

```

1 // LAB 02
2 import { StatusBar } from 'expo-status-bar';
3 import React, { Component } from 'react';
4 import { Platform, StyleSheet, Text, View, Image } from 'react-native';
5
6 //Lab 02-1
7 export default class App extends Component {
8   render() {
9     return (
10       <View style={styles.container}>
11         <Text style={{fontSize: 50, fontWeight: 'bold'}}>Hello World!</Text>
12       </View>
13     );
14   }
15 }
16
17 //Lab 02-2
18 export default class App extends Component {
19   render() {
20     return (
21       <View style={styles.container}>
22         <View style={styles.row}>
23           <View style={styles.square_red}></View>
24           <View style={styles.square_green}></View>
25         </View>
26         <View style={styles.row}>
27           <View style={styles.square_blue}></View>
28           <View style={styles.square_pink}></View>
29         </View>
30       </View>
31     );
32   }
33 }
34
35 //Lab 02-3
36 export class ListItem extends Component{
37   render(){
38     let image = {
39       uri: 'https://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/React-
40 icon.svg/512px-React-icon.svg.png'
41     };
42     return(
43       <View style={styles.listContainer}>
44         <View>
45           <Image source={image} style={styles.image}/>
46         </View>
47         <View>
48           <Text style={styles.title}>{this.props.title}</Text>
49           <Text style={styles.text}>{this.props.description}</Text>
50         </View>
51       </View>
52     );
53   }
54
55 export default class App extends Component {
56   render() {
57     let lorem = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
58     tempor incididunt ut labore et dolore magna aliqua.";
59     return (
60       <View style={styles.container}>

```

```
60      <ListItem title="Item 01" description={lorem}></ListItem>
61      <ListItem title="Item 02" description={lorem}></ListItem>
62      <ListItem title="Item 03" description={lorem}></ListItem>
63    </View>
64  );
65}
66}
67
68const styles = StyleSheet.create({
69  container: {
70    backgroundColor: '#fff', flexDirection: 'column',
71    fontFamily: 'sans-serif', paddingTop: 20,
72  },
73  listContainer: {flexDirection: 'row', fontFamily: 'sans-serif',
74    paddingTop: 20},
75  row:{flexDirection: 'row'},
76  title:{fontSize: 20, fontWeight: 'bold'},
77  text:{fontSize: 16, flexWrap: 'wrap', width: '33%'},
78  image:{width: 100, height: 100, marginRight: 10},
79  square_red: {width: 50, height: 50, backgroundColor: 'red'},
80  square_blue: {width: 50, height: 50, backgroundColor: 'blue'},
81  square_green: {width: 50, height: 50, backgroundColor: 'green'},
82  square_pink: {width: 50, height: 50, backgroundColor: 'pink'}
83 },
84 });
85
```

```
1 // LAB 03-1 BMI
2 import { StatusBar } from 'expo-status-bar';
3 import React, { Component } from 'react';
4 import { Platform, StyleSheet, Text, View, TextInput, TouchableOpacity } from 'react-native';
5
6 export default class makingBMI extends Component {
7   constructor(props) {
8     super(props);
9     this.state = {
10       weight: '0', height: '0', bmi: '0.00',
11       bmiClass: '', textColor: 'black',
12     };
13   }
14   compute = () => {
15     console.log(this.state);
16     let weight = parseFloat(this.state.weight);
17     let height = parseFloat(this.state.height);
18     let BMI = (weight / Math.pow(height / 100, 2)).toFixed(2);
19     this.setState({ bmi: BMI });
20     if (BMI >= 30) {
21       this.setState({ bmiClass: 'Obese', textColor: 'red' });
22     } else if (BMI >= 25) {
23       this.setState({ bmiClass: 'Overweight', textColor: 'yellow' });
24     } else if (BMI >= 18.5) {
25       this.setState({ bmiClass: 'Normal', textColor: 'green' });
26     } else {
27       this.setState({ bmiClass: 'Underweight', textColor: 'orange' });
28     }
29   };
30   render() {
31     console.log(this.state);
32     return (
33       <View style={styles.container}>
34         <Text style={styles.appTitle}>BMI Calculator</Text>
35         <View style={styles.group}>
36           <Text style={styles.title}>Weight (KG)</Text>
37           <TextInput
38             keyboardType="numeric"
39             style={styles.input}
40             value={this.state.weight}
41             onChangeText={(weight) => this.setState({ weight })}
42           />
43         </View>
44         <View style={styles.group}>
