KWARA STATE POLYTECHNIC, ILORIN

INSTITUTION OF INFORMATION COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

A TECHNICAL REPORT OF THE STUDENTS INDUSTRIAL WORK EXPERIENCE SCHEME (SIWES)

UNDERTAKEN AT:

G-TECHNOLOGY TRAINING INSTITUTE

ILORIN KWARA STATE

SUBMITTED BY:

OLADEPO EUNICE BOLUWATIFE

ND/23/COM/PT/0144

SUBMITTED TO:

THE DEPARTMENT OF COMPUTER SCIENCE

PERIOD OF ATTACHMENT:

AUGUST - NOVEMBER 2024

DEDICATION

I dedicate my report on industrial training to the Almighty God, who has granted me the opportunity to take part in the SIWES program, as well as to my parents and numerous others who have made significant contributions to the success of my industrial training.

ACKNOWLEDGEMENT

I give thanks to my industrial supervisor who helped me with my industrial training and to God for seeing me through my SIWES program. I also like to thank my coworkers, friends, and professors for their moral support.

My sincere gratitude goes out to my amazing and charming parents, Mr. and Mrs. OLADEPO, who supported me financially, with prayers, and in other ways.

ABSTRACT

The Web Development SIWES program offers a comprehensive learning experience in building and maintaining websites, focusing on the essential skills needed to design, develop, and optimize dynamic web applications. Over the course of the program, participants gained hands-on experience with front-end and back-end technologies, including HTML, CSS, JavaScript, PHP, and various databases. Emphasis was placed on creating responsive, user-friendly designs and implementing efficient coding practices. Additionally, the program explored web frameworks, version control systems, and best practices for optimizing web performance and security. By the end of the program, participants were equipped with the practical knowledge and skills to build functional websites and applications, preparing them for careers in web development or further specialized training.

TABLE OF CONTENT

3.5 **JAVASCRIPT**

- Data type in JavaScript
- Comment in JavaScript
- Conditional /decision making in JavaScript

CHAPTER FOUR

4.0 Challenges

4.1 Recommendations

CHAPTER FIVE

5.0 Conclusion

CHAPTER ONE

1.0 INTRODUCTION

The Student Industrial Work Experience Scheme (SIWES) is a critical initiative designed to enrich the educational journey of students in technical and vocational fields. Typically undertaken by students in their penultimate or final years of study, SIWES provides a unique platform for integrating theoretical knowledge with practical applications in real-world environments. As students transition from academic institutions to the professional workforce, this program offers invaluable opportunities for hands-on experience in various industries, fostering essential skills and competencies needed to excel in their careers.

SIWES is not merely an internship; it is a structured program that allows students to immerse themselves in the dynamics of the workplace, engage with industry professionals, and apply their academic learning to address actual challenges facing businesses today. This experiential learning approach promotes personal and professional growth, enhances students' understanding of industry standards, and prepares them for competitive job markets. Through SIWES, students can also establish vital networks that may facilitate future employment opportunities.

1.1 OBJECTIVES OF SIWES:

- **Practical Experience:** To give students hands-on experience in their field of study, bridging the gap between theoretical knowledge and practical application.
- **Skill Development:** To develop and enhance students' technical and soft skills, preparing them for the workforce.
- **Industry Exposure:** To expose students to the working culture and environment of various industries, helping them understand how businesses operate.
- **Networking Opportunities:** To provide students with opportunities to build professional networks that can facilitate future job placements or internships.
- **Problem-Solving Abilities:** To help students learn to identify and solve real-world problems, fostering critical thinking and innovation.
- **Research and Development:** To encourage students to participate in research projects that can contribute to their academic and professional knowledge.
- **Career Guidance:** To provide students with insights into career paths and job roles in their field, helping them make informed decisions about their future.

CHAPTER TWO

2.0 DESCRIPTION OF THE ESTABLISHMENT OF ATTACHMENT

G-tech is an entirely homegrown Nigerian firm that offers clients innovative IT and security solutions that are both high-quality and cost-effective. The company places a strong emphasis on ensuring complete client satisfaction. Over the years, G-tech has built a reputable standard thanks to its comprehensive technical expertise and proficiency in project management and implementation.

