Programming Evaluation for:

PHP

Catalyst IT Australia Pty Limited

Version: 1.1

September 2015

Commercial in Confidence

Suite 501–504, Level 5, 89 York Street, Sydney 2000
PO Box Q573, Queen Victoria Building, NSW 1230, Australia
+61 2 8203 9777 // info@catalyst-au.net // www.catalyst-au.net

Table of Contents

Revision History			
Modified by	**Date**	**Version**	**Change**
Matt Porritt	15/9/2014	0.1	Initial draft
Matt Porritt	18/9/2014	1.0	Review Updates
Matt Porritt	9/9/2015	1.1	Updates

1. Script Task

Create a PHP script that is executed form the command line that accepts a CSV file as input (see command line directives below) and processes the CSV file. The parsed file data is to be inserted into a MySQL database. A CSV file will be provided as part of this task that contains test data, the script must be able to process this file.

The PHP script will need to correctly handle the following criteria:

- CSV file will contain user data and have three columns: name, surname, email (see table definition below)
- CSV file will have an arbitrary list of users
- Script will iterate through the CSV rows and insert each record into a dedicated MySQL database into the table “users”
- The users database table will need to be created/rebuilt as part of the PHP script. This will be defined as a Command Line directive below.
- Name and surname field should be set to be capitalised e.g. from “john” to “John” before being inserted into DB
- Emails need to be set to be lower case before being inserted into DB
- The script should validate the email address before inserting to make sure that it is valid (valid means that it is a legal email format e.g. “xxxx@asdf@asdf is not a legal format). In the instance that an email is invalid, no insert should be made to database and error message reported to STDOUT.

We are looking for a script that is robust and gracefully handles errors/exceptions.

The PHP script command line argument definition is outlined in section 5: Script Command Line Directives. However, user documentation will be looked upon favourably.

1.1. Source Control

The code for the test is to be managed using “git” as the Version Control System, with the repository made available via online repository: GitHub (github.com), bitbucket (bitbucket.org) etc. This will be how the sample code will be delivered to Catalyst at the completion of development.

A repository with only one commit is not acceptable. Showing the development process is just as important as the task itself.

1.2. Assumptions

- The deliverable will be a running PHP script – it will be executed on an Ubuntu 14.04 instance
- Catalyst would like to see your development process history in git – not just a completed script.
- There may be some libraries that need to be installed via apt-get or pear. This is fine but these dependencies should be outlined in provided install documentation.
- MySQL database server is already installed and is version 5.x – MySQL user details should be configurable.
- PHP script will be called – user_upload.php
- CSV file will be called users.csv and is provided with this document.

If there are any unclear details here, then you are welcome to make assumptions as long as they are clearly stated and documented as part of the deliverables.

1.3. User Table Definition

The MySQL table should contain at least these fields

- name
- surname
- email (email should be set to a UNIQUE index)

1.4. Script Command Line Directives

The PHP script should include these command line options (directives)

- --file [csv file name] – this is the name of the CSV to be parsed
- --create_table – this will cause the MySQL users table to be built (and no further
- action will be taken)
- --dry_run – this will be used with the --file directive in the instance that we want to run the script but not insert into the DB. All other functions will be executed, but the database won't be altered.
- -u – MySQL username
- -p – MySQL password
- -h – MySQL host
- --help – which will output the above list of directives with details.

1.5. Questions

The aim of this task is to test both your development skills as well as simulate a real world project task. Guidance can be sort regarding the requirements and deliverables of this task. Questions on “how to do it” won't be accepted.

2. Logic Test

Create a PHP script that is executed form the command line. The script should:

- Output the numbers from 1 to 100
- Where the number is divisible by three (3) output the word “foo”
- Where the number is divisible by five (5) output the word “bar”
- Where the number is divisible by three (3) and (5) output the word “foobar”
- Only be a single PHP file

2.1. Example

An example output of the script would look like:

1, 2, foo, 4, bar, foo, 7, 8, foo, bar, 11, foo, 13, 14, foobar

2.2. Deliverable

The deliverable for this task is a php script called foobar.php. There is no need to put this task into source control (but you can if you want to).

2.3. Questions

The aim of this task is to test your development skills. Guidance can be sort regarding the requirements and deliverables of this task. Questions on “how to do it” won't be accepted.