RadBid - A Blockchain-based E-Auction System

Presented By:

Aditya Kumar Bej, Guneet Mummaneni, Komal Bakshi, Madhumitha Santhanakrishnan, Shristi Suman

UCDAVIS
COMPUTER SCIENCE

Table of Contents

- Features of the application

- Tech Stack

- Functionality Flow

- Smart Contract Logic

- UI Screenshots

Features

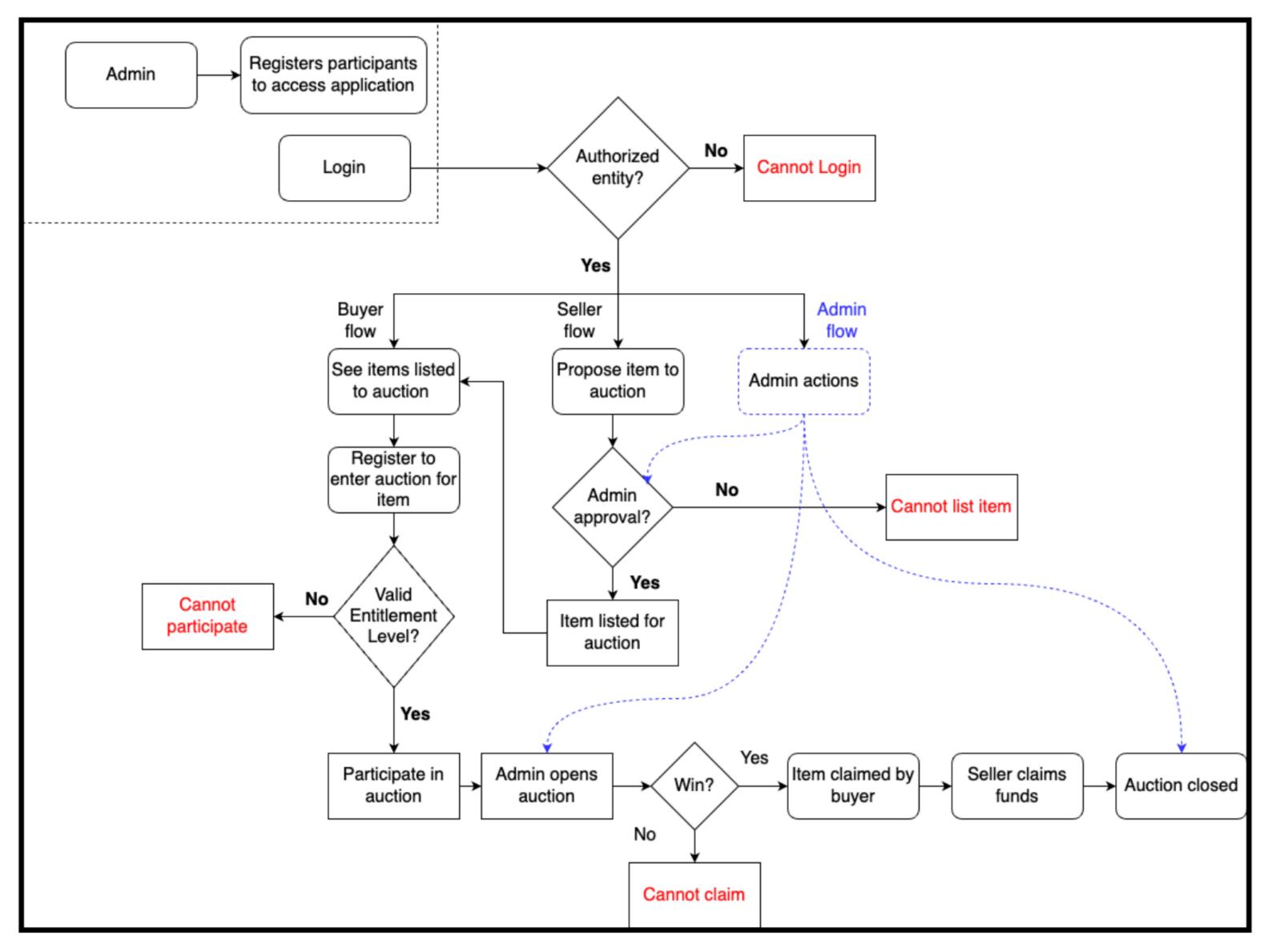**Core of it - Decentralized Auction Application**

- English-type auction application with an open bid system

- Actors in the system - **Admin and Participants** (Buyers/Sellers)

- Authorized access - Only registered participants can access the application

- Managing User Entitlements - **Access Control** using **Badges**

- Register an auction item for listing and place quick bids as participants (buyers)

- View the list of available and sold-out auction items with a purchase log for Admin and Participant

Tech Stack

What did we use?

