

## 1 - Drupal - Content Management System

- Q What is Content Management System? Explain features & implementation
- Computer Software used to manage the creation and modification of digital content.
  - Used for enterprise content management & web content Management
  - CMS comes with 3 standard content types :-
    - Document
    - Image
    - News
  - Custom Content Types :-
    - Blogs
    - Posts
    - Product Announcements
  - Key functions of most CMS Applications
    - Storing
    - Indexing
    - Search & Retrieval
    - Format Management
    - Revision Control
    - Access Control
    - Publishing
    - Reporting
  - CMS Examples :-
    - Wordpress
    - Drupal
    - Joomla
    - Wix
    - SquareSpace
    - HubSpot
    - Sitecore
    - Magento
    - TYPO3
    - Shopify
    - BigCommerce
    - MODX
    - Ghost



What is Drupal?

free and Open Source Content Management System.

Enables user to build & Manage Websites.

Provides frameworks for organizing, managing and publishing content.

Drupal is written in PHP & uses a database to store content and configuration.

Pros & Cons :-

Pros of Drupal :-

Huge User Community :-

- Open Source CMS with a significant community of developers and enthusiasts, providing support and resources.

Accessibility Prioritized : (Prioritized Accessibility)

- Drupal is committed to accessibility, ensuring compliance with guidelines and accessibility features.

Robust Platform :-

- Drupal websites offer a solid and customizable design platform for creating and delivering content.

Cons of Drupal :-

Security Concerns :-

- Being an open-source platform, Drupal can be susceptible to security vulnerabilities if not properly maintained & updated.

Not good for Non-technical users

Not user friendly for Non technical users :-

- Drupal can be complex and challenging for non-developers, requiring technical expertise for customization and feature development.

Limited Customization Options:- (Hard to Customize)

- The availability of modules for customization in Drupal can be limited, and popular modules often come with additional costs.

Lack of Support :-

- As an open-source solution, Drupal lacks dedicated company support, requiring users to handle support and troubleshooting themselves or pay for costly support from trusted Drupal.

Scalability Challenges :- (Not so scalable)

- Flexibility and Scalability can be challenging with Drupal, requiring additional development work and specialized technical knowledge to leverage the platform fully.

Terminologies :-

**Content** : Refers to text, images, and other graphics items displayed on a website.

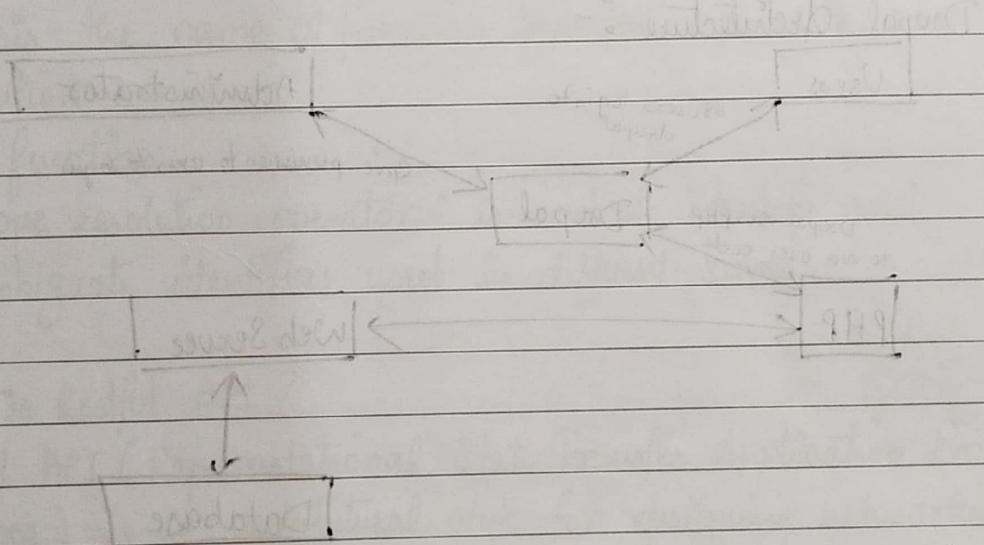
**Views** :- Created using the Drupal View module, it displays content items in various formats, such as lists, grids or tables.

- ③ Content Type (Entity Bundle, Entity Type): A grouping of fields used for a specific purpose to store and display content. Each content type represents a different type of content, like blog posts or news article.
- ④ Content Item or Node: An individual instance of a content type, representing a specific piece of content on a website.
- ⑤ Core: The central codebase of Drupal that provides the basic functionality for every Drupal website.
- ⑥ Contributed: Modules and Themes developed by the Drupal community and made available for free to other members.
- ⑦ Modules: Collection of files containing code that extend the functionality of a Drupal website. They can be core modules, contributed modules, or custom modules.
- ⑧ Themes: Sets of files that define the visual appearance and layout of a Drupal site.
- ⑨ Taxonomy: A way to classify and organize content by assigning taxonomy terms or categories to content items. It enables grouping related content & creating views based on taxonomy terms.

Menus :- Collections of links used for website navigation.

Drupal includes standard menus like the Main menu and User menu and additional menus can be created and customized.

The Drupal flow :-



Data Node - The base layer of the Drupal system where data is input and stored.

Modules - Functional plugins that extend Drupal's core functionality. They can be part of the core or contributed by the Drupal community. Modules allow customization of data items, e-commerce setup, content display, and more.

