Connor Kealey, Alden Roy

http://flip1.engr.oregonstate.edu:15650/

Executive Summary:

Below will be a summary of all of our changes from step 1 to now. In the ERD diagram at the very bottom, we have changed the two non-solid lines to solid lines to be consistent. We're not sure why this happened but we fixed it. We changed "Trasaction_id" to "Transaction_id", we also changed "medium" to "Medium" even though they will all be lowercase in the database, we might as well have them all consistent because of autocorrect. We changed the spelling of "adress" to "address". "Transaction_has_Frames" is now "Transaction_Has_Frames" for consistency in our document.

Next we changed our attribute names to lowercase in our actual database, not in the doc because of autocorrect, because at least they're consistent now. Then based on the feedback, several reviewers noticed our Address variables was missing from the artists table in the schema, so we added that in. We also had some crossing lines that were fixed to improve visibility of the diagrams. Reviewers mentioned that we were missing some cascade operations from our DDL.SQL file, so those were added in. There were also several capitalizations errors in the schema and ERD that we changed to make everything looks better and to improve consistency.

Someone said, "The Sales page is kind of confusing, as an Administrator are you going to update the sales page? I think you would want this to be the Transactions page showing transactions and maybe just deleting transactions/sales or adding a sale. I think you are implementing the sale page to create transactions?" This was just a bug on our part, we don't have a sales page we got rid of it and put transactions in. It was just linking an incorrectly named page from our prototype names. Someone said we had no delete on cascade, so we added delete on cascade to all our entities. Then someone said, "I think the Sales tab in the HTML UI is not consistent with the schema, perhaps change the Sales to Transaction. Also the intersection table's patron_id has the frame data.", so we changed frame data and made the schema consistent by deleting the sales tab. We fixed the error where our inserted data wasn't able to be deleted so now all data in patrons can be deleted correctly. We added auto filling data to the update form when update is clicked to make the UI better as well.

Project Outline and Overview:

The Corvallis Art Gallery has an annual art sale which brings in around $5 million dollars annually to help fund their gallery and local art ventures. Artwork will be brought in to be bid on by artists from throughout the West Coast. A database driven website for the Corvallis art gallery would solve the problem of keeping track of all the bids, customers who enter, buy, and bid on artwork, as well as keep track of all artwork and details about the art. This website will help to support and record the average of 3,000 bids by over 250 unique customers on 100 pieces of vintage and modern artworks.

CONSIDERATIONS

- What if an artist is also a customer? Is it worth consolidating people into a “patrons” table where they can be referenced as an artist?
- Transactions could also refer to intake of a piece of a piece into the inventory as well as a sale (may be beyond the scope of this project)
- How are multiple pieces of work included in a transaction?
    - Maybe have separate transaction and invoice entities? That way an invoice could refer to a transaction that contains multiple pieces of artwork (I think this would be an intersection table of sorts).

Database Outline:

Entities

- Patrons: records the patrons who buy artwork at the gallery
    - Patron_id: int, NOT NULL, PK
    - first_name: varchar(45), NOT NULL
    - Last_name: varchar(45), NOT NULL
    - Address: varchar(45), NOT NULL
    - Email: varchar(45)
    - Is_artist: boolean
    - Relationships: 1:M is implemented between patrons and artwork to keep track of the art a patron owns, 1 person can buy many pieces of art. 1:M between patrons and transactions is implemented with patron_id as an FK to keep track of a patrons transactions.