- Frontend - **React.js**

- Backend/Middleware - **Node/Express**

- Ledger - **Radix as a ledge**r

- Smart Contracts - **Scrypto**

- Local Simulator - **Resim**

- Live Network - **Radix Babylon Alphanet**

FUNCTIONALITY FLOW

High-level working of the auction application

UC DAVIS
COMPUTER SCIENCE

Smart Contracts
Authorization

Blueprint - Authorization		
Struct	**Parameters**	**Description**
UserBadge	username: **String**	Approved members of the protocol
UserAuth	admin_badge_address: **ResourceAddress**	Given to admins to make permissioned method calls
	temp_badge_address: **ResourceAddress**	Given to participants requesting to be authorized to access the application
	component_badge_vault: **Vault**	Badge stored in vault of this component to mint or burn tempbadge and userbadge NFTs
	pending_users: **HashMap<String,NonFungibleId>**	Record of pending users for approval. String-username
	approved_users: **HashMap<NonFungibleId, NonFungibleId>**	Record of approved users.
Methods	**Arguments**	**Description**
instantiate_user_auth	**ComponentAddress, Bucket**	Makes Component addressable and returns admin badge via Bucket
approve_user	username: **String**	Approves user
claim_user	temp_badge: **Bucket**	Approved user to claim minted UserBadge NFT
request_user	username: **String**	Requests being an user

Smart Contracts
Participants

Blueprint - Participants		
Struct	**Parameters**	**Description**
Participants	username: **String**, firstname: **String**, lastname: **String**, middleinitial**: String ,** title: **String,** email: **String,** phone: **String,** dob: **String,** gender: **String,** tag: UserBadge **Non Fungible ID**, items_bought: **NonFungibleId**, items_sold: **NonFungibleId**	Capturing additional details of users through a form
Related Functions		
get_list_of_approved_users	Get list of all approved users from the Users Vault	
get_list_of_pending_users	Get list of all pending users from the Users Vault. This list will allows admins to approve pending users into the system	
get_particular_user	Fetch details of a particular approved user to generate their user profile page or so	

