**Magento 2 Theme Development**

**Magento Installation**

We need magento 2.2.0 installation with sample data.

**Theme Structure**

Theme\_directory/

Vendor\_Module/

web/

css/

source/

layout/

override/

templates/

etc/

i18n/

media/

web/

css/

source/

fonts/

images/

js/

composer.json

registration.php

theme.xml

**Theme Location**

Custom theme are located at /app/design/frontend/vendor /custom\_theme\_name

Built-in theme i.e. Luma and Blank theme are located at

/vendor/magento/theme-front-luma

**Static Files Location**

Static files are published in following folders

/pub/static/frontend/vendor/theme/language

theme\_dir/media

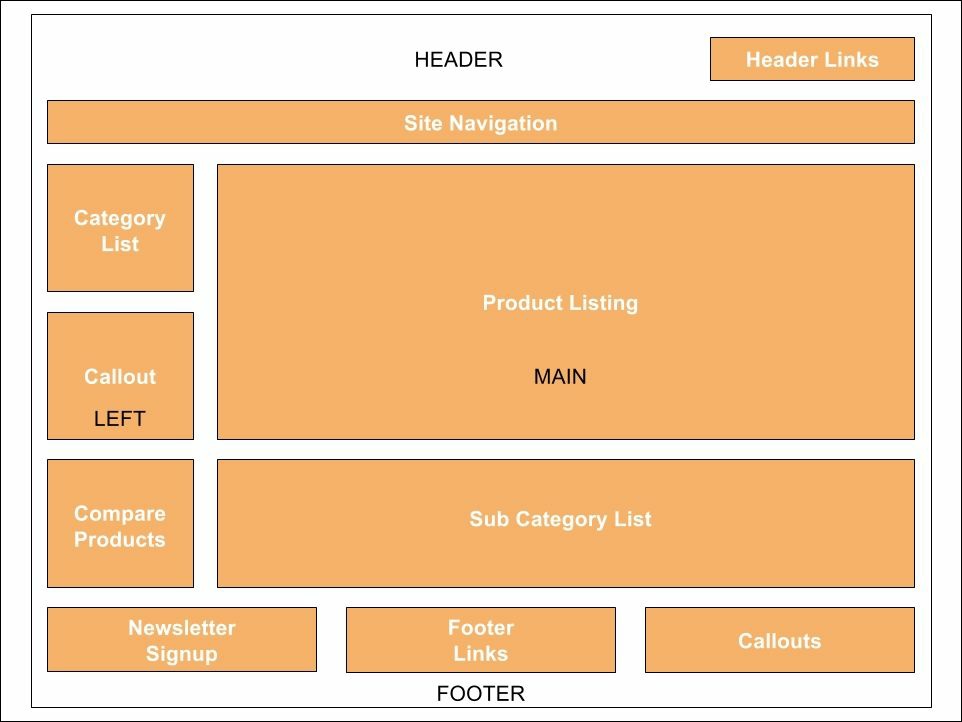
theme\_dir/web

**Magento theme**

Provide visual appearance to design to entire application using custom template, layout

, styles and images.

**Magento Page Layout**



**Magento theme structure**

theme\_dir

vendor\_module

web

css

source

layout

override

templates

etc

i18n

media

web

css

source

fonts

images

js

composer.json

registration.php

theme.xml

**Create One Web 🡪 Three Store 🡪 Three view of a store**

Login to admin 🡪 Store 🡪 All Store 🡪 Click create website 🡪 give name & code 🡪

Again click on store and create store under website 🡪 then click on website and create

website under store.

**Magento 2 theme customization**

We can every aspect of magento store through layout, CSS, Javascript, template and

local files.

**Magento 2 Luma Theme**

Luma theme implement Responsive Web Design (RWD). Inherits code and layout from

Blank theme.

Luma theme style is based on magento User Interface (UI) Library and CSS3 Media

queries for screen width.

**Blocks and Pages**

Magento use blocks to differentiate between various components of it’s functionality.

Example of Blocks are

The search feature

Product listing

Mini cart

Category listing

Site Navigation links

Callouts (advertising blocks)

We can create block and page by login to admin 🡪 Content 🡪 Pages or Block

**Custom Variables**

Custom Variable are piece of code that have specific value like programming variable.

For example

{{config path =”web/unsecure/base\_url”}}

this variable will show URL store

To create custom variable login with admin 🡪 System --. Custom Variable 🡪 Click Add

🡪 create variable with code and it’s value

Now to use it open content 🡪 pages 🡪 About us 🡪 enter following code in content

area {{CustomVar code=”dev\_name”}}

save the page and browse from frontend.

**Chapter No. 3 Magento 2 Theme Layout**

**Magento 2 Model – View – Architecture**

**Model** model handle system business rules and data persistence

**View** this is responsible for display layer

**Controller** this layer define main actions, request and response, that may change

the model layer

1. Request comes from browser to Controller 🡪 Controller forward request to Model 🡪 Model return data results to Controller 🡪 Controller pass results data to View 🡪 View return GUI content back to controller 🡪 Controller return back HTTP Response to Browser

**Command line Interface**

php bin/magento list

**Magento 2 Theme files**

Each theme have two types of files

1. Static view files (images , javascripts, CSS )
2. Dynamic view files ( .less files )

**Caching System**

We can flush magento 2 cache by running CLI commands

**Magento 2 Layout System**

Magento 2 layout system is define by XML files.

