



ADVANCED PROGRAMMING

Assignment # 1



Hafiz Muhammad Yaseen

Registration # 222192

BSCS 7-B

SEPTEMBER 16, 2019

Magento 2

Magento is a feature-rich, professional open source e-commerce web application that was launched on March 31, 2008. Magento offers merchants complete flexibility and control over the look, content and functionality of their online store. Magento's intuitive administration interface contains content management, powerful marketing and merchandising tools to give merchants the ability to create sites that are fully fit their unique business needs, putting no constraints on business processes and flow.

When Magento was released in 2008, it transformed the ecommerce platform market. In a period of 18 months, Magento surpassed one million downloads and now records billions of dollars of transactions worldwide by tens of thousands of merchants. Purchased by eBay in early 2011, it continues to be the fastest growing ecommerce platform in the world. There is some amazing information of Magento:

- Over 25% of online businesses in the world choose Magento as their platform.
- Over \$25 billion transaction was made each year.
- Over 300 solution & industry partners in the world.

Here are few reasons for people to choose Magento:

- * Easily to control and manage your online store.
- * Flexible to meet your unique needs.
- * Enterprise edition features.
- * Scalable to grow with your business.
- * Thousands of Magento extensions to meet your requirements.

System Requirements to install Magento:

Our system requirements include required software to run the Magento application, supported web browsers, and also recent requirements to securely connect with our software repository and with PayPal.

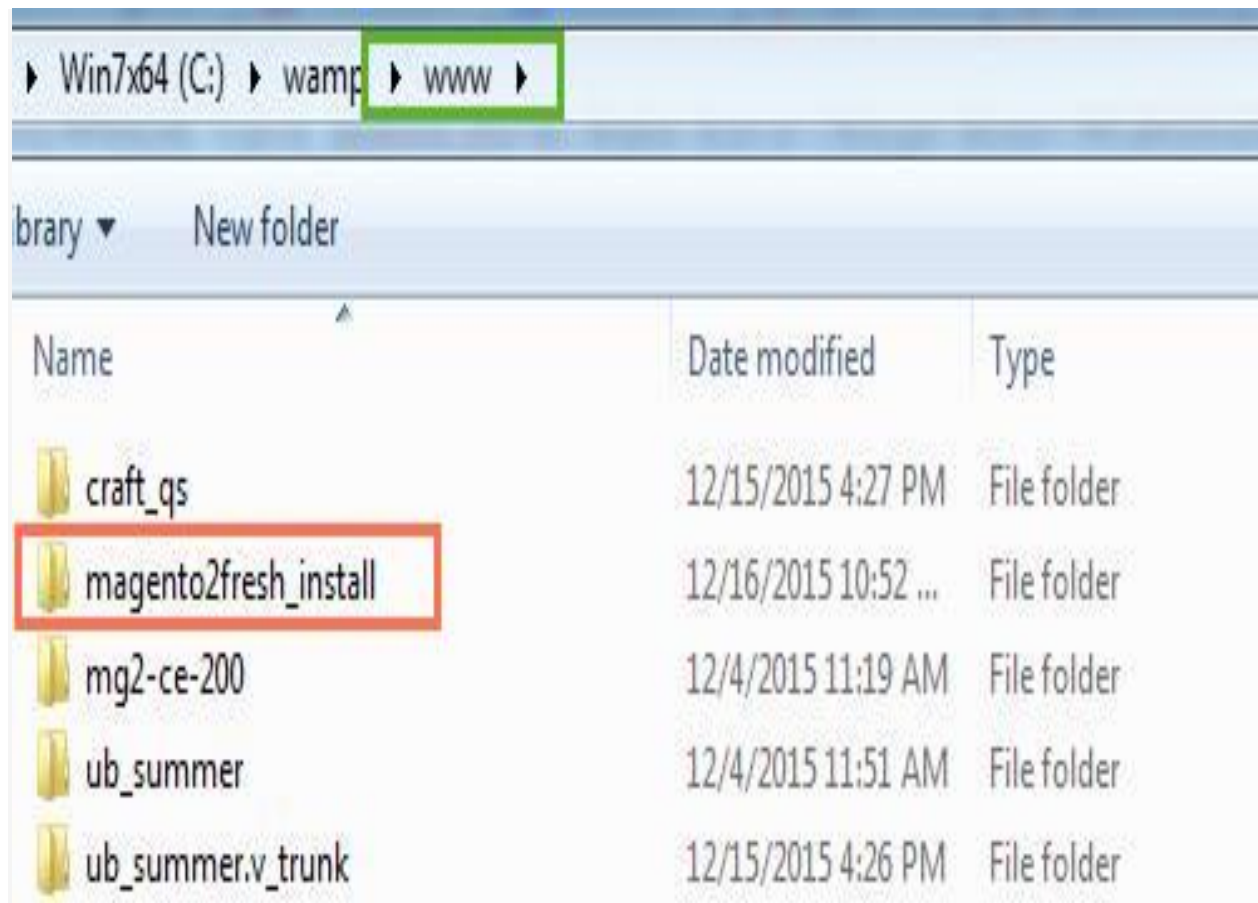
See the following for more detail:

- Magento technology stack requirements
- Supported browsers
- TLS requirement for repo.magento.com
- TLS 1.2 requirement for PayPal
- Apache 2.2 or later

- PHP 5.5.x or PHP 5.6.x
Note: Magento 2 does not support PHP 5.4.
- Required PHP extensions:
 - PDO/MySQL
 - mbstring
 - mcrypt
 - mhash
 - simplexml
 - curl
 - gd2, ImageMagick 6.3.7 (or later) or both
 - soap
 - intl
 - bc-math
 - openssl
- MySQL 5.6.x
- Mail Transfer Agent (MTA) or an SMTP server
-

