

# TEAM NAME:VIJAY04



PACKAGES\_USED:

- 1.KIVY
- 2.PYTEZOS

Github: IN4111

Web3 integration with Web2 Track-Fair Reselling Marketplace Using Tezos Technology.

# HOW IT WORKS?

- EACH TICKET HAS A SECRET TEXT/HASH
- ON SUCCESSFUL PURCHASE ,RESELLER GET THE TICKET HASH OF THE TICKETS
- THIS HASH IS THEN EVALUATED AND VERIFIED BY GOING THROUGH TICKET PURCHASE HISTORY, TO WHOM IT SOLD ,WHAT DATE SOLD AND SO ON.
- RESELLERS CAN BE REMOVED FROM ACCESSING TICKETS

# APP BUILD FOR ADMIN

### Welcome to Dashboard

The dashboard features three main sections: Tickets, Resellers, and History. The Tickets section lists five items with their IDs, creation dates, and prices, each with 'Sell' and 'remove' buttons. The Resellers section lists three resellers with their names, Tezos Public Keys, and creation dates. The History section lists two sold tickets with their details, including Ticket Hashes and Transaction hashes.

Ticket ID	Created Date	Price
ticket_id:45234	2023-08-15	15.12 XTZ
ticket_id:8721	2023-08-15	29.23 XTZ
ticket_id:12352	2023-08-15	12.9 XTZ
ticket_id:8727	2023-08-15	12.2 XTZ
ticket_id:8712	2023-08-15	

Name	Tezos Public Key	Created On
reseller_1	tz1PtXvDfB FsD5GjEmrCdXiqabETCvLVAyP	2023-08-15
reseller_2	tz1TGhf9Q8P zR2ddFLWfhHdQvb3AthLYGTKe	2023-08-15
reseller_3	tz1R7NScUmj Uo38CyDDjfxAbkqy2oJ9nbEfK	2023-08-15

Ticket ID	Price	Sold To	Sold On	Ticket Hash	Transaction Hash
4112	12.12	reseller_1	2023-08-15	2872c509c246f08c03de181e81eaa19ad751b3cc6c37f501dd0214a2c55c085	op5VWm8twb8dn2xMz4ELrKitxufWDJXmiFwKxsbQwXPSUy41z4
6472	56.212	reseller_2	2023-08-15	a37302b8da057d9cd579abb15495ee9ec0887d861c8d83e35e51021dff844cd	

[Add a Ticket](#)

[Add a Reseller](#)

[Remove a Reseller](#)

# APP BUILD FOR RESELLERS

### Welcome to Dashboard



Tickets

ticket\_id:12352  
created\_date:2023-08-15  
price:12.9 XTZ

Buy

ticket\_id:8727  
created\_date:2023-08-15  
price:12.2 XTZ

Buy

ticket\_id:8712  
created\_date:2023-08-15  
price:21.64 XTZ

Buy

ticket\_id:231  
created\_date:2023-08-15  
price:12.12 XTZ

Buy

ticket\_id:31  
created\_date:2023-08-15  
price:123.23 XTZ

Buy

Your purchases

Ticket\_id:4112  
Price:12.12  
SoldTo:reseller\_1  
Sold On:2023-08-15

Ticket\_Hash: 2872c509c246f08c03de181e81eaa19a  
d751b3cc6c37f501dd0214a2c55c085

Transaction\_hash: op5VWm8twb8dn2xMz4ELrKt  
ixufWDjXmiFwKxsQwXPSUy41z4

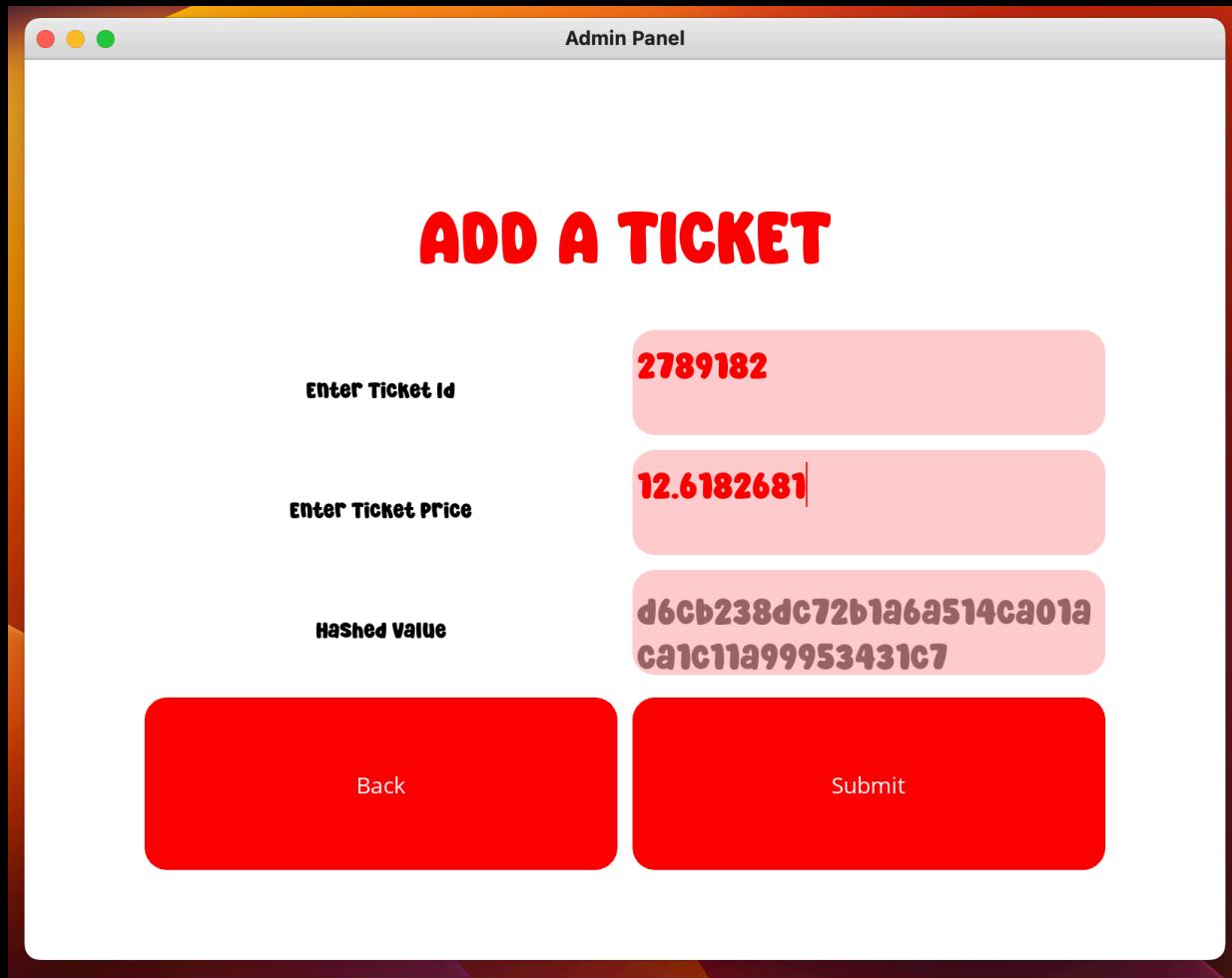
Ticket\_id:271782  
Price:10.23  
SoldTo:reseller\_1  
Sold On:2023-08-15

Ticket\_Hash: b48ee0492aaaf052a643299bf2b440bdf  
3388fdbd072aff79f347c03ae7098f1

