

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 editor window for a project named 'expoRouterApp'. The Explorer sidebar on the left shows the file structure with 'index.js' selected in the 'app' directory. The main editor area displays the content of 'index.js', which is a simple React Native page component. The code is as follows:

```
1 import { View, Text } from 'react-native'
2 import React from 'react'
3
4 export default function Page() {
5   return (
6     <View>
7       <Text>Home Page</Text>
8     </View>
9   )
10 }
```

Below the editor, the Terminal window is active, displaying the following instructions:

```
> 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 for your project will appear below. Press Ctrl+C to exit.
> Opening on iOS...
> Opening exp://192.168.0.104:8081 on iPhone 14 Pro Max
> Press ? | show all commands
Unable to resolve asset "../assets/images/icon.png" from "icon" in your app.json or app.config.js
iOS Bundling complete 3917ms
```


Expo Router



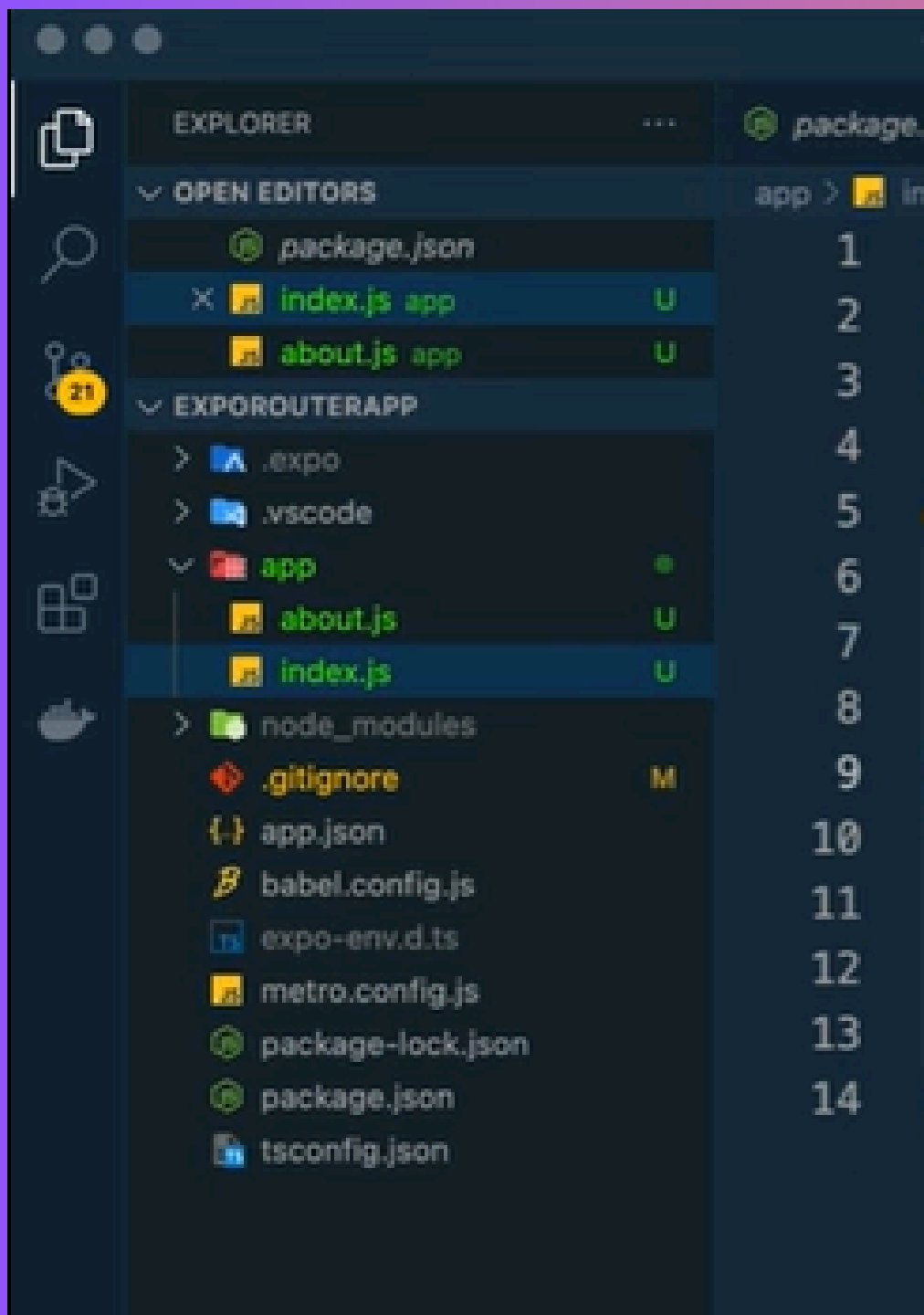
Creating Pages/Screens

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 editor window for an Expo Router application. The Explorer sidebar on the left displays the project structure, including the 'app' directory with 'about.js' and 'index.js' files. The main editor area shows the code for 'index.js', which imports 'View', 'Text', and 'Link' from 'react-native' and 'expo-router'. The code defines a 'Page' component that returns a JSX element containing a 'Text' component with the text 'Home Page' and a 'Link' component with the href attribute set to '/about'. The 'Link' component is highlighted with a blue selection bar.

```
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>

```
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 }
```

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



Routing Between Pages

Use of Button Component

```
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 }
```

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

```
package.json
index.js app
about.js app
index.js app/blog
EXPOROUTERAPP
  .expo
  .vscode
  app
    blog
      index.js
      about.js
      index.js
  node_modules
  .gitignore
  app.json
  babel.config.js
  expo-env.d.ts
```

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

Expo Router



Routing between Pages

Using a Link Component with child Button