45           <Text style={styles.title}>Height (CM) </Text>
46           <TextInput
47             keyboardType="numeric"
48             style={styles.input}
49             value={this.state.height}
50             onChangeText={(height) => this.setState({ height })}
51           />
52         </View>
53         <View style={styles.center}>
54           <View
55             style={[styles.group, {color: this.state.textColor}]}>
56             <Text style={[styles.title,{color: this.state.textColor}]}>
57               BMI: {this.state.bmi}' '
58             </Text>
59             <Text style={{ ...styles.title, color: this.state.textColor }}>
60               {this.state.bmiClass}' '
61           </View>
62         </View>
63       </View>
64     );
65   }
66 }
```

```
61         </Text>
62     </View>
63     <View style={styles.group}>
64         <TouchableOpacity style={styles.button} onPress={this.compute}>
65             <Text style={styles.buttonText}>Compute</Text>
66         </TouchableOpacity>
67     </View>
68     <StatusBar style="auto" />
69 </View>
70 );
71 }
72 }
73 }
74
75 const styles = StyleSheet.create({
76     container: {flex: 1, justifyContent: 'center',
77         flexDirection: 'column', padding: 20,
78     },
79     group: {marginTop: 20},
80     button: { backgroundColor: 'lightblue', padding: 20, borderWidth: 1},
81     buttonText: {fontSize: 20},
82     input: {padding: 10, height: 40, borderWidth: 1},
83     appTitle: {fontSize: 20, fontWeight: 'bold', textAlign: 'center'},
84     title: {fontSize: 18,fontWeight: 'semibold'},
85     center: {alignItems: 'center', textAlign: 'center'},
86 });
87
```

```

1 // LAB 03-2 Stopwatch
2 import React, { Component } from 'react';
3 import {AppRegistry, StyleSheet, Text, View,
4   TextInput, TouchableHighlight } from 'react-native';
5 import formatTime from 'minutes-seconds-milliseconds';
6
7 export class LapTime extends Component {
8   render() {
9     return (
10       <View key={this.props.index} style={styles.lap}>
11         <Text style={styles.lapText}>Lap #{this.props.index + 1}</Text>
12         <Text style={styles.lapText}>{formatTime(this.props.time)}</Text>
13       </View>
14     );
15   }
16 }
17
18 export default class Stopwatch extends Component {
19   constructor(props) {
20     super(props);
21     this.state = {
22       timeElapsed: null, //Difference between the current time and the startTime
23       running: false, //Does the clock is ticking?
24       startTime: null, //Record the startTime, when user press start.
25       laps: [], //Array from lap records
26     };
27     this.handleStartPress = this.handleStartPress.bind(this);
28     this.startStopButton = this.startStopButton.bind(this);
29     this.handleLapPress = this.handleLapPress.bind(this);
30   }
31
32   laps() {
33     let numOflaps = this.state.laps.length;
34     return this.state.laps.map(function (time, index) {
35       if(numOflaps - index <= 5){
36         return (
37           <LapTime index={index} time={time}/>
38         );
39       });
40     }
41
42   startStopButton() {
43     var style = this.state.running ? styles.stopButton : styles.startButton;
44
45     return (
46       <TouchableHighlight
47         underlayColor="gray"
48         onPress={this.handleStartPress}
49         style={[styles.button, style]}>
50         <Text>{this.state.running ? 'Stop' : 'Start'}</Text>
51       </TouchableHighlight>
52     );
53   }
54
55   lapButton() {
56     return (
57       <TouchableHighlight
58         style={styles.button}
59         underlayColor="gray"
60         onPress={this.handleLapPress}>
61         <Text>{this.state.running ? 'Lap' : 'Reset'}</Text>
62       </TouchableHighlight>
63     );
64   }
65
66   handleLapPress() {
67     if (!this.state.running) {
68       this.setState({laps: [], startTime: null, timeElapsed: null});

```

```

69     return;
70   }
71   var lap = this.state.timeElapsed;
72   this.setState({
73     startTime: new Date(),
74     laps: this.state.laps.concat([lap]),
75   });
76 }
77
78 handleStartPress() {
79   if (this.state.running) {
80     clearInterval(this.interval);
81     this.setState({ running: false });
82     return;
83   }
84   this.setState({ startTime: new Date() });
85   this.interval = setInterval(() => {
86     this.setState({
87       timeElapsed: new Date() - this.state.startTime,
88       running: true,
89     });
