**CHAPTER ONE**

1. **PURPOSE OF TRAINING**

The student industrial work experience scheme (SIWES) popularly called Industrial Training (IT) by Nigerian Students is a yearly program design by the institution in collaboration with the industries to give students the opportunity to gain practical experience in their various field of study or area of specialization. It is an effort to bridge the existing gap between classroom theories and practical’s in engineering, management and other professional programs in the Nigerian tertiary institutions.

Training is the key factor in changing expertise of a workforce. The world is passing through one of the worst economic crises in recent time. Both the developed and developing economics are experiencing serious economic downturns.

It is through this industrial training that the educational systems aims at helping students acquire appropriate skills, abilities and competences, both mental and physical as well as equip the individual to live in society. The focus of the Industrial Training Fund (ITF) is for the industries of our countries to succeed in the face of the current economic meltdown.

No society can achieve meaningful progress without encouraging its youth to acquire necessary practical skills. Such skills enable them to harness available resources to meet the needs of the society.

* 1. **BACKGROUND OF SIWES**

Before the establishment of the scheme, there was a growing concern that graduates of institutions of higher learning lacked adequate practical knowledge and that the theoretical education in higher institutions was not responsive to the needs of the employers of labor.

As a result of this, the Industrial Training Fund (ITF) was initiated, designed and introduced SIWES Scheme in 1973 to acquaint students with the skills of handling equipment and machinery.

The students Industrial Work Experience Scheme (SIWES) is a skill training program designed to prepare and expose students of universities, polytechnics, colleges of education etc. for the industrial work experience they are likely to meet after graduation.

The Industrial Training Fund (ITF) solely funded the scheme during its formative years. However, due to financial constraints, the fund withdrew from the scheme in 1978. The federal government noting the significance of the skills training handed the management of the scheme to its National Universities Commission (NUC) and the National Board for Technical Education (NBTE) in 1979. In November, 1984 management and implementation of the scheme was again reverted to ITF with the funding to solely borne by the federal government.

* 1. **AIMS AND OBJECTIVES OF SIWES**

The specific objective of SIWES were summarized by the federal government as follows is meant to ensure.

* Provide adequate skill acquisition for tertiary institution students
* It is designed to give these students an idea of real work situations and experience they are likely to encounter after graduation.
* They also include providing a structural attachment program with emphasis applications, management and hands-on experience for students to apply knowledge acquired.
* Advanced countries, with over 100 years of sustained industrial development and requisite technical and human infrastructure, have been able to adequately implement training for their students.
* Moreover, it helps them to gain interpersonal and entrepreneurship skills and also instill in them the right kind of work altitudes and professionalism through interactions with peoples in the organization and observation of their future role in the industry.
  1. **BENEFITS OF INDUSTRIAL TRAINING**
* Provide avenue for students in institution of higher learning to acquire industrial skills and experience in their course of study.
* Provide students with an opportunity to apply their knowledge in real work situation thereby bridging the gap between theory and practical
* Makes the transition from school to the world of work easier and enhance students contact for later job placement.
* Expose students to work methods and techniques in handling equipment and machinery that may not be available in their institutions;
* Enlist and strengthen employers’ involvement in the entire educational process and prepare students for employment after graduation.

The major benefits accruing to students who participate conscientiously in industrial are the skills and competencies they acquire. These relevant production skills (RPSs) remain a part of the recipient of industrial training as lifelong assets which cannot be taken from them. This is because the knowledge and skills acquired through training are internalized and become relevant when required to perform jobs or functions. Several other benefits can accrue to students who participate in industrial training.

* 1. **COMPANY VISION**

To become a center of research and quality ICT service provider where future generation embrace technologies and use it’s possibilities to create a friendly user interface between the technology and common human.

* 1. **COMPANY AREA OF SPECIALIZATION**

With a team of software developers and IT personnel with a sound experience in software development. The company offer the various services such as web management, web development, software development, web design, internet and E-mail services.

**CHAPTER TWO**

**1.0 VAM TECHNOLOGY VENTURES HISTORY**

VAM TECHNOLOGY VENTURES is an ICT and Computer Cyber Service provider company, incorporated with an RC number of 2898768.

The services being rendered by Vam Technology Ventures includes:

* Graphic Design
* Internet Services
* Online Registrations and Recruitment
* Online Research
* Uploading and downloading
* Website Design
* Website Development
* Website Management
* Printing
* Photocopy
* Scanning
* Laminating
* Binding
* Computer Training
* Software Installation
  1. **VISION STATEMENT**

To become a Centre of research and quality ICT Service provider where future generation embrace technologies and use it’s possibilities to create a friendly user interface between the technology and common human.

* 1. **MISSION STATEMENT**
* To provide a complete solution of ICT and computer services to the general public.
* To train more youth on technology to help alleviate the rate of unemployment in the country.
  1. **COMPANY RULES AND REGULATIONS**

VAM TECHNOLOGY VENTURES is all about a safe and pleasant working atmosphere for both workers and clients. With workers & Team committed to co-operate and commit to the appropriate standards of behavior and interaction with clients.

