Programming Evaluation for:

PHP

Catalyst IT Australia

Open Source Technologists

Version 1.4

September 2019

Commercial in Confidence

catalyst
open source technologists

Level 9, Suite 903, Tower B, The Zenith, 821 Pacific Highway, Chatswood, 2067
+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
Matt Porritt	5/8/2016	1.2	Defining PHP version
Matt Porritt	17/7/2017	1.3	PHP and Ubuntu version update
Arjen Lentz	14/6/2019	1.4	PHP, Ubuntu, PostgreSQL version update. Proofread and template fixups

1. Script Task

Create a PHP script, that is executed from the command line, which 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 PostgreSQL database. A CSV file is provided as part of this task that contains test data, the script must be able to process this file appropriately.

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 PostgreSQL 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 case that an email is invalid, no insert should be made to database and an error message should be 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 1.4 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 is to 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 18.04 instance
- PHP version is: 7.2.x
- 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, pear or composer. This is fine but these dependencies should be outlined in provided install documentation

- PostgreSQL database server is already installed and is version 9.5 (higher versions are fine) – DB 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, 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 PostgreSQL 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 PostgreSQL users table to be built (and no further action will be taken)
- --dry_run – this will be used with the --file directive in case 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 – PostgreSQL username
- -p – PostgreSQL password
- -h – PostgreSQL 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 sought regarding the requirements and deliverables of this task. Questions on “how to do it” won't be accepted.