

Expo Router

Introduction

Expo Router

-  **Structure of the app and Routing**
-  **Navigating between different screens**
-  **Basic React Native Elements**
-  **Basic Expo API**
-  **How Pass Parameters from one Screen to another**
-  **How to Create Dynamic Routes**
-  **How to Create a 404 of the app**
-  **Stack Navigation**
-  **Tab Navigation**
-  **Drawer Navigation**

Expo Router



Installation of Expo Router



```
npx create-expo-app@latest
```



<https://docs.expo.dev/>



[https://docs.expo.dev/router/layouts
/#native-layouts](https://docs.expo.dev/router/layouts/#native-layouts)

Expo Router



Lets Clean Up

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure. The `index.js` file under the `app` folder is open in the editor.
- Editor View:** Displays the code for `index.js`. The code defines a `Page` component that returns a `View` containing a `Text` component with the text "Home".

```
import { View, Text } from 'react-native'
import React from 'react'

export default function Page() {
  return (
    <View>
      <Text>Home</Text>
    </View>
  )
}
```
- Terminal View:** Shows command-line shortcuts and logs.
 - Shortcuts:
 - Press `j` | open debugger
 - Press `r` | reload app
 - Press `m` | toggle menu
 - Press `o` | open project code in your editor
 - Press `?` | show all commands
 - Logs:
 - Logs for your project will appear below. Press `Ctrl+C` to exit.
 - Opening on iOS...
 - Opening exp://192.168.0.104:8881 on iPhone 14 Pro Max
 - Press `?` | show all commands
 - Bottom status: Unable to resolve asset "./assets/images/icon.png" from "icon" in your app.json or app.config
 - Bundling complete 3917ms

Expo Router



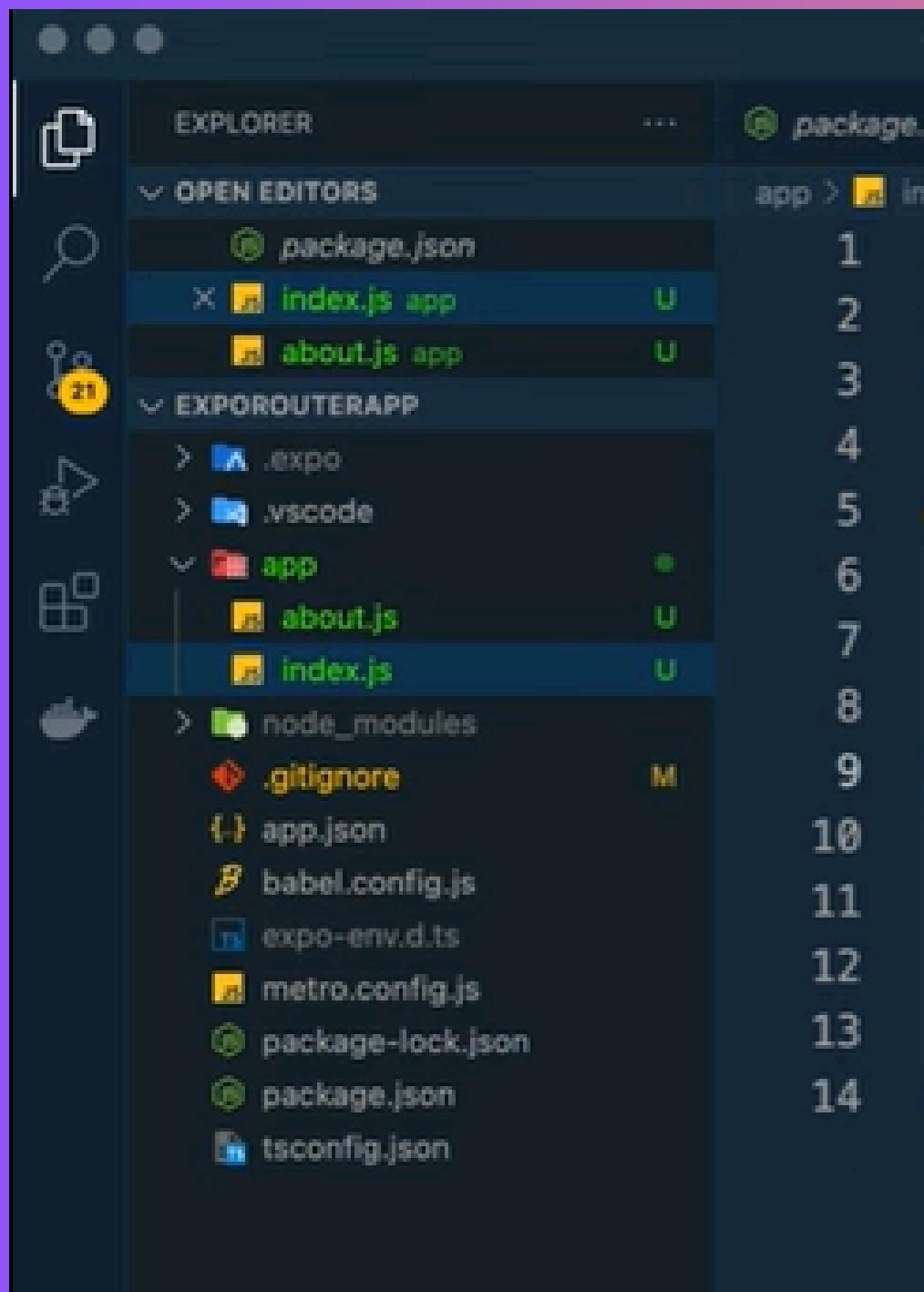
Creating Pages/Screens

We use file based routing same as of Next JS.

index.tsx => Represents the Home Page

about.tsx => Represents the About Page

<https://docs.expo.dev/router/create-pages/>



Expo Router



Routing Between Pages

Use of Link Component

<https://docs.expo.dev/router/navigating-pages/>

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure with files like package.json, index.js, about.js, .expo, .vscode, app, node_modules, .gitignore, app.json, babel.config.js, expo-env.d.ts, metro.config.js, package-lock.json, package.json, and tsconfig.json.
- Editor View:** The file index.js is open, showing the following code:

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Link } from 'expo-router'
4
5 export default function Page() {
6   return (
7     <View>
8       <Text>Home Page</Text>
9       <Link href={'/about'}>
10         <Text>Go to About Page</Text>
11       </Link>
12     </View>
13   )
14 }
```

Expo Router



Routing Between Pages Use of Button Component

<https://docs.expo.dev/router/navigating-pages/#buttons>

The screenshot shows a VS Code interface with the following details:

- Explorer:** Shows the project structure with files like package.json, index.js, and about.js.
- Editor:** The file about.js is open, showing the following code:

```
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import {useRouter} from 'expo-router';

4

5 export default function About() {
6   const router = useRouter();
7   return (
8     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
9       <Text>About Page</Text>
10      <Button onPress={() => router.back()} title='Go Back' />
11    </View>
12  )
13 }
```
- Terminal:** Shows the output of a command, likely related to image processing or caching, with several lines of log output.

Expo Router



Routing Between Pages

Use of Button Component

The screenshot shows a VS Code interface with the title bar "expoRouterApp". The Explorer sidebar on the left lists files and folders, including "package.json", "index.js", "about.js", ".expo", ".vscode", and "node_modules". The "OPEN EDITORS" section shows "about.js" is the active editor. The code in "about.js" is:

```
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import {useRouter} from 'expo-router';
4
5 export default function About() {
6   const router = useRouter();
7   return (
8     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
9       <Text>About Page</Text>
10      <Button onPress={() => router.back()} title='Go Back' />
11    </View>
12  )
13 }
```

The "TERMINAL" tab at the bottom shows the following stack trace:

```
18)
    at Object.createCacheKeyWithDirectoryAsync (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@expo/image-utils/build/Cache.js:24:33)
    at Object.generateImageAsync (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@expo/image-utils/build/Image.js:151:34)
    at async Object.getFaviconFromExpoConfigAsync (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@expo/cli/build/src/export/favicon.js:64:25)
    at async FaviconMiddleware.handleRequestAsync (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@expo/cli/build/src/start/server/middleware/FaviconMiddleware.js:24:26)
    at async internalMiddleware (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@expo/cli/build/src/start/server/middleware/ExpoMiddleware.js:51:24)
```

Expo Router



Create a Page with in a directory

We use file based routing same as of Next JS.

index.tsx => Represents the Home Page

about.tsx => Represents the About Page

blog/index.tsx => Represents the Blog Page

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - OPEN EDITORS: package.json, index.js (app), about.js (app), index.js (app/blog)
 - EXPOROUTERAPP: .expo, .vscode, app (with blog folder containing index.js, about.js, and index.js)
 - node_modules, .gitignore, app.json, babel.config.js, expo-env.d.ts, metro.config.js
- package.json:** Shows the file path: app > blog > index.js. The code is as follows:

```
1 import
2 import
3 import
4
5 export const
6 return
7 <Vi
8 <
9 <
10 <
11 </V
12 )
13 }
```

Expo Router



Routing between Pages

Using a Link Component with child Button

The screenshot shows the VS Code interface with the file `index.js` open in the editor. The code defines a `Page` component that contains a `View` with centered text and two `Link` components. The first `Link` has a href of `'/about'`. The second `Link` has a href of `'/blog'` and includes a `Button` with the title `'Go to Blog Page'` as its child.

