Project Group 50: Jacob Hathaway and Kevin Kraatz
Flask App (be sure to use the VPN):
http://flip1.engr.oregonstate.edu:53266

Executive Summary:

Over the course of this project we have gained many valuable insights through the development of the Local Library Database Management Application. During the project creation phase, we spent a lot of time focusing on the best way to organize the data into manageable entities and their relationships. Overall, the data organization remained fairly consistent throughout the project. We made one name change based on user feedback from “artist” to “creator” to make the terminology more generic and more applicable to all media types. Originally we had chosen to have a primary and alternate supplier for each media item. However, based on peer feedback, we thought that this information was unnecessary and so we restricted it to just an optional, primary supplier. We also had two additional tables in the beginning for media type (book, music, movie, magazine) and for transaction type (checkin, checkout) and based on grader feedback, we removed those tables and simply made them attributes since they weren’t likely to change and didn’t require a table. This decreased our design complexity and simplified the implementation.

During the development of the SQL files, we also learned a great deal about the best way to write the queries. When we created the sample insert data, we wrote separate insert statements. This made the file a lot longer and unnecessarily complicated. After a suggestion from a peer reviewer, we were able to simplify the queries by chaining the insert statements together so that only one insert statement was necessary per table. We also made a change to allow item replacement cost and patron fine amount to be null. These changes allow the web application user to simply leave those fields blank when they do not apply or the information is not known at that time.

Finally, during the development of the UI, we made quite a few changes throughout the course of the project based on feedback from users and their experiences. At first we implemented some of the CRUD operations that were dependent on foreign keys as text fields where the user would have to specify the correct foreign key. This was not practical since it could be easy to type incorrectly. To solve this problem, we implemented combination boxes with searchable and scrollable dropdown menus. This allows a user to search for and select the right option right in the field. We also implemented tooltips to enhance user experience and to describe what should go in each field, formatting for phone numbers, and dynamic messages that say what edit or delete would take place if a button is clicked. We felt that these UX features enhance the usability of the application. Another design element that we changed early on was the ordering of the information on the webpages. Originally we had the search results table at the top of the page, but we quickly realized that if the dataset contained a lot of entries, users would have to scroll all the way to the bottom to add a new entry.

Overall, the project was a great success thanks to the help and feedback from instructors and peer reviewers and it greatly improved the final quality of our product.

Project Outline:

A local library that serves a small community with a population of 10,000 residents needs a way to track and manage its business data. The database will contain entity tables for at most 5 *Employees*, a few local *Suppliers*, 2,000 *Patrons*, a few different *Media_Types* with descriptions such as music, book, magazine, and movie, and a table with all of the 20,000 *Media_Items* that the library owns and can check out. There will also be a table that records checkout and checkin transactions called *Media_Transactions*. The web application will be used to manage the different entity tables. For example, if new *Employees* are hired or leave, if new *Patrons* accounts are created or closed, new *Suppliers* are added or current suppliers are removed, or as various *Media_Items* are removed from circulation or new ones are added. Transactions at this local library are only accomplished by employees since there is no self-check option. Each transaction into the database will be per item. In order to maintain business data and provide support for auditing, our database will restrict deletion of patrons, employees, and media items that have been involved in media transactions. We will also restrict deletion of a creator when we have media items by that creator in our database. This will also allow us to use friendly names when searching and working with data.

Database Outline:

1. Employees
    - Attributes including data types/constraints
        - employee_id
            - INT
            - Unique
            - Auto-Increment
            - Primary Key
            - Not Null
        - first_name
            - NVARCHAR(50)
            - Not Null
        - last_name
            - NVARCHAR(50)
            - Not Null
        - phone_number
            - NVARCHAR(15)
            - NULL DEFAULT NULL
        - address
            - NVARCHAR(200)
            - Not Null
    - Relationships
        - 1:M with Media_Transactions
        - One employee will check out items to a patron or check items in for a patron.

