**Logo

Description automatically generated with medium confidence**

**STUDENTS’ WORK EXPERIENCE PROGRAM**

**(SWEP)**

**AT**

**xPlug Technologies Limited.**

**Surulere.**

**Lagos.**

**PREPARED BY:**

**Ugochukwu Chukwuebuka Chidiebere**

**22/ENG02/074**

**SUBMITTED TO:**

**THE COLLEGE OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**AFE BABALOLA UNIVERSITY, ADO-EKITI**

**EKITI STATE.**

**DATE**

## CERTIFICATION

This is to certify that this work titled Students’ Work Experience Program (SWEP) was undertaken at xPlug Technologies Limited by **Ugochukwu Chukwuebuka Chidiebere**

**(22/ENG02/074)** and presented to the Department of Electrical and Electronics Engineering, Afe Babalola University, Ado-Ekiti, Ekiti State during the 2024/2025 Students’ Work Experience Program (SWEP).

--------------------------------- Signature and Date

SIWES Coordinator

Engr. Dr. Femi Onibonoje. Signature and Date

Head of Department

Electrical and Electronics Engineering

## DEDICATION

I would like to dedicate this report to God Almighty, whose unwavering strength and protection were my anchors throughout my three months of industrial work experience. His guidance inspired me to persevere and stay motivated   
(especially when I didn’t feel like it), and His support made this journey not only possible but meaningful. Thank you for being my constant source of strength. And also to my loving parent, who continuously personified the love of God in my life, it would be foolish for me to be ungrateful, so thank you very much

## ACKNOWLEDGEMENT

My deepest acknowledgement and gratitude goes out to God Almighty for his gift of life and strength, making it possible for me to complete my SWEP in this reputable organization.

My profound appreciation goes to the Provost of the College of Engineering Engr. Prof J.O. Dada and the Head of Department, Electrical and Electronics Engineering Engr. Dr. Femi Onibonoje.

I extend my sincere gratitude to xPlug Technologies Limited for graciously allowing me to complete my SWEP training within their esteemed organization. The knowledge and experience gained during this period will undoubtedly prove invaluable in my future endeavors.

I am also deeply appreciative of the unwavering support and guidance provided by all my lecturers in the Electrical and Electronics Department of ABUAD. Your dedication has been truly inspiring, and I am immensely grateful.

Lastly, I would like to express my heartfelt thanks to my parents, Mr. & Mrs. Ugochukwu for their unyielding encouragement, unwavering support, and provision of a nurturing environment during this phase of my professional development. I am truly blessed to have such remarkable parents.

## TABLE OF CONTENTS

[**CERTIFICATION 1**](#_Toc133768935)

[**DEDICATION 2**](#_Toc133768936)

[**ACKNOWLEDGEMENT 3**](#_Toc133768937)

[**TABLE OF CONTENTS 4**](#_Toc133768938)

[**CHAPTER ONE 5**](#_Toc133768939)

[**1.0 INTRODUCTION 5**](#_Toc133768940)

[**1.1 INTRODUCTION TO STUDENTS’ WORK EXPERIENCE PROGRAM (SWEP) 5**](#_Toc133768941)

[**1.2 OBJECTIVES OF SWEP 5**](#_Toc133768942)

[**1.3 BENEFITS OF SWEP 6**](#_Toc133768943)

[**1.4 HISTORY OF SWEP/SIWES 7**](#_Toc133768944)

[**1.5 LOGBOOK 7**](#_Toc133768945)

[**1.6 BACKGROUND OF THE COMPANY 7**](#_Toc133768946)

[**CHAPTER TWO 8**](#_Toc133768947)

[**2.0 THEORY OF WORK DONE 8**](#_Toc133768948)

[**2.1 SOFTWARE DEVELOPMENT 8**](#_Toc133768949)

[**2.2 NETWORK SECURITY 9**](#_Toc133768950)

[**2.3 CLOUD COMPUTING 12**](#_Toc133768951)

[**2.4 TECHNICAL SUPPORT 12**](#_Toc133768952)

[**2.5 CYBERSECURITY 17**](#_Toc133768953)

[**CHAPTER THREE 18**](#_Toc133768954)

[**3.0 WORKDONE 18**](#_Toc133768955)

[**3.1 ACTUAL WORKDONE 18**](#_Toc133768956)

