Facilitator: Ukwuoma Victor.

Facilitating Group: Coxygen Global Haskell Plutus GROUP 14

JULY 2025 REPORT

Table of Contents

Summary Report

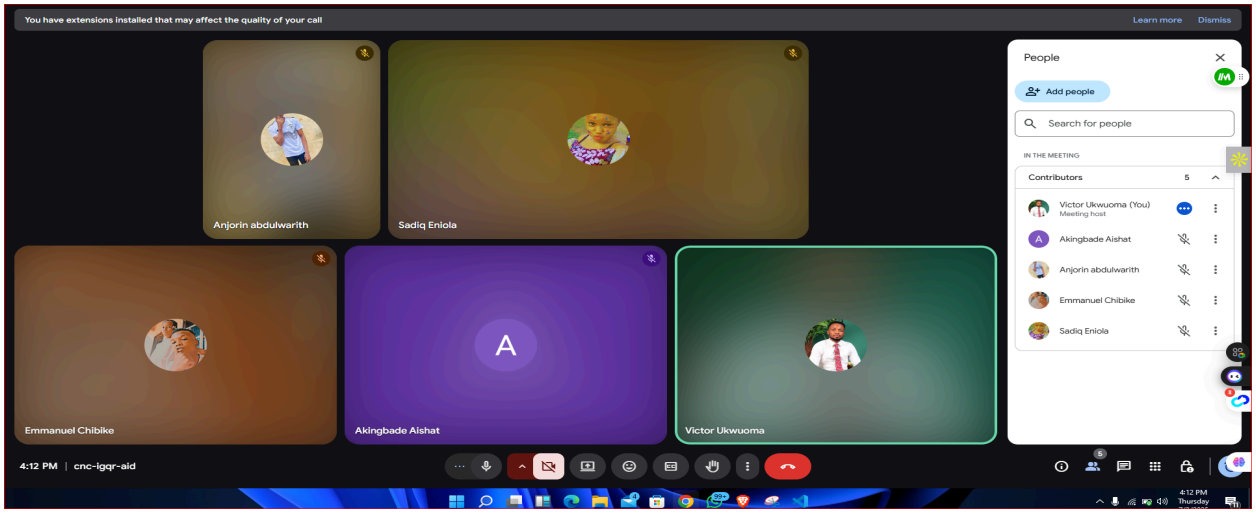
I began the month of July by reaching out to some of the inactive students in my group. In the first week, I conducted several one-on-one sessions to assist with registration and email verification. We had a total of eight general class sessions covering revision of Chapter One through Chapter Four in Haskell. Due to the complexity of the topic, I allocated extra time for review to ensure a solid foundational understanding. Additionally, I scheduled extra support sessions for students who found certain topics challenging.

Numerical Growth.

Currently, I have a total of 19 active students enrolled in the program, showing a gradual increase from 11 students in June. Of these, 14 students are consistently engaged and actively participating, while I have continued to monitor and provide support to the remaining 5 students to encourage full engagement. Notably, 11 students have successfully completed their on-chain credentials up to Chapter 5 and have also uploaded the corresponding chapters to GitHub.

Haskell Group 14 Facilitation Report

July 3, 2025: Live Session with G-014 Students on Thursday, 3rd July

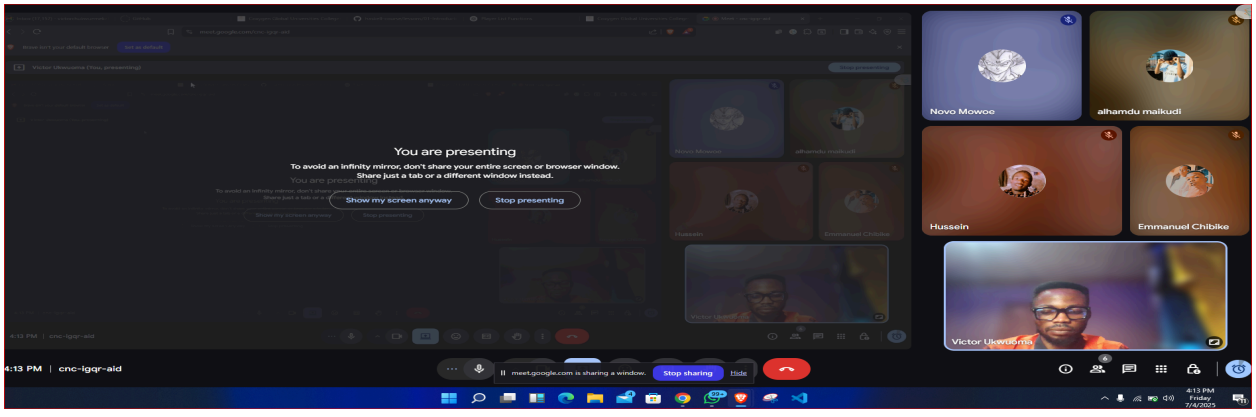


🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd
We took an in-depth study of infix and prefix notations, common data types in Haskell, and polymorphic values and type variables, as well as understanding of coding and indentation rules.