```
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    </View>
16  )
17 }
```

The screenshot shows a VS Code editor with the Expo Router code for the `Page` component. The code imports `View`, `Text`, and `Button` from `react-native`, and `Link` from `expo-router`. The `Page` component returns a `View` with a flex layout, containing a `Text` element with a large font size, a `Link` component with a `href` of `"/about"` containing a `Text` element with a smaller font size, and another `Link` component with a `href` of `"/blog"` and the `asChild` prop, containing a `Button` with the title "Go to Blog Page".

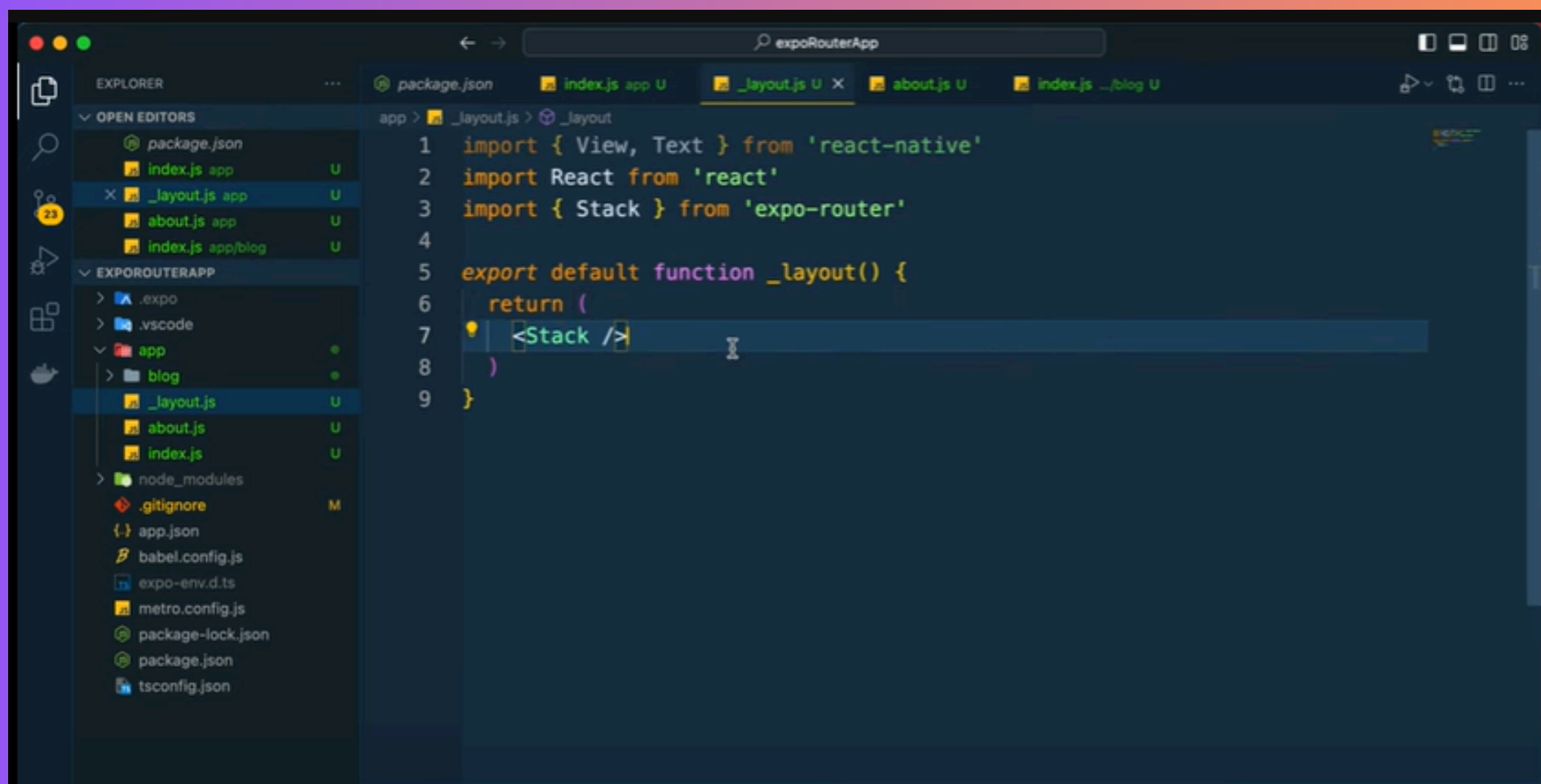
Expo Router



Creating Layout file

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

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



```
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



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/>

```
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)

```
OPEN EDITORS
package.json
index.js app U
_layout.js app U
about.js app U
index.js app/blog U
EXPOROUTERAPP
.expo
.vscode
app
blog
_layout.js U
about.js U
index.js U
node_modules
.gitignore M
app.json
babel.config.js
expo-env.d.ts
metro.config.js
package-lock.json
package.json
tsconfig.json

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 }
```

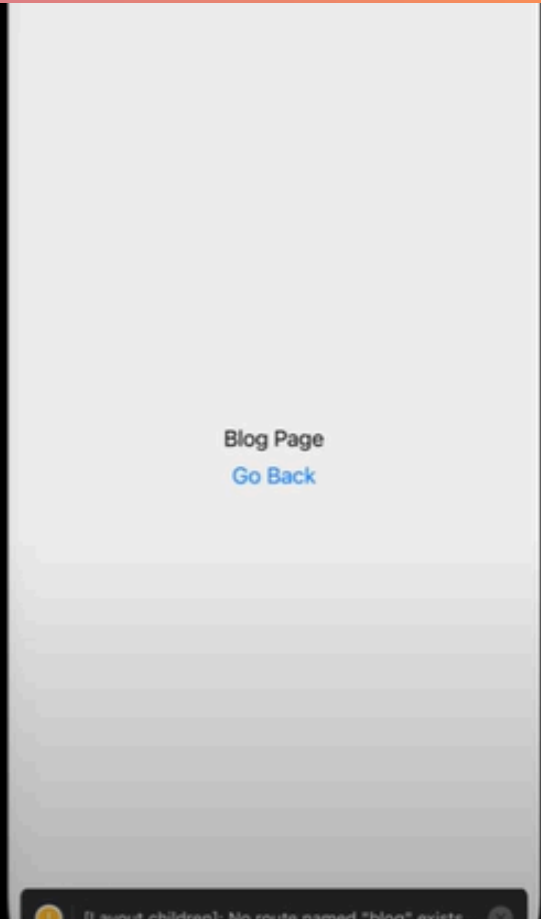
Expo Router



Layout file /Customising the App Bar

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