```
import { View, Text, Button } from 'react-native'
import React from 'react'
import { Link } from 'expo-router'

export default function Page() {
  return (
    <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
      <Text style={{fontSize:30}>Welcome to our app</Text>
      <Link href={'/about'}>
        <Text style={{fontSize:18}}>Go to About Page</Text>
      </Link>
      <Link href={'/blog'} asChild>
        <Button title='Go to Blog Page' />
      </Link>
    </View>
  )
}
```

Expo Router



Creating Layout file

The Layout file enables us to create the app bar for our app

<https://docs.expo.dev/router/layouts/>

```
import { View, Text } from 'react-native'
import React from 'react'
import { Stack } from 'expo-router'

export default function _layout() {
  return (
    <Stack />
  )
}
```

Expo Router



Layout file /Customising the App Bar

We can Now Customise the App Bar for each Page by expanding the Stack component and we add diff screens(pages)

<https://docs.expo.dev/router/advanced/stack/>

The screenshot shows a VS Code interface with the following details:

- Open Editors:** The left sidebar shows the project structure with files like package.json, index.js, _layout.js, about.js, and index.js under app/blog.
- File:** The main editor area displays the content of the _layout.js file. The code defines a function _layout() that returns a <Stack> component containing two <Stack.Screen> components: one for 'index' with headerTitle 'Home' and another for 'about' with headerTitle 'About'.

```
app > _layout.js > _layout > headerTitle
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Stack } from 'expo-router'
4
5 export default function _layout() {
6   return (
7     <Stack>
8       <Stack.Screen name='index' options={{headerTitle:'Home'}} />
9       <Stack.Screen name='about' options={{headerTitle:'About'}} />
10    </Stack>
11  )
12 }
```

Expo Router



Layout file /Customising the App Bar

We can Now Customise the App Bar for each Page by expanding the Stack component and we add diff screens(pages)

The screenshot shows the VS Code interface with the file `_layout.js` open in the editor. The code defines a `_layout()` function that returns a `<Stack>` component containing two `<Stack.Screen>` components. The first screen is named 'index' with a header title 'Home'. The second screen is named 'about' with a header title 'About'. The code is as follows:

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Stack } from 'expo-router'
4
5 export default function _layout() {
6   return (
7     <Stack>
8       <Stack.Screen name='index' options={{headerTitle:'Home'}} />
9       <Stack.Screen name='about' options={{headerTitle:'About'}} />
10    </Stack>
11  )
12 }
```

Expo Router



Layout file /Customising the App Bar

For a route that is a folder such as blog/index.tsx, it gives an error “No route named “blog” exists

The screenshot shows a developer tools interface with several panels:

- Code Editor:** Displays the following code in a file named _layout.tsx:

```
5 export default function _layout() {
6   return (
7     <Stack>
8       <Stack.Screen name='index' options={{headerTitle:'Home'}} />
9       <Stack.Screen name='about' options={{headerTitle:'About'}} />
10      <Stack.Screen name='blog' options={{headerTitle:'All Blog Posts'}} />
11    </Stack>
12  )
13 }
```
- Terminal:** Shows the command zsh and some log output:

```
at App
at withDevTools(App) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:1305:27)
at ErrorToastContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:174:24)
at ErrorOverlay
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at AppContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40220:36)
at main(RootComponent) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:1254:28)
```
- Browser Preview:** Shows a mobile application screen with a navigation bar. The first two items ('index' and 'about') have correct titles ('Home' and 'About'). The third item, 'blog', has an incorrect title ('All Blog Posts'). Below the screen are links 'Blog Page' and 'Go Back'.
- Bottom Status Bar:** Shows the status bar with icons for battery, signal, and other system information.

Expo Router



Layout file /Customising the App Bar

For a route that is a folder such as `blog/index.tsx`, you have to provide a name as below

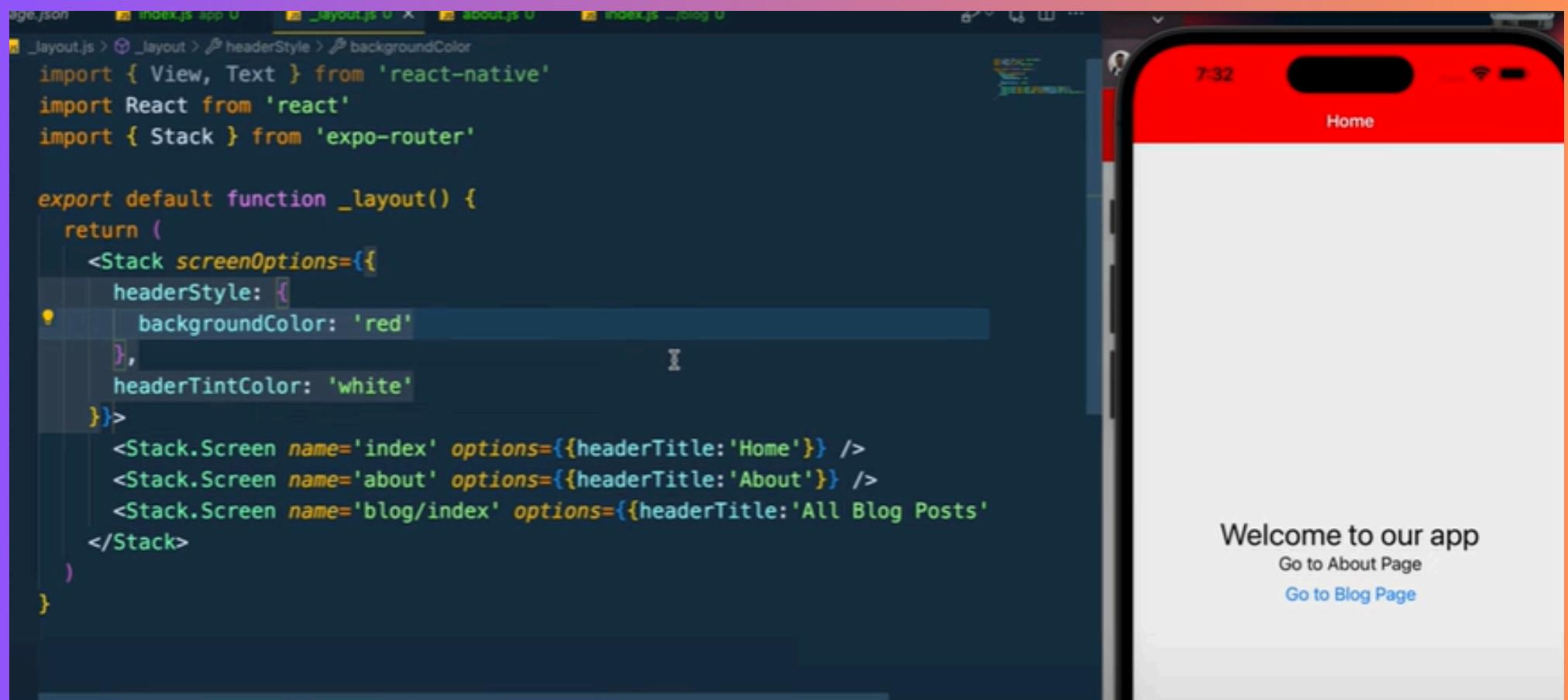
```
app > _layout.js > _layout
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Stack } from 'expo-router'
4
5 export default function _layout() {
6   return (
7     <Stack>
8       <Stack.Screen name='index' options={{headerTitle: 'Home'}} />
9       <Stack.Screen name='about' options={{headerTitle: 'About'}} />
10      <Stack.Screen name='blog/index' options={{headerTitle: 'All Blog Posts'}} />
11    </Stack>
12  )
13}
```

Expo Router



Customise the Status Bar for all Screens

We can also Customise the Look of the Status Bar Via Screen Options on the Stack Container



Expo Router



Routing between Pages Via header

We can also navigate to other Pages Via header using the `headerRight` option

The image shows a split-screen view. On the left is a code editor displaying `_layout.js` with the following code:

```
10     backgroundColor: "red",
11   },
12   headerTintColor: "white",
13 }
14 >
15 <Stack.Screen
16   name="index"
17   options={{
18     headerTitle: "Home",
19     headerRight: () => <Button onPress={() => {router.push('contact')}}
```

On the right is a mobile application running on an iPhone X simulator. The screen shows a red header bar with "Home" and "Contact" buttons. Below the header, the text "Welcome to our app" is displayed, followed by three blue links: "Go to About Page", "Go to Blog Page", and "Go to Contact Page".

Expo Router



Changing Page Presentation

We can also change how the page is presented via the Stack Options using the presentation option

```
tack.Screen
  name="index"
  options={{
    headerTitle: "Home",
    headerRight: () => <Button onPress={() => {router.push('contact')}} title='
  }>

tack.Screen name="about" options={{ headerTitle: "About" }} />
tack.Screen
  name="blog/index"
  options={{ headerTitle: "All Blog Posts" }}>
  Screen(props:
    ScreenProps<NativeStackNavigationOptions
    StackNavigationState<ParamListBase>,
    NativeStackNavigationEventMap>): null

tack.Screen name="contact" options={{ headerTitle: "About", presentation: ''
  ack>
  card
  containedModal
  containedTransparentModal
  formSheet
  fullScreenModal
  modal
  transparentModal

: OUTPUT TERMINAL PORTS DEBUG CONSOLE
.js:10862:23)
lowParserMixin.parseUpdate (/Users/pradipdebnath/Desktop/Tutorials/
5:21)
lowParserMixin.parseMaybeBinary (/Users/pradipdebnath/Desktop/Tutori
18821:23)
lowParserMixin.parseMaybeBinaryOrPrivate (/Users/pradipdebnath/Deskt...
index.js:10659:61)
lowParserMixin.parseExprOps (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/inde
64:23)
lowParserMixin.parseMaybeConditional (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser
ex.js:10641:23)
lowParserMixin.parseMaybeAssign (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/
```

Expo Router



Presenting a Page as Modal

We can also change how the page is presented via the Stack Options using the presentation option

<https://docs.expo.dev/router/advanced/modals/>

```
    stack.Screen  
    name="index"  
    options={  
      headerTitle: "Home",  
      headerRight: () => <Button onPress={() => {router.push('contact')}} title='>  
>}
```

```
stack.Screen name="about" options={{ headerTitle: "About" }} />
stack.Screen
  name="blog/index"
  options={{ headerTitle: "All Blog Posts" }}
```

```
ack.Screen name="contact" options={{ headerTitle: "About", presentation: "card" }}
```