Smart Contracts
Auction Listings

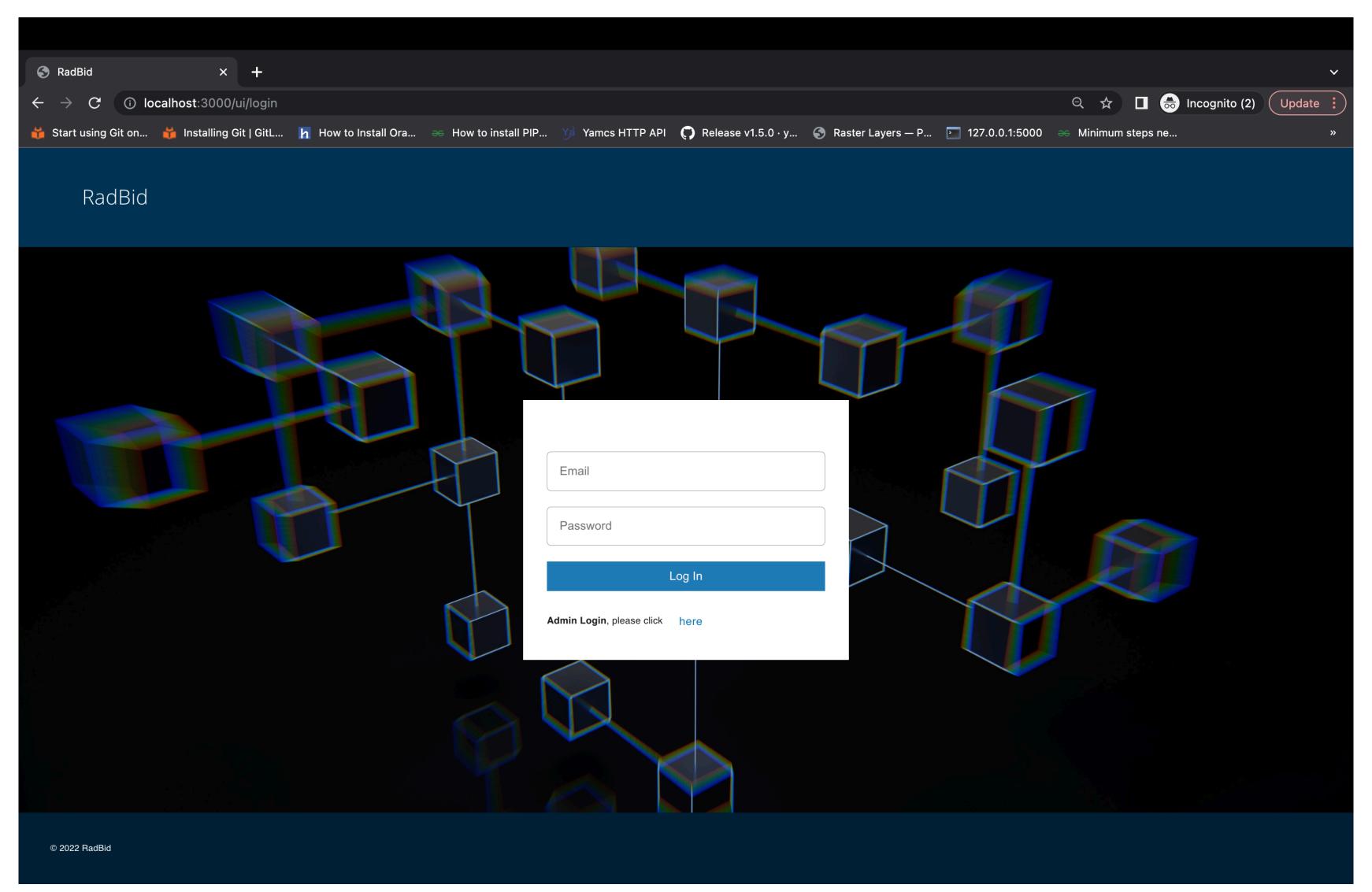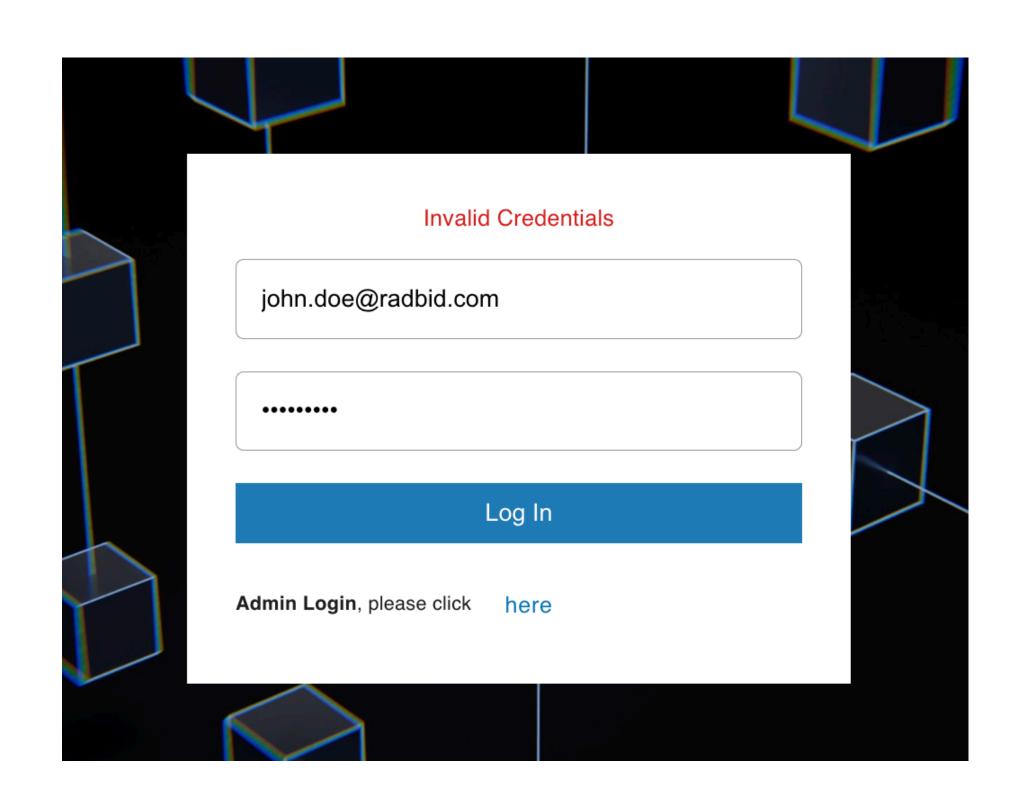| Blueprint - Auction Listings		
Struct	**Parameters**	**Description**
Auction Item Registration	item_name: **String,** Tags: **String**, Status: **Enum ("Sold", "Active", "Cancelled"),** date_listed: **String,** date_sold: **String,** expiry_period: **String,** sold_price: **Decimal,** last_bid_price: **Decimal,** activeListing: **Bool**	Capturing details of the item to be listed for auction
Related Functions		
get_all_listings	List down all the auction items	
get_particular_item	Get a particular auction item	
update_particular_listing	Update a particular auction item such as changing status	