90   }, 30);
91 }
92
93 // if the open View tag is on the same line as return, don't need to wrap in ()
94 // but if it's on a new line, need to wrap everything in () because
95 // JS will insert ; right after return
96 render() {
97   return (
98     <View style={styles.container}>
99       <View style={styles.header}>
100         <View style={styles.timerWrapper}>
101           <Text style={styles.timer}>
102             {formatTime(this.state.timeElapsed)}
103           </Text>
104         </View>
105         <View style={styles.buttonWrapper}>
106           {this.lapButton()}
107           {this.startStopButton()}
108         </View>
109       </View>
110       <View style={styles.footer}>{this.laps()}</View>
111     </View>
112   );
113 }
114 }
115
116 const styles = StyleSheet.create({
117   container: {
118     flex: 1, // Fill the entire screen
119     margin: 20,
120   },
121   header: {flex: 1},
122   footer: {flex: 1},
123   timerWrapper: {flex: 5, justifyContent: 'center', alignItems: 'center'},
124   buttonWrapper: {flex: 3, flexDirection: 'row',
125     justifyContent: 'space-around', alignItems: 'center'},
126   lap: {justifyContent: 'space-around', flexDirection: 'row',
127     backgroundColor: 'lightgray', padding: 10, marginTop: 10,},
128   button: {borderWidth: 2, height: 100, width: 100, borderRadius: 50,
129     justifyContent: 'center', alignItems: 'center'},
130   timer: {fontSize: 60 },
131   lapText: {fontSize: 30 },
132   startButton: {borderColor: 'green' },
133   stopButton: {borderColor: 'red' },
134 });
135
136 AppRegistry.registerComponent('simpleApps', () => Stopwatch);
137

```

```

1 // LAB-04 APP.JS
2
3 import React, { Component } from 'react';
4 import { StyleSheet, Text, View, Dimensions, StatusBar } from 'react-native';
5 import { LineChart } from 'react-native-chart-kit';
6 import StockButton from './StockButton';
7 import API from './api';
8 import Switch from './switch';
9
10 const chartConfig = {
11   backgroundGradientFrom: '#1E2923',
12   backgroundGradientTo: '#08130D',
13   color: (opacity = 1) => `rgba(26, 255, 146, ${opacity})`, // color of background
14   strokeWidth: 2 // optional, default 3
15 }
16
17 export default class Stocks extends Component{
18   constructor(props){
19     super(props);
20     this.changeIndex = this.changeIndex.bind(this);
21     this.handleSwitchChange = this.handleSwitchChange.bind(this);
22     this.state = {
23       stockCode: 'Not Chosen Yet',
24       stockName: 'N/A',
25       dates: ["01/01", "02/01", "03/01", "04/01", "05/01", "06/01", "07/01", ],
26       prices: [1,2,3,4,5,6,7],
27       switch: false
28     };
29   }
30
31   changeIndex(stockCode,stockName){
32     console.log("Change Index is called");
33     let stockMode = (this.state.switch) ? "Weekly" : "Daily";
34     console.log(stockCode,stockName,stockMode);
35     API(stockCode,stockMode).then((stock) => {
36       let datesArray = [];
37       if(stockMode == "Daily"){
38         datesArray = Object.keys(stock["Time Series (Daily)"]).slice(0,7);
39       }else{
40         datesArray = Object.keys(stock["Weekly Time Series"]).slice(0,7);
41       }
42       let closingPrice = [];
43       datesArray.forEach((day) => {
44         if(stockMode == "Daily"){
45           closingPrice.push(stock["Time Series (Daily)"][day]["4. close"]);
46         }else{
47           closingPrice.push(stock["Weekly Time Series"][day]["4. close"]);
48         }
49       })
50       let datesArrayRev = datesArray.reverse();
51       let dayMonthArray = [];
52       datesArrayRev.forEach((element) => {
53         let dayArray = element.split('-');
54         dayMonthArray.push(dayArray[2]+'/'+dayArray[1]);
55       });
56       this.setState({
57         stockCode: stockCode,
58         stockName: stockName,
59         dates: dayMonthArray,
60         prices: closingPrice,
61       });
62       console.log("Finished change index");
63     }).catch(function(error){
64       console.log(error);
65     });
66   }
67
68   handleSwitchChange(){

```

```

69     this.setState({
70       switch : !this.state.switch
71     });
72   }
73
74   render(){
75     return(
76       <View style={styles.container}>
77         <View style={styles.header}>
78           <Text style={styles.title}>Company: {this.state.stockCode} / Mode: {(this.state.switch) ? "Weekly" : "Daily"}</Text>
79           <LineChart
80             , data={{