8 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
.js:18962:23)          ↳ fullScreenModal
  lowParserMixin.parseUpdate (/Users/pradipdebnath/Desktop/Tutorials/
  S:21)                  ↳ modal
  lowParserMixin.parseMaybeUnary (/Users/pradipdebnath/Desktop/Tutori
18821:23)          ↳ transparentModal
  lowParserMixin.parseMaybeUnaryOrPrivate (/Users/pradipdebnath/Deskt
index.js:18659:61)
  lowParserMixin.parseExprOps (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index.js:184:23)
  lowParserMixin.parseMaybeConditional (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/ex.js:18641:23)
  lowParserMixin.parseMaybeAssignment (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/
```

Expo Router



Dynamic Routes

We create dynamic Route such as blog detail page
we create the same way in next js => [id].tsx

<https://docs.expo.dev/router/create-pages/#dynamic-routes>

The screenshot shows a code editor with a dark theme. The file being edited is named 'index.js' and is located in a folder named 'blog'. The code implements a dynamic route for blog posts:

```
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import {useRouter} from 'expo-router';
4
5 export default function Page() {
6   const router = useRouter();
7   return (
8     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
9       <Text style={{fontSize:18}}>Blog Page</Text>
10      <Button onPress={() => router.push('/blog/1')} title='Go To Blog 1' />
11      <Button onPress={() => router.push('/blog/2')} title='Go To Blog 2' />
12      <Button onPress={() => router.push('/blog/3')} title='Go To Blog 3' />
13      <Button onPress={() => router.back()} title='Go Back' />
14    </View>
15  )
16}
```

Expo Router



Dynamic Routes

We create dynamic Route such as blog detail page
we create the same way in next js => [id].tsx

To get the id on Detail Page Use “useLocalSearchParams”

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure:
 - OPEN EDITORS: package.json, index.js app, _layout.js app, [id].js app/blog (selected), contact.js app, about.js app, index.js app/blog.
 - EXPORROUTERAPP: .expo, .vscode, app, blog: [id].js, index.js, _layout.js, about.js, contact.js, index.js.
 - node_modules, .gitignore, app.json, babel.config.js, expo-env.d.ts, metro.config.js, package-lock.json, package.json.
- CODE EDITOR**: The [id].js file is open, showing the following code:

```
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import {useLocalSearchParams} from 'expo-router';
4
5 export default function Page() {
6   const { id } = useLocalSearchParams();
7
8   return (
9     <View style={{flex:1,justifyContent:'center',alignItems:'center'}>
10      <Text style={{fontSize:18}>Blog Post Details {id}</Text>
11      <Button onPress={() => router.back()} title='Go Back' />
12    </View>
13  )
14 }
```
- STATUS BAR**: Shows the current file path: app > blog > [id].js > Page

Expo Router



Customising the Header for the Detailed Page

We use Stack Component to Customise the Header for the Detailed Page

Expo Router



Passing Multiple Params

We can also pass multiple parama as shown below

```
blog > index.js > Page
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import {useRouter} from 'expo-router';
4
5 export default function Page() {
6   const router = useRouter();
7   return (
8     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
9       <Text style={{fontSize:18}}>Blog Page</Text>
10      <Button onPress={() => router.push('/blog/1')} title='Go To Blog 1' />
11      <Button onPress={() => router.push('/blog/2')} title='Go To Blog 2' />
12      <Button onPress={() => router.push('/blog/3?author=john')} title='Go T
13        <Button onPress={() => router.back()} title='Go Back' />
14      </View>
15    )
16  }
17 }
```

Expo Router



Passing Multiple Params

We can also pass multiple params as shown below

```
og > [id].js > Page
import { View, Text, Button } from "react-native";
import React from "react";
import { useLocalSearchParams, router, Stack } from "expo-router";

export default function Page() {
  const { id, author } = useLocalSearchParams();

  return (
    <>
      <Stack.Screen options={{ headerTitle: `Article ${id}` }} />

      <View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
        <Text style={{ fontSize: 18 }}>Blog Post Details {id}</Text>
        <Text style={{ fontSize: 18 }}>Written by {author}</Text>
        <Button onPress={() => router.back()} title="Go Back" />
      </View>
    </>
  );
}
```

Expo Router



Using Link to pass Multiple Params

We can also pass multiple parama as shown below

<https://docs.expo.dev/router/navigating-pages/#linking-to-dynamic-routes>

```
blog > index.js > Page > params
3 import {Link, useRouter} from 'expo-router';
4
5 export default function Page() {
6   const router = useRouter();
7   return (
8     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
9       <Text style={{fontSize:18}}>Blog Page</Text>
10      <Button onPress={() => router.push('/blog/1')} title='Go To Blog 1' />
11      <Button onPress={() => router.push('/blog/2')} title='Go To Blog 2' />
12      <Button onPress={() => router.push('/blog/3?author=john')} title='Go T
13      <Link href={{
14        pathname: 'blog/4',
15        params: { author: 'Jenny' }
16      }>
17        <Text>Go to blog 4</Text>
18      </Link>
19      <Button onPress={() => router.back()} title='Go Back' />
20    </View>
21  )
22 }
```

Expo Router



Tab Navigator and Groups

Start by Creating a route group (tabs)
and Inside Create some Pages



Expo Router



Linking to a page in a group

A screenshot of a code editor showing a file named `index.js` with the following content:

```
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import { Link } from 'expo-router'

4
5 export default function Page() {
6   return (
7     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
8       <Text style={{fontSize:30}}>Welcome to our app</Text>
9       <Link href={'/about'}>
10         <Text style={{fontSize:18}}>Go to About Page</Text>
11       </Link>
12       <Link href={'/blog'} asChild>
13         <Button title='Go to Blog Page' />
14       </Link>
15       <Link href={'/blog'} asChild>
16         <Button title='Go to Contact Page' />
17       </Link>
18       <Link href={'/(tabs)/feed'} asChild>
19         <Button title='Go to Tabs' />
20       </Link>
21     </View>
22   )
23 }
```

The code demonstrates linking to pages within a group. The `/about` link is a standard `Link`. The links to `/blog` and `/(tabs)/feed` are wrapped in `asChild` components, which is a requirement for linking to pages within a group.

Expo Router



Create Layout to bring the Tabs

<https://docs.expo.dev/router/advanced/stack/>

```
o > (tabs) > _layout.js > _layout
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Tabs } from 'expo-router'

4

5 export default function _layout() {
6   return (
7     <Tabs />
8   )
9 }
```

Expo Router



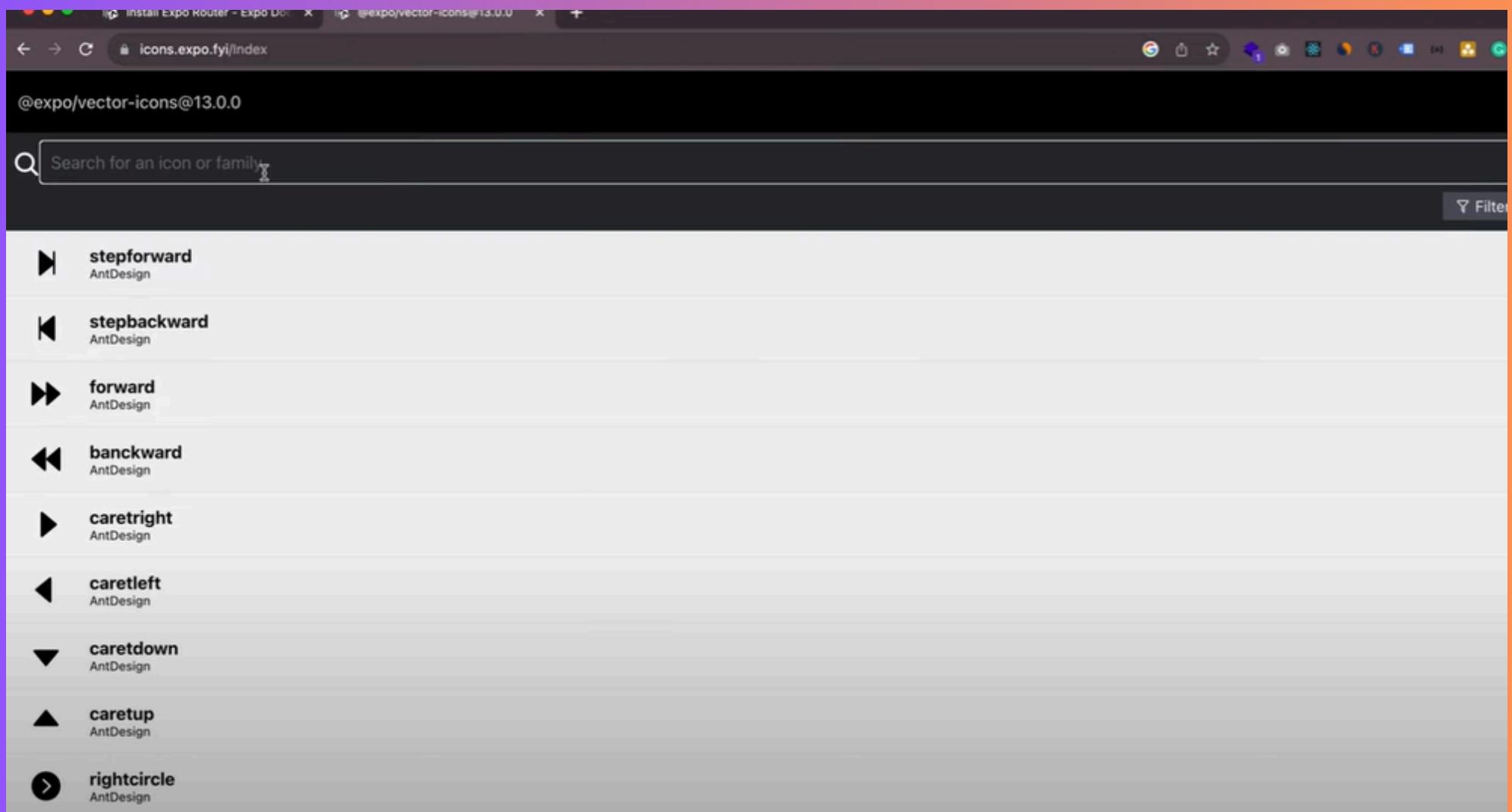
Customise the Tabs as By expanding the Tabs