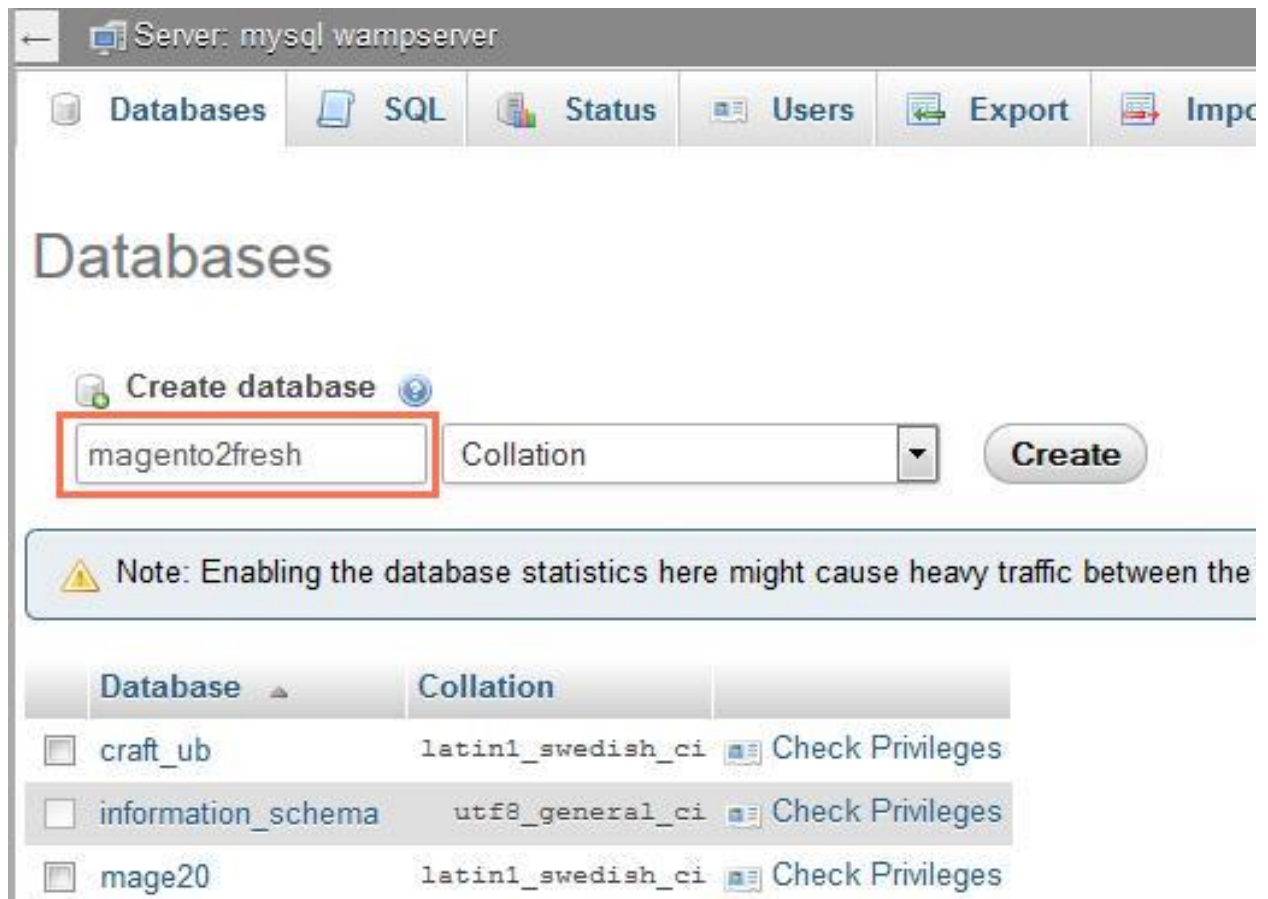
When you are done with verifying your prerequisites, follow the steps below to install Magento 2.

1. Upload package to your web root



- Create DataBase

Open PHPmyadmin, then create Database for Magento 2.0



Follow the 7 steps below to start your Magento installation

From your browser, navigate to the location of the Magento downloader on your server.

Step 1 : Agree and Setup

The Magento Downloader welcome page appears.



Version 2.0.0

Welcome to Magento Admin, your online store headquarters.
Click 'Agree and Set Up Magento' or read [Getting Started](#) to learn more.

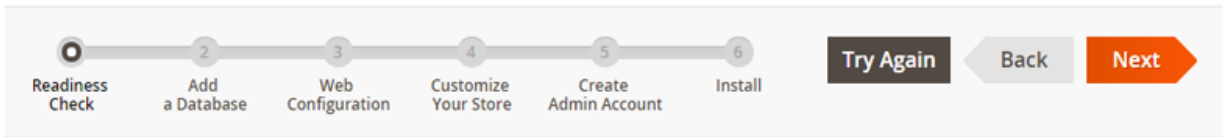
[Terms & Agreement](#)

Agree and Setup Magento

Start setup

Step 2 : Verify System requirements

To validate your server environment, click the Continue button. If error warnings appear, you must correct the errors and validate your environment again




Step 1: Readiness Check

✓ **Completed!** You can now move on to the next step.