2. Patrons
    - Attributes including data types/constraints
        - patron_id
            - INT
            - Unique
            - Auto-Increment
            - Primary Key
            - Not Null
        - first_name
            - NVARCHAR(50)
            - Not Null
        - last_name
            - NVARCHAR(50)
            - Not Null
        - phone_number
            - NVARCHAR(15)
            - Not Null
        - fine_amount
            - Decimal(19,2)
            - NULL DEFAULT NULL
        - address
            - NVARCHAR(200)
            - Not Null
    - Relationships
        - 1:M with Media_Transactions
        - M:M with Media_Items (Media_Transactions is the intersection table)
        - One patron may check items in or out from a library employee

3. Media_Items
    - Attributes including data types/constraints
        - item_id
            - INT
            - Unique
            - Auto-Increment
            - Primary Key
            - Not Null
        - media_type
            - NVARCHAR(20)
            - Not Null
        - supplier_id
            - INT
            - Null (A particular item may not have a supplier at times)
            - Foreign key with Suppliers.supplier_id
        - title

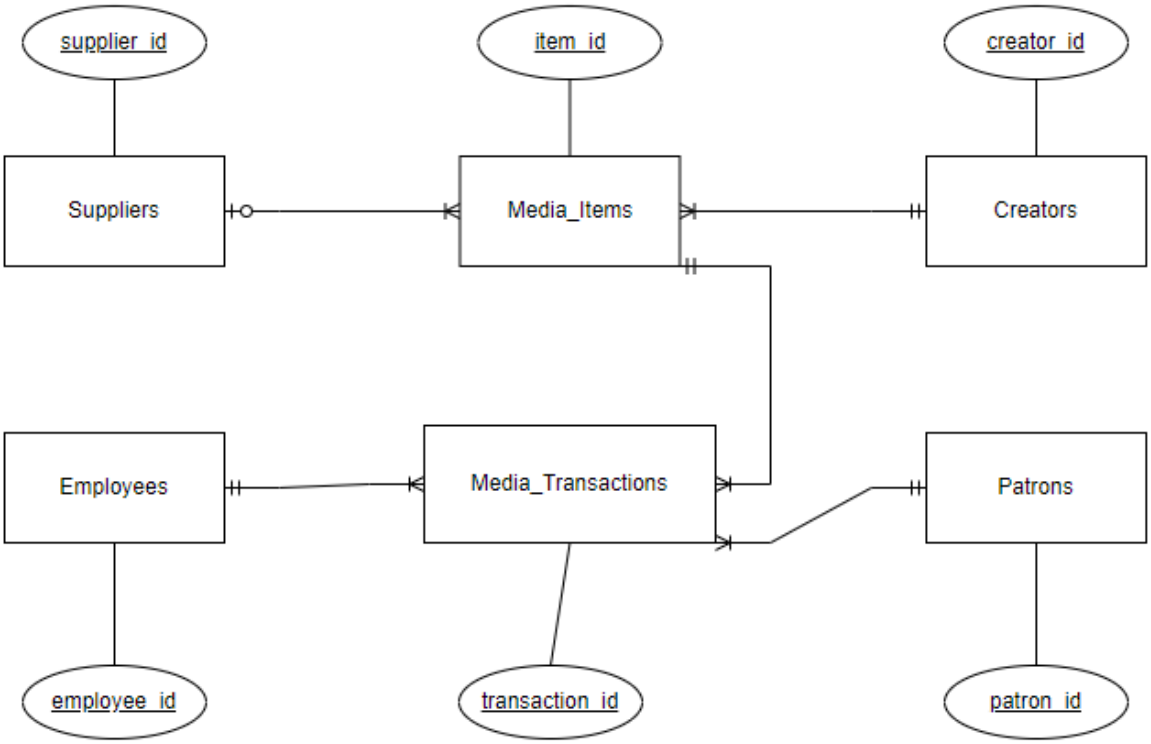
            - NVARCHAR(50)
            - Not Null
        - creator_id
            - INT
            - Not Null
            - Foreign key with Creators.creator_id
        - release_date
            - DATE
            - Not Null
        - replacement_cost
            - Decimal(19,2)
            - Null Default Null
        - quantity_owned
            - INT
            - Not Null
    - Relationships
        - 1:M with Media_Transactions
            - Each media item may be checked out many times
        - 1:M with Suppliers
        - 1:M with Creators

4. Suppliers
    - Attributes including data types/constraints
        - supplier_id
            - INT
            - Not Null
            - Primary Key
            - Unique
            - Auto-Increment
        - name
            - NVARCHAR(50)
            - Not Null
        - address
            - NVARCHAR(200)
            - Not Null
        - phone_number
            - NVARCHAR(20)
            - Not Null
    - Relationships
        - 1:M with Media_Items
            - 1 supplier may supply many different media items
5. Media_Transactions
    - Attributes including data types/constraints
        - transaction_id

