

Session Reliability and Break-Cycle Work Report

#Overview

Today's work continued from the updated vision-gap plan, with the focus narrowed to the remaining timer and session-state gaps rather than broader feature expansion.

The first goal was to finish the last missing part of the focus-and-break loop by making the app distinguish between a short break and a long break in a way that matches an actual study cycle instead of using total historical session count.

After that, the scope shifted into reliability work because the remaining highest-risk issue was not missing UI, but the possibility that local active-session state and recorded session history could drift apart. That led to a review of how the dashboard, setup flow, task details screen, and timer screen each handled expired, cancelled, or replaced sessions.

Later in the same work session, the focus narrowed again into wording and flow polish on the timer screen. The break and sprint descriptions were rewritten so they better reflect the app's goal of supporting structured study behavior, and two runtime regressions reported after testing were fixed in the timer flow itself.

After that, the work shifted into the remaining first-time-user gap from the vision plan. The login and tab flows were tightened so incomplete users are routed into guided setup automatically, and the first guided sprint was changed into a short onboarding demo instead of dropping a new user straight into a normal 25-minute timer.

The final pass of the day was smaller, but still tied to the same product goal. The help modals on the dashboard and subjects screens were rewritten so they explain the focus-session and break rhythm in a more human way instead of sounding like a rigid step list.

After that, one final navigation-polish pass was added on the tabs layout itself. The bottom tabs were given explicit icons so the app's primary navigation reads faster at a glance and feels less unfinished.

After the branch work was merged back into 'main', one more cleanup pass was needed. That merge left multiple screen files in a syntactically broken state, so the final work shifted away from feature work and into restoring the current main branch to a usable state before report-focused delivery work.

#ImplementedFeatures

#LongBreakCycleCompletion

Finished the missing long-break part of the focus-and-break loop:

- extended the shared session defaults in 'lib/sessionDefaults.ts'
- added:
 - 'DEFAULT_LONG_BREAK_DURATION_MINUTES'
 - 'FOCUS_SESSIONS_PER_LONG_BREAK'
 - 'STUDY_CYCLE_IDLE_RESET_MINUTES'
- updated the timer flow so the next break is chosen from a local study-cycle model instead of total historical session count

The long-break rule now follows a small continuous-cycle interpretation of study flow rather than incorrectly counting unrelated sessions from earlier in the day or from previous days.

#StudyCycleState

Added a small local study-cycle state to support the new break behavior:

- extended 'lib/asyncStorage.ts'
- added 'StudyCycle'
- added:
 - 'SaveStudyCycle'
 - 'GetStudyCycle'
 - 'RemoveStudyCycle'
- tracked:
 - the task tied to the current cycle
 - how many focus sessions have been completed in that cycle
 - the last completed session type

- the last completion timestamp

This keeps the long-break offer tied to the current study run instead of to the user's entire database history.

#DynamicBreakPrompt

Updated the timer completion overlay so post-focus actions are based on the actual cycle state:

- expanded the post-session prompt model to include the next break type
 - replaced the hardcoded short-break action with a dynamic break action
 - updated the overlay text and button label so the user now sees:
 - 'Start short break'
 - or 'Start long break'
- depending on the current cycle

This completes the missing loop from the plan where break behavior should feel intentional rather than unfinished.

#SessionLifecycleConsistency

Introduced a shared finalization path for active sessions:

- added 'lib/sessionLifecycle.ts'
- introduced 'finalizeStoredSession(...)'
- moved repeated session-finalization behavior into one place:
 - remove local active-session state
 - clear study-cycle state when the final status is not 'completed'
 - finalize the same 'sessionId' in Supabase with:
 - 'completed'
 - 'cancelled'
 - 'expired'

This reduced the risk that one screen would only clear local storage while another screen properly finalized the database record.

#CrossScreenReliabilityFixes

Applied the shared finalization path across the screens that handle active-session recovery or replacement:

- 'app/task/timer.tsx'
- 'app/(tabs)/index.tsx'
- 'app/task/viewDetailsTask.tsx'
- 'app/setup.tsx'

The updated paths now explicitly finalize sessions when they are:

- expired on reopen
- expired while being observed from the dashboard
- cancelled because the user replaces one active sprint with another

This closes a real reliability gap where the app could previously lose the active local session while leaving the recorded session in the database unfinalized.

