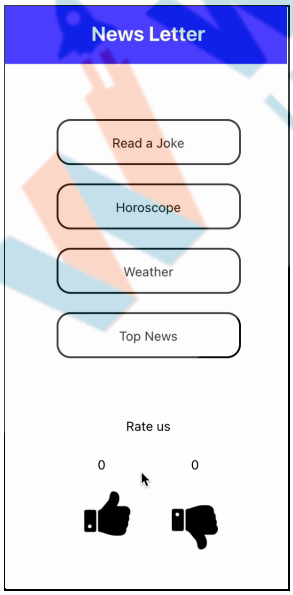
Topic	Quiz Master App	
Class Description	**Students build a Quiz Master Admin App which displays the names of the teams in the order in which they press the buttons. Students learn about sort and map methods in javascript defined over arrays.**	
Class	**C60**	
Class time	**45 mins**	
Goal	• Use sort method and compare function to sort an array in ascending order. • Use map method to display the team names on the app. • Create a reset button to reset the database to its default state.	
Resources Required	• Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Snack Account • Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Snack Account	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

- **WARM-UP SESSION - 5 mins**

Teacher starts slideshow  **from slides 1 to 14**

Refer to speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
Hi, how have you been? Are you excited to learn something new? ***Run the presentation from slide 1 to slide 5.*** **The following are the warm-up session deliverables:** • Reconnect with previous class topics. • Warm-Up quiz session.	**ESR**: Varied Response. Click on the slide show tab and present the slides.
QnA Session	
Question	**Answer**
Which of the following options will set the state of object “like” by incrementing the current like by 1. 	**B**

A.
```
this.setState =
  like : this.state.like + 1
```

B.
```
this.setState({
  like : this.state.like + 1
});
```

C.
```
this.setState({
  this.state.like + 1
});
```

D.
```
this.setState(
  like = this.state.like + 1
);
```

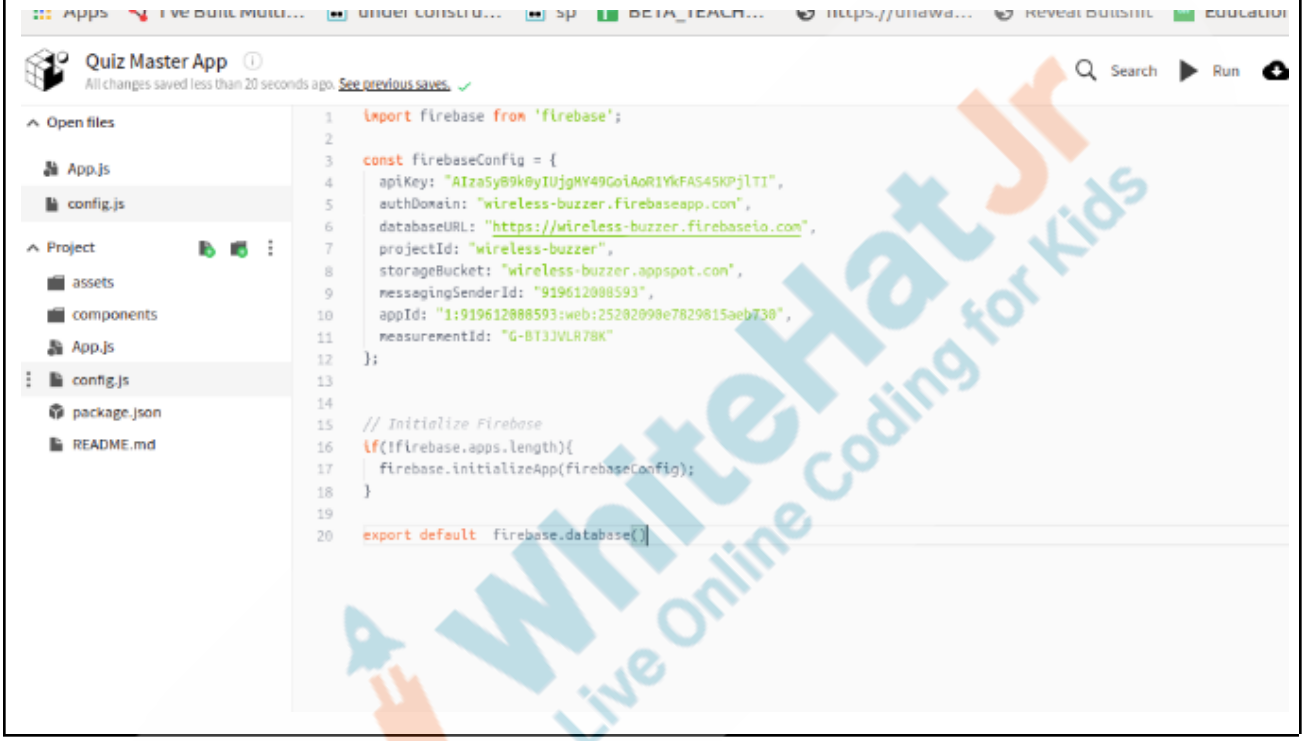
Which of the following options will set the state of object “dislike” by incrementing the current dislike by 1. A. `this.setState({ dislike: this.state.dislike - 1 });` B. `this.setState({ dislike = 1 });` C. `this.setState({ dislike: -1 });` D. `this.setState({ dislike: this.state.dislike + 1 });`	**D**

