

# Assignment 1 - Advanced Programming

# Submitted to: Dr Sidrah Sultana

# Submitted by: Muhammad Zakiullah Usman, BSCS 7A, 208655

# Exploring Open Source Software

# Introduction

I choose the eCommerce website type to work on. Ecommerce websites have long interested me and I really want to take the opportunity provided by this assignment to explore it as much as I can. So far, I have a long experience of using eCommerce websites like Daraz and Aliexpress. As a result, I am very familiar to the user experience of these kinds of websites.

# Why I Chose Magento?

I already knew a little bit about other OSS which facilitate in the creation of eCommerce websites like Spotify and the Wordpress plugin called WooCommerce. I wanted to try something new. I searched for the top 10 OSS for developing eCommerce websites and found Magento. Upon exploring it a little, I found that it is in development by Adobe. Adobe’s reputation made me choose this over others. Moreover, it had a free Community version available with demoes that I downloaded to explore.

# Step 1: Configuration

I already had Wamp server installed on my laptop along with an understanding on what it does and how to use it. After downloading Magento, I placed the files in the ‘www' folder.

The purpose of Wamp is to set up a local server on which to host the website. When running a website locally, it won’t be accessible from any other system.

As per instructions, I learned that first I will need to create a database for my site. To achieve this, I opened the Phpmyadmin tool from the Wamp localhost. I created the database by the name of ‘Magento’. The next step was to change some settings of Wamp server to ensure an error-free installation. I activated the curl php extension and the Apache rewrite module.

The next step was to make some changes in the ‘php.ini’ file. The objective was to increase the maximum execution time and I set it to 1500 seconds. I did the same for the max input time. Finally, all was set to install Magento.

After opening Magento from Project Files, an installation wizard guided the process. The installation completed and the option of launching Magento admin became visible. I got separate URL(s) for both access as a user and admin access.

However, after several attempts I could not access either of the URLs, so I resorted to another option.

Since I already had bought a hosting account from GoDaddy, I wanted to see if I could run Magento on a registered domain. So instead of using local server like Wamp or Xamp, I used my hosting account.

To install, I navigated to cPanel and selected Magento to install. I learned that it was very similar to installing Wordpress on a hosting. In this method, I did not have to set up the database. Overall the process was really simple. All that was needed was to click install. Unlike the local server method which required several edits in the php.ini and also did not work.

I learned that in order to keep it separate, I should install it in a different directory so it does not interfere with the existing WooCommerce application. I installed it at:   
[www.shopieshub.com/ecommerce/](http://www.shopieshub.com/ecommerce/) (site is live)

# Step 2: Exploring as a user

Before making any changes, I opened the website to explore its basic design and features as a user. A user typically only interacts with the frontend, so website was opened without logging in as admin.

The site had a top bar which contained a placeholder for message holder and an option to login and register. I learned that all the functionality for creating a user id was implemented as a basic form. There is also a search bar with search functionality implemented.

Other than that there is an option to subscribe email address.

# Step 3: Exploring the database structure as a database administrator

I signed in the admin URL with the admin credentials to explore the database structure. But this was a dashboard to manage the site, so I opened the backend.

There are a lot of tables in the PhpMyAdmin accessed through cPanel. Each handling a separate aspect of the entire interaction with eCommerce stores. There are various relationships within the tables that maintains the database so that appropriate changes are reflected in all relevant tables.

I learned that most of the database type was of InnoDB. I could make any changes in any table, for example to manually add a user.

# Step 4: Exploring the coding structure as a coder

Unlike in Wordpress, there was no option to customize theme and view all the code in the magento dashboard. To view the basic code, I opened the website and used ‘inspect element’.

I studied the code and saw the usage of several web technologies like javascript, html, css and ajax, among others. From the sources tab, its file system revealed folders for storing images, css, jquery, fonts. All these folders are for easy readability and management of the code.

# Step 5: Adding plugins for importing functionality

I browser for the Geo Lock extension which provided the functionality to restrict certain IP addressing from accessing my website. It also shows where the visitors are accessing a site from and can restrict users based on location.

I intended to install it through the Extensions manager. The process is not as straightforward as compared with Wordpress. It required the creation of an Access key from the authorized marketplace before allowing installation. Afterwards a readiness check started which failed in several places. As a result, I could not complete the installation.