

CHAPTER ONE

INTRODUCTION

1.1 Background of SIWES

The Student Industrial Work Experience Scheme (SIWES) is a skills training programme established to expose and prepare students of universities and other tertiary institutions for the industrial work situation they are likely to encounter after graduation. SIWES was introduced in Nigeria by the Industrial Training Fund (ITF) in 1973 and has since become a key component of Nigerian higher education, particularly for students in science, engineering, and technology disciplines.

The scheme provides Nigerian students with an opportunity to apply their academic knowledge to real-world problems and to acquire practical experience that is not available within the classroom. At the end of a successful training period, students are expected to have developed their technical skills, improved their professional discipline, and gained a better understanding of the work environment.

1.2 Objectives of SIWES

The SIWES programme is designed with the following key objectives:

1. To provide students with an opportunity to apply knowledge and skills acquired in the classroom to real industrial problems.
2. To expose students to work methods and techniques in handling equipment and tools that may not be available in their institutions.
3. To make the transition from school to the world of work easier, thereby enhancing students' contacts for job placement after graduation.
4. To enlist and strengthen employers' involvement in the educational process and prepare students for employment in industry.
5. To develop in students the necessary work habits, resilience, and professional attitude required for the field.

1.3 Scope of Training

This report covers the industrial training period from [Start Date] to [End Date], undertaken at StoneTech Square. The training encompassed three major technology areas: Microsoft Excel for data analysis and management, Power BI for business intelligence and

interactive visualisation, and Python programming for data analysis, automation, web development, and artificial intelligence applications.

CHAPTER TWO

OVERVIEW OF THE ORGANISATION

2.1 History of the Company

Stonetech is a technology organisation based in Nigeria, working across Africa to help people and businesses build something that lasts. We started as an innovation hub in Kaduna in 2023 and have since grown into an organisation that supports founders, creatives, learners, and businesses at every stage of what they are building.

Our work sits across four areas. We run training programmes that give people practical digital skills they can use immediately. We support creative professionals who want to turn their work into sustainable careers. We help early stage startups go from an idea to a funded, functioning business. And we work with communities and institutions to put technology in the hands of people who need it most.

We believe Africa's best solutions will come from within Africa itself. Everything we do our hub, our venture studio, our business services, and our partnerships is built around that belief.

CHAPTER THREE

TRAINING ACTIVITIES AND EXPERIENCE

3.1 Microsoft Excel

3.1.1 Introduction to Excel

Microsoft Excel is a powerful spreadsheet application developed by Microsoft Corporation and is part of the Microsoft Office Suite. It is widely used in businesses, educational institutions, and research environments for data entry, organisation, analysis, and visualisation. During the training, I was introduced to the Excel interface, including the Ribbon, cells, rows, columns, worksheets, and workbooks. I learned how to navigate the spreadsheet environment efficiently and understood the purpose of key Excel components such as the formula bar, Name Box, and status bar.

Excel operates on a grid of cells identified by column letters and row numbers (e.g., A1, B3). Each cell can hold different types of data including text, numbers, dates, or formulas. Understanding this structure formed the foundation for all subsequent Excel training activities.

3.1.2 Data Analysis in Excel

Data analysis is one of the most powerful capabilities of Microsoft Excel. During the training, I learned how to use Excel to clean, organise, and analyse datasets of varying sizes. Key skills acquired include:

- Using built-in functions such as SUM, AVERAGE, MIN, MAX, COUNT, COUNTIF, SUMIF, and VLOOKUP for calculations and data retrieval.
- Applying sorting and filtering to organise data and identify trends.
- Using conditional formatting to highlight key data points and anomalies.
- Creating charts and graphs including bar charts, pie charts, and line graphs to represent data visually.

Data analysis in Excel helped me understand how raw data could be transformed into meaningful insights that support decision-making in professional environments.

3.1.3 Nominal and Ordinal Data

As part of the data analysis training, I was introduced to the concept of data types, specifically nominal and ordinal data. Nominal data refers to categorical data that has no intrinsic order or ranking, such as names, colours, or gender. Examples encountered during training included employee departments, product categories, and location names.

Ordinal data, on the other hand, refers to categorical data that has a meaningful order or ranking, but where the differences between ranks are not necessarily equal. Examples include survey satisfaction ratings (e.g., Poor, Fair, Good, Excellent) and academic grade classifications. Understanding the distinction between these data types was important for choosing the correct Excel functions and chart types for analysis.

