React Native Project with NodeJS and MongoDB- Part 2

Nabendu Biswas · Follow

8 min read · Apr 27, 2022

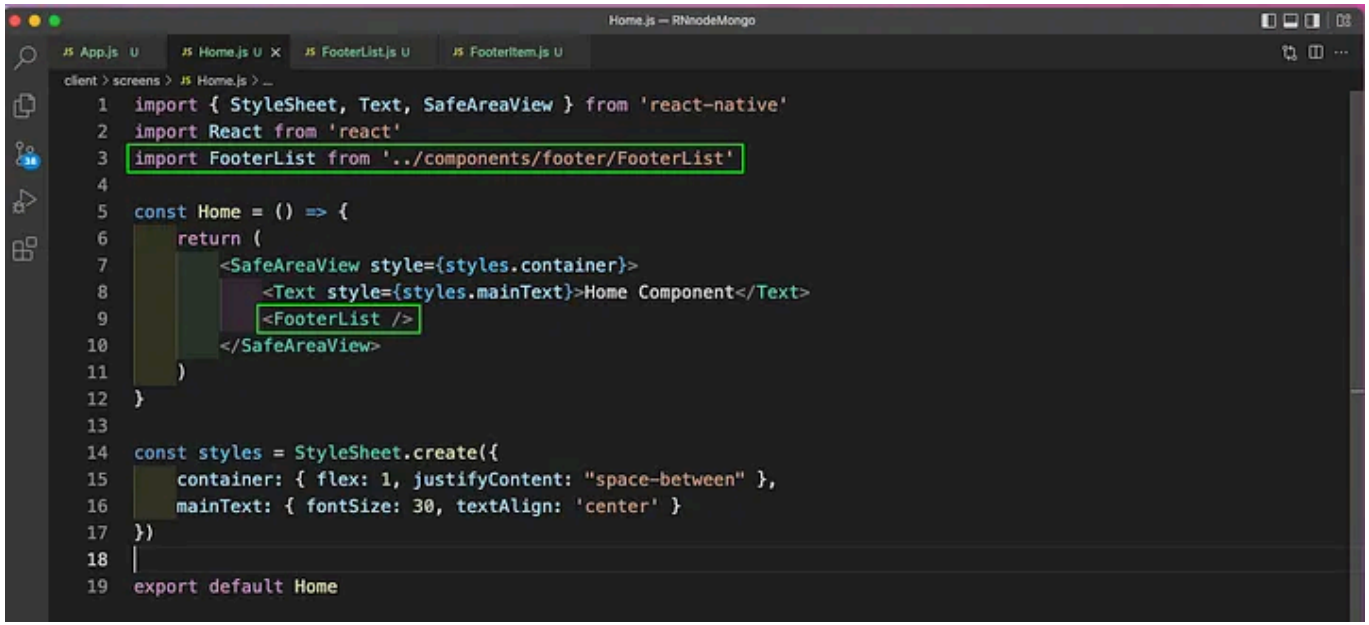
React Native

Welcome to the part-2 of the series and we will start where we left. You can find part-1 here.

We will first install react native vector icons in our project.

```
nabendubiswas@Nabendus-Mac-mini ~/Projects/RNnodeMongo/client (master)$ npm i react-native-vector-icons

added 10 packages, and audited 921 packages in 5s

34 packages are looking for funding
  run `npm fund` for details

5 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
nabendubiswas@Nabendus-Mac-mini ~/Projects/RNnodeMongo/client (master)$
```

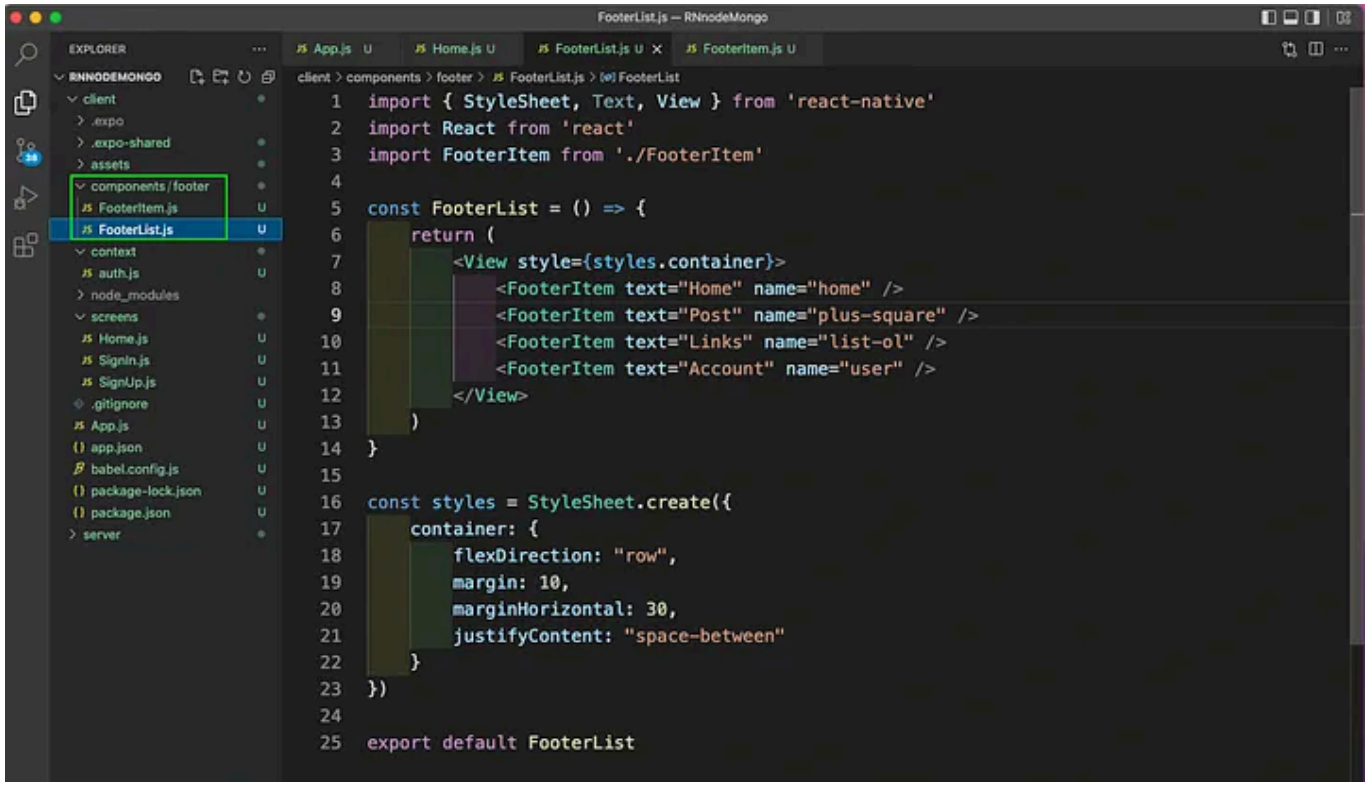
After that include a **FooterList** component in the **Home.js** file.



Home.js

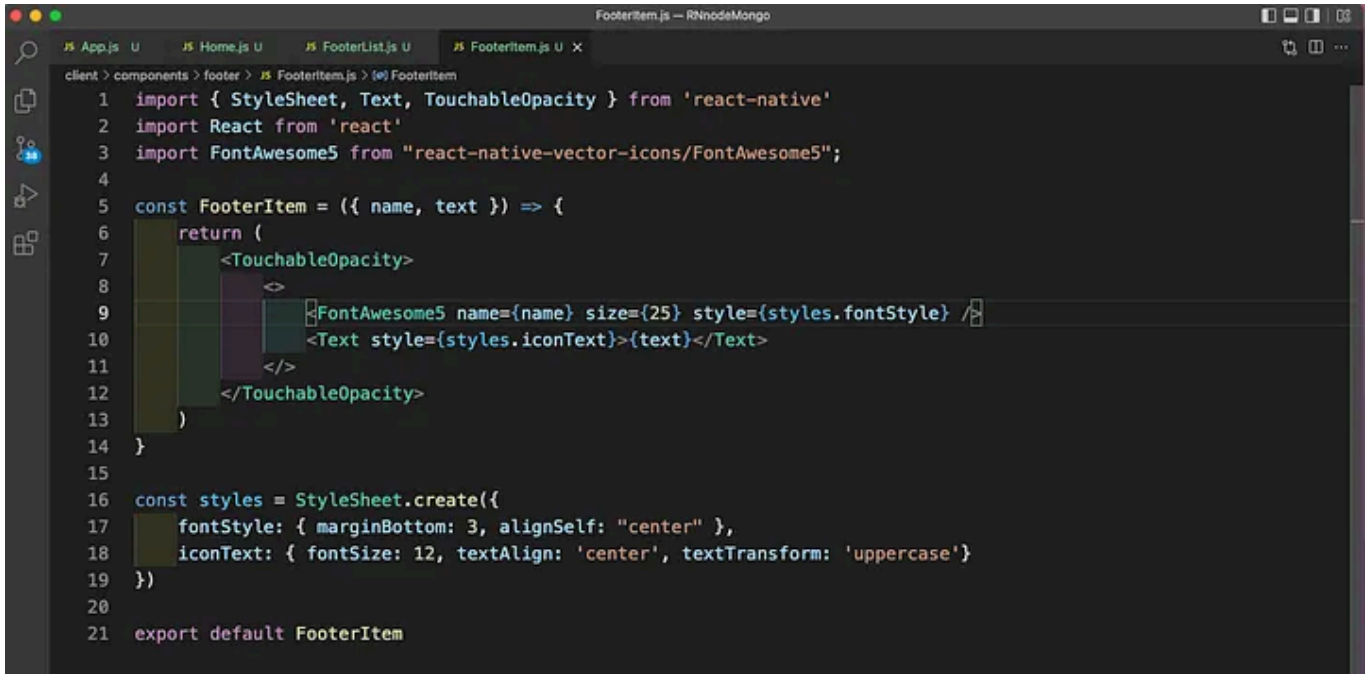
Next, create a **footer** folder in the **components** folder and a file **FooterList.js** inside it. Add the below content in it.

Here, we are sending the **text** and the **name** props to a **FooterItem** component.



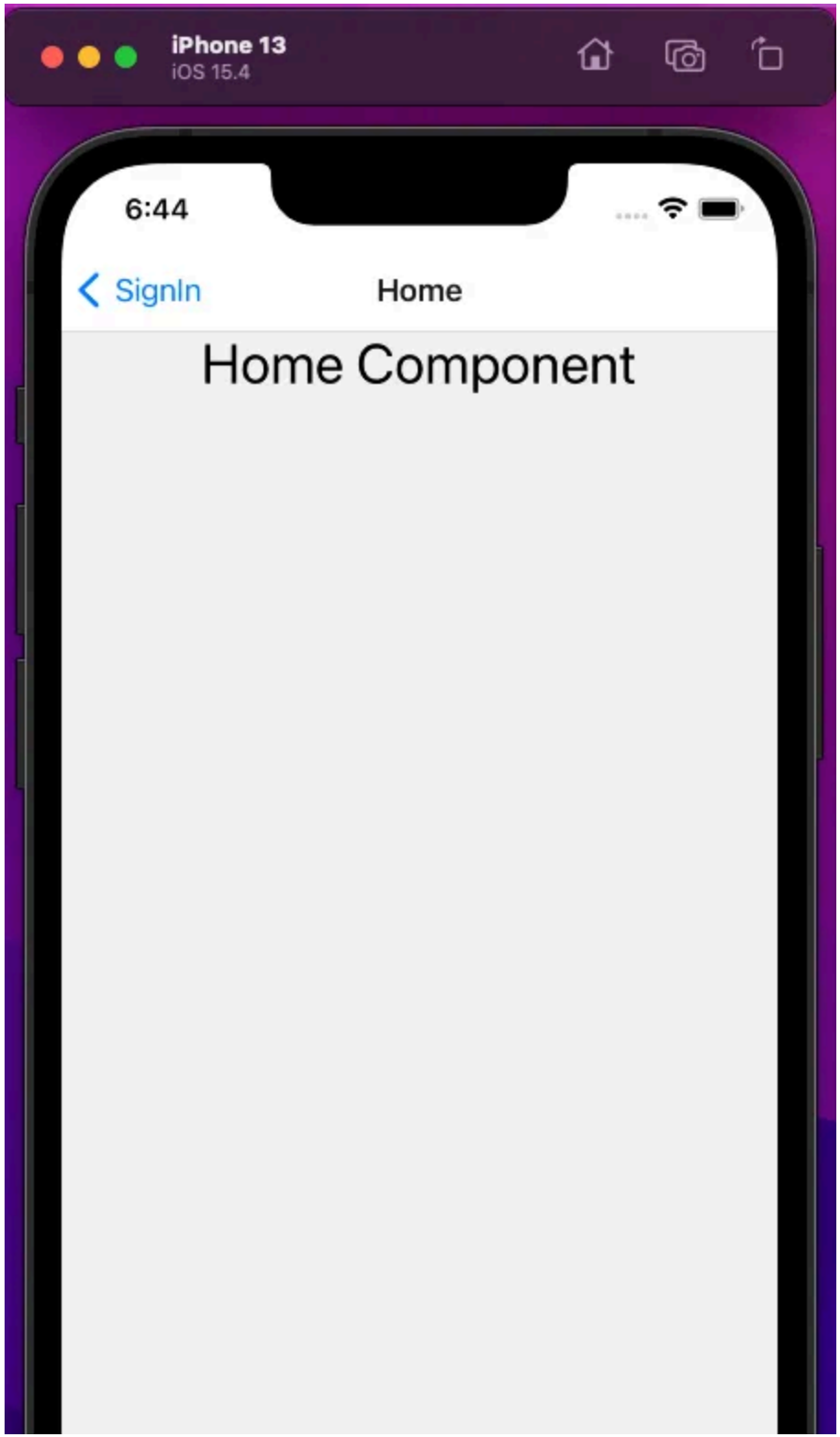
FooterList.js

Now, create a **FooterItem.js** file inside the **footer** folder. Here, we are using the FontAwesome5 icons with the passed props.

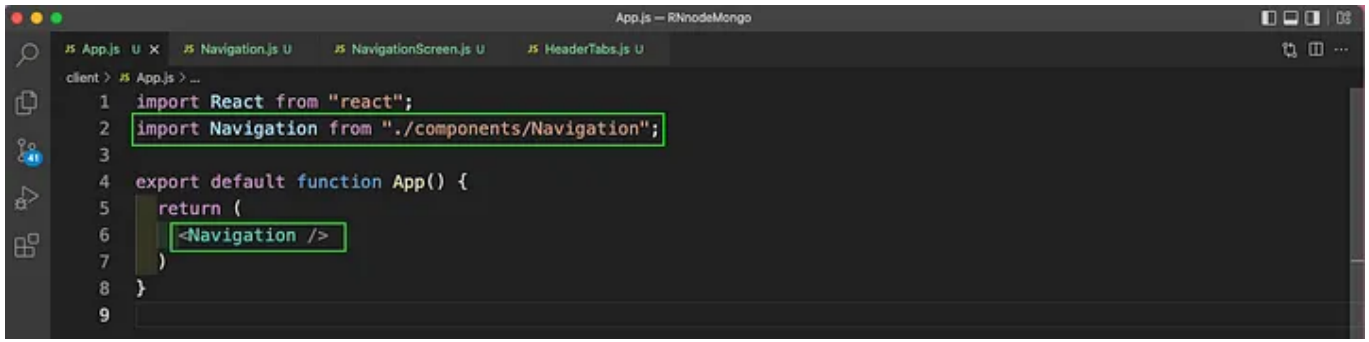


FooterItem.js

Now, in our Home component we will see nice icons in footer.


iPhone 13
iOS 15.4
6:44
SignIn
Home
Home Component

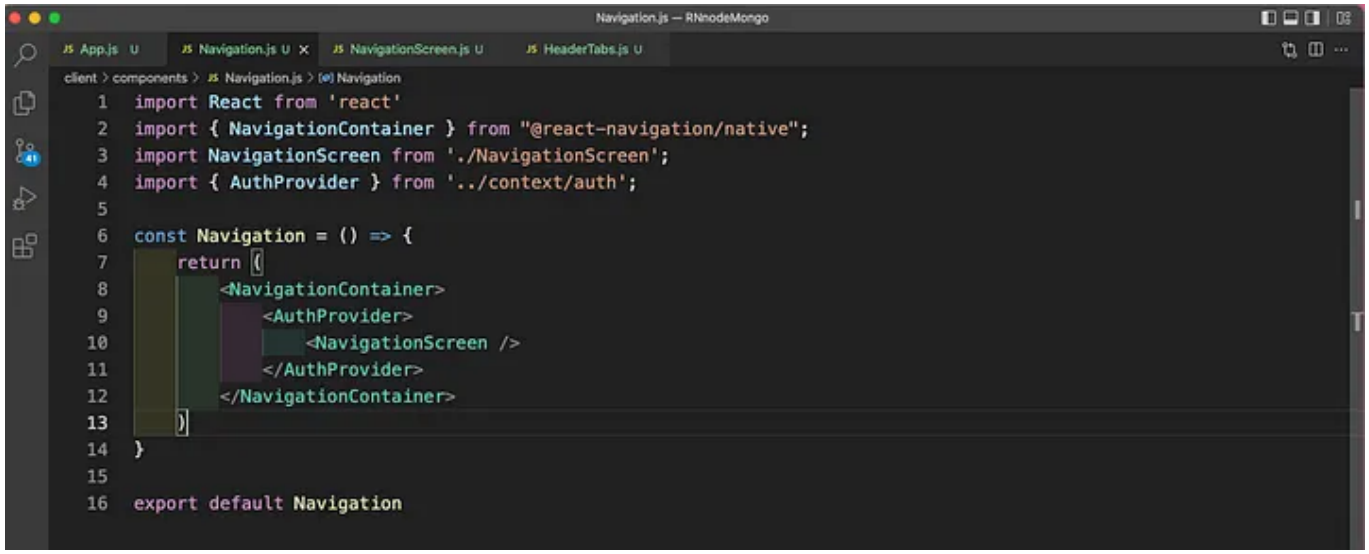
Now, we will refactor our code by moving all content from **App.js** file. Here, we have inserted a **Navigation** component.



App.js

Next create a **Navigation.js** file inside the components folder and add the below code in it.

Here, we are calling a **NavigationScreen** component.

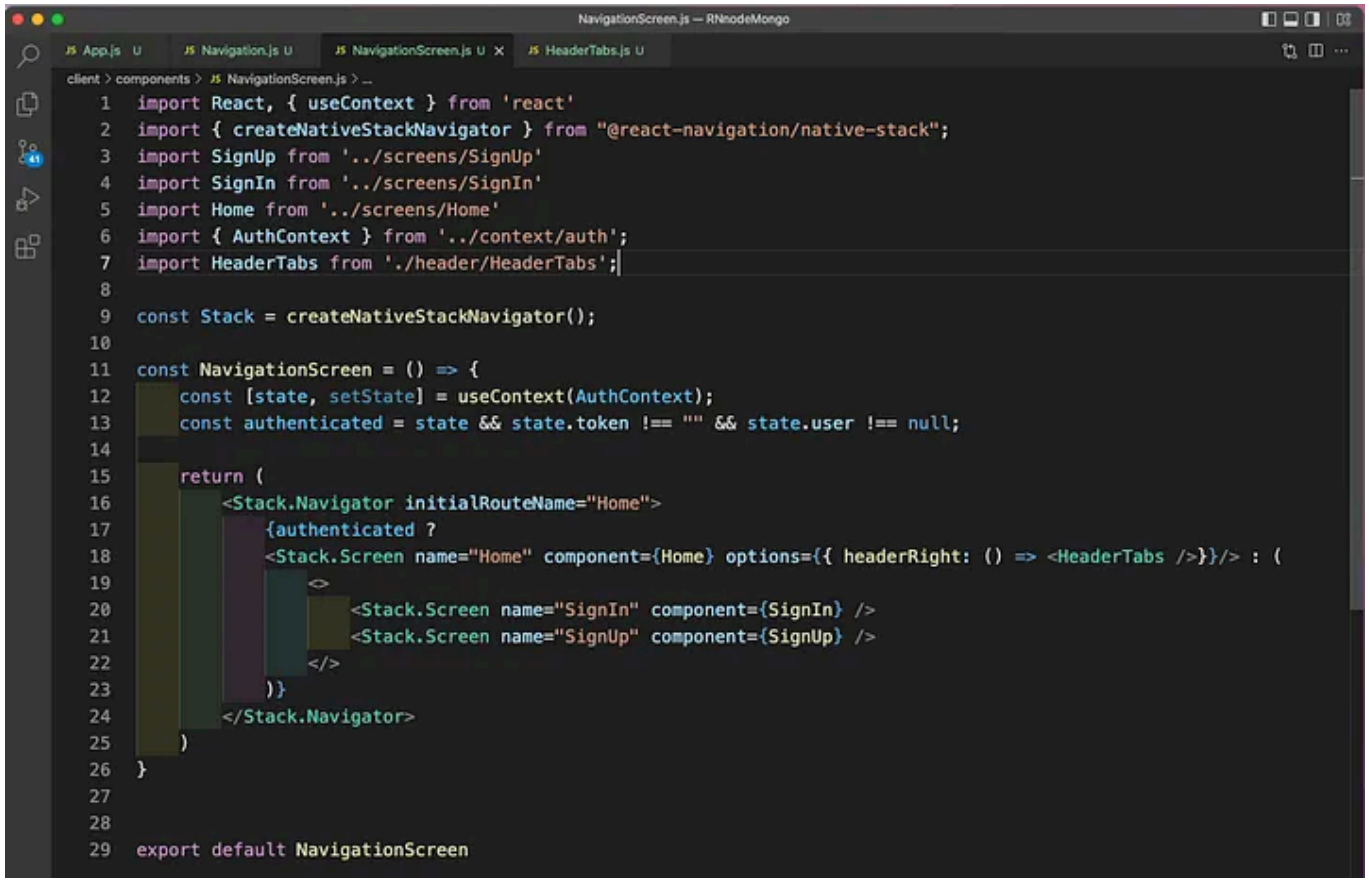


Navigation.js

In the **NavigationScreen.js** file we have most of the code from App.js file. But we have also added the concept of authentication.

