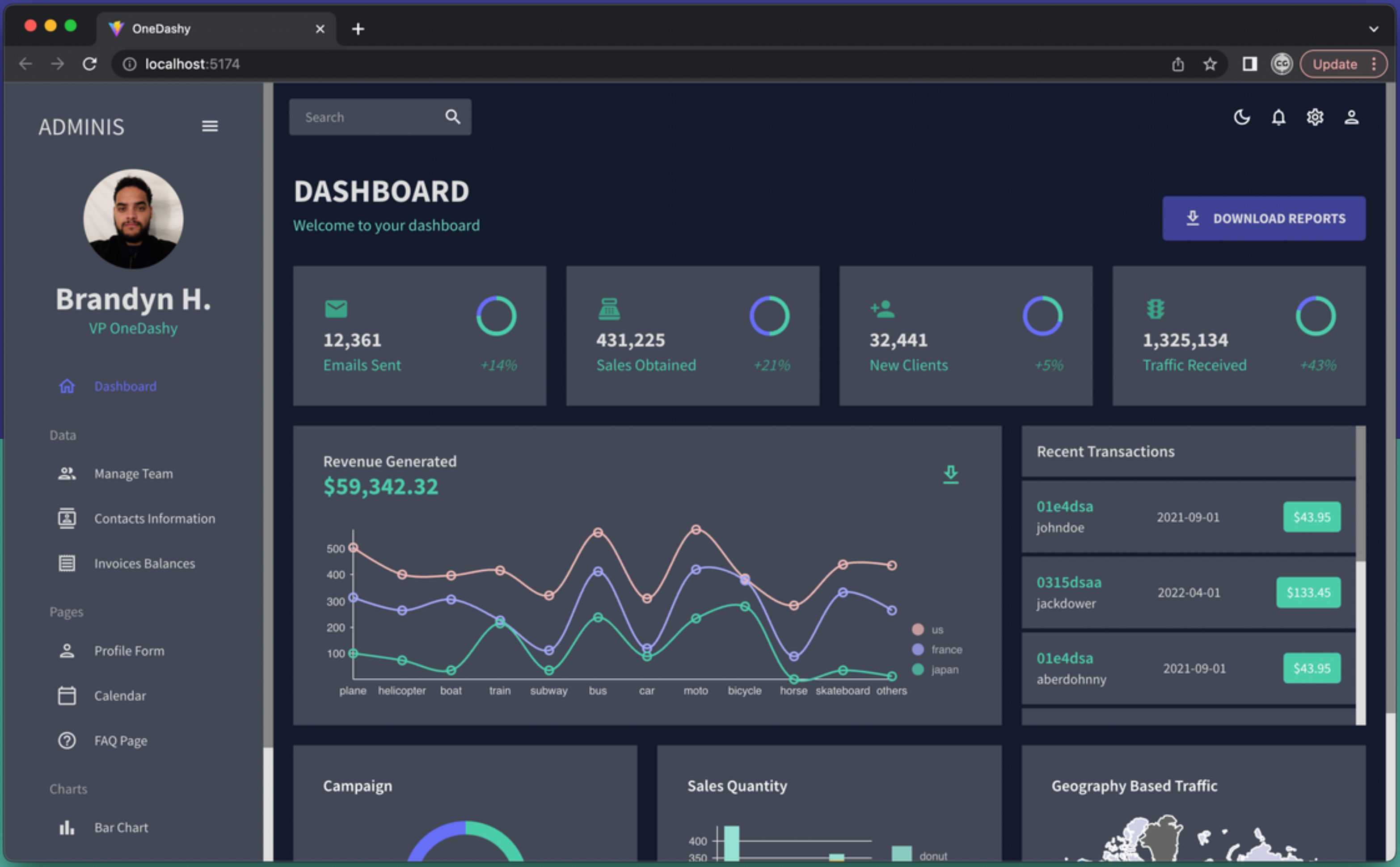
ONEDASHY
CASE STUDY



Brandyn H.
VP OneDashy

DASHBOARD
Welcome to your dashboard

12,361 Emails Sent +14%

431,225 Sales Obtained +21%

32,441 New Clients +5%

1,325,134 Traffic Received +43%

Revenue Generated
$59,342.32

Recent Transactions

01e4dsa johndoe	2021-09-01	$43.95
0315dsaa jackdower	2022-04-01	$133.45
01e4dsa aberdohnny	2021-09-01	$43.95