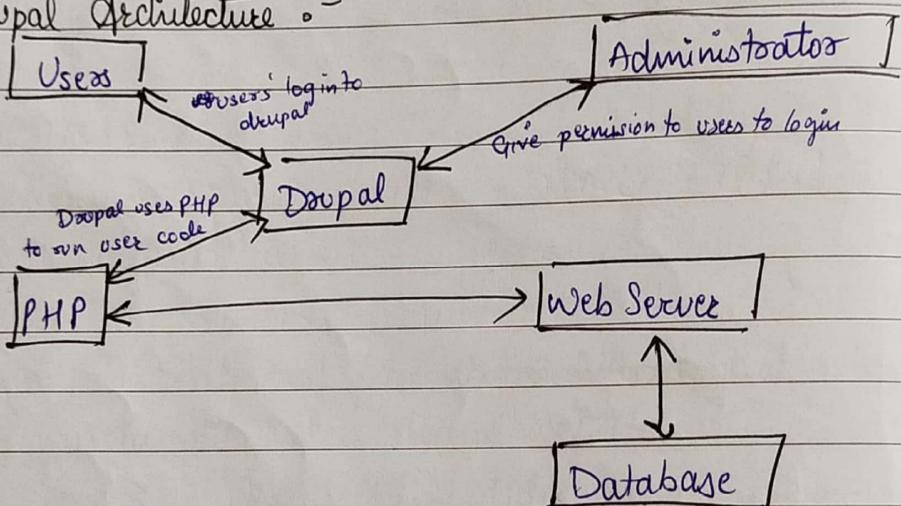
Blocks and Menus :- Blocks are output from modules or custom content that can be placed in various regions of the website's template. Menus define the navigation structure.

and links to different pages.

4. User Permissions :- assigning permissions to roles determine what users can do and see on the website.

5. Site theme/template :- Top layer of Drupal, consisting of HTML, CSS and PHP variables. The theme controls the visual appearance and layout of the website. Templates and functions can be customized to override default module outputs & provide granular control.

\* Drupal architecture :-



- Drupal is a web content management platform used for building simple and complex websites.
- The architecture of Drupal includes layers such as Users, Administrator, Drupal, PHP, Web Server, and Database.
- Users interacts with the Drupal CMS through web browsers, search engines, etc.

- The administrator manages access permissions and administers the site.
- Drupal is a free and open-source CMS built on PHP.
- PHP is used to create applications in Drupal and interacts with the web server to fetch data from the database.
- The web server handles HTTP requests and serves web pages to users.
- The database stores user information, content, and other data required for the drupal site.

★ Homework :- 17/23

- Q What is the name of function type which uses scope resolution operator?
- static function.
- The scope resolution operator : is used to identify and disambiguate identifiers used in different scopes
- Q What is Restful API ?
- RESTful API (Representational State Transfer Application Programming Interface) is an architectural style for designing networked applications.
- Restful API is an interface that two computer systems use to exchange information securely over the internet.
- Eg:- to generate monthly payslips, your internal accounts system has to share data with your customer's banking system to automate invoicing & communicate with an internal timesheet application.

Q Can you use radio buttons to select more than one option?  
How do you group radio buttons?

- No, radio buttons are ~~meant~~ designed to ~~allow~~ user to select only one option from a set of mutually exclusive choices.
- You can set the Choice Value of each option, for each button, as well as group these buttons by giving them the same group Name.

4/July/23

Q Firewalls :-

- A Network Security device to monitor & control incoming and outgoing network traffic.
- Acts ~~as~~ as barriers between internal & external networks.  
↳ helping to protect against unauthorized access & malicious activities.
- Can analyze - packets, filter traffic  
↳ blocks or allows specific types of ~~network~~ connections
- Plays crucial role in preventing unauthorized access & safeguarding sensitive information.

## OWASP Top 10 :- [ ]

### 1) Broken Access Control :-

- Allows attackers to bypass authentication safeguards and performs tasks as privileged users.
- Ability to destroy or expose sensitive information and performs network functioning.

### 2) Cryptographic failures :-

Exploits information transferred over secure communication between two parties.

Often provides useful information to attackers

Advanced persistent threat → attackers gain ~~access~~ unauthorized access to a computer network & remains undetected for an extended period.

### Injection

King of vulnerabilities

A vulnerable computer program by introducing code that changes the way of program to be executed.

Eg :- By injecting code into web application, an attacker can steal authentication ~~cookies~~ cookies and using them on online services tricking these services into thinking that the attacker is you.

- ④ Insecure Design :-
- Important consideration whenever a new application is added into network infrastructure.
  - Developers are informed to add more security patterns and principles by design.
  - Eg Solarwinds supply chain attacks :- The hackers used a supply chain attack to insert malicious piece of code into the Orion framework.  
↳ affected 3<sup>rd</sup> party service provider.

⑤ Security Misconfiguration :-

- Failure to implement security controls for a server or web applications or implementing sc but with errors.
- Human Errors :-
  - mistakes made under time pressure
  - default login credentials
  - Lack of Experience in ~~control~~ computers
  - Misinterpreting a system implementation

⑥ Vulnerable and Outdated Components :-

- The components used in web application website or application is outdated or is vulnerable itself which leads to compromising whole application.
- Developer should always know the versions of components being used & should perform regular scans
- Protective measure → remove ~~one other~~ unnecessary features, unused dependencies, components, files & documentation.

## Identification and authentication failure :-

Compromise of passwords where attackers guess users' passwords by from their identity; or any personal information;  
To prevent it:- by protection of multi-factor authentication to protect usernames & passwords.

## Software & Data Integrity failures :-

Code & infrastructure that does not protect against integrity violations can lead to software & data integrity

Eg:- Wordpress relies on plugins; one compromise plugin can lead attackers to access your application.

## Security Logging and Monitoring failures :-

Maintenance of Secure Infrastructure.

View logs regularly for acting fast in case any potential dangerous activity is noticed.

## SSRF - Server Side Request Forgery :-

An exploit where hackers access service functionality access/manipulate data/information.

## Web Crawler

A spider or web spider that is an automated script or program that systematically browses `WWW` in methodical & automated manner.

travels through web pages, follow links, & extract information for indexing or data gathering purposes.

Commonly used by Google Search Engine.

- \* TFTP :- Trivial File Transfer Protocol:-
- Simple transfer protocol operates on UDP (User datagram Protocol)
  - used for transferring files between network devices
  - lacks some of security features
  - does not provide encryption or user authentication, making it less secure

\* Socket Creation in C++ and Multithreading :-

- Client and Server socket

\* XSS

- web application vulnerability
- allows attacker to inject malicious code into webpages viewed by other users.
- The injected script is in the form of JS.
- Happens when application does not properly validate user supplied input and allows injection of untrusted code.
- Happens when a victim visits a vulnerable webpage, the injected script is executed within the browser & enables the attacker to steal sensitive information, manipulate content, or perform unauthorized actions on behalf of user.