July 4, 2025: Live Session with G-014 Students on Friday, 4th July



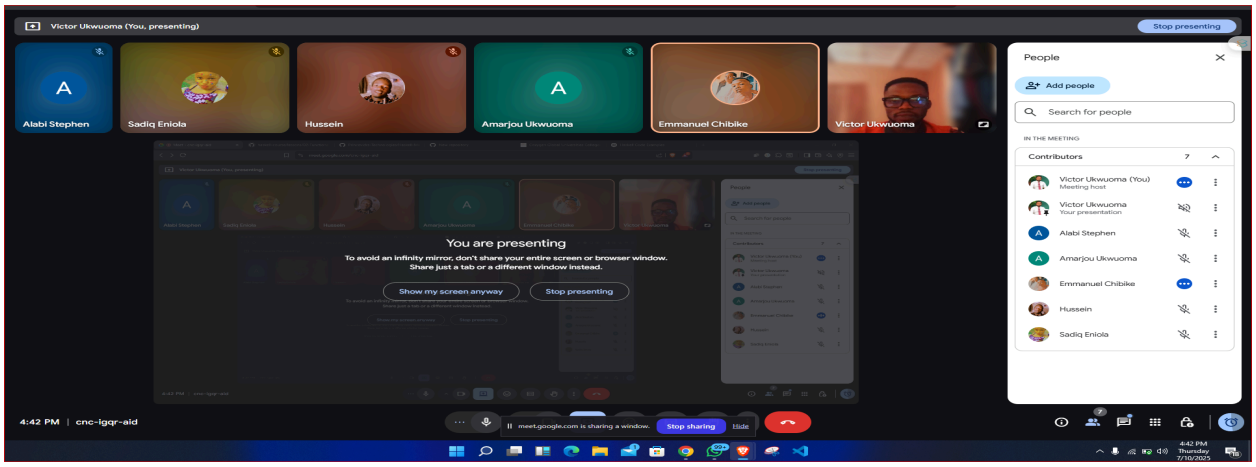
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

We started conditions and helper constructions as we delved into if-then-else expressions.

July 10, 2025: Live Session with G-014 Students on Thursday July 10, 2025



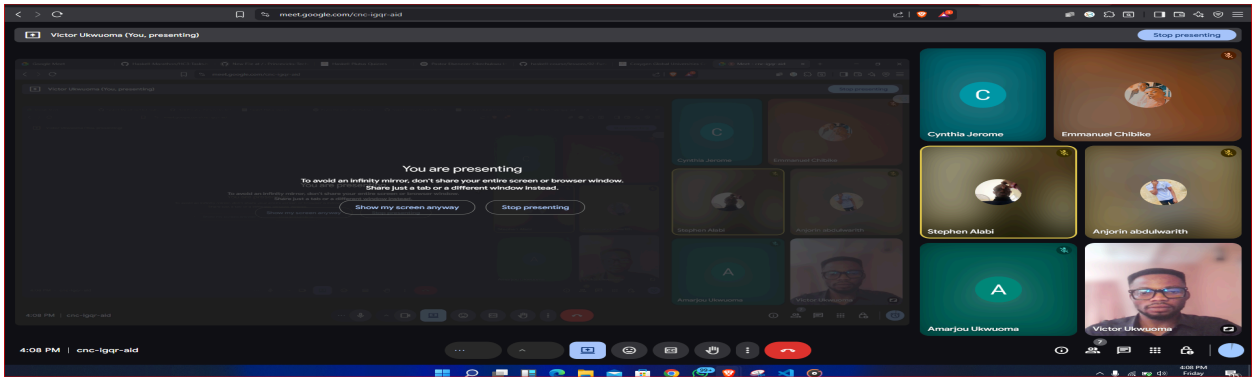
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

We continued with if-then-else expressions, deepening our understanding of guards and the differences between the two approaches to conditional statements.

July 11, 2025: Live Session with G-014 Students on Friday, July 11, 2025



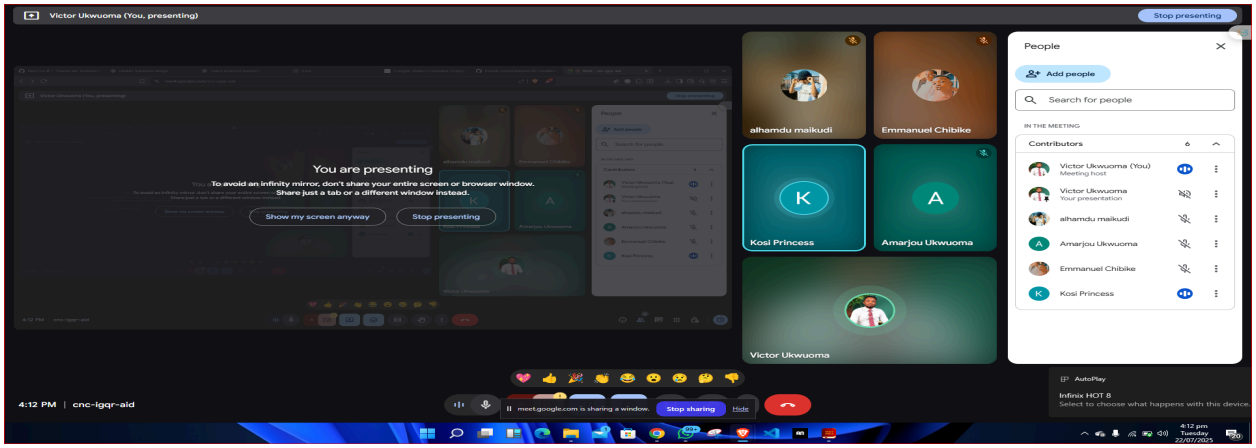
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

We looked at let and where expressions.