OVERVIEW:

OneDashy is a React Desktop administration dashboard. This is designed to be used in a production level environment. This application consists of mock data to replicate employees, or other members of the application. With this mock data, the application mimics how it can operate in a professional field, having a Teams page, Contacts page, Invoices page, Profile page, FAQ page, Calendar page, and Charts pages holding other professional data. This applications has a light and dark mode and makes use of React, Material UI, Nivo Charts, Formik, Yup, FullCalendar, and Data Grid.

GOAL:

The overall goal of this application was to present a project that could be used in a professional & production level environment, while integrating the use of quality grade APIs.

TOOLS USED:

- React
- Material UI
- Nivo Charts
- Formik
- Yup
- FullCalendar
- Data Grid

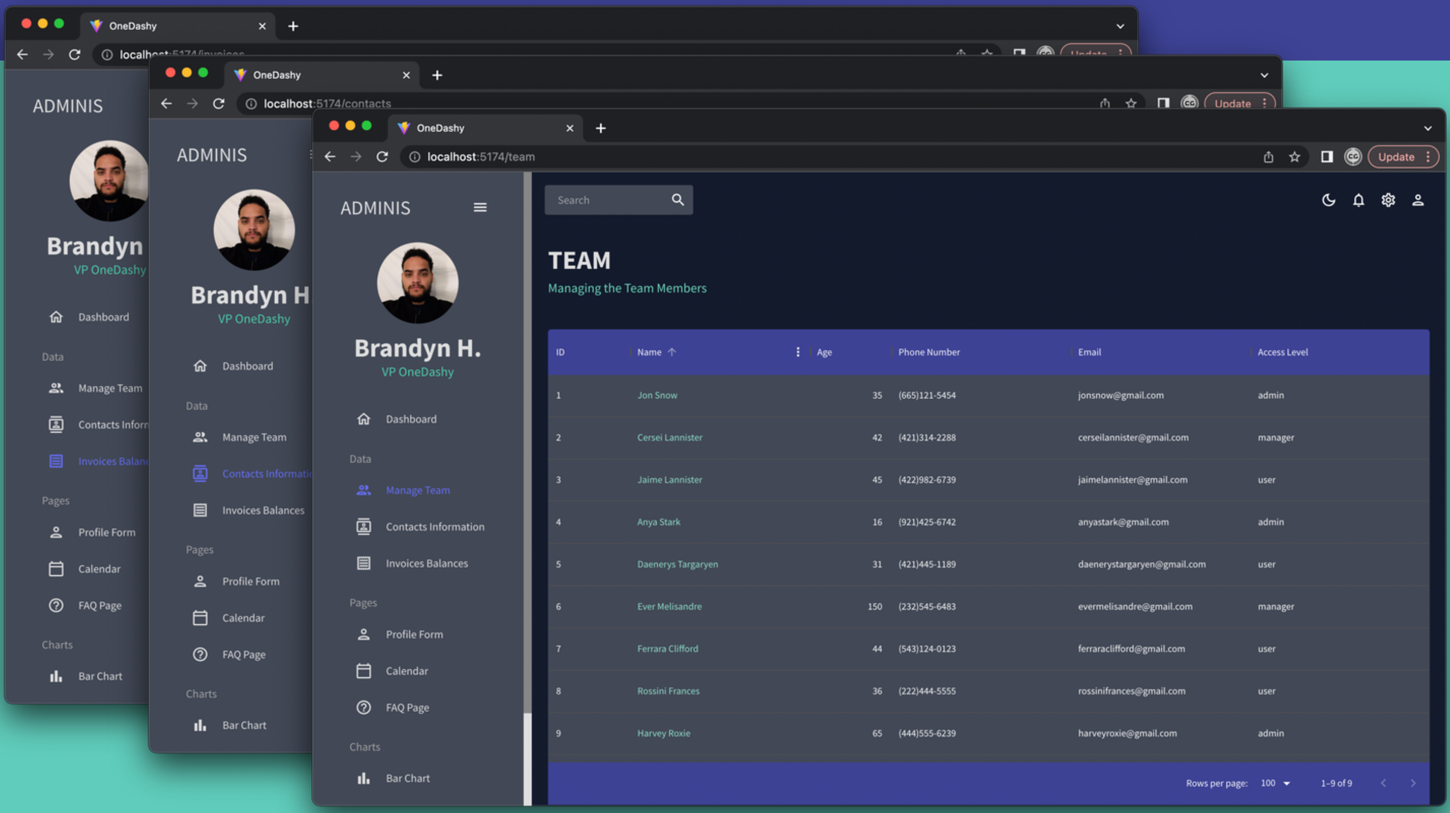
APPROACH:

