Mobile App UI using React Native

Advance Web Development

Time: 60 mins

Introduction

In this class, student/s will be introduced to the React Native framework by designing the UI of a mobile app. Student/s will learn to use the Snack Expo online editor by creating the first level of UI for the Language translator App. Student/s will learn to add React Native components and images to the App.

React Native Commands and Components Introduced

● <Text></Text>	Creates a Text component in the App.
● <View></View>	Creates a View container in the App.
● <TextInput></TextInput>	Creates a TextInput component in the App.
● <StyleSheet></StyleSheet>	Creates a StyleSheet component in the App.
● <Image></Image>	Creates an Image component in the App.
● .create()	Creates a StyleSheet style reference from the given object.
● placeholder	A temporary text displayed in the textinput.
● multiline	Sets the textinput to be singleline or multiline.
● numberOfLines	The number of lines of a textinput.
● flex	Sets a flex layout for the container.
● flexDirection	Sets the direction of components in a flex container. To be row or column.
● boxShadow	Sets a shadow in the hspae of a box around an element

Vocabulary

- **React Native:** A software framework used to create mobile applications that run on Android, iOS, and also on the web.
- **Snack Expo:** An open-source online editor used to create and run React Native Apps
- **React Native Components:** Components can be predefined as well as user-defined and are the basic building blocks of an App.

Learning Objectives

Student/s should be able to:

- ***Recall*** HTML tags and JavaScript syntax.
- ***Explain*** The similarities and differences between syntax JSX and HTML-JavaScript combination.
- ***Demonstrate*** how to use the Snack Expo to view the App on a mobile phone.
- ***Display*** the UI of a Language Translator App on the preview pane of iOS, Android, and the Web.

Activities

1. **Class Narrative:** *(2 mins)*
    - Brief the student/s that they will create the basic screen for a language translator App using Snack Expo. The App created will have a title, names, and images of the languages and two text fields for entering the text and displaying the translated text.

2. **Concept Introduction Activity:** *(5 mins)*
    - Let the student/s play the explore-activity and observe the working of the Language Translator App. The "from" language can be selected by clicking on the language button and the text can be entered in the first text input box By clicking on the "to" language button, the translated phrase and its pronunciation can be viewed in the second text input. The pronunciation of the phrase can be heard by clicking on the speaker button.
    - Guide the student/s to view the same App on the Web, Android, iOS, and mobile phones.
    - Guide the student/s to install Snack Expo Go on their devices. Explain the steps to scan the QR code of the app to view and use the language translator app on their device.
    - Using the slides, explain how:
        - Display the title of the App.
        - Style the title of the App and add text input fields.
        - Add the image and name of the languages of both the original text and translated text.

3. **Activity 1::** *(12 mins)*

    Teacher Activity: *(6 mins)*
    - Introduce to the student/s the Snak Expo online editor. Walk them through the file explorer, editing window, and the preview pane.
    - Explain to the student/s how the import statement is written and how the <Text> component is imported. Use <Text> to display the title of the App and display it in Web, Android, and iOS.

    Student Activity: *(6 mins)*
    - Guide the student/s to import and use the <Text> component to display the title of the App.
    - Explain to the student/s how syntax errors are displayed in the preview pane. Guide them to locate the error using the line number.

4. **Activity 2::** *(10 mins)*

    Teacher Activity: *(5 mins)*
    - Explain to the student/s the concept of Views and how Views can be placed inside Views to create a tree-like structure and components can be placed inside them. Explain that views can be used to give different styling to different parts of the screen.

- Create a container View and add a View for the title of the App and another view for the text input and add these components within these. Make the textinput editable, multiline, and add a placeholder for the text input.
- Show the student/s how StyleSheet is imported and defined in App.js. Add the styles to the Views and make the student/s observe the changes in the output. Set the flex value to 20 and see how the title takes up most of the screen.

Student Activity: *(5 mins)*

- Guide the student/s to add a second text input for the translated text.
- Guide the student/s to style the textinput and set the flex such that the text inputs and title are aligned as required in the App.

Probing question: What is the advantage of using flex boxes while styling?
Expected answer: The elements in a flexbox can be arranged and spaced according to the space available. The elements will automatically adjust their size and space between them based on the size of the screen.