81               , labels: this.state.dates,
82               , datasets: [
83                 , data: this.state.prices,
84                 , color: (opacity = 1) => `rgba(134, 65, 244, ${opacity})`, // color of the line
85                 , strokeWidth: 2 // optional
86               ]
87             }
88             width={Dimensions.get('window').width}
89             height={400}
90             chartConfig={chartConfig}
91             style={{paddingVertical:10}}
92           />
93         </View>
94         <View style={[styles.rowContainer,styles.footer]}>
95           <View style={[styles.rowContainer]}>
96             <Switch onValueChange={this.handleSwitchChange} value={this.state.switch}></Switch>
97           </View>
98           <View style={styles.rowContainer}>
99             <StockButton code="AAPL" name="Apple" onPress={this.changeIndex}></StockButton>
100            <StockButton code="GOOGL" name="Google" onPress={this.changeIndex}></StockButton>
101            <StockButton code="UBER" name="Uber" onPress={this.changeIndex}></StockButton>
102          </View>
103        </View>
104      </View>
105    );
106  }
107}
108
109 const styles = StyleSheet.create({
110   container: { flex: 1, paddingTop: 40 },
111   title:{ flex: 2, marginTop: 15, fontSize: 24,
112     color: 'white', fontWeight: 'bold' },
113   header:{ flex: 2, justifyContent: 'center',
114     alignItems: 'center', backgroundColor: 'black' },
115   rowContainer:{ flexDirection: 'row',
116     justifyContent:'space-between'
117   },
118   columnContainer:{ flexDirection: 'column', justifyContent: 'space-between'
119   },
120   footer:{ flex: 1, flexDirection: 'row',
121     flexWrap: 'wrap', alignItems:'center',
122     justifyContent: 'space-evenly', backgroundColor: 'gray'
123   },
124   button:{ margin: 10, borderWidth: 1, width: 100,
125     height: 50, borderRadius: 10, justifyContent: 'center',
126     alignItems: 'center', backgroundColor: 'lightgray'
127   }
128 });
129
130
131
132 });
133

```

```

1 // LAB 04 SWITCH.JS
2 // minimum working example for Switch
3 import React, { Component } from 'react';
4 import { Platform, StyleSheet, Text, View, Switch } from 'react-native';
5
6 export default class App extends Component {
7
8     constructor(props) {
9         super(props);
10        this.state = {switch : false}
11    }
12
13    render() {
14        let message
15        if (this.props.value) {message = 'Weekly';}
16        else {message = 'Daily';}
17        return (
18            <View style={styles.container}>
19                <Switch onValueChange={this.props.onValueChange}>
20                    , value={this.props.value}/>
21                    <Text style={{fontSize:20}}>" "+message</Text>
22                </View>
23        );
24    }
25 }
26
27 const styles = StyleSheet.create({
28     container: {
29         flex: 1, flexDirection: 'row',
30         justifyContent: 'center', alignItems: 'center',
31         backgroundColor: 'white', padding: 10
32     }
33 });
34
35 // LAB 04 STOCKBUTTON.JS
36 import React, {Component} from 'react';
37 import { StyleSheet, Text, TouchableOpacity } from 'react-native';
38
39 export default class StockButton extends Component{
40     render(){
41         return(
42             <TouchableOpacity style={styles.button}
43             , onPress={()=>this.props.onPress(this.props.code,this.props.name)}>
44                 <Text>{this.props.code}</Text>
45             </TouchableOpacity>
46         );
47     }
48 }
49
50 const styles = StyleSheet.create({
51     button:{
52         margin: 10,
53         height: 50,
54         width: 100,
55         borderWidth: 1,
56         borderRadius: 10,
57         alignItems: 'center',
58         justifyContent: 'center',
59         backgroundColor: 'lightgray'
60     }
61 });
62

```

```

1 // LAB 05 APP.JS
2 import React from 'react';
3 import { Button, View, Text, TouchableHighlight,
4   StyleSheet, FlatList, Image, TextInput } from 'react-native';
5 import { NavigationContainer } from '@react-navigation/native';
6 import { createStackNavigator } from '@react-navigation/stack';
7 import Toast from 'react-native-toast-message';
8 import API from './api';
9 const IMAGE_NOT_FOUND = require('./image_not_found.png');
10 const NUM_MOVIES = 5;
11
12 class ListScreen extends React.Component {
13   constructor(){
14     super();
15     this.loadMovie = this.loadMovie.bind(this);
