

## OpenAI Deep Research

Given the Software Engineering issues 1 to 10 in the following repository, analyze each of them and give a thorough description of what is required of each, also give all the requirements for each in point format. Finally, give a score out of 10 for the complexity of each and an estimated number of minutes to solve each:

<https://github.com/openai/SWELancer-Benchmark/tree/main/issues>

I will analyze Software Engineering issues 1 to 10 from the provided repository. My analysis will include:

- A thorough description of each issue and its requirements.
- A detailed list of requirements in point format.
- A complexity score (out of 10) for each issue.
- An estimated time (in minutes) to solve each issue.
- Categorization of each issue (e.g., bug fix, feature request, documentation improvement, etc.).
- Suggested approaches for solving each issue.
- Dependencies or related components impacted by each issue.

I will update you as soon as my research is complete.

## Analysis of Issues 1–10 (SWELancer-Benchmark Repository)

### Issue 1: Expense – Hold Option Missing in Context Menu

**Detailed Description:** ([SWELancer-Benchmark/issues/1/issue\\_data.json](https://github.com/openai/SWELancer-Benchmark/issues/1/issue_data.json) at main · openai/SWELancer-Benchmark · GitHub) In the Expensify app, right-clicking an expense preview should show a “**Hold**” option in the context menu (as in production). However, this “**Hold**” option is currently not appearing at all ([SWELancer-Benchmark/issues/1/issue\\_data.json](https://github.com/openai/SWELancer-Benchmark/issues/1/issue_data.json) at main · openai/SWELancer-Benchmark · GitHub). The issue requires adding or restoring the missing **Hold** action so that users can place an expense on hold via the context menu.

## Requirements:

- Ensure the “**Hold**” option is present in the expense context menu for relevant expense items ([SWELancer-Benchmark/issues/1/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).
- The **Hold** action should function as expected (placing the expense on hold) consistent with production behavior.
- Verify the context menu UI updates correctly across supported platforms (web, mobile) when adding the **Hold** option.
- No regression: other context menu options should remain unaffected and continue to work.

**Complexity:** 3/10 (Relatively straightforward UI fix in context menu).

**Estimated Time:** ~90 minutes to implement and test.

**Category:** Bug Fix (UI functionality missing).

**Suggested Approach:** Identify where the expense context menu items are defined in the code and add the missing **Hold** menu item. This likely involves enabling a flag or adding an entry for **Hold** in the menu configuration. Review production code or specs to ensure the **Hold** option’s visibility conditions are met, then implement the change and test by right-clicking an expense to confirm the **Hold** option appears and is clickable.

**Dependencies:** This fix impacts the **Money Requests/Expense** component that renders the context menu. It may involve files related to expense context menu logic or localization of menu items. Ensure that any state or permission checks for showing **Hold** (if any) are handled. Other components (expense list, context menus system-wide) should be checked to ensure adding this option doesn’t interfere with their behavior.

## Issue 2: Subscription – Currency Selection Resets Incorrectly After Reload

**Detailed Description:** ([SWELancer-Benchmark/issues/2/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) On the **Subscription** page, changing the currency and then reloading the page leads to an incorrect selection shown. For example, if a user selects **AUD** and reloads, the page still highlights **AUD** as selected, whereas it should default back to **USD** (the expected default) upon reload ([SWELancer-Benchmark/issues/2/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). This issue is about fixing the currency dropdown state so it reflects the correct default after a page refresh.

## Requirements:

- After a page reload on the Subscription page, the currency dropdown should display the correct default currency (**USD** in this case) ([SWELancer-Benchmark/issues/2/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

- Changing the currency should only persist if intended; if not saved, the UI should not “remember” the last selection across reloads.
- Ensure the selected currency in the UI always matches the actual currency setting in effect (no stale or incorrect state).
- Test that this behavior is consistent in staging and production environments, and on all platforms (web, desktop).

