

RadBid - A Blockchain-based E-Auction System

Presented By:

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

Table of Contents

- Features of the application
- Tech Stack
- Functionality Flow
- Smart Contract Logic
- UI Screenshots

Features

Core of it - Decentralized Auction Application

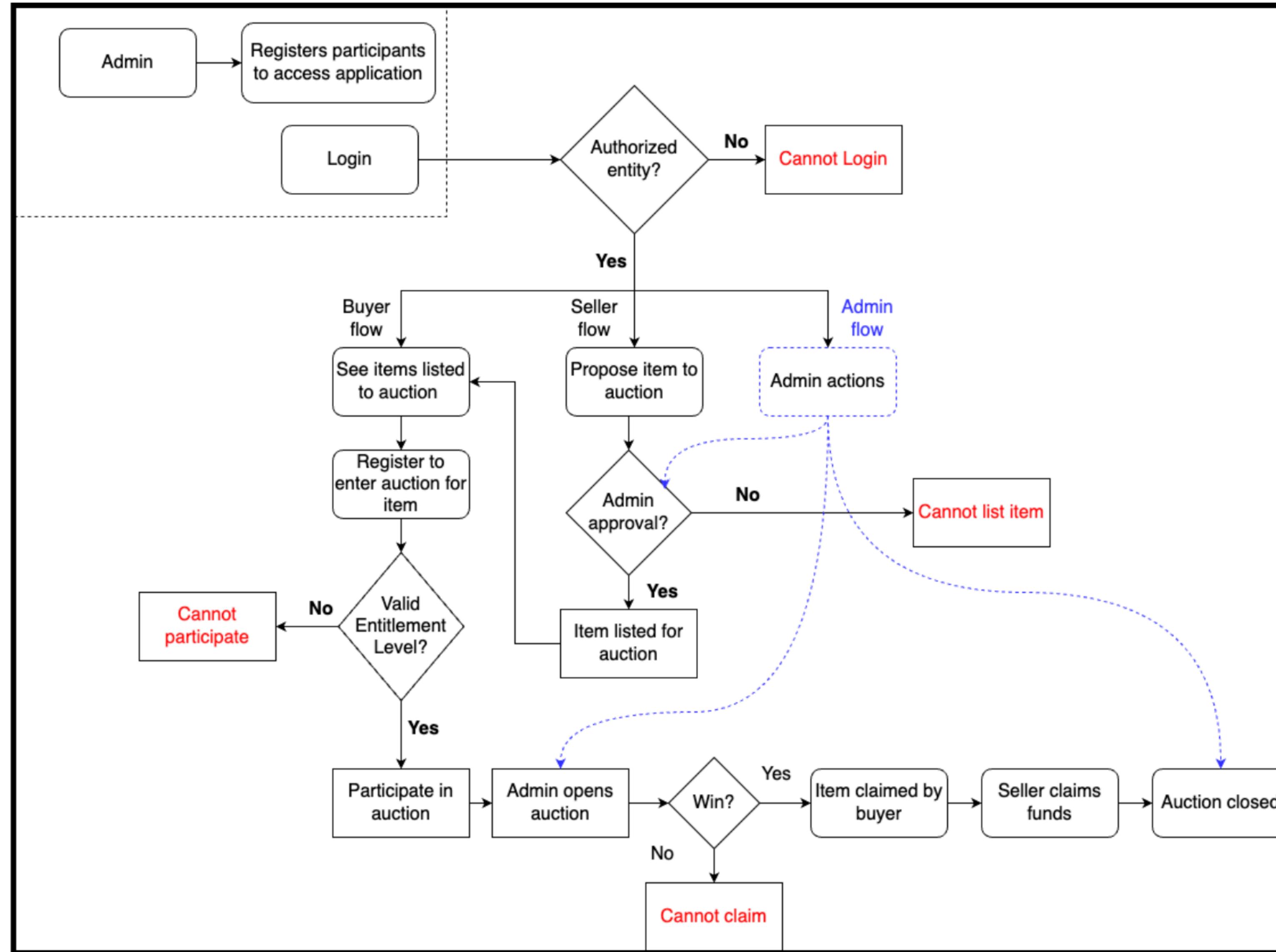
- English-type auction application with an open bid system
- Actors in the system - **Admin and Participants** (Buyers/Sellers)
- Authorized access - Only registered participants can access the application
- Managing User Entitlements - **Access Control using Badges**
- Register an auction item for listing 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 ledger**
- Smart Contracts - **Scrypto**
- Local Simulator - **Resim**
- Live Network - **Radix Babylon Alphanet**