5 | July | 2023

## \* Wireframe

- Menu details
- Login page
- How normal users, admin, chef views the page (interface page) and what all they can access.
- Progress Plan

5 | July | 23

## [Classwork]

- ↳ Homework Given
- ↳ Wireshark and firewall [explore by installing software]

## CHAPTER 4 - Configuration Management, Module and Theme Configuration

- \* Configuration → drupal.org → Configuration Management
  - ① Taxonomy vocabularies? → don't know?
  - ② Making configuration changes on a live site is not recommended.
    - because there are records that are update
    - It is easy but not good.
  - Version Control → your repository (github, etc)
  - YAML files → in XML we have nodes & values; it's same here but uses in different way.

- \* How to import / Export and Synchronize? →
  - Synchronization means making changes at both places at same time
  - Site's UVID's? → Unique User Identification
  - CLI → command line Interface

- \* Do's & Don'ts for Configuration:-

\* Theme Name :- flow to get themes?

① Get themes using CLI or by searching using theme name

\* Process to install Themes :-

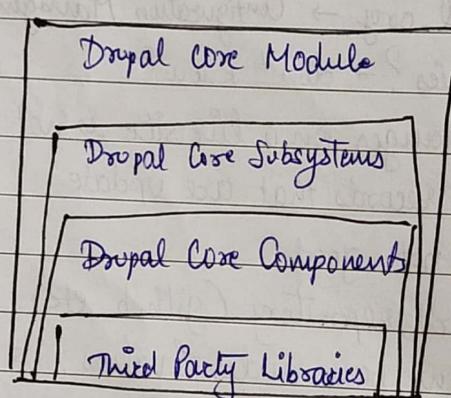
Themes can be installed using PHP.

9 | July | 23

- Chp 1 {
- Headless CMS :- All layers are bifurcated.
  - Types of Services we can use :- Tomcat, lightTPD.
  - Use of Database :- PHP MySQL, MongoDB,

Chp 2 :- Theming and Modules :-

→ Diagram for Drupal Components :-



★ Chapter 3 :- Security :-

① → Update Drupal and Modules

② Perform Regular Drupal website Backups

③ Use crafty username and passphrases for Drupal Security

\* How file Permission Numbers are Generated ?

↳ Homework Question

3

Drush → <sup>a</sup> class

→ generates forms ; generate class,

→ Invoice Page Layout

Project → ~~the~~ "Add to Cart" button & Quantity.

→ Invoice Page Layout

for Project:- Go to Manage → Content, Structure, Appearance, Extend,  
Select the theme Configuration, People, Reports, Help.

For no. of visitors → Google Analytics

Configuration → Performance → Clear caches (if any module is not visible).

Appearance → Change theme

Extend → ~~the~~ Add or Extend functionality of Modules.

→ There are different modules available but not enabled

Configuration →

People → User groups ; access or granted or revoke permissions.

Reports → Drupal version, Database, php; site errors, warnings, log messages -  
for pages, if added.

htpdocs > drupal folder > modules > custom > [create directory]

Create new folder ; then "login\_module" ← eg ↓

To find module in drupal → go to Extend > write down "login" in filter section.

OR

in sites directory there are ~~self~~ dynamically made modules by drupal.

→ In login Module folder for custom module:-

→ Src

→ login-module.info.yml [Yaml]

→ login-module.routing [Yaml]

In VS Code :-

[We can install but no functionality]

in login-module.info.yml

name: My login Module

description: login module created on \_\_\_\_\_

type: module

core-version-requirement: ^10 | ^8

package: Custom

no space

for my triple  
version →  
or just

^10

→ underscore

This module  
will be  
visible in custom module  
section

Go to Manage > Extend > Search "login"

to check your module.

★ PHP Database Connection :-

[you have the connection.php code]

9/July/23 Basic Structure:- [taught through website]

drupal/modules/custom/

↓

Src → custom php file for code

30 July 23

05 Aug 23 Complete document about Views in Drupal 9, Drupal 10 | D4DRUPAL

### \* Practice :-

#### \* What is CMS ?

- CMS (Used for creation to manage creation & modification of digital content)
- CMS comes with 3 standard content types :-
- RP of most CMS applications
  - ↳ ~~Storage~~ Access Control, Publishing, Reporting, Revision Control, Indexing, Storing, Searching & Retrieval.
- Eg → Joomla, WordPress, SilverStripe, Drupal, Wix, Hubspot etc.

#### \* What is Drupal ?

- CMS → uses PHP & uses database to store content & configure it.
- provides frameworks for used for managing, organizing & publishing content.
- used for making / designing websites.

#### \* Pros of Drupal

- Huge user community
- Prioritizes Accessibility
- Robust platform

#### Cons of Drupal

- Security concerns
- Not good for non technical users
- Hard to customize
- Lack of support
- Not so scalable

## \* Drupal Terminologies:-

- |                                      |   |
|--------------------------------------|---|
| ① Content - graphics items.          | ⑥ contributed                             |
| ② Views - RSS feeds.                 | ⑦ Modules - core, contrib, custom         |
| ③ Content type - grouping of content | ⑧ Themes                                  |
| ④ Content node or item               | ⑨ Taxonomy - classify or organize content |
| ⑤ Core                               | ⑩ Menus                                   |

## \* flow of data in Drupal / Drupal Flow :-

### ⑤ Templates

- Top layer comprising XHTML, CSS and PHP variables, controlling visual appearance of the websites. It includes functions to modify module-generated content & templates based on user permissions.

### ④ User Permissions :-

- Determine user access and actions, assigned to roles & then users.

### ③ Blocks and Menus :-

- Blocks are outputs from modules or custom content, placed in different areas of the website.
- Menus defines navigation link to different webpages.

