Bug Verification Report - Bug #1816: Duplicate Wish Scheduling Prevention

Date: 2026-01-09 **Bug ID:** #1816 **Title:** Duplicate wish can be scheduled when user clicks "Confirm Schedule" again after success popup **Status:** Fixed

Issue Description

Users were able to schedule the same wish multiple times because the "Confirm Schedule" button remained active and the form was not reset after a successful submission. If a user closed the success alert and clicked the button again, a duplicate wish was created.

Epic: #1599 Occasion & Event Management **User Story:** #1651 Create birthday or anniversary event

Fix Implementation

Multiple layers of prevention were added to ensure a wish is only scheduled once per session.

Changes:

- **Loading Guards**: Added `if (loading) return;` at the beginning of the submission handlers in both `dashboard/page.js` and `CalendarView.jsx`. This prevents concurrent API calls from multiple clicks.
- **Persistence of Loading State**: For scheduled wishes, `setLoading(false)` is now only called AFTER the success modal is dismissed or an action is taken. This keeps the submit button in a "disabled" state while the success message is visible.
- **Wizard Reset (Remounting)**: Introduced a `wizardKey` state in `dashboard/page.js`. Whenever a wish is successfully scheduled, the key is incremented, forcing the `CreateWishWizard` component to completely unmount and remount. This clears all form state and resets the wizard to Step 0.
- **Redirect on Modal Close**: Updated the `onClose` handler of the `WishSuccessModal` to redirect the user to the "Overview" tab. This ensures that even if they try to bypass the wizard reset, they are moved away from the submission form.

Verification Steps

Manual Verification

1. **Duplicate Click Prevention**:
   - Navigated to "Create Wish".
   - Clicked "Confirm Schedule" and immediately tried to click it again several times.
   - **Expected Result**: Only one API call was triggered; subsequent clicks were ignored due to the loading guard.
2. **Form Reset After Success**:
   - Completed a wish schedule.
   - Observed the success modal.
   - Clicked "X" to close the modal.
   - **Expected Result**: Redirected to the Dashboard Overview. Returning to the "Basic" tab showed a fresh, reset wizard at Step 0.
3. **Confirming Button State**:
   - While the success modal was open, observed that the wizard in the background (if visible) had the button in a loading/disabled state.
4. **Calendar Resilience**:
   - Created a wish from the Calendar.
   - **Expected Result**: Modal closed immediately on success, and the calendar refreshed, preventing any re-submission of the same data.

Conclusion

The issue is resolved. The system now robustly prevents duplicate wish scheduling through UI guards, state resetting, and automatic redirection.