

SmartSuggest State Management Fix - Technical Analysis

Executive Summary

The Roblox Verifier app has critical state management issues preventing SmartSuggest from working repeatedly. After one SmartSuggest flow, subsequent searches fail to trigger SmartSuggest and show “Not Found” instead. This document details all root causes and comprehensive fixes.

Root Cause Analysis

Issue #1: Input Field Being Cleared (Line 201-203)

Location: `handleSubmit` function, lines 201-203

Problem:

```
if (!skipInputClear) {  
  setInput('');  
}
```

Root Cause:

- The `skipInputClear` parameter defaults to `false` in the function signature
- When `handleSelectCandidate` calls `handleSubmit`, it passes `false` for `skipInputClear`
- This causes the input field to be cleared after every submit, including after selecting a SmartSuggest candidate

Impact: User must re-type the search query after every verification

Issue #2: SmartSuggest Not Re-triggering After First Use

Location: `handleSubmit` function, lines 89-96 and 205-241

Problem:

```
// Lines 89-96: State reset at start
setResult(null);
setBatchResults([]);
setScoredCandidates([]);
setOriginalDisplayNameQuery('');

// Lines 205-241: Result rendering logic
if (out.status === 'Verified') {
  setScoredCandidates([]); // ← Clears candidates
  setResult(<div>...Verified...</div>);
} else if (out.status === 'Not Found') {
  setScoredCandidates([]); // ← Clears candidates
  setResult(<div>...Not Found...</div>);
}
```

Root Causes:

1. **Premature State Reset:** Lines 92-95 clear `scoredCandidates` and `originalDisplayNameQuery` at the START of `handleSubmit`, before any API calls
2. **Sticky “Verified” State:** After selecting a candidate and verifying (line 206), `scoredCandidates` is cleared but `result` is set to the Verified component
3. **Missing State Reset in “Not Found” Path:** When a subsequent search returns “Not Found”, the code clears candidates (line 233) but doesn’t properly reset to allow `SmartSuggest` to work again
4. **Race Condition:** The state is cleared at the beginning (line 94), but if the previous search had candidates, they’re gone before the new search completes

Flow Breakdown:

1. First search: “John” → No exact match → `SmartSuggest` shows candidates ✓
2. User clicks “Select & Verify” on a candidate → Verified state set, candidates cleared
3. Second search: “Jane” → State cleared at start → No exact match → Should trigger `SmartSuggest`
4. **BUG:** The outputs array gets “Not Found” status because the `displayName` search path (lines 130-150) correctly sets suggestions, BUT the result rendering logic (lines 232-240) shows “Not Found” UI instead of letting `SmartSuggest` render

The Critical Bug:

The issue is in the result rendering logic. When `out.status === 'Not Found'`, it renders a red error box (lines 232-240) instead of checking if there are candidates to show. The `SmartSuggest` component only renders when `scoredCandidates.length > 0` (line 365), but the “Not Found” result box takes precedence.

Issue #3: Incorrect Status Logic for SmartSuggest

Location: Lines 144-149 and 182-187

Problem:

```
outputs.push({
  input: singleInput,
  status: candidates.length > 0 ? 'Suggestions' : 'Not Found',
  suggestions: candidates,
  details: candidates.length === 0 ? 'No matches' : undefined,
});
```

Root Cause:

- The status is set to "Not Found" when `candidates.length === 0`
- The result rendering logic (lines 232-240) then shows a red error box for "Not Found"
- This prevents SmartSuggest from rendering even when it should

Correct Logic Should Be:

- If candidates exist → Don't set `result` state, let SmartSuggest component render
- Only show "Not Found" error when truly no candidates found

Issue #4: handleSelectCandidate Implementation

Location: Lines 274-276

Problem:

```
const handleSelectCandidate = async (username: string) => {
  await handleSubmit({ preventDefault: () => {} } as React.FormEvent, [username], false);
};
```

Root Causes:

1. Passes `false` for `skipInputClear`, causing input to be cleared
2. Passes username as batch input array, which triggers different code path
3. Doesn't properly reset state before re-submitting

Comprehensive Fix Strategy**Fix #1: Preserve Input After Submit**

Change: Modify `handleSelectCandidate` to pass `true` for `skipInputClear`

```
const handleSelectCandidate = async (username: string) => {
  await handleSubmit({ preventDefault: () => {} } as React.FormEvent, [username], true);
};
```

Fix #2: Proper State Management Flow**Changes:**

1. **Don't clear state prematurely** - Only clear state that needs to be cleared
2. **Don't render "Not Found" when candidates exist** - Check for candidates before showing error
3. **Reset all relevant state at the right time** - Clear previous results but not mid-flight

