LAB 01

- Change the verbosity level of the errors logged the error.log log file of nginx to be info. Try accessing a page that does not exist on a website served by nginx. Now change the verbosity to be emerg and do the same action. Observe the difference.

LAB 02

1. Create a new configuration file myserver.conf under /etc/nginx/conf.d.
2. Ensure that nginx does not load the default.conf file and load myserver.conf file instead.
3. Inside the file, create a definition for two websites: drupal and magento.
4. Configure the server name for drupal to be drupal.example.com and for magento to be magento.example.com
5. Create the necessary directories for both websites.
6. Inside each directory, place two files: index.html and home.html
7. Add some content to both files to be recognizable.
8. Reload nginx (do not restart).
9. Change your hosts file (or update your DNS server if you are using one) to map drupal.example.com and magento.example.com to the appropriate IP address of nginx.
10. Navigate to drupal.example.com and magento.example.com. Make sure that index.html contents of each website are displayed.

LAB 03:

- Change the default page for magento to be home.html instead of index.html. Reload nginx and ensure that the new default page is displayed when you navigate to magento.nginx.com

LAB 04:

- Ensure that nobody can access the nginx server using it's public IP. Only localhost or 127.0.0.1 can be used to access the server. (Hint: magento and wordpress are already accessed using their appropriate addresses so you don't need to make changes to them).

LAB 05:

1. Add some images to drupal website (jpg, png, gif…etc.)
2. Name one of the images: allowed.jpg.
3. Create a new directory inside drupal, call it "special" and place some images inside it.
4. Configure nginx so that users trying to access JPG files on drupal (case insensitive) should receive a 402 error.
5. Users trying to access allowed.jpg should be able to view the image.

LAB 06

1. Modify the configuration done in the previous lab so that users trying to access images under the “special” directory should receive 403 error.

LAB 07

1. Create a new file, fallback.html inside magento directory. Place some content inside this file to be recognizable.
2. Modify nginx configuration so that when a user cannot find a URL on magento, fallback.html contents should be displayed.

LAB 08:

- Create a new rewrite rule for drupal so that when a user navigates to drupal.example.com/modules, the server should receive the request as if it was drupal.example.com/index.html?id=modules.

LAB 09:

1. Create a friendly error message page, called friendlyerrors.html inside magento directory.
2. Instruct nginx to serve this friendlyerrors.html file whenever an error code that starts with 4 is encountered (that is 404, 403, 406…etc.).

LAB 10:

- Ensure that users can access the “special” directory under drupal and list the files inside it.

LAB 11:

- Ensure that users are totally denied access from magento website.