**Complexity:** 4/10 (State management issue, moderate complexity).

**Estimated Time:** ~120 minutes to diagnose and fix.

**Category:** Bug Fix (State/UX bug).

**Suggested Approach:** Investigate how the Subscription page stores the selected currency. The likely cause is that the selection state is being cached or not reset on component mount. The fix could involve clearing the selection state on page load or ensuring the dropdown is initialized from the actual subscription settings each time. Implement a solution such as resetting the dropdown value to the default (USD) on render if no saved preference exists. After the fix, test by selecting a different currency, reloading, and confirming the dropdown shows the correct default (and not the previously selected currency).

**Dependencies:** Affects the **Subscription Settings** component and its currency selector. It may involve the form state or Redux store (if used) for subscription info. Ensure that related components (like payment method or subscription summary) that display currency are in sync. No other components should be impacted beyond the subscription currency selection logic.

## Issue 3: Unable to Leave a Workspace as an Invited Member

**Detailed Description:** ([SWELancer-Benchmark/issues/3/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) Users who are invited to a workspace (with a member role) have **no option to leave that workspace** on their own. In the workspace settings, an invited member is expected to have a way to remove themselves (leave the workspace), but currently \*\*no “Leave workspace” button or similar control is provided ([SWELancer-Benchmark/issues/3/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). This means members are stuck in a workspace unless an admin removes them, which is a usability issue.

### Requirements:

- Add a **“Leave Workspace”** feature for non-admin workspace members (e.g., a button in workspace settings).
- On triggering leave, remove the user’s membership from that workspace (backend call to remove member).
- Provide confirmation or feedback to the user (e.g., a prompt “Are you sure?” and success message upon leaving).
- Ensure proper permission checks: only allow this option for members who are not the sole admin/owner of the workspace.

- After leaving, the workspace should no longer appear in the user's workspace list, and they should lose access to its data.

**Complexity:** 5/10 (Introduces a new flow – moderate effort).

**Estimated Time:** ~180 minutes to implement and test across platforms.

**Category:** Feature Improvement (Usability enhancement; could be seen as a bug/oversight).

**Suggested Approach:** In the workspace settings UI, add a **“Leave workspace”** button for eligible users (members, not owners). When clicked, call the existing API or action to remove the current user from that workspace's members. Handle edge cases, such as if the user is the last admin (in which case leaving might not be permitted without transferring ownership). Implement a confirmation dialog to prevent accidental removals. After a successful removal, update the app state to reflect that the workspace is gone from the user's list and navigate the user appropriately (perhaps to the home or workspace list screen). Test by inviting a user to a workspace and then using the new feature to leave it, verifying they no longer see the workspace and no errors occur.

**Dependencies:** Involves the **Workspace Settings** component and possibly the backend API for workspace membership management. It impacts the membership data for workspaces – ensure the front-end uses the correct endpoint to remove a user. Related components include the workspace list (needs to refresh) and any badge/notification counts for that workspace. Documentation (contributing guidelines, if any) might also be updated to mention how members can leave a workspace.

## Issue 4: Reports – “Pay” Button Not Changing to “Review” After Category Removal

**Detailed Description:** ([SWELancer-Benchmark/issues/4/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) On a **Report** (particularly a group expense report in a workspace), if the setting *“Members must categorize all expenses”* is enabled, removing the category from an expense should introduce a policy violation. In that case, the report's **“Pay”** button is expected to change to a **“Review”** button (indicating attention needed) immediately ([SWELancer-Benchmark/issues/4/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). **Actual behavior:** After unselecting the expense's category, the button stays as **“Pay”** and only switches to **“Review”** after a full page refresh, which is incorrect (the UI isn't updating in real-time).

### Requirements:

- When a user removes/unselects a category from an expense in a report, the report's primary action button should update from **Pay** to **Review** immediately (without needing a refresh) ([SWELancer-Benchmark/issues/4/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).
- Conversely, if the category is re-added or all violations resolved, the button should switch back to **Pay** dynamically.

- Ensure the violation indicator/state for the report is updated as soon as an expense becomes uncategorized (triggering the state change for the button).
- This should apply consistently in the Outstanding report view and anywhere else the Pay/Review status may be shown for the report.
- No new issues should be introduced: e.g., the button text should not flicker or show wrong status in other scenarios.

