to book parts, portraying the principle subjects you'll cover in areas of the article.

Resulting headers, H2 to H6, fill in as extra sub-headings inside each segment; similarly as a book part might be separated by numerous sub-chapters.

2) Go through Headers to Break Text

A readable article is a decipherable article, and a comprehensible article is one that is bound to perform well in the webcrawlers.

Searchability is so imperative to the accomplishment of an article, that distributions like Forbes have gone so far to call it "the most neglected factor of substance showcasing."

3) Remember Catchphrases for Your Header Labels -

While headers won't affect your search engine optimization as a backlink from a legitimate site, Google actually sees them to assess the setting for your page.

Likewise with anything Google focuses on, that implies it means remembering catchphrases for your header labels.

This doesn't mean you should stockpile catchphrases in no matter what.

Be wise, not malicious.

#### 4) Advance for Highlighted Bits -

One spot where header labels do appear to have a beneficial outcome in Website optimization is with highlighted scraps in two specific manners:

The first is advancing your header tag for a long-tail voice search watchword, and afterward putting the inquiring straight-forwardly beneath utilizing scraps inside <P> passage labels.

The second is utilizing ensuring, more market headings to layout distinctive rundown things. Google utilizes those headers to make its own bulletted and numbered records included bit results.

#### 5) Just Utilize One H1 -

We should scatter a typical Search engine optimization legend.

"Google has said there is no issue with utilizing different H1s." ← the legend

In any case, that doesn't mean it's a Search engine optimization best practice to utilize various H1s on a page.

H1s are huge, and they look like titles to perusers. Utilize numerous H1s on your page, and it begins to glance somewhat wild.

\*Need to ensure you don't have any different H1s existing on your site?\*

\*Run your area through a crawler device like Shutting Frog.\*

\*Switch over to the H1 tab to see initially whether you have any pages that are missing H1s completely or have different H1s.\*

#### 6) Keep Your Header Labels Reliable -

A decent general guideline for UX is to keep a predictable encounter for clients. At the point when a site accomplishes that down to the best detail, it's amazing. In tend to intrigue with steady header labels.

\*On the off chance that you decide to utilize title case design, stay with that across the entirety of your pages (and the other way around on the off chance that you pick sentence case).\*

Additionally, keep your headings on the more limited side.

A header tag isn't the spot to accommodate a square of watchword rich content for Google.

Deal with it like a small scale title for the accompanying part of text.

### ⑤ Make Them Fascinating

The standard applies to all your copywriting, not simply the headers. Your underlying draft may have garbage headers that you use to make your framework, however you ought to constantly survey and modify them prior to distributing to make them convincing for guests. Indeed, your header labels make an article searchable. Yet, in a perfect world, they don't check the entire route through. Interesting header labels urge guests to take a beat and read for some time. Spot exceptional significance on your H1 tags specifically. Clients like H1s. In exceptional part, your H1 may direct whether guests trouble to look down the page by any means. Put forth a valiant effort to find a lone great H1 label that answers the clients' pursuit expectation, and get them armed up for perusing your article.

Remember<sup>10</sup> Google Analytics can be easily integrated with our React App using the react-ga module, which makes things straightforward.

Remember<sup>11</sup> You can also send page view events from navigating across the app using the react-router module: import { BrowserRouter } from react-router-dom. If you are looking for good SEO crawler devices along the line of Screaming Frog SEO Spider, then maybe know of Apify, WildStarts SEO Spider Tool, DeepCrawl Scrapper. Why do we need to remember these two things, and what does it mean to integrate G.A. and page view events tracking navigation? Because in today's data driven world, web analytics has become a key tool for understanding user behavior, optimizing web usage and driving business decisions. Analytics platforms measure various types of interactions in a website including for example (but not limited to): how many users visit, how long they stay, which pages they visit, which buttons they click and whether they come by following a link or not.\*

Adding any kind of tracking to our project though always seems to be an afterthought, causing our app to be littered with third-party javascript

introduction calls. But with careful consideration and thoughtful design, we can meet all the linking requirements of our website while keeping our React code relatively simple and side-effect free.

npm install react-ga --save

```
import {ReactGA} from 'react-ga';
```

```
(ReactGA) initialize('Ut-XXXXXXX');
```

```
import React from 'react';
```

```
import ReactGA from 'react-ga';
```

```
export default class App extends React.Component {
```

```
  onClickLink() {
```

```
    *ReactGA.event({
```

```
      category: 'Link',
```

```
      action: 'click',
```

```
    });
```

```
    render() { return ( <div> <a href="#" onClick={()=> {this.onClickLink()}}>
```

```
      Link </a> </div> ); } }
```

```
import React from 'react';
```

```
import ReactDOM from 'react-dom';
```

```
import { BrowserRouter } from 'react-router-dom';
```

```
import ReactGA from 'react-ga';
```

```
export class HomeComponent extends React.Component {
```

```
  static propTypes = { location: PropTypes.object.isRequired }
```

```
  onPageView() { *ReactGA.set({ page: this.props.location.pathname }) }
```

```
  *ReactGA.pageview(this.props.location.pathname) }
```

```
  *componentDidMount() { this.onPageView(); }
```

```
  render() { return <h1>Hello world! </h1> }
```

What is rendered with ReactDOM?

```
let render = () => { ReactDOM.render(<BrowserRouter> <Route>
```

```
  <Route path="/" component={HomeComponent} /> </Route>
```

```
  </BrowserRouter>, document.getElementById('root')) ); }
```

For "clicking in links" event

For "viewing page" event

based on React Router

git remote -v means, show URLs of remote repositories when listing your current remote connections.

'<https://www.ganley.com>' → '<https://www.ganley-frontend-test-2.herokuapp.com>'

\* The trailing slash is generally used to distinguish a directory which has the \*  
\* suffixes from a file that does not have the trailing slash. \*

The trailing slash not on req.headers.original, indicating that:

- ① Is there a distinction made between directories → yes? ✓
- ② Or not.

\* <https://ganley-frontend-test-2.herokuapp.com> \*

### Git connected Using Google Analytics with React

How to set up analytics tracking in React.js with Google Analytics.

\* Need: to replace TrackingId with your actual Google Analytics tracking id. To find your tracking id:

- ① Sign in to <https://analytics.google.com>
- ② Click Admin. ③ Select an account, select a property ④ Tracking Info  
if you have a 'Sign Up Now!' button, you could then Tracking Code  
measure how many times the button was clicked is how many users  
actually signed up.

\* ReactGA.event({ ← Usage with Custom Events

category: "Sign Up",

action: "User pressed the big blue sign up button";

\*}); The event would then → Google → Available for analysis.  
Pageview Tracking

Saturday May 8, 2021

On gslely-frontend-test-2,

Hold On!  
Network Error

Signup

- ⑩ Access to XMLHttpRequest at '<https://backend.gslely.com/signup/saveUserToDatabase>' from origin '<https://gslely-frontend-test-2.herokuapp.com>' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.  
and then we get  
net::ERR\_FAILED
- ⑪ POST <https://backend.gslely.com/signup/saveUserToDatabase>  
net::ERR\_FAILED
- ⑫ POST <https://www.googleapis.com/identitytoolkit/v3/relyingparty/verifyPassword?key=AIza...VCY> 400  
which means  
GET <https://backend.gslely.com/user/getProfileData> 400  
(Bad Request)

Sohan says, we don't need to add it to the Firebase API, just google cloud console.

issue: domain itself or its communication with the backend server.  
It's not a problem with the origin and it's not a problem with the backend code.

It's a problem with CORS because localhost:3000 and localhost:5000 are working and gslely.com and backend.gslely.com working,  
however: localhost:3000/signup and backend.gslely.com are not working  
and this is with the same code.

Similarly, https://gslely-frontend-test-2.herokuapp.com  
and backend.gslely.com do not mesh.