3.1.4 Pivot Tables

A Pivot Table is one of the most powerful data analysis tools available in Microsoft Excel. It allows users to summarise, reorganise, and aggregate large datasets quickly and interactively without altering the original data. During the training, I received detailed instruction on the concept, purpose, and practical use of pivot tables.

3.1.5 Understanding Pivot Tables

A pivot table works by allowing users to drag and drop data fields into four areas: Rows, Columns, Values, and Filters. The "Rows" area determines how data is grouped vertically, the "Columns" area creates horizontal groupings, the "Values" area defines the aggregation method (such as SUM, COUNT, or AVERAGE), and the "Filters" area allows users to narrow down the dataset being displayed. This flexible structure makes pivot tables ideal for exploring large datasets and producing summary reports quickly.

3.1.6 Steps on How to Create a Pivot Table

The following steps were followed to create pivot tables during training:

1. Select any cell within the dataset to be analysed.
2. Click on the "Insert" tab in the Excel Ribbon.
3. Select "PivotTable" from the Tables group.
4. In the dialog box that appears, confirm the data range and choose whether to place the pivot table in a new worksheet or an existing one, then click OK.
5. In the PivotTable Fields pane that appears on the right, drag the desired fields into the Rows, Columns, Values, and Filters areas as required.

6. Customise the pivot table by right-clicking on values to change the aggregation method, applying number formatting, or adding slicers for interactive filtering.
7. Refresh the pivot table whenever the source data is updated by right-clicking and selecting "Refresh".

3.1.7 Types of Data Analysis

During the training, I was introduced to the following types of data analysis commonly performed in Excel:

- **Descriptive Analysis:** Summarising and describing the main features of a dataset using measures such as mean, median, mode, and standard deviation.
- **Comparative Analysis:** Comparing datasets across different categories, time periods, or groups to identify differences and patterns.
- **Trend Analysis:** Using line charts and moving averages to identify patterns over time.
- **What-If Analysis:** Using Excel tools such as Goal Seek and Data Tables to explore how changes in input variables affect outcomes.
- **Statistical Analysis:** Using Excel's built-in statistical functions and the Analysis ToolPak add-in for more advanced computations.

3.2 Power BI

3.2.1 Introduction to Power BI

Microsoft Power BI is a business analytics and data visualisation tool developed by Microsoft. It enables users to connect to a wide variety of data sources, transform raw data into structured formats, and create interactive dashboards and reports that support informed business decisions. Power BI is available in several versions including Power BI Desktop (a free downloadable application), Power BI Service (a cloud-based platform), and Power BI Mobile. During the training, I primarily worked with Power BI Desktop.

The Power BI interface consists of three main views: Report View (for building visualisations), Data View (for examining loaded tables), and Model View (for defining relationships between tables). Understanding this environment was the starting point of all Power BI training activities.

3.2.2 Types of Power BI

The different versions and components of Power BI encountered during training include:

- Power BI Desktop: A free application installed on a local computer, used to build reports and models before publishing them.
- Power BI Service: A cloud-based platform (app.powerbi.com) where reports are published, shared, and collaborated on.
- Power BI Mobile: A mobile application available on Android and iOS that allows users to view and interact with published dashboards on mobile devices.
- Power BI Report Server: An on-premises solution for organisations that need to host reports within their own infrastructure.
- Power BI Embedded: An Azure service that allows developers to embed Power BI reports and dashboards into custom applications.

3.2.3 Introduction to Power Query

Power Query is the data transformation and preparation engine built into both Microsoft Excel and Power BI. It provides a user-friendly interface for connecting to data sources, cleaning messy data, reshaping tables, and combining datasets before loading them into the data model for analysis. Power Query uses the M language in the background, but most transformations can be performed through its graphical interface without writing code.

During training, I used Power Query in Power BI Desktop to connect to Excel files and CSV files, remove duplicate rows, rename columns, change data types, filter unnecessary

rows, and merge multiple tables into a unified dataset. The result of each transformation step is recorded in the "Applied Steps" pane, making the process transparent and reversible.