```
app > (tabs) > _layout.js > _layout > tabBarIcon
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Tabs } from 'expo-router'
4 import { Feather } from '@expo/vector-icons';
5
6 export default function _layout() {
7   return (
8     <Tabs>
9       <Tabs.Screen name='feed' options={{
10         tabBarIcon: () => (
11           <Feather name="list" size={24} color="black" />
12         )
13       } />
14     </Tabs>
15   )
16 }
```

Expo Router



For Icons Use expo vector icons



Expo Router



For more Customisations of the Tabs

```
pp > (tabs) > _layout.js app U > _layout > headerTitle
7   return (
8     <Tabs>
9       <Tabs.Screen name='feed' options={{
10         tabBarIcon: ({color}) => (
11           <Feather name="list" size={24} color={color} />
12         ),
13         tabBarLabel: 'Feed',
14         headerTitle: 'Feed',
15       } />
16       <Tabs.Screen name='profile' options={{
17         tabBarIcon: ({color}) => (
18           <AntDesign name="user" size={24} color={color} />
19         ),
20         tabBarLabel: 'Profile',
21         headerTitle: 'Profile'
22       } />
23     </Tabs>
24   )
25 }
```

Expo Router



To remove 2 Bars on the Page, we add the tabs in the layout

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows files like index.js app, _layout.js app/(tabs), [id].js app, contact.js app, about.js app, index.js app/blog, feed.js app/(tabs), and profile.js app/(tabs).
- _layout.js Content:**

```
13     });
14   >
15   <Stack.Screen
16     name="index"
17     options={{
18       headerTitle: "Home",
19       headerRight: () => <Button onPress={() => {router.push('contact')}}>
20     }}
21   />
22   <Stack.Screen name="about" options={{ headerTitle: "About" }} />
23   <Stack.Screen
24     name="blog/index"
25     options={{ headerTitle: "All Blog Posts" }}
26   />
27   <Stack.Screen name="contact" options={{ headerTitle: "Contact", presen
28   <Stack.Screen name="(tabs)" options={{ headerShown: false }} />
29   </Stack>
30 };
31 }
```
- TERMINAL:** Shows command-line output related to expo-router setup.
- Bottom Status Bar:** Shows the time as 8:51 and the device as iPhone 13 Pro Max (iOS 16.4).

Expo Router



Creating Nested pages within the Tab Navigator

The screenshot shows a VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - index.js app
 - _layout.js app/(tabs)
 - _layout.js app
 - [id].js app/blog
 - contact.js app
 - about.js app
 - index.js app/blog
 - index.js app/(tabs)/feed
 - new.js app/(tabs)/feed
- OPEN EDITORS:** Shows the file _layout.js being edited. The code is as follows:

```
return (
  <Tabs>
    <Tabs.Screen name='feed/index' options={{
      tabBarIcon: ({color}) => (
        <Feather name="list" size={24} color={color} />
      ),
      tabBarLabel: 'Feed',
      headerTitle: 'Feed',
    }/>
    <Tabs.Screen name='profile' options={{
      tabBarIcon: ({color}) => (
        <AntDesign name="user" size={24} color={color} />
      ),
      tabBarLabel: 'Profile',
      headerTitle: 'Profile'
    }/>
  </Tabs>
)
```

Expo Router



To remove Unwanted Tab, Create another layout with feed folder

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - OPEN EDITORS: 1 unsaved
 - app > (tabs) > feed > _layout.js
 - EXPOROUTERAPP > .expo
 - EXPOROUTERAPP > vscode
 - EXPOROUTERAPP > app
 - (tabs)
 - feed
 - _layout.js
 - index.js
 - new.js
 - _layout.js
 - profile.js
 - blog
 - _layout.js
 - about.js
 - contact.js
 - index.js
 - node_modules
- EDITOR:** The current file is _layout.js under the feed directory. The code is as follows:

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Stack } from 'expo-router'
4
5 export default function _layout() {
6   return (
7     <Stack />
8   )
9 }
```

Expo Router



Use the folder name as the name in the layout

The screenshot shows a VS Code interface with the following details:

- EXPLORER:** Shows the project structure under "OPEN EDITORS".
 - app > (tabs) > _layout.js
 - index.js app
 - _layout.js app/(tabs)
 - _layout.js app/(tabs)/feed
 - _layout.js app
 - [id].js app/blog
 - contact.js app
 - about.js app
 - index.js app/blog
 - index.js app/(tabs)/feed
- EDITOR:** The file `_layout.js` is open, showing the following code:

```
return (
  <Tabs>
    <Tabs.Screen name='feed' options={{
      tabBarIcon: ({color}) =>
        <Feather name="list" size={24} color={color} />
      ),
      tabBarLabel: 'Feed',
      headerTitle: 'Feed',
    }} />
    <Tabs.Screen name='profile' options={{
      tabBarIcon: ({color}) =>
        <AntDesign name="user" size={24} color={color} />
      ),
      tabBarLabel: 'Profile',
      headerTitle: 'Profile'
    }} />
  )
}
```
- PROBLEMS:** Shows several errors related to asset resolution, such as "Unable to resolve asset ... from 'icon' in your app.json or app.config.json".

Expo Router



Add the header Right Button on the Feed Screen

The screenshot shows a code editor with a dark theme. The file being edited is `Layout.js`, specifically the `headerRight` prop of a `<Tabs>` component. The code defines two tabs: 'feed' and 'profile'. The 'feed' tab has a header right button that, when pressed, pushes a new route 'feed/new'.

```
7  return (
8    <Tabs>
9      <Tabs.Screen name='feed' options={{
10        tabBarIcon: ({color}) => (
11          <Feather name="list" size={24} color={color} />
12        ),
13        tabBarLabel: 'Feed',
14        headerTitle: 'Feed',
15        headerRight: () => <Button onPress={() => router.push('feed/new')} title='push(href: Href): void' />
16      ) />
17      <Tabs.Screen name='profile' options={{
18        tabBarIcon: ({color}) => (
19          <AntDesign name="user" size={24} color={color} />
20        ),
21        tabBarLabel: 'Profile',
22        headerTitle: 'Profile'
23      ) />
24    </Tabs>
25  )
26}
```

The code editor's status bar at the bottom shows the following tabs: PROBLEMS, OUTPUT, TERMINAL, PORTS, DEBUG CONSOLE. The TERMINAL tab is active, displaying the command `zsh`. The bottom of the terminal window shows several error messages related to asset resolution and bundle completion.

```
73:27)
  at ErrorToastContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:13:8474:24)
  at ErrorOverlay
  at RCTView
  at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
  at RCTView
  at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
  at AppContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:48220:36)
  at main(RootComponent) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:12:8954:28)
Unable to resolve asset "./assets/images/icon.png" from "icon" in your app.json or app.config.js
ios Bundling complete 52ms
```

Expo Router



Remove the unnecessary Header bar

The screenshot shows a development environment with a code editor and a mobile app preview. On the left, a code editor displays a file named `_layout.js` with the following content:

```
app > (tabs) > feed > _layout.js > headerShown
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import { Stack } from 'expo-router'
4
5 export default function _layout() {
6   return (
7     <Stack screenOptions={{headerShown:false}} />
8   )
9 }
```

The line `<Stack screenOptions={{headerShown:false}} />` is highlighted. Below the code editor is a terminal window showing logs and errors. The terminal output includes:

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
73:27)
    at ErrorToastContainer (http://192.168.0.104:8881/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:13
0474:24)
    at ErrorOverlay
    at RCTView
    at View (http://192.168.0.104:8881/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
    at RCTView
    at View (http://192.168.0.104:8881/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
    at AppContainer (http://192.168.0.104:8881/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40220:36)
    at main(RootComponent) (http://192.168.0.104:8881/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:12
8954:28)
    Unable to resolve asset "./assets/images/icon.png" from "icon" in your app.json or app.config.js
    info Bundling complete! Done
```

To the right of the terminal is a mobile app preview for iOS. The app has a white background and a black header bar. The header bar contains the text "Feed" and "Add Post". Below the header bar, the text "Feed Page" is visible. The status bar at the top of the phone screen shows the time as 8:57.

Expo Router



Adding Button on the New Page

```
> (tabs) > feed > newjs > Page
1 import { View, Text, Button } from 'react-native'
2 import React from 'react'
3 import { router } from 'expo-router'