```
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 }
```



```
at App
at withDevTools(App) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&shot=false&lazy=true:1305:27)
at ErrorToastContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&shot=false&lazy=true:1374:24)
at ErrorOverlay
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&shot=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&shot=false&lazy=true:40380:43)
at AppContainer (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&shot=false&lazy=true:40220:36)
at main(RootComponent) (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle//&platform=ios&dev=true&shot=false&lazy=true:1254:28)
```

[Layout children]: No route named "blog" exists...

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

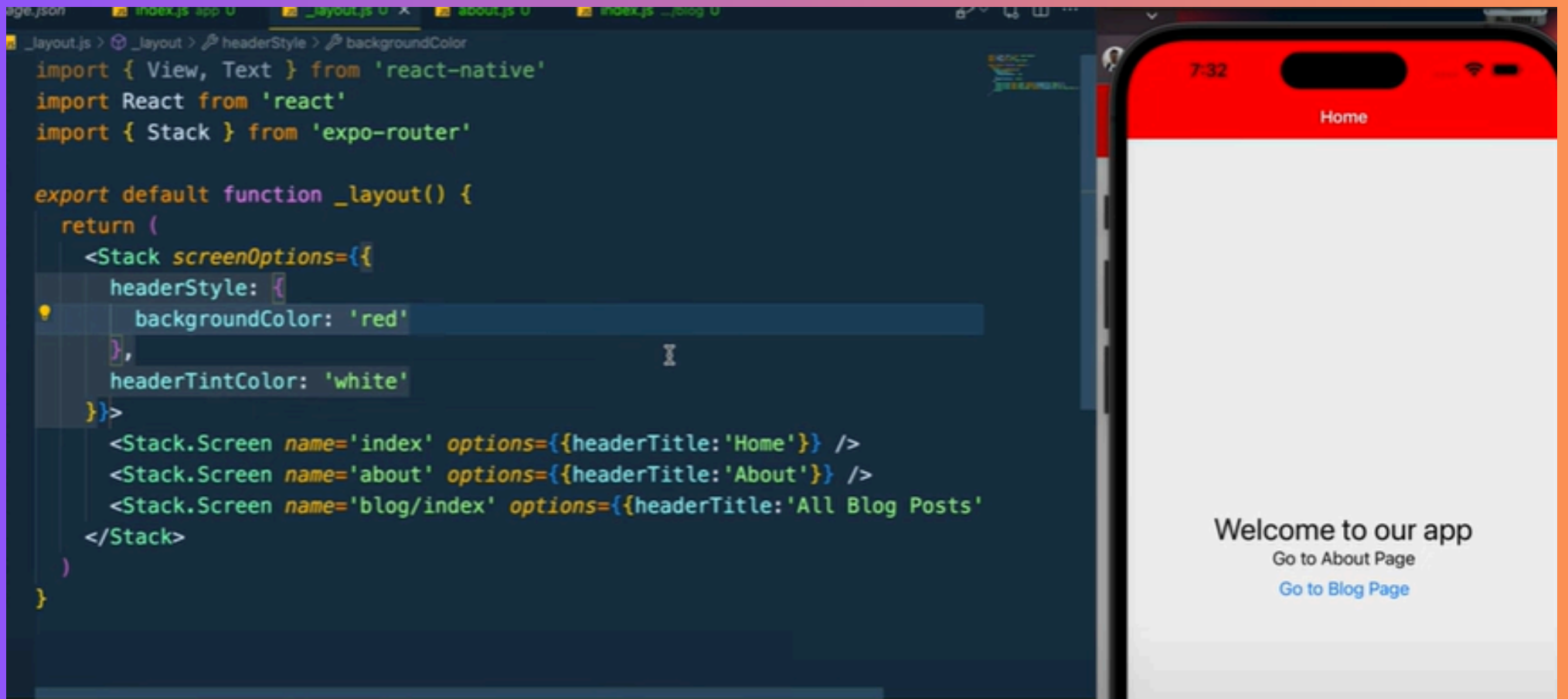
```
app > _layout.js > _layout
1  import { View, Text } from 'react-native'
2  import React from 'react'
3  import { Stack } from 'expo-router'
4
5  export default function _layout() {
6    return (
7      <Stack>
8        <Stack.Screen name='index' options={{headerTitle:'Home'}} />
9        <Stack.Screen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

```
package.json index.js app U _layout.js U x contact.js U about.js U index.js .../blog U
app > _layout.js > _layout
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')}}
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>
28 );
29 }
30
```


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/>

```
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" }}
```

```
tack.Screen name="contact" options={{ headerTitle: "About", presentation: "
  >
```

```
Screen(props:
  ScreenProps<NativeStackNavigationOptions
  StackNavigationState<ParamListBase>,
  NativeStackNavigationEventMap>): null
```

- card
- containedModal
- containedTransparentModal
- formSheet
- fullScreenModal
- modal
- transparentModal

MS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
ex.js:10862:23)
FlowParserMixin.parseUpdate (/Users/pradipdebnath/Desktop/Tutorials/
845:21)
FlowParserMixin.parseMaybeUnary (/Users/pradipdebnath/Desktop/Tutori
s:10821:23)
FlowParserMixin.parseMaybeUnaryOrPrivate (/Users/pradipdebnath/Deskt
b/index.js:10659:61)
FlowParserMixin.parseExprOps (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index
8664:23)
FlowParserMixin.parseMaybeConditional (/Users/pradipdebnath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser
index.js:10641:23)
FlowParserMixin.parseMaybeAssign (/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>

```
expo-router-app
├── index.js app U
├── _layout.js U
├── [id].js U
├── contact.js U
├── about.js U
└── index.js _/blog U x

> 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')} title='Go To Blog 3' />
13       <Button onPress={() => router.back()} title='Go Back' />
14     </View>
15   )
16 }
```