Updated handleSubmit logic:

```

const handleSubmit = async (e: React.FormEvent, batchInputs: string[] = [], skipInputClear: boolean = false) => {
  e.preventDefault();
  setLoading(true);

  // Clear previous results but NOT candidates yet (they might be needed for comparison)
  setResult(null);
  setBatchResults([]);

  // Only clear candidates and query if starting a fresh search (not from candidate selection)
  if (batchInputs.length === 0) {
    setScoredCandidates([]);
    setOriginalDisplayNameQuery('');
  }

  setIsBatchMode(batchInputs.length > 0);
  // ... rest of logic

```

Fix #3: Correct Result Rendering Logic

Change: Only show “Not Found” error when truly no candidates, otherwise let SmartSuggest render

```

if (!isBatchMode && outputs.length === 1) {
  const out = outputs[0];

  if (!skipInputClear) {
    setInput('');
  }

  if (out.status === 'Verified') {
    // Clear candidates only after successful verification
    setScoredCandidates([]);
    setOriginalDisplayNameQuery('');

    setResult(/* Verified UI */);
  } else if (out.status === 'Not Found' && (!out.suggestions || out.suggestions.length === 0)) {
    // Only show Not Found if truly no candidates
    setScoredCandidates([]);
    setOriginalDisplayNameQuery('');

    setResult(/* Not Found UI */);
  } else if (out.status === 'Suggestions' && out.suggestions && out.suggestions.length > 0) {
    // Don't set result - let SmartSuggest component render
    // Candidates are already set in setScoredCandidates earlier
  }
}

```

Fix #4: Clean State Transitions

Principle: Each search should be independent with clean state

State Reset Points:

1. **Start of new search (user types and submits):** Clear all previous state
2. **After verification:** Clear candidates and query, show verified result

3. **After “Not Found”:** Clear candidates and query, show error
 4. **After candidate selection:** DON'T clear input, proceed with verification
-

Implementation Changes

Modified Functions

1. `handleSubmit` (Lines 89-245)

- Remove premature state clearing of `scoredCandidates` and `originalDisplayNameQuery`
- Add conditional state clearing based on whether it's a fresh search or candidate selection
- Fix result rendering logic to not show “Not Found” when candidates exist
- Ensure proper state cleanup after verification

2. `handleSelectCandidate` (Lines 274-276)

- Change `skipInputClear` parameter from `false` to `true`
-

Testing Acceptance Criteria

✓ Test Case 1: Repeated SmartSuggest Usage

1. Search for “John Doe” (non-exact match)
2. SmartSuggest shows candidates
3. Click “Select & Verify” on a candidate
4. Verification succeeds
5. Search for “Jane Smith” (non-exact match)
6. **Expected:** SmartSuggest shows candidates again
7. **Previous Bug:** Showed “Not Found” instead

✓ Test Case 2: Input Persistence

1. Type “John Doe” in input field
2. Submit search
3. SmartSuggest shows candidates
4. Click “Select & Verify”
5. **Expected:** Input field still shows “John Doe”
6. **Previous Bug:** Input field was cleared

✓ Test Case 3: True “Not Found” Scenario

1. Search for “xyzabc123nonexistent”
2. No candidates found
3. **Expected:** Red “Not Found” error box appears
4. **Should Work:** This case should still work correctly

✓ Test Case 4: Multiple Searches Without Refresh

1. Search “Alice” → SmartSuggest → Select → Verify
2. Search “Bob” → SmartSuggest → Select → Verify
3. Search “Charlie” → SmartSuggest → Select → Verify

4. **Expected:** All three searches work independently without refresh
 5. **Previous Bug:** Only first search worked, rest showed “Not Found”
-

Code Quality Improvements

State Management Best Practices Applied

1. **Single Source of Truth:** Each state variable has clear ownership
2. **Predictable State Transitions:** Clear flow from search → candidates → verification
3. **No Stale State:** Proper cleanup at appropriate times
4. **Idempotent Operations:** Each search is independent and repeatable

Performance Considerations

- No unnecessary re-renders
 - State updates batched where possible
 - Async operations properly handled
-

Conclusion

The root cause was a combination of:

1. **Premature state clearing** at the start of handleSubmit
2. **Incorrect result rendering logic** that showed “Not Found” instead of SmartSuggest
3. **Input clearing** on every submit including candidate selection
4. **Missing conditional logic** to differentiate between fresh searches and candidate selections

The fix ensures:

- ☒ SmartSuggest works repeatedly without refresh
 - ☒ Input persists after submit
 - ☒ Clean state transitions between searches
 - ☒ “Not Found” only appears when truly no candidates exist
 - ☒ No sticky state between independent searches
-

Files Modified

1. `src/app/page.tsx` - Main component with state management fixes

Files Created

1. `SMARTSUGGEST_FIX_ANALYSIS.md` - This technical analysis document