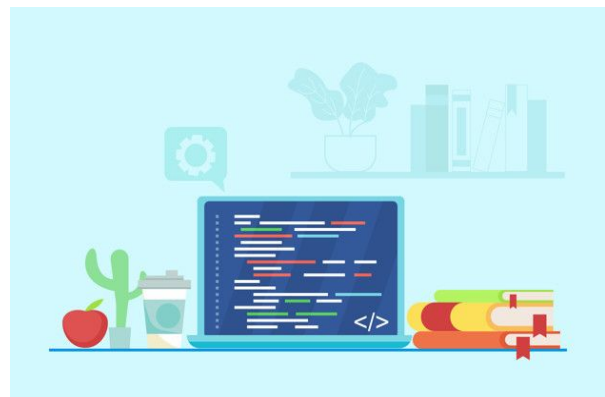


## Book Santa- SignUp and Login



### What is our GOAL for this MODULE?

We got introduced to the Book Santa application. We identified the problem the app is trying to solve and how it aims to do so. We wrote the user stories for the app and learned to design the workflow of the application.

### What did we ACHIEVE in the class TODAY?

- Create Sign Up and Login Screen
- Authenticate users after they have signed up
- Add Lottie Animation on Sign Up Page

### Which CONCEPTS/CODING BLOCKS did we cover today?

- User Authentication
- Lottie Animation

### How did we DO the activities?

## Activity 1: Create new Expo project and Design UI for SignUp Login page.

We started by creating a new expo project.

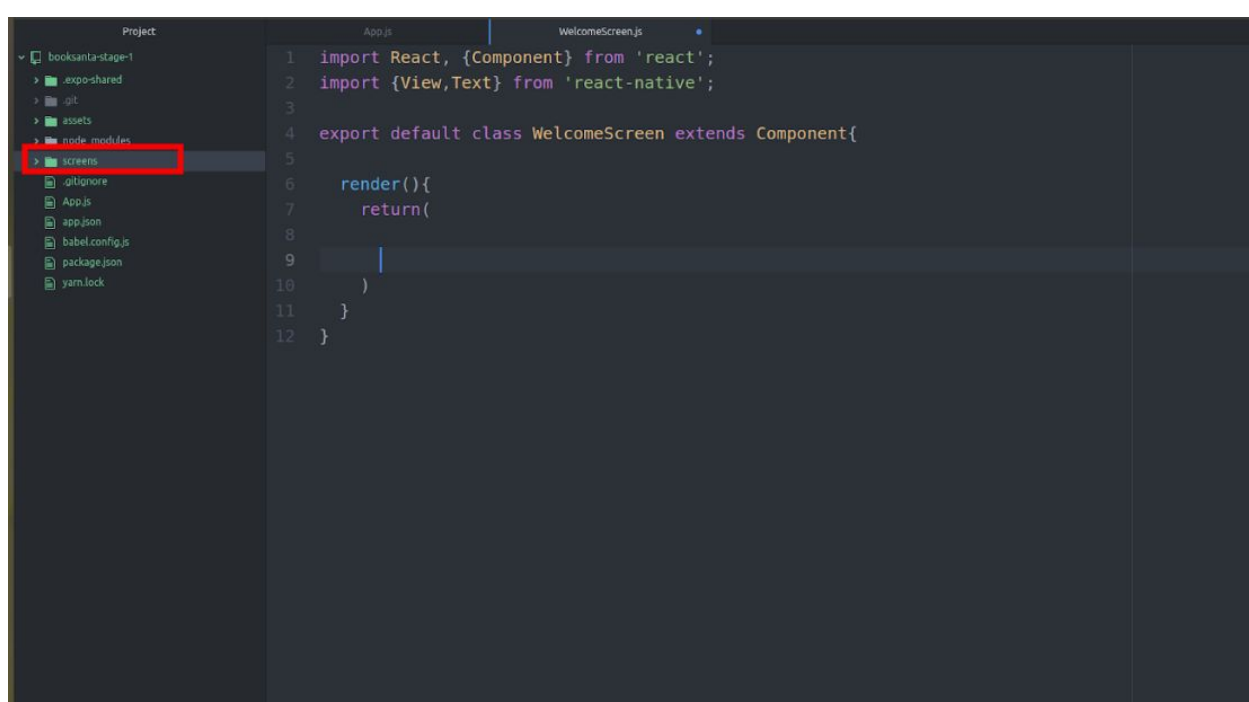
1. We ran the **expo init bookSanta-stage-1** command in the terminal.

```
vishu@oasis ~/Desktop> expo init bookSanta-stage-1

There is a new version of expo-cli available (3.13.1).
You are currently using expo-cli 3.6.0
Install expo-cli globally using the package manager of your choice; for example: 'npm install -g expo-cli' to get the latest version

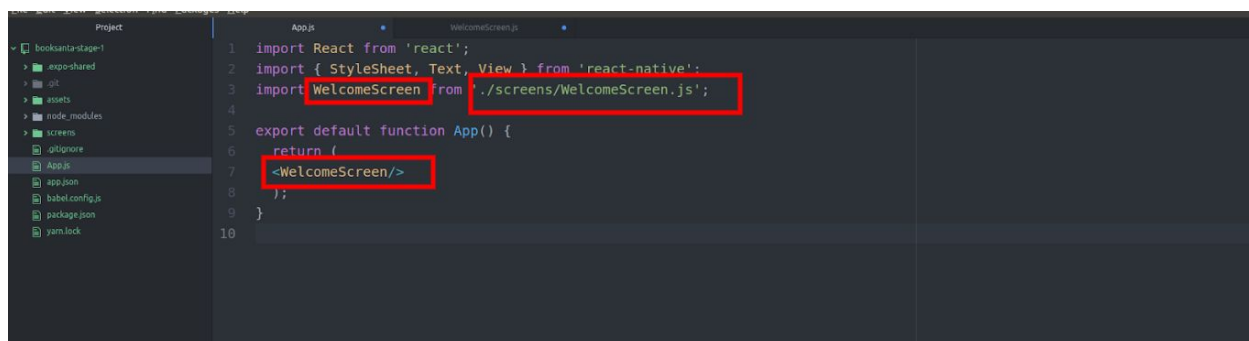
? Choose a template: expo-template-blank
✓ Please enter a few initial configuration values.
  Read more: https://docs.expo.io/versions/latest/workflow/configuration/ · 100% completed
? Yarn v1.21.1 found. Use Yarn to install dependencies? Yes
Extracting project files...
Customizing project...
Initialized a git repository.
Installing dependencies...
yarn install v1.21.1
info No lockfile found.
[1/4] Resolving packages...
warning expo > fbemitter > fbjs > core-js@1.2.7: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > fbjs > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > fbjs-scripts > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > metro-babel-register > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native-web > deep-assign@3.0.0: Check out 'lodash.merge' or 'merge-options' instead.
[2/4] Fetching packages...
info fsevents@1.2.11: The platform "linux" is incompatible with this module.
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
warning Your current version of Yarn is out of date. The latest version is "1.22.0", while you're on "1.21.1".
info To upgrade, run the following command:
```