                - INT
                - Not Null
                - Primary Key
                - Unique
                - Auto-Increment
            - patron_id
                - INT
                - Not Null
                - Foreign key with Patrons.patron_id
            - transaction_type
                - NVARCHAR(20)
                - Not Null
            - employee_id
                - INT
                - Not Null
                - Foreign key with Employees.employee_id
            - item_id
                - INT
                - Not Null
                - Foreign key with Media_Items.item_id
            - transaction_date
                - DATETIME
                - Not Null
    - Relationships
        - 1:M with Patrons
        - 1:M with Employees
        - 1:M with Media_Items

6. Creators
    - Attributes including data types/constraints
        - creator_id
            - INT
            - Unique
            - Not Null
            - Auto-Increment
            - Primary Key
        - full_name
            - NVARCHAR(50)
            - Not Null
    - Relationships
        - 1:M with Media_Items

Entity Relationship Diagram



Database Schema:

Sample Data:

Suppliers

supplier_id	name	address	phone_number
1	Ingram Content Group, Inc.	One Ingram Blvd. La Vergne, TN 37086	(855) 867-1920
2	Publishers Group West	1700 Fourth Street Berkeley, CA 94710	(510) 809-3700
3	Media Solutions	9632 Madison Blvd. Madison, AL 35758	(800) 476-5872
4	Small Changes Inc.	1418 NW 53rd Street Seattle, WA 98107	(206) 382-1980

Employees

employee_id	first_name	last_name	phone_number	address
1	Jaimie	Melliard	(702) 445-4215	6 Eliot Point Las Vegas, NV 89115
2	Darian	Labern	(415) 391-7377	08854 Armistice Junction San Francisco, CA 94110
3	Cathee	Pluthero	(407) 216-1318	97 Del Mar Circle Pensacola, FL 32595
4	Lena	Bausor	(773) 616-9766	3 Nelson Pass Chicago, IL 60657

Creators

creator_id	full_name
1	Stephen King
2	Joanna Gaines
3	George R.R. Martin
4	Katy Perry
5	Quentin Tarantino

Patrons

patron_id	first_name	last_name	phone_number	fine_amount	address
1	Dianna	MacTrustie	(979) 856-0499	0.00	2568 Mifflin Parkway Bryan, TX 77806
2	Lorain	Burren	(916) 468-6616	2.75	45 Melrose Way Sacramento, CA 94250
3	Mildrid	McArte	(212) 760-3859	19.95	1 Algoma Hill New York City, NY 10184
4	Jeramy	McCann	(503) 550-4845	0.00	68967 Stuart Alley Beaverton, OR 97075

Media_Items

item_id	media_type	supplier_id	title	creator_id	release_date	replacement_cost	quantity_owned
1	Book	1	The Shining	1	1977-01-28	7.99	2
2	Book	2	A Game of Thrones	3	1996-08-01	15.29	4
3	Movie	3	Pulp Fiction	5	1994-10-14	15.50	1
4	Music	3	Teenage Dream	4	2010-08-24	8.01	1
5	Magazine	4	Magnolia Journal	2	2022-01-01	20.00	1

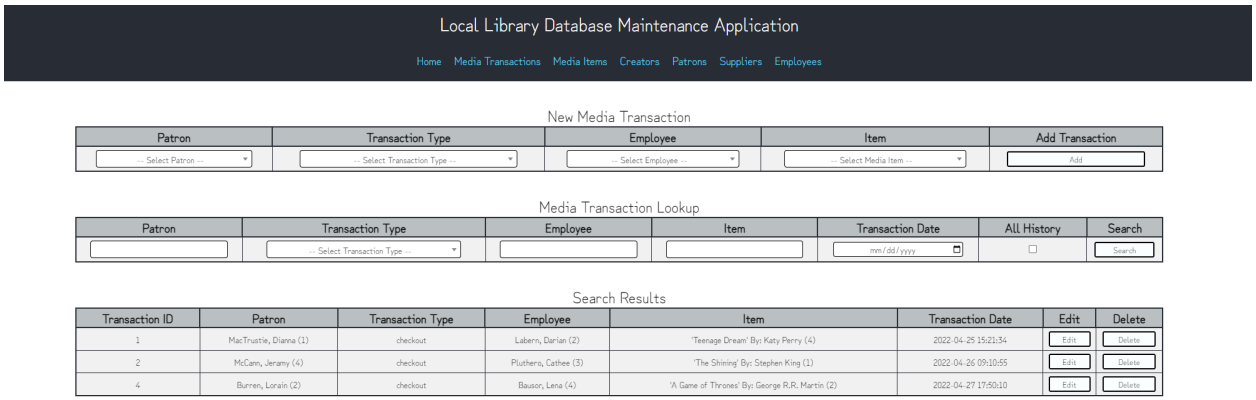
Media_Transactions

transaction_id	patron_id	transaction_type	employee_id	item_id	transaction_date
1	1	checkout	2	4	2022-04-25 15:21:34
2	4	checkout	3	1	2022-04-26 09:10:55
3	3	checkin	1	5	2022-04-26 11:46:34
4	2	checkout	4	2	2022-04-27 17:50:10

