November Report

Facilitator: Name: Ukwuoma Victor
Facilitator Group: G-0014
Profession: Haskell Plutus Developer, Technical Writer
Number of Students: Ten(10) Students from Nigeria
Date Submitted: Monday, 1st December, 2025

November 2025 REPORT

Table of Contents

Executive Summary

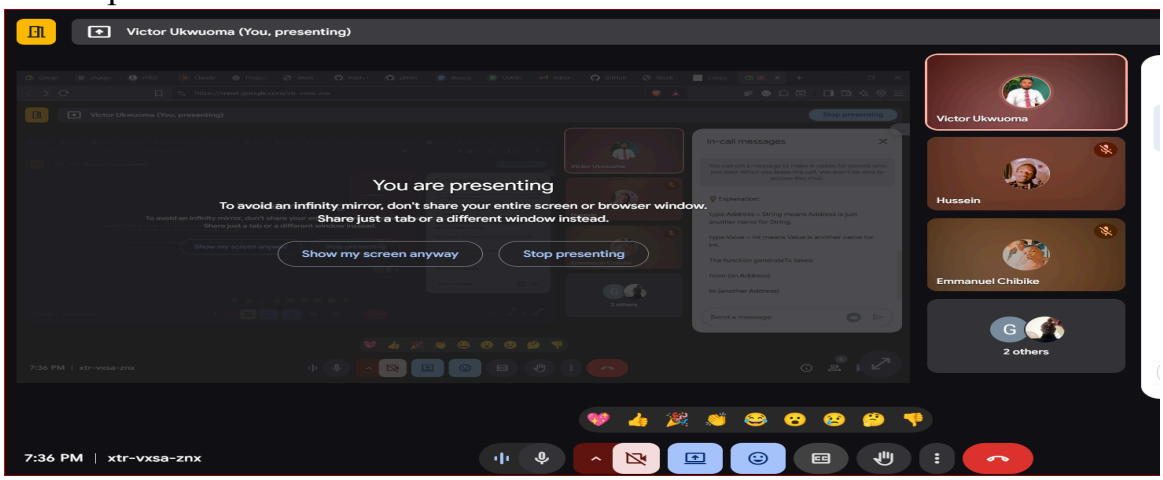
In the month of November, we made steady progress in our classes. Much of the focus shifted toward beginner sessions as we welcomed several new members. These foundational classes helped introduce core principles, ensured proper understanding from the start, and created a strong base for future lessons. Overall, November strengthened our learning structure, integrated new participants effectively, and positioned us for more advanced teaching in the coming weeks.

I made major progress in Haskell and Plutus smart contract development. I created and improved several Plutus V2 contracts, including escrow, voting, ticketing revenue, batch auction, limit order, liquidity mining, swap escrow, launchpad, protocol fee router, and different loan contract versions. These contracts included full datums, redeemers, validator logic, signature checks, expiry handling, and ADA value transfers, using consistent Plutus imports and serialization tools.

I significantly expanded my GitHub repository by adding more than fifty new files, organizing them into structured folders such as `src`, `tests`, `assets`, and `lecture`. I resolved cabal dependency errors, merge conflicts, and path issues while building and cleaning the project inside the Nix environment. I made many commits and successfully pushed large updates.

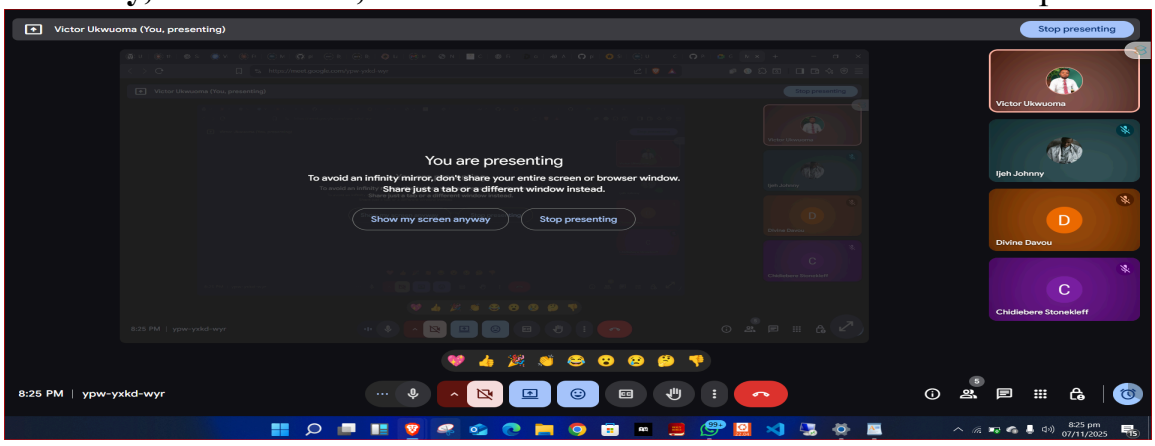
1.0 Haskell Group 14 Facilitation Report

