PLANORAMA

Sprint 3 Manual Testing Report

Team 22: Sruthi Bhamidipati, Yana Sheoran, Vihaan Khajanchi, Jack Kenny

User Story #1

As a user, I would like to receive smart suggestions for breaking up long tasks into smaller subtasks.

Test cases:

1. Description: Verify that a suggestion prompt appears when a task matches a known phrase in the suggestion bank.
   Input: Create a new task titled "Write essay about World War II", save the task, then click "Edit" to reopen the task modal.
   Observed output: Suggested subtasks like "Research topic", "Create outline", and "Draft essay" appear in grey containers above the "+ Add Subtask" button.
   Acceptable output: yes no
2. Description: Verify that fallback subtasks are shown in the task modal when the task description is long and doesn't match the suggestion bank.
   Input: Create a new task titled "Submit report", and in the description field enter: "Finish the quarterly report, send the summary to stakeholders, and prepare slides". Save the task, then click "Edit" to reopen the task modal.
   Observed output: Fallback subtasks like "Finish the quarterly report", "send the summary to stakeholders", and "prepare slides" appear in grey containers above the "+ Add Subtask" button.
   Acceptable output: yes no
3. Description: Verify accepting a suggested subtask adds it as a normal editable subtask.
   Input: In the task modal, click the "Accept" button on the suggestion labeled "Research topic".
   Observed output: "Research topic" disappears from the suggestions list and appears in the main subtasks section above as a regular subtask (editable, completable).
   Acceptable output: yes no
4. Description: Verify editing a suggested subtask before accepting replaces it with the new name.
   Input: Click the "Edit" button on the suggestion "Outline structure", type "Write abstract" in the input field, then confirm.
   Observed output: The suggestion updates to "Write abstract" and is converted to a regular subtask in the main list.
   Acceptable output: yes no
5. Description: Verify rejecting a suggested subtask removes it permanently from the suggestion list.
   Input: Click the "Reject" button on the suggestion "Proofread and revise".
   Observed output: "Proofread and revise" disappears and does not show up in the subtask list.
   Acceptable output: yes no
6. Description: Ensure rejected suggestions do not reappear on reopening the task modal.
   Input: Reject "Write conclusion", close the task modal, then reopen it.
   Observed output: "Write conclusion" is still not present in either the grey suggestions or the regular subtasks.
   Acceptable output: yes no
7. Description: Verify accepted suggestions persist across sessions and reopen as normal subtasks.
   Input: Accept "Write introduction" and "Write body paragraphs" and close the task modal. Reload page or start a new session.
   Observed output: "Write introduction" and "Write body paragraphs" remain in the task table and task modal as regular subtasks.

Acceptable output: yes no

User Story #2

As a user, I would like to receive a weekly summary of my most and least productive days so that I can adjust my workflow.

Test cases:

1. Description: Verify that a chart displays productivity scores for each day (Sunday–Saturday) of the current week.
   Input: Complete tasks and log time on Monday, Wednesday, and Friday. Open the "Weekly Summary" page.
   Observed output: An area chart appears with 7 labeled data points (Sunday through Saturday). The chart's shape reflects the relative productivity score for each day.
   Acceptable output: yes no
2. Description: Verify that the most and least productive days are displayed in labels beneath the area chart.
   Input: Complete 2 tasks on Sunday, 6 tasks on Monday, 5 tasks on Tuesday, 3 tasks on Wednesday, 4 tasks on Thursday, 1 task on Friday, and 1 task on Saturday. Open the summary.
   Observed output: Labels under the chart read: "**Most productive:** Monday" and "**Least productive:** Friday, Saturday".
   Acceptable output: yes no
3. Description: Verify that task-only or time-only data still contributes to the area chart and productivity calculation.
   Input: Complete only tasks (no time) on Tuesday; log only time (no tasks) on Friday.
   Observed output: Area chart still shows visible scores for Tuesday and Friday based on available metrics. Both days are included in productivity comparison.
   Acceptable output: yes no
4. Description: Verify that the summary shows only partial week data when viewed midweek.
   Input: Open the "Weekly Summary" page on a Wednesday.
   Observed output: The area chart shows data points for Sunday through Wednesday. Thursday to Saturday are flat, but most and least productive labels reflect only completed days (Sunday through Wednesday).
   Acceptable output: yes no
5. Description: Verify that hovering or clicking on a day in the area chart displays a detailed productivity breakdown.
   Input: Hover over the data point for "Wednesday" in the Weekly Summary area chart.
   Observed output: A dark tooltip appears and includes: the day name at the top (e.g., Wednesday), a "Tasks completed" section listing completed task names or "No tasks completed", a "Time logged" section with a bulleted list of task names and time logged in the day or "No time logged", and a "Productivity Score" section with the total productivity score for the day.
   Acceptable output: yes no
6. Description: Verify that selecting a previous week from the dropdown updates the area chart and labels.
   Input: Use the dropdown to select "Week of Apr 13", assuming that the week has had activity logged.
   Observed output: Area chart will display 7 days of data for the week and most and least productive day labels will be populated accordingly underneath.
   Acceptable output: yes no
7. Description: Verify that a week with no tasks or time logs will display a message indicating as such.
   Input: Use the dropdown to select "Week of Apr 6", assuming that the week has not had activity logged.
   Observed output: Instead of an area chart, a message saying "No tasks/time logged for this week" underneath the dropdown.
   Acceptable output: yes no

User Story #3

As a user, I would like to receive notifications so that I can be reminded of important tasks and stay productive.

Test cases:

1. Description: Verify that valid notifications are shown on session start when notifications are enabled.
   Input: Enable notifications in settings. Create 3 tasks: one due today, one overdue, and one marked high priority. Refresh or re-login to start a new session.
   Observed output: Notification panel appears automatically with all 3 tasks listed. Notification badge (e.g., bell icon) is active.

Acceptable output: yes no

2. Description: Verify that dismissing a single notification hides it for the remainder of the session.
   Input: Start session with a task due today. Click the "x" icon next to that task in the notification panel.
   Observed output: The dismissed task disappears from the panel. It does not reappear until the next session, as long as it's still valid.
   Acceptable output: yes no
3. Description: Verify that clicking "Dismiss All" hides all notifications for the session.
   Input: Start a session with multiple valid task notifications. Click "Dismiss All" in the notification panel.
   Observed output: All notifications disappear immediately and do not return until a new session begins.
   Acceptable output: yes no
4. Description: Verify that disabling notifications in settings prevents them from appearing on session start.
   Input: In the settings page, toggle off task notifications. Start a new session.
   Observed output: Notification panel does not open, and no notification badge appears regardless of pending tasks.
   Acceptable output: yes no
5. Description: Verify that a notification appears dynamically if a task becomes due while the user is active.
   Input: With notifications enabled, stay logged in. Create a high-priority task (or one that is overdue or due today). Save the task.
   Observed output: A new notification appears automatically in the panel without requiring a page refresh.
   Acceptable output: yes no
6. Description: Verify that clicking a task in the notification panel highlights the task in the table.
   Input: Click a notification from the notification panel.
   Observed output: The corresponding task row in the table is visually highlighted via a black border. Highlight disappears after clicking elsewhere on the page.
   Acceptable output: yes no
7. Description: Verify that dismissed notifications reappear next session if still valid.
   Input: Dismiss a "due today" task notification during a session. Refresh the page or open a new session.
   Observed output: The same task reappears in the notification panel because it's still due today.
   Acceptable output: yes no
8. Description: Verify that a task removed from notification criteria no longer triggers notifications.
   Input: Mark a high-priority task as "Completed". Refresh session.
   Observed output: Task is no longer shown in the notification panel.
   Acceptable output: yes no

User Story #4

As a user, I would like to set an estimated time and an actual time for each of my tasks, and see the difference between the two.