Continue the warm-up session

<table>
<tr><th colspan="2">Activity details</th><th>Solution/Guidelines</th></tr>
<tr><td colspan="2">Run the presentation from slide 6 to slide 14 to set the problem statement.

The following are the warm-up session deliverables:
• Review code from the previous class.
• Quiz master App functionality and pseudo-code for the app.</td><td></td></tr>
<tr><td colspan="3">Teacher ends slideshow  </td></tr>
<tr><td colspan="3">TEACHER-LED ACTIVITY - 15 mins</td></tr>
<tr><td colspan="3">Teacher Initiates Screen Share</td></tr>
<tr><td colspan="3"><u>CHALLENGE</u>
• Use sort method to arrange the teams in the order in which they pressed the buzzer.
• Use map method to display the team names on the app.</td></tr>
<tr><td>Step 2:
Teacher-led Activity
(15 min)</td><td>Our new App called - Quiz Master App will read from our database we had created earlier for the Quiz Buzzer. It will then display the order in which the Buzzer buttons were pressed by the team.
How do we need to connect to our database?</td><td>ESR:
We need the config keys.</td></tr>
<tr><td></td><td>Guide me on how to create a config.js file where we can initialize our firebase and export ‘firebase.database()’.</td><td>The student guides the teacher on how to create the config.js file.</td></tr>
</table>

	Student opens **Teacher Activity 1** Check if the student remembers and understands how any value can be exported from a file.	



	Awesome! Now our Quiz master app is going to be a single screen App. It will simply display the names of the team in the order in which they pressed the buttons. It will also contain a reset button which will reset the team fields to default values so that the teams can play another round.	The student listens.
	Let us create a state in our App class Component which will hold the names of the teams which have pressed the buzzer button in an array.	The student guides the teacher on how to create ‘state’ for the App class component.

	Initially, the array will be empty when the app loads. Later it will get the team names from the database. Can you guide me on how to create a state for the app component? Check if the student remembers the use of constructor(),super(), initializing state etc.	
 Search ``` import React, { Component } from 'react'; import { Text, View, StyleSheet, Button } from 'react-native'; import db from './config'; export default class App extends Component { constructor() { super(); this.state = { teamsRank: [], }; } render() { return <View style={{ flex: 1 }} />; } } ```		
	Good job! Now we want a function which will read all the teams who have pressed the buttons from the database and arrange them according to the timestamp. Remember the structure of our	The student observes.

	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().	

