

A scenic landscape of a mountain range with autumn foliage in the foreground and middle ground, and a body of water in the foreground.

# macOS High Sierra

[localhost](#)[macOS](#)[macOS Sierra](#)[PHP](#)[MySQL](#)[HTTPS](#)[SSL](#)[Apache](#)

## Set up localhost on macOS High Sierra (Apache, MySQL, and PHP 7) with SSL/HTTPS

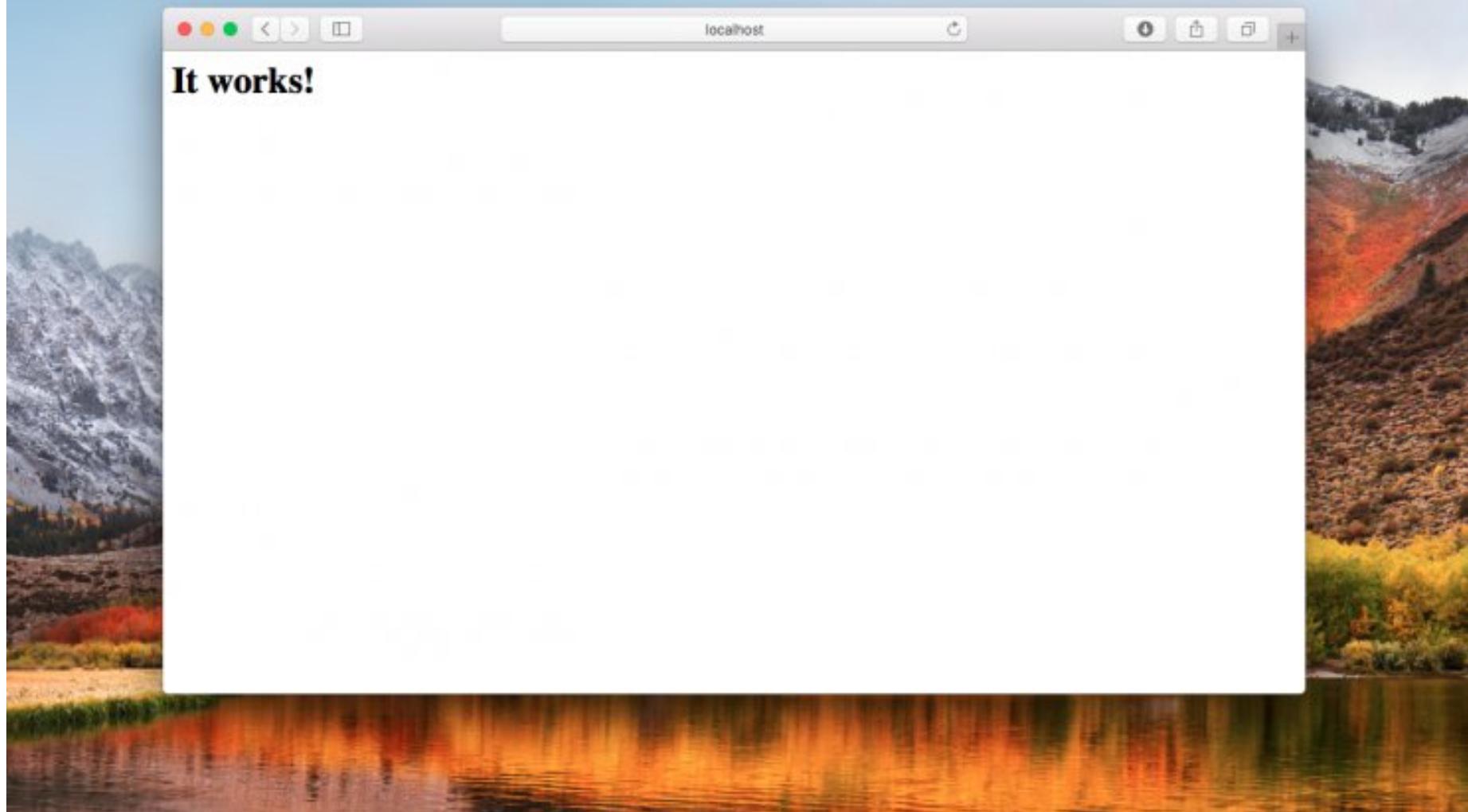
Sep 20, 2017

 [Table of Contents](#)

### Turn on Apache

1. Open Terminal by clicking on the magnifying glass at the top right corner of your screen and searching for Terminal
2. Type `sudo apachectl start` and press enter
3. Open Safari (or your browser of choice), type `localhost` in the address bar, and press enter

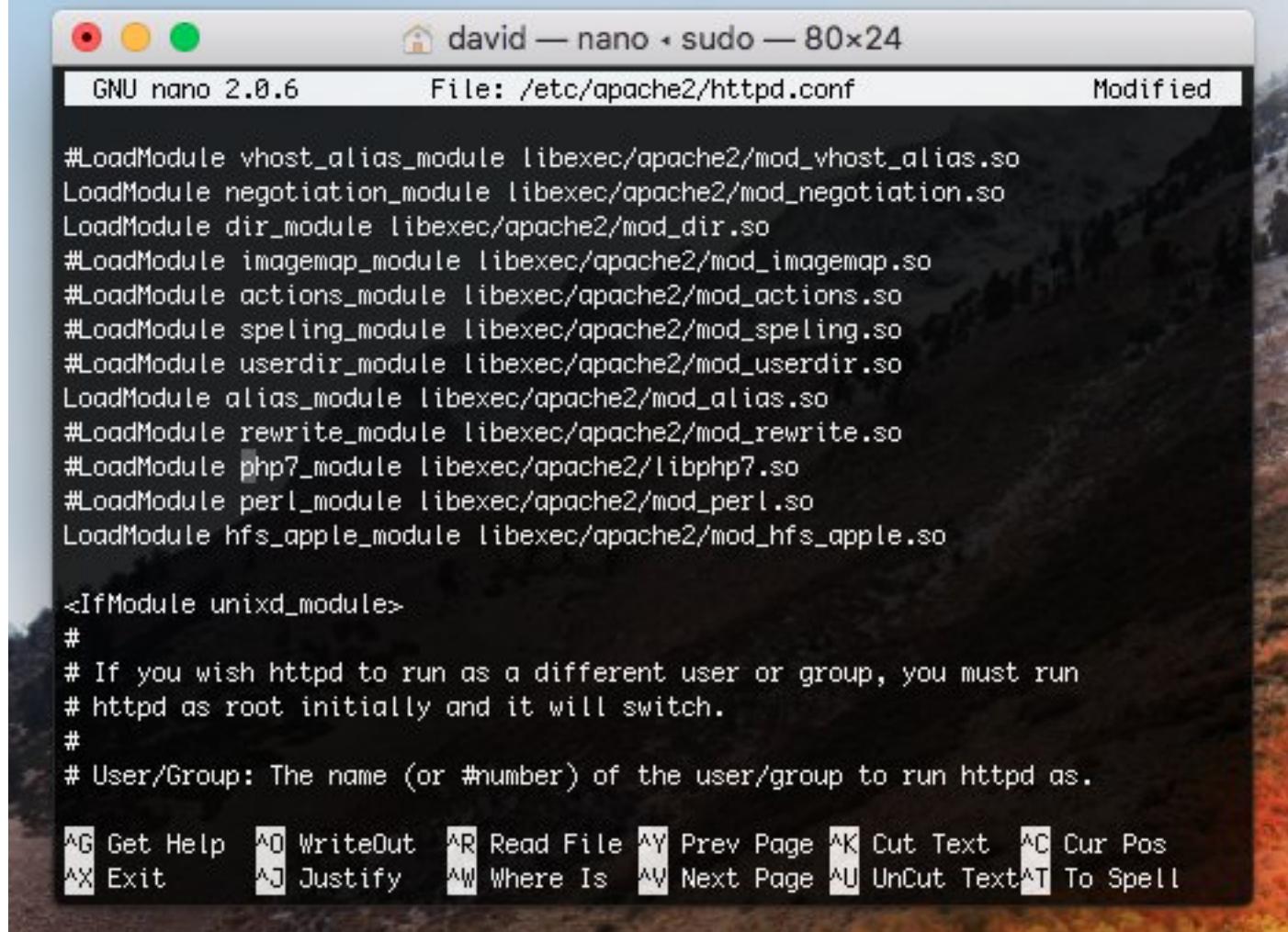
Can't believe it's that easy? It is! You should now see `It works!` in the browser.