2.1 THE OBJECTIVES OF THE COMPANY

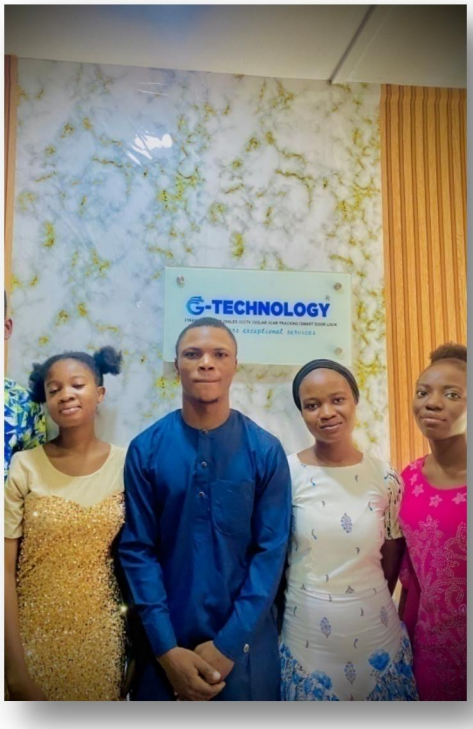
1. To offer a comprehensive and up-to-date curriculum that encompasses the latest technologies and industry trends to ensure our students acquire relevant competencies.
2. To provide practical, hands-on training experiences through labs, workshops, and projects that enable students to apply theoretical knowledge in real-world scenarios.
3. To promote accessibility in technology education by offering diverse programs that cater to individuals from various backgrounds, ensuring an inclusive learning environment.

2.2 CORE VALUES

- Honesty
- Service
- Commitment
- Excellence
- Professionalism

2.3 VISION OF THE COMPANY

To be a leading institution in technology education, empowering individuals with the skills, knowledge, and confidence to innovate and excel in a rapidly evolving digital landscape. We envision a future where our graduates are at the forefront of technological advancement, contributing to sustainable development and impactful solutions for global challenges.



2.4 COMPANY AREA OF SPECIALIZATION

With a team of professional Computer programmers, web development, data science, Graphic designers, UI/UX, and cyber-security, the Company has a reputable recognition in the following areas:

1. Software Development

Software development involves the process of designing, coding, testing, and maintaining software applications. This field encompasses a variety of programming languages, frameworks, and methodologies. Software developers work on applications that run on various platforms, including desktop, mobile, and cloud environments. Key concepts include software engineering principles, version control, agile methodologies, and user experience design.

Developers may specialize in different areas, such as front-end development (user interface) or back-end development (server-side logic).

2. Web Development

Web development focuses specifically on the creation and maintenance of websites and web applications. It includes both front-end development, which involves the design and interactivity of the website that users see, and back-end development, which deals with server-side logic, databases, and application integration. Technologies such as HTML, CSS, JavaScript, and various frameworks (e.g., React, Angular, and Node.js) are commonly used. Web developers must ensure websites are functional, responsive, and accessible across different devices and browsers.

3. Cyber Security

Cyber-security is the field dedicated to protecting systems, networks, and data from cyber threats and attacks. It involves identifying vulnerabilities, implementing security measures, and responding to incidents. Cyber-security specialists work to safeguard information through various practices, including encryption, firewalls, intrusion detection systems, and security protocols. They also focus on compliance with regulations and standards to ensure data privacy. With the growing prevalence of cyber threats, the demand for cyber-security professionals continues to rise.

4. Data Science

Data science is the study and analysis of complex data sets to derive insights, make informed decisions, and drive business strategy. This interdisciplinary field combines statistics, mathematics, programming, and domain expertise to extract meaning from structured and unstructured data. Data scientists use tools and programming languages like Python, R, and SQL, along with machine learning techniques and data visualization tools, to analyze trends and patterns. The profession is crucial in various industries, enabling organizations to harness the power of data for predictive modeling, customer segmentation, and more.