16     this.state = {
17       listData: Array(5).fill(null),
18       movieSearch: 'Ocean'
19     };
20     this.loadMovie('Ocean');
21   }
22
23   loadMovie(title){
24     console.log("Search using ",title);
25     let listData = Array(5).fill(null);
26     API(title).then(
27       (data) => {
28         let results = data.results;
29         let movies = results.slice(0,NUM_MOVIES);
30         if(movies.length > 0){
31           movies.forEach((element,index) => {
32             listData[index] = element;
33           });
34         }
35         this.setState({listData: listData})
36       }
37     ).catch((error) => {
38       Toast.show({
39         type: 'error',
40         text1: 'Error while searching your movie',
41         autoHide: false,
42         position: 'bottom'
43       });
44     });
45   }
46
47   handleButtonPress = () => {
48     let query = this.state.movieSearch;
49     this.loadMovie(query);
50   }
51
52   render() {
53     let data = [];
54     const listData = this.state.listData;
55     console.log(listData);
56     console.log(listData[0] !== null);
57     if(listData[0] !== null){
58       data = [];
59       listData.forEach(
60         (list) => {
61           if(list !== null){
62             data.push({
63               key: list.title,
64               imgSource: {uri: 'https://image.tmdb.org/t/p/w200/' + list.poster_path},
65               vote_average: list.vote_average,
66               overview: list.overview,
67               release_date: list.release_date
68             })
69         }
70       );
71     }
72     return (
73       <View style={styles.container}>
74         <Text>{this.state.movieSearch}</Text>
75         <FlatList
76           data={listData}
77           renderItem={({item}) =>
78             <Image source={{uri: item.imgSource}} style={styles.image} />
79             <Text>{item.key}</Text>
80             <Text>{item.vote_average}</Text>
81             <Text>{item.overview}</Text>
82             <Text>{item.release_date}</Text>
83           }
84         </FlatList>
85       </View>
86     );
87   }
88 }

```

```

69         }
70     }
71   );
72 }else{
73   console.log('Toast must shown');
74   Toast.show({
75     type: 'error',
76     text1: 'Not found the movie you are looking for',
77     text2: 'Please try again using different query',
78     autoHide: false,
79     position: 'bottom',
80     keyboardOffset: 20
81   });
82 }
83
84 return (
85   <View style={{padding: 20}}>
86     <View style={{marginBottom: 10}}>
87       <TextInput keyboardType='default' style={styles.input} value={this.state.movieSearch}
88       , onChangeText={({movieSearch) => {this.setState({movieSearch})}}}></TextInput>
89       <Button title='Search' color={'#333'} onPress={this.handleButtonPress}></Button>
90     </View>
91     <Text style={{fontSize:24, fontWeight: '600'}}>
92       Movie List
93     </Text>
94     <FlatList
95     , data={data}
96     , style={{marginTop: 5}}
97     , renderItem = { ({item}) => {
98       return (
99         <TouchableHighlight
100        , onPress={() => {
101          this.props.navigation.navigate('Details',item)
102        }}>
103         <View style={styles.row}>
104           <Image style={styles.image} source={item.imgSource}/>
105           <Text style={styles.title}>{item.key}</Text>
106         </View>
107         <TouchableHighlight>
108           );
109         });
110       />
111     );
112   );
113 }
114 }
115
116 class DetailsScreen extends React.Component {
117   render() {
118     const movieDetail = this.props.route.params;
119     return (
120       <View style={{padding: 20, flexDirection: 'column'}}>
121         <Text style={{fontSize: 24, marginBottom: 10, fontWeight: 'bold'}}>{movieDetail.key}</Text>
122         <View style={{flexDirection:'row'}}>
123           <View style={{flex: 2}>
124             <Image style={{height: 200}} source={movieDetail.imgSource}/>
125           </View>
126           <View style={{flex:3, padding:10}}>
127             <Text style={styles.text}>"Released on: "+movieDetail.release_date</Text>
128             <View style={{height:1, backgroundColor:'lightgray', margin:5}}>
129               <Text style={styles.text}>"Rating: "+movieDetail.vote_average</Text>
130             <View style={{height:1, backgroundColor:'lightgray', margin:5}}>
131               <Text style={styles.text}>"Plot: "+movieDetail.overview</Text>