Test cases:

1. Description:.Verify that the estimated time entered during task creation or editing is saved and displayed.
   Input: Create/edit a task and enter "90" into the "Estimated Time" field. Save the task and reopen the modal.
   Observed output: Estimated time of 90 minutes appears in the task details.
   Acceptable output: yes no
2. Description: Verify that actual time logged for a task is displayed alongside the estimated time.
   Input: Log 80 minutes for a task that has an estimated time of 90.
   Observed output: Both estimated (90 min) and actual time (80 min) are displayed in the task modal.
   Acceptable output: yes no
3. Description: Verify that the difference between estimated and actual time is displayed when both exist.
   Input: Estimated: 90 min, Actual: 80 min
   Observed output: A "10 min" difference is shown.

Acceptable output: yes no

4. Description: Verify that time difference is color-coded: green for under, red for over, gray for exact.
   Input: Estimated: 90, Actual: 80 → Green, Estimated: 60, Actual: 75 → Red, Estimated: 45, Actual: 45 → Gray
   Observed output: Color-coded appropriately next to the time difference display.
   Acceptable output: yes no
5. Description: Verify that tasks can be filtered/sorted by time performance (under/over/on time).
   Input: Use the filter dropdown to select "Under Time".
   Observed output: Only tasks completed in less time than estimated are shown.
   Acceptable output: yes no
6. Description: Verify that entering non-numeric characters in "Estimated Time" shows an error.
   Input: **:** Enter "–" in the Estimated Time field and try to save.
   Observed output: Error message appears: "Please enter a valid number."
   Acceptable output: yes no
7. Description:Verify that leaving the estimated time field empty shows an error.
   Input: Leave the Estimated Time field blank and try to save.
   Observed output: Error message appears: "Estimated time is required."
   Acceptable output: yes no

User Story #5

As a user, I would like to compare my productivity across different weeks/months to track my long-term progress.

Test cases:

1. Description: Verify that the current week's total logged time is displayed.
   Input: **:** Log tasks across multiple days in the current week in the "Productivity Tracker" section.
   Observed output: A total (e.g., "Total: 320 minutes") appears for the current week.
   Acceptable output: yes no
2. Description: Verify that the system shows a comparison between two selected weeks.
   Input: Log tasks in Week A and Week B. In the Productivity Tracker, select both weeks for comparison.
   Observed output: Display shows the total time for each week with a graph comparison
   Acceptable output: yes no
3. Description: Verify that weekly productivity is shown using a clear visual format.
   Input: Verify that weekly productivity is shown using a clear visual format.
   Observed output: Weekly totals are represented in a bar chart
   Acceptable output: yes no
4. Description: Verify that productivity increases are highlighted in green.
   Input: Log more time this week than the previous. View the current week in the Productivity Tracker.
   Observed output: A green bar appears with an icon of how much % is up
   Acceptable output: yes no
5. Description: Verify that previous weekly logged time is displayed in a historical view.
   Input: Navigate to the "History" tab or section of the Productivity Tracker.
   Observed output: A timeline or list shows weekly logged times.
   Acceptable output: yes no
6. Description: Verify that a message is shown when no task data is available for the current or past week.
   Input: Ensure no tasks are logged for the current week in the Productivity Tracker.
   Observed output: Message shown: "No task data available for this week. Please make sure tasks are logged."
   Acceptable output: yes no

User Story #6

As a user, I would like to be given gentle reminders through pop-ups or alerts suggesting I take breaks after long periods of continuous work.

