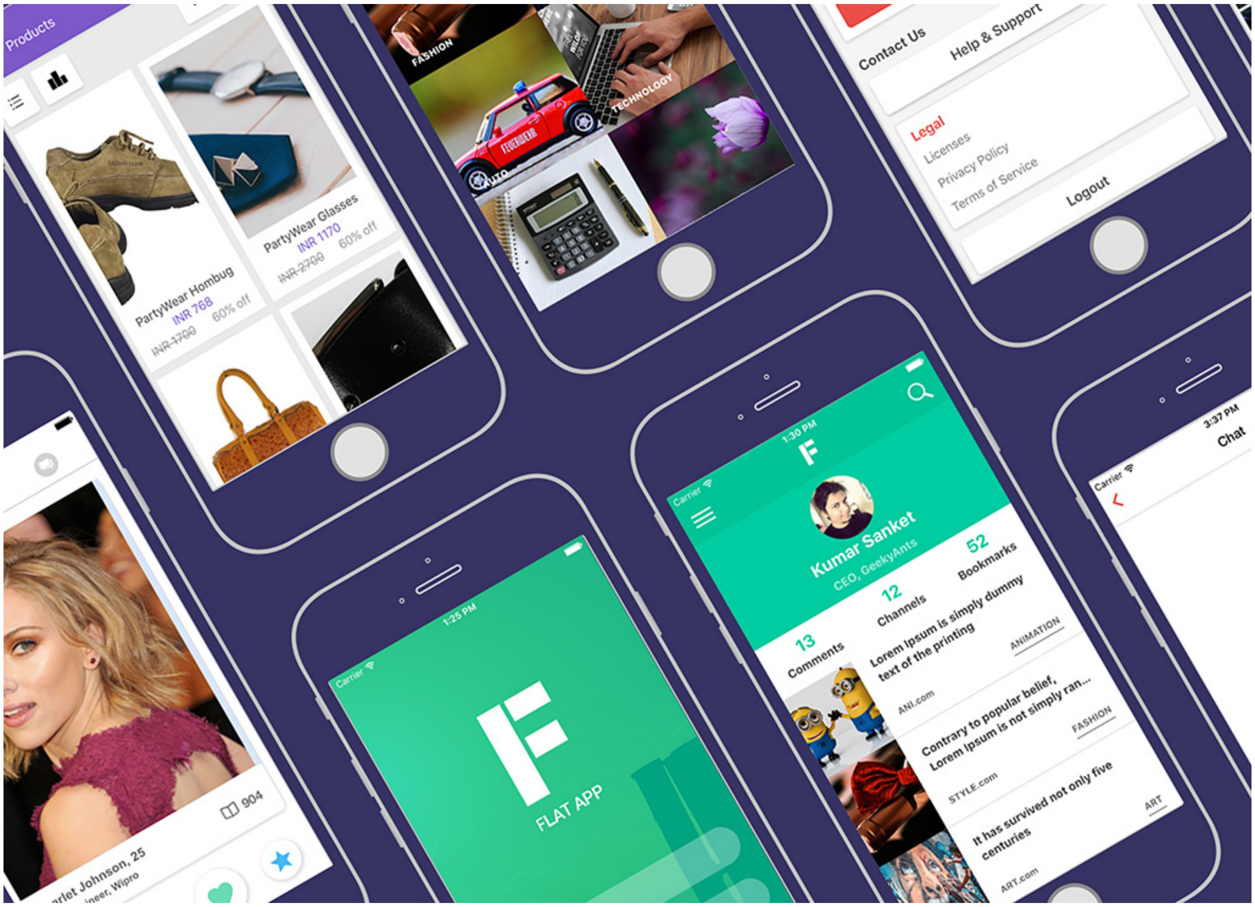
# Styling your App with NativeBase

Graphic design in an important part of App development. If you are not a graphic designer I recommend using NativeBase to style your App. NativeBase makes it easy to create a theme for your app.