The Following is a list of behavior that the company considers unacceptable. Any employee found engaging in these behaviors will be subject to disciplinary actions including reprimand, warning, layoff or dismissal

* Failure to be at the work place, ready to work at the regular starting time.
* Damaging, destroying or stealing properties that belongs to a fellow employee or the company.
* Fighting or engaging in horseplay or disorderly conduct.
* Coming to work under the influence of alcohol or any hard drug.
* Falsifying any result.
* Intentionally giving any false or misleading information to obtain a leave or an excuse.
* Using threatening or abusive language towards a fellow employee.
* Being tardy or taking unexcused absences from work.
* Ignoring work duties

**THESE SHOULD BE NOTED ALSO**

* While at the office, do not entertain visitors coming to the office frequently, in order to increase your productivity.
* Use of phone is allowed online if you are free but not while on any services or attending to a customer.
* Maintain a high level of cleanliness in communal spaces.
* Complaints regarding office inconveniences should be discussed with the CEO for a better improvement and it's meant to be confidential to avoid creating bad impact concerning the company to the public.
* Dress properly to work and while in office (Business or Casual wears).

**VAM TECHNOLOGY VENTURES** was caved out from the name of the team **VALENTINE ASIERIKA MMADUABUCHI** with the sole proprietor **EZENWAKA INNOCENT MMADUABUCHI** who hails from Ifite Nanka in Orumba North Local Government Area, Anambra State in Nigeria.

**CHAPTER THREE**

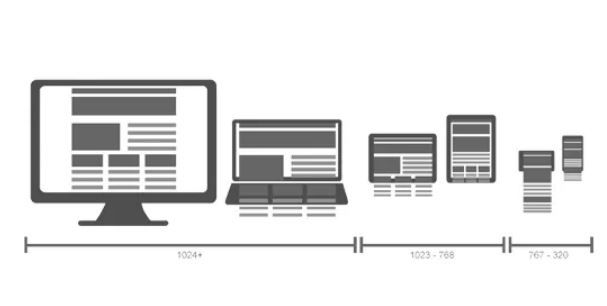
**3.1 INTRODUCTION**

Web development is the work involved in developing a website for the internet or the intranet (system in which multiple PCs are networked to be connected to each other.) web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. Among web professionals web development usually refers to the main non design aspect of building web sites; writing mark-up languages and coding. Web development may be a collaborative effort between department rather than the domain of a designated department.

There are three kinds of web development specialization: front-end development, back-end development and full-stack development.

* Front-end development: This entails the development of the graphical user interface of the web site, through the use of HTML, CSS and JAVA SCRIPT, so that users can use and interact with that website.
* Back-end development: This is everything that is not visible on the website, and what makes the web site or web application to work properly. The back-end is extremely important because it defines the correct operation of the entire technical background of a given application or website.
* Full-stack development: It refers to the development of both the front end (client side) and back end (server side) portion of web application.

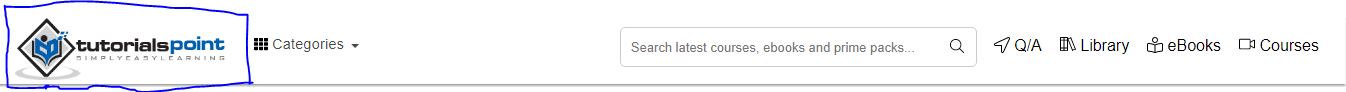
**3.2 DEFINITION OF TERMS**

* WEB SITE: A web site is a set of related web pages having content such as images, text, videos, audios etc. web sites are hosted on at least one web server, accessible via a network such as internet or private LAN through an internet address known as URL(Universal Resource Locator). A public accessible web sites constitute of the World Wide Web (WWW)
* RESPONSIVE DESIGN

*Images source* [*https://cdn.mos.cms.futurecdn.net/efd6f83197d2d2a97f029b16f7475c17-970-80.png.webp*](https://cdn.mos.cms.futurecdn.net/efd6f83197d2d2a97f029b16f7475c17-970-80.png.webp)

A responsive design is one that adapts to the user’s device and, in an ideal world, the user’s context so that it displays the required context in the most appropriate and accessible manner. Regardless of what kind of web connected devise is used to access it. This means that a web page will re paginate itself as the screen size increase or decrease, displaying in multiple column when we view on a desktop computer of a laptop, but only a single column when viewed with a mobile phone.