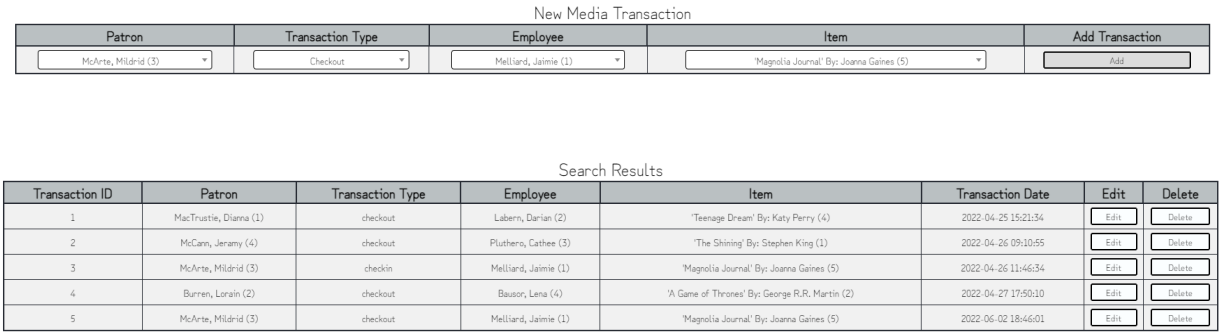
Screen Captures:

Local Library Database Maintenance Application Home Page.

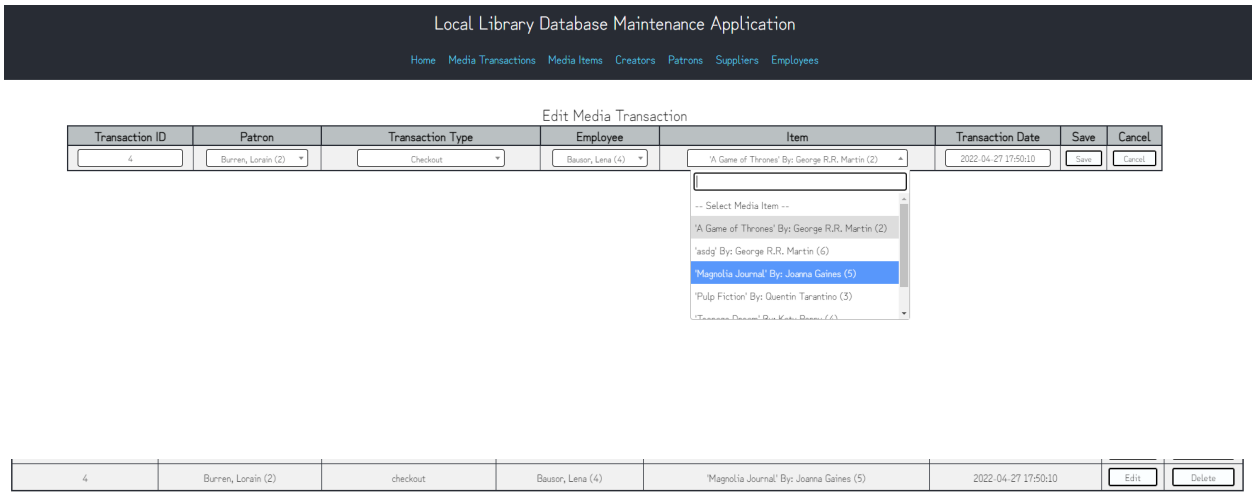
Media Transactions Page with Create, Retrieve, Update, and Delete functionality. Search results are displayed below, and each entry has an Edit and Delete button.



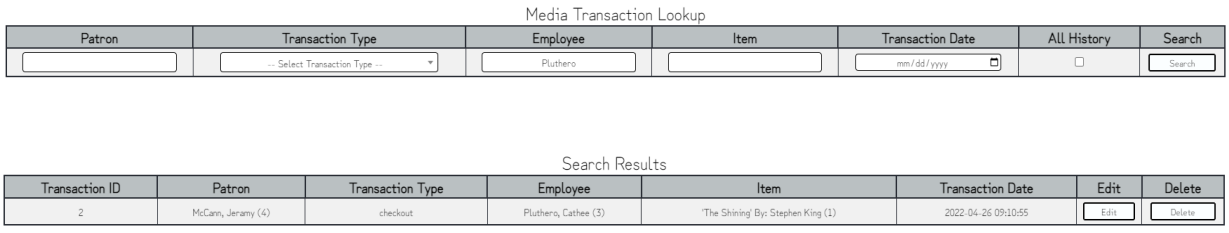
Adding a new entry.