July 22nd, 2025: Live Session with G-014 Students on



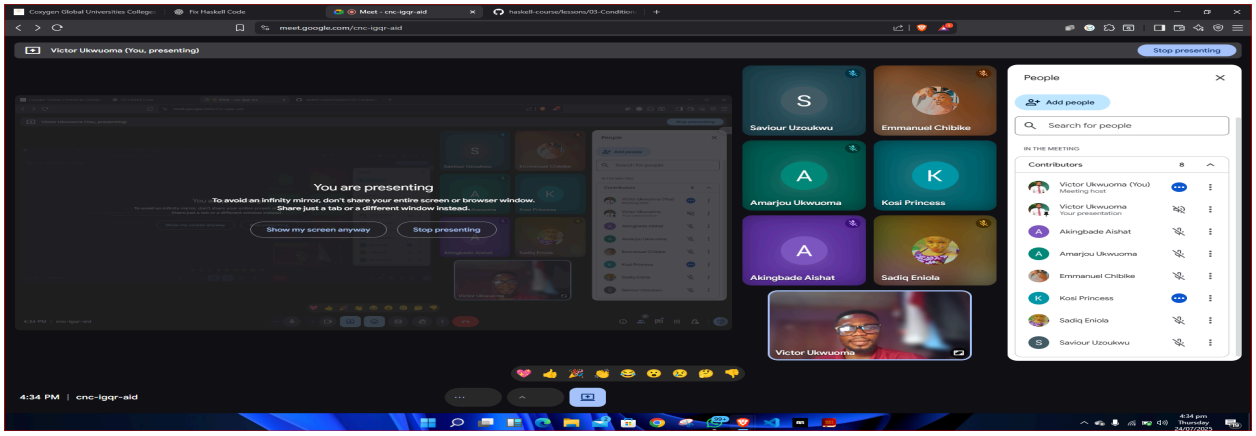
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

We further explore let-then-where expressions. This part was a little bit challenging to the students; hence, we had more time dedicated to it.

24th July, 2025: Live Session with G-014 Students on



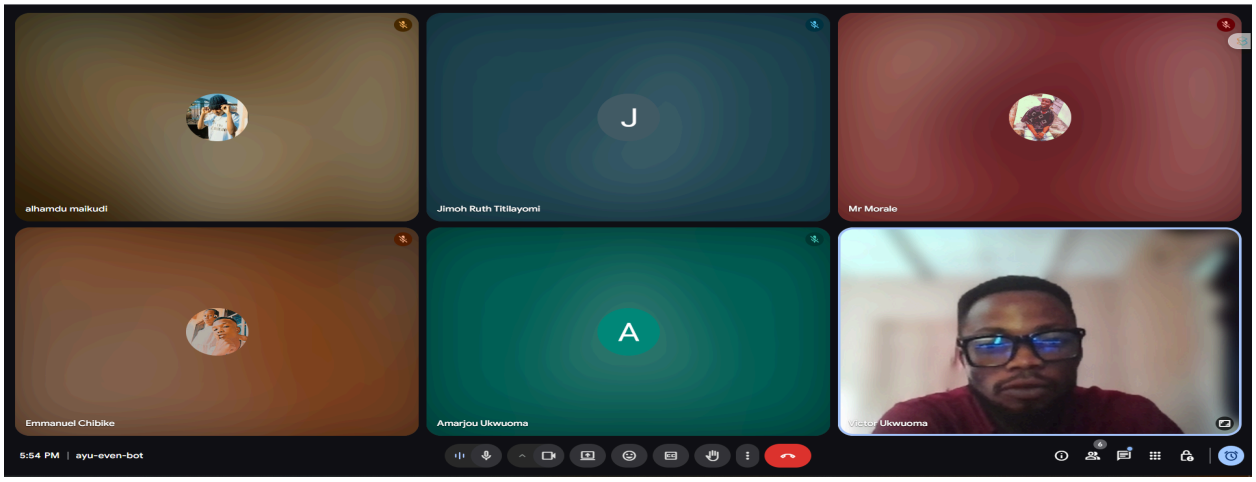
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

We took an introductory lesson on pattern matching in functions.

Friday, 25th July, 2025: Live Session with G-014 Students.



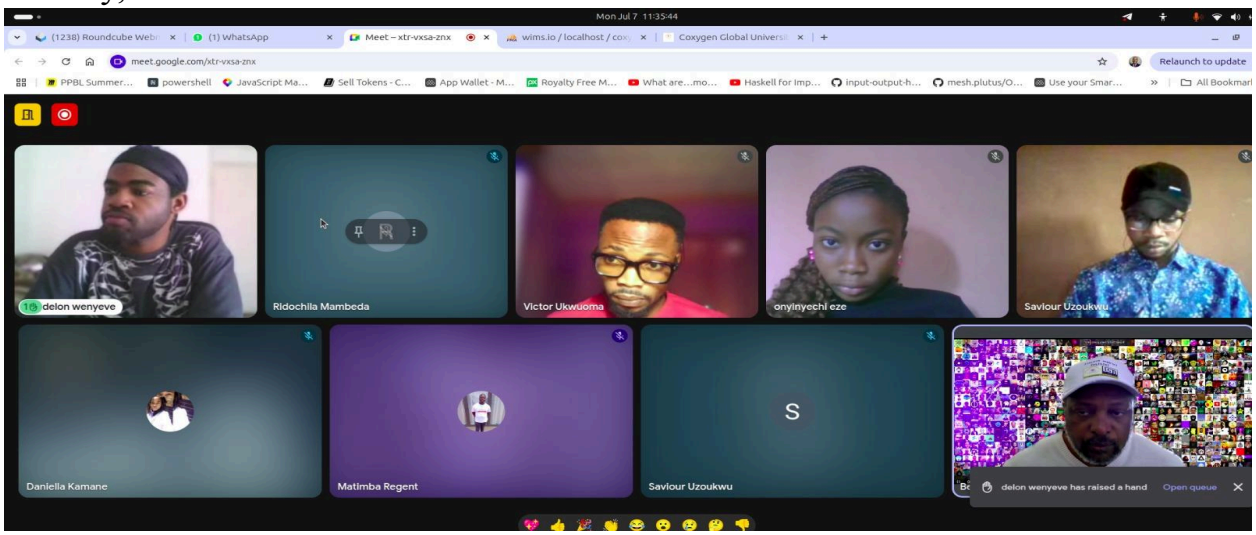
🕔 Time: 4–5PM (WAT)

🔗 Join us via the link below every Thursday and Friday:

https://meet.google.com/wfh-dvdc-ebd

Facilitators Meeting with Mr. Benard S.