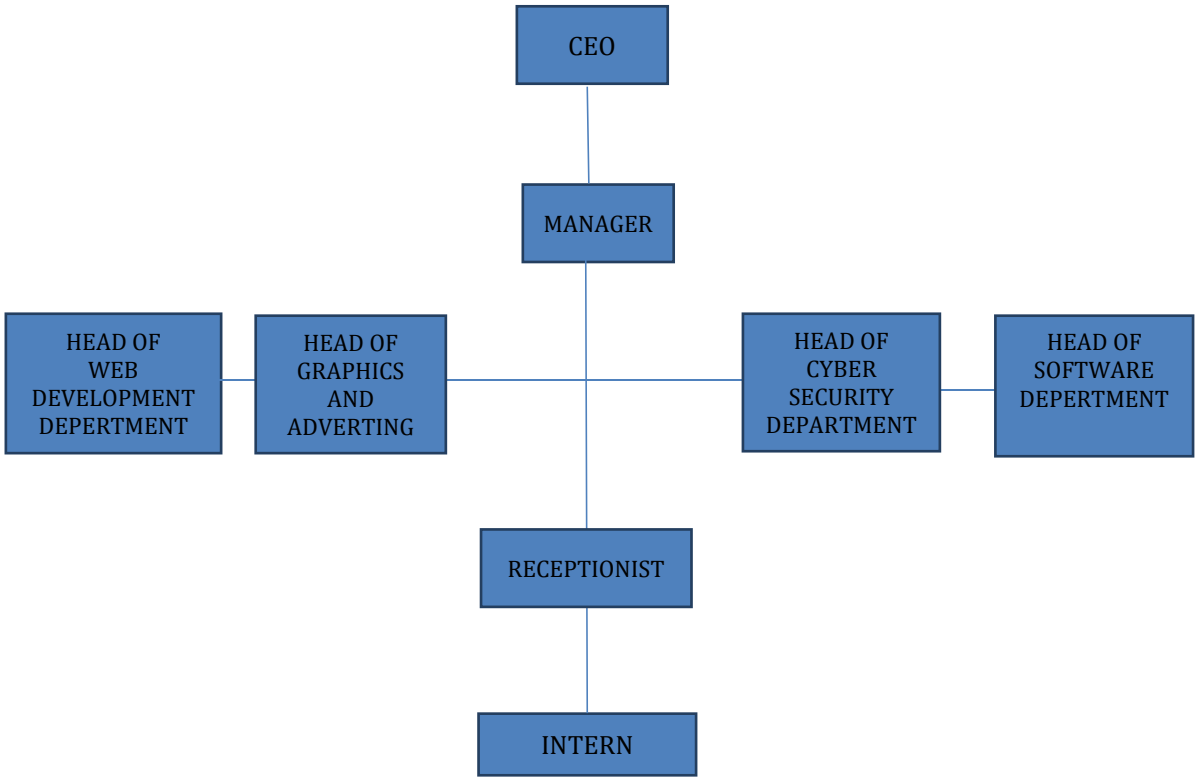
5. **Graphic Design**

Graphic Design is a field of computer science that involves the visual communication of ideas and messages through the use of typography, color, and images. It involves the creation of visual elements such as logos, graphics, icons, and graphics for websites, magazines, newspapers, and other media.

6. **UI/UX DESIGN**

Our business focuses in developing user interface (UI) and user experience (UX) designs that improve digital interactions by being clear and captivating. In order to create solutions that are not only aesthetically beautiful but also incredibly practical, we concentrate on comprehending user behavior, preferences, and demands.

2.5 STRUCTURE DIAGRAM OF G-TECHNOLOGY TRAINING INSTITUTE



CHAPTER THREE

3.0 INDUSTRIAL EXPERIENCE

During my SIWES in web development i gain practical, hands-on experience in the design, development, and deployment of web applications. During this period, I sought to enhance my technical skills in programming languages such as HTML, CSS, and JavaScript. Also, I aimed to understand the software development lifecycle, collaborate with a team of developers, and learn how to effectively gather and implement user requirements to create functional and user-friendly web solutions.

3.1 INTRODUCTION TO WEB DEVELOPMENT

Web development is the process of building and maintaining websites and web applications for the internet or an intranet. It involves a combination of various tasks and skills to create a functional and visually appealing online presence. It is divided into (3) which involve:

Frontend development

1. Backend development
2. Full – stack development

Web development involves various aspects such as:

1. Web design
2. User experience (ux) design
3. Web security
4. Mobile application development
5. Web assembly
6. Search engine optimization (SEO)
7. E-commerce development

Definition Web Developer

Web Developer: (web programmer or web coder) is a person who builds and maintains website, web application and mobile application using various programming language, framework and tools.

Web developer uses various tools such as:

1. Text editor (e.g. visual studio code)
2. Integrated development environment (IDE)
3. Database management system

4. Testing framework
5. Version control system

3.2 STATUS CODE

A status code is a three- digit number that a web server user to communicate the outcome of a request to a client (usually a web browser). It's a way for the server to say "here's what happened with your request ".

FIVE CATEGORIES OF STATUS CODES

1. **1xx :** informational (request received, processing)
2. **2xx:** success (request successful data returned)
3. **3xx:** redirection (resource moved, redirect to another URL)
4. **4xx:** client error (request invalid, access denied)
5. **5xx:** server error (server-side error, request failed)

SOME COMMON STATUS CODE INCLUDES:

- **200:** ok (request successful)
- **404:** not found (resource not found)
- **500:** internal server error (server-side error)
- **301:** moves permanently (resource moved, new URL)
- **400:** bad request (invalid request error in syntax)

3.3 FRONTEND DEVELOPMENT

- **HTML (HYPERTEXT MARKUP LANGUAGE)**

Html (Hypertext Markup Language): is the standard markup language used to create web page it's the backbone of a website, providing the structure and content that the web browser render the user. It defines the structure and layout of a web page, html consists of series of element, represented tags (< >), these elements include:

1. Heading (h1 – h6)
2. Image (img)
3. Span (span)
4. Title (title)
5. Paragraph (p)
6. Form (form input, select) etc.

HTML LISTS

HTML lists : are used to group related items together and can be presented in various formats.

There are three main types of lists in HTML:

1. Ordered Lists (<ol>): These lists display items in a specific order, usually numbered. Each item in an ordered list is wrapped in a <li> (list item) tag. The numbering is automatically generated by the browser.