Here, we are checking whether we have the token and if we have it then showing the Home Screen or else showing the SignIn component.

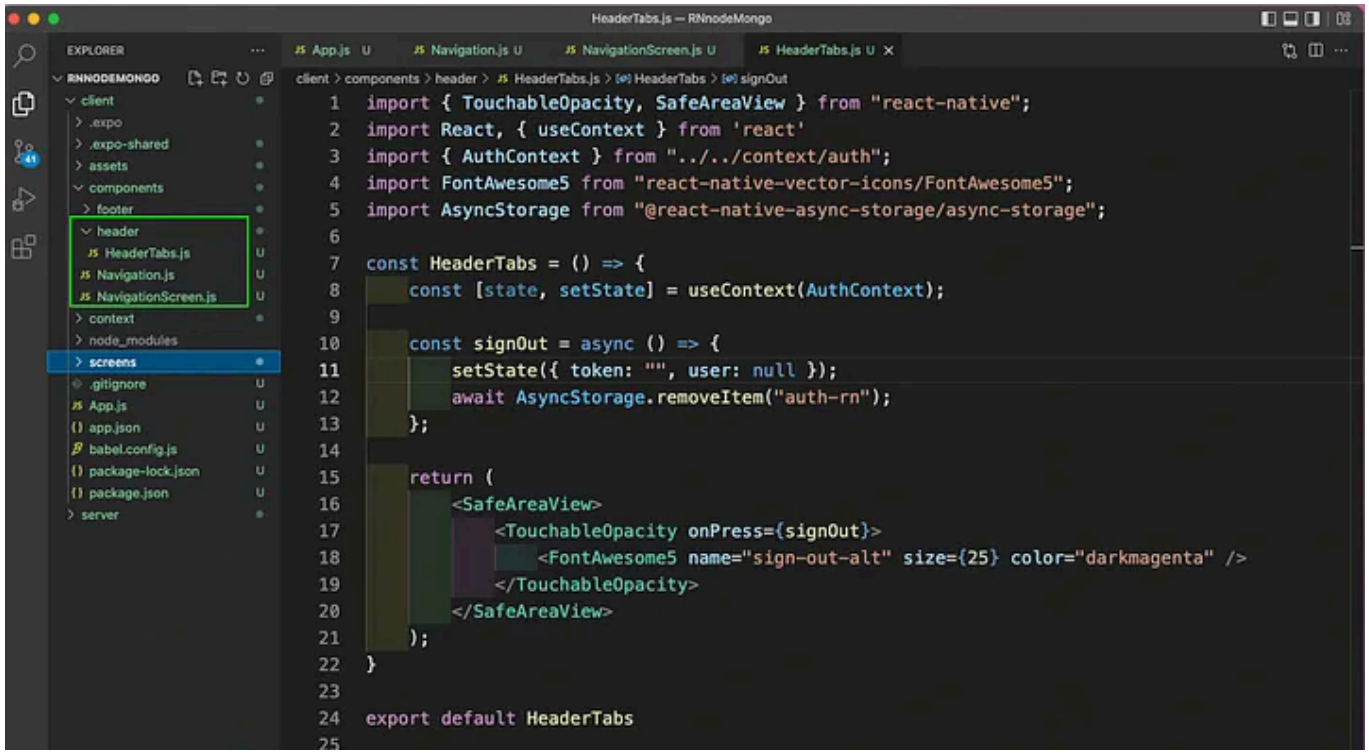
We have also included a new **HeaderTabs** in the Home Screen.



NavigationScreen.js

Now, create a **header** folder inside the **components** folder and a **HeaderTabs.js** file inside it.

Here, we have a Signout buttton which will remove our Async Storage.

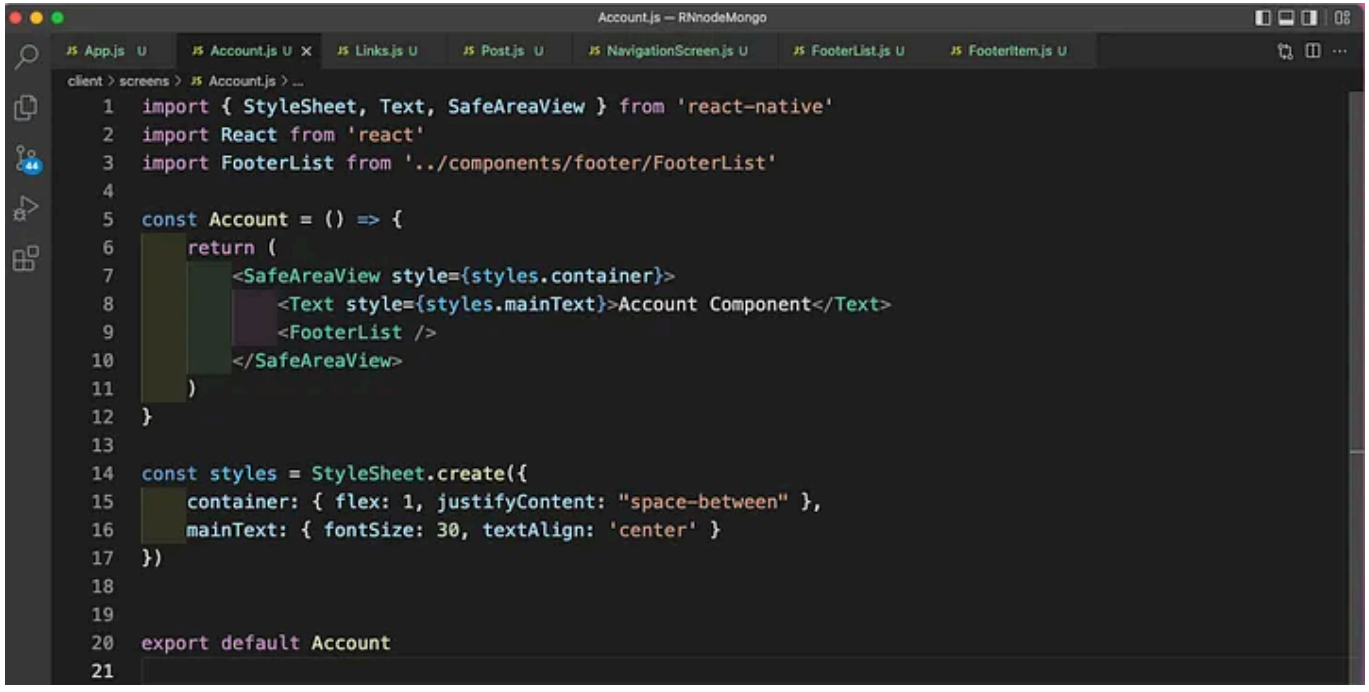


HeaderTabs.js

Now, our app have a working Signout button.

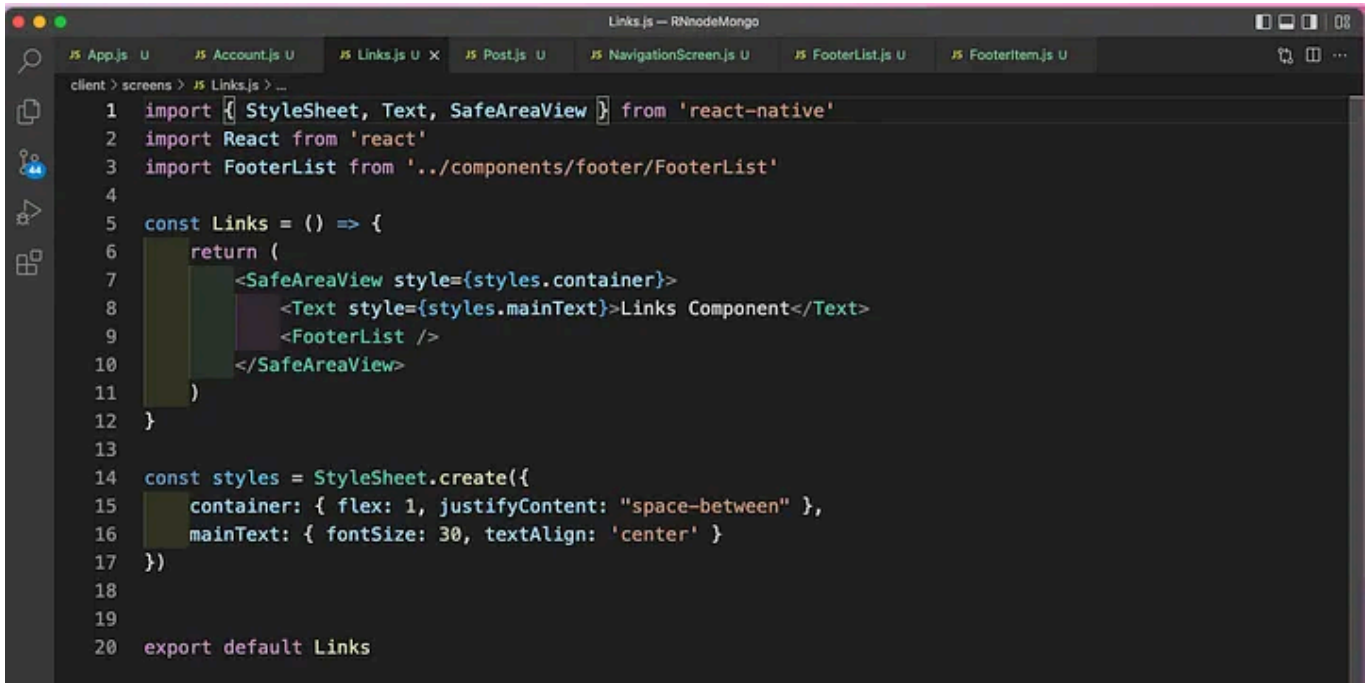

iPhone 13
iOS 15.4
7:24
Home
Home Component

Now, we will create some new screens. So, first create a file **Account.js** inside the **screens** folder and add the below content in it.



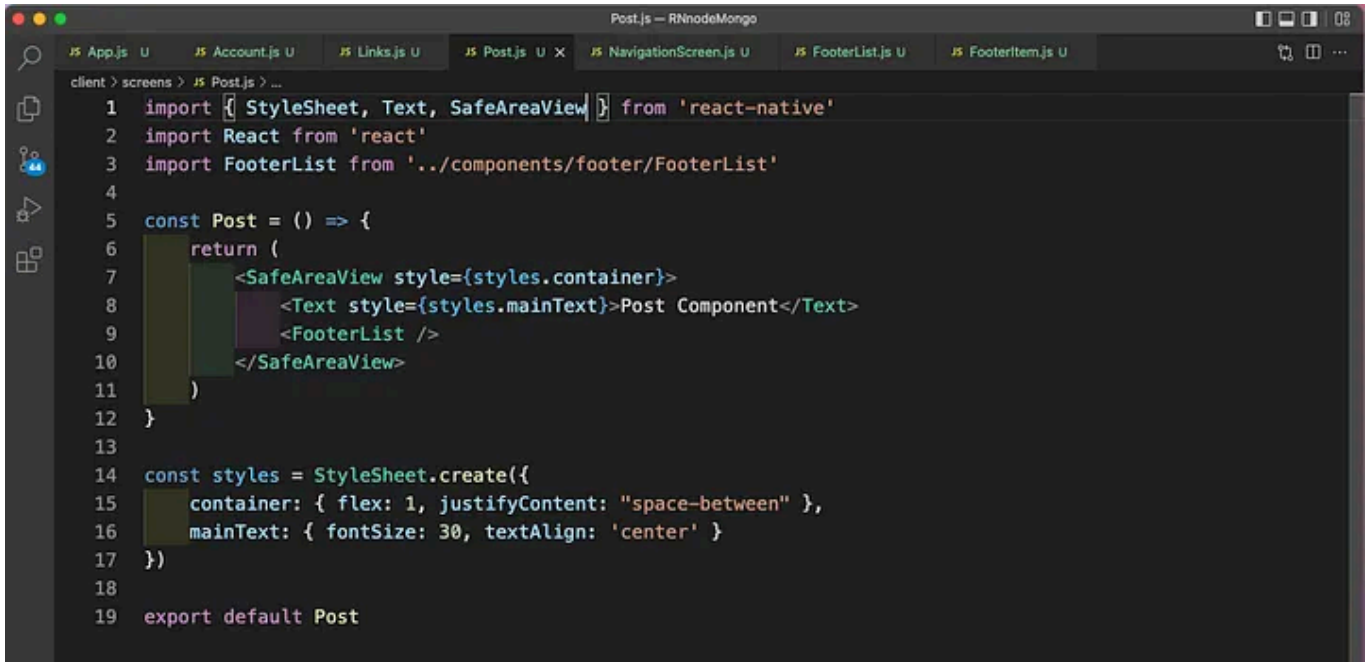
Account.js

Next create a file **Links.js** inside the **screens** folder and add the below content in it.



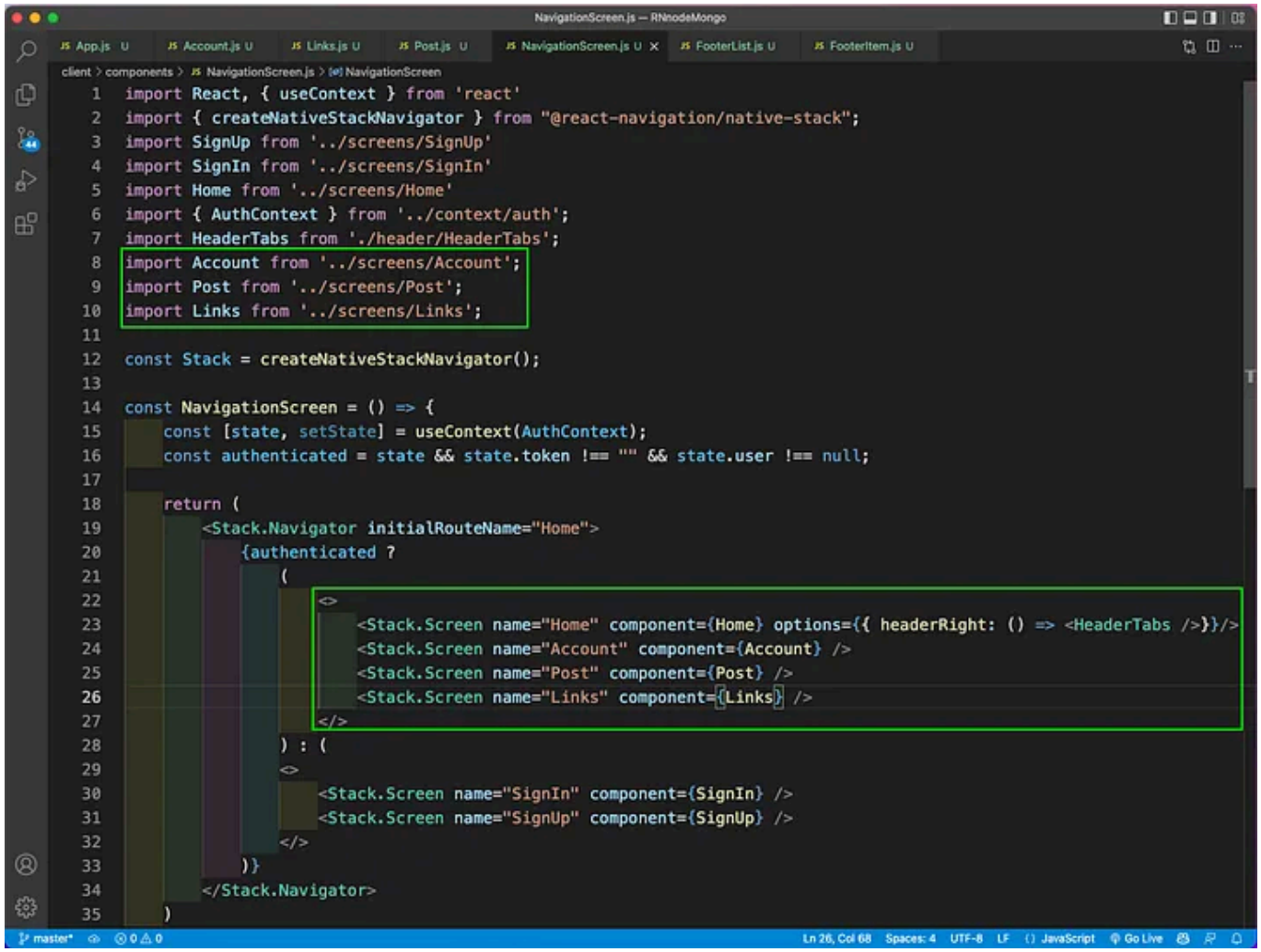
Links.js

Lastly, create a file **Post.js** inside the **screens** folder and add the below content in it.



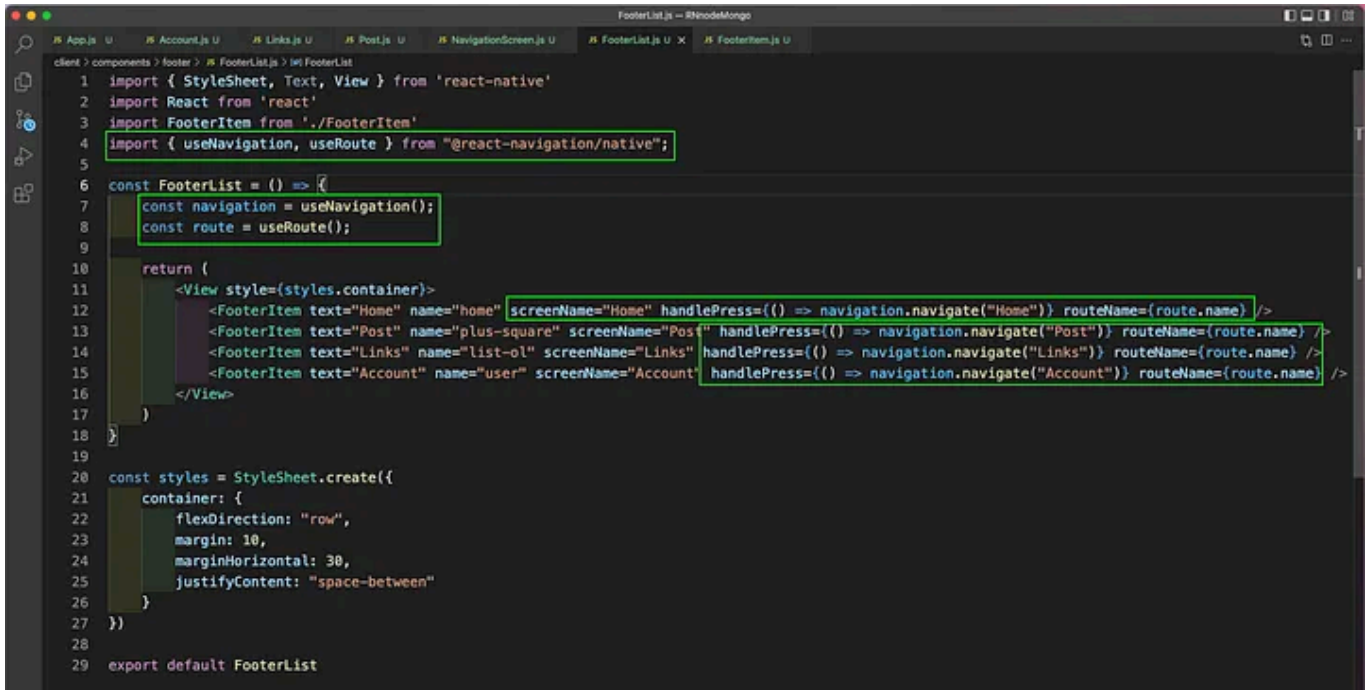
Post.js

Now, we will add these new components route in our **NavigationScreen.js** file.



