**A TECHNICAL REPORT ON**

**STUDENT INDUSTRIAL WORK EXPERIENCE SCHEME (SIWES)**

**CARRIED OUT AT**

**UNIVERSITY OF MKAR,**

**GBOKO, BENUE STATE**

**FROM: 4TH JANUARY, 2023**

**TO: 4TH MAY, 2023**

**BY**

**AJAVER TIMOTHY JEREMIAH**

**ST/CS/ND/21/114**

**SUBMITTED TO**

**THE DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF SCIENCE AND TECHNOLOGY**

**FEDERAL POLYTECHNIC MUBI**

**IN PARTIAL FULFILLMENT FOR THE AWARD OF NATIONAL DIPLOMA (ND) IN COMPUTER SCIENCE**

**JUNE, 2023**

**APPROVAL PAGE**

This is to certify that this report was compiled and written by Ajaver Timothy Jeremiah with Registration Number ST/CS/ND/21/114 from the department of Computer Science of the Federal Polytechnic Mubi, Adamawa State and was read and approved by

**TPL. KADALA SOLOMON BAUDE** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

School Supervisor Date

**MR. SIMON GALADIMA** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department SIWES Coordinator Date

**MALLAM ADAMU GARBA** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department Date

**DEDICATION**

I dedicate this report to the Almighty God for the guidance and protection accorded me throughout the programme.

**ABSTRACT**

*The report summarises the result of work done during my SIWES experience, the technical report consists of four chapters, which comprises of introduction, history and criteria’s of SIWES in chapter one followed by their aims and objectives, historical background, organizational structure of the organization in chapter two, while chapter three consist of the work actually carried out during the SIWES programme and lastly chapter four consist of the summary, conclusions and recommendation.*

**Table of Contents**

**Title page - - - - - - - - - - i**

**Approval page - - - - - - - - ii**

**Dedication - - - - - - - - - - iii**

**Abstract - - - - - - - - - - iv**

**CHAPTER ONE**

1.0 Introduction - - - - - - - - - 1

1.1 Background of SIWES - - - - - - - 1

1.2 Brief History of SIWES - - - - - - - 1

1.3 Aims and Objectives of SIWES - - - - - - 2

**CHAPTER TWO**

2.1 Brief History of the Organization. - - - - - - 3

2.2 The Objective of the Organization. - - - - - - 3

2.3 Organizational Chart - - - - - - - - 4

**CHAPTER THREE**

**EXPERIENCE ACQUIRED**

# 3.1 Introduction HTML - - - - - - - - 5

## 3.2 What is CSS - - - - - - - - - 11

3.3 Introduction to JavaScript - - - - - - - 15

**CHAPTER FOUR**

4.1 Summary - - - - - - - - - 16

4.2 Conclusion - - - - - - - - - 16

4.3 Problems - - - - - - - - - 17

4.4 Recommendations - - - - - - - - 18

**CHAPTER ONE**

**1.0 INTRODUCTION**

The concept behind industrial training schemes is to acquire practical, knowledge in addition to what has been learn institution. The four months mandatory supervised industrial attachment is program instituted by the federal government to help student for their different field of studies.

### 1.1 BACKGROUND OF SIWES

It aims at exposing student to the practical and the actualization of work situation which they may meet after graduation and the learning is meanly expected to produce graduate qualification to meet man power of employments of (ITF), there was growing concern amount the industrialist that graduate of higher learning do not passes adequate background student for the opinion that theoretical education going higher institution was responsive enough to meet the need of employers of the labour as a requirement of national board for technical education (NBTE) polytechnic training before obtaining the National Diploma (ND).

Experience scheme (SIWES), is a National programme introduced by Government in 1974 for student in tertiary institution

### 1.2 BRIEF HISTORY OF SIWES

**SIWES** was established by **ITF** in 1973 to solve the problem of lack of adequate practical skills preparatory for employment in industries by Nigerian graduates of tertiary institutions.

The program “Student industrial work experience scheme” is a pure skill-acquisition program structured for the tertiary institutions as a complementary and enhancement program to the theoretical education, laboratory and workshop practices engaged in by student in different higher institutions. The SIWES was initiated to improve the student’s technical abilities (performance/knowledge) and expose them to industrial culture thereby preparing them to be acquainted with the roles to play towards the technological advancement of the nation.

