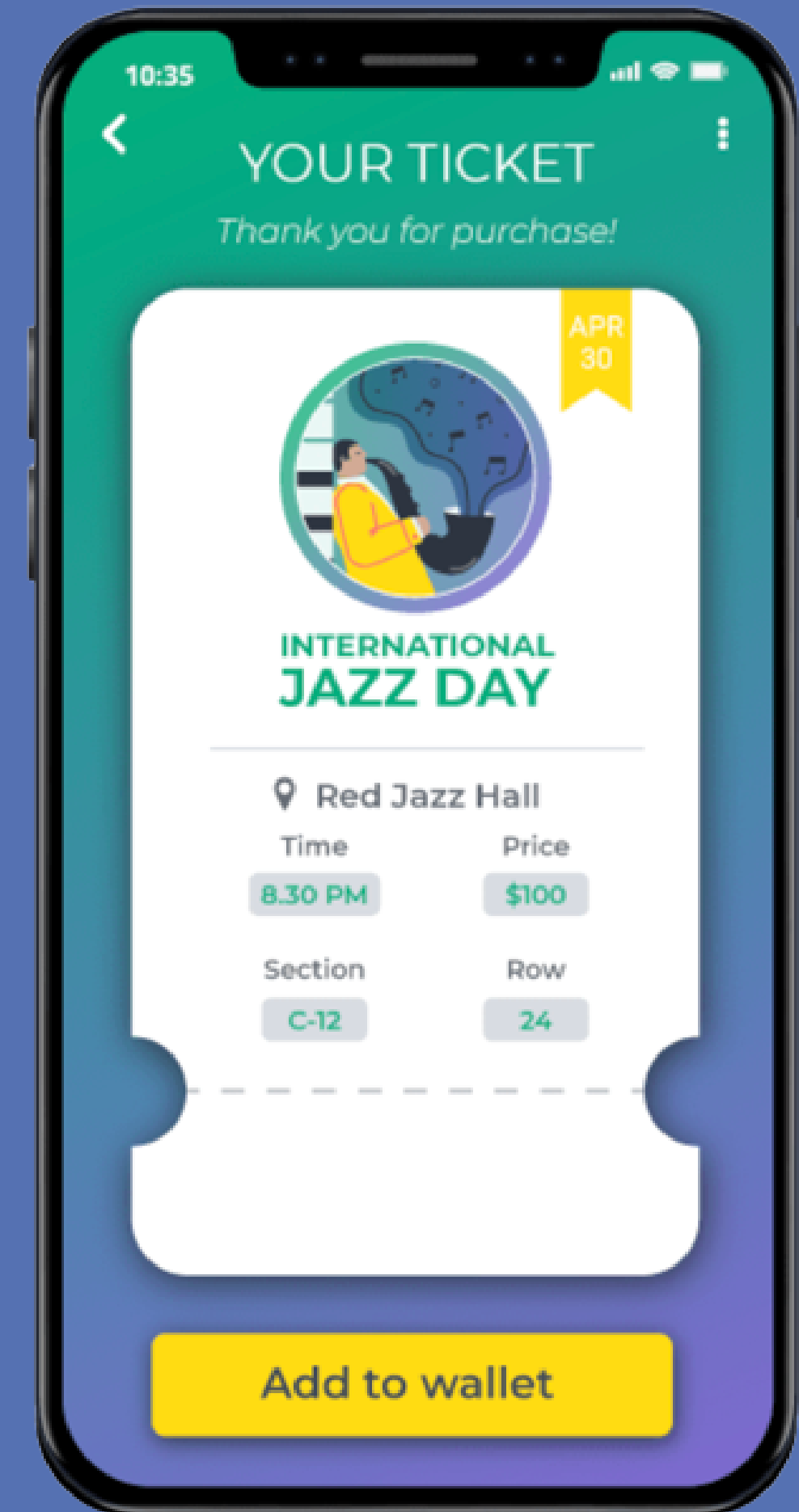


# TkTblock

## Blockchain Digital Ticketing

Fintech Bootcamp  
Team 2



# Content

**01**

Our Team

**02**

Problem Statement

**03**

Solution

**04**

Entities and Journey Mappings

**05**

Tools and Technologies

**06**

Demo

**07**

Lessons Learned and Next Steps

