Final Documentation

Bytewise Fellowship

FrontEnd Web Development

INTERNEE'S NAME:

Adeel Amin

[Pick the date]

Task 1:

Summary :

Today I learned the most important fundamentals of front-end-developers. That is HTML,CSS,and javaScript.

HTML: Stands for hyper text markup language and it is used to structure web pages.

CSS: stands for cascading styling sheets and is used to make them beautiful.

JavaScript:it is used to program them.

If we learn all these fundamentals then we should learn frameworks of front-end-developer.Frameworks are important in front-end. These frameworks are popular React,Angular, and Vue. Framework force our application into structure.

Also I learned how web works/ It works through combination of technologies, including:

Hypertext Transfer Protocol (HTTP) - a protocol that allows web browsers and servers to communicate with each other.

Uniform Resource Locators (URLs) - addresses that identify the location of web pages and other resources on the internet.

Hyperlinks - clickable links that allow users to navigate between web pages and other resources.

Web browsers - software applications that allow users to access and view web pages.

When a user enters a URL into a web browser, the browser sends an HTTP request to the web server hosting the requested web page. The server then responds with the requested web page, which the browser renders and displays to the user.

The web also includes other technologies such as HTML, CSS, and JavaScript, which are used to create and design web pages, and databases and server-side scripting languages, which are used to generate dynamic content.

Task 2:

Summary :

Today I learned about HTML.

What is HTML?

HTML stands for Hyper Text Markup LaNguage.It is not a programming language.It is used for creating web pages/Document and building blocks of the web. We use elements in HTML. An HTML element is defined by a start tag, some content, and an end tag.

How can we create an HTML file?

Files must end with >=.html extension. It is run in a web browser.Index.html is a root OR home of a web page. We do not need a server.

HTML Tags Syntax:Element name surrounded by angle brackets. Elements normally come in pairs(start tag and end tag). i.e <tagname>Content goes here...</tagname>.

HTML Page Structure:

```
<html>
  <head>
        <title>Page title</title>
  </head>
  <body>
        <h1>this is a heading</h2>
  </body>
</html>
```

Inline vs Block level elements:

**Inline Elements: these elements do no start on a new line and take a necessary width.(
,<a>,<img>,)**

Block Level Elements: these elements start on a new line and take full width available.(<p>,<h6>, ,<forms>)

Html Attribute: All HTML elements can have attributes.Attributes provide additional information about elements.Attributes are always specified in the start tag.Attributes usually come in name/value pairs like: name="value"

Html Semantic tags: A semantic element clearly describes its meaning to both the browser and the developer.

Task 3:

Summary :

Today I learned about semantic element in html.

Semantic Elements: A semantic element clearly describes its meaning to both the browser and the developer. In HTML<There are many semantic elements:
<article>,<aside>,<details>,<figcaption>,<figure>,<footer>,<header>,<main>,<mark>,<nav>,<section>, <summary>,<time>

<main>: For the main content of a webpage, unique to that page.

<section>:Defines a certain section of a webpage.

<article>:Defibes a bit of content which makes up an article.

<aside>: Defines some content related to something else.

<header>: For the header of a website.

<footer>: For the footer of a website

Task 4:

Summary :

Today I learned about media elements in HTML.

Media Elements: on the web is sound,music, videos, movies,and animations.There are some media elements (<video>,<audio>).

<video>: this element is used to show a video on a web page. The easiest way to play videos in HTML is to use YOUTUBE.(Playing a YT video in HMTL)

1.GO to the YT video,which you have to use

2.Right click on this video nd select "embed code"(This video 's embed code is copied)

3.Paste this code on your web page.

OR

1.Upload the video to YouTube

2.Take a note of the video id

3.Define an <iframe> element in your web page

4.Let the src attribute point to the video URL

5.Use the width and height attributes to specify the dimension of the player

6.Add any other parameters to the URL

<audio>element is used to play a sound on a web page.

Task 5:

Summary :

Today I learned about CSS and fundamental of css,

What is CSS?

CSS stands for Casading styling sheet. It is not a programming language.

It is used to style HTML Doc. CSS describes how HTML elements should be displayed .

CSS saves a lot of work. It can control the layout of multiple web pages all at once

External stylesheets are stored in CSS files.

There are three ways to insert CSS in HTML:

1.External CSS: With an external style sheet, you can change the look of an entire website by changing just one file.Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

2.Internal CSS: An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

3. Inline CSS:An inline style may be used to apply a unique style for a single element.To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS properties.

The basic fundamentals of CSS include the following:

Selectors: CSS selectors are used to identify the HTML elements to which the styles should be applied. Selectors can be based on element types, class names, IDs, or other attributes.