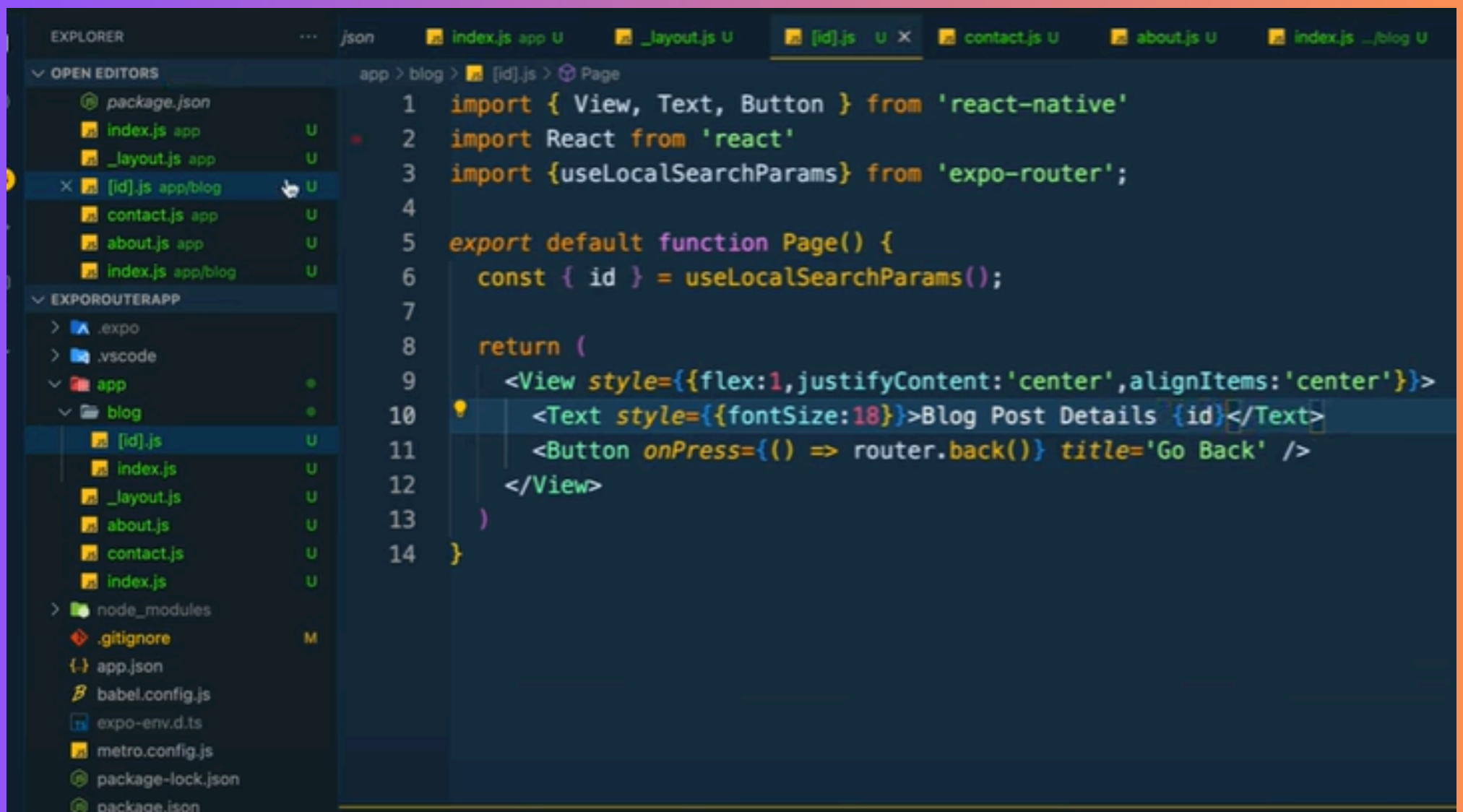

Expo Router



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**”



```
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 }
```

Expo Router



Customising the Header for the Detailed Page

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

```
index.js app U  _layout.js U  [id].js U x  contact.js U  about.js U  index.js .../blog U
> blog > [id].js > Page > headerTitle
1  import { View, Text, Button } from 'react-native'
2  import React from 'react'
3  import {useLocalSearchParams, router, Stack} from 'expo-router';
4
5  export default function Page() {
6    const { id } = useLocalSearchParams();
7
8    return (
9      <Stack.Screen options={{headerTitle: `Article ${id}`} />
10     <View style={{flex:1,justifyContent:'center',alignItems:'center'}}>
11       <Text style={{fontSize:18}}>Blog Post Details {id}</Text>
12       <Button onPress={() => router.back()} title='Go Back' />
13     </View>
14   </>
15 )
16 }
17 }
```

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 }
```

Expo Router



Passing Multiple Params

We can also pass multiple parama as shown below

```
bg > [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
import { Link, useRouter } from 'expo-router';

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

```
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 }
```