132             </View>
133           </View>
134         </View>
135       );
136     }
137   }

```

```
138 |
139 const Stack = createStackNavigator();
140
141 export default class App extends React.Component {
142   render() {
143     return (
144       <NavigationContainer>
145         <Stack.Navigator initialRouteName="Home">
146           <Stack.Screen
147             name="Home"
148             component={ListScreen}
149             options={{
150               title: 'Movie Explorer',
151               headerStyle: {
152                 backgroundColor: 'darkred',
153               },
154               headerTintColor: '#fff',
155               headerTitleStyle: {
156                 fontWeight: 'bold',
157               },
158             }}
159           />
160           <Stack.Screen
161             name="Details"
162             component={DetailsScreen}
163             options={{
164               title: 'Movie Explorer',
165               headerStyle: {
166                 backgroundColor: 'darkred',
167               },
168               headerTintColor: '#fff',
169               headerTitleStyle: {
170                 fontWeight: 'bold',
171               },
172             }}
173           />
174         </Stack.Navigator>
175         <Toast/>
176       </NavigationContainer>
177     );
178   }
179 }
180
181 const styles = StyleSheet.create({
182   container: {flex: 1, justifyContent: 'center',
183     flexDirection: 'column', padding: 20},
184   row: {flexDirection: 'row', height: 110, flex: 1,
185     borderBottomWidth: 1, borderBottomColor: '#ddd', marginVertical: 5},
186   image: {height: 110, flex: 2},
187   title: {fontSize: 18, flex: 5, padding: 15},
188   subtitle: {fontSize: 14, flex: 5, padding: 15, color: 'gray'},
189   text: {fontSize: 16},
190   input: {padding: 10, height: 50, borderWidth: 1, borderColor: '#666'}
191 });
```

```

1 // LAB 06 APP.JS
2 // LOCATION AND MAPS IS HERE
3 import React, { Component } from 'react';
4 import { Platform, StyleSheet, Text, View, Dimensions, Button, Image} from 'react-native';
5 import Constants from 'expo-constants';
6 import * as Location from 'expo-location';
7 import MapView from 'react-native-maps';
8 let {width: SCREEN_WIDTH, height: SCREEN_HEIGHT} = Dimensions.get('window');
9 export default class App extends Component {
10   constructor(props) {
11     super(props);
12     this.state = {
13       location: null,
14       errorMessage: null,
15       currentLocationRegion: null,
16       region: {
17         latitude: 13.764884, longitude: 100.538265,
18         latitudeDelta: 0.005, longitudeDelta: 0.005,
19       },
20       markers:[
21         {
22           latlng: {latitude: 13.764884, longitude: 100.538265},
23           title: "Victory Monument",
24           description: "A large military monument in Bangkok, Thailand.",
25           photo: require('./images/Victory_Monument.jpg')
26         },
27         {
28           latlng: {latitude: 13.763681, longitude: 100.538125},
29           title: "Saxophone Club",
30           description: "A music pub for saxophone lover",
31           photo: require('./images/Saxophone.jpg')
32         },
33         {
34           latlng: {latitude: 13.764595, longitude: 100.537438},
35           title: "Coco Department Store",
36           description: "A fashion department store",
37           photo: require('./images/coco.jpg')
38         }
39       ],
40       pin: false
41     };
42     this.onRegionChangeComplete = this.onRegionChangeComplete.bind(this);
43     this.moveMaptoLocation = this.moveMaptoLocation.bind(this);
44   }
45
46   onRegionChangeComplete(region) { this.setState({region});}
47
48   moveMaptoLocation(region) {
49     this.refs.map.animateToRegion(region, 1000);
50   }
51
52   handleMarkerButtonPress(region, index){
53     this.moveMaptoLocation({...region, latitudeDelta: 0.005,longitudeDelta: 0.005});
54     this.refs["marker"+index].showCallout();
55   }
56
57   handleCurrentLocationButtonPress(){
58     if(!this.state.pin){
59       this.moveMaptoLocation(this.state.currentLocationRegion);
60     }
61     this.setState({pin: !this.state.pin});
62   }
63
64   componentDidMount() {
65     if (Platform.OS === 'android' && !Constants.isDevice) {
66       this.setState({
67         errorMessage: `Oops, this will not work on Sketch in an Android emulator.