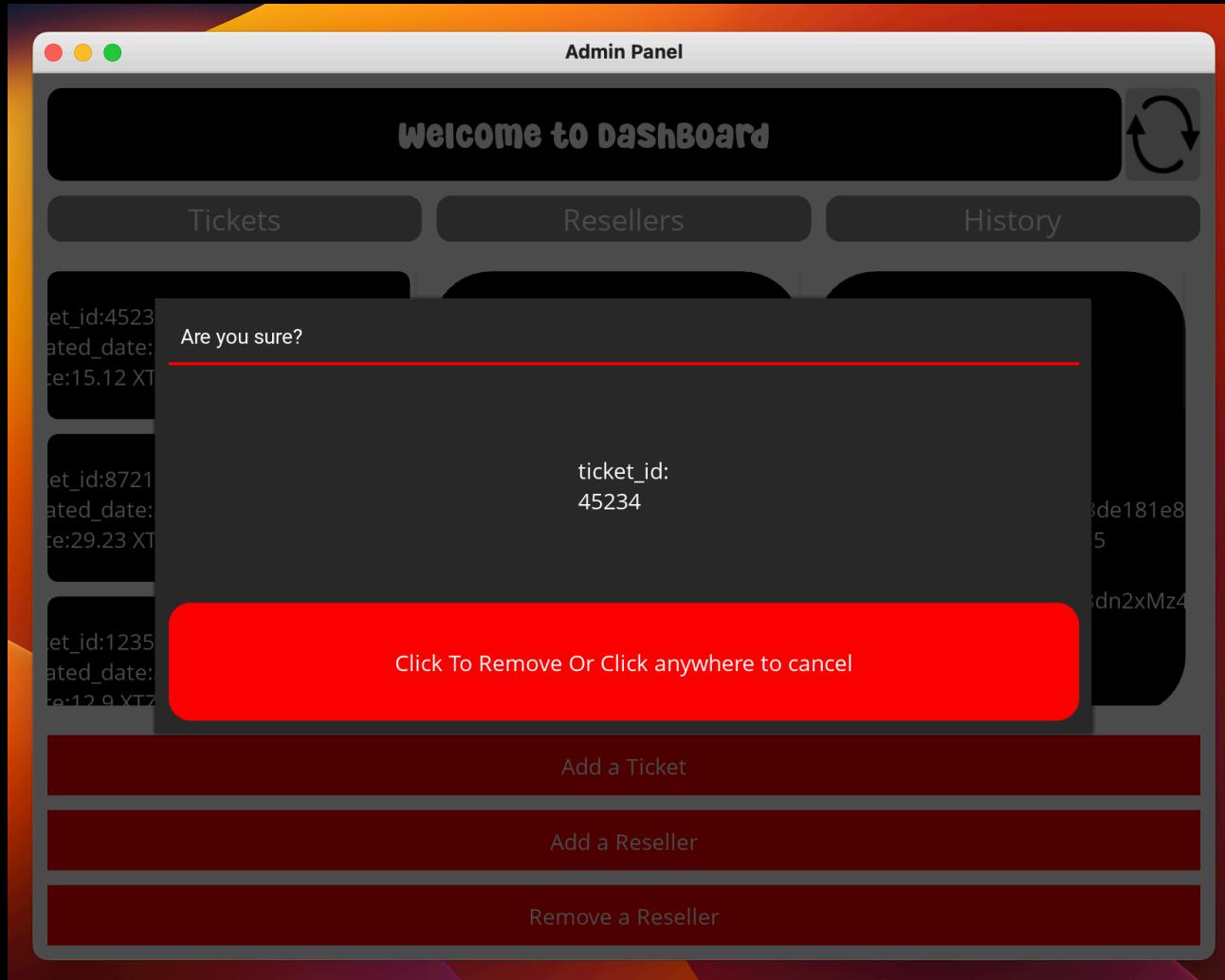
Transaction hash: oosDLg5GRkvCF1Fud1gPS2N