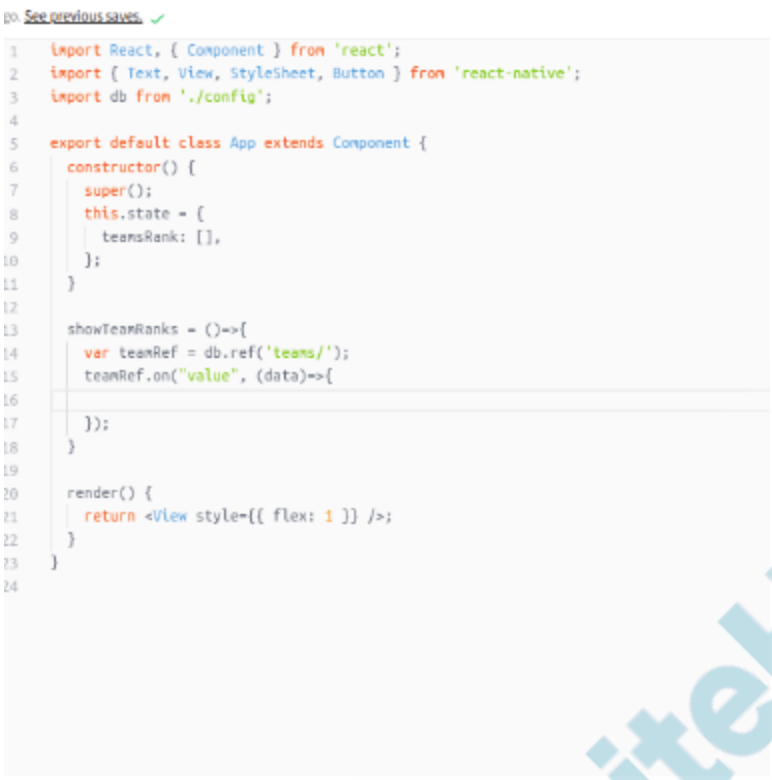
	Let's try to read the value stored inside teams from our database. Do you remember how we can do that?	ESR: - We need to get a database reference first. - We need to create a listener which triggers a callback() function whenever the function is triggered.
	Help me do that.	The student helps the teacher create a database reference for the teams and listener which triggers a callback function when any value in the database is changed.

```
import React, { Component } from 'react';
import { Text, View, StyleSheet, Button } from 'react-native';
import db from './config';

export default class App extends Component {
  constructor() {
    super();
    this.state = {
      teamsRank: [],
    };
  }

  showTeamRanks = ()=>{
    var teamRef = db.ref('teams/');
  }

  render() {
    return <View style={{ flex: 1 }} />;
  }
}
```



```
import React, { Component } from 'react';
import { Text, View, StyleSheet, Button } from 'react-native';
import db from './config';

export default class App extends Component {
  constructor() {
    super();
    this.state = {
      teamsRank: [],
    };
  }

  showTeamRanks = ()=>{
    var teamRef = db.ref('teams/');
    teamRef.on("value", (data)=>{

    });
  }

  render() {
    return <View style={{ flex: 1 }} />;
  }
}
```

