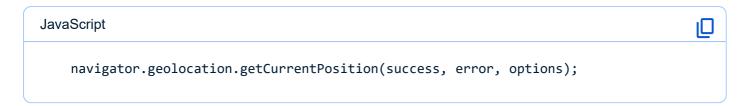
Allowing users to choose between navigation apps like Google Maps and Waze in a React application involves using device-specific APIs or third-party libraries to handle the navigation requests. The implementation will differ slightly between web and mobile (React Native) environments.

Web Implementation

• **Determine User Location:** Use the Geolocation API to get the user's current coordinates.



Present Navigation Options:

Display buttons or a select dropdown for the user to choose between Google Maps and Waze.

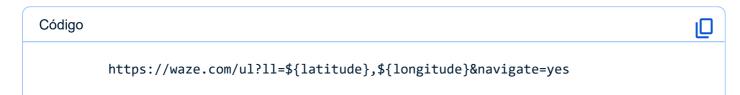
Generate Navigation URLs:

Construct the appropriate URL based on the selected app and destination coordinates.

Google Maps:



waze.



 Open the Navigation App: Use window.open() to open the selected navigation URL in a new tab or window.

```
Código

const handleNavigation = (app) => {
  const destination = { latitude: 34.0522, longitude: -118.2437 }; // Example destination
  let url = '';

if (app === 'googleMaps') {
  url = `https://www.google.com/maps/dir/?api=1&destination=${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${destination.latitude},${
```

React Native Implementation

Install a Navigation Library:

Use a library like react-native-map-link or react-native-launch-navigator for handling navigation requests.

Determine User Location:

Use react-native-geolocation-service or similar to get user coordinates.

Present Navigation Options:

Similar to the web implementation, present navigation app choices to the user.

Use the Navigation Library:

Call the appropriate function from the chosen library to open the selected app with the destination coordinates.

```
import MapLink from 'react-native-map-link';

const handleNavigation = (app) => {
  const destination = { latitude: 34.0522, longitude: -118.2437 }; // Example destination

MapLink.openMap({
  latitude: destination.latitude,
  longitude: destination.longitude,
  app: app,
  apps: [app]
  });
};

return (
```

```
<div>
     <Button title="Open in Google Maps" onPress={() => handleNavigation('google-maps')} /
     <Button title="Open in Waze" onPress={() => handleNavigation('waze')} />
     </div>
);
```

Ensure that the necessary permissions for location access are requested and granted in both web and React Native environments. For React Native, configure the info.plist (iOS) and AndroidManifest.xml (Android) files to include the relevant permissions.