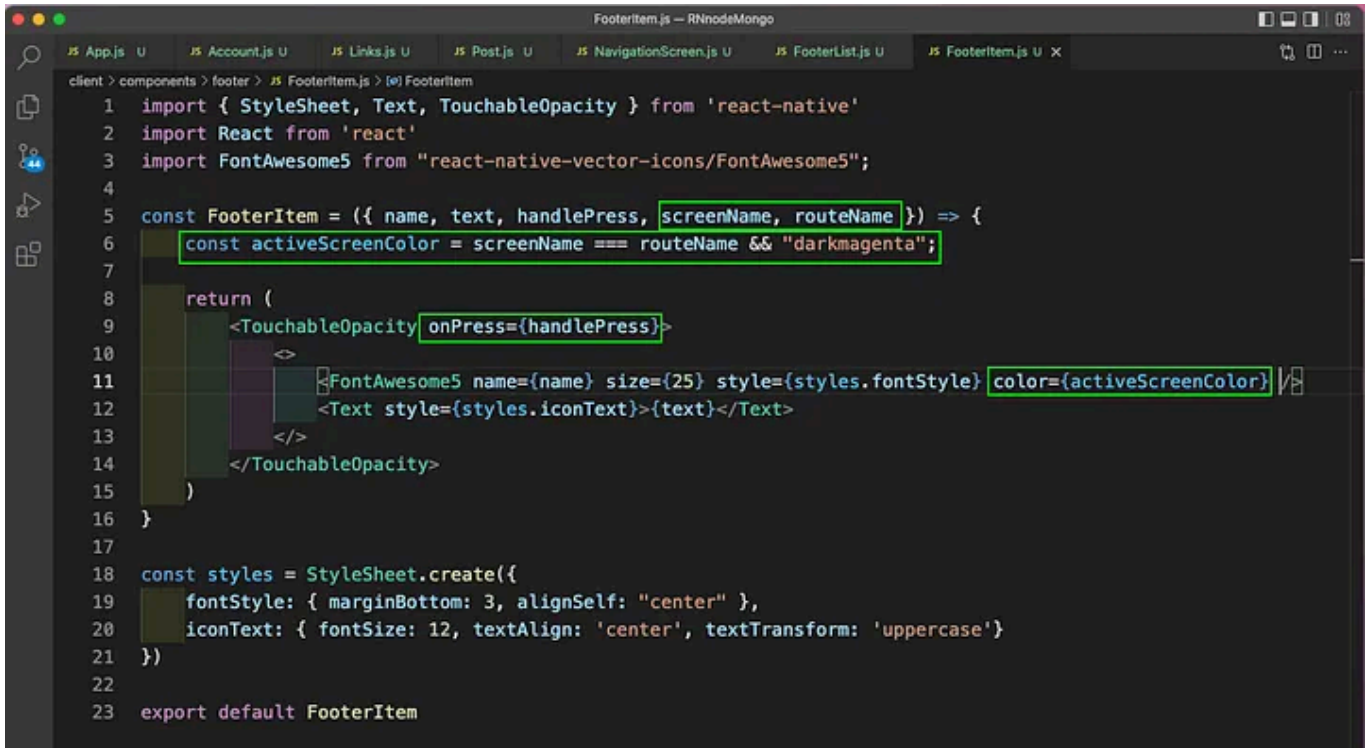
NavigationScreen.js

Next, in the **FooterList.js** file we will add three new props to the FooterItem component. They are routeName, screenName and handlePress.



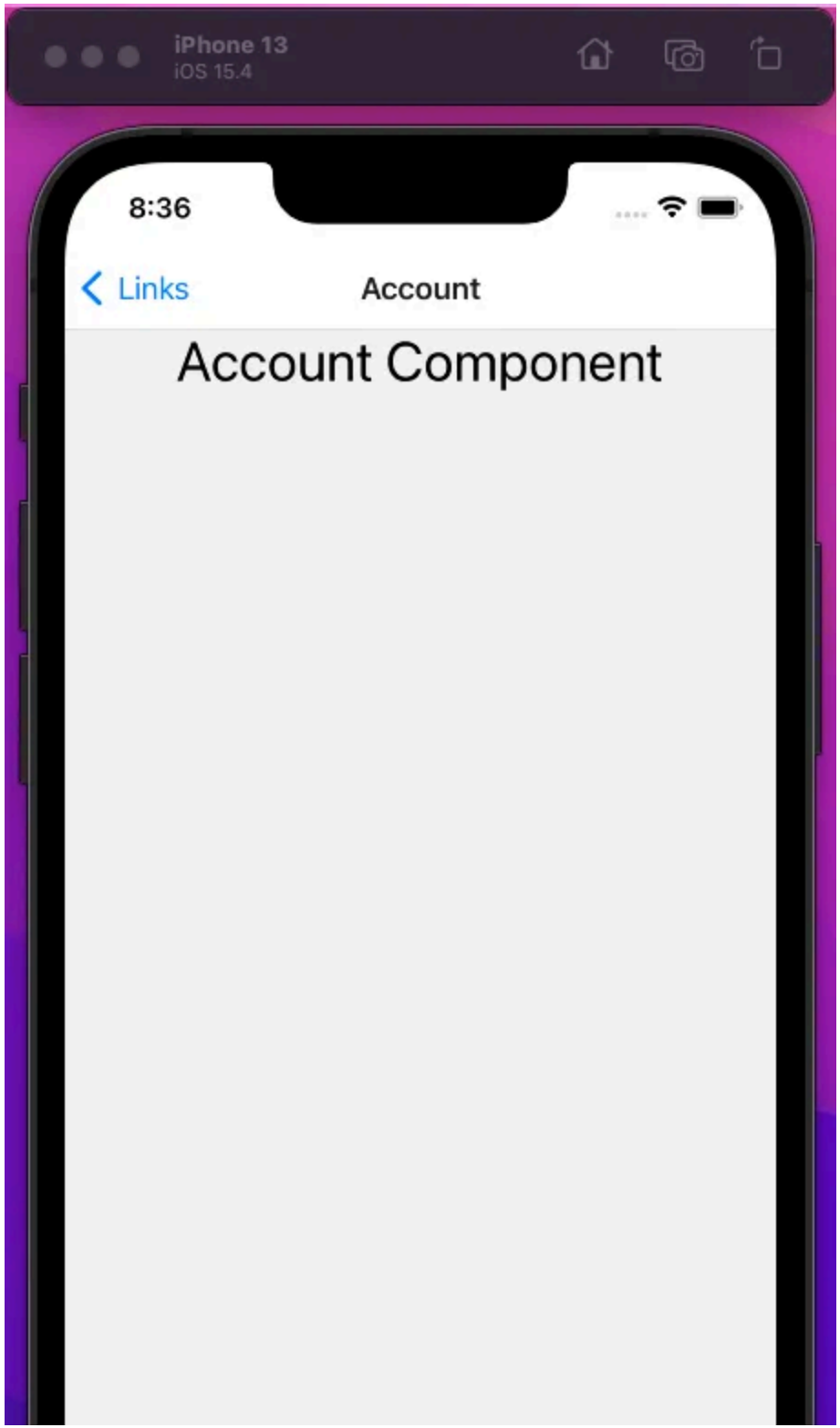
FooterList.js

Now, we will update our **FooterItem.js** file. Here, by using the screenName and routeName we are showing an active tab. Also, onPress have the handlePress function.

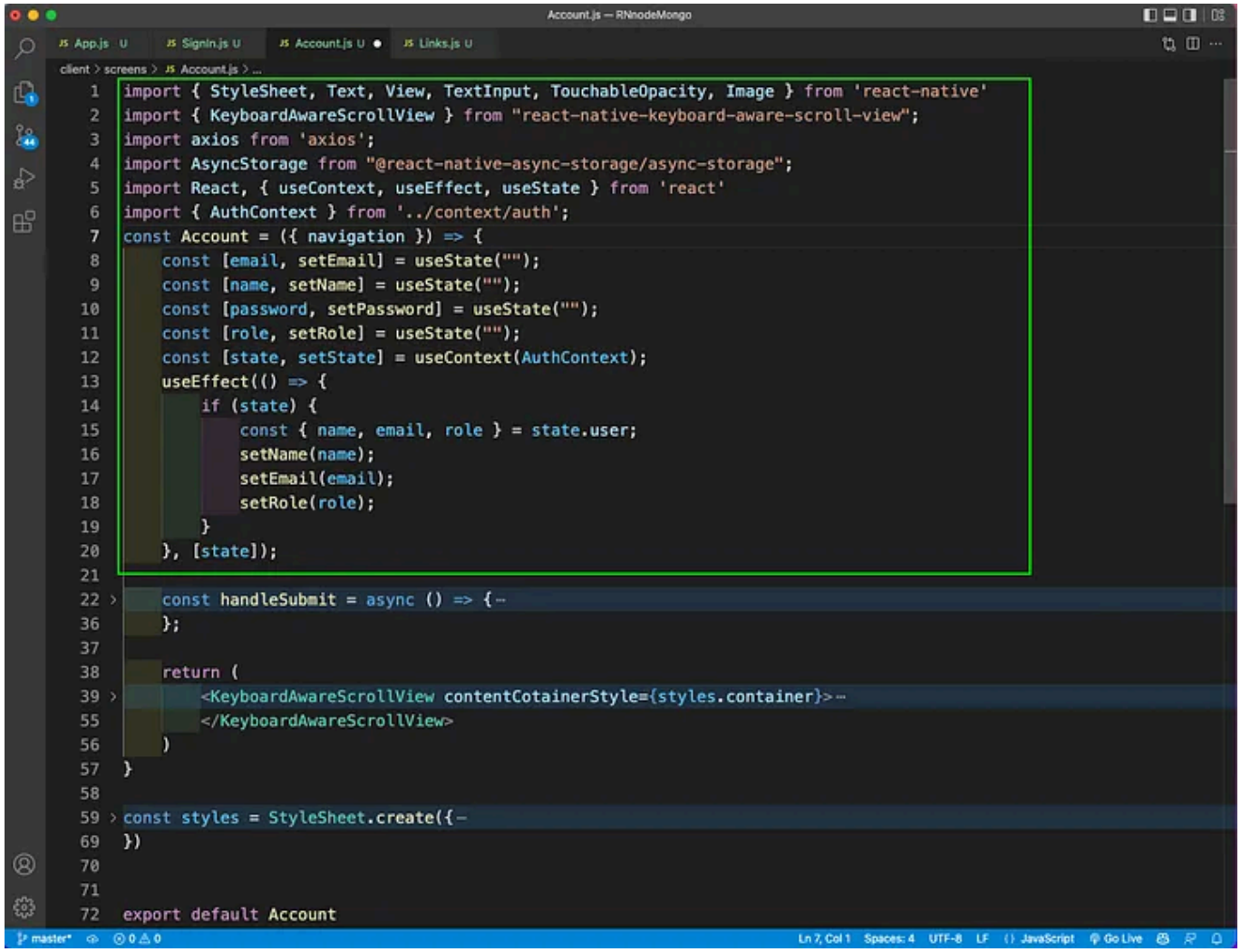


FooterItem.js

Now, our components are changing perfectly with active one shown in purple.

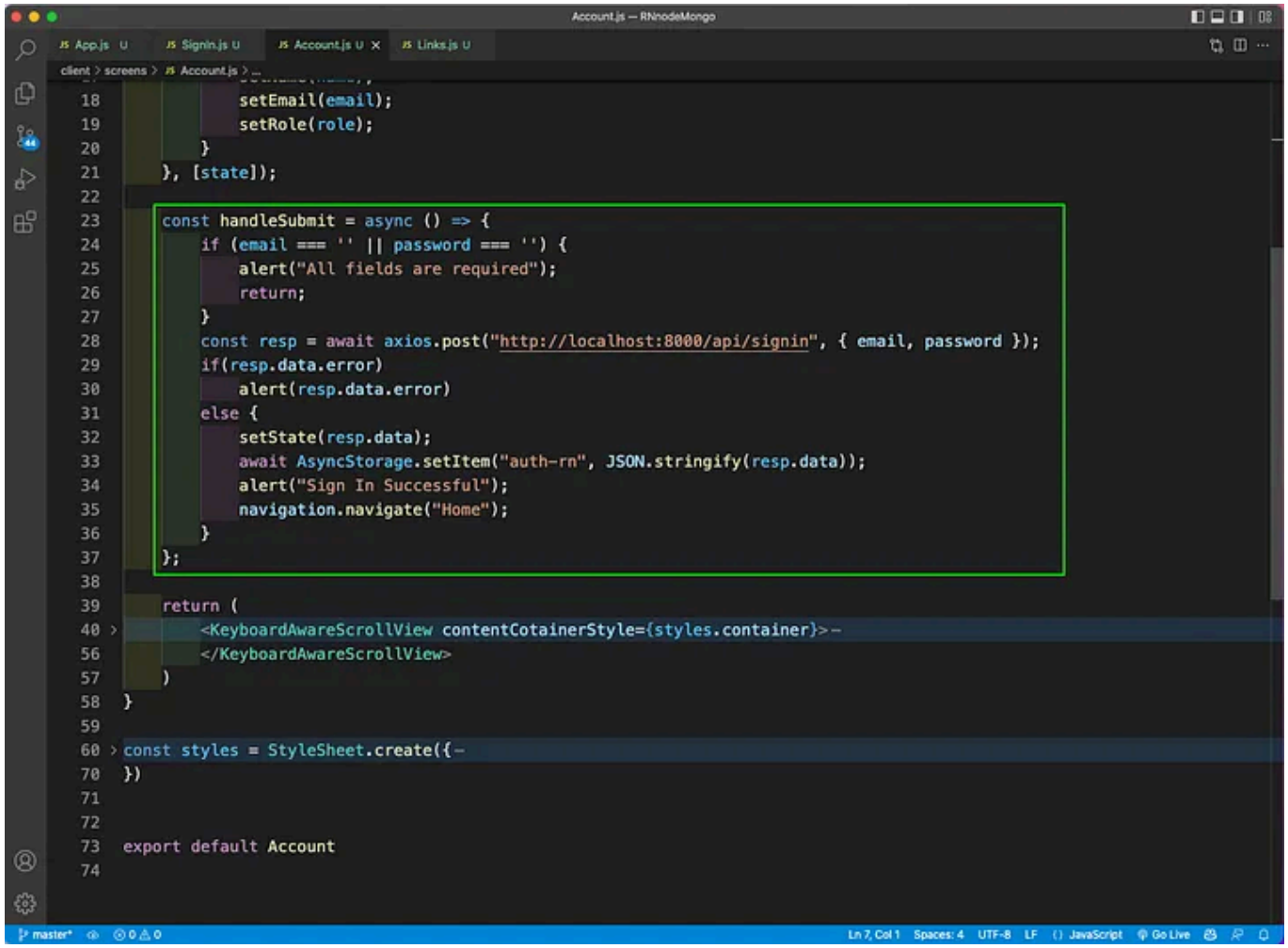

iPhone 13
iOS 15.4
8:36
Links
Account
Account Component

Now, we will update our **Account.js** file to first have the required imports and useState. We are also getting the name, email and role from the context API.



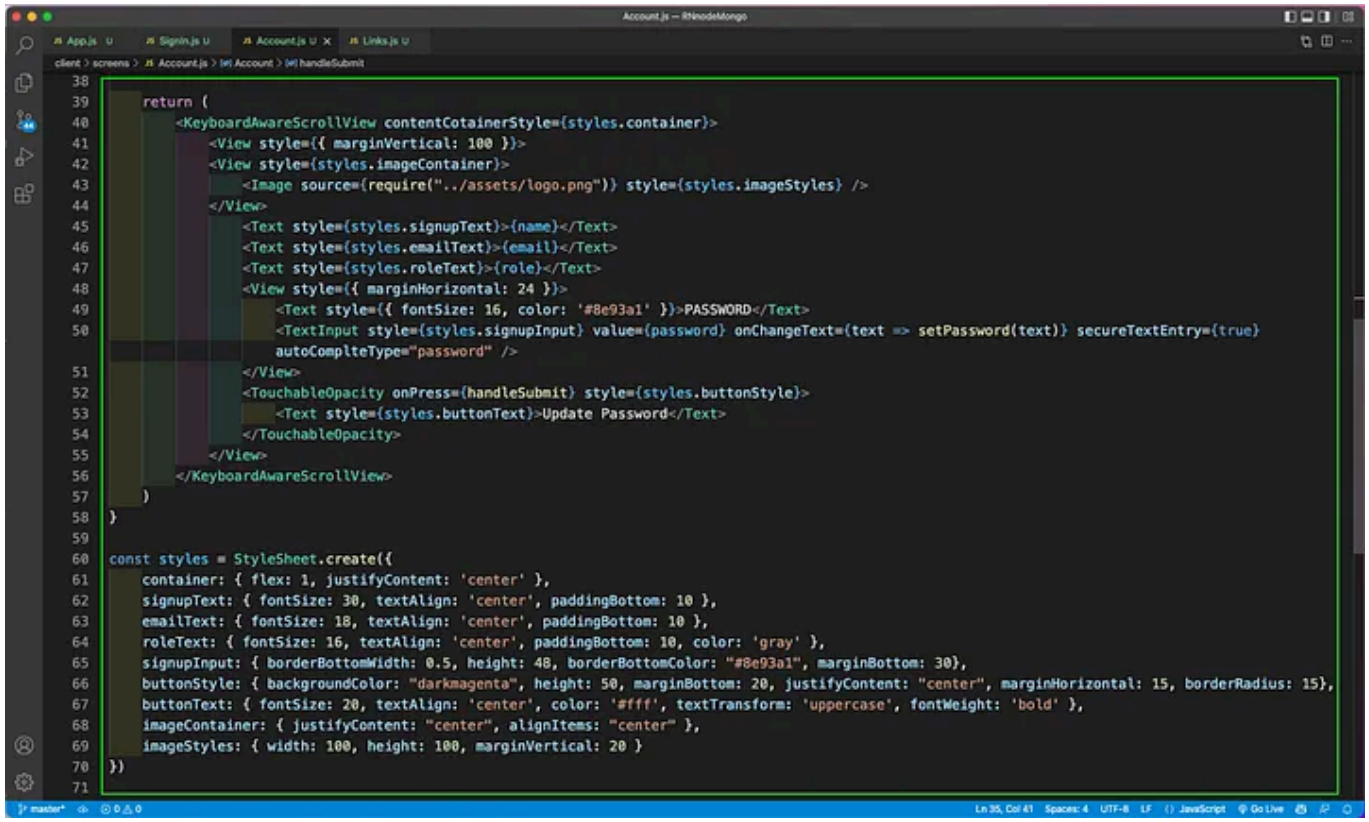
Account.js

Next, we will create the handleSubmit which is quite similar to the SignIn component.