While working and following along on this React application process, "Production Level" was always on forefront of the mind. Having the application look profession, in its design and usability, along with the technologies used, needed to be quality level. Making use of the various APIs was the decision to make this work. Mock data was also created to replicate the professional level environment by having many employees or users of the application, which was used throughout the application in all of its different pages and views.

MOCK DATA:

The mock data was used on the Dashboard to present in a Teams, Contact, and Invoice page. The data has their names, age, phone numbers, email, address, city, zip-code, & invoices.

Data Grid API Has been used for the layout of these pages.



TEAM

Managing the Team Members

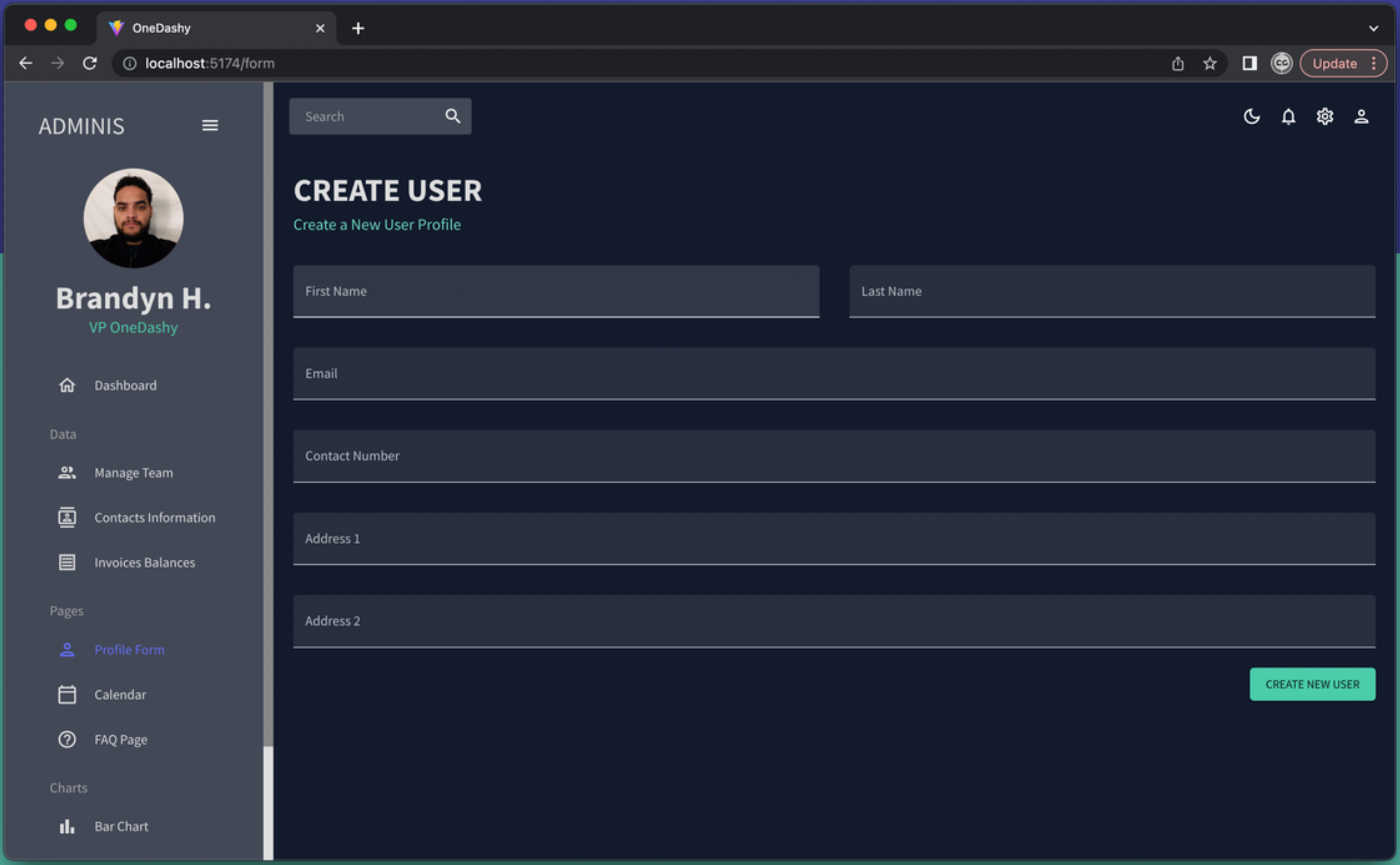
ID	Name	Age	Phone Number	Email	Access Level
1	Jon Snow	35	(665)121-5454	jonsnow@gmail.com	admin
2	Cersei Lannister	42	(421)314-2288	cerseilannister@gmail.com	manager
3	Jaime Lannister	45	(422)982-6739	jaimelannister@gmail.com	user
4	Anya Stark	16	(921)425-6742	anyastark@gmail.com	admin
5	Daenerys Targaryen	31	(421)445-1189	daenerystargaryen@gmail.com	user
6	Ever Melisandre	150	(232)545-6483	evermelisandre@gmail.com	manager
7	Ferrara Clifford	44	(543)124-0123	ferraraclifford@gmail.com	user
8	Rossini Frances	36	(222)444-5555	rossinifrances@gmail.com	user
9	Harvey Roxie	65	(444)555-6239	harveyroxie@gmail.com	admin