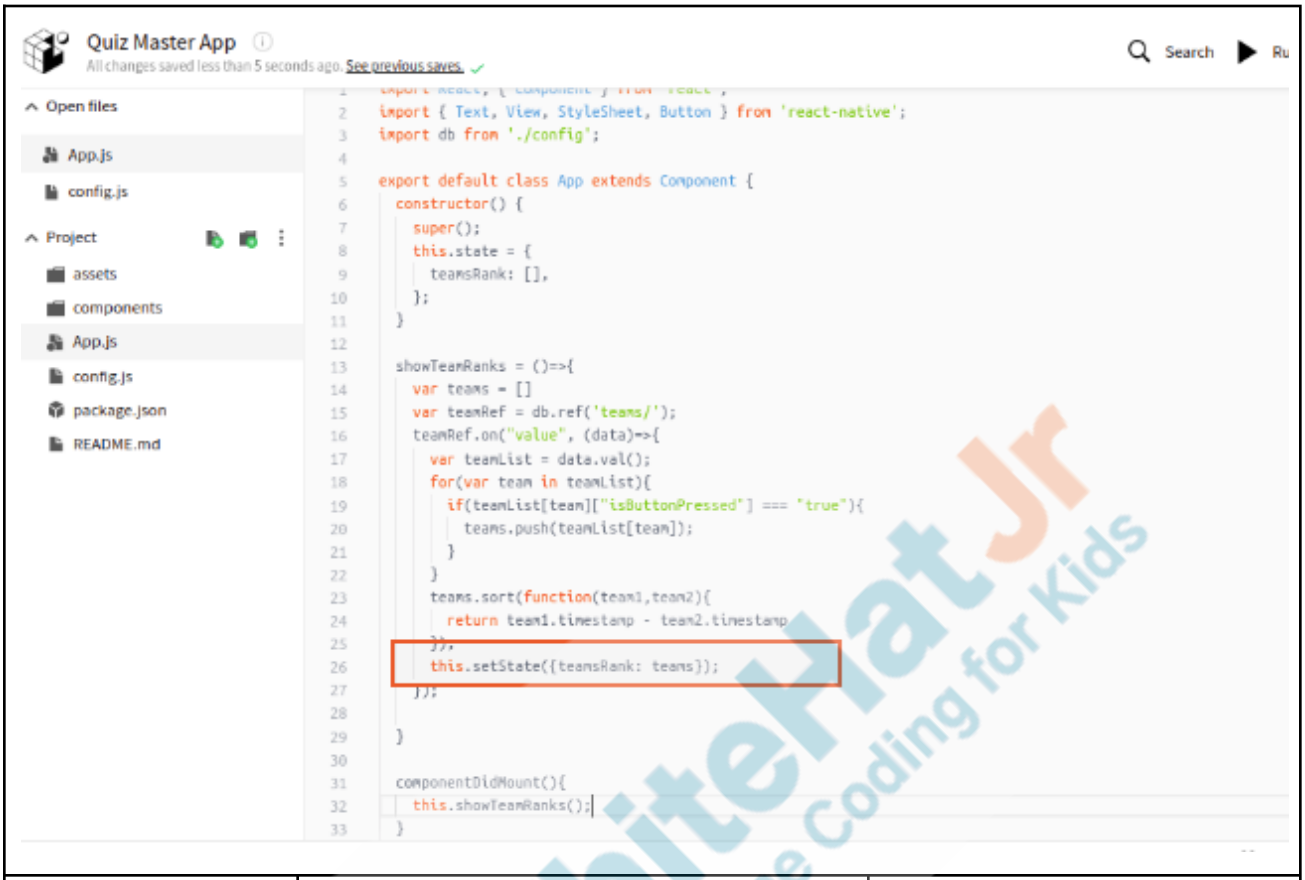
	For now, let's simply store the data we are getting from the database inside a variable called 'teamList' and let's try to console log it. We will need to call the 'showTeamRanks' function somewhere so that it is called when the app loads. Where can we call it? Teacher calls the function inside the 'componentDidMount' and shows the output inside the console.	ESR: Inside 'componentDidMount()' function which is called when the app component has mounted.

	You can see that the output is an object containing the list of teams and their keys - ‘isButtonPressed’ and ‘timestamp’. You can change the database directly or through the Quiz Buzzer App to see the output change in the console. Teacher shows the change in the output when the buzzer button is pressed from the previous app.	The student observes the change in the output when the buzzer button is pressed.
	Now, we want to loop over all the teams inside the teamList and check if any of the teams have ‘isButtonPressed’ to 'true. Teacher writes the code to loop over the ‘teamList’ object	The student observes the code and asks questions.

	Clearly explain the for(var team in teamList) loop. The for loop runs over each object. ‘team’ is the key inside the teamList and they represent blue, red, green and yellow. Each team has "isButtonPressed" and "timeStamp" property. **Edit: true is a boolean value and SHOULD NOT be inside quotes.**	
		
	Now, what do we want to do if the button is pressed for a team? Great! Let’s create an array called	ESR: We want to push the team in an array to be sorted by their timestamp.

	teams and push the teams which have pressed the buzzer inside them. Teacher writes the code.	
	We now need to sort the teams array according to their timestamp. There is a function which can help us sort the teams array. 'array.sort()' can sort any 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. Let me show you how. Teacher writes the sort() function and explains. Each two teams in the array are compared using team1.timestamp - team2.timestamp. If the result is greater than 0, the larger of teams is pushed at the end of the array. If this is done repeatedly, the array gets sorted in ascending order.	The student understands how sort() function is used and asks questions to the teacher.

	We have done well so far. Now, we have an updated sorted list of teams in the array called teams. We just have to update the state. Do you remember how to update the state of the component?	ESR: Using ‘this.setState()’.
	Yes! Let's update the state then. Teacher writes the code to update the state.	The student helps the teacher.



