What is Git?

Git is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

Git is foundation of many services like **GitHub** and **GitLab**, but we can use Git without using any other Git services. Git can be used **privately** and **publicly**.

Features of Git

Some remarkable features of Git are as follows:

Scalability
Security
Open Source
Features of Git
Speed

- **Open Source**
  Git is an **open-source tool**. It is released under the **GPL** (General Public License) license.
- **Scalable**
  Git is **scalable**, which means when the number of users increases, the Git can easily handle such situations.
- **Distributed**
  One of Git's great features is that it is **distributed**. Distributed means that instead of switching the project to another machine, we can create a "clone" of the entire repository. Also, instead of just having one central repository that you send changes to, every user has their own repository that contains the entire commit history of the project. We do not need to connect to the remote repository; the change is just stored on our local repository. If necessary, we can push these changes to a remote repository.

- **Security**

Git is secure. It uses the **SHA1 (Secure Hash Function)** to name and identify objects within its repository. Files and commits are checked and retrieved by its checksum at the time of checkout. It stores its history in such a way that the ID of particular commits depends upon the complete development history leading up to that commit. Once it is published, one cannot make changes to its old version.

- **Speed**

Git is very **fast**, so it can complete all the tasks in a while. Most of the git operations are done on the local repository, so it provides a **huge speed**. Also, a centralized version control system continually communicates with a server somewhere

Search Engine Optimization (SEO)

SEO stands for **S**earch **E**ngine **O**ptimization. SEO is all about optimizing a website for search engines. SEO is a technique for:

- designing and developing a website to rank well in search engine results.
- improving the volume and quality of traffic to a website from search engines.
- marketing by understanding how search algorithms work, and what human visitors might search.

SEO is a subset of search engine marketing. SEO is also referred as SEO copyrighting, because most of the techniques that are used to promote sites in search engines, deal with text.

If you plan to do some basic SEO, it is essential that you understand how search engines work.

How Search Engine Works?

Search engines perform several activities in order to deliver search results.

- **Crawling** - Process of fetching all the web pages linked to a website. This task is performed by a software, called a **crawler** or a **spider** (or Googlebot, in case of Google).
- **Indexing** - Process of creating index for all the fetched web pages and keeping them into a giant database from where it can later be retrieved. Essentially, the process of indexing is identifying the words and expressions that best describe the page and assigning the page to particular keywords.
- **Processing** - When a search request comes, the search engine processes it, i.e. it compares the search string in the search request with the indexed pages in the database.
- **Calculating Relevancy** - It is likely that more than one page contains the search string, so the search engine starts calculating the relevancy of each of the pages in its index to the search string.
- **Retrieving Results** - The last step in search engine activities is retrieving the best matched results. Basically, it is nothing more than simply displaying them in the browser.

Sass

Sass stands for **S**yntactically **A**wesome **S**tyle**s**heet. It is a CSS pre-processor and it reduces repetition of CSS and therefore saves time.

Why Use Sass?

Stylesheets are getting larger, more complex, and harder to maintain. This is where a CSS pre-processor can help.

Sass lets you use features that do not exist in CSS, like variables, nested rules, mixins, imports, inheritance, built-in functions, and other stuff.

How Does Sass Work?

A browser does not understand Sass code. Therefore, you will need a Sass pre-processor to convert Sass code into standard CSS.

This process is called transpiling. So, you need to give a transpiler (some kind of program) some Sass code and then get some CSS code back.

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

Why Use Bootstrap?