Smart Contracts
Auction Process

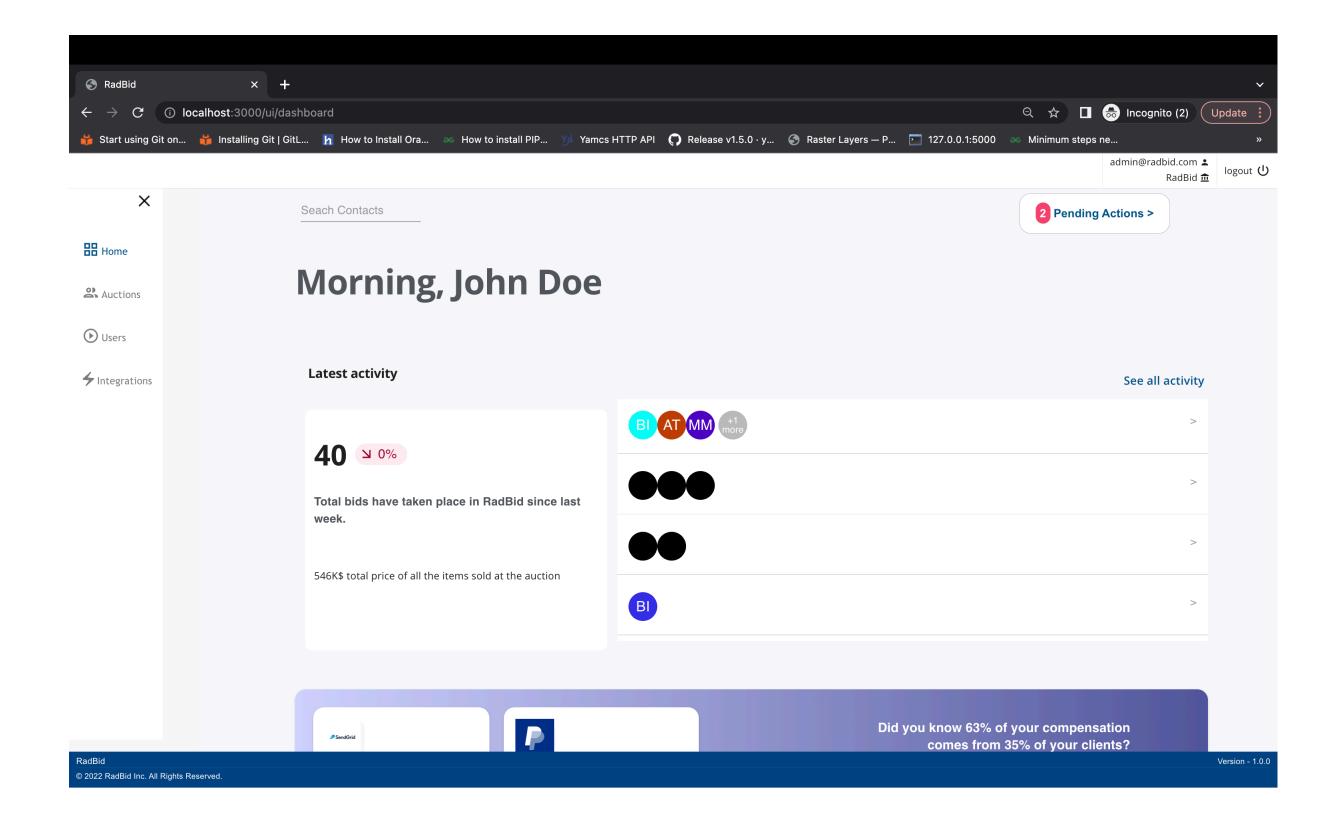| Blueprint - Auction Process		
Struct	**Parameters**	**Description**
Start Auction	items_vaults: **HashMap<NonFungibleId, Vault>,** bid_vaults: **HashMap<NonFungibleId, Vault>,** payment_vault: **Vault,** bidders_badge: **ResourceAddress,** auction_time: **u64,** state: **Enum ("Active", "Inactive", "Cancelled")**	Blueprint to start an english auction
Functions		
instantiate_auction	Instantiates a new english auction for the passed items. Returns **ComponentAddress** and **Bucket.**	
bid_auction	Allows a participant to bid in this auction	
increase_bid	Increases the bid price of the item once a participant raises bid	
cancel_bid	Cancel bid. Ensure conditions such as auction settlement, state of the auction are checked before cancellation	
claim_funds	Claim NFTs from the component post auction settlement	
check_auction_settlement	Check function to ensure auction is settled. Called before cancelling auction	

