Locate a Doctor

Native App Development

Time: 60 mins

Introduction

In this class, the student(s) will learn to render maps in the web view and the mobile view of the BriskMed app using the google-map-react and react-native-maps libraries. Student(s) will also display the users(*doctors and patients*) on the map based on the coordinates of their location(latitude and longitude).

New Commands Introduced

<GoogleMapReact>	The component used to render a map in the web view.
<MapView>	The component used to display a map in the mobile view.
Platform.OS	Return the platform on which the app is running.
Location.requestForegroundPermissionsAsync	Request permission for accessing geo-location of the user when the app is in use.
getCurrentPositionAsync	Get the current position of the user.
Math.random	Returns a float number which is greater than or equal to 0 but less than 1.
CustomMarker	Displays a marker to identify a location on the map.
Marker	Displays a marker to identify a location on the map, used in the mobile view of the map.
MapView	Map component for mobile view

Vocabulary

- Latitude : Latitude is a geographic coordinate that specifies the east-west position of a point on the surface of the earth.
- Longitude: Longitude is a geographic coordinate that specifies the north-south position of a point on the surface of the earth.

Learning Objectives

student(s) should be able to:

- ***Recall*** the concept of sign in and sign out of the briskmed app.
- ***Demonstrate*** how location of different users are shown on the map.
- ***Explain*** how latitude and longitude of users is stored in the database.
- ***Illustrate*** how a different users can be displayed on the map screen depending on the user type.

Activities

1. **Class Narrative:** *(2 mins)*
   - Brief the student(s) that they will be exploring the map feature to display the geo location of the doctors.
   - The location will be displayed on the map based on the location of the user's device.

2. **Concept Introduction Activity:** *(5 mins)*
   - Let the student(s) explore the map feature on the BriskMed app.
   - Make student(s) sign in as a Patient as well as a Doctor to understand the difference between signing in as a doctor and signing in as a patient.
   - Ask student(s) to observe how the doctors are displayed on the map for different devices.
   - Ask student(s) to notice how the doctors are shown in the surroundings of the user.
   - Using the slides, explain that the student(s) will learn to:
     - Access User Location
     - Render the map in Web View
     - Render the map in Mobile View

3. **Activity 1: Access User Location:** *(12 mins)*

   Teacher Activity: *(6 mins)*
   - Explain to the student(s) how to get the user location using the expo-location library.
   - Demonstrate to the student(s) how to create a function called getLocation() which will ask for geolocation permission from the user.
   - Explain how the location state is initialized by storing the geolocation and updated using the setLocation() function.
   - Demonstrate to the student(s) how to use the location state to add the latitude and longitude of the user device in the user object and offset it randomly using the Math.random() function.
   - Let the student know that the location icon has to be slightly shifted to avoid overlapping icons. It might not be the case in real-world apps as users are placed far enough geographically but we have used it in our app for the sake of testing it, in case we login from our location using multiple user ids.
   - Explain to the student that the getLocation() function is called within the useEffect() hook so that it is called as soon as the app loads at the user end on signing up.

   Note: Make sure to delete the collections named ‘doctors’ and ‘patients’ from the database before starting this activity as the new users will be created while signing up and the existing users won’t have latitude and longitude which will show some errors.

   Student Activity: *(6 mins)*
   - Facilitate the student(s) by explaining the expectations from this activity. Explain that the user location is being tracked only on sign up and the student needs to add the functionality to update the user location every time they sign in. Tell the student(s) that we have provided the

getLocation function with two parameters and added a location state variable already. Guide the student(s) to add a reference to the database path based on the userType inside the getLocation function where the user information needs to be stored and update the current latitude and longitude of the user in the database using the update() function.

- Guide the student(s) to create the getFellowUsers function to get all the users' data based on the userType from the database and pass the data to the bottom tabs.
- Explain that the condition for checking the user type to update the database path in the db variable for the getLocation function is different from the condition used to update the databasePath variable used within the getFellowUsers function. The db variable in the getLocation function is initialized to the doctors' database path if the user is a doctor and vice-versa as the user-specific information has to be stored. In contrast, the database Path variable in the getFellowUsers function is initialized to the patients' database path if the user is a doctor and vice-versa to either display the doctors' location on a map for the patient's home screen or to display the list of patients on the doctors' home screen.

Note: Guide the student(s) to delete the collections called 'doctors' and 'patients'.

4. **Activity 2: Render the Map in Web View:** *(12 mins)*

Teacher Activity: *(6 mins)*

- Ask the student(s) to describe how the map opens differently in the web view.
- Explain how to add the google-map-react library using the keyword, require and use the GoogleMapReact component to display the map.
- Explain that the keyword, require is used to import the google-map-react library instead of the keyword import for conditionally rendering the map in the web view. We will learn more about this when we conditionally render the map in two different views in the next activity.
- Demonstrate how CustomMarkers are used to pin the doctors on the map.
- Explain that the GoogleReactMap component helps to display the map with given coordinates.