Example:

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

2. Unordered Lists (<ul>): These lists display items without a specific order, typically using bullet points. Like ordered lists, each item is wrapped in a <li> tag.

Example:

```
<ul>
  <li>Item A</li>
  <li>Item B</li>
  <li>Item C</li>
</ul>
```

3. Description Lists (<dl>): These lists are used for terms and their descriptions. A description list consists of <dt> (description term) and <dd> (description definition) tags.

Example:

```
<dl>
  <dt>HTML</dt>
  <dd>A markup language for creating web pages.</dd>
  <dt>CSS</dt>
  <dd>A style sheet language used for describing the presentation of a document written in HTML.</dd>
</dl>
```

BLOCK AND INLINE ELEMENT

Block Elements

Block elements occupy the full width available (from left to right) and create a new line (or block) before and after the element. They are typically used for larger structural components of a webpage.

Characteristics of block element

- Start on a new line.
- Take up the full width of their parent container by default.
- Can contain other block elements and inline elements.

Common Block Elements

- : A generic container for grouping elements.
- <h1>, <h2>, ..., <h6>: Headers of different importance levels.
- <p>: Paragraphs.
- and <ol>: Unordered and ordered lists, respectively.
- <blockquote>: For quoting sections of text.

Example:

```
<div>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</div>
```

Inline Elements

Inline elements do not start on a new line and only occupy the width necessary to contain their content. These elements are typically used for smaller parts of content within a block element.

Characteristics of inline element

- Do not start on a new line.
- Only take up as much width as they need.

. Cannot contain block elements (but can contain other inline elements).

Common Inline Elements

- : A generic inline container for text.
- <a>: Anchor or hyperlink tags.
- <img>: For embedding images.
- : For important text (usually bold).
- : For emphasized text (usually italicized).

Example:

This is a bold word in a paragraph.

This is a link

HTML DOM [DOCUMENT OBJECT MODEL]

CODE INSPECTION

Code Inspect: typically refers to the ability to view and interact with the underlying code of a webpage directly within a web browser. This is most commonly done through a feature known as the "Developer Tools" or "DevTools", which is available in most modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

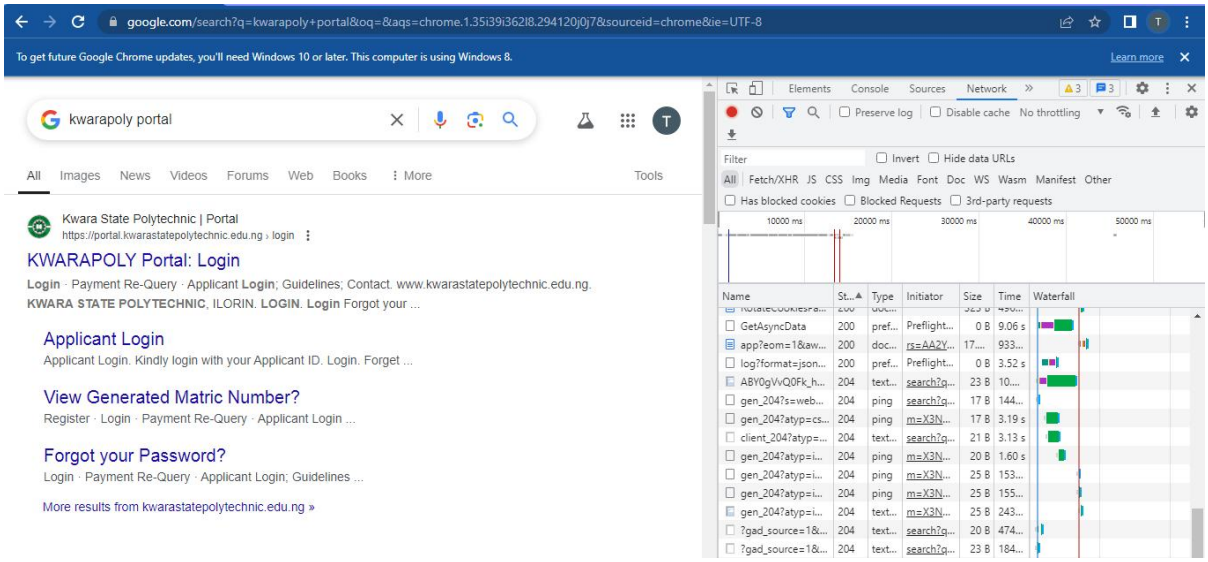
Character Of Developer Tools:

1. Inspect Elements: You can right-click on any element on a webpage and select "Inspect" (or "Inspect Element") to view its HTML structure, CSS styles, and other attributes. This allows developers and designers to see how various elements of a page are structured.
2. Debug JavaScript: The console allows you to run JavaScript code directly on the page and debug scripts. You can set breakpoints, step through code, and inspect variables.
3. Network Monitoring: You can view all network requests made by the webpage, including the time taken to load resources, independent of whether they are images, scripts, or styles. This is particularly useful for performance tuning.
4. Storage Inspection: Developers can view and manipulate cookies, local storage, indexedDB, and session storage to debug issues related to client-side storage.