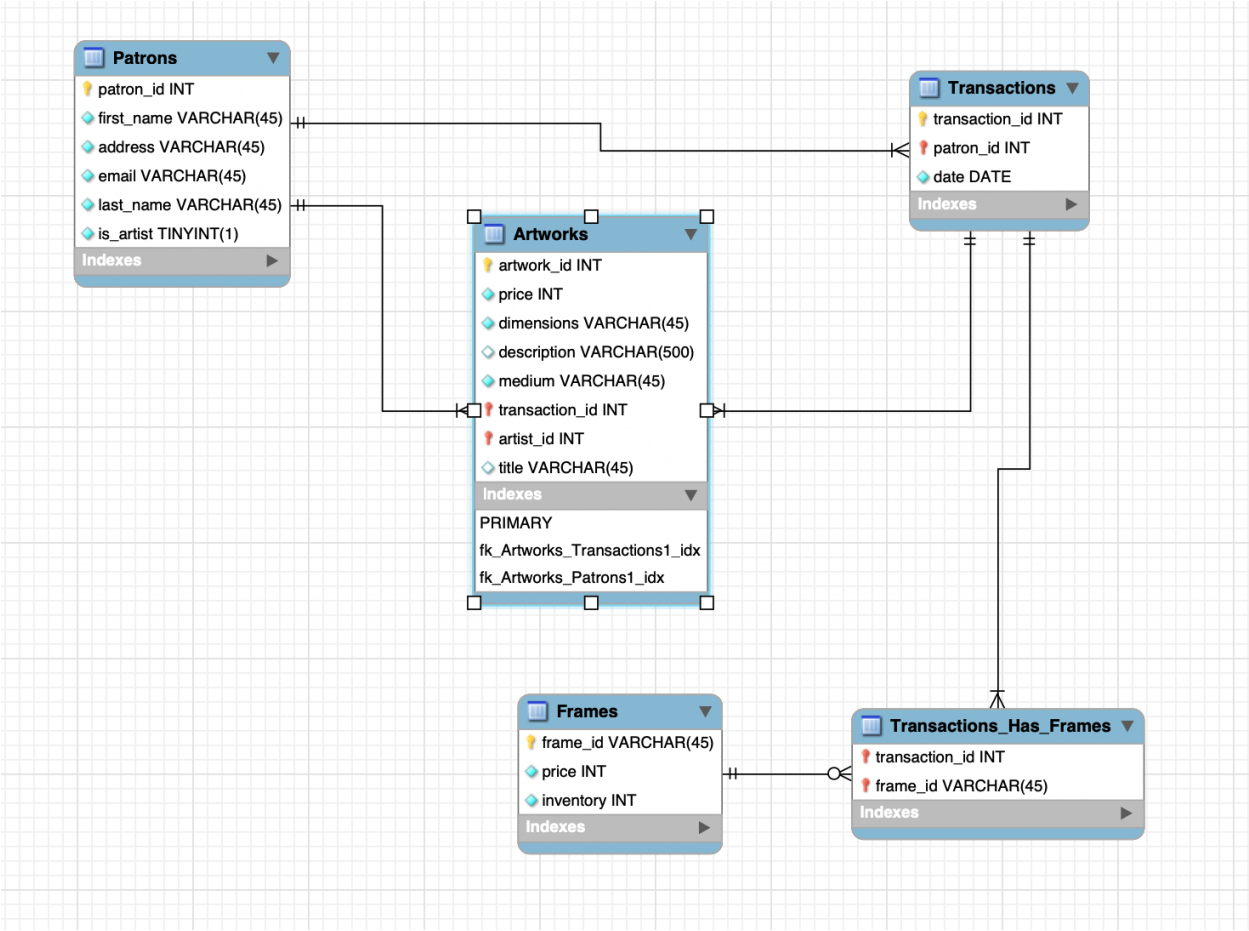
- Transactions: records transactions that occur in the gallery
    - Transaction_id: int, NOT NULL, PK
    - Patron_id: int, NOT NULL, FK
    - Date: datetime, NOT NULL

    - Relationships: There is a 1:M relationship between this and artwork implemented with transaction_id in artwork as an FK because a transaction can have many pieces of art listed on it. There is also a 1:M between patrons and transactions is implemented with patron_id as an FK to keep track of a patrons transactions. Also a M:M with frames implemented with a transition table with FKs transaction_id and frame_id because a frame can be a part of many transactions but a transaction can have many pieces of art which have many frames.