3.2.4 Steps on How to Create Tables and Charts Using Power Query

The following steps were followed when creating tables and charts using Power Query in Power BI:

8. Open Power BI Desktop and click "Get Data" from the Home tab.
9. Select the appropriate data source (e.g., Excel, CSV, or SQL Server) and connect to the file or server.
10. In the Power Query Editor that opens, apply the necessary transformations such as removing blank rows, changing data types, and renaming columns.
11. Click "Close & Apply" to load the transformed data into the Power BI data model.
12. In the Report View, select a visualisation type from the Visualisations pane (e.g., table, bar chart, pie chart, or line chart).
13. Drag the relevant fields from the Fields pane into the visual's field wells (Axis, Values, Legend, etc.).
14. Format the visual using the Format pane to adjust colours, labels, titles, and borders.
15. Add slicers and filters to make the report interactive.

3.2.5 Power Pivot

Power Pivot is an advanced data modelling feature available in both Microsoft Excel and Power BI. It allows users to work with large datasets from multiple sources and create sophisticated relationships between tables, similar to a relational database. Power Pivot uses the DAX (Data Analysis Expressions) language for creating calculated columns and measures.

During training, I learned how to define relationships between tables in the Model View of Power BI by dragging fields to connect primary and foreign keys. I also wrote basic DAX formulas including calculated columns (e.g., Total Revenue = Quantity * Unit Price) and measures (e.g., Total Sales = SUM(Sales[Revenue])) to enrich the data model.

3.2.6 Pivot Table in Power BI

Similar to Excel, Power BI supports the equivalent of pivot table functionality through its Matrix visual. The Matrix visual allows users to display data in a cross-tabulated format with rows, columns, and values, making it ideal for comparing performance across multiple

dimensions simultaneously. During training, I used the Matrix visual to create summary tables comparing data across different categories and time periods.

3.2.7 Pivot Chart Steps in Power BI

The following steps were used to create pivot-style charts in Power BI:

16. Load the data into Power BI using Power Query and apply all necessary transformations.
17. In the Report View, select a chart visual from the Visualisations pane (e.g., Clustered Bar Chart or Stacked Column Chart).
18. Drag the categorical field (e.g., Product Category) to the Axis field well.
19. Drag the numeric field (e.g., Sales Amount) to the Values field well.
20. Drag an additional field (e.g., Region) to the Legend field well if a comparative breakdown is required.
21. Add a slicer visual by selecting the Slicer from the Visualisations pane and dragging a filter field (e.g., Year) into it.
22. Adjust chart formatting, colours, and titles to finalise the visual.

3.3 Python

3.3.1 Introduction to Python

Python is a high-level, general-purpose programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python has become one of the most widely used programming languages in the world, with applications spanning data analysis, machine learning, web development, automation, game development, and artificial intelligence. During the training, I was introduced to Python's syntax, fundamental programming concepts, and several of its key application areas.

Python uses an interpreter-based execution model, meaning code is executed line by line. The Python development environment used during training was Python IDLE and Visual Studio Code. Python scripts are saved with the ".py" file extension.

3.3.2 Python Syntax

Python syntax refers to the set of rules that define how Python code is written and structured. Key syntax rules encountered during training include:

- Python uses indentation (whitespace) to define code blocks rather than curly brackets. This makes the code more readable and enforces clean formatting.
- Comments are written using the hash symbol (#) and are ignored during execution.
- Statements do not require a semicolon at the end, unlike C or Java.
- Variable names are case-sensitive (e.g., Name and name are different variables).
- Python uses the colon (:) to begin code blocks such as functions, loops, and conditionals.

3.3.3 Data Types in Python

Python supports several built-in data types. The major ones encountered during training are:

- Integer (int): Whole numbers without decimal points, e.g., 5, 100, -20.
- Float (float): Numbers with decimal points, e.g., 3.14, 2.718.
- String (str): A sequence of characters enclosed in single or double quotes, e.g., "Hello", 'Python'.
- Boolean (bool): Represents one of two values — True or False.
- List: An ordered, mutable collection of items, e.g., [1, 2, 3, "apple"].

- Tuple: An ordered, immutable collection of items (covered in detail in section 3.3.5).
- Dictionary (dict): An unordered collection of key-value pairs, e.g., {"name": "Haruna", "age": 20}.
- Set: An unordered collection of unique items, e.g., {1, 2, 3}.