# Our Team



Mario  
Gunawardena



Joseph  
Scorsone



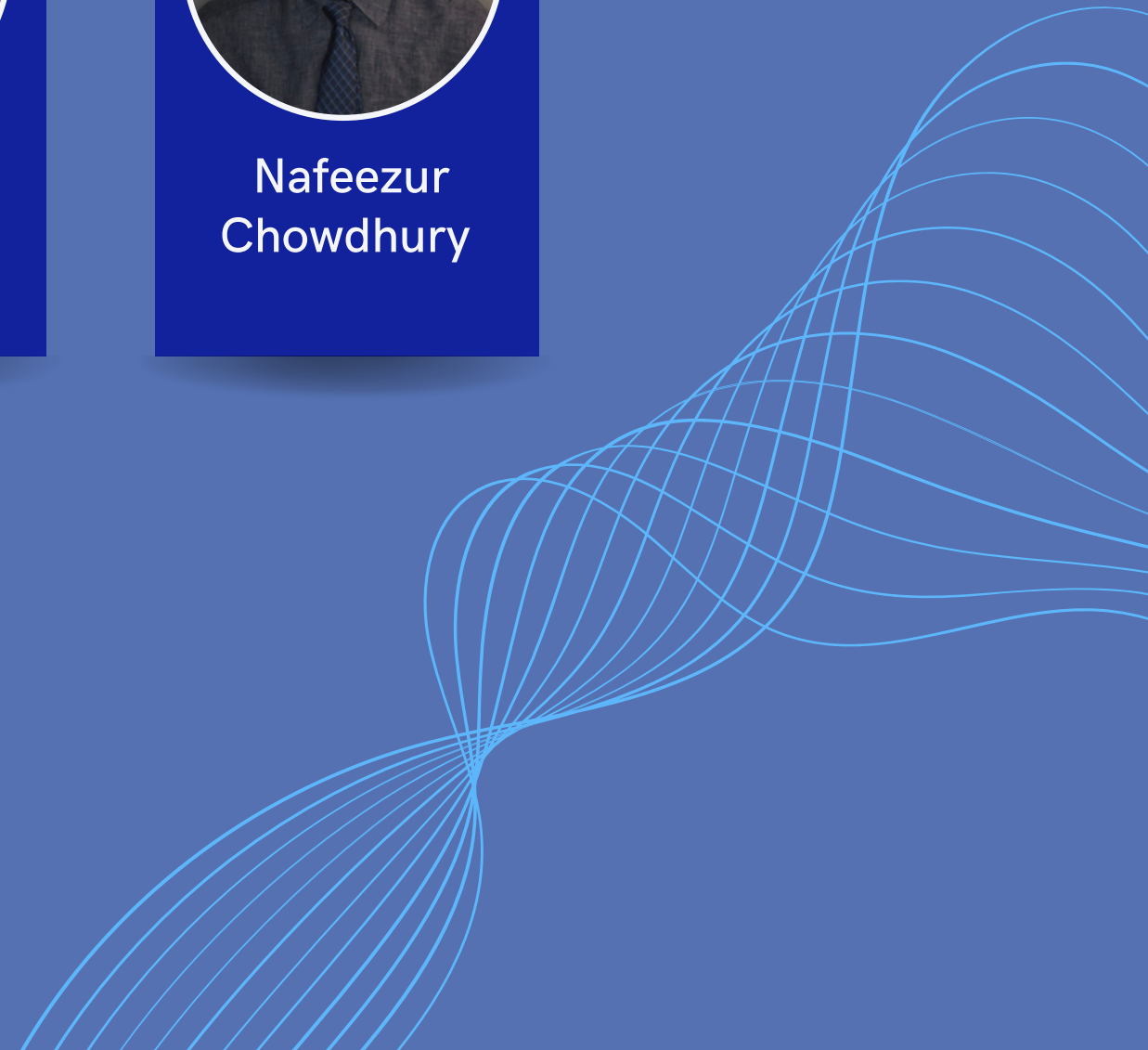
Alghalia  
Alsammak



Luis Rosales  
Garcia



Nafeezur  
Chowdhury

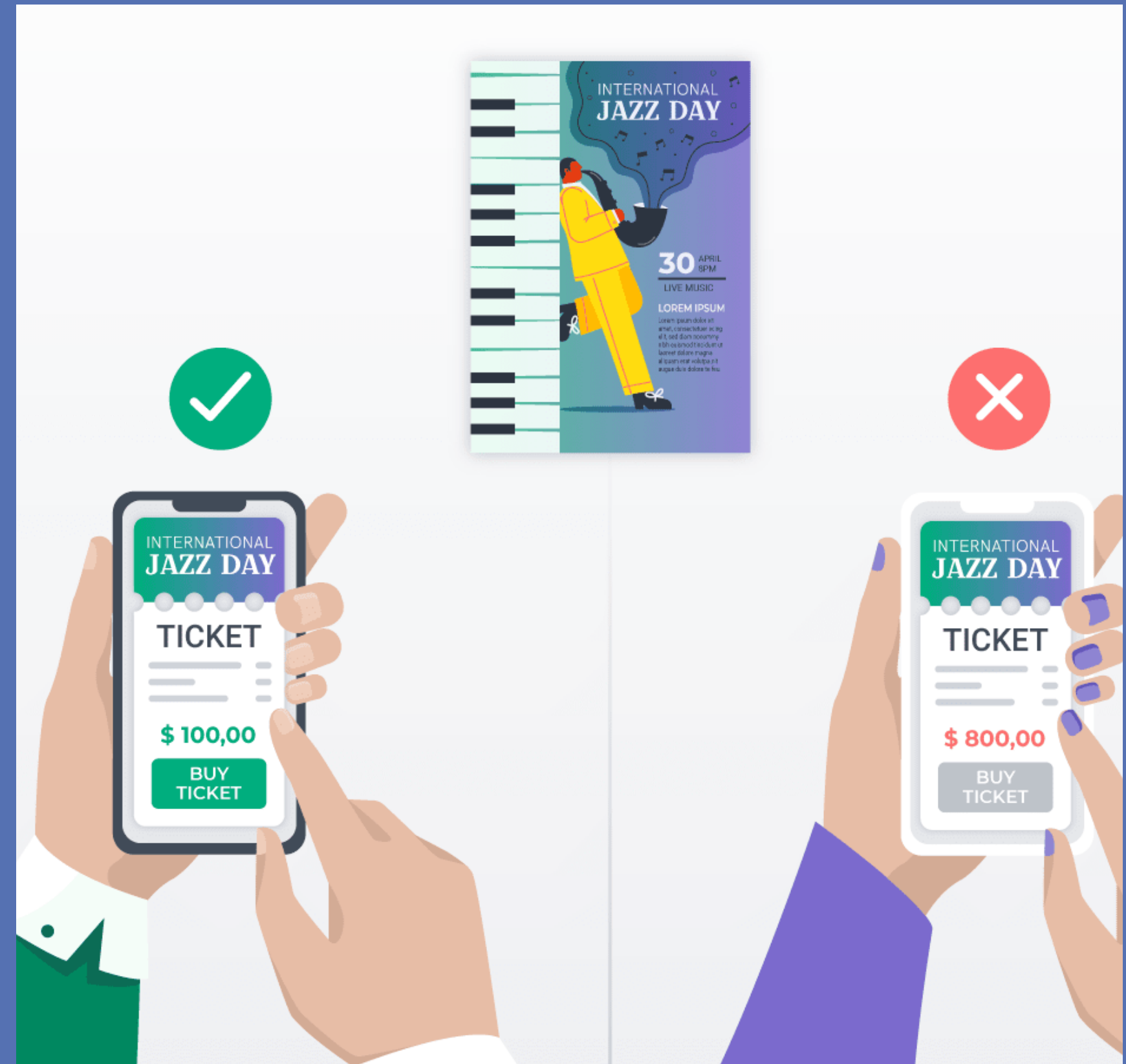


# Problem

Fans, event organizers, and ticket vendors are facing many issues in the events and ticketing market.