1. **Thursday, November 6, 2025:** Live Session with G-014 Student Developers



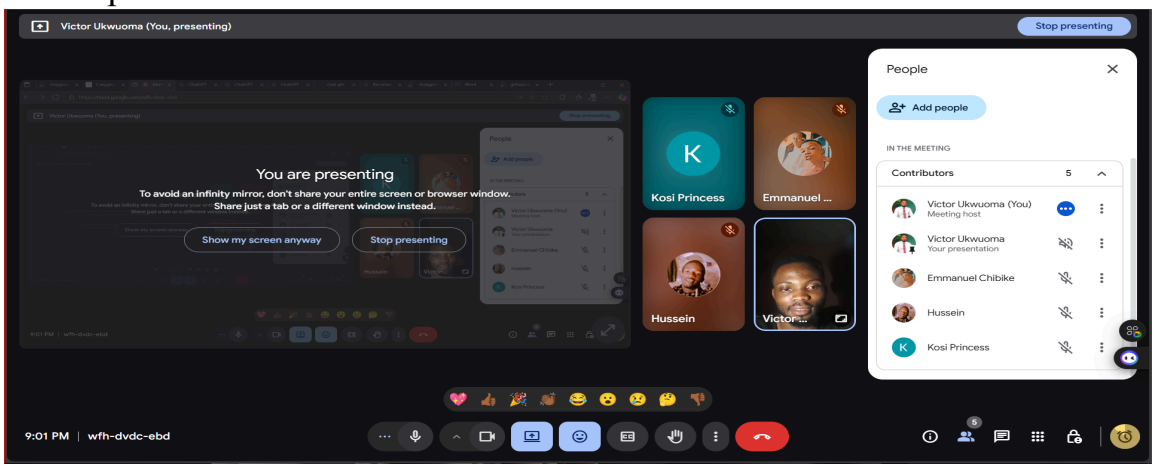
🕔 Time: 8–9PM (WAT)
🔗 Join us via the link below every Sunday, Teusday, Thursday and Friday:
https://meet.google.com/wfh-dvdc-ebd
Attendance: Emmanuel, Cynthia, John, Divine, and Hussein
Activities: We revisited key Haskell foundations to strengthen understanding for both new and continuing learners, covering basic syntax, functional concepts, core data types, and essential programming patterns for clearer, more confident progress.

2. **Friday, November 7, 2025:** Live Session with G-014 Student developers



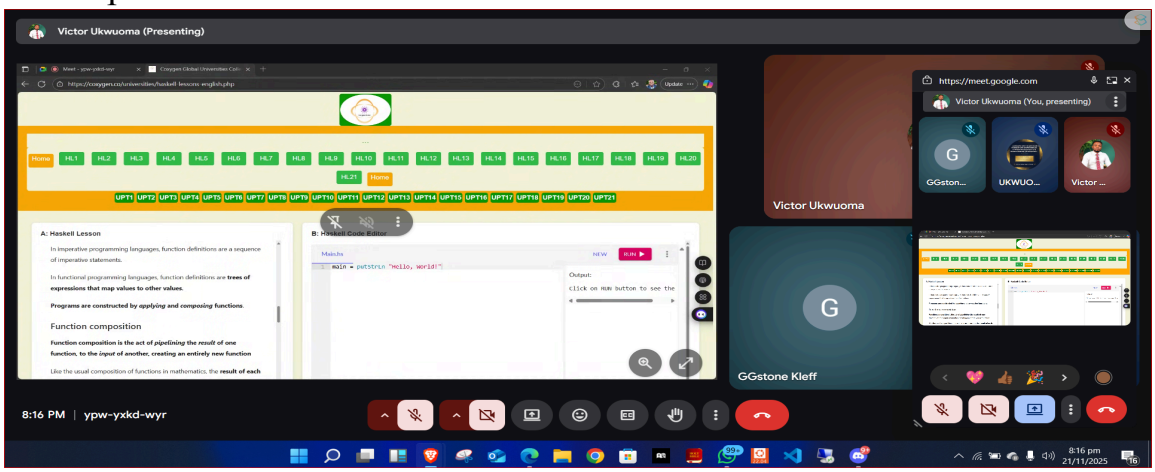
🕔 Time: 8–9PM (WAT)
🔗 Join us via the link below every Thursday and Friday:
https://meet.google.com/wfh-dvdc-ebd
Attendance: Emmanuel, Hussein, Alabi, John and Aishat, Divine
Activities: We held live sessions exploring core Haskell data types, Integer, Double, Float, Lists, and ordering, using clear, practical examples to help beginners understand value behavior and foundational functional programming principles.