Rows per page: 100 1–9 of 9

CREATING USERS:

Although the application is using mock data to be represented in OneDashy, a Profile Forms page has been added for the future ability to add users to a database through means of an external API to store the new members information, like **MongoDB** or **PostgreSQL**. When admins use the application to add new members, they could use this form in uniform with either of these APIs to hold and fetch this information using HTML requests.

The **Formik & Yup APIs** have been used in this application for user authentication and validation throughout the Profile Forms page.

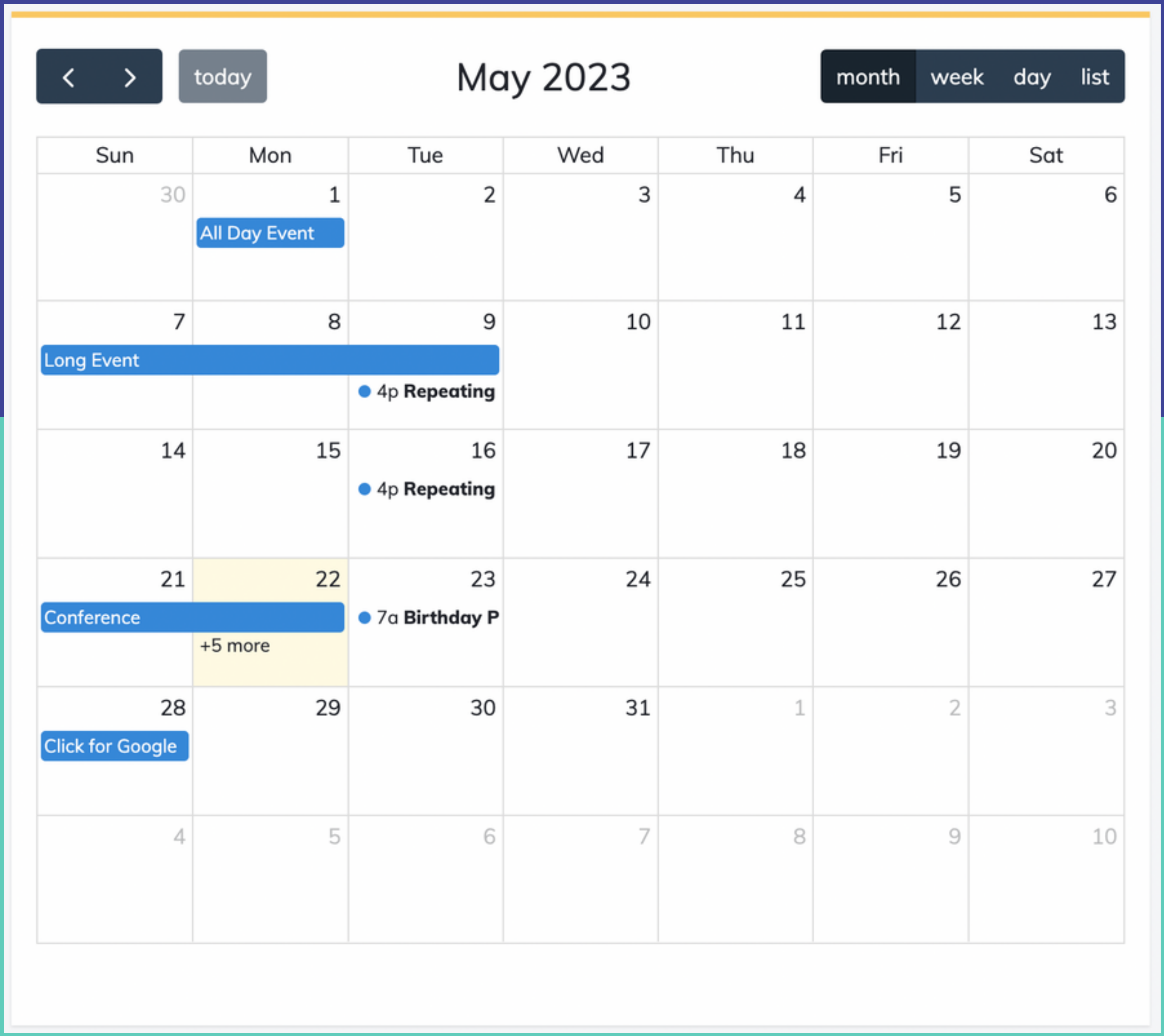


CREATING & SHARING EVENTS:

In a professional and administrative environment, sharing events and needing to kept update is just another part of the lifestyle. Needing to be aware of what's coming up and sharing the events with your team is needed in such an environment.

The **FullCalendar API** have been used in this application for replicating this environment.

demo application is having issues running FullCalendar API, as soon as the issue is resolved this will become available again. In the meantime, refer to the online demo, https://fullcalendar.io/demos