We created a **WelcomeScreen.js** file in a screens folder and Designed our UI in it.



The screenshot shows the VS Code editor with a project named 'booksanta-stage-1'. The file explorer on the left shows the 'screens' folder highlighted. The main editor displays the 'WelcomeScreen.js' file with the following code:

```
1 import React, {Component} from 'react';
2 import {View,Text} from 'react-native';
3
4 export default class WelcomeScreen extends Component{
5
6   render(){
7     return(
8
9     |
10  )
11  }
12 }
```



The screenshot shows the VS Code editor with the same project. The 'WelcomeScreen.js' file has been updated with the following code:

```
1 import React from 'react';
2 import { StyleSheet, Text, View } from 'react-native';
3 import WelcomeScreen from './screens/WelcomeScreen.js';
4
5 export default function App() {
6   return (
7     <WelcomeScreen/>
8   );
9 }
10
```

```

Project
  booksanta-stage-1
    expo
    expo-shared
    git
    assets
    node_modules
    screens
    gitignore
    App.js
    app.json
    babel.config.js
    package.json
    yarn.lock

App.js
  2 import {View,Text,TextInput,StyleSheet} from 'react-native';
  3
  4 export default class WelcomeScreen extends Component{
  5   constructor(){
  6     super();
  7     this.state={
  8       emailId:'',
  9       password:''
  10    }
  11  }
  12  render(){
  13    return(
  14      <View style={styles.container}>
  15        <View>
  16          <Text style={styles.title}>Book Santa</Text>
  17
  18          </View>
  19          <TextInput
  20            style={styles.loginBox}
  21            placeholder="abc@example.com"
  22            keyboardType ='email-address'
  23            onChangeText={(text)=>{
  24              this.setState({
  25                emailId: text
  26              })
  27            }}
  28          />
  29          <TextInput
  30            style={styles.loginBox}
  31            secureTextEntry = {true}
  32