Running UI Instance - Local
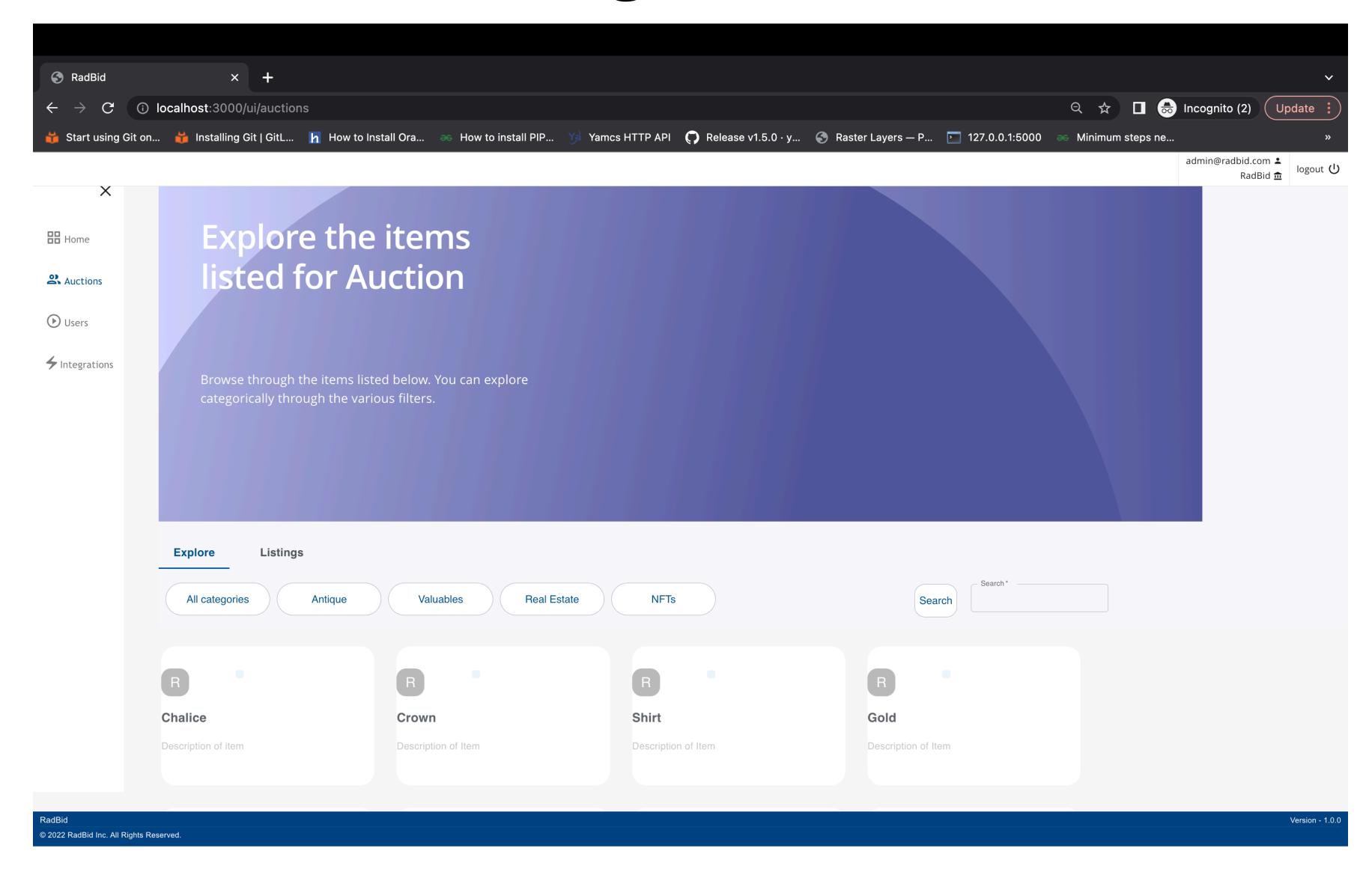