[**CHAPTER FOUR 18**](#_Toc133768957)

[**4.0 EXPERIENCE GAINED AND CHALLENGES FACED 18**](#_Toc133768958)

[**4.1 EXPERIENCE GAINED 18**](#_Toc133768959)

[**4.2 CHALLENGES FACED 18**](#_Toc133768960)

[**CHAPTER FIVE 19**](#_Toc133768961)

[**5.0 CONCLUSION AND RECOMMENDATION 19**](#_Toc133768962)

[**5.1 CONCLUSION 19**](#_Toc133768963)

[**5.2 RECOMMENDATION 19**](#_Toc133768964)

[**REFERENCES 19**](#_Toc133768965)

## 

## CHAPTER ONE

## INTRODUCTION

## INTRODUCTION TO STUDENTS’ WORK EXPERIENCE PROGRAM (SWEP)

The Student Work Experience Program (SWEP) was established in 1973 by the Industrial Training Fund (ITF) with the objective of providing university students with practical technical knowledge related to their course of study before completing their studies. SWEP is a core training program which forms an integral part of the approved minimum academic standards for various degree programs in Nigerian universities. The program aims to bridge the gap between theoretical knowledge and practical application in areas such as engineering, technology, physical sciences, agriculture, medicine, environmental sciences, technical and science education and other vocational training programs offered in Nigerian tertiary institutions. Participation in SWEP is a requirement for the award of diplomas and certificates in certain disciplines in most universities in the country, which is in line with the government's education policy.

## OBJECTIVES OF SWEP

The objectives of SWEP according to ITF (2003) are as follows:

* To prepare the students for industrial work situations they are likely to face after graduation.
* To strengthen employers’ involvement in the entire educational process.
* To make the labor force more vibrant and simultaneously making the economic sector more buoyant.
* To provide students with industrial skills and needed experience while the course of study.
* To assess the interest and suitability of the students in what discipline of work he/she is studying.
* To prepare specialists who will be ready for any working situations immediately after graduation.
* To give students the ability to try and apply the given knowledge.
* To provide students with the knowledge of real work situations.
* To expose the students to work methods not taught in the classroom and to provide access to production equipment.
* To enlighten students on the pitfalls to avoid in the business world as well as to maximize profit in their various fields.
* To make the transition from school to the labor market easier for the students. 10 create conditions and circumstances, this can be as close as possible to the actual workflow.
* To make the transition from school to the world of work easier and enhance the students’ contacts for a possible job offer in the future. i.e., to bridge the gap between theory and practice by providing students with the opportunity to apply their educational knowledge in real work situations.

## 1.3 BENEFITS OF SWEP

SWEP affords students with the following opportunities:

* Appreciate the connection between their courses of study and other related disciplines in the production of goods and services.
* Appreciate the role of professions in their various fields as the creators of change and wealth and indispensable contributors to growing the economy and national development.
* Bridge the gap between the knowledge acquired in institutions and relevant production skills required in work organizations.
* Enhance students’ contact with potential employers while on training.
* Develop and enhance personal attributes such as critical thinking, creativity, initiative, resourcefulness, leadership, time management, presentation skills and interpersonal skills.
* Blend theoretical knowledge acquired in the classroom with practical hands-on application of knowledge.
* Appreciate work method and gain experience in handling equipment and machinery which may not be available in our institution.

## 1.4 HISTORY OF SWEP/SIWES

The Student Work Experience Scheme (SWEP) was launched in 1974 with 748 students from 11 universities. By 1978, the number of participants had increased to about 5,000 students from 32 universities. However, the Industrial Training Fund withdrew from running the program in 1979 due to organizational logistical problems and the increasing financial burden of the rapid expansion of SWEP/SIWES (ITF, 2003). As a result, the Federal Government funded the program through the National Universities Commission (NUC) and the National Board of Technical Education (NBTE) to implement it for a period of five years (1979-1984). During this period, the regulators (NUC and NBTE) worked with the respective universities to implement the program. The program was subsequently revised by the Federal Government, with Decree No. 16 of August 1985 mandating that "all students enrolled in the fields of engineering, technology, business, applied science and applied arts must complete, as part of their studies, a supervised internship". At the same time, the ITF was instructed by the Federal Government to take over the management of SIWES/SIWES in collaboration with the Federal Government.

