Bug Fix Verification Report

Bug Verification Report - Bug #1822: Data Integrity & Backend Validation

Date: 2026-01-09

Bug ID: #1822

Title: Wish is created successfully without selecting mandatory Wish Details

Status: Fixed

Issue Description

Users were able to create/schedule wishes even when mandatory fields (Event Name, Date, Type) were omitted. While frontend validation was partially addressed in #1817, the backend was still accepting incomplete requests, leading to invalid data in the database. Additionally, some parts of the UI still used native browser alerts for feedback.

Epic: #1599 Occasion & Event Management

User Story: #1651 Create birthday or anniversary event

Fix Implementation

A two-pronged approach was taken to ensure data integrity and a premium UI experience.

1. Backend Enforcement (Endpoints & Models)

- **Strict Pydantic Validation**: Updated `ScheduleRequest` in `endpoints.py`. Fields like `recipient_name`, `event_name`, and `scheduled_time` are now strictly mandatory at the schema level.

- **Custom Validators**: Added `@validator` decorators to the Pydantic model to ensure strings are not just present, but also not empty or whitespace-only.

- **Explicit Safeguards**: Added explicit `400 Bad Request` exceptions in the `schedule_wish` endpoint if mandatory fields are missing, providing a last line of defense against corrupted data.

2. Frontend Safeguards & UI Polish

- **Removed Default Values**: Changed the initial state of `CreateWishWizard.js` to use empty strings for `event_name` and `event_type`. This forces the user to interact with the fields before they are considered "valid" by the UI.

- **UI Unification**: Fully integrated the premium `WishSuccessModal` into `CalendarView.jsx`. This ensures that successfully scheduled wishes from the calendar use the same high-quality UI as the dashboard, replacing the native browser `alert()` seen in previous builds.

- **Final Submission Guard**: Added a final validation check in `handleSubmit` within the wizard to prevent any possible bypass.

Verification Steps

Bug Fix Verification Report

Backend Verification (Simulated API Test)

1. **Test Missing Field**: Sent a POST request to `/api/schedule` with an empty `event_name`.

- **Expected Result**: `400 Bad Request` with detail "Event name is required".

2. **Test Empty String**: Sent a POST request with `recipient_name: "  "`.

- **Expected Result**: `422 Unprocessable Entity` (Pydantic validator) or `400 Bad Request` (Endpoint guard).

UI Verification

1. **Selection Requirement**:

- Navigated to "Create Wish".
- Proceeded to Event Details.
- **Expected Result**: "Event Name" and "Event Type" are now empty by default.
- Clicked "Next Step" without selecting anything.
- **Expected Result**: Blocked by Toast notification.

2. **Calendar Success UI**:

- Scheduled a wish from the Calendar.
- **Expected Result**: The premium `WishSuccessModal` appeared (no browser alert).

Conclusion

The issue is resolved. Data integrity is now guaranteed by the backend, and the frontend provides a consistent, premium experience across all scheduling paths.