```

```

WelcomeScreen.js — booksanta-stage-1/screens — ~/Desktop/booksanta-stage-1 — Atom

Edit View Selection Find Packages Help

Project
  booksanta-stage-1
    expo
    expo-shared
    git
    assets
    node_modules
    screens
    gitignore
    App.js
    app.json
    babel.config.js
    package.json
    yarn.lock

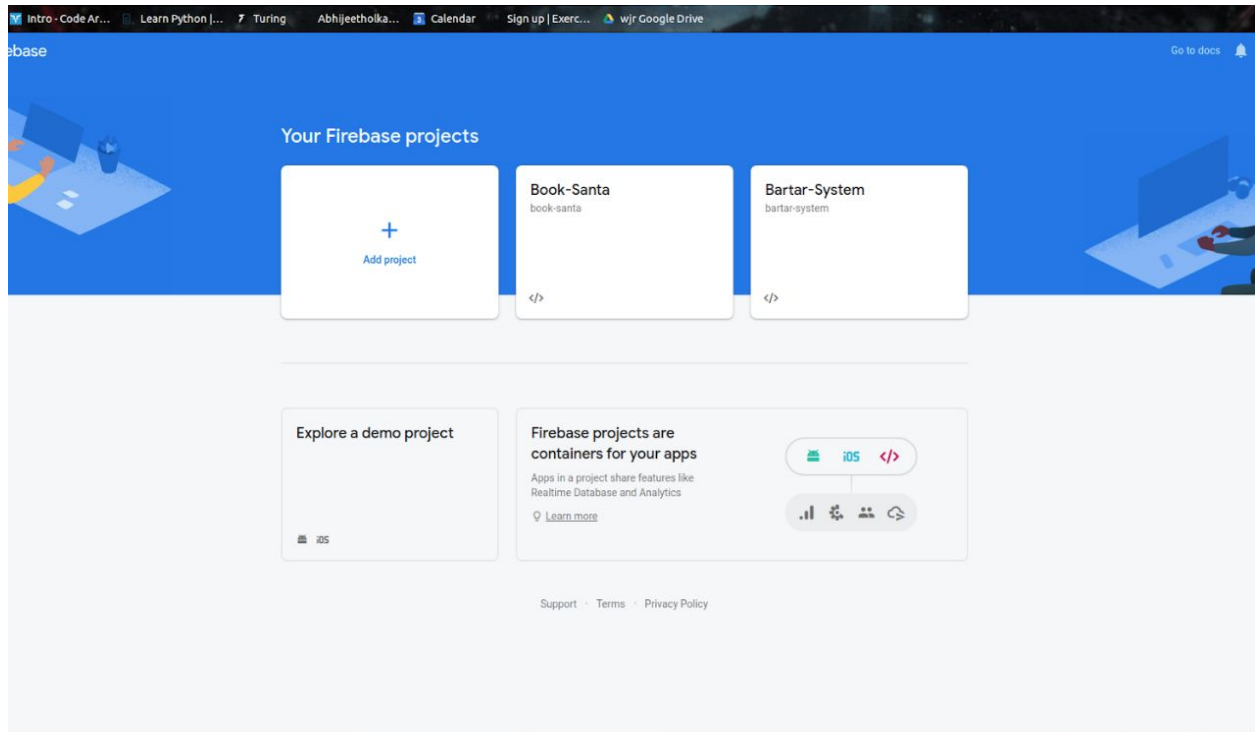
App.js
  21 style={styles.loginBox}
  22 placeholder="abc@example.com"
  23 keyboardType ='email-address'
  24 onChangeText={(text)=>{
  25   this.setState({
  26     emailId: text
  27   })
  28 }}
  29 />
  30 <TextInput
  31   style={styles.loginBox}
  32   secureTextEntry = {true}
  33   placeholder="enter Password"
  34   onChangeText={(text)=>{
  35     this.setState({
  36       password: text
  37     })
  38   }}
  39 />
  40 </View>
  41 </View>
  42 )
  43 }
  44 }
  45
  46 const styles = StyleSheet.create({
  47   container:{
  48     flex:1,
  49     backgroundColor: '#F8BE85'
  50   },
  51   profileContainer:{

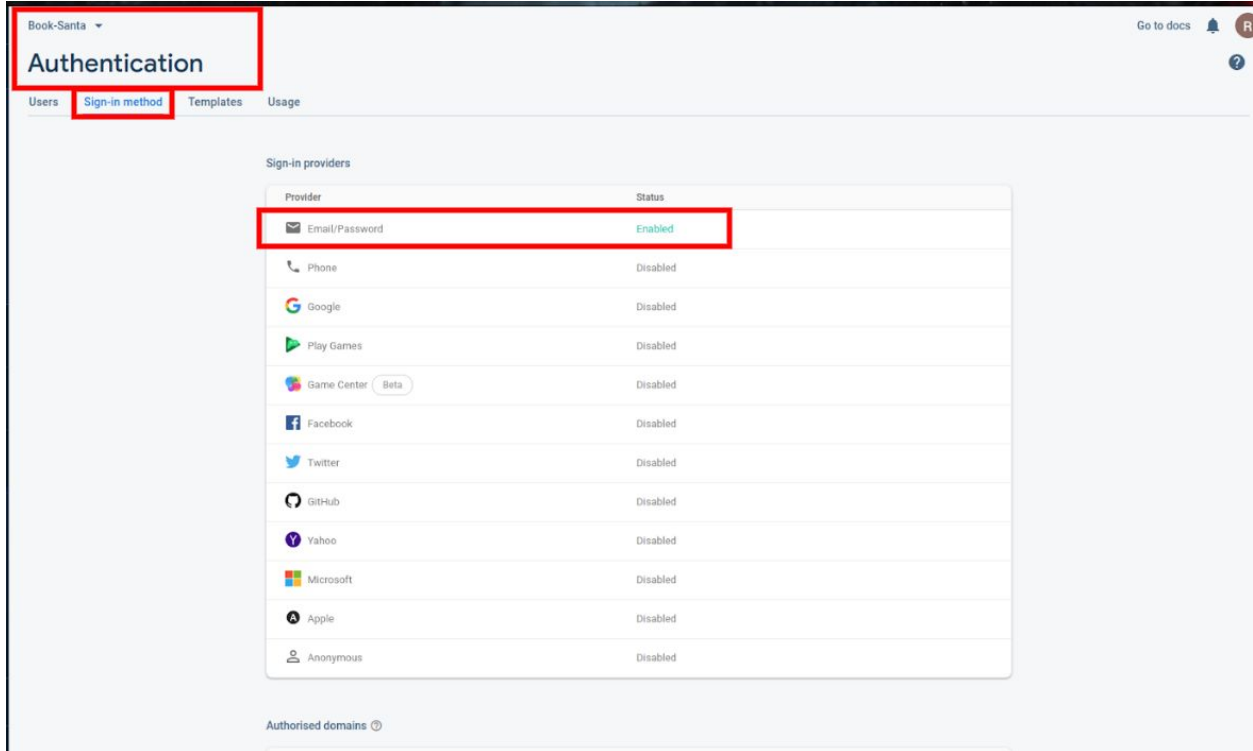
```