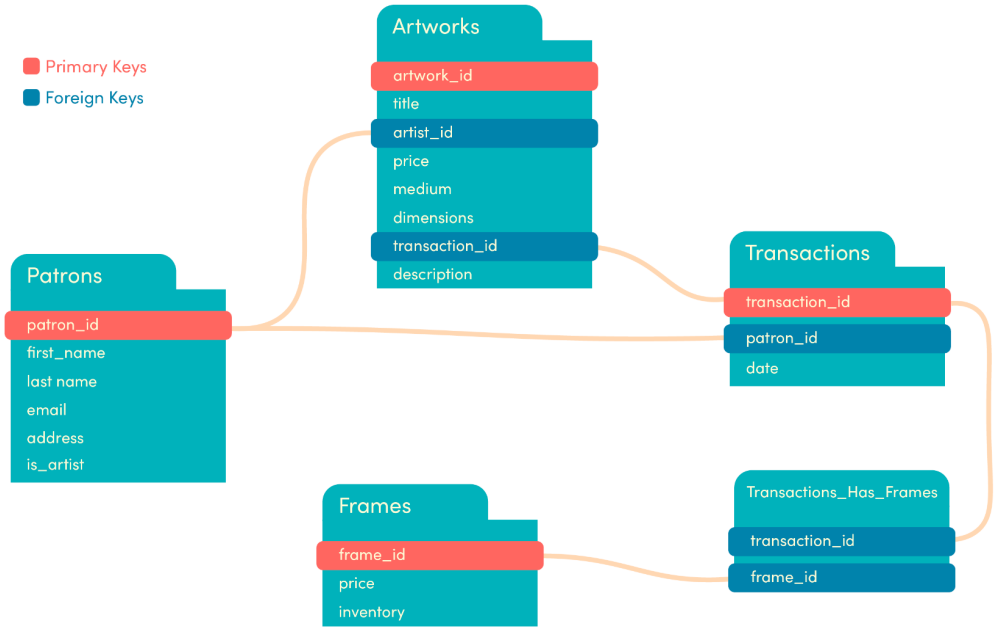
- Artwork: records data about each piece of artwork being sold.
    - Artwork_id: int, NOT NULL, PK
    - Artist_id: int, NOT NULL, FK
    - Price: int, NOT NULL
    - Dimensions: varchar(45), NOT NULL
    - Description: varchar(45)
    - Medium: varchar(45), NOT NULL
    - Transaction_id: int, NOT NULL, FK
    - Relationships: There is a 1:M relationship between this and transactions implemented with transaction_id as FK. Also a 1:M is implemented between patrons and artwork to keep track of the art a patron owns.
- Frames: records data on the frame of the artwork
    - Frame_id: varchar(45), PK
    - Price: int
    - Inventory: int
    - Relationships: M:M with transactions implemented with a transition table with FKs transaction_id and frame_id

- Transactions_has_Frames: transition table to create M:M between transactions and frames
    - Transaction_id: int, NOT NULL, FK
    - Frame_id: varchar(45), NOT NULL, FK
    - Relationships: M:1 with frames implemented with frame_id as a FK. M:1 with transactions implemented with transactions_id as an FK.

ERD Diagram:


Patrons
patron_id INT
first_name VARCHAR(45)
address VARCHAR(45)
email VARCHAR(45)
last_name VARCHAR(45)
is_artist TINYINT(1)
Indexes
Transactions
transaction_id INT
patron_id INT
date DATE
Indexes
Artworks
artwork_id INT
price INT
dimensions VARCHAR(45)
description VARCHAR(500)
medium VARCHAR(45)
transaction_id INT
artist_id INT
title VARCHAR(45)
Indexes
PRIMARY
fk_Artworks_Transactions1_idx
fk_Artworks_Patrons1_idx
Frames
frame_id VARCHAR(45)
price INT
inventory INT
Indexes
Transactions_Has_Frames
transaction_id INT
frame_id VARCHAR(45)
Indexes

Schema



Updated schema based on feedback from the reviewers.

Normalization: Our database is currently in 2NF likely into partially 3NF. It is 2NF because each table represents a single entity or idea which fulfills 1NF and the database is in 1NF and each table except the transition table which I believe is excluded, has its own primary key which fulfills 2NF. I think it's almost to 3NF but we may have a few non-key attributes that determine another non-key attribute in the table.