- ✓ **PHP Version Check**
Your PHP version is correct (5.6.15).
- ✓ **PHP Settings Check**
Your PHP settings are correct.
- ✓ **PHP Extensions Check**
You meet 14 out of 14 PHP extensions requirements. [Show detail](#)
- ✓ **File Permission Check**
You meet 4 out of 4 writable file permission requirements. [Show detail](#)

Step 3 : Setup the database



1

Readiness
Check

2

Add
a Database

3

Web
Configuration

4

Customize
Your Store

5

Create
Admin Account

6

Install

Back

Next

Database Server Host *

localhost

Database Server Username *

root

Database Server Password

(not always necessary)

Database Name *

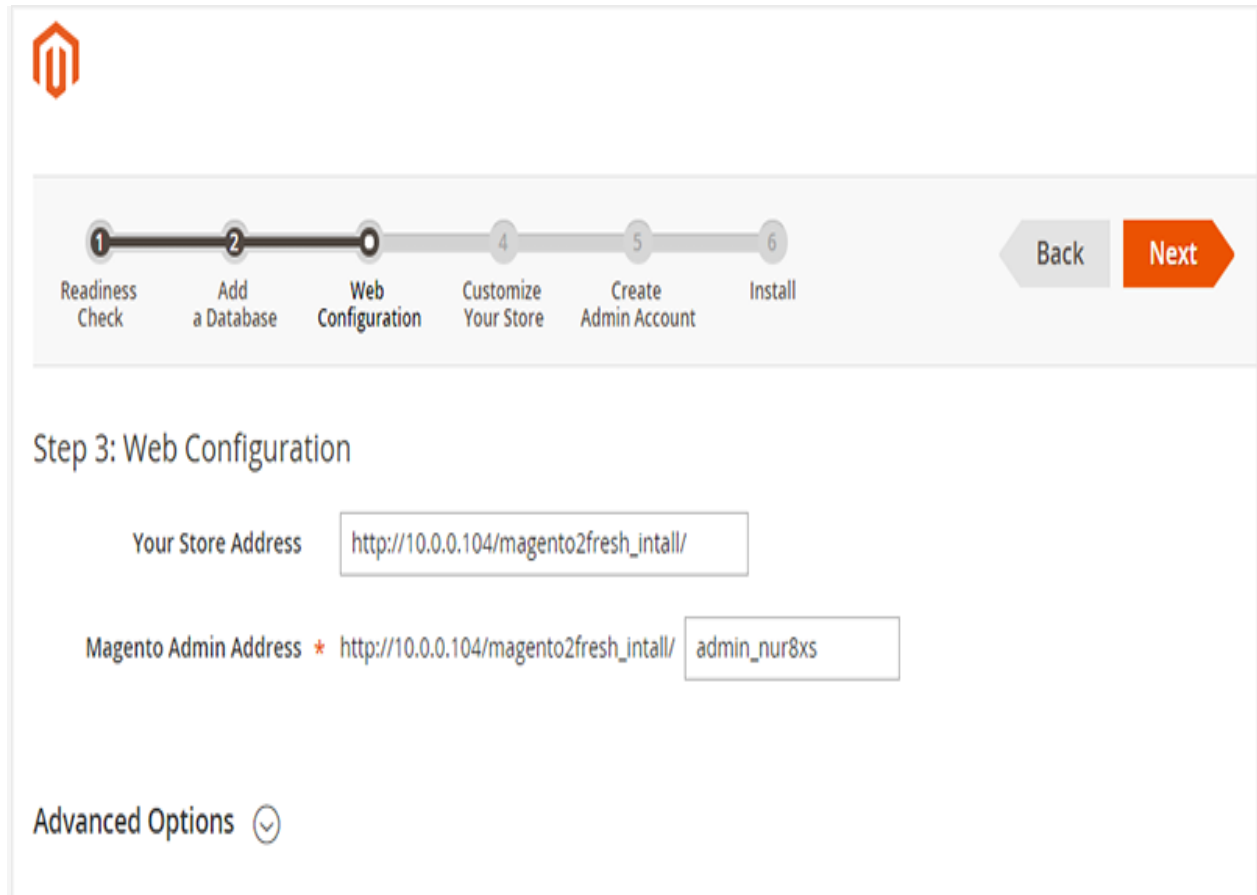
magento2fresh

Enter the name of your store's database.


Table prefix

(optional)

Step 4 : Web Configuration



The image shows the Magento 2 installation wizard interface. At the top left is the Magento logo. Below it is a progress bar with six steps: 1. Readiness Check, 2. Add a Database, 3. Web Configuration (current step), 4. Customize Your Store, 5. Create Admin Account, and 6. Install. To the right of the progress bar are 'Back' and 'Next' buttons. The main content area is titled 'Step 3: Web Configuration'. It contains two input fields: 'Your Store Address' with the value 'http://10.0.0.104/magento2fresh_intall/' and 'Magento Admin Address' with a red asterisk, followed by the value 'http://10.0.0.104/magento2fresh_intall/' and a separate input field containing 'admin_nur8xs'. At the bottom is an 'Advanced Options' section with a downward arrow icon.



1 Readiness Check 2 Add a Database 3 Web Configuration 4 Customize Your Store 5 Create Admin Account 6 Install

Back Next

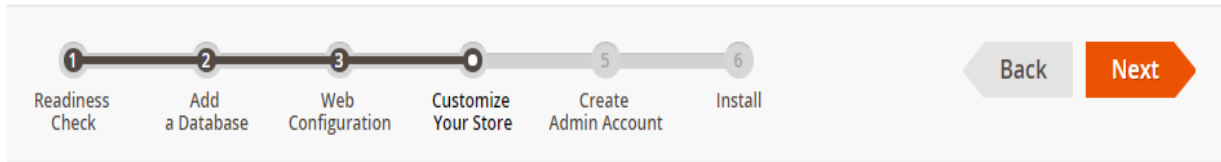
Step 3: Web Configuration

Your Store Address

Magento Admin Address *

Advanced Options 

Step 5 : Configure Language, Time Zone and Currency



Step 4: Customize Your Store

Store Default Time Zone * GMT (UTC) ▼

Store Default Currency * US Dollar (USD) ▼

Store Default Language * English (United States) ▼

Advanced Modules Configurations ⌵

Step 6 : Create Admin Account



Back

Next

Step 5: Create Admin Account

Create a new Admin account to manage your store.

