1. What is Drupal, and how does it differ from other content management systems (CMS)?

Drupal is a free and open-source content management system (CMS) written in PHP and distributed under the GNU General Public License. Drupal is a popular choice for building websites and web applications of all sizes, from small personal blogs to large enterprise websites.

Drupal is known for its flexibility, scalability, and security. It is a modular system, which means that it can be extended and customized with a wide variety of modules and themes. Drupal is also a highly secure platform, thanks to its active community of developers who are constantly working to identify and fix security vulnerabilities.

Some of the features that make Drupal a popular choice for website development include:

**Flexibility**: Drupal is a highly flexible platform that can be used to build a wide variety of websites.

**Scalability**: Drupal is a scalable platform that can be used to build websites of all sizes.

**Security**: Drupal is a secure platform that is constantly being updated to fix security.

**Community**: Drupal has a large and active community of developers who are always willing to help.

**Multilingual Support**: Drupal has built-in multilingual capabilities, allowing websites to be translated into multiple languages. It provides tools for content translation, language detection, and multilingual site management, making it an attractive choice for organizations with global audiences

If you are looking for a flexible, scalable, and secure CMS, Drupal is a great option.

Here are some examples of websites that are built on Drupal:

-The Economist

-The White House

-The New York Times

-The Guardian

-UNESCO

-Indian Government

-NASA

1. What are the key features and improvements introduced in Drupal 9 compared to Drupal 8?

Drupal 9, released in June 2020, builds upon the foundation of Drupal 8 while introducing several key features and improvements. Here are some of the notable changes in Drupal 9 compared to Drupal 8:

1. **Updated Dependencies**: Drupal 9 updates the underlying dependencies of the core system, including Symfony(Symfony is a popular PHP web application framework that follows the Model-View-Controller (MVC) architectural pattern. Symfony is composed of many standalone and reusable components, which can be used independently in any PHP project), Twig(Twig is a powerful and flexible templating engine for PHP, primarily used in web development within the Symfony framework Control structures, such as loops and conditionals, are written using tags like **{% %}**. For example, **{% if condition %} ... {% else %} ... {% endif %}**.),and other third-party libraries. This ensures compatibility with newer versions and provides better performance, security, and maintainability.
2. **Deprecated Code Removal**: Drupal 9 removes deprecated code from Drupal 8, meaning that modules and themes compatible with Drupal 8 are more likely to work with Drupal 9. This removal streamlines the code base and encourages developers to adopt modern best practices.
3. **Compatibility with Contributed Projects**: With the removal of deprecated code and updated dependencies, contributed modules and themes developed for Drupal 8 can be easily upgraded to Drupal 9. This reduces the effort required for developers to keep their projects up to date.
4. **Cleaner and More Efficient Code base**: Drupal 9 focuses on improving code quality, reducing technical debt, and optimizing performance. It introduces various refinements and enhancements to make Drupal more efficient and easier to work with for developers.
5. **Improved Upgrade Process**: Upgrading from Drupal 8 to Drupal 9 is smoother compared to previous major version upgrades. Since Drupal 9 retains backward compatibility with Drupal 8, the upgrade primarily involves removing deprecated code and updating dependencies. This allows for a more seamless transition for Drupal 8 sites.
6. **Enhanced Media Management**: Drupal 9 introduces improvements to the media management system with the stable Media Library module. It provides a user-friendly interface for managing and reusing media assets, such as images and videos, across the site.
7. **Easy Accessibility**: Drupal 9 puts a strong emphasis on accessibility by incorporating accessibility improvements from Drupal 8.8 and 8.9, ensuring that websites built on Drupal are more inclusive and usable for individuals with disabilities.

**8) Automated Updates**: Drupal 9 offers automated update capabilities for both the core software and contributed modules. This simplifies the process of keeping Drupal sites up to date with the latest security patches and feature enhancements.

1. New features of Drupal 9?

C:\xampp\apache\conf\extra => vhosts

C:\Windows\System32\drivers\etc => hosts

Drupal 9 is the latest version of the popular open-source content management system (CMS). It was released in June 2021 and is a major upgrade over Drupal 8. Drupal 9 introduces a number of new features and improvements, including:

**Twig integration:** Drupal 9 now uses Twig as its templating engine, replacing the previous PHP-based system. This makes it easier to create and maintain custom templates, and it also makes Drupal more compatible with other modern web development technologies.

**Improved performance:** Drupal 9 has been optimized for performance, making it faster and more responsive than previous versions. This is due in part to the use of Twig, as well as other performance improvements such as lazy loading of assets.

