Adding the Umoove library to your React project

1. Add ‘umoove_wasm’ folder to public folder in the project
2. Add the ‘UmooveApi.js’ file to ‘src’ folder in the project (Should be available for all the components of the app)
3. Add the following code to **index.html** file, anywhere before the :

```
<!-- UMOOVE LIBRARY CODE -->
   <div class="wasm_load_div" id="wasm_load_div" style="visibility: hidden; position: absolute;left: 0px;top: 0px;"></div>
   <!-- Necessary wasm preloading -->
   <script  type="text/Javascript" src="./umoove_wasm/wasm_load.js" ></script>
   <!-- Umoove Library -->
   <script async type="text/Javascript" src="./umoove_wasm/umoove_eye.js"></script>
   <!-- Umoove Video management and interface -->
   <script type="text/JavaScript" src="./umoove_wasm/umoove_load_eye.js"></script>

   <div class="cameraWindow">
        <video id="video" class="video" width = "640" height = "480" style="display: none"></video>
   </div>
   <script>
       var currentUmooveResults = []
       var readingData = []
       var newData = false
       var collectData = false
       var newUmooveData = (umoooveResults) => {
               newData = true;
               currentUmooveResults = umoooveResults;
               if(collectData){
                 readingData.push({time:Date.now(),eye_data:umoooveResults})
               }
             };
   </script>
   <!-- END UMOOVE LIBRARY CODE -->
```

Umoove API Calls

The following Api calls are available in 'UmooveApi.js':

1. **API_loadUmooveLibrary() -** Initiating the library and starting the video stream from the camera
2. **API_getUmooveStream() –** Returns **stream** object which can be used as a source of a <video> tag to present video
3. **API_startUmoove() -** Start the eye tracking process
4. **API_getUmooveTracking() -** Checks If tracking if user is detected and tracking is active - returns true/false
5. **API_stopUmoove() -** Stop the eye tracking process
6. **API_getDistance() -** Returns the current distance (in CM) of the user from the camera
7. **API_startReading() -** Starts collecting eye tracking data while user reading the paragraph
8. **API_stopReading(num_of_lines)** - Receive number of lones in paragraph - Stops collecting eye tracking data and calculates the reading rate per line – returns an array (size num_of_lines-1) of time in ms of each line in paragraph