Lovable Prompt: Fix Critical UX Bug in Customer Address Flow

Project: Lawnly.com.au - Critical UX Bug Fix

Component: Customer Address Modal (`Add New Address`)

1. Context

The customer dashboard features a "Getting Started" checklist. The first step is "Add Your Address," which opens a multi-step modal. In Step 1 of this modal, the user enters their street address into an input field that is integrated with the Google Places Autocomplete API. As the user types, a dropdown of suggested addresses appears. When the user selects an address from this list, the form fields for Street, Suburb, State, and Postal Code are correctly auto-populated.

2. The Problem (The Bug)

After the Google Places Autocomplete populates the address fields, the **"Next: Mark Lawn Areas" button remains disabled**. The form does not recognize that the fields have been filled, and the user is blocked from proceeding to the next step.

A non-obvious workaround exists: if the user manually clicks into one of the populated fields (e.g., "Suburb") and then clicks out of it, the form's state updates, and the "Next" button becomes enabled. This is a critical UX failure, as the vast majority of users will not discover this. They will assume the website is broken and abandon the process.

3. Expected Behavior (The Solution)

The "Next: Mark Lawn Areas" button should be **enabled automatically and immediately** after the user selects an address from the Google Places Autocomplete dropdown. The user should not be required to perform any additional actions.

The correct flow should be:

1. User types their address.
2. User clicks a valid address from the autocomplete suggestions.
3. The address fields are auto-populated.
4. The "Next: Mark Lawn Areas" button immediately transitions from a disabled to an enabled state.
5. User can click "Next" to proceed to Step 2 (Mark Your Lawn Areas).

4. Technical Implementation Guidance

This bug is likely caused by the form's state not being updated when the Google Places Autocomplete populates the fields programmatically. The fix involves triggering a state update or validation check within the autocomplete's event listener.

- **Event Listener:** The Google Places Autocomplete widget has an event listener for when a user selects a prediction. The event is typically `places_changed`.
- **Trigger Validation:** Inside the `places_changed` event handler, after you have retrieved the address components and populated the form fields, you must **programmatically trigger the form's validation logic**.
- **Example (React):** If you are using a library like React Hook Form, you might call a function like `trigger()` after setting the field values with `setValue()`. If using standard React state, ensure you are setting the state correctly for each field, which should cause a re-render and re-evaluation of the button's disabled status.

5. Acceptance Criteria

To confirm the bug is fixed, the following must be true:

- [ ] When the "Add New Address" modal is first opened, the "Next: Mark Lawn Areas" button is disabled.
- [ ] After typing an address and selecting a valid option from the Google Places Autocomplete dropdown, the "Next: Mark Lawn Areas" button becomes enabled.
- [ ] This enabling must happen **without** any further user interaction (i.e., no extra clicks in or out of fields).
- [ ] The user can successfully click the enabled "Next" button to proceed to the "Mark Your Lawn Areas" step.
- [ ] The form validation should still work correctly if the user tries to manually clear a required field after autocompletion (i.e., the "Next" button should become disabled again).