3.3.4 Simple Program to Print Hello World

The classic first program in any programming language is the "Hello World" program. In Python, this is written using the `print()` function, which outputs text to the console. The programme below was the very first Python code I executed during training:

```
# My First Python Programme
print("Hello, World!")
```

When this programme is executed, it produces the output: Hello, World! This simple exercise demonstrated how Python code is written and run, and introduced the `print()` function which is used extensively for displaying output.

3.3.5 Tuples in Python

A tuple is an ordered, immutable sequence of elements in Python. Like lists, tuples can store multiple items of different data types, but unlike lists, tuples cannot be modified after they are created (i.e., elements cannot be added, removed, or changed). Tuples are defined using round brackets `()`, while lists use square brackets `[]`.

Example of a tuple:

```
my_tuple = ("Haruna", 20, "Computer Engineering", True)
```

3.3.6 Different Data Types in a Tuple

Tuples can hold elements of different data types within a single tuple. The types that can appear in a tuple include:

- String values: e.g., "Kaduna", "COE24ND0053"
- Integer values: e.g., 2024, 100
- Float values: e.g., 3.75, 99.5
- Boolean values: e.g., True, False
- Nested tuples: e.g., (1, (2, 3), "hello")

Tuples are used when data should remain constant throughout a programme's execution, such as storing coordinates, database records, or configuration settings.

3.3.7 Basic Concepts of Python

The following fundamental Python programming concepts were covered during training:

- Variables: Named storage locations used to hold data values. e.g., `name = "Haruna"`
- Operators: Symbols used to perform operations on values, including arithmetic (+, -, *, /), comparison (==, !=, >, <), and logical (and, or, not) operators.
- Conditional Statements: `if`, `elif`, and `else` statements are used to execute code blocks based on conditions.
- Loops: `for` loops iterate over sequences (e.g., lists, strings), while `while` loops repeat a block of code as long as a condition is `True`.
- Input and Output: The `input()` function receives user input, and `print()` displays output.
- Indentation: Code blocks in Python are defined by consistent indentation (typically 4 spaces).

3.3.8 Uses of Python

Python is one of the most versatile programming languages available. The major application areas covered during training include:

a) Web Development

Python is widely used for building websites and web applications using frameworks such as Django and Flask. Django is a high-level web framework that follows the Model-View-Template (MVT) architecture, enabling rapid development of secure and scalable websites. Flask is a lightweight framework suitable for smaller applications and APIs. During training, the concept of how Python handles server-side logic, URL routing, and database connectivity in web applications was discussed.

b) Data Analysis

Python is the dominant language for data analysis, supported by powerful libraries such as Pandas (for data manipulation), NumPy (for numerical computing), and Matplotlib/Seaborn (for data visualisation). During the training, I was introduced to how Pandas DataFrames are used to load, clean, and analyse tabular data, mirroring many of the operations performed in Microsoft Excel but with greater automation capability.

c) Machine Learning

Python is the primary language used for machine learning, with libraries such as Scikit-learn, TensorFlow, and PyTorch. Machine learning involves training algorithms on

historical data so they can make predictions or decisions on new data. During training, the conceptual foundation of supervised learning (classification and regression) and unsupervised learning (clustering) was explained, highlighting Python's role as the standard tool in this field.

d) Artificial Intelligence

Artificial Intelligence (AI) involves creating computer systems that can perform tasks that normally require human intelligence, such as understanding natural language, recognising images, and making decisions. Python serves as the backbone of AI development due to its rich ecosystem of AI libraries and frameworks including TensorFlow, Keras, and OpenCV. The training highlighted how AI is increasingly integrated into everyday applications such as voice assistants, recommendation systems, and fraud detection.

e) Automation

Python is extensively used to automate repetitive tasks, saving time and reducing human error. Common automation use cases include file management (renaming, moving, and organizing files), web scraping (extracting data from websites using libraries like BeautifulSoup and Selenium), scheduled tasks, and automated report generation. During training, simple Python scripts were written to automate file operations and demonstrate the potential of automation in a professional setting.

f) Game Development

Python can be used for developing 2D games using the Pygame library. While Python is not the most common language for large commercial game development, it is widely used for prototyping, educational games, and small indie games. The training introduced the concept of Pygame and how Python manages game loops, event handling, and graphics rendering.