Example Data:

```
MariaDB [cs340_royal]> SELECT * FROM Artworks;
+------------+-------------+-----------+-------+--------------+------------+----------------+-----------------------------------+
| artwork_id | title       | artist_id | price | medium       | dimensions | transaction_id | description                       |
+------------+-------------+-----------+-------+--------------+------------+----------------+-----------------------------------+
|          1 | Fishnets    |         3 |  5000 | Digital      | 27x40      |              1 | Woman wearing a diving helmate    |
|          2 | Royal Frog  |         6 | 10000 | Acrylic      | 10x10      |              3 | Frog wering a frilled collar      |
|          3 | Otter       |         4 |   300 | Digital      | 8.5x11     |              1 | Low-poly otter floating on its back |
|          4 | Frog Valley |         7 |  4000 | Screen Print | 11x17      |              5 | Two frogs having a picnic         |
+------------+-------------+-----------+-------+--------------+------------+----------------+-----------------------------------+
```

```
MariaDB [cs340_royal]> SELECT * FROM Frames;
+------------+-------+-----------+
| frame_id   | price | inventory |
+------------+-------+-----------+
| 10x10      |    25 |        27 |
| 11x17      |    30 |        14 |
| 22x28      |    40 |        24 |
| 27x40      |    40 |         5 |
| 4x6        |    15 |        30 |
| 8.5x11     |    20 |        16 |
| 8x8        |    20 |        64 |
+------------+-------+-----------+
```

```
MariaDB [cs340_royal]> SELECT * FROM Patrons;
+-----------+------------+-----------+-------------------------------+--------------------+-----------+
| patron_id | first_name | last_name | email                         | address            | is_artist |
+-----------+------------+-----------+-------------------------------+--------------------+-----------+
|         1 | Hunter     | Cram      | huntercooks@goodfood.com      | 156 S Pan St       |         0 |
|         2 | Ryan       | Dillard   | rdillard@garmen.biz           | 561 Takeoff Ave    |         0 |
|         3 | Connor     | Kealey    | creations@connorkealey.design | 101 Art Dr         |         1 |
|         4 | D          | Tilson    | artpdf@tilson.graphics        | 189 Kirkland St    |         1 |
|         5 | Benny      | Beaver    | is@sortacreepy.com            | 147 Monroe Ave     |         0 |
|         6 | Alex       | Koetje    | alex@darling.beads            | 11 Taco St         |         1 |
|         7 | April      | Friend    | april@good.vibes              | 137 Portland Place |         1 |
+-----------+------------+-----------+-------------------------------+--------------------+-----------+
```

```
MariaDB [cs340_royal]> SELECT * FROM Transactions;
+----------------+------------+-------------+
| transaction_id | patron_id | date        |
+----------------+------------+-------------+
|              1 |          1 | 2023-01-02 |
|              2 |          3 | 2022-12-20 |
|              3 |          5 | 2023-02-06 |
+----------------+------------+-------------+
```

```
MariaDB [cs340_royal]> SELECT * FROM Transactions_Has_Frames;
+----------------+----------+
| transaction_id | frame_id |
+----------------+----------+
|              1 | 27x40    |
|              1 | 8.5x11   |
|              2 | 10x10    |
|              3 | 11x17    |
+----------------+----------+
```

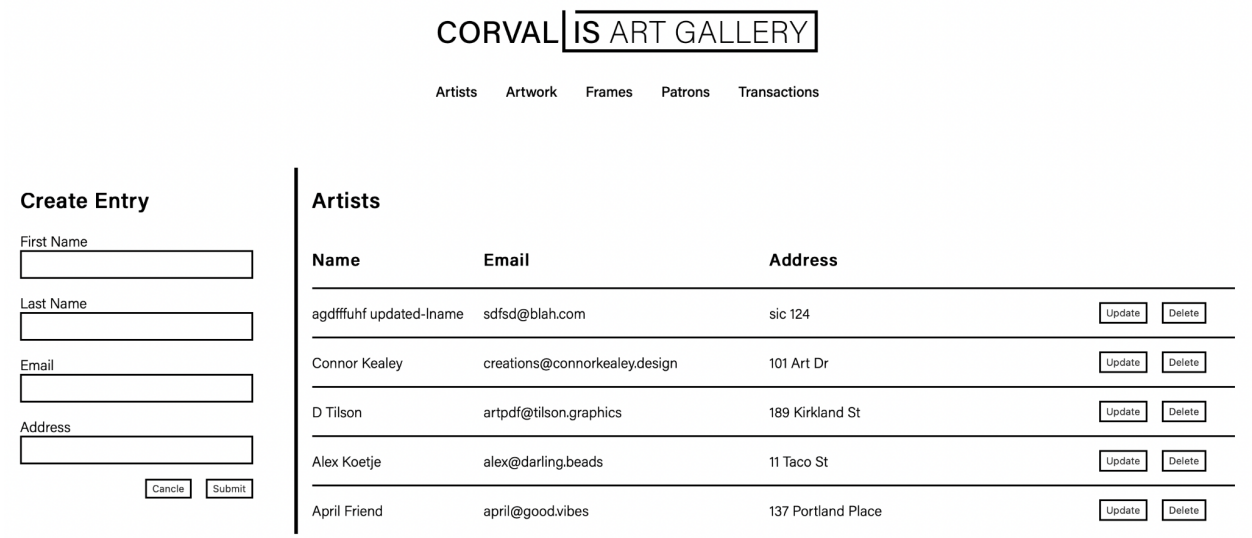
UI Screenshots:

Home Page:

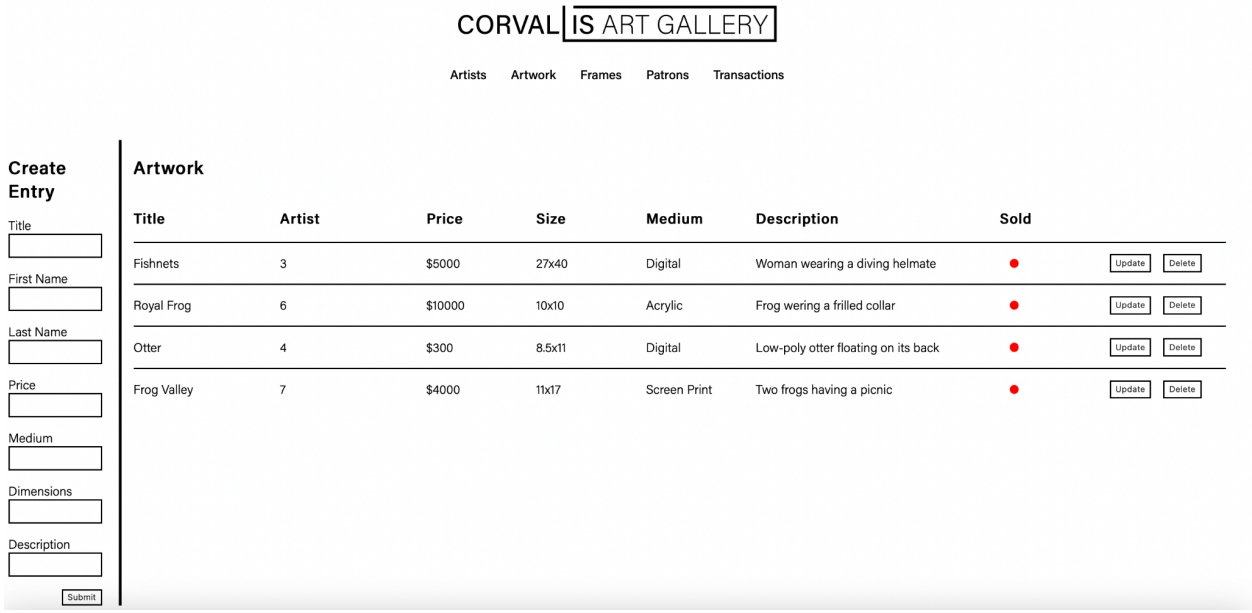
Provides links to all entity pages and a welcome screen to all users.

Welcome to the corvallis art gallery database.

