



## JUNE REPORT

AUTHOR: EZE-NWAOBASI ONYINYECHI PRINCESS

Facilitator at G-015

Haskell Plutus Developer at Wims Cardano...

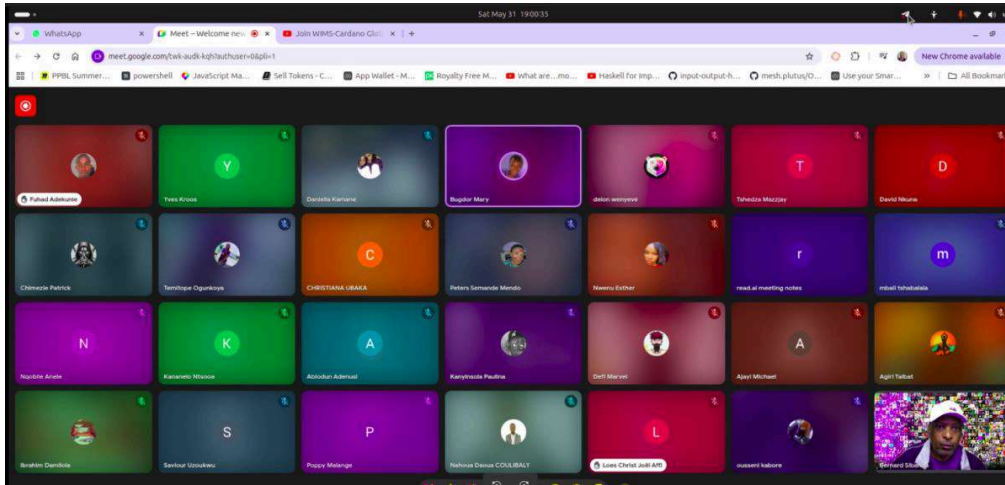
## Table of Contents

### Meetings:

|  |    |
|--|----|
| .....  | 2  |
| Facilitation (Group: G015) and general meetings                  | 2  |
| .....  | 2  |
| 6/2/25 Live session with Mr Bernard S and all New Members.....   | 2  |
| 6/6/25, First weekly meeting with student developers             | 2  |
| .....  | 2  |
| 6/9/25, Onchain Credentials for new students                     | 3  |
| .....  | 3  |
| 6/9/25, General meeting with Mr Bernard S and facilitators ..... | 3  |
| 6/12/25, General meeting with Mr Bernard S and facilitators..... | 3  |
| 6/12/25, Dev ex weekly meeting with Mr Bernard S.....            | 3  |
| 6/13/25, Enrollment of New Students                              | 4  |
| .....  | 4  |
| 6/18/25, General meeting with Mr Bernard S and facilitators..... | 5  |
| 6/23/25, General meeting with Mr Bernard and facilitators.....   | 5  |
| Developer Session And Source Codes.....                          | 11 |
| Multisig wallet:   | 11 |
| .....  | 11 |

## Facilitation (Group: G015) and general meetings

### 6/2/25 Live session with Mr Bernard S and all New Members:

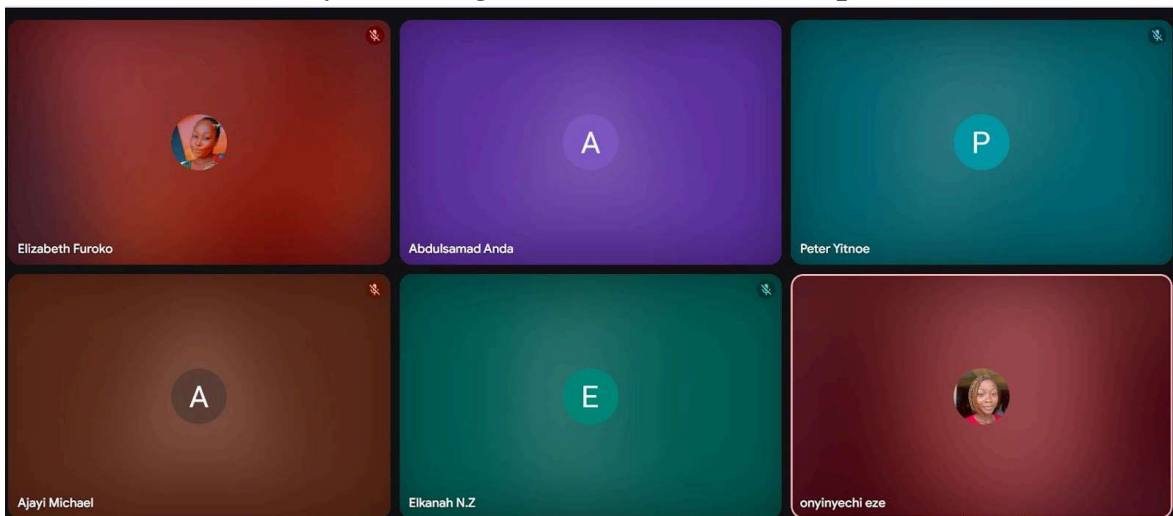


🕒 Time: 9:30–2:00PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAYS and SUNDAYS:

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

### 6/6/25, First weekly meeting with student developers:



🕒 Time: 8–9PM (Nigerian Time) | 9-10 PM (SAT)

🔗 Join us via the link below every FRIDAYS and SUNDAYS:

<https://meet.google.com/bwd-cnty-ump>

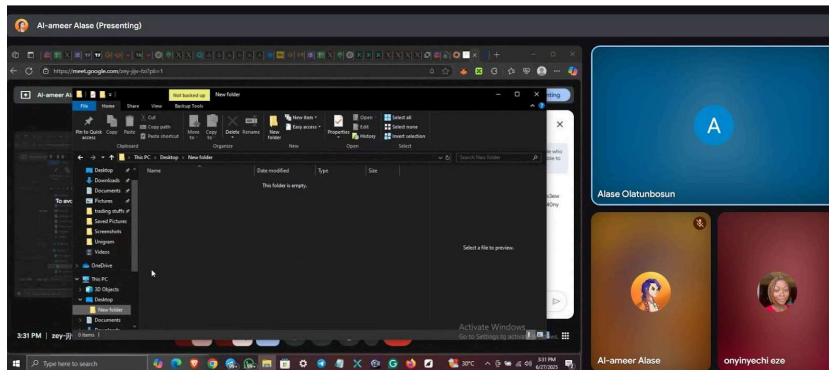
We started with Introduction to Haskell Programming Language, What are Functions, Examples of Haskell Codes, Job Opportunity

## Onchain Credentials

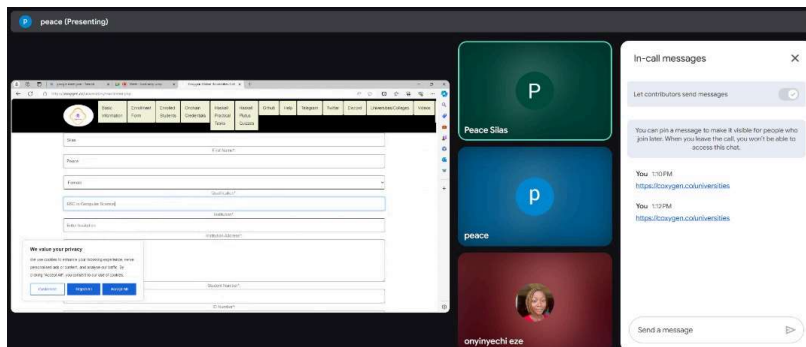
### 1. On the 26th June, 2025, Damilola Ibrahim did her Onchain credentials

| No | Task Code        | Task Title   |
|----|------------------|--|
| 1  | HC1102120250626  | HC1-1. Haskell Chapter 1 Practical Tasks   |
| 2  | HC5102120250626  | HC5-5. Haskell Chapter 5 Practical Tasks: Improving-and-combining-functions        |
| 3  | HC41102120250626 | HC4-4. Haskell Chapter 4 Practical Tasks - Pattern Matching in Functions           |
| 4  | HC31102120250626 | HC3-3. Haskell Chapter 3 Conditions-and-helper-constructions                       |
| 5  | HC21102120250626 | HC2-2. Haskell Chapter 2 Practical Tasks - Data types, Signatures and Polymorphism |

