TEST SCENARIO TS-02: FULL APPLICATION FUNCTIONALITY

Project Name: AniTrack

Module: Full Functionality Testing

Author: Matyáš Prokop

Date: 5.1.2026

Test Objective

To verify comprehensive application features, including CRUD operations for all entities, data validation, M:N relationship management (Anime-Genres), database transactions (Watchlist Transfer), CSV data import, and automated history logging via triggers.

Prerequisites

- System Status: Scenario TS-01 (Installation & Launch) must be successfully completed.
- Application: Flask server must be running at `http://127.0.0.1:5000`.
- Database: Tables must be empty (fresh installation state)

Test Steps

1. User Management (CRUD)

- Create: Navigate to "Users" and add `testuser1` (test1@example.com) and `adminuser` (admin@example.com, Admin: Yes).
- Read/Search: Search by ID "1", name "admin", and email "test1".
- Update: Edit `testuser1` to change the email to `newemail@example.com`.

2. Genre Management

- Create: Add genres: `Action`, `Comedy`, `Drama`, `Romance`, and `Sci-Fi`.
- Update: Rename `Sci-Fi` to `Science Fiction`.

3. Anime Management (M:N Relationships)

- Create: Add `Shingeki no Kyojin` (Genres: Action, Drama).
- Create: Add `Kimetsu no Yaiba` (Genre: Action).
- Create: Add `Spy x Family` (Genres: Action, Comedy).
- Update: Edit `Spy x Family` status to `FINISHED` and add Romance genre.

4. Watchlist & History (Transactions & Triggers)

- Add Entry: Assign `Attack on Titan` to `testuser1` (Status: WATCHING, Progress: 25).
- Transaction: Use the "Transfer" tool to move `Demon Slayer` from `testuser1` to `adminuser`.
- Verify History: Check the "History" tab to confirm automated logs for `INSERT` and `UPDATE` actions with old/new value tracking.

5. CSV Data Import

- Genres: Import `test_genres.csv` (Horror, Fantasy, Thriller).
- Anime: Import `test_anime.csv` (Death Note, Naruto).

6. Deletion & Constraints

- Watchlist: Delete a specific entry and verify it is removed from the view.
- Constraint Test: Attempt to delete the `Action` genre while anime are still assigned to it (Expected: Error).
- Cleanup: Delete the `Horror` genre (Expected: Success).

Success Criteria

ID	Success Criterion	Expected Result	Status
1	**User CRUD**	All user operations (Create, Search, Update) function without errors.	[ ]
2	**Genre CRUD**	Genres are successfully managed and displayed in the list.	[ ]
3	**M:N Anime-Genres**	Anime can be linked to multiple genres and updated correctly.	[ ]
4	**Watchlist Logic**	Watchlist entries accurately reflect user progress and anime status.	[ ]
5	**Triggers/History**	The history table automatically logs all changes (INSERT/UPDATE/DELETE).	[ ]
6	**Transactions**	Watchlist transfers move data between users as a single atomic unit.	[ ]
7	**CSV Import**	External data is correctly parsed and inserted into the database.	[ ]
8	**Summary Report**	The dashboard displays correct aggregated counts and statistics.	[ ]

| 9 | **Data Integrity** | Foreign key constraints prevent the deletion of genres currently in use. | [ ] |

Test Data Summary

Category	Details
Users	2 Users (1 Admin, 1 Standard)
Anime	5 Anime total (3 Manual, 2 CSV Import)
Relationships	M:N Anime-Genres junction table active
Imported Files	test_genres.csv, test_anime.csv
Database Logic	3 Active Triggers for History, 1 Transactional Transfer
Access URL	http://localhost:5000