COMP 4541 Individual Project

Lottery DApp

Student Name: Tsang Hing Ki

Student ID: 20865468

Background

This report details the COMP4541 Lottery DApp, a decentralized application developed to emulate the core mechanics and appeal of popular lottery systems, with specific reference to the Hong Kong Mark Six. The Mark Six is a well-established and widely participated-in lottery in Hong Kong, known for its classic 6-number draw format from a pool of 49, plus an additional "extra number," and its ability to generate significant community interest and substantial jackpots. This project seeks to translate that familiar and engaging experience into a decentralized environment. By leveraging blockchain technology and smart contracts, the DApp aims to address inherent limitations of traditional, centralized lotteries by offering enhanced transparency in all operations, provably fair random number generation for draws (utilizing Chainlink VRF), increased security of prize pools, and an auditable, immutable record of all transactions and outcomes. The goal is to provide a more trustworthy, accessible, and modern platform for lottery participation.

Lottery Rules

The decentralized lottery operates under clearly defined regulations for participation and prize distribution. Participant entry involves acquiring a ticket at a fixed cost, for which they select six unique integers from a pool of one to forty-nine. The determination of winning numbers, six main and one supplementary 'extra' number, is conducted through a verifiably random and fair process, ensuring integrity. The total prize fund for each draw instance is an aggregation of revenues from all tickets sold for that round, less a predetermined ten percent deducted as an operational commission, and augmented by any unclaimed prize money carried forward from previous draws. This net prize fund is then distributed among winners according to a predefined schedule of prize tiers, with specific percentage allocations from the total prize fund as follows:

- Matching all six main numbers awards 55% of the prize pool.
- Matching five main numbers plus the supplementary extra number awards 25%.
- Matching five main numbers alone awards 10%.
- Matching four main numbers plus the supplementary extra number awards 7%.
- Matching four main numbers alone awards 6%.
- Matching three main numbers plus the supplementary extra number awards 4%.
- Matching three main numbers alone awards 3%.

Should multiple participants achieve the criteria for a specific prize tier, the allocated sum for that tier (derived from its designated percentage of the total prize fund) is divided equally amongst them. In instances where no tickets match the criteria for the highest prize categories, the corresponding prize money is added to the fund for the subsequent draw, creating a rollover effect. Winners are subsequently able to initiate a claim for their awarded cryptocurrency.

Lottery Implementation

The Lottery DApp operates on a structured, cyclical protocol designed to ensure fairness, transparency, and adherence to predefined rules, heavily inspired by the Mark Six lottery format. The mechanism can be dissected into several distinct phases:

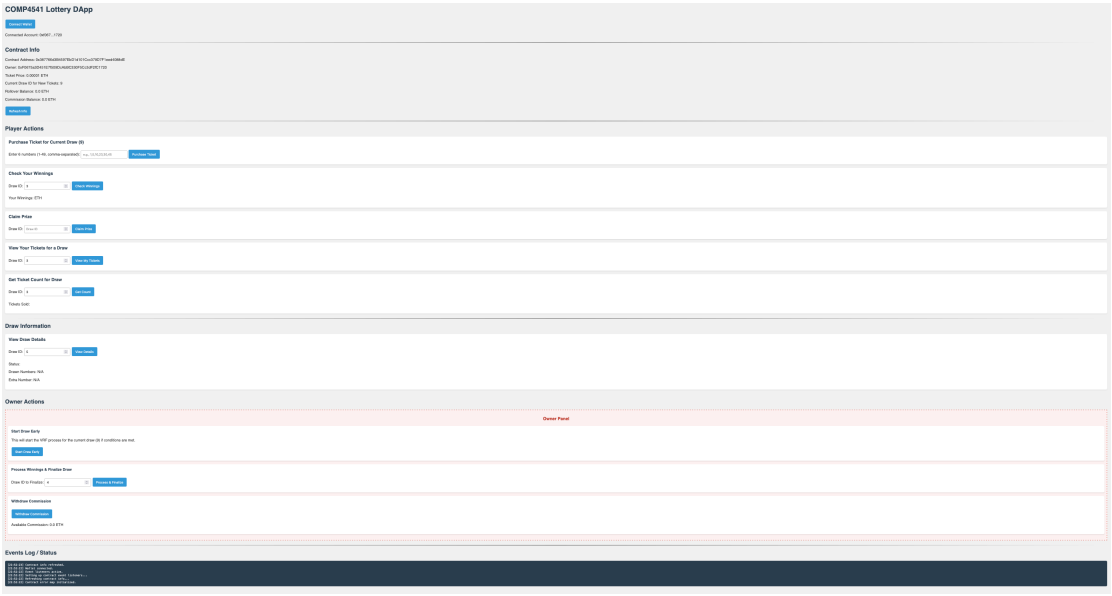
1. **Participation and Ticket Acquisition:**
   Participants engage in the lottery by acquiring digital tickets for the *currentDrawId* (the identifier for the ongoing or next upcoming draw). This acquisition is facilitated by a smart contract function (*purchaseTicket*) requiring the participant to remit a predetermined *ticketPrice* in Ether (ETH). Concurrently, the participant must submit a selection of six distinct integers, chosen from a valid range of 1 to 49. The smart contract enforces the uniqueness of these numbers and their adherence to the specified range, rejecting invalid submissions. Each valid purchase is recorded on the blockchain, associating the player's address and their chosen numbers with the specific *currentDrawId*.
2. **Draw Initiation and Random Number Generation (RNG):**
   The draw process, which determines the winning numbers, is triggered either by a predefined time interval (*i_drawInterval*) managed by Chainlink Automation (*checkUpkeep* and *performUpkeep*) or, under specific conditions, can be initiated prematurely by the contract owner (startDrawEarly). Critically, the selection of winning numbers is not performed by a centralized entity. Instead, the DApp integrates Chainlink's Verifiable Random Function (VRF V2Plus). Upon draw initiation, the smart contract requests random numbers from the VRF Coordinator. The VRF service then generates a cryptographically secure set of random values, which are subsequently fulfilled back to the lottery smart contract (*fulfillRandomWords*). This process yields six main winning numbers and one "extra number," all derived from the VRF output, ensuring provable fairness and tamper-resistance. The status of the draw transitions to reflect that VRF has been requested and then fulfilled (*DrawStatus.VRF_REQUESTED*, *DrawStatus.VRF_FULFILLED_NUMBERS_SET*).

3. **Winnings Determination and Prize Pool Distribution:**
   Once the winning numbers (6 main + 1 extra) are set for a given *drawId*, the *processWinningsAndFinalizeDraw* function (callable via Chainlink Automation or by any address once conditions are met, or by the owner) calculates the winnings. The total prize pool for a draw consists of the revenue from ticket sales for that draw (less a predefined *COMMISSION_PCT* which is allocated to the *commissionBalance*) plus any *rolloverBalance* carried forward from previous draws where top-tier prizes were not won.
   A tiered prize structure, defined within *prizeTierRules*, dictates payouts. Participants' tickets are compared against the drawn numbers. Winnings are awarded based on the number of main numbers matched and, for certain tiers, whether the extra number was also matched. For example, matching all six main numbers typically corresponds to the highest prize tier. The smart contract iterates through all purchased tickets for the draw, identifies winners for each prize tier, and records their entitlements (winnings mapping). If no ticket matches the criteria for the highest prize tiers (or any tier that would exhaust the pool), the unallocated portion of the prize pool contributes to the *rolloverBalance* for the subsequent draw. The draw is then marked as *DrawStatus.RESULTS_CALCULATED* and *drawFinalized*.
4. **Prize Claim:**
   Participants holding winning tickets for a finalized draw can actively claim their prizes by calling the *claimPrize* function, specifying the relevant *drawId*. The smart contract verifies the claimant's entitlement based on the recorded winnings for their address and that *drawId*. If valid, the corresponding amount of ETH is transferred from the contract to the claimant's wallet, and their recorded winnings for that draw are zeroed out to prevent duplicate claims.