Hypothesis: Firebase is enforcing SRS  
and Google APIs is enforcing CORS.

given that we can fetch data from the backend,  
which does not go through Firebase and/or Google? ✓

 we got our site on <https://chalkboard-1.herokuapp.com/>!  
To deploy our site on Heroku Create Heroku App, CD git clone  
<https://github.com/gldishd/chalkboard.git> because it has Zoom Version 1.2.5,  
hidden scrollbars, and Zoom Credentials, (2) git remote add heroku  
<https://git.heroku.com/chalkboard-1.git> for which chalkboard-1 could be the  
new Heroku App name, (3) npm install, (4) ln -snt ./.auth zoomSecrets.js  
and add on the first line: process.env['REACT\_APP\_ZOOM\_API\_KEY'] =  
'Kmc2reNRimighQJ6P1muQ' and on the second line: process.env['REACT\_  
APP\_ZOOM\_API\_SECRET\_KEY'] = 'RDA3sLNpGFMWdYh85CeyCPRir  
QDsAjrQRTE' or if you want new credentials then update ./client/  
components/Zoom/Zoom.js based on the Zoom doc, and create ./zoomSecrets.js  
with JWT API Key and API Secret from marketplace zoom us/developer/create,  
(5) heroku login, npm run deploy, on dashboard.herokuapp.com/apps/chalkboard-1/  
resources in Heroku add Heroku Portfiles, (6) heroku run bash, then npm install,  
npm install -dev, exit, heroku config:set PORT\_MODE=development, heroku run bash,  
npm run seed. Remember to connect with HTTPS in the browser!  
\*If you want new credentials then make sure to update ./client/components/Zoom/Zoom.js  
and ./zoomSecrets.js

\* The problem is with the Learn More button itself, which should ~~not~~  
take us to /profile/edit; it should take us to /signup.

The reason that

```
var response = await axios.get(  
  process.env.APT_URL + '/user/getProfileData',
```

{

withCredentials: true,

)

for

? What is the Axios logger called? ✓

? What is the difference between dev dependencies and regular dependencies?  
in package.json and why?

npm install react-ga --save

Google Analytics Tracking ID:

- Sign in to https://analytics.google.com
- Find Tracking Info then Tracking Code.

?) "Couldn't send you a prompt because your phone can't be reached!"  
What does this mean?

UA-161923401-1

to, and how does one use it?

?) What is Global Site Tag (gtag.js), and do we really have to use it?

What does it mean to fire a GT tracking event?

What are the ingredients?

```
import ReactGA from 'react-ga';
ReactGA.initialize('UA-161923401-1');
```

Let's suppose somebody clicks on a link

```
import React from 'react';
// Component stuff //
handleClick() {
  ReactGA.event({
    category: 'Link',
    action: 'Click',
  });
}
```

How we handle click!

\* fires a GT tracking event  
with the event details.

Within the render section

```
<a onClick={()=>{this.handleClick()}}> Link </a>
```

Send Page Views using react-route Wrapper This is in the Basic Index

```
import React from 'react';
import Component from './Component';
```

```
import ReactDOM from 'react-dom'
```

```
import { BrowserRouter } from 'react-router-dom'
```

```
import ReactGA from 'react-ga';
```

These are import statements

// component stuff

```
static propTypes = {
```

location: PropTypes.object.isRequired,

```
}
```

```
onPageView() {
```

```
  ReactGA.set({page: this.props.location.pathname})
  ReactGA.pageview(this.props.location.pathname)
```

```
}
```

```
componentDidMount() { this.onPageView(); }
```

• What's in the render section of this main index component?

• `render = () => {`

`ReactDOM.render(`

`<BrowserRouter>`

`<Route>`

`<Route path="/" />`

`component={HomeComponent} />`

`</Route>`

`</BrowserRouter>,`

`document.getElementById('root')`

`)`

I what should document.get.. refer to?

}; What do we have this in the document?!

On any component we desire,

We can simply fire ReactGA.event for clicks

and on the

index tree root component, the one which renders Routes,

We are

able to track page views. Is it

by page directory is it a catch-all, ... or

does ReactGA count... something entirely unexpected?

① npm install react-ga --save is a real thing.

How can we not save it? Such that git stash doesn't work.

② How do React apps know to look for index.js?

We want: index page for Routes.

Want to have the greatest index of all time!

- \* The gtag.js library is a JavaScript tracking framework and API that sends events to Google Ads, Campaign Manager, Display & Video 360, Doubleclick Search, and Google Analytics.

So we can actually send stuff to Google Ads!

Anyway: actually, no, we don't want to have the greatest index of all time, we want to simply do an insert with regard to ReactDOM.

- \* Did we already import React from 'react' and did we already import ReactDOM from 'react-dom'?

Yes? Good. Then: `import { BrowserRouter } from 'react-router-dom';  
import ReactGA from 'react-ga';`

- \* PropTypes exports a range of validators that can be used

to make sure the data you receive is valid.

In this example, we're using PropTypes.string.  
`static propTypes = { location: PropTypes.object.isRequired, }`

// BECAUSE we want to make sure we can access this.props.location.pathname on `PageView()`

`ReactPropTypes.page: this.props.location.pathname })`

`ReactGA.pageview(this.props.location.pathname)`

— Call this on componentDidMount().

// How are we even getting this information? We're using  
`import { BrowserRouter } from 'react-router-dom';`



~~function = () => {~~

~~ReactDOM.render(~~

~~<BrowserRouter>~~

~~<Route path="/>~~

~~component={HomeComponent} />~~

~~</Route>~~

~~</BrowserRouter>,~~

~~document.getElementById('root');~~

~~);~~

~~});~~

~~just check if this.props.location.pathname exists~~

Page Views are integrable as long as we use the package  
import { BrowserRouter } from 'react-router-dom';

import ReactGA from 'react-ga';

npm install reactga --save

ReactGA.initialize('UA-161923401-1');

eventHandler() {  
    ReactGA.event({  
        category: 'Link',  
        action: 'Click'  
    });  
}

? In ReactGA, what kind of category and action are there?

<a eventHandler={()=>{this.onClickLink()}}> Link </a>

What are the URLs and their corresponding components?

/privacyPolicy

/signup  
/login  
/about  
/termsAndConditions

\* directories on which to put buttons which respond with ReactGA events.

internally, this is:

Router.push href=

Pagelet Integration w/ ReactJS

✓ /  
✓ /sign/  
✓ /logic/  
✓ /oxic/  
✓ /smashConditions  
✓ /privacyPolicy  
✓ local? userId=max-fisnos  
✓ googlebiz? true Does GA include /googlebiz? true ?  
✓ logout  
✓ sendReviewForm  
✓ /googlebiz/corporate  
✓ /customBuckets  
✓ /googlebiz/user  
✓ /googlebiz/

How are client-side try/ catch blocks triggered by server sending 400 error HTTP status back?

★ Accessed by this.props.query.true

★ Accessed by this.props.query.userID

local? userID=max-fisnos ★

googlebiz? true Does GA include /googlebiz? true ?

and /googlebiz? ? yes. Search query is in the URL

★ window.location.pathname + window.location.search

On analytics.google.com/analysis/web/

submit-time is inaccessible directly but without

the name time it redirect you to this address

maybe make it accessible directly via redirect

Soham, if /user/ isloggedin is just meant to return true or false, then are you just saving the response object by converted, that is, the real point is whether we get an error or not?

★ That is, does Google Analytics include /register or not?

all visits not just to /googlebiz but also to

/googlebiz? true and /googlebiz/ and all sub-categories of

/googlebiz? and /sendReviewForm which requires administrator.

★ From Web SDK API

Handle Dispositions

and Google Analytics API

Routing: Next.js has a file-system based router built on the config of pages.

When a file is added to the 'pages' directory it's automatically available as a route.

The files inside the 'pages' directory can be used to define most common patterns.

★ Plus, the window.location.pathname returns the pathname of the current page: window.location.pathname → /js/js-with-location.js



Q Is it customary to use HTTP response codes as a criterion for continuing to run a block of code?

If so, then how can we

Logged-in status should not be indicated by HTTP response code.

• return res.status(200).json({ loggedIn: true })

• return res.status(400).json({ error: 'User is not logged in' })

• This will actually throw an error message which will stop the rest of the code from running.

\* what happens when window is undefined?

Then, manually type about, for example,

```
import ReactGA from 'react-ga';
ReactGA.initialize('UA-16923401-1');
ReactGA.pageview(window.location.pathname);
await axios.get(process.env.API_URL + '/user/isLoggedIn', {
  withCredentials: true
})
```

Router.push('/goalbybiz/')

```
catch(err) {
  console.log(err)
```

\* AND Want: measure all clicks on all buttons on

- ① Main Page
- ② Login Page
- ③ Signup Page.

\* axios-logger package: Beautify Axios Logging Messages.

? Warning: Setting up fake worker ← What does that mean? ↗  
at http://jsbin.com/125A

- ★ Web Workers are a simple means for web content to run scripts in background threads. What does "thread" mean? !
- The worker thread can perform tasks without interfering with the User interface.
- In addition, they can perform I/O using XMLHttpRequest.  
Using the responseXML and channel attributes are always null or broken with no such restrictions.

