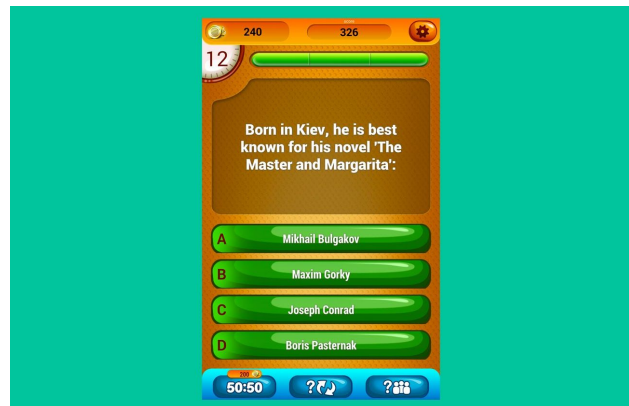


## Quiz Master App

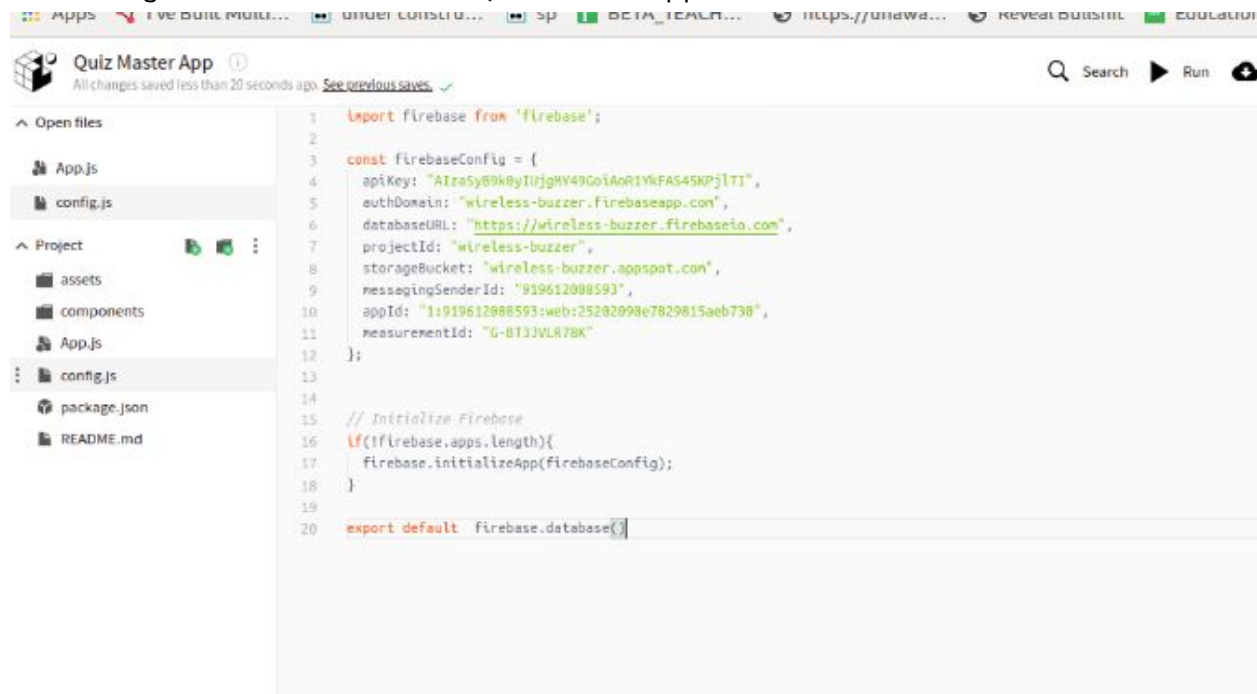


### What we did:

- Used the sort method and compare function to sort an array in ascending order.
- Used the map method to display the team names on the app.
- Created a reset button to reset the database to its default state.

### How we did it:

Connecting the database with the Quiz Buzzer App:



Quiz Master App ⓘ  
All changes saved less than 10 seconds ago. [See previous saves.](#) ✓

Open files

- App.js
- config.js

Project

- assets
- components
- App.js
- config.js
- package.json
- README.md

```

1  import React, { Component } from 'react';
2  import { Text, View, StyleSheet, Button } from 'react-native';
3  import db from './config';
4
5  export default class App extends Component {
6    render() {
7      return (
8        <View style={{ flex: 1 }}>
9
10       </View>
11     );
12   }
13 }
14

```

We created a state in our App class Component which will hold the names of the teams which have pressed the buzzer button in an array.

Search ▶ Ri

```

1  import React, { Component } from 'react';
2  import { Text, View, StyleSheet, Button } from 'react-native';
3  import db from './config';
4
5  export default class App extends Component {
6    constructor() {
7      super();
8      this.state = {
9        teamsRank: [],
10      };
11    }
12    render() {
13      return <View style={{ flex: 1 }} />;
14    }
15  }
16

```

The structure of our database. Every team had two keys - 'isButtonPressed' and 'timestamp'.

We will use 'isButtonPressed' to identify if the team has pressed the button.