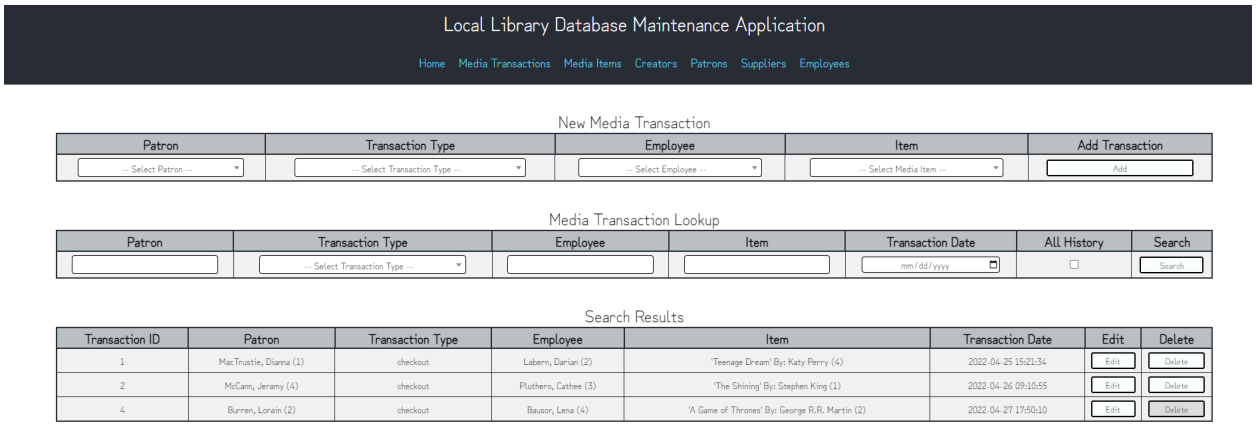
Media Transaction Edit Page for editing a transaction entry. Foreign keys are selected and edited via ComboBox as shown.

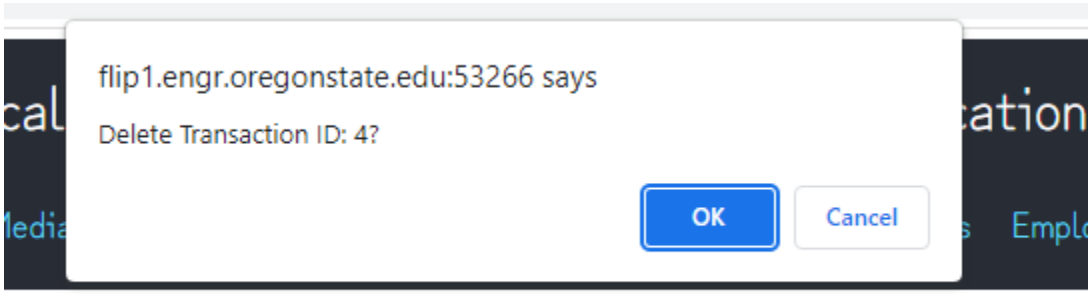


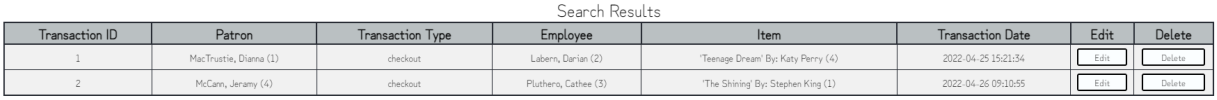
Entries can be searched by attribute.



Clicking the Delete button for an entry will delete that transaction from the database.