LogOut

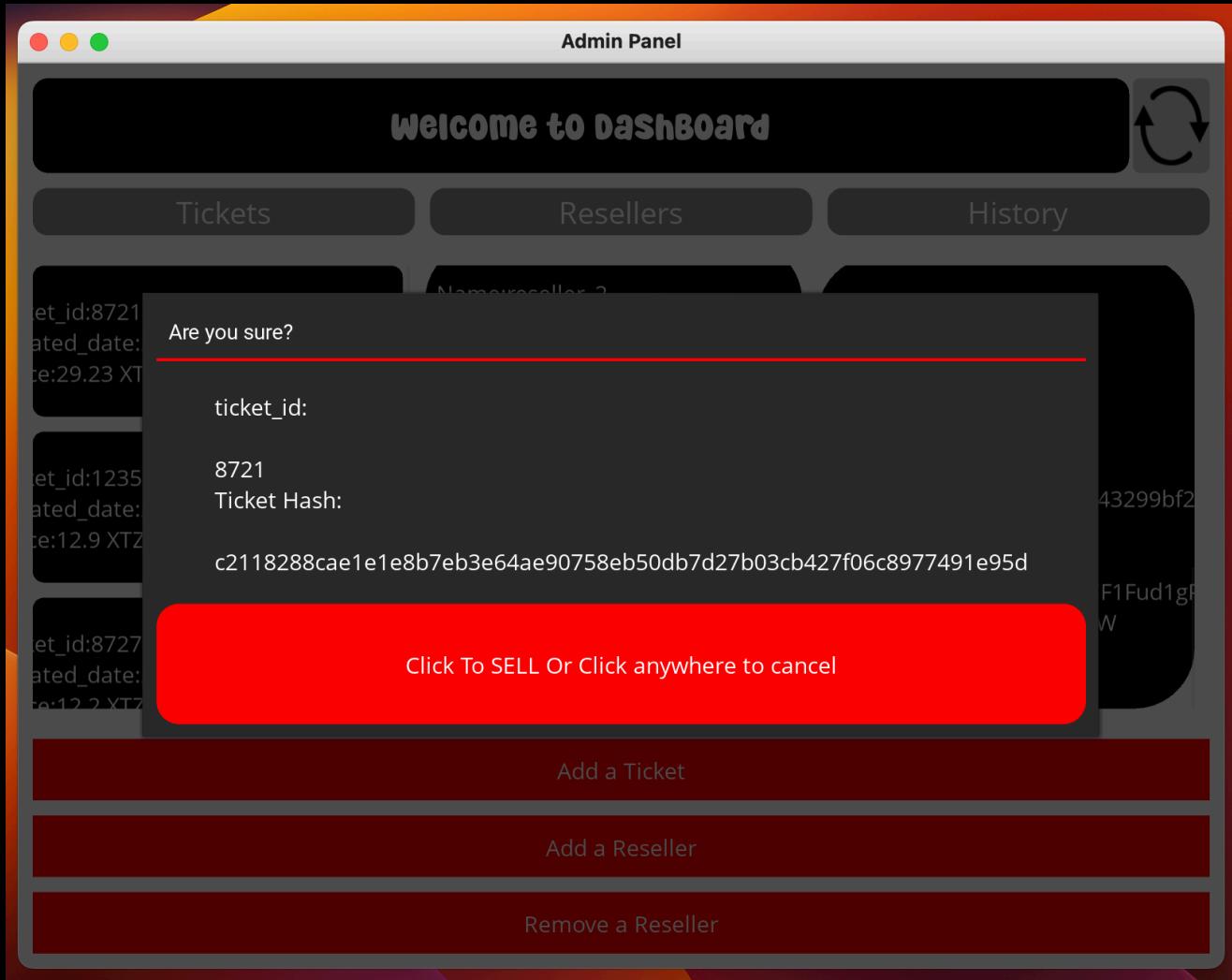
# ADDING TICKETS



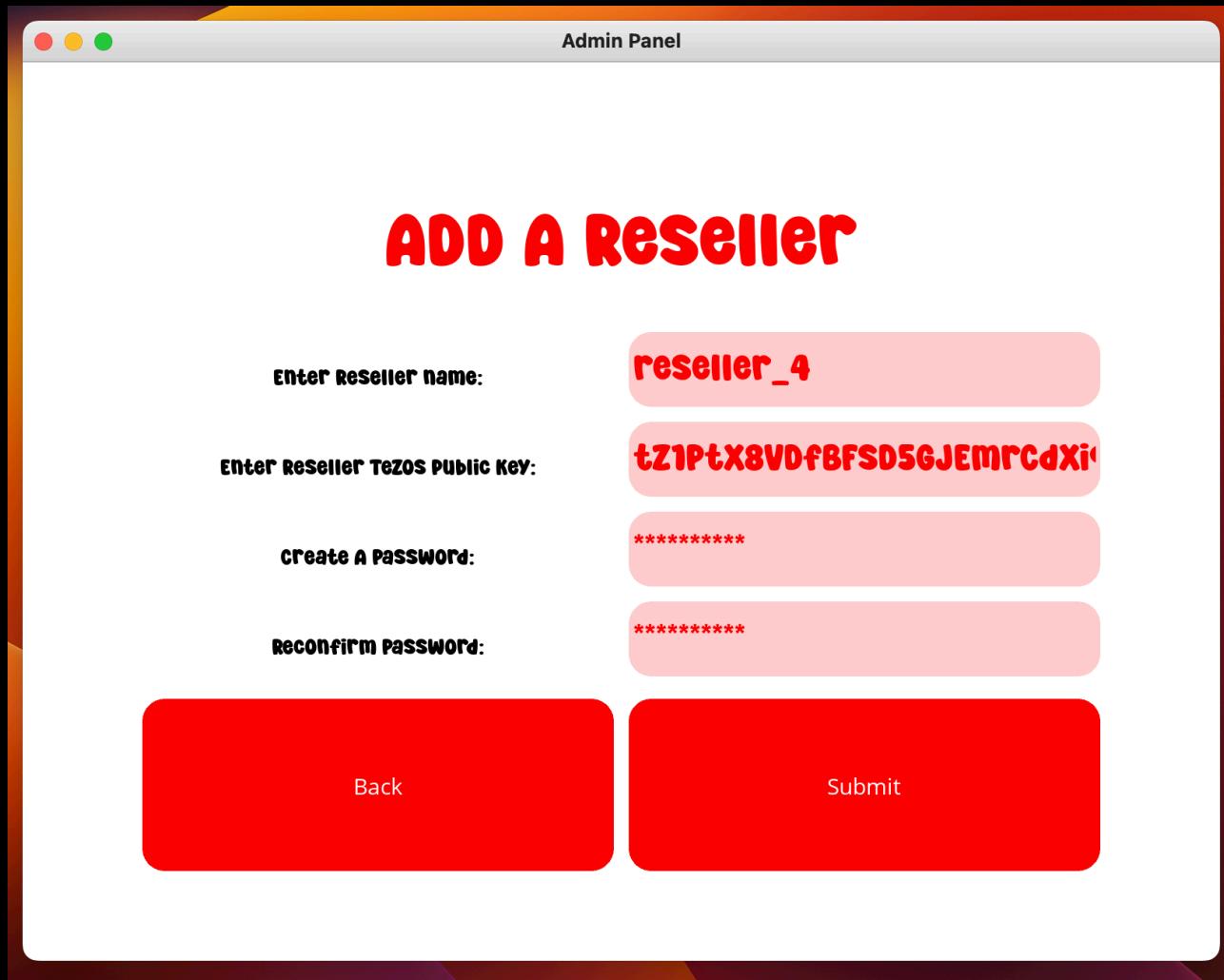
# MOVING A TICKET



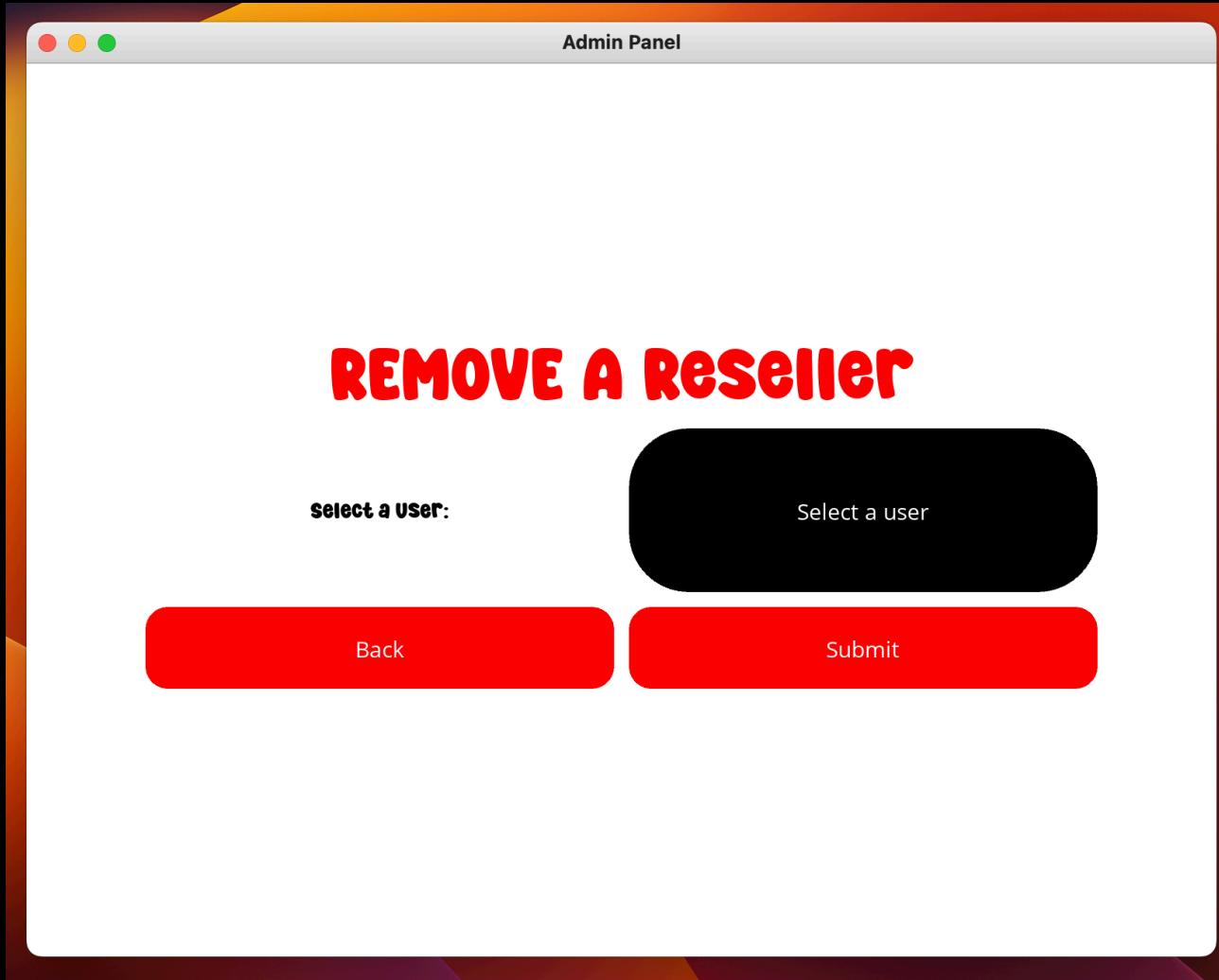
# SELLING TICKETS IN LOCAL AND REMOVING FROM TICKETS