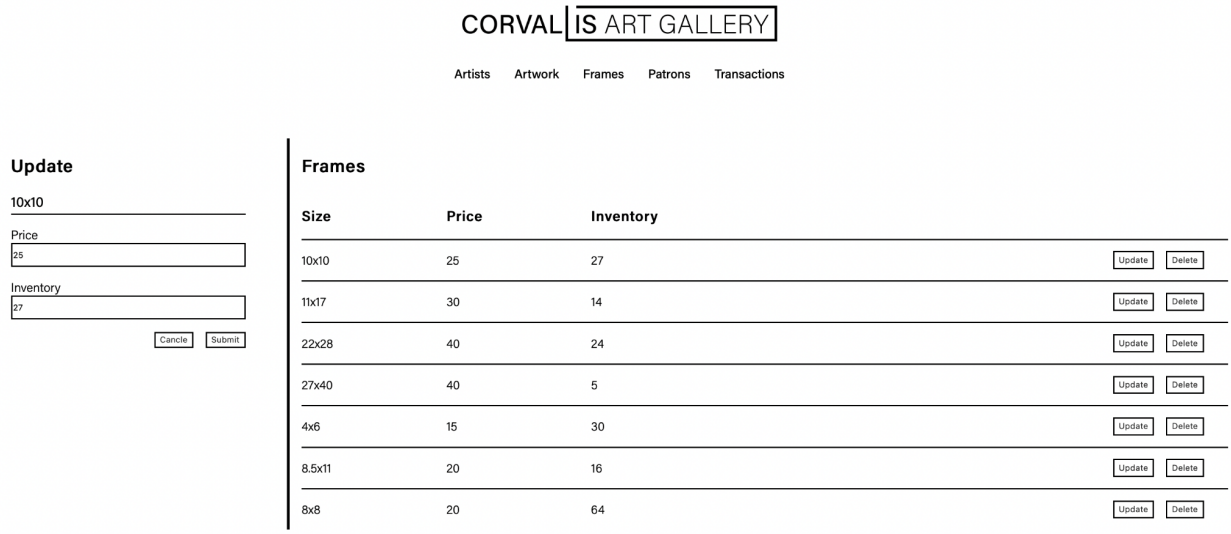
Artists(Create, Read, Update, Delete):
Allows users to read data from the DDL and create new artists to the main database. Clicking delete next to each row will delete that row and item from the database. The update button will auto-populate the "Create Entry" form with the current data for that row and allow the user to change that and click submit to make the changes occur.



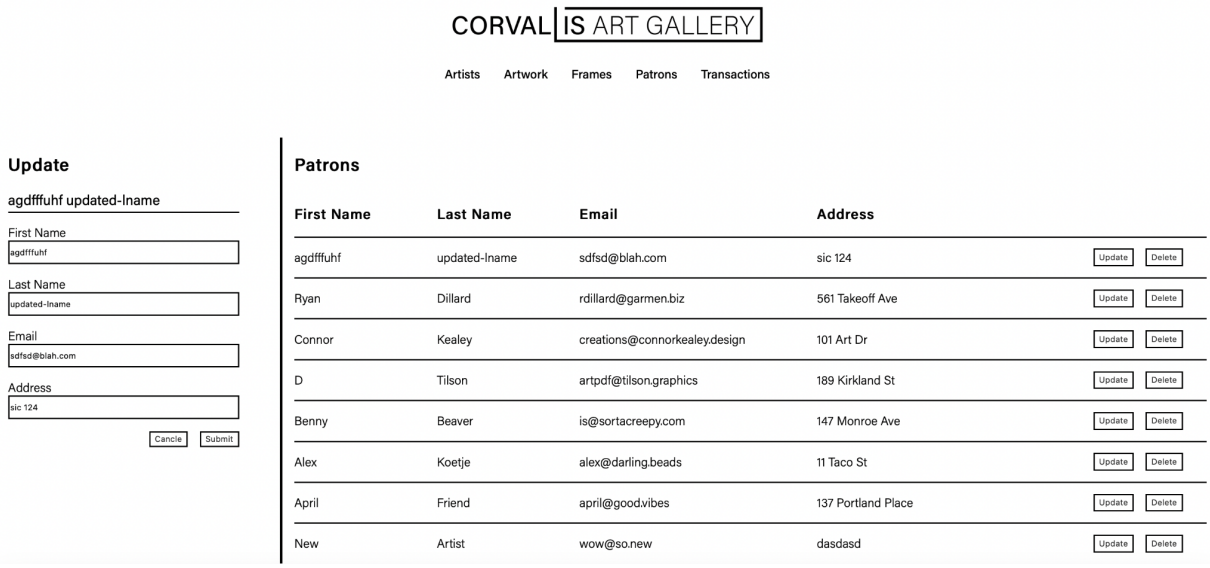
Artworks(Create, Read, Update, Delete):
Allows users to create new artworks, update these artworks, read from the database, and delete artworks from the corvallis art gallery page.



Frames(Create, Read, Update, Delete):
Allows users to create new frames, update these frames, read from the database, and delete artworks from the corvallis art gallery page. Also has links to the other entity pages like the other pages.



Patrons(Create, Read, Update, Delete):
Allows users to create new patrons, update these patrons, read from the database, and delete patrons from the corvallis art gallery page. We have shown the autofilled update form in this example.



Transactions:

This is the page for transactions, unfortunately, we don't have crud implemented for this page.