5. **Activity 3::** *(10 mins)*

Teacher Activity: *(5 mins)*

- Demonstrate to student/s how images can be added and styled on the screen.

Student Activity: *(5 mins)*

- Guide the student/s to add and style the image for the second selected image.
- Guide the student/s to add the names of the languages and style them using the same style.

6. **Introduce the Post class project:** *(2 min)*
   - Add and style the title and search bar for the Search image screen.

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 text and image in the Monkey Chunky App.
   - Encourage the student/s to style the components added in the Monket Chunky App.

9. **State the Next Class Objective:** *(1 min)*
   - We will learn to add touchable components in the screen.

U.S. Standards:

CSTA: 2-AP-13, 2-AP-16, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**

Class Presentation	Mobile App UI using React Native	https://s3-whjr-curriculum-uploads.whjr.online/337ff716-1801-47bb-95ff-4bc62d49e6fc.html
Explore Activity	Language Translator App	https://snack.expo.dev/@procodingclass/wad-402---51-sa3-pronounce-the-translation---solution
Snack Expo on PlayStore	Expo - Downloadable link	https://play.google.com/store/apps/details?id=host.exp.exponent
Snack Expo on AppStore	Expo Go - Downloadable link	https://apps.apple.com/in/app/expo-go/id982107779
Teacher Activity 1	Introduction to React Native	https://snack.expo.dev/@procodingclass/wad-402---49-ta1-introduction-to-react-native
Teacher Activity 1 Solution	Introduction to React Native - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-ta1-introduction-to-react-native---solution
Student Activity 1	Add Title	https://snack.expo.dev/@procodingclass/wad-402---49-sa1-add-title
Teacher Reference: Student Activity 1 Solution	Add Title - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-sa1-add-title---solution
Teacher Activity 2	Add initial-textinput and Style the App	https://snack.expo.dev/@procodingclass/wad-402---49-ta2-add-initial-textinput-and-style-the-app
Teacher Activity 2 Solution	Add initial-textinput and Style the App - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-ta2-add-initial-textinput-and-style-the-app---solution
Student Activity 2	Add translated-textinput and Style the App	https://snack.expo.dev/@procodingclass/wad-402---49-sa2-add-translated-textinput-and-style-the-app
Teacher Reference: Student Activity 2 Solution	Add translated-textinput and Style the App - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-sa2-add-translated-textinput-and-style-the-app---solution
Techer Activity 3	Add from-Image and Language Name	https://snack.expo.dev/@procodingclass/wad-402---49-ta3-add-initial-flag-icon-and-language-name
Techer Activity 3 Solution	Add from-Image and Language	https://snack.expo.dev/@procoding

	Name	class/wad-402---49-ta3-add-initial-flag-icon-and-language-name---solution
Student Activity 3	Add and Style Language Name and Flag	https://snack.expo.dev/@procodingclass/wad-402---49-sa3-add-and-style-language-name-and-flag
Teacher Reference: Student Activity 3 Solution	Add and Style Language Name and Flag - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-sa3-add-and-style-language-name-and-flag---solution
Student's Additional Activity 1	Create the Monkey Chunky App Screen	https://snack.expo.dev/@procodingclass/wad-402---49-aa1-create-the-monkey-chunky-app-screen
Teacher Reference: Student's Additional Activity 1 Solution	Create the Monkey Chunky App Screen - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-aa1-create-the-monkey-chunky-app-screen---solution
Student's Additional Activity 2	Style the Monkey Chunky App	https://snack.expo.dev/@procodingclass/wad-402---49-aa2-style-the-monkey-chunky-app
Teacher Reference: Student's Additional Activity 2 Solution	Style the Monkey Chunky App - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-aa2-style-the-monkey-chunky-app---solution
Post Class Project	Create and Style the Image Search App	https://snack.expo.dev/@procodingclass/wad-402---49-pcp-create-and-style-the-image-search-app
Teacher Reference: Post Class Project Solution	Create and Style the Image Seach App - Solution	https://snack.expo.dev/@procodingclass/wad-402---49-pcp-create-and-style-the-image-search-app---solution