**What is Magento?**

Magento is an open source E-commerce software, created by Varien which is useful for online business and it has the flexible modular architecture. It is scalable and it has many control options that helps the user. Magento uses E-commerce platform which offers companies ultimate E-commerce solutions and extensive support network.

**Why to use Magento?**

* Magento is open source E-commerce software.
* It is scalable and offers small companies to build business.
* It provides the searching and sorting of products in several ways.
* Easily integrates with many of the third-party sites which are needed to run effective E-commerce website.
* Using this, customer can order or purchase number of products. There are no limits on number of purchasing products.

**What are disadvantages of Magento?**

* Magento uses larger disk space and memory.
* It takes much time to build the customized functionality.
* It is very slow compared to other E-commerce sites.
* It needs proper hosting environment, if the hosting environment is improper the user can face the problems.

**Name the product types that are available in Magento?**

The product types available in Magento are −

* Simple Products
* Grouped Products
* Configurable Products
* Virtual Products
* Bundled Products
* Downloadable Products

**What is grouped product?**

This is a group of simple products. In this type, you cannot specify a specific price for product; you can just specify the discount.

**What is Configurable Products**

In this type, customer can select products according to their color and size before purchasing.

Example − Cell phones obtained in different colors and sizes.

**What are Bundled products?**

Bundled products are those products which cannot be sold separately and doesn't give any choice for end user.

**What are Magento product types?**

* **Magento simple product:** It is used for a single item without any specific selectable variations. For example, a pen, copy, etc.
* **Magento grouped product:** It is used for a combination of Magento simple product. For example, a pen and copy together.
* **Magento configurable product:** It is used for a single item with specific selectable variations. For example, a pen with different color options.
* **Magento virtual product:** It is used for a virtual item i.e: non touchable item. For example, reservation, insurance, etc.
* **Magento bundle product:** It is used for a bundle of simple products. For example, laptop with various items such as processor, hard disk, RAM, etc.
* **Magento downloadable product:** It is used for online software items. For example, PowerPoint presentation, MP3 files, servers, etc.

**What is Inventory?**

Inventory allows setting a product's stock quantity. For instance, you have a product with 100 units in stock. If you set the stock availability to "Out of Stock" then it will force the item to be out of stock.

**Name the categories of E-Commerce?**

The categories of E-Commerce are:

* Business to Business(B2B)
* Business to Consumer(B2C)
* Consumer to Consumer(C2C)
* Consumer to Business(C2B)

**What is the used of Zero Subtotal Checkout panel?**

The Zero Subtotal Checkout panel is a payment option that displays when order total is zero and not required to enter payment details for the customer.

**Which are the methods of PayPal Payment Gateways?**

The two methods of PayPal Payment Gateways are −

* Payflow Pro (Includes Express Checkout)
* Payflow Link (Includes Express Checkout)

**What is Payflow Pro?**

The Payflow Pro option is customizable payment gateway which can be used with merchant account to process credit card transactions.

**What is Payflow Link?**

Payflow Link option often called as hosted payment gateway that keeps customer on your site by providing fast and easy way to add transaction processing to your site.

**What is the use of My Cart Link panel?**

My Cart Link panel specifies whether the number of quantities in the cart should be shown or whether the number of different products should be shown using the Display Cart Summaryfield.

**What is Google checkout in Magento?**

Google Checkout is online payment processing service provided by Google. Magento allows integration of online stores with Google checkout. It is like PayPal that simplifies the process of paying for online purchases.

**What is Magento Manage Order?**

Order management is important thing which allows business to run smoothly and keeps customers happy, making them more likely to visit your site in the future.

**What is Magento Google Analytics?**

Google Analytics is a finest Google service for those who are actively managing websites and adds analytics to Magento store including Ecommerce tracking and conversions of their websites.

**What is the use of Page Layout in Magento?**

Layout files are useful in rendering front pages of Magento.

**What is Magento Content Management System(CMS)?**

Magento CMS (Content Management System) section is used to manage all web site pages. It is a way of promoting the products by providing valuable information to the customers and increases visibility to search engines.

**What are Static Blocks?**

Static block is a piece of content can be used anywhere in the pages. Magento allows creating blocks of content that can be used through the store and can be added to any page or another block.

**What are Polls?**

Polls are used to get customer's opinions and preferences. The poll results appear immediately after response is submitted.