Student Activity: *(6 mins)*

- Guide the student(s) to display the map using styling and alignment property.
- Guide the student(s) to display the user on the map using the custom marker
- Guide the student(s) to import the WebMap.js file in the map.js file to display the map in the web view.

.

5. **Activity 3: Render the Map in Mobile View:** *(8 mins)*

Teacher Activity: *(6 mins)*

- Explain how to add the MapView component from the react-native-maps library using the keyword, require.
- Explain that the coordinates within MapView are used to specify the coordinates of the user's location.
- Demonstrate how to display the doctors on the map using the Marker and the CustomMarker components.
- Draw the attention of the students to the conditional rendering of maps explained after the activity and highlight that this would not be possible using the keyword, import.

Student Activity: *(8 mins)*

- Guide the student(s) to use the CustomMarker and Marker component to display the users in the MobileMap.js file.
- Guide the student to display the map either in the web view or in the mobile view based on the user's selection by writing this code within the Map.js file where both WebMap.js and MobileMap.js have been imported.

  Note: require is used to import the map libraries for conditional selection between web view and mobile view, which is not possible with the import keyword.

6. **Introduce the Post class project:** *(2 min)*
   - Inform the student(s) that a map is already displayed in the mobile view of the campus security app.
   - Student(s) need to render the map in the web view of the app to display the location of the students.
   - Import the WebMap.js file in the Map.js file and display the web view using conditional rendering.

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

8. **Additional activities:**
   - Encourage the student(s) to highlight the current user in a different color.
   - Encourage the student(s) to debug the app code to update the user's location when they sign in.

9. **State the Next Class Objective:** *(1 min)*
   - We will learn to add a feature to click open custom markers to show the details of doctors and swipe through them.

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	Locate the Doctor	https://s3-whjr-curriculum-uploads.whjr.online/c0148efb-91a1-42ff-888b-bed598200847.html
Explore Activity	BriskMed App	https://snack.expo.dev/@procodingclass/wad-402-c59-sa3-view-location-on-a-mobile-device---solution?platform=web

Teacher Activity 1	Access user location	https://snack.expo.dev/@procodingclass/wad-402-c59-ta1-access-user-location?platform=web
Teacher Activity 1 Solution	Access user location	https://snack.expo.dev/@procodingclass/wad-402-c59-ta1-access-user-location---solution?platform=web
Student Activity 1	Access user location	https://snack.expo.dev/@procodingclass/wad-402-c59-sa1-access-user-location?platform=web
Teacher Reference: Student Activity 1 Solution	Access user location	https://snack.expo.dev/@procodingclass/wad-402-c59-sa1-access-user-location---solution?platform=web
Teacher Activity 2	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-ta2-render-the-map-in-web-view?platform=web
Teacher Activity 2 Solution	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-ta2-render-the-map-in-web-view--solution?platform=web
Student Activity 2.1	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa2-render-the-map-in-web-view?platform=web
Teacher Reference: Student Activity 2.1 Solution	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa2-1-render-the-map-in-web-view-solution-?platform=web
Student Activity 2.2	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa2-2-render-the-map-in-web-view?platform=web
Teacher Reference: Student Activity 2.2 Solution	Render the Map in Web View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa2-render-the-map-in-web-view--solution?platform=web
Teacher Activity 3	Render the Map in Mobile View	https://snack.expo.dev/@procodingclass/wad-402-c59-ta3-render-the-map-in-mobile-view?platform=web
Teacher Activity 3 Solution	Render the Map in Mobile View	https://snack.expo.dev/@procodingclass/wad-402-c59-ta3-render-the-map-in-mobile-view---solution?platform=web
Student Activity 3	Render the Map in Mobile View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa3-render-the-map-in-mobile-view?platform=web
Teacher Reference: Student Activity 3 Solution	Render the Map in Mobile View	https://snack.expo.dev/@procodingclass/wad-402-c59-sa3-render-the-map-in-mobile-view--solution?platform=web
Student's Additional Activity 1	Highlight the Current User	https://snack.expo.dev/@procodingclass/wad-402-c59-aa1-highlight-the-current-user?platform=web
Teacher Reference: Student's Additional Activity 1 Solution	Highlight the Current User	https://snack.expo.dev/@procodingclass/wad-402-c59-aa1-highlight-the-current-user---solution?platform=web

Student's Additional Activity 2	Debug the Code	https://snack.expo.dev/@procodingclass/wad-402-c59-aa2-debug-the-code?platform=web
Teacher Reference: Student's Additional Activity 2 Solution	Debug the Code	https://snack.expo.dev/@procodingclass/wad-402-c59-aa2-debug-the-code---solution?platform=web
Post Class Project	Show Users on Map	https://snack.expo.dev/@procodingclass/wad-402-c59-pcp-show-users-on-the-map?platform=web
Teacher Reference: Post Class Project Solution	Show Users on Map	https://snack.expo.dev/@procodingclass/wad-402-c59-pcp-show-users-on-the-map--solution?platform=web