68         Try it on your device!`,

```

```

69     });
70   } else { this._getLocationAsync(); }
71 }
72
73 async _getLocationAsync() {
74   let { status } = await Location.requestForegroundPermissionsAsync();
75   if (status !== 'granted') {
76     this.setState({
77       errorMessage: 'Permission to access location was denied',
78     });
79   }
80   let location = await Location.getCurrentPositionAsync({});
81   const currentLocationRegion = {
82     latitude: location.coords.latitude,
83     longitude: location.coords.longitude,
84     latitudeDelta: 0.005, longitudeDelta: 0.005,
85   };
86   this.setState({location: location, currentLocationRegion: currentLocationRegion});
87 }
88
89 helperText(){
90   let text = 'Waiting for the current location...';
91   if (this.state.errorMessage) {
92     text = this.state.errorMessage;
93   }
94   else if (this.state.location) {
95     text = 'Current Location is available';
96   }
97   return text;
98 }
99
100 render() {
101   return (
102     <View style={styles.mapContainer}>
103       <MapView style={styles.map}
104         , region={this.state.region}
105         , onRegionChangeComplete={this.onRegionChangeComplete} ref="map">
106         {this.state.markers.map((marker,i) =>
107           return(
108             <MapView.Marker
109               , props={{key:i}} coordinate={marker.latlng}
110               , title={marker.title} description={marker.description} ref={"marker"+i}>
111               <MapView.Callout>
112                 <View style={styles.callout}>
113                   <Image style={styles.calloutPhoto} source={marker.photo}/>
114                   <Text style={styles.calloutTitle}>{marker.title}</Text>
115                   <Text>{marker.description}</Text>
116                 </View>
117               </MapView.Callout>
118             </MapView.Marker>));
119         )})
120         { this.state.pin ?
121           <MapView.Marker
122             , coordinate={{
123               latitude: this.state.location.coords.latitude,
124               longitude: this.state.location.coords.longitude
125             }}
126             , title="Your Current Location"
127             ,
128             >
129             <MapView.Callout>
130               <View style={styles.callout}>
131                 <Text style={styles.calloutTitle}>Your Current Location</Text>
132                 <Text>Latitude: {this.state.location.coords.latitude}{'\n'}
133                 Longitude: {this.state.location.coords.longitude}</Text>
134               </View>
135             </MapView.Callout>
136           </MapView.Marker>: null
137         }
138     </MapView>

```

```
138     <View style={styles.container}>
139       <View style={{padding: 5}}>
140         {
141           this.state.markers.map((marker,i) => {
142             return(
143               <Button
144                 key={i} style={styles.button}
145                 onPress={() => this.handleMarkerButtonPress(marker.latlng, i)}
146                 title={marker.title} color="#66A500"
147                 />
148             );
149           })
150         }
151     </View>
152     <View style={{padding: 5, flexDirection: 'column',
153       justifyContent: 'center', alignItems: 'center'}}>
154       <Text>{this.helperText()}</Text>
155       <Button disabled={!this.state.location}
156         title={this.state.pin ? 'Hide Marker At The Current Location' :
157           'Show Marker At The Current Location'}
158         onPress={() => this.handleCurrentLocationButtonPress()}
159         />
160       </View>
161     </View>
162   </View>
163 );
164 }
165 }
166
167 const styles = StyleSheet.create({