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



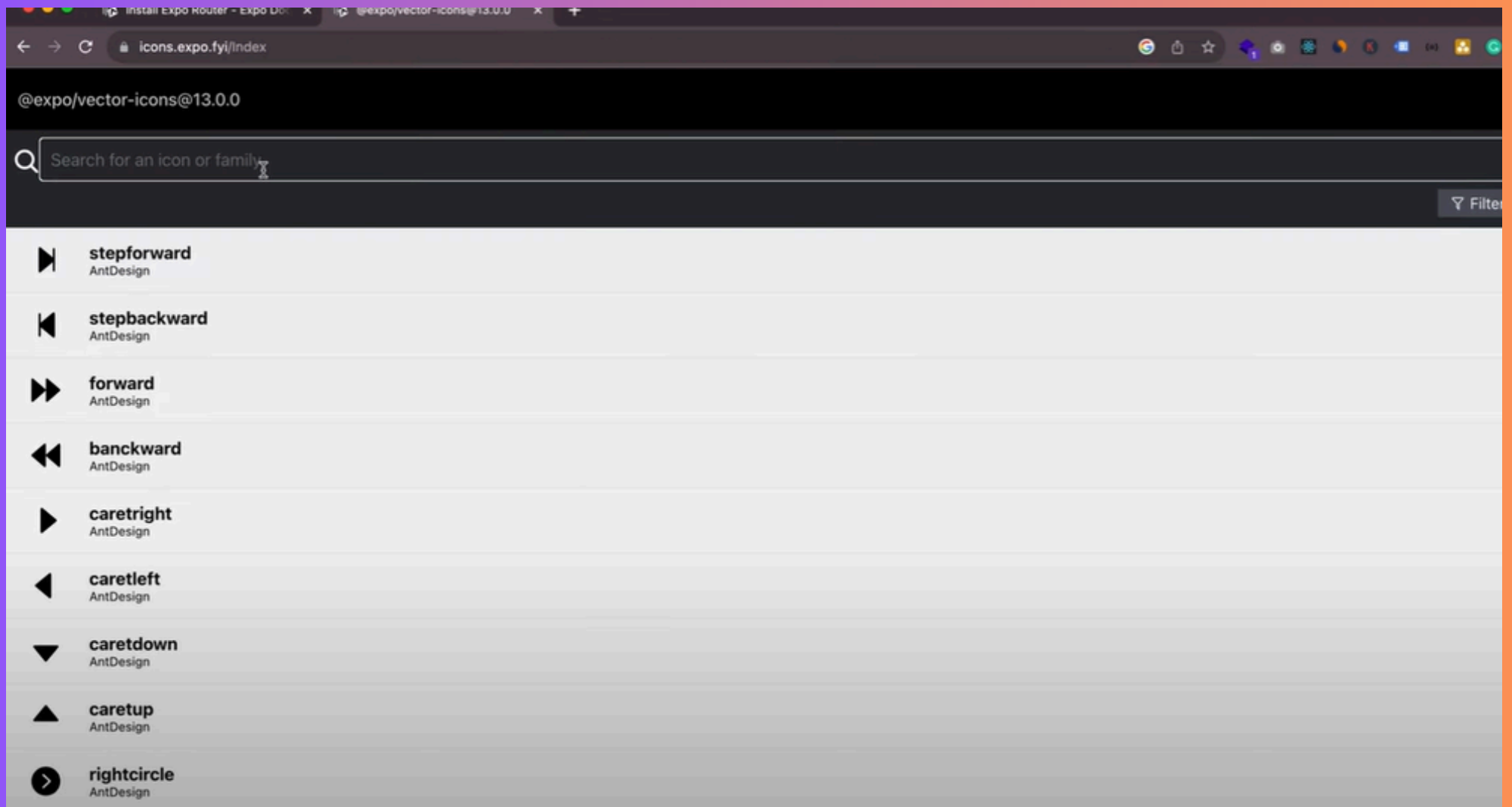
Customise the Tabs as By expanding the Tabs

```
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 > _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

```
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", present
28 <Stack.Screen name="(tabs)" options={{ headerShown: false }} />
29 </Stack>
30 );
31 }
32
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
index.js:6920:17)
    at FlowParserMixin.parseExprAtom (/Users/pradipdeb Nath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index.js:6932:19)
    at FlowParserMixin.parseExprSubscripts (/Users/pradipdeb Nath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index.js:10862:23)
    at FlowParserMixin.parseUpdate (/Users/pradipdeb Nath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index.js:10845:21)
    at FlowParserMixin.parseMaybeUnary (/Users/pradipdeb Nath/Desktop/Tutorials/react-native/expoRouterApp/node_modules/@babel/parser/lib/index.js:10821:23)
    Unable to resolve asset "../assets/images/icon.png" from "icon" in your app.json or app.config.js
405 Bundling complete 67ms
```

Expo Router



Creating Nested pages within the Tab Navigator

The screenshot shows a VS Code editor with the Expo Router configuration for a nested page. The Explorer sidebar on the left shows the project structure with the 'feed' folder expanded, containing 'index.js', 'new.js', and '_layout.js'. The main editor displays the following code in '_layout.js':

```
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 Explorer sidebar on the left and a code editor on the right. The Explorer sidebar shows the project structure for 'EXPOROUTERAPP', including folders like '.expo', '.vscode', 'app', '(tabs)', and 'feed'. The 'feed' folder is expanded, showing files like '_layout.js', 'index.js', 'new.js', '_layout.js', 'profile.js', 'blog', '_layout.js', 'about.js', 'contact.js', and 'index.js'. The code editor shows the content of the selected '_layout.js' file in the 'feed' folder, which is currently empty except for the closing tag of a Stack component.

```
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 editor window with the following structure:

- EXPLORER:** Shows a project structure with folders like `app`, `blog`, and `node_modules`. The `app` folder contains a `(tabs)` subfolder with a `feed` subfolder. The `feed` folder contains `_layout.js`, `index.js`, and `new.js`.
- EDITOR:** Displays the content of `app > (tabs) > _layout.js`. The code defines a `<Tabs>` component with two screens: `feed` and `profile`. The `feed` screen uses a `<Feather name="list" size={24} color={color} />` icon, and the `profile` screen uses a `<AntDesign name="user" size={24} color={color} />` icon.
- TERMINAL:** Shows a stack trace error: `Unable to resolve asset "../assets/images/icon.png" from "icon" in your app.json or app.config.js`. The error originates from `ErrorToastContainer` and `ErrorOverlay` components.

Expo Router



Add the header Right Button on the Feed Screen

```
app > (tabs) > _layout.js > _layout > headerRight
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')} tit
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 }
```

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
0474:24) 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:12
0954:28)
Unable to resolve asset "../assets/images/icon.png" from "icon" in your app.json or app.config.js
105 Bundling complete 57ms

Expo Router



Remove the unnecessary Header bar

The image shows a development environment with a code editor on the left and a mobile app preview on the right. The code editor displays the following JavaScript code:

```
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 mobile app preview shows a screen with the title "Feed" and a button "Add Post". The main content area is labeled "Feed Page". At the bottom of the screen, there is a "Refreshing..." indicator and a message: "Don't see your changes? Reload the app".