Teacher shows the database structure to the student.

Let's call this function 'showTeamRanks()' and let us define it inside our app class.

Teacher writes an empty function showTeamRanks().



```
1 import React, { Component } from 'react';
2 import { Text, View, StyleSheet, Button } from 'react-native';
3 import db from './config';
4
5 export default class App extends Component {
6   constructor() {
7     super();
8     this.state = {
9       teamsRank: [],
10     };
11   }
12
13   showTeamsRank = ()=>{
14
15   }
16
17   render() {
18     return <View style={{ flex: 1 }} />;
19   }
20 }
21
```

Read the value stored inside teams from our database.

- We need to get a database reference first.
- We need to create a listener which triggers a callback() function whenever the function is triggered.

```

1  import React, { Component } from 'react';
2  import { Text, View, StyleSheet, Button } from 'react-native';
3  import db from './config';
4
5  export default class App extends Component {
6    constructor() {
7      super();
8      this.state = {
9        teamsRank: [],
10      };
11    }
12
13    showTeamRanks = ()=>{
14      var teamRef = db.ref('teams/');
15    }
16
17    render() {
18      return <View style={{ flex: 1 }} />;
19    }
20  }
21

```


(30) [See previous saves.](#) ✓

```

1  import React, { Component } from 'react';
2  import { Text, View, StyleSheet, Button } from 'react-native';
3  import db from './config';
4
5  export default class App extends Component {
6    constructor() {
7      super();
8      this.state = {
9        teamsRank: [],
10      };
11    }
12
13    showTeamRanks = ()=>{
14      var teamRef = db.ref('teams/');
15      teamRef.on("value", (data)=>{
16
17      });
18    }
19
20    render() {
21      return <View style={{ flex: 1 }} />;
22    }
23  }
24

```

We called the 'showTeamRanks' function inside the 'componentDidMount' so that it is called when the app loads.



```
1 import React, { Component } from 'react';
2 import { Text, View, StyleSheet, Button } from 'react-native';
3 import db from './config';
4
5 export default class App extends Component {
6   constructor() {
7     super();
8     this.state = {
9       teamsRank: [],
10     };
11   }
12
13   showTeamRanks = () => {
14     var teamRef = db.ref('teams/');
15     teamRef.on("value", (data) => {
16       var teamList = data.val();
17       console.log(teamList);
18     });
19   }
20
21   componentDidMount() {
22     this.showTeamRanks();
23   }
24 }
```

ERRORS LOGS

Android SDK built for x86: \* { blue: { \_ }, green: { \_ }, red: { \_ }, yellow: { \_ } }

Android SDK built for x86: \* { blue: { \_ }, green: { \_ }, red: { \_ }, yellow: { \_ } }

We changed the database directly or through the Quiz Buzzer App to see the output change in the console.

Quiz Master App ⓘ  
 All changes saved half a minute ago. [See previous saves](#) ✓

Search ▶ Run

Open files

- App.js
- config.js
- project
- assets
- components
- App.js
- config.js
- package.json
- README.md

```

1  import React, { Component } from 'react';
2  import { Text, View, StyleSheet, Button } from 'react-native';
3  import db from './config';
4
5  export default class App extends Component {
6    constructor() {
7      super();
8      this.state = {
9        teamsRank: [],
10      };
11    }
12
13    showTeamRanks = () => {
14      var teams = [];
15      var teamRef = db.ref('teams/');
16      teamRef.on("value", (data) => {
17        var teamList = data.val();
18        for(var team in teamList){
19          if(teamList[team]["isButtonPressed"] === "true"){
20            teams.push(teamList[team]);
21          }
22        }
23      });
24    }
25
26    componentDidMount(){
27      this.showTeamRanks();
28    }
29
30    render() {
31      return <View style={{ flex: 1 }} />;
32    }
33  }
  
```




To sort the team's array according to their timestamp, there is a function which can help us sort it. 'array.sort()' can sort an array according to the rule we define. It takes a comparison function as an argument. It runs the comparison function repeatedly over the elements of the array until the array is completely sorted.


Last saved 10 minutes ago. [See previous saves.](#)

```

1 import React, { Component } from 'react';
2 import { Text, View, StyleSheet, Button } from 'react-native';
3 import db from './config';
4
5 export default class App extends Component {
6   constructor() {
7     super();
8     this.state = {
9       teamsRank: [],
10     };
11   }
12
13   showTeamRanks = ()=>{
14     var teams = [];
15     var teamRef = db.ref('teams/');
16     teamRef.on("value", (data)=>{
17       var teamList = data.val();
18       for(var team in teamList){
19         if(teamList[team]["isButtonPressed"] === "true"){
20           teams.push(teamList[team]);
21         }
22       }
23       teams.sort(function(team1,team2){
24         return team1.timestamp - team2.timestamp
25       });
26     });
27   }
28
29   componentDidMount(){
30     this.showTeamRanks();
31   }
32
33   render() {
  
```

We wrote the code to update the state.

