1. Introduction

This document serves as the final demonstration and remote access guide for our Library Management System (LMS) database. The purpose of this guide is to showcase the interactive capabilities of our deployed system, explain its architecture, and provide remote instructions for the TAs to independently validate all core functionalities.

Our solution provides an intuitive web-based interface allowing SQL queries to be executed live against our Supabase PostgreSQL database. Built using modern frontend and backend tools, the application supports query testing, result rendering, and edge-case handling.

2. How to Use the Live Query Demo

Live App URL:

https://eecs-447-team-27.vercel.app

2.1. Instructions for Use

1. Navigate to the application link above.
2. You will see a textbox labeled **"Type a query..."**.
3. Type in any SQL statement in the textbox. Press `Enter` or click on **Execute** button to run a single-line query or `Shift+Enter` for multiline queries.
4. The output will render below the textbox in a responsive table.

2.2. Example Queries

2.2.1. Get all active users

```
SELECT * FROM Users WHERE Account_Status = 'Active';
```

2.2.2. List all currently borrowed items

```
SELECT t.Transaction_ID, u.Name, l.Title, t.Borrow_Date,
t.Due_Date
FROM Transactions t
JOIN Users u ON t.User_ID = u.user_id
JOIN LibraryItems l ON t.Item_ID = l.Item_ID
WHERE t.Return_Date IS NULL;
```

2.2.3. All reservations by a specific user

```
SELECT l.Title, r.Reservation_Date, r.Expiration_Date
FROM Reservations r
JOIN LibraryItems l ON r.Item_ID = l.Item_ID
WHERE r.User_ID = 99;
```

2.2.4. Get all overdue items

```
SELECT t.Transaction_ID, u.Name, l.Title, t.Due_Date
FROM Transactions t
JOIN Users u ON t.User_ID = u.user_id
JOIN LibraryItems l ON t.Item_ID = l.Item_ID
WHERE t.Return_Date IS NULL AND t.Due_Date < CURRENT_DATE;
```

2.2.5. List all unpaid fees

```
SELECT f.Fee_ID, u.Name, f.Amount
FROM Fees f
JOIN Users u ON f.User_ID = u.user_id
WHERE f.Paid_Status = 'Unpaid';
```

2.2.6. Fetch all reviews for a specific item

```
SELECT u.Name, r.Rating, r.Review_Text, r.Review_Date
FROM Reviews r
JOIN Users u ON r.User_ID = u.user_id
WHERE r.Item_ID = 87;
```

2.2.7. Top 3 users by total unpaid fees

```
SELECT u.Name, SUM(f.Amount) AS TotalDue
FROM Users u
JOIN Fees f ON u.user_id = f.User_ID
WHERE f.Paid_Status = 'Unpaid'
GROUP BY u.Name
ORDER BY TotalDue DESC
LIMIT 3;
```

2.2.8. Borrow-return cycle analysis per user

```
SELECT u.Name, COUNT(t.Transaction_ID) AS TotalBorrowed,
       SUM(CASE WHEN t.Return_Date IS NOT NULL THEN 1 ELSE 0 END)
AS Returned,
        SUM(CASE WHEN t.Return_Date IS NULL THEN 1 ELSE 0 END) AS
Outstanding
FROM Users u
LEFT JOIN Transactions t ON u.user_id = t.User_ID
GROUP BY u.Name
ORDER BY TotalBorrowed DESC;
```

2.2.9. Books with most late returns

```
SELECT l.Title, COUNT(*) AS LateReturns
FROM Transactions t
JOIN LibraryItems l ON t.Item_ID = l.Item_ID
WHERE t.Return_Date IS NOT NULL AND t.Return_Date > t.Due_Date
GROUP BY l.Title
ORDER BY LateReturns DESC
LIMIT 5;
```

3. System Architecture

Our LMS system was designed with a full-stack cloud-first approach for fast development and easy accessibility:

3.1. Tech Stack

3.1.1. Frontend

Next.js (React), TailwindCSS, ShadCN UI, Framer Motion

3.1.2. Backend

Supabase RPC (Stored Procedures), REST APIs

3.1.3. Database

Supabase PostgreSQL

3.1.4. Deployment

Vecel (Frontend), Supabase (DB + Backend)

3.2. Architecture Highlights

- Supabase PostgreSQL hosts the physical database, allowing for role-based access control, CSV data imports, and stored procedure execution.
- `custom_sql_runner` RPC function inside Supabase securely executes user-submitted queries.
- Frontend interfaces with Supabase via this stored function to prevent direct SQL injection risks.
- The UI is styled with calm pastel colors, responsive layout, and dynamic animations using Framer Motion.

4. Remote TA Access Guide

4.1. Supabase Credentials

Access has been secured via the `custom_sql_runner` function. No login is required from the TA.

4.2. Testing Steps

1. Visit https://eecs-447-team-27.vercel.app.
2. Run any of the queries listed in Section 2.2 or test your own.
3. Results will be fetched and shown in the table below the textbox.

   Note: Direct Supabase login is not required. All functionality has been exposed via our frontend with controlled RPC access.