Once created, a worker can send messages to the JavaScript code that created it by posting messages to an event handler specified by that code and vice versa.

- ★ A worker is an object created using a constructor \*

- e.g. [Worker] - this runs a named JavaScript file.
  - This file contains the code that will run in the worker thread.
  - Workers run in another global context (?) What is the worker global context? That is different from the current window. Thus,
  - Using the window shortcut to get the current global scope instead of (self) within a Worker will return an error.
  - The worker context is represented by a DedicatedWorkerGlobalScope object in the case of dedicated workers; standard workers global are utilized by a single script; shared workers use SharedWorkerGlobalScope.
  - A dedicated worker is only accessible from the script that first spawned it, whereas shared workers can be accessed from multiple scripts.
- "The default worker is an inline file worker. To improve performance, either add the worker-hander (preferred) to your Webpack config or use the inline workersyntax." - [bestofjs.org](http://bestofjs.org)

- Workers are accessible from scripts.
- ?) Why encourage writing a non-in-line worker in a separate JavaScript file?
- ?) What is the window context? • What is the worker global context?

How can we fetch data from the Google Analytics API?

You query the Core Reporting API for Google Analytics report data.

Each query requires a view (profile) ID, a start and end date, and at least one metric. You may also supply additional query parameters such as dimensions, filters, and segment to refine your query.

See the Overview Guide to understand how all these concepts work together.

### Request

The API provides a single method to request data:

`AnalyticsService ga.get()`

This method is exposed in various client libraries and has language specific interfaces to set the query parameters.

The API also can be queried as a RESTful endpoint:

Authorization: Bearer {oauth2-token}

GET <https://www.googleapis.com/analytics/v3/data/ga>

?ids=ga:12345

&start-date=2008-10-01

&end-date=2008-12-31

&metrics=ga:sessions,ga:bounces

Each URL query parameter specifies an API query parameter that  
must be URL encoded.

What if we do not set query parameters?

## ① Enable Analytics Reporting API

### ② Get Client ID.

How to develop a custom Google Analytics Dashboard using Google Analytics Reporting API v4 and React.js.

- You can use it for measuring a website performance by a large amount of metrics.
- But what if you need to build a custom app where you can see just some metrics that your customer want to measure?

React.js and Google Analytics Reporting API.

Prerequisites:

- Google Analytics account with an existing website app setup
- Basic knowledge of React.js

### Part 1: Enable the API

Link → "Create a project" → "Continue"  
When you see The API is enabled → Click "Go to credentials"  
answer 3 questions: Analytics Reporting API, Web browser (Javascript),  
User data; press "What credentials do I need?" button →

See modal window to: set up OAuth consent screen.  
Press "Set up consent screen" button → in the next minute  
choose External User Type and press "Create", on the next page  
it is sufficient just to fill the App name and Support email (filled automatically).

Leave all other fields empty. Press "Save" button. The last thing. Navigate to "Credentials" menu on the left bar then press "Create credentials".

In the dropdown menu select "Client ID".

Application type field: Web Application

AUTH. JS origins field: http://localhost:3000

Enter the name of the app: Website 1 by default. Press "Save". ★

★ The data with your Client ID and Client Secret appears.

We always can see the credentials in the developer console.

? Can we get statistics without the Google Sign-in button?

## Part 2: Setup React App:

~~npm create-react-app react-google-analytics~~

- delete all demo code in the App.js file.

- inside the head tag of the public/index.html file:

```
<script src="https://apis.google.com/js/client:platform.js">
</script>
```

- Create utils.js file in src folder

src/utils.js

```
const initAuth = () => {
```

```
  return window.gapi.auth2.init({
    client_id: "YOUR-CLIENT-ID", // paste your client ID here
    scope: "https://www.googleapis.com/auth/analytics.readonly",
  });
};
```

```
export const checkSignedIn = () => {
```

```
  return new Promise((resolve, reject) => {
```

```
    initAuth() // call the previous function
```

```
    .then(() => {
```

```
      const auth = window.gapi.auth2.getAuthInstance();
```

```
      // returns the GoogleAuth object
```

```
      resolve(auth.isSignedIn.get());
```

```
      // returns whether the current user is currently signed in
```

```
    })
```

```
    .catch((error) => {
```

```
      reject(error);
    });
  });
};
```

```
};
```

Also, we need to add the Google Sign-in button.

```
export const renderButton = () => {
  window.gapi.load('signin2', render('signin-button', {
    scope: "profile email",
    width: 240,
    height: 50,
    language: true,
    theme: "dark",
    onSuccess: onSuccess,
    onFailure: onFailure,
  }));
};

const onSuccess = (googleUser) => {
  console.log("Logged in as: " + googleUser.getBasicProfile().getName());
};

const onFailure = (error) => {
  console.error(error);
};
```

That's it for the util functions.

Now let's code the main logic of the app: on page load, check if user is signed in.

If not - render the sign-in button, if yes - request Google Analytics Report (for example, daily user's visits for the last 10 days) and show it in the browser.

App.js



- P.S. ReactGA events in all buttons for each of these directories:
- /signup /login /about /termsAndConditions  
 /privacyPolicy

### Tracking Events:

export const GTEvent = (categoryName, eventName) => {  
 ReactGA.event({

required category: categoryName, 'Link',  
 required action: eventName, 'Click',  
 optional label: 'labelName', // e.g. alongside the Click action, we could add the name of  
 optional value: 10, // a component as the label.  
 optional noInteraction: false, // a means of recording a numerical value against an event.  
 // if an event is not triggered by a user interaction, but instead by someone  
 // (e.g. on page load, it should be flagged as a noInteraction event)

// to avoid skewing bounce rate data.)      <Navbars.Brand>

onClick={handleButtonClick} >

</>      X

this invokes the function

<Navbars.Brand className='qualegLogo' href='/'  
 onClick={handleButtonClick}  
 onClick={(e) => handleButtonClick(e)}>

</>      ✓

✓

In handleButtonClick, e.target.href refers not to but to the full url value

(http://localhost:3001/)

★ Make sure to test the Navbars on Login-In view. ★  
 - Make sure the Qualeg Logo shows up.

Want `onClick={() => handleButtonClick(e, city.toString())}`

Recall: `import handleButtonClick from '../components/buttonClick'`

that first card click to call fn.

pages/index.js - spa-ly-frontend

```
  this.state.carouselDisplay['Comedy']
    |----- displayfunc[category] renders {carousel}
  for(var category of categories) {
    var carousel = []
    ... and then carousel.push(<a href={`/local?userId=' />
      import ReactGA from 'react-ga'
```

## Quodley Frontend

- On the Mobile View:
- ① Gallery for Ads → enlarge ✓
  - ② Flexible our Experts
  - ③ Filter by Price and City → Center ✓
  - ④ SEO: searchenginewatch.com/18/104/04/a-quick-and-easy-guide-to-met-tags-in-seo/
  - ⑤ On Our Customers section: two icons per row ✓

2. if (this.state.profileData.approvedProfile)

versus

↳ if (this.state.loggedIn & this.state.profileData != null)

className = 'expertCarousel'

Issue: whenever you reload the page, the content stays up as expected; when you switch screen type from mobile to web and back, you get lost tracking.

★ In VS Code you can click and drag to get split view tabs! ★  
margin auto; center div!

Toggle device toolbar does not cause a re-render!

★ add increased width for About Me, Request Booking, Why Should you book me? sections.

git commit -amend

Especially on mobile. Click on picture to go full screen

React FullscreenImage library to render images that can auto-fit fullscreen when clicked.

page/test.js either I'm logged in and looking at my own profile,

or

If I'm not logged in (first time), I'm looking at someone else's profile.  
either way I want to make the photos expand when you click them:

from `function StandardItem`

- 1 `this.state = { ... }`  
`StandardItem: []`,
- 2 `var item = this.state.StandardItem // the first time this is set, it has [] as its value.`  
`: getNextUpdate`

`this.setState({`

`StandardItem: item,`

~~\* modify the image which we are pushing to the items array. \*~~

`<img style={{ width: '100%', height: 500, objectFit: 'contain' }}  
src={item.url} className='reactCardMedia' />`

X Global CSS ~~can't~~ be imported from within getNextUpdate.

maybe need to work on the CSS imports?