**How to Optimize the Magento Environment?**

* It uses complex database, so that it needs to be run on dedicated servers.
* Magento application could be optimized by using cloud computing.
* Merge you JavaScript and CSS files which reduces the load time dramatically since its loading only one merged file.
* Proper MySQL configuration is one of the most important aspects in terms of performance.
* Always upgrade to the latest Magento version allows to perform better.

**How to Optimize Magento Configuration?**

* To speed up Magento performance, don't run mysql and web server on the same machine.
* Do not host files on your web server that you do not use.
* Optimization of session storage.
* Enabling Magento flat catalog. (Magento uses a complex and resource-intensive Entity Attribute Value [or EAV] based catalog). After initial catalog establishment, enabling the flat catalog can dramatically improve database querying time.
* Identification and disabling of unused Magento modules.

**What is the process of Code Optimization?**

The process of Code Optimization are −

* Removal of unused or unnecessary code processes.
* To optimize Magento performance, JavaScript and CSS files need to be compressed and aggregated.
* Conformance of all site images to optimal web image sizes.
* Identification of bottlenecks(process that causes the entire process to slow down or stop) processes in both front-end and back-end.

**How to improve the performance of database?**

The following points describe how to improve the performance of database −

* Unused data must be cleaned up regularly for better performance.
* Optimization of database queries.
* Configuration of setting and limits of the database server (e.g. memory setting, query cache, sort buffer optimization).

**What is WSDL?**

Web Services Description Language. It is used for describing web services and how to access them.

**What is the process of Order Life cycle in Magento?**

Orders follow a standard life cycle process. When customer place product orders, it arrive in the administration interface with a pending status. When order is processed, the status of order changes according to current state in the processing workflow. Once the invoice is created for the order, the status changes from pending to processing status. Next it creates shipment for an order which changes the status from pending to complete status.

**Which are the two sections present in Design Section?**

The two sections present in Design section are −

* Page Layout
* Custom Design

**What does the Page Layout section contains?**

The Page Layout section contains a Layout option which allows selecting layout as per your choice and Layout Update XML option inserts the XML code.

**What is the use of Meta Data section while setup new pages?**

The Meta Data section contains Keywords and description of the page.

**How to subscribe to Newsletters using Magento?**

Customer can subscribe to Newsletters using Magento. Customer can sign up for the Newsletter when he creates a new customer account which contains checkbox for signing up. For creating Newsletters you need to enable Newsletter option in your magento to make sure that customer has confirmed to receive Newsletter.

**How to optimize the Magento front-end performance?**

The following points show how to optimize the Magento front-end performance −

* Use the latest version of php, so that you can perform front-end operation much better and faster. The newest released version may cause the errors, so carefully read the release notes and check out the new version.
* Use the clean database to improve the performance of magento. The database logs need to be clear regularly. The database stores the automatically created logs to keep track of record session and interaction.

**What is the use of Backorder field in Product Stock Option Panel?**

If it is enabled, customer can buy products even if they are out of stock.

**What are Tax rules?**

Tax rules are entities that combine product tax classes, customer tax classes and tax rates.

**What is the use of Manage store section?**

Manage Stores section, you will see website names, store names and Store View Name columns.

**What is E-commerce?**

E-commerce (Electronic Commerce) is a type of business that involves the commercial transaction or purchasing or selling of goods and services through electronic channels known as internet.

**What are the features of Magento?**

The features of Magento are −

* Magento provides different payment methods such as credit cards, PayPal, cheques, money order, Google checkouts.
* It provides shipping of products in one order to the multiple addresses.
* Easy to manage the orders by using admin panel.
* It filters the products and displays in grid or list format.

**What are the advantages of Magento?**

The advantages of magento are −

* It is user friendly E-commerce software.
* It is compatible with Smartphone's, tablets and other mobile devices.
* It provides multiple payment options so every visitor can make payment based on their preferred payment gateway.
* It has many extensions which supports for the development of an online store.

**Which is the MySql version used for magento?**

MySQL 5.1

**What scripting language and database management system does Magento use?**

Magento uses PHP as a scripting language and MySQL for the database. Database: MySQL 5.6 (Oracle or Percona). PHP version: PHP 5.4, PHP 5.5 or above.