**Complexity:** 4/10 (Front-end state update issue).

**Estimated Time:** ~120 minutes to fix and validate.

**Category:** Bug Fix (UI state/logic bug).

**Suggested Approach:** Investigate how the report's pay/review status is determined. It likely checks if any expenses have rule violations (like missing category) to decide the button label. The bug suggests that this check isn't happening after the category removal event. The fix could involve triggering a state update or re-computation when an expense's category is cleared. For example, after an expense edit (unselecting category), dispatch an update to the report or recalc the policy violations count. Implement an event or callback so that when the expense is updated, the parent report component knows to refresh its data (perhaps via context or by refetching the report info). Test by following the exact steps: mark an expense without category and confirm the button changes to **Review** immediately, then add a category back to see it revert to **Pay**, all without manual refresh.

**Dependencies:** Affects the **Reports/Expenses** components and how they interact. The **Outstanding report view** and the right-hand panel (RHP) for expense editing are involved. Ensure that any state management (Redux store or local component state) for the report's completeness is updated when an expense changes. This might involve the rule enforcement logic (likely in the front-end for immediate feedback). Other components (like summary badges or lists showing count of violations) should also reflect changes if present.

## Issue 5: Expense – “Canceled” Spelling and Punctuation in System Message

**Detailed Description:** ([SWELancer-Benchmark/issues/5/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) When a payment on an expense is canceled, the system generates a chat message indicating this action. Currently, that message reads that the payment was “**cancelled**” (with two “l”s) and includes a full stop (period) at the end ([SWELancer-Benchmark/issues/5/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). The expected copy is to use American spelling “**canceled**” (one “l”) **without a period** at the end of the sentence ([SWELancer-Benchmark/issues/5/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). This is a minor text/consistency issue.

### Requirements:

- Update the system message text for canceled payments to use “**canceled**” (single / spelling) instead of “cancelled” ([SWELancer-Benchmark/issues/5/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

- Remove the trailing period in that system message (the message should end without a full stop as per style guide).
- Ensure the message format matches the expected copy exactly: “[User] canceled the [amount] payment” (with no punctuation after)  
([SWELancer-Benchmark/issues/5/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).
- Verify this text change across all platforms (it’s likely a shared string for web, mobile, etc.) and that it doesn’t break any automated tests or string keys.
- Check if any other related messages use the word “cancelled” and correct those for consistency, if needed.

**Complexity:** 1/10 (Simple text change).

**Estimated Time:** ~15 minutes (find and replace text, plus quick test).

**Category:** Bug Fix (UI/Text polish).

**Suggested Approach:** Locate the source of the system message (perhaps in the localization strings or directly in the code where the cancel action is handled). It might be a constant string or constructed message when a payment is canceled. Change “**cancelled**” to “**canceled**” and remove the period. Since this is a user-facing string, double-check if it’s coming from a translations file; if so, update the appropriate locale (likely English) string. After the change, test by performing the cancel payment action on an expense: the resulting system message in the expense chat should read exactly as intended (with correct spelling and no period).

**Dependencies:** The change touches the **system message generation** for expense payments. It may reside in a function that posts system comments or in a language file. The impact is limited to the display text of that message. No broader functionality is affected, but if there are unit or integration tests checking the exact message text, those will need updating to match the new string.

## Issue 6: Workspace Chat – “Split Expense” Should Not Be the First Create Option

**Detailed Description:** ([SWELancer-Benchmark/issues/6/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) In a workspace chat, clicking the “+” (create) menu shows options like “Split expense” and “Submit expense.” Currently, “**Split expense**” appears as the first option in the menu, which is not desired. The expected behavior is that “**Submit expense**” (or the standard expense submission) is listed first, with “**Split expense**” after it ([SWELancer-Benchmark/issues/6/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). This issue is about reordering those menu options for better UX.

### Requirements:

- Change the order of items in the workspace chat’s create menu so that “**Submit expense**” is the first option and “**Split expense**” is later (second in the list)  
([SWELancer-Benchmark/issues/6/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).



- Ensure the menu labeling is correct (the term might be “Submit expense” or “New expense” – use the same wording as elsewhere in the app for consistency).
- Verify the new ordering on all clients (web, mobile) – the first menu item should be the regular expense creation.
- Make sure that the functionality of each option remains the same after reordering (only the positions swap).
- If there are any screenshots or docs referencing this menu, update them if necessary (likely not critical unless documented).

