The LSS-100P Web Application site is common file structure containing the index.php and post.php at the root with an .htaccess file. This site require re-write rule to be enabled and requires a server configuration command “a2enmod rewrite” for Debian and for Red Hat it’s typically already loaded and the directive in .htaccess is applied when the module is loaded. For Red Hat the yum install rewrite_mod will update the configuration with the installed module. To view current modules installed enter: apachectl -M

rewrite_module (shared)
ssl_module (shared)

The following is the contents of the .htaccess file:

```
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteBase /
   RewriteCond %{REQUEST_FILENAME} !-f
   RewriteCond %{REQUEST_FILENAME} !-d
   RewriteRule . /index.php [L]
</IfModule>
```

For simple single website the following root files and directories are as shown below:

```
drwxrwxr-x  4 root www-data 4096 Jun 15 20:55 api
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 app
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 components
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 controllers
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 css
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 directives
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 fonts
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 images
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 includes
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 lib
drwxrwxr-x  4 root www-data 4096 Jun 30 16:12 nbproject
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 partials
drwxrwxr-x  3 root www-data 4096 Jun 18 00:23 serverservices
drwxrwxr-x  2 root www-data 4096 Jun 15 20:55 services
-rw-rw-r--  1 root www-data 6821 Jul  2 18:55 index.php
-rw-rw-r--  1 root www-data 5507 Jun 30 16:12 post.php
-rwxr--r--  1 root www-data  196 Jun 15 20:56 .htaccess
```

The web site files are located by default at /var/www/html and can be located in another fold as required for a multiple site configuration. In our case we have a configuration for both port 80 and 443 that requires a virtual host declaration in the configuration files. In addition to virtual host there are a couple of directory directives that control access. The following are some excerpts from the /etc/httpd/httpd.conf for Red Hat and /etc/apache2/apache2.conf for Debian.

```
# AccessFileName: The name of the file to look for in each
# directory for additional configuration directives.  See
# also the AllowOverride directive.

AccessFileName .htaccess
```

**Note that for Red Had this directive maybe located in another configuration file.*

```
#
# Relax access to content within /var/www.
#
<Directory "/var/www">
   AllowOverride None
   Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/lssdb">
   #
   # Possible values for the Options directive are "None",
   # "All", or any combination of:
   # Indexes Includes FollowSymLinks SymLinksifOwnerMatch
   # ExecCGI MultiViews
   #
   # Note that "MultiViews" must be named
   # *explicitly* --- "Options All" doesn't give it to you.
   #
   # The Options directive is both complicated and
   # important.  Please see
   # http://httpd.apache.org/docs/2.4/mod/core.html#options
   # for more information.
   Options Indexes FollowSymLinks MultiViews

   #
   # AllowOverride controls what directives may
   # be placed in .htaccess files.
   # It can be "All", "None", or any combination of the
   # keywords: Options FileInfo AuthConfig Limit
   #
   AllowOverride All

   #
   # Controls who can get stuff from this server.
   #
   Require all granted
</Directory>

# The following lines prevent .htaccess and .htpasswd
# files from being viewed by Web clients.
#
<Files ".ht*">
   Require all denied
</Files>

* Note: other properties like “Allow from all” are available.
```

The basic VirtualHost directives are located by default in different files between Red Hat and Debian. As a note the Red Hat distribution will have most of the https 443 created by the yum install ssl_mod in the /etc/httpd/conf.d/ssl.conf file. With Debian it's located in the /etc/apache2/sites-available/default file. With regards to configuring the Apache 2 server for the two distributions see the respective https setup documentation provided by USL.

When the LSS-100P is commanded to execute measurement(s), the results are sent as a post to the LSS Database Web service, the site's post.php file. This is done by defining a script in the LSS-100P that executes measurements and sends a “lss100.sys.log_post” command with the pipe followed by url as shown below in the example:

lss100.sys.log_post|http://lssdb.uslinc.com/post.php

When the post is received, it contains html text describing a number of records (samples) that are parsed an logged into the database. The data is stored in the RawData table that is used for generating email notifications. These batched when the cron process executes. The cron process is located within the “serverservices” folder as shown in shell directory screen capture:

-rw-r--r-- 1 root root 3420 Jun 15 20:55 cron.php

The Linux operating system has an application scheduling feature called “cron” that can be setup to execute at either a set interval or time. For defining an exact time use “crontab” and for a daily execution use the cron.daily method. To see the exact time that daily is executed enter the following in the command shell:

~# cat /etc/crontab

For the simplest implementation create a file in /etc/cron.daily as shown:

-rwxr-xr-x 1 root root 48 Jun 15 22:53 lsscron

The lsscron is a shell script as follows that will be executed daily that then invokes the LSS-100 Web Application's cron.php script. When invoked, the notification entries are then processed to send emails to the users that are assigned to the theater.

```
#!/bin/sh
php /var/www/serverservices/cron.php
```