- **Fees** - Higher commissions and convenience fees from the providers
- **Lack of choice** - Few controlling majority of the market share and setting rules for artist, venues, customers and promotions.
- **Freedom to Customers** - Fans lack access to buying tickets, as they are unethically sold-out online in a matter of seconds.
- **Fraud** - Lost revenue from tickets being resold by secondary sites and scalpers and fake tickets

“We have a monopoly that covers an entire supply chain.” -  
Diana Moss, president of the American Antitrust Institute.



# Solution

TKTblock has the potential to resolve many pressing issues currently facing the event industry and offers a faster way to create traceable and transferable tickets.



## Benefits

- **Grant control** back to the customers and artist
- Use of **decentralized**, highly available systems
- **Transparency** on event details and ticket supply
- Verification of **authenticity** of tickets to minimize fraud
- Minimal commission **fees**
- Easy **transferability** by the people
- Ability to use multiple **digital wallets**
- **Promote** ticket sales using multiple channels
- **Flexible commissions** agreed by artists, event spaces and agents
- **Promotes competition** in the industry



# Entities

Multiple parties are involved in event related activities

## Contract Owner

- Setup the smart contract
- Initiates the event with details
- Setup the **commission** shares between Artist, Event space and contract Owner
- **Free ticket give aways** (airdrop)
- **Event changes** are made
- Event promotions

## Artist

- Specifies the event agenda
- Requests the **commission shares**
- Organizes the event
- **Performs** the event
- Promotes the event

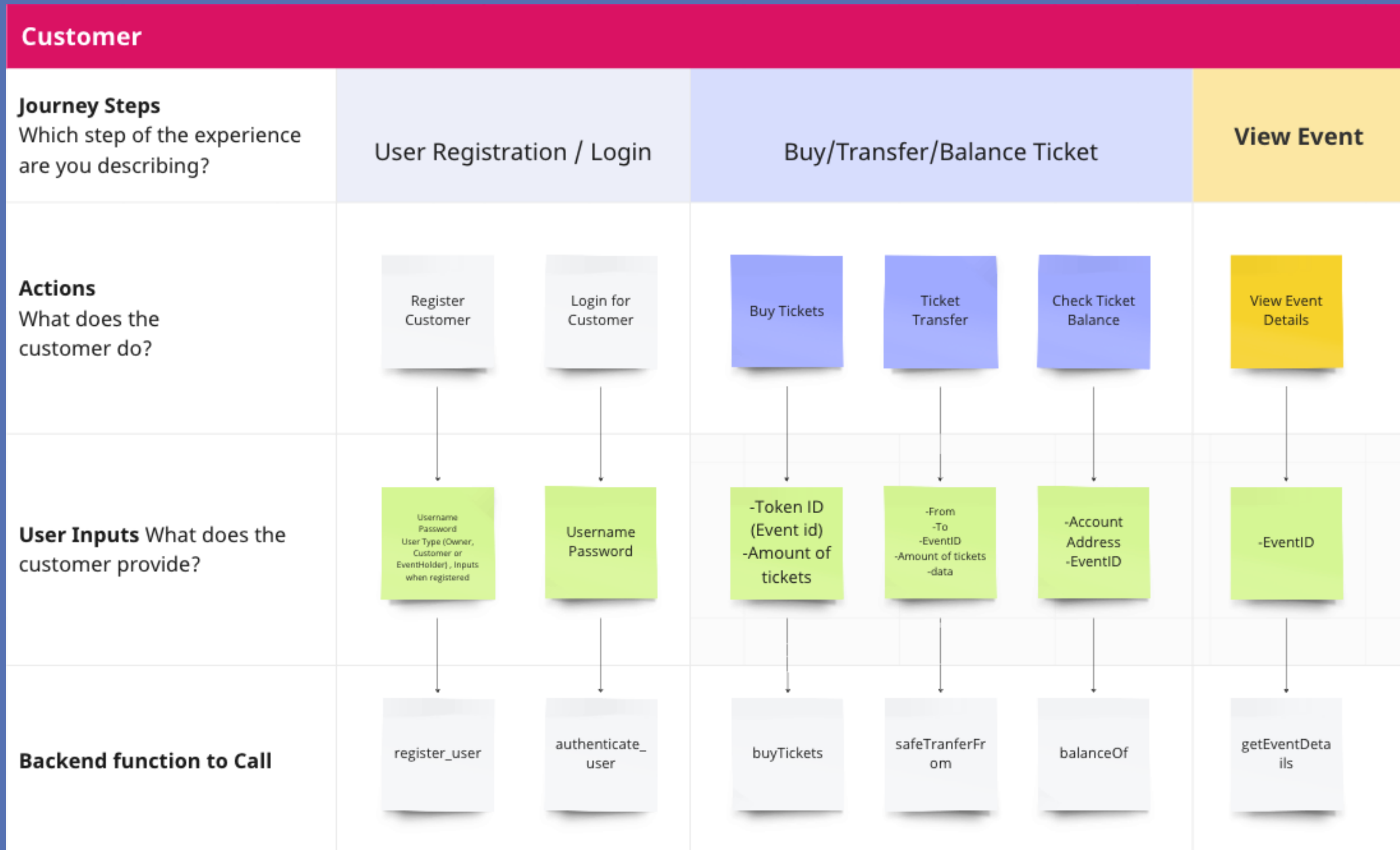
## EventSpace

- Provides seating capacity
- **Admits** the fans
- Checks ticket balances
- **Verifies ticket authenticity**
- Requests the commission shares
- Promotes the event