7th July, 2025 MEETING WITH MR. BERNARD AND THE FACILITATORS



🕔 Time: 9:30–2:00 PM (Nigerian Time) | 10:30-3:00 PM (SAT)

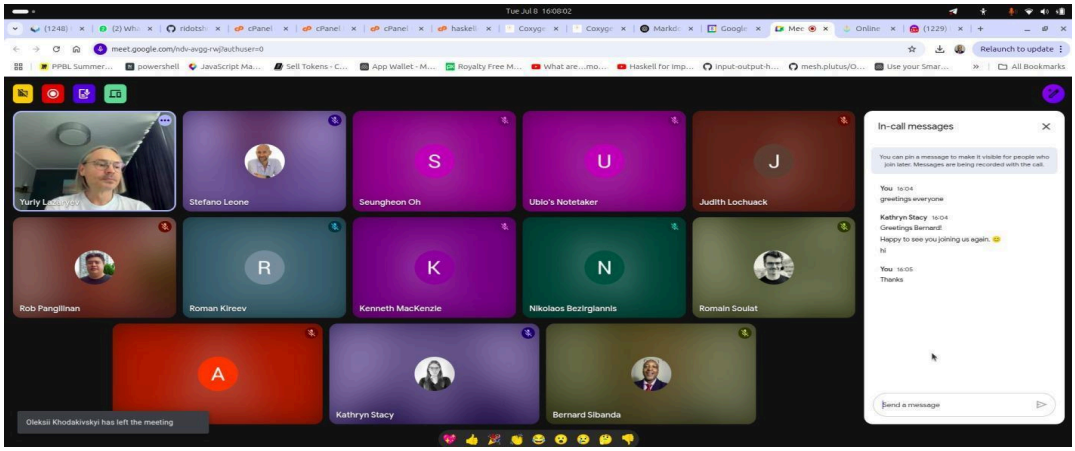
🔗 Join us via the link below every FRIDAY and SUNDAY:

https://meet.google.com/xtr-vxsa-znx

The meeting focused entirely on auditing the new website and test-running its various components to ensure full functionality. The team reviewed the site's structure, design, and features to identify any errors or inconsistencies. Functionality tests were

conducted across different pages and devices to verify that all interactive elements worked as intended. The aim was to ensure that the website is fully operational, user-friendly, and ready for public use.

July 8, 2025 PLUTUS WORKING GROUP LIVE SESSIONS



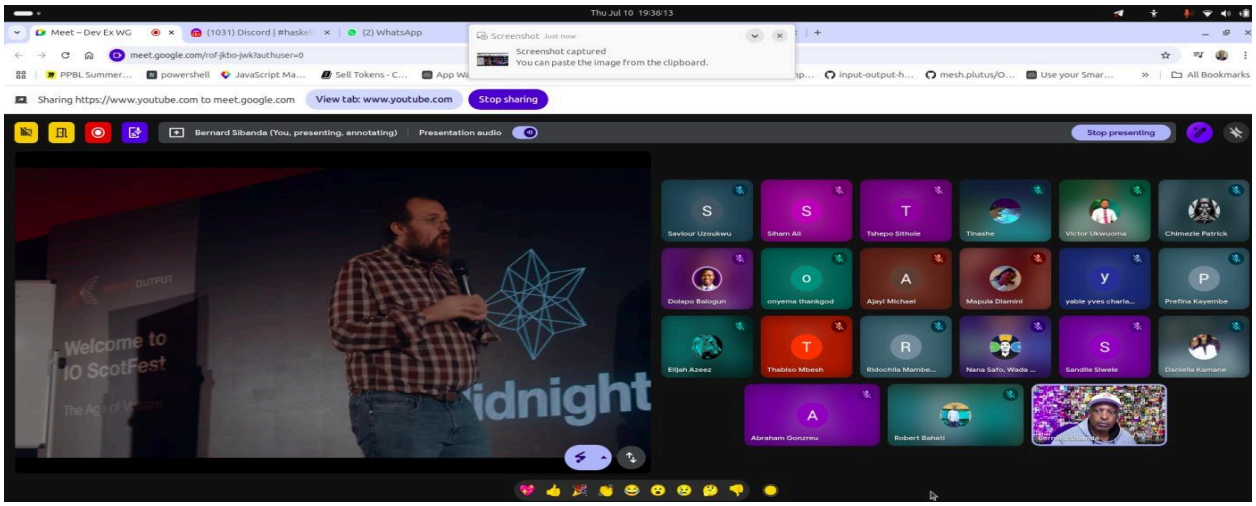
🕔 Time: 9:30–2:00 PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAY and SUNDAY:

https://meet.google.com/xtr-vxsa-znx

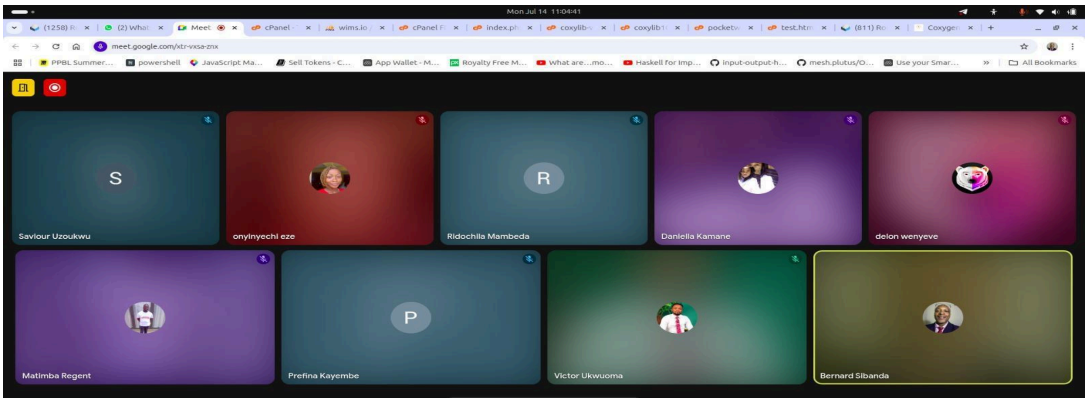
The meeting centered on the continued auditing of the website to eliminate detected bugs and inconsistencies in its functionalities. The team revisited previously identified issues, such as phone number and email verification code delays during registration.

