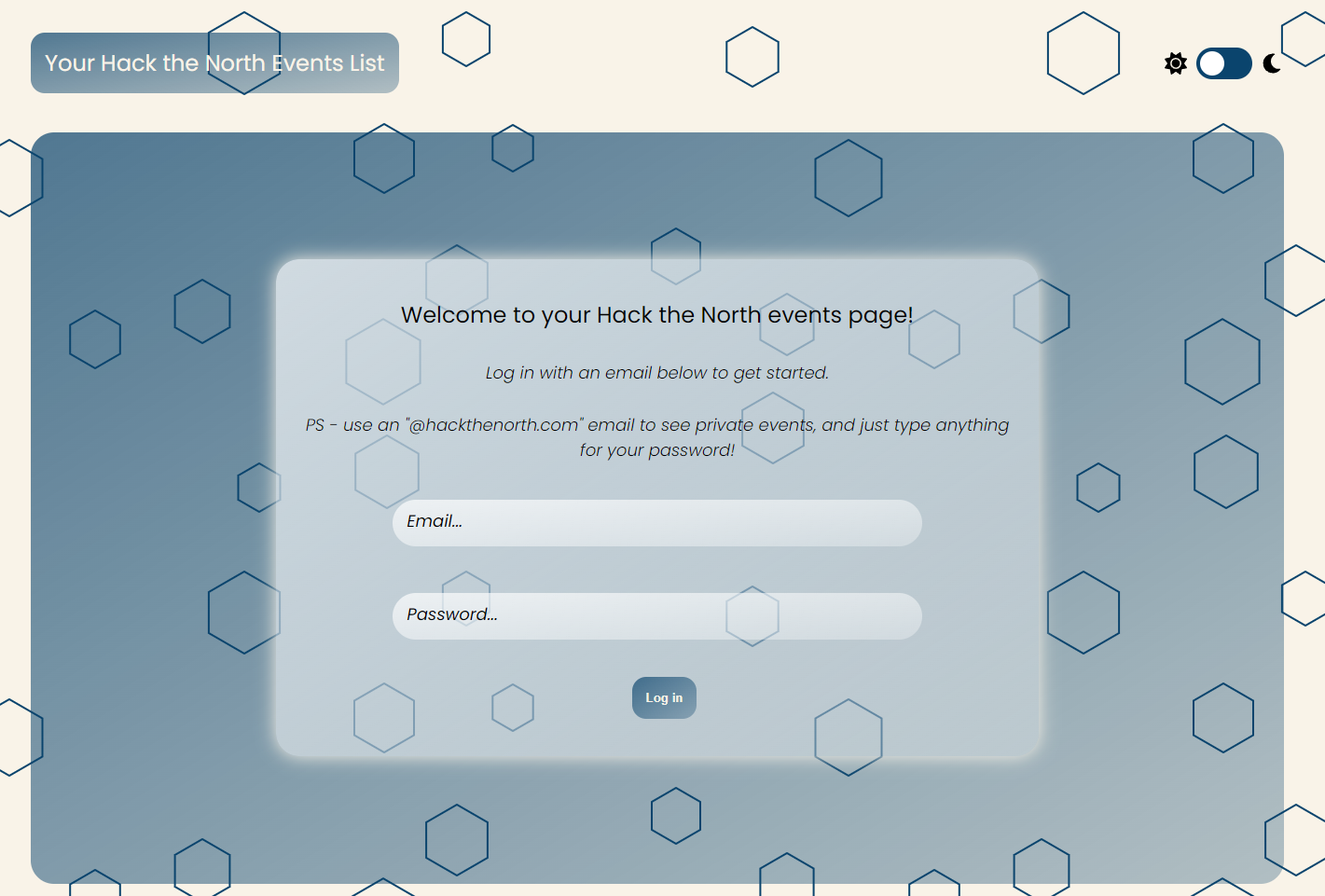
Hack the North Front-end Coding Challenge