New Username *

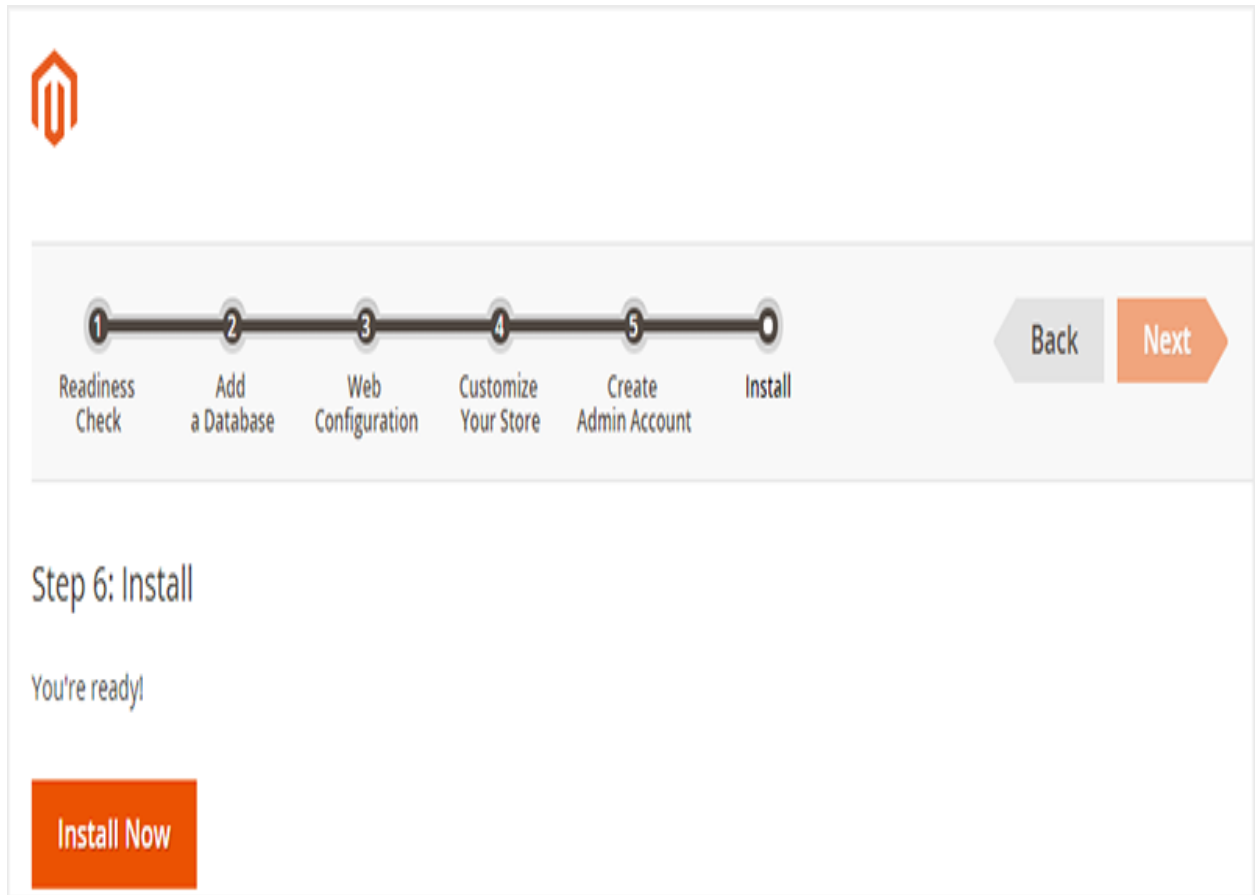
admin

New Email *

New Password *

Confirm Password *

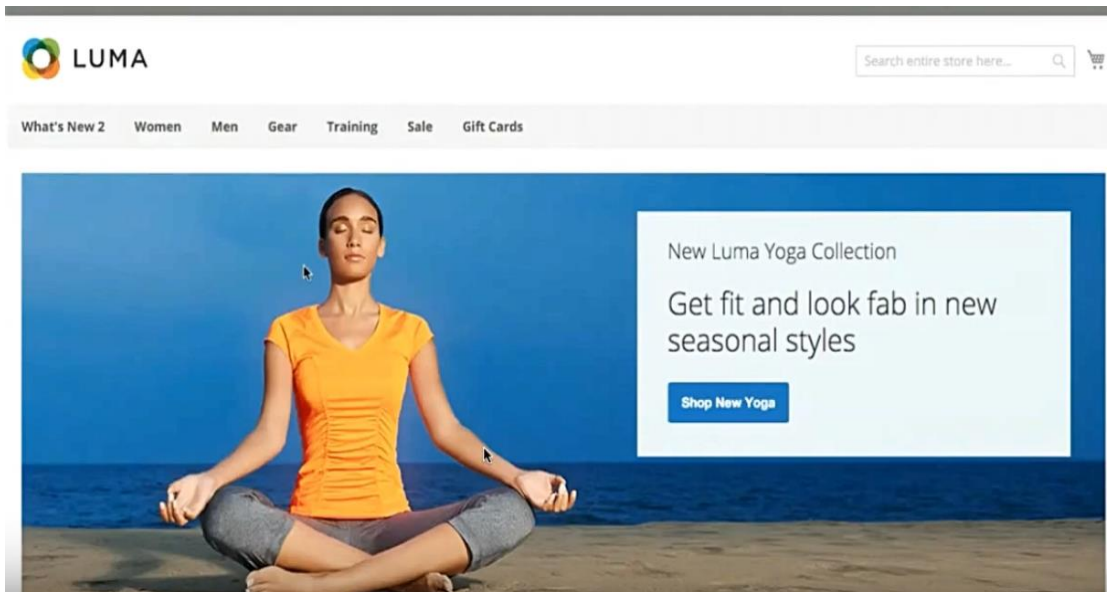
Step 7 : Install & Finish



Exploration Being a USER:

User or client is a person who comes to your website to interact with your website and is attracted by the structure of your website, your design or the attraction you have made to convince the user to come to your website and stay long on your site. Magento provides a lot of user interaction to interact user with your site like editing the content, giving reviews, commenting, adding things to the cart, making and maintaining a wishlist and so on.....