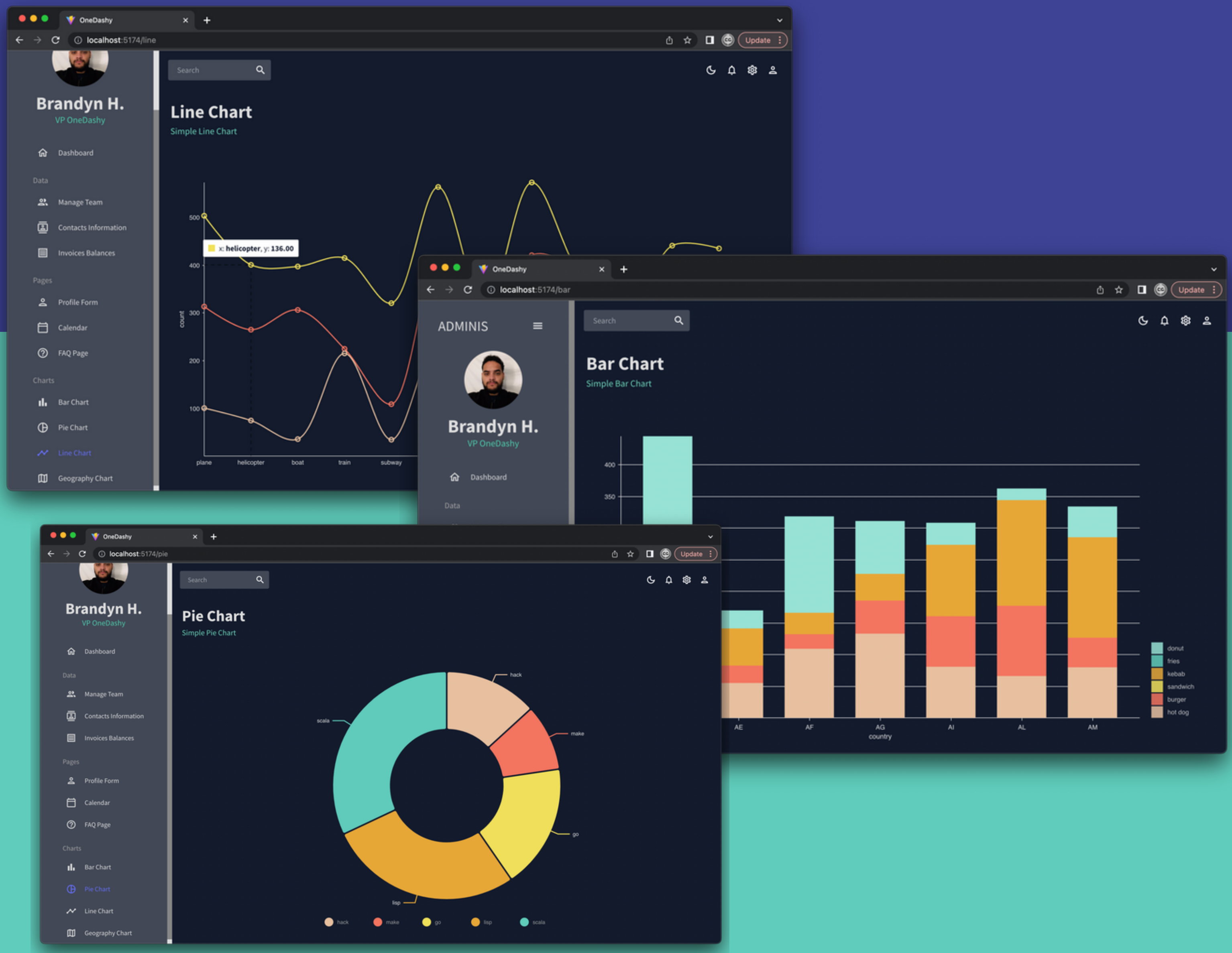


Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1 All Day Event	2	3	4	5	6
7 Long Event	8	9 ● 4p **Repeating**	10	11	12	13
14	15	16 ● 4p **Repeating**	17	18	19	20
21 Conference	22 +5 more	23 ● 7a **Birthday P**	24	25	26	27
28 Click for Google	29	30	31	1	2	3
4	5	6	7	8	9	10

SHARING DATA, NUMBERS & CHARTS:

In many professional fields and environments, the sharing of information and displaying number through charts is a valuable resource. Again, we used mock data for the information provided to replicate this type of environment. In a production level application, we would use another external data base to use HTML requests to add, update, remove, delete new numbers and data.