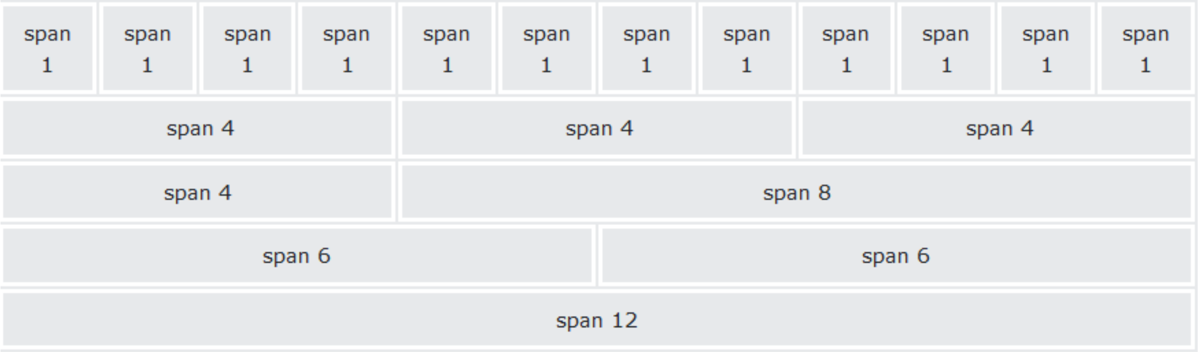
Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap 5 is compatible with all modern browsers (Chrome, Firefox, Edge, Safari, and Opera). **Note** that if you need support for IE11 and down, you must use either BS4 or BS3.

Bootstrap 5 Grid System

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:



<table>
<tr><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td></tr>
<tr><td colspan="4">span 4</td><td colspan="4">span 4</td><td colspan="4">span 4</td></tr>
<tr><td colspan="4">span 4</td><td colspan="8">span 8</td></tr>
<tr><td colspan="6">span 6</td><td colspan="6">span 6</td></tr>
<tr><td colspan="12">span 12</td></tr>
</table>

The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

Grid Classes

The Bootstrap 5 grid system has six classes:

- `.col-` (extra small devices - screen width less than 576px)
- `.col-sm-` (small devices - screen width equal to or greater than 576px)
- `.col-md-` (medium devices - screen width equal to or greater than 768px)
- `.col-lg-` (large devices - screen width equal to or greater than 992px)
- `.col-xl-` (xlarge devices - screen width equal to or greater than 1200px)
- `.col-xxl-` (xxlarge devices - screen width equal to or greater than 1400px)

The classes above can be combined to create more dynamic and flexible layouts.

VueJS

VueJS is an open source progressive JavaScript framework used to develop interactive web interfaces. It is one of the famous frameworks used to simplify web development. VueJS focusses on the view layer. It can be easily integrated into big projects for front-end development without any issues.

Features

Following are the features available with VueJS.

Virtual DOM

VueJS makes the use of virtual DOM, which is also used by other frameworks such as React, Ember, etc. The changes are not made to the DOM, instead a replica of the DOM is created which is present in the form of JavaScript data structures. Whenever any changes are to be made, they are made to the JavaScript data structures and the latter is compared with the original data structure. The final changes are then updated to the real DOM, which the user will see changing. This is good in terms of optimization, it is less expensive and the changes can be made at a faster rate.

Data Binding

The data binding feature helps manipulate or assign values to HTML attributes, change the style, assign classes with the help of binding directive called **v-bind** available with VueJS.

Components

Components are one of the important features of VueJS that helps create custom elements, which can be reused in HTML.

Event Handling

v-on is the attribute added to the DOM elements to listen to the events in VueJS.

Animation/Transition

VueJS provides various ways to apply transition to HTML elements when they are added/updated or removed from the DOM. VueJS has a built-in transition component that needs to be wrapped around the element for transition effect. We can easily add third party animation libraries and also add more interactivity to the interface.

Computed Properties

This is one of the important features of VueJS. It helps to listen to the changes made to the UI elements and performs the necessary calculations. There is no need of additional coding for this.

Templates

VueJS provides HTML-based templates that bind the DOM with the Vue instance data. Vue compiles the templates into virtual DOM Render functions. We can make use of the template of the render functions and to do so we have to replace the template with the render function.

Directives

VueJS has built-in directives such as v-if, v-else, v-show, v-on, v-bind, and v-model, which are used to perform various actions on the frontend.

Watchers

Watchers are applied to data that changes. For example, form input elements. Here, we don't have to add any additional events. Watcher takes care of handling any data changes making the code simple and fast.

Routing

Navigation between pages is performed with the help of vue-router.