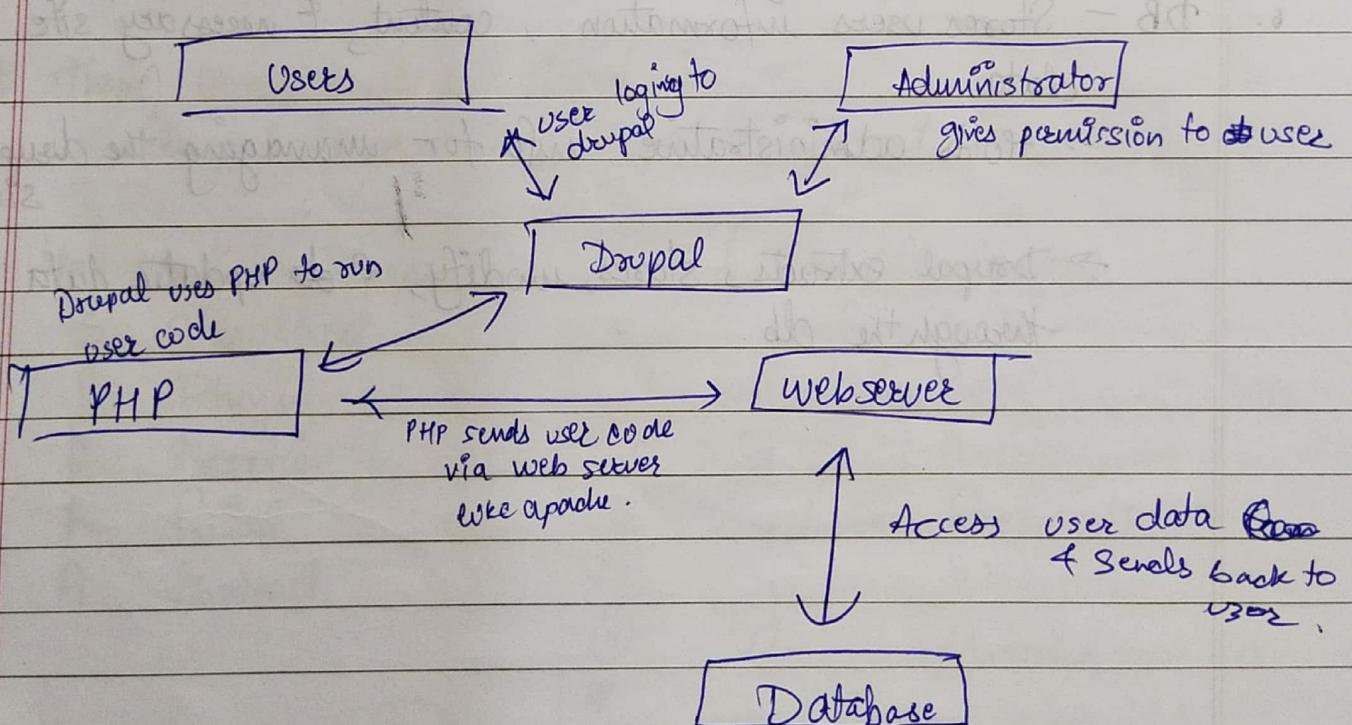
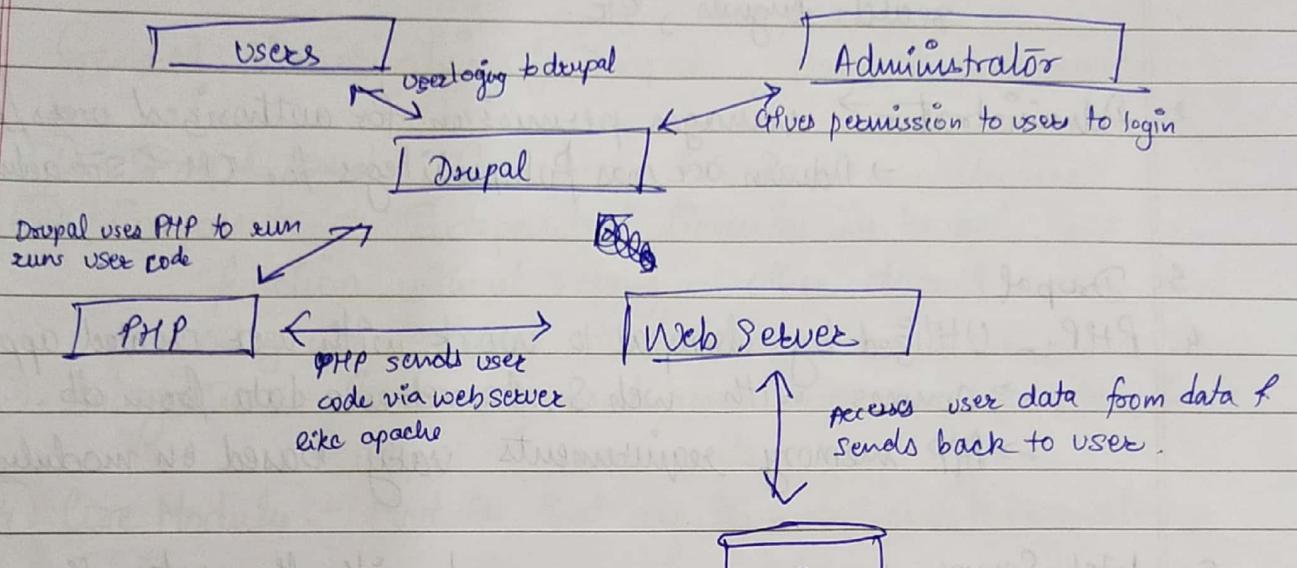
### ② Modules :-

- Functional ~~program~~ plugins that expands drupal's core functionality, allowing customization, e-commerce setup & content display.

### ① Data (Nodes, etc).

- foundation of system containing nodes or data inputs.

