**DRUPAL**

Drupal is an open source CMS (Content Management System) that allows organizing, managing and publishing content. It is based on PHP environments and is carried out under GNU (General Public License) where everyone can download freely and share it. Drupal is used on million sites such as WhiteHouse.gov, World Economic Forum, Stanford University, Examiner.com and many more.

Content Management System is a software which keeps all data that is available in the website like photos, text, documents, etc. CMS helps in editing, modifying and publishing the content of the website.

Firstly, Drupal was released in Jan 2001.

Why Drupal: Developing a website using CMS is easy with Drupal. New users can easily edit the content without any knowledge of HTML and other web technologies. It can easily interact with other technologies and can handle complex forms and workflows. It is available with more than 16000 modules can be addressed with Drupal core and add-on modules.

Features:

1. Easy to create and manage site
2. Built in user interface is used for translating anything in the system.
3. Our website can be connected to other sites using feeds and search engine connection capabilities.
4. No licensing costs as it is an open source software.
5. Drupal allows to publish content on social networking sites such as facebook and twitter.
6. Provides number of customizable themes.
7. Manages content on informational sites, media sites, intranets and web application.

Advantages:

1. Different types of content like blog, menu handling, video, text can be handled flexibly.
2. No need of starting the application from scratch as multiple templates are available for developing the web application.
3. It have 7000 plug ins which boost up website and can also create own plug ins.
4. Provides interesting templates and themes that gives attractive look to the website.

Disadvantages:

1. Low performance compared to other CMS’s
2. Not compatible with other software’s.
3. UI is not user friendly. Need advanced knowledge and few basic things for installation of the platform.

Drupal Architecture: It has following layers i.e Users, PHP, database, WebServer, Administrator.

Users: These are the users on the Drupal community. The user sends a request to a server using Drupal CMS and where web browsers, search engines etc acts like clients.

**Administrator**: Administrator can provide access permission to authorized users and will be able to block unauthorized access. The Administrative account will be having all privileges for managing content and administering the site.

**Drupal**: Drupal is a free and open source Content Management System (CMS) that allows organizing, managing and publishing your content and built on PHP based environments. Drupal CMS is very flexible and powerful and can be used for building large, complex sites. It is an easy way of interacting with other site and technologies using Drupal CMS. Further, you will be able to handle complex forms and workflows.

**PHP**: Drupal uses PHP in order to work with an application which is created by a user. It takes help of web server to fetch data from the database. PHP memory requirements depend on the modules which are used in your site. Drupal 6 requires at least 16MB, Drupal 7 requires 32MB and Drupal 8 requires 64MB.

**Web Server**: Web server is a server where the user interacts and processes requests via HTTP (Hyper Text Transfer Protocol) and serves files that form web pages to web users. The communication between user and server takes place using HTTP. You can use different types of web servers such as apache, IIS, nginx, lighttpd etc.

**Database:** The database stores the user information, content and many more required data of the site. It is used to store the administrative information to manage the Drupal site. Drupal uses the database to extract the data from a database and enables to store, modify and update the database.

**Main Menu**: Menus are very important to navigate your website easily. Menus offer a set of links as a result, you can navigate your website. The Drupal menu allows you to add, remove and rename the menus and menu items.

**Blocks & Regions**: Blocks are container objects that are used to organize your content of your website. It can be displayed in the regions on your page.

* Block title: Enter the title for your block.
* Block description: Enter the description about block.
* Block body: Enter the content of the block.
* Text format: Select the format for your text i.e. Filtered HTML, FULL HTML, and Plain text.

**Themes& Layouts**: Drupal will install Bartik theme as default theme during installation. You can select paid or free themes from Drupal official site. In general, layout is an arrangement of text and graphics. It is a good thought to choose a theme, keeping what different layouts to use on your site.

**Front Page**: If you have a good content on your website and expect your visitor to see that particular content first, then that article must be on the front page.

**Static Pages:** Main purpose of static pages is speed and steadiness. They are faster than dynamic pages.

Click on **Content**->**Add Content**->**click on Basic page. Title** specifies the title of static page. **Body** displays the content of the page. **URL path settings** can specify alternative URL by which content can be accessed. We can select Publish options based on the requirement.

**Creating Article**: Creating article enables users to know about the website when they visit.

**Drupal Optimization:** Performance of Drupal website can be improved by enabling **cache blocks and cache pages from anonymous users**. Having multiple css files can lower your website speed. **Compress the files** by enabling the **aggregate and compress files**. Moving media files and static files to a CDN(Content Delivery Network) that hosts media and static files. **CDN acts as external cache for files** and they store in **high performance servers** so that it is optimized. **Optimize Images Size and Their Number: Do not have many images which consume more space as this leads to slow website.** Always resize your images and have few that are related to your website.

**Site Backup:** This helps in backing up the files and database files. For this, we need to install FileZilla client on the system.

**Backup files in Drupal:**

Step 1: open FileZilla client

Step 2: Enter Host, username, Password as same as use to login into cpanel. And then click QuickConnect.

Step 3: Displays all files and folder of Drupal website at the right side.

Step 4: Select all files related and folders and right click and click on download. All the files will get saved on the system.

