Add, Save and View Comments

Native App Development

Time: 60 mins

Introduction

In this class, the student/s will learn to create the comments screen for the feeds of the Snap Buzz app. They will also learn to add the comment for a selected feed, save and view the comments using API calls.

New Commands Introduced

<KeyboardAvoidingView>... </KeyboardAvoidingView>	Automatically adjusts the component's height, position, or bottom padding based on the keyboard height to remain visible
behavior={Platform.OS === 'ios' ? 'padding' : 'height'}	Specifies how to react to the presence of the keyboard for android and ios.
keyboardVerticalOffset={65}	Adds the distance between the top of the user screen and the React Native view
setAllComments([newComment, ...allComments])	Updates the value of setter function setAllComments by adding the newComment and list of comments within the allComments

Vocabulary

- ***KeyboardAvodingView*** component will automatically adjust components height, position, or bottom padding based on the keyboard height to remain visible.
- ***Spread operator*** is a JavaScript operator denoted by three dots (…). It is used when all elements from an object or array should be included in a new array or object, or should be applied one-by-one in a function call's arguments list.

Learning Objectives

Student/s should be able to:

- ***Recall*** the concept of API call to save the comment.
- ***Demonstrate*** how to save and fetch the comments from the database.
- ***Explain*** how state variables can be used to navigate between the comment and feed screen.
- ***Illustrate*** how a keyboard can be displayed without overlapping over a text box to remain visible.

Activities

1. **Class Narrative:** *(2 mins)*
    - Brief the student/s that they will be creating a comment screen to post and display the comments for the feeds.

2. **Concept Introduction Activity:** *(5 mins)*
    - Let the student/s play the explore-activity and explore the comment feature of the SnapBuzz app.
    - Ask students to observe how we can add and view comments within the specific feed's comment screen by clicking the comment icon.
    - Ask student/s to notice how on clicking the comment icon, one can navigate to the comment screen of the selected feed.
    - Using the slides, explain that the student/s will learn:
        - navigate to the comment screen
        - add keyboard without overlapping view
        - post and view the comments

3. **Activity 1: Navigate to the Comment Screen:** *(12 mins)*

    Teacher Activity: *(6 mins)*
    - Explain to the student/s how to navigate to the comment section for a selected feed.
    - Recall the use of state variables to handle screen state and store the selected feed and display its comment section.

    Student Activity: *(6 mins)*
    - Guide the student/s to navigate to the selected feed's comment screen.

4. **Activity 2: Adjust the Overlapping Keyboard:** *(12 mins)*

    Teacher Activity: *(6 mins)*
    - Ask the students to observe how the keyboard overlaps with the text box where we type the comments.
    - Explain the use of KeyboardAvoidingView component to avoid the keyboard overlapping with text input.
    - Demonstrate how we can avoid overlapping the keyboard with the text input using the KeyboardAvoidingView component.

    Student Activity: *(6 mins)*
    - Guide the student/s to debug the overlapping of keyboard with the text input using keyboardOverlappingView component.

    .

5. **Activity 3: Post and View Comments:** *(8 mins)*

    Teacher Activity: *(6 mins)*
    - Explain how a comment can be saved on an online storage using POST API call and using spread operator in JavaScript. Also explain how students will retrieve the comment from the online storage to view it on the comme.
    - Recall Scrollview to fetch and view all comments at a time.

Student Activity: *(8 mins)*

- Guide the student/s to save and display the comments as a list with scroll view.

6. **Introduce the Post class project:** *(2 min)*
    - Create a bidding screen to store and display the bid feed using a hook useEffect() and flatList component.

7. **Test and Summarize the class learnings:** *(5 mins)*
    - Check for understanding through quizzes and summarize learning after respective missions.
    - Summarize the overall class learning towards the end of the class.

8. **Additional activities:**
    - Encourage the student/s to add functionality to reply to a comment.
    - Encourage the student/s to add a functionality to display all the replies on comments.

9. **State the Next Class Objective:** *(1 min)*
    - We will learn to navigate to different screens of the SnapBuzz App.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Add, Save and View Comments	https://s3-whjr-curriculum-uploads.whjr.online/e781b199-77a4-433a-bb6d-b88098ead63f.html
Explore Activity	SnapBuzz App	https://snack.expo.dev/@procodingclass/wad-402---55-sa3_1-save-the-comments--solution
Teacher Activity 1	Navigate to the comment screen	https://snack.expo.dev/@procodingclass/wad-402---55-ta1-navigate-to-comments-screen
Teacher Activity 1 Solution	Navigate to the comment screen	https://snack.expo.dev/@procodingclass/wad-402---55-ta1-navigate-to-comments-screen--solution
Student Activity 1	Navigate to the comment screen	https://snack.expo.dev/@procodingclass/wad-402---55-sa1-navigate-to-comments-screen
Teacher Reference: Student Activity 1 Solution	Navigate to the comment screen	https://snack.expo.dev/@procodingclass/wad-402---55-sa1-navigate-to-comments-screen--solution
Teacher Activity 2	Adjust the Overlapping Keyboard	https://snack.expo.dev/@procodingclass/wad-402---55-ta2-adjust-the-overlapping-keyboard

Teacher Activity 2 Solution	Adjust the Overlapping Keyboard	https://snack.expo.dev/@procodingclass/wad-402---55-ta2-adjust-the-overlapping-keyboard--solution
Student Activity 2	Adjust the Overlapping Keyboard	https://snack.expo.dev/@procodingclass/wad-402---55-sa2-adjust-the-overlapping-keyboard
Teacher Reference: Student Activity 2 Solution	Adjust the Overlapping Keyboard	https://snack.expo.dev/@procodingclass/wad-402---55-sa2-adjust-the-overlapping-keyboard--solution
Teacher Activity 3	Save the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-ta3_1-save-the-comments
Teacher Activity 3 Solution	Save the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-ta3_1-save-the-comments--solution
Student Activity 3.1	Save the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-sa3_1-save-the-comments
Teacher Reference: Student Activity 3.1 Solution	Save the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-sa3_1-save-the-comments--solution
Student Activity 3.2	View the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-sa3_2-view-the-comments
Teacher Reference: Student Activity 3.2 Solution	View the Comments	https://snack.expo.dev/@procodingclass/wad-402---55-sa3_2-view-the-comments--solution
Student's Additional Activity 1	Reply to a Comment	https://snack.expo.dev/@procodingclass/wad-402---55-aa1-reply-to-a-comment
Teacher Reference: Student's Additional Activity 1 Solution	Reply to a Comment	https://snack.expo.dev/@procodingclass/wad-402---55-aa1-reply-to-a-comment---solution
Student's Additional Activity 2	Display the Comment Replies	https://snack.expo.dev/@procodingclass/wad-402---55-aa2-display-comment-replies
Teacher Reference: Student's Additional Activity 2 Solution	Display the Comment Replies	https://snack.expo.dev/@procodingclass/wad-402---55-aa2-display-comment-replies---solution
Post Class Project	Create a Bidding Screen	https://snack.expo.dev/@procodingclass/wad-402---55-pcp-create-a-bidding-screen
Teacher Reference: Post Class Project Solution	Create a Bidding Screen	https://snack.expo.dev/@procodingclass/wad-402---55-pcp-create-a-bidding-screen---solution