**Mention what is the limitation of Magento?**

Magento is supported by PHP, comparatively to other e-commerce solutions, Magento might be slow in performance

It becomes a complex system if it is not using object-oriented programming

Magento requires much space and memory

**Explain how you can change Magento Core API settings?**

To change Magento Core API settings, you have to

* Go to Admin menu, choose **System -> Configuration**
* Select Magento Core API on the left side of the **Configuration Panel**, under **Services**
* Tap on to expand the **General Settings** section and you can
* Type the name of the **Default Response Charset**that you want to use
* Determine the **Client Session Timeout** in seconds
* Click the **Save Config** button when complete

**Explain how Magento can be made more secure for the client?**

Best practices for Magento includes

* Using a strong password and changing them at regular interval
* Disable remote access to Magento Connect Manager
* Disable Downloader on production sites
* Restrict access to safe IP addresses

**Mention what is the benefit of applying Connect Patches in Magento?**

In Magento, applying connect patch

* It enables easy installation of packages with installation; it overwrites any existing translations for the same time
* For enhancing security, by default **Magento Connect** uses HTTP to download extensions instead of FTP
* With a dash character in the name, extension developers can now create extensions
* Magento administrators will be informed now who tries to install an extension with insufficient file system privileges

**Explain how you can configure Magento to work with another domain?**

To do that, in the admin area, you have to change the Magento Base URL option. To do this, visit to your Magento admin area> System > Configuration and click Web on the left menu. Select the **Unsecure option**after that edit the base URL field to change the URL that will be used for normal (HTTP) connections.

**In Magento 2, what are the different deploy modes and what are their differences?**

**Developer**

In this mode, all the files in pub/static/ are symlinks to the original file. Exceptions are thrown and errors are displayed in the front end. This mode makes pages load very slowly, but makes it easier to debug, as it compiles and loads static files every time. Cache can still be enabled.  
**Default**

This default is enabled out-of-the-box. It is a state in between production and developer, as the files are generated when they are needed. I.e. CSS files are generated using several LESS files in several locations. These files will be generated only when they are needed by the front end, and will not be generated again the next time they are needed.  
**Production**

This mode should be enabled for all Magento 2 websites in production, as all the required files are generated and placed in the pub/static folder.

**In Magento 2, what is dependency injection and what are its advantages?**

Dependency injection is a design pattern strategy that relegates the responsibility of injecting the right dependency to the calling module or framework. This is the Hollywood Principle: “Don’t call us, we’ll call you.”

The responsibility of calling the right dependency is no longer handled by the function and respects [the SOLID principle](https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design).

Its main advantages are that it makes code:

1. Easier to test
2. Easier to re-use
3. Easier to maintain

**In Magento 2, what is a factory class and how does it work?**

Factory classes are generated when code generation happens. They are created automatically for models that represent database entities.

Factory classes are used to create, get, or change entity records without using the ObjectManager directly, as its direct usage is discouraged by Magento. (This is because it goes against the principles of dependency injection.)

These classes do not need to be manually defined, but they can be, in case you need to define a specific behavior.

**What is the difference between a store and a website?**

Some parameters are defined by a store and some are defined by a website:

| **Parameter** | **Scope** |
| --- | --- |
| Product settings | Default, Store View |
| Product prices | Default, Website |
| Product tax class | Default, Website |
| Base currency | Default, Website |
| Display currency | Default, Store view |
| System configuration settings | Default, Website, and Store view |
| Root category configuration | Store group |
| Orders | Store view |
| Customers | Default, Website |
| Category settings | Default, Store view |

For example, if you need to define different base currencies, you will need two different websites.

**In Magento 1, how can you change the behavior and extend Magento core functionality? If there are multiple ways, explain their differences and advantages/disadvantages.**

There are three ways to override core functionalities:

1. **Monkey patches**: Because Magento loads modules in a specific order, you can override the modules located in the core and community code pools by copying them into the local code pool. In that case, you will rewrite the whole class. This is the least preferred method.
2. **Rewrites**: You can rewrite a function by specifying a class in your config file to rewrite an existing class. In this case, you can extend the parent class and rewrite only one function.
3. **Observers**: Magento throws events when specific actions are done. If there is an event that is thrown before or after the action you want to interact with, you can intercept it and modify it. This is the preferred method.