**Complexity:** 2/10 (Minor UI change).

**Estimated Time:** ~30 minutes (simple reordering fix).

**Category:** UX Improvement (minor tweak, likely labeled as a bug).

**Suggested Approach:** Find where the workspace chat FAB (floating action button) or “+” menu options are defined in the code. This could be a static list or JSX structure listing “Split Expense” and “Submit Expense.” Reorder these items so the intended one is first. If the items are dynamically generated or sorted, adjust the sorting criteria or the order of insertion. After making the change, run the app and open a workspace chat, click “+”, and confirm that **Submit/New Expense** is the first option visible, with **Split Expense** following. Test on a couple of platforms or screen sizes to ensure the menu renders correctly in the new order.

**Dependencies:** Involves the **Workspace Chat compose/create menu** component. This likely does not affect anything outside of that menu, since it’s an isolated UI element. Just ensure that if the menu order was assumed elsewhere (for example, if a default selection or focus was on the first item), those assumptions still hold logically with the new order (likely not an issue). No backend impact.

## Issue 7: Onboarding – Multiple Workspaces Created by Navigating Back and Forth

**Detailed Description:** ([SWELancer-Benchmark/issues/7/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) During the **new account onboarding** flow, users choose an option to create a workspace (e.g., selecting company size). The bug is that if the user goes *back* and changes their selection, the app calls the **CreateWorkspace** action **again**, resulting in multiple workspaces being created for one user ([SWELancer-Benchmark/issues/7/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) ([SWELancer-Benchmark/issues/7/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). The expected outcome is to create only **one** workspace per onboarding session, regardless of navigation; changing options should not spawn additional workspaces ([SWELancer-Benchmark/issues/7/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

### Requirements:

- Ensure that only **one workspace** is created during the onboarding process for a new user ([SWELancer-Benchmark/issues/7/issue\\_data.json at main ·](#)

[openai/SWELancer-Benchmark · GitHub](#)). Going back and selecting a different employee count (or similar step) should not trigger another workspace creation.

- Modify the flow so that if a workspace was already created on a previous step, it doesn't create a duplicate. For instance, only call the workspace creation API once, either at the final confirmation or the first time through, and reuse that if the user changes selection.
- If multiple employee count selections need to be explored by the user, perhaps delay creating the workspace until the choice is finalized (i.e., after they click "Continue" on the last step of that flow).
- Clean up any partially created duplicate data if necessary (though primarily a front-end fix, we should ensure no orphaned workspaces remain – however, in staging/prod this likely already happened, but that's outside the scope).
- Test the entire onboarding: selecting an option, continuing, going back, selecting another, and finishing – confirm that only one workspace exists in the end.

**Complexity:** 5/10 (Workflow logic change – moderate complexity).

**Estimated Time:** ~180 minutes (needs careful tweaking and testing of flow).

**Category:** Bug Fix (Logic/Workflow bug).

**Suggested Approach:** Review the onboarding modal code to see where `CreateWorkspace` is called. The description indicates it's called immediately upon choosing an employee range option. A better approach would be to call `CreateWorkspace` only once, perhaps when the user has finalized the onboarding choices. There are a few strategies:

- **Option A:** Defer workspace creation until the end of onboarding. Collect the choices in state and only create the workspace when the user finishes onboarding (e.g., clicks a final "Done" or moves past the last step).
- **Option B:** If creation must happen early (for some reason, like needing a workspace ID to proceed), then track that a workspace has been created (store the ID in context/state) and skip creating another if the user navigates back. Instead, reuse the existing workspace (maybe update it if needed, though likely just use it).
- Implement one of these solutions. The simplest might be deferring creation. After the change, test by simulating a new signup: go through onboarding, choose an option (notice if a workspace is created at that moment or later), go back, pick a different option, continue, and then check the database or UI to ensure only one workspace exists for the account.