### 2. Alase Al-Ameer did his Onchain Credentials



### 3. 27th June, 2025 Peace Silas was enrolled and verified



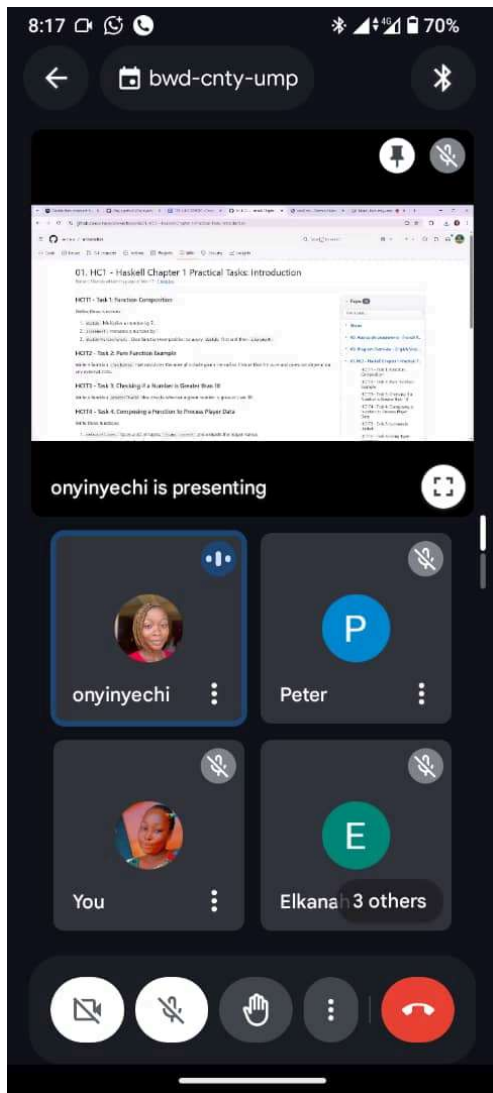
🕒 Time: 8–9PM (Nigerian Time) | 9-10 PM (SAT)

🔗 Join us via the link below every FRIDAYS and SUNDAYS:

<https://meet.google.com/bwd-cnty-ump>

#### 4. LIVE SESSION WITH G-015 STUDENTS: 27th June, 2025

We discussed Haskell Function Composition, Pure Functions and Real live coding sessions.

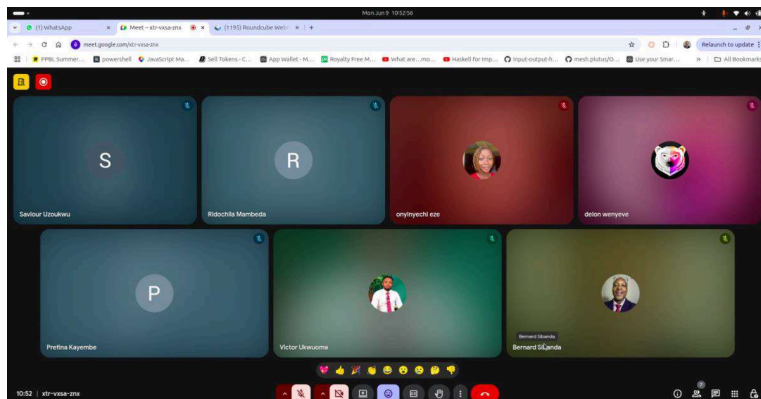


🕒 Time: 8–9PM (Nigerian Time) | 9-10 PM (SAT)

🔗 Join us via the link below every FRIDAYS and SUNDAYS:

<https://meet.google.com/bwd-cnty-ump>

## MEETING WITH MR BERNARD AND THE FACILITATORS ON 9th June, 2025

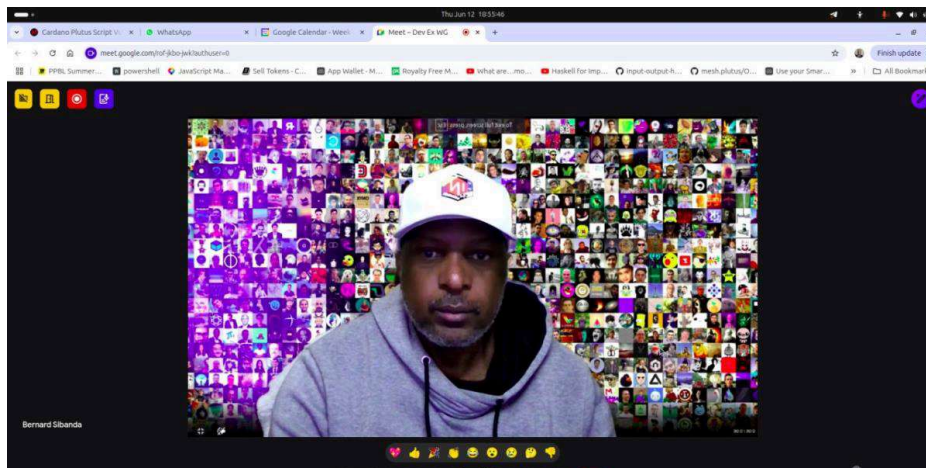


🕒 Time: 9:30–2:00PM (Nigerian Time) | 10:30-3:00 PM (SAT)