`node_modules/react-native/Libraries/react-native.css`

Reach out to the maintainer and ask for them to publish a compiled version of their dependency.

- What does it mean to compile a dependency?
- What is the dependency development lifecycle; how do you write dependencies for npm?
- What does a compiled version of their dependency look like?
- Would removing ~~QSS imports~~ work?

Read the Next.js

How does fullscreen-click work from a CSS standpoint?

TestFlight: push to TestFlight iOS App

```
Suppose you set width: (80vw)  
height: 10vh,  
display: block,  
position: fixed;
```

so click  
jQuery?

- Does it make sense to include npm inside?  
Not really though?
- Where can you put a const like  
in a React component?

className = 'reactCarouselMedia'

this does not seem to trigger componentDidMount!

However You Choose

\* and we get "click on picture to go full screen not necessary for MVP !!!!"

increase in size

- className = 'expertCarousel'
- className = 'expertCarouselMobileVersion'

Style = {! this.state.showCarousel ? {display: 'none'} : {display: 'block'}}}

arrows = {this.state.showCarousel ? true : false}



but not for the mobile version!  
only for the traditional view.

# Dooley App Customer Side!

Dear Gladys    dooleygladys@example.com password123  
↓

## View 1      Browse Activities

Virtual Only  ← does this literally refer to events

No activities to display which occur virtually?

\* Certainly there is a way to improve the design of the user interface!

## View 2      My Bookings

Have not booked an activity yet!

① What is the database origin of activities, and what is the database origin of bookings?

Workflow: ① Test locally first: expo start

② expo build for simulator → <https://expo.io>

③ expo build for App Store → <https://help.apple.com>

\* Distribute an app using TestFlight (iOS, tvOS, watchOS)

1. Archive your app

2. Validate and upload the archive to App Store Connect <https://appstoreconnect.apple.com>

3. In App Store Connect, view build details and add metadata to the app version

4. Invite internal and external testers

5. Users download TestFlight and accept your invitations [reinhardtive.com](https://reinhardtive.com)

How To Upload Your iOS app to TestFlight for distribution/beta

Create your App Bundle ID or have it created for you

<https://com.yourcompany.yourappname>

<https://com.Dooley.DooleyTraveler>

Step 1 - Set up your project in the App Store

Sign in to <https://developer.apple.com> → Certificates, Identifiers & Profiles

→ App IDs → App ID Description → Explicit App ID option.

## Step 2 - Add your details to the App Store project

<https://developer.apple.com> iTunes Connect → My Apps + New App.

Complete the details and make sure to choose your app's bundle id correctly.

## Step 3 - Upload your project

In Xcode, add your bundle ID to the app's settings  
and select the company from the signing options.

Select Generic iOS Device from the top menu.

Select Scheme from the Product menu.

Once this completes, it will open a window called the Organizer.

You can find it at Window → Organizer

in the top menu of Xcode.

Click the Upload to App Store button followed by Next and wait for it to upload.

If for any reason it fails at this point, it will be due to a certificate error and

you will have to contact someone from your Apple Developer Account team.

## Step 4 - push it live

Log in to <https://itunesconnect.apple.com> In your app, click on TestFlight in the top menu.  
You should see the build uploaded. Be sure to note down your build number for adding  
to a beta group.

Look through the scheduling backend logic on the BacklogZig App.

① pages/NavPages/CreateActivityPages/NameActivity.js  
"Let's name your activity!"

pages/NavPages/CreateActivityPages/TagsActivity.js

② pages/NavPages/CreateActivityPages/MediaComponents/MediaTool.js

③ pages/NavPages/CreateActivityPage/DescribeActivity.js  
"Tell us about your activity!"

④ pages/NavPages/CreateActivityPages/CategoryActivity.js  
"How would you categorize your activity?"

⑤ pages/NavPages/CreateActivityPages/MediaLocation.js  
"What kind of activity is it?"

⑥ pages/NavPages/CreateActivityPage/NotesActivity.js  
"Help improve your traveler(s) experience!"

⑦ pages/NavPages/CreateActivityPages/DurationActivity.js  
"Click below to set the length of your activity."

⑧ pages/NavPages/CreateActivityPages/RangeActivity.js  
"Click the dates below to choose a start and end date for your activity."

⑨ pages/NavPages/CreateActivityPage/DaysActivity.js  
"Click the day(s) of the week at the top to select the different days you want to offer your activity."

⑩ pages/NavPages/CreateActivityPages/SessionActivity.js  
"Select a day of the week and add sessions corresponding to that day."

username: Seanfeldish1@gmail.com

password is password

Qooley Firebase app users → Seanfeldish1@gmail.com

① Is there an admin dashboard for the QooleyBiz app?

The data is being posted to the same Firebase database.

① Name  
Let's name your activity!

NEXT → ② Add media

③ Description  
Tell us about your activity!  
What are some cool things you want people  
to know about the activity?  
Activity Description

NEXT → ④ Categories

How would you categorize  
your activity?

Arts and Entertainment, Food and Drink,  
Professional Training, Health and Wellness,  
Educational, History and Culture,

Volunteer Experience, Outdoor Activities

⑤ Meeting Location  
What kind of activity is it?  
 Virtual    Live

NEXT → ⑥ Notes

Help improve your traveler's  
experience!  
Ex. Please wear comfortable shoes

Notes

⑦ Activity Duration  
Click below to set the length  
of your activity.  
15 minutes

NEXT → ⑧ Activity Range

Click the dates below to choose a start  
and end date for your activity  
Start Date: 2021-07-06 End Date: 2021-07-09

⑨ Activity Days  
Click the day(s) of the week  
at the top to select the different  
days you want to offer your activity.  
Days  
Su Mo Tu We Th Fr Sa  
< July 2021 >

NEXT → ⑩ Activity Sessions

Select a day of the week and add  
sessions corresponding to that day.

Select a day of the week ✓

NEXT →

⑪ Price  
How much will you charge?  
Set a price that your user will pay  
\$0.00

⑫ Preview  
Submit

⑪ pages/Maypages/CreateActivityPages/PriceActivity.js  
"How much will you charge?"

⑫ ←

Preview  
General Information  
Name  
Description  
Category  
Notes  
Schedule Information  
Start Date  
End Date  
Activity Duration  
Activity Sessions  
Pricing Information  
Activity Price  
**SUBMIT**

pages/Maypages/CreateActivityPages/PreviewActivity.js

NameActivity → TypeActivity → MediaCard → DescriptionActivity

CategoryActivity → MeetingLocation → NoteActivity

DurationActivity → RangeActivity → DaysActivity

SessionActivity → PriorityActivity → ProminenceActivity

NameActivity / navigation.addListener('focus', () => {

// The screen is focused

console.log('FOCUS ED')

// console.log(date.name)

// if (date.name) {

// setTextField(date.name)

// }

// call any action

})

Keyboard.addListener('KeyboardWillShow', -KeyboardDidShow)

Keyboard.addListener('KeyboardWillHide', -KeyboardDidHide)

// cleanup function

return () => {

Keyboard.removeListener('KeyboardWillShow', -KeyboardDidShow)

Keyboard.removeListener('KeyboardWillHide', -KeyboardDidHide)

}, [])

const -KeyboardDidShow = () => [

setIsFocused(true)

}

const -KeyboardDidHide = () => [

setIsFocused(false)

↓

```
const [textField, setTextField] = React.useState("")
```

```
const [keyboard, setKeyboard] = React.useState()
```

```
const onsubmit = () => {
```

```
route.params.updateValue('name', textField)
```

```
navigation.navigate('MeiaActivity')
```