	Let's quickly console log the teams. Teacher console logs the teams and presses the buzzer using the Quiz Buzzer App. As you can see the array 'teams' is sorted by timeStamps. However, the team names (keys) are missing. Only the values are present. We can fix this by creating a 'teamName key' and pushing it in the array 'teams'. Teacher console logs again to show the output.	The student observes and asks questions.

```
import { Text, View, StyleSheet, Button } from 'react-native';
import db from './config';

export default class App extends Component {
  constructor() {
    super();
    this.state = {
      teamsRank: [],
    };
  }

  showTeamRanks = () => {
    var teams = [];
    var teamRef = db.ref('teams/');
    teamRef.on('value', data => {
      var teamList = data.val();
      for (var team in teamList) {
        if (teamList[team]["isButtonPressed"] === true) {
          teamList[team]["teamName"] = team;
          teams.push(teamList[team]);
        }
      }
      console.log(teams);
      this.setState({ teamsRank: teams });
    });
  };

  componentDidMount() {
    this.showTeamRanks();
  }

  render() {
```

Prettier {}

	Alright, now we want to render the team names using the 'teams' in the App state - 'teamsRank'. Where can we render the team names?	ESR: Inside render() function in app.
	Ideally we would like to loop through all the elements inside 'teamsRank' and display each team name inside text. We can do that using the 'map()' function. 'map()' function can loop through each element in an array. It takes a	The student understands how to use map() to iterate through an array.

	function which can render JSX tags for each element of the array. Isn't that amazing! Teacher shows how to use the '.map()' function to iterate through the state array.	
		
	We can add some inline style or we can use stylesheets.	The student helps in styling the view.

```
});
};

componentDidMount() {
  this.showTeamRanks();
}

render() {
  return (
    <View style={{flex:1}}>
     <View
     style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
       {this.state.teamsRank.map((team) => (
         <View
           style={{
             width: 140,
             height: 55,
             borderWidth: 2,
             margin: 5,
             justifyContent: 'center',
             alignItems: 'center',
             backgroundColor: team.teamName,
           }}>
           <Text>{team.teamName.toUpperCase()}</Text>
         </View>
       ))}
      </View>
    </View>
  );
 }
}
```

	Now, can you complete the app by adding the header and a reset button.	Yes.
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

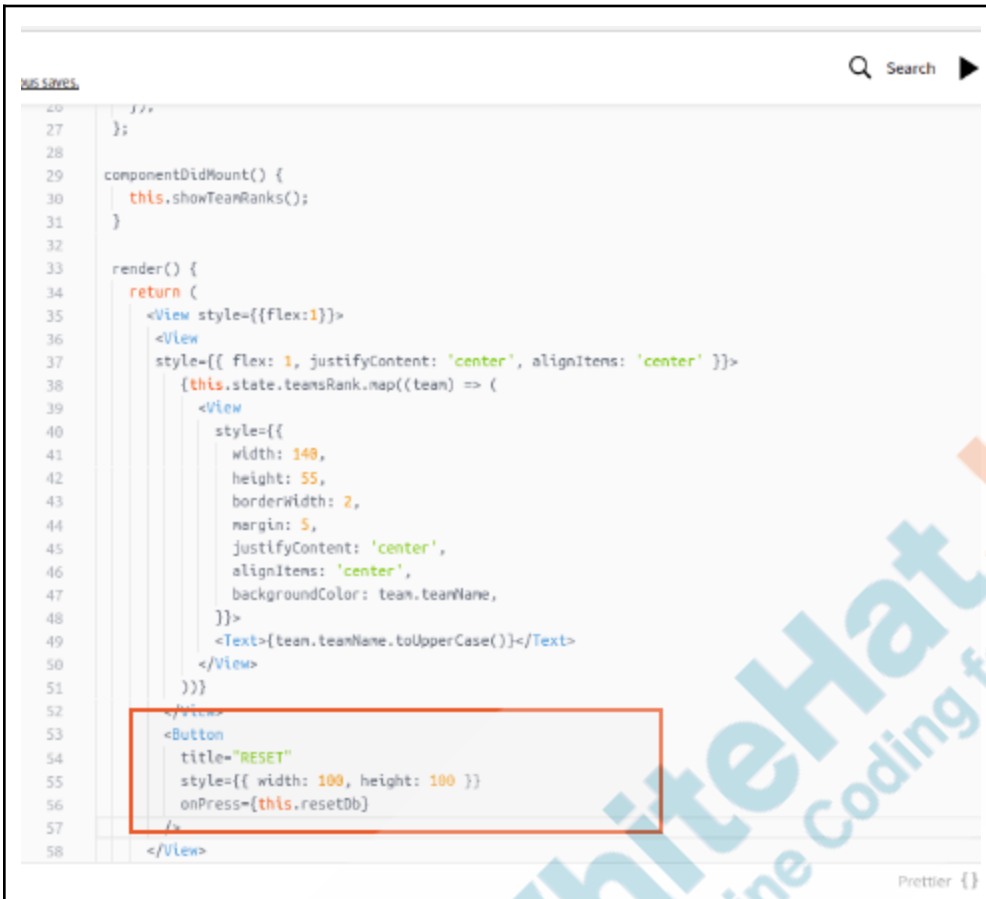
- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

