HLL Part 3 – PHP

Lab Sheet 2 – Tentative Steps in PHP database connectivity

This exercise sheet is aimed at helping you getting a connection to a mysql database via your good old PHP scripts. But first of all you are going to need a mysql account with the university mysql server and get all of the software running correctly for username. DO THIS AS SOON AS POSSIBLE – you must not be doing this as the coursework deadline begins to loom.

MySQL Introduction

MySQL is installed on *mysql.cs.nott.ac.uk* and can also be accessed from *Robin*, *Tuck*, *Much*, *Scarlet*, and *Marian* also as well as any local machine that has suitable client software installed. You will be accessing it directly from your PHP scripts but you may also want to check your data from the command line.

Configuration for MySQL

To access the MySQL database from the Unix system, you will need to add **mysql** to your current list of packages. For accounts that are set up to use bash by default (accounts created during Summer 2000 or afterwards), simply edit the *.profile* file in your home directory, search for the line that beings *PACKAGES=* and add a space and **mysql** to this variable, remebering to save the file and log out of Unix completely afterwards (or use . ./.profile).

Obtaining a MySQL Account

If you have not used MySQL before, you will need an account setting up, you can do this yourself by logging into a unix server that is set up for correctly (*Robin*, *Tuck*, *Much*, *Scarlet*, or *Marian*) and typing:

```
create_mysql
```

Remeber to make a note of the password you are given, you should change this as soon as you log into *mysql*.

Using MySQL

Once you have added the *mysql* package as detailed above (this command won't be recognised otherwise), you can access the **MySQL** command line by typing:

```
mysql
```

At the command-line prompt, or:

```
mysql -u [USERNAME] -p [DATABASE NAME]
```

To specify a particular account to log into, where *USERNAME* is the MySQL username (which in most cases is the same as your Unix username) and

DATABASE NAME is the name of the database (which is also usually the same as your Unix username). The *-p* option ensures you are prompted for a password.

Once logged in you can change your password by typing:

```
set PASSWORD=PASSWORD('YourNewPassword');
```

Replace *YourNewPassword* with whatever you wish your password to be. Now you have access to mysql and your up and running – you can type in those mysql commands directly to the mysql command line. There is a full manual for mysql on the TSG website but here are some basics below for you to try. The key to remember when coding MySQL statements directly in is that they only process when you type a semi-colon – not just when you press return:

MySQL Commands

SHOW TABLES This will show you what tables you have created in your database account and hence what tables you can use. To start with, this will of course be empty…

CREATE TABLE This statement will setup a table for you to start putting data into – try out a line such as:

```
 CREATE TABLE people (firstname varchar(30), lastname
varchar(30), age integer, bio text);
```

INSERT This statement actually adds lines into your database:

```
INSERT INTO people VALUES (“Sven-Goran”, “Erikkson”, 56,
“England Football Manager”);
```

Try adding more insert statements - maybe your favourite lecturers (!?) – until you have a few records in your database.