Final Project

Paola Petri

1 Implementation

The frontend for the smart contracts has been developed using *Python* as programming language, *Flask* microframework, *Web3* library for Python.
The blockchain was simulated using *Ganache*.
The contracts in the project have been compiled using *Truffle*.

2 Front end

The front end has been developed using a common interface for the lottery manager and the users until they log in in the system. From the home page everyone can mint tokens and enter the **Lottery** section.

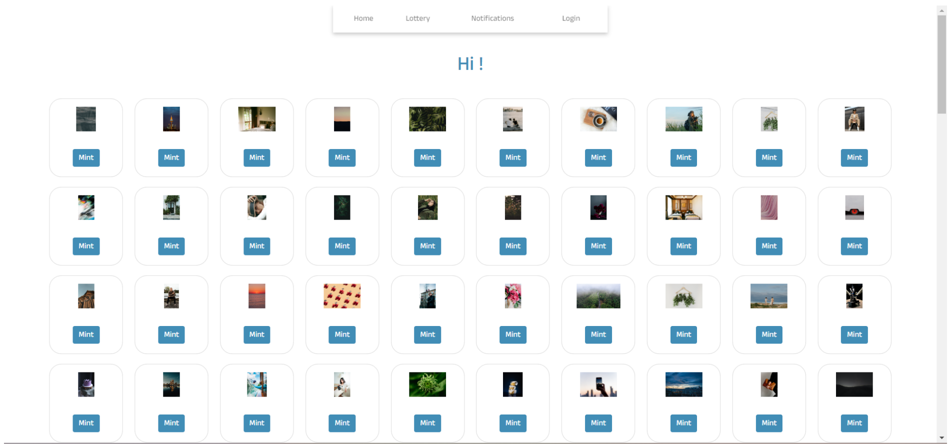


Figure 1: Home page

For each operation is required to log in to the system, so, if the user is not logged in, when the request starts, it will be redirected in the login section. There is also a *Login* button, to log in arbitrarily.

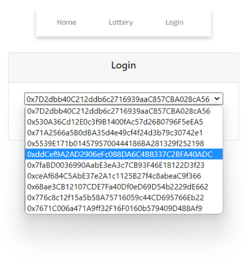

Home
Lottery
Login
Login
0x7D2dbb40C212ddb6c2716939aaC857CBA028cA56
0x7D2dbb40C212ddb6c2716939aaC857CBA028cA56
0x530A36Cd12E0c3f9B1400fAc57d26B0796F5eEA5
0x71A2566a5B0dBA35d4e49cf4f24d3b79c30742e1
0x5539E171b0145795700444186BA281329f252198
0xddCef9A2AD2906eFc088DA6C4B8337C2BFA40ADC
0x7faBD0036990AabE3eA3c7CB93F46E18122D3f23
0xceAf684C5AbE37e2A1c1125B27f4c8abeaC9f366
0x68ae3CB12107CDE7Fa40Df0eD69D54b2229dE662
0x776c8c12f15a5b58A75716059c44CD695766Eb22
0x7671C006a471A9ff32F16F0160b579409D488Af9

Figure 2: Login page

The login page offers a dropdown menu in which we can choose an account from the accounts offered by Ganache. An account is the **manager** of the lottery (see Chapter 3), the others are **users**.

2.1 Lottery Manager

The manager, after the login, can manage the lottery buying collectibles to be used as prizes, start and stop the lottery and the rounds. After the login he can start the lottery and subsequently a round. Then he can manage the lottery operation such as buying future prizes or draw numbers and assign prizes.