4

5 export default function Page() {
6   return (
7     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
8       <Text style={{fontSize:18}}>Add New Post Page</Text>
9       <Button onPress={() => router.back()} title='Go Back' />
10    </View>
11  )
12}
```

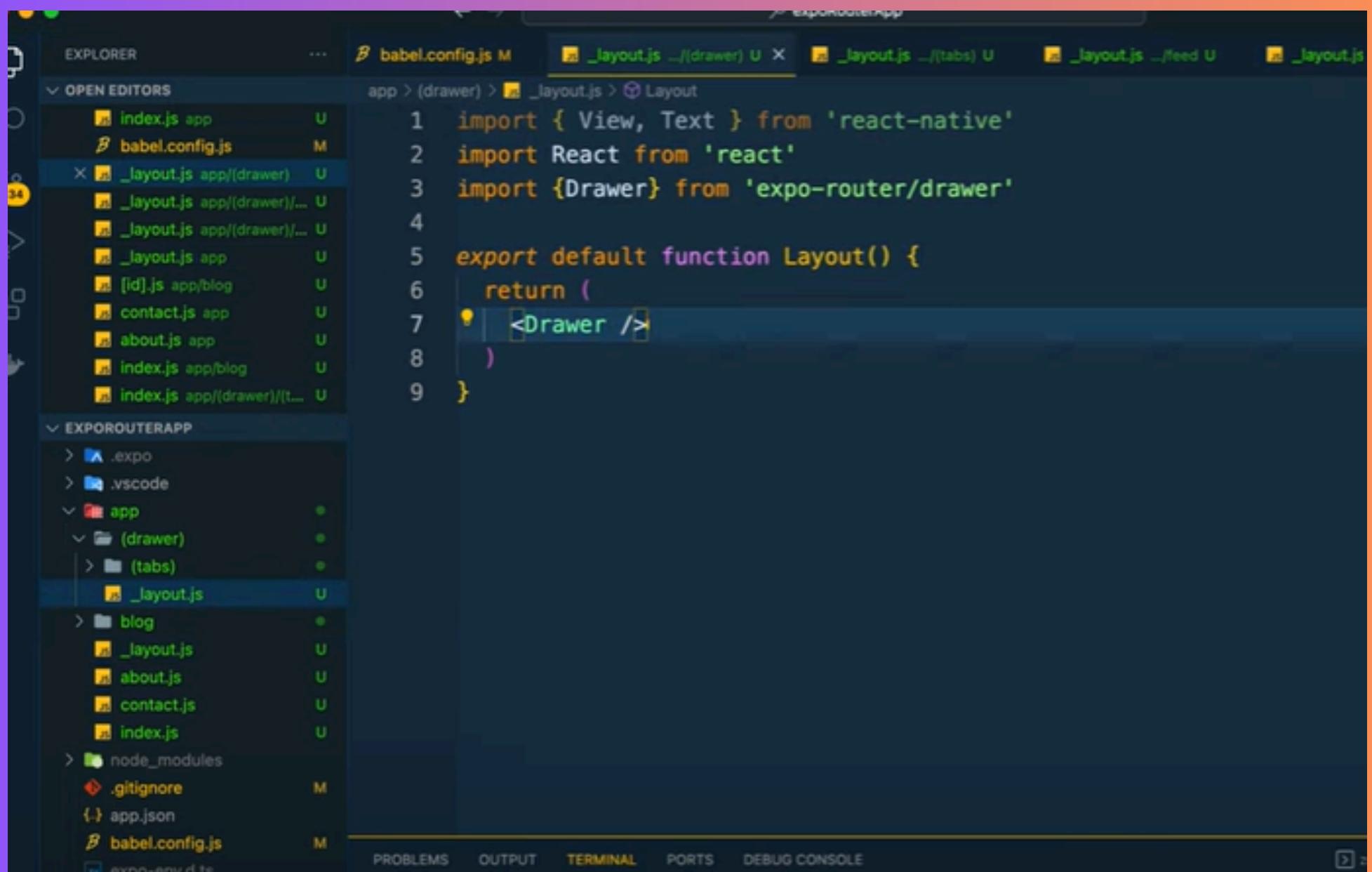
```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
at ErrorToastContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:13:4:24)
at ErrorOverlay
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at AppContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40228:36)
at main(ReactComponent) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:12:4:28)
```

Expo Router



Drawer Navigation

<https://docs.expo.dev/router/advanced/drawer/>



A screenshot of the VS Code interface showing a file named `_layout.js` open in the editor. The code uses the `expo-router/drawer` package to create a drawer navigation layout. The code is as follows:

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import {Drawer} from 'expo-router/drawer'
4
5 export default function Layout() {
6   return (
7     <Drawer />
8   )
9 }
```

The Explorer sidebar on the left shows the project structure, including files like `index.js`, `babel.config.js`, and various `_layout.js` files for different routes. The `(drawer)` folder under `app` is selected.

Expo Router



Drawer Navigation

Creating Custom Drawer Component to to be
Customise the Drawer content

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure for "expoRouterApp". It includes files like `index.js`, `babel.config.js`, and `_layout.js` under the `app` directory, which is further divided into `(drawer)` and `(tabs)`.
- Code Editor:** The current file is `_layout.js` located in the `(drawer)` folder. The code defines a `CustomDrawerContent` component using `DrawerContentScrollView` from `@react-navigation/drawer`. It also exports a `Layout` function that returns a `Drawer` component with `drawerContent` set to `CustomDrawerContent`.
- Terminal:** At the bottom, the terminal shows logs from the Expo DevTools, including entries for `withDevTools`, `ErrorToastContainer`, `ErrorOverlay`, and `AppContainer`.

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3 import {Drawer} from 'expo-router/drawer'
4 import { DrawerContentScrollView } from '@react-navigation/drawer'

6 const CustomDrawerContent = (props) => {
7   <DrawerContentScrollView {...props}>
8     |
9   </DrawerContentScrollView>
10 }

12 export default function Layout() {
13   return (
14     <Drawer drawerContent={(props) => <CustomDrawerContent {...props} />} />
15   )
16 }
```

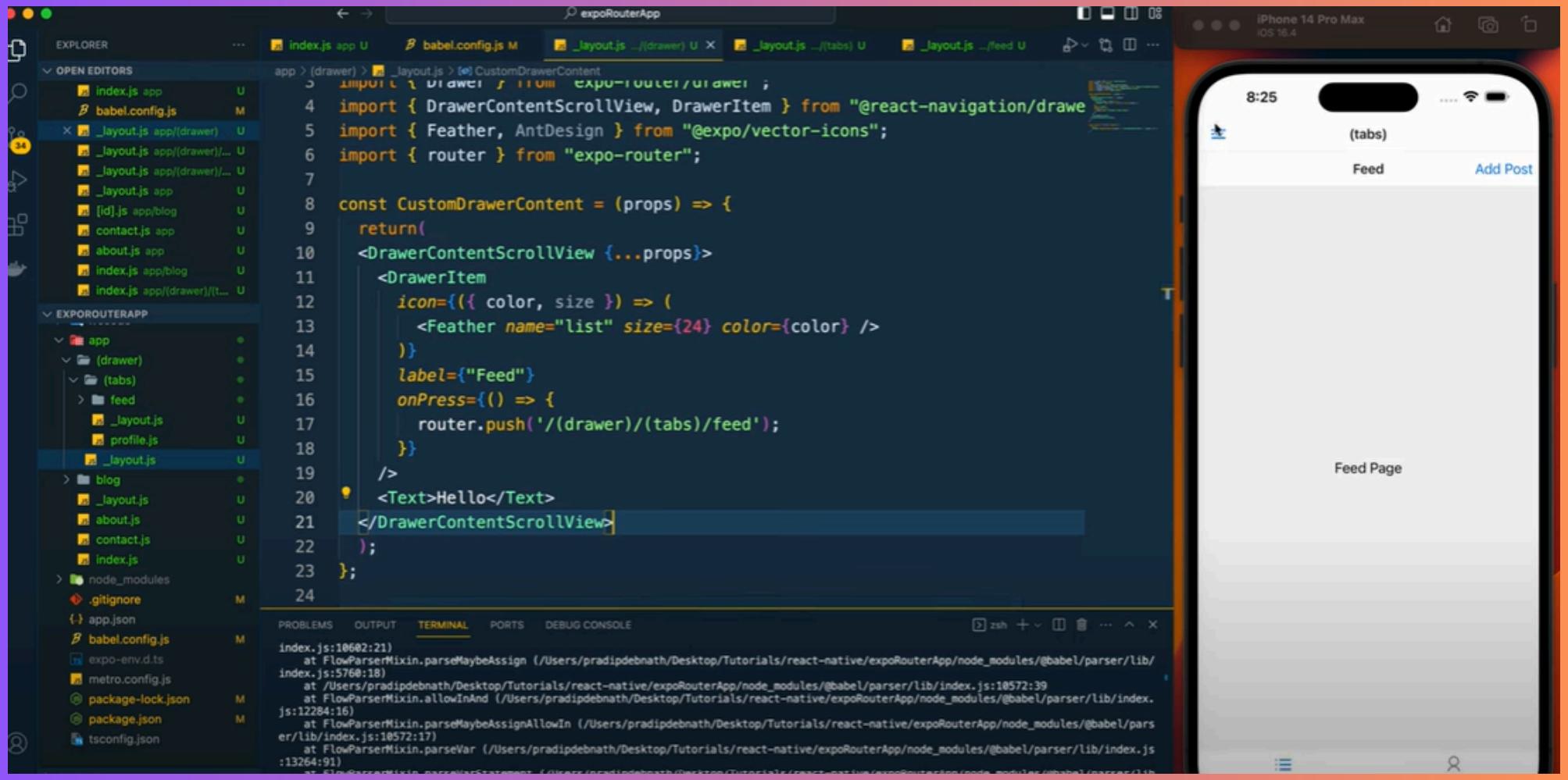
```
at withDevTools(App) (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:1305
73:27)
at ErrorToastContainer (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:13
0474:24)
at ErrorOverlay
at RCTView
at View (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at RCTView
at View (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at AppContainer (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40220:36)
at main(bootComponent) (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:12
```

Expo Router



Drawer Navigation

We can add Tabs Links to the Custom Drawer



The screenshot shows a development environment with a code editor and a mobile application preview.

Code Editor (VS Code):

```
index.js app U babel.config.js M _layout.js app/(drawer) U _layout.js .../(drawer) U _layout.js .../(tabs) U _layout.js .../feed U
1 import { DrawerContentScrollView, DrawerItem } from "@react-navigation/drawer";
2 import { Feather, AntDesign } from "@expo/vector-icons";
3 import { router } from "expo-router";
4
5 const CustomDrawerContent = (props) => {
6   return(
7     <DrawerContentScrollView {...props}>
8       <DrawerItem
9         icon={({ color, size }) => (
10           <Feather name="list" size={24} color={color} />
11         )}
12         label="Feed"
13         onPress={() => {
14           router.push('/(drawer)/(tabs)/feed');
15         }}
16       >
17         <Text>Hello</Text>
18       </DrawerContentScrollView>
19     );
20   };
21
22
23
24
```

Mobile Application Preview (iPhone 14 Pro Max):

The mobile application displays a drawer navigation interface. The top bar shows the time as 8:25 and the device as iPhone 14 Pro Max (iOS 16.4). The drawer menu has a tab labeled '(tabs)' which is currently selected. Below it are links for 'Feed' and 'Add Post'. The main content area is labeled 'Feed Page'.

Expo Router



Drawer Navigation

We can add other pages that not on the tabs

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure under `EXPORROUTERAPP`. It includes files like `index.js`, `babel.config.js`, `app` (with `(drawer)` and `(tabs)` subfolders containing `_layout.js`, `favourites.js`, `settings.js`, `about.js`, `contact.js`, and `index.js`), `blog` (with `_layout.js`, `about.js`, `contact.js`, and `index.js`), `node_modules`, `.gitignore`, `app.json`, `babel.config.js`, `expo-env.d.ts`, `metro.config.js`, `package-lock.json`, `package.json`, and `tsconfig.json`.
- OPEN EDITORS**: Shows the file `_layout.js` open in the editor. The code is as follows:

```
32     })
33     label="Favourites"
34     onPress={() => {
35       router.push('/favourites');
36     }
37   }
38   <DrawerItem
39     icon={({ color, size }) =>
40       <Feather name="list" size={24} color={color} />
41     }
42     label="Settings"
43     onPress={() => {
44       router.push('/settings');
45     }
46   }
47   </DrawerContentScrollView>
48 );
49 };
50
51 export default function Layout() {
52   return (

```

- PROBLEMS**: Shows several errors related to FlowParserMixin and Babel parser issues.
- OUTPUT**: Shows the command `zsh` and other terminal output.
- TERMINAL**: Shows the command `zsh` and other terminal output.
- PORTS**: Shows port information.
- DEBUG CONSOLE**: Shows the command `zsh` and other terminal output.

Expo Router



Drawer Navigation

Styling the Icon

The screenshot shows a code editor and a running mobile application. The code editor displays a portion of `_Layout.js` with the following content:

```
app > (drawer) > _Layout.js > CustomDrawerContent > fontSizes
29   <DrawerItem
30     icon={({ color, size }) =>
31       <MaterialIcons name="favorite-outline" size={size} color={color} />
32     }
33     label="Favourites"
34     onPress={() => {
35       router.push('/favourites');
36     }}
37   />
38   <DrawerItem
39     icon={({ color, size }) =>
40       <Ionicons name="settings-outline" size={size} color={color} />
41     }
42     label="Settings"
43     labelStyle={{marginLeft: -20, fontSize:18}}
44     onPress={() => {
45       router.push('/settings');
46     }}
47   />
48   </DrawerContentScrollView>
49 );
```

The code editor's bottom bar shows tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, and DEBUG CONSOLE. The TERMINAL tab is active, displaying the command `zsh`. The DEBUG CONSOLE tab shows a stack trace:

```
at withDevTools(App) (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:1305:73:27)
at ErrorToastContainer (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:131:8474:24)
at ErrorOverlay
at RCTView
at View (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at RCTView
at View (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40380:43)
at AppContainer (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&hot=false&lazy=true:40220:36)
```

To the right of the code editor is a mobile application window titled "iOS 16.4". The screen shows a drawer navigation menu with four items: "Feed", "Profile", "Favourites", and "Settings". The "Favourites" item is currently selected.

Expo Router



Drawer Navigation

To add active Drawer item , we use Pathname

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure for "expoRouterApp". The "app" folder contains a "drawer" folder which includes "(tabs)", "_layout.js", "favourites.js", and "settings.js". Other files like "index.js", "babel.config.js", and "node_modules" are also listed.
- CODE EDITOR:** The file "CustomDrawerContent.js" is open. It imports View, Text, StyleSheet from react-native, React, useEffect from react, and Drawer from expo-router/drawer. It also imports DrawerContentScrollView, DrawerItem from @react-navigation/drawer, and Feather, AntDesign, MaterialIcons, Ionicons from @expo/vector-icons. It uses the usePathname hook from expo-router. The code defines a CustomDrawerContent component that logs the pathname to the console when the effect runs.
- TERMINAL:** At the bottom, the terminal shows the output of the application, including logs for ErrorOverlay, RCTView, View, AppContainer, and mainRootComponent, along with profile and feed logs.

```
1 import { View, Text, StyleSheet } from "react-native";
2 import React, { useEffect } from "react";
3 import { Drawer } from "expo-router/drawer";
4 import { DrawerContentScrollView, DrawerItem } from "@react-navigation/drawer";
5 import { Feather, AntDesign, MaterialIcons, Ionicons } from "@expo/vector-icons";
6 import { router, usePathname } from "expo-router";
7
8 const CustomDrawerContent = (props) => {
9   const pathname = usePathname();
10
11   useEffect(() => {
12     console.log(pathname);
13   }, [pathname]);
14
15   return(
16     <DrawerContentScrollView {...props}>
17       <DrawerItem
18         icon={({ color, size }) => (
19           <Feather name="list" size={size} color={color} />
20         )}
21         label="Feed"
22         labelStyle={styles.navItemLabel}
23       </DrawerItem>
24     </DrawerContentScrollView>
25   );
26 }
27
28 export default CustomDrawerContent;
```

Expo Router



Drawer Navigation

Using the pathname to show the Active Item

```
> (drawer) > _layout.js > (e) CustomDrawerContent > <View style={...}>
21   <Text>Custom Drawer</Text>
22   <DrawerItem
23     icon={({ color, size }) => (
24       <Feather
25         name="list"
26         size={size}
27         color={pathname === "/feed" ? "#fff" : "#000"}
28       />
29     )}
30     label="Feed"
31     labelStyle={[
32       styles.navItemLabel,
33       { color: pathname === "/feed" ? "#fff" : "#000" },
34     ]}
35     style={{backgroundColor: pathname === "/feed" ? "#333" : "#fff"}}
36     onPress={() => {
37       router.push("/:drawer)/(tabs)/feed");
38     }}
39   />
40   <DrawerItem
41     icon={({ color, size }) => (
42       <AntDesign name="user" size={size} color={color} />
```

Expo Router



Drawer Navigation

Add Other Content in the Drawer

The screenshot shows a development environment with a code editor and a simulator.

Code Editor:

```
app > (drawer) > _layout.js > [e] CustomDrawerContent
20  return (
21    <DrawerContentScrollView {...props}>
22      <View style={styles.userInfoWrapper}>
23        <Image
24          source={{ uri: "https://randomuser.me/api/portraits/women/26.jpg" }
25          width={80}
26          height={80}
27          style={styles.userImg}
28        />
29        <View style={styles.userDetailsWrapper}>
30          <Text style={styles.userName}>John Doe</Text>
31          <Text>john@email.com</Text>
32        </View>
33      </View>
34      <DrawerItem
35        icon={({ color, size }) => (
36          <Feather
37            name="list"
38            size={size}
39            color={pathname === "/feed" ? "#fff" : "#000"}
40          />
41        )}
42      </DrawerItem>
43    </DrawerContentScrollView>
44  </View>
45</CustomDrawerContent>
```

Terminal:

```
LOG /settings
LOG /settings
LOG /feed
LOG /profile
LOG /profile
LOG /favourites
```

Simulator:

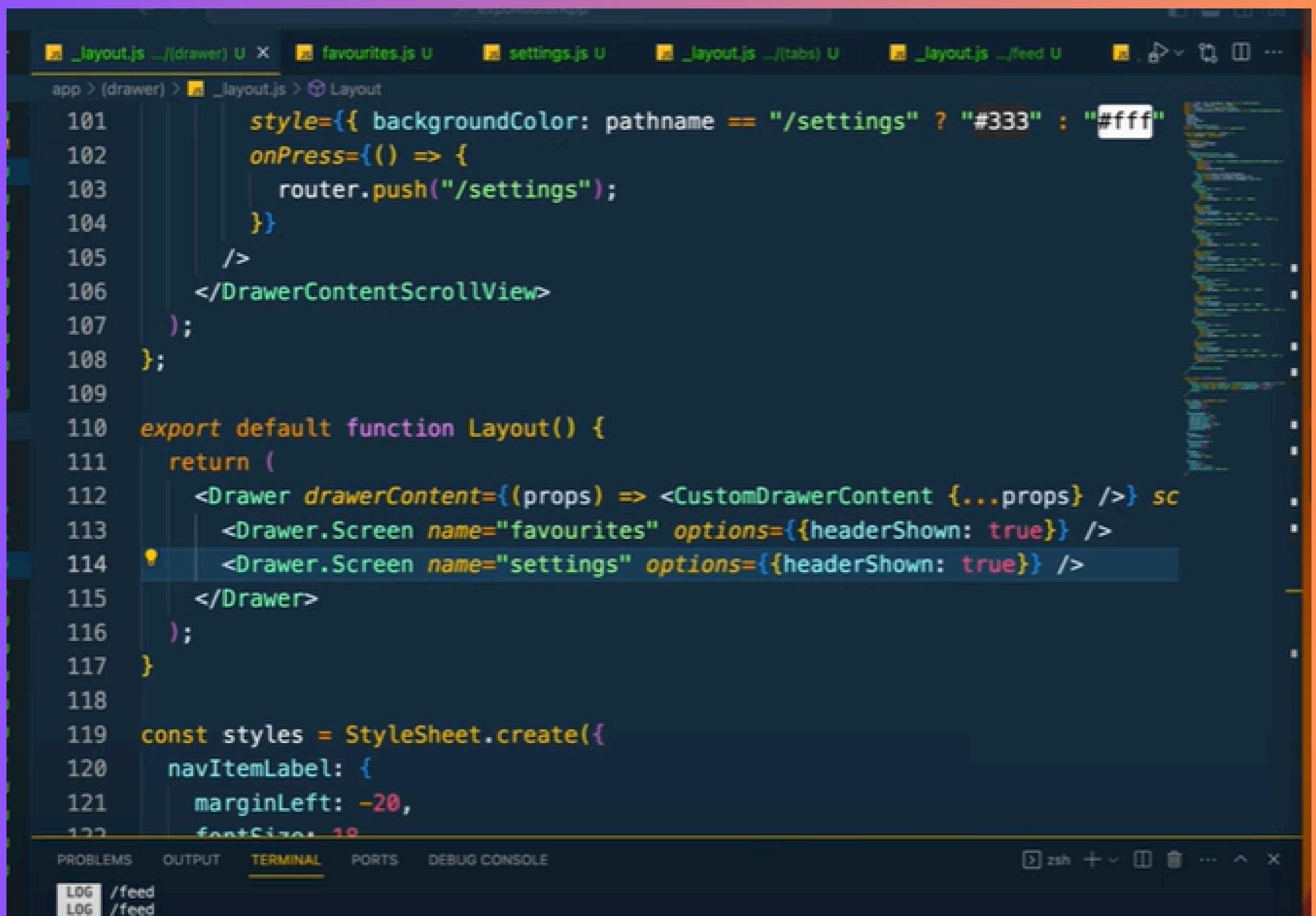
An iPhone X simulator running iOS 16.4 displays a drawer navigation interface. The top bar shows the time as 8:49. The user profile section displays a placeholder profile picture of a woman, the name John Doe, and the email john@email.com. The drawer menu contains four items: Feed (selected), Profile, Favourites, and Settings.

Expo Router



Drawer Navigation

Handling the Drawer Header, for the links that are not in the tabs



The screenshot shows a code editor in VS Code with the file `_Layout.js` open. The code defines a `Layout` component that returns a `Drawer` component with `drawerContent` set to a function that returns `CustomDrawerContent`. Inside `CustomDrawerContent`, there are two `Drawer.Screen` components: one for "favourites" and one for "settings", both with `headerShown: true`.