Test cases:

1. Description: Verify that a break reminder appears when the user reaches the work threshold.
   Input: Log tasks continuously until total reaches 2 hours without taking a break.
   Observed output: A modal or pop-up appears saying "Time for a break!".
   Acceptable output: yes no
2. Description: Verify no reminder is shown if the threshold hasn't been reached.
   Input: Log tasks totaling less than 2 hours.
   Observed output: No pop-up or alert appears.
   Acceptable output: yes no
3. Description: Verify that clicking "Accept" dismisses the reminder.
   Input: When the break reminder appears, click "Accept".
   Observed output: The modal disappears and no further reminders are shown until the next cycle.
   Acceptable output: yes no
4. Description: Verify that accepting the break resets the continuous work timer.
   Input: Click "Accept" on a break reminder, then log new time after acceptable time has passed
   Observed output: System starts counting work time from 0 again for the new session.
   Acceptable output: yes no
5. Description: Verify that clicking "Snooze" delays the break reminder.
   Input: Click "Snooze" when the break reminder appears.
   Observed output: Modal disappears and reappears after 5 minutes.
   Acceptable output: yes no
6. Description: Verify that the break threshold can be customized by the user.
   Input: Set the break threshold to 90 minutes in reminder settings and log 90 minutes of task time.
   Observed output: A break reminder appears after 90 minutes instead of 120.
   Acceptable output: yes no
7. Description: Verify that snoozing is disabled after 3 times.
   Input: Click "Snooze" on the break reminder three times in a row.
   Observed output: On 3rd time, "Snooze" option is disabled or unavailable. User must "Accept".
   Acceptable output: yes no
8. Description: Verify that task logging is disabled during a break.
   Input: Verify that task logging is disabled during a forced break.
   Observed output: Alert appears: "Break in progress. Please wait X minutes." and the log is not saved.
   Acceptable output: yes no

User Story #7

As a team member, I would like to be able to add tasks.

Test cases:

1. Description: When creating a task, it should not let you submit if the date is not filled in.
   Input: Log in using username "admin", password "pass", go to teams, click "Create Team", make a team and click on it. Click "Create Task", fill in any name. Leave the date empty.
   Observed output: It does not submit, and shows a message saying all fields need to be filled in.
   Acceptable output: yes no
2. Description: When creating a task, it should not let you submit if the name is not filled in.
   Input: Click "Create Task", fill in a date by clicking "today". Leave the name empty.

Observed output: It does not submit, and shows a message saying all fields need to be filled in.
Acceptable output: yes no

3. Description: When creating a task, it should not let you submit if both fields are not filled in.
   Input: Click "Create Task", leave both fields empty.
   Observed output: It does not submit, and shows a message saying all fields need to be filled in.
   Acceptable output: yes no
4. Description: If there are no tasks, it should say "No tasks".
   Input: Return to the team page if you are not already there.
   Observed output: It says "No tasks" in the task table.
   Acceptable output: yes no
5. Description: If all fields are valid, creating a task should populate on the table with the correct information.
   Input: Click create task, enter any name and a date today onwards.
   Observed output: The task shows up in the table with the name and date I entered.
   Acceptable output: yes no
6. Description: The team leader can delete a task.
   Input: Add a new task and click delete on it. Make sure you still have a task left for the next test.
   Observed output: The task we deleted disappears.
   Acceptable output: yes no
7. Description: Verify that added tasks show up for every member, members can add tasks, and members cannot delete tasks.
   Input: Click "add member", enter "a" in the search bar, invite admin 1, 2, and 3. Click log out in the top right, log in using "admin[number]" & "pass", go to the teams page, click "Join" on the invite in the bottom right, click on the team. Try creating a task and check the options you see for the tasks. STAY on a member account for the next test.
   Observed output: Members can each add a task, which shows up for everyone. They do not see the task that was deleted. Also, they do not see the delete option.
   Acceptable output: yes no

User Story #8

As a team member, I would like to be able to claim tasks.