FUNCTIONALITY FLOW



High-level working of the auction application

Smart Contracts

Authorization

Blueprint - Authorization		
Struct	Parameters	Description
UserAuth	username: String	Approved members of the protocol
	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 NonFungible 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	<code>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</code>	Capturing details of the item to be listed for auction
Related Functions		
<code>get_all_listings</code>	List down all the auction items	
<code>get_particular_item</code>	Get a particular auction item	
<code>update_particular_listing</code>	Update a particular auction item such as changing status	

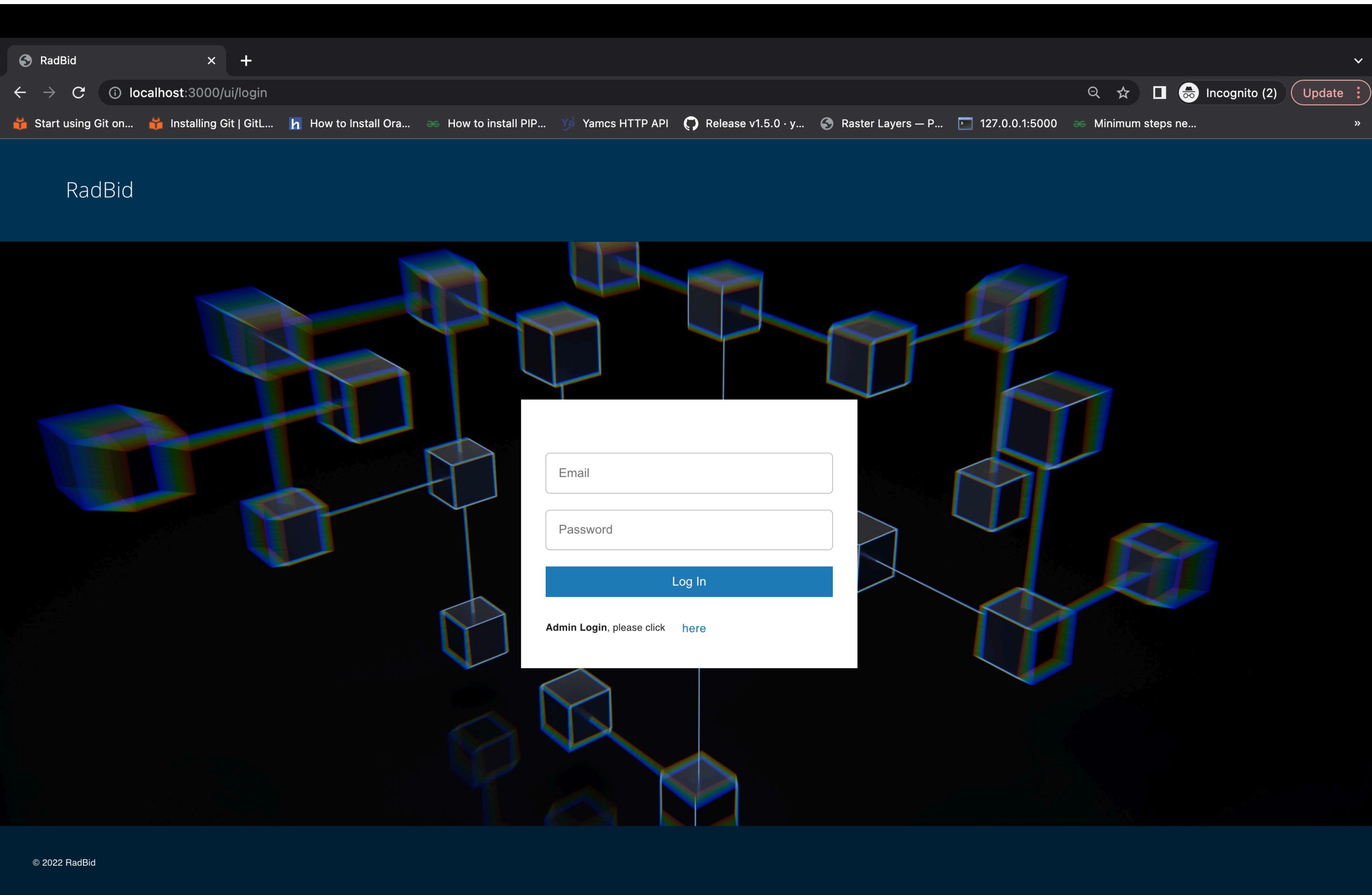
Smart Contracts

Auction Process

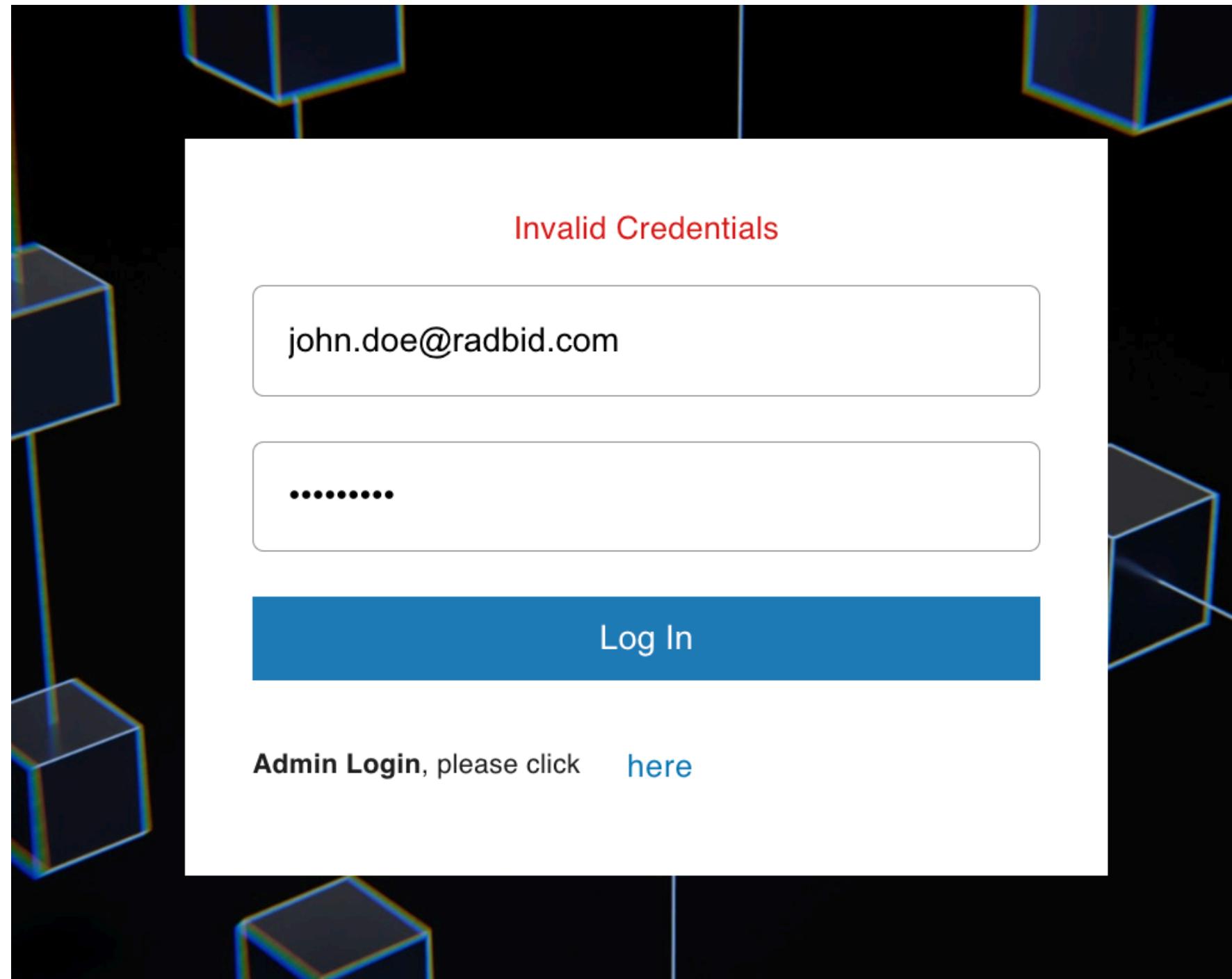
Blueprint - Auction Process		
Struct	Parameters	Description
Start Auction	items_vaults: HashMap<NonFungibleId, Vault> , bid_vaults: HashMap<NonFungibleId, Vault> , payment_vault: Vault , bidders_badge: ResourceAddress , auction_time: u64 , state: Enum ("Active", "Inactive", "Cancelled")	Blueprint to 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