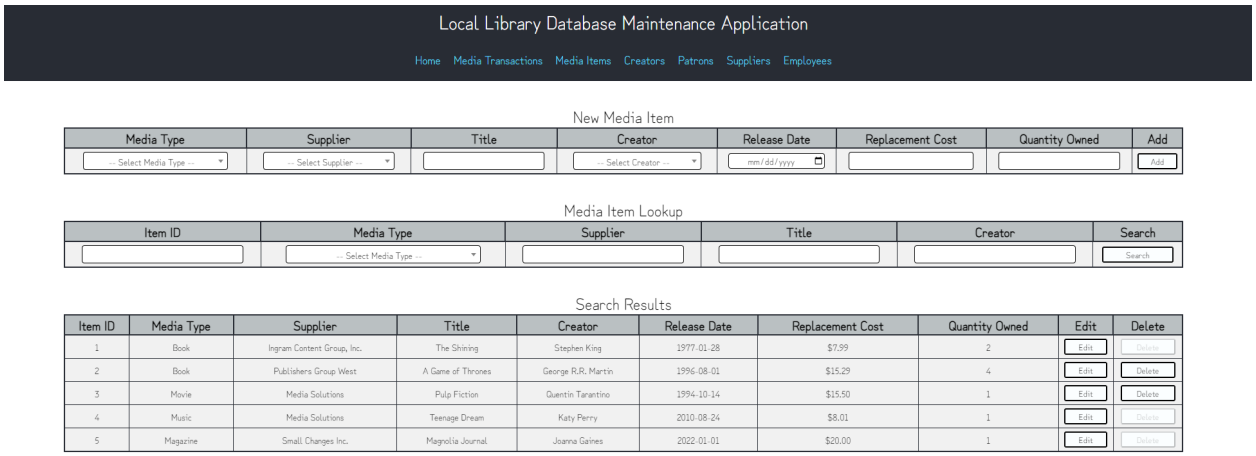




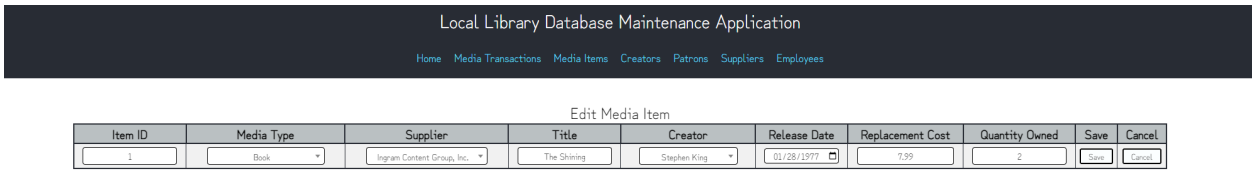


Functionality is the same for each page in the application.

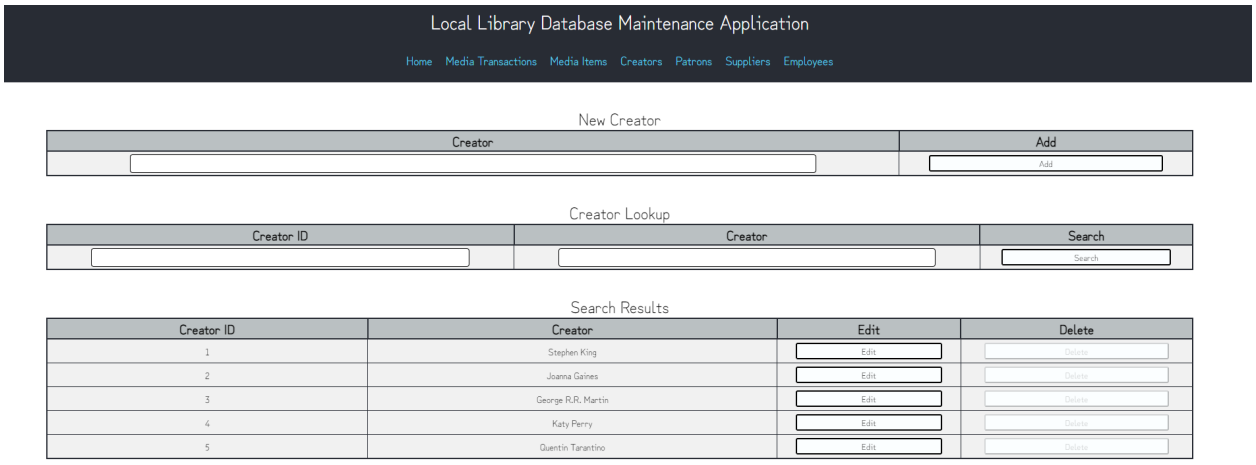
Media Items Page with Create, Retrieve, Update, and Delete functionality.