ACTIVITY

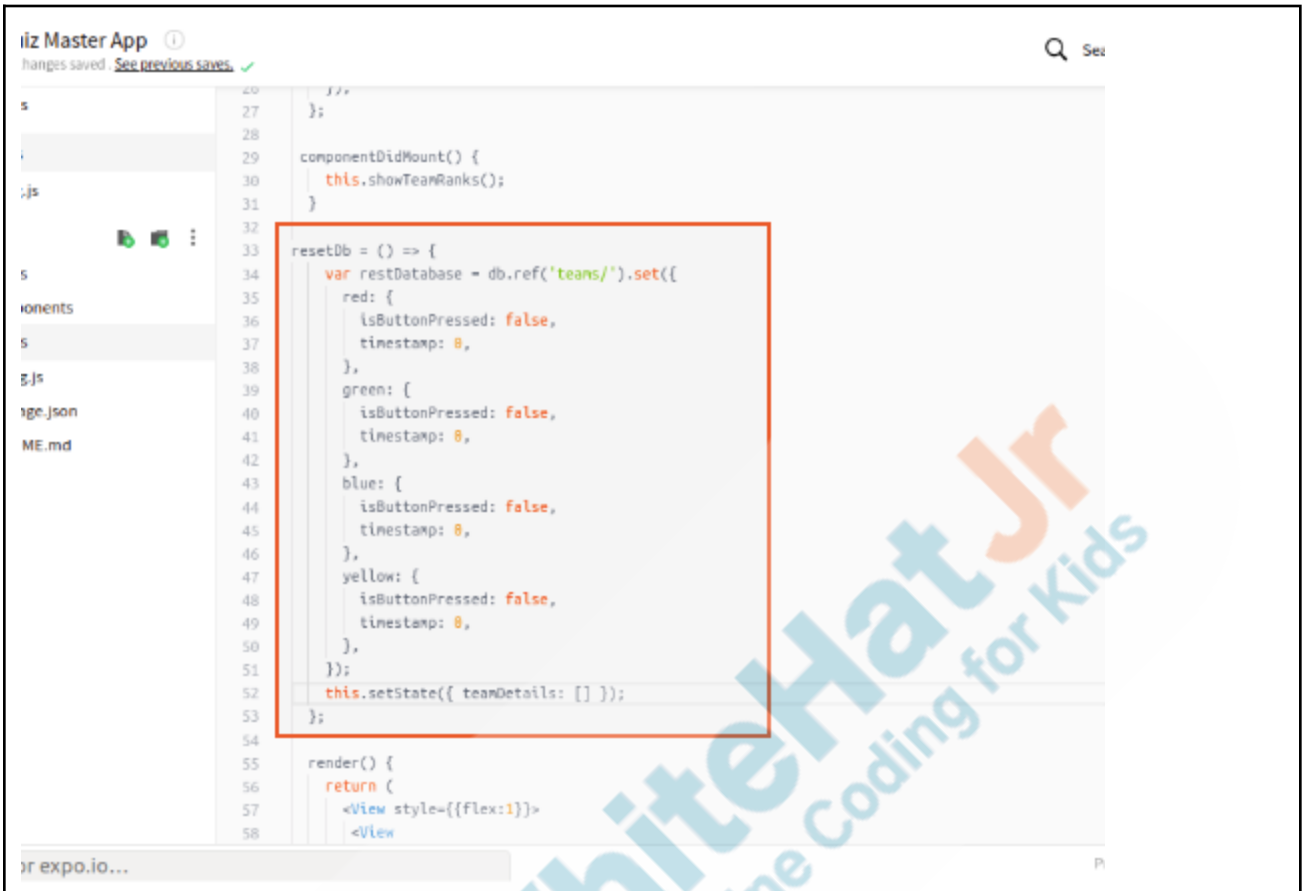
- **Create a reset button to reset the database to its default state.**