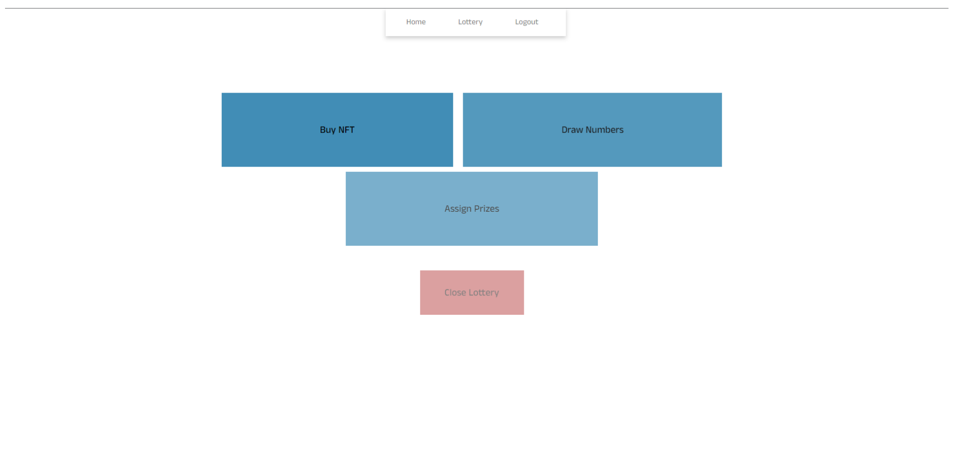


Figure 3: Lottery Manager page

The button ”Buy NFT” redirect the manager to the home page, but the function called from the contract is the one that mint a collectible and assign it a class to be used as prize.

2.2 Users

A normal user, after the login, can mint tokens from the home page, and can subscribe to the lottery to receive notification and participate buying tickets in the lottery section.

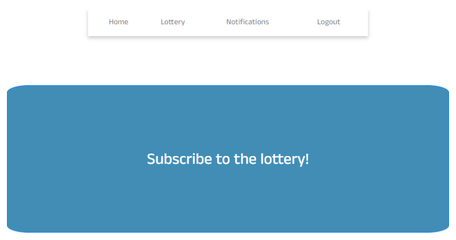


Figure 4: Lottery Subscribe page

After subscription, if the lottery manager has opened a round, the user can buy tickets for all the duration of the round and wait for the notification of the extracted numbers.

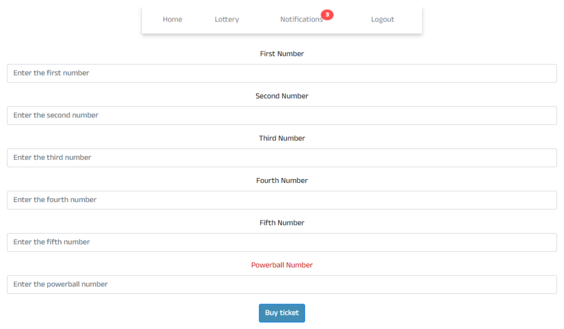
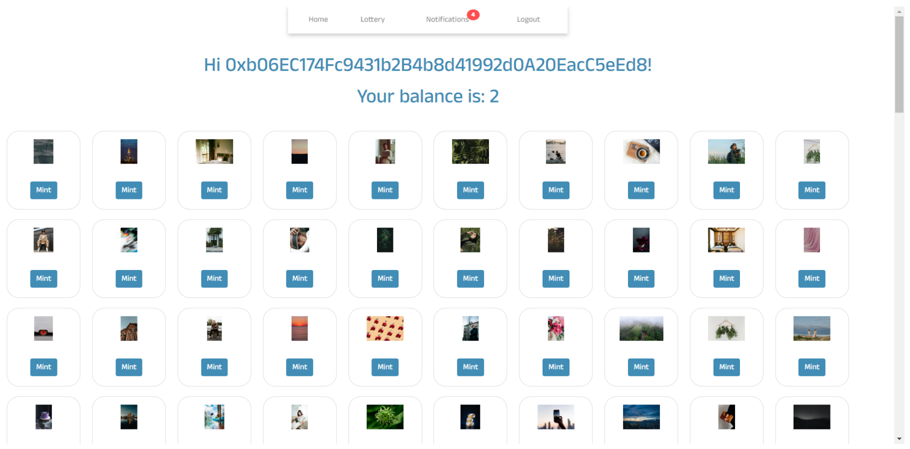


Figure 5: Buy ticket page

If a user wins some collectibles, he can see his balance in the home page:


Hi 0xb06EC174Fc9431b2B4b8d41992d0A20EacC5eEd8!