We create 2 buttons using Touchable opacity and Text SignUp and Login respectively.

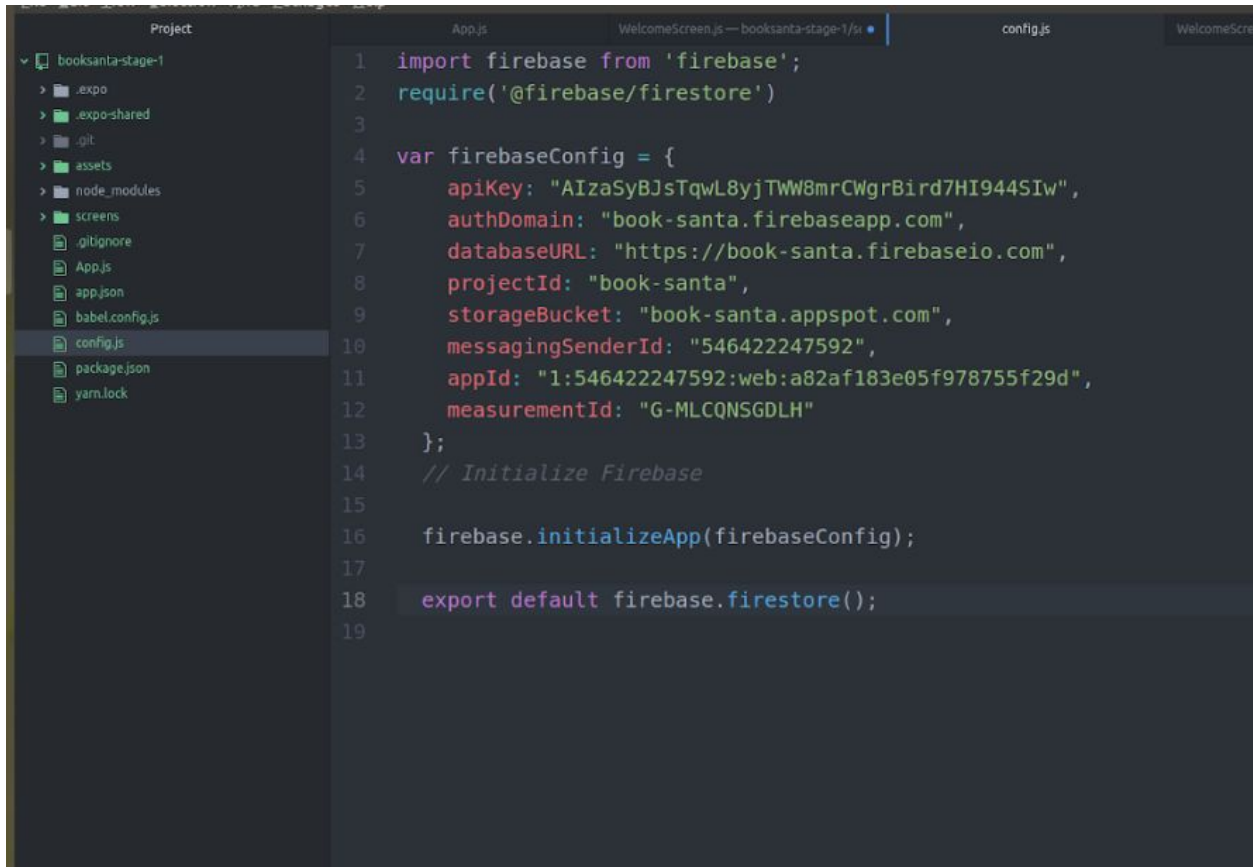
## Activity 2: Create firebase database and import firebase in our project.