Test cases:

1. Description: Given a team member clicks the claim button, then their name should show up in the assignee section.
   Input: Click "claim" on any of the tasks you made.
   Observed output: The user's name shows up in the "Assigned to" section
   Acceptable output: yes no
2. Description: Given a team member clicks remove for themselves, their name should be removed.
   Input: Click "unclaim" on the task you just claimed.
   Observed output: The task shows "Unassigned" again.
   Acceptable output: yes no
3. Description: Given a task already has an assignee, then users should not be able to add themselves.
   Input: Click "claim" on any task, then log out and go to another member's account. Enter the team again and see if you can claim that task.
   Observed output: No claim or unclaim options appear.
   Acceptable output: yes no
4. Description: Given a task is assigned to a different user, then the current user should not have the option to remove them unless they are the owner.
   Input: From test 3 you already know you can't remove it as another member, so log out and go to the team leader's account. Click "Unassign" on the task that is claimed by a member. STAY on this page for the next test.

Observed output: Assignee becomes “Unassigned”.
Acceptable output: yes no

User Story #9

As a team leader, I would like to be able to assign tasks.

Test cases:

1. Description: Given the user is the owner of the team, they should have a different option than everyone else when assigning. They should have a list to choose any member.
   Input: On the task you just unassigned, click “Assign”. Click “choose” on one of the members that shows up.
   Observed output: The selected user shows up in the assignee section.
   Acceptable output: yes no
2. Description: Given a task already has an assignee, the team leader cannot add another unless they remove it first.
   Input: Click the “unassign” button. Then click “assign”, and assign someone again.
   Observed output: When a task is assigned, you only see the unassign button. When you click it, the assign button shows up again, and it successfully assigns a user again.
   Acceptable output: yes no
3. Description: Given the team owner selects a member to assign, their name should show up for everyone as the assignee.
   Input: Assign a task to a member. Log out and log back in as a member. Return to this team’s page. You can stay here for the next test.
   Observed output: The assignment made by the owner is visible to other members.
   Acceptable output: yes no

User Story #10

As a team member, I would like to mark team tasks as completed.

Test cases:

1. Description: Given the user clicks an expand button, the completed section should show its contents and clicking it again will close the section.
   Input: Click on the arrow next to “Completed”. Click it a few times.
   Observed output: The button shows and hides the table.
   Acceptable output: yes no
2. Description: Given the user clicks the button to mark as completed, the task should disappear from the main table and move to a “completed” section.
   Input: Create a couple tasks. Mark one as completed while the “completed” table is open and one when it is closed. Open it up again.
   Observed output: Both times the task was removed from the main table and populated in the completed table.
   Acceptable output: yes no
3. Description: Given the user was an assignee, then completing the task will show they were the one who completed it.
   Input: Create a task and click “Assign”. Then, mark it as completed. Create another task and click “Assign”, then switch to another member’s account, return to the same team page and mark as complete from that account.
   Observed output: Both times the correct user shows up under “Completed by”.
   Acceptable output: yes no
4. Description: Given a task does not have an assignee, then completing the task will credit the person who clicked complete.
   Input: Create a task. Click the check mark. Then go to another account and repeat this.
   Observed output: The person who clicked the check mark shows up under “Completed by”.
   Acceptable output: yes no
5. Description: Given the user undoes a completion, the task should show back up in the team task table.

Input: For some of the tasks you marked as complete, click the undo button under “Mark incomplete”
Observed output: The task shows up in the main table, keeping the name and deadline but becoming unassigned.
Acceptable output: yes no

User Story #11

As a team member, I would like to leave comments on tasks so that I can communicate updates.

Test cases:

1. Description: Given the user clicks the comment button the comment UI with its features should appear by it.
   Input: The user clicks the comment button. Then observe what happens.
   Observed output: A comment UI to type your comment and save it appears.
   Acceptable output: yes no
2. Description: Given that the user clicks the save comment button, the comment should be put into a list of all the comments for that task.
   Input: The user clicks the comment button, then types in the comment “hello” into the input box, then presses the save button.
   Observed output: The comment in the input box disappears and the comment “hello” appears within the comment list.
   Acceptable output: yes no
3. Description: Given that the comment is successfully moved into the comment list, it should say who on the team made the comment.
   Input: The user creates a comment, then opens the comment list, then finds their comment.
   Observed output: Their comment has their display name listed below the comment.
   Acceptable output: yes no
4. Description: Given that the user clicks the comment list button, the entire list of comments will be shown to the user.
   Input: The user makes several comments themselves and then clicks the comment list button.
   Observed output: The comment list UI appears and a scrollable menu is shown with a list of all the comments and who made them.
   Acceptable output: yes no
5. Description: Given that the user made a comment, they should be able to delete the comment from the list.
   Input: The user opens the comment list, then they find one of the comments they’ve made and click the delete button underneath it.
   Observed output: The comment stays in the comment list.
   Acceptable output: yes no
6. Description: Given that there are multiple members in a team, when multiple users make a comment on one task, all the users should be able to see what the comment is and who made it.
   Input: One user named “admin” types the comment “hello” onto task 1. Then another user named “guest” types the comment “bye” onto task 1. Both admin and guest then click the comment list to see what comments are in the list.
   Observed output: When admin opens the comment list they see both comments and who made them. When guest opens the comment list they see both comments and who made them.
   Acceptable output: yes no
7. Description: Given that a user logs out or shuts down their computer, the data should persist across sessions.
   Input: The user refreshes the page on their browser.
   Observed output: The comments stay within the comment list along with who made them.
   Acceptable output: yes no

User Story #12

As a user, I would like to be able to roll over incomplete tasks to the following day, either automatically or manually.

Test cases:

1. Description: Given that a task is past deadline, it should be verified that the task has a visual indicator that it is overdue and can be "rolled over"
   Input: The user waits a day or puts the deadline of a task behind the current day.
   Observed output: The task's color is changed to a unique red color exclusive to the roll over feature and a roll over button appears on the task.
   Acceptable output: yes no
2. Description: Verify that the roll over button will set the deadline from the past due date to a day in front of the current day.
   Input: The user clicks the roll over button on 4/25/2025 for a task that has a deadline of 4/04/2025.
   Observed output: The task's deadline is changed from 4/04 to 4/26 and the red color is changed back to its original color.
   Acceptable output: yes no
3. Description: Verify that if the user rolls over a task multiple times, a counter will keep track of how many times that task has been rolled over.
   Input: The user will take a task that has a deadline of 4/04/2025 and roll over the task to 4/26/2025. The user will then change the deadline back to 4/04/2025 and click the roll over button again.
   Observed output: The counter in the roll over column goes from 0 to 1 to 2.
   Acceptable output: yes no
4. Description: Verify that the auto roll over check mark turns on the auto rollover and has the time to reset appear.
   Input: The user clicks the auto roll over check mark.
   Observed output: The check mark has a time input that the user can set to be what time they want the auto reset to happen.
   Acceptable output: yes no
5. Description: Verify that the time input into the auto roll over time will roll over the tasks that are overdue at the time specified.
   Input: The user puts in the time 1:02pm when it is 1:01pm. The user then waits until it is 1:02pm.
   Observed output: The roll over happens and the roll over count is increased by 1 and the task is no longer overdue.
   Acceptable output: yes no
6. Description: Verify that the roll over count and deadline persist across different sessions after a user has rolled over the task.
   Input: The user rolls over tasks 1 and 3 and then refreshes the page.
   Observed output: Tasks 1 and 3 remain rolled over and their number of roll over persists as well.
   Acceptable output: yes no
7. Description: Verify that the automatic rollover persists across sessions so the user does not need to keep inputting the time.
   Input: The user checks the auto roll over checkmark and then inputs the time "12:00pm" and then refreshes the page
   Observed output: The checkmark is unchecked and the auto roll over time is reset.
   Acceptable output: yes no

User Story #13

As a user, I would like to be able to change my display name for different teams to best suit each team.

Test cases:

1. Description: Verify that if a user is in a team, a change display name button will appear next to their username.
   Input: The user clicks the teams page, creates a team, clicks into the team, and checks next to their name.
   Observed output: The change display name button appears next to their name.
   Acceptable output: yes no
2. Description: Verify that when the user clicks the change display name button the change display UI will appear that will allow them to change their name.
   Input: The user clicks the change display name next to their name within the team "t".
   Observed output: An input box with the button "save" and "reset" below it appears.