🔗 Join us via the link below every FRIDAYS and SUNDAYS:

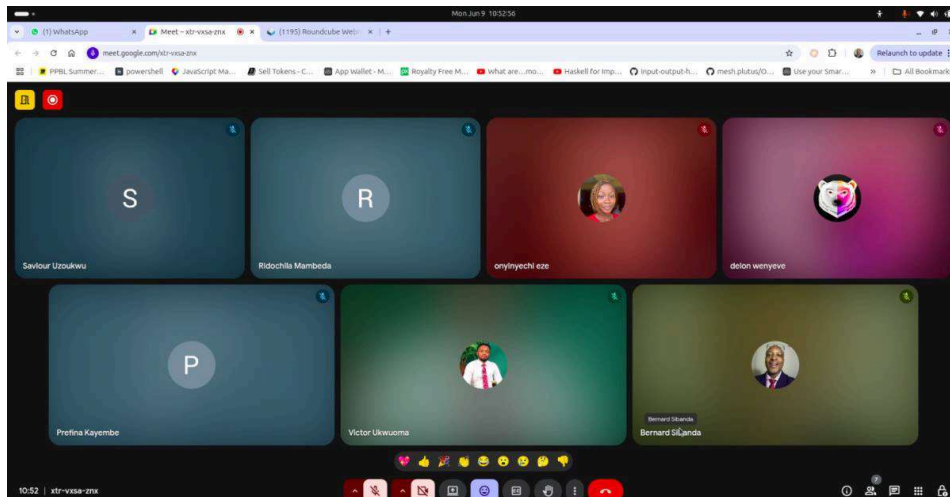
<https://meet.google.com/xtr-vxsa-znx>

## DEV EX WEEKLY MEETING WITH MR BERNARD ON 12th June, 2025





## MEETING WITH MR BERNARD AND THE FACILITATORS ON 18th June, 2025

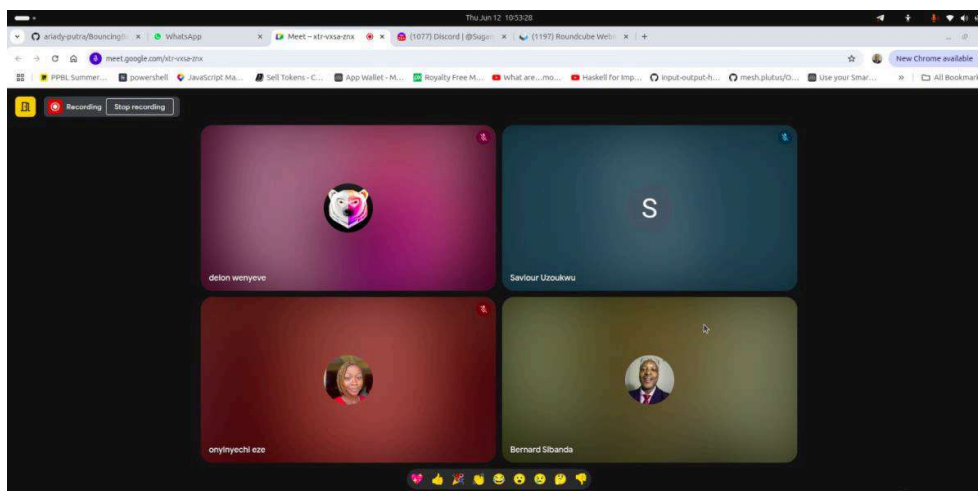


🕒 Time: 9;30–2:00PM (Nigerian Time) | 10:30-3:00 PM (SAT)

👁 Join us via the link below every FRIDAYS and SUNDAYS:

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

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

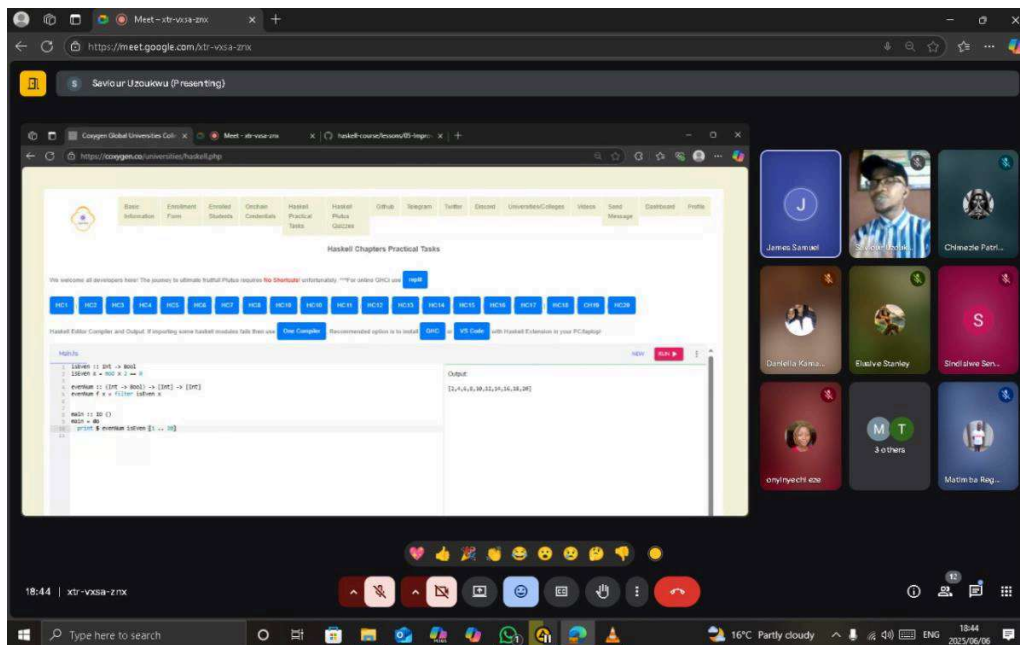


🕒 Time: 9;30–2:00PM (Nigerian Time) | 10:30-3:00 PM (SAT)

👁 Join us via the link below every FRIDAYS and SUNDAYS:

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

## LIVE SESSION WITH MR SAVIOUR AND STUDENT DEVELOPERS



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

👁 Join us via the link below every FRIDAYS and SUNDAYS:

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

## ENROLLMENT OF NEW STUDENTS

ID: 266 : Furoko Adebawale elizabeth  
Gender:No, Email Verified [Yes]

Qualification : **BSC IN ENGLISH  
LANGUAGE**

University/College : **COXYGEN  
GLOBAL UNIVERSITIES**

Address : **7 Gold Reef Rd 7  
Gold Reef Rd, Ormonde,  
Johannesburg South, 2091**

Date Joined : **2025-06-27**  
**10:58:35**

Wallet Address  
Reminder(ID:266)

ID: 262 : Anda Abdulsamad  
Gender:Yes, Email Verified [Yes]

Qualification : **BSC IN  
COMMUNICATION**

University/College : **UNIVERSITY OF ABUJA**

Address : **SF5 WUSE2**  
Date Joined : **2025-06-25**  
**22:17:39**

Wallet Address  
Reminder(ID:262)

ID: 253 : Zokdat Elkanah  
Gender:Yes, Email Verified [Yes]

Qualification : **UNDERGRADUATE**  
University/College : **PLATEAU  
STATE UNIVERSITY, BOKKOS,  
PLATEAU STATE**

Address : **Bokkos, Plateau  
State**  
Date Joined : **2025-06-21**  
**10:59:39**

Wallet Address  
Reminder(ID:253)



## ACHIEVEMENTS

1. Nine (9) Students have been clarified
2. I studied haskell up till Chapter 7






## Developer Session And Source Codes

### 1. Onchain Governance Marathon

**Description:** a hands-on coding challenge focused on building a basic decentralized governance system using Haskell <https://github.com/Onyinyechi46/Haskell-Basics>

#### Goal

Build a lightweight DAO voting engine in Haskell that supports:

-  Onchain-style proposal creation and voting logic
-  Vote tracking per proposal (Yes/No)
-  A voting deadline or timer for each proposal
-  Prevention of late votes after the timer expires
-  Execution logic based on vote results (approve or reject)

#### (B) Haskell Governance Timer

**Description:**It provides a precise, customizable countdown utility that can be used during proposal deliberation periods, voting sessions, or task windows in decentralized autonomous organizations (DAOs) or blockchain governance platforms.