3. Sunday, November 16, 2025: Live Session with G-014 Student Developers



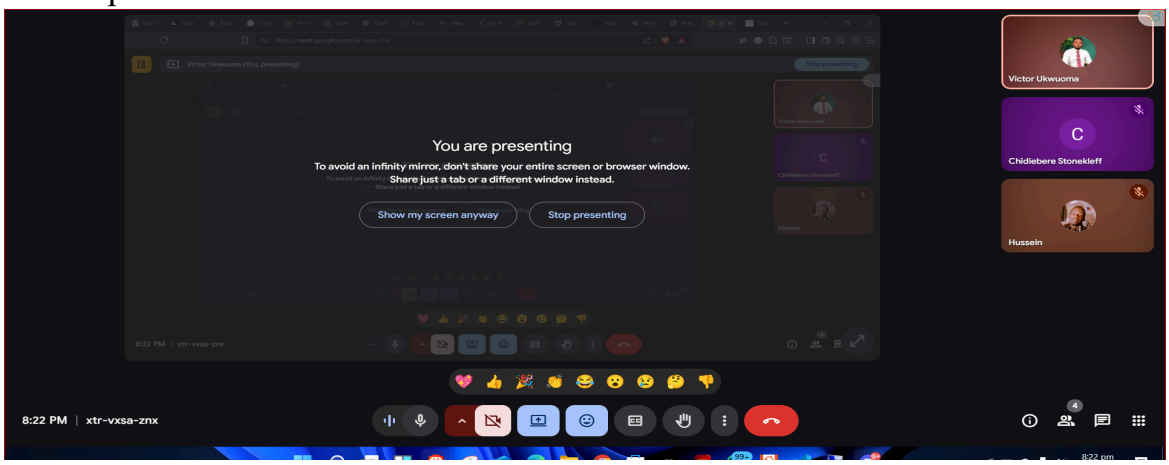
🕔 Time: 8–9PM (WAT)
🔗 Join us via the link below every Thursday and Friday:
https://meet.google.com/wfh-dvdc-ebd
Attendance: Cynthia, Hussein, John, Divine and Emmanuel.
Activities: We revisited the Integral, Floating, Read, and Show typeclasses, gaining a deeper understanding of their roles and applications.

4. Friday, November 21, 2025: Tutorial Session with G-014 Student Developers.



🕔 Time: 8–9PM (WAT)
🔗 Join us via the link below every Thursday and Friday:
https://meet.google.com/wfh-dvdc-ebd
Attendance: Aishat, Cynthia, Divine and John.
Activities: We looked at the most general valid type Class and Constraints for multiple type variables

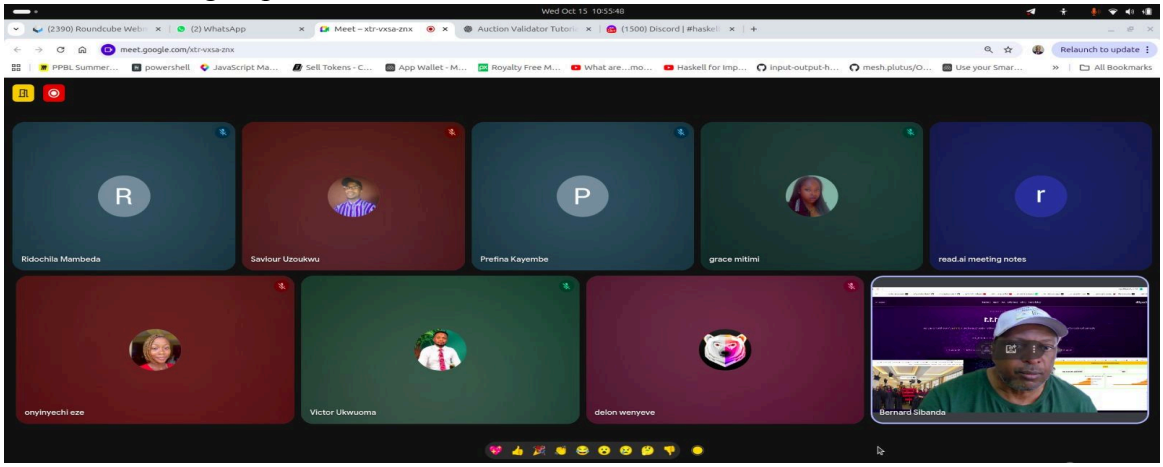
5. Friday, November 26, 2025: Tutorial Session with G-014 Student Developers



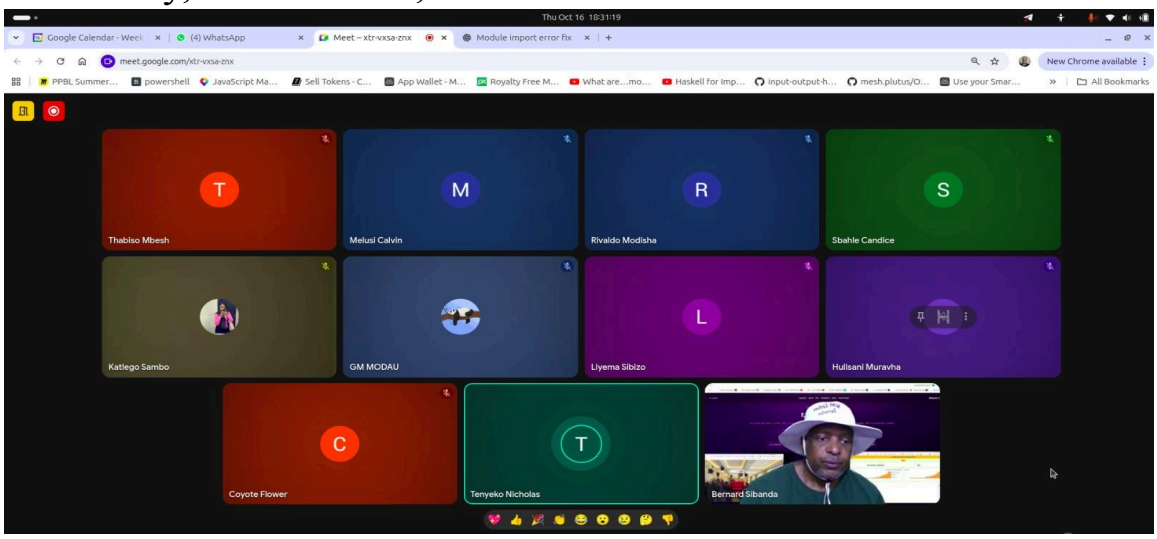
🕔 Time: 4–5PM (WAT)
🔗 Join us via the link below every Thursday and Friday:
https://meet.google.com/wfh-dvdc-ebd
Attendance: Divine, Cynthia, John.
Activities: We again held live sessions exploring core Haskell data types—Integer, Double, Float, Lists, and ordering—then applied these concepts functionally, demonstrating practical uses and reinforcing foundational functional programming principles for beginners.