Properties: CSS properties are used to define the visual style of HTML elements. Examples of CSS properties include color, font-size, margin, padding, and background-color.

Values: CSS values are used to set the values of CSS properties. For example, the value for the color property can be a named color, an RGB value, or a hexadecimal value.

Cascading: CSS stands for Cascading Style Sheets, which means that styles can be inherited and overridden in a cascading manner. This allows for consistent styling throughout a website while also allowing for customization on specific pages or elements.

Box model: The CSS box model is used to define the layout of HTML elements. It consists of the content area, padding, border, and margin.

Units of measurement: CSS supports various units of measurement, such as pixels, ems, rems, and percentages, which are used to define the size and position of elements.

Media queries: Media queries allow developers to create responsive websites by applying different styles based on the size of the device or screen. This enables the website to look good on various devices, such as desktops, tablets, and smartphones.

Task 6:

Summary :

Today I learned about flexbox in css. What is flexbox?

Flexbox is a CSS layout module that provides a flexible way to lay out and align content within a container. It allows for dynamic positioning and sizing of elements based on available space, making it a powerful tool for creating responsive designs.

To use Flexbox, you first define a container element and set its display property to flex. This tells the browser that the container should use Flexbox layout. You can then use various Flexbox properties to control the behavior of the container and its child elements.

Here are some common Flexbox properties:

flex-direction: Defines the main axis of the container and the direction in which the child elements will be laid out. Values can be row (default), row-reverse, column, or column-reverse.

justify-content: Defines how the child elements are aligned along the main axis of the container. Values can be flex-start (default), flex-end, center, space-between, space-around, or space-evenly.

align-items: Defines how the child elements are aligned along the cross axis of the container. Values can be stretch (default), flex-start, flex-end, center, or baseline.

flex-wrap: Defines whether the child elements should wrap to a new line if there is not enough space on the current line. Values can be nowrap (default), wrap, or wrap-reverse.

align-content: Defines how multiple lines of child elements are aligned along the cross axis of the container. This property only applies if flex-wrap is set to wrap or wrap-reverse. Values can be flex-start, flex-end, center, space-between, space-around, or stretch (default).

In addition to these properties, there are many other Flexbox properties that can be used to achieve specific layout goals. By mastering Flexbox, you can create complex and responsive layouts with ease.

Task 9:

Summary :

I made a blog-website. In this site, I faced to many errors but also debuged them and learned many new things.

Here My project Blogsite live link:

https://github.com/adeelamin872/Blog-website.git

in which I am using HTML CSS AND JAVASCRIPT.

Task 10:

Summary:

Today I learned about Git and GitHub .

What is Git?

Git is a distributed version control system that is used to track changes in source code during software development. It allows developers to work collaboratively on the same codebase and keep track of changes over time. Git was created by Linus Torvalds in 2005 and has since become one of the most popular version control systems in use today.

Here are some commonly used Git commands:

git init: Initializes a new Git repository in the current directory.

git add [file]: Adds a file or files to the staging area for the next commit.

git commit -m "commit message": Commits changes to the local repository with a brief message describing the changes.

git status: Shows the current status of the repository, including any modified files and files that have been added or deleted.

git push: Pushes changes to a remote repository, such as a repository hosted on GitHub.

git pull: Pulls changes from a remote repository and merges them with the local repository.

git branch: Lists all the branches in the repository.

git checkout [branch]: Switches to the specified branch.

git merge [branch]: Merges the specified branch into the current branch.

git clone [repository URL]: Creates a local copy of a remote repository.

What is GitHub?

GitHub, on the other hand, is a web-based hosting service for Git repositories. It provides a platform for developers to store, manage, and collaborate on Git repositories, as well as a variety of tools for managing code reviews, issue tracking, and continuous integration and deployment.

GitHub allows developers to easily share their code with others and contribute to open-source projects. It provides a platform for developers to work on code collaboratively, manage project workflows, and track issues and bugs. GitHub also provides a range of tools for project management, including wikis, project boards, and team management features.

GitHub has become an important tool for software developers and open-source communities, allowing them to share and collaborate on code in an efficient and effective manner. Many popular open-source projects are hosted on GitHub, and it is widely used by developers and organizations around the world.

Task 11:

Summary:

Today I know about this 200+ design resources for developers. We can use these resources what we need in our project. And also learned about framework.

What is framework and why use framework?

Framework:

In general, a framework is a structure or set of rules, assumptions, or procedures that provide a standardized way to approach a particular problem or task. In computer science, a framework refers to a set of pre-built components, modules, and libraries that are designed to simplify and streamline the development of software applications.

Frameworks provide a foundation upon which developers can build their applications, without having to write everything from scratch. They can include everything from basic utility libraries to full-fledged application architectures, and they often come with pre-built functionality, such as authentication, data storage, and user interface components.

