

## CALIBRAINT TECHNOLOGIES

### MILESTONE 6 - 07.09.2024

#### CDN

Bootstrap, Tailwind

#### CDN

- A CDN, or Content Delivery Network, is a network of servers that delivers web content to users based on their location. CDNs are made up of servers that are strategically placed around the world in data centers. When a user visits a website, the content is stored on a CDN server that's closer to the user, which speeds up the loading time.
- Why is CDN important ?
  - The primary purpose of a content delivery network (CDN) is to reduce latency, or reduce the delay in communication created by a network's design. Because of the global and complex nature of the internet, communication traffic between websites (servers) and their users (clients) has to move over large physical distances. The communication is also two-way, with requests going from the client to the server and responses coming back.
  - A CDN improves efficiency by introducing intermediary servers between the client and the website server. These CDN servers manage some of the client-server communications. They decrease web traffic to the web server, reduce bandwidth consumption, and improve the user experience of your applications.
- Here are some benefits of using a CDN:
  - Faster load times
    - CDNs can help reduce load times and improve performance, which can boost user satisfaction.
  - Better SEO rankings
    - CDNs can help improve SEO rankings by reducing bandwidth and improving load times.
  - Reduced buffering
    - CDNs can help reduce buffering and improve the quality of images and sound.
  - Reduced latency
    - CDNs can help reduce latency by using servers that are closer to the user.
  - Load balancing
    - CDNs can help load balance traffic to prevent servers and routers from becoming overwhelmed.
    - CDNs are used by many industries, including media and entertainment, and online gaming.

- What internet content can a CDN deliver?
  - A content delivery network (CDN) can deliver two types of content: static content and dynamic content.
  - Static content
    - Static content is website data that does not change from user to user. Website header images, logos, and font styles remain the same across all users, and the business does not change them frequently. Static data does not need to be modified, processed, or generated and is ideal for storage on a CDN.
  - Dynamic content
    - Dynamic content such as social media news feeds, weather reports, login status, and chat messages vary among website users. This data changes based on the user's location, login time, or user preferences, and the website must generate the data for every user and every user interaction.
- How does a CDN work?
- Content delivery networks (CDNs) work by establishing a point of presence (POP) or a group of CDN edge servers at multiple geographical locations. This geographically distributed network works on the principles of caching, dynamic acceleration, and edge logic computations.
  - Caching
    - Caching is the process of storing multiple copies of the same data for faster data access. In computing, the principle of caching applies to all types of memory and storage management. In CDN technology, the term refers to the process of storing static website content on multiple servers in the network. Caching in CDN works as follows:
      1. A geographically remote website visitor makes the first request for static web content from your site.
      2. The request reaches your web application server or origin server. The origin server sends the response to the remote visitor. At the same time, it also sends a copy of the response to the CDN POP geographically closest to that visitor.
      3. The CDN POP server stores the copy as a cached file.
      4. The next time this visitor, or any other visitor in that location, makes the same request, the caching server, not the origin server, sends the response.
  - Dynamic acceleration
    - Dynamic acceleration is the reduction in server response time for dynamic web content requests because of an intermediary CDN server between the web applications and the client. Caching doesn't work well with dynamic web content because the content can change with every user request. CDN servers have to reconnect with the origin server for every dynamic request, but they accelerate the process by optimizing the connection between themselves and the origin servers.
    - If the client sends a dynamic request directly to the web server over the internet, the request might get lost or delayed due to network latency. Time might also be spent opening and closing the connection for

security verification. On the other hand, if the nearby CDN server forwards the request to the origin server, they would already have an ongoing, trusted connection established. For example, the following features could further optimize the connection between them:

- Intelligent routing algorithms
- Geographic proximity to the origin
- The ability to process the client request, which reduces its size

- Edge logic computations
  - You can program the CDN edge server to perform logical computations that simplify communication between the client and server. For example, this server can do the following:
    - Inspect user requests and modify caching behavior.
    - Validate and handle incorrect user requests.
    - Modify or optimize content before responding.
- Distribution of application logic between the web servers and the network edge helps developers offload origin servers' compute requirements and improve website performance.
- Bootstrap CDN is the distribution of the most popular CSS framework in the world using a content delivery network. It is a geographically distributed group of servers that work together to provide fast delivery of Internet content.
- BootstrapCDN is a public content delivery network. It enables users to load CSS, JavaScript, and images remotely from its servers. Used by more than 7.9 million websites worldwide (including 30% of the top-10k websites), BootstrapCDN serves more than 70 billion requests a month
- **How do Core Bootstrap 4.0.0 CSS and js files use bootstrap cdn?**
  - Using Bootstrap CDN over manual installation is a little bit advantageous; it keeps your code short and clean, quick response of your site on access, short loading time and reduced bounce overs. Anyhow, there is a little problem- downtime or slow server connection may result in some missing features. However, if you use CDN, two things you need to take care of:
    - Position your Bootstrap CDN in proper place. Place it into the header of your site , just above the custom css stylesheet that you created to add custom classes to your site. This is done to override any default bootstrap classes that you wish to modify, later on to your project.
    - Now when it comes about the js files, you should place it in the footer section, just above the closing body tag. Additional steps should be taken care of; Bootstrap4 requires an additional popper.js and the usual jquery to run all the features. Maintain the order properly in your footer. The order should be-
      - on top- jquery.min.js
      - second- popper.js
      - third and last position- bootstrap.min.js

- This is how you should be able to add Bootstrap4 CDN into your project.