## 1.5 LOGBOOK

The daily report of the routine work done including discrepancies and corrective action during routine and preventive maintenance and repair are contained in the logbook, also the means of television and radio broadcasting was well stated in the logbook. The logbook was inspected, cross examined and signed by both my Industrial based supervisor and my Institutional based supervisor. It is going to contain your basic information such as:

1. Your recently snapped passport
2. Full name
3. Sex (Male or Female)
4. Matriculation Number
5. Department
6. Faculty
7. Year of Study (200, 300, 400)
8. Name & mailing address of company attached to
9. Names of Industry-Based Supervisor(s)

## 1.6 BACKGROUND OF THE COMPANY

xPlug Technologies Limited is a leading software company in Nigeria, known for crafting innovative solutions that powers businesses of all sizes. Founded on the principles of service integrity and relentless pursuit of excellence, xPlug began its journey in 2009 to redefine the standards of quality service delivery in the technology industry. Our dedication to writing working code ensures our products are robust and functional, with a focus on quality at every step. We are committed to customer satisfaction with unparalleled passion, striving to exceed expectations and build lasting relationships. Embedded in our history is the ethos of empowering businesses with integrity-driven solutions, and as we continue to grow, our commitment to driving positive change remains at the heart of xPlug Technologies Limited. With a vision of to build a strong ICT-based community where businesses are empowered to the grassroots and strong human capital development is the hub of success. And on a mission to provide excellent services in the Area of Information Technology; empowering human capacity for perfect delivery of such services by riding on the platform of service integrity.

Some of their clients include:

* Central Bank of Nigeria
* Lagos State Government
* Green Lotto

## CHAPTER TWO

## THEORY OF WORK DONE

A front-end developer is a type of software developer who specializes in creating and designing the user interface (UI) and user experience (UX) of websites and web applications. The primary responsibility of a front-end developer is to ensure that the visual and interactive aspects of a website or application are user-friendly, aesthetically pleasing, and functionally efficient.

Front-end developers work with various technologies, tools, and languages, including:

1. HTML (Hyper-Text Markup Language): The standard markup language used to create the structure and layout of web pages.
2. CSS (Cascading Style Sheets): A stylesheet language used to control the presentation, formatting, and appearance of web pages, such as colors, fonts, and layout.
3. JavaScript: A programming language that allows developers to add interactivity, animations, and other dynamic elements to websites and web applications.

Front-end developers may also use libraries and frameworks, such as React, Angular, or Vue.js, to streamline their work and create more sophisticated and interactive UIs. Additionally, they often collaborate with back-end developers, who are responsible for the server-side logic and data management, to ensure seamless integration between the front-end and back-end components of a web application or website.

## 2.1 SOFTWARE DEVELOPMENT

**What is software development?**

Software development refers to a set of computer science activities that are dedicated to the process of creating, designing, deploying, and supporting software.

Software itself is the set of instructions or programs that tell a computer what to do. It is independent of hardware and makes computers programmable. There are three basic types:

**System software** to provide core functions such as operating systems, disk management, utilities, hardware management and other operational necessities.

**Programming software** to give programmers tools such as text editors, compilers, linkers, debuggers, and other tools to create code.

**Application software** (applications or apps) to help users perform tasks. Office productivity suites, data management software, media players and security programs are examples. Applications also refer to web and mobile applications like those used to shop on Amazon.com, socialize with Facebook or post pictures to Instagram.

## 2.2 SOFTWARE TESTING LIFE CYCLE

Every production process, including quality assurance (QA), must include testing. Furthermore, the process used to carry out these tests for production frequently consists of a set of coordinated actions known as the "life cycle."

Software items are not an exception. It also passes examinations that are enforced through a final procedure made up of stages or qualifying rounds. It is necessary to first "understand the definition of software testing" in order to have a good understanding of this. Software testing is a "set of processes aimed at investigating, evaluating, and ascertaining the completeness and quality of computer software," according to Techopedia. It is a system that aids in guaranteeing that a software product complies with business, technical, functional, regulatory, and user requirements. To put it simply, software testing consists of:

* Verifying the performance of the software’s functional features
* Conducting technical assessment to identify bugs/errors that ensures the software remains error-free and rightfully executes its functions
* Analyzing usability for different user scenarios, security aspects, localization, and compatibility, and/or installation issues

Software Testing Life Cycle (STLC) is a systematic process aimed at ensuring the quality and functionality of software through its various phases. It is a series of phases that guide the testing process to ensure the quality and functionality of the software. Each phase contains specific activities and goals aimed at identifying defects and ensuring that the software meets the specified requirements.