<https://github.com/Onyinyechi46/Haskell-Basics/blob/main/Marathons/Haskell%20Governance%20Timer>

### 2. Mini-Wallet

A simple command-line wallet sign-up system built with Haskell. Users can create wallets with unique usernames and passwords, demonstrating basic I/O, data handling, and user interaction in Haskell. [http://commerce\\_://github.com/Onyinyechi46/Mini-Wallet](http://commerce_://github.com/Onyinyechi46/Mini-Wallet)

### 3. Cardano-Ecommerce-Sim

**Description:** A Haskell-based Cardano e-commerce simulation where users buy products with ADA, enjoy limited-time discounts, and receive delivery confirmations. Features smart contract-style logic like balance validation, timed offers, and instant transaction feedback

<https://github.com/Onyinyechi46/Cardano-Ecommerce-Sim>

### 4. taskFlow

**Description:** A simple To Do List manager written entirely in Haskell

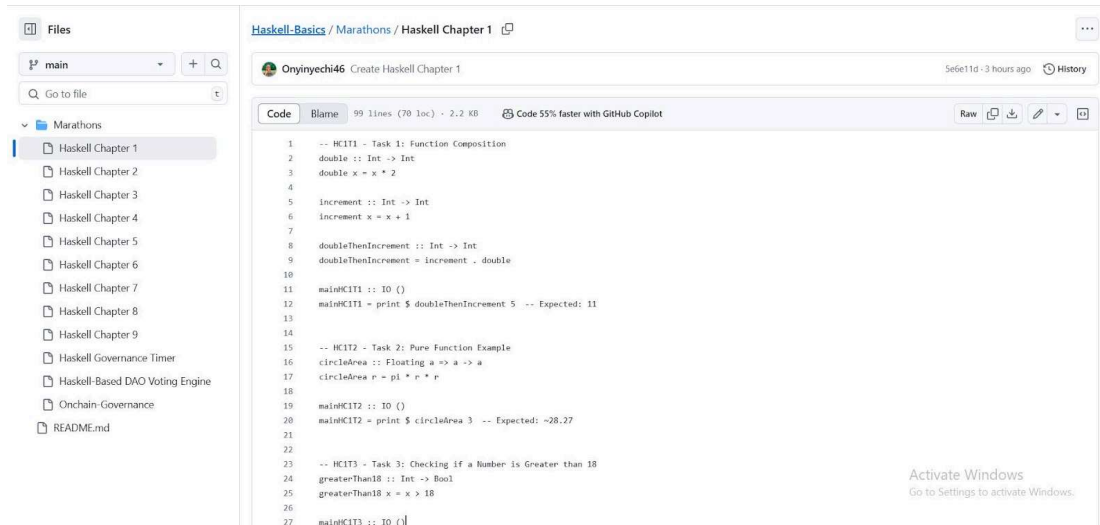
The screenshot shows the taskFlow web application interface. At the top, it says "Welcome, Onyinyechi Eze" with a taskFlow logo and the date/time "6/29/2025, 4:34:20 PM". Below this is a "New task..." input field. A "Pick due date:" section features a calendar grid with days of the week (Sun-Sat) and dates (1-30). A blue "Add Task" button is positioned below the calendar. The task list at the bottom contains three items: "Onchain credentials" (Due: 2025-06-01), "Student enrollment" (Due: 2025-06-10), and "Session with Mr Bernard" (Due: 2025-06-15). Each item has a red 'x' icon for deletion.

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |
| 29  | 30  |     |     |     |     |     |

- **Onchain credentials** ✕  
Due: 2025-06-01
- **Student enrollment** ✕  
Due: 2025-06-10
- **Session with Mr Bernard** ✕  
Due: 2025-06-15

<https://github.com/Onyinyechi46/TaskFlow>

## 5. Haskell Basics



```
1  -- HC111 - Task 1: Function Composition
2  double :: Int -> Int
3  double x = x * 2
4
5  increment :: Int -> Int
6  increment x = x + 1
7
8  doubleThenIncrement :: Int -> Int
9  doubleThenIncrement = increment . double
10
11 mainKC111 :: IO ()
12 mainKC111 = print $ doubleThenIncrement 5 -- Expected: 11
13
14
15 -- HC112 - Task 2: Pure Function Example
16 circleArea :: Floating a => a -> a
17 circleArea r = pi * r * r
18
19 mainKC112 :: IO ()
20 mainKC112 = print $ circleArea 3 -- Expected: ~28.27
21
22
23 -- HC113 - Task 3: Checking if a Number is Greater than 18
24 greaterThan18 :: Int -> Bool
25 greaterThan18 x = x > 18
26
27 mainKC113 :: IO ()
```

<https://github.com/Onyinyechi46/Haskell-Basics/blob/main/Marathons/Haskell%20Chapter%201>

**THANK YOU**