It is there for a practical aspect of the academic works, which the students may not be opportune to carry out throughout their stay in the higher institutions. Based on this fact, the Federal Government decided to establish a body (regulatory) which engages in the training of the technical manpower. The body is named industrial Training Fund (ITF). The government also undertook to make up for the deficiencies by structuring and established Students Industrial Training (SIT) as it was then called but in 1973 the SIWES was formed which is the subsidiary.

### 1.3 AIMS AND OBJECTIVES OF SIWES

1. One of the objectives of SIWES is to create an avenue for students to acquire industrial skills to complement their theoretical knowledge and improve their experiences in their course of study.
2. The program prepares the students for industrial working conditions prior to their graduation.
3. To enable the students to learn personal relationship with employers and co-employees on graduation.
4. It exposes the student to working methods and techniques in handling equipment and machinery.
5. It develops the student in make critical and realistic approaches to solving problem.
6. It serves as a way of improving the abilities of the students and to contribute to the growth of the nation.
7. It gives the students the opportunity to practice what they have been taught in various disciplines.
8. The SIWES program strengthens the employer’s involvement in preparing the students for employment.

# CHAPTER TWO

### 2.1 BRIEF HISTORY OF University of Mkar

The University of Mkar is a private university located in Gboko, Benue State, Nigeria. It was established in 2005 by the Universal Reformed Christian Church (URCC) as a Christian institution of higher learning. The university started as the Mkar Community College in 1986, offering basic education to students in the community. In 2005, the URCC decided to upgrade the college to a university, and the University of Mkar was born. The institution was officially licensed by the National Universities Commission (NUC) in 2005. Since its establishment, the University of Mkar has continued to grow and expand its academic programs, facilities, and student population. It currently offers undergraduate and graduate programs in various fields such as agriculture, business administration, computer science, education, environmental management, and theology. The university has also established several research and community development centers to promote research and outreach activities. These include the Center for Agribusiness and Sustainable Development, Center for Entrepreneurship Development, Center for Gender and Social Development, and Center for Christian Studies and Social Development.

**Mission Statement**

To provide a conducive and nurturing environment for the development of human capital through quality teaching, research, and community service, anchored on Christian values and principles, towards the transformation of society.

**Vision Statement**

To be a world-class Christian institution of higher learning, producing graduates who are intellectually, spiritually, and morally equipped for service to God and humanity.

**2.2 ORGANIZATIONAL CHART OF BRIATEK COMPUTERS**

Board of Directors

Managing Director/CEO

Director of Human Resources

Director of Institute

Director of research and development

Director operations

Clerical Analyst

Program/Institute coordinator

Program Analyst

System Analyst/Data

Driver

Instructors

Program Developers

Data processing officers

Security

IT/SIWES students

Instructors

Data processing Assistant

Figure 2.1: Organizational chart

**CHAPTER THREE**

**3.0 INTRODUCTION to HTML**

**3.1 What is HTML?**

* HTML is the standard markup language for creating Web pages.
* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
* Browsers do not display the HTML tags, but use them to render the content of the page

**HTML Tags**

HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

* HTML tags normally come in pairs like <p> and </p>
* The first tag in a pair is the start tag, the second tag is the end tag
* The end tag is written like the start tag, but with a forward slash inserted before the tag name

**WEB BROWSERS**

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:

**HTML VERSIONS**

Since the early days of the web, there have been many versions of HTML:

**Version Year**

HTML 1991

HTML 2.0 1995

HTML 3.2 1997

HTML 4.01 1999

XHTML 2000

HTML5 2014

**HTML EDITORS**

Web pages can be created and modified by using professional HTML editors.

* Notepad (Windows) or TextEdit (Mac).
* Sublime
* Notepad ++
* Atom, etc.

We believe using a simple text editor is a good way to learn HTML.

Follow the four steps below to create your first web page with Notepad or TextEdit.

**HTML PARAGRAPHS**

HTML paragraphs are defined with the <p> tag:

Example

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

**HTML Headings**

Headings are important in HTML documents.

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

»

**HTML Tip - How to View HTML Source**

Have you ever seen a Web page and wondered "Hey! How did they do that?"

To find out, right-click in the page and select "View Page Source" (in Chrome) or "View Source" (in IE), or similar in another browser. This will open a window containing the HTML code of the page.

