COSC349 Assignment 1

Elliot Thompson
ID 9194588 – Username thoel044

The repository on GitHub is found here: https://github.com/elliot4600/COSC349Assignment1

This project aims to use multiple connected virtual machines to allow the public to view a list of books available within a public community library. The book list can be added and changed by the admin which is reflected in the database and therefore, also the book list.

This application uses vagrant to build three different VMs with differing functions. The application makes use of three virtual machines, a database server, and two webservers. One webserver is a public resource where people can view the books currently available within the library. The other webserver allows the organizers of the library to add new books to the database. The database server stores the available books within an SQL database.

Websiteserver	Allows addition of new books to SQL database via a PHP form.
|	
Database Server	Stores book list in an mySQL database
|	
Query Server	Lists the books available from querying Database

The websiteserver and queryserver do not directly interact at any point. The websiteserver is developed to assist with management of the database contents through a simpler interface than the mySQL terminal commands.

All VM's interact over a private network which has been created by the configuration specified within the Vagrantfile (the configuration file which is ran when **vagrant up** is called to initialize the VM network). PHP is used for the webservers dynamic forms which interact with the database.

Setup notes:

Please note: The download sizes mentioned here were accurate as of October 2020.

I chose to implement Ubuntu (Xenial 64 bit) as the OS for our virtual machines. This will only need to be downloaded once for the first time running of this application, as the file will be stored locally on the host machine. This will reduce the time taken to recreate the application for developers when implementing new builds which require reconfiguration of the VMs. As Ubuntu is a common choice for VMs, this may allow for easier troubleshooting if problems are experienced due to the widespread use of the software. The download size of this is approximately 250-300MB.

When calling **apt-get-update** and **apt-get upgrade y** within the config files, will install the most recent updates for the appropriate software for the Ubuntu distribution. The updates downloaded for the webservers are approximately 15 MB, and the database updates are approximately 40MB.

The initial build takes between 5-6 minutes to construct when creating the entire application for the first time (by using **vagrant up** on the repository). When rebooting the application from powered down VMs, it takes approximately 1.5 minutes.

Usage instructions:

This VM application makes use of the following tools to run. They can be downloaded from the links below, and their approximate download size is also displayed (for the latest versions as of September 2020).

Vagrant:
https://www.vagrantup.com/downloads
Download size: 30-40MB for Mac OS X & Linux, ~240MB for Windows (64-Bit)

Virtualbox:
https://www.virtualbox.org/wiki/Downloads
Download size: 100-120MB

The repository then needs to be either cloned or downloaded and is stored on github.

To clone the repository using git with this command:

```
git clone https://github.com/elliot4600/COSC349Assignment1.git
```

Alternatively, this repository can be downloaded as a .zip folder from:
https://github.com/elliot4600/COSC349Assignment1/archive/master.zip

After cloning/downloading the repository, navigate to the directory of the repository. Run the command below to initialise the VM's using vagrant:

```
vagrant up --provider virtualbox
```

The public can access the library contents page within a web browser at the URL:

http://127.0.0.1:8081

The library administration can edit the database contents and add new books within a web browser at the URL:

http://127.0.0.1:8080

For database and administration purposes, the username and password for the database are both "**admin**".

Possible modifications:

A developer could further extend this application through modification of the existing code. After changes are made, the application must be rebuilt. To rebuild the application, the VMs created through vagrant must be destroyed and rebuilt with the following commands in terminal:

```
Vagrant global-status          (Provides ID's of all VMs constructed with vagrant)
vagrant destroy [id of VM]     (Calling the destroy method on the unique ID of a machine)
vagrant up                     (Called from within the location of the repository)
```

Vagrant reload is the commend used to power-down and power-up the VMs, however sometimes the changes made to the code, particularly with regards to IP allocation via vagrant, require the VMs to be destroyed and recreated in order to properly show the changes made.

Possible changes other developers could implement within this application involve extension of the administration/content addition form webpage, as well as modifications to allow more interaction from the public with the query server.

At present, this application can only add new books to the database, which will be displayed on the query table to the public. An extension could be to integrate further interaction with the mySQL database through additional forms to prevent any need to connect to the mySQL database via SSH. Implementation of a removal function for existing books, and modification to existing books to correct if information has been entered incorrectly.

For the public site, the table queries the mySQL database and displays the book name & author pairings in a table which is read-only. Submission forms may help the community to more directly interact with the library project. Examples of these forms could be forms which allow requests for new books or requesting the library to put a hold on a book for a specific member. For these extensions, additional tables could be added to the mySQL database, which store input from the user in a similar vein to the form used to add books to the availablebooks table within the database.

Additional notes:

When completing this assignment, an odd error was encountered sometimes when creating the VMs using **vagrant up**, in which the mySQL database was not properly initialized. If this error is encountered, I found that accessing the database VM via SSH, then logging into the mySQL service allowed for testing if the table storing available books has been properly initialized.

To do so, call the following commands from the same location as the VMs.

```
Vagrant ssh dbserver
mysql –u admin –p
```

(The password is **admin** when prompted)

You should now be accessing the mySQL service.
From here, **SHOW DATABASES** will list all current databases. Navigate into the desired database by **USE** [database name]. The tables can then be viewed by calling **SHOW TABLES**.