## Turn on PHP

Mac already comes with PHP, so you don't need to install it from PHP.net. Before High Sierra (version 10.13), Mac came with PHP 5 installed. This caused users to get annoyed because it didn't come with PHP 7, so you'd have to go through extra work to upgrade from PHP 5 to PHP 7. Thankfully High Sierra already comes with PHP 7.1!!!

1. Open Terminal and type `sudo nano /etc/apache2/httpd.conf` and press enter
2. Press **Ctrl+W** which will bring up a search
3. Search for **php** and press enter. You'll see the following:



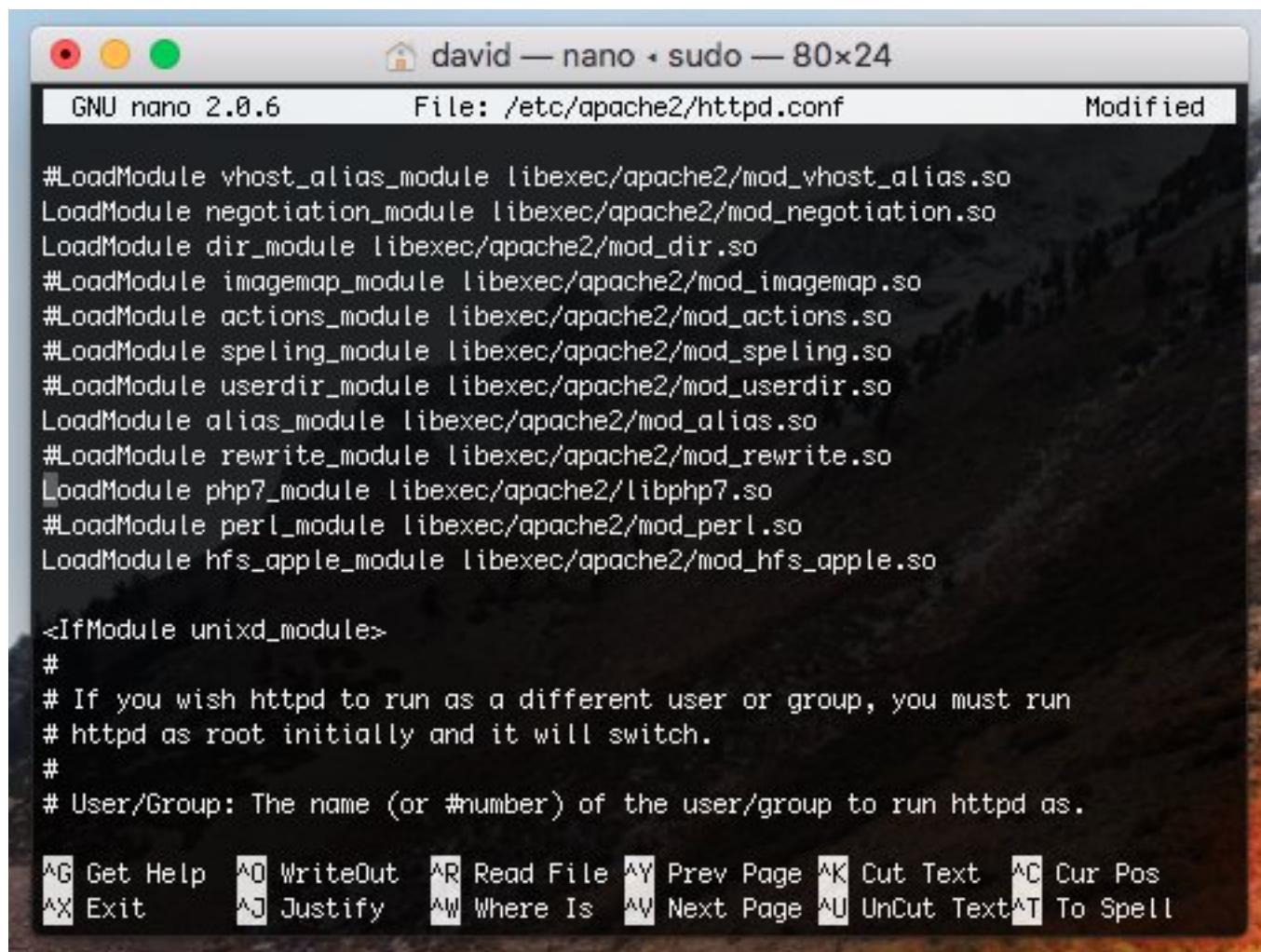
```
GNU nano 2.0.6           File: /etc/apache2/httpd.conf           Modified

#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
#LoadModule speling_module libexec/apache2/mod_speling.so
#LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
#LoadModule rewrite_module libexec/apache2/mod_rewrite.so
#LoadModule php7_module libexec/apache2/libphp7.so
#LoadModule perl_module libexec/apache2/mod_perl.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.

AG Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

4. Delete the # from `#LoadModule php7_module libexec/apache2/libphp7.so`



```
GNU nano 2.0.6           File: /etc/apache2/httpd.conf           Modified

#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
LoadModule imagemap_module libexec/apache2/mod_imagemap.so
LoadModule actions_module libexec/apache2/mod_actions.so
LoadModule speling_module libexec/apache2/mod_speling.so
LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
LoadModule php7_module libexec/apache2/libphp7.so
#LoadModule perl_module libexec/apache2/mod_perl.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so