The terminal at the bottom of the IDE shows the following error message:

```
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
0474:24)
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:40388:43)
at RCTView
at View (http://192.168.0.104:8081/node_modules/expo-router/entry.bundle/?platform=ios&dev=true&hot=false&lazy=true:40388:43)
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:12
0954:28)
Unable to resolve asset "../assets/images/icon.png" from "icon" in your app.json or app.config.js
105 Bundling complete 52ms
```

Expo Router



Adding Button on the New Page

```
> (tabs) > feed > new.js > 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:134:24)
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:124:28)
```

Expo Router



Drawer Navigation

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

The screenshot shows a VS Code editor with the following content:

```
EXPLORER
  OPEN EDITORS
    index.js app U
    babel.config.js M
    _layout.js app/(drawer) U
    _layout.js app/(drawer)/... U
    _layout.js app/(drawer)/... U
    _layout.js app U
    [id].js app/blog U
    contact.js app U
    about.js app U
    index.js app/blog U
    index.js app/(drawer)/(t... U
  EXPOROUTERAPP
    .expo
    .vscode
    app
      (drawer)
        (tabs)
        _layout.js U
      blog
        _layout.js U
        about.js U
        contact.js U
        index.js U
    node_modules
    .gitignore M
    app.json
    babel.config.js M
    expo-env.d.ts

babel.config.js M
_layout.js _/(drawer) U x
_layout.js _/(tabs) U
_layout.js _/feed U
_layout.js

app > (drawer) > _layout.js > Layout
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 }
```

Expo Router



Drawer Navigation

Creating Custom Drawer Component to to be Customise the Drawer content

```
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'
5
6 const CustomDrawerContent = (props) => {
7   <DrawerContentScrollView {...props}>
8     |
9   </DrawerContentScrollView>
10 }
11
12 export default function Layout() {
13   return (
14     <Drawer drawerContent={({props}) => <CustomDrawerContent {...props} /> /> />
15   )
16 }
```

The screenshot shows a VS Code editor window for an Expo Router application. The Explorer sidebar on the left shows the project structure, including a 'drawer' directory with a '_layout.js' file. The main editor displays the code for 'CustomDrawerContent' and the 'Layout' function. The 'Layout' function uses the 'Drawer' component from 'expo-router/drawer' and the 'DrawerContentScrollView' from '@react-navigation/drawer' to render the custom drawer content. The terminal at the bottom shows the application running on a device, with a stack trace indicating an error in the drawer content rendering process.

Expo Router



Drawer Navigation

We can add Tabs Links to the Custom Drawer

The image shows a VS Code editor window on the left and a mobile app preview on the right. The editor displays the following code in `_layout.js`:

```
import { DrawerContentScrollView, DrawerItem } from "@react-navigation/drawer";
import { Feather, AntDesign } from "@expo/vector-icons";
import { router } from "expo-router";

const CustomDrawerContent = (props) => {
  return(
    <DrawerContentScrollView {...props}>
      <DrawerItem
        icon={({ color, size }) => (
          <Feather name="list" size={24} color={color} />
        )}
        label="Feed"
        onPress={() => {
          router.push('/(drawer)/(tabs)/feed');
        }}
      />
      <Text>Hello</Text>
    </DrawerContentScrollView>
  );
};
```

The mobile app preview on the right shows a screen with a top bar containing the text "(tabs)" and "Feed", and a blue "Add Post" button. The main content area displays "Feed Page".

Expo Router



Drawer Navigation

We can add other pages that not on the tabs

The screenshot shows a VS Code editor window with the following structure:

- EXPLORER:** Shows the project structure with folders like `app`, `blog`, and `node_modules`. The `app` folder contains subfolders `(drawer)` and `(tabs)`. The `(drawer)` folder contains `_layout.js`, `favourites.js`, and `settings.js`.
- OPEN EDITORS:** Shows several instances of `_layout.js` files, with the active one being `app > (drawer) > _layout.js`.
- Code Editor:** Displays the content of `CustomDrawerContent` in `_layout.js`. The code defines a `DrawerContentScrollView` with two `DrawerItem` components:
 - The first `DrawerItem` has a label "Favourites" and an `onPress` handler that pushes the router to `/favourites`.
 - The second `DrawerItem` has a label "Settings" and an `onPress` handler that pushes the router to `/settings`.
- TERMINAL:** Shows the output of the application, including error messages from the Babel parser.

Expo Router



Drawer Navigation

Styling the Icon

```
app > (drawer) > _layout.js > CustomDrawerContent > fontSize
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 );
```

PROBLEMS OUTPUT **TERMINAL** PORTS DEBUG CONSOLE

```
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:40300:43)
at RCTView
at View (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//platform=ios&dev=true&hot=false&lazy=true:40300:43)
at AppContainer (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle//platform=ios&dev=true&hot=false&lazy=true:40220:36)
```


Expo Router



Drawer Navigation

To add active Drawer item , we use Pathname

The screenshot shows a VS Code editor with the following code in `_layout.js` inside the `(drawer)` folder:

```
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       />
24     </DrawerContentScrollView>
25   );
26 }
```

The terminal at the bottom shows the following output:

```
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(RootComponent) (http://192.168.0.102:8081/node_modules/expo-router/entry.bundle/?platform=ios&dev=true&hot=false&lazy=true:120954:28)
LOG /profile
LOG /feed
```


Expo Router



Drawer Navigation

Using the pathname to show the Active Item

```
> (drawer) > _layout.js > CustomDrawerContent > backgroundColor
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} />
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

zsh + v [] [] ... ^ X

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)

Expo Router



Drawer Navigation

Add Other Content in the Drawer

The image displays a development environment with two main components: a code editor on the left and a mobile app preview on the right.

Code Editor (Left): Shows the implementation of a drawer navigation menu in a file named `CustomDrawerContent`. The code uses `DrawerContentScrollView` to wrap the content. It includes a user profile section with an image and text, and a `DrawerItem` for a navigation menu. The menu items are:

- Feed
- Profile
- Favourites
- Settings

The `DrawerItem` uses a `Feather` icon named "list" and dynamically sets its color based on the current route: `color={pathname === "/feed" ? "#fff" : "#000"}`.

Mobile App Preview (Right): Shows a mobile application interface. At the top, it displays the time "8:49" and the user's profile information: "John Doe" and "john@email.com". Below the profile, there is a navigation drawer menu with the following items: "Feed", "Profile", "Favourites", and "Settings". The "Feed" item is currently selected and highlighted.

Expo Router



Drawer Navigation

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