**Phases**

1. - **Requirements analysis**

This is the initial stage that involves comprehending the project's requirements, examining testable functions, and setting objectives for the evaluation. It provides a framework for the entire testing process. The main purpose of this step is to understand what needs to be tested and define clear test objectives.

1. - **Test Planning**

In the test planning section, a detailed test plan describes the test approach, objectives, scope, schedule, and resource allocation. Using this step, you can assess the possible risks and plan to reduce them. It helps companies stay efficient, allocate resources and provide comprehensive testing options.

1. - **Test design**

Based on requirements, test cases and test scenarios are created. The main goal is to use all available methods to provide complete test coverage. This step helps to analyze the expected results and determine their validation.

1. - **The test environment**

Before running the test, it is important to set up the test environment. The design includes hardware, software and test data. To get the right results, this step ensures that the test environment is the same as the production environment.

1. — **Test Management**

This is where tests are run on cases and planned tests are run. This step records the test results, including pass/fail status and any errors found. This helps companies identify problems or bugs in software.

1. - **Defect Tracking**

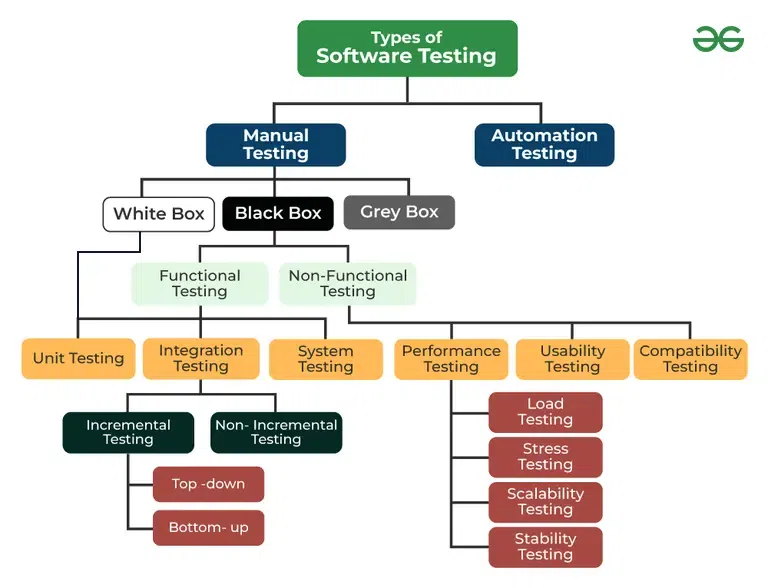
Defects are discovered during testing; defect records are recorded and appropriate resolution is sought. This phase includes documenting all identified issues, prioritizing them, and working with the development team to determine the right timeline.

1. - **Test Report**

Finally, in this section, a test report is created that shows test progress, test coverage, error metrics, and software quality. These reports help stakeholders make decisions and assess the software's readiness for deployment.

These 7 components of STLC work together to provide a systematic and structured approach to testing that results in a high quality systems that meets user expectations.

***TYPES OF SOFTWARE TESTING***



**Based on Execution:**

* **Manual Testing:** Testing is performed by human testers without the use of automation tools. Useful for exploratory testing and usability evaluations.
* **Automated Testing:** Testing is executed using automated tools and scripts. Ideal for repetitive and regression testing, improving efficiency and accuracy.

**Based on Testing Phase:**

* **Unit Testing:** This focuses on testing individual parts of the software. It is typically performed by software-developers during the development phase.
* **Integration Testing:** This stage tests the interaction between integrated components or systems to ensure they work as they should.
* **System Testing:** This checks the complete and integrated software system against the specified requirements. It is aimed at checking if the software works as a whole.
* **Acceptance Testing:** This affirms that the software meets business requirements and is ready for deployment.