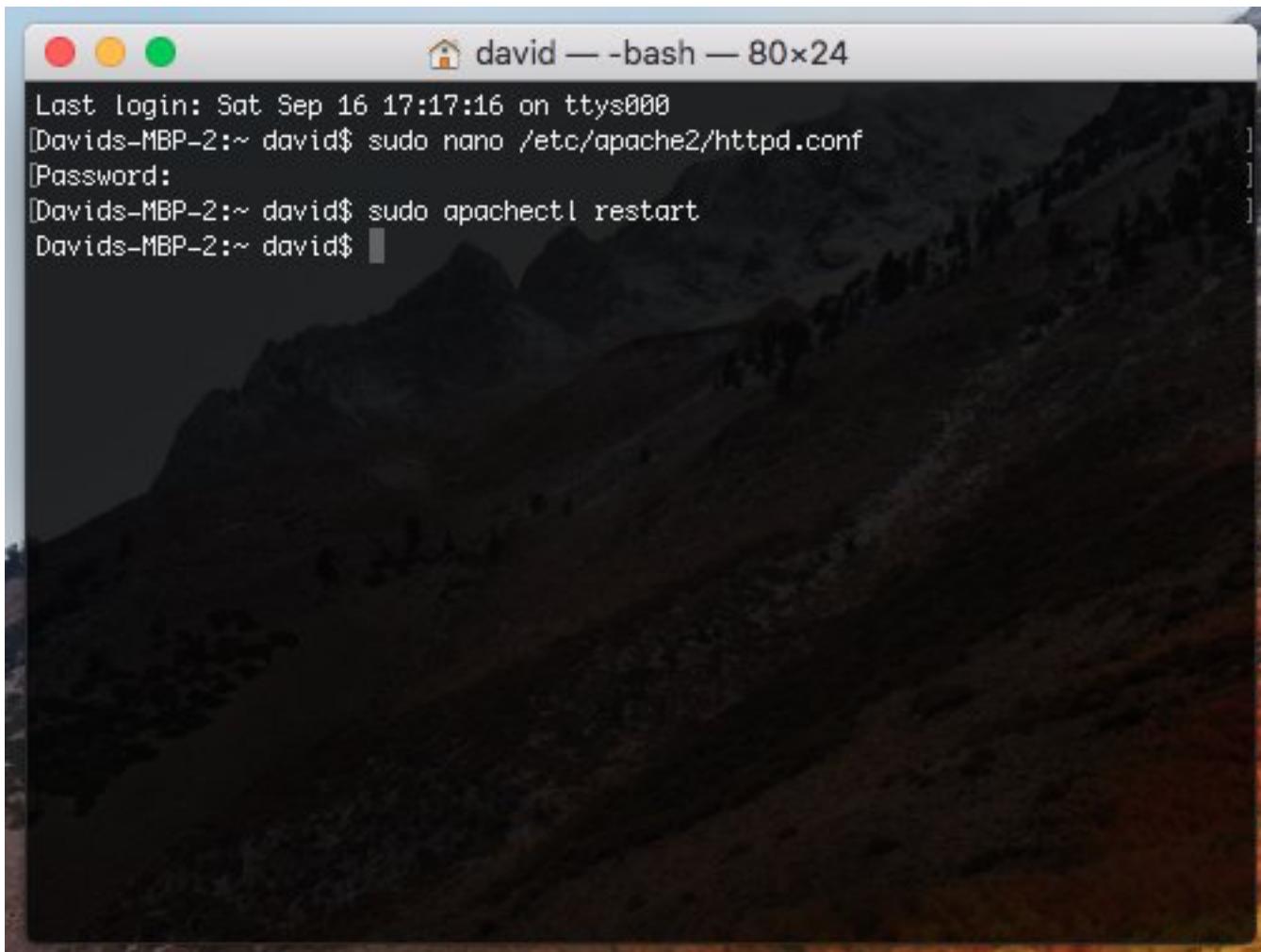
<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.

AG Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

5. Press **Ctrl+O** followed by Enter to save the change you just made

6. Press **Ctrl+X** to exit nano

7. Type sudo apachectl restart and press enter

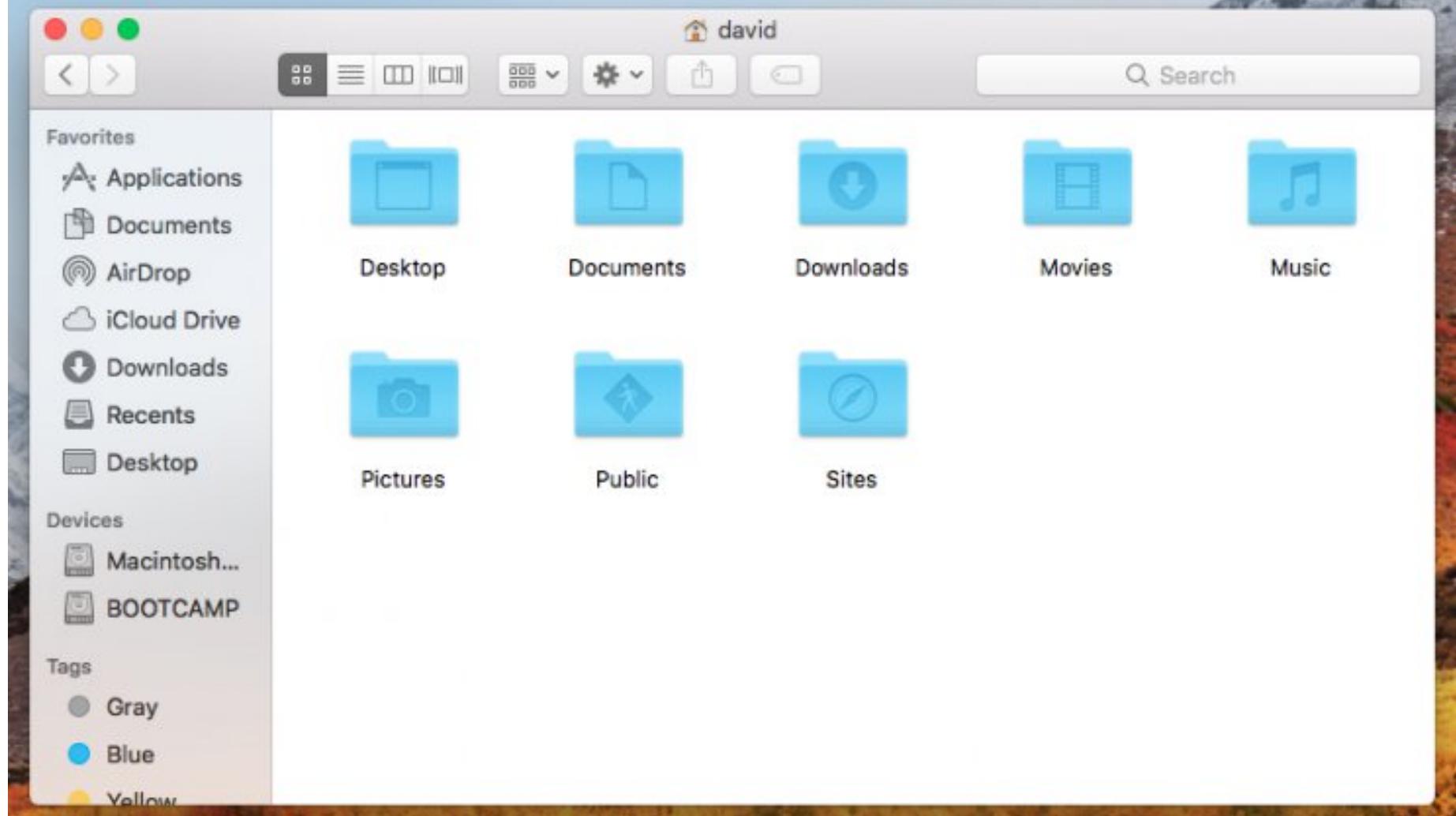


```
Last login: Sat Sep 16 17:17:16 on ttys000
[Davids-MBP-2:~ david$ sudo nano /etc/apache2/httpd.conf
>Password:
[Davids-MBP-2:~ david$ sudo apachectl restart
Davids-MBP-2:~ david$ ]
```

You just turned PHP 7 on. Great work so far!