Frameworks are designed to be flexible, extensible, and reusable, so developers can customize them to suit their specific needs. Some popular examples of software frameworks include Ruby on Rails, Laravel, Angular, React, and Django.

Why use framework:

There are several reasons why developers use frameworks when building software applications:

Speed and efficiency: Frameworks provide pre-built components and libraries that can help developers to build applications more quickly and efficiently than if they had to write everything from scratch.

Standardization: Frameworks often come with standardized coding conventions and best practices that can help ensure that the application is consistent, maintainable, and easy to read.

Security: Many frameworks provide built-in security features and modules that can help protect the application from common security vulnerabilities.

Scalability: Frameworks are designed to be scalable, so as the application grows and evolves over time, the framework can adapt and support new features and functionality.

Community and support: Many frameworks have active communities of developers who contribute to the development and maintenance of the framework, which can provide a valuable source of support, documentation, and resources.

Overall, using a framework can help developers save time, increase productivity, and build more robust, secure, and scalable applications.

What is Bootstrap?

Bootstrap is a free and open-source CSS framework that was originally developed by Twitter. It provides a set of pre-built HTML, CSS, and JavaScript components that can be used to quickly and easily build responsive and mobile-first web applications.

Bootstrap includes a wide range of pre-built UI components such as navigation menus, forms, buttons, typography, tables, modals, and more. It also includes a responsive grid system, which allows developers to create flexible and dynamic layouts that adapt to different screen sizes and devices.

One of the main benefits of using Bootstrap is that it simplifies the process of creating a visually appealing and functional website or web application. Bootstrap provides a standardized and consistent set of design patterns, which means that developers can focus on the functionality of their application, rather than worrying about the design and layout.

Bootstrap also provides a large community of developers and designers who contribute to its development and provide support and resources to other users. This makes it a popular choice for developers who want to build high-quality web applications quickly and easily.

Task 15:

Summary of task15:

Today I'm learing about JS.

Learning JavaScript (JS) requires a solid understanding of its core concepts and features. Here's an overview of what you need to learn to become proficient in JavaScript

1. Basic Syntax: Start by understanding the basic syntax of JavaScript, including variables, data types, operators, and control structures like conditionals and loops.

2. Functions: Learn how to define and use functions in JavaScript. Understand the concept of parameters, return values, and the scope of variables within functions.

3. Objects and Prototypes: JavaScript is an object-oriented programming language, so it's crucial to understand objects and prototypes. Learn how to create objects, define properties and methods, and work with prototypes for inheritance.

4. Arrays: Arrays are fundamental data structures in JavaScript. Learn how to create, manipulate, and iterate over arrays. Familiarize yourself with array methods like push, pop, splice, and map.

5. DOM Manipulation: The Document Object Model (DOM) allows you to interact with web page elements using JavaScript. Learn how to select and modify HTML elements, handle events, and dynamically change the content and style of a webpage.

6. Asynchronous Programming: JavaScript is single-threaded, but it supports asynchronous programming through concepts like callbacks, promises, and async/await. Learn how to handle asynchronous operations such as fetching data from APIs or making network requests.

7. Error Handling: Understand how to handle errors and exceptions in JavaScript. Learn about try-catch blocks and how to throw custom errors.

8. Modules: JavaScript has a built-in module system that allows you to organize your code into reusable and separate files. Learn how to import and export modules using the ES6 module syntax.

9. Browser APIs: JavaScript provides access to various browser APIs, such as the Web Storage API, Fetch API, and Geolocation API. Explore these APIs and learn how to use them to enhance your web applications.

10. Popular Libraries and Frameworks: JavaScript has a vast ecosystem of libraries and frameworks that can simplify web development tasks. Familiarize yourself with popular libraries like React, Vue.js, or Angular, depending on your specific interests.

Task 12 and 13:

In this task I am performing bootstrap bootcamp project :

Githup link page:

https://github.com/adeelamin872/BootCamp.git

netlify link:

https://app.netlify.com/sites/moonlit-conkies-cdaafa/deploys/64a1620fafc4984b60351ae5

Task 14:

In this task I am performing bootstrap CSS tailwind project :

Tailwind project link:

https://github.com/adeelamin872/Tailwind-project.git

netlify link:

https://jocular-shortbread-37f119.netlify.app/

Task 18:

In this task I ma performing react base project

React page link:

https://github.com/adeelamin872/React-App.git

live link:

https://adeelamin872.github.io/React-App/

netlify link:

https://master--splendid-otter-9a8897.netlify.app/

This is all my work and projects which I am doing during the fellowship

THANKS :

BYTEWISE FELLOWSHIP