We created a new firebase database with name of Book-Santa and created a config.js file





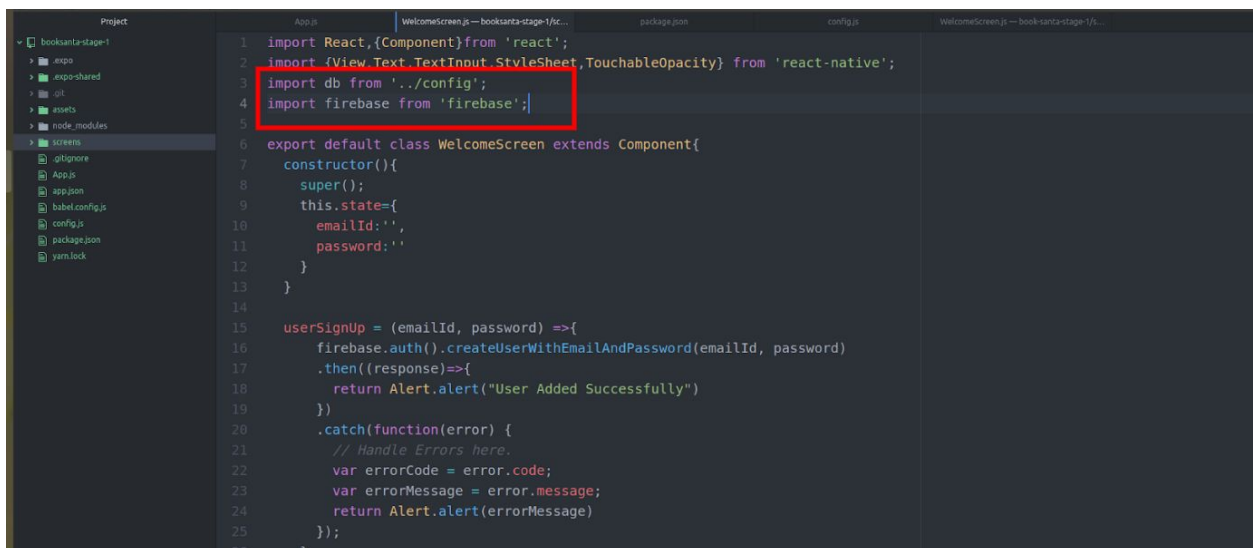
Adding config.js file and importing firebase to app.



```

1  import firebase from 'firebase';
2  require('@firebase/firestore')
3
4  var firebaseConfig = {
5    apiKey: "AIzaSyBJsTqwL8yjTWw8mrCWgrBird7HI944SIw",
6    authDomain: "book-santa.firebaseio.com",
7    databaseURL: "https://book-santa.firebaseio.com",
8    projectId: "book-santa",
9    storageBucket: "book-santa.appspot.com",
10   messagingSenderId: "546422247592",
11   appId: "1:546422247592:web:a82af183e05f978755f29d",
12   measurementId: "G-MLCQNSGDLH"
13 };
14 // Initialize Firebase
15
16 firebase.initializeApp(firebaseConfig);
17
18 export default firebase.firestore();
19