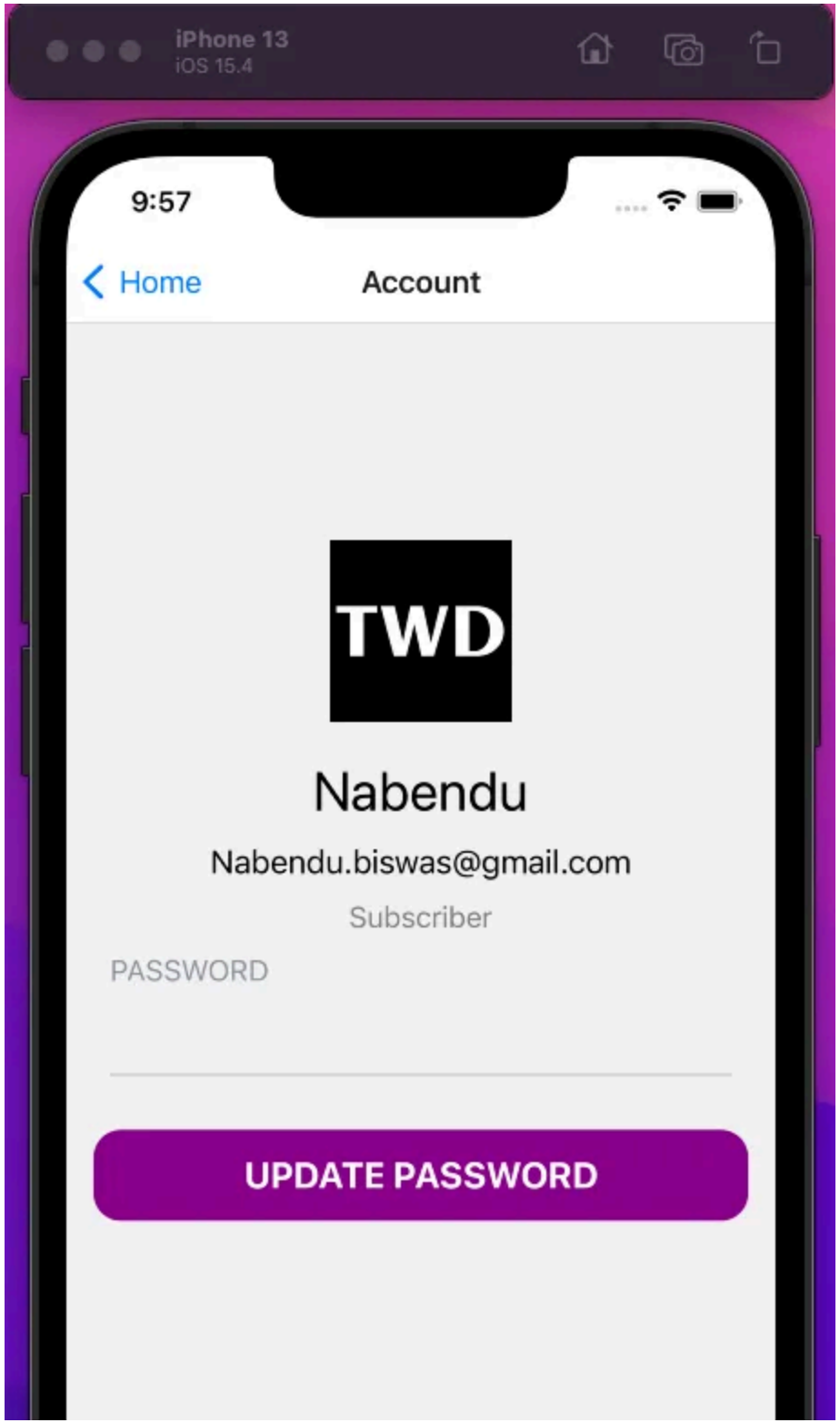
Account.js

Lastly we will add the code in return and the styles for it. Again it is quite similar to the SignIn component.

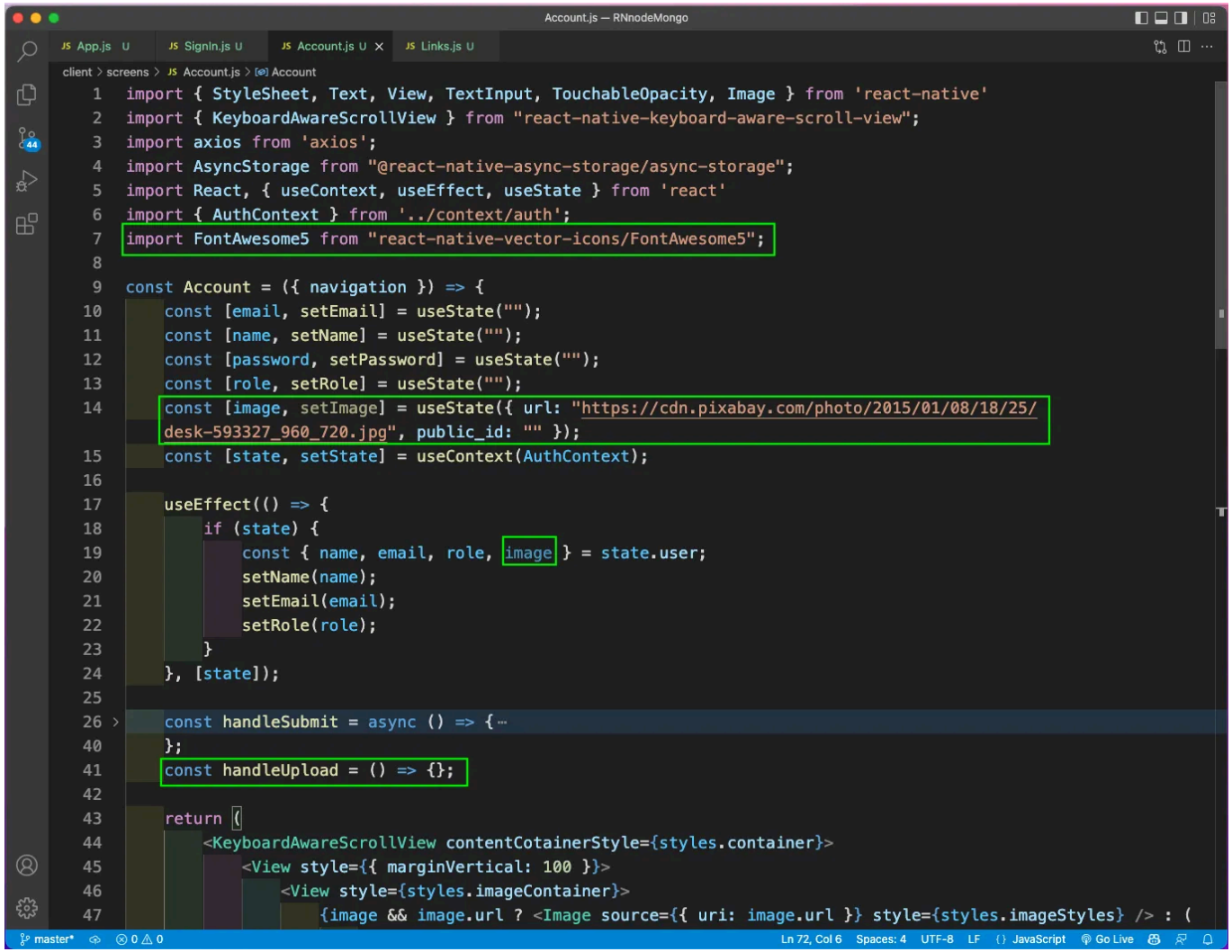


Account.js

Now, our App will show this nice view in Account.

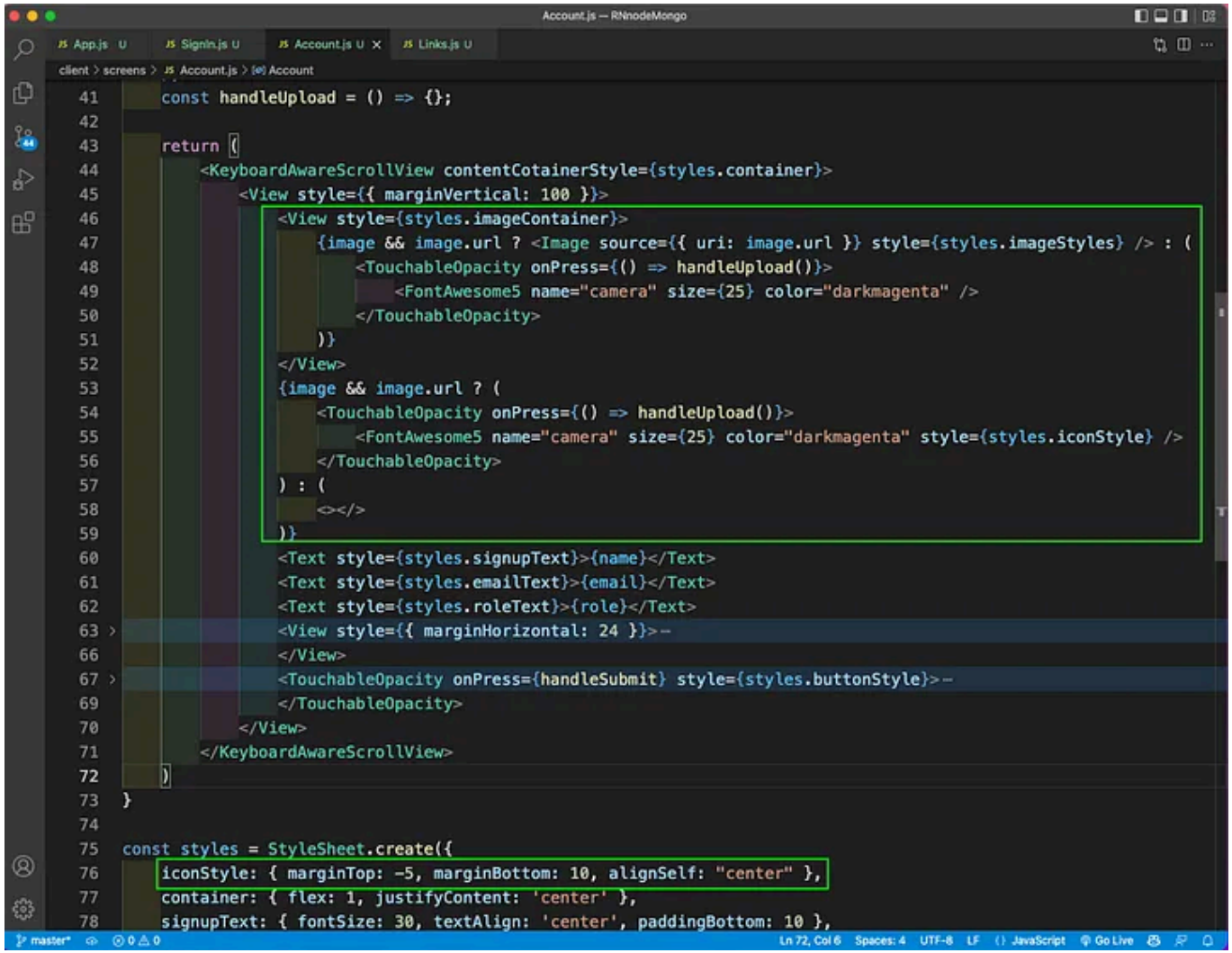

iPhone 13
iOS 15.4
9:57
Home
Account
TWD
Nabendu
Nabendu.biswas@gmail.com
Subscriber
PASSWORD
UPDATE PASSWORD

Now, we will add the logic to upload image in Account.js file. Here, we are creating a image state variable and having a hard-coded url as of now.



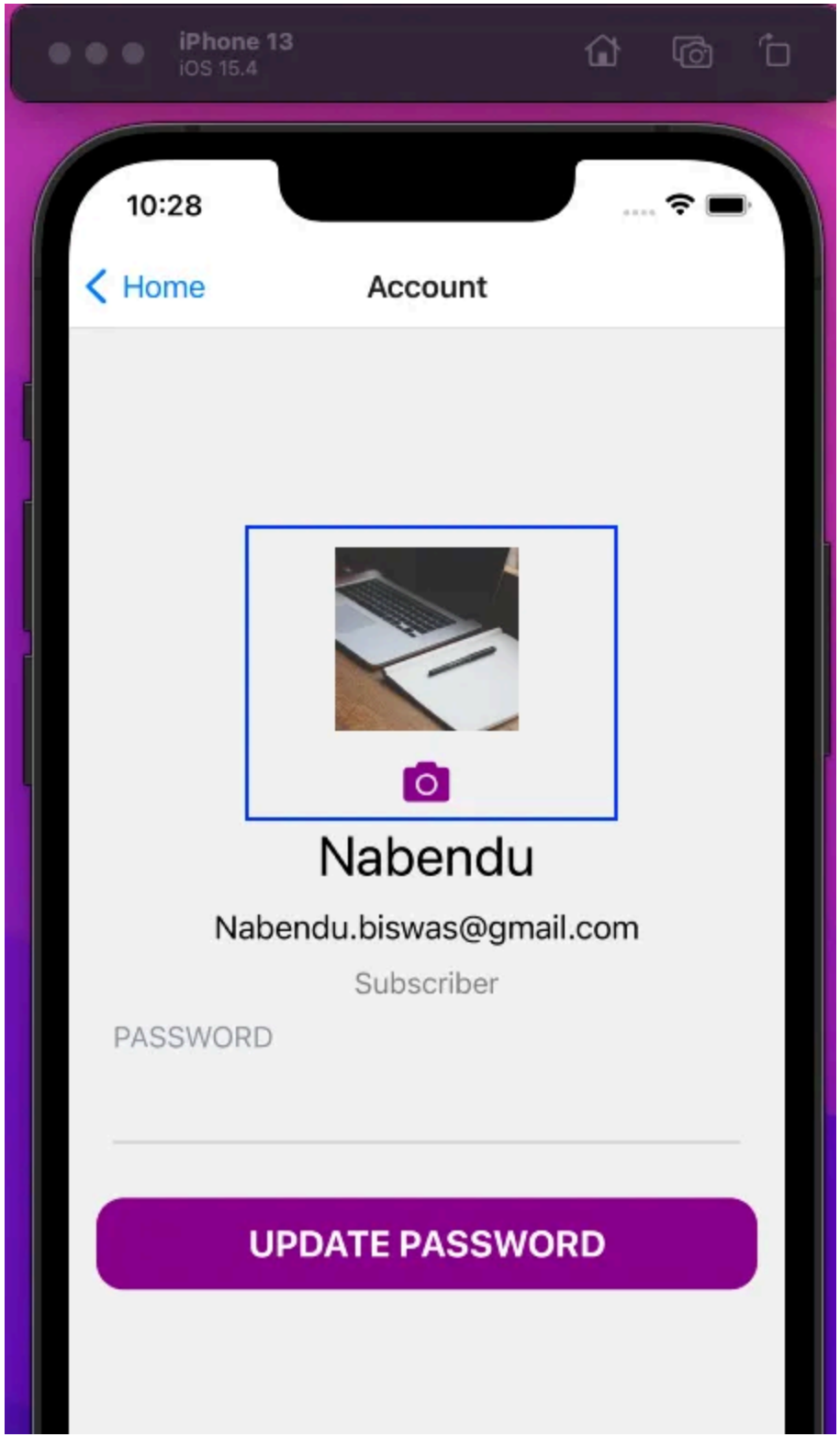
Account.js

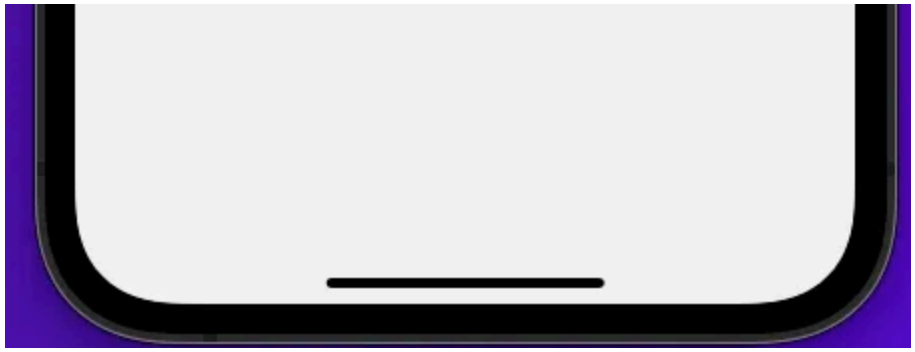
Next, we will show an image and also a camera icon to upload image. Notice that when we will not have a image for the first time, we are showing the camera icon.



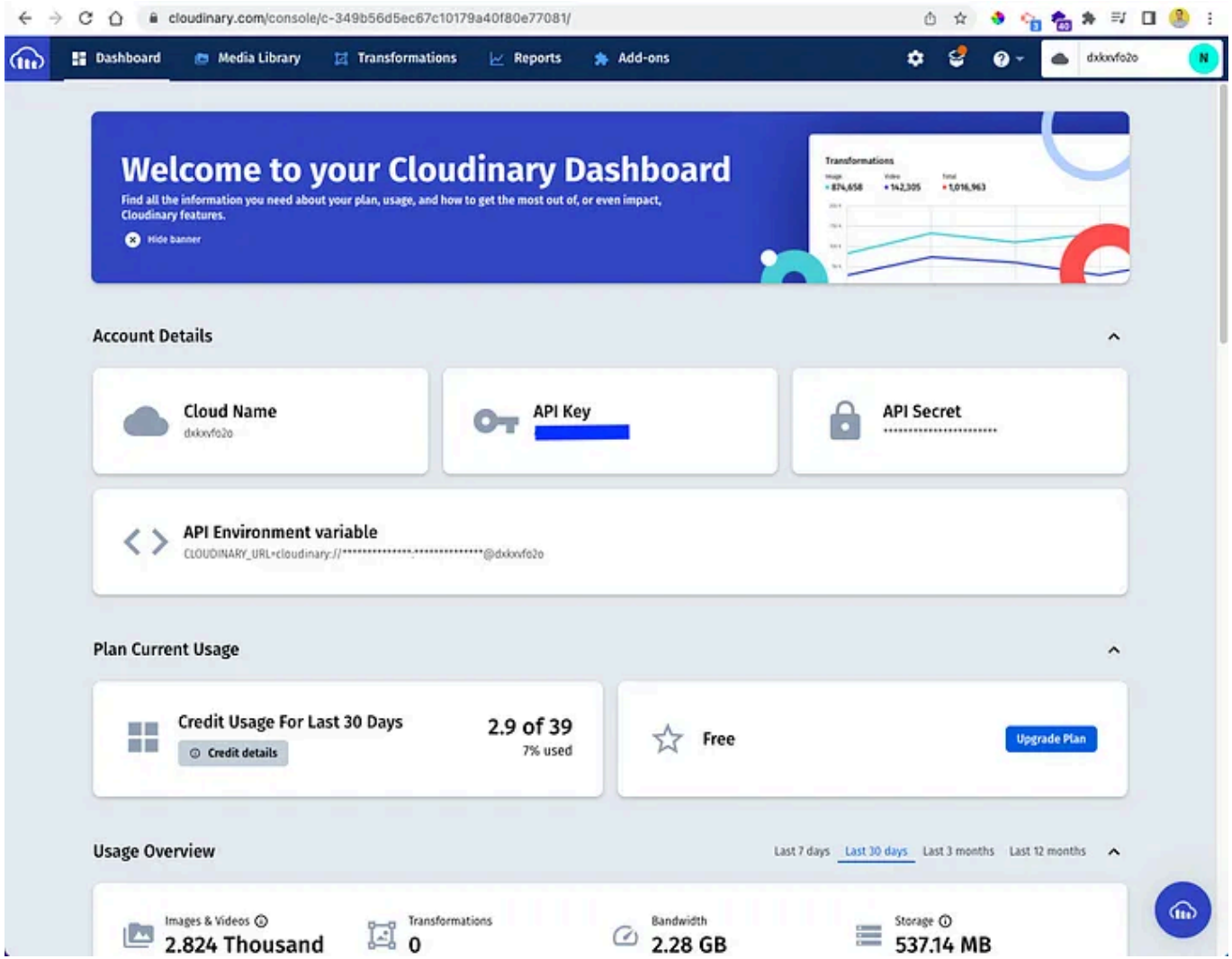
Account.js

Now, a nice image is shown in the app.