```
app > (drawer) > _layout.js > Layout
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: 19,
```

Expo Router



Drawer Navigation

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

The image shows a VS Code editor window with the Expo Router code for a drawer navigation setup. The code is in the file `app > (drawer) > (tabs) > _layout.js`. The code imports `tabs` and `router` from `expo-router`, `Feather` and `AntDesign` from `@expo/vector-icons`, and `DrawerToggleButton` from `@react-navigation/drawer`. The `_layout` function returns a `Tabs` component with two screens: `feed` and `profile`. The `feed` screen has a `headerLeft` prop set to `<DrawerToggleButton />`. The `profile` screen has a `headerTitle` prop set to `'Profile'`. The `terminal` panel at the bottom shows the following output:

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

On the right, a mobile emulator (iPhone 14 Pro Max, iOS 16.4) shows the app running. The top status bar displays the time 8:55. The app interface shows a drawer navigation with a hamburger menu icon on the left and the text "Profile" on the right. The main content area is titled "Profile Page".

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

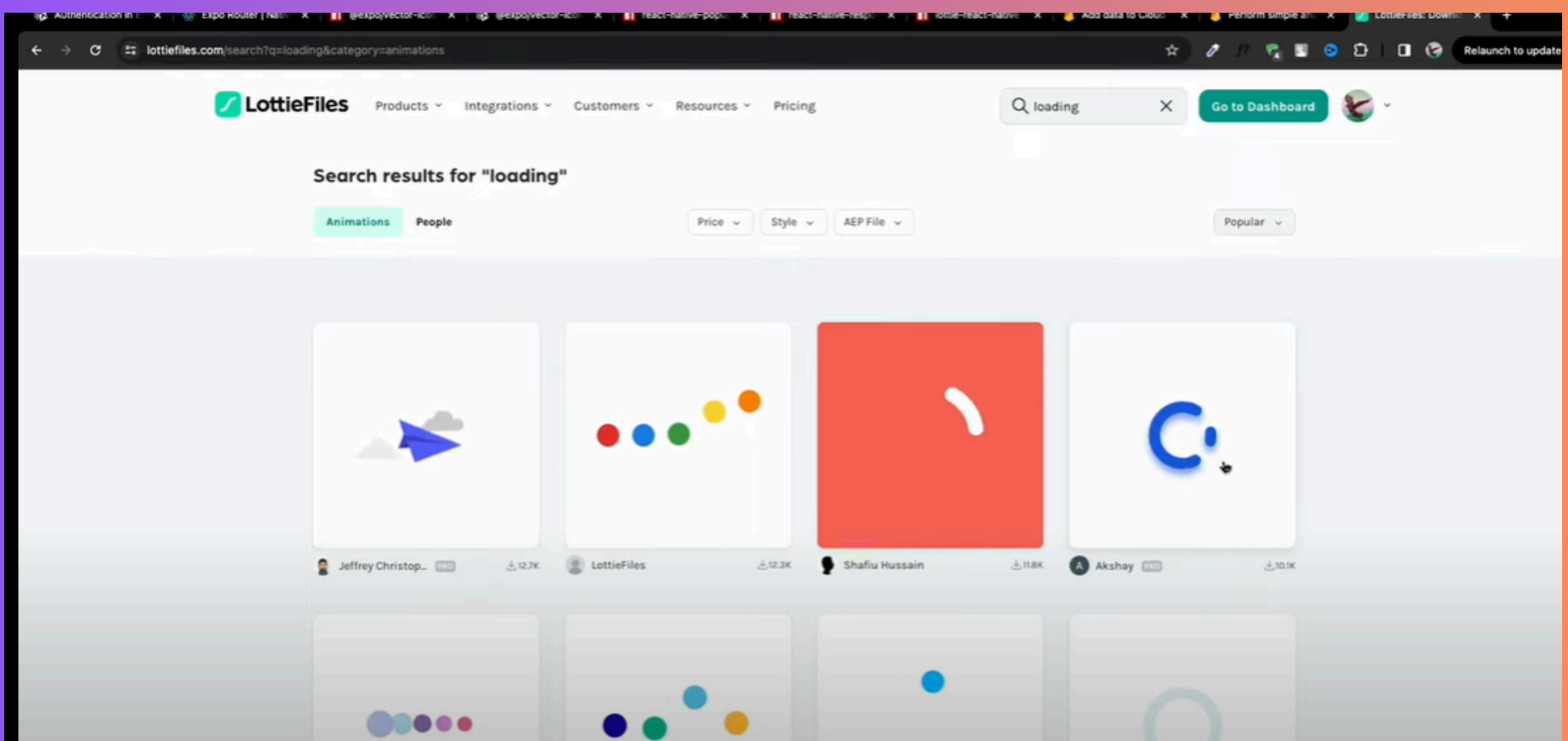
First Install the Package