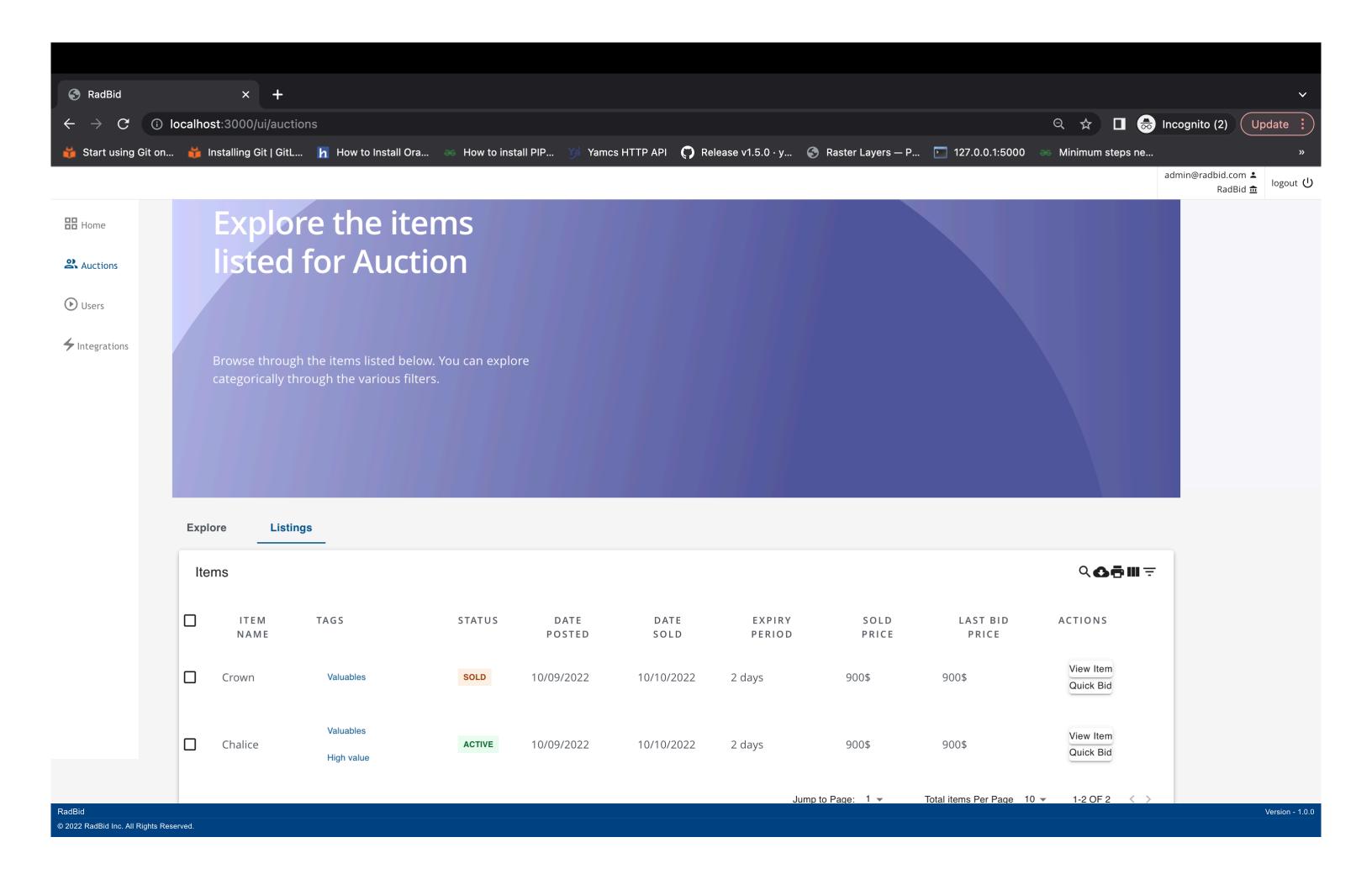
Login Page