Acceptable output: yes no

3. Description: Verify that if the user clicks the save button with text within the input box, the user's display name will be changed to their input.
   Input: The user clicks the change display name button, then inputs the name "new name". They then click the save button.
   Observed output: The user's normal name of "admin" changes to "new name" within the team user list.
   Acceptable output: yes no
4. Description: Verify that when the user's display name has changed, all other things within the team that use their name also change to the new display name.
   Input: The user "admin" changes their display name to "new name". The user then assigns themselves a task and creates a comment. The user then clicks the comment list button.
   Observed output: The assigned name has been also changed to "new name". All the comments within the comment list made by the user are now under "new name".
   Acceptable output: yes no
5. Description: Verify that if there are multiple users within a team, that all the members show the new display name as provided by the user.
   Input: Two users "admin" and "guest" change their names to "name 1" and "name 2" respectively.
   Observed output: On both ends the names appear as they put them in. On admin's end, guest is now "name 2" and on guest's end, admin is now "name 1".
   Acceptable output: yes no
6. Description: Verify that the reset button properly resets the user's name back to their account name.
   Input: The user changes their display name from "admin" to "new name". The user then goes into the change display name menu and clicks "reset"
   Observed output: The user's display name "new name" is reverted back to their original account name "admin".
   Acceptable output: yes no
7. Description: Verify that the display names persist across sessions.
   Input: The user changes their display name from "admin" to "new name" and then refreshes the page.
   Observed output: The display name remains as "new name" and does not reset to "admin" unless the user does it themselves.
   Acceptable output: yes no

Sprint 2: Incomplete User Stories, Tasks, & Acceptance Criteria

User Story #4

As a user, I would like to be able to delete tasks.

Test cases:

1. Description: Verify that when the user deletes a task they are given a time frame(around 10 seconds) to undo the task before it is moved into the trash bin.
   Input: The user clicks the delete button on the task they want deleted.
   Observed output: There is no timer to countdown, the undo button just remains there until they decide.
   Acceptable output: yes no
2. Description: Verify that the tasks within the trash bin and the deleted tasks persist across sessions.
   Input: The user deletes 4 tasks and then refreshes the page.
   Observed output: The tasks are no longer in the trash bin but are in the main task menu.
   Acceptable output: yes no
3. Description: Verify that when the user deletes more than 10 tasks the oldest ones will be removed from the trash bin and replaced with the newer tasks.

Input: The user deletes 12 tasks and clicks the trash bin.
Observed output: The first 2 tasks deleted are no long in the trash bin, only the latest 10.
Acceptable output: yes no

User Story #4

As a user, I would like to archive completed tasks so that I can keep my workspace clutter-free.

Test cases:

1. Description: Verify that the mark as complete check mark visually displays the task as having been complete.
   Input: The user clicks the check mark for the desired task.
   Observed output: The task's color becomes a distinct green that is exclusive to the completed tasks and an archive button appears underneath it.
   Acceptable output: yes no
2. Description: Verify that the archive button only appears for completed tasks and puts tasks into the archived section when clicked.
   Input: The user clicks the mark as a complete check mark and sees that an archive button appears. The user then clicks that archive button.
   Observed output: The task disappears from the main task page and then is put into the archived section for retrieval if needed.
   Acceptable output: yes no
3. Description: Verify that the archived section is accessible and the tasks within it are retrievable.
   Input: The user clicks the archived tasks button in the bottom left corner and a UI appears with all the tasks that have been previously archived. The user then clicks the checkmark next to the top task.
   Observed output: The task that the user clicked is put back into the main task page as a completed task and removed from the archived tasks section.
   Acceptable output: yes no
4. Description: Verify that the auto archival time is functioning and sets tasks into the archived section at the specified time.
   Input: The user marks the top three tasks as complete and then goes into the archival section. The user then checks the check mark to enable the auto archival and sets the time to daily and 12:04 pm(the users time is 12:03pm). The user then waits until 12:04pm has come and gone.
   Observed output: After 12:04pm comes and goes, nothing happens and the tasks do not go into the archival section.
   Acceptable output: yes no
5. Description: Verify that the tasks marked as complete or the tasks in the archived section persist across multiple sessions.
   Input: The user marks 10 tasks as complete and archives 5 of those tasks. Then the user refreshes the page.
   Observed output: All 10 tasks go back to being on the main page as non-complete.
   Acceptable output: yes no