MAIN Page of Sample Website:



As a user there are a lot of feature through user can interact with the website like:

1. The main Menu bar of the website allows the user to see and open the main and important services of the website and main categories.
2. The main page of the website provides the attractive and convincing interface and user can easily get the idea what about this website is.
3. Scrolling the main page can give a complete tour of the website.
4. A user can also get the news of the new arrival of the stock.
5. A user will be able to maintain a wishlist of the product he likes after the complete exploration of the website which engages him to the website for long.
6. User can also comment or review about the product which helps the aftercoming to decide about the quality and experience of the products.
7. User can also select different styles, categories and colors which are provided by the framework Magento as:

Hoodies & Sweatshirts

Shopping Options

Items 1-9 of 12

Sort By Position

STYLE

CLIMATE

COLOR

MATERIAL

PATTERN

PRICE

Compare Products

You have no items to compare.

My Wish Lists

You have no items in your wish list.



Mona Pullover Hoodie

★★★★★ 3 reviews

£57.00



Hera Pullover Hoodie

★★★★★ 3 reviews

£48.00



Autumn Pulle

★★★★★ 3 reviews

£57.00



Miko Pullover Hoodie

★★★★★ 3 reviews

£69.00



- The user can explore and search the product of his own choice and category either for men and women and can take an overview and detailed description of the product.

Related Products

Check items to add to the cart or select all

Daria Bikram Pant
£40.80 Regular Price
£51.00

Iris Workout Top
£29.00

Layla Tee
£29.00

Sahara Leggings
£60.00 Regular Price
£75.00

Fusion Backpack
£59.00

Endeavor Daytrip Backpack
£33.00

We found other products you might like!

magento2ee.phplab.co.uk/sahara-leggings.html

- User can place the order by giving the important and reachable credentials:

The screenshot displays the LUMA checkout interface. At the top left is the LUMA logo. A progress bar indicates the current step is 'Shipping' (marked with a checkmark) and the next step is 'Review & Payments' (marked with a '2'). In the top right corner, there is a 'Sign In' link. The main section is titled 'Shipping Address' and contains several input fields: 'Email Address *' (with a red asterisk and a help icon), 'First Name *', 'Last Name *', 'Company', and 'Street Address *'. A red error message 'This is a required field.' is visible below the email field, and a note states 'You can create an account after checkout.' To the right of the form is a grey box titled 'Order Summary' which shows '2 Items in Cart' with a dropdown arrow.

Exploration Being a Database Developer:

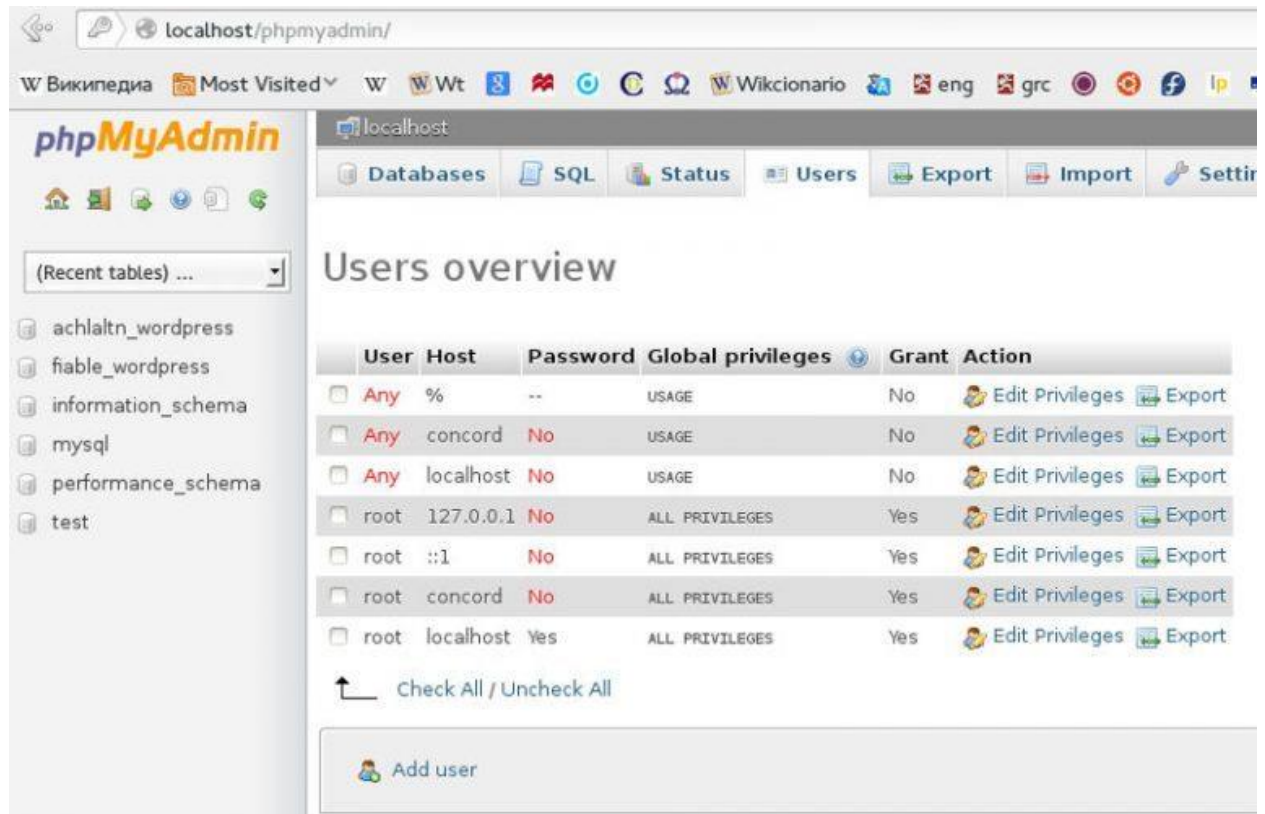
A Database Developer is primarily responsible for creating and implementing databases. He determines the best **database** management system for a particular client, as well as test **database** programs for efficiency and performance and troubleshoot and correct problems. In Magento a database is developed on the SQL server and is being managed and implemented by the database developer. Database Developer maintain the records of the users who have logged in to the website and stores their liked products and also show the content accordingly. Also the database developer make the updates in the website and create the database of newly added products to the website. Storing user comments and user reviews to the database and analyzing that feedback is also an important task of the of the DB Developer.