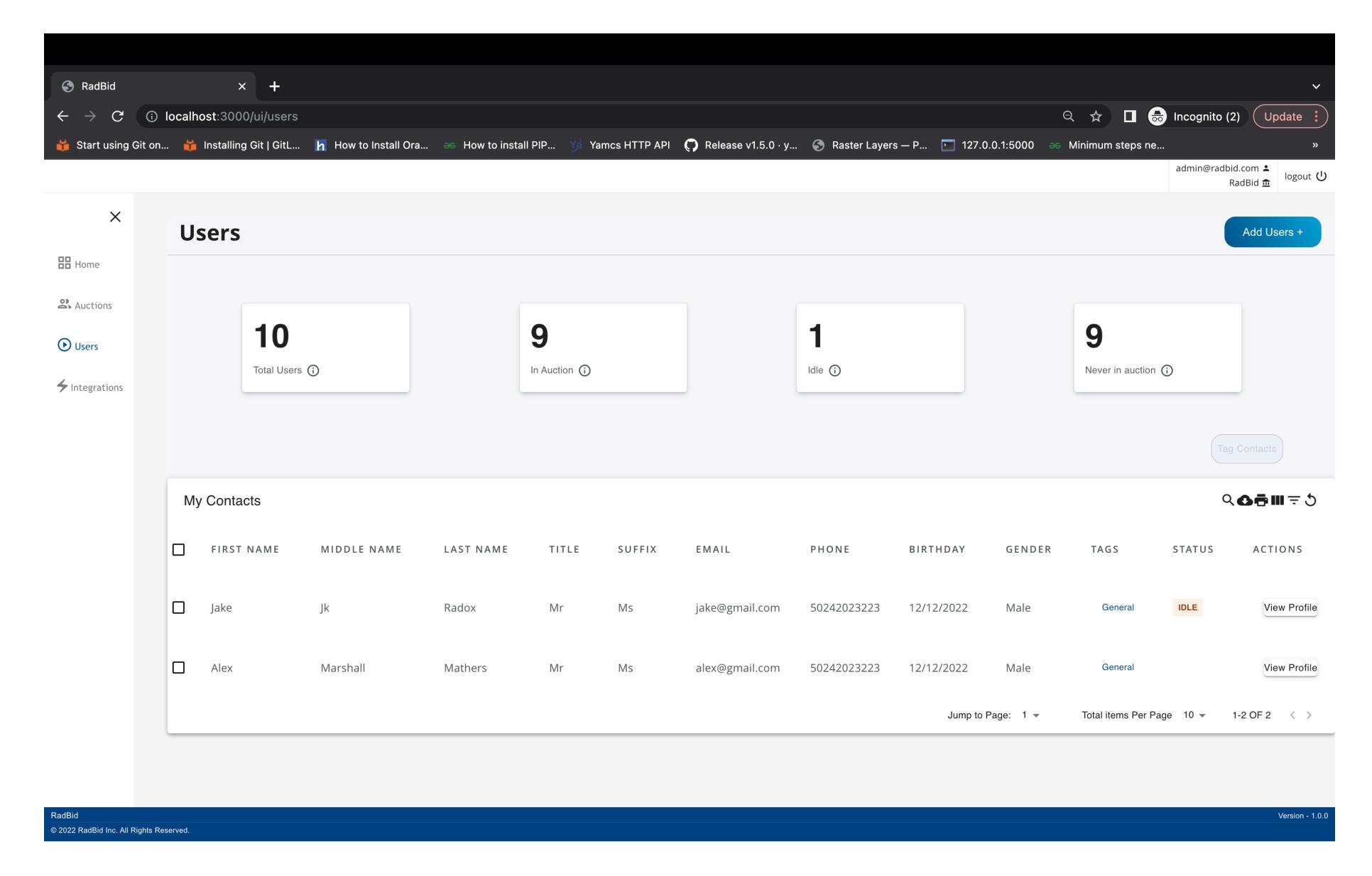# Invalid vs Valid Authorization

Auction Items Listings

Auction Items Listings

Users Page - Admin

FUTURE WORK

END