Brayden Royston

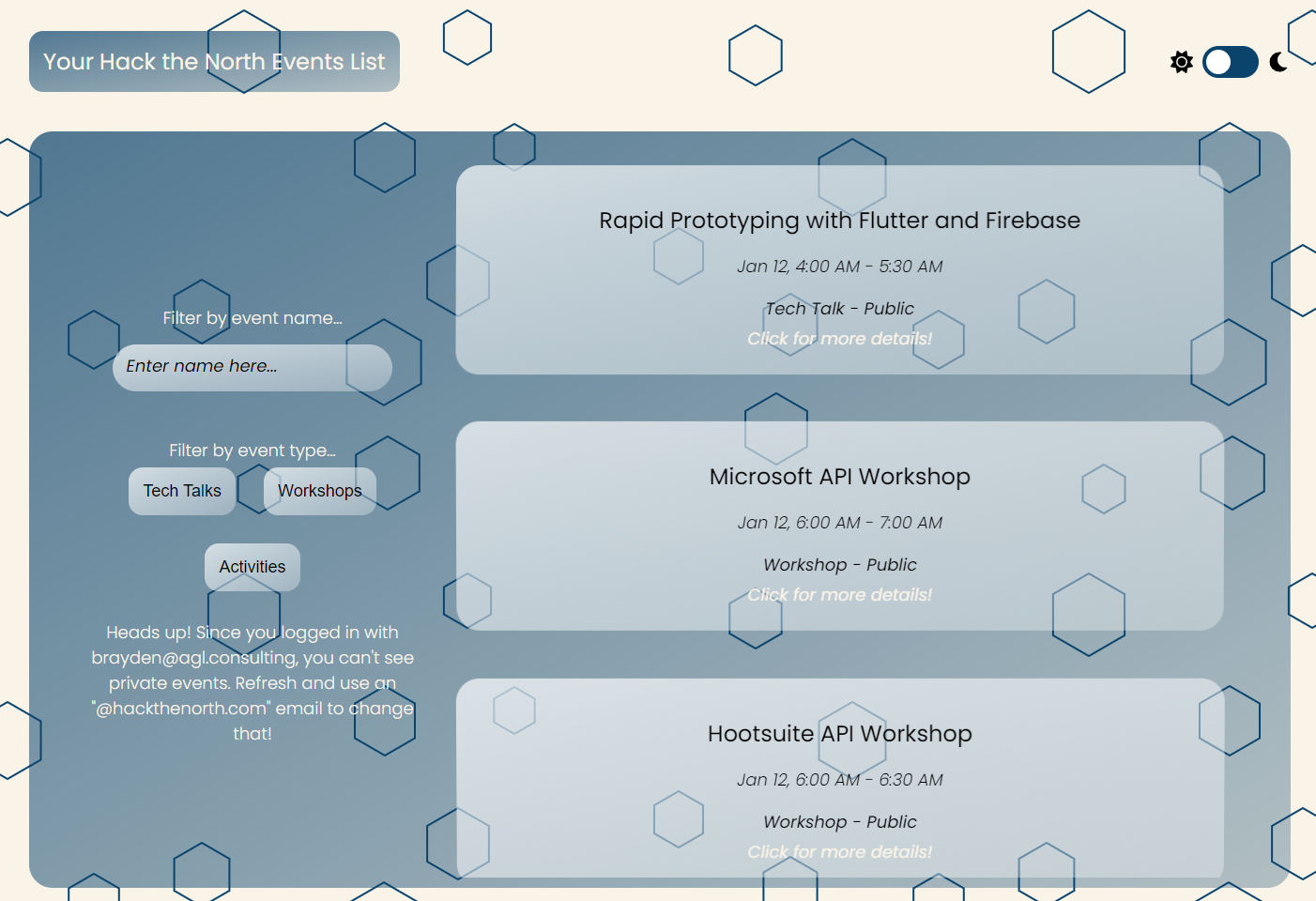
***Part Zero – Using my app***

*Step One: Logging In*



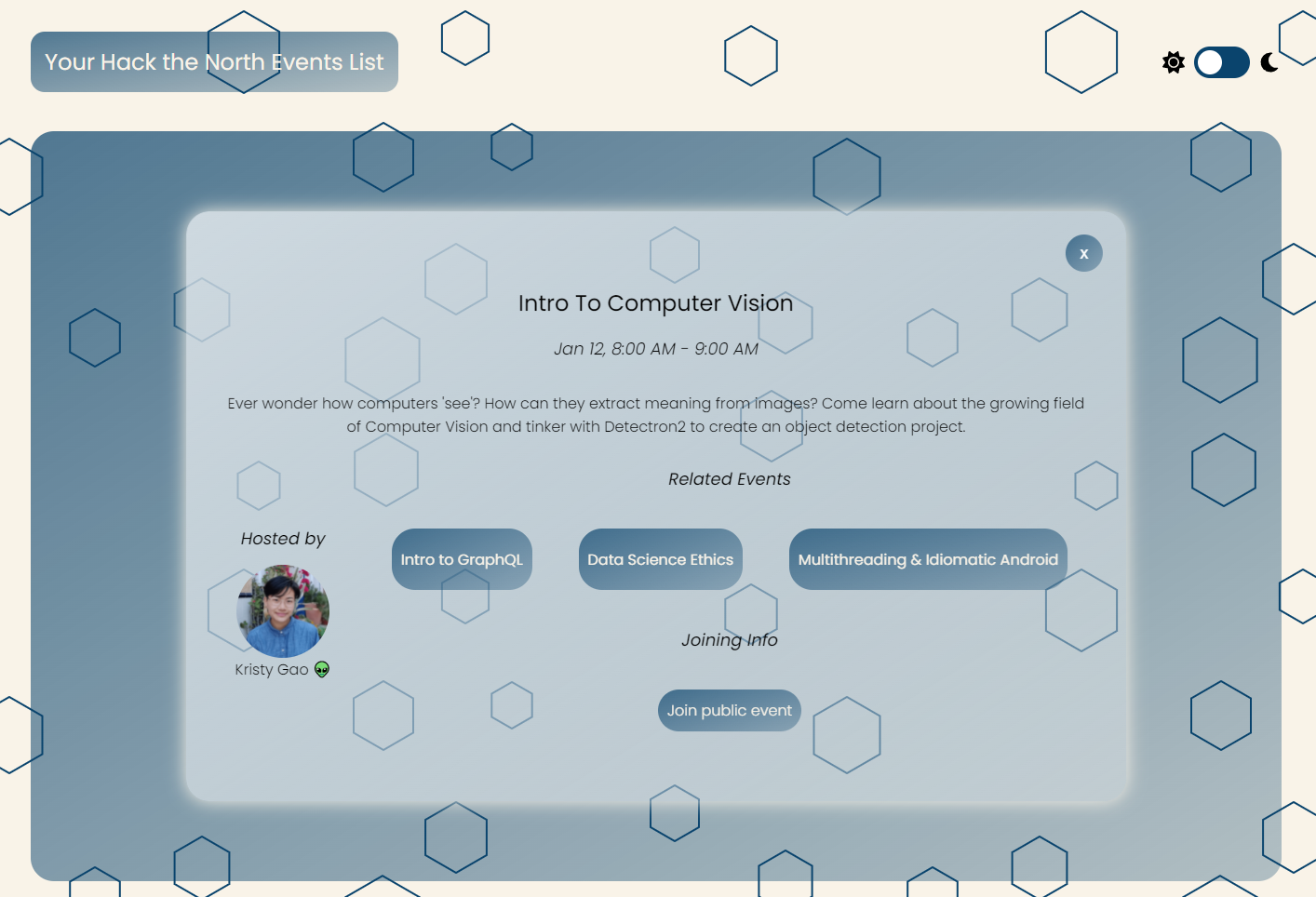
The user is first pushed to a log-in page where they can enter an email and a password to enter the app. I didn’t set up any real authentication for this application, so I’ve made it so that if you log in with an @hackthenorth.com email then you are given access to the private events, and if not you only will have access to the public events. Additionally, the user can toggle lightmode/darkmode at any time via the toggle switch in the upper-right-hand-corner of the app.

*Step Two: Browsing Events*



After logging in, the user is then shown the list of events and can scroll through them to view the ‘highlights’ of each event (Name, Date, Event Type, Event Perms). The user has access to a control panel on the left side of the screen, which allows them to select filters so that they can view events by type, and also allows them to type in a custom filter of their own if they want to really narrow down their search. For the purposes of this app, I also give a brief message to tell the user whether or not they’re verified and can see private events.