```

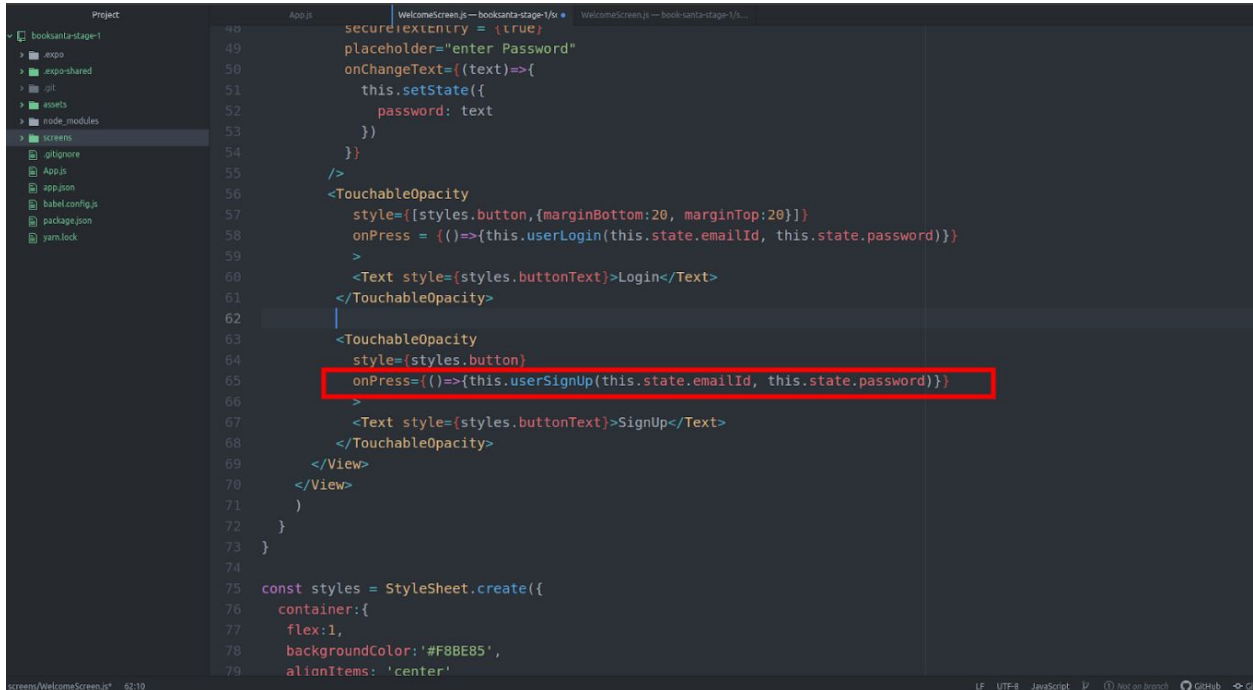


```

1  import React, {Component} from 'react';
2  import {View, Text, TextInput, StyleSheet, TouchableOpacity} from 'react-native';
3  import db from '../config';
4  import firebase from 'firebase';
5
6  export default class WelcomeScreen extends Component{
7    constructor(){
8      super();
9      this.state={
10        emailId:'',
11        password:''
12      }
13    }
14
15    userSignUp = (emailId, password) =>{
16      firebase.auth().createUserWithEmailAndPassword(emailId, password)
17        .then((response)=>{
18          return Alert.alert("User Added Successfully")
19        })
20        .catch(function(error) {
21          // Handle Errors here.
22          var errorCode = error.code;
23          var errorMessage = error.message;
24          return Alert.alert(errorMessage)
25        });
26    }
27  }

```

We create a userSignUp function which accepts email and password as arguments. We call the function when the SignUp Button is pressed.