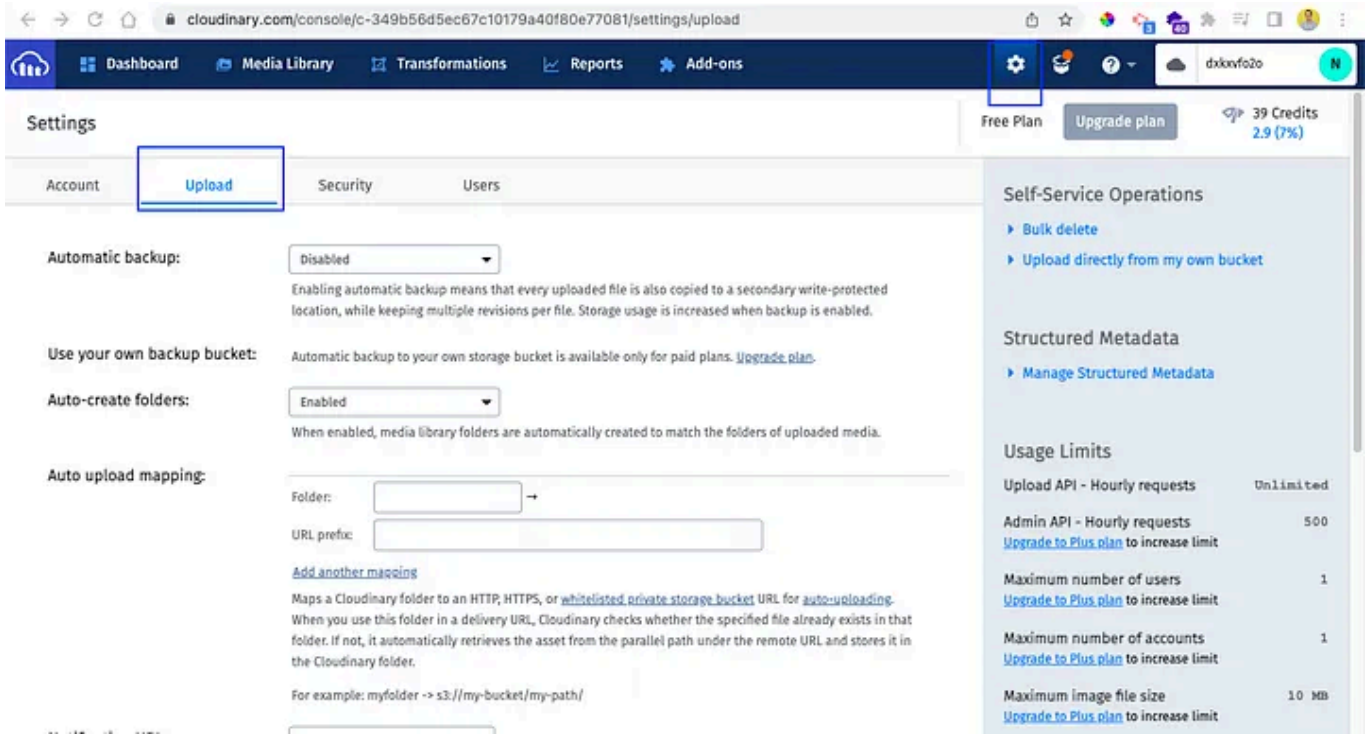

iPhone 13
iOS 15.4
10:28
Home
Account
Nabendu
Nabendu.biswas@gmail.com
Subscriber
PASSWORD
UPDATE PASSWORD



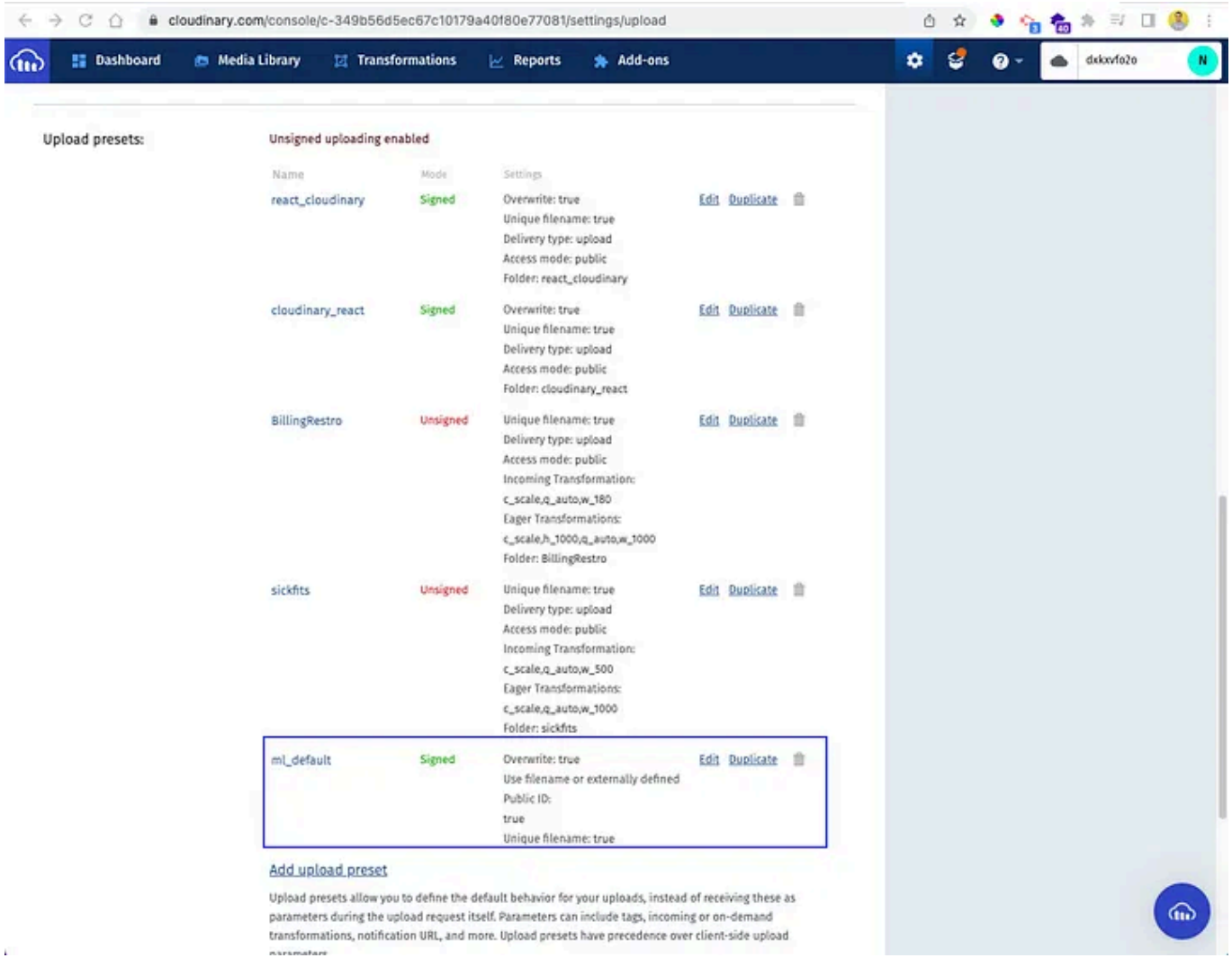
Now, we will use cloudinary to upload our images. So, login to your cloudinary account.



After that click on Settings and then Upload tab.



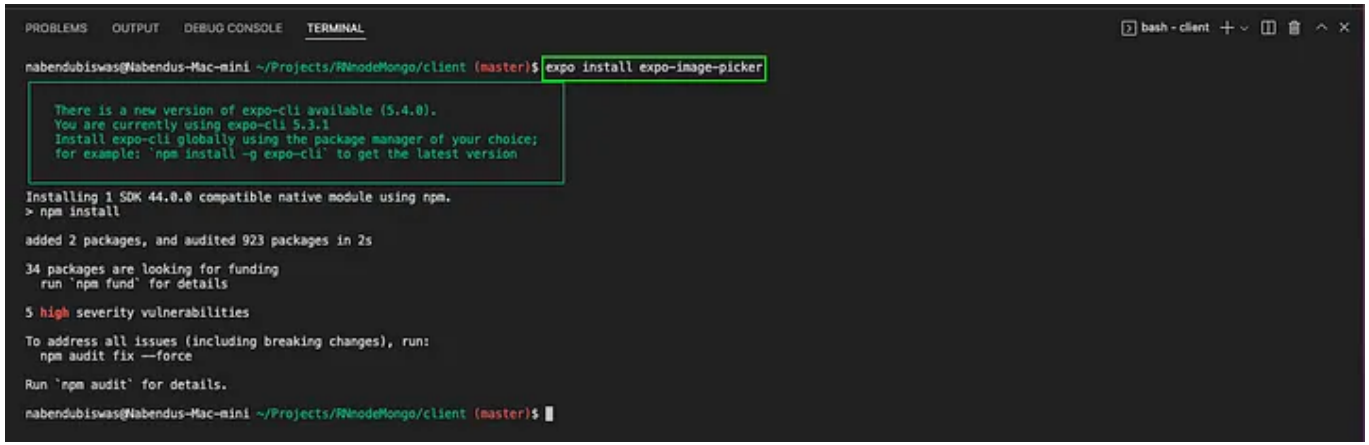
After scrolling a bit down, we will get Upload presets. Here the ml_default should be signed.



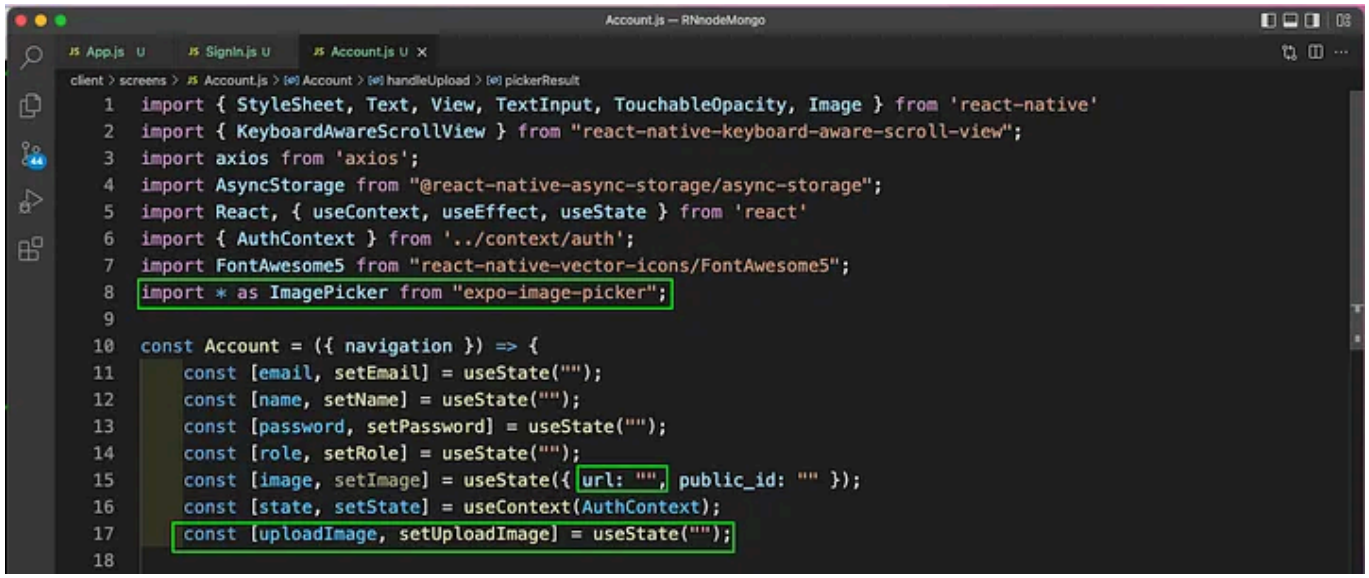
Now, we will take the cloudinary name, API key and secrets from the home page and create variables from the same in **.env** file inside server.

.env

Now, we will add expo image picker in our client side.



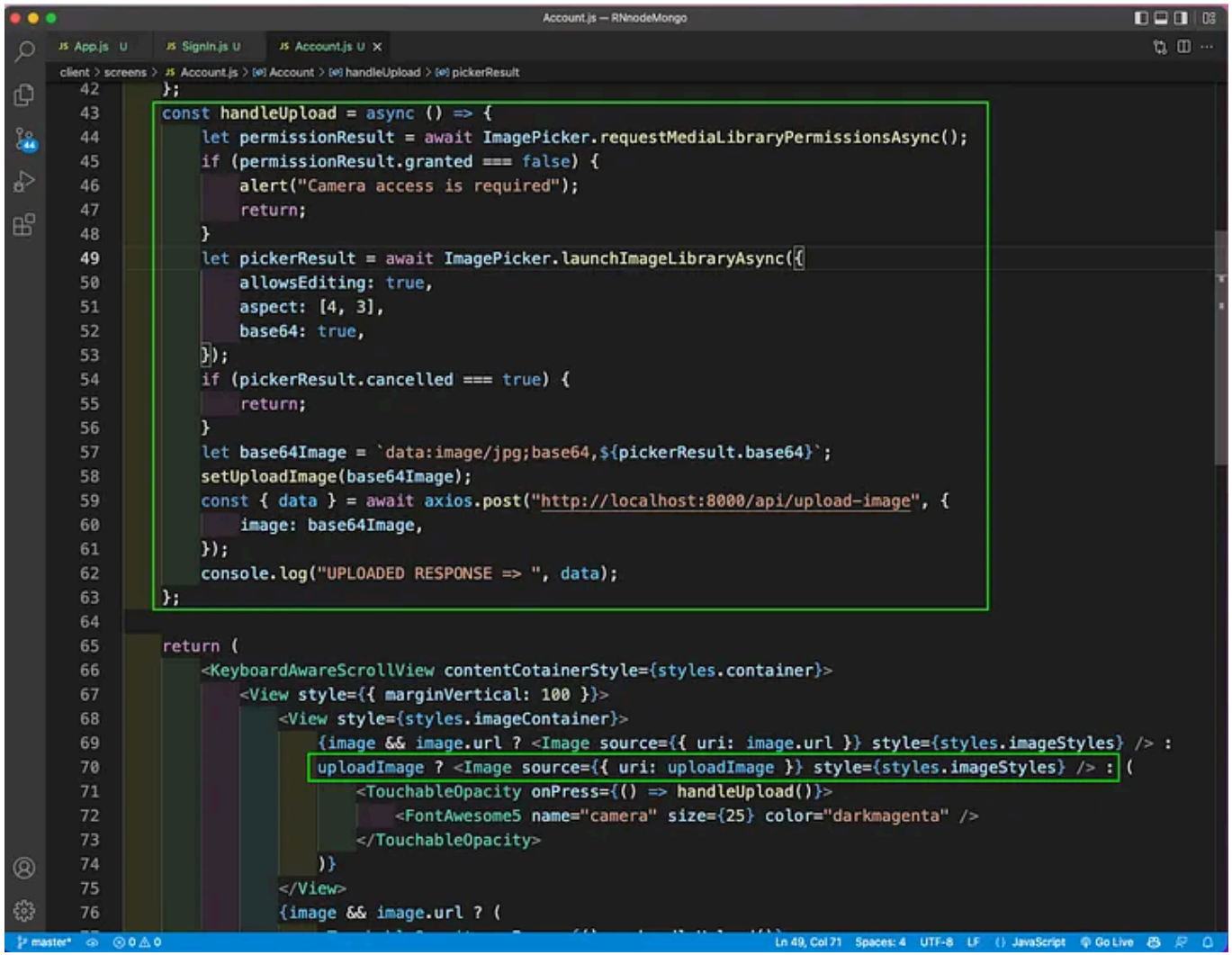
Next, we will use it in the **Account.js** file. Here, we are adding a new state variable and also removed the hard-coded url.



Account.js

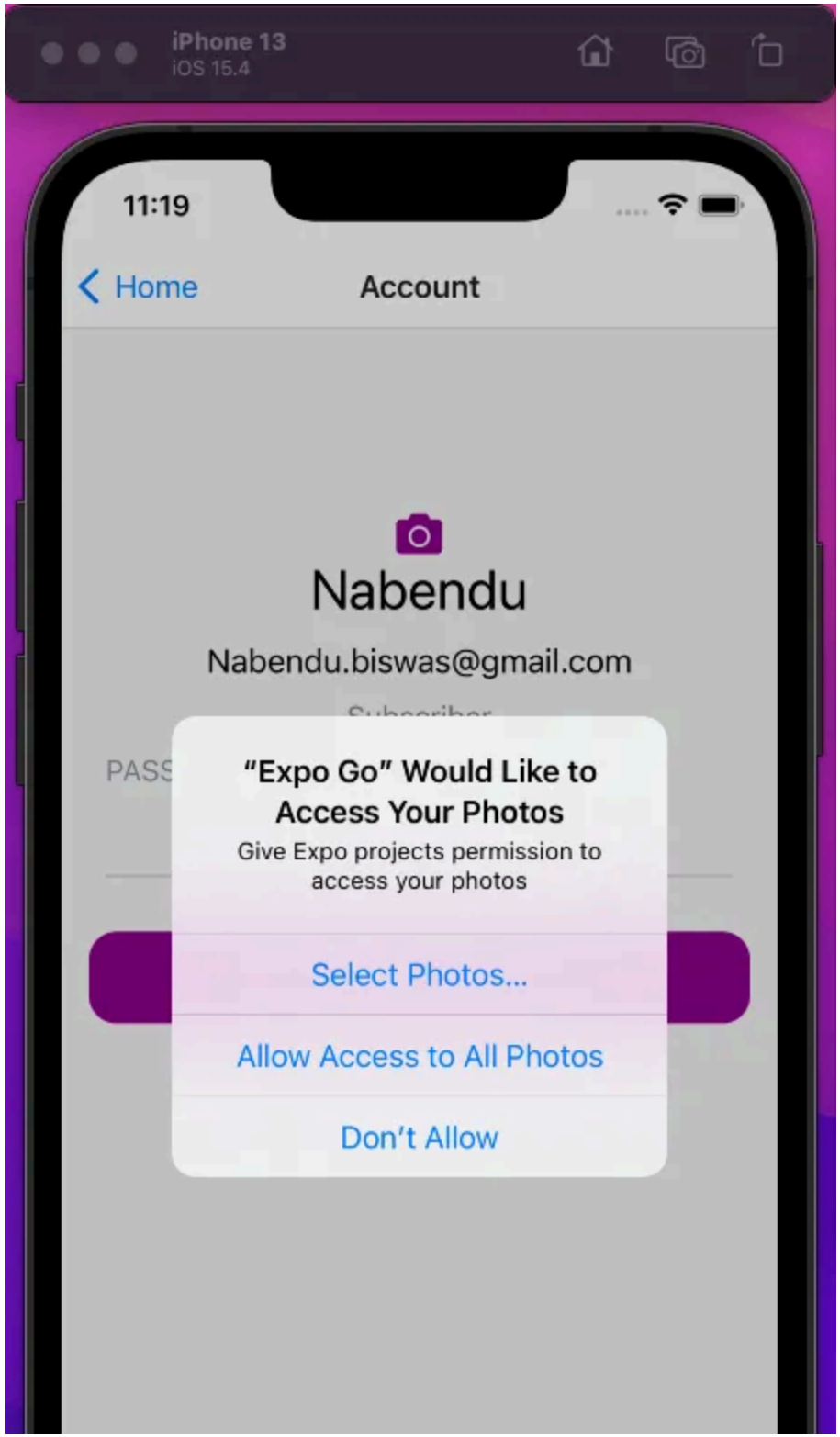
Now, we will update our handleUpload function. Here, we are using the ImagePicker functions to get the image and set the base64Image to uploadImage.

We are also passing this to the backend route, which we need to create.