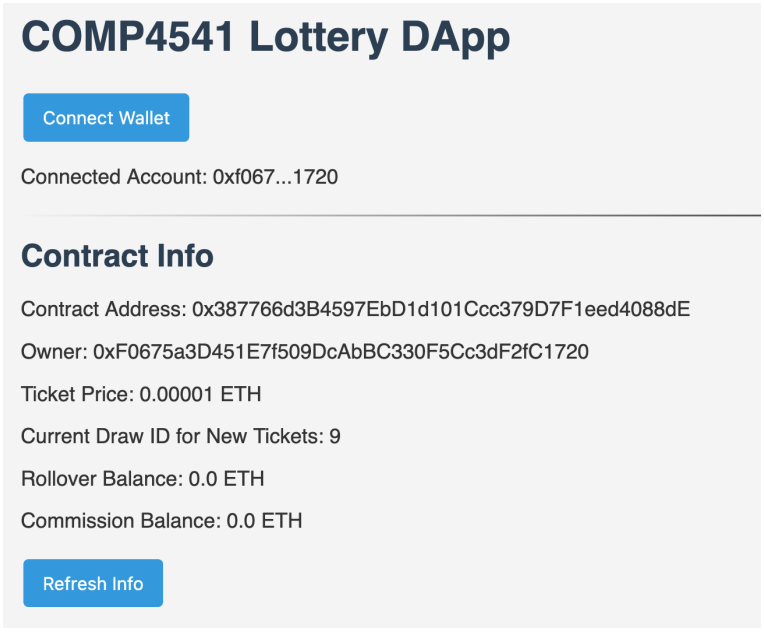
This cyclical process—ticket sales, VRF-based draw, winnings calculation, and prize claims—then repeats for the next *currentDrawId*, with the *s_lastDrawTimestamp* updated to manage the interval for automated draw initiation.

Detail Description of the App Frontend

1. Main Page



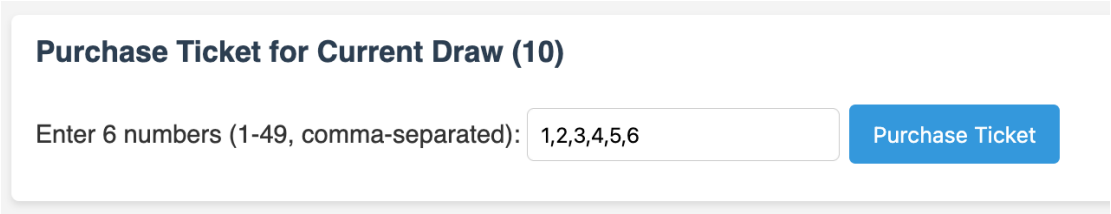
This screenshot shows the Main Page of the Decentralized Lottery.



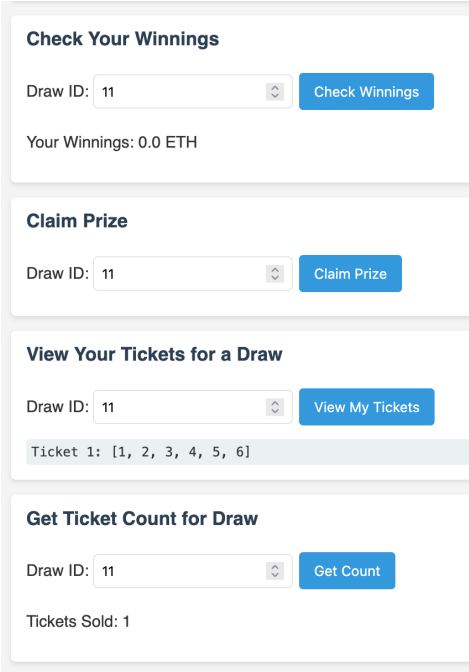
User can Click on the "Connect Wallet" Button. Metamask should pop ip. Approve the connection. The information of the contract and the lottery will be displayed.