The DB Developer login into the Admin system and can see and make changes in the dataase accordingly. He can see the customer information, orders and so on as:

The screenshot shows the Magento Go administrative interface. At the top, there's a navigation bar with 'Start My Store' and the Magento Go logo. Below this is a menu with various options like Dashboard, Sales, Catalog, Design, Customers, Promotions, Newsletter, CMS, Reports, System, and Add-ons. A search bar for 'Global Record Search' is also present. A status bar indicates the latest finished task as 'Regenerate config task' and mentions 27 unread messages. The main content area is titled 'Customer Information' and features a sidebar with a 'Customer View' dropdown and a list of links: Account Information, Addresses, Orders, Billing Agreements, Shopping Cart, Wishlist, Newsletter, Product Reviews, Product Tags, and Store Credit. The main panel shows details for 'Peter Smith', including buttons for Back, Reset, Create Order, Delete Customer, Manage Shopping Cart, Save Customer, and Save and Continue Edit. The 'Personal Information' section displays fields for Last Logged In, Confirmed email, Account Created on, Account Created in, and Customer Group. The 'Default Billing Address' section shows the customer's name, address, city, state, zip, and phone number. Below this is a 'Store Credit Balance' section with a table showing 'Balance' and 'Website' columns, indicating 'No records found.'

There are a lot of controls available to the DB Developer and he can update the data of users, tables and other available privileges. He can add and remove users and products according to the need and requirements. The Users Tab allows the control over the user which includes creating new user and managing them this data is directly linked to the database along with a lot of other data in phpMyAdmin.

The DB Developer has the power and ability to create new users, new tables, new databases according to the need of the store and requirements.



Being a Backend Developer:

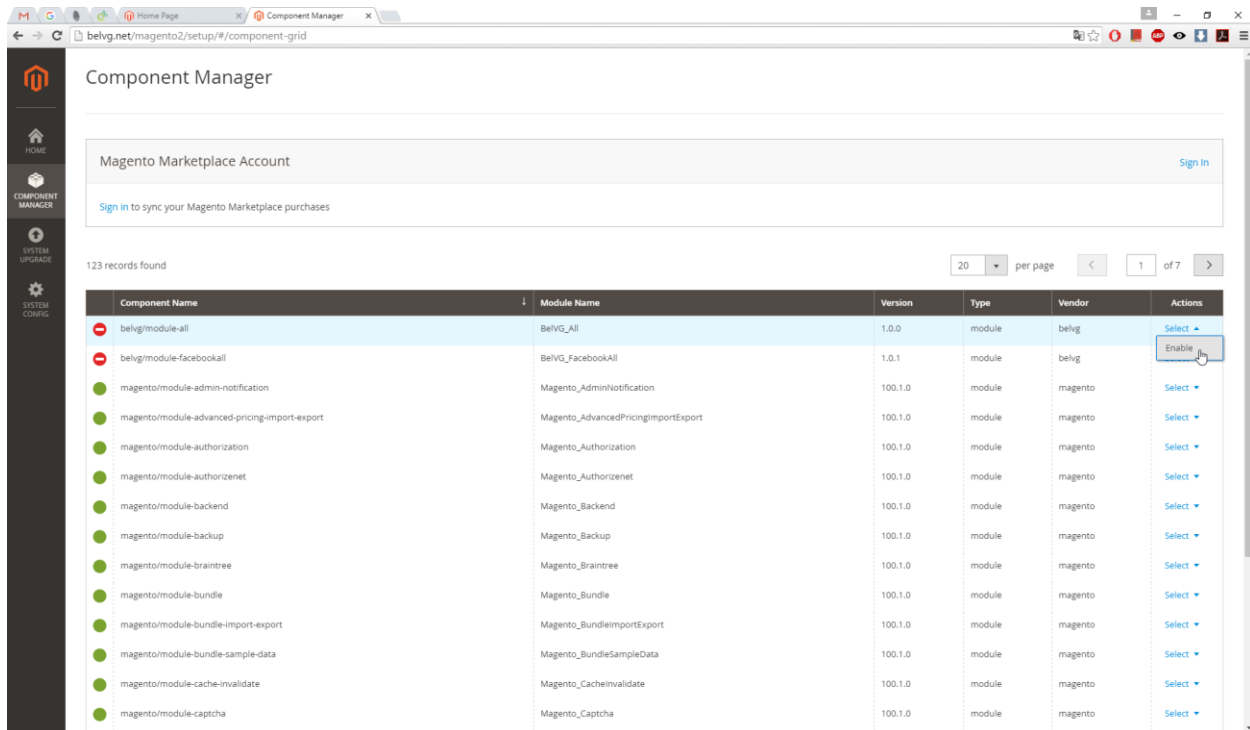
As a backend developer, a backend developer has the power given by the magento framework to change and manipulate the contents of the website. Either it is HTML, or CSS or JAVAscript a backend developer can change the product display styles, their arrangements, prices, categories, designs through this framework. According the need and requirements and most of the time from the feedback given by the users and reviews, a backend developer decides to change the style, price or listing of the products. Updating the css files will change the look of the website, js files are responsible for the dynamic changes, php is needed to communicate with the server. In order to make changes to the files a coder needs access to the admin.