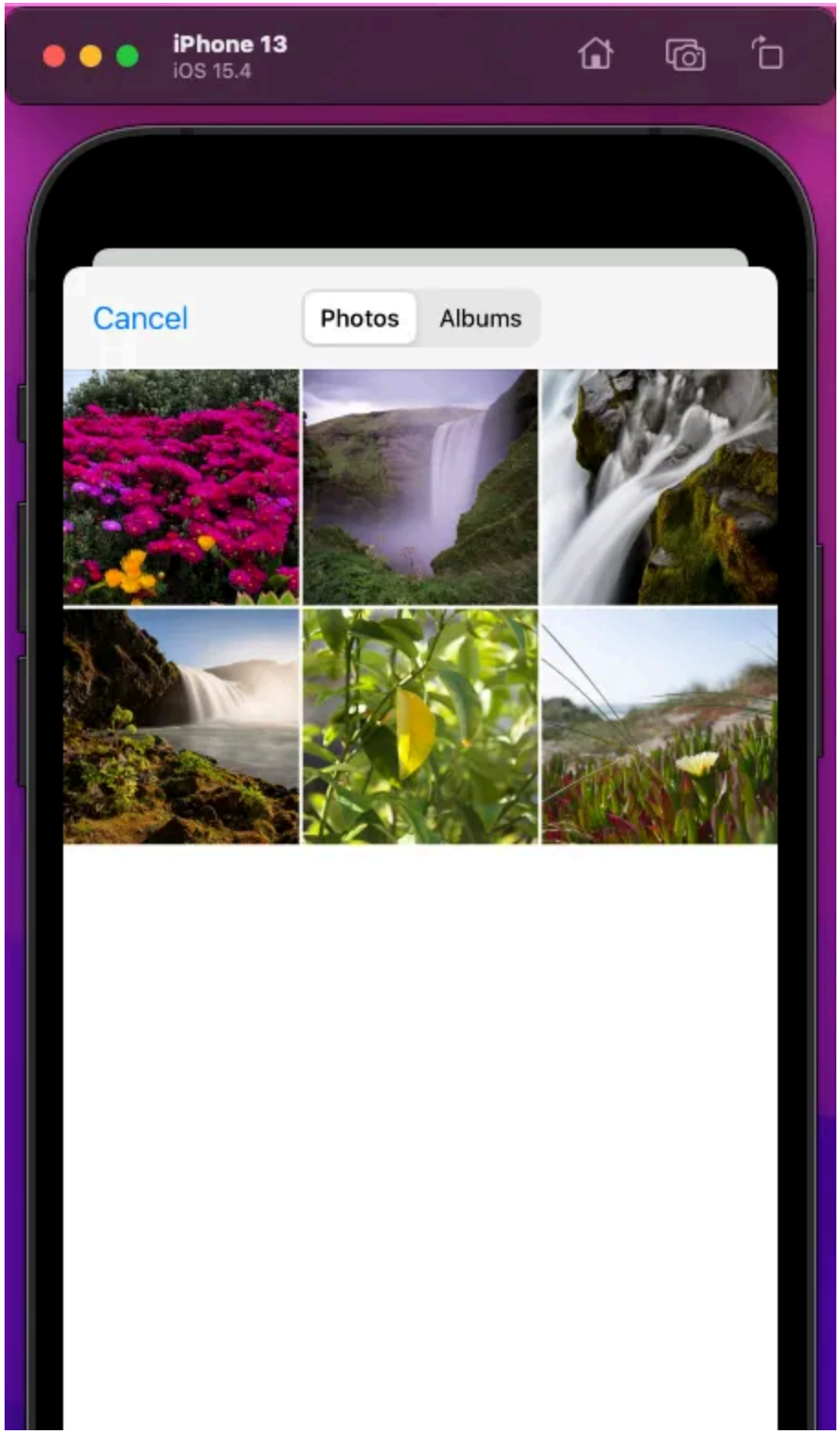


Account.js

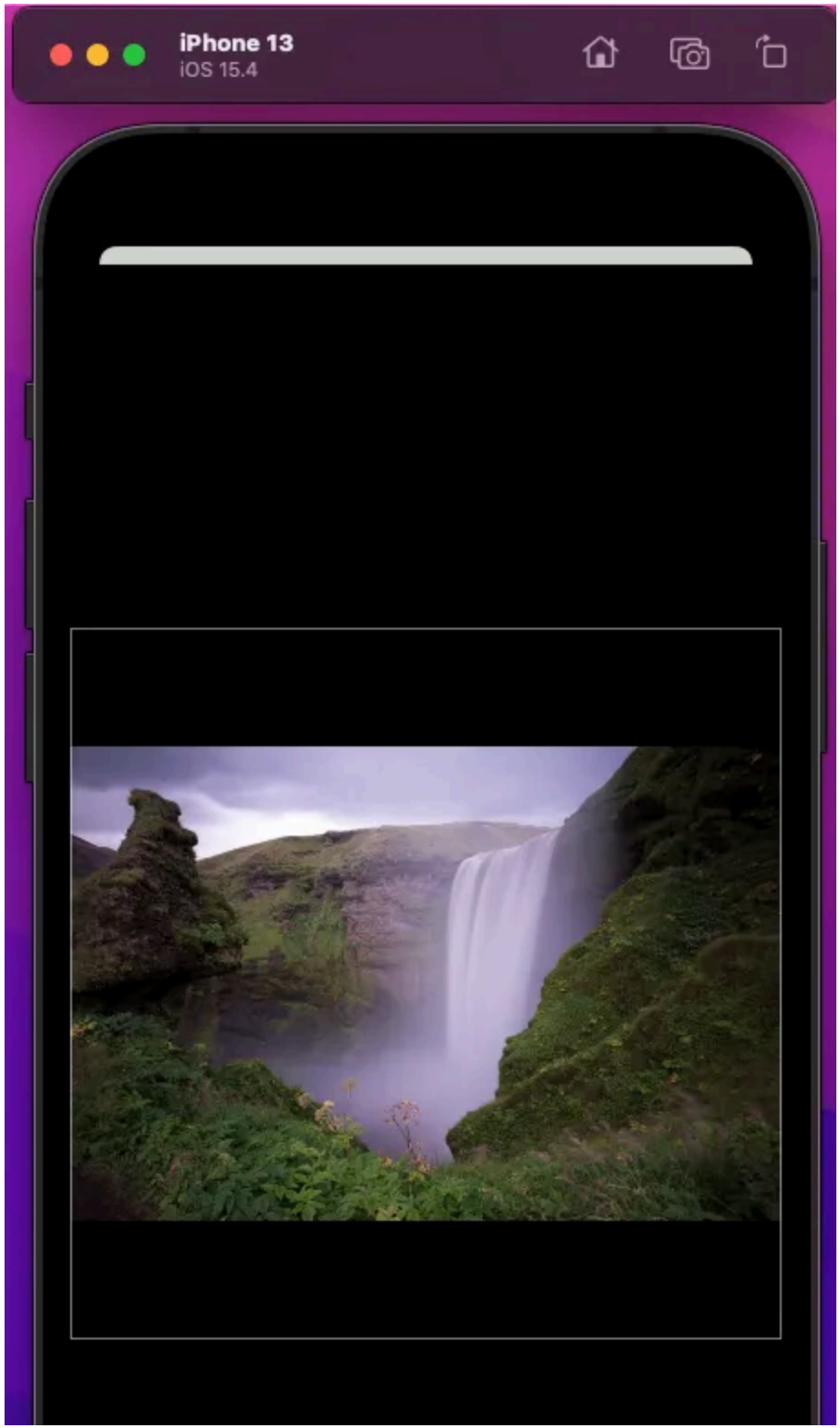
Now, in the app when we click on the Upload button, we will first be asked for permissions.

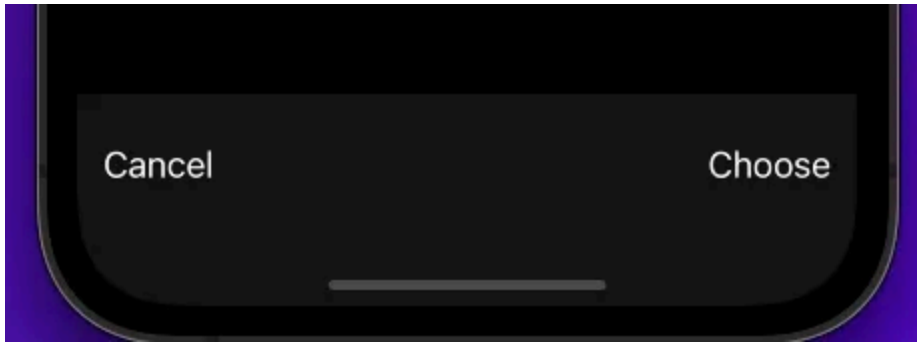

iPhone 13
iOS 15.4
11:19
Home
Account
Nabendu
Nabendu.biswas@gmail.com
"Expo Go" Would Like to Access Your Photos
Give Expo projects permission to access your photos
Select Photos...
Allow Access to All Photos
Don't Allow

Next, we will be taken to the simulator photo gallery.

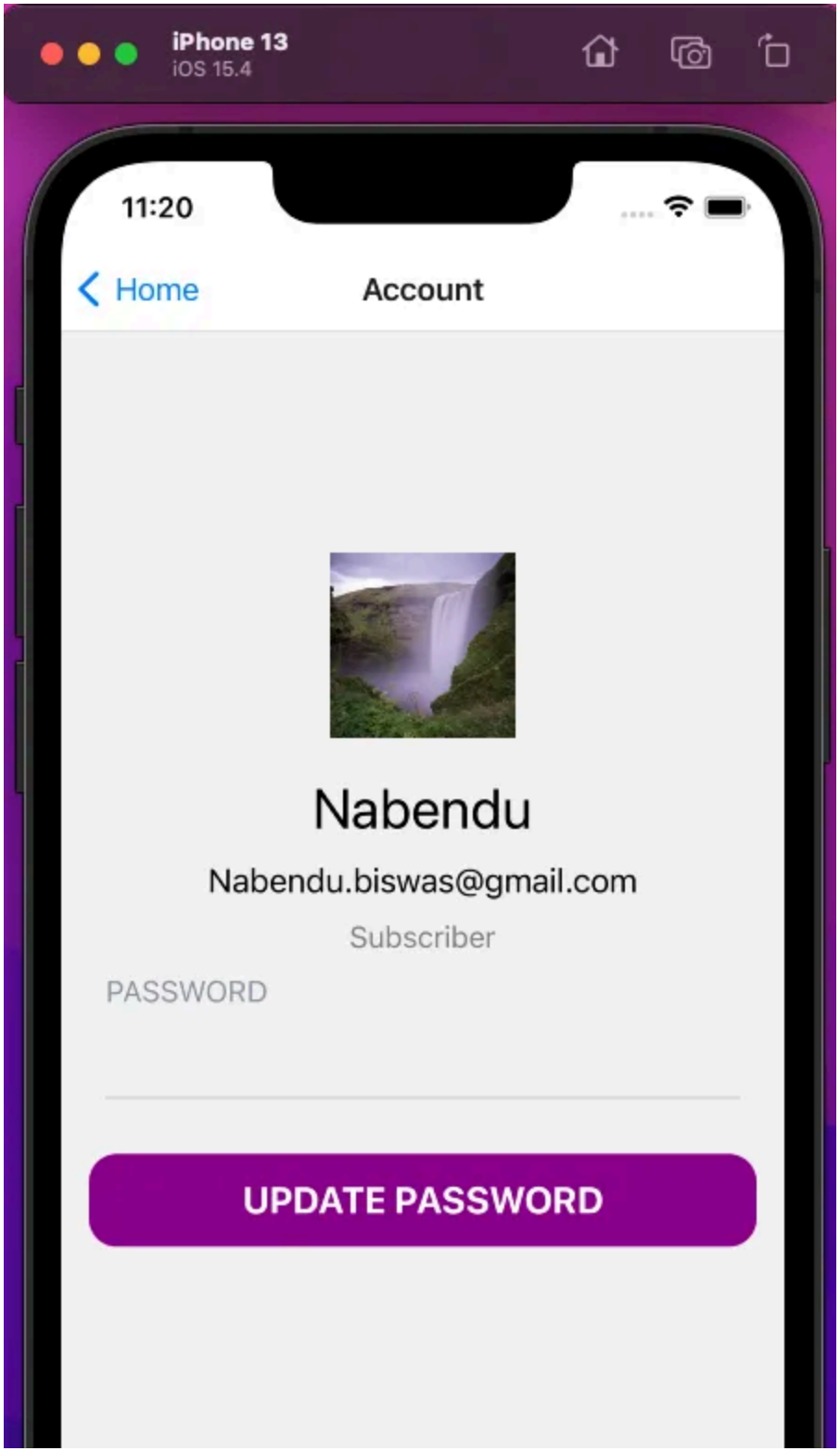

iPhone 13
iOS 15.4
Cancel
Photos
Albums

Here, on picking a image we will be taken to the next scree. Here, press on Choose.


iPhone 13
iOS 15.4

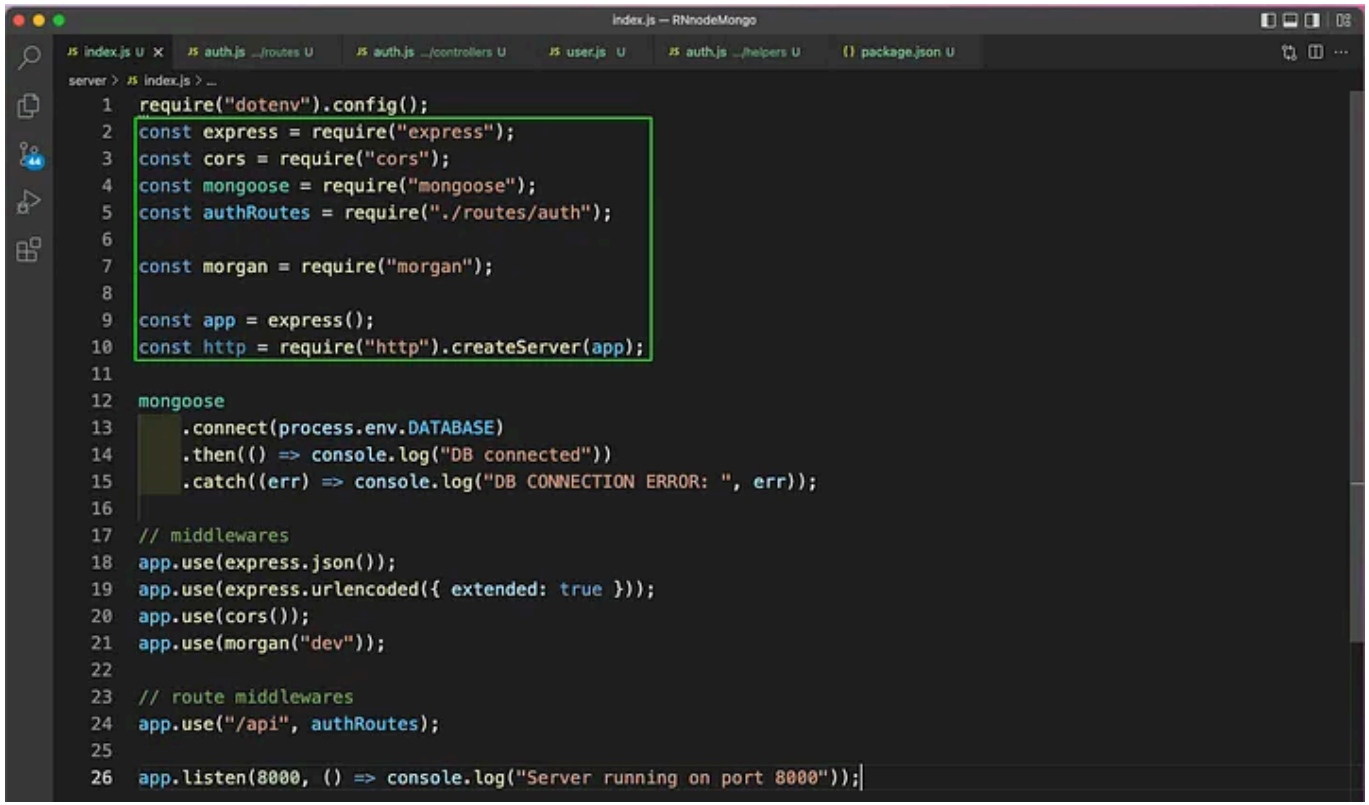


Now, our uploaded image will be shown in the component.


iPhone 13
iOS 15.4
11:20
Home
Account
Nabendu
Nabendu.biswas@gmail.com
Subscriber
PASSWORD
UPDATE PASSWORD

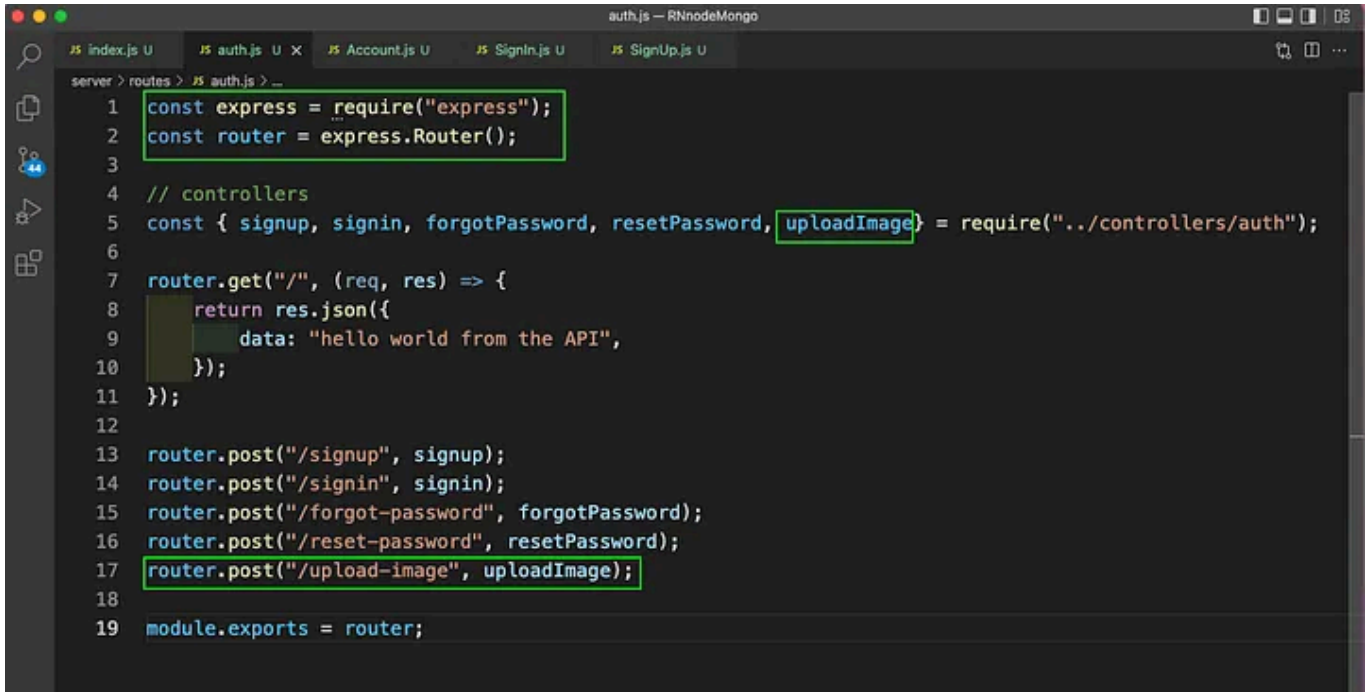
Next, we need to create the upload-image route in the server. But first we will change all of our imports to the old NodeJS kind import, because some old node_modules were creating problem.

First update them in **index.js** file.



