NFT Artwork Auction

Presented by:
Leigh Badua, Loc Thai, Amine Baite, Christopher Diamond, Eunice Huang

Executive Summary

NFT auction application allows digital asset owner registering NFT for auction, public bidding on NFT, and inspector to authenticate NFT via Python, Blockchain and Smart Contract.

Python serves front end communication with users. Blockchain provides benefits of decentralization, distributed architecture, trust, record keeping, and transparency of bidding/asset transfer history. Smart Contract sets a set of rules and allows credible transactions of digital assets without third parties.

The deApp provides trust via decentralized network without central governance and transaction efficiency and transparency of NFT creator/owner exchange history.

Project Approach

- Define User Story → Identify steps and task → coding/testing/integration

Streamlit (Front End)

NFT Bidding
Input: Enter Bid
Output: HexBytes Trx#

NFT Register
Input: Asset info
Output: HexBytes Trx#

SQL (DataStorage)

Data Storage

Blockchain (Python)

- Check account balance
- sign/send transaction
- Send contract/transaction info

Smart Contract (Solidity)

- Highest Bid requirement
- Credible transaction and asset transfer

- Application requirements
- Tokenize Property

Testing/Evaluating/codes

```python
# Pass the connection string to the SQLAlchemy create_engine function
engine = sqlalchemy.create_engine(database_connection_string, echo=True)

create_nft_table = """
CREATE TABLE nft_info (
    "filename"          VARCHAR(50),
    "filepath"          VARCHAR(50),
    "filesize"          VARCHAR(50),
    "Owner_Name"        VARCHAR(50),
    "Public_Key"        VARCHAR(200),
    "Asset_name"        VARCHAR(50),
    "bid_start_amount"  INT,
    "Bid_close_date"    VARCHAR(50)
  );
 """

engine.execute(create_nft_table)



#EH: Get seller information
st.header('NFT Submission for Auction Application')


username=st.text_input(label='Username')
public_key=st.text_input(label='Public Key')
asset_caption=st.text_input(label='Asset Name')
bid_start=st.number_input("Enter Desired Bid Start amount in Token",min_value=1000,step=1000)

close_date_request = st.date_input(
     "Please enter bid close date(UTC)",
     datetime.now()+timedelta(days=7),min_value=datetime.now() +timedelta(days=7))
st.write('Your close date (UTC) request is:', close_date_request)


def load_image(image_file):
    img = Image.open(image_file)
    return img

def insert_data(nft_df):
    for index, row in nft_df.iterrows():
        engine.execute("INSERT INTO nft_info(filename, filepath, filesize, Owner_Name, Public_Key, Asset_name, bid_start_amount, Bid_close_date) values(?,?,?,?,?,?,?,?)", row.filename, row.filepath,
            row.filesize,row.Owner_Name, row.Public_Key, row.Asset_name, row.bid_start_amount, row.Bid_close_date)

    sql_nft_info_df = pd.read_sql_table('nft_info', con=engine)

    st.write(sql_nft_info_df)
```

filetype

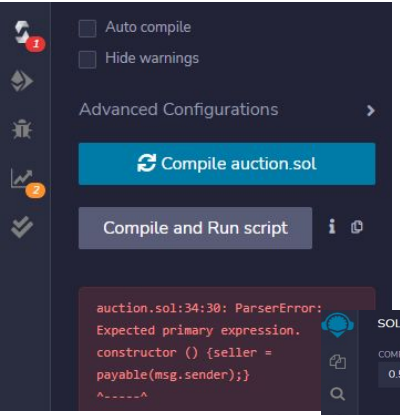
filepath

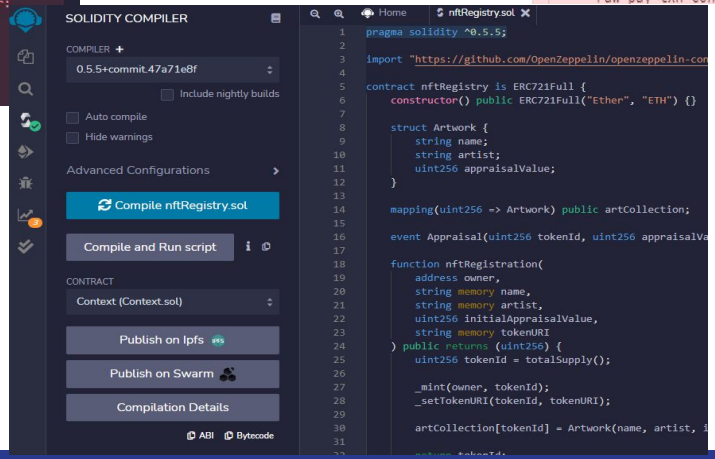
You are screen sharing

Stop

Testing/Evaluating/codes

- Smart Contract compiler/Deploy error messages





- Streamlit python errors

TypeError: Unsupported type: '<class 'numpy.float64'>'. Must be one of: bool, str, bytes, bytearrayor int.

Traceback:

```
File "C:\Users\eunic\anaconda3\envs\dev\lib\site-packages\streamlit\scriptrunner\script_runner.py", line 554, in _run_script
    exec(code, module.__dict__)
File "nft_sale_stl.py", line 126, in <module>
    raw_pay_txn=contract.functions.pay_property(public_key1).buildTransaction(payload)
```

```
# "st.session_state object:",st.session_state

if 'prev_bid' not in st.session_state:
    st.session_state['prev_bid']=nft_database[nft_option][1]

if 'counter' not in st.session_state:
    st.session_state['counter']=0

if 'current_bid' not in st.session_state:
    st.session_state['current_bid']=nft_database[nft_option][1]

if 'bid_history' not in st.session_state:
    st.session_state['bid_history']=pd.DataFrame(columns=["Asset
```

Demo

Challenges

- Streamlit session state between input selections
- SQL–datatype, syntax, url address
- Solidity, python datatype sync-up issue such int value
- Testing through remix, ganache, Metamask, streamlit

Next Step

- Take bidder deposit
- Monetize platform by taking fee from sales
- Use Pinata to store digital asset and release after sale
- Final approval requirement in Smart Contract

Thank you