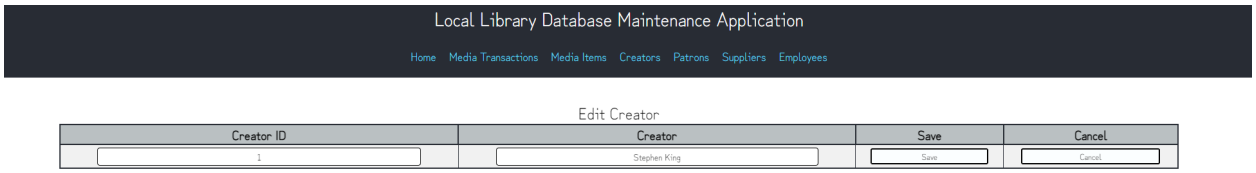
Edit Media Items Page.



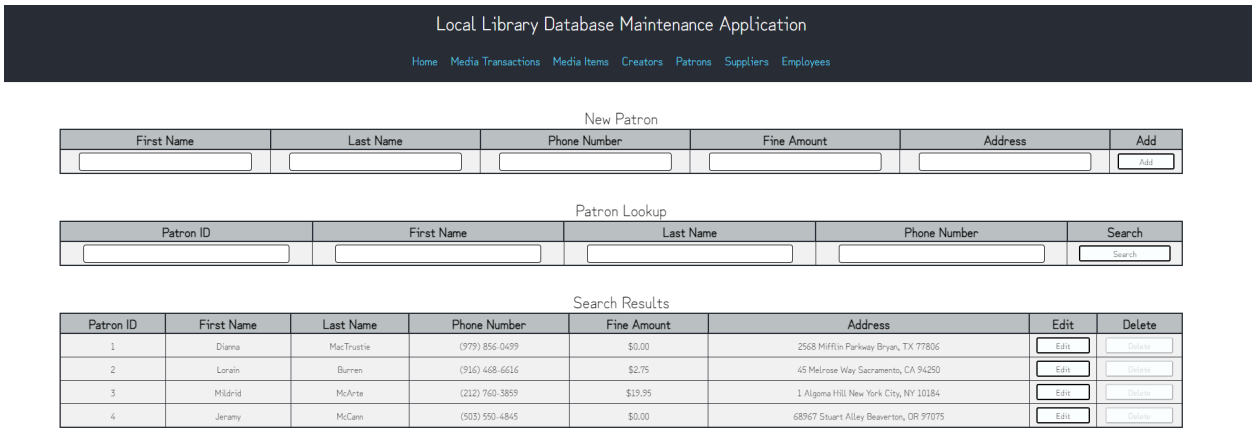
Creators Page with Create, Retrieve, Update, and Delete functionality.



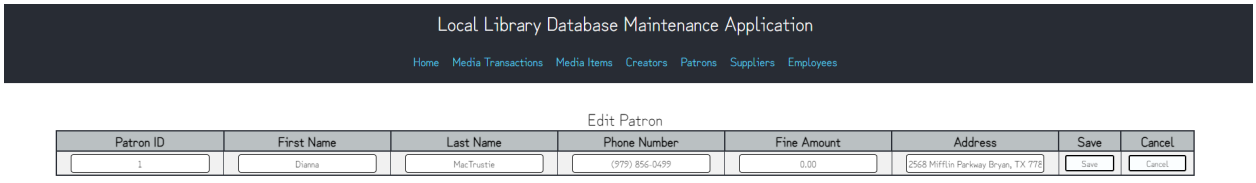
Edit Creators Page.



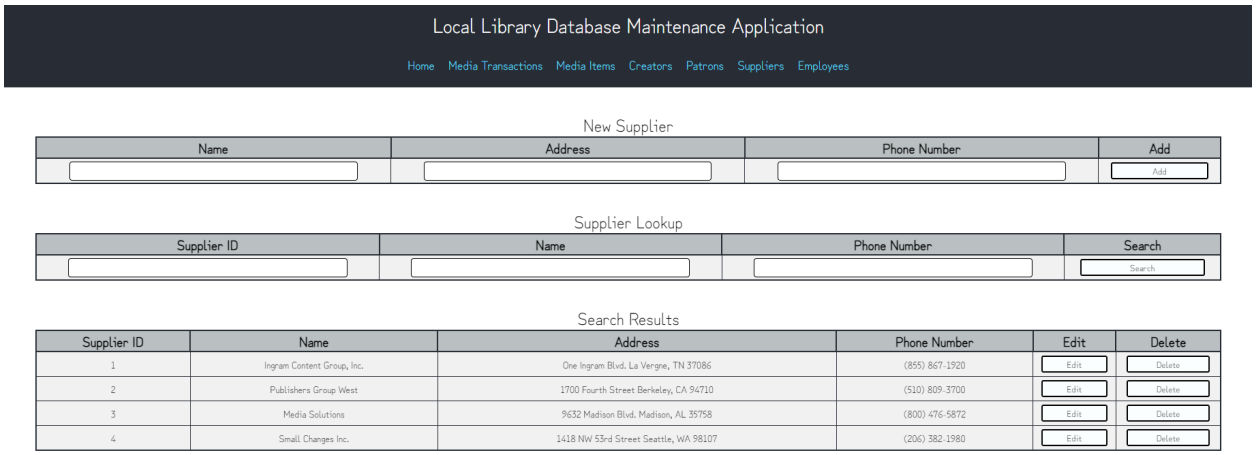
Patrons Page with Create, Retrieve, Update, and Delete functionality.