* **WEB PAGES:** web pages is a document available on the world wide web (www) web pages are stored on the web server and can be viewed using a web browser. A web page can contain huge information including text graphics, audio, video, and hyper link. These hyperlinks are the links to other web pages or a different web site entirely.
* **URL:** This stands for Universal resource locator which is just a given address for a unique resource on the web. In theory it points to a unique resource. This resource can be an HTML or a CSS document or images.
* **404:** An error that a user sees when there try to reach a nonexistent page on the web site. Usually this can be caused by the user typing the wrong URL or attempts to access a page that was already deleted from the site. An effective **404** should tell the user the page does not exist and what the user can do.
* **DOMAIN**: The name of the website that people type into a browser to visit it. For example, my portfolio website domain is franklin-raph.github.io
* **FAVICON: A small icon image, often a company logo, that displays on the title bar or tab of a browser.**



***Image source:****https://www.tutorialspoint .com*

* **HOSTING: The web servers houses and maintain the web sites files, this web servers are computer running web server software connected to the internet which allows visitors to access different websites through the internet connected web browser.**
* **LANDING PAGE: This is the page a user sees after a search result, or a marketing email, social media post or online advertisement.**
* **NAVIGATION: This usually appears on the side of the web which directs the user through the site. It can also include links in the footer, at the bottom page of the site.**

**3.3 WEB DEVELOPMENT TOOLS**

This are used to write a web code, there are different types and different uses, each having a unique attribute which makes web development easy, some of this tools are used with specific operating system like the mac, windows, Linux etc.

Some tools include;

* SKECTH:  Some of the features including the ability to add simple animations and creation of clickable prototypes. It also lets the user create a master symbol, that can then be used anywhere in the document, support only MAC.
* Invision cloud
* Sublime text
* Foundation
* Chrome devtools
* Visual studio code
* Bracket text editor
* Git
* Github
* Atom text editor
* Notepad ++
* Vim
* Emacs
* Adobe Dream Weaver e.t.c.

**CHAPTER FOUR**

**4.1 WHAT IS HTML**

HTML stands for **Hyper Text Markup Language** which is used to design the front end portion of the web page using mark language. It acts as the skeleton of the web site since it used to make the structure of the web site, tell the web browser how to display the content.

As the name suggests, HTML is a markup language which means you use HTML to simply “mark-up” a text document with tags that tell a Web Browser how to display the content of the web page.

Originally HTML was developed with the intent of defining the structure of documents like heading, paragraphs, lists and so forth to facilitate the sharing of scientific knowledge between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language. With the popular version of html which is html5 it becomes easy for developers to build web pages.

-**What does HTML look like?**



The picture above gives a pictorial form of how the HTML format is, it contains different tag which are used for better representation by the browser.

-The <title>tag which is used to give the title of the HTML page name.

-The <body> tag which contains everything that would be displayed by the browser.

**How to create and view HTML**

Because HTML is a markup language, it can be created and viewed in any text editor (web tool such as visual studio code, sublime etc.) if saved as index.html or with a .htm or .html file extension. However, most web developers finds it easy to design and create web pages in HTML using text editors editors. Once the HTML file is created it can viewed locally or uploaded to a web server to be viewed online using a browser.

**4.1.2 Html tags**

HTML tags are like keyword which defines how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are called self closing tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tag have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. some of this essential tags are described below.

|  |  |
| --- | --- |
| Tag | Description |
| <html> | This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>…. </head> and document body which is represented by <body>…</body> tags |
| <head /> | This tag represents the document’s header which can keep other HTML tags like <title>, <link>, <script defer src=”index.js”/> |
| <title> | The <title> tag is used inside the <head> tag to mention the document title. |
| <body> | This tag represents the document body which keeps other HTML tags like <h1>, <div>, <p>, <table> etc. |
| <h1>, <h2>, <h3>, <h4>, <h5>, <h6> | This tags represent the heading tags, each tag has a different font-size. |
| <p> | This tag represents a paragraph |
| <br/>, <li>, <ol>, <ul> | Line break, list, ordered list, unordered list |

The HTML tag syntax is given as <tag name>content</tag name>, this HTML tag names are always written in lower case.

Some example of how this tags work is below;

<p> Paragraph Tag </p>

**<h2> Heading Tag </h2>**

<b> **Bold Tag** </b>

<i> *Italic Tag* </i>

<u> Underline Tag</u>

Other types of HTML tags include:

* UNCLOSED HTML TAGS

Some HTML tags are not closed for example br and hr.

<br>this stands for a line break, it breaks the line of code.

<hr> Stands for horizontal rule. This tags is used to put a line across the web page.

* HTML TEXT TAGS

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <strong>, <em>, <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <pre>, <samp>, <var> and <br> etc.

* HTML LINK TAGS

<a> and <base>

* HTML LIST TAGS

<ul>, <ol>, <li>, <dl>, <dt> and <dd>

These are just few of many other tags in HTML, the HTML tags must be enclosed within < > these bracket.