#TimerAndBreakCopyPolish

Rewrote the timer and break descriptions so they better match the product's intended tone:

- updated the pre-start sprint description
- updated the pre-start break description
- updated the focus fallback description on the running timer
- updated the post-focus and post-break explanation copy

The new wording emphasizes that structured focus and intentional breaks matter for studying, instead of sounding like placeholder or utility-only text.

#ReportedRegressionFixes

Fixed two runtime issues discovered during manual testing after the earlier session-cycle changes:

- after cancelling a focus session, the sprint-duration view could appear visually blank until the user manually dragged the picker

- after completing a long break, 'Continue with same task' could route the user back to the dashboard instead of returning to the correct task flow

The fixes were:

- reinitializing the picker-offset path when the timer returns to a non-running state
- preserving 'returnTaskId' inside the stored active-session shape so break sessions keep the correct task context all the way through completion

#OnboardingRoutingGuard

Closed the remaining onboarding-routing gap so incomplete users are pushed into guided setup instead of being left in the dashboard tabs:

- added 'lib/setupStatus.ts'
- moved the shared setup-completion rule into one place
- updated:
 - 'app/login.tsx'
 - 'app/(tabs)/_layout.tsx'
 - 'app/(tabs)/index.tsx'
 - 'app/(tabs)/subjects.tsx'
- setup completion is now checked from the same source in login, tab entry, dashboard, and subjects

This made the setup flow enforceable instead of depending on the user noticing the guided-setup card in the dashboard.

#FirstSprintDemoFlow

Adjusted the first guided sprint so the first-time experience better matches the low-friction vision goal:

- extended 'lib/asyncStorage.ts'
- added:
 - 'GetSetupSprintDemoUsed'
 - 'SaveSetupSprintDemoUsed'
- updated 'app/task/upsertTask.tsx' and 'app/setup.tsx' so the first setup sprint uses:
 - 'durationSeconds: '5''
 - 'onboardingDemo: 'true''
- updated 'app/task/timer.tsx' so that onboarding-demo sprint completion:
 - skips the normal session-complete modal
 - routes directly to the dashboard

This keeps the first sprint short enough to demonstrate the flow without locking a new user into a full focus block, while still falling back to the normal focus-session duration after the demo has been used once.

#HelpModalFlowCopy

Updated the help modals on the dashboard and subjects screens so they explain the intended study rhythm more naturally:

- updated:
 - 'app/(tabs)/index.tsx'
 - 'app/(tabs)/subjects.tsx'
- rewrote the flow-step descriptions so they feel less mechanical
- added clearer wording about the actual intended loop:
 - focus session
 - short pause
 - focus session again
 - longer pause after a few rounds

This better matches the app's tone and makes the focus/break cycle easier to understand from inside the product itself.

#TabBarIconPolish

Added explicit icons to the bottom-tab navigation so the two primary surfaces are easier to scan:

- updated 'app/(tabs)/_layout.tsx'
- reused the existing 'MaterialIcons' set already used elsewhere in the app
- assigned:
 - 'dashboard' to the dashboard tab

- 'menu-book' to the subjects tab

This was a small UI polish pass, but it improves immediate navigation clarity and makes the tab bar feel more intentional instead of placeholder-like.

#MainBranchMergeCleanup

Repaired merge-related breakage after switching back to 'main':

- fixed 'app/(tabs)/subjects.tsx'
- fixed 'app/subject/viewDetailsSubject.tsx'
- rebuilt 'app/task/viewDetailsTask.tsx' into a consistent working version

The merge had left these files with duplicated blocks, broken hook structure, and invalid JSX. The cleanup work focused on restoring the intended screen behavior rather than changing product scope.

The result was:

- subjects screen logic restored to a valid loading/setup/render flow
- subject details screen header and progress area reconstructed
- task details screen restored with working context, study activity, and sprint-start actions

This was not new feature work, but it was necessary delivery work because the main branch was no longer in a reliable edit/test state.

#ProblemsAndSetbacks

#SessionTruthDivergence

The main reliability issue uncovered today was not in the timer animation itself, but in how different screens treated expired or replaced sessions.

Several screens could detect that a stored session was no longer valid, but some of them only removed the local active-session entry instead of also finalizing the matching 'sprint_sessions' row in Supabase. That created a risk where the UI and the database could tell different stories about the same session.

