User Story #1671: Convert SQLite to MySQL

Date: 2025-12-30 **Tool Created**: backend/migrate_sqlite_to_mysql.py **Unit Test**: backend/tests/test_migration_script.py

Migration Plan

Since the production database credentials are typically managed via environment variables and not hardcoded, we have prepared the application for a seamless transition.

1. Dependencies

- Added pymysql to backend/requirements.txt to support the MySQL driver.

2. Migration Script

A custom script has been created to migrate data from the local sql_app.db (SQLite) to any target MySQL database.

Usage:

```
export MYSQL_DATABASE_URL='mysql+pymysql://user:password@host:port/dbname'
python backend/migrate_sqlite_to_mysql.py
```

3. Verification

A unit test (tests/test_migration_script.py) was executed to verify the migration logic without needing an active MySQL server. It mocks the database connections and asserts that: 1. Source (SQLite) is queried. 2. Target (MySQL) connection is established. 3. Records are iterated and added to the target session. 4. Transaction is committed.

Test Results:

```
platform darwin -- Python 3.9.20
rootdir: /Users/trickshot/Desktop/AI_based_tools/Wishing_tool/backend
collected 1 item

tests/test_migration_script.py .                    [100%]

==================== 1 passed in 0.16s ====================
```

Conclusion

The application is now **MySQL-ready**. To complete the “Conversion”: 1. Provision a MySQL database. 2. Run the migration script above. 3. Update the DATABASE_URL environment variable in your deployment environment (e.g., Vercel) to point to the new MySQL instance.