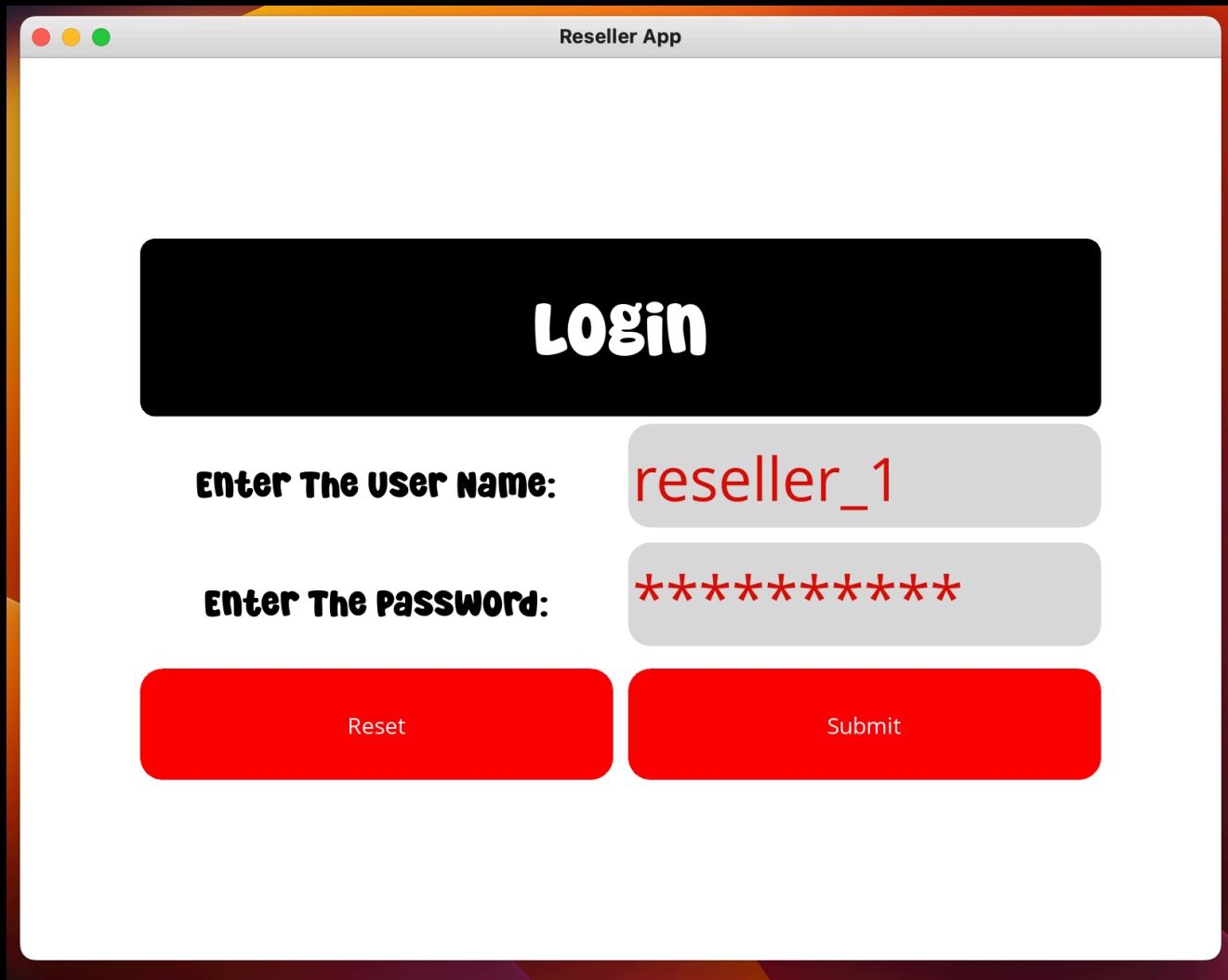
# ADDING A RESELLER



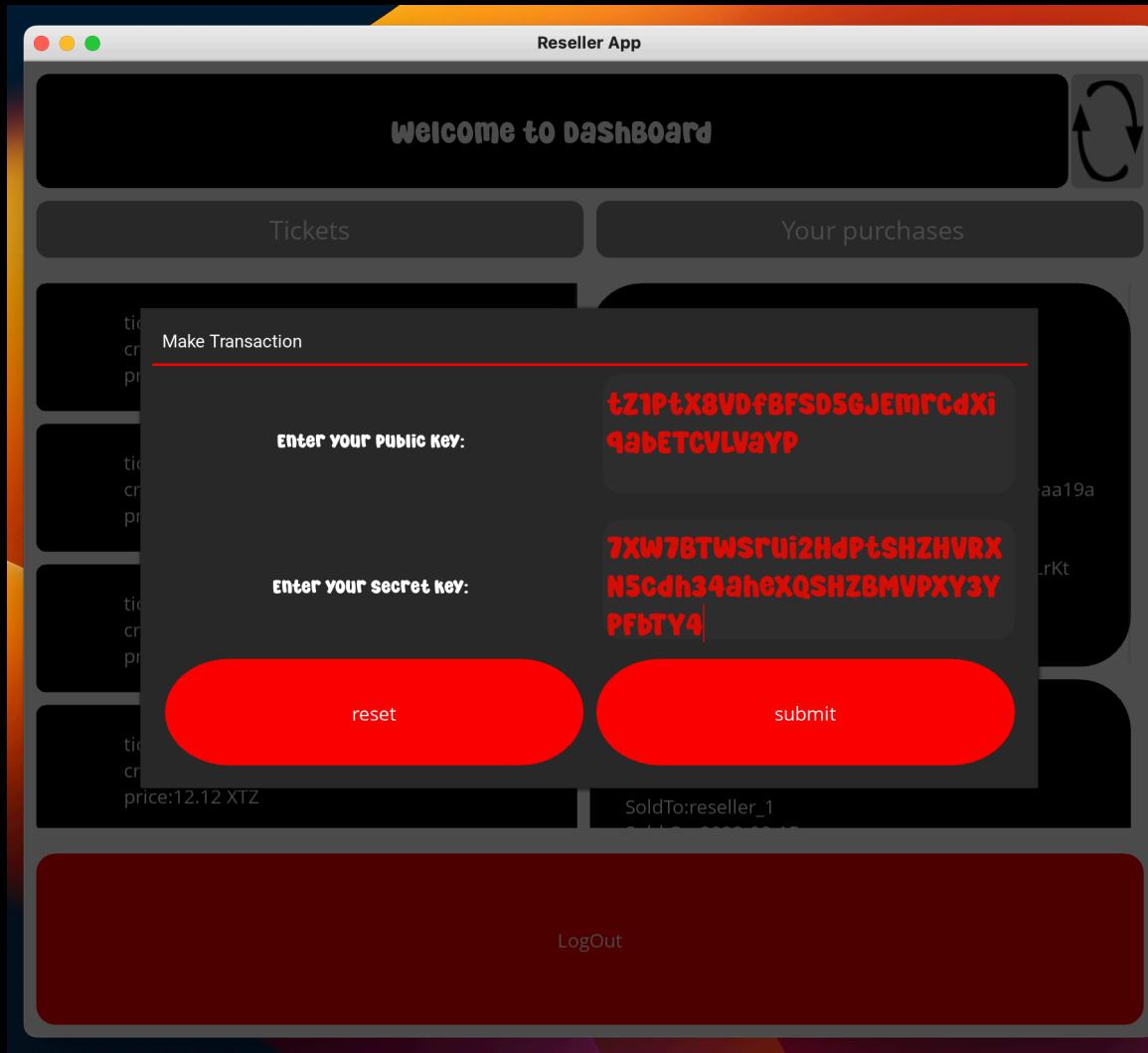
# MOVING A RESELLER



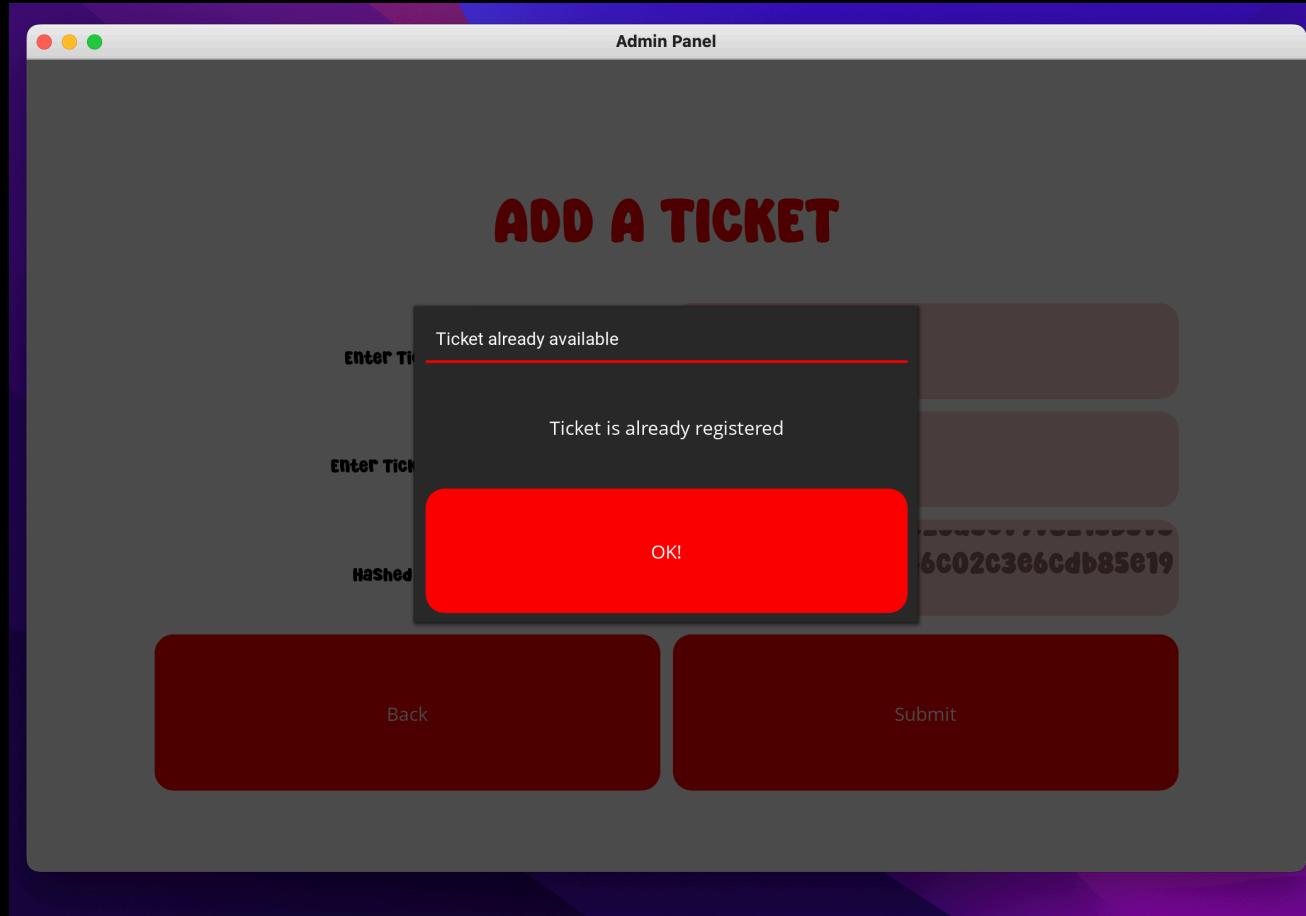
# LOGIN IN (RESELLERS)



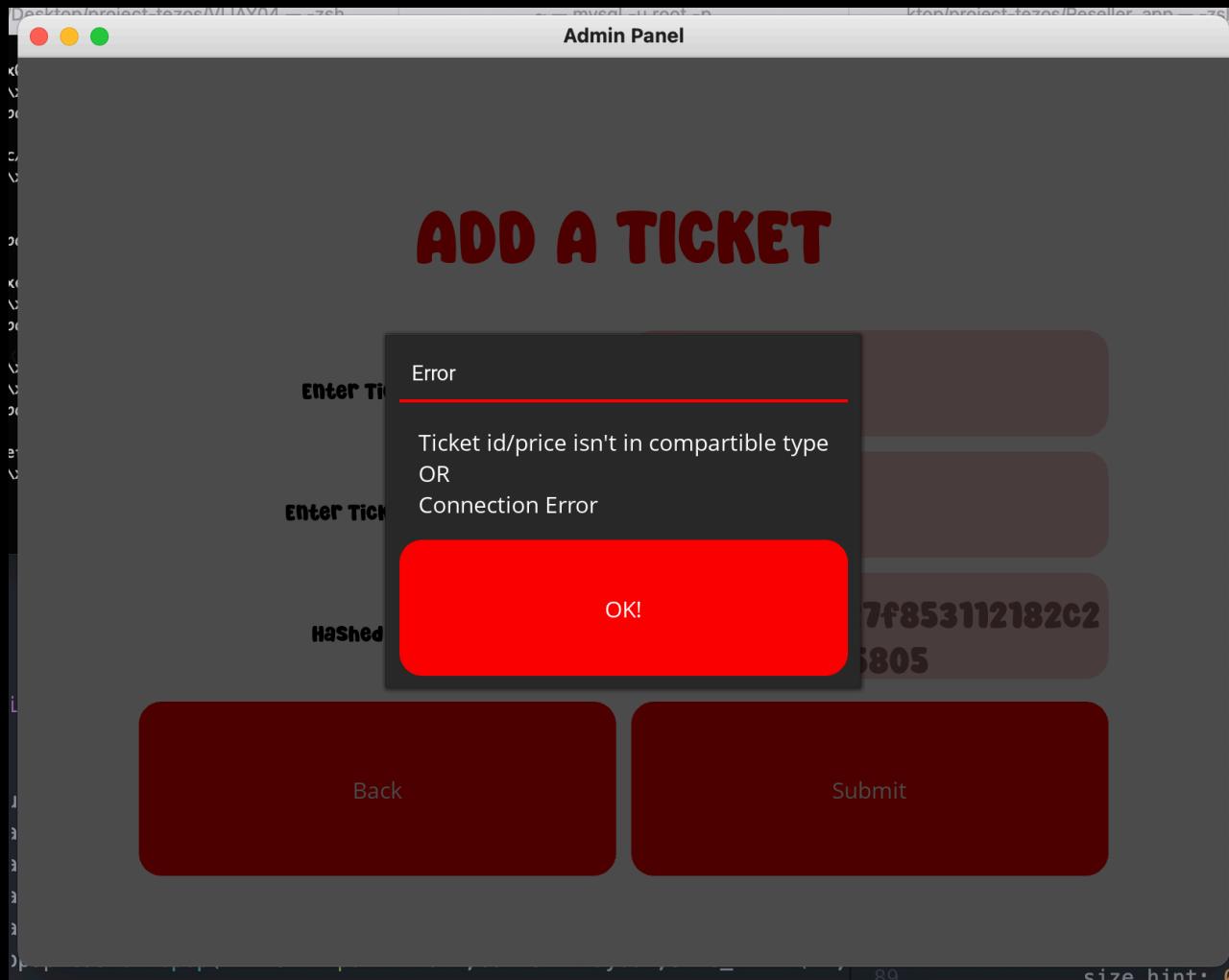
# BUYING A TICKET (RESELLER)