## **Bootstrap, Tailwind**

- **BOOTSTRAP**

- Bootstrap has a system with 12 columns that changes size based on how big the screen is, making it great for viewing on phones, tablets, and computers.
- It comes with lots of ready-to-use pieces like menus, buttons, and alerts. You can change how these look with some coding skills.
- Bootstrap is made to be mobile-friendly from the start and works well in different web browsers. This means websites built with Bootstrap look good on any device.
- You can easily change the look of your site with Bootstrap's theme system, like adjusting colors and more.
- There's a lot of help and examples online to get you started quickly.
- Big names like NASA and Spotify use Bootstrap for their websites.

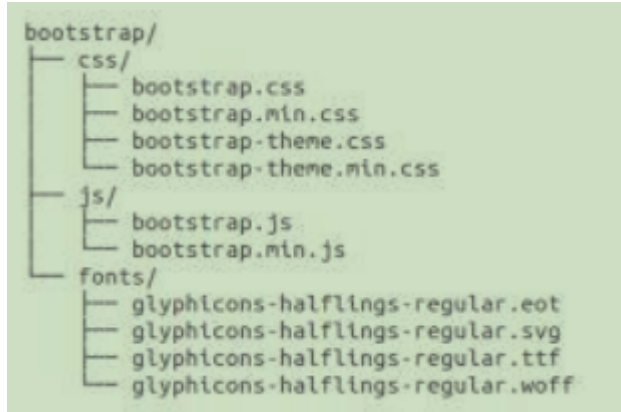
- **Why use Bootstrap ?**

- Mobile first approach: Since Bootstrap 3, the framework consists of Mobile first styles throughout the entire library instead of in separate files.
- Browser Support: It is supported by all popular browsers.
- Easy to get started: With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
- Responsive design: Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about responsive design in the chapter Bootstrap Responsive Design.
- Provides a clean and uniform solution for building an interface for developers.
- It contains beautiful and functional built-in components which are easy to customize.
- It also provides web based customization.
- And best of all it is an open source.

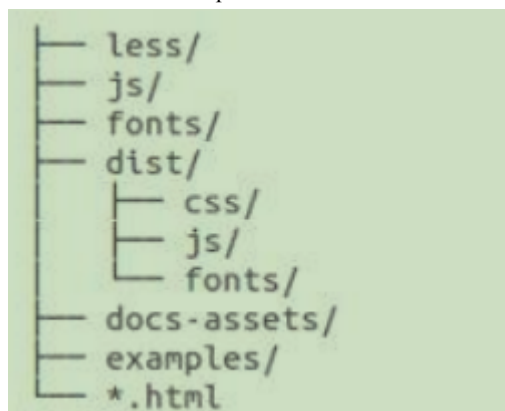
- **What Bootstrap Package Includes?**

- Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, background. This is covered in detail in the section Bootstrap Basic Structure
- CSS: Bootstrap comes with feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section Bootstrap with CSS.
- Components: Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, popovers, and much more. This is covered in detail in the section Layout Components.

- JavaScript Plugins: Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section Bootstrap Plugins.
- Customize: You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.
- File Structure
- Precompiled bootstrap
  - Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure:



- Bootstrap source code
- If you downloaded the Bootstrap source code then the file structure would be as follows:



- REFERENCE LINK - <https://wiki.lib.sun.ac.za/images/0/07/Bootstrap-tutorial.pdf>

- **TAILWIND**

- Tailwind CSS is a tool for building websites that was created by Adam Wathan and Steve Schoger in 2017. It's different because it gives you a lot of small tools called utility classes that let you design web pages in detail. Think of it like having a big box of LEGO bricks where each brick can be used in many ways to make exactly what you want. Here are some important things to know about Tailwind CSS:
- Utility-First Approach: Tailwind CSS is all about giving you these utility classes for things like colors, spacing, and how text looks. This means you can mix and match them to make your web page look just right.

- **Highly Customizable:** You can change how Tailwind CSS works to fit your needs. This includes changing the color scheme, the fonts, and more right in the settings.
- **Purges Unused CSS:** Tailwind CSS helps keep your website fast by getting rid of styles you're not using. This means your website loads quicker because it's not weighed down by unnecessary stuff.
- **Works with Other Tools:** Whether you're using React, Vue, or another web development tool, Tailwind CSS fits right in. It's easy to add to whatever project you're working on.
- **Mobile-First Styling:** Tailwind CSS is made to make sure your website looks good on phones first, then adjusts for bigger screens like tablets and computers.
- In short, Tailwind CSS is all about letting you build your website your way, with lots of options for customization. It's great for making sure your site is fast and looks good on all devices, thanks to its focus on utility classes and mobile-first design.
- **How to install Tailwind ?**
  - Use the CDN - <https://unpkg.com/tailwindcss/dist/tailwind.min.css>
  - Install Tailwind via NPM
    - `npm install --save-dev tailwindcss` (or)
    - `Yarn add -D tailwindcss`
- **REFERENCE LINK -**  
<https://www.drupalcampatlanta.com/sites/default/files/slides/taking-flight-tailwind-css.pdf>