**4.1.3 Html attributes**

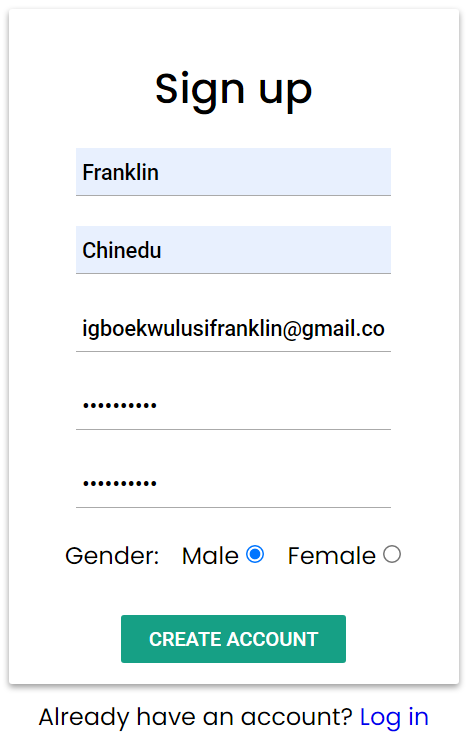
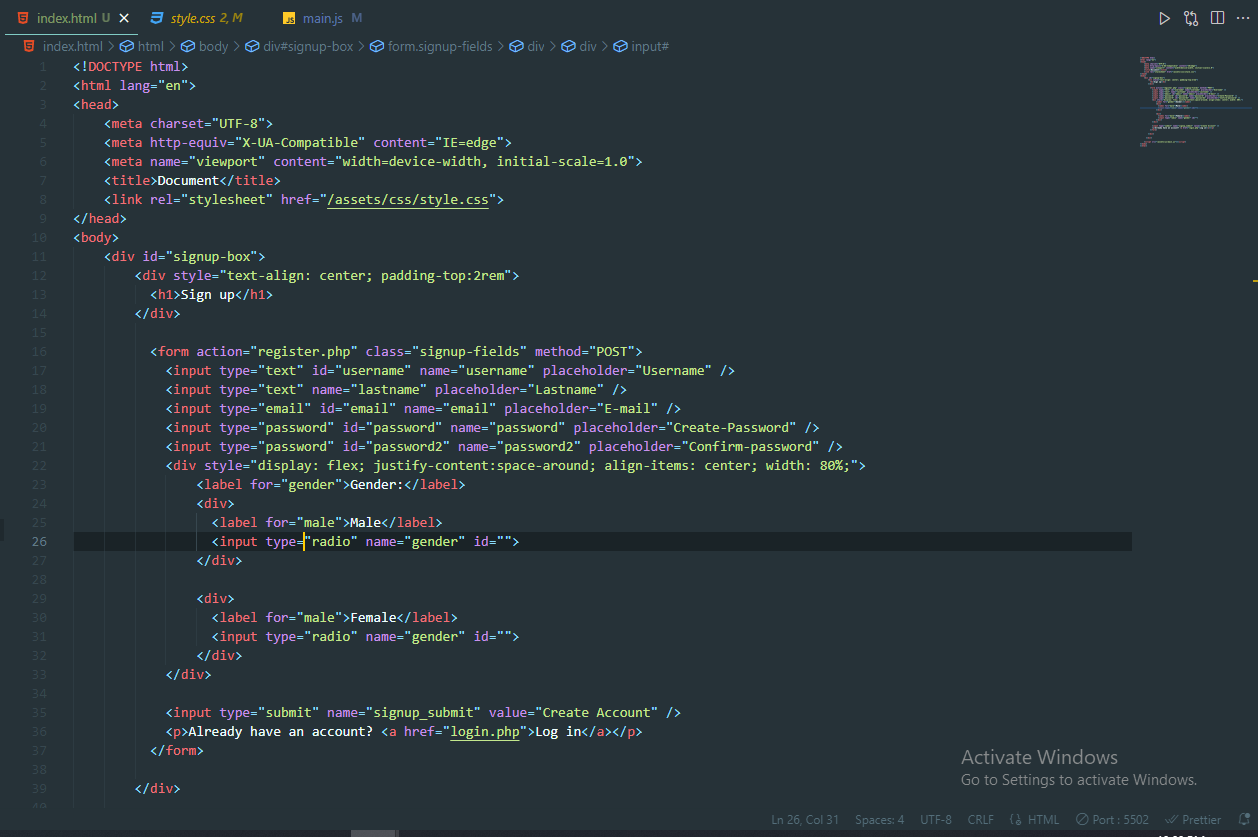
An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts − a **name** and a **value**

* The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
* The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left, center** and **right**.

These attributes are not case sensitive thus the upper case or the lower case can be used.

Some examples of the attributes are;

|  |  |  |
| --- | --- | --- |
| Attributes | Options | Function |
| Align | Right, left, center | Horizontally aligns tags |
| Align | Top, middle, bottom | Vertically aligns tags |
| Bgcolor | numeric, hexidecimal, RGB values | Places a background color behind an element |
| Background | URL | Places a background image behind an element |
| Id | User Defined | Names an element for use with Cascading Style Sheets. |
| Class | User Defined | Classifies an element for use with Cascading Style Sheets. |
| Width | Numeric Value | Specifies the width of tables, images, or table cells. |
| Height | Numeric Value | Specifies the height of tables, images, or table cells. |
| Title | User Defined | "Pop-up" title of the elements. |



The picture above is the output of the HTML code with some little css for styling it.