**What is the best way to count the items in a collection? Explain the differences with other method(s).**

The best way is to use the method getSize(). This function will not load the collection each time to count the items but store it. So every time you need this value you will not have to recalculate it. Moreover, it uses the SQL COUNT() function in order to speed up the counting process. However, if the collection has been modified, this value can become inconsistent.

In contrast, the count() method will load the collection and count its items every time it is called. This can become very resource demanding.

**Mention what is the architecture of Magento? What are the different versions of Magento?**

Magento architecture is a typical PHP **MVC (Model-View-Controller)** application, where all the **Controllers** will be in one folder and all the**Models** in another, etc.  Based on their functionality files are grouped together, which are referred as modules in Magento.

Different version of Magento includes

* Magento Enterprise
* Magento .go
* Magento Community

**Mention what is the difference between Mage::getModel() and Mage::getSingletone() in Magento?**

**Mage::getModel():** It always creates a new object

**Mage: :getSingleton():** It always look for an existing object and if not then creates a new object

**Mention what are the basic features of Magento?**

Basic features of Magento includes

* Reporting and Analytics
* Product and Catalog Browsing
* Customer Accounts
* Order Management
* Payment
* Site Management
* Shipping
* Search engine optimization
* Marketing promotions and tools
* Checkout
* International Support

**Explain Registry pattern in Magento?**

Registry pattern is basically a pattern that allows any object or data to be available in a public global scope for any resource to use. In Magento, you can register an object or data with Mage::register function, Mage::register('identifier', $object\_or\_data); After it is registered, you can call it with Mage::registry by passing identifier name: Mage::registry('identifier'); You can also unregister an object at any time with Mage::unregister Mage::unregister('identifier'); This is especially help

**In Magento which product types implement a parent-child relationship between product entities?**

Configurable, bundle and grouped products implement a parent-child relationship between product entities in Magento.

**Explain Magento’s Compilation feature?**

The Compilation feature of Magento will allow you to compile all files of a Magento installation in order to create a single include path to increase performance. The performance increase is between 25% - 50% on page loads. In order to use this tool, the directory ‘includes’ and the file includes/config.php must both be writable.

**What are the differences between EAV and Flat model?**

EAV is an entity attribute value database model, where data is fully in normalized form. Each column data value is stored in their respective data type table. Example, for a product, product ID is stored in catalog\_product\_entity\_int table, product name in catalog\_product\_entity\_varchar, product price in catalog\_product\_entity\_decimal, product created date in catalog\_product\_entity\_datetime and product description in catalog\_product\_entity\_text table. EAV is complex as it joins 5-6 tables even if you want to get just one product’s details. Columns are called attributes in EAV.

The flat model uses just one table, so it’s not normalized and uses more database space. It clears the EAV overhead, but not good for dynamic requirements where you may have to add more columns in database table in future. It’s good when comes to performance, as it will only require one query to load whole product instead of joining 5-6 tables to get just one product’s details. Columns are called fields in flat model

**What is the difference between “pay” and “capture” operations in Magento?**

**Pay**: This basically sets the invoice state is ‘paid’, this works for both when payment is online or offline.

**Capture**: This is when actual payment processing happens online, and the capture() method in our payment method is called. After capture, pay() is called.

### What is EAV in Magento?

EAV stands for Entity Attribute Value. It is a technique that facilitates users to add unlimited columns to their table virtually.

### How many tables will be created in EAV module in Magento? Name them.

EAV module will create 6 tables in database.They are as follows:

 module  
 module\_datetime  
 module\_decimal  
 module\_int  
 module\_text  
 module\_varchar

### Explain the difference between EAV and flat model.

EAV database model is fully in normalized form. Each column's value is stored in their respective data type table which makes it more complex as they have to join 5-6 tables even if you need only one detail. In EAV, columns are called attributes.

Flat model uses just one table. It is not normalized and uses more database space. It is not good for dynamic requirements where you may have to add some more columns in future. Its performance is fast as it needs only one query instead of joining 5-6 tables. In flat model, columns are called fields.

**Explain different modules in Magento.**

* Core modules
* Commercial modules
* Community modules

**Explain ORM in Magento.**

ORM stands for Object Relational Mapping. It is a programming technique used to convert different types of data into objects and vice versa.

There are two types of ORM:

* Converts different types of data into objects.
* Converts objects to various types of data.