Adding Modules to Magento

This method requires Cron tasks set up and proper running. In case it has not performed yet — do it. You will need it further, not only while extensions installing.
So if Cron is set up correctly, just follow the path in admin panel to initialize the module:

System/ Web Setup Wizard/ Component Manager

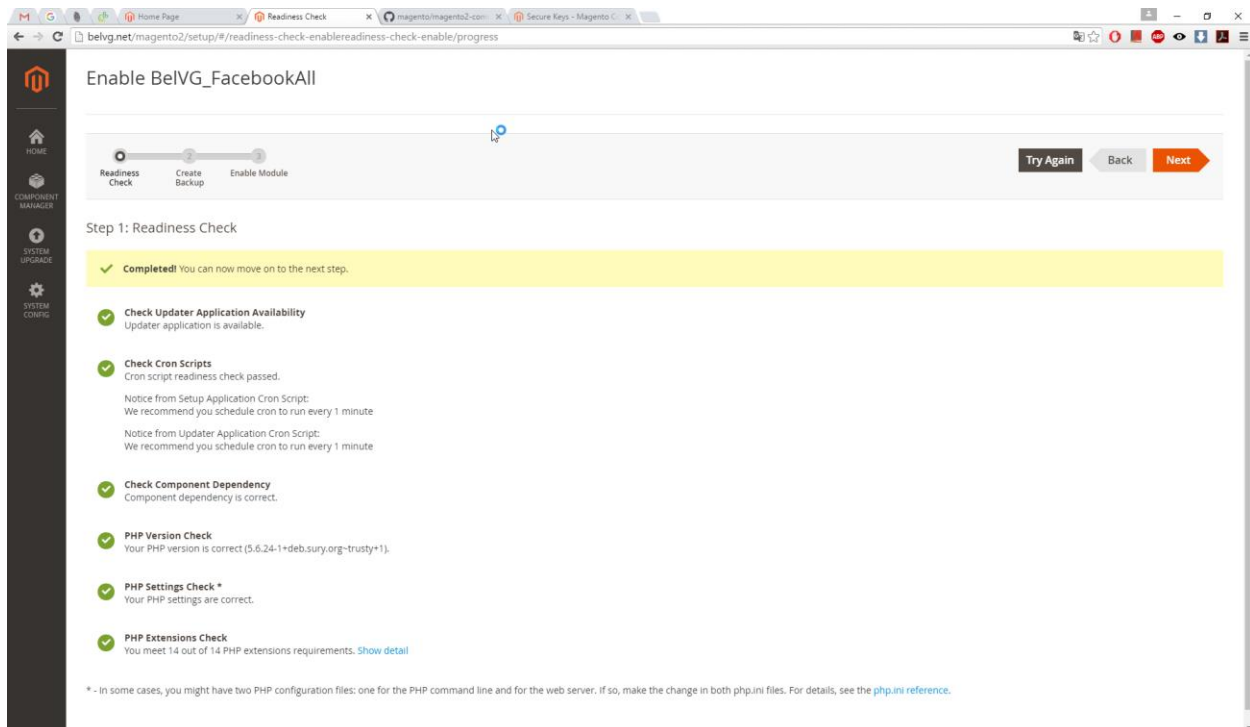
Here you can enable all available modules.



The screenshot shows the Magento Component Manager interface. At the top, there's a 'Magento Marketplace Account' section with a 'Sign In' link. Below this, a table lists 123 records found. The table has columns for Component Name, Module Name, Version, Type, Vendor, and Actions. The first two rows are highlighted in blue. The first row is 'belvg/module-all' with Module Name 'BelVG_All', Version '1.0.0', Type 'module', and Vendor 'belvg'. The second row is 'belvg/module-facebookall' with Module Name 'BelVG_FacebookAll', Version '1.0.1', Type 'module', and Vendor 'belvg'. Both rows have an 'Enable' button in the Actions column. The rest of the table lists various Magento modules like 'magento/module-admin-notification', 'magento/module-advanced-pricing-import-export', etc., all with version '100.1.0' and vendor 'magento'.

Component Name	Module Name	Version	Type	Vendor	Actions
belvg/module-all	BelVG_All	1.0.0	module	belvg	Enable
belvg/module-facebookall	BelVG_FacebookAll	1.0.1	module	belvg	Enable
magento/module-admin-notification	Magento_AdminNotification	100.1.0	module	magento	Select
magento/module-advanced-pricing-import-export	Magento_AdvancedPricingImportExport	100.1.0	module	magento	Select
magento/module-authorization	Magento_Authorization	100.1.0	module	magento	Select
magento/module-authorizenet	Magento_Authorizenet	100.1.0	module	magento	Select
magento/module-backend	Magento_Backend	100.1.0	module	magento	Select
magento/module-backup	Magento_Backup	100.1.0	module	magento	Select
magento/module-braintree	Magento_Braintree	100.1.0	module	magento	Select
magento/module-bundle	Magento_Bundle	100.1.0	module	magento	Select
magento/module-bundle-import-export	Magento_BundleImportExport	100.1.0	module	magento	Select
magento/module-bundle-sample-data	Magento_BundleSampleData	100.1.0	module	magento	Select
magento/module-cache-invalidate	Magento_CacheInvalidate	100.1.0	module	magento	Select
magento/module-captcha	Magento_Captcha	100.1.0	module	magento	Select

In case Magento is set up properly you'll see the notification that all of the systems run correctly.