**4.2** **CSS AND MEANING**

**4.2.1 What is the meaning of CSS**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup language HTML.

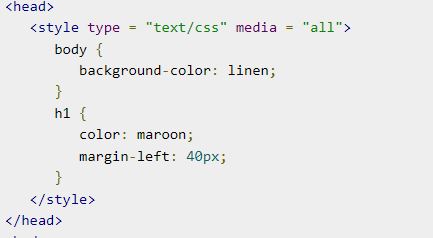
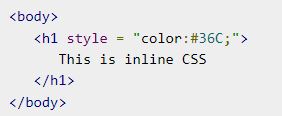
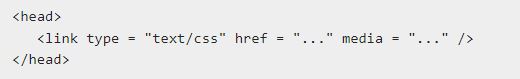
**4.2.2 Methods used by CSS in Formatting HTML Document**

1. **Inline Style:** it is used to apply a unique style to a single HTML element. An inline CSS uses the style attribute of an HTML element.
2. **Embedded / Internal Style:** You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

1. **External Style:** This is used to apply style to multiple HTML pages. With an external style you can change the look of an entire website with just one file. Each page must include a reference inside the <link> tag in the <head> segment of your html page. Placing CSS in a complete page is a best practice in designing a website.

External CSS file is a file that contain only css codes and has a file extension of css eg style.css.

**Image 1 Image 2**

**  **

The images above give a pictorial form of how the css styling can be and linked.

**4.2.3** **CSS syntax**

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
* **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
* **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

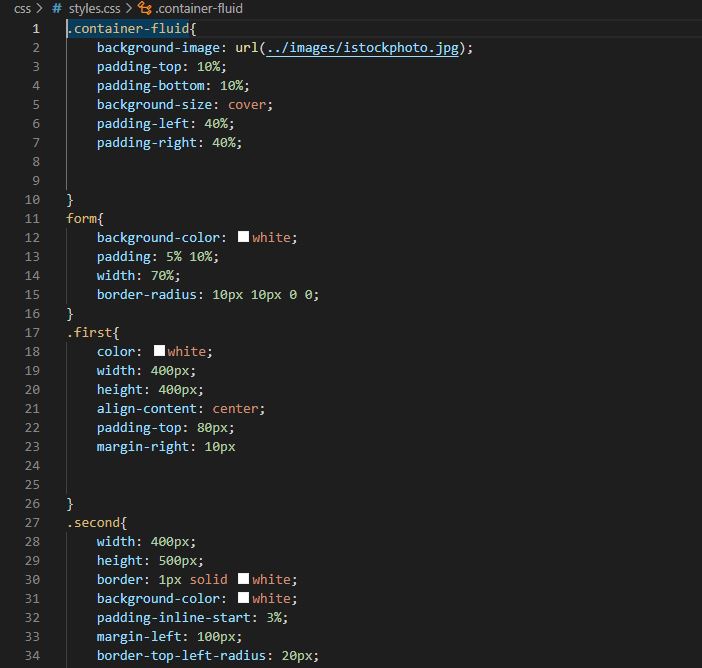
You can put CSS Style Rule Syntax as follows −

selector {

property: value

}

Image of a CSS document



**4.2.4** **Css selector**

**CSS selectors** define the elements to which a set of CSS rules apply. The selectors help to properly design the html document, as the name implies, they are used in selecting a particular HTML tag for styling. Some basic selectors include the following;

1. **Element Selector:** The element selector select element based on the element name.
2. **Id Selector:** The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within the page. So the id selector is used to select one HTML element. E.g <div id=”container” ></div>. To select elements with a specified Id, use a hash tag character [#] followed by the name of the Id e.g #container when HTML looks like <div id=”container”></div>
3. **Class Selector:** the class selector selects elements with specified class attribute. To select elements with a specified class, use a period character [.] followed by the name of the class e.g .container when HTML looks like <div class=”container”></div>.
4. **Attribute selector:** this selects all HTML element that has been given an attributename. **syntax:** [attr] [attr=value] [attr~=value] [attr|=value] [attr^=value] [attr$=value] [attr\*=value]

**Example** [auto play] this selects all element with the attribute name.

**Css comment**

This are writings for easy understanding of the code written, written usually inside the style sheet as

/\*.....this is a comment in style sheet.....\*/.