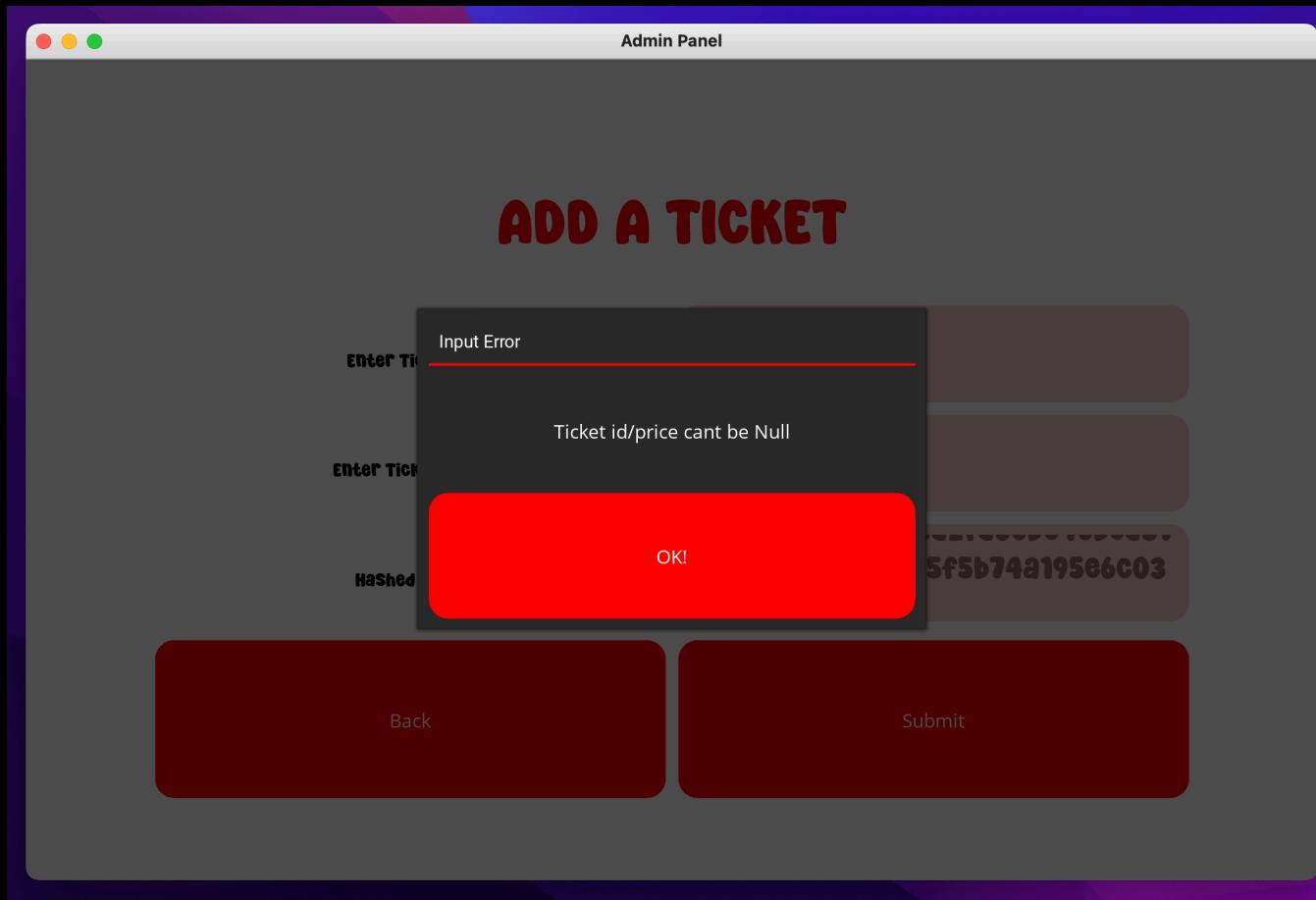
# ERRORS HANDLING : DUPLICATE TICKETS



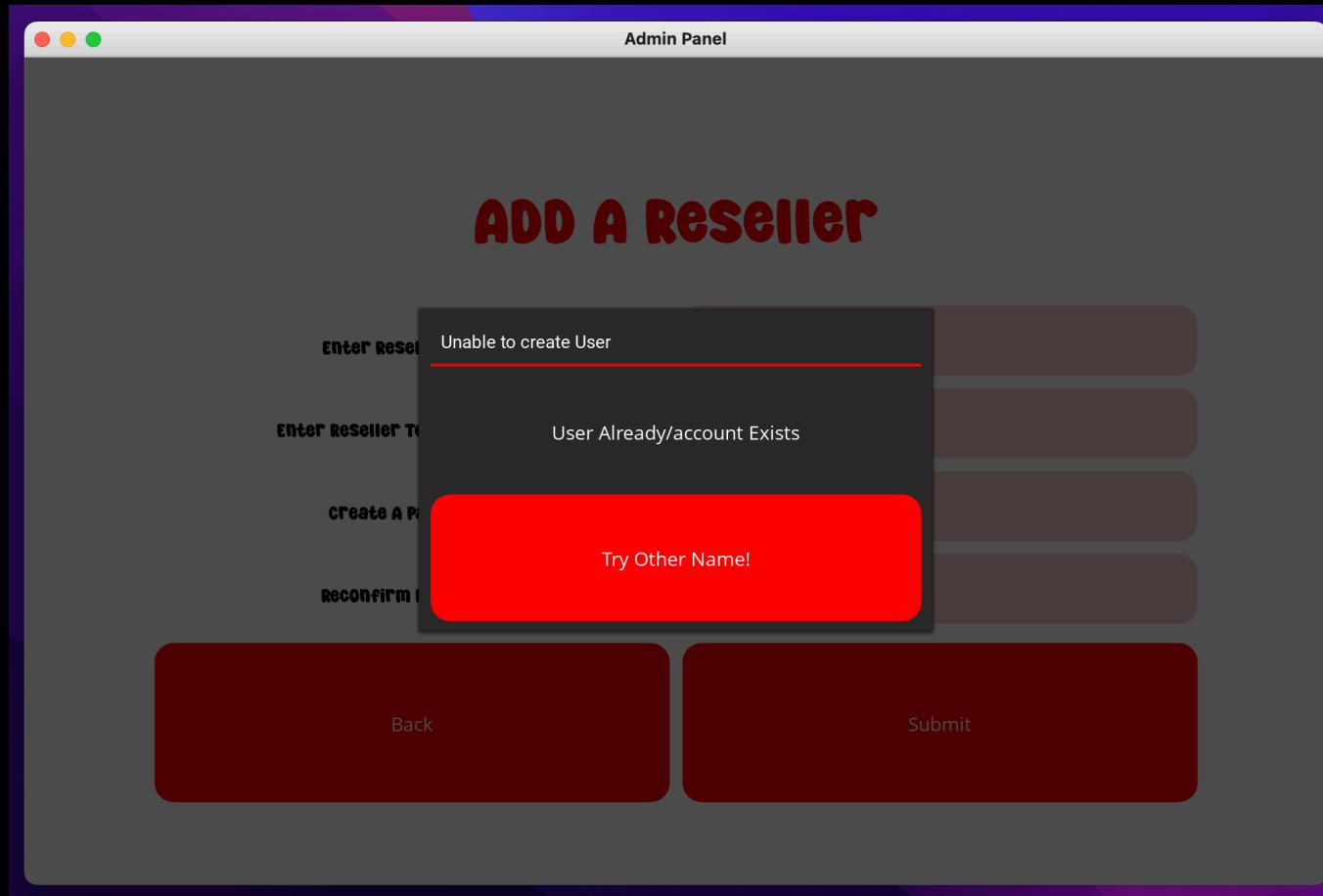
# ERRORS HANDLING :NON-Compatible type



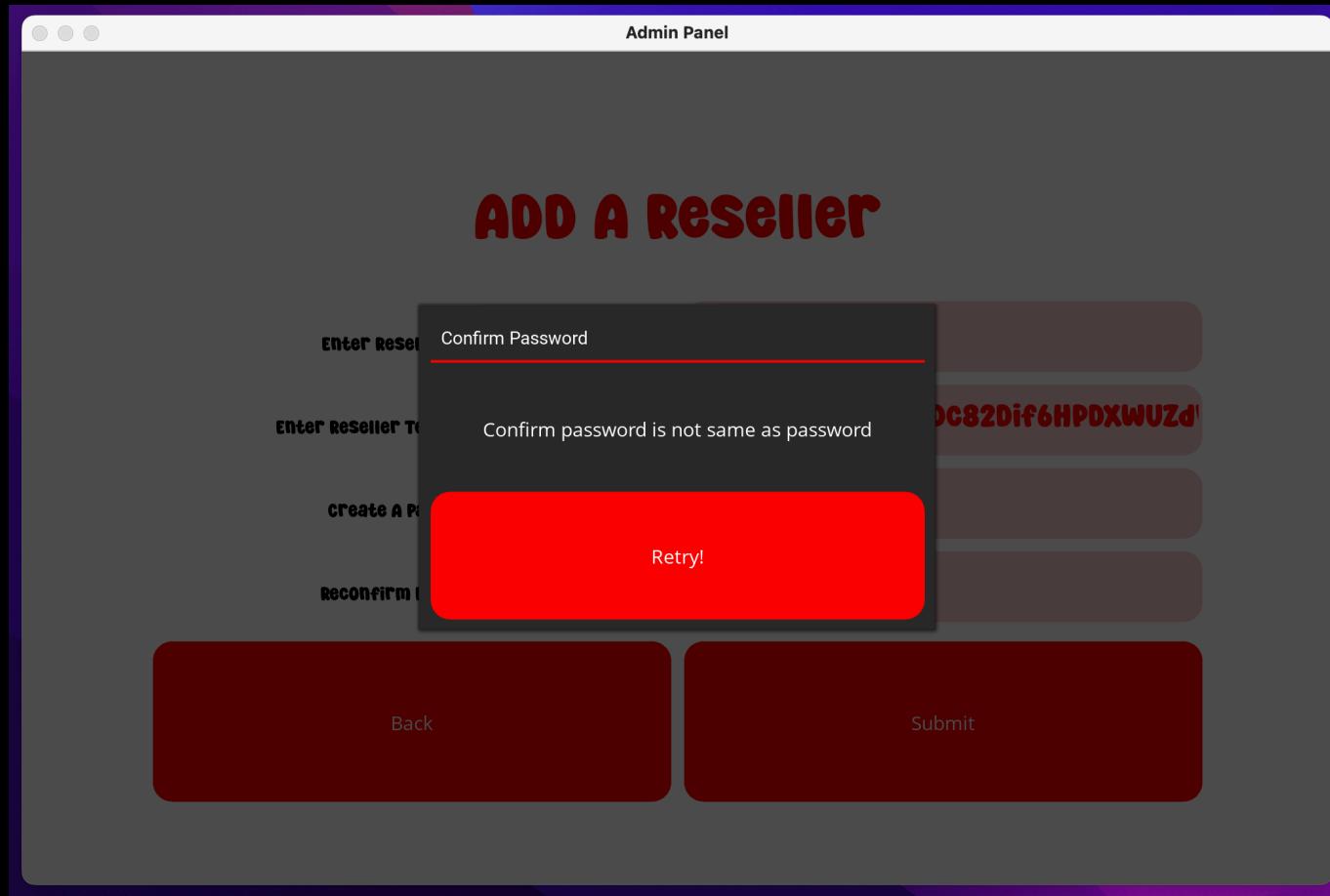
# ERRORS HANDLING : NULL ENTRY



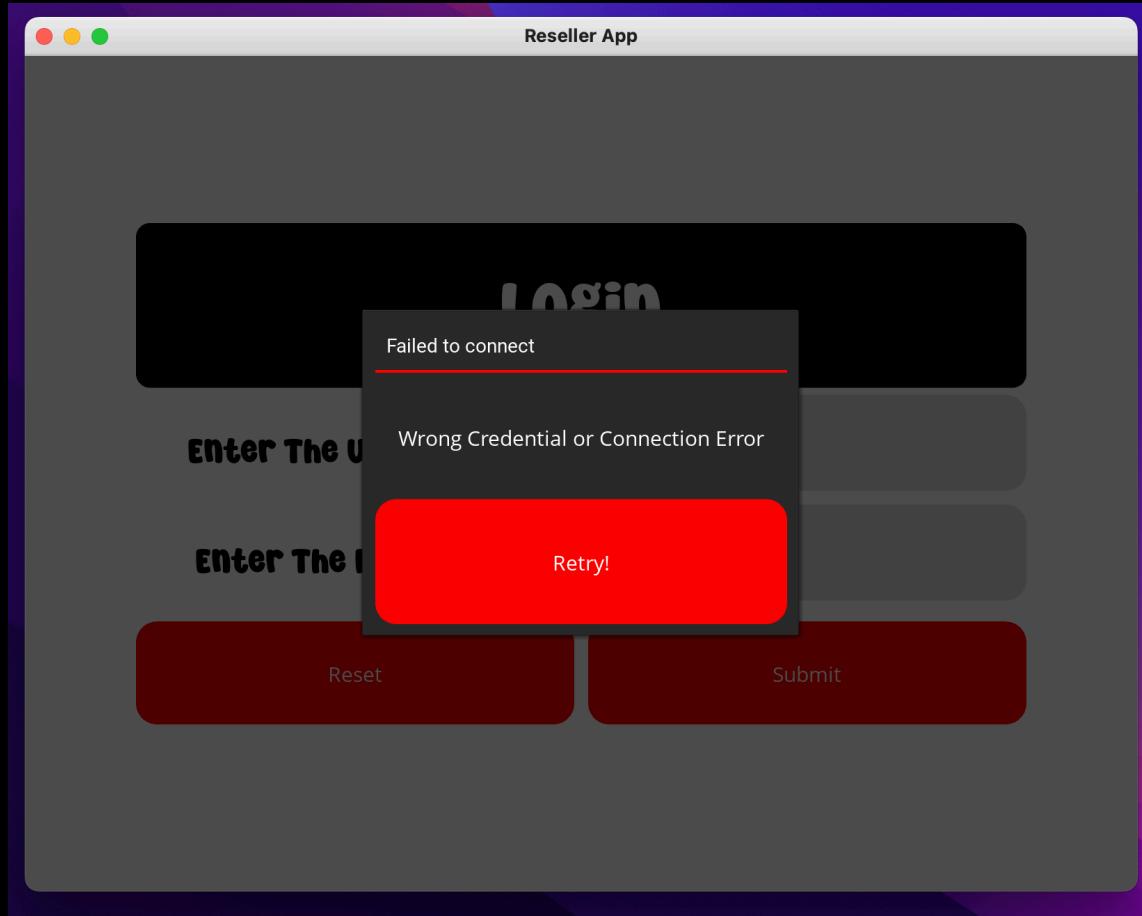
# ERRORS HANDLING :DUPLICATE USER



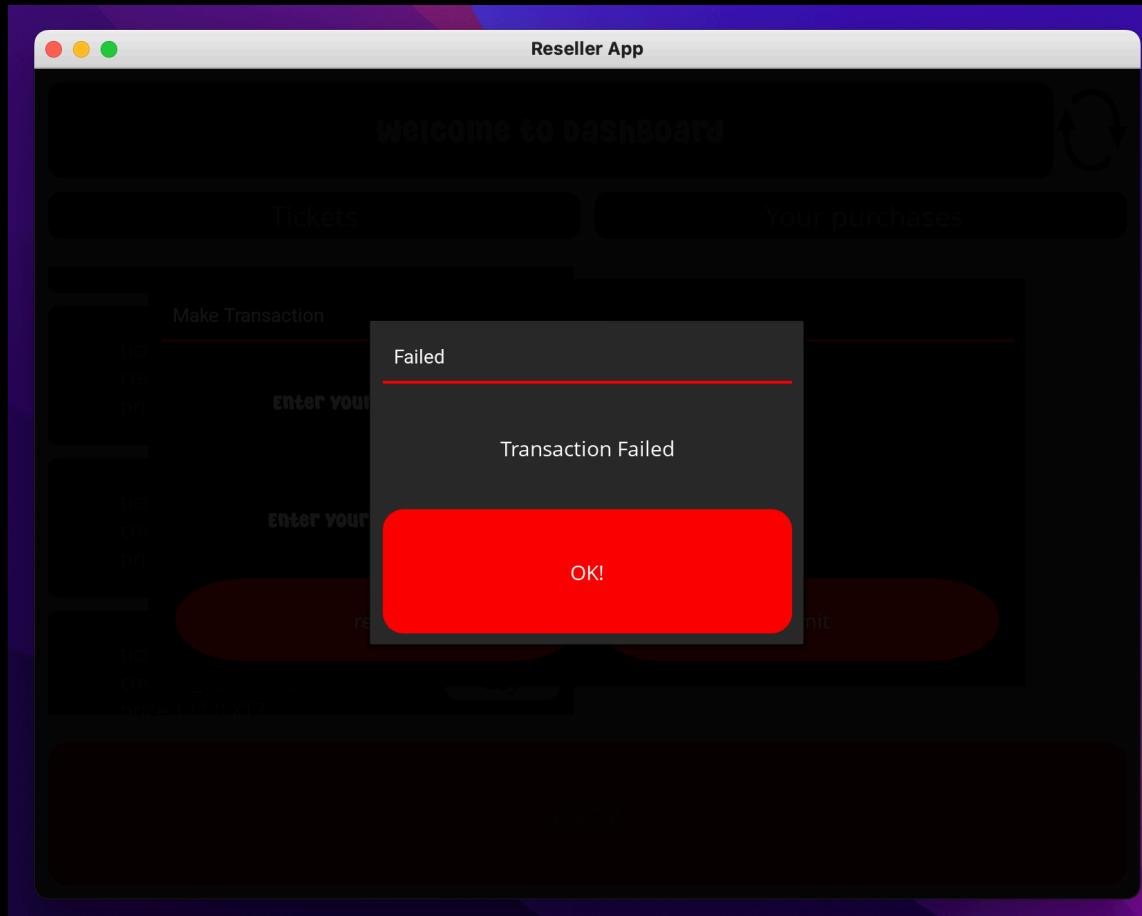