Thursday, July 10, 2025: Dev Ex WG Meeting



🔗 Thursday, 10 July · 🕔 7:00 – 8:00pm
Time zone: Africa/Johannesburg
Google Meet joining info: Video call link: https://meet.google.com/rof-jkbo-jwk

14th July, 2025 MEETING WITH MR. BERNARD AND THE FACILITATORS

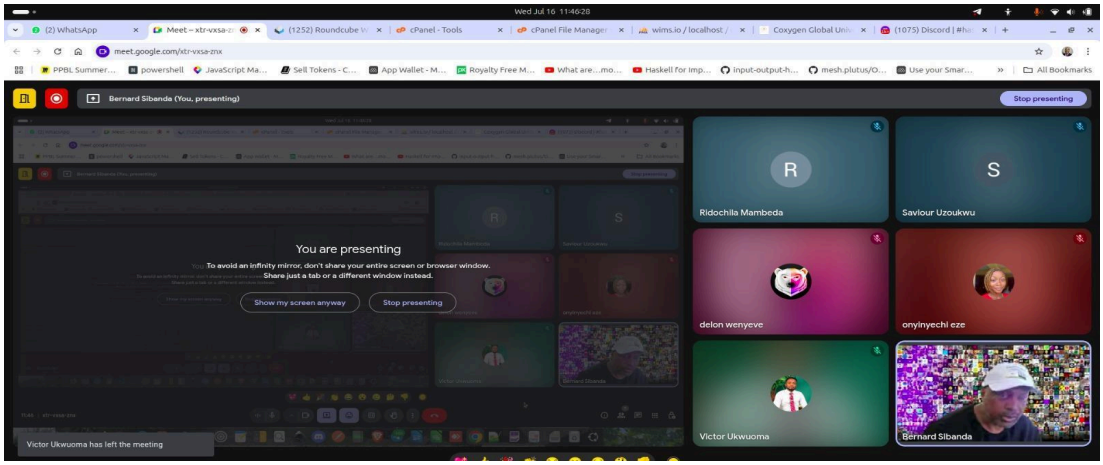


🕔 Time: 9:30–2:00 PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAY and SUNDAY:

https://meet.google.com/xtr-vxsa-znx

16th July, 2025 MEETING WITH MR. BERNARD AND THE FACILITATORS

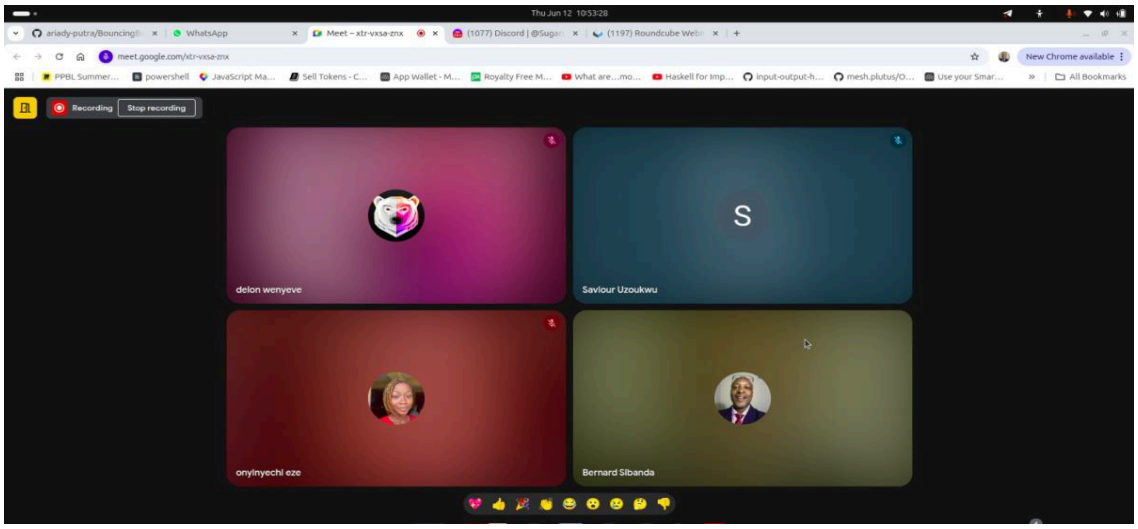


🕔 Time: 9:30–2:00 PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAY and SUNDAY:

https://meet.google.com/xtr-vxsa-znx

MEETING WITH MR. BERNARD AND THE FACILITATORS ON 23rd June, 2025



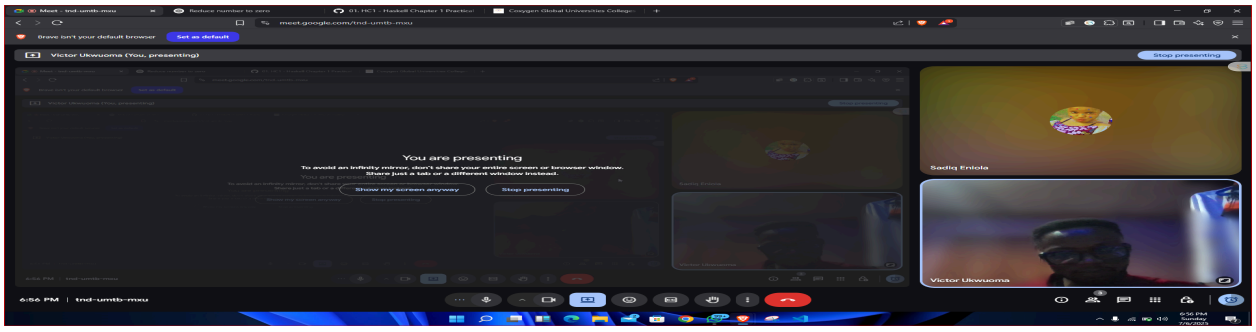
🕔 Time: 9:30–2:00 PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAY and SUNDAY:

https://meet.google.com/xtr-vxsa-znx

Extra Lessons/Training sessions with student developers

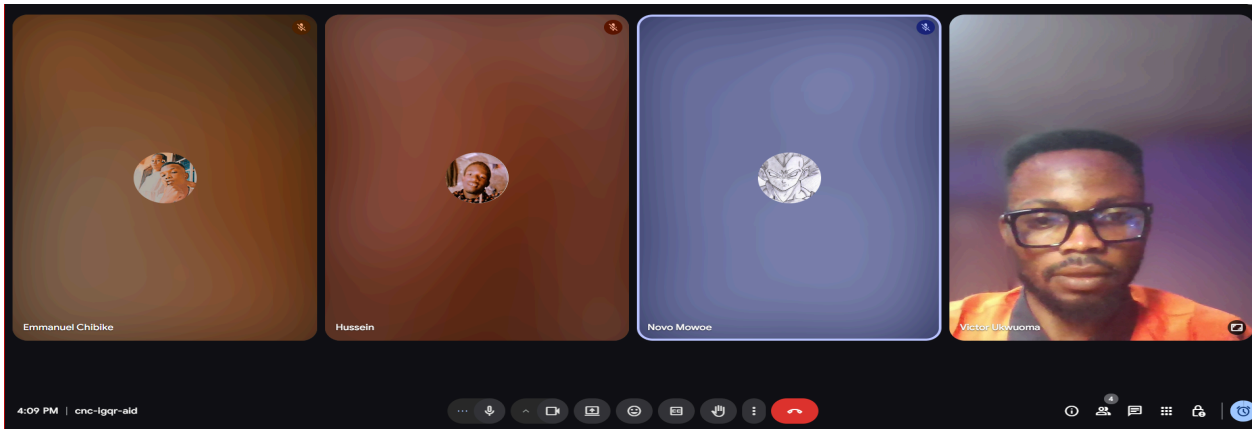
July 6, 2025: Extra lesson session with student developers on Haskell data types.



🕔 Time: 6–7PM (WAT)

https://meet.google.com/wfh-dvdc-ebd

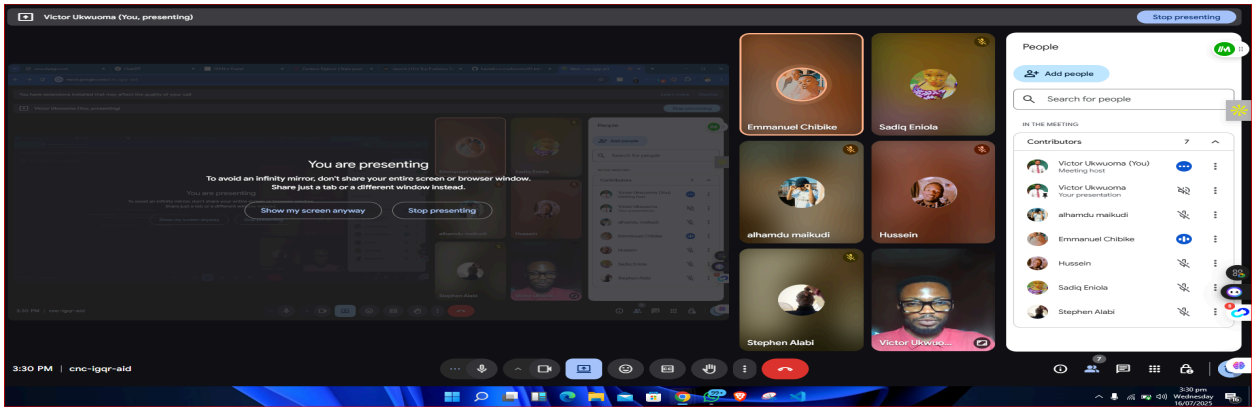
9th July, 2025: Extra lesson Meeting session with student developers



🕔 Time: 8–9 PM (WAT)

https://meet.google.com/wfh-dvdc-ebd

Wednesday, 16th July, 2025: Extra lesson meeting session with student developers on GitHub uploading.



🕔 Time: 3–4PM (WAT)

https://meet.google.com/wfh-dvdc-ebd

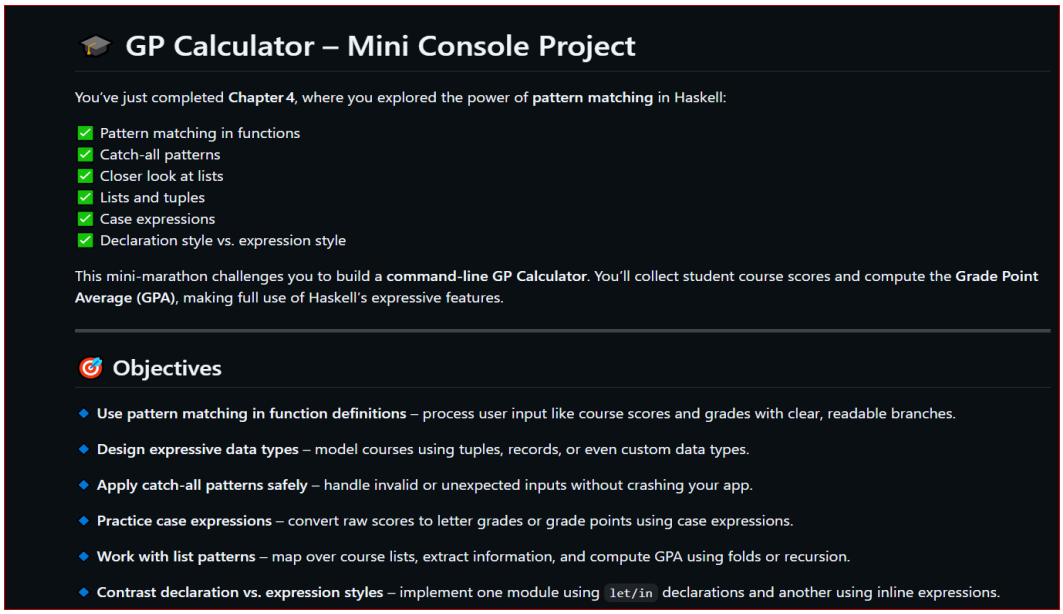
ACHIEVEMENTS

1. I was able to recruit 6 students for the month of July, among whom 4 have completed their registration and verification.
2. I completed Haskell chapters up to chapter 15 and uploaded them to my GitHub.