A screenshot of a web-based dashboard for "RadBid". The browser address bar shows "localhost:3000/ui/dashboard". The top navigation bar includes links for "Home", "Auctions", "Users", and "Integrations". On the right, there is a sidebar with a "Pending Actions" notification (2). The main content area displays a greeting "Morning, John Doe" and a section titled "Latest activity". It shows a summary: "40" (with a -0% change), "Total bids have taken place in RadBid since last week.", and "546K\$ total price of all the items sold at the auction". At the bottom, there is a purple footer bar with logos for "SendGrid" and "PayPal", and a statistic: "Did you know 63% of your compensation comes from 35% of your clients?". The footer also includes the text "Version - 1.0.0".

Auction Items Listings

The screenshot shows the 'Auctions' section of the RadBid application. At the top, there's a large purple banner with the text 'Explore the items listed for Auction'. Below the banner, there's a sub-headline: 'Browse through the items listed below. You can explore categorically through the various filters.' On the left, a sidebar menu includes 'Home', 'Auctions' (which is selected), 'Users', and 'Integrations'. The main content area has tabs for 'Explore' (selected) and 'Listings'. Under 'Explore', there are buttons for 'All categories', 'Antique', 'Valuables', 'Real Estate', and 'NFTs'. A search bar with a placeholder 'Search*' is also present. Below these, four items are listed in cards: 'Chalice' (Description of item), 'Crown' (Description of Item), 'Shirt' (Description of Item), and 'Gold' (Description of Item). Each card features a small circular icon with a letter 'R' and a light blue dot. At the bottom, a footer bar contains the text 'RadBid', '© 2022 RadBid Inc. All Rights Reserved.', and 'Version - 1.0.0'.

Auction Items Listings

The screenshot shows the RadBid application interface for managing auction items. The top navigation bar includes links for various documentation and tools, such as 'Start using Git on...', 'Installing Git | GitL...', 'How to Install Oracle...', 'How to install PIP...', 'Yamcs HTTP API', 'Release v1.5.0 · y...', 'Raster Layers — P...', '127.0.0.1:5000', and 'Minimum steps ne...'. The user is logged in as 'admin@radbid.com'.

The main content area features a large blue banner with the text 'Explore the items listed for Auction'. Below the banner, a message encourages users to 'Browse through the items listed below. You can explore categorically through the various filters.'

The interface includes a sidebar with navigation links: Home, Auctions, Users, and Integrations. The 'Listings' tab is currently selected, showing a table of auction items:

ITEM NAME	TAGS	STATUS	DATE POSTED	DATE SOLD	EXPIRY PERIOD	SOLD PRICE	LAST BID PRICE	ACTIONS
Crown	Valuables	SOLD	10/09/2022	10/10/2022	2 days	900\$	900\$	View Item Quick Bid
Chalice	Valuables High value	ACTIVE	10/09/2022	10/10/2022	2 days	900\$	900\$	View Item Quick Bid

At the bottom, there are pagination controls: 'Jump to Page: 1', 'Total items Per Page: 10', '1-2 OF 2', and 'Version - 1.0.0'.

Users Page - Admin

The screenshot shows the RadBid Admin interface on a web browser. The title bar indicates the site is running on localhost:3000/ui/users. The top navigation bar includes links for various documentation and tools, and a user account section for admin@radbid.com.

The main content area is titled "Users". It displays four summary cards:

- Total Users: 10
- In Auction: 9
- Idle: 1
- Never in auction: 9

Below these cards is a "My Contacts" section with a table of user data:

	FIRST NAME	MIDDLE NAME	LAST NAME	TITLE	SUFFIX	EMAIL	PHONE	BIRTHDAY	GENDER	TAGS	STATUS	ACTIONS
<input type="checkbox"/>	Jake	Jk	Radox	Mr	Ms	jake@gmail.com	50242023223	12/12/2022	Male	General	IDLE	<button>View Profile</button>
<input type="checkbox"/>	Alex		Marshall	Mr	Ms	alex@gmail.com	50242023223	12/12/2022	Male	General		<button>View Profile</button>

At the bottom of the page, there are navigation links for "Jump to Page", "Total items Per Page", and "1-2 OF 2". The footer contains copyright information for RadBid Inc. and the version number 1.0.0.

FUTURE WORK

END