**Based on Test Approach:**

* **Black-Box Testing:** This tests the functionality of the application without knowledge of the internal code structure. It is usually done by people with little to no knowledge of the coding structure.
* **White-Box Testing:** This tests the internal logic and structure of the code. It has a requisite of knowledge of the code and is used to ensure internal operations are functioning correctly.
* **Gray-Box Testing:** This tests elements of both black-box and white-box testing. Testers have partial knowledge of the internal workings.

**Based on Test Objective:**

* **Functional Testing:** This tests the software against functional requirements or specifications. Ensures the software performs its intended functions.
* **Non-Functional Testing:** This tests the non-functional aspects such as performance, usability, and security. Includes performance testing, load testing, and stress testing.
* **Regression Testing:** This tests that new changes or enhancements do not adversely affect existing functionalities.
* **Smoke Testing:** It’s a preliminary test to check the basic functionality of the software. Also known as "Build Verification Testing."
* **Sanity Testing:** Aimed at checking if specific functionalities after changes or bug fixes to ensure they work as intended. A subset of regression testing.

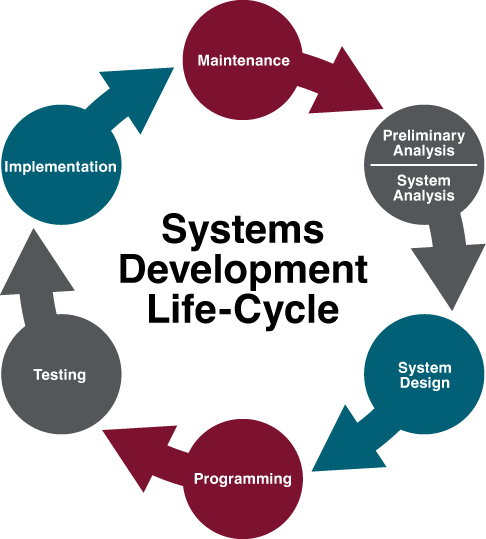
**Based on Test Documentation:**

* **Static Testing:** This test involves reviewing the code and documentation without executing the code. Includes code reviews and static analysis.
* **Dynamic Testing:** It is aimed at executing the code to verify its behavior during runtime. Includes both the functional and non-functional testing.

## 2.3 SOFTWARE DEVELOPMENT LIFE CYCLE

Software development life cycle (SDLC) is the term used in the software industry to describe the process for creating a new software product. [Software developers](https://www.coursera.org/articles/software-developer) use this as a guide to ensure software is produced with the lowest cost and highest possible quality in the shortest amount of time.

In the 1960s, the Systems Development Life Cycle (SDLC) was created for the purpose of managing sizable software projects that were executed on corporate mainframes. This methodical, risk-averse approach to software development is intended to oversee sizable projects involving numerous programmers and systems. It is not flexible enough to allow for design changes and necessitates a clear, upfront understanding of what the software is supposed to do. This method is akin to an assembly line procedure in that it is obvious to all parties involved what the final product should accomplish and that significant modifications are expensive and time-consuming to execute. The SDLC procedure is as follows. Although there are many definitions of the SDLC methodology, the majority include the following stages.