## \* Architecture of Drupal :-



- Page 20
1. **Users**:- Individuals who interact with the drupal CMS by sending requests to the server through web browser, search engines, etc.
  2. **Administrator** → manages permission for authorized users  
→ Admin acc has full privileges for CM & site administration
  3. **Drupal**.
  4. **PHP**:- Utilized by drupal to work with user created app.  
→ Comms with web S. to receive data from db.  
→ PHP memory requirements vary based on modules
  5. **Web Server** :- ~~to~~ users engaged with the system through HTTP, serving web pages & processing requests.
  6. **DB** - Stores users information, content, & necessary site data.  
→ Holds administrative info for managing the drupal site.  
→ Drupal extracts, stores, modify, & ~~to~~ updates data through the db.

## CHAPTER 2

★ General Visualization of Drupal / Drupal components.

- ① **Drupal kernel** :- Core part of Drupal's architecture
- ② **Components** :- Independent libraries in Drupal that can function without relying on other Drupal code, and may use external lib/packages.
- ③ **Subsystem** :- Core system elements that built upon the Kernel & components.
- ④ **Core Modules** :- Modules that are ~~not~~ essential for Drupal's functioning.
- ⑤ **API documentation** :- Detailed information about drupal core API.

→ These components are stored in :-  
`/drupal/core/lib/Drupal/Component`

→ Has various classes :- (Total 120 classes)

S Standard  
P Plugin  
E Exception  
f final  
A Abstract

Drupal Components :-

- ① Third party libraries
- ② Drupal Core Components
- ③ 11 Subsystems
- ④ 11 Modules

### \* Core Modules :-

- Date time module
- History Module
- Filter Module
- File Module
- Node Module
- Options Module

→ Comment Module

→ Contact Module

→ Image Module

→ Media Module

→ Path Module

→ Link Module

### \* Core Themes :-

- Claro Theme

→ Oliveto Theme → StarkT → Starkkit

### \* Drupal themes Folder :-

- Core > modules
  - > themes > bartik
  - > seven

- modules

- themes > contrib > zen

    |  
    > basic

    |  
    > bluemagin

    |  
    > custom > fluffiness

### \* Drupal theming with .info.yml

- name your theme

    |  
    → folder → fluffiness

    |  
    → machine name → fluffiness

[ very-fuffy ]

- small letters

- not more than 50 letters

- underscore

- unique

Base level directories :-

default.

- /core → Contains files necessary for Drupal's ~~base~~ functionality
- /libraries → Holds third-party libraries used by Drupal modules.
- /modules → Contains custom and contributed modules, grouped into subdirectories → 'contrib' & 'custom'.
- /profile → Stores contributed & custom installation profiles.
- sites / [domain OR default] / {modules, themes}
- sites / [domain OR default] / files.
- /themes :- Holds contributed & custom themes & sub-themes, structured like 'contrib' & 'custom'.
- /vendor :- Contains external libraries.

Core folder Directories :-

- /core/assets - Various external libraries used by core
- /core/includes - Base level functionality that Drupal uses through other / core folders.
- /core/lib - Drupal core classes.
- /core/misc - Frontend code that drupal core depends on.
- /core/modules -
- /core/profiles - ~~base~~ D's core installation profiles.
- /core/scripts - CLI
- /core/tests\* - D's core tests
- /core/themes - D's core themes

## UNIT - 3

\* Drupal's security ensures :-

- User Access Control
- Database Encryption
- Information sharing via security reports
- Auto-update & core validation work in partnership with Git.
- Prevention of malicious data entry.
- Mitigation of Denial of Service (DoS) attacks
- Patching of issues before they're exploited.

\* Security Measures :-

① Update Drupal and Modules :-

- Regularly update both Drupal core & modules to patch vulnerabilities and protect against attacks.

② Use trusted modules and Themes :-

- Install modules & themes only from reputable source to minimize security risks.

③ Configure Trusted host patterns :-

- Set up configuration to allow incoming requests from trusted sources only to prevent unauthorized access.

④ Perform regular backups :- Backups your website's core files & modules to quickly recover in case of an attack.

Unnecessary Bot traffic  
Update Drupal & Modules  
use trusted Modules & themes  
implement security measures.  
Block Access to imp files  
Perform security audits.

Use strong usernames & passwords :- Create unique user names and complex passwords to prevent brute-force attacks.

### Implement Security Modules :-

- Login Security Module - Limit login attempts and block IP addresses to prevent brute force attack
- Password Security Module - Configure user password policies and enforce regular password changes.
- CAPTCHA Module - Implement captcha to differentiate between human & bot login attempts.
- Security Review Module - Conduct security audits to identify vulnerabilities & recommend improvement.
- Update Manager Module - Stay informed about latest software & theme updates
- Duo - 2 factor authentication - Add an extra layer of security with two-factor authentication.
- file Integrity Check Module - Monitor changes to core files & modules for potential modifications.

Block Malicious Bot Traffic:- Prevent bad bots, scrapers, and crawlers from consuming bandwidth & potentially causing harm.

Always connect securely:- use secure connections like SFTP or SCP to protect data during file transfers.

Secure ~~the~~ drupal File Permissions :- Configure appropriate file permissions to prevent unauthorized access while allowing necessary operations.

10. Block Access to important files:-  
→ Restrict access to files like authorize.php & install.php to authorized users only.
11. Harden HTTP Security Headers :-  
→ Configure HTTP security headers to enhance browser security and mitigate various threats.
12. Use drupal malware scanner:-  
→ Regular scan for hidden malware on your website to prevent harmful consequences.
13. Deploy a web application firewall :-  
→ Implement a WAF like Arbor to continuously monitor & block various cyber threats.
14. Perform Security Audits :-  
→ Regularly conduct manual security audits to identify vulnerabilities & ensure comprehensive protection.
15. Sanitize Inputs - Validate & sanitize user inputs, especially in text fields & upload sections, to prevent SQL Injections & XSS attacks.

## CHAPTER - 4

\* flow to import, Export, Synchronize :-

→ With Configuration Manager core module you can I/E/S site configurations via Manager > Configuration > Development > Configuration synchronization.