2. Player Actions

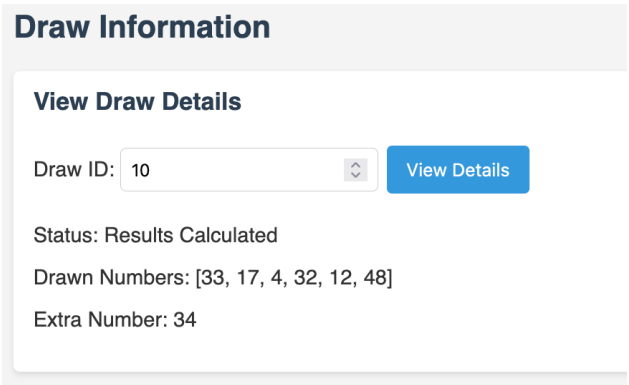
There are 5 player actions: Purchase Ticket for Current Draw, Check Your Winnings, Claim Prize, View Your Tickets for a Draw, and Get Ticket Count for Draw.

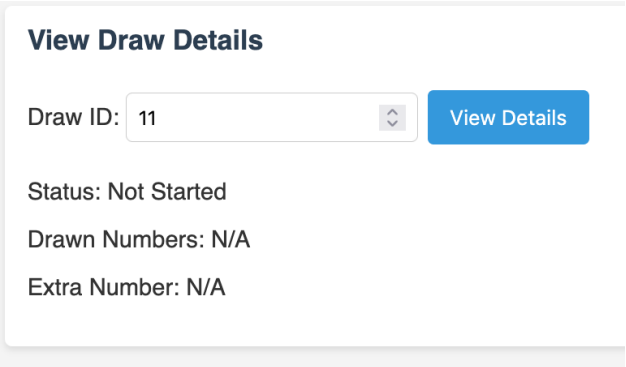


Player can enter the 6 numbers they pick and click the "Purchase Ticket" button. Metamask will pop up asking the user to confirm the transaction.



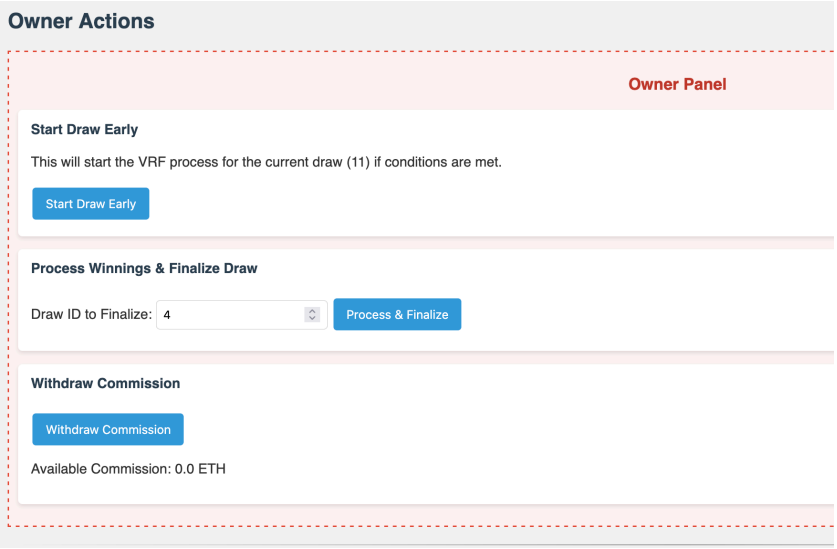
Player can also enter the corresponding "Draw ID" to view the lottery details, like the tickets they buy, the total number of tickets sold. They can also check their winnings and claim their prize here.





Anyone can enter the "Draw ID" to check the corresponding draw details.

3. Owner Panel



A specific owner panel is provided in the frontend for the owner to start the draw early and withdraw their commission.