**4.2.5 Css frameworks**

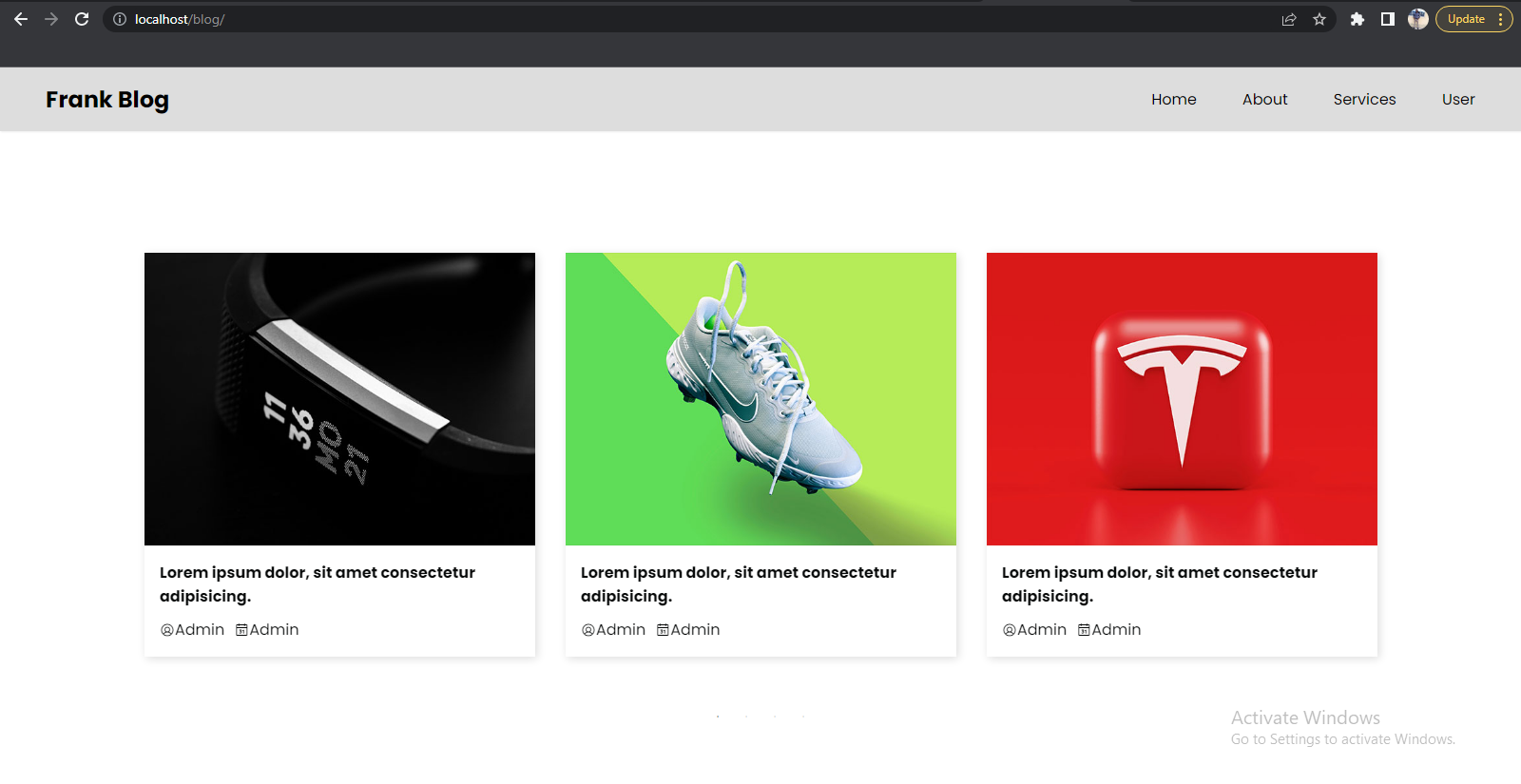
CSS frameworks are tools which developers use to make their job easier. Rather than reinventing the wheel each time a new project comes up, frame works gives the developer the tool to quickly spin up user interfaces that can be tweaked and iterated on throughout a project instead of spending time from a blank document. In its simplest form, a CSS framework is a collection of CSS stylesheets that are prepared and ready to use. They’re tailored for use in common situations, like setting up navbars, accordions, offcanvas, drop-downs, buttons, alerts, cards, forms, list groups, e.t.c and are often expanded upon by other technologies such as SASS and JavaScript.

Examples of css frameworks are Bootstrap, Materialize css, Bulma css, Skeleton css, Tailwind css, e.t.c one of the css framework mostly used is the **Bootstrap version 5.2**

**Web design using HTML and CSS**

So with this two technologies you can design and create a beautiful and responsive site.

**Image of a site built with HTML and CSS**

****

4.3 **JAVASCRIPT**

WHAT IS JAVASCRIPT?

JavaScript is a programming language. It is lightweight and scripting language which makes it fast to run on the web, it is the most common tool used to create web pages, whose implementation interact with the user and create dynamic pages. It is an interpreted programming language with object oriented capabilities. it was developed by Brenan eich 1995-1996, it is an implementation of ECMA script (European Computers Manufacturers Association). It was also known as livescript later change to javaScript. It is used for writing client-side and nowadays server-side of the web.

**4.3.1 What does javascript really do**

The core client-side JavaScript language consists of some common programming features that allow you to do things like:

* Store useful values inside variables. In the above example for instance, we ask for a new name to be entered then store that name in a variable called name.
* Operations on pieces of text (known as "strings" in programming). In the above example we take the string "Player 1: " and join it to the name variable to create the complete text label, e.g. ''Player 1: Chris".
* Running code in response to certain events occurring on a web page. We use a [click](https://developer.mozilla.org/en-US/docs/Web/API/Element/click_event) event to detect when the button is clicked and then run the code that executes a command.
* It is commonly used for writing client-side scripting language with its robust libraries and frameworks, although, javascript can also be used to write the server-side.
* It is used to improve the UI/UX of our web pages by creating animation and dynamic web pages.
* It can also be used for validating forms before submitting to the server for processing.