2.0 Facilitators Meeting with Mr. Benard S.

1. Monday, November 3, 2025: Meeting with Mr Benard S.
We deliberated on the attendance of the students Developers to the EBU training. We created the 2nd group for Fascilitators

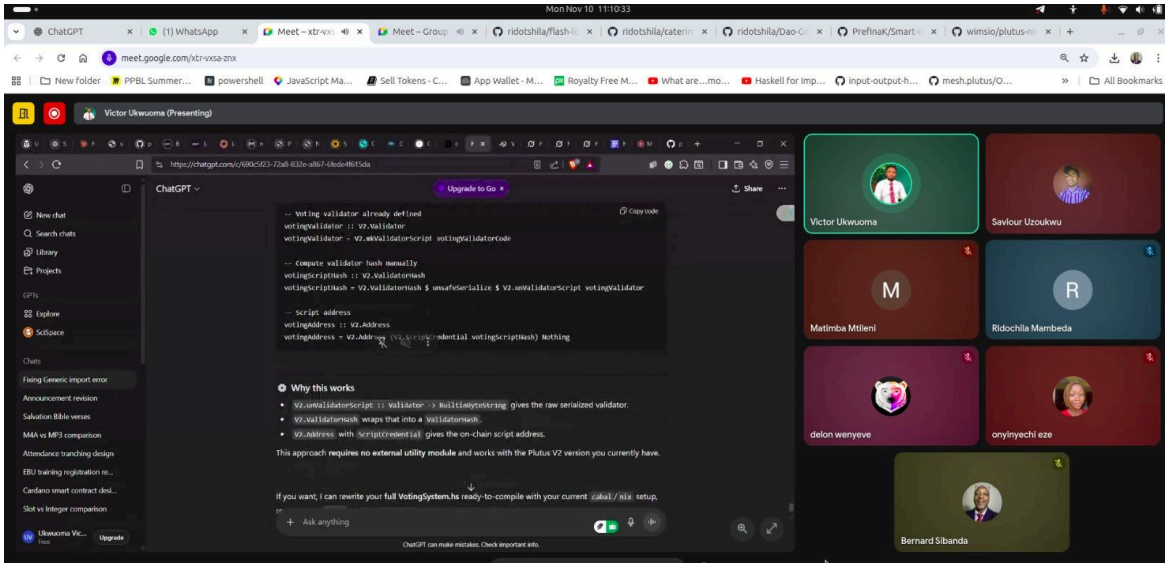


2. Teusday, November 4, 2025 Live session with Mr Banard S.



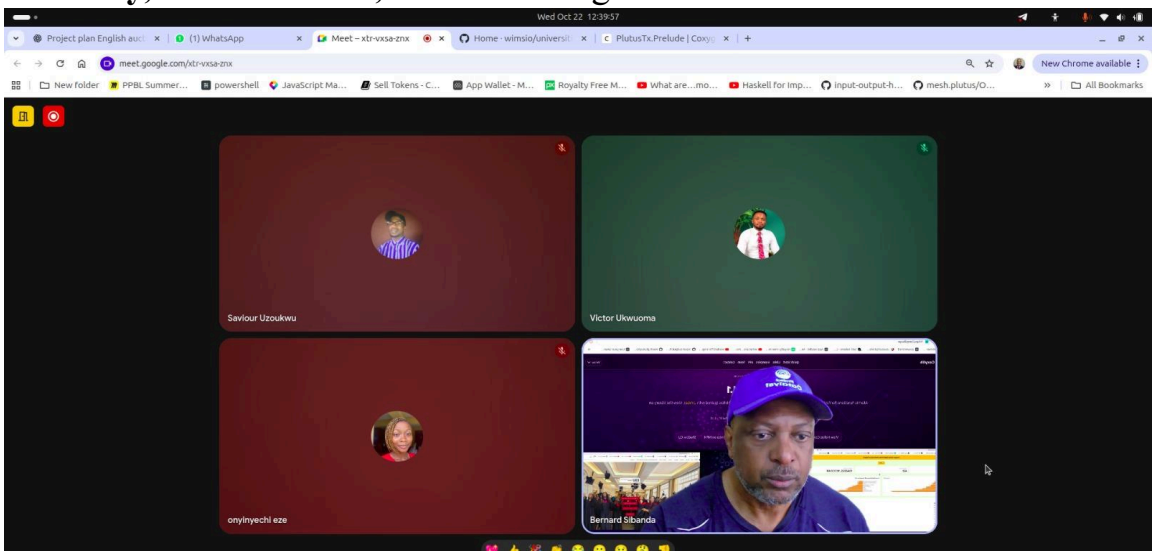
🕔 Time: 10:30AM (SAT)
🔗 Join us via the link below every Monday through Friday:
https://meet.google.com/xtr-vxsa-znx