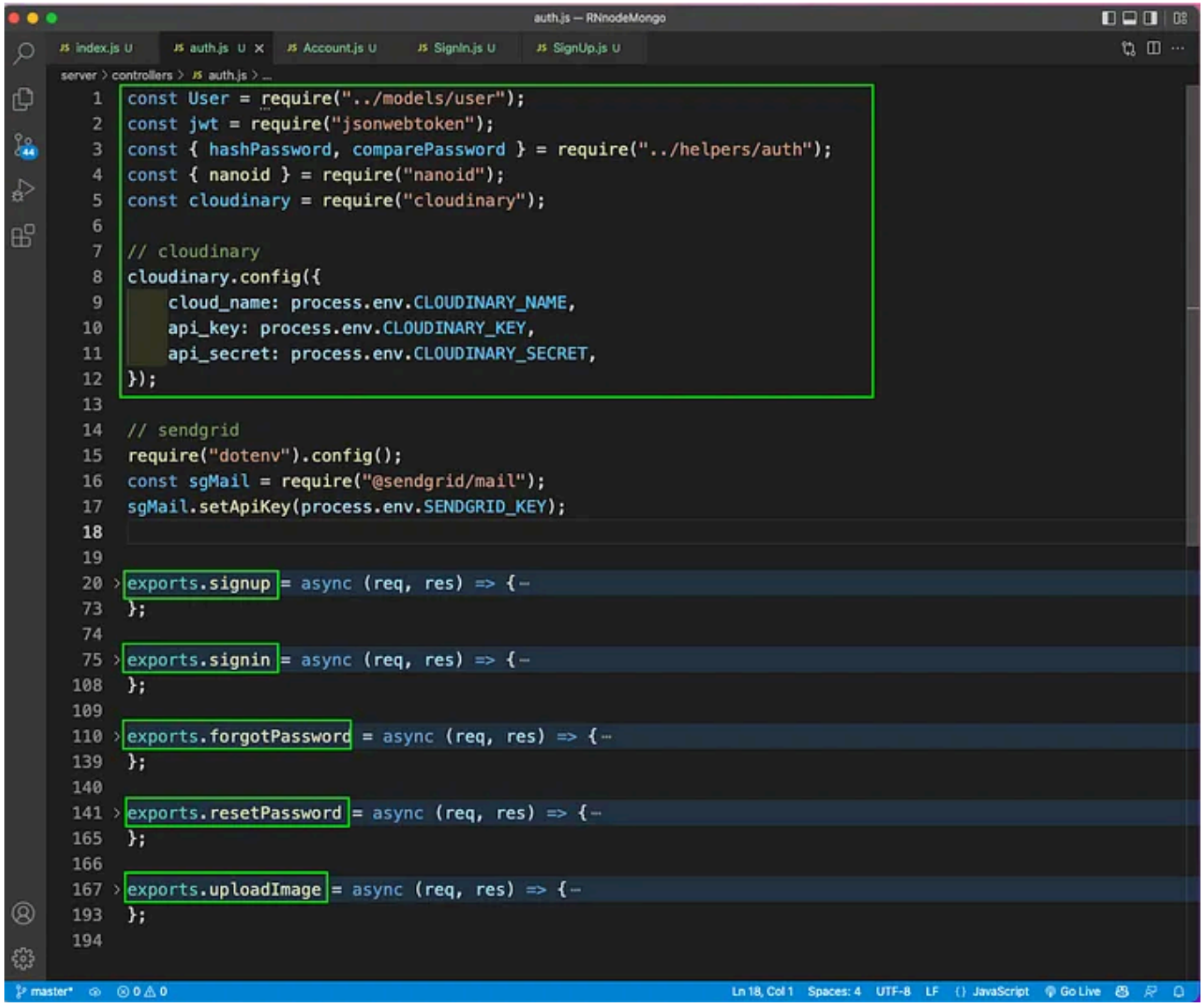
index.js

Next, we will update them in **auth.js** file in **routes** folder. Here, we are also creating a new route for **upload-image**.



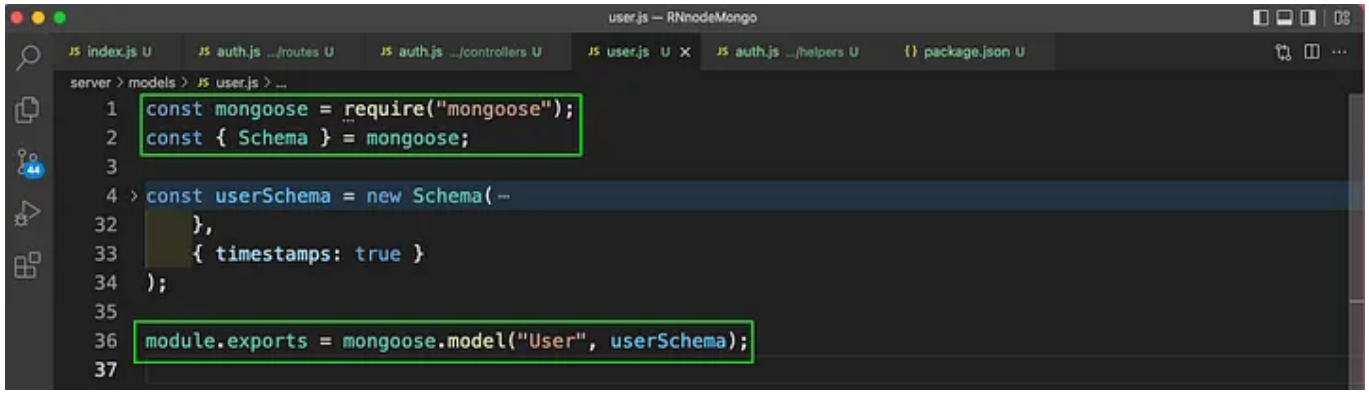
auth.js

Next, we will update the imports in **auth.js** for the **controllers.** We are also adding the cloudinary config.



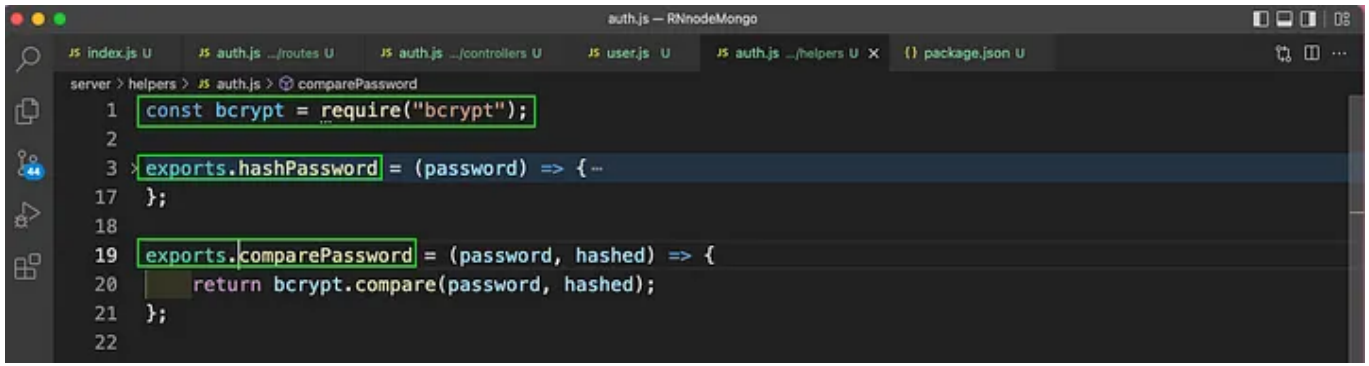
auth.js

Now, we will update the imports in **user.js** for the **models.**



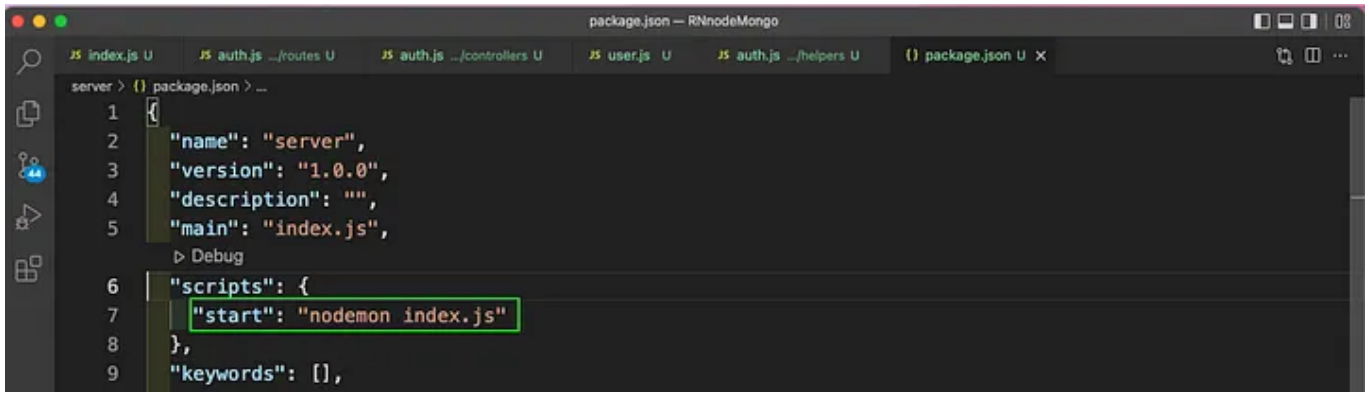
user.js

Next, we will update imports in **auth.js** for the **helpers.**



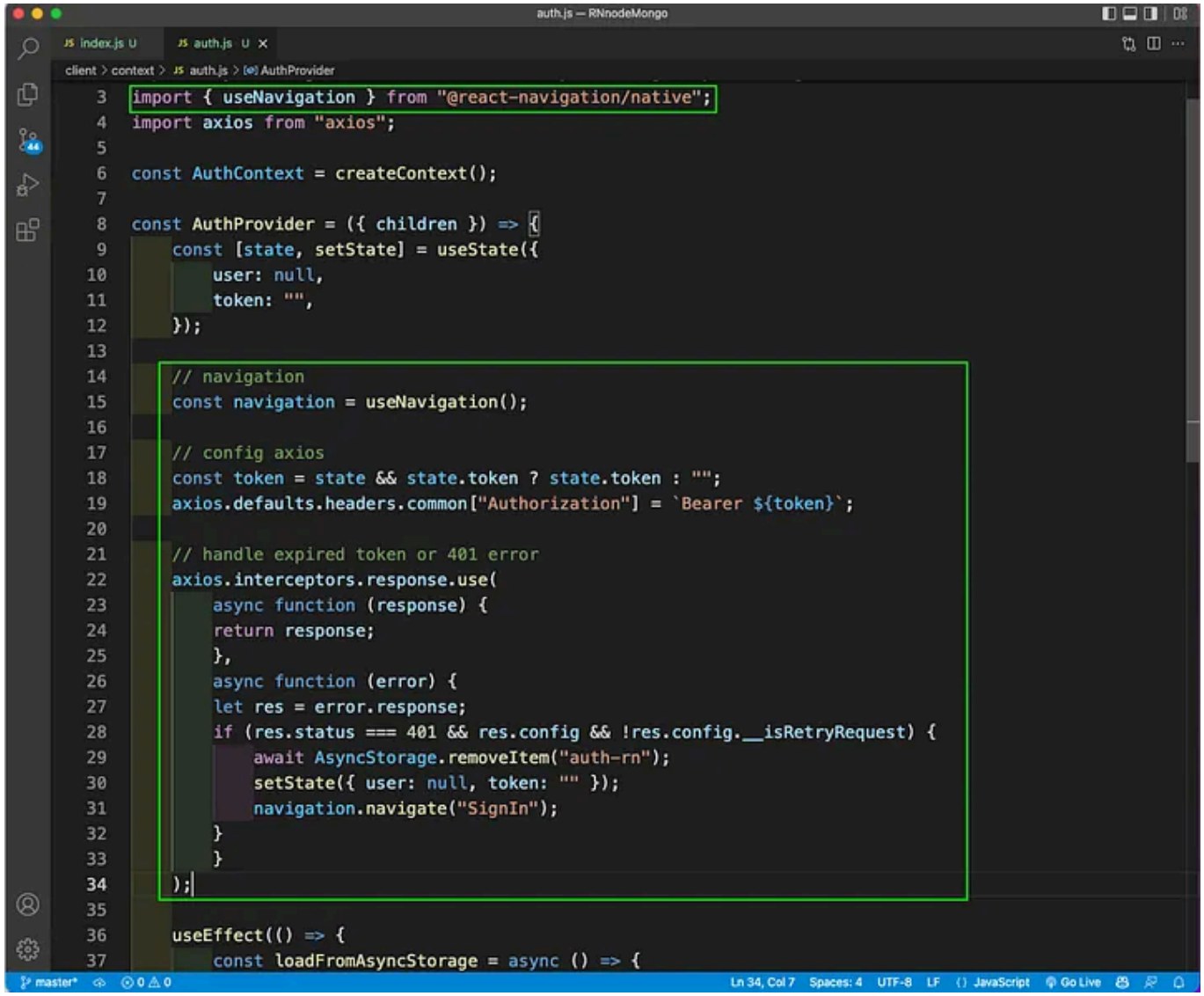
auth.js

We are also changing our nodemon in **package.json** file.



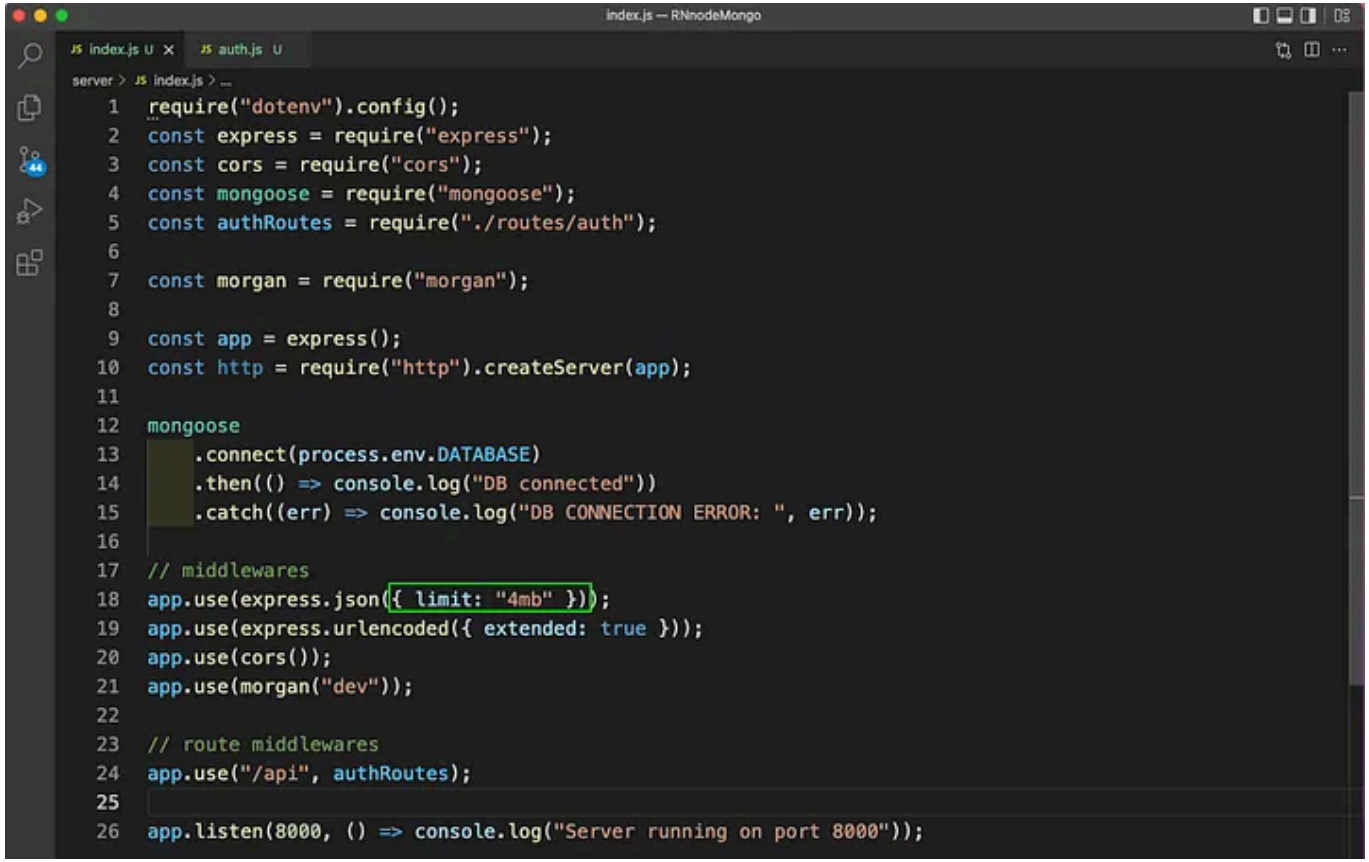
package.json

Now, in the **auth.js** in the **context** folder in client, we are adding the required header for POST request. We have also added logic to handle token expiry.



auth.js

We were getting an error for json request, so we have to update the limit of it in **index.js** file inside the server.



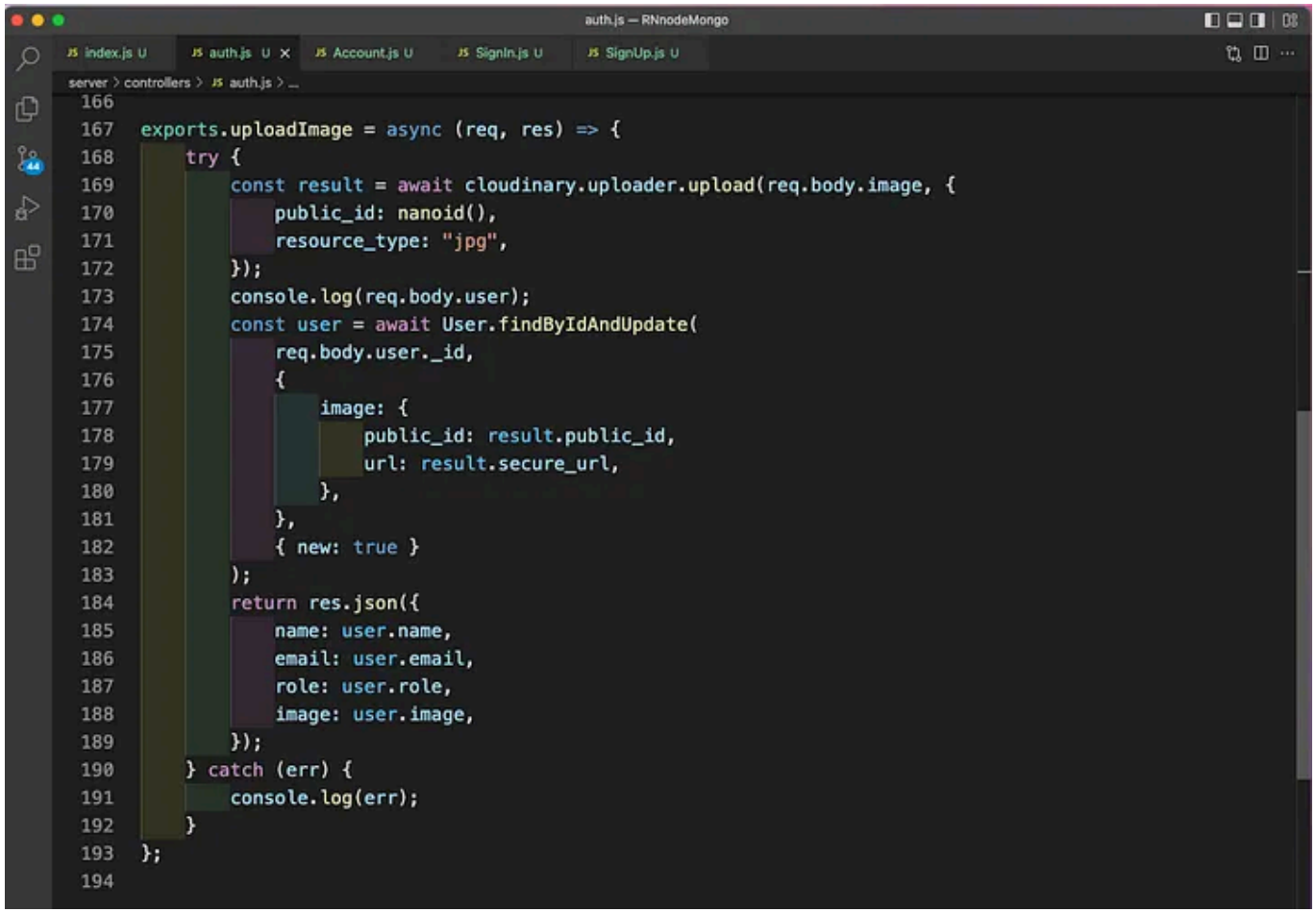
index.js

Next, we will add the cloudinary package in the server.