**4.3.2 Advantages of using Javascript**

1. Less server interaction: you can validate user input before sending the data off to the server, which means fewer load on the server.
2. Increased interactivity: you can create interfaces that react when a user clicks a button, input on a text field, hover a field etc.
3. Richer interfaces: you can use javascript to include item like drag and drop, slider functionality, and some animation effect on your web page.

One of the major strengths of javascript, it is easy to setup. One can begin with a text editor like vs code, atom, brackets or vim. Since it is interpreted, you don’t need a compiler.

**4.3.3 Javascript syntax**

JavaScript syntax refers to a set of rules that determine how the language will be written (by the programmer) and interpreted (by the browser). By learning JavaScript you will become familiar with terms such as variables, functions, statements, operators, data types, objects etc.

**Getting started with javascript**

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page. The HTML [<script>](https://www.quackit.com/html/tags/html_script_tag.cfm) tags tells the browser to expect a script in between them and it is usually placed at the bottom of <body> tag to increase performance.

**An image of a javascript code**



**4.3.4 JavaScript variable**

As with any programming language, JavaScript variables are like a container that contains or holds a value — a value that can be changed as required. For example, you could prompt your website users for their first name. When they enter their first name you could store it in a variable called say, firstName. First, you need to declare your variables. You do this using the **var**, **let** or **const** keyword. You can declare one variable at a time or more than one. You can also assign values to the variables at the time you declare them.

Eg var firstName = “stephen”;

Rules for JavaScript Variables

* Can contain any letter of the alphabet and can be followed by the underscore characters and digits.
* No spaces.
* No punctuation characters (eg comma, full stop, etc).
* The first character of a variable name cannot be a digit.
* JavaScript variables' names are case-sensitive. For example *firstName* and *FirstName* are two different variables.
* You should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section. For example, **break** or **boolean** variable names are not valid.
* JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, **123test** is an invalid variable name but **\_123test** is a valid one.

<**script type = “text/JavaScript> var money; var name;**

**let age; let department;**

**const level; const faculty;**

**</script>**

Multiple variables can also be declared by using the same **var , let or const** keyword as follows

<**script type= “text/JavaScript”> var money, name;**

**let age, department;**

**const level, faculty;**

**</script>**

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

For instance, you might create variable named **money** and assign the value 2000.50 to it later. For another variable, you can assign a value at the time of initialization as follows.

<**script type = “text/javascript”>**

**var name = “Ali”;**

**var money;**

**money = 2000.50;**

**</script>**

The scope of a variable is the region of your program in which it is defined. JavaScript variables have only two scopes.

* **Global variable:** A global variable has global scope which means it can be defined anywhere in your JavaScript code.
* **Local variable:** A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

A list of all the reserved words in JavaScript is given in the following table. They cannot be used as JavaScript variables, functions.

|  |
| --- |
|  |
| abstract | else | instanceof | switch |
| boolean | enum | int | synchronized |
| break | export | interface | this |
| byte | extends | long | throw |
| case | false | native | throws |
| catch | final | new | transient |
| char | finally | null | true |
| class | float | package | try |
| const | for | private | typeof |
| continue | function | protected | var |
| debugger | goto | public | void |
| default | if | return | volatile |
| delete | implements | short | while |
| do | import | static | with |
| double | in | super | forEach |

**4.3.5 JavaScript Operators**

JavaScript operators are used to perform an operation. There are different types of operators for different uses.

Below is a listing of JavaScript operators and a brief description of them.

## Arithmetic Operators

|  |  |
| --- | --- |
| Operator | Description |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus (remainder of a division) |
| ++ | Increment |
| -- | Decrement |

## Assignment Operators

|  |  |
| --- | --- |
| Operator | Description |
| = | Assign |
| += | Add and assign. For example, x+=y is the same as x=x+y. |
| -= | Subtract and assign. For example, x-=y is the same as x=x-y. |
| \*= | Multiply and assign. For example, x\*=y is the same as x=x\*y. |
| /= | Divide and assign. For example, x/=y is the same as x=x/y. |
| %= | Modulus and assign. For example, x%=y is the same as x=x%y. |

## Comparison Operators

|  |  |
| --- | --- |
| Operator | Description |
| == | Is equal to |
| === | Is identical (is equal to and is of the same type) |
| != | Is not equal to |
| !== | Is not identical |
| > | Greater than |
| >= | Greater than or equal to |
| < | Less than |
| <= | Less than or equal to |

**4.3.6 Conditional Statement**

When you write code, you will often need to use conditional statements, such as "if" statements. Here's an explanation of the JavaScript If statement.