**How to change theme for login users?**

To change theme for login users,

1. if(Mage::getSingleton('customer/session')-**>**isLoggedIn()):
2. Mage::getDesign()-**>**setPackageName('package\_name')-**>**setTheme('themename');
3. endif;

**How can you add an external JavaScript/ CSS file to Magento?**

1. css/yourstyle.css

or

1. skin\_jsjs/ yourfile.js
2. skin\_csscss/yourstyle. css

**State the syntax to call a CMS page in your module's PHTML file.**

1. $this-**>**getLayout()-**>**createBlock('cms/block')-**>**setBlockId('blockidentifier')-**>**toHtml();

### When you need to clear the cache to see the changes made in Magento?

When you have added or modified XML, CSS or JS files.

**How to run custom query in Magento?**

To run custom query,

1. $db = Mage::getSingleton('core/resource')-**>**getConnection('core\_write');
2. $result=$db-**>**query('SELECT \* FROM users where id=4');

### How to enable product's custom attribute visibility in frontend?

In **Manage Attributes** section under custom attribute, select **Yes** for "Visible on Product View Page on Frontend" and "Used in Product Listing".

### State whether namespace is mandatory while creating a custom module in Magento?

No, namespace is not mandatory while creating custom module.

### Is it possible to have more than one grid in Magento?

Yes it is possible.

**List the magic methods in Magento?**

Magic methods in Magento:

* \_\_get()
* \_\_set()
* \_\_isset()
* \_\_call()
* \_\_tostring()
* \_\_construct()
* \_\_has()
* \_\_uns()

**How many types of sessions are there? Why we use different sessions in Magento?**

There are namely three sessions in Magento:

* customer session
* checkout session
* core session

All these sessions are stored in one session only. We use different sessions because sometimes we need to clear only a particular session data and not all session data.

**How can you reset Magento Files and Directory permissions?**

Change the directory to the directory where Magento is installed and execute the following commands.

1. find. -type d -exec chmod 755
2. chmod+x magento

**How will you get first and last item from the collection in Magento?**

$collection-**>**getFirstItem() and $collection-**>**getLastItem();

### Explain the use of namespace in Magento?

Magento core modules are placed in mage namespace, **core/Mage/Catalog** and all custom modules are placed in **local/CustomModule.**

You can have more than one module with same name but they need to be placed in different namespaces.

### Explain handles in Magento?

Handles control the structure of the page to be displayed. It decides which block will be placed where in the page. Handle is called for every page and every page request can have several unique handles.

### Explain compilation feature in Magento?

Compilation feature allows us to compile all Magento files to create a single include path to increase performance.

**How to enable Maintenance mode in Magento?**

Create a file named as maintenance.flag and upload it to Magento home directory containing following code.

1. $maintenanceFile = 'maintenance.flag';
3. if (file\_exists($maintenanceFile)) {
4. include\_once dirname(\_\_FILE\_\_) . '/errors/503.php';
5. exit;
6. }

**How to convert default currency to others in Magento?**

To convert default currency to others, select the currency and import currency rates from **System-> Manage currency-> Rates.**

**Syntax:**

1. $convertedPrice = Mage::helper('directory')-**>**currencyConvert($price, currentCurrency, newCurrency);

### Explain Google checkout in Magento.

Magento allows the integration of online stores with Google checkout. Google checkout is the online payments service provided by the Google. It works like PayPal.

**Explain how to change Magento core API setting?**

You have to follow these steps to change Magento core API setting.

* Go to Admin menu, choose System -> Configuration
* Select Magento Core API on the left side of the Configuration Panel, under Services
* Click on to expand the General Settings section
* Type name of the Default Response Charset that you want to use
* Determine the Client Session Timeout in seconds
* Click the Save Config button when completed

**Can all billing information be managed through Magento?**

You can do the following things through client Magento account:

* You can update your billing address.
* You can add a credit card.
* You can view your billing history.
* You can add a PayPal account.
* You can produce a print ready receipt.

**What are the advantages of applying Connect Patches in Magento?**

In Magento, applying Connect Patches provides following features:

* Enable easy installation of packages with installation and overwrite any existing translations for the same time
* Enhance security, by default Magento Connect uses HTTP to download extensions instead of FTP
* Facilitate the extension developers to create new extensions with a dash character in the name
* Magento administrators will be informed now who tries to install an extension with insufficient file system privileges.