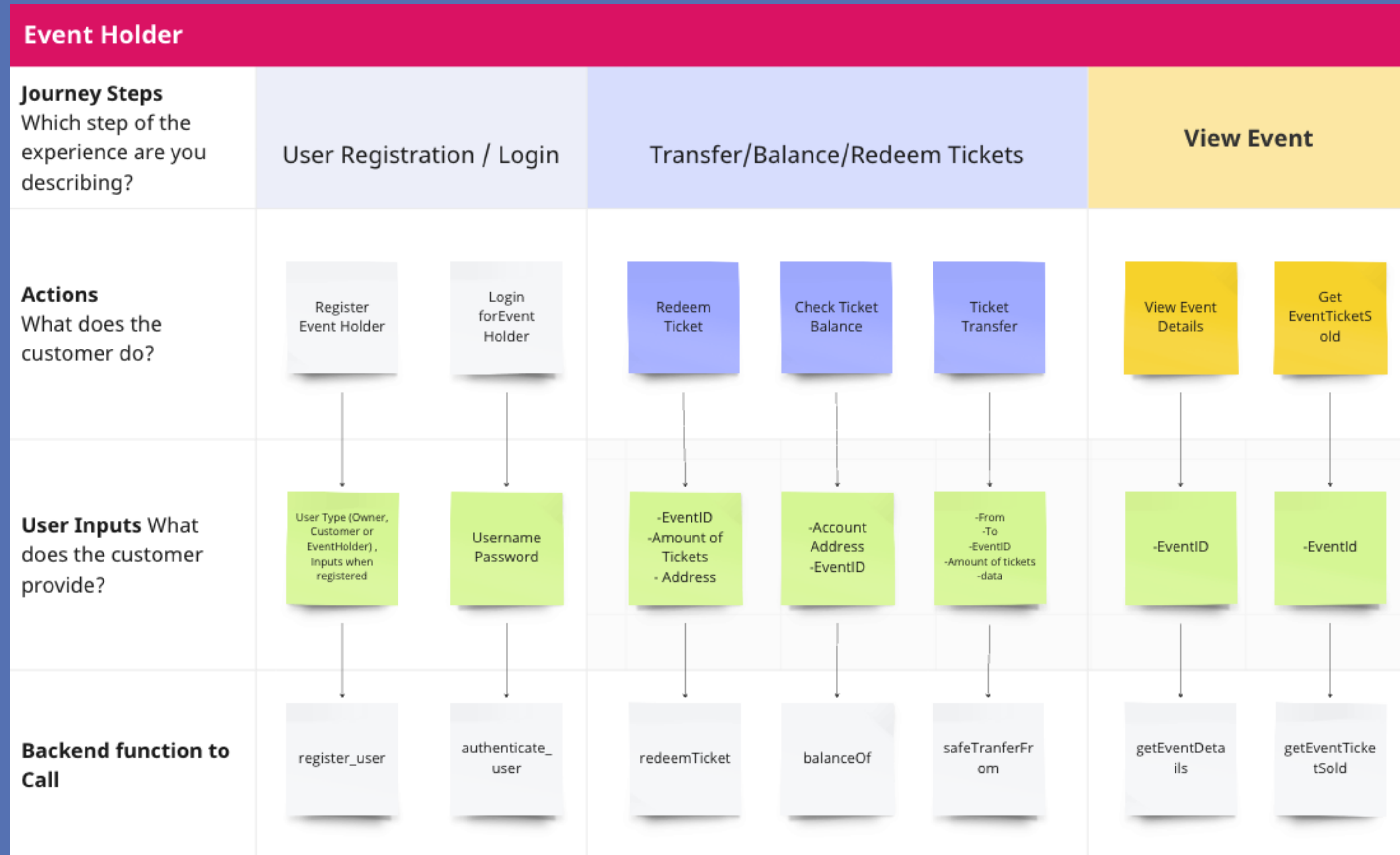
## Customer

- **Buy tickets**
- Check balances
- **Transfer tickets**
- Check event details

# User Journey - Customer



# User Journey - EventHolder





# User Journey - EventOwner/Organizer

Contract Owner / Event Organizer															
Journey Steps Which step of the experience are you describing?	User Registration / Login		Create an Event			View / Update Event / Transfers								Payments	
Actions What does the customer do? What information do they look for? What is their context?	Register Owner	Login for Owner	Deploy Contract	Create event with Details	Instantiate tickets	View Event Details	Get EventTicketSold	Get Event Ticket Price	Get Event URL	AirDrop Tickets	Check Shares	Get Ticket Supply	Ticket Tranfer	Check Ticket Balance	Release commission
User Inputs What does the customer provide?	Username Password User Type (Owner, Customer or EventHolder), inputs when registered	Username Password	-Payee Address [] - Shares for Payees[]	<div>eventOwner (bool) eventOwner (bool) eventDate (Date) eventLocation (bool string) eventTicketPrice (int uint256) eventTicketSupply (number of tickets) eventURL (bool) eventTokenHolderId (uint256) - the first owner that will be able to release tickets for the event</div>	-Token ID (Event id) -Amount of tickets	-EventID	-EventId	-EventId	-EventId	-EventId -recipients[] -amount of tickets	-Address	-EventId	-From -To -EventID -Amount of tickets -data	-Account Address -EventID	- Address
Backend function to Call	register_user	authenticate_user	Contract Deployment	addEvent	buyTickets	getEventDetails	getEventTicketSold	getEventTicketPrice	uri	airdropTickets	shares	GetEventTicketSupply	safeTransferFrom	balanceOf	release

# Tools & Technologies

## CONTRACT TYPE

ERC721

Non-Fungible

Require creation of a smart contract for each new token.

Only allows a single operation for each transaction, it is expensive and time-consuming

Single transactions

Static metadata about tokens

ERC 1115

Mix of **Fungible** and **Non-Fungible**

Single smart contract can be used to mint either fungible/Non-fungible tokens

Multiple operations in a single transaction, therefore **cost effective**

Bulk minting and safe transfers

Multiple URIs can be linked to tokens

# Tools & Technologies

## OPENZEPPPLIN

### ● OpenZeppelin contracts

```
import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC1155/extensions/ERC1155Supply.sol";
import "@openzeppelin/contracts/utils/Strings.sol";
import "@openzeppelin/contracts/security/Pausable.sol";
import "@openzeppelin/contracts/finance/PaymentSplitter.sol";
```

### ● Data Structures - Structs & Mapping and Variables

```
// Create a struct to hold the event information
struct Event {
    uint128 _eventId;
    string _eventName;
    string _eventDate;
    string _eventVenue;
    uint128 _eventTicketPrice;
    uint128 _eventTicketSupply;
    string _eventURL;
    uint128 _eventTicketSold;
    address _eventHolder;
}

// Create a mapping of events with the identifiers
mapping(uint => Event) public events;
```

# Tools & Technologies

## TESTNET INTEGRATIONS

### GoreliEth integration

**Etherscan** Home Blockchain Tokens NFTs Misc

**Contract** 0x3845AAaA2c13C6dF2C9B31331c504afe4829481

**Overview**

ETH BALANCE  
200 wei

**More Info**

CONTRACT CREATOR  
0x7f9389...90e5473f at txn 0xa2c39c4953ce9cf3f...