A *conditional statement* refers to a piece of code that does one thing based on one condition, and another based on another condition. In fact, you could have as many conditions as you like.

JavaScript If statements are an example of conditional statements. With If statements, you can tell the browser to execute a piece of code only *if* a given condition is true.

### **To create a JavaScript If statement**

1. Start with the word if.
2. Between open and closed brackets, write the actual condition that is being tested (i.e. if something is equal to something else).
3. Between open and closed curly brackets ({}), specify what will happen if the condition is satisfied.

Example: *if (20 > 30) {*

*Console.log(“True”)*

*}*

# **JavaScript If Else statement**

The above code is OK if you only want to print out ***“True”*** when the condition is true. But what if you want to print out something else like ***“False”*** when the condition is not true. For example, what if the 20 is not greater than 30?

Example:

*if (20 > 30) {*

*console.log(“True”)*

*} else {*

*console.log(“False”)*

*}*

This is where an If Else statement comes in handy. The else part is what we're interested in here. The else is what tells the browser what to do if the condition is not true.

**Switch Statement**

The Switch statement can be used in place of the If statement when you have many possible conditions.

An advantage of using the switch statement is that it uses less code, which is better if you have a lot of conditions that you need to check for.

**4.3.7 Javascript Loops**

Loops can execute a block of code a number of times.

If you want to run the same code over and over again, each time with a different value.

**Different Kinds of Loops**

JavaScript supports different kinds of loops:

* for - loops through a block of code a number of times
* while - loops through a block of code while a specified condition is true
* do/while - also loops through a block of code while a specified condition is true

**While Loop**

In JavaScript (and most other languages), "loops" enable your program to continuously execute a block of code for a given number of times, or while a given condition is true. The JavaScript While loop executes code while a condition is true.

Syntax:

while (condition) {  
*// code block to be executed*  
}

Example:

let i = 0 while (i < 10) {  
  console.log(i);  
  i++;  
}

## The Do While Loop

The do while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax:

do {  
*// code block to be executed*}  
while (condition);

Example:

The example below uses a do while loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

do {  
  console.log(i);  
  i++;  
}  
while (i < 10);

**The for Loop**

The JavaScript For loop executes code for a specified number of times

Syntax:

for (*expression 1*; *expression 2*; *expression 3*) {  
  // *code block to be executed*  
}

**Expression 1** is executed (one time) before the execution of the code block.

**Expression 2** defines the condition for executing the code block.

**Expression 3** is executed (every time) after the code block has been executed.

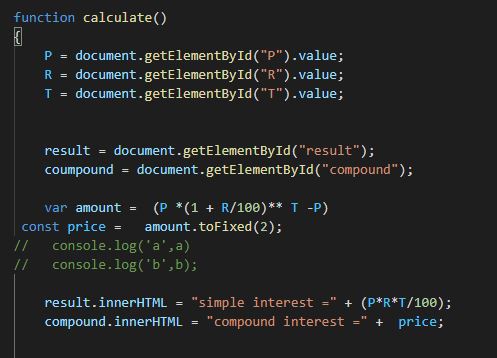
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}

**4.3.8 Javascript functions**

A JavaScript function is a block of code designed to perform a particular task, this function is executed when something invokes it (calls it ).

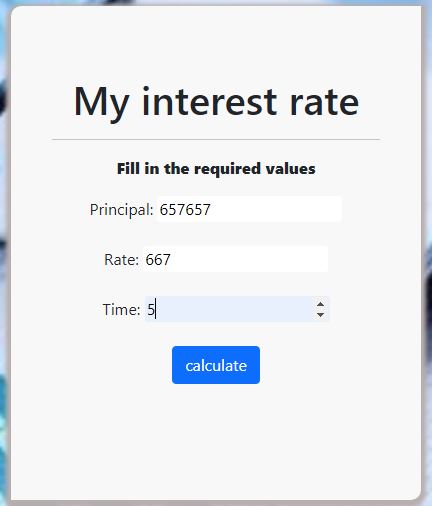
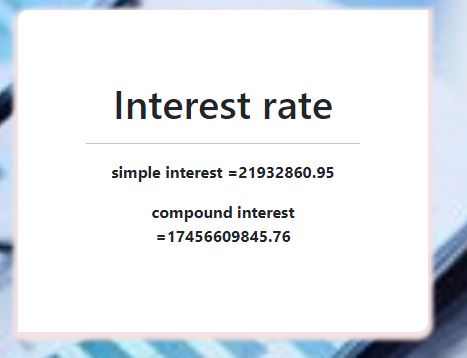
A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses **()**. Function names can contain letters, digits, underscores, and dollar signs.

The parentheses may include parameter names separated by commas:  
**(*parameter1, parameter2, ...*)**The code to be executed, by the function, is placed inside curly brackets: **{}** an example is showed below



The JavaScript code shown above calculates a simple interest, when the numbers are inputted the function is called or invoked by clicking the calculate button.

The output of the code is shown below;

**Image 1 Image 2**