**Introduction**

Welcome to Section 9 of this course! You can code along by following the [GitHub repo](https://github.com/Cyfrin/foundry-smart-contract-lottery-cu) associated with this section. This project will be a valuable addition to your portfolio, as we'll develop a **Verifiably Random Lottery Smart Contract** that contains a lot of best coding practices.

🗒️ **NOTE**  
We won't be deploying this to ZkSync because of current integration constraints.

In this lesson, we will cover **events**, **true random numbers**, **modules**, and **automation**. You can preview the final project by cloning the repository and checking the makefile, which lists all the specific versions of dependencies needed to compile our contract.

The main contract that we'll work on will be src/Raffle.sol. It contains detailed comments and professional-looking NAT spec, such as:

/\*\*

\* @title A sample Raffle Contract

\* @notice This contract is for creating a sample raffle contract

\* @dev This implements the Chainlink VRF Version 2

\*/

This smart contract allows for a **fully automated** Smart Contract Lottery where users can buy a lottery ticket by entering a raffle. Functions like checkUpkeep and performUpkeep will automate the lottery process, ensuring the system runs without manual intervention.

We'll use **Chainlink VRF** version 2.5 for randomness. The fulfillRandomWords function will handle the selection of the random winner and reset the lottery, ensuring a provably fair system.

We'll also write advanced **scripts** that you can find inside the makefile. These include various commands to interact with the smart contract, such as creating subscriptions and adding a consumer.

Let's dive in and start building this exciting project!