5. Accessibility Audit: Tools available in DevTools can help you analyze a webpage's accessibility and check for any common accessibility issues.

6. CSS Editing: You can directly modify HTML and CSS in the browser to see how changes affect the appearance and behavior of the page in real-time.

To access Developer Tools, you can usually right-click on a webpage and select "Inspect" or press F12 or Ctrl + Shift + I (or Cmd + Option + I on Mac) depending on the browser you are using.



3.4 CSS (CASCADING STYLE SHEETS)

CSS (Cascading Style Sheets): is a styling language used to control the layout and visual appearance of web page written in html. It also, used in target specific html element to apply style and consist of properties and value example (color: red; padding: 10px ;,) etc.

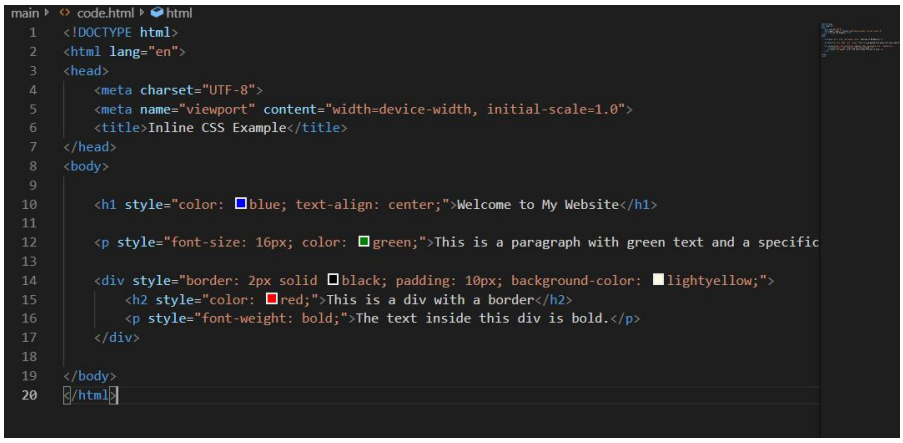
HOW TO CONNECT YOUR CSS TO YOUR HTML

We can connect CSS to html in 3 ways:

1. Inline CSS
2. Internal CSS
3. External CS

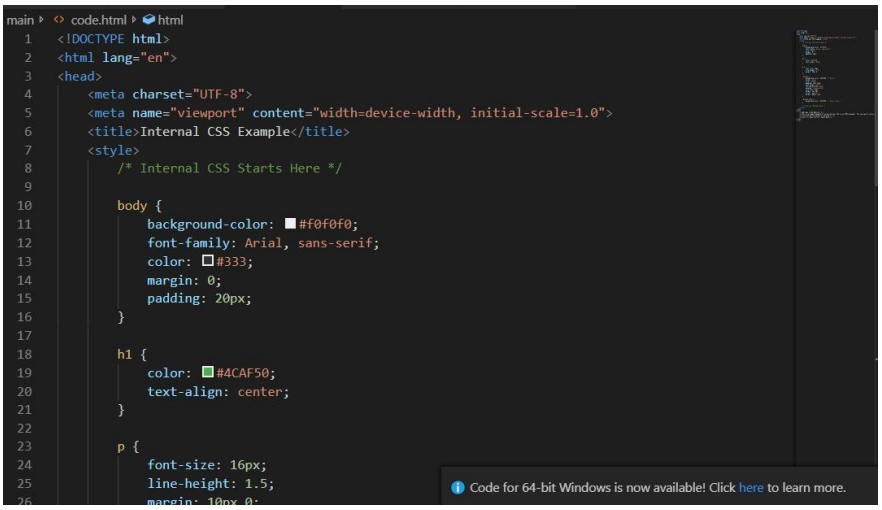
1. **Inline CSS:** written directly within an html element using the style attribute.

   Example; <p style = “color: blue ;”> this is inline CSS </p>



```
main ▸ code.html ▸ html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inline CSS Example</title>
</head>
<body>

    <h1 style="color: blue; text-align: center;">Welcome to My Website</h1>

    <p style="font-size: 16px; color: green;">This is a paragraph with green text and a specific

    <div style="border: 2px solid black; padding: 10px; background-color: lightyellow;">
        <h2 style="color: red;">This is a div with a border</h2>
        <p style="font-weight: bold;">The text inside this div is bold.</p>
    </div>

</body>
</html>
```

2. **Internal CSS:** written within an html file using the <style> tag in the <head> section.
   Example: <style> /* style here */ </style>



```
main ▸ code.html ▸ html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Internal CSS Example</title>
    <style>
        /* Internal CSS Starts Here */

        body {
            background-color: #f0f0f0;
            font-family: Arial, sans-serif;
            color: #333;
            margin: 0;
            padding: 20px;
        }

        h1 {
            color: #4CAF50;
            text-align: center;
        }

        p {
            font-size: 16px;
            line-height: 1.5;
            margin: 10px 0;
Code for 64-bit Windows is now available! Click here to learn more.
```

3. **External CSS**: is separate from html while internal and inline CSS are embedded within html. It generally preferred for maintainability and scalability.