TOKEN TRACKER  
ERC1155

**Multi Chain**

MULTICHAIN ADDRESSES  
1 address found via Blockscan

**Transactions** Token Transfers (ERC-20) Contract Events

Latest 4 from a total of 4 transactions

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0x9797fc54fe66fa312...	0xa225e20c	9523291	6 hrs 46 mins ago	0x7f9389...90e5473f	0x3845AA...e4829481	0 ETH	0.00046885
0xcd955f8dc5543f71c...	Buy Tickets	9523273	6 hrs 51 mins ago	0x7f9389...90e5473f	0x3845AA...e4829481	200 wei	0.00031288
0x77f4145d8c34c24cc...	0xa225e20c	9523269	6 hrs 52 mins ago	0x7f9389...90e5473f	0x3845AA...e4829481	0 ETH	0.00046891
0xa2c39c4953ce9cf3f...	0x60806040	9523240	7 hrs ago	0x7f9389...90e5473f	Contract Creation	0 ETH	0.00489931

### GoreliEth integration

**Etherscan**

**Transaction Details**

**Overview** Logs (1) State

[ This is a Goerli Testnet transaction only ]

Transaction Hash: 0xcd955f8dc5543f71c37d068f41e25e041e1690cfb1469ed59d62f5d23cbfc3c

Status: Success

Block: 9523273 23093 Block Confirmations

Timestamp: 4 days 1 hr ago (Aug-15-2023 05:39:00 PM +UTC)

From: 0x7f93896d3064DCEf21efc282A3B49E1f90e5473f

Interacted With (To): 0x3845AAaA2c13C6dF2C9B31331c504afe4829481

ERC-1155 Tokens Transferred: ERC-1155 For 2 of Token ID [1] From 0x0000000...00000000 To 0x7f9389...90e5473f

# Tools & Technologies

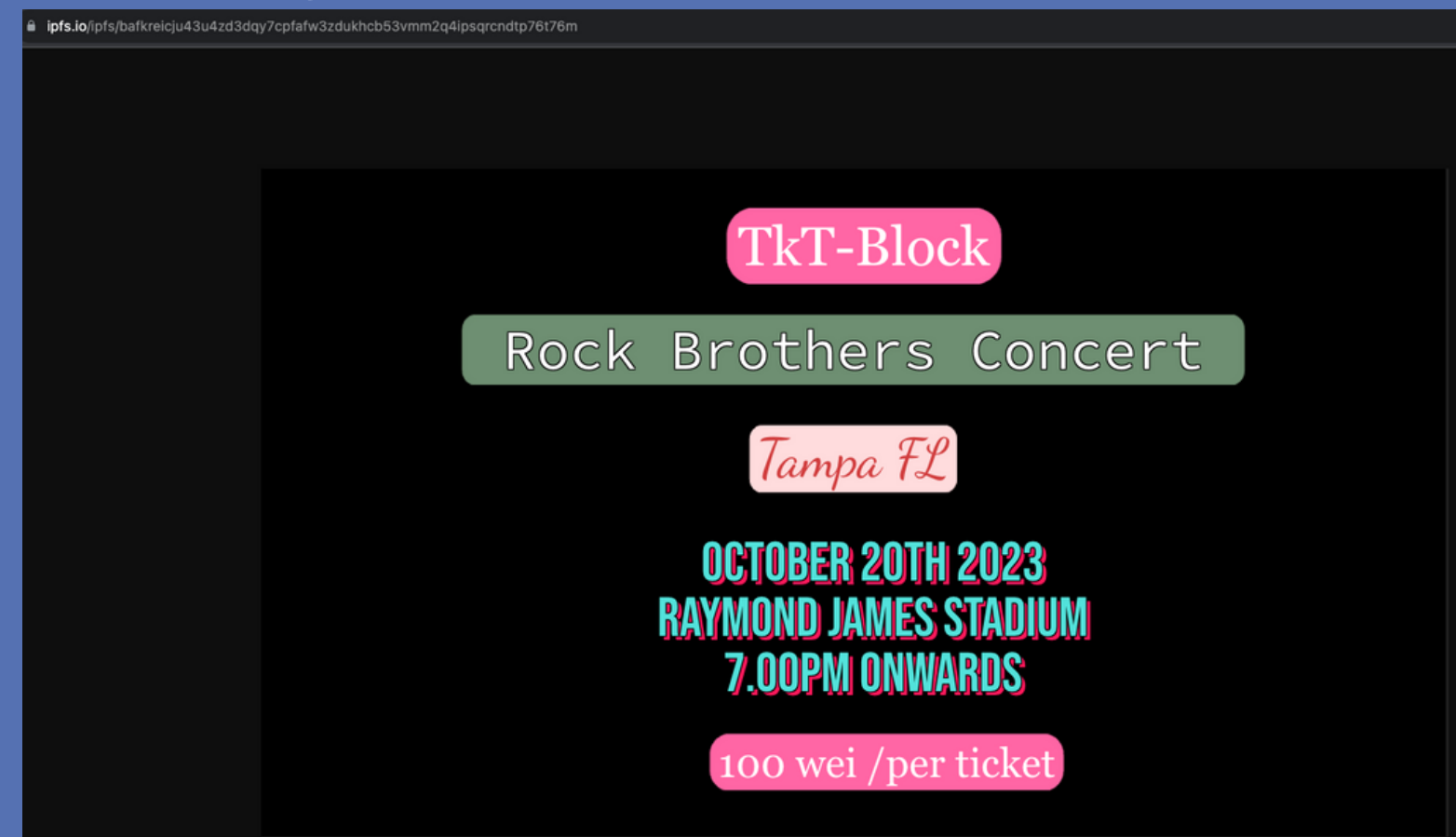
## IPFS INTEGRATIONS

### IPFS Image Metadata

```
← → ↻ ipfs.io/ipfs/bafybeidl22wc73c75gdqnl26wvjzhykwglx5djiyjdggplclnxxl3m/1.json

{
  "name": "Rock Brothers Concert",
  "description": "Rock Brothers Concert, Tampa FL, October 2023",
  "image": "https://ipfs.io/ipfs/bafkreicju43u4zd3dqy7cpfafw3zdukxcb53vmm2q4ipsqrcndtp76t76m"
}
```