*Step Three: Diving into Event Details*

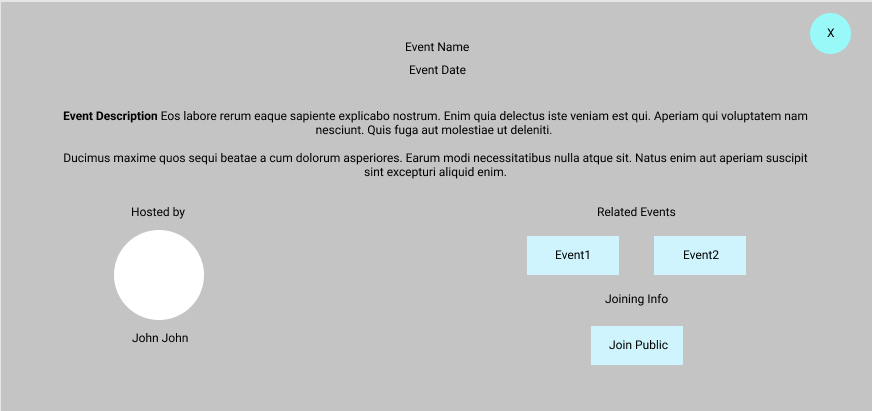


When a user selects one of the event cards, they are shown an EventModal that gives them some more details about the event that they selected. From here, they can read the event description, view who is hosting the event, and click on any related events that might catch their eye (related event buttons send them to another EventModal that displays the event they clicked on. Note that related events that are of permission == private do not get show if you are an unverified user). In addition, they can join the vent via the buttons at the bottom of the modal (only the public button is shown if the user is not verified). The user can press the close button on the top right of the modal at any time to return to the events list view screen.

***Part One – Development Process***

When it came to designing this application, the first step that I took was familiarizing myself with the event data structure by making a quick call to the events endpoint. I took a look at all of the fields that were within an event, and then began brainstorming about how I could display this info to a user in a way that allowed them to easily navigate and find the events that they cared about.

Admittedly I’m far (very far) from an expert in design, so I begun this project by sketching a brief blueprint of what I wanted the UI to look like. I knew that from the start that I wanted this app to have a glassmorphism theme throughout it, and I also wanted to take the single-page-application route because I personally feel that those are often the easiest to navigate and use, especially when it comes to applications for purposes such as hackathons. For the colour scheme I took some inspiration from the colour palette on the HTN 2021 website, and from there I was pretty much ready to start coding. This thought process eventually transferred into a super low-fidelity Figma, which allowed me to plan out how I wanted the layout on my main ‘Event Card’ to look:



Once I had an idea in mind for how I wanted things to look design and layout-wise, I began developing the app. The tech that I used for the app was as follows:

**Main Application**: ReactJS

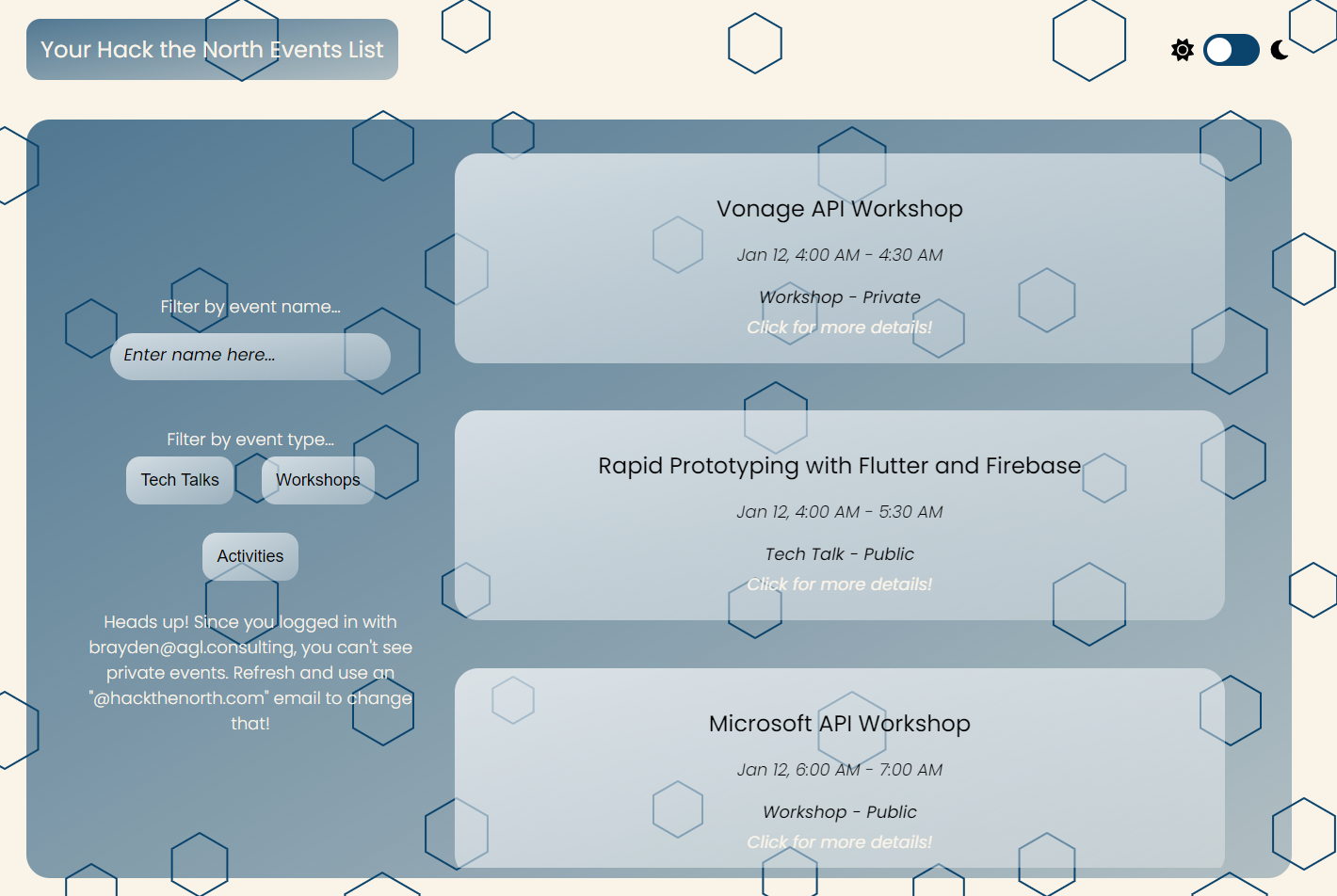
**Libraries:**

* Axios
  + http client
* react-spring
  + provides a physics engine that allows for some interesting effects used on the EventCard and EventModal
* styled-components
  + allows for styling of components using CSS in the same file as the rest of your JSX
  + I come from a VueJS background so I use this library mostly due to the fact that I enjoy having all my styling, JS, and JSX/Markup in one place.
  + Also allows for easy props-based conditional styling which was used to implement a darkmode/lightmode throughout the app

**Problems & Solutions**

I think that the main problem that I encountered while creating this application was the design aspect. I think that when given a Figma I can fairly comfortably code up pretty much anything, but when it came to actually designing the layout and styling of this application on my own I really struggled with making things look appealing (as you could probably tell LOL). This design process was actually a super interesting challenge for me and it’s something that I really want to work on improving for the future. In terms of short-term solutions to this issue while I was coding, I unfortunately had to cut a lot of corners when it came to design and just stick to things that I felt got my message across and made the user’s life as easy as possible. The EventModal is something that I really wish I could have spent more time experimenting with, because I wasn’t satisfied with how the layout and overall design of that component turned out.

**Code I like**

To be perfectly honest with you I think that this project wasn’t really my best work due to it being a bit of a time crunch with work/school/etc., but one part that I’m relatively happy with was the events list view portion of the project show below:

This was my first time taking a stab at glassmorphism and I think that this page does a decent job at actually bringing that style to life. It was a good time experimenting with react-spring to apply the ‘floaty’ effects to the event cards, and I was pretty happy with this UI overall. The actual file that contains this portion of the project is EventsView.js if you want to check it out.

***Part Two – Improvements***

There are countless improvements that could be made to this app to make it hacker-ready. Here’s a few that I wish I had time for:

* responsiveness and cross-device compatibility
  + I was able to implement the app to work on my desktop and my laptop, but I didn’t have time to test it on a variety of devices and screen-sizes. At the moment, the app isn’t functional on a mobile device since I coded all components from scratch and didn’t have the normal cross-device compatibility that comes out of the box with something like bootstrap. Getting this responsivity aspect correct would be a must if this was to become a real product.
* Proper authentication
  + needs to happen, private event data is sensitive and would have to be protected behind a proper authentication wall
* store cookies to memorize app state
  + probably not super useful in an app as small as this, but a good quality of life improvement for the user nonetheless
* use redux for some things:
  + could have been useful for global access to lightmode/darkmode state
  + could have been useful for events management
  + overall if I was working on this app any longer, I would have used redux because it makes the developer life easier in a lot of ways.
* styling
  + some of my components are super cluttered code-wise, definitely would have spent time improving that and providing better documentation
  + also from a design perspective I really wish I spent more time experimenting and finding layouts that I actually thought were visually appealing and effective, there’s a lot of work to be done here in my opinion.
* functionality
  + I was able to implement most of the extra functionality that I wanted to such as filters, a search bar, and a lightmode/darkmode, but I think that it would have been cool to also add an interested/going state for each user on each event, this would allow the HTN team to have a better idea of how many people are going to show up to each event which would be great.

***Part Three – Other thoughts***

Overall this was a fun challenge and I enjoyed it! It was a fun time for me to experiment with some things that I don’t actually have a lot of experience in such as the design process, and using new libraries such as react-spring. Thanks for reviewing my application and I hope to be in touch soon :D