The screenshot shows the 'Enable BelVG_FacebookAll' progress screen. At the top, there's a progress bar with three steps: 'Readiness Check' (completed), 'Create Backup', and 'Enable Module'. Below the progress bar, there's a 'Try Again' button, a 'Back' button, and a 'Next' button. The main content area is titled 'Step 1: Readiness Check' and shows a 'Completed!' message. Below this, there are several checkmarks indicating successful checks: 'Check Updater Application Availability', 'Check Cron Scripts', 'Check Component Dependency', 'PHP Version Check', 'PHP Settings Check', and 'PHP Extensions Check'. Each check has a brief description of what was verified. At the bottom, there's a note about PHP configuration files.

Step 1: Readiness Check

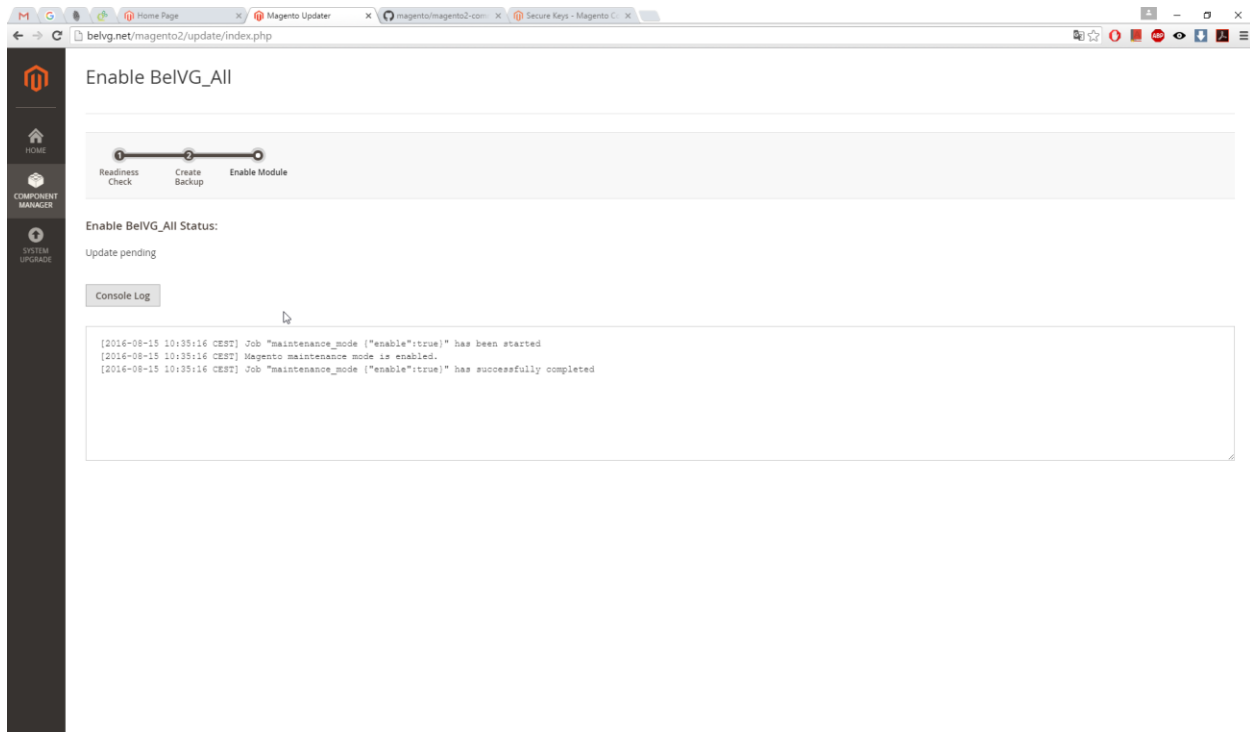
Completed! You can now move on to the next step.

- ✓ Check Updater Application Availability
Updater application is available.
- ✓ Check Cron Scripts
Cron script readiness check passed.
Notice from Setup Application Cron Script:
We recommend you schedule cron to run every 1 minute
Notice from Updater Application Cron Script:
We recommend you schedule cron to run every 1 minute
- ✓ Check Component Dependency
Component dependency is correct.
- ✓ PHP Version Check
Your PHP version is correct (5.6.24-1+deb.sury.org-trusty*1).
- ✓ PHP Settings Check *
Your PHP settings are correct.
- ✓ PHP Extensions Check
You meet 14 out of 14 PHP extensions requirements. [Show detail](#)

* - In some cases, you might have two PHP configuration files: one for the PHP command line and for the web server. If so, make the change in both php.ini files. For details, see the [php.ini reference](#).

Further, before you enable the custom module in Magento 2, Backup creation will be proposed.

And finally we will see the following screen:




The module has been enabled. And now we can switch back to the module list and enable (install) the next one.

Problems I Faced During the Assignment

Following are the major problems I faced during the assignment:

Problem # 1

 **Error!** You need to resolve these issues to continue.



PHP Version Check

Your PHP version is correct (7.2.22).



PHP Settings Check *

Your PHP settings are correct.



PHP Extensions Check

3 missing PHP extensions. [Hide detail](#)

The best way to resolve this is to install the correct missing extensions. The exact server, your host, and other system variables.
Our [PHP extension help](#) can get you started.

I searched on internet and finally solved the problem.

Solution:



Open your Php.ini file available at the location `{your_installation_drive}:\xampp\php`

0

Search for the below given extension over there.



```
extension=intl.dll  
extension=php_soap.dll  
extension=php_xsl.dll
```

May be available at line no `884, 906 & 911` respectively just remove the `;` symbol before the extensions.

Save the `php.ini` file & restart the the server again

If still facing the same problem feel free to revert

Thank you!

Step 2: Add a Database

✖ SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using password: NO)

Database Server Host *

Database Server Username *

Database Server Password

Database Name *

I also solved this issue by searching on the internet.