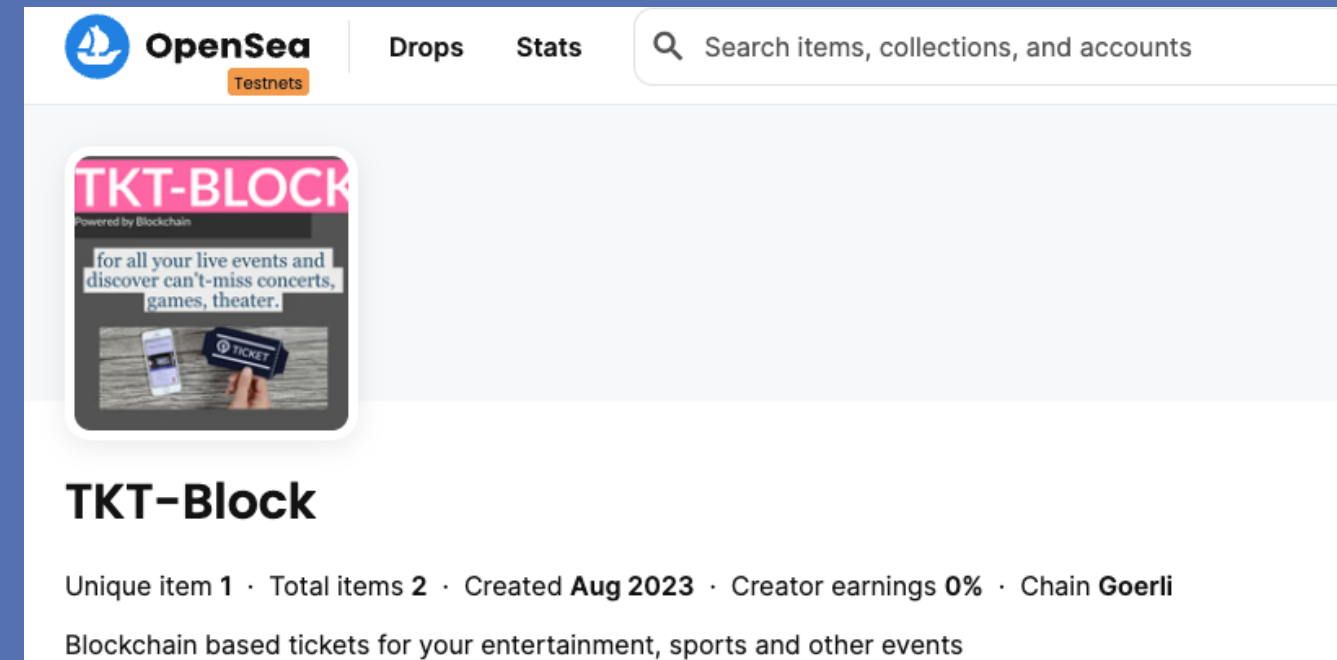
### IPFS Image URLs



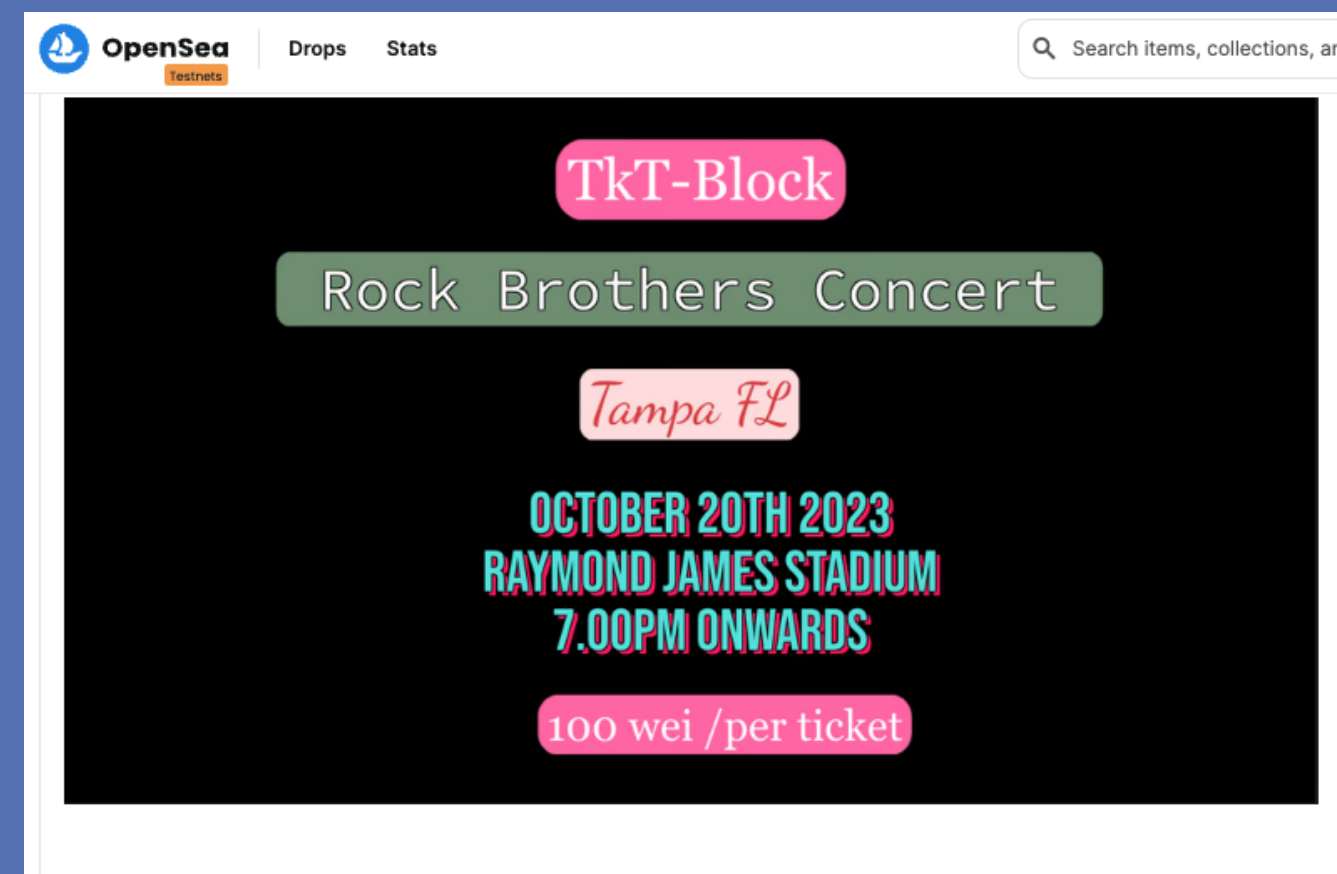
# Tools & Technologies

## OPENSEA INTEGRATIONS

### Open Sea integration



### Open Sea Token

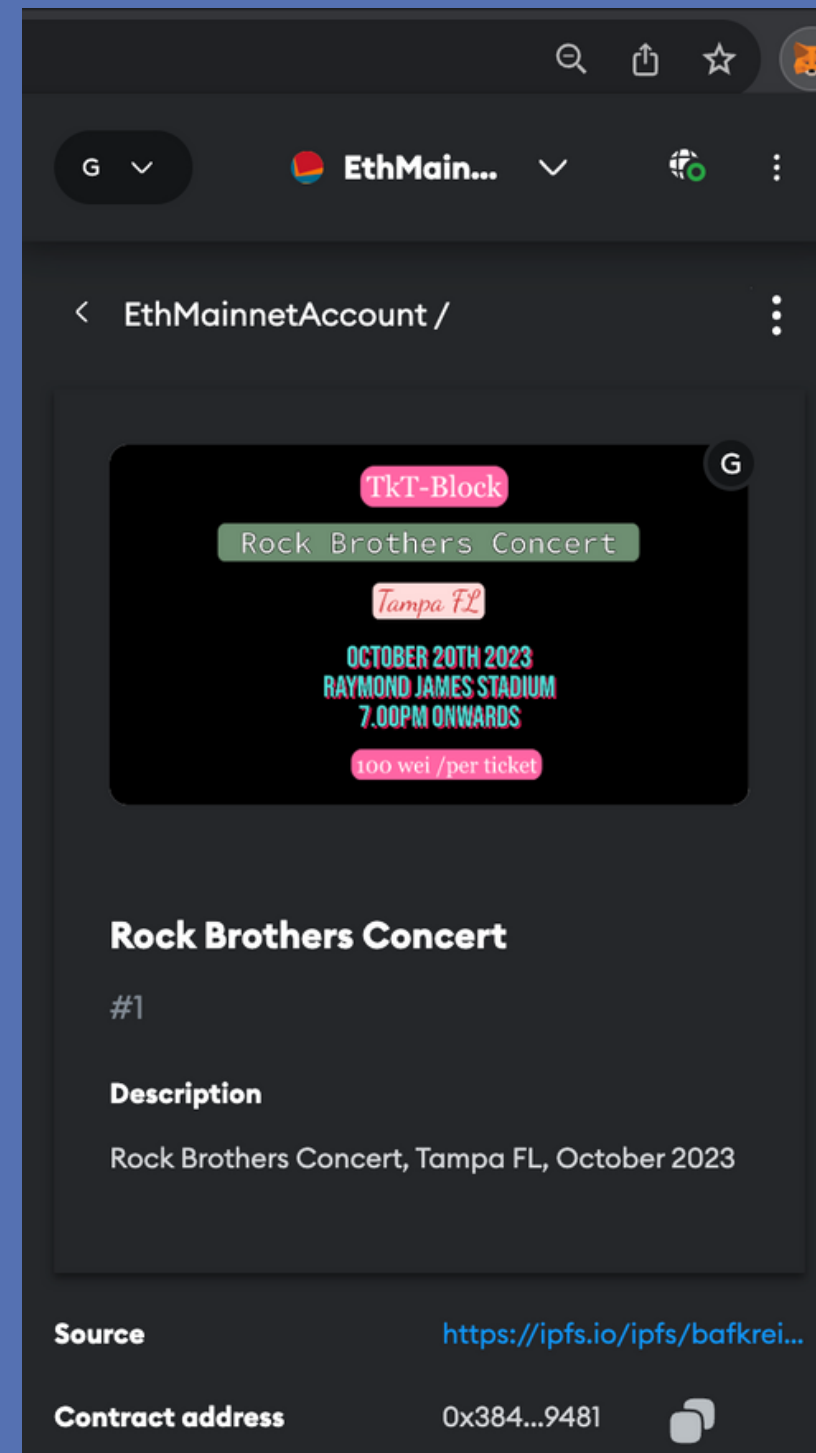




# Tools & Technologies

METAMASK  
INTEGRATION

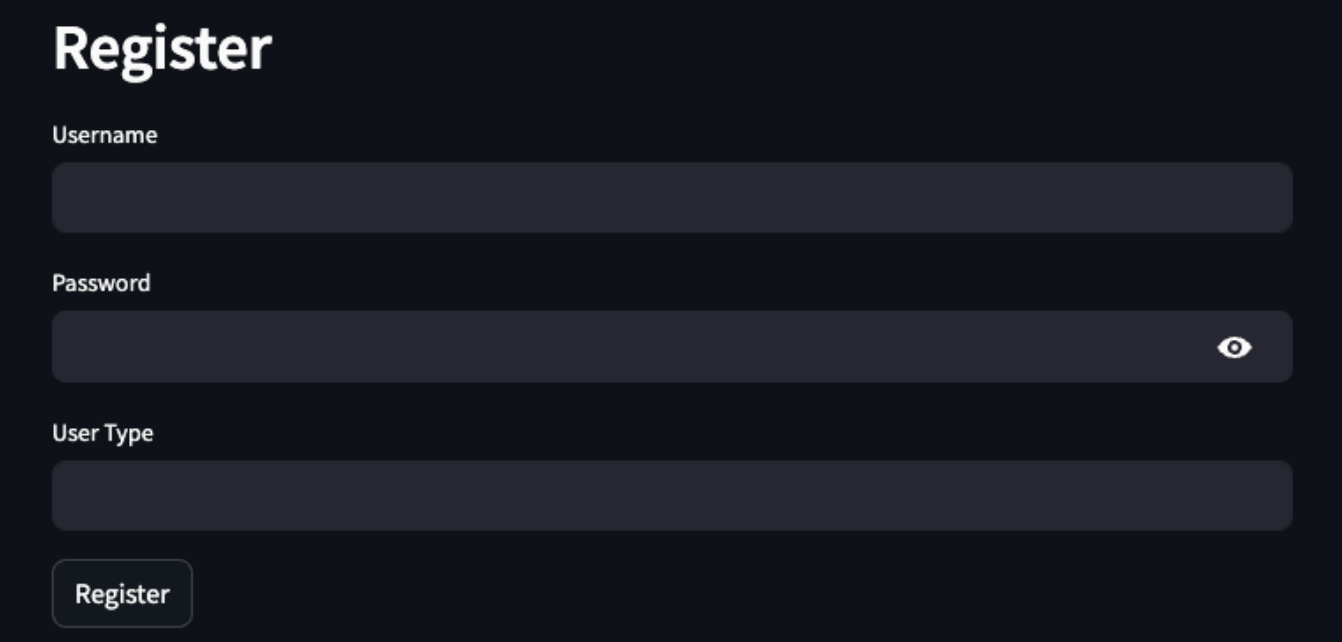
## ● NFT Imports into Wallet



# Tools & Technologies

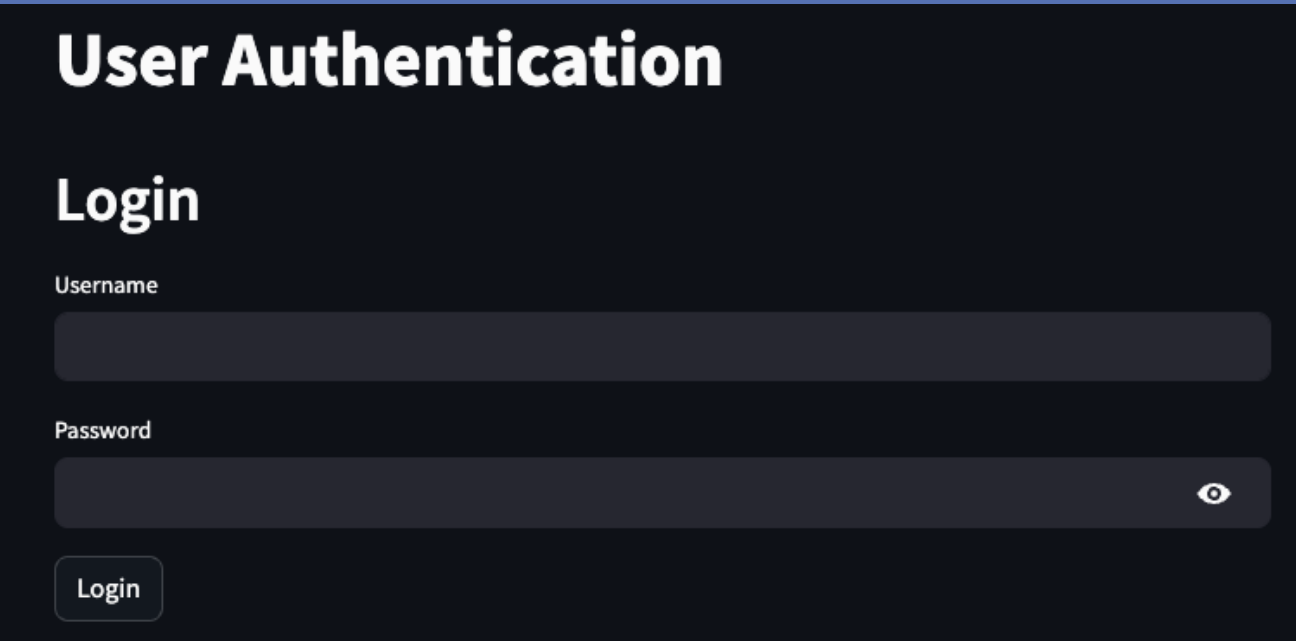
STREAMLIT  
INTEGRATION

## ● Streamlit User Registration with Roles



A screenshot of a Streamlit user registration form. The form is titled "Register" in bold white text. It contains three input fields: "Username", "Password", and "User Type". The "Password" field has a toggle icon (an eye) on the right side. Below the input fields is a "Register" button.

## ● Streamlit Authentication



A screenshot of a Streamlit user authentication form. The form is titled "User Authentication" in bold white text. Below the title is the word "Login" in bold white text. It contains two input fields: "Username" and "Password". The "Password" field has a toggle icon (an eye) on the right side. Below the input fields is a "Login" button.

# Lessons Learned and Next Steps

## Challenges faced

- ERC 1155 token does not let you add specific info for copies of the token which limited us adding seat info
- Contract size limitation (24,000 bytes) made us optimize the contract
- Getting testnet ether was limited to per day quotas for Goreli testnet, which limited the testing ability
- Some testnets don't integrate with solutions like OpenSea

## Future Improvements

- Ability to add barcodes to tickets
- Ability for Artist to interact with the smart contract
- Ability to include seat assignments to contracts
- Integration with a mobile application / Wallet
- Ability to transfer tickets and get funds from the transfer

# Demo

**Thank You!**

# Presentation Order and assignments

Total 12 mins

Intro - Entities (1-6) - AlGhalia (2 mins)

Journey (7-9) - Joe (3 mins)

Technologies and Integration (10-15) - (2 min) Mario

Lessons Learned and Next Steps (17) - (1 min) Naf

Demo (18) - Luis (4 mins)