```
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

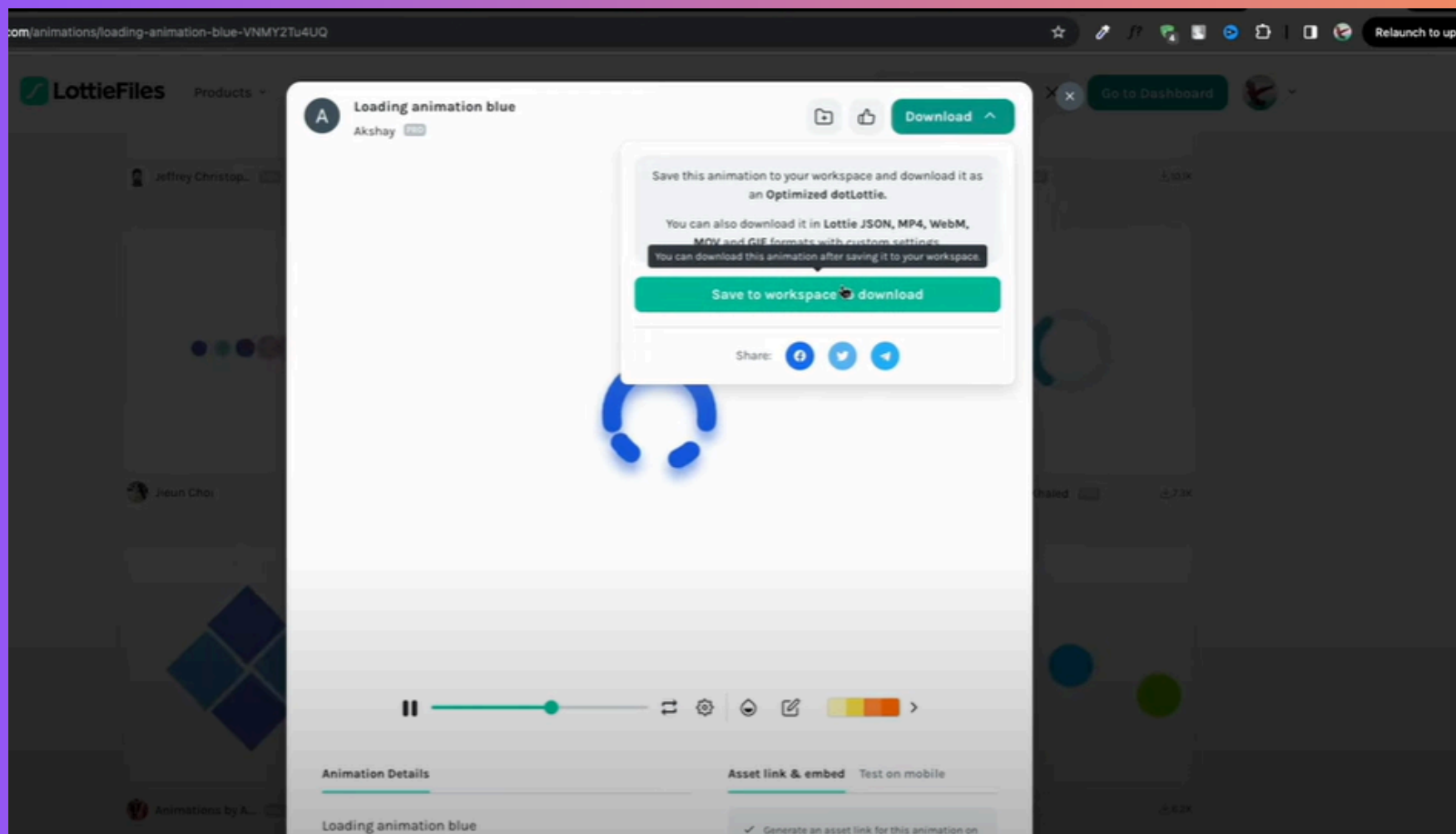
First Install the Package

```
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

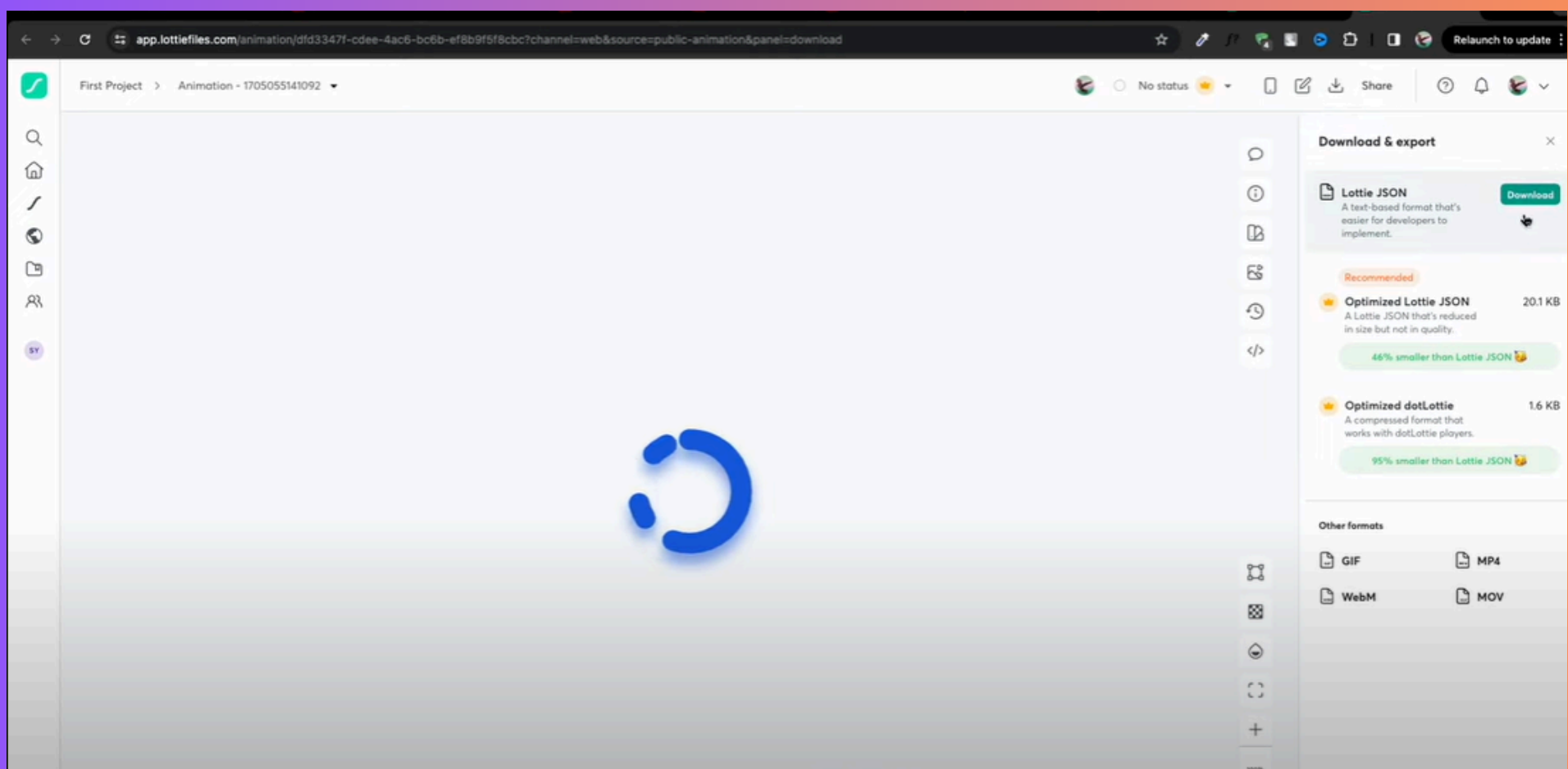
First Install the Package

```
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

```
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)}} clas
            <Text style={{fontSize: hp(2.7)}} className="text-white font-bol
              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/>