Teacher starts slideshow  **for slide 15 and 16.**

	Now it's your turn. Please share your screen with me.	
Teacher ends slideshow 		
Step 3: Student-Led Activity (15 min)	Guide the student to create the new Quiz Admin App.	The student creates the new Quiz Admin app.
	Guide the student to - - create 'teamsRank' State. - create 'showTeamsRank' function where we sort the teams according to their timestamp. - update the 'teamsRank' state - call the 'showTeamsRank' function in an array.	The student creates the 'showTeamsRank' function, sorts the teams and updates the state.
	Guide the student to render the teams name using .map function() for teams array	The student renders the team names on the app.
	Guide the student to create a reset button.	The student creates the reset button on the screen.



	Guide the student to create a ‘resetdb()’ function where the state of the App and database is reset.	The student creates the ‘resetdb()’ function where the App and database is reset.



Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 5 Mins

Teacher starts slideshow **from slide 17 to slide 27**

Activity details	Solution/Guidelines
Run the presentation from slide 17 to slide 27 **Following are the wrap-up session deliverables:** • **Explain the facts and trivias** • **Next class challenge** • **Project for the day** • **Additional Activity**	Guide the student to develop the project and share with us.

Quiz time - Click on in-class quiz	
Question	**Answer**
Why did we use sort() in our code? A. to sort the teams who have pressed the buzzer B. to sort the teams who have not pressed the buzzer C. to sort the teams who have pressed the buzzer in order of the values of timestamp D. to sort according to the names of the teams	**C**
How does the map() work? A. it loops through all the values of an array B. in map() every value is associated with a unique key. C. it takes a function which can render JSX tags for each element of the array D. all of the above	**D**
What is the functionality of the reset function? A. database is reset B. state of the app is reset C. state of the app and database is reset D. navigate to the next screen	**C**

FEEDBACK

- **Let the student experiment more with sort and map methods on arrays.**
- **Encourage the student to make reflection notes in the markdown format.**
- **Complement the student for her/his effort in the class.**

	You get a “hats off”. In the next class, we 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.	Make sure you have given at least 2 Hats Off during the class for: 

	Till then, goodbye!	 
Project Pointers and Cues (5 min)	***This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.** SCHOOL ATTENDANCE APP **Goal of the Project:** In Class 60, you have learnt about sort and map methods in javascript defined over arrays. You have used Firebase Database to create the Quiz Master App. In this project, you will be implementing the same concepts to create a Student Attendance App. **Story:** In this COVID-19 Pandemic, your school wants you to put your coding skills to use! They are finding it very difficult to manually take the attendance, maintain registers and give the data to the admin. School team has created an application where teachers can see the list of students, marked present/absent for a particular date. Could you please create another application?	**Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.**

	I am very excited to see your project solution and I know you both will do really well. Bye Bye!	
Teacher Clicks 		
Additional Activities	Encourage the student to write reflection notes in their reflection journal using markdown. Use these as guiding questions: • What happened today? - Describe what happened - Code I wrote • How did I feel after the class? • What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult?	The student uses the markdown editor to write her/his reflection in a reflection journal.

Activity	Activity Name	Links
Teacher Activity 1	Class activity	https://snack.expo.io/@whitehatjr/pro-c60-quiz-master:-class-activity
Teacher Activity 2	Reference	https://snack.expo.io/@whitehatjr/pro-c60-quiz-master-app

Student Activity 1	Class activity	https://snack.expo.io/@whitehatjr/pro-c60-quiz-master:-class-activity
Project Solution	School Attendance App	https://snack.expo.io/@snerrus/40df0f4401c411b95d129d2f1281fd76
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO_C60_withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/34b7403c-c90b-478e-ae5a-d6ac2a69e4ca.pdf