**HTML Text Formatting Elements**

**Tag Description**

<b> Defines bold text

<em> Defines emphasized text

<i> Defines italic text

<small> Defines smaller text

<strong> Defines important text

<sub> Defines subscripted text

<sup> Defines superscripted text

<ins> Defines inserted text

<del> Defines deleted text

<mark> Defines marked/highlighted text

**HTML Comments**

Comment tags are used to insert comments in the HTML source code.

You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

With comments you can place notifications and reminders in your HTML:

Example

<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Remember to add more information here -->

»

**HTML Colors**

In HTML, a color can be specified by using a color name, an RGB value, or a HEX value.

**Color Names**

In HTML, a color can be specified by using a color name:

Example

**Color Name**

Red

Orange

Green

Black

HTML supports 140 standard color names.

**RGB Value**

rgb(255,0,0)

rgb(255,255,0)

»

**HEX Value**

In HTML, a color can also be specified using a hexadecimal value in the form: #RRGGBB, where RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and FF (same as decimal 0-255). For example, #FF0000 is displayed as red, because red is set to its highest value (FF) and the others are set to the lowest value (00).

Example

**HEX Value**

#FF0000

#FFFF00

#CCCCCC

#FFFFFF

**HTML Links**

Links are found in nearly all web pages. Links allow users to click their way from page to page.

**HTML Links - Hyperlinks**

HTML links are hyperlinks. You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

**Note**: A link does not have to be text. It can be an image or any other HTML element.

**HTML Images**

JPG Images

GIF Images

PNG Images

Example

<!DOCTYPE html>

<html>

<body>

<h2>Spectacular Mountain</h2>

<img src="pic\_mountain.jpg" alt="Mountain View" style="width:304px;height:228px; ">

</body>

</html>

»

**HTML Images Syntax**

In HTML, images are defined with the <img> tag. The <img> tag is empty, it contains attributes only, and does not have a closing tag. The src attribute specifies the URL (web address) of the image:

<img src="url" alt="some\_text" style="width:width;height:height;">

If a browser cannot find an image, it will display the value of the alt attribute:

Example

<img src="wrongname.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

»

The alt attribute is required. A web page will not validate correctly without it.

**HTML Tables**

An HTML table is defined with the <table> tag.

Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

Example

<table style="width:100%">

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Age</th>

</tr>

<tr>

<td>Jill</td>

<td>Smith</td>

<td>50</td>

</tr>

<tr>

<td>Eve</td>

<td>Jackson</td>

<td>94</td>

</tr>

</table>

**HTML Lists**

HTML List Example

An Unordered List:

1. Item
2. Item
3. Item
4. Item

An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

**Unordered HTML List**

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

Example

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

**Ordered HTML List**

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

Example

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

**Ordered HTML List - The Type Attribute**

The type attribute of the <ol> tag, defines the type of the list item marker:

**Type Description**

type="1" The list items will be numbered with numbers (default)

type="A" The list items will be numbered with uppercase letters

type="a" The list items will be numbered with lowercase letters

type="I" The list items will be numbered with uppercase roman numbers

type="i" The list items will be numbered with lowercase roman numbers

## **3.2 What is CSS?**

**CSS** stands for **C**ascading **S**tyle **S**heets. CSS describes **how HTML elements are to be displayed on screen, paper, or in other media.** 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.** CSS is a language that describes the style of an HTML document.

### CSS Example

body {  
    background-color: lightblue;}  
 h1 {  
    color: white;  
    text-align: center;}  
p {  
    font-family: verdana;  
    font-size: 20px;}

**Note:** that CSS code is simply written instructions that tells Web browsers (like FireFox and Internet Explorer) how to display things on a page. For example:

1. make text bold.
2. position things a page.
3. set the font style for a page or paragraph etc.

# Applying CSS

There are three ways to apply CSS to HTML: **Inline**, **internal**, and **external**.

## **Inline**

Inline styles are plonked straight into the HTML tags using the style attribute.

They look something like this:

<p style="color: red">text</p>

This will make that specific paragraph red.

But, if you remember, the best-practice approach is that the HTML should be a stand-alone, **presentation free** document, and so in-line styles should be avoided wherever possible.