3.3.9 Functions in Python

A function in Python is a reusable block of code that performs a specific task. Functions are defined using the `def` keyword, followed by the function name, a pair of parentheses (which may contain parameters), and a colon. The function body is indented. Functions help to organise code, avoid repetition, and make programmes easier to read and maintain.

The general syntax for defining a function is:

```
def function_name(parameters):  
    # function body  
    return value
```

The following function examples were practised during training:

Example 1 — A function that greets a user:

```
def greet(name):  
    print("Hello, " + name + "!")  
  
greet("Haruna") # Output: Hello, Haruna!
```

Example 2 — A function that adds two numbers:

```
def add_numbers(a, b):  
    result = a + b  
    return result  
  
print(add_numbers(10, 5)) # Output: 15
```

Example 3 — A function that checks if a number is even or odd:

```
def check_even_odd(number):  
    if number % 2 == 0:  
        return "Even"  
    else:  
        return "Odd"  
  
print(check_even_odd(7)) # Output: Odd
```

These examples demonstrated how functions improve code reusability and modularity, core principles of professional software development.

CHAPTER FOUR

PROBLEMS ENCOUNTERED

Like any real-world work experience, the training was not without its challenges. The following are the major difficulties encountered during the SIWES period:

4.1 Limited Prior Knowledge in Some Areas

Certain topics encountered during the training, particularly Python programming and Power BI DAX formulas, were relatively new to me at the commencement of training, as they had not yet been fully covered in my academic curriculum. This initial knowledge gap meant that I required additional reading and personal study outside of working hours to keep pace with the technical demands of the training.

4.2 Time Management

Balancing structured training activities, personal study, and SIWES logbook updates was initially challenging. With guidance from my industrial supervisor, Ruqayyah Umar Salisu, I developed a more structured daily schedule that improved my time management and productivity considerably as the training progressed.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The Student Industrial Work Experience Scheme undertaken at StoneTech Square was an exceptionally rewarding and enriching experience. The training provided a practical bridge between the theoretical knowledge gained in the classroom and the actual demands of a modern data-driven work environment.

Through the various activities carried out including comprehensive Microsoft Excel training for data analysis and pivot tables, Power BI for business intelligence and interactive reporting, and Python for programming, data analysis, machine learning, web development, and automation I was able to significantly expand both my technical competence and my professional disposition. The challenges encountered further strengthened my problem-solving abilities and resilience, qualities that are essential for a successful career in computer engineering and technology.

It is evident that SIWES plays a vital and irreplaceable role in the training of engineering students in Nigeria. The scheme equips graduates with the practical skills and workplace awareness that formal academic instruction alone cannot provide, making them more competitive, confident, and work-ready upon graduation.

5.2 Recommendation

Based on my experience during the training, the following recommendations are made:

1. Organisations hosting SIWES students should ensure adequate computing resources and software licences are available to allow all trainees to practise simultaneously and develop their skills without unnecessary delays.
2. Kaduna Polytechnic should update the academic curriculum to introduce foundational programming (Python) and data analysis tools (Excel and Power BI) at the ND1 level, so that students are better prepared and can make more productive use of their SIWES period.
3. Students should be encouraged to begin personal study of data tools and programming languages before their SIWES period commences, as prior exposure significantly improves performance during training.

4. The Industrial Training Fund (ITF) and polytechnic management should ensure that SIWES stipends are disbursed promptly to enable students to meet their transportation and logistical needs throughout the training period.
5. Future SIWES students should maintain detailed logbooks throughout the training and document practical exercises, as this reinforces learning and makes report writing considerably easier at the end of the programme.

REFERENCES

1. Industrial Training Fund (ITF). (2023). *SIWES Handbook and Guidelines for Students and Supervisors*. Industrial Training Fund, Jos, Nigeria.
2. McKinney, W. (2022). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media.
3. Microsoft Corporation. (2024). *Microsoft Power BI Documentation*. Retrieved from <https://docs.microsoft.com/en-us/power-bi/>
4. Microsoft Corporation. (2024). *Microsoft Excel Help and Learning*. Retrieved from <https://support.microsoft.com/en-us/excel>
5. Python Software Foundation. (2024). *Python 3 Documentation*. Retrieved from <https://docs.python.org/3/>
6. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
7. Kaduna Polytechnic Academic Board. (2022). *National Diploma Computer Engineering Curriculum*. Kaduna Polytechnic Press.