### NameActivity Rewrite

```
import ...
```

```
export default function NameActivity ({ route, navigation, state }) {
```

```
const [textField, setTextField] = React.useState("")
```

```
const [keyboard, setKeyboard] = React.useState()
```

```
useEffect(() => {
```

```
navigation.addListener('focus', () => {
```

```
console.log('FOCUSED')
```

```
)
```

```
Keyboard.addListener('keyboardDidShow', -keyboardDidShow)
```

```
Keyboard.addListener('keyboardDidHide', -keyboardDidHide)
```

```
return () => {
```

```
Keyboard.removeListener('keyboardDidShow', -keyboardDidShow)
```

```
Keyboard.removeListener('keyboardDidHide', -keyboardDidHide)
```

```
}
```

```
, [])
```

```
const -keyboardDidShow = () => { setKeyboard(true) }
```

```
const -keyboardDidHide = () => { setKeyboard(false) }
```

```
const onsubmit = () => {
```

```
route.params.updateValue('name', textField)
```

```
navigation.navigate('MeiaActivity')
```

```
}
```

```

return (
  <View
    style={{  

      backgroundColor: 'white',  

      height: hp('100%')  

    }}  

>
  <View style={styles.headerSection}>  

    <TouchableOpacity  

      onPress={() => {  

        route.params.cancelActivity()  

      }}>  

      <Icon name='close' size={25} style={{ color: 'white' }} />  

    </TouchableOpacity> Size of the icon can also be passed as fontSize in the style object.  

    <Text style={styles.headerText}>Name </Text>  

</View> in Google Chrome pixel is the default unit that's specified for css  

<View style={{ paddingLeft: wp(10), paddingRight: wp(10) }}>  

  <Text style={styles.inputText}>Let's name your activity! </Text>  

  <Text style={styles.descriptionText}>  

    Here are some examples: Standup Comedy Show, Virtual Magic Session,  

    Museum Tour, etc.  

</Text>  

<TextInput  

  style={styles.textInput}  

  returnKeyType='done' // will automatically hide the keyboard.  

  blurOnSubmit={true} // If true, the text field will blur when submitted.  

  value={textField} // Callback that is called when the text inputs text changes  

  placeholder='Activity Name' // Changed text is passed as a single string argument  

  onChangeText={text => setTextField(text)} // To the callback handler.  

/>

```

```

</View> // It is a component to solve the common problem of views
<KeyboardAvoidingView // that need to move out of the way of the virtual keyboard
  behavior='padding' // Specify how to react to the presence of the keyboard.
  style={[flex: 1, justifyContent: 'flex-end', alignItems: 'center']}>
  > // Android and iOS both interact with this prop differently.

  <TouchableOpacity // Which type of button to display, disabled or not!
    style={disabled ? styles.disabledButton : styles.button} // Opacity is controlled by wrapping the children in an AnimatedView
    > // which is added to the view hierarchy. Because
      > // that this can affect layout.

      <Text // Keyboard
        style={disabled ? styles.disabledButtonKeyboard : styles.buttonKeyboard}
        > // textField.length == 0
        > // styles.disabledButton
        > // styles.button
        > // textField.length == 0
        > // styles.disabledButtonKeyboard
        > // styles.buttonKeyboard
      </Text>
    </TouchableOpacity>
  </KeyboardAvoidingView>
</View>
}

```

disabled = { textField.length == 0 }

onPress={() => onSubmit()}

<Text style={styles.buttonText}>NEXT</Text>

</TouchableOpacity>  
</KeyboardAvoidingView>

</View>

export default function NameActivity is in DescribeActivity.js. MeaningLocation is NameActivity is  
NoteActivity is TagActivity is TempPage.js.

CreateNewActivity.js imports NameActivity, initialRoute on this state equal to 'NameActivity' which is  
the monitor for routes on this props navigation and names within renderNavigation, within which the  
NameActivity component is rendered, which is the reason that NameActivity is imported.

② TagActivity is quite similar to NameActivity its process

③ MediaCard simply allows us to upload and display the photo

- ④ Describability simply allows us to set text fields
- ⑤ Categoryativity has selectable, TouchableOpacity controls
- ⑥ NestingLocation gives two choices: Virtual and Live, either or
- ⑦ Noteability is input notes!
- ⑧ Functionality is shows an action sheet
- ⑨ Rangeability is given as a date picker
- ⑩ Dayability is calendar container element
- ⑪ Seasonability - calendar scheduledComponents
- ⑫ Priceability: TextInputMask
- ⑬ Preemptability deriving date from props for the "add a new activity" flow preview!
- ⑭ ActivityCard.js initializes an ActivityCard component with a selectedIndex object, and renders stuff like props.previewImageURL, prop.name, and includes the MultiBold font family.
- ⑮ DraftabilityCard is in MyActivities.js, that is DraftabilityCard is not being pushed to this state. Draftability.
- ⑯ Mediability.js - at the same time, while configuration.navigate ("Mediability") is called within Mediability.js and within TempPage.js, and while Mediability is in fact called as a component within CreateActivity.js, Mediability itself consists of an AddMediaButton and a MediaPreviewCard; addMediaItem of AddMediaButton is a function and screenshot of MediaPreviewCard is a function, and this setState (in mediaPreview previous, mediaItem: 'Save', mediaItems: this.props.date.mediaItem, type: [...this.props.date.mediaItem].pop() type, url: [...this.props.date.mediaItem].pop(), url).
- ⑰ Scheduleability.js is imported into CreateActivity.js and exported using the same option only from Dayability and
- ⑱ TempPage.js demonstrates the effect of "screen focus" event listener for which the beginnings of a "create your own activity" flow have been fleshed out!

pages - /authPages - AuthNavigator.js, CreatePhonePassword.js, LoginAuth.js, SetPhoneAuth.js,  
SignUpAuth.js, SignUpAuth(PhoneNumber).js, SignUpPhoneAuth.js

\ NewPages - CreateActivityPages

MediaComponents - AddMedia.js, MediaCard.js, MediaPreviewCard.js

ScheduleComponents - AddScheduleSession.js, ScheduleSession.js

. /pages / authPages / AuthNavigator.js      CreatePhonePassword.js  
                  LoginAuth.js      SetPhoneAuth.js  
                  SignUpAuth.js      SignUpAuth(PhoneNumber).js  
                  SignUpPhoneAuth.js

/ NewPages / CreateActivityPages / MediaComponents / AddMedia.js  
                                                  MediaCard.js  
                                                  MediaPreviewCard.js  
                                                  AddScheduleSession.js  
                                                  ScheduleSession.js  
                                                  TempPage.js

  / ActivityCard.js  
    CategoryActivity.js  
    DayActivity.js  
    DescribeActivity.js  
    DraftActivityCard.js  
    DurationActivity.js  
    MediaActivity.js  
    MeetingLocation.js  
    NameActivity.js  
    NotesActivity.js  
    PreviewActivity.js  
    PriceActivity.js  
    RangeActivity.js  
    ScheduleActivity.js  
    SessionActivity.js  
    TagActivity.js

Pages / NavPages / ProfileScreens /

- EditProfile.js
- EditName.js
- MyProfile.js
- MyBookings.js
- PageNavigator.js
- ProfileNavigator.js
- ProfilePage.js

/

- ActivityComponent.js
- BanksAccount.js
- BookingsDetails.js
- CreateNewActivity.js
- History.js
- HistoryBookingsDetails.js
- MyActivities.js

./App.js

- app.json
- babel.config.js
- firebaseConfig.json
- GoogleService-Info.plist
- package.json
- package-lock.json

## Dooley Admin Dashboard

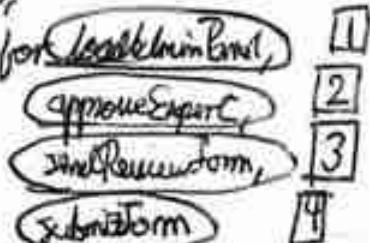
- Within a for-loop in JavaScript, the upper bound gets recalculated along with -!
- Separate graph for the Dooley application. ← Dooley App!
- Where is `baseConfig`?
  - `baseConfig` objects are being declared in multiple files!
- What is the reason behind no longer importing into `Logistics.js` and `Signups.js`, the assets `Google.png` and `Facebook.png`?
- How does the admin dashboard work for the regular non-mobile artists, and how can we make it work for the ones who sign up on and for the mobile `DooleyBiz` app?

Key Features: \* import the entire Firebase library.

(We have `firebaseConfig = { apiKey: authDomain: databaseURL: projectId: storageBucket: messagingSenderId: appId: }`)

initializing with `firebase.initializeApp()`,

and then \* on the component: this binding for `localhost:Port`,



Functionality: the first thing we should do is this `localhost:Port` which occurs on `componentDidMount`.

Are there thumb creators? How is the database interacted with?

[3] `SendReviewForm` makes an `ajaxPost` request

`POST /sendReviewForm,`

`/process.env.API_URL + '/sendReviewForm'`

`git diff HEAD^HEAD`

`localId: booking.localId,`

`reviewerName: booking.requesterName,`

`reviewerEmail: booking.requesterEmail,`

`localName: booking.localName`

`},`  
[with `individual: true`]

① localAdminPanel = await () => {  
try {

var response = await axios.get()

process.env.API\_URL + '/admin/loadDashboard',  
[withCredentials: true]

), response.data has boolean isAuthenticated

and has requestSettings equal to response.data.settings.

response.data.isAuthenticated is a thing!

and we sent that to state.