## **Internal**

Embedded, or internal, styles are used for the whole page. Inside the [head](http://www.htmldog.com/references/html/tags/head/) element, the [style](http://www.htmldog.com/references/html/tags/style/) tags surround all of the styles for the page.

<!DOCTYPE html>

<html>

<head>

<title>CSS Example</title>

**<style>**

**p {**

**color: red;**

**}**

**a {**

**color: blue;**

**}**

**</style>**

This will make all of the paragraphs in the page red and all of the links blue.

Although preferable to soiling our HTML with inline presentation, it is similarly usually preferable to keep the HTML and the CSS files separate, and so we are left with our saviour.

## **External**

External styles are used for the whole, multiple-page website. There is a **separate CSS file**, which will simply look something like:

p {

color: red;

}

a {

color: blue;

}

If this file is saved as “style.css” in the same directory as your HTML page then it can be linked to in the HTML like this:

<!DOCTYPE html>

<html>

<head>

<title>CSS Example</title>

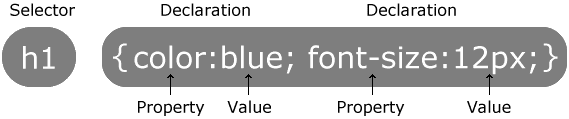
<link rel="stylesheet" href="style.css">

...

# CSS Syntax and Selectors

## **CSS Syntax**

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

## **Lengths and Percentages**

There are many property-specific units for values used in CSS, but there are some general units that are used by a number of properties and it is worth familiarizing yourself with these before continuing.

* **px** (such as font-size: 12px) is the unit for pixels.
* **em** (such as font-size: 2em) is the unit for the calculated size of a font. So “2em”, for example, is two times the current font size.
* **pt** (such as font-size: 12pt) is the unit for points, for measurements typically in printed media.
* **%** (such as width: 80%) is the unit for… wait for it… percentages.

Other units include picas, centimeters, millimeters and inches.

## **CSS Selectors**

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

## **The Element Selector**

The element selector selects elements based on the element name. You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

### Example

p {  
    text-align: center;  
    color: red;  
}

## **The 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 a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

### Example

#para1 {  
    text-align: center;  
    color: red;  
}

## **The Class Selector**

The class selector selects elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and center-aligned:

### Example

.center {  
    text-align: center;  
    color: red;  
}

## **CSS Comments**

Comments are used to explain the code, and may help when you edit the source code at a later date. A CSS comment starts with /\* and ends with \*/. Comments can also span multiple lines:

### Example

p {  
    color: red;  
    /\* This is a single-line comment \*/  
    text-align: center;  
}

**3.3 INTRODUCTION TO JAVASCRIPT**

**3.3.1 WHAT IS JAVASCRIPT**

JavaScript is a programming language, and like most programming languages, it has some basic constructs that we’ll look at. A program in JavaScript is like a sequence of steps. Similar to how we give directions to a stranger, a computer needs detailed instructions, defined as steps, to accomplish any simple or complex action.

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [Client-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [Server-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. Javascript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like [Array](https://www.geeksforgeeks.org/arrays-in-javascript/), [Date](https://www.geeksforgeeks.org/javascript-date-objects/), and [Math](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like operators, control structures, and statements.

1. **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**. Useful libraries for the client-side are [**AngularJS**](https://www.geeksforgeeks.org/introduction-to-angularjs/), [**ReactJS**](https://www.geeksforgeeks.org/react-js-introduction-working/), **VueJS** and so many others.
2. **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is [**node.js**](https://www.geeksforgeeks.org/introduction-to-nodejs/).
3. **Imperative language –**In this type of language we are mostly concern about how it is to be done . It simply control the flow of computation . The procedural programming approach , object, oriented approach comes under this like async await we are thinking what it is to be done further after async call.
4. **Declarative programming –**In this type of language we are concern about how it is to be done , basically here logical computation require . Here  main goal is to describe the desired result without direct dictation on how to get it like  arrow function do .