## Create Sites Folder

1. Click on Finder at the bottom left corner of your screen and click on **Go > Home** the top navigation bar. This will take you to your home directory, which will also be whatever your computer is named; in this case it's **david**.
2. Create a new folder and name it **Sites** (The Safari icon shown below gets added automatically to the Sites folder as shown below)



3. Open your favorite text editor and create a file called **index.php** with the following code:

```
1 | <?php  
2 | echo "Hello From Sites Folder!";  
3 | phpinfo();  
4 | ?>
```

4. Save **index.php** in the **Sites** folder you created

5. Go back to Terminal and enter `sudo nano /etc/apache2/httpd.conf`

6. Press **Ctrl+W** to bring up search

7. Search for **Library** and press enter. You should see this:

```
GNU nano 2.0.6           File: /etc/apache2/httpd.conf

# you might expect, make sure that you have specifically enabled it
# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/Library/WebServer/Documents"
<Directory "/Library/WebServer/Documents">
    #
    # 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.
    #

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page ^U Uncut Text ^T To Spell
```

8. Replace **both** occurrences of `/Library/WebServer/Documents` with `/Users/david/Sites` (instead of david use your name which can be found at the top of your terminal next to the home icon)

```
GNU nano 2.0.6           File: /etc/apache2/httpd.conf           Modified

# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/Users/david/Sites"
<Directory "/Users/david/Sites">
    #
    # 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"
    #

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page ^U Uncut Text ^T To Spell
```

9. Press **Ctrl+O** followed by Enter to save these changes

10. Press **Ctrl+X** to exit nano

11. Type `sudo apachectl restart` and press enter

Go back to Safari and refresh the localhost page and you'll see Hello From Sites Folder! with PHP info that shows PHP 7 is being used.

The screenshot shows a web browser window with the address bar set to "localhost". The main content area displays the PHP info page. At the top, it says "Hello From Sites Folder!" and "PHP Version 7.1.7". On the right, there is a large "php" logo. Below this, there is a table with the following data:

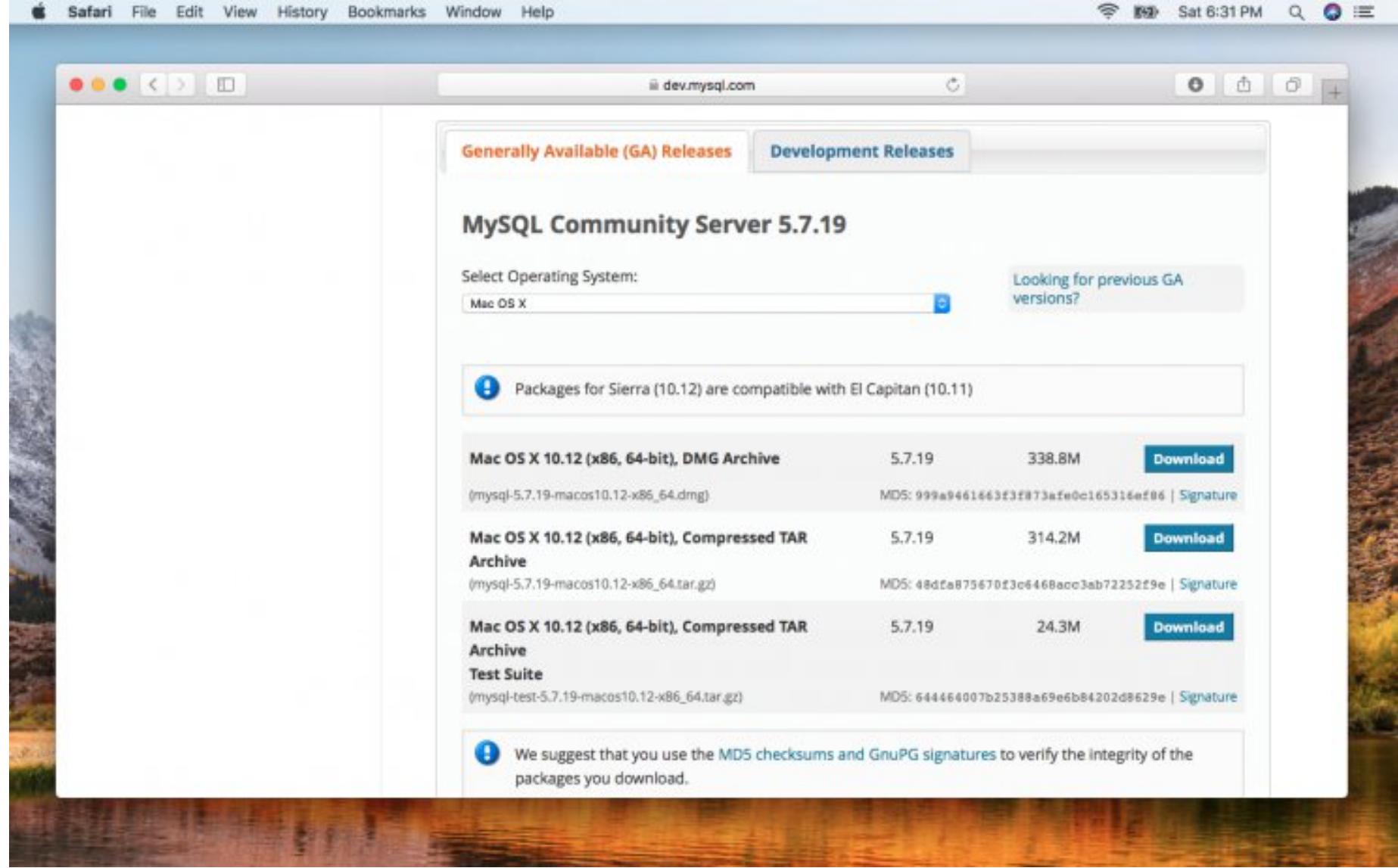
System	Darwin Davids-MBP-2 17.0.0 Darwin Kernel Version 17.0.0: Thu Aug 24 22:01:05 PDT 2017; root:xnu-4570.1.46~3RELEASE_X86_64 x86_64
Build Date	Jul 15 2017 18:01:36
Configure Command	/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Objects/php/configure' '--prefix=/usr' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--disable-dependency-tracking' '--sysconfdir=/private/etc' '--with-libdir=lib' '--enable-cli' '--with-iconv=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-config-file-path=/etc' '--with-libxml-dir=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-openssl=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr/local/libressl' '--with-kerberos=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-zlib=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-bcmath' '--with-bz2=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-calendar' '--disable-cgi' '--with-curl=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-dba' '--with-ndbm=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-exif' '--enable-fpm' '--enable-ftp' '--with-gd' '--with-png-dir=/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Root/usr/local' '--with-jpeg-dir=/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Root/usr/local' '--enable-gd-native-ttf' '--with-icu-dir=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-ldap=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOS

## Install MySQL

1. Go to <https://dev.mysql.com/downloads/mysql> (<https://dev.mysql.com/downloads/mysql>)

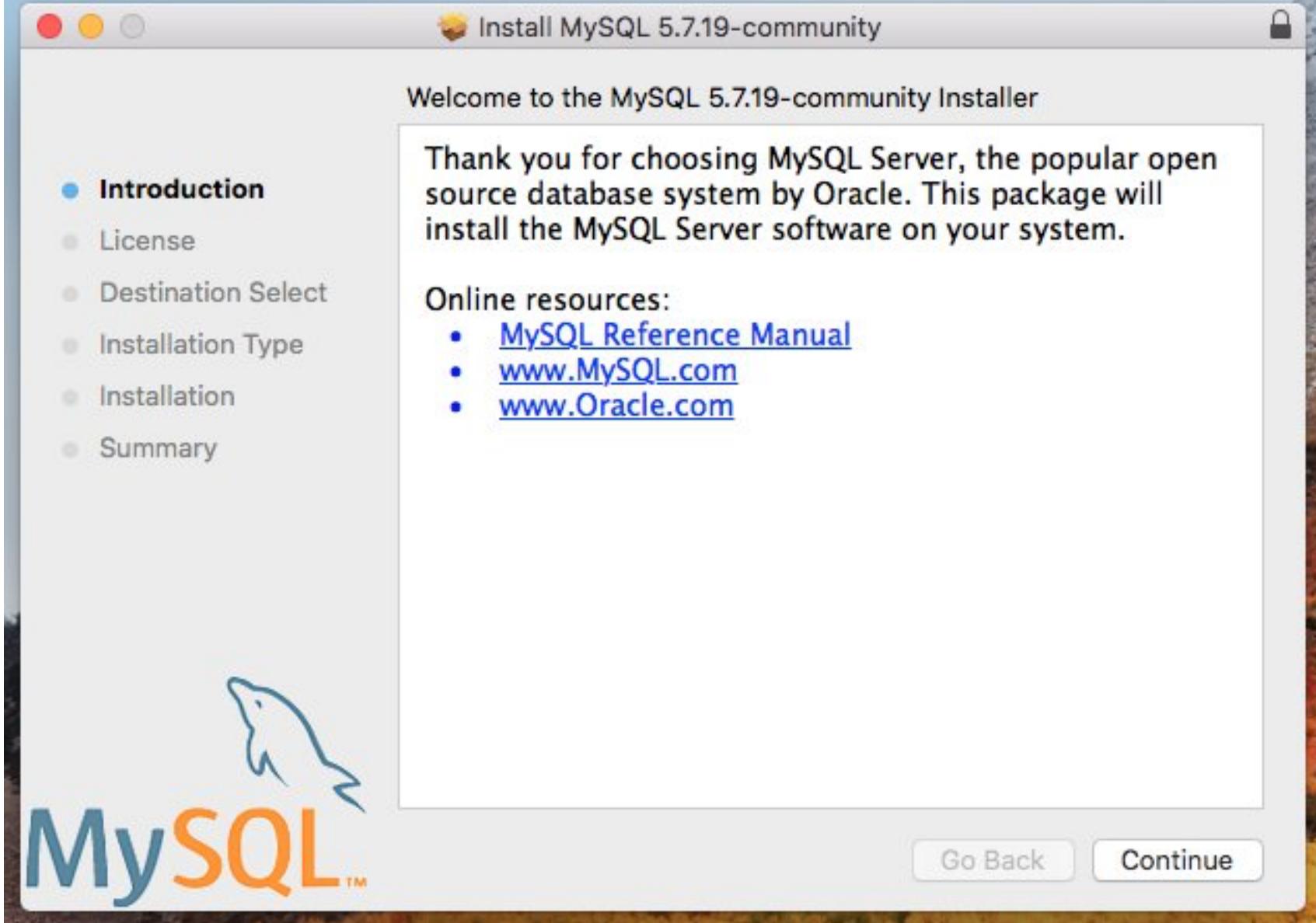
2. Scroll down until you see three download buttons and click on the one next to DMG

