Cps 301 Intro to Database Management Systems

Application Development II

Date: 9th December 2025
Team Members: John Tam & David Geddam

Features Implemented

1. **Main Home Page (Service List)**

- When we open the app, it pulls all the existing services from the database and displays them in a table format. It shows date, theme and the respective links.

2. **Service Details Page**

- Clicking a service takes you to the respective information about that service, i.e; songleader, theme. We display a table of every event in that service sorted by their sequence number.
- Create Service Form: Below the table, we added a form. It defaults to the current time and date, it has a dropdown menu for Songleader. We queried the database to find those who are songleaders and listed them by last name.

3. **Service (Stored Procedure)**

- When you fill out the details, and hit "Create Service", the python app calls our MySQL stored procedure that we created.
- It checks if a service exists at that time.
- Then, it creates the new service record.
- Then, it looks up the songleader's ID according to the name selected and links them to the service.
- Then, it copies all the events from the old service to the new one, keeping the structure and notes, and setting the specific songs and people to NULL.

Work Documentation

- **David:** Worked on flask setup, main page routing, and debugging database connection issues.
- **Hours Invested:** 6 hours
- **John:** Wrote the stored procedure in sql, setting the html templates for the details page, formatted code and tables, and debugging database connection issues.
- **Hours Invested:** 6 hours

Demo Video Link:

https://www.youtube.com/watch?v=z5g9XBvcRYw

SQL Code for Stored Procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `create_service`(
    IN templateID INT,
    IN newDateTime DATETIME,
    IN newTheme VARCHAR(255),
    IN newSongleader VARCHAR(255),
    OUT statusCode INT
)
BEGIN
    DECLARE existingService INT;
    DECLARE songleaderID INT;

    -- check date/time for existing service
    SELECT COUNT(*)
    INTO existingService
    FROM service
    WHERE Svc_DateTime = newDateTime;

    IF existingService > 0 THEN
        SET statusCode = 1;        -- service already exists
    ELSE

        -- insert new service
        INSERT INTO service (Svc_DateTime, Theme_Event)
        VALUES (newDateTime, newTheme);

        SET @newServiceID = LAST_INSERT_ID();


        -- get new songleader name if provided
        IF newSongleader IS NOT NULL AND newSongleader != '' THEN

            SELECT Person_ID
            INTO songleaderID
            FROM person
            WHERE CONCAT(First_Name, ' ', Last_Name) = newSongleader
            LIMIT 1;

            IF songleaderID IS NOT NULL THEN
                INSERT INTO fills_role (Service_ID, Person_ID, Role_Type, Confirmed)
                VALUES (@newServiceID, songleaderID, 'S', 'N');
            END IF;

        END IF;


        -- copy records into the new service
        INSERT INTO service_item
            (Service_ID, Seq_Num, Event_Type_ID, Title, Notes, Confirmed,
             Person_ID, Ensemble_ID, Song_ID)
        SELECT
            @newServiceID,      -- new Service_ID
            Seq_Num,
            Event_Type_ID,
            NULL AS Title,
            Notes,
            'N' AS Confirmed,
            NULL AS Person_ID,
            NULL AS Ensemble_ID,
            NULL AS Song_ID
        FROM service_item
        WHERE Service_ID = templateID;

        SET statusCode = 0;      -- success

    END IF;
END
```

By affixing our signatures below, We certify that the accompanying work represents our own intellectual effort. Furthermore, we have received no outside help.

Signed: David Geddam & John Tam