Developer Session And Source Codes

Developer Session

1) HASKELL CHAPTER 4 PROJECT TASK FOR STUDENTS DEVELOPERS ON REAL-LIFE ISSUES: Grade Point Calculator.



🎓 GP Calculator – Mini Console Project

You've just completed **Chapter 4**, where you explored the power of **pattern matching** in Haskell:

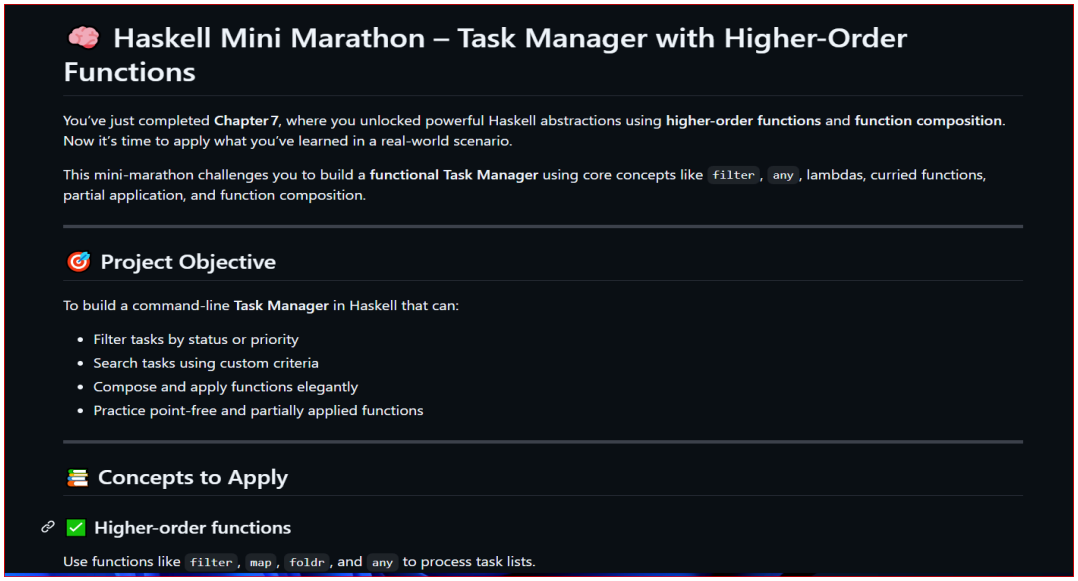
- ✅ Pattern matching in functions
- ✅ Catch-all patterns
- ✅ Closer look at lists
- ✅ Lists and tuples
- ✅ Case expressions
- ✅ Declaration style vs. expression style

This mini-marathon challenges you to build a **command-line GP Calculator**. You'll collect student course scores and compute the **Grade Point Average (GPA)**, making full use of Haskell's expressive features.

🎯 Objectives

- 🔹 **Use pattern matching in function definitions** – process user input like course scores and grades with clear, readable branches.
- 🔹 **Design expressive data types** – model courses using tuples, records, or even custom data types.
- 🔹 **Apply catch-all patterns safely** – handle invalid or unexpected inputs without crashing your app.
- 🔹 **Practice case expressions** – convert raw scores to letter grades or grade points using case expressions.
- 🔹 **Work with list patterns** – map over course lists, extract information, and compute GPA using folds or recursion.
- 🔹 **Contrast declaration vs. expression styles** – implement one module using `let/in` declarations and another using inline expressions.

2) HASKELL CHAPTER 5 PROJECT TASK FOR STUDENTS DEVELOPERS ON REAL-LIFE ISSUES: Task Manager with higher-order functions.



🧠 Haskell Mini Marathon – Task Manager with Higher-Order Functions

You've just completed **Chapter 7**, where you unlocked powerful Haskell abstractions using **higher-order functions** and **function composition**. Now it's time to apply what you've learned in a real-world scenario.

This mini-marathon challenges you to build a **functional Task Manager** using core concepts like `filter`, `any`, lambdas, curried functions, partial application, and function composition.

🎯 Project Objective

To build a command-line **Task Manager** in Haskell that can:

- Filter tasks by status or priority
- Search tasks using custom criteria
- Compose and apply functions elegantly
- Practice point-free and partially applied functions

📚 Concepts to Apply

✅ Higher-order functions

Use functions like `filter`, `map`, `foldr`, and `any` to process task lists.

Repository link: https://github.com/Princevicks-Technologies/Haskel-Marathon-Chapter-1-19

2) HASKELL CHAPTER 6 PROJECT TASK FOR STUDENTS DEVELOPERS ON REAL-LIFE ISSUES: Recursive Budget Analyzer—Mini Console.

Recursive Budget Analyzer – Mini Console Project

You've just completed Chapter 6, where you unlocked the power of recursion and folds in Haskell:

- ✅ Why recursion is core to Haskell
- ✅ Thinking recursively – breaking down problems into base & recursive steps
- ✅ Recursive patterns: sum, product, and, length, reverse, drop, take, map, filter
- ✅ Steps to create your own recursive function
- ✅ Extracting the `foldr` pattern
- ✅ The `foldl` and `foldl'` functions
- ✅ Choosing the right fold for performance and clarity

This mini-marathon challenges you to build a **console-based Recursive Budget Analyzer**. You'll process daily income and expense records using custom recursion and folds.

