**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.