3. Teusday, November 11, 2025. Meeting with Mr Benard S.



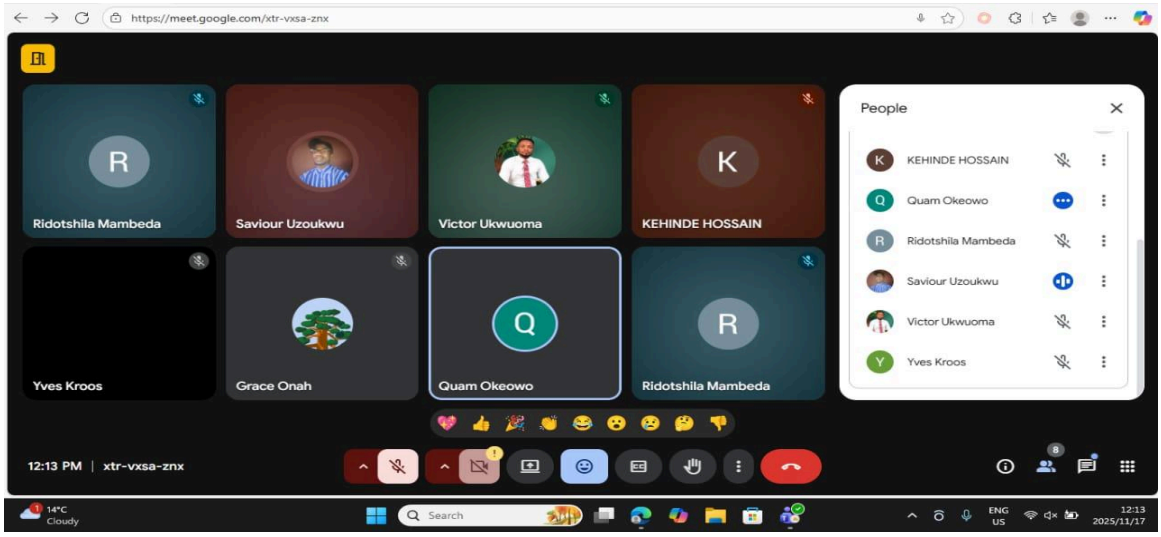
🕔 Time: 10:30AM (SAT)
🔗 Join us via the link below every Monday through Friday:
https://meet.google.com/xtr-vxsa-znx

4. Friday, November 14, 2025. Meeting with Mr Benard S.



🕥 Time: 10:30AM (SAT)
🔗 Join us via the link below every Monday through Friday:
https://meet.google.com/xtr-vxsa-znx

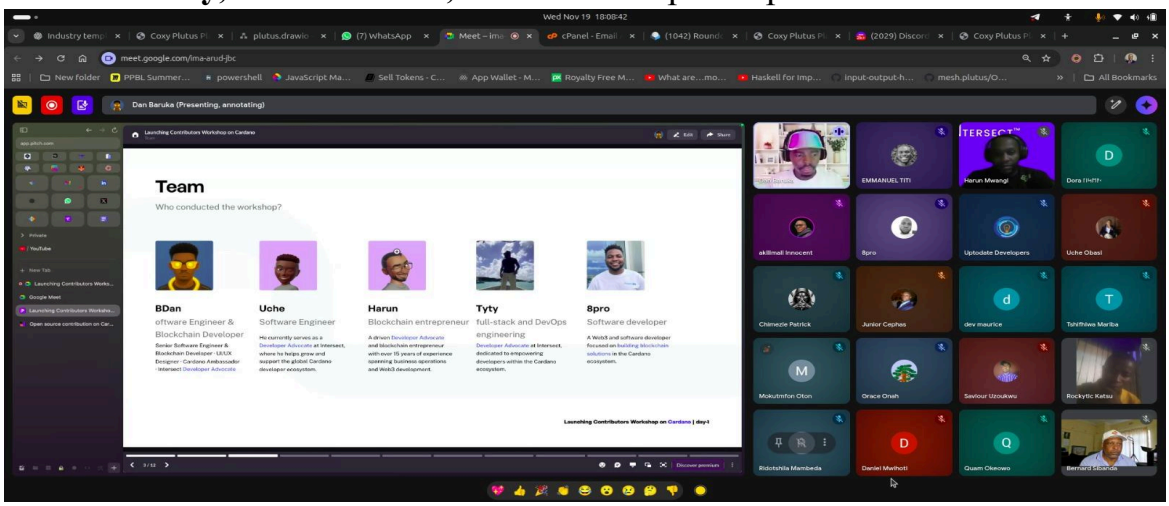
5. Monday, November 17, 2025: Live session with Mr Benard S



🕥 Time: 10:30AM (SAT)
🔗 Join us via the link below every Monday through Friday:
https://meet.google.com/xtr-vxsa-znx

3.0 Dev Ex Meeting Sessions

1. Wednesday, 19 November, 2025: Developer Experience WG



Dev Ex WG
Thursday, 2 November · 7:00 – 8:00pm
Time zone: Africa/Johannesburg
Google Meet joining info
Video call link: https://luma.com/join/g-PUogcQCbXejHUcp