**How to fetch 5 bestsellers products programmatically in Magento?**

Mage::getResourceModel('reports/product\_collection')

-**>**addOrderedQty()

-**>**addAttributeToSelect('\*')

-**>**setPage(1, 5)

-**>**load();

### What is codePool?

Code pool is a concept to pull the code in Magento structured format. It is specified when you register new module in app/etc/modules/Company\_Module.xml

There are 3 codePools in Magento: core, community and local, which reside at app/code/ directory.

**CodePools:  
  
\_community:** It is generally used by 3rd party extensions.  
**\_core:** It is used by Magento core team.  
**\_local:** Local codePool should be used for in-hour module development and overriding of core and community modules for custom requirement.

**Residing directory:**

**\_app  
\_code**So in short, you can say that codePool helps Magento to locate module inside app/code/ for processing

**Block Remove:**

<referenceBlock name="copyright" remove="true"/>

**Magento2 Command Line**

**Setup Upgrade Using Command Line**

php bin/magento setup:upgrade

If you don’t want to remove pub/static files while installing/updating database then use following command.

php bin/magento setup:upgrade --keep-generated

**Cache Clean Using Command Line**

php bin/magento cache:clean

**Cache Flush Using Command Line**

php bin/magento cache:flush

**View cache status Using Command Line**

php bin/magento cache:status

**Enable Cache Using Command Line**

php bin/magento cache:enable [cache\_type]

**Disable Cache Using Command Line**

php bin/magento cache:disable [cache\_type]

**Static Content Deploy Using Command Line**

php bin/magento setup:static-content:deploy

**Static Content Deploy For Particular Language Using Command Line**

php bin/magento setup:static-content:deploy en\_US

**Static Content Deploy For Magento Backend Theme Using Command Line** (Working on 2.1.1 or later)

php bin/magento setup:static-content:deploy --theme="Magento/backend"

**Static Content Deploy For Specific Themes Using Command Line** (Working on 2.1.1 or later)

php bin/magento setup:static-content:deploy --theme Magento/luma --theme Magento/second\_theme

**Exclude Themes on Static Content Deploy and does not minify HTML files Using Command Line** (Working on 2.1.1 or later)

php bin/magento setup:static-content:deploy en\_US --exclude-theme Magento/luma --no-html-minify

Reindexing Using Command Line

php bin/magento indexer:reindex

**View the list of indexers Using Command Line**

php bin/magento indexer:info

**View indexer status Using Command Line**

php bin/magento indexer:status

**Show the mode of all indexers Using Command Line**

php bin/magento indexer:show-mode

**Enable module Using Command Line**

php bin/magento module:enable Namespace\_Module

**Disable module Using Command Line**

php bin/magento module:disable Namespace\_Module

**Uninstall Module Using Command Line**

php bin/magento module:uninstall Namespace\_Module

**Check Current Mode Using Command Line**

php bin/magento deploy:mode:show

**Change To Developer Mode Using Command Line**

php bin/magento deploy:mode:set developer

**Change To Production Mode Using Command Line**

php bin/magento deploy:mode:set production

**Run the single-tenant Compiler Using Command Line**

php bin/magento setup:di:compile

**Steps for custom theme creation:**

1. Create a directory for the theme under app/design/frontend/<your\_vendor\_name>/<your\_theme\_name>
2. Add or copy from an existing theme.xml to your theme directory app/design/frontend/<Vendor>/<theme>

<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Config/etc/theme.xsd">

<title>New theme</title> <!-- your theme's name -->

<parent>Magento/blank</parent> <!-- the parent theme, in case your theme inherits from an existing theme -->

<media>

<preview\_image>media/preview.jpg</preview\_image> <!-- the path to your theme's preview image -->

</media>

</theme>

Save as theme.xml

1. To distribute your theme as a package, add a composer.json file to the theme directory.
2. {
3. "name": "magento/theme-frontend-luma",
4. "description": "N/A",
5. "require": {
6. "php": "~5.5.0|~5.6.0|~7.0.0",
7. "magento/theme-frontend-blank": "100.0.\*",
8. "magento/framework": "100.0.\*"
9. },
10. "type": "magento2-theme",
11. "version": "100.0.1",
12. "license": [
13. "OSL-3.0",
14. "AFL-3.0"
15. ],
16. "autoload": {
17. "files": [
18. "registration.php"
19. ]
20. }
21. }

4. To register your theme in the system, in your theme directory add a registration.php file with the following content.

<?php

/\*\*

\* Copyright © 2015 Magento. All rights reserved.