Security Considerations

The development of a decentralized lottery application, particularly one managing user funds and relying on probabilistic outcomes, necessitates a rigorous approach to security. Several key areas of concern inherent to smart contracts and blockchain-based lotteries have been identified and addressed in the COMP4541 Lottery DApp's design and implementation:

1. Random Number Generation (RNG) Integrity and Fairness

On-chain RNG is susceptible to manipulation by miners or predictable if based on block variables. This undermines the fundamental fairness of a lottery. The DApp integrates

Chainlink Verifiable Random Function (VRF V2Plus). This service provides provably fair and verifiable randomness by generating random numbers off-chain and delivering them with cryptographic proof to the smart contract. The *fulfillRandomWords* function, which processes the VRF output, can only be called by the trusted VRF Coordinator, ensuring that the drawn numbers are tamper-proof and sourced from a secure, external oracle. This decouples random number generation from on-chain execution, preventing any single entity within the blockchain ecosystem from influencing draw outcomes.

2. Smart Contract Vulnerabilities

Smart contracts can be susceptible to common vulnerabilities such as reentrancy attacks, integer overflows/underflows, transaction-ordering dependence (front-running), and flawed access control. While not using the deprecated *transfer()* or *send()*, the prize claim mechanism (*claimPrize*) updates the user's winnings balance (*winnings[_drawId][msg.sender] = 0;*) before initiating the external call for ETH *transfer (msg.sender.call{value: amount}(""))*. This adheres to the checks-effects-interactions pattern, mitigating reentrancy risks during prize payouts. Robust input validation is enforced for critical functions like *purchaseTicket* (checking *msg.value,* number uniqueness, and range) and draw progression is managed by a *DrawStatus* enum, preventing actions from being performed out of sequence (e.g., purchasing tickets after a draw has commenced or finalizing a draw before numbers are set). Custom errors (e.g., *DecentralizedLottery_IncorrectPayment, DecentralizedLottery_BettingClosedForDraw*) provide explicit revert reasons. Sensitive administrative functions such as setting the ticket price (*setTicketPrice*), updating VRF parameters (*setVrfParameters*), withdrawing accumulated commission (*withdrawCommission*), and initiating an early draw (*startDrawEarly*) are restricted to the contract owner using an onlyOwner modifier (inherited from a standard library or implemented within).

3. Prize Pool and Fund Security

The prize pool could be illegitimately drained or mismanaged. The logic for prize pool calculation, commission deduction, and rollover balance updates is embedded within the immutable smart contract code. The commission (*COMMISSION_PCT*) is a fixed percentage, and its withdrawal is an owner-privileged action. Prize distribution to winners is governed by predefined *prizeTierRules* and automated calculations within

_calculateAndRecordWinningsInternal and related functions, minimizing manual intervention and potential for error or bias in payouts.

4. Transaction Integrity and Atomicity

Partial execution of critical operations or race conditions could lead to inconsistent states. Blockchain transactions are atomic; they either complete fully or revert entirely. This ensures that operations like purchasing a ticket (payment and ticket issuance) or claiming a prize (balance update and fund transfer) are executed as a single, indivisible unit. State variables like *drawFinalized* and *drawStatuses* ensure that draws are processed in a coherent and irreversible manner once certain conditions are met.

5. Oracle Reliability and Automation

Over-reliance on a single point of failure for external data (RNG) or automation triggers. Chainlink VRF is a decentralized oracle network, providing high availability and resilience. Similarly, Chainlink Automation (*checkUpkeep* and *performUpkeep*) is designed for reliable, decentralized execution of smart contract functions based on predefined conditions (e.g., time intervals, specific contract states), reducing reliance on a centralized operator for routine draw initiation and finalization. However, the owner retains the ability to *startDrawEarly* and any address can call *processWinningsAndFinalizeDraw* if conditions are met, providing fallback mechanisms.

Conclusion

In conclusion, the COMP4541 Lottery DApp demonstrates the viability of translating a traditional lottery model, inspired by the Hong Kong Mark Six, into a decentralized and transparent blockchain-based system. By leveraging smart contracts for core logic and Chainlink VRF for provably fair random number generation, this project has established a secure, auditable, and user-centric platform for lottery participation. The DApp effectively addresses common concerns of traditional lotteries by enhancing fairness and transparency, showcasing the potential of decentralized technologies to innovate within established gaming paradigms and foster greater trust among participants.