# ERRORS HANDLING :FAILED CONFIRMATION



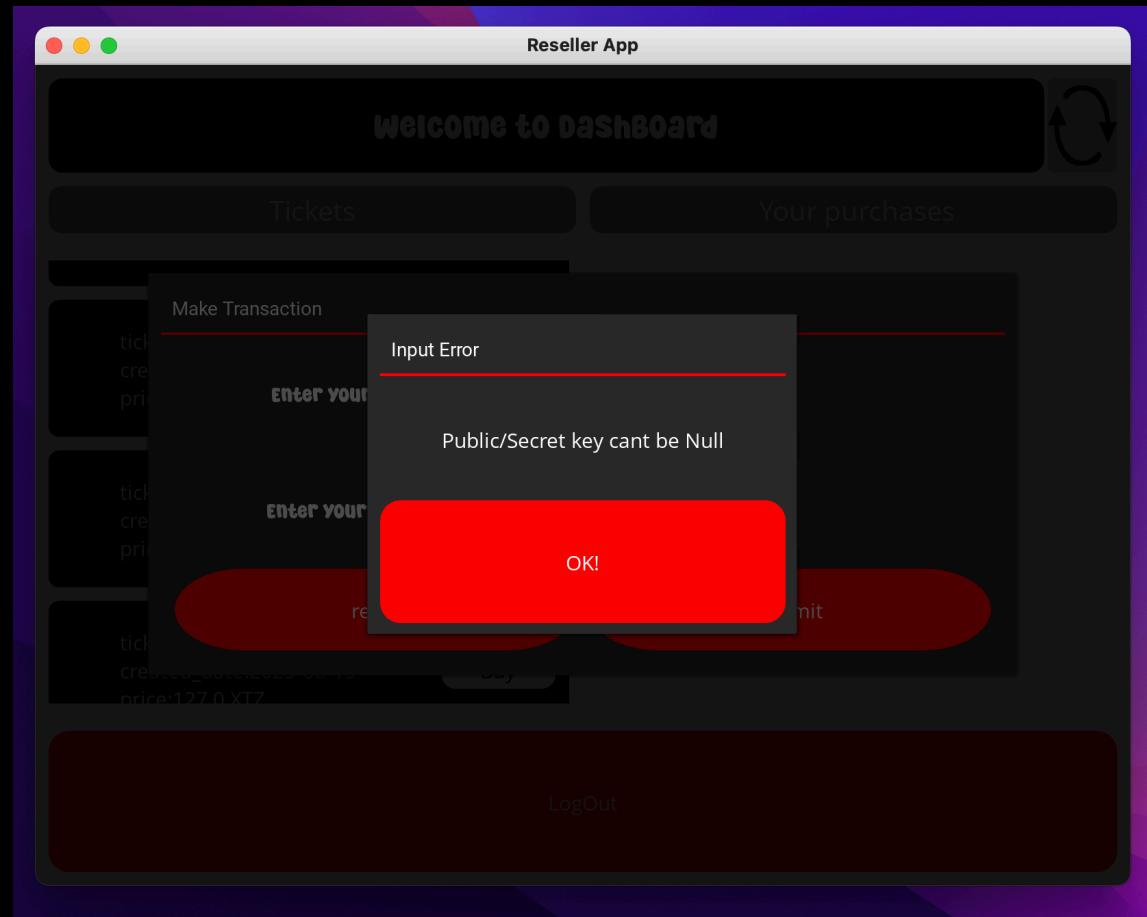
# ERRORS HANDLING :INVALID CREDENTIAL



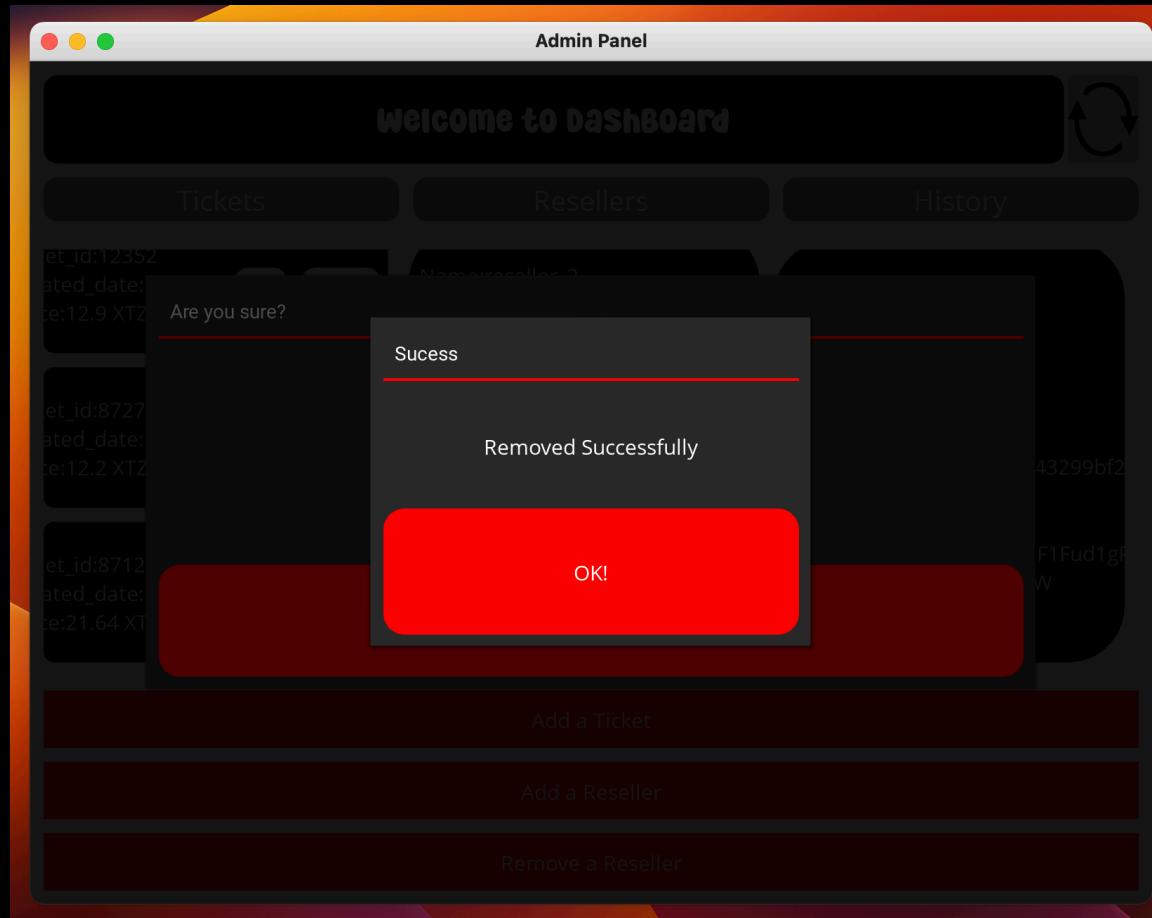
# ERRORS HANDLING : TRANSACTION FAILED



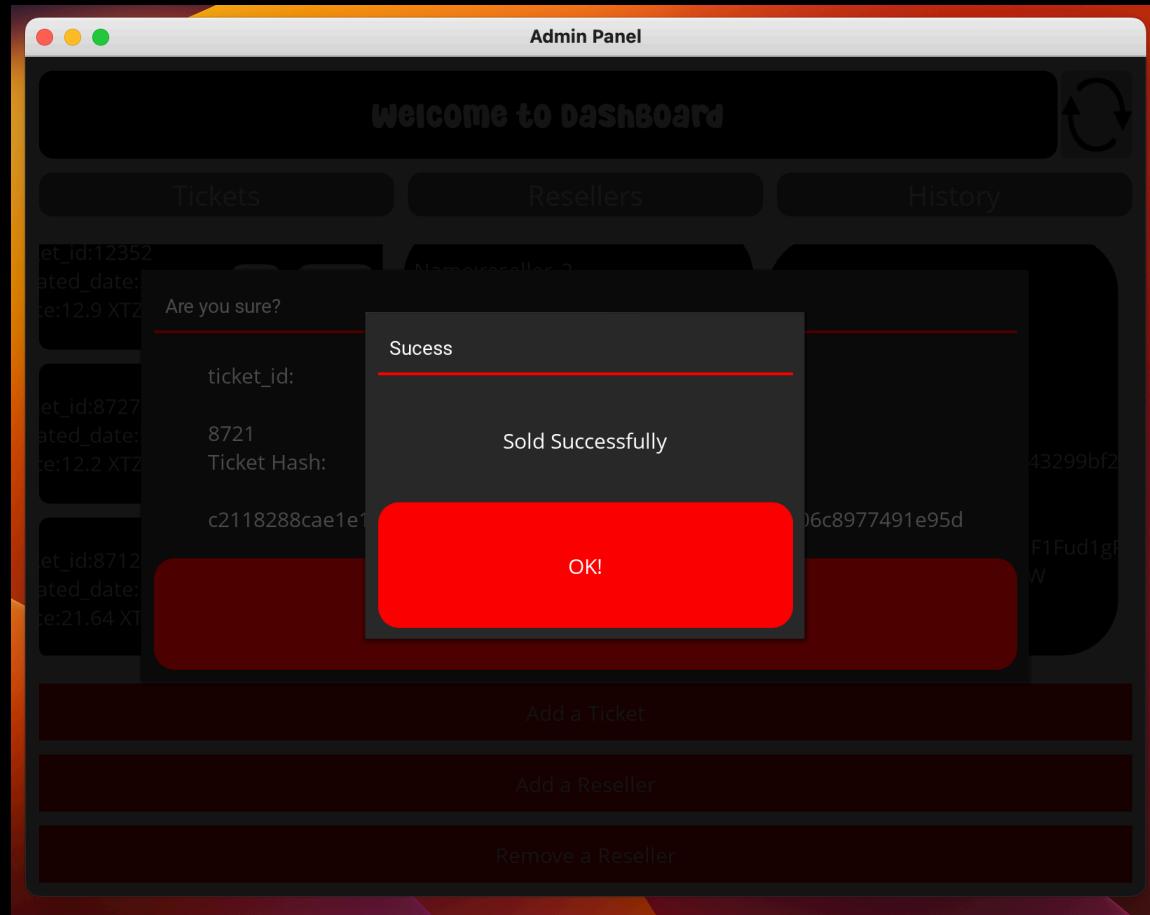
# ERRORS HANDLING :NULL INPUT



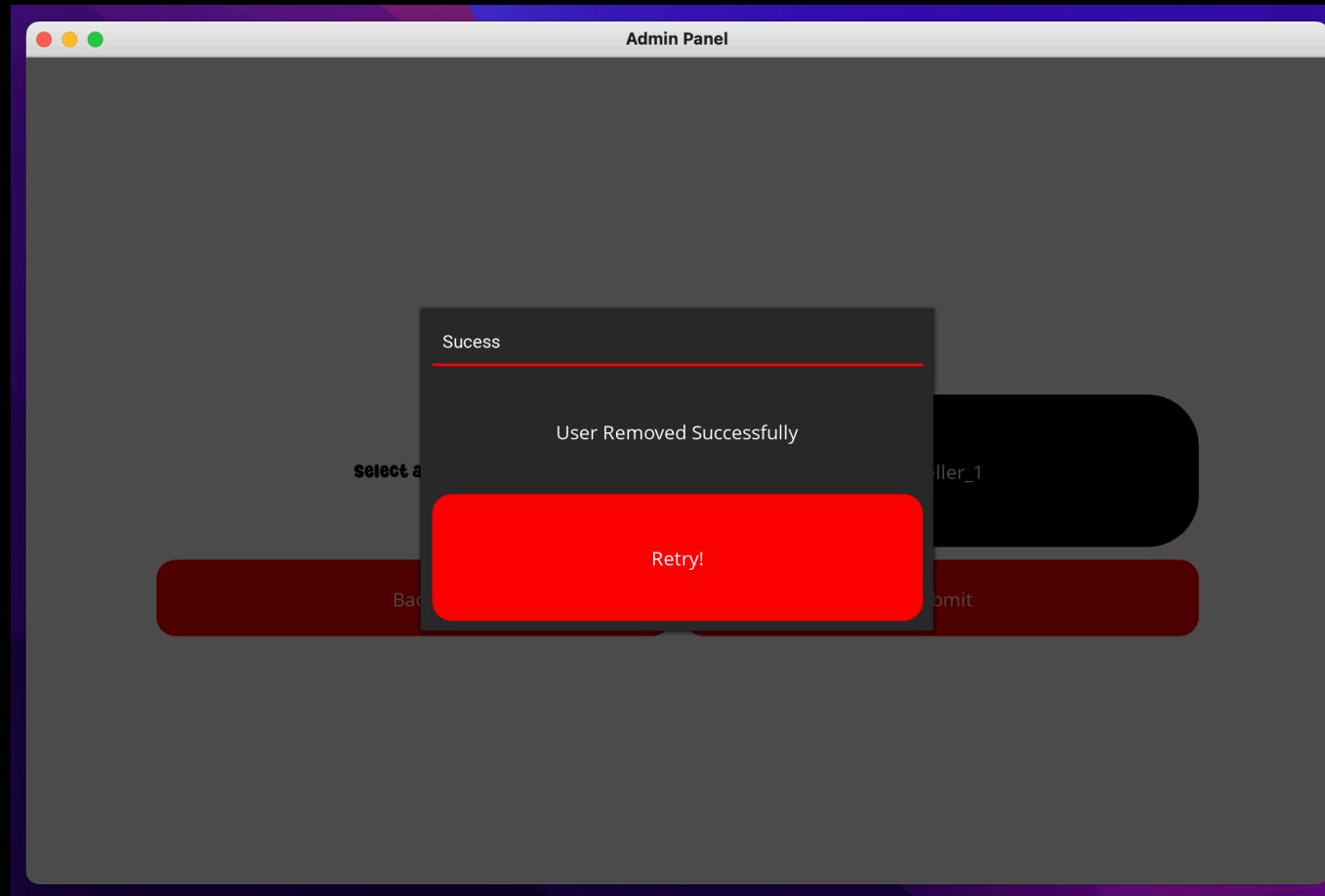