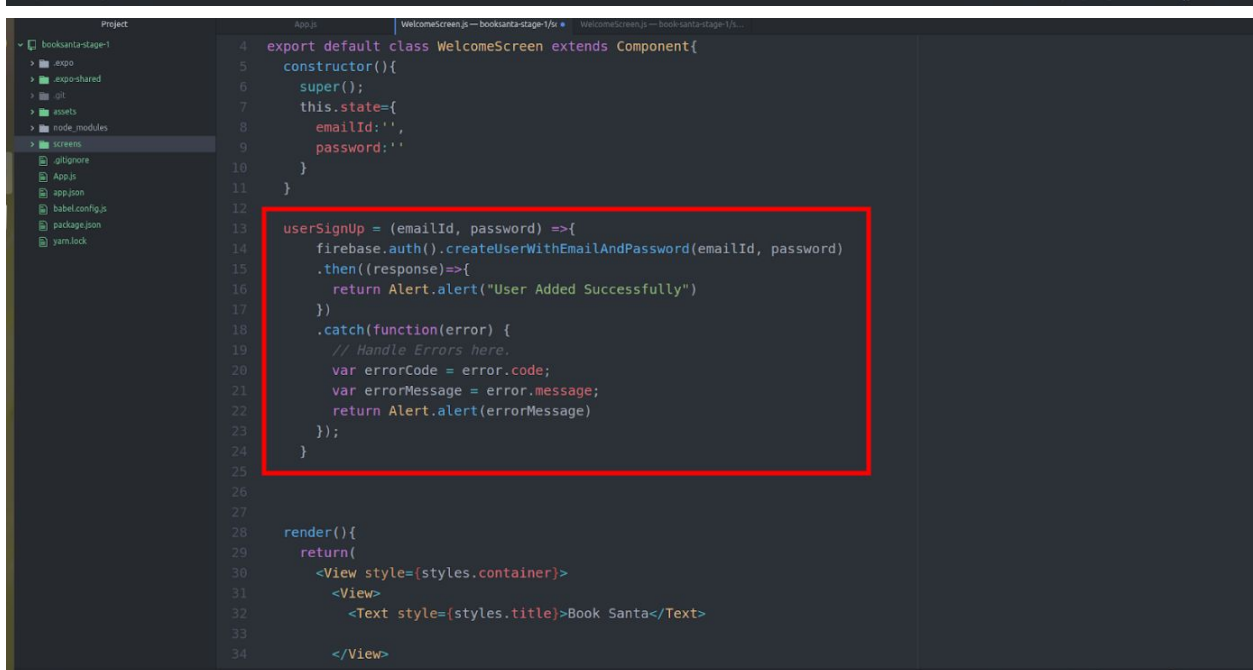


```

Project: booksanta-stage-1
App.js: WelcomeScreen.js — booksanta-stage-1/x... WelcomeScreen.js — book-santa-stage-1/x...

49 secureEntry = {true}
50 placeholder="enter Password"
51 onChangeText={(text)=>{
52   this.setState({
53     password: text
54   })
55 }}
56 </>
57 <TouchableOpacity
58   style={[styles.button,{marginBottom:20, marginTop:20}]}
59   onPress={()=>{this.userLogin(this.state.emailId, this.state.password)}}
60 >
61   <Text style={styles.buttonText}>Login</Text>
62 </TouchableOpacity>
63 |
64 <TouchableOpacity
65   style={styles.button}
66   onPress={()=>{this.userSignUp(this.state.emailId, this.state.password)}}
67 >
68   <Text style={styles.buttonText}>Sign Up</Text>
69 </TouchableOpacity>
70 </View>
71 </View>
72 )
73 }
74
75 const styles = StyleSheet.create({
76   container:{
77     flex:1,
78     backgroundColor:'#F8B85',
79     alignItems: 'center'

```



```

Project: booksanta-stage-1
App.js: WelcomeScreen.js — booksanta-stage-1/x... WelcomeScreen.js — book-santa-stage-1/x...

4 export default class WelcomeScreen extends Component{
5   constructor(){
6     super();
7     this.state={
8       emailId:'',
9       password:''
10    }
11  }
12
13  userSignUp = (emailId, password) =>{
14    firebase.auth().createUserWithEmailAndPassword(emailId, password)
15    .then((response)=>{
16      return Alert.alert("User Added Successfully")
17    })
18    .catch(function(error) {
19      // Handle Errors here.
20      var errorCode = error.code;
21      var errorMessage = error.message;
22      return Alert.alert(errorMessage)
23    });
24  }
25
26
27
28  render(){
29    return(
30      <View style={styles.container}>
31        <View>
32          <Text style={styles.title}>Book Santa</Text>
33        </View>
34      </View>

```

We create a userLogin function which accepts email and password as arguments. We call the function when the Login Button is pressed.



```

Project
  booksanta-stage-1
    expo
    expo-shared
    git
    assets
    node_modules
    screens
    .gitignore
    App.js
    app.json
    babel.config.js
    package.json
    yarn.lock


App.js
  45 //
  46 <TextInput
  47   style={styles.loginBox}
  48   secureTextEntry = {true}
  49   placeholder="enter Password"
  50   onChangeText={({text})=>{
  51     this.setState({
  52       password: text
  53     })
  54   }}
  55 >/>
  56 <TouchableOpacity
  57   style={{styles.button, {marginBottom:20, marginTop:20}}}
  58   onPress = {()=>{this.userLogin(this.state.emailId, this.state.password)}}
  59 >
  60   <Text style={styles.buttonText}>Login</Text>
  61 </TouchableOpacity>
  62 |
  63 <TouchableOpacity
  64   style={styles.button}
  65   onPress={()=>{this.userSignUp(this.state.emailId, this.state.password)}}
  66 >
  67   <Text style={styles.buttonText}>SignUp</Text>
  68 </TouchableOpacity>
  69 </View>
  70 </View>
  71 )
  72 }
  73 }
  74
  75 const styles = StyleSheet.create({

WelcomeScreen.js — booksanta-stage-1/js
  17 .then((response)=>{
  18   return Alert.alert("User Added Successfully")
  19 })
  20 .catch(function(error) {
  21   // Handle Errors here.
  22   var errorCode = error.code;
  23   var errorMessage = error.message;
  24   return Alert.alert(errorMessage)
  25 });
  26 }
  27
  28 userLogin = (emailId, password)=>{
  29   firebase.auth().signInWithEmailAndPassword(emailId, password)
  30   .then(()=>{
  31     return Alert.alert("Successfully Login")
  32   })
  33   .catch((error)=> {
  34     var errorCode = error.code;
  35     var errorMessage = error.message;
  36     return Alert.alert(errorMessage)
  37   })
  38 }
  39
  40
  41 render(){
  42   return(
  43     <View style={styles.container}>
  44       <View>
  45         <Text style={styles.title}>Book Santa</Text>
  46       </View>
  47