④ submitForm = async e => {

POST /auth/dashboardLogin,

var response = await axios.post()

process.env.API\_URL + '/auth/dashboardLogin';

username: form.elements.email.value,  
password: form.elements.password.value

[withCredentials: true]

I believe this is just to set the cookie!

② async approveExport(expert) {

var response = await axios.post

process.env.API\_URL + '/user/approveProfile',

email: expert.email

POST /user/approveProfile,

[withCredentials: true]

why do we need to send  
the email here?

this.localAdminPanel()

③ What does the tag withCredentials: true mean? withCredentials: true is included  
in all of our queries

- ① GET /admin/loadDashboard
- ② POST /user/approveProfile
- ③ POST /sendReviewForm
- ④ POST /auth/dashboardLogin

## Signup Page, Frontend:

Change title of field to First Name, Last Name? split in the middle on same line.

On the ~~submit~~ submit and accept a lastname field.

submission = async e  $\Rightarrow$  {

pages/signup.js

var form = e.target.elements ...

[var lastname = form.lastname.value]

① How are we going to receive the isCustomer and isLocal boolean variable being submitted to the database if we are even using them?

if (password == confirmPassword) {

var requestBody = {

userId: email,

data: {

isCustomer: true,  
isLocal: false,

email: email,

name: name,

[lastName: lastname], ...

authType: 'email',

profilePic: 'https:...'

}

authType is only ever used within the createUserWithEmailAndPassword function of pages/Auth/signUp.js. index.js once not been used!

What are the other authTypes?

Will user still be able to sign in given that profile picture is supposed to be mandatory?

★ firebase Do we call this twice. Shall we call auth() twice?

auth() according to the docs, creates a new account by passing the new

createUserWithEmailAndPassword(email, password) user's email address and password.

then async response  $\Rightarrow$  {

to createUserWithEmailAndPassword.

firebase

auth()

signInWithEmailAndPassword(email, password)

```

.then(async response => {
  var response = await axios.post(
    process.env.API_URL + '/auth/setLogin',
    { user: { email: email } },
    { withCredentials: true }
  )
  var response4 = await axios.post(
    process.env.API_URL + '/signup/saveUserToDatabase',
    requestBody,
    { withCredentials: true }
  )
})
)
)

```

★ Signup Page assume component different  
 checks that we're logged in and  
 pushes us to `goobyby` if so and if not catches this as an error and  
 does not performing action.

These are the elements being pulled off of the form:

`form = e.target.elements` through the event!  
`form.email.value`   `form.password.value`   `form.repeated.value`  
`form.name.value`.

`<Form.Control`

`className='signupPageLastName'`  
`name='lastName'`

`type='text'`

~~What are the types besides text and password?~~

`placeholder='last name'`

~~What is controlId for?~~

this needs to be wrapped in

~~<Form.Group controlId='formBasicEmail'>~~  
~~</Form.Group>~~

1>

~~Bootstrap supports all the HTML input types: text, password,  
 datetime, datetime-local, date, month, time, week, number, email, url, search, color and color.~~

~~controlId is only ever needed so that `FormControl` has an id and `Form.Label` has an htmlFor  
 to which we can refer the label styling and text location to the proper  
 Form Group.~~

# Google Sign Up

Last Piece of Code:

```
<GoogleLogin
```

```
    className='signupPageSignupWithGoogle'
```

```
    clientId=
```

```
        buttonText='Sign up with Google'
```

```
    onSuccess={response=>this.onSuccess(response)}
```

```
    // onFailure={response=>console.log(response)}
```

```
    cookiePolicy='single-host-only'
```

/>

```
onSuccess=async state=>{
```

```
    var reqBody = {
```

cancel delete?

```
        accessToken: state.accessToken,
```

```
        user: state.profileObj ★
```

```
}
```

```
const credential = firebase.auth.GoogleAuthProvider.credential(
```

```
    null,
```

```
    state.accessToken
```

```
)
```

```
var user = state.profileObj
```

```
firebase
```

```
.auth()
```

```
.signInWithCredential(credential)
```

```
.then(async response=>{
```

```
    var response = await axios.post(
```

```
        process.env.API_URL + '/auth/setlogin',
```

```
        { user: { email: state.profileObj.email } },
```

```
        { withCredentials: true }
```

```
)
```

```
var regBody = {  
    userId: user.email,  
    slate: {  
        idLocal: true,  
        email: user.email,  
        authType: 'google',  
        name: user.givenName + ' ' + user.familyName,  
        profilePic: user.imageUrl.slice(0, -6)  
    }  
}
```

unuseel

```
var response2 = await axios.post(  
    process.env.API_URL + '/signup/savetherDatabase',  
    regBody,  
    { withCredentials: true })
```

```
} )  
    Router.push('/goolebiz')
```

End of Firebase database method chaining block!

- Therefore within the Google Signup \*this is called, the success function.

① Grab the Firebase credentials

② Axios POST '/auth/setLogin'

③ Axios POST '/signup/savetherDatabase'

④ Router push to '/goolebiz'

- And within the regular signup: \*this is called, the submitform function.

① await axios POST '/auth/setLogin'

② await axios POST '/signup/savetherDatabase',

everything after that is usual warnings!

- With that in mind we are going to change the regular signup for now.  
and what happens on the auth/setLogin route?

- app.post('/auth/setLogin',  
 simply adds the user email to req.session.user.   
 Literally: req.session.user = {email: user.email} )  
 this will subsequently be used within the Signup/SellerToDatabase   
 'POST' request!
- app.post('/signup/sellerToDatabase'  
 async (req, res) =>  
 data = req.body.data  
 var user = req.session.user  
 user.id = req.body.userId

- On the database, first name should be required and last name should be required!  
if they are not present then display a swal alert!  
this requirement should be carried out among the API routes!

Errors:

- ① Please confirm both passwords
- ② When the firebase authentication <sup>or</sup> user with email and password creation fails
- ③ When the firebase authentication, sign-in, so axios post requests fail

★ swal({

title: "Hold On!",

text: "Please confirm both ~~both~~ missing fields"

"Please ensure [missing fields] are completed.",

icon: "warning",

button: "OK",

});

★ the thing about swal is that it doesn't have to be done as part of a try/catch error block; it can be done within an else block.

## \* Error codes matter from the frontend perspective!

What does error code 400 mean?

that the server cannot or will not process the request due to something that is perceived to be a client error

e.g. malformed request syntax,  
invalid request message framing, or  
deceptive request routing.

~~Custom HTTP headers for decision making.~~ When such a header is absent, stale, incorrect, or otherwise not processable.

As a general rule, anything malformed or "wrong" coming from the public Internet can safely be assumed malicious (or "deceptive").

- delimiters, request line, headers, length prefix, end of header body
- It could be argued that having incorrect data in your request is a syntax error, even if your actual request at the HTTP level (Request line, headers etc) is syntactically valid.

• the trigger block is working!

- getting profile data occurs right after Error: Please ensure last name is complete.  
 ⇒ `app.get('/user/getProfileData')` is subsequently called
- maybe don't immediately push to localhost right after?  
 surely, yes try removing localhost.

★ We don't even get to localhost.



Hold On!

\* Request failed with status code 400

2 API routes: /auth/setLogin

POST

/signup/saveUserToDatabase

function

.auth()

.createUserWithEmailAndPassword(email, password)

occurs before the API routes are POSTed to:

↳ are for field validation, and return messages for success or failure.

Resulting in the error message:

The email address is already in use by another account.

★ What does the createUserWithEmailAndPassword function actually do?

I suppose we should console log the response object from

• SignInWithEmailAndPassword.

★ email shows up on the Authentication Tab on Firebase!

② On the Authentication Tab: We can create:

• createUserWithEmailAndPassword

(What is the counterpart? For example:

• deleteUserWithEmailAndPassword ?

This should be avoided because otherwise customers and experts  
will encounter the ①

Holy Oh! error code auth/email-already-in-use

The email address is already in use by another account  
Email alert.

instead of being able to sign up!

★ The authentication tab is for storing user unique identifier,  
email identifier, provider (email or by Google), date of creation,  
date of signing and does not store

any other information about the user!

★ In that case since not all of none of the response variables are actually  
useful, we can make the API POST requests right away!

• move the order around

and then the method chaining might stop right there!