→ To directly edit site config entities, you can use  
drush config-edit

OR

drupal config:edit

\* The 'Authenticated user' role user accounts :-

→ The 'Anonymous' user role accounts

→ You are in control

→ Additional user roles

→ Your module's permission

\* API :-

① Entity API :-

→ The entity system is the API for entity manipulation  
CRUD :

② DB API :-

### Module API :-

- core API in Drupal.
- Allows developers to create custom modules that extend Drupal's functionality.
- Devs use hooks to modify or add features, alter behaviour, & integrate their code seamlessly with the core system.

### Entity API :-

- used to create & manage navigation menus for their Drupal's site.
- Allows devs to define
- used to create & manage content entities in Drupal such as nodes, users, taxonomy terms, & more.
- provides a unified way to interact with & manipulate different types of data entities.

### Form API :-

- Allows devs. to create & handle forms in D.
- provides a structured way to build forms with validation, processing logic, & user input handling.
- helps to ~~create~~ create everything from simple contact forms to complex configuration interfaces.

#### ④ filter API in

↳ Allows administrator to define sets of text filters that process & modify user-entered content. These filters are applied to different text formats, like node content or comments, and can convert urls into links, apply HTML markup, prevent security issues like XSS attacks.

#### DB connection Code :-

<? php

```
$servername = "localhost";  
$username = "username";  
$password = "password";
```

// Create Connection

```
$conn = new mysqli($servername, $username, $password);
```

# Check connection

```
if ($conn->connect_error){
```

```
die("Connection failed: ". $conn->connect_error)
```

```
}
```

```
echo "Connected successfully";
```

```
?>
```

- 15 questions solve 14 questions / 5 marks each  
→ 1 question compulsory → Create module (Custom)  
    → How to write  
    → Proper Code  
    → Proper Indentation

After 05 Aug :-

02 - Sept ; 03 Sept

## Chapter 21 -

Plugins :-

- Plugins are small pieces of functionality that are swappable.
- Plugins perform similar functionality are the same of plugin type.
- Plugin systems have 3 base elements :-
  - ① Plugin types :-
  - ② → central controlling class that defines how the plugins of this type will be discovered & initiated. instantiated

③ Plugin Discovery within the

↳ finding plugins ~~that are available code base~~ that are qualify for use.

④ Plugin factory :-

↳ Responsible for instantiating the specific plugins chosen for a given use case

Components → Plugin Derivatives, Discovery Decorators

## Chapter 6

- ★ **Usability Testing :-** <sup>method of testing</sup>
- It is the functionality of a website, app, or digital product by observing real users as they attempt a task.

Types :-

- ① Moderated Usability Testing :-
  - A facilitator introduces tests to participant, ask them questions, answer their queries
- ② Unmoderated Usability Testing :-
  - The participants conduct the test without direct supervision, usually with a script.
- ③ Remote Usability test :-
  - The test participants conduct the test online, more rarely over the phone.
- ④ In Person UT :-
  - The test participants & the ~~researcher~~ researchers are at the same location.



## \* Usability Testing vs User Testing :-

→ User testing is a research method that uses real people to evaluate a product or service by observing their interaction & gathering information.

## \* Why usability testing is important ?

- Understand how your site works and don't get 'lost' or ~~confuse~~ confused.
- Can complete the main actions they need to
- Don't encounter usability issue or bugs
- Have a functional and efficient experience
- Notice any other usability problems.

## \* Top 8 benefits of website usability testing :-

- Validate your prototype
- Confirm your product ~~not~~ meets expectations
- Identify issues with complex flows.
- Complement and illuminate other data points
- Catch minor errors.

## \* What usability testing is not :-

### A/B testing →

→ A/B testing is a way to experiment with multiple versions of a web page to see which is most effective. While it can be used to test changes based on user testing, it is not a usability testing tool.

Focus grp :-

Heatmaps -  
Surveys -

User acceptance testing

Page 33

In-house proper use  
testing.

- Focus group :-

- focus group are a type of user testing, for which researchers gather a group of people together to discuss a specific topic. Usually, the goal is to learn people's opinions about a product or service, not to test how they use it.

- Surveys :-

- Use surveys to gauge user experience.

## ★ Paper Pattern :-

- There will be 16 questions
- 1-15 - 5 marks each - solve 14 questions
- 16 question - mandatory - 10 Marks; custom modules.

### \* Chapter 1 :-

- 2-3 questions
  - What is DCMs, implementation, key features
  - Drawbacks, Pros & cons
  - What is Headless Content Management.
  - Content, views & ~~term~~ technologies
  - Entity bundle, entity type.

### \* Chap 2 :- (core level or base directories)

- Directory structure (core, contrib, & core, profile)
- Plugins
- Drupal kernel & components (~~components~~)
- relationship between core modules & core subsystems
- Drupal 8 the most famous version, base level directories introduced in that drupal version.

## \* Chapter 3 :-

- Each individual directory ; explain , there will be 4-5 directories and then explain those directories
- The preprocessor, twig files

## \* Chapter 3 :- (3 questions - 5 marks each)

- Questions
- what regular backups are req?
- Key features / purpose of ~~Paranoia~~ module in drupal Paranoia
- Significance of help button on the extend page.
- Secure configuration - what is this.
- Configuration management & content management

## Practice :- (10 Marks questions)

- yml file - explain everything
- php → connection string
- Overall process to explain the module
- Directory Structure
- routing.yml ← to just the routing files .

## \* Chapter 4 and 5<sup>th</sup> combine 5 questions :- 5marks each.

- Significance of ~~formation~~, documentation, readme file
- Guidelines for ~~fed~~ Readme files
- Drupal modules info file
- Key components ~~selected~~ needs to be included in project page
- Usability Testing ← 1 or 2 questions
  - ↳ difference user vs usability with examples,
  - ↳ Heatmaps; A/B testing, Survey, focus groups (not usability)

## \* REVISION :-

### \* Headless CMS vs Drupal.

#### Headless CMS

Decoupled architecture

Separates the CMS (frontend) & presentation layer (backend) providing flexibility & adaptability

Content first approach

emphasizes content as primary focus, streamlining content creation & delivery, reducing project delays.

