What to do over the Christmas break

While everyone needs time to unwind, over indulge and recharge over the holidays there will be some downtime when you want to have something to work on or look ahead.

Here is my list of things that you can do.

1. Practice working with GitHub and uploading and downloading files from GitHub.

2. Install Eclipse with the ADT plugin, JDK and SDK.

3. Try to build an Android PhoneGap app that uses at least one feature from the PhoneGap script library, other than navigator.notification.alert();

4. Read ANOTHER chapter from **PhoneGap Essentials**

5. Have a good rest.

Part Two of the Final Project is to take the jQuery Mobile Trivia Quiz Web App that you have already built and turn it into an Android PhoneGap App.

There are a few enhancements to add to the app too, before converting it to a PhoneGap App.

1. The highscores need to be sent to the server via an AJAX call. A Provided PHP page called set-highscores.php will accept three pieces of information - username, score and your account name (eg:abcd0001). This page will save the scores in a database and return an error code and message to you.

2. The highscore list needs to be retrieved via an AJAX call using a provided PHP page. You pass your account name (eg: abcd0001) to the PHP page via an AJAX call and it will return a JSON file with the list of scores.

3. When the game is complete you should display a button in the header that lets the user Play Again. Make sure the Play Again button appears in the header for ONLY the Scores pages.

[highscore.ziphttps://algonquin.instructure.com/images/popout.png](https://algonquin.instructure.com/courses/757/files/333319/download?wrap=1) - this zip file includes the two PHP pages that you will need to include in your folder on Edumedia in order to make the score functionality work. Note: the AJAX calls for scores will likely only work if tested on Edumedia.

Here is an example of how you would call the get-highscores.php page to fetch a JSON object that contains a list of scores:

$.ajax({  
 url:'get-highscore.php',  
 data:'account=abcd0001',  
 dataType:"json",  
 type:"POST"  
}).done(function(scoreData, textStatus, jqXHR){  
 //got the list of scores  
 console.log( scoreData );  
}).fail(function(jqXHR, textStatus, errorThrown){  
 //failed to get data...  
 console.log( "ERR: " + errorThrown );  
});

The data parameter in the ajax call will pass the account name to the php page. This account will fetch YOUR list of high scores. Simply replace the abcd0001 with your account name.

The response that comes back from the call to the php page will provide, at a minimum, a property called **code** and a second property called **message**. If **code**is zero then there were no errors. If the **code** is any other number then there was an error and the **message** is something that can be displayed to users.

In order to upload a new score, when the user finishes playing the game, then you would make an AJAX call to the set-highscores.php page. This call will also return a **code** and a **message** property. The data property for the AJAX call to set the score requires more information than just the account. It would require additionally, a username and a score.

$.ajax({  
 url:'set-highscore.php',  
 data:'account=abcd0001&username=bubba&score=55',  
 dataType:"json",  
 type:"POST"  
}).done(function(scoreData, textStatus, jqXHR){  
 //this is the code and message about the success or failure to save a score  
 console.log( scoreData );  
}).fail(function(jqXHR, textStatus, errorThrown){  
 //failed to save the score...  
 console.log( "ERR: " + errorThrown );  
});

Part Two of the Final Project is to take the jQuery Mobile Trivia Quiz Web App that you have already built and turn it into an Android PhoneGap App.

There are a few enhancements to add to the app too, before converting it to a PhoneGap App.

1. The highscores need to be sent to the server via an AJAX call. A Provided PHP page called set-highscores.php will accept three pieces of information - username, score and your account name (eg:abcd0001). This page will save the scores in a database and return an error code and message to you.

2. The highscore list needs to be retrieved via an AJAX call using a provided PHP page. You pass your account name (eg: abcd0001) to the PHP page via an AJAX call and it will return a JSON file with the list of scores.

3. When the game is complete you should display a button in the header that lets the user Play Again. Make sure the Play Again button appears in the header for ONLY the Scores pages.

[highscore.ziphttps://algonquin.instructure.com/images/popout.png](https://algonquin.instructure.com/courses/757/files/333319/download?wrap=1) - this zip file includes the two PHP pages that you will need to include in your folder on Edumedia in order to make the score functionality work. Note: the AJAX calls for scores will likely only work if tested on Edumedia.

Here is an example of how you would call the get-highscores.php page to fetch a JSON object that contains a list of scores:

$.ajax({  
 url:'get-highscore.php',  
 data:'account=abcd0001',  
 dataType:"json",  
 type:"POST"  
}).done(function(scoreData, textStatus, jqXHR){  
 //got the list of scores  
 console.log( scoreData );  
}).fail(function(jqXHR, textStatus, errorThrown){  
 //failed to get data...  
 console.log( "ERR: " + errorThrown );  
});

The data parameter in the ajax call will pass the account name to the php page. This account will fetch YOUR list of high scores. Simply replace the abcd0001 with your account name.

The response that comes back from the call to the php page will provide, at a minimum, a property called **code** and a second property called **message**. If **code**is zero then there were no errors. If the **code** is any other number then there was an error and the **message** is something that can be displayed to users.

In order to upload a new score, when the user finishes playing the game, then you would make an AJAX call to the set-highscores.php page. This call will also return a **code** and a **message** property. The data property for the AJAX call to set the score requires more information than just the account. It would require additionally, a username and a score.

$.ajax({  
 url:'set-highscore.php',  
 data:'account=abcd0001&username=bubba&score=55',  
 dataType:"json",  
 type:"POST"  
}).done(function(scoreData, textStatus, jqXHR){  
 //this is the code and message about the success or failure to save a score  
 console.log( scoreData );  
}).fail(function(jqXHR, textStatus, errorThrown){  
 //failed to save the score...  
 console.log( "ERR: " + errorThrown );  
});