Quiz Master App   Search  Run

All changes saved less than 5 seconds ago. [See previous saves.](#) 

```

1 import React, { Component } from 'react';
2 import { Text, View, StyleSheet, Button } from 'react-native';
3 import db from './config';
4
5 export default class App extends Component {
6   constructor() {
7     super();
8     this.state = {
9       teamsRank: [],
10     };
11   }
12
13   showTeamRanks = ()=>{
14     var teams = [];
15     var teamRef = db.ref('teams/');
16     teamRef.on("value", (data)=>{
17       var teamList = data.val();
18       for(var team in teamList){
19         if(teamList[team]["isButtonPressed"] === "true"){
20           teams.push(teamList[team]);
21         }
22       }
23       teams.sort(function(team1,team2){
24         return team1.timestamp - team2.timestamp
25       });
26       this.setState({teamsRank: teams});
27     });
28   }
29
30   componentDidMount(){
31     this.showTeamRanks();
32   }
33
34   render() {
  
```

We console logged the teams.

```

1 import React, { Component } from 'react';
2 import { Text, View, StyleSheet, Button } from 'react-native';
3 import db from './config';
4
5 export default class App extends Component {
6   constructor() {
7     super();
8     this.state = {
9       teamsRank: [],
10     };
11   }
12
13   showTeamRanks = () => {
14     var teams = [];
15     var teamRef = db.ref('teams/');
16     teamRef.on('value', data => {
17       var teamList = data.val();
18       for (var team in teamList) {
19         if (teamList[team]['isButtonPressed'] === true) {
20           teamList[team]['teamName'] = team;
21           teams.push(teamList[team]);
22         }
23       }
24       console.log(teams);
25       this.setState({ teamsRank: teams });
26     });
27   };
28
29   componentDidMount() {
30     this.showTeamRanks();
31   }
32
33   render() {

```

We used the '.map()' function to iterate through the state array.

See previous saves ✓

```

16 teamRef.on('value', data => {
17   var teamList = data.val();
18   for (var team in teamList) {
19     if (teamList[team]['isButtonPressed'] === true) {
20       teamList[team]['teamName'] = team;
21       teams.push(teamList[team]);
22     }
23   }
24   console.log(teams);
25   this.setState({ teamsRank: teams });
26 });
27 };
28
29 componentDidMount() {
30   this.showTeamRanks();
31 }
32
33 render() {
34   return (
35     <View style={{flex:1}}>
36       <View>
37         {this.state.teamsRank.map((team) => (
38           <View>
39             <Text>{team.teamName.toUpperCase()}</Text>
40           </View>
41         ))}
42       </View>
43     </View>
44   );
45 }
46 }
47

```



We added some inline style or we can use stylesheets.

```

26     });
27   };
28
29   componentDidMount() {
30     this.showTeamRanks();
31   }
32
33   render() {
34     return (
35       <View style={{flex:1}}>
36         <View
37           style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
38           {this.state.teamsRank.map((team) => (
39             <View
40               style={{
41                 width: 140,
42                 height: 55,
43                 borderWidth: 2,
44                 margin: 5,
45                 justifyContent: 'center',
46                 alignItems: 'center',
47                 backgroundColor: team.teamName,
48               }}>
49             <Text>{team.teamName.toUpperCase()}</Text>
50           </View>
51         ))}
52       </View>
53     </View>
54   );
55 }
56 }

```

We completed the app by adding the header and a reset button.

```

27   };
28
29   componentDidMount() {
30     this.showTeamRanks();
31   }
32
33   render() {
34     return (
35       <View style={{flex:1}}>
36         <View
37           style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
38           {this.state.teamsRank.map((team) => (
39             <View
40               style={{
41                 width: 140,
42                 height: 55,
43                 borderWidth: 2,
44                 margin: 5,
45                 justifyContent: 'center',
46                 alignItems: 'center',
47                 backgroundColor: team.teamName,
48               }}>
49             <Text>{team.teamName.toUpperCase()}</Text>
50           </View>
51         ))}
52       </View>
53       <Button
54         title="RESET"
55         style={{ width: 100, height: 100 }}
56         onPress={this.resetDb}
57       />
58     </View>

```

iliz Master App ⓘ

Changes saved. [See previous saves.](#) ✓

Search 🔍 Set

```

20
21
22
23
24
25
26
27   };
28
29   componentDidMount() {
30     this.showTeamRanks();
31   }
32
33   resetDb = () => {
34     var restDatabase = db.ref('teams/').set({
35       red: {
36         isButtonPressed: false,
37         timestamp: 0,
38       },
39       green: {
40         isButtonPressed: false,
41         timestamp: 0,
42       },
43       blue: {
44         isButtonPressed: false,
45         timestamp: 0,
46       },
47       yellow: {
48         isButtonPressed: false,
49         timestamp: 0,
50       },
51     });
52     this.setState({ teamDetails: [] });
53   };
54
55   render() {
56     return (
57       <View style={{flex:1}}>
58       <View
  
```

expo.io...

### What's next?:

In the next class, you will learn how to fix a few minor bugs which might have crept in and also learn how to make 'apk' or 'ios' files for installing the app on your phone. f