\* See COPYING.txt for license details.

\*/

\Magento\Framework\Component\ComponentRegistrar::register(

\Magento\Framework\Component\ComponentRegistrar::THEME,

'frontend/<Vendor>/<theme>',

\_\_DIR\_\_

);

5. Create folder with name “media” and place inside it an image “preview.jpg”.

6. Create “web” folder and place all static content inside it (css,js,images).

**Steps for custom theme integration:**

1. Create a folder to connect layout theme <Magento\_Theme> in the following path:

app/design/frontend/<your\_vendor\_name>/<your\_theme\_name>/ Magento\_Theme

1. Copy layout and template folders from \vendor\magento\module-theme\view\frontend

and place in the Magento\_Theme.

1. To override the default theme, override the phtmls placed in Magento\_Theme/templates/html.
2. Remove the default css and add the custom css files in the following path.

app\design\frontend\keerthi\mytheme\Magento\_Theme\layout\ default\_head\_blocks

<!-- Remove blank/luma theme styling if you declared a parent -->

<remove src="css/styles-m.css" />

<remove src="css/styles-l.css" />

<remove src="css/print.css" />

<!-- Add your own CSS files -->

<css src="css/override.css" />

<css src="css/main.min.css" />

1. Include your js code with in below function in .js files to make js work.
2. require(['jquery', 'jquery/ui'], function($){
3. //your js code here
4. });

6. Enable Template Path Hints to make overriding the existing theme process easy.

Store->Configuration->Developer->Advanced

Open Debug dropdown and set Enabled Template Path Hints for Storefront to yes.

7. Design your custom header and footer by overriding container.phtml and bugreport.phtml.

**Fun City Theme Integration:**

1. Create a custom CMS page.
2. Create a root category(eg:UAE) and multiple sub categories with 3 custom category attributes like class\_name, id, data\_title.

Custom attribute creation:

<http://inchoo.net/magento-2/add-category-attribute-magento-2/>

1. Create a phtml file in the following path:

app\design\frontend\keerthi\mytheme\Magento\_Catalog\templates\catalog\product

and include this phtml file in the custom CMS page created above.

**Including a phtml to a CMS page:**

{{block class="Magento\Framework\View\Element\Template" template="Magento\_Catalog::catalog/product/storecategories.phtml"}}

1. Write the Jquery required for your theme.

How to build Custom Theme in Magento 2

Frontend:

**Theme folder structure:**

\app

\design

\frontend

\**Bfl <vender name>**

\**customtheme**

\Magento-Theme

\layout

\default.xml

\media

\preview.png

\web

\images

\custom-logo.png

\registration.php

\theme.xml

\composer.json

Admin:

**URL:**

<http://localhost:8080/magento_store/admin_store/admin/>

**Add Secret Key to URLs** :  
  
<http://localhost:8080/magento_store/admin_store/admin/>

Store 🡪 Configuration 🡪 ADVANCED 🡪 admin 🡪 Security 🡪 Add Secret Key to URLs

**Change Theme:**

CONTENT 🡪 Configuration 🡪 Edit

commands

php bin/magento setup:upgrade

php bin/magento setup:static-content:deploy –f

php bin/magento indexer:reindex

php bin/magento cache:flush

php bin/magento cache:clean

php bin/magento deploy:mode:show

php bin/magento deploy:mode:set developer

c:\xampp\htdocs\magento\_store>php bin/magento deploy:mode:set developer

The directory "C:/xampp/htdocs/magento\_store/pub/static/frontend/Magento/blank/e

n\_US/Magento\_OfflineShipping/js/model" cannot be deleted Warning!rmdir(C:/xampp/

htdocs/magento\_store/pub/static/frontend/Magento/blank/en\_US/Magento\_OfflineShip

ping/js/model): Directory not empty

php bin/magento module:status

php bin/magento module:enable Bfl\_CustomModule

php bin/magento setup:di:compile

Info: Some modules might require static view files to be cleared. To do t

n 'module:enable' with the --clear-static-content option to clear them.

Delete Folders: before cache clear

1). var/view\_preprocessed

2). pub/static/frontend/Storename