Michael McCrary

Winter Engineering and Electronic Dudes

Web Page Limitations Document

Purpose:

The purpose of this document will be to research methods of implementation of the core functionality of our application, Attendance Made Easy, using modern and/or available capabilities of HTML and physical hardware. The specificity of this task is set on figuring out if it is possible for a web page to look at what time it is currently, whether it needs to be open or not, and get the user's geolocation at the time the user's data is consensually gathered.

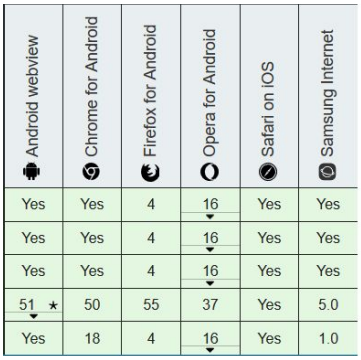
Research Methods:

I used the Internet to explore and read about different methods and requirements of how different kinds of browsers utilize the HTML 5 API in order to be properly integrated with mobile use. HTML code was also analyzed to gain at least a smaller understanding of this functionality before full implementation based on the research gathered here.

Results:

Most, if not all, sources researched all seemed to agree that using the HTML5 Geolocation API is the best and most modern way to gather a phone's geolocation, when the device is on and the page gathering the location is open. Most modern smartphones are able to detect accuracy within

3 meters of the person's actual geolocation. Sources also list and indicate compatibility specifications for mobile and desktop browsers, some of which are as seen here:



Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
Yes	Yes	4	16	Yes	Yes
Yes	Yes	4	16	Yes	Yes
Yes	Yes	4	16	Yes	Yes
51 *	50	55	37	Yes	5.0
Yes	18	4	16	Yes	1.0

The compatibility for mobile is the only one seen here, since it is most likely going to be the most important side of making sure our website works across all mobile platforms. However, there is also built-in capability of determining if a browser maintains Geolocation compatability, seen here:

```
// check for Geolocation support
if (navigator.geolocation) {
  console.log('Geolocation is supported!');
}
else {
  console.log('Geolocation is not supported for this Browser/OS.');
}
```

The three main functions of this API are getCurrentPosition(), watchPosition(), and clearWatch; getCurrentPosition does exactly what it says, with a GeolocationPosition object being returned and other return data either being guaranteed as standard return data or some other kinds being only available via utilization of the function in higher end browsers or with higher hardware capable phones. watchPosition() will store this gathered information and return a long value that invokes what callback function will be used when the position changes, and clearWatch removes all the data of the user gathered with the two previous functions. All of this will, of course, be executed only if the user grants permission for the application to access the location of the device. This idea of application is just as easy as implementing the few lines of code necessary to gather location based on user consent, but the next step would have to be storing this data or updating it in the application's database.

Conclusions:

Overall, a large understanding was gained about the surprisingly simplistic ability to gather a geolocation from a device. The more difficult part of implementation will be more in regards to how the data is interacted with and stored, something that will presumably require more direct JavaScript and database interaction.

Sources:

https://developer.mozilla.org/en-US/docs/Web/API/Geolocation

https://www.w3schools.com/html/html5_geolocation.asp

https://developers.google.com/web/fundamentals/native-hardware/user-location

https://www.htmlgoodies.com/html5/navigating-html5-geolocation.html

https://www.storelocatorwidgets.com/blogpost/20453/Everything_you_ever_wanted_to_know_about_HTML5_Geolocation_Accuracy