```
101     style={{ backgroundColor: pathname == "/settings" ? "#333" : "#fff" }}  
102     onPress={() => {  
103       router.push("/settings");  
104     }}  
105   />  
106   <DrawerContentScrollView>  
107 );  
108 };  
109  
110 export default function Layout() {  
111   return (  
112     <Drawer drawerContent={(props) => <CustomDrawerContent {...props} />} sc  
113       <Drawer.Screen name="favourites" options={{headerShown: true}} />  
114       <Drawer.Screen name="settings" options={{headerShown: true}} />  
115     </Drawer>  
116   );  
117 }  
118  
119 const styles = StyleSheet.create({  
120   navItemLabel: {  
121     marginLeft: -20,  
122     fontSize: 10  
123   }  
124 });
```

Below the code editor, the VS Code interface shows the following tabs: PROBLEMS, OUTPUT, TERMINAL (which is selected), PORTS, and DEBUG CONSOLE. The bottom left shows a log entry: LOG /feed.

Expo Router



Drawer Navigation

Handling the Drawer Header, for the links that are in the tabs by using DrawerToggleButton button

The screenshot shows a development environment with a code editor and a mobile application preview.

Code Editor (VS Code):

```
app > (drawer) > (tabs) > _layout.js > _layout > headerLeft
  3 import { Iaods, router } from 'expo-router';
  4 import { Feather, AntDesign } from '@expo/vector-icons';
  5 import { DrawerToggleButton } from '@react-navigation/drawer';
  6
  7 export default function _layout() {
  8   return (
  9     <Tabs screenOptions={{headerLeft: () => <DrawerToggleButton />}}>
10       <Tabs.Screen name='feed' options={{
11         tabBarIcon: ({color}) => (
12           <Feather name="list" size={24} color={color} />
13         ),
14         tabBarLabel: 'Feed',
15         headerTitle: 'Feed',
16         headerRight: () => <Button onPress={() => router.push('feed/new')} title='New' />
17       } />
18       <Tabs.Screen name='profile' options={{
19         tabBarIcon: ({color}) => (
20           <AntDesign name="user" size={24} color={color} />
21         ),
22         tabBarLabel: 'Profile',
23         headerTitle: 'Profile'
24       } />

```

Mobile Application Preview:

An iPhone 14 Pro Max simulator is running the app. The screen shows a navigation bar with a "Profile" tab selected. Below the navigation bar, there is a drawer toggle button. The main content area displays a list of items under the "Feed" tab, with a "New" button at the bottom right.

Expo Router



Drawer Navigation

Changing the Tint Color of the button

```
app > (drawer) > (tabs) > _Layout.js > _Layout > headerLeft
  3   import { Tabs, router } from 'expo-router';
  4   import { Feather, AntDesign } from '@expo/vector-icons';
  5   import { DrawerToggleButton } from '@react-navigation/drawer';
  6
  7   export default function _layout() {
  8     return (
  9       <Tabs screenOptions={{headerLeft: () => <DrawerToggleButton tintColor='#000' />}}
 10         <Tabs.Screen name='feed' options={{
 11           tabBarIcon: ({color}) => (
 12             <Feather name="list" size={24} color={color} />
 13           ),
 14           tabBarLabel: 'Feed',
 15           headerTitle: 'Feed',
 16           headerRight: () => <Button onPress={() => router.push('feed/new')} title='>' />
 17         ) />
 18         <Tabs.Screen name='profile' options={{
 19           tabBarIcon: ({color}) => (
 20             <AntDesign name="user" size={24} color={color} />
 21           ),
 22           tabBarLabel: 'Profile',
 23           headerTitle: 'Profile'
 24         ) />
```

Expo Router



Lottie Animations

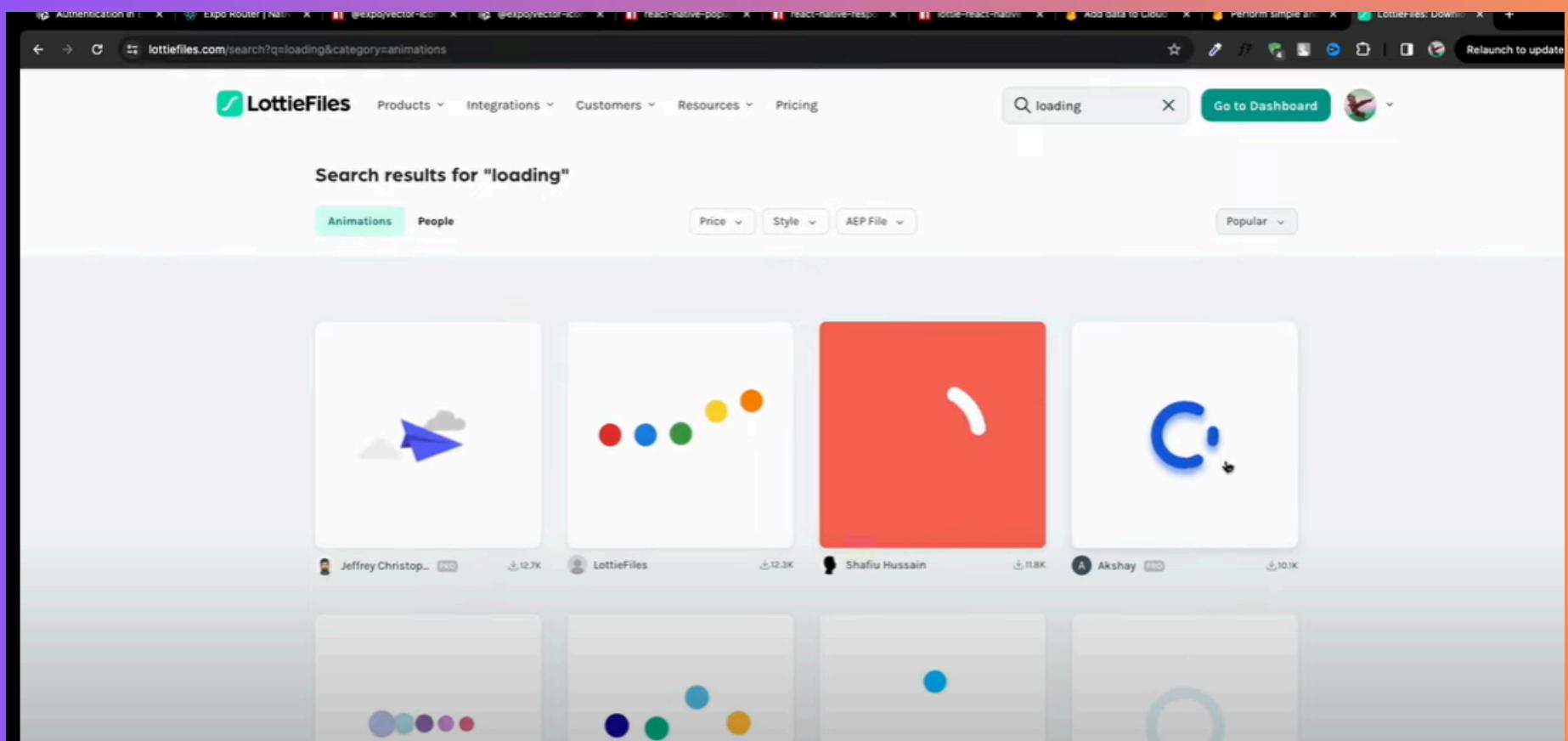
First Install the Package