Close Expo on your computer if it is already running. Make sure you are in your AwesomeProject folder and run the following commands to install Nativebase.

```
npm install native-base --save
npm install react-native-easy-grid --save
```



Test NativeBase by updating your App.js file with the following code and starting Expo.

```
import React from 'react';
import { AppLoading } from 'expo';
import { Container, Header, Content, Body, Left, Right, Text } from
'native-base';
import { Col, Row, Grid } from 'react-native-easy-grid';
import * as Font from 'expo-font';
import { Ionicons } from '@expo/vector-icons';

export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      isReady: false,
    };
  }

  async componentDidMount() {
    await Font.loadAsync({
      Roboto: require('native-base/Fonts/Roboto.ttf'),
      Roboto_medium: require('native-base/Fonts/Roboto_medium.ttf'),
      ...Ionicons.font,
    });
    this.setState({ isReady: true });
  }

  render() {
    if (!this.state.isReady) {
      return <AppLoading />;
    }

    return (


      <Container>
        <Header>
          <Left style={{backgroundColor: 'silver'}}>
            <Text style={{color: '#ffffff'}}>Left</Text>
          </Left>
          <Body>
            <Text style={{color: '#ffffff'}}>My Header</Text>
          </Body>
          <Right style={{backgroundColor: 'silver'}}>
            <Text style={{color: '#ffffff'}}>Right</Text>
          </Right>
        </Header>
        <Content>
          <Grid>
            <Col>
              <Text>First Column</Text>
            </Col>
            <Col>
              <Text>Second Column</Text>
            </Col>
          </Grid>
        </Content>
      </Container>

    );
  }
}
```

Now NativeBase isinstalled you can experiment with it’s Component Library here, https://docs.nativebase.io/

# Navigation with react-navigation

Understanding navigation in your app is really important. Thankfully there are modules available to help manage navigation in your app. Expo recommends the react-navigation module as it works well with Expo.

Create a new Expo project to test navigation.

```
expo init ExampleNavigation
```

Install the react-navigation module and supporting modules..

```
npm install @react-navigation/native --save

expo install react-native-gesture-handler react-native-reanimated
react-native-screens
expo install react-native-safe-area-context
@react-native-community/masked-view

npm install @react-navigation/stack  --save
```

Remember to add NativeBase to your new project

```
npm install native-base --save
npm install react-native-easy-grid --save
```

Test React Navigation by updating your App.js file with the following code and starting Expo.

```
import * as React from 'react';
import { View, Text } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import { Container, Button, Input, Item } from 'native-base';

function HomeScreen({ navigation }) {
  // this is React wizardry...it's called a state hook
  // text is a state and setText is a function to change text
  const [text, setText] = React.useState('');

  return (
    // In Button below react-navigation uses the onPress property
    // {() => navigation.navigate('Welcome', {myName: text})}
    // 'Welcome' is the screen referenced in line 52
    // {myName: text} allows you you to send data between pages
    // In this case the value of the text variable in line 11
    <Container style={{ flex: 1, alignItems: 'center',
justifyContent: 'center', padding: 20 }}>
      <Item regular>
        <Input
          placeholder='What is your name?'
          value={text} // text const from line 13
          onChangeText={text => setText(text)} // this is a shorthand
way of writing js functions
        />
      </Item>
      <Button onPress={() => navigation.navigate('Welcome', {myName:
text})} block bordered primary>
        <Text>Submit</Text>
      </Button>
    </Container>
  );
}

function WelcomeScreen({ route, navigation }) {

  const { myName } = route.params;

  return (
    <Container style={{ flex: 1, alignItems: 'center',
justifyContent: 'center' }}>
      <Text>Welcome {myName}</Text>
    </Container>
  );
}

// this last section always needs to be in your App.js file
const Stack = createStackNavigator();

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Welcome" component={WelcomeScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

There is more information about getting started with React Navigation here;

https://reactnavigation.org/docs/getting-started/