The fix was to stop duplicating that logic screen by screen and route those paths through a shared finalization helper instead.

#PostChangeRuntimeRegressions

After the cycle and reliability changes landed, manual testing surfaced two smaller regressions in the timer screen:

- the duration screen could look empty after cancelling a focus session
- the break-return flow lost its task target after a long break

These were not architectural problems, but they were both important because they affected the user's immediate understanding of the timer flow after interacting with it.

#OnboardingFlowMismatch

Manual testing later uncovered a smaller flow mismatch inside guided setup:

- the first task created in setup could still open the timer with the normal 25-minute focus default
- returning to the guided-setup screen afterwards could then launch a different 5-second demo path

The problem was that task creation in setup and the setup screen itself were using two different timer-entry paths. The fix was to make those paths share the same one-time onboarding-demo rule.

#PostMergeCodeBreakage

The final setback of the day came after the branch merge itself rather than from the timer/session work.

Several files on 'main' were left in a partially merged state with duplicated code fragments and broken JSX structure. That meant the next pass could not start from feature verification alone, because basic app screens were no longer parseable.

The practical fix was to repair those files directly first, then re-run targeted verification on the restored app files before deciding whether any real feature regressions were still present.

#CurrentState

The timer and session model are now closer to the final intended behavior in the vision-gap plan.

The app now supports:

- a simple long-break rule tied to the current study cycle
- a local cycle model that avoids counting unrelated older sessions
- a post-focus overlay that correctly offers either a short break or a long break
- a shared session-finalization path used across timer, dashboard, setup, and task-details flows
- better consistency between active local session state and recorded session history
- more intentional sprint and break wording on the timer screen
- preserved task-return context across long-break completion
- corrected timer-screen recovery after cancelling a focus session
- automatic routing into guided setup for incomplete users after login and tab entry
- a one-time onboarding sprint demo that uses a 5-second timer
- direct dashboard routing after the onboarding demo completes, without the normal completion modal
- help modals that explain the study loop in a more natural way
- explicit tab icons that make dashboard and subjects easier to distinguish at a glance
- repaired 'main'-branch versions of the subjects, subject-details, and task-details screens after merge corruption

At this point, the timer/session work is closer to a finished loop, and the first-time-user path is more in line with the intended product vision. The biggest remaining work is now less about feature gaps and more about making sure the final report and final app behavior stay aligned.

#Verification

Static checks were run after the main implementation work and again after the regression fixes:

```
““text
npx tsc --noEmit
exited successfully
```

```
npm run lint
exited successfully
```

```
npx tsc --noEmit
exited successfully
```

```
npm run lint
exited successfully
```

```
npx tsc --noEmit
exited successfully
```

```
npm run lint
exited successfully
```

```
npx tsc --noEmit
exited successfully
```

```
npm run lint
exited successfully
““
```

Manual testing also confirmed most of the intended behavior from today's scope. Two regressions were found during that testing, both inside the timer flow, and both were fixed in the same work session:

- blank sprint-duration state after cancelling a focus session
- incorrect dashboard return after pressing 'Continue with same task' following a long break

Later manual testing also validated the guided-setup flow after the onboarding fixes:

- incomplete users were routed into guided setup instead of landing in dashboard tabs
- the first setup sprint used the intended 5-second demo timer
- after the demo finished, the user was sent directly to the dashboard without seeing the normal session-complete modal

The final UI pass for the day was lighter and did not change behavior, but the resulting tabs-layout diff was reviewed directly and confirmed to be limited to navigation presentation:

- explicit 'MaterialIcons' import in the tabs layout
- 'dashboard' icon for the dashboard tab
- 'menu-book' icon for the subjects tab

The final repair pass on 'main' was verified separately:

- 'npx eslint app/(tabs)/subjects.tsx app/subject/viewDetailsSubject.tsx app/task/viewDetailsTask.tsx'
- 'git diff --check'
- no remaining merge markers were found in 'app/', 'lib/', 'components/', or 'notes/'

One broader static check still failed afterwards:

- 'npx tsc --noEmit'

That remaining failure was no longer caused by the repaired app screens. The reported errors were instead in the test setup under '__tests__/', where Jest/testing-library types and modules were not currently configured in this branch.