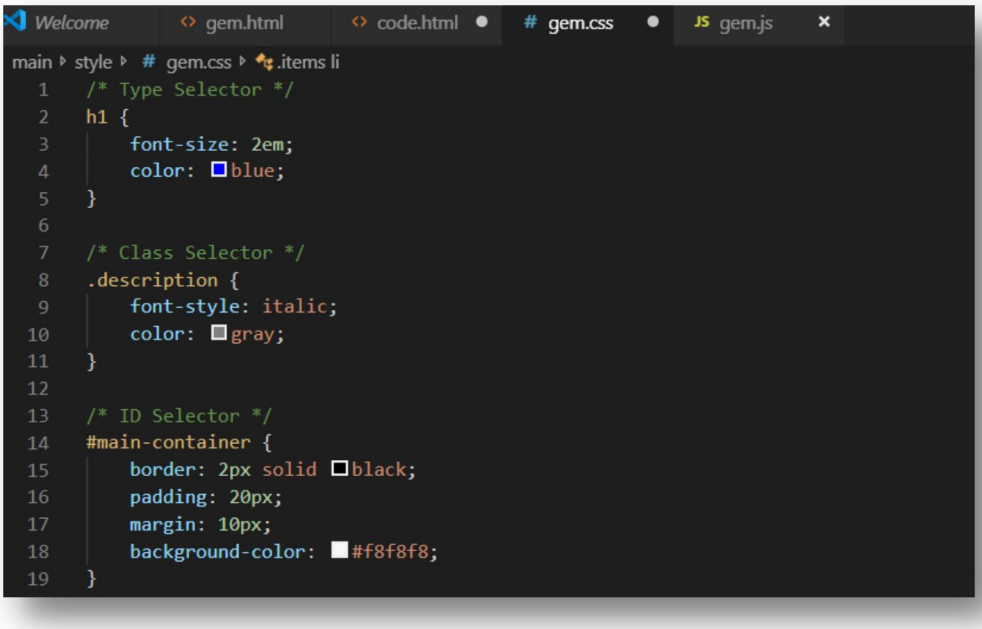
```
Welcome | gem.html | code.html | # gem.css | JS gem.js
main ▸ style ▸ # gem.css ▸ .hub
body {
    position: relative;
    text-align: center;
    background: purple;

}

h1 {
    color: white;

}

.hub {

    width: 350px;
    border: solid black 5px;
```

CSS SELECTOR

CSS (Cascading Style Sheets) selectors are patterns used to select elements in an HTML document that you want to style.

TYPE OF CSS SELECTOR

1. Type Selector: Targets all elements with a specific tag name (e.g., h1, p).
2. Class Selector: Targets all elements with a specific class (e.g., .nav, .header).
3. ID Selector: Targets a single element with a specific ID (e.g., #header, #footer).



CSS POSITION

In CSS (Cascading Style Sheets), the position property is used to control the positioning of an element in the document. It defines how an element is positioned in relation to its normal position, its parent element, or the viewport. There are several values that the position property can take, each affecting layout in different ways:

1. **Static:** This is the default value. Elements are positioned according to the normal flow of the document. Setting the top, right, bottom, or left properties will have no effect on a statically positioned element.

```
.element {
   Position: static;
}
```

2. **Relative:** The element is positioned relative to its normal position. Setting top, right, bottom, or left will move the element from where it would normally be in the document flow without affecting other elements.

```
.element {
   Position: relative;
   Top: 10px; /* Moves the element down 10 pixels */
```

3. **Absolute:** The element is positioned relative to the nearest positioned ancestor (an ancestor with a position value of anything other than static). If there is no such ancestor, it is positioned relative to the initial containing block (usually the viewport). The element is removed from the normal document flow.

```
.element {
   Position: absolute;
   Top: 50px; /* 50 pixels from the top of the nearest positioned ancestor */
}
```