168   mapContainer: { flex: 1, flexDirection: 'column',
169     justifyContent: 'flex-end', alignItems: 'center',
170     backgroundColor: '#F5FCFF', paddingTop: Constants.statusBarHeight },
171   button:{ padding: 5 },
172   container: { flex: 1, justifyContent: 'center',
173     alignItems: 'center', backgroundColor: '#F5FCFF' },
174   map: { width: SCREEN_WIDTH, height: Math.floor(SCREEN_HEIGHT*2/3) },
175   callout:{ flex: 1, paddingRight: 10,
176     paddingBottom: 10, marginRight: 10, marginBottom: 10 },
177   calloutPhoto:{ flex: 1, width: 166, height: 83 },
178   calloutTitle:{ fontSize: 16, fontWeight: 'bold' }
179 });
180 }
```

```

1 // LAbB07 APP.JS ANIMATION
2
3 import React, { Component } from 'react';
4 import { Platform, StyleSheet, Text, View, Dimensions, Button, Image} from 'react-native';
5 import { GestureHandlerRootView,
6   TapGestureHandler, PanGestureHandler } from 'react-native-gesture-handler';
7 import Animated, { useSharedValue, useAnimatedStyle,
8   useAnimatedGestureHandler, withSpring} from 'react-native-reanimated';
9 let {width: SCREEN_WIDTH, height: SCREEN_HEIGHT} = Dimensions.get('window');
10 const EventsExample = () => {
11   const pressed = useSharedValue(false);
12   const startingPosition = 0;
13   const x = useSharedValue(startingPosition);
14   const y = useSharedValue(startingPosition);
15   const eventHandler = useAnimatedGestureHandler({
16     onStart: (event, ctx) => {
17       pressed.value = true; ctx.startX = x.value; ctx.startY = y.value;
18     },
19     onActive: (event, ctx) => {
20       x.value = ctx.startX + event.translationX;
21       y.value = ctx.startY + event.translationY;
22     },
23     onEnd: (event, ctx) => {
24       pressed.value = false;
25       let corner_threshold = 150;
26       let snapCornerX = x.value;
27       let snapCornerY = y.value;
28       let xLeft = (((0-SCREEN_WIDTH)/2)+corner_threshold);
29       let xRight = (SCREEN_WIDTH/2)-corner_threshold;
30       let yTop = (SCREEN_HEIGHT/2)-corner_threshold;
31       let yBottom = (((0-SCREEN_HEIGHT)/2)+corner_threshold);
32       if (x.value < xLeft && y.value > yTop){ // top left corner
33         snapCornerX = (((0-SCREEN_WIDTH)/2)+25);
34         snapCornerY = ((SCREEN_HEIGHT/2)-25);
35       } else if (x.value > xRight && y.value > yTop){ // top right corner
36         snapCornerX = ((SCREEN_WIDTH/2)-25);
37         snapCornerY = ((SCREEN_HEIGHT/2)-25);
38       } else if (x.value > xRight && y.value < yBottom){ // bottom right corner
39         snapCornerX = ((SCREEN_WIDTH/2)-25);
40         snapCornerY = (((0-SCREEN_HEIGHT)/2)+25);
41       } else if (x.value < xLeft && y.value < yBottom){ // bottom left corner
42         snapCornerX = (((0-SCREEN_WIDTH)/2)+25);
43         snapCornerY = (((0-SCREEN_HEIGHT)/2)+25);
44       }
45       x.value = withSpring(snapCornerX); y.value = withSpring(snapCornerY);
46     },
47   });
48   const uas = useAnimatedStyle(() => {
49     return {
50       backgroundColor: pressed.value ? '#FEEF86' : '#001972',
51       transform: [{ translateX: x.value }, { translateY: y.value }],
52     };
53   });
54   return (
55     <PanGestureHandler onGestureEvent={eventHandler}>
56       <Animated.Image source={{uri:
57 ,           'image here'}} style={[styles.ball, uas]} />
58     </PanGestureHandler>
59   );
60 };
61
62 export default function App() {
63   return (
64     <GestureHandlerRootView style={styles.container}>
65       <EventsExample/>
66     </GestureHandlerRootView>
67   );
68 }
69
70 const styles = StyleSheet.create({
71   container: { flex: 1, justifyContent: 'center', alignItems: 'center'},
72   ball: { width: 100, height: 100, borderRadius: 100, alignSelf: 'center'},
73 });

```