**3.3.2 HOW TO INSERT JAVASCRIPT**

JavaScript can be added to your HTML file in [two ways](https://www.geeksforgeeks.org/where-to-put-javascript-in-an-html-document/):

1. **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
2. **External JS:** We can write JavaScript code in other file having an extension.js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

**Syntax:**

<script>

// JavaScript Code

</script>

**Example:**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">    <head>      <title>          Basic Example to Describe JavaScript      </title>  </head>    <body>        <!-- JavaScript code can be embedded inside          head section or body section -->      <script>          console.log("Welcome to JavaScript Programming");      </script>  </body>    </html> |

**Output:** The output will display on the console.

Welcome to JavaScript Programming

**3.3.3 Applications of JavaScript**

**Web Development:** Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.

1. **Web Applications:** With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React is helpful in this department.
2. **Server Applications:** With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful on the server-side.
3. **Games:** Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.
4. **Smartwatches:** JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the internet for its functioning.
5. **Art:** Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, and make the sound more effective also can be used [**p5.js**](https://www.geeksforgeeks.org/p5-js-introduction/) library.
6. **Machine Learning:** This JavaScript ml5.js library can be used in web development by using machine learning.
7. **Mobile Applications:**JavaScript can also be used to build an application for non-web contexts. The features and uses of JavaScript make it a powerful tool for creating mobile applications. This is a Framework for building web and mobile apps using JavaScript. Using React Native, we can build mobile applications for different operating systems. We do not require to write code for different systems. Write once use it anywhere!

JavaScript is a dynamically typed (also called loosely typed) scripting language. That is, in JavaScript variables can receive different data types over time. Datatypes are basically typed data that can be used and manipulated in a program.

**The latest ECMAScript(ES6) standard defines following data types**: Out of which six data types are Primitive(predefined).

1. **Numbers**: Represent both integer and floating-point numbers. Example: 5, 6.5, 7 etc.
2. **String**: A string is a sequence of characters. In JavaScript, strings can be enclosed within the single or double quotes. Example: “Hello GeeksforGeeks” etc.
3. **Boolean**: Represent a logical entity and can have two values: true or false.
4. **Null**: This type has only one value : *null.*
5. **Undefined**: A variable that has not been assigned a value is *undefined.*
6. **Symbol:**Unlike other primitive data types, it does not have any literal form. It is a built-in object whose constructor returns a symbol-that is unique.
7. **bigint:**The bigint type represents the whole numbers that are larger than 253-1. To form a bigint literal number, you append the letter n at the end of the number.
8. **Object**: It is the most important data-type and forms the building blocks for modern JavaScript. We will learn about these data types in detail in further articles.

**Variables in JavaScript**

Variables in JavaScript are containers that hold reusable data. It is the basic unit of storage in a program.

1. The value stored in a variable can be changed during program execution.
2. A variable is only a name given to a memory location, all the operations done on the variable effects that memory location.
3. In JavaScript, all the variables must be declared before they can be used.

**Before ES2015**, JavaScript variables were solely declared using the *var* keyword followed by the name of the variable and semi-colon. Below is the syntax to create variables in JavaScript:

*var* var\_name;

*var x;*

The var\_name is the name of the variable which should be defined by the user and should be unique. These types of names are also known as **identifiers**. The rules for creating an identifier in JavaScript are, the name of the identifier should not be any pre-defined word(known as keywords), the first character must be a letter, an underscore (\_), or a dollar sign ($). Subsequent characters may be any letter or digit or an underscore or dollar sign. Notice in the above code sample, we didn’t assign any values to the variables. We are only saying they exist. If you were to look at the value of each variable in the above code sample, it would be *undefined.*

We can initialize the variables either at the time of declaration or also later when we want to use them. Below are some examples of declaring and initializing variables in JavaScript:

*// declaring single variable*

**var name;**

*// declaring multiple variables*

**var name, title, num;**

*// initializing variables*

**var name = "Harsh";**

**name = "Rakesh";**

JavaScript is also known as **untyped** language. This means, that once a variable is created in JavaScript using the keyword var, we can store any type of value in this variable supported by JavaScript. Below is the example for this:

*// creating variable to store a number*

**var num = 5;**

*// store string in the variable num*

**num = "Javascript";**

The above example executes well without any error in JavaScript, unlike other programming languages.   
Variables in JavaScript can also evaluate simple mathematical expressions and assume their value.

// storing a mathematical expression

var x = 5 + 10 + 1;

console.log(x); // 16

**After ES2015**, we now have two new variable containers: let and const. Now we shall look at both of them one by one. The variable type **Let** shares lots of similarities with var but unlike var, it has scope constraints. To know more about them visit [let vs var](https://www.geeksforgeeks.org/difference-between-var-and-let-in-javascript/). Let’s make use of the let variable:

*// let variable*

let x; // undefined

let name = 'Mukul';

*// can also declare multiple values*

let a=1,b=2,c=3;

*// assignment*

let a = 3;

a = 4; // works same as var.