1. ***Preliminary Analysis:***First, a request for a new or replacement system is examined. In the review, there are inquiries like: What is the issue that has to be resolved? Can a solution be created? What other options are there? And what is being done about it right now? Is our company a good fit for this project? "Needs analysis" is the term used to describe this procedure. Following the resolution of these issues, a feasibility study is started (more on this later). This is a crucial stage in deciding whether to start the project.
2. ***System Analysis.*** In this phase one or more system analysts work with different stakeholder groups to determine the specific requirements for the new system. No programming is done in this step. Instead, procedures are documented, key players/users are interviewed, and data requirements are developed in order to get an overall impression of exactly what the system is supposed to do. The result of this phase is a system requirements document and may be done by someone with a title of Systems Analyst.
3. ***System Design.*** In this stage, a designer develops the precise technical specifications needed for the system using the system requirements document produced in the previous phase. The business needs are converted into precise technical requirements during this step. Here, the designs for the database, user interface, data inputs and outputs, and reports are created. This phase produces a document that describes the system design. Depending on the size of the project, someone with the title of Systems Analyst, Developer, or Systems Architect may develop the system using the information in this document, which contains everything a programmer needs to actually create the system.
4. ***Programming.*** The code finally gets written in the programming phase. Using the system design document as a guide, programmers develop the software. The result of this phase is an initial working program that meets the requirements specified in the system analysis phase and the design developed in the system design phase. These tasks are done by persons with titles such as Developer, Software Engineer, Programmer, or Coder.
5. ***Testing.*** Using the system requirements document created in the previous step, a designer creates the precise technical specifications required for the system in this stage. In this step, the business needs are translated into exact technological requirements. This is where the designs for the reports, data inputs and outputs, user interface, and database are made. This stage results in a document that outlines the architecture of the system. This document contains all the necessary information for a programmer to actually create the system, therefore depending on the magnitude of the project, someone with the title of Systems Analyst, Developer, or Systems Architect may use it to design the system.
6. ***Implementation****.* Once the new system is developed and tested, it has to be implemented in the organization. This phase includes training the users, providing documentation, and data conversion from the previous system to the new system. Implementation can take many forms, depending on the type of system, the number and type of users, and how urgent it is that the system becomes operational. These different forms of implementation are covered later in the chapter.
7. ***Maintenance.*** The last phase takes place after implementation phase is complete. In the maintenance phase the system has a structured support process in place. Reported bugs are fixed and requests for new features are evaluated and implemented. Also, system updates and backups of the software are made for each new version of the program. Maintenance ensures that the software continues to operate effectively and adapts to changing requirements or environments.

**Types of Maintenance**

* ***Corrective Conservation***

This involves fixing bugs and crimes in the software that were n't discovered during the original development phase. It's about resolving issues that affect the software's functionality.

* ***Adaptive conservation***

This type of conservation is performed to keep the software up- to- date with changes in its terrain. For illustration, conforming the software to work with a new operating system or tackle.

* ***Perfective conservation***

This involves perfecting or enhancing the software to meet new stoner conditions or to ameliorate performance. It's about making the software more, indeed when it's not rigorously necessary.  It can simply range from changing the GUI of the software to make it more attractive and user friendly to making drastic changes in the core code to improve the runtime and performance

* ***Preventative conservation***

This type of conservation is performed to help unborn problems. It involves streamlining attestation, optimizing law, or restructuring the software to make it easier to maintain in the future.

## 2.4 UI/UX DESIGN (WITH ACCESSIBILTY IN MIND)

**What is User Interface (UI) Design?**

***User interface (UI) design*** is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms—e.g., voice-controlled interfaces.

In summary, A user interface (UI) is the point of communication between a person and a machine. It’s what you see, hear, say, and touch in order to give instructions to a device or receive information back from it.

**What makes up a user interface?**

* **Input components: Anything you use to initiate an interaction**

E.g. Menu buttons, checkboxes, dropdown lists, free text fields, radio buttons, and anything you can tap or swipe.

* **Navigational components: The elements used to move around an environment**

A progress bar, breadcrumbs, and search boxes.

* **Graphics and visuals: The components that form the basis for the UI**

Windows, transitions, menus, icons, text, fonts, images, color palettes, layout, and any other signposting.

* **Feedback: How the UI communicates back to the user**

Text boxes/messages, color changes, sound prompts, virtual assistants, and haptic feedback.

**What is User Experience?**

It refers to the feeling users experience when using a product, application, system, or service. Regardless of whether different aspects of the experience are under the direct control of the product or are merely associated with the product, the total experience is considered part of the UX from the user’s perspective.

**What are UX design principles?**

1. **Focus on the User**: Make sure to understand who will be using your design, what they need, and what they want. Your design should solve their problems and make their experience better.
2. **Be Consistent**: Keep things uniform throughout your design, like colors, fonts, and layouts. Consistency helps users get used to how things work and makes your design easier to use.
3. **Keep It Simple**: Avoid adding unnecessary features or details. Make the design as straightforward as possible so that users can figure out how to use it without getting confused.
4. **Think About Accessibility**: Design so that everyone, including people with disabilities, can use your product. This might mean making sure text is readable or that the design works with screen readers.
5. **Give Feedback**: Let users know what’s happening when they interact with your design. For example, if they click a button, they should see a response that shows if the action was successful or if something went wrong.
6. **Make It Usable**: The design should be easy to navigate and should help users complete their objectives quickly and with ease. This often involves testing and tweaking the design to make sure it’s user-friendly.
7. **Ensure Visual Hierarchy**: Arrange all elements in a way that makes it easy for users to see what’s most important. Use it to make them focus on what you want them to focus on.