User Story #5

As a user, I would like to jot notes down about tasks that I have or have not completed so that I can reference thoughts for later.

Test cases:

1. Description: Verify that the notes button appears on every task and allows the user to type in notes individually into tasks
   Input: The user creates two tasks. Then they open each one and type "test 1" and "test 2" respectively.
   Observed output: The two notes do not interfere with each other and remain separate.
   Acceptable output: yes no
2. Description: Verify that the bold option works when the user highlights text and selects the bold button.

Input: The user selects the first task and opens its notes. Then the user types “test” into the bottom section, highlights the text in the top section, and clicks bold

Observed output: The text in the top section is bolded as intended.

Acceptable output: yes no

3. Description: Verify that the italics option works when the user highlights text and selects the italics button.
   Input: The user selects the first task and opens its notes. Then the user types “test” into the bottom section, highlights the text in the top section, and clicks italics.
   Observed output: The text in the top section is italicized as intended.
   Acceptable output: yes no
4. Description: Verify that the bullet option works when the user highlights text and selects the bullet button..
   Input: The user selects the first task and opens its notes. Then the user types “test” into the bottom section, highlights the text in the top section, and clicks the bullet button.
   Observed output: The text in the top section is bulleted as intended.
   Acceptable output: yes no
5. Description: Verify that the delete notes button works properly and deletes the contents of the note when clicked.
   Input: The user opens the notes and attempts to click the delete button.
   Observed output: The user sees that the delete button does not exist so they are not able to successfully delete the tasks.
   Acceptable output: yes no
6. Description: Verify that the notes persist across sessions.
   Input: The user creates a note in task 1 and in task 2 and then refreshes the page.
   Observed output: The notes in both tasks are no longer there.
   Acceptable output: yes no

User Story #6

As a user, I would like to set default task templates so that I can quickly create repetitive tasks.

Test cases:

1. Description: Verify that the task template section exists and that all of the features of the template are within it.
   Input: The user clicks the “Templates” button on the bottom left of the screen.
   Observed output: A UI appears with different options to create or add task templates.
   Acceptable output: yes no
2. Description: Verify that a default task template can be created using the create task template button.
   Input: The user clicks the task template button, inputs the information for the task, and then clicks save.
   Observed output: Once the template has been saved the user is shown a list of all their task templates where different options for each one present themselves.
   Acceptable output: yes no
3. Description: Verify that once the user clicks “create task” for a task template that task is created with the same specifications as the template.
   Input: The user creates a task template with the information they want and then they press the “create task” button underneath the template they just created.
   Observed output: They are prompted with a message saying that it was successful and the task is then put into the main task window.
   Acceptable output: yes no
4. Description: Verify that multiple task templates work and create the tasks they were defaulted to individually.
   Input: The user creates two different task templates and clicks each one two times.
   Observed output: Each of the tasks they are templated for appear in the main task interface twice.
   Acceptable output: yes no

5. Description: Verify that template editing works and the user is able to change the specification of the template they have made.
   Input: The user clicks the edit button beneath a template and changes the date from “05/01/2025” to “05/05/02025” and then clicks the “update template” button.
   Observed output: The template’s date is updated to the new date specified.
   Acceptable output: yes no
6. Description: Verify that the delete template button works as intended and allows the user to delete unwanted templates.
   Input: The