✓ it worked!

submitForm to continuity if (password == confirmPassword)

Once reqBody is declared to be a variable,  
try/catch block is initiated with

await axios.post - /auth/setLogin

\* /signup/saveUserToDatabase

before anything to do with Firebase!

\* /auth/setLogin does not do anything special

other than extracting values from req and from res.

specifically, req.body.username → res.session.user

res.status(200).json({ success: true })

\* /signup/saveUserToDatabase: the first try/catch block does field validation  
and ③ references on req.body.data against

① db.collection('users').doc(userEmail)

② db.collection('usernames').get()

this one is used for /user/usernameSignup

and therefore:

① Determine 'user already exists in database'

② Determine 'thank you google local for signing up as a user'  
'thank you google user for signing up as a local'

③ And finally what is this?

~~return response~~ await db.collection('users').doc(userEmail).set(data) } and finally we return res.json({ res.status(200).json({ success: message }) })

this is the collections tab: the very first tab  
on the Firebase database

this is the second tab on Firebase!

and this last statement indeed does set an item  
at that document location: the third column

Request failed with status code 400 →

① The error caught on the frontend

~~is not the~~ is different from

The error being thrown ~~for~~ on the backend,

Sent back as part of the 400 w. status reply!

How to include the response returned from the server/backend  
as a part of the error thrown on the frontend?

We did not get a response ~~object~~ from the 400 port.

The error is caught before the response is caught.

The response doesn't actually ~~exist~~ on the frontend,  
is shown through the use of an ~~interim~~ log/ketab block.

② Restart the server after making changes!

③ How to send an error like 400 back with an  
accessible response body?

④ What is the type of errors thrown in node.js  
given that they have to be converted to ~~text~~?

from string to number?

⑤ data.password = await bcrypt.hash(req.body.data.password, saltRounds)

What does this mean?

⑥ Are there any advantages to the use of named response variables?

• How does Google decide what's the first thing you see on the page?

• How does the Google search algorithm work?

• View Profile Page: Google Logo.

Changed Google Text → Google Logo!

How does the Edit Profile page interact with the database?

There is no edit name text field on the Edit Profile page!

Edit Profile is governed by pages/goolebiz/index.js

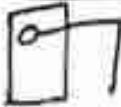
Search for Best password

→ this would impact the username of anyone/expert who ever changes their name, so  
maybe idea: ① create an entirely new user

② edit the username field simultaneously!

③ experts should be able to change the auto-generated username.

account\_settings.png



Main

menu

reviews

Logo

my name → my account

Edit Profile

My Bookings → My Bookings List

History → History List

Payment → Payment Codes and Methods

Contact Us → Leave Feedback

Be a Gooley Local → Become Gooley Local

bizaccount.png



About Me  
Interests ★  
Phone  
Email

New Bank Account  
Edit Activity

Background Check Info

Review/Submit → Activities/Thank you! → Past Activity

Report Incident → ex. lost item

- menu bar
- 3) bizappaddactivity.png all kinds of stuff for the "add a new activity" flow!
- 4) bizapploginsignup.png "Forgot Password" flow and "Signup" flow are very similar!
- 5) bizappmanagebookings-1.png  
New Bookings, Traveller's Responses, Upcoming Activity, Cancel Booking
- 6) bizappmanagebookings.png  
New Bookings, Traveller's Responses, Active Now, Cancel Booking
- 7) browse\_activities\_and\_make\_booking.png  
Destination Notifications Date/Time  
Payment Method
- 8) my-bookings\_and\_notifications.png
- 9) onboarding\_flow.png

# Publishing to TestFlight

expo publish build the bundle

index.ios.js      index.android.js

- /Users/leenyadish/react-native-gosly-app-front-end/\*\*/\*

expo build:ios

will ask you to log in with Apple ID.

\* Will you provide your own Apple distribution certificate?

(yes) ?   No ✓

expo.dev/testflight-status

NPM install -g eas-cli

~~Do I want access to Searh?      Expo Account?~~

~~to check the build status?~~

expo login

selma@gooley.com

gooley2020!

to eas submit --platform ~~ios~~ ios

→ Need access to EAS Priority Features

Unless you have a macOS device

★ docs.expo.io/distribution/uploading-apps/ ★

## Transporter App

ipa file comes from <https://expo.io/artifacts/>

on the transporter

Dockey  
1.0.0 (223)  
1273758598

Before expo publish,

remember to change the version  
and the build number!

Using My Apple account and  
Soham's Expo account

\* \* \*

and then check the builds on  
[appstosecret.apple.com](https://appstosecret.apple.com)

takes a long time to build!

Push notification Simulator?

right now, I'm publishing to the App Store.

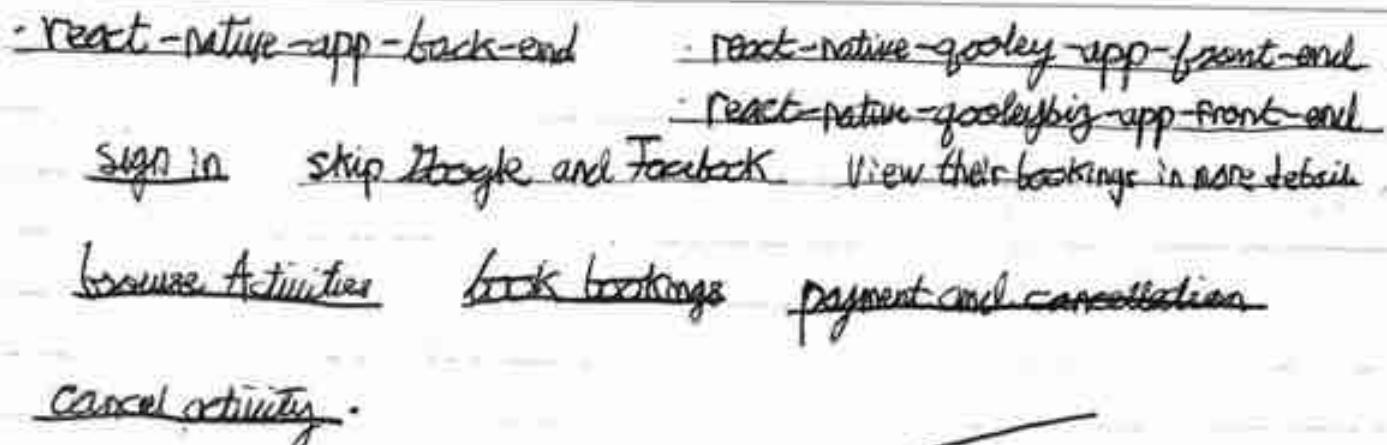
expo dev/turtle status

Does expo process builds  
One by one?

Very slow build speed!

once it passes the verification step...

Does Version 1.0.0 translates to 1.0 on App Store  
Correct?



### Key functionalities for the MVP Gooley Customer

account-settings.png    What do we need for the app to be reviewed  
 browse-activities-and-make-bookings.png the Apple App Store?  
 my-bookings-and-notification.png  
 Onboarding-flow.png

### App Store Connect Email

Dear Gooley, LLC,

The following build has completed processing:

Platform: iOS

App Name: Gooley

Build Number: 2.2.4

Version Number: 1.0

App Store: Gooley 01

App Apple ID: 1234567890

the internal # of builds that have been made until the version #

You can now use this build for TestFlight testing or submit it to the App Store.

- a difference in build number represents a recompilation of the same source.
  - Release 1.0.0 might be build 542. It took 542 builds to get to a 1.0.0 release.
  - Release 1.0.1 might be build 578.
  - Release 1.1.0 might be build 694.
  - Release 2.0.0 might be build 949.

I want to go back from TestFlight to the simulator

(appstoreconnect.apple.com/apps

Am I using the right (most recently  
edited code)

GitHub repository?



expo build:ios

? Choose the build type you would like:

> - Use device-keys. Return to submit.

or > archive - Deploy the build to the store

> simulator - Run the build on a simulator

- What does the expo build:ios command actually do?

? building JS bundle takes a long time!  
What does the build and bundle process look like over time?

long screenshots (multiple screen heights)

We decide between ...simulator.tar.gz and .ipa because we are the ones using the Expo CLI and decide in the terminal.

npm install  
package versions



npm install generator@~23.14.x

npm install generator@^20.x.x

latest patch version

latest minor version

npm ls --

Npm view → Version

expo install \_\_\_\_\_ will install automatically the right version  
check There is a new version of expo-cli available  
npm install -g expo-cli

install expo-cli globally using the package manager of your choice ✓

A. import Constants from 'expo-constants' A

What is in here? Constants, Device exists,  
what else is there?

almost got the iOS flow

Want: to be able to run the app on a simulator.