**Const** is another variable type assigned to data whose value cannot and will not change throughout the script.

const name = 'Mukul';

name = 'Mayank'; // will give Assignment to constant variable error.

**Variable Scope in Javascript**

Scope of a variable is the part of the program from where the variable may directly be accessible.   
In JavaScript, there are two types of scopes:

1. **Global Scope** – Scope outside the outermost function attached to Window.
2. **Local Scope** – Inside the function being executed.

Let’s look at the code below. We have a global variable defined in the first line in the global scope. Then we have a local variable defined inside the function fun().

|  |
| --- |
| let globalVar = "This is a global variable";    function fun() {    let localVar = "This is a local variable";      console.log(globalVar);    console.log(localVar);  }    fun(); |

**CHAPTER FOUR**

**SUMMARY, CONCLUSION, PROBLEMS AND RECOMMENDATION**

# 4.1 SUMMARY

The relevance of the SIWES program cannot be over emphasized considering the fact that it has significantly reduced the gap between my theoretical and practical knowledge about computer hardware and software, installations, maintenance and networking. The processes of communication which include data and telecommunication the use of switch in the networking and what networking is all about.

The program is indeed a commendable one in that it affords students ample opportunities of being exposed to good working relationship with colleagues and the field experience with customers. This little exposure has widened my knowledge about my course of study, not only that it has automatically changed my views about lift in general. The firm at large has taught me how to be independent of my own how to be conscious of my health and safety at its peak relating to the environment where I carried out my SIWES program. It was indeed a highly rewarding experience to be with University of Mkar, Gboko, Benue State.

**4.2 CONCLUSION**

In conclusion, I thank ITF in general for their effort towards the Student Industrial Training Scheme. The contribution that the industrial training offered to student will not be over emphasized. It has exposed me seriously to a certain depth and length of practical capability on Web Designs.

It has also acquainted me with the working condition, which I am expected to encounter in the near future. I will say that SIWES has a greater advantage on me, it has greatly exposed me to the practical application of all that I have been through in the school, SIWES is an experience that all student must pass through this is because it gives a full practical knowledge of what has been through in classroom.

Finally, I have a strong believe that this comprehensive based on the experience, I acquired during the industrial training scheme will convince every user training is not difficult.

I therefore strong conclude that the continuous existence of SIWES programme as it is very necessary since it plays a dominant role in the development of student of Computer Science in the acquisition of practical experience.

# 4.3 PROBLEMS OBSERVED DURING MY PROGRAM

1. The time frame set for the program is too short as some of the aspects of the program where not completed.
2. Lack of Financial support from the company to aid transportation to and from training.
3. Attentions are not given to the IT students by the workers it is learn if you want to learn or ask if you want to know.
4. Cost of Training: The Student has to be registered as a student of a particular organization in order to carry out the program.
5. Lack of Visits to the various places of Attachment by the ITF Officials to ensure that the student is actually carrying out the program.

**4.4 RECOMMENDATIONS**

Below are the recommendations that should be given serious consideration so as alleviate the suffering of students undergoing SIWES.

Firstly, the Federal Government of Nigerian should make a positive effort in reducing the overall cost of production so that companies should be producing to fill capacity and accommodate SIWES populaces. Also, certain monthly allowance may be given to the student by company accepted then (student to ease transportation problem).

Secondly, the Industrial Training Fund (ITF), should try and increase the money paid at the end of SIWES to the student so as to justify the Cost of Living we experienced.

Thirdly, the period set for the SIWES should be increase to six mouths so as to enable the student involved and gain enough experience since it is widely believed that experience is the best teacher. Again, this period should also be in line with Nigerian University Commission curriculum for the university undergraduate student on industrial attachment.

Finally, the ITF official should please continue visiting the students, to ensure that what they are learning is in line with the ITF requirement.

**REFERENCES**

Edward, B. M.D (2003). The Effect of SIWES on a student, Federal Ministry of Works, Lagos. Nigeria.

John, Doe (2008). Introduction to web development: responsive designs, McGraw Print New York.