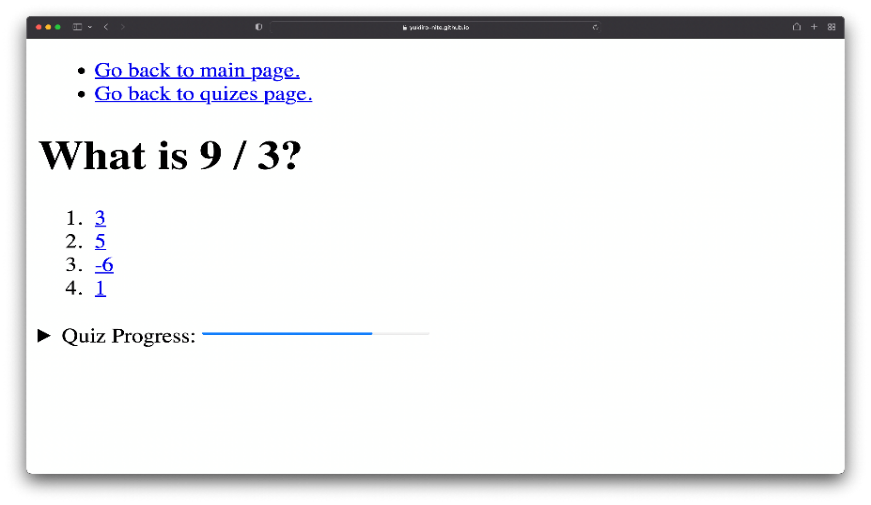
## 2.5 TECHNOLOGIES USED AND SKILLS LEARNED

**Introduction**

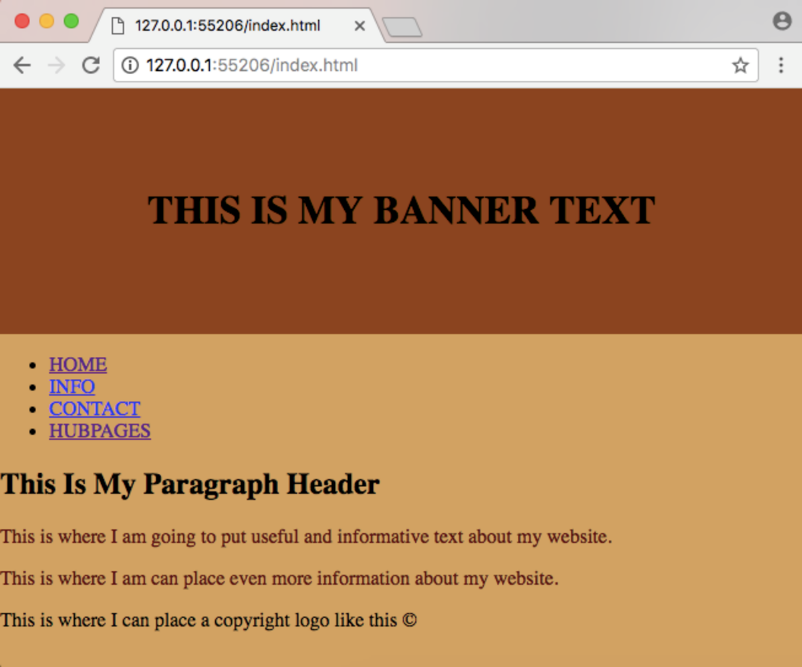
The story begins with Tim Berners-Lee, a British computer scientist working at CERN. In the late 1980s, he conceived the idea of the World Wide Web as a way to share information among researchers. In 1990, he proposed the concept of the web and created the first web browser and web server software. In 1991, Berners-Lee introduced HTML (Hypertext Markup Language), the fundamental language of the web. HTML allowed the structuring of documents with hyperlinks, forming the basis of web pages. The first-ever website, which explained the World Wide Web, went live in 1993.

At the core of web development lie 3 fundamental technologies namely: HTML, CSS, and JavaScript. They are the basic building blocks of every website.

* **HTML (Hypertext Markup Language)** serves as the backbone of web content. It provides the structure for web pages, defining the meaning and hierarchy of information. HTML ensures that content is organized logically and can be understood by both browsers and assistive technologies, making it essential for accessibility and search engine optimization.