### Stripe Migrations

① react-native-godley-app-front-end

package-lock package JSON

ActivityDetails ~~AltCardComponent~~ BookActivity

BookingDetails PayActivity PaymentMethods PreviewBooking

JS

② react-native-godleybiz-app-front-end

BankAccount

BookingDetails

HistoryBookingDetails

JS

for OAuth

- stripe cart creation and API route are commented out!

③ react-native-app-back-end

package-lock ~~JSON~~ package

VS Code full file

Screenshot

see CodeSnap VS Code

extension!

?) We have ~~two~~ package.json files!

and index.js contains all Stripe-related functions!



① react-native-app-back-end  
· stripe version 8.135.0 in functions/package.json  
in package.json and package-lock.json

· stripe version 8.141.0 in ./package.json  
and ./package-lock.json

what do we do in our interaction with Stripe?

functions/index.js  
const stripe = require('stripe')('sk-test-74...L3')

APP.GET /stripe/connect/oauth

· Send the authorization code to Stripe's API.

function saveAccountID = save the connected account ID from the  
response to your database.

APP.GET /stripe/getCustomer

APP.POST /stripe/addPaymentMethod

· calls stripe.customers.createSource  
based on req.session.user

POST /stripe/deletePaymentMethod

· calls stripe.customers.deleteSource

POST /stripe/payForActivity based on req.session.stripeCustomerId

and then

## ② react-native-paypal-app-front-end

"expo-payments-stripe", a key from package-lock.json  
"stripe-client", which contains NPM registry addresses with integrity.  
Version 8.2.1

Version 1.1.5

- interface with 'expo-payments-stripe': we were going to do something with that, I suppose and then chickened out at the end

stripe.createToken and

global API-URL + '/stripe/payForActivity'

focus on AddCartComponent

var stripe = require('stripe-client')('pk-test-NDRV...7B')

When is Stripe eventually called?

\* Var card = await stripe.createToken(information)

\* Var token = card.id ← is the card valid?

\* this.props.addCard(token) ← Does the this.props.addCard

this.setState({isBillingCard: false }) ← function complete its mission?

React Native | Went console log!

Error: listen EADDRINUSE: address already in use : 19000

it appears that npm is using expo-cli@4.9.3

fixed this error

before expo start, so that we don't get it during expo start process.

So Expo Start

react-native-focally-app-front-end

pages/NavPages/PremiumBookings

This global API-URL + '/stripe/payForActivity'

prerequisites:

- expo-payments-stripe  
- stripe-client

package.json

pages/NavPages/ActivityDetails.js

import { PaymentsStripe as Stripe } from 'expo-payments-stripe'  
... and then NOTHING... /stripe/payForActivity

is commented out.

pages/NavPages/AddCardComponent.js

var stripe = require('stripe-client')('pk-test-NDR...7B')

PaymentsStripe is imported and not used

instead handleButtonPress instead

var card = await stripe.createToken(cardInformation)

pages/NavPages/BookActivity.js

also here

some

var card = await... is commented out

pages/NavPages/BookingDetails.js

PaymentsStripe is imported and not used

some card = await is commented out

pages/NavPages/BuyActivity.js

var stripe = require('stripe-client')('pk-test-NDR...7B')

What are you planning to do with PaymentsStripe in AddCardComponent,

BookActivity, BookingDetails, BuyActivity,

- async refreshCard() has var response = await axios.get(global.API-URL + '/stripe/getCustomer', { withCredentials: true })

- addCard has var response = await axios.post(global.API-URL + '/stripe/addPaymentMethod', { cardToken: token }, { withCredentials: true })

pages/NavPages/PaymentMethods.js

- Var stripe = require('stripe-client')('pk-test...9B')

- async refreshCard() Var response = await axios.get(global.API-URL + '/stripe/getCustomer');

- handleButtonPress has var card = await stripe.createToken(cardInformation)

• Yes, what is Payment Stripe used for?

/pages/NovPages

: PreviewBookings

API URL

/stripe/payForActivity

: ActivityDetails

: AddCardComponent

handleButtonPress invokes stripe.createToken(information)

for which

var stripe = require('stripe-client')('pk-test...')

: BookActivity

: BookingDetails

: PayActivity

var stripe = require('stripe-client')('pk-test...')

async refreshCard():

API URL

/stripe/getCustomer

addCard API URL /stripe/addPaymentMethod

: PaymentMethods

var stripe = require('stripe-client')('pk-test...')

async refreshCard():

API URL

/stripe/getCustomer

handleButtonPress invokes stripe.createToken(information)

What are the principal elements of the API URLs?

/stripe/payForActivity does

POST > /stripe/payForActivity

① Grab the user from req.session servante

② Grab the activity from req.body

③ Turn the activity's price into an application fee  
using a proprietary formula

$$\text{application\_fee} = \frac{\text{activity\_price}}{10} + \text{activity\_price} \cdot 0.029 + 30$$

④ From req.body      1) if customerToken is not available,  
then

charge = await stripe.charges.create

amount: activityPrice,

currency: 'usd',

customer: customer.id,

applicationFeeAmount: applicationFee,

transferData: {

destination: 'acct-1...'

on the other hand, 2) if customerToken is available,  
then

charge = await stripe.charges.create

amount: activityPrice

currency: 'usd'

customer: req.body.customerToken,

applicationFeeAmount: applicationFee,

transferData: {

destination: 'acct-1...'

return res.status(200).json({charge: charge})

an asset is not banking!

### GET /stripe/getCustomer

- ① Take the current req.session user
  - ② Get data from db.collection('app-users')  
.doc(user.email).get().data()
- and then ③ stripe.customers.retrieve(customerId,  
async function (err, customer) {  
if err then try/catch await stripe.customers.create  
({ email: user.email })  
await db.collection('app-users')  
.doc(user.email).update([{ stripeCustomerId : customerId }])

After the try/catch block is completed,  
whether or not there was err then

place req.session.stripeCustomerId = customerId

return res.status(200).json({ cards: customer })

### POST /stripe/addPaymentMethod

if req.session.stripeCustomerId is true,

- ① stripe customers.createSource

req.session.stripeCustomerId, { source: req.body.cardToken },  
function (err, card) {

if (err) { return res.status(400).json({ error: err }) }

```
• imports React, React-Native, Expo-Font, Expo-Linking, Axios,  
  React-Native-Responsive-Screen, Firebase, NavigationContainer,  
  createStackNavigator; createDrawerNavigator, expo-notifications,  
  expo-permissions,  
  expo-constants
```

// imports SignUp Pages Login Page

App Pages

global.API\_URL = 'https://us-central1-gookeybig.cloudfunctions.net/app'

App

loadFontSync async await Font.loadAsync(customFont)

RegistersForPushNotificationSync

gets permission AS LONG AS isDevice  
and generates Alert messages, checkLogos for  
Firebase.auth().onAuthStateChanged for which we can Axios POST auth/login  
and for which we have setLogout and setLogout, and what does our  
componentDidMount do?  
await this.\_loadFontSync, await this.\_checkLogos  
this.\_RegistersForPushNotificationSync()

render()

if this.state.dataLoaded {  
 return

<NavigationContainer>  
<Stack.Navigator>

! this.state.isLoggedIn ? (

<Stack.Screen name='Signup1'>

<Stack.Screen name='Signup2'>

Signup3

Signup4

LoginAuth

SignupAuth

SignupForgot

CreationReward

SetPassword

```
(| < Stack.Screen name='PageNavigator'>  
initialParams, component
```

else if this.state.detected is not loaded then

```
<View <(image source={assets/splash-screen.png}>>>
```

{props =>} <EditProfile {...props} data={this.state.detected} />

<Stack.Screen>

→ ProfileNavigator

→ PageNavigator

```
* <MyDrawer.Screen name='ProfileNavigator'>
```

props =>

<ProfileNavigator

refreshDrawer={this.updateDrawer}  
{...props}

data={this.state.detected}

/>

```
)</MyDrawer.Screen>
```

<Stack.Screen

name='PageNavigator'

initialParams={OnLayout: this.setLayout.bind(this)}  
component={PageNavigator})

/>

i have never seen one, but I know they actually exist!