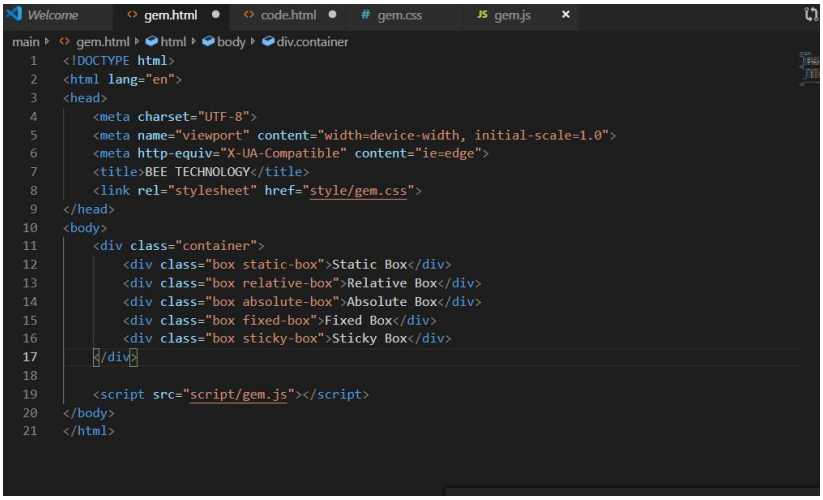
4. **Fixed:** The element is positioned relative to the viewport, which means it stays in the same position even when the page is scrolled. It is also removed from the normal document flow.

```
.element {
   Position: fixed;
   Top: 0; /* Sticks to the top of the viewport */
}
```

5. **Sticky:** The element toggles between relative and fixed positioning depending on the user's scroll position. It is treated as relative until it crosses a specified threshold (like top, right, bottom, or left), at which point it becomes fixed.

```
.element {
   Position: sticky;
   Top: 10px; /* Becomes fixed when scrolled to 10px from the top */
}
```

CSS POSITION USING EXTERNAL CSS CONNECTION TO HTML



3.4 JAVASCRIPT

JavaScript: is a high-level dynamic and interpreted programming language used for client-side scripting on the web. It allows developer to create interactive web page, web application and mobile application. It is used for:

1. Client – side scripting
2. Server – side programming
3. Game development
4. Mobile development

WAY TO DECLARE A VARIABLE IN JAVASCRIPT

1. Var.
2. Let.
3. Const.

1. **Var.:** is a global object that can be access anywhere. It didn't have a scope.

Example:
```
var. name: "mercy";
Console.log (name)
```

2. **Let :** is a block scope ,

Anything inside curly bracket define scope

Example:
```
{
  Let name = "john";
}
Console.log (name);
```

CONCATENATION

Concatenation: mean joining two things together to become one. Using template string to concatenate:

1. Back tick
2. Dollar sign
3. Curly bracket

Example:

```
Let first name = "jerry"
Let second name = "Akanke"
let full name = $`{first name}${second name}`
Console.log (full name);
```

- **Console:** this is a feature in almost all browser it work with java Script to output a value to the console.
- **Alert [ ] method:** is a built in function that display a message box with a specified message and an ok button. it is often used to alert or notify the user of something important

COMMENT IN JAVASCRIPT

Comments in JavaScript are used to provide insights, explanations, or annotations within the code. They are ignored by the JavaScript engine when the code is executed, allowing developers to leave notes for themselves or others without affecting the program's functionality.

There are two types of comments in JavaScript:

1. **Single-line comments:** These comments are created using two forward slashes (//). Everything following // on that line will be treated as a comment.

For example:

```
// this is a single-line comment
Let x = 5; // Assigns 5 to x
```

2. **Multi-line comments:** These comments can span multiple lines and are created by enclosing the comment text between /* and */.

For example:

```
/* this is a multi-line comment
  That can span multiple lines */
Let y = 10; /* this is an inline comment */
```

DATA TYPE IN JAVASCRIPT PROGRAMMING

Data type is a classification of data based on its format, size, and set of values, it can hold. JavaScript

Is a dynamically-typed language, which means that you don't need to declare the data type of a variable before using it?

TYPE DATA TYPE IN JAVASCRIPT

1. **Number:** represents a numeric value e.g., 42,3.14
2. **String:** represent a sequence of character e.g., "hello"
3. **Boolean:** represent a true or false value.
4. **Null:** represent the absence of any value.

5. **Variable:** Are containers that hold a value think of it like a labeled box where you can store value. E.g.,

Age =20,

console.log (age)

Type of Variable

- **Local variable:** is anything that inside the curly bracket
- **Global variable:** is outside the curly bracket.

6. Array: represent a collection of value, e.g., [1,2,3,4,5,6]

Method of array

1. **Push:** it is use to push information into array

Example:

```
Let user = [
"Samuel",
"Jade",
]
ser. Push ("favor")
Console.log (user)
```

2. **Pop :** is used for delete item from the back

Example:

```
Let user = [ ]
User. push ("Tom")
User. push ("Sunday")
User.pop ("favor")
Console.log (user)
```

3. **Shift () method:** is used to remove the first element from an array and return that element. It change the length of the array and shift all the element down by one position

```
/*
Let color = ["red", "green", "blue", "yellow",]
Console.log (color. shift ()); // output: "red",
Output ["green", "blue", "yellow",]        */
```

4. **Un-shift () method**: is a built-in JavaScript array method that adds one or more elements to the beginning of an array. It modifies the original array and returns the new length of the array after the elements have been added.