🎯 Objectives

🔁 Use recursion to define list-processing functions:
Implement custom versions of `sum`, `filter`, `map`, `reverse`, and more.

💡 Think recursively:
Break list operations into base and recursive steps.

🧩 Use pattern matching to deconstruct and process transactions.

⚖️ Apply `foldr`, `foldl`, and `foldl'`:

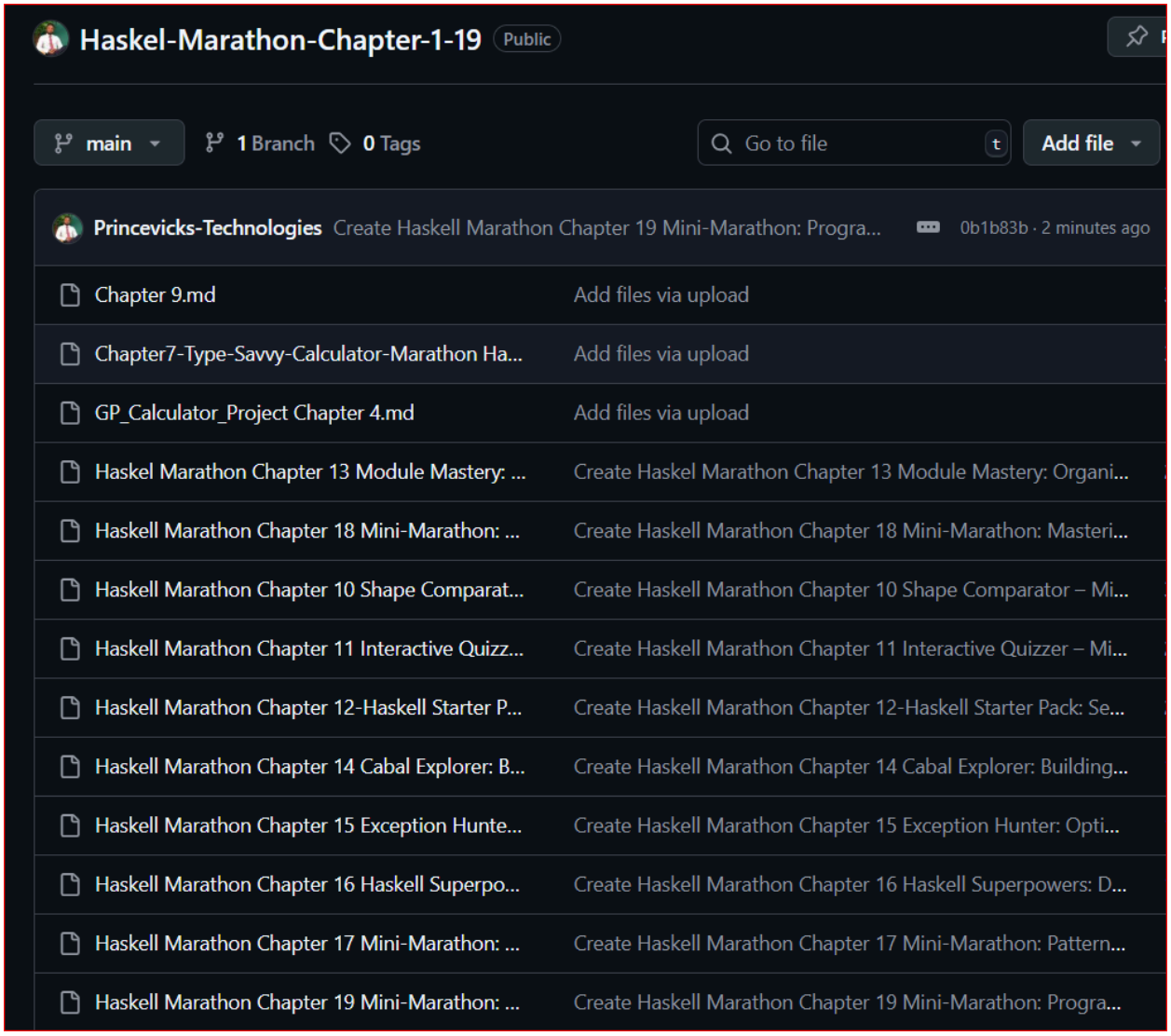
Repository link: https://github.com/Princevicks-Technologies/Haskel-Marathon-Chapter-1-19

Haskell chapters 1-15 GitHub uploads

Name	Last commit message
.gitignore	Initial commit
HC-13 Tasks	Create HC-13 Tasks
HC-14 Task.hs	Create HC-14 Task.hs
HC-15 Task.hs	Create HC-15 Task.hs
HC10-Tasks.hs	Create HC10-Tasks.hs
HC11-Tasks.hs	Create HC11-Tasks.hs
HC12-Tasks.hs	Create HC12-Tasks.hs
HC1Tasks.hs	Add files via upload
HC2-Tasks.hs	Rename HC-2.hs to HC2-Tasks.hs
HC3-Tasks.hs	Rename HC3Tasks.hs to HC3-Tasks.hs
HC4-Tasks.hs	Create HC4-Tasks.hs
HC5-Tasks.hs	Create HC5-Tasks.hs
HC6-Tasks.hs	Create HC6-Tasks.hs
HC7-Tasks.hs	Create HC7-Tasks.hs
HC8-Tasks.hs	Create HC8-Tasks.hs
HC9-Tasks.hs	Create HC9-Tasks.hs

Repository link: https://github.com/Princevicks-Technologies/Haskel-Marathon-Chapter-1-19

Hakel Marathon Chapters 1-19



Repository link: https://github.com/Princevicks-Technologies/Haskel-Marathon-Chapter-1-19