**Dependencies:** Touches the **Onboarding flow** and **workspace creation logic**. It involves the front-end workflow, but it also has implications for backend (to avoid duplicate entries). If the backend doesn't prevent duplicates, the front-end must handle it carefully. Ensure that whatever flag or state used to prevent duplicate creation is properly reset if the user truly restarts onboarding or logs out/in again. Other parts of the app that assume one workspace per new user (like post-onboarding experience) will benefit from this fix.



## Issue 8: Bank Account – State Search Query Not Cleared on Modal Close

**Detailed Description:** ([SWELancer-Benchmark/issues/8/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) When connecting a bank account manually through **Workspace settings > Workflows**, there is an address form where you select a **State** from a list. If you type a search query to filter the state list and then close the list (navigate back), the next time you open the state selector, the **previous search term is still present**, filtering the list ([SWELancer-Benchmark/issues/8/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). Expected behavior is that leaving the state selection should **clear any search query** so that when reopened, the full state list is shown with an empty search field ([SWELancer-Benchmark/issues/8/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

### Requirements:

- Clear the state filter query when the state selection modal/list is exited. In other words, the **State search box should be reset to empty** whenever you close the state picker.
- Ensure that upon re-opening the state list, all states are visible (no lingering filter from last time).
- This should hold true across platforms (the bug is noted on Android, iOS, web, desktop – basically everywhere – so likely a shared component).
- Also ensure consistency: if there are similar search-in-list interfaces (for country, city, etc.), consider applying a similar reset logic there, so the UX is consistent (though the issue specifically calls out the State field).
- Test by performing the exact steps: enter a query (e.g., type “New”) in State selection, hit back, then go into State again – the search field should be blank and the list unfiltered.

**Complexity:** 2/10 (Small UI state fix).

**Estimated Time:** ~45 minutes to implement and test.

**Category:** Bug Fix (UX polish).

**Suggested Approach:** The likely solution is to reset the search input state for the state picker component either when it unmounts or when it is about to be shown. For instance, if the state list is a separate component or modal, add a cleanup in its `componentWillUnmount` or equivalent (for functional components, use `useEffect` cleanup) to clear the query state. If the search query is stored in a higher-level state or Redux store, dispatch an action to clear it when the user exits the screen. Another approach is to re-initialize the state list component fresh each time it opens (not preserving its state between opens). Implement the chosen fix, then test by reproducing the scenario: after closing the state list with a query, verify the query is gone when reopening.

**Dependencies:** Concerns the **Personal Information / Address form** within the **Bank account connection workflow**. The component likely affected is the **State selection dropdown/modal**. It may use a generic picker component – if so, changes might be in a generic place (ensuring the picker doesn’t remember old input unless intended). Ensure that clearing the query doesn’t

interfere with any “recent selection” logic (typically not applicable for a state list). Other form fields with search filtering (perhaps country picker) should be checked if they exhibit similar behavior; if yes, applying a similar fix would be wise for consistency.

## Issue 9: Workspace “Default” Label Tooltip Truncated in Spanish

**Detailed Description:** ([SWELancer-Benchmark/issues/9/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) In **Workspace Settings** (Espacios de trabajo) when the app is in Spanish, a workspace marked as “**Predeterminado**” (Default) has a tooltip explanation. The content of this tooltip is longer in Spanish and currently **gets cut off (truncated)** when displayed ([SWELancer-Benchmark/issues/9/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). The expected result is that the tooltip should display the full text without truncation, regardless of language ([SWELancer-Benchmark/issues/9/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

### Requirements:

- Adjust the tooltip for the “Default” workspace label so that its content is **fully visible** in Spanish (and by extension, any language).
- The tooltip should dynamically accommodate longer text – this may mean allowing the tooltip to wrap text to multiple lines or increasing its width.
- Ensure the tooltip still looks good in English and other languages (no excessive blank space or odd formatting if the text is short).
- Test the fix with Spanish language enabled: hover over “Predeterminado” and confirm the entire explanation is readable. Also test in English to ensure it still appears normally.
- If the tooltip is a common component used elsewhere, verify that the change doesn’t negatively affect other tooltips in the app.