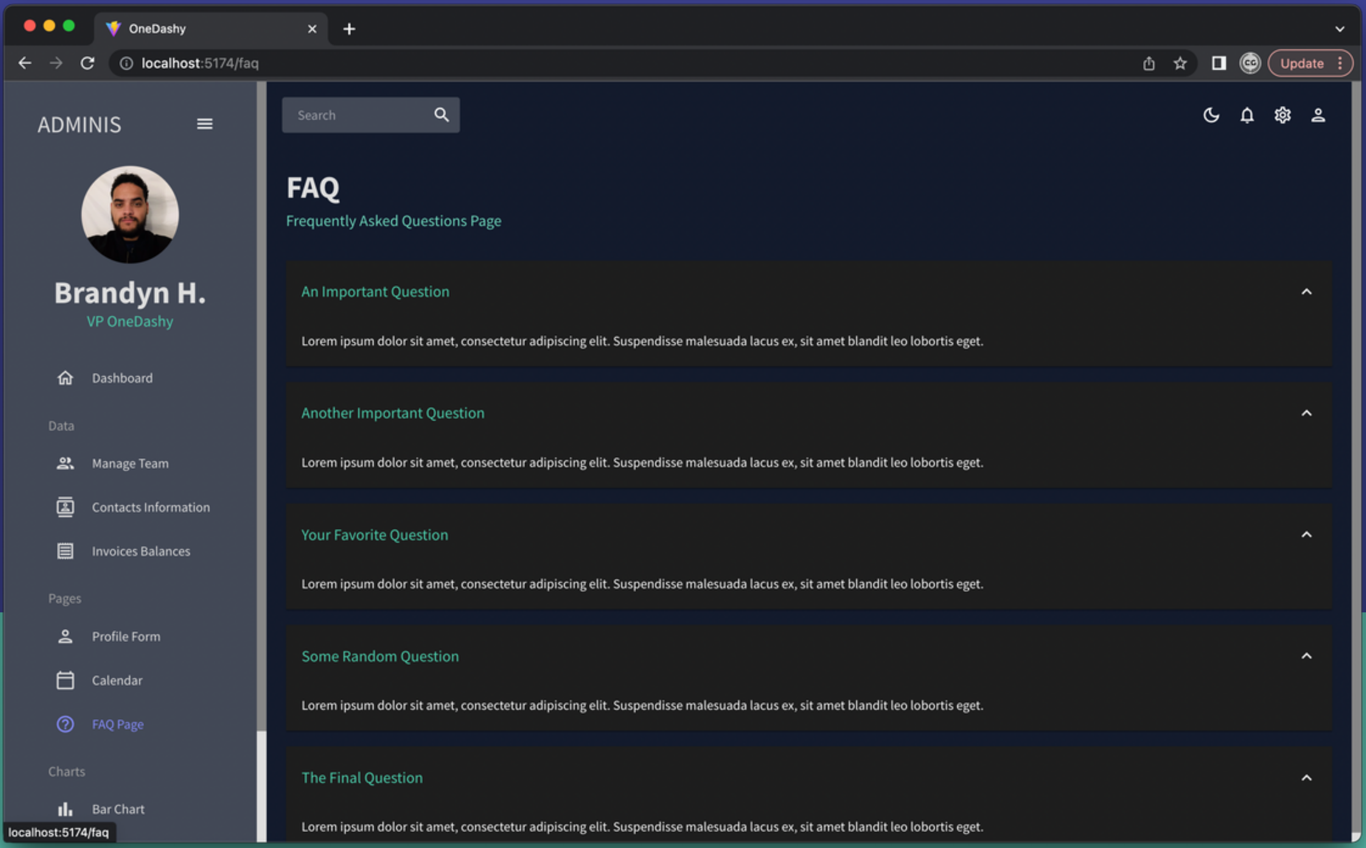
Nivo Charts API have been used in this application for to display a few different charts to display the mock data below.



FAQ:

In a professional and administrative environment, a frequently asked question page is vital to keeping things moving smoothly when help is needed or questions need to be asked/answered.

The **Material UI Accordion** have been used in this application for replicating this environment



FAQ

Frequently Asked Questions Page

An Important Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

Another Important Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

Your Favorite Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

Some Random Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

The Final Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

CONCLUSION:

In conclusion, this application is intended to be used in a professional atmosphere that would require and administrative dashboard in which its users/employees can use to stay connected and informed.

CHALLENGES:

Throughout this project I have had some trouble trying to use the various APIs in conjunction together involving mock data. One problem I had trying to solve was trying to get FullCalendar API and React to work together in their respective versions. FullCalendar API was responding to Reacts new v18. I was able to resolve the issue by using different imports that React would respond with cooperatively to pull from FullCalendar's API. I found the solution while digging through new documentation and searching through others who had similar issues with React v18.