Worksheet 8
Introduction to the Slim Framework

IMPORTANT: You must catch up first!

Please note that if you are behind with any of the following exercises, you should catch up with them first before attempting object-oriented PHP. The exercises below are important to complete to get a C or B in the assignment or complete the TCA, while object-oriented PHP is only required for a Grade A on the assignment.

You MUST complete the following before you start this worksheet:

-Worksheet 3: PHP and databases, questions 1-8 (the main questions, not the "further questions")
-Worksheet 4: multi-page PHP applications, questions 1-7 (note that q5-7 are "further questions" but still important)
-Worksheet 5: logins, questions 1-9 (note that q8,9 are "further questions" but still important)
-Worksheet 6: SQL injection section. Ensure your SQL queries are converted to prepared statements.
- Worksheet 7: object-oriented PHP, question 1 (creating and using a basic class). DAO not needed.

Do this work on Edward2, not Neptune. Slim is only setup fully on Edward2.

1) Try out the basic Slim example from the lecture notes with two routes, '/hello' and '/time'.

2) Create a Slim HitTastic! application. It should have these routes:

- the 'root' route (/). This should display ALL songs in the database. Hint: create a string variable, add (append) each result in turn to it in your while loop, and then output the string variable as your response. There is no need to use PHP templates just yet.
- the route /artist/{artist}. This should show all songs by the artist specified in the placeholder, using the same technique as the 'root' route.
- the route /artist/{artist}/song/{song}. This should show a given song by a given artist.

3) Add these further routes:
- the route '/addsong'. This should read in song details (using POST) and add them to your 'wadsongs' table.

- the route '/inputsong'. This should show a form allowing the user to add the song details. Use a PHP template for the form. In this case, the template can be plain HTML - no PHP will be needed. This should post to your 'addsong' route.

The action on the form should be 'addsong' so that when the form details have been entered, they are sent to the 'addsong' route.

- the route '/song/{id}'. **Using PHP templates**, display the full details of the song with the given ID.