```
npm i lottie-react-native
```

For the animations Visit the Website

<https://lottiefiles.com/>

Start by Searching for the animation eg loading



Expo Router



Lottie Animations

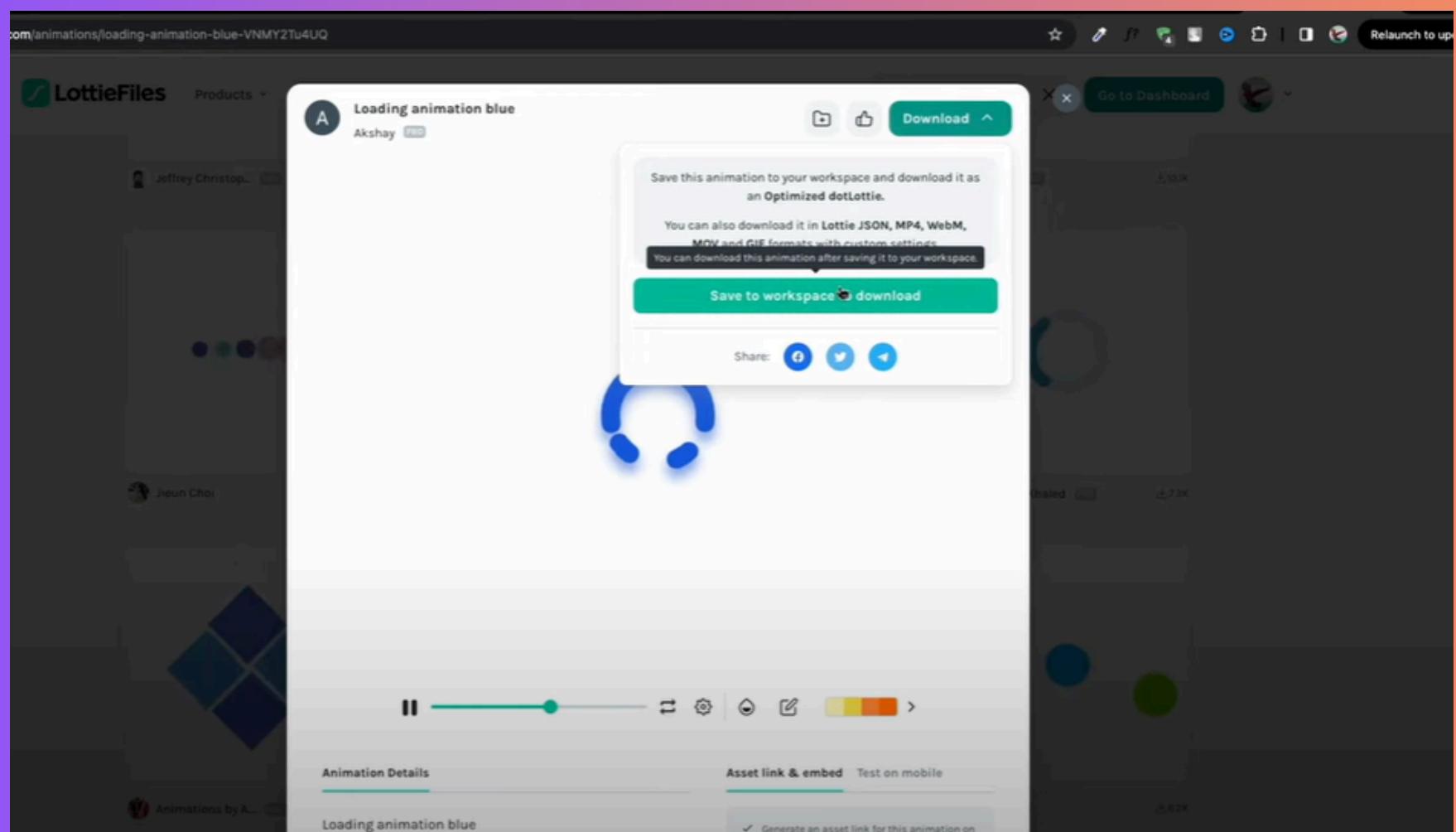
First Install the Package

```
npm i lottie-react-native
```

For the animations Visit the Website

<https://lottiefiles.com/>

Click on the animation and save it to workspace



Expo Router



Lottie Animations

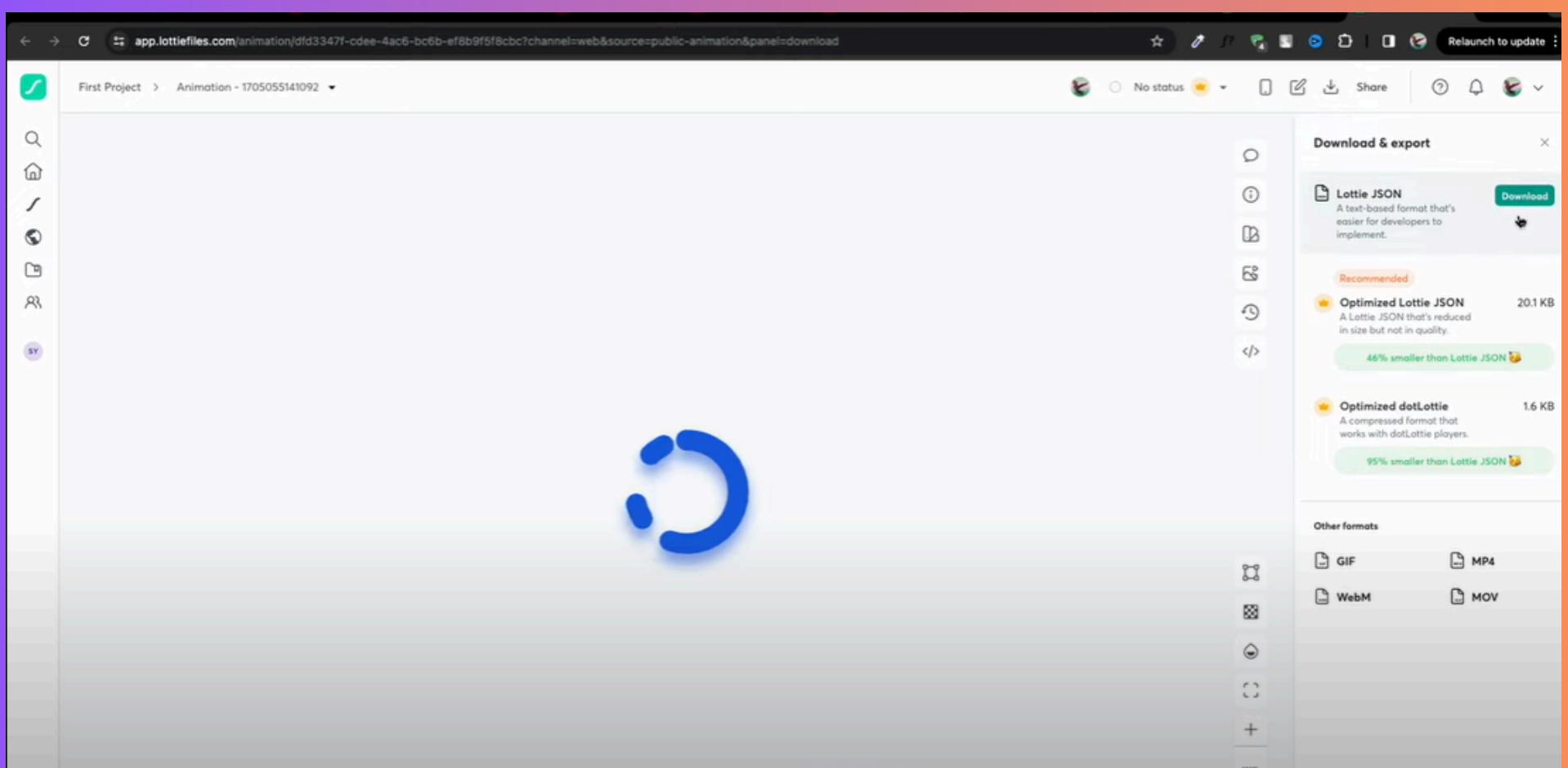
First Install the Package

```
npm i lottie-react-native
```

For the animations Visit the Website

<https://lottiefiles.com/>

In the Editor download the Json File and save it ur assets such as loading.json



Expo Router



Lottie Animations

First Install the Package

```
npm i lottie-react-native
```

For the animations Visit the Website

<https://lottiefiles.com/>

Create the Loading Component and use the Package we installed

```
EXPLORER ... JS index.js U JS authContext.js U JS signin.js U JS Loading.js U X JS _layout.js .../(app) U JS home.js U JS ④ ...  
OPEN EDITORS  
FIREBASE-CHAT  
.expo  
app  
(app)  
JS _layout.js U  
JS home.js U  
JS _layout.js U  
JS index.js U  
JS signin.js U  
JS signUp.js U  
assets  
components  
JS Loading.js U  
constants  
context  
JS authContext.js U  
node_modules  
.gitignore M  
app.json  
babel.config.js M  
TS expo-env.d.ts  
# global.css U  
components > JS Loading.js > Loading  
1 import { View, Text } from 'react-native'  
2 import React from 'react' 6.9k (gzipped: 2.7k)  
3 import LottieView from 'lottie-react-native';  
4  
5 export default function Loading({size}) {  
6   return (  
7     <View style={{height: size, aspectRatio: 1}}>  
8       <LottieView style={{flex: 1}} source={require('../assets/images/loading.json')} autoPlay loop  
9     </View>  
10   )  
11 }
```

Expo Router



Lottie Animations

First Install the Package

```
npm i lottie-react-native
```

For the animations Visit the Website

<https://lottiefiles.com/>

Use the Loading Component

```
JS signin.js > ⚡ Signin
    /* submit button */

    <View>
        {
            loading? (
                <View className="flex-row justify-center">
                    <Loading size={hp(6.5)} />
                </View>
            ) : (
                <TouchableOpacity onPress={handleLogin} style={{height: hp(6.5)}}
                    <Text style={{fontSize: hp(2.7)}} className="text-white font-bold">
                        Sign In
                    </Text>
                </TouchableOpacity>
            )
        }
    </View>

    /* sign up text */

    <View className="flex-row justify-center">
```

Expo Router



Illustrations

Here are some links

<https://www.freepik.com/free-photos-vectors/concept-illustration>

https://www.vecteezy.com/search?qterm=login-concept&content_type=vector

<https://freeillustrations.xyz/#browse>

<https://webpixels.io/illustrations>

Expo Router



Ecommerce designs

Here are some links

<https://dribbble.com/tags/ecommerce-mobile-app>

<https://www.behance.net/search/projects/Ecommerce%20App%20Design>

https://www.behance.net/gallery/197142383/Plant-Shop-Ecommerce-Mobile-App-UIUX-Design-Figma?tracking_source=search_projects|Ecommerce+App+Design&l=22

<https://uiplace.com/ui-school/top-20-ecommerce-app-ui-design-inspiration-1>

https://www.behance.net/gallery/196535627/Grocery-App-Ui-Design-Grocery-Delivery-App-Design?tracking_source=search_projects|Ecommerce+App+Design&l=24

<https://www.insightlancer.com/figma-ui-kits/mobile-app-ui-kit/e-commerce-app-ui-kit/>