**Complexity:** 3/10 (Styling/UI fix with i18n consideration).

**Estimated Time:** ~60 minutes to fix and test.

**Category:** Bug Fix (Internationalization/UI issue).

**Suggested Approach:** Likely this involves CSS or style adjustments. Inspect the tooltip component (could be a custom tooltip or a third-party one). If it has a fixed width that’s too small, consider increasing the max-width or allowing it to auto-size based on content. Another solution is to enable text wrapping inside the tooltip container so longer words/phrases can break into a second line instead of overflowing. For example, adding CSS like `white-space: normal` and maybe a larger width constraint can help. Implement the change, then test by switching the app to Spanish and hovering over the default label – the tooltip should now show the full sentence (which was previously cut). Also test in a scenario where the tooltip text is short (like in English “Default” case) to ensure it doesn’t introduce a huge empty tooltip; ideally, the tooltip should size just large enough for content.

**Dependencies:** Affects the **Tooltip UI component** for workspace default labels. If a common tooltip component is used application-wide, the style change will impact all tooltips – which is usually fine if it's an improvement (most tooltips should be allowed to show full text). Just ensure other tooltips with potentially long content also benefit (or at least are not broken by this). It primarily impacts the front-end presentation; no backend or logic changes are involved.

## Issue 10: Category “Receipt Required” Amount Shows Wrong Default

**Detailed Description:** ([SWELancer-Benchmark/issues/10/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)) In **Workspace Settings > Categories**, there is a setting “**Require receipts over [amount]**” which defaults to a certain value (e.g., £25.00). The bug is that when viewing a category's settings, the default threshold is incorrectly displayed as **£2000.00** instead of the actual workspace default £25.00 ([SWELancer-Benchmark/issues/10/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)). Internally, the rule is still £25 (as evidenced by violations triggering at £25), so this is a **display issue** where the UI is pulling or showing the wrong value ([SWELancer-Benchmark/issues/10/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).

### Requirements:

- Display the correct default “**require receipts over**” amount in the category settings panel. It should match the workspace's configured value (e.g., £25.00 by default) ([SWELancer-Benchmark/issues/10/issue\\_data.json at main · openai/SWELancer-Benchmark · GitHub](#)).
- Ensure that the value shown updates if the workspace rule is changed – i.e., it's not a hardcoded number, but the actual setting from the workspace or policy.
- Confirm that this is just a front-end issue: the backend uses the correct value (the note says backend is correct). So the task is to fetch/use the right data for the UI.
- After the fix, test by checking a workspace's receipt requirement setting (e.g., if it's default £25) and then opening a category's settings to see that same value. If possible, also test by changing the workspace's receipt threshold (if the app allows) and verifying the category view reflects the new value.
- The code reference given (CategoryRequireReceiptsOverPage.tsx line 54) should be reviewed to ensure it uses the proper prop/state (likely it was mistakenly using a max expense amount or some other constant).

**Complexity:** 4/10 (Data wiring bug – moderate).

**Estimated Time:** ~120 minutes to identify the data flow and fix it.

**Category:** Bug Fix (Data Display issue).

**Suggested Approach:** Examine the component that renders the “**Require receipts over [amount]**” field in category settings (the hint suggests

`CategoryRequireReceiptsOverPage.tsx`). It's likely using an incorrect value – possibly a placeholder or a constant (like the maximum allowed value) instead of the actual policy value. Update the component to use the correct source: it should retrieve the workspace's policy setting for receipt requirement. This data might be available in the workspace or policy object in the app's store. If the data isn't being passed down, modify the logic to fetch it (perhaps the categories page should receive the policy info). Once corrected, the UI will show the right default. Test by following the steps: look at the main **Workspace Rules** page to see the "Require receipts over" value (e.g., £25), then open a category's settings and ensure it shows £25 as well (not £2000). Also verify that the violation behavior still works (to be sure the change didn't affect logic, which it shouldn't, as it's purely display).

**Dependencies:** This fix touches the **Workspace Categories settings** UI. It depends on workspace policy data (the receipt threshold). Ensure that the component has access to that data – it might require passing a prop or selecting from a store. No other components should be negatively impacted, but any component that shows the "receipt required" default should be checked (in case this value is shown elsewhere, though it's likely just here). The backend remains unchanged; it's a front-end consistency fix.