4.0 Monthly Achievements

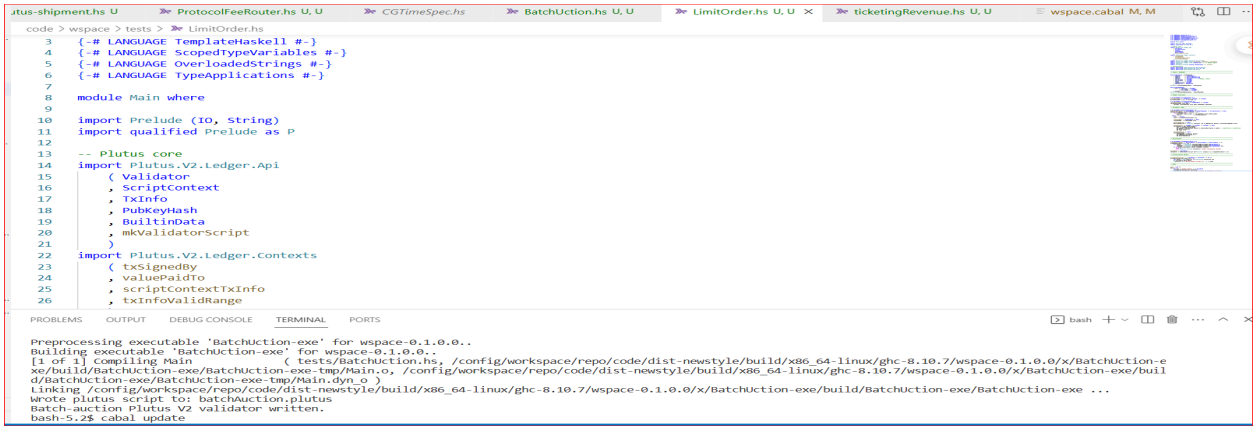
1. Student Recruitment & Enrollment.

In November, 2nd, I successfully recruited 2 students and enrolled them in the beginner Haskell course up to Chapter 2.

5.0 Surce code projects

1. PlutusBatchAuction

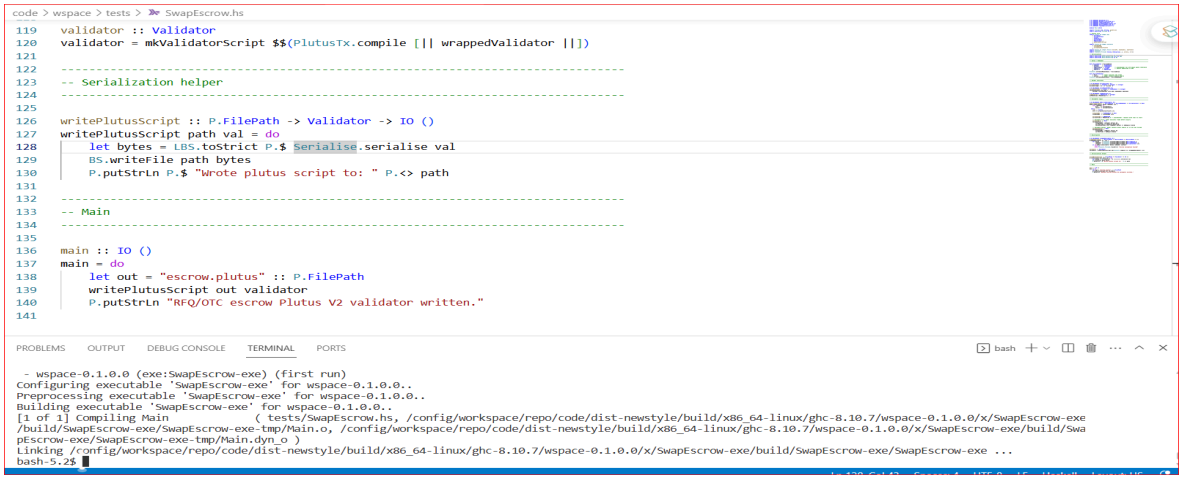
This project implements **PlutusBatchAuction**, a **Batch Auction smart contract** on Cardano using Plutus V2. It enables users to place buy or sell orders with specified quantities and prices. Orders can be **matched** at a clearing price or **canceled** by the owner. The contract enforces order constraints, prevents overfills, and ensures only the owner can cancel orders. The validator is serialized into a Plutus script for on-chain deployment, providing a secure, deterministic, and decentralized mechanism for batch trading.



GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/BatchUction.hs

2. PlutusEscrow

This project implements **PlutusEscrow**, a **decentralized escrow smart contract** on Cardano using Plutus V2. It facilitates secure transactions between a buyer and a seller, with an optional arbiter to handle disputes. The contract supports **mutual release**, **refunds after deadlines**, and **arbitration decisions**, ensuring funds are only transferred according to agreed rules. It enforces escrow states (`Pending`, `Released`, `Refunded`) and checks for proper signatures, deadlines, and transaction validity. The validator is serialized into a Plutus script, enabling deployment on-chain for reliable, deterministic, and tamper-resistant escrow operations.