* **CSS (Cascading Style Sheets)** is responsible for the presentation and layout of web pages. It controls how HTML elements are displayed, including their color, size, position, and responsiveness to different screen sizes. CSS allows developers to create visually engaging designs and ensure a consistent user experience across various devices and platforms.



* **JavaScript** is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

## 2.6 PROJECT DOCUMENTATION

## CHAPTER THREE

## WORKDONE

## 3.1 ACTUAL WORKDONE

## CHAPTER FOUR

## EXPERIENCE GAINED AND CHALLENGES FACED

## 4.1 EXPERIENCE GAINED

## 4.2 CHALLENGES FACED

## CHAPTER FIVE

## CONCLUSION AND RECOMMENDATION

## 5.1 CONCLUSION

## 5.2 RECOMMENDATION

## REFERENCES

[What Is Software Development? | IBM](https://www.ibm.com/topics/software-development#:~:text=Software%20development%20refers%20to%20a,hardware%20and%20makes%20computers%20programmable.)

[What Is the Software Development Life Cycle? SDLC Explained | Coursera](https://www.coursera.org/articles/software-development-life-cycle)

[Software Testing Life Cycle: Everything You Need to Know (radixweb.com)](https://radixweb.com/blog/software-testing-life-cycle?utm_source=MediumAB&utm_medium=Pratik)

[Understanding the Objectives of Software Testing Life Cycles | Medium](https://pratikmistry.medium.com/demystifying-the-software-testing-life-cycles-with-7-key-phases-fa5efe11b0c3)

[7.3. Systems Development Life Cycle – Information Systems for Business and Beyond (pressbooks.pub)](https://ecampusontario.pressbooks.pub/informationsystemscdn/chapter/7-3-systems-development-life-cycle/)

Thales Group. (n.d.). 4 Types of Software Maintenance & What is Software Maintenance?. <https://cpl.thalesgroup.com/software-monetization/four-types-of-software-maintenance>

[Types of Software Testing | GeeksforGeeks](https://www.geeksforgeeks.org/types-software-testing/)

[What is User Interface (UI) Design? — updated 2024 | IxDF (interaction-design.org)](https://www.interaction-design.org/literature/topics/ui-design?srsltid=AfmBOoqP6FcPXoKWBGXng6uuWHkMM7DSlmzX3ER-lQKV7dP69-IogAlp)

[What is User Experience? | Definition and Overview (productplan.com)](https://www.productplan.com/glossary/user-experience/)

[The History and Evolution of Web Development: From HTML to the Modern Web | by Abhishek Tiwari | Medium](https://medium.com/@Abhishek-Tiwari/the-history-and-evolution-of-web-development-from-html-to-the-modern-web-982e3f90e891)

1. Introduction
   * Brief overview of the company
   * Duration of the training
   * Your role/position during the training
2. Learning Objectives
   * What you aimed to learn during this training
3. Technologies and Skills Learned  
   3.1 HTML
   * Basic structure of web pages
   * Semantic HTML elements
   * Forms and input types

3.2 CSS

* + Styling web pages
  + Layout techniques (e.g., Flexbox, Grid)
  + Responsive design principles

3.3 Web Design

* + Design principles
  + User experience (UX) considerations
  + Tools used (e.g., Figma, Adobe XD)

3.4 JavaScript Basics

* + Variables and data types
  + Functions and event handling
  + DOM manipulation

1. Bookstore Website Project
   * Project overview and objectives
   * Your role in the project
   * Technologies used
   * Key features implemented
   * Challenges faced and how you overcame them
   * Screenshots or link to the final product (if possible)
2. Work Environment and Company Culture
   * Team structure and collaboration
   * Communication methods
   * Agile/Scrum practices (if applicable)
3. Personal Growth and Skill Development
   * How your skills improved over the 3 months
   * Soft skills developed (e.g., teamwork, time management)
4. Challenges and Learning Experiences
   * Difficulties you encountered
   * How you solved problems
   * What you learned from these experiences
5. Conclusion
   * Summary of your overall experience
   * How this training has prepared you for your future career
   * Acknowledgments to mentors or team members who helped you
6. Appendices (if applicable)
   * Code samples
   * Project documentation
   * Certificates or acknowledgments received