Example:

```
Let fruits = ['banana', 'orange', 'apple'];
Fruits. Un-shift ('mango'); // Adding a single element
console.log (fruits); // Output: ['mango', 'banana', 'orange', 'apple'].
```

7. **Object:** is a complex data type that allows you to store collections of data and more complex entities. Objects can hold multiple values in the form of key-value pairs, where each key (also known as a property) is a string (or a Symbol) that map to a value, which can be of any data type, including another object or even a function.

Example;

```
 Const person = {
   Name: "Alice",
   Age: 25,
   Is Student: false
};
console.log (person)
```

8. **Function:** is a reusable block of code that performs a specific task. Functions allow you to encapsulate logic, making it easier to write, manage, and reuse code. Functions can take inputs (called parameters), perform operations, and return outputs (values).

<u>2 Common Way To Declare A Function</u>

1. **Function Declaration:** is a way to define a named function using the function keyword. This is one of the most common ways to create functions in JavaScript. Function declarations are hoisted, meaning they can be called before they are defined in the code, making them flexible for organizing code.

Example:

```
Function functionName (parameter1, parameter2) {
   // Code to be executed
   Return result; // Optional
}
```

Example:

```
Functions add (a, b) {
   Return a + b;
}
console.log (add (3, 5)); // Output: 8
```

2. **An arrow function:** is a feature in JavaScript introduced with ECMAScript 6 (ES6) that provides a more concise syntax for writing function expressions.

```
Const function Name = (parameter1, parameter2) => {
   // Code to be executed
   Return result; // Optional
};
```

Example:

```
Const subtract = (a, b) => a - b; // Single expression, implicit return
console.log (subtract (10, 4)); // Output: 6
```

CONDITIONALS/DECISION MAKING IN JAVASCRIPT

Condition or decision-making in JavaScript allows you to execute different blocks of code based on certain conditions. This is a fundamental aspect of programming that enables dynamic and flexible application behavior. JavaScript provides several statements for decision-making, including if, else, switch, and the ternary operator.

1. **if Statement**

If statement: evaluates a condition and executes a block of code if the condition is true.

Example:

```
Let temperature = 30;
If (temperature > 25) {
   console.log ("It's a hot day.");
}
```

2. **if...else Statement**

You can provide an alternative block of code to execute if the condition is false using else.

Example:

```
Let temperature = 20;
If (temperature > 25) {
   console.log ("It's a hot day.");
} else {
   console.log ("It's not a hot day.");
}
```

3. **else if Statement**

For multiple conditions, you can use else if to test additional conditions.

Example:

```
Let temperature = 10;
If (temperature > 25) {
   console.log ("It's a hot day.");
} else if (temperature < 15) {
   console.log ("It's a cold day.");
} else {
   console.log ("It's a mild day.");
}
```

CHAPTER FOUR

4.0 CHALLENGES

I find it challenging to comprehend and use version control systems like Git, which could cause problems with code management and teamwork. Making it a habit to write test cases and check the quality of the code may be neglected, which could result in production bugs. It can be very annoying to troubleshoot bugs and errors, particularly if you have never dealt with similar issues before. If proper breaks and self-care are not taken , long hours of coding and problem-solving can result in burnout. Notwithstanding these difficulties, I learned a lot from the program. Although these difficulties can be intimidating, they also offer worthwhile educational opportunities that can improve my abilities and fortitude as a web developer. By accepting these difficulties, asking for assistance when necessary, and keeping a growth mindset, I am better able to manage my SIWES program.

4.1 RECOMMENDATIONS

I implore the federal government to give the SIWES program more consideration, particularly for me. T students and the Applied Sciences, as the world is moving in the direction of I. T. . For the program's goal to be preserved, the government should make sure that SIWES students are properly supervised. Given the program's potential to improve pool I's quality, the federal government should also include sufficient funding for SIWES in the yearly budget. The nations available T skills. To help SIWES students find employment in the industry, a thorough and detailed directory of sponsors is desperately needed. To keep the SIWES students enthusiastic and motivated, the ITF should also make sure that the student allowance is paid on schedule. The Nigerian educational system needs to be reviewed, in my opinion, because what is taught in the classroom completely deviates from what is required in the workplace.

CHAPTER FIVE

5.0 CONCLUSION

My SIWES was a huge success, and I gained knowledge about the field of information technology. I now understand the fundamentals of web development. This will enable me to set a goal for myself to write code for various programs and create a comprehensive website. I have been genuinely exposed to the difficulties of a developing self through SIWES as a course. a world where computers are essential.