API-centric

Provides full API for accessing and displaying content, allowing content to be delivered to various platforms and channels in a highly flexible manner.

Cloud

Content as a service

#### DRUPAL

evolving towards headless CMS, but still remains some monolithic architecture.

Drupal may not prioritize this aspect as effectively.

Does offer API capabilities, but configuring it for headless CMS use may require additional time & effort.

## CHAPTER-28-

### ★ DIRECTORY STRUCTURE :-

#### ① Base level Directories :- (8)

/core :- all the files needed for the core functionality of Drupal excluding some files that have specific reason to be placed in the root (" / ") directory.

/libraries - stores all third party external libraries that drupal uses,

- This directory is not included in Drupals' core installation but utilized by various contributed modules.

/modules - directory where all custom & contributed modules are stored

- To organize you can create contrib and custom subfolders in the directory to keep track of your module.

/profile - All contributed & custom installation profiles.

/sites/[domain or default]/[modules, themes] -

keeps sites-specific modules and themes at specific place, this prevents from appearing on every site and provides more flexibility in managing sites functionality & designs.

/sites/[domain or default]/files - used to store site specific files, including user uploaded content like images and site configuration files, both active & staged

/themes - All contributed, custom & subthemes  
- To organize, you can create custom and contrib subfolders to keep track of your themes.

/vendor - has backend external libraries that drupal core is dependent on, for eg symfony & twig are essential for drupal's functionality.

② Core folder Directories :- (9)

/core/scripts - Various CLI scripts, mainly used by developers

/core/lib - Drupal core classes

/core/includes - Base level functionality that drupal uses through core folders

/core/misc - frontend code that drupal core depends on

/core/module - Drupal's core modules

/core/profiles - Drupal's core installation profiles. These are Minimal, Standard and Testing.

/core/assets - Various external libraries used by core

/core/tests - Drupal core test

/core/themes - Drupal's core themes

③ Composer Build Directories :-

/vendor - various external composer php packages

/web - parent directory for above base level and core directories

## ★ PLUGINS → Pg 30

- Small pieces of functionality that are swappable
- These functionalities are much similar to Plugin type

1. **Plugin Type** → ~~specifies~~ defines spec of plugins that describes how they are discovered and created
2. **Plugin** → They categorized plugins as centralized function
2. **Plugin Discovery** → Process of searching & identifying plugin from the code base that can be used within the specific plugin type.
3. **Plugin Factory** :- ~~factory~~ It is responsible for creating instances of specific plugins that is selected for particular use case.

### - Some useful Components :-

- **Plugin Derivatives**, **Discovery Decorators** & **Plugin Mapper** are powerful tools that can be used to create more flexible & user friendly plugin system.

**Plugin Derivatives** → allows a single plugin to get placed of many  
→ this can reduce the no. of plugin needed & make it easier for administrator to manage them.

**Discovery Decorators** → used to ~~add~~ add additional functional to existing discovery method such as caching

Plugin Mapper → Allows you to map something, this make it easier to create & configure plugin.

## \* Drupal Kernel and Components :- Pg 21

- ① Kernel is the core heart of Drupal  
↳ Responsible for handling requests, loading modules & themes and managing db.

## \* ② Third Party Libraries :-

- These parties are not developed by the Drupal community.
- often used to provide additional functionality or features to drupal website.
- Common third party libraries commonly include :-
  - Jquery      Bootstrap      FontAwesome      Slick
  - Swiper

## ③ Drupal Core Components :-

- independent lib developed by Developers.
- May not depend on Drupal code but dependent on Drupal Components or external libraries.
- used to provide essential functionality to Drupal website.

## ④ DC Subsystems :-

- Dependent on Drupal kernel & Drupal Core Components.
- set of Drupal core modules that provide a specific functionality.

## ⑤ DC Modules :-

- Has ability to create & manage content, users & themes.

## CHAPTER 3

Q Why regular backups are required?

(i) **Data Security** :- Regular backups can ensure that your website's data is safe and can be restored in case of a data breach, data loss or corruption.

- This safeguards your website's valuable information.

(ii) **Disaster Recovery** :- Backups serve as the safety net in the event of a catastrophic event, such as a cyberattack, server failure, or accidental data deletion.

-> Enable quick recovery

(iii) **Version Control** - Backup includes ~~the~~ Drupal core & module files, allowing you to rollback to previous version if there is any issues found while making any change in the site, this helps in maintaining site stability.

(iv) **Testing Env.** - Backups can be used to create sandbox environments for testing, updates & changes before making the site live. This helps to reduce bugs or issues.

(v) **One-Click Recovery** - Pantheon offers one-click backup & restore solution which helps to simplify the recovery process. Can help quickly rollback to previous stable state.

## \* Relationship between Core Modules & Core Subsystems :

### ① Core Modules :-

- Set of pre-built features & functionalities that are included as the part of Drupal core package.
- Has essential features like content management, user authentication, & basic site building tools.
- Designed to be general-purpose & can be enabled & disabled from site-administrators.
- Can be found under /core/modules

### ② Core Subsystems :-

- Foundational components of the Drupal system that provide a framework for various functionalities & features.
- Supports core modules and ensure they work together.
- Can be found in /core/lib/Drupal/core
- They define set of rules, API, and conventions that core modules adhere to, making it easier to develop & maintain Drupal website.

## \* Preprocessor function in Twig file :-

- It is a PHP function used to preprocess variables before they pass to twig template for rendering.
- Integral part of Drupal theming and allows you to modify, enhance or manipulate variables to tailor the output of your theme.

## A Key features & Purpose of Paranoia Module:-

→ Paranoia Module is a security tool designed to identify common vulnerabilities such as PHP SQL Injection.

### Purpose :-

→ Mitigate PHP SQL Injection :-

↳ Most common security vulnerability in Drupal.

↳ It achieves this by blocking & evaluating PHP code executed via Drupal interface.

### Key Features :-

#### ① Blocking Privilege Escalation :-

↳ Prevents attackers from using "use PHP" for block visibility privilege to gain higher access.

#### ② Disabling PHP filter :-

It stops PHP filter from creating input formats to prevent PHP execution, which is a major risk.

#### ③ Protecting Main Configuration :-

↳ restrict unauthorized access to core configuration.

#### ④ Restricting Risky Permissions :-

↳ It enforces restrictions on permissions, ensures only authorized users can modify vital settings.

#### ⑤ Non-Disabling Security :-

This module cannot be disabled without proper authorization.

- \* Significance of help button on Extend Page :-
  - Provides users valuable information about the various modules & themes available for installation.
  - Extend tab helps to
    - manage modules & themes
    - quick access to documentation on how the module works, features, purpose etc.
  - Provides user guidance to users who are not familiar with all available extensions.
- Reduce confusions → vast amount of modules & themes are available on the Drupal ; this helps users to select a proper module & theme for their site.
- Effective decision Making
  - ↳ Documentation helps user to by guiding
- Enhance User Experience :- 'help' button helps them to explore & evaluate available extensions with confidence.

|            |                      |                   |
|------------|----------------------|-------------------|
| Complexity | $O(N!)$              | Factorial         |
|            | $O(N^N)$             | Exponential       |
|            | $O(N^3)$             | Cubic             |
|            | $O(N^2)$             | Quadratic         |
|            | $O(N \times \log N)$ | $N \times \log N$ |
|            | $O(N)$               | Linear            |
|            | $O(\log N)$          | Logarithmic       |
|            | $O(1)$               | Constant          |

### ⑤ Storage & Active Configuration :-

→ Active site configuration is stored in db for performance & security.

### ⑥ Import & Export in YAML :-

Configurations can be imported & exported in YAML using drush & drush console.

### ⑦ Environments & Deployments :-

Configuration changes can be safely moved between different environments.

### ⑧ Environment & Consistency :-

Drupal checks for ensures configuration consistency between environments using SITESINI.DRUPAL.

### \* ReadMe file :- (guidelines)

#### ① Provide an Overview :-

→ what the module does & how someone can use it.

→ Concise summary of module's purpose and functionality

#### ② Repeat Synopsis :-

→ Repeat the synopsis found in Drupal's module page

→ Ensure description & clarity of module.

#### ③ Use additional files for Complexity :-

→ Drupal contributed module becomes lengthy or complex  
information ; so, it's better to split it in  
Readme file & Install file.

## \* Configuration Management :-

- fundamental aspect while deploying a website settings and configurations.
- Involves storing and managing sites' configuration data settings.
- The following are the key points to understand :-

### ① Consistent Configuration Storage :-

- Drupal stores configuration data in a consistent manner.
- This means everything related to your site is stored in consistent manner / organized and structured uniformly.
- Includes various components :- content types, fields & other site configurations.

### ② Avoiding <sup>live</sup> Site Changes :-

- Making configuration changes directly on a live site is not recommended.
- because such changes made can cause site's unexpected behaviour

### ③ Development and Deployment Behaviour :-

- This involves taking the live site's configuration, testing proposed changes in the local development env, and then exposing these changes to configuration files.

### ④ Integration with version -

Configuration changes can be stored in the codebase & tracked with version control like Git

## ④ Support & Collaboration :-

→ Documentation fosters collaboration with the drupal community.

## ★ Usability Testing

### Codes

↳ folder location → /drupal/modules/custom/[module name]  
↳ Directory Structure :-  
custom/register

base src  
↳ register.info.yml  
↳ register.install  
↳ register.module  
↳ register.routing.yml  
\$ src

↳ Controller  
↳ UserController.php  
↳ Form

① info.yml.

Name : name : Register Module

Type : type : module

Desc : description : User Registration

package : package : custom

version : version : 1.0

core : core : 8.x

core-version-requirements : ^8 || ^9 || ^10

compatible with drupal 8

⑨ Use Unix-style line endings.  
 → Make sure Readme; Install & other plain text files use Unix style line endings.

⑩ Format for 80 characters :-  
 → File should be formatted to hand wrap text at 80 characters per line.  
 → Maintains readability & consistency.

⑪ Use README template :-  
 → Readme file should be based on Readme template.

## \* SIGNIFICANCE OF DOCUMENTATION :-

① User Guidance & Clarity :-  
 → Documentation provides users with clear & structure ~~and~~ guidance how to install, configure & use Drupal Module  
 → Helps users understand modules' purpose  
 → Important for modules with ~~such~~ complex features.

② Enhanced User Experience :-  
 → Proper documentation helps users to quickly find their answers, troubleshoot issues,  
 → saves times & frustrations.

③ Troubleshooting & Issues Resolution.  
 → Common issues/questions can be mentioned in documentation which helps users to solve their easily.

### ② install

- used for db related tasks when module is installed or uninstalled.
- you can define db schema changes, create tables, insert default values

### ③ routing

- defines routes for your module
- defines how URLs are mapped to controllers & pages within your module.
- ~~defines~~ path specifies the path, the controller to be used, access permissions & other routes related settings.

### ④ form.php

- has PHP classes that define forms for your module.
- handles rendering, submission & validation.

### ⑤ Controller :-

- responsible for handling requests & retrieving responses.

\* UT & Chap 1

subtask assigned : 9/10

subtask : 9/10

incomplete task : 9/10

notes : 9/10

O.L : 9/10

X.B : 9/10

Q1^ || P^ || 8^ : 9/10

Chances - 9/10

## \* README FORMAT :-

1. Project Name :- Write project name at the first line with a large '#' symbol.
2. Heading :- Use capitalize heading with initial capital. Indicate level of heading with one or more '#' symbols.  
for eg → "# Project Overview" - top level heading  
"## Installation" - second level heading  
(###)
3. Spacing - Leave two blanks before second & third level heading (# ##)
4. Lists - use bulletin list (-) & ordered list '1'
5. Links - Use meaningful links "Drupal" instead of URLs
6. Text width - Keep the text around 80 columns for readability.