```

### Activity 3: Add lottie View

We add lottie to our app using `yarn add lottie-react-native /npm install lottie-react-native` command

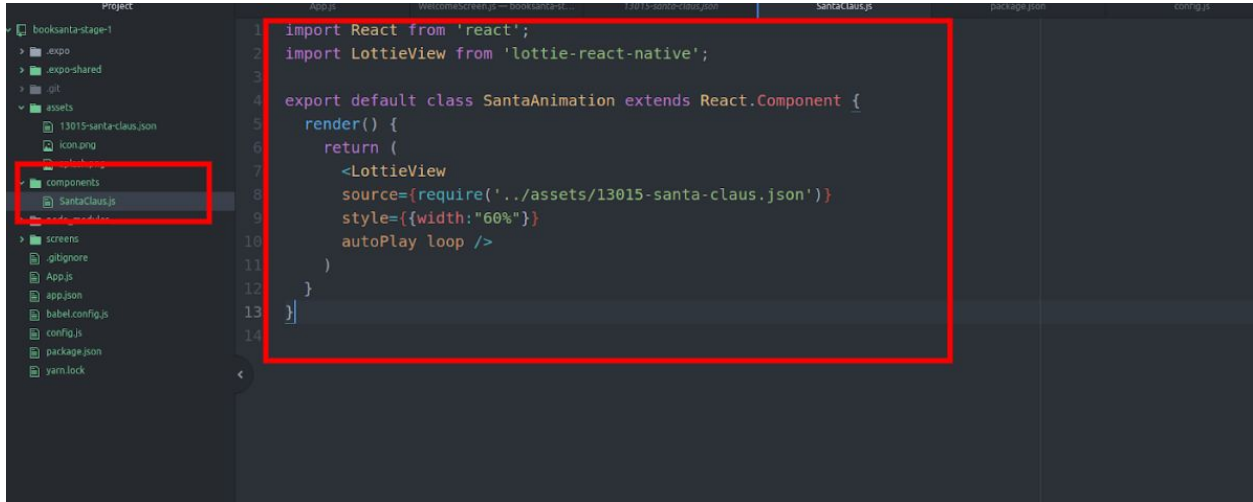


```
To run the app with live reloading, choose one of:
• Sign in as @vishalgaddam073 in Expo Client on Android or iOS. Your
  projects will automatically appear in the "Projects" tab.
• Scan the QR code above with the Expo app (Android) or the Camera app (iOS).
• Press a for Android emulator.
• Press e to send a link to your phone with email.

Expo Press ? to show a list of all available commands.
Logs for your project will appear below. Press Ctrl+C to exit.
Finished building JavaScript bundle in 48905ms.
Running application on vivo Y51L.
Finished building JavaScript bundle in 111ms.
Running application on vivo Y51L.

Stopping packager...
> Closing Expo server
> Stopping Metro bundler
Packager stopped.
vishu@oasis ~/D/booksanta-stage-1$ yarn add lottie-react-native
yarn add v1.21.1
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.11: The platform "linux" is incompatible with this module.
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 4 new dependencies.
info Direct dependencies
├─ lottie-react-native@3.3.2
info All dependencies
├─ dedent@0.6.0
├─ lottie-ios@3.1.6
├─ lottie-react-native@3.3.2
└─ react-native-safe-modules@1.0.0
Done in 10.32s.
vishu@oasis ~/D/booksanta-stage-1$
```


We write code to add lottie animation to app



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'assets', 'components', 'screens', and 'App.js'. The 'components' folder is expanded, and 'SantaClaus.js' is selected. The code editor shows the following code:

```
1 import React from 'react';
2 import LottieView from 'lottie-react-native';
3
4 export default class SantaAnimation extends React.Component {
5   render() {
6     return (
7       <LottieView
8         source={require('../assets/13015-santa-claus.json')}
9         style={{width:"60%"}}
10        autoPlay loop />
11     )
12   }
13 }
14 ]]
```

Run the code and test the app



# Book Santa

Login

SignUp

**What's NEXT?**

We will design the sign up form for the Book Santa App.

PRO-C77



EXTEND YOUR KNOWLEDGE