Now, we will update our **uploadImage** function in **auth.js** inside **controllers.**

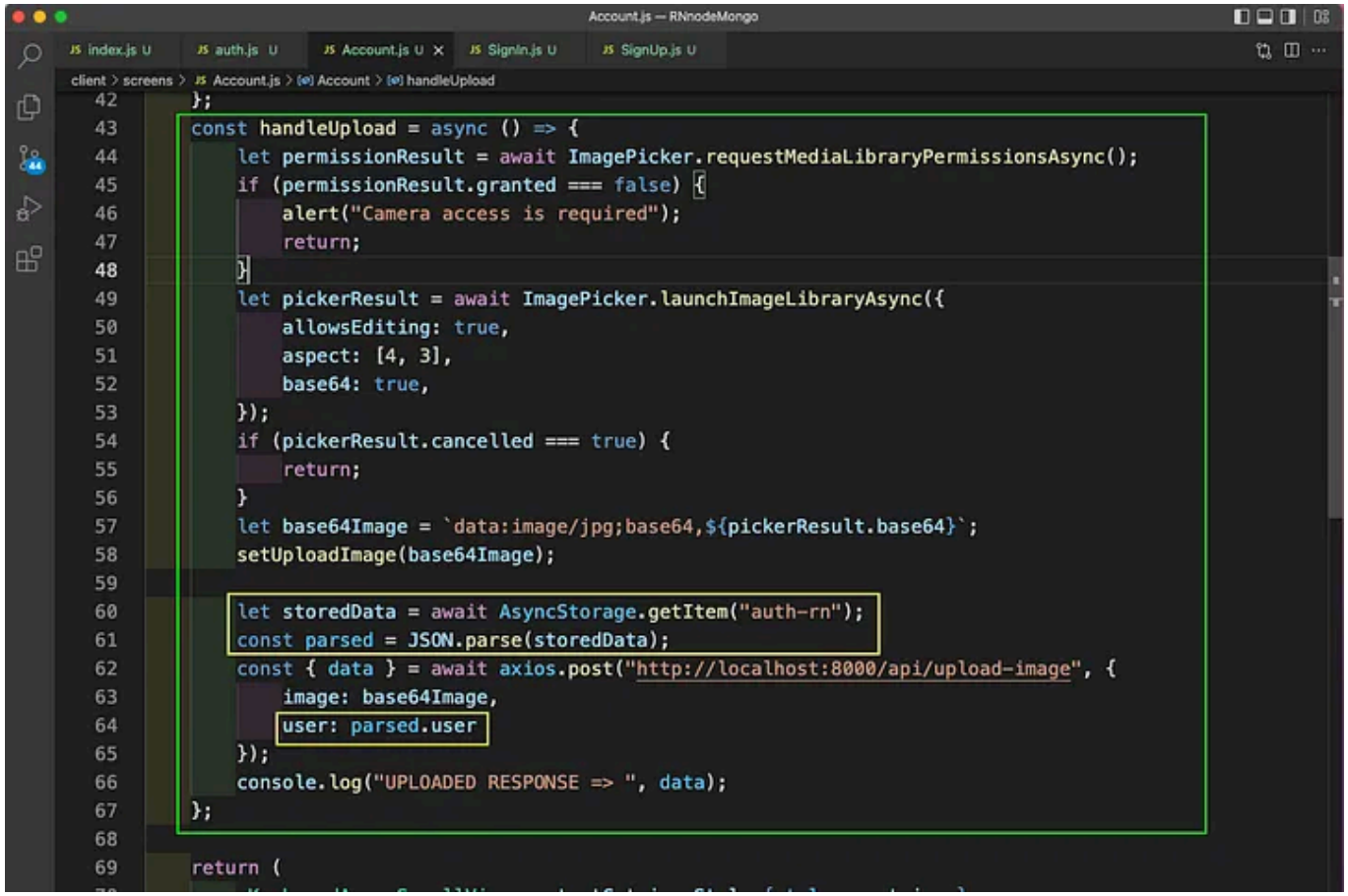
Here, we are using cloudinary to upload our image to cloudinary. Also, getting a unique id with nanoid().

After that updating the record to include the public_id and url from cloudinary.



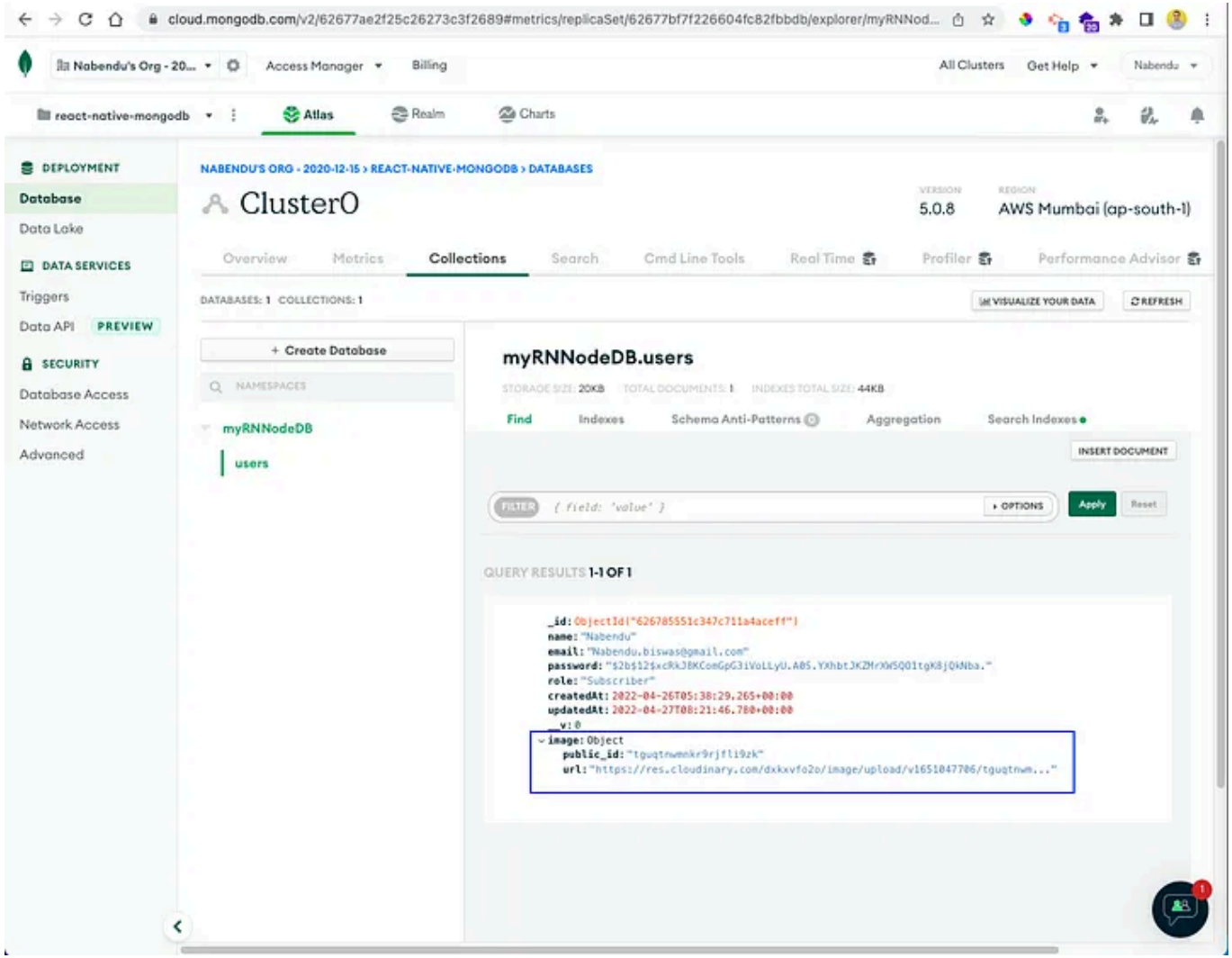
auth.js

Next, we will update our handleUpload function to get the user from the stored Async storage and pass it to upload-image endpoint in **Account.js** file.

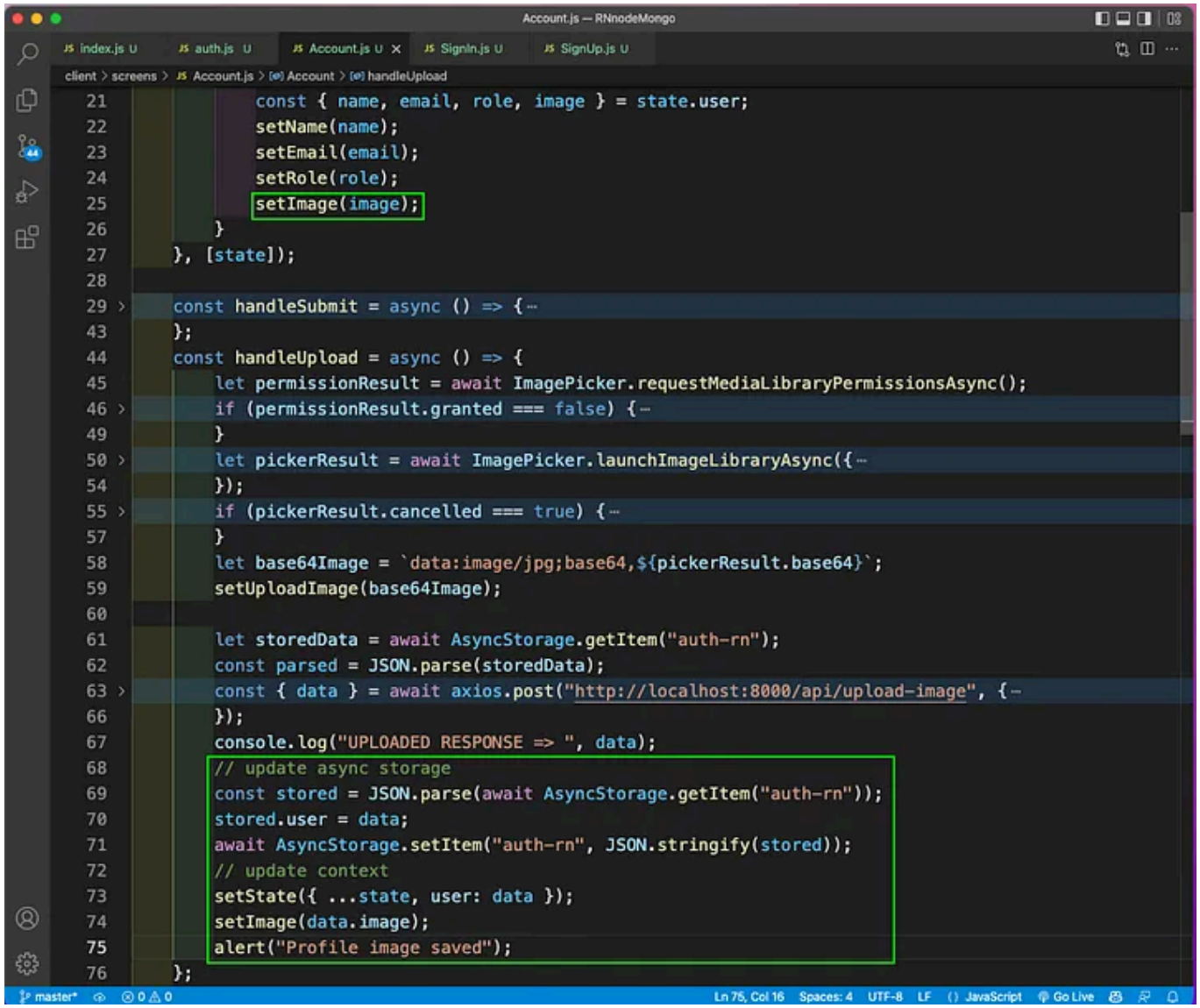


Account.js

Now, when we upload an image, it is updated in the database.

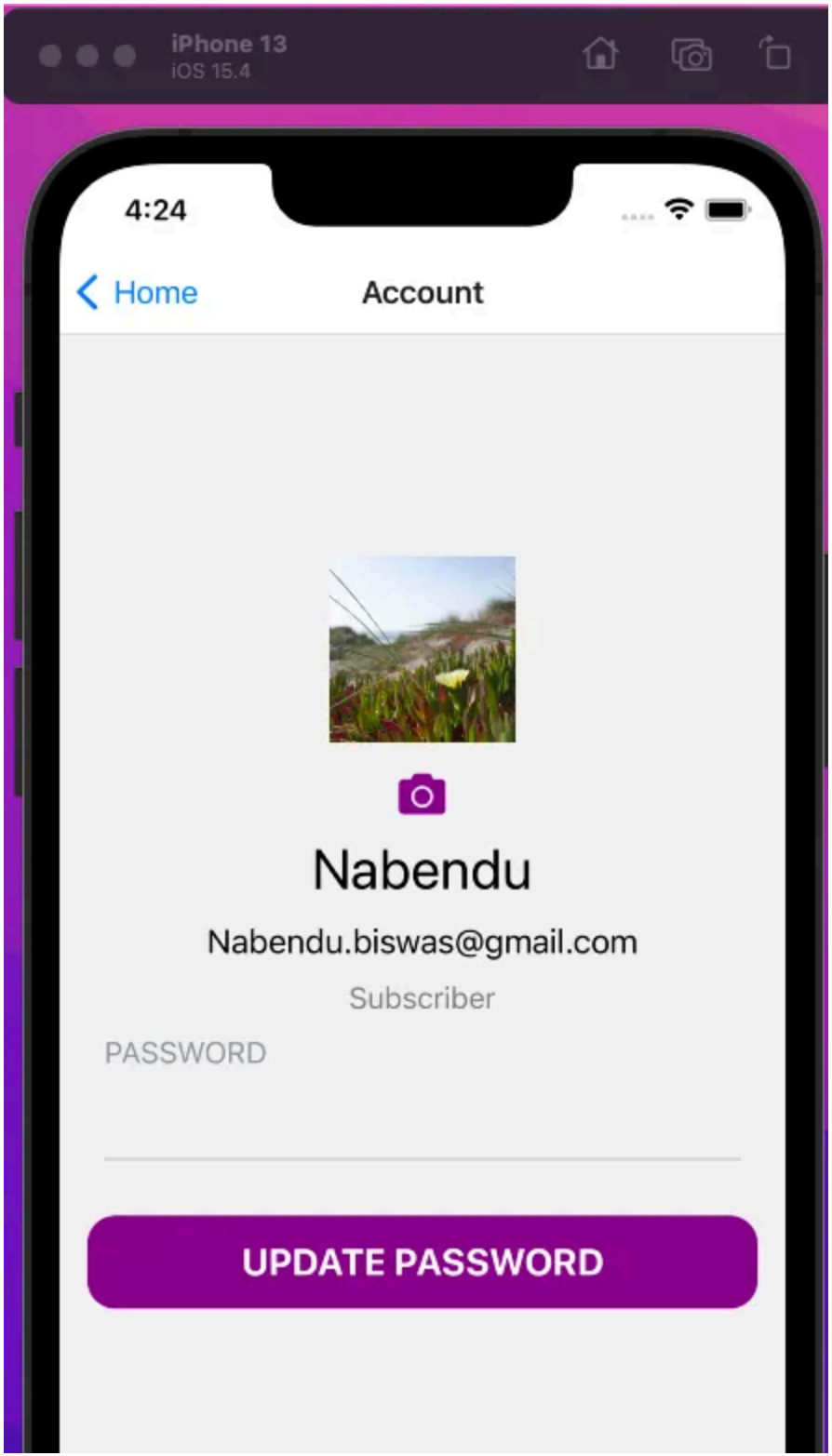


Next, we will update our Async storage also so that the user image is saved, even if we refresh the device.



Account.js

Now, our image will be shown permanently.


iPhone 13
iOS 15.4
4:24
Home
Account
Nabendu
Nabendu.biswas@gmail.com
Subscriber
PASSWORD
UPDATE PASSWORD

This completes part-2 of the series. See you soon in part-3

React Native | React Native Development | React Native Developers | React Native App | React Native Tutorial

Written by Nabendu Biswas

Follow

903 Followers

Architect, ReactJS & Ecosystem Expert, Youtuber, Blogger

More from Nabendu Biswas

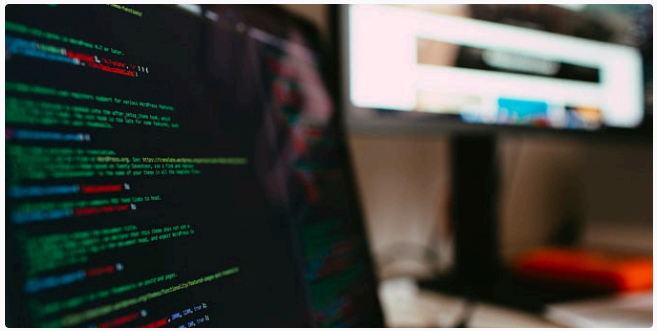
Nabendu Biswas

React Testing with Jest and RTL in TypeScript

In this post we are going to learn to test a React app. Here, we are going to use Jest an...

15 min read · Jul 8, 2023

17 2



Nabendu Biswas

NextJS 14 Blog app

In this post we will learn to create a simple blog app using the latest NextJS 14. This blo...

15 min read · Mar 2, 2024

144 3

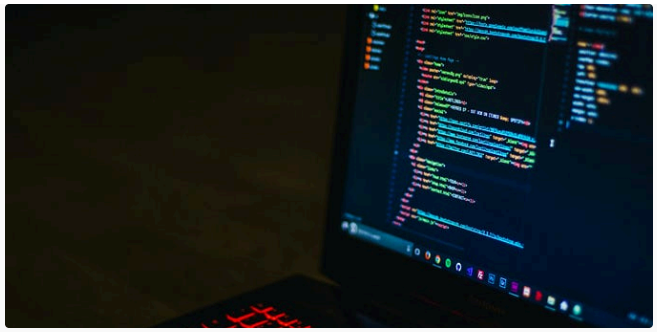
Nabendu Biswas

React Native Project with NodeJS and MongoDB- Part 1

In this post we will create a major project with React Native, which will have data stored in ...

12 min read · Apr 26, 2022

92 1



Nabendu Biswas

TypeScript Interview Questions

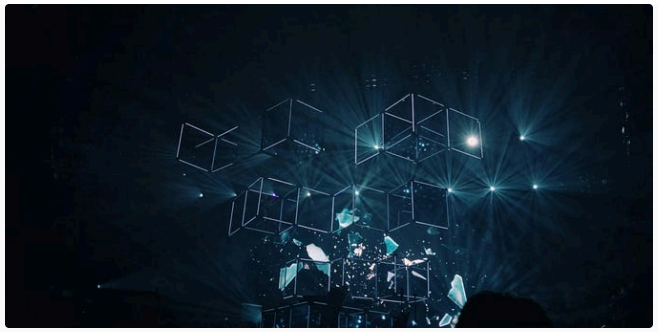
Nowadays most modern ReactJS and NodeJS apps are created using TypeScript. Angular...

10 min read · Apr 14, 2024

28 1

See all from Nabendu Biswas

Recommended from Medium



Onix React

Release React Native 0.74.0

React Native 0.74 has been released, bringing with it a slew of updates designed to enhanc...

4 min read · Apr 23, 2024

578

Matheshyogeswaran

Building Authentication in React Native with Firebase: A Step-by-...

Here, we are going to delve into setting up React Native with Firebase and then setting...

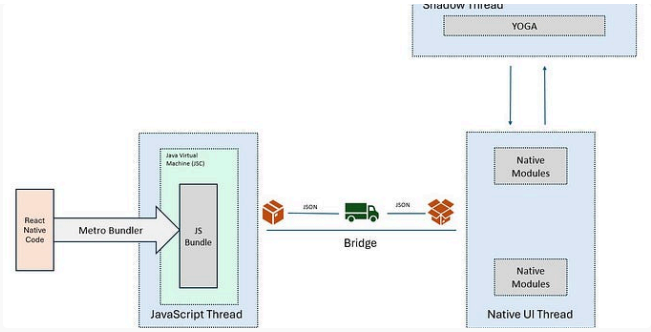
4 min read · Feb 22, 2024

2 1

Lists

Stories to Help You Grow as a Software Developer

19 stories · 1027 saves



Sanjana Shivananda

Mastering User Authentication: MERN Stack Login Page (Part 1—...

Introduction

9 min read · Dec 19, 2023

7

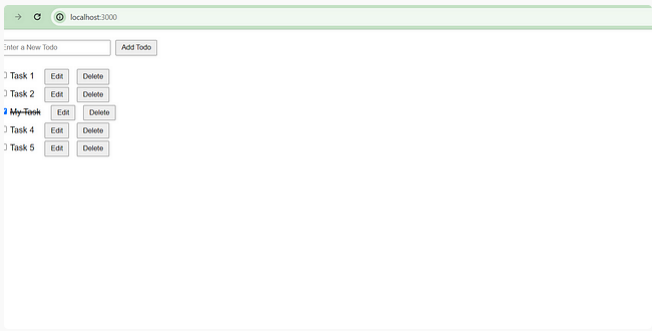
Harshit Kishor

React Native New Architecture

React Native New Architecture vs Old Architecture

8 min read · 6 days ago

5



Charisma Kurniawan Aji

Make your React Native code cleaner

Feeling overwhelmed by a cluttered mess of React Native code? As your app expands, so...

Maitrik Thakkar

Simple Todo App using Redux Toolkit

Install React Project

3 min read · Mar 8, 2024

46 1

3 min read · Feb 25, 2024

2

See more recommendations