Your balance is: 2

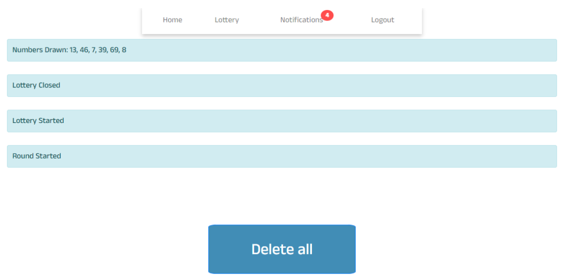
Figure 6: Balance page

The user part of the project has been implemented using some parameters of the *session* dict offered by Flask microframework, for example to enable notifications and playing lottery for the users who subscribed to the lottery.

2.3 Notifications

The notifications have been implemented using the main events generated from the smart contracts and creating an *event filter*. The notification section is limited to users, so, when the lottery manager will login, won't see this button in the navbar overhead. The user instead, will see in real time the notifications with a red circle in the navbar without the need to refresh or change the pages. This has been implemented with a script related to the notification button in the navbar. The notifications are related to:

- Lottery closing/opening
- Round starting
- Numbers drawing
- Prize assigned



Numbers Drawn: 13, 46, 7, 39, 69, 8

Lottery Closed

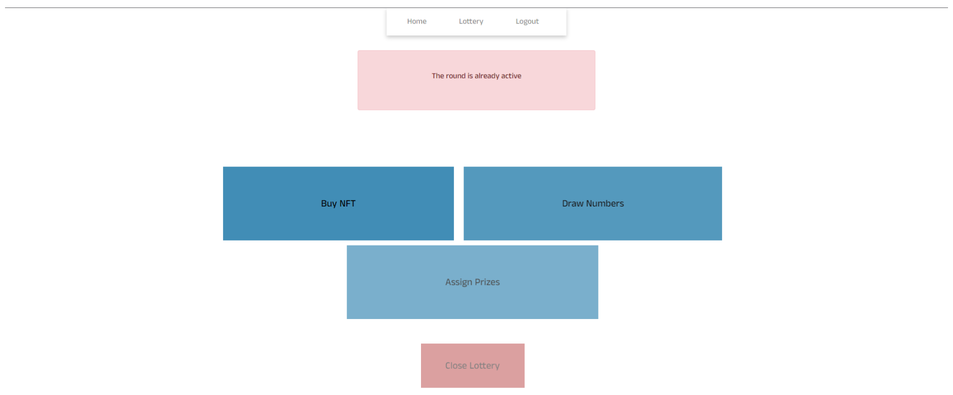
Lottery Started

Round Started

Figure 7: Buy ticket page

2.4 Transaction status

The transaction outcome has been always checked to guarantee, both to users and manager, to know which is the main reason of the failure of a transaction. The requirements of each transaction are checked if the status means failure and the error related to the failing requirements is displayed. For example this appears if the manager tries to draw numbers even if the round is not finished:



The round is already active

Figure 8: Buy ticket page

3 Instructions

1. Open a terminal
2. type `virtualenv venv`
3. type `source venv/bin/activate`
4. type `pip install -r requirements.txt`
5. type `sudo truffle compile`
6. type `ganache`
7. Choose and copy from the private keys displayed in the terminal, the one related to the lottery manager (be careful: if you choose the first key, the lottery manager account will be the first displayed in the accounts dropdown menu in the login page)
8. Paste the key in the file *app.py* in the *owner* variable
9. Open a new terminal
10. Repeat 3
11. run `python3 app.py`