**Backup database :**

**Step 1:** goto phpmyadmin

**Step 2:** Select the drupal database

**Step 3:** Click Export tab

**Step 4:** Two methods will be available to export i.e Quick and Custom. Select any method and click go. Database file will be available in the system.

**Caching:** There are two types of users they are **Authenticated Users and Anonymous Users**. Caching for Authneticated is challending whereas Anonymous can be fully offloaded to a cache.

Basic performance tuning at **Administration > Configuration > Performance > Development > Performance.**

**It is always recommended to enable block and page caching. Instead of dynamic retrieval from db using sql queried it caches this for fast retrieval.**

If caching is enabled on the site, we need to clear all the cache to see if any changes is done to the site. For best performance, caching must be disabled while developing and should always be enabled in the production.

**Devel module** has quick cache fushing in the menu. We can use DRUSH.

**Drush cc all**- This is the shell command used for flushing the cache.

**Database Caching**

Database caching can be provided by a distributed memory object caching system, such as **[Memcache](https://www.drupal.org/project/memcache)**. **Memcached** allows you to **allocate memory where you have more than you need and make it accessible to areas where you have less than you need**. When implemented with Drupal, Memcached can store the result of your database query’s in the memory for a specified time, reducing database traffic.

**Web Server (Proxy) Caching**

HTTP acceleration, or Web Server (Proxy) Caching, will significantly reduce resource requirements and page load times. [**Varnish Cache**](https://www.drupal.org/project/varnish), is a widely-used HTTP accelerator for Drupal sites.

HTTP acceleration is handled by a reverse proxy. A reverse proxy is a type of proxy server that retrieves resources (pages, images, files, etc.) prior to being requested by a website visitor. These resources are stored in virtual memory and are quickly retrieved by the proxy server when requested. Reverse proxies are heavily threaded and optimized for data retrieval.

**Performance Modules:**

If better optimization is required, you might **install one or two performance modules**. For **example**, if the site is on a shared server, try [**Boost**](https://www.drupal.org/project/boost)**(static page caching for non-logged in visitors), complemented with [Authcache](https://www.drupal.org/project/authcache) -in CR's file (for logged in members)**.

Cron job running command- 0 \* \* \* \* wget -O - -q -t 1 <http://CRON_URL>

wget -O - -q –t 1🡪basically tells the server to request a URL

0 \* \* \* \*🡪 0 indicates the minute and \*’s indicate the hour, day, month and day of the week. A wild card \* is used for specifying to run the job everytime.

DRUSH Commands: DRUSH is a command line tool to work with drupal from terminal. We have bunch of useful commands by default like downloading, enabling or even updating the module. We can also create our own drush commands to perform operations.

**Cache:**

**drush cache-clear** or **drush cc**-clears all the cache data. It allows one argument where we can specify the particular cache to be deleted.

**Cache-get**: Fetch the cached object and display it.

**Drush cache-get** or **drush cg**- It allows two arguments.

**cid-id** of the object to fetch

**bin:** The cache bin to fetch from.

Cache-get accepts **10 options** like --format,--fields like which field to output etc.

**Drush cache-rebuild**:rebuilds the site and clears all cache.

Two aliases: **drush cr** and **drush rebuild**.

**Drush cache-set**: cache an object expressed in JSON or var\_export() format and alias is **drush cs**

**Coder:**

**drush coder-format🡪**re- formats and re writes the code according to drupal coding standards.

It allows one argument i.e path of the file.

**Ex: drush coder-format sites/all/modules/coder/coder.module**

**SQL:**

**Drush sql-cli –** opens the sql client

**Drush sql-create🡪**creates the database

**Drush sql-connect🡪** string for connecting to the db

**Drush sql-drop🡪**drops all tables in a given database

**Drush sql-dump🡪**exports frupal db as sql using mysql dump or equivalent

**Drush sql-query🡪**executes a query against db

**Drush sql-sanitize**🡪runs sanitization operations on the database i.e refreshing the database.

**Drush sql-sync**🡪copies data from source to target. Transfers database dump via rsync.

**Runserver:**

**Drush runserver🡪**runs PHP’s built in http server for development alias **drush rs.**

**Pm:**

**Drush pm-disable:** disabling the module

**Drush pm-download:** downloading content from drupal.org

**Drush pm-info:** info about module

**Drush pm-enable:** enabling the module

**Drush pm-update:** update drupal core and apply any pending db updates

**Drush pm-uninstall:** uninstall one or more modules.

**Role:**

**Drush role-add-perm(rap)🡪** Grant specified permissions to a role. Arguments are role and permissions.

**Drush role-create(rcrt)🡪** create a new role. Arguments are machine name and human-readable name.

**Drush role-delete(rdel)🡪** Delete a role. Arguments are machine name.

**Drush role-list(rls)🡪** Displays all the list of roles. If a role name is provided then all the permissions of that role are listed. Arguments is role.

**Drush Role-remove-perm (rmp)🡪** Remove specified permissions from a role.

**Field:**

**Drush field-clone**🡪clone a field and all its instances. Arguments are source name and target name.

Ex: **field-clone tags labels**

**Drush field-create🡪create fields and its instances.Returns fields for url editing. Arguments are bundle and field-spec**

**drush field-create article**

**open `drush field-create article`**