# SUCCESS ALERT :REMOVING TICKET



# SUCCESS ALERT :SELLING TICKET IN LOCAL



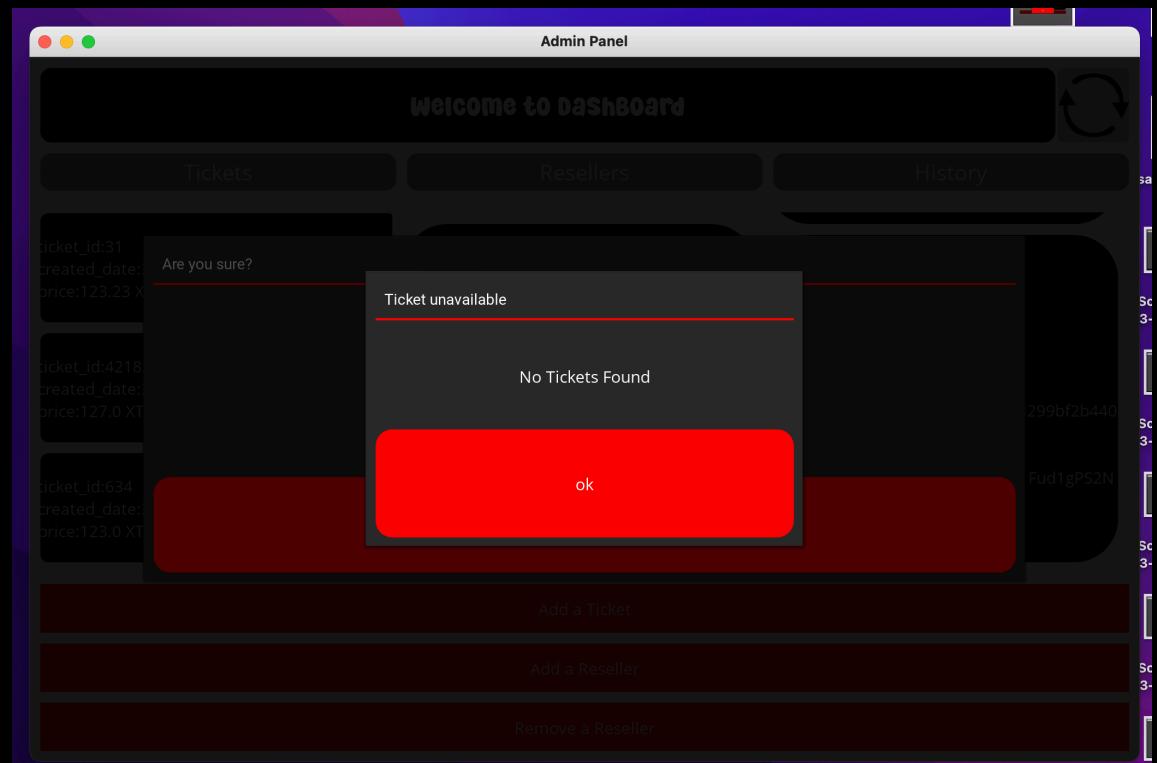
# SUCCESS ALERT :REMOVING RESELLER



# WHAT IF USER DELETE TICKET AND RESELLER MAKING A TRANSACTION?

ANSWER:

1. while making a transaction on a ticket where a reseller try to buy it, we used a function `refresh_connection()` which refreshes data every time and for every clicks.



# ERROS HANDLING

- ACCOUNTS WITH **TEZOS ACCOUNT** CAN ONLY BE ADD TO A RESELLERS,  
UNAUTHORISED USERS CANT BE ADDED
- **NULL** ENTRY(GIVES WARNING)
- **FAILED TRANSACTION ALERT**
- **ATOMICITY** IN ALL DATA OF DATABASE(NO DUPLICATE ACCOUNTS CAN BE  
CREATED)
- **TRANSACTION AND SELLING** CANT BE DONE IN SAME TIME, WHERE WE  
REFRESH EVERY DATA IN THE DATABASE FOR EVERY EVENT.