Archive

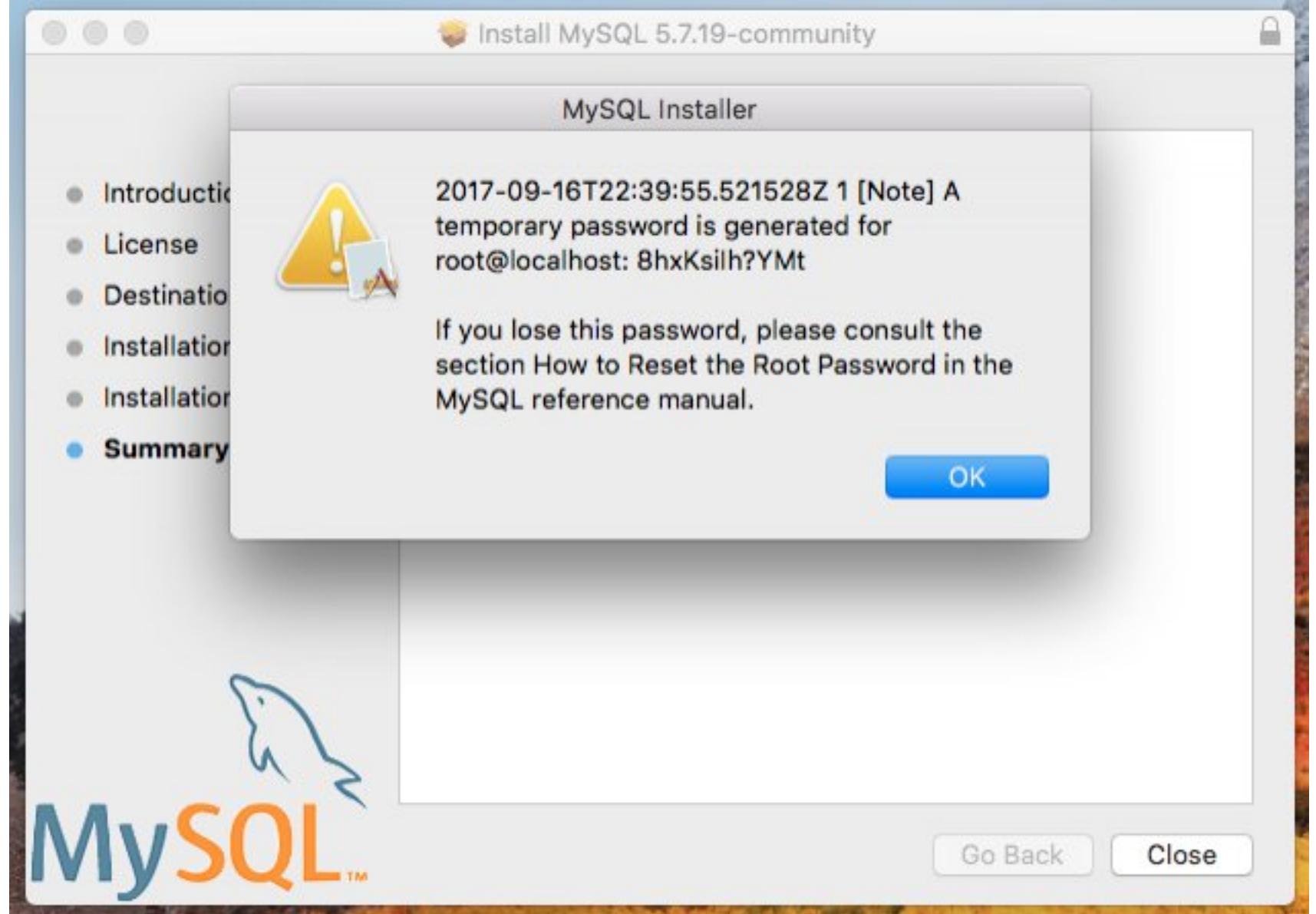


3. Scroll down and click **No thanks, just start my download**

4. Click on the downloaded file, double click the pkg file, and go through the installation which should look like this:



5. Once you get to the end of the installation, you'll see a popup that gives you your MySQL username (root) and password (8hxKsilh?YMt). Your password will be different. Copy the password to text file (or take a screenshot) and press okay. **Do this now, as it will be pain to reset the password if you don't know it.**



6. Press the Apple logo at the top left of your screen and go to System Preferences
7. Click on MySQL and you'll see that it's turned off



8. Press **Start MySQL Server** to turn it on and you'll see:



9. Go back to Terminal and type `sudo /usr/local/mysql/bin/mysql -u root -p`
10. When you press enter it will ask you for your Password. This is the password you use when you login to your Mac
11. Then it says Enter Password: which is `8hxKsilh?YMt` for me (you should use the password you copied earlier instead of `8hxKsilh?YMt`). You'll see this screen:

A screenshot of a terminal window titled "david — mysql • sudo — 80x24". The window shows the MySQL monitor starting up. It displays the last login information, the command used to start the MySQL monitor, and the MySQL connection details. It then shows the welcome message, the server version (5.7.19), and the copyright notice. Finally, it provides help instructions and the MySQL prompt "mysql>".

```
Last login: Sat Sep 16 19:08:31 on ttys000
[Davids-MBP-2:~ david$ sudo /usr/local/mysql/bin/mysql -u root -p
>Password:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.7.19

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

12. Type ALTER USER 'root'@'localhost' IDENTIFIED BY 'newpassword'; and press enter

```
Last login: Sat Sep 16 19:25:52 on ttys000
[Davids-MBP-2:~ david$ sudo /usr/local/mysql/bin/mysql -u root -p
>Password:
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.7.19

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

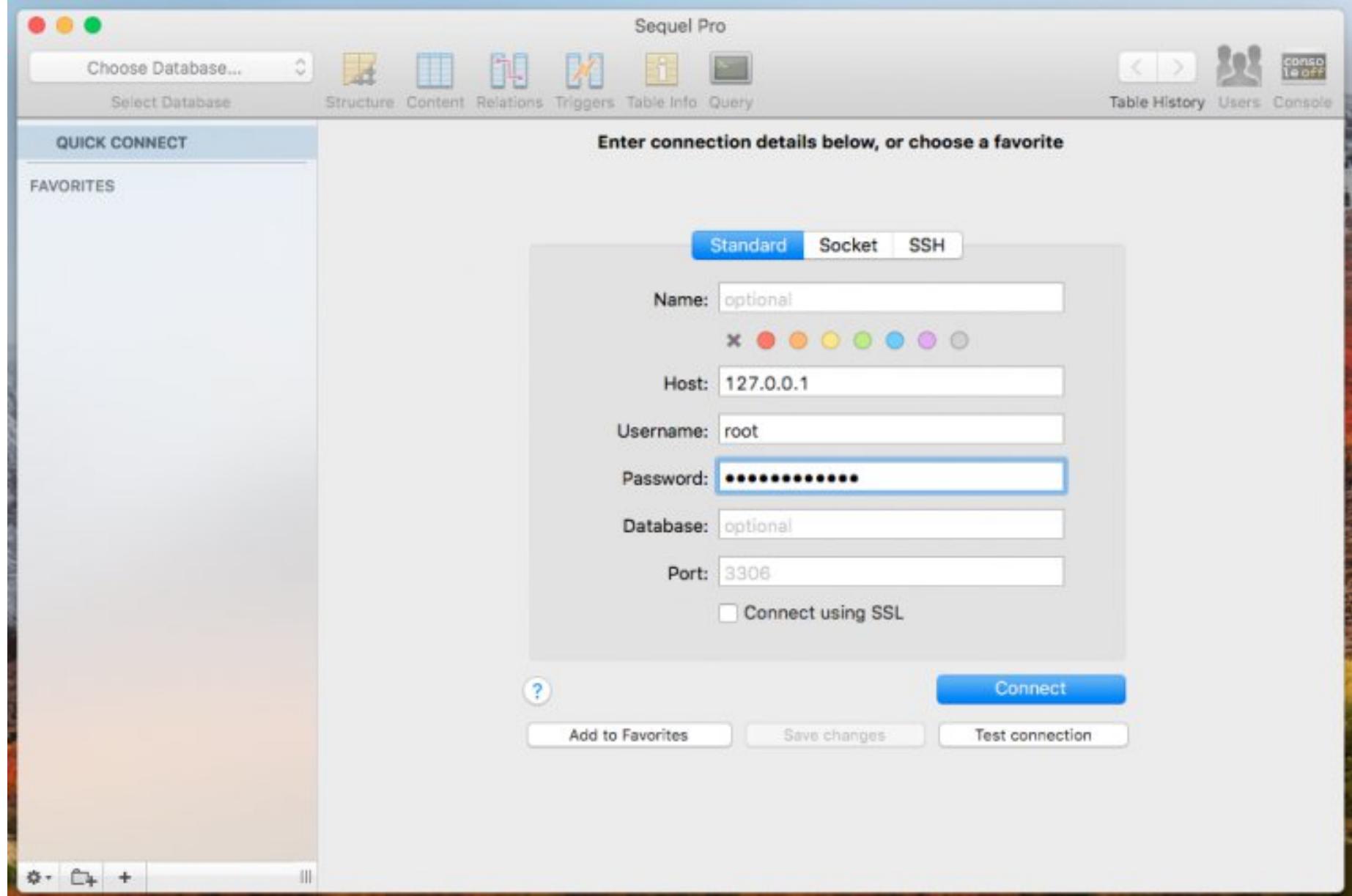
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

[mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'newpassword';
Query OK, 0 rows affected (0.05 sec)

mysql>
```

13. Go to <https://www.sequelpro.com> (<https://www.sequelpro.com>) and download Sequel Pro. This is like phpMyAdmin, but better because it's a desktop app.

14. Go through the installation, open Sequel Pro, and enter **127.0.0.1** for the Host. Enter **root** for the Username and **newpassword** for the Password. Press Connect



15. Click **Choose Database...** at the top left followed by **Add Database...**
16. Give it the name **mydb** and press enter
17. Then click the plus sign at the bottom left to create a new table. Name the table **mytable** and press add
18. Click the plus sign right above the word INDEXES and name the field **message**. Set the type to VARCHAR and give it a length of 200.
19. Click on **Content** at the top.
20. Click the plus sign at the bottom (the one to the left of the minus sign)
21. Give the new entry an id of **1** and set the message to **MySQL works perfectly!**

The screenshot shows the MySQL Workbench interface. At the top, it displays the connection information: (MySQL 5.7.19) root@127.0.0.1/mydb/mytable. Below this is a toolbar with icons for Select Database, Structure, Content, Relations, Triggers, Table Info, Query, Table History, Users, and Console. The 'Content' tab is selected. On the left, there's a 'TABLES' list with 'mytable' selected. The main pane shows a table with two columns: 'id' and 'message'. A single row is present with id 1 and message 'MySQL works perfectly!'. Below the table, the 'TABLE INFORMATION' section lists various details about the table, such as created date, engine type (InnoDB), and number of rows (1). The bottom of the window has a toolbar with various icons and a status bar indicating '1 row in table'.

22. Open **index.php** in your Sites folder and change the code to

```
1 <?php
2 $con = new mysqli("127.0.0.1", "root", "newpassword", "mydb");
3 $message = $con->query("SELECT message FROM myTable")->fetch_object()->message;
4 $con->close();
5 echo "$message <br/>";
6 echo "Hello From Sites Folder!";
7 phpinfo();
```

Obviously this example is immune to SQL injection, since there are no user-inputted values. However, if you don't understand MySQLi prepared statements as well as you'd like, check out this post (<https://websitebeaver.com/prepared-statements-in-php-mysqli-to-prevent-sql-injection>) for an excellent tutorial on how to prevent SQL injection 😊. There's even a PDO version (<https://websitebeaver.com/php-pdo-prepared-statements-to-prevent-sql-injection>), if you prefer that more.

23. Refresh localhost on Safari and you should see:

System	Darwin Davids-MBP-2 17.0.0 Darwin Kernel Version 17.0.0: Thu Aug 24 22:01:05 PDT 2017; root:xnu-4570.1.46~3RELEASE_X86_64 x86_64
Build Date	Jul 15 2017 18:01:36
Configure Command	/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Objects/php/configure' '--prefix=/usr' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--disable-dependency-tracking' '--sysconfdir=/private/etc' '--with-libdir=lib' '--enable-cgi' '--with-iconv=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-config-file-path=/etc' '--with-libxml-dir=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-openssl=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr/local/libressl' '--with-kerberos=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-bz2=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-calendar' '--disable-cgi' '--with-curl=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-dba' '--with-ndbm=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--enable-exif' '--enable-fpm' '--enable-ftp' '--with-gd' '--with-png-dir=/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Root/usr/local' '--with-jpeg-dir=/BuildRoot/Library/Caches/com.apple.xbs/Binaries/apache_mod_php/install/TempContent/Root/usr/local' '--enable-gd-native-ttf' '--with-icu-dir=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-ldap=ldap:/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-sasl=sasl:/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr' '--with-libedit=/BuildRoot/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.13.Internal.sdk/usr'

Great! You now have MySQL working, so you can use localhost to store information in databases.

## Make SEO Friendly URLs Work

Instead of going to **localhost/profile.php?user=david**, it looks nicer when you can go to **localhost/profile/david**. Let's make this work by enabling **mod\_rewrite** so you can use **RewriteRules**.

1. Go to Terminal and type `sudo nano /etc/apache2/httpd.conf` followed by pressing enter
2. Press **Ctrl+W**, type **rewrite**, and press enter
3. Remove the **#** in `#LoadModule rewrite_module libexec/apache2/mod_rewrite.so`

```
GNU nano 2.0.6          File: /etc/apache2/httpd.conf

#LoadModule asis_module libexec/apache2/mod_asis.so
#LoadModule info_module libexec/apache2/mod_info.so
#LoadModule cgi_module libexec/apache2/mod_cgi.so
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
#LoadModule speling_module libexec/apache2/mod_speling.so
#LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
LoadModule php7_module libexec/apache2/libphp7.so
#LoadModule perl_module libexec/apache2/mod_perl.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.

[ Wrote 546 lines ]

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page   ^K Cut Text
^X Exit         ^J Justify      ^W Where Is     ^V Next Page   ^U Uncut Text
^C Cur Pos      ^T To Spell
```

4. Press **Ctrl+O** followed by Enter to save

5. Press **Ctrl+X** to exit nano

6. Type `sudo apachectl restart`

Now you can use friendly URLs when you develop using localhost.

## Make .htaccess Work

This isn't necessary to make vanity URLs work, but I personally prefer using an .htaccess file, as it allows me version control it.

1. Go to Terminal and type `sudo nano /etc/apache2/httpd.conf` followed by pressing enter

2. Press **Ctrl+W**, type `AllowOverride controls`, and press enter

3. Change `AllowOverride None` to `AllowOverride All`

```
# 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 FollowSymLinks Multiviews
# MultiviewsMatch Any

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

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

#
^G Get Help      ^O WriteOut     ^R Read File     ^Y Prev Page    ^K Cut Text      ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is      ^N Next Page    ^U UnCut Text    ^T To Spell
```

4. Press **Ctrl+O** followed by Enter to save

5. Press **Ctrl+X** to exit nano

6. Type `sudo apachectl restart`

Now you can add a .htaccess file in the Sites folder and it will work well.

## Turn On SSL/HTTPS

Instead of accessing your website through localhost, you might want to access your website by going to `https://localhost`. If you turn on SSL, you'll see a padlock next to localhost in the URL bar.

1. Go to Terminal and type `sudo nano /etc/apache2/httpd.conf` followed by pressing enter
2. Press **Ctrl+W**, type **socache\_shmcb\_module**, and press enter

The screenshot shows a terminal window titled "david — nano — 91x26". The file being edited is "/etc/apache2/httpd.conf". The content of the file is a list of Apache modules, many of which are commented out with a '#' symbol at the beginning of the line. The modules listed include authz\_dbd, authz\_core, authnz\_ldap, access\_compat, auth\_basic, auth\_form, auth\_digest, allowmethods, file\_cache, cache, cache\_disk, cache\_socache, socache\_shmcb, socache\_dbm, socache\_memcache, watchdog, macro, dbd, dumpio, echo, and buffer. At the bottom of the screen, there is a menu bar with various keyboard shortcuts for navigating the nano editor.