Layout layer is divided into following sections

Page Layout this is responsible for general structure of page

Page configuration this is used for reporting meta information, content type

Generic Layout this is same like page layout but is used for AJAX request

Magento Modes

Magento have three mode

Default

Production Mode

Developer Mode

Creating our first theme

Create a new directory in /app/design/frontend/Packt

Create another directory inside Packt directory with name helloworld

Create a file with name theme.xml inside helloworld directory

and following code in it.

<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:noNamespaceSchemaLocation="urn:magento:framework:Config/etc/theme.xsd">  
 <title>Hello World theme</title>  
 <parent>Magento/blank</parent>  
 *<!-- <media>  
 <preview\_image>media/preview.jpg</preview\_image>  
 </media>-->*

</theme>

Create a preview.jpg of 800X800 pixel size and save in media directory

Create a file with name registration.php inside helloworld directory

Create Static Files directories

Static files are save in web directory , create a web directory inside helloworld

web 🡪

css 🡪source

fonts

images

js

Creating Logo

By default logo is recognize with name logo.svg with default location

theme\_dir/web/images/logo.svg

if we have logo.svg file, simply putting in above directory will work

but if we want to work with different name and format then we have to define

with layout files.

Create a directory with name Magento\_Theme/web/images inside helloworld

and put the logo.png file

them create directory layout in Magento\_Theme and create default.xml inside it.

Magento 2 theme debugs

To enable debug feature, follow below procedure

Login to admin 🡪 Store 🡪 configuration 🡪 Advanced -> Developer 🡪 Debug 🡪 Enable

1st and 4th option to yes 🡪 flush cache 🡪 check from frontend

Magento User Interface Library (UI)

Following Technology are used by magento

HTML5 + CSS + LESS

RWD Them

Magento 2 UI

Built-in PHP LESS compiler

Web content accessibility guide lines (WCAG 2.0) compliant

LESS Language

LESS is a programming language which add additional functionality to CSS

Variable usage

Function Usage

Mixins

Magento 2 UI

LESS and CSS preprocessing

LESS compilation in Magento 2

Magento 2 UI Library

Magento UI library is located at /lib/web/css/source/lib

Here we have \_breadcrum.less file, magento blank them call function of this file

at location magento\_blank\_thme/web/css/soure/\_breadcrumb.less

Testing Magento 2 UI Library

Let make some customization in helloworld theme using Magento UI library.

CREATING PRACTICAL MAGENTO 2 THEME

Prerequisites

1. Latest browser
2. Knowledge of CSS, CSS3
3. Knowledge of LESS
4. Knowledge of HTML , HTML5
5. Knowledge of XML
6. Knowledge of Javascript
7. Knowledge of PHTML
8. Knowledge of PHP
9. Knowledge of Responsive Web Design (RWD)

Clearing Cache

To clear the cache and Static file Cache, run below commands

php bin/magent cache:clean

rm -rf pub/static/\* remember not to delete the .htaccess file

rm -rf var/view\_preprocessed/\*

Theme

Install a Third-party theme

Install a theme manually

🡪 Upload your theme files to /app/design/frontend directory 🡪

run command php bin/magento setup:upgrade and then activate

from admin

Install theme as composer package

Not required by time being

Install theme as extension

If theme is distribution on magento marketplace, use web wizard

Building BookStore theme

1. Create a directory with name bookstore in Packt/ directory

Developing magento 2 template

Magento Frontend Development

Magento cache

Cache Location

Client

Server

Cache Systems

Files system (Default, no need for setup)

Varnish

Redis

Database

Theme

Install a third party store front theme

We can install magento theme by two way

Manually

Copy the theme file in magento-root-dir/app/design/frontend directory

Composer

**To install theme, we must set magento mode to developer mode**

Registering Theme

After installing the theme, we need to register the theme in magento admin.

Uninstalling a storefront theme

Make sure magento is an developer mode

Theme is not active theme

Make sure theme is not parent theme for a child theme

Delete the theme directory from app/design/frontend/vendor directory

Remove the record from database using below query

**mysql -u <user> -p -e "delete from <dbname>.theme where theme\_path ='<Vendor>/<theme>' AND**

**area ='frontend' limit 1"**

Creating Admin Theme

Content Menu

Pages

Create Pages with text, images, blocks , variable and widget and connect to

navigation of store and linked to other pages

Blocks

We can create block of contents without writing any code. Block can contain

text, images and video and can be assign to any part of page.

Dynamics Block

This option in only available in Magento commerce

Widgets

Widget display dynamic data and add blocks, links and interactive elements

anywhere in store.

Media

Media Gallery helps store owner to organize the media files in admin area

We must enable media gallery for this option

Design

Manage the visual presentation of store, apply theme and schedule design change

Staging

This option in only available in Magento commerce