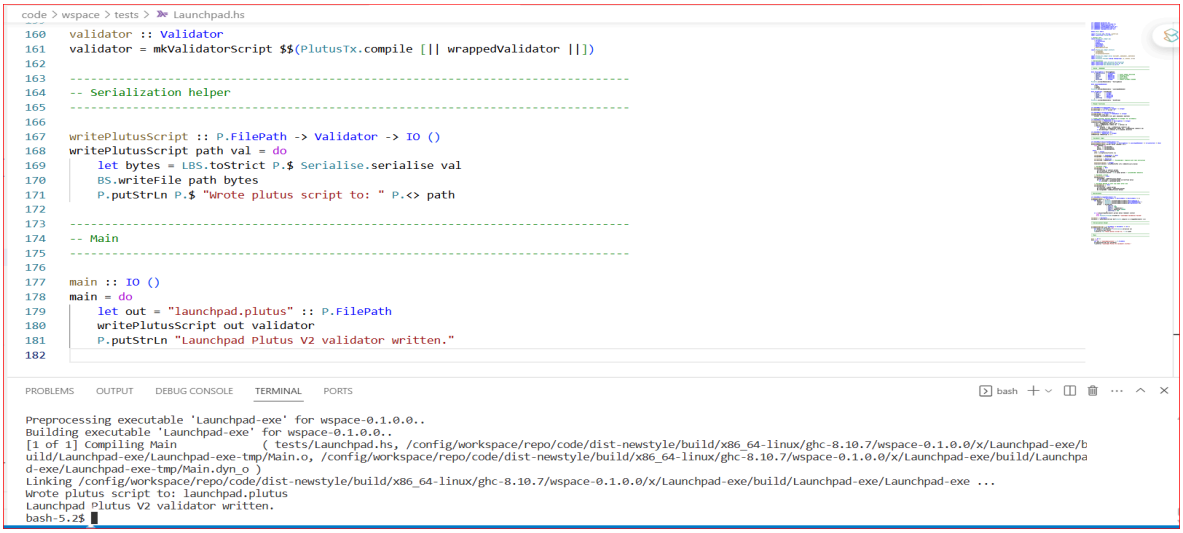


GitHub: Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/DisputeEscrow.hs

3. PlutusLaunchpad

`PlutusLaunchpad` is a **decentralized token launchpad and vesting smart contract** on Cardano using Plutus V2. It allows participants to **buy tokens during a sale period**, claim **linearly**

vested tokens over time, and request **refunds** if the soft cap is not met. The contract enforces strict sale rules using `SaleParams` for price, cap, start/end times, and soft cap. Vesting is tracked with `VestingDatum`, ensuring that beneficiaries can only claim what has been unlocked according to a cliff and linear schedule. Transactions are validated by checking signatures, current time, vesting states, and contribution amounts. This script ensures **secure, automated, and deterministic token sales and vesting schedules** on-chain.



GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/Launchpad.hs

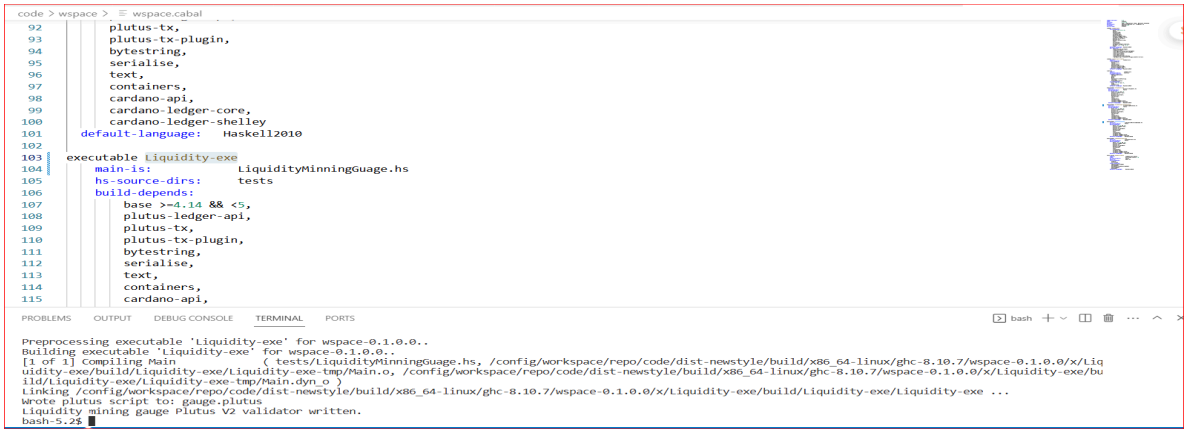
4. PlutusLimitOrder

Description:
`PlutusLimitOrder` is a **decentralized on-chain limit order validator** for Cardano. It allows users to create **buy and sell orders** with specified price limits and quantities. Orders are valid until a defined **deadline**, ensuring trades occur within a specific time window. The contract enforces that **execution prices meet or exceed the order limits**, and it prevents over-filling of orders. Order owners can also **cancel unfilled orders** before expiry. This ensures **secure, deterministic, and automated matching of limit orders** directly on-chain, without relying on intermediaries.

GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/LimitOrder.hs

5. PlutusLiquidityGauge

Description:
`PlutusLiquidityGauge` is a **decentralized staking and rewards contract** for Cardano liquidity providers. Users can **stake LP tokens**, **unstake**, and **claim accrued rewards** based on their contribution and elapsed time. The contract tracks each LP's position, staked amount, last update time, and rewards accrued. Admins can **adjust the reward rate** per epoch to incentivize liquidity. Rewards are calculated proportionally to each staker's share of the total pool weight, ensuring **fair and deterministic distribution**. This validator enables **secure, automated, on-chain liquidity mining**, supporting tokenized incentives in DeFi ecosystems.

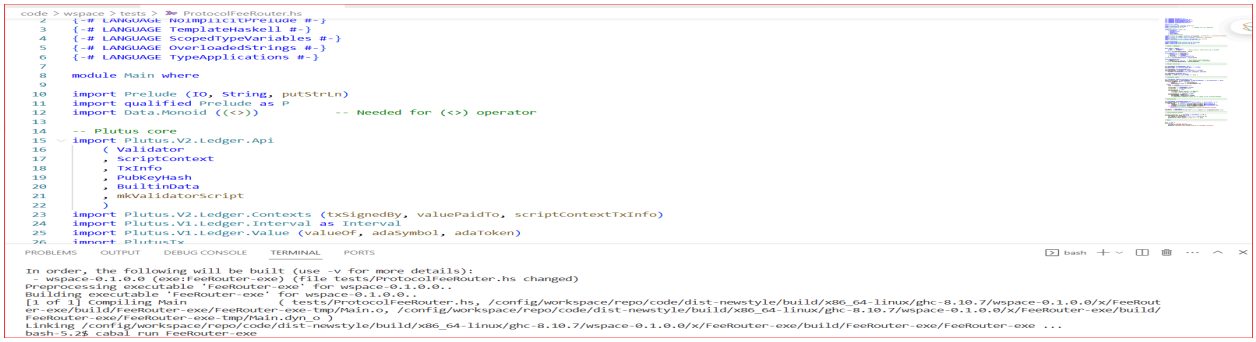


GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/LiquidityMinningGuage.hs

6. PlutusFeeRouter

Description:
`PlutusFeeRouter` is an **on-chain protocol fee router** for the Cardano blockchain. It allows the **automated distribution of collected fees** to multiple recipients according to predefined **basis point splits**. Key features include:

Controlled distribution: Only the `controller` can update fee splits.
Timelocked updates: Supports future enhancements to enforce time-based restrictions on changes.
Deterministic splits: Ensures total basis points sum to 10,000 (100%) for accurate distribution.
Versioned contract: Supports version tracking and upgradeability via `version` and `routerNFT`.
This Plutus V2 validator enables **secure, trustless, and automated fee routing** within DeFi protocols, reducing manual intervention and ensuring correct, on-chain accounting for rewards or protocol fees.



GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/ProtocolFeeRouter.hs

7. PlutusShipmentEscrow

Description:
PlutusShipmentEscrow is an on-chain milestone-based shipping escrow contract for the Cardano blockchain. It facilitates secure, automated payments for shipments tied to milestone completion.
Key features include:

- Milestone-based fund release: Carrier receives payment progressively as each shipment milestone is submitted and approved.
- Custodian oversight: A custodian initializes funding and can reclaim funds if milestones are incomplete after the deadline.
- Deadline enforcement: Ensures payments or reclamation actions occur only within valid timeframes.
- Escrowed funds: Total payment is held on-chain, reducing risk for both shipper and carrier.
- Traceable progress: Shipment progress is tracked on-chain via milestone indexing.

This validator enables trustless, automated, and verifiable shipment payments, ensuring carriers are paid fairly while shippers retain control over escrowed funds until milestones are met.

GitHub Link:
https://github.com/Princevicks-Technologies/plutus-nix/blob/main/code/wspace/tests/plutus-shipment.hs

Challenges

1. Impact of Examinations on Participation
Additionally, some of my students had examinations, which affected their participation in our weekly classes. To support them, I organised make-up sessions to ensure that those who missed classes were properly guided and brought up to speed.

Students onchain progress

No	Last Name	First Name	Github username	Onchain Progress
1	Mowoe	Novo	https://github.com/Novo-Mowoe	5
2	Hussein	Sulaiman	https://github.com/Hussman256	22
3	Maikudi	Alhamadu	https://github.com/Jiggycoinz	5
4	Aishat	Akingbade	https://github.com/aishat-akingbade	5
5	Chibuike	Emmanuel	https://github.com/surflex16	10
6	Oyinlola	Timilehin	https://github.com/Timskid11	-
7	Anjorin	Adeniyi	https://github.com/Hardhenhiyi	-
8	Eze	Emeka	https://github.com/CbmloArt	5
9	Ezeaku	Cynthia	https://github.com/cynthiakosi110	10
10	Eniola	Temitope	https://github.com/Temmy1744	5
11	Iwunosa	Jegson	https://github.com/iwunosaJIN	-
12	Alabi	Stephen	https://github.com/alabistephen	
13	Muhammad	Abdulhamid	https://github.com/hamid	-
14	Odey	Miracle	https://github.com/odeymiracle	10
15	Kasis	New		-
16	Davou	Divine		-
17	John	Ijeh	https://github.com/Johnijeh	-
18	Fuhad	Mohammadu		-
19	Kelvin	Polycarp	https://github.com/Johvialkevin	10