```
GNU nano 2.0.6 File: /etc/apache2/httpd.conf

#LoadModule authz_dbd_module libexec/apache2/mod_authz_dbd.so
LoadModule authz_core_module libexec/apache2/mod_authz_core.so
#LoadModule authnz_ldap_module libexec/apache2/mod_authnz_ldap.so
LoadModule access_compat_module libexec/apache2/mod_access_compat.so
LoadModule auth_basic_module libexec/apache2/mod_auth_basic.so
#LoadModule auth_form_module libexec/apache2/mod_auth_form.so
#LoadModule auth_digest_module libexec/apache2/mod_auth_digest.so
#LoadModule allowmethods_module libexec/apache2/mod_allowmethods.so
#LoadModule file_cache_module libexec/apache2/mod_file_cache.so
#LoadModule cache_module libexec/apache2/mod_cache.so
#LoadModule cache_disk_module libexec/apache2/mod_cache_disk.so
#LoadModule cache_socache_module libexec/apache2/mod_cache_socache.so
#LoadModule socache_shmcb_module libexec/apache2/mod_socache_shmcb.so
#LoadModule socache_dbm_module libexec/apache2/mod_socache_dbm.so
#LoadModule socache_memcache_module libexec/apache2/mod_socache_memcache.so
#LoadModule watchdog_module libexec/apache2/mod_watchdog.so
#LoadModule macro_module libexec/apache2/mod_macro.so
#LoadModule dbd_module libexec/apache2/mod_dbd.so
#LoadModule dumpio_module libexec/apache2/mod_dumpio.so
#LoadModule echo_module libexec/apache2/mod_echo.so
#LoadModule buffer_module libexec/apache2/mod_buffer.so

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell
```

3. Delete the # from `#LoadModule socache_shmcb_module`

`libexec/apache2/mod_socache_shmcb.so`

4. Press **Ctrl+W**, type **mod\_ssl**, and press enter

5. Delete the # from `#LoadModule ssl_module libexec/apache2/mod_ssl.so`

6. Press **Ctrl+W**, type **httpd-ssl**, and press enter

7. Delete the # from `#Include /private/etc/apache2/extra/httpd-ssl.conf`

8. Press **Ctrl+O** followed by Enter to save

9. Press **Ctrl+X** to exit nano

10. Type `sudo nano /etc/apache2/extra/httpd-ssl.conf` and press enter

11. Press **Ctrl+W**, type **ServerName**, and press enter

```
david — nano — 131x34
GNU nano 2.8.6          File: /etc/apache2/extra/httpd-ssl.conf

# Define a relatively small cache for OCSP Stapling using
# the same mechanism that is used for the SSL session cache
# above. If stapling is used with more than a few certificates,
# the size may need to be increased. (AH01929 will be logged.)
#SSLStaplingCache "shmcb:/private/var/run/ssl_stapling(32768)"

# Seconds before valid OCSP responses are expired from the cache
#SSLStaplingStandardCacheTimeout 3600

# Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

## 
## SSL Virtual Host Context
## 

<VirtualHost _default_:443>

# General setup for the virtual host
DocumentRoot "/Library/WebServer/Documents"
ServerName www.example.com:443
ServerAdmin you@example.com
ErrorLog "/private/var/log/apache2/error_log"
TransferLog "/private/var/log/apache2/access_log"

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

AG Get Help      ^O WriteOut     ^R Read File     ^Y Prev Page    ^K Cut Text     ^C Our Pos
^X Exit         ^J Justify      ^W Where Is      ^V Next Page    ^U UnCut Text   ^T To Spell
```

12. Replace **www.example.com:443** with **localhost**

13. Right above you'll see **/Library/WebServer/Documents** . Replace that with  
**/Users/david/Sites** . (Use your name instead of david just like you did earlier)

14. Right underneath **<VirtualHost\_default\_:443>** add (but make sure to replace david with your name):

```
1 <Directory "/Users/david/Sites">
2   Options All
3   MultiviewsMatch Any
4   AllowOverride All
5   Require all granted
6 </Directory>
```

15. Your terminal should now look like this:

GNU nano 2.8.6

File: /etc/apache2/extr~~a~~/httpd-ssl.conf

Modified

```
# Define a relatively small cache for OCSP Stapling using
# the same mechanism that is used for the SSL session cache
# above. If stapling is used with more than a few certificates,
# the size may need to be increased. (AH01929 will be logged.)
#SSLStaplingCache "shmcb:/private/var/run/ssl_stapling(32768)"

# Seconds before valid OCSP responses are expired from the cache
#SSLStaplingStandardCacheTimeout 3600

# Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

## 
## SSL Virtual Host Context
## 

<VirtualHost _default_:443>

<Directory "/users/david/Sites">
    Options All
    MultiviewsMatch Any
    AllowOverride All
    Require all granted
</Directory>

# General setup for the virtual host
DocumentRoot "/Users/david/Sites"
ServerName localhost
ServerAdmin you@example.com
```

AG Get Help      ^O WriteOut      AR Read File      ^W Prev Page      ^K Cut Text  
AX Exit      AJ Justify      AW Where Is      ^N Next Page      AU Uncut Text      AC Cur Pos  
AT To Spell

16. Press **Ctrl+O** followed by Enter to save

17. Press **Ctrl+X** to exit nano

18. Type `sudo nano /etc/ssl/openssl.cnf` and press enter

19. At the bottom of **openssl.cnf** add this:

1   [ san ]	
2   subjectAltName	= DNS:localhost

20. Your terminal should look like this:

```
[ req ]  
#default_bits      = 2048  
#default_md        = sha256  
#default_keyfile   = privkey.pem  
distinguished_name = req_distinguished_name  
attributes         = req_attributes  
  
[ req_distinguished_name ]  
countryName          = Country Name (2 letter code)  
countryName_min      = 2  
countryName_max      = 2  
stateOrProvinceName  = State or Province Name (full name)  
localityName         = Locality Name (eg, city)  
0.organizationName   = Organization Name (eg, company)  
organizationalUnitName = Organizational Unit Name (eg, section)  
commonName           = Common Name (eg, fully qualified host name)  
commonName_max       = 64  
emailAddress         = Email Address  
emailAddress_max     = 64  
  
[ req_attributes ]  
challengePassword    = A challenge password  
challengePassword_min = 4  
challengePassword_max = 20  
  
[ san ]  
subjectAltName        = DNS:localhost
```

Get Help

^O WriteOut

X Exit

^R Read File

W Where Is

^Y Prev Page

N Next Page

^K Cut Text

U Uncut Text

^C Cur Pos

T To Spell

21. Press **Ctrl+O** followed by Enter to save

22. Press **Ctrl+X** to exit nano

23. Type

```
sudo openssl req -extensions san -config /etc/ssl/openssl.cnf -x509 -nodes -newkey rsa
```

and press enter. (You can use your information instead, but don't change localhost)

24. Type

```
sudo security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain /
```

and press enter

25. Type sudo apachectl restart

26. Now go to **https://localhost** in your browser, and you'll see a padlock next to the url which means you're done setting up SSL!

As always, if you have any questions please leave them in the comments, and we'll do our best to help you.

---

---

## **Author - David Marcus**

---