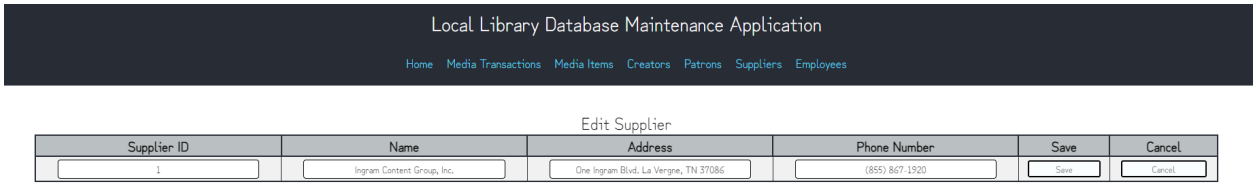
Edit Patrons Page.



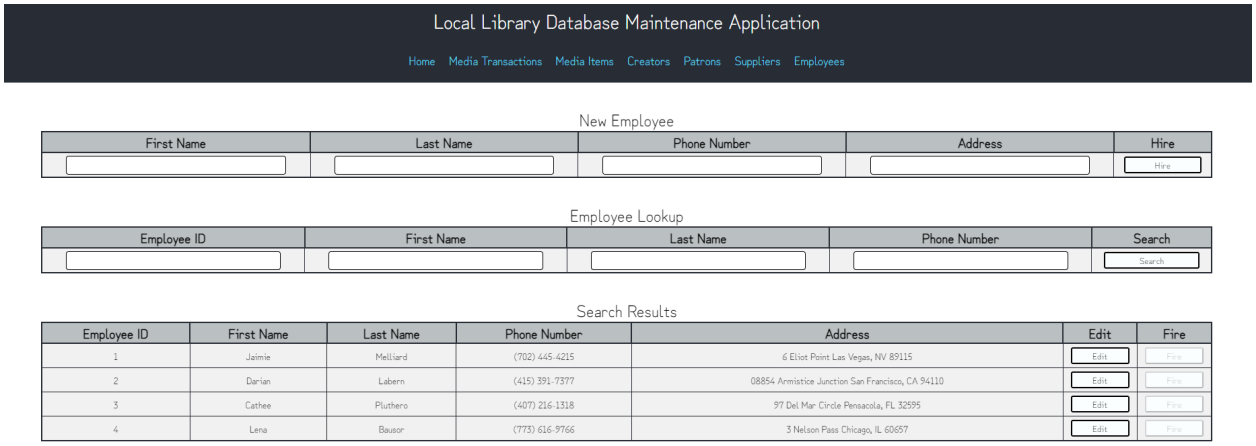
Suppliers Page with Create, Retrieve, Update, and Delete functionality.



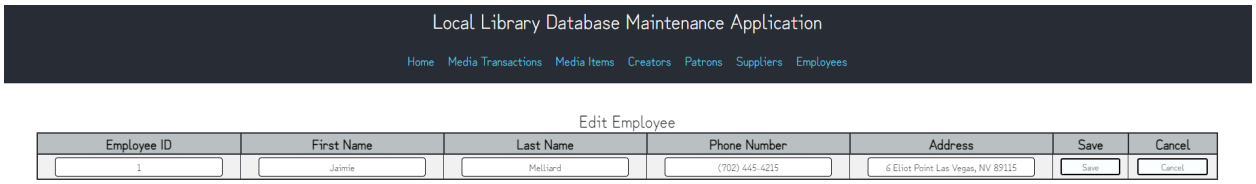
Suppliers Edit Page.



Employees Page with Create, Retrieve, Update, and Delete functionality.



Edit Employees Page.



Edit Employee

Employee ID	First Name	Last Name	Phone Number	Address	Save	Cancel
1	Jaimie	Melliard	(702) 445-4215	6 Eliot Point Las Vegas, NV 89115	Save	Cancel