**Enhanced security:** Drupal 9 has been hardened against security vulnerabilities, making it a more secure platform for running websites. This is due in part to the use of Symfony 4, which is a more secure framework than the previous version of Symfony used by Drupal 8.

**New features and modules:** Drupal 9 includes a number of new features and modules, such as a *new block layout system, a new media library, and a new translation system*. These new features make it easier to build and manage complex websites.

1. Can you explain the role of a front-end developer in Drupal development?

A front-end developer is responsible for the user interface (UI) and user experience (UX) of a Drupal website. They use HTML, CSS, and JavaScript to create the visual elements that users see and interact with on the website. Front-end developers also work with back-end developers to integrate the UI and UX with the underlying Drupal code.

**Theme Development:** Front-end developers in Drupal primarily focus on theming. They take the design assets provided by the UI/UX designers and convert them into functional and visually appealing Drupal themes. This involves writing HTML, CSS, and JavaScript code to create the structure, styles, and interactivity of the website.

**Templating:** Front-end developers work with Drupal's templating system, which utilizes the Twig templating engine. They create and customize Twig templates to control the display of various Drupal components, such as nodes, blocks, views, and forms. Templating involves organizing and presenting the data from the back-end in a user-friendly manner.

**Responsive Design:** Front-end developers ensure that Drupal websites are responsive and optimized for different screen sizes and devices. They use responsive design techniques, such as media queries and flexible layouts, to adapt the website's appearance and behavior to different screen resolutions, including desktops, tablets, and mobile devices.

**Cross-Browser Compatibility:** Front-end developers test and ensure that Drupal websites function correctly and display consistently across different web browsers (such as Chrome, Firefox, Safari, and Internet Explorer) and their various versions. They address browser-specific quirks and implement necessary fallbacks or workarounds when needed.

**Accessibility:** Front-end developers play a crucial role in making Drupal websites accessible to all users, including those with disabilities. They follow accessibility guidelines and best practices to ensure that the website's content and functionality are perceivable, operable, understandable, and robust for individuals using assistive technologies.

**Performance Optimization**: Front-end developers optimize the performance of Drupal websites by optimizing code, compressing and minifying assets (HTML, CSS, JavaScript), and implementing techniques like lazy loading and caching. They aim to reduce page load times and improve the overall user experience.

**Integration of Front-end Libraries and Frameworks**: Front-end developers integrate and work with popular front-end libraries and frameworks (e.g., jQuery, Bootstrap, React, Vue.js) within Drupal projects. They leverage these tools to enhance interactivity, create dynamic components, and improve the user interface.

**Collaboration with Back-end Developers and Designers**: Front-end developers closely collaborate with back-end developers who handle the Drupal site's functionality and data management. They also collaborate with UI/UX designers to ensure that the front-end implementation aligns with the design vision and effectively communicates the website's purpose and message.

1. Can you explain the theming system in Drupal and how you customize and create themes?

The theming system in Drupal allows you to customize the appearance and presentation of your Drupal website. It separates the design and layout from the underlying functionality, enabling you to create unique and visually appealing themes. Here's an overview of how you can customize and create themes in Drupal:

**Understanding the Drupal Theming Architecture:** Drupal uses a template-based theming system. Templates are files that control the HTML markup and structure of different components, such as pages, blocks, nodes, views, and forms.

Drupal 8 and Drupal 9 use the Twig templating engine for creating templates. Twig provides a flexible and secure environment for theming.

**Identifying the Base Theme:** Before creating a custom theme, you can start with an existing base theme, such as the Drupal core themes (like Bartik or Stable) or contributed themes (like Bootstrap or Zen).

Base themes provide a foundation with pre-built templates, styles, and functionality that you can leverage and extend in your custom theme.

**Creating a Custom Theme Directory:** In the "themes" directory of your Drupal installation, create a new directory for your custom theme. For example, "themes/custom/my\_theme".

**Creating Theme Files:** Inside your custom theme directory, create a file called "my\_theme.info.yml" (replace "my\_theme" with your theme's name).

This file contains metadata and settings for your theme, such as name, description, base theme, regions, stylesheets, scripts, and more. You can also specify the template overrides you want to apply.

**Creating Template Overrides:** To customize the HTML structure and presentation of Drupal components, you can create template override files.

Copy the original template file from the base theme or Drupal core to your custom theme directory and modify it according to your needs. Maintain the same file structure and naming conventions.

For example, to override the node template, copy "node.html.twig" from the base theme or core to your custom theme's template directory and modify it as required.

**Modifying CSS and JavaScript:** To style your custom theme, create or modify CSS and JavaScript files.

In the .info.yml file, specify the CSS and JavaScript files to include in your theme.

You can use pre process functions in your theme's .theme file to add custom classes, alter variables, and apply dynamic changes to the rendered output.

Enabling and Configuring the Custom Theme:

Log in to your Drupal site as an administrator.

Go to the "Appearance" page (admin/appearance) and find your custom theme in the list of available themes.

Enable the theme by selecting the corresponding checkbox and save the configuration.

Configure the theme settings, such as regions and block placements, based on your specific layout requirements.

**Clearing the Cache:** After enabling or modifying the theme, Drupal needs to rebuild its cache to reflect the changes.

You can clear the cache by visiting the "Performance" page (admin/config/development/performance) and clicking the "Clear all caches" button.

By customizing and creating themes in Drupal, you can control the look and feel of your website, adapt it to your branding requirements, and provide a unique user experience. Remember to follow best practices, leverage Drupal's theming documentation and community resources, and test your theme across various devices and browsers to ensure compatibility and consistency.

1. Can you explain the concept of Drupal blocks and how you create and customize them in front-end development?

In Drupal, a block is a self-contained piece of content that can be placed in a region on a page. Blocks can contain a variety of content, such as text, images, menus, and forms. They can be created by modules or by site administrators.

Once you have created a block layout, you can add the block to a region on a page. To do this, you need to go to the Structure > Block layout page. In the Block layout page, you will see a list of all the regions on your page. To add a block to a region, simply click on the Place block button next to the region.

Once you have selected the block and the pages that you want to display it on, click on the Save button. The block will be added to the region and will be displayed on the pages that you selected.

Here are some examples of blocks that can be created in Drupal:

**Promotional blocks**: These blocks can be used to promote your products or services. For example, you could create a block that displays a list of your latest blog posts or a block that displays a rotating image slider of your products.

**Navigation blocks**: These blocks can be used to provide your site visitors with a way to navigate your site. For example, you could create a block that displays a site map or a block that displays a list of your most popular pages.

**Contact blocks**: These blocks can be used to provide your site visitors with a way to contact you. For example, you could create a block that displays your contact information or a block that displays a contact form.

**Social media blocks**: These blocks can be used to connect with your site visitors on social media. For example, you could create a block that displays your latest tweets or a block that displays your Facebook likes.

Advertising blocks: These blocks can be used to display advertising on your site. For example, you could create a block that displays a list of your latest Google AdSense ads.

**Custom blocks**: You can also create custom blocks that display any type of content that you want. For example, you could create a block that displays a weather forecast or a block that displays a countdown timer.

1. How to create module in drupal 9?

To create a custom module in Drupal 9, you will need to:

1. Create a directory for your module in the *web/modules/custom* directory.
2. Create an *info.yml* file in the root of your module directory. This file will contain information about your module, such as its name, description, and version.
3. Create a module.php file in the root of your module directory. This file will contain the main logic of your module.
4. Create any other files that your module needs, such as templates, CSS files, or JavaScript files.

Once you have created all of the necessary files, you can enable your module in Drupal. To do this, go to the admin/modules page and click the Enable button next to your module.

1. What exactly is PDO?

PHP Data Objects (PDO) is the acronym for **PHP Data Objects**. **It is a reliable way to access the database.** **It makes it easy for developers to write code.** **It's related to a single API-based data access layer**. Drupal has a database abstraction layer that allows developers to interact with multi-database servers more simply.

It's utilized to keep SQL's syntax and power while working with more complicated features. It uses security checks and best practices to give a defined interface for dynamic queries. This technique is built on top of the PDO database API.

1. In Drupal, what are hooks?

In Drupal, **hooks are functions** that are *called at specific points in the execution of the Drupal code*.***They are used to allow modules to interact with each other and with the Drupal core code.***

Hooks are a powerful way to extend the functionality of Drupal. By implementing hooks, modules can add new features, change the behavior of existing features, and even override the core Drupal code.

There are many different types of hooks in Drupal. Some of the most common hooks include:

**hook\_init()**: This hook is called when Drupal starts up. It can be used to initialize module-specific data and settings.

**hook\_menu()**: This hook is used to define the menus for a module.

**hook\_theme()**: This hook is used to define the theme functions for a module.

**hook\_node\_insert()**: This hook is called when a new node is created. It can be used to modify the node before it is saved to the database.

**hook\_form\_alter()**: This hook is used to alter the form for a specific entity.

1. Explain how caching works in Drupal?

Drupal allows to **speed up websites** using different shapes like **page** **caching**, **block caching**, and **lifetime** for cached pages through caching.

**Page Caching**: It enables the entire HTML of each page to be stored in the database. It reduces the number of queries needed.

**Block Caching**: You can set the cache settings for a block in views when a block is created by views.

**Minimum cache lifetime**: It is the amount of time before the page cache is cleared. On each run, page caches are cleared.

**Expiration of cached pages**: It only applies to an external mechanism, for example, your browser cache or varnish.

11. Tell me, what is Drush?

Drush is a **command-line shell and scripting interface for Drupal**. It is a powerful tool that can be used to perform a variety of tasks,

**Including:**

-Managing Drupal sites

-Executing SQL queries

-Running cron jobs

-Generating code And much more

Drush is a free and open-source project that is maintained by a community of developers. It is available for all major platforms, including Linux, Windows, and macOS.

Here are some of the benefits of using Drush:

-It can ***save your time by automating tasks*** that would otherwise be done manually.

-It can help you to ***avoid errors*** by providing a consistent and reliable way to perform tasks.

-It can make **it *easier to collaborate with others*** by providing a common way to interact with Drupal sites.

1. Explain the concept of regions and blocks in Drupal.

In Drupal, a **block is a piece of content that can be displayed in a region on a page**. Blocks can be anything from **text** and **images** to **menus** and **widgets**. They are a powerful way to customize the look and feel of your Drupal site.

A region is an area on a page where blocks can be placed. Regions are defined by the theme you are using. Each theme has its own set of regions. Some common regions include:

-Header

-Footer

-Sidebar

-Main content

Structure > Block layout. In this page, you will see a list of all the blocks that are available for your site. You can also see a list of all the regions that are available in your current theme.

To add a block to a region, simply drag and drop the block into the desired region. You can also change the order of the blocks in a region by dragging and dropping them.

Once you have added blocks to regions, you can preview your changes by clicking the Preview button at the top of the page.

Here are some additional tips for using blocks:

1. Use blocks to add content that is not part of your main content. This could include things like menus, widgets, and social media buttons.
2. Use blocks to create a consistent look and feel across your site. By using the same blocks in different places, you can create a sense of unity and coherence.
3. Use blocks to make your site more user-friendly. By placing blocks in strategic locations, you can make it easy for users to find the information they need.
4. What are the different types of modules in Drupal?

A module is a set of codes that extend Drupal features and functionality.

There are 3 main types of modules in Drupal: core modules, contributed modules, and custom modules.

**Core modules:**  are comes with every Drupal installation. They provide basic functionality, such as user management, content management, and navigation.

**Contributed modules:**  are created by third-party developers and are available for download from the Drupal website. They provide a wide range of additional functionality, such as e-commerce, social media integration, and multilingual support.

**Custom modules:** are created specifically for a particular Drupal site. They can be used to add custom functionality that is not available in core or contributed modules.

In addition to these three main types, there are also a number of other types of modules, such as:

**Deprecated modules:** are modules that are no longer supported by the Drupal community. They may still work, but they may not be compatible with newer versions of Drupal.

**Obsolete modules:**  are modules that have been removed from the Drupal website. They cannot be installed or used.

When choosing modules for your Drupal site, it is important to consider your needs and requirements. Core modules provide a good starting point, but contributed modules can add a lot of additional functionality. If you need custom functionality, you can work with a developer to create a custom module.

Here are some of the most popular **core modules:**

**User -** This module provides basic user management functionality, such as creating and editing user accounts, assigning roles, and managing permissions.

**Node** - This module provides basic content management functionality, such as creating and editing content types, adding fields to content types, and managing nodes.

**Field** - This module provides a flexible field system that can be used to add custom data to content types.

**Views** - This module provides a powerful content management system that can be used to create custom views of content.

**Menu** - This module provides a flexible menu system that can be used to create custom menus.

Here are some of the most popular **contributed modules**:

**Pathauto** - This module automatically generates aliases for your content, making it easier for users to find your content.

**Search API** - This module provides a powerful search engine that can be used to search your content.

**Views Infinite Scroll** - This module allows users to scroll through your content infinitely without having to reload the page.

**Rules** - This module provides a powerful rule engine that can be used to automate tasks and add custom logic to your site.

**CKEditor** - This module provides a powerful WYSIWYG editor that can be used to create and edit content.

These are just a few of the many modules that are available for Drupal. With so many options to choose from, you can easily find the modules that you need to create the perfect Drupal site.

1. How do you install and configure drupal?

To install and configure Drupal, you can follow these general steps:

**System Requirements**: Ensure that your web server meets the system requirements for running Drupal. These requirements typically include a web server (e.g., Apache or Nginx), a supported version of PHP, and a database server (e.g., MySQL or PostgreSQL).

Additionally, make sure you have a domain name or a local development environment set up.

**Download Drupal**: Visit the official Drupal website (https://www.drupal.org/) and download the latest stable release of Drupal core. Extract the downloaded files to a directory accessible by your web server.

**Create a Database**: Set up a database for your Drupal installation. This usually involves creating a new database and a database user with appropriate privileges. You can use tools like phpMyAdmin or the command line to manage your database.

**Start Installation**: In your web browser, navigate to the directory where you extracted Drupal files. You should see the Drupal installation wizard. Select your preferred language and click on the "Save and continue" button.

**Verify Requirements**: Drupal will check if your server environment meets the necessary requirements. If any issues are found, address them before proceeding. Once all requirements are met, click on "Save and continue."

**Database Configuration**: Provide the database details you created in Step 3, including the database type, database name, username, and password. You can also customize the database prefix if desired. Click on "Save and continue."

**Install Profile**: Choose an installation profile based on your needs. The "Standard" profile is suitable for most general-purpose websites. Click on "Save and continue."

**Site Configuration**: Enter your site name, email address, and site maintenance account details. You can also configure additional options such as site slogan, default country, and time zone. Click on "Save and continue."

**Installation Progress**: Drupal will start the installation process, creating the necessary database tables and configuring your site. This may take a few moments. Once the installation is complete, you'll be directed to the "Congratulations" page.

**Configure Site**: On the "Congratulations" page, you can configure additional settings like site appearance, modules, and permissions. It's recommended to review and adjust these settings according to your requirements.

**Clear Cache**: After configuring your site, it's a good practice to clear the Drupal cache. This ensures that your changes take effect. You can do this by navigating to the "Performance" section in the admin interface and clicking on the "Clear all caches" button.

1. How do you create and manage content in Drupal?

To create and manage content in Drupal, you need to have access to the Drupal administration pages. Once you have access, you can follow these steps:

-Go to the Content page in the administration menu.

-Click on the Add content button.

-Select the content type that you want to create.

-Fill in the form with the content that you want to create.

-Click on the Save button.

1. What is a patch in drupal?

A patch in Drupal is a **text file that contains a list of changes to be made to a Drupal module or theme**. Patches are used to fix bugs, add new features, or make other changes to Drupal code.

**To apply a patch** in Drupal, use the **drush patch command**. For example, to apply the patch file my-patch.patch, you would use the following command:

drush patch my-patch.patch

//Patch applied successfully.

//If the patch fails, you will see an error message.

You can also apply patches manually by editing the Drupal code and then using the drush pm-update command to update the modules or themes.

Here are some of the benefits of using patches in Drupal:

Patches can be used to fix bugs in Drupal modules and themes..

1. What is the purpose of the .info.yml file in a Drupal theme?

The .info.yml file is a YAML file that **contains metadata about a Drupal theme**. It is used by Drupal to understand the theme and to display information about it in the administrative interface.

The .info.yml file contains the following information:

-The theme name

-The theme description

-The theme version

-The theme dependencies

-The theme assets

-The theme settings

The .info.yml file is a required file for all Drupal themes. It is located in the theme's root directory.

Here is an example of a .info.yml file:

**Code snippet**

name: My Theme

description: A simple theme that changes the look and feel of Drupal.

core: 8.x

version: 1.0

dependencies:

- drupal: 8.x

libraries:

- core/drupal.theme

This .info.yml file tells Drupal that the theme is named "My Theme", that it is a simple theme that changes the look and feel of Drupal, that it is compatible with Drupal 8.x, and that it has no dependencies. It also tells Drupal that the theme uses the core/drupal.theme library.

The .info.yml file is a valuable tool for developers who create and maintain Drupal themes. It makes it easy for Drupal to understand the theme and to display information about it in the administrative interface.

1. What is cron job in drupal?

The term 'Cron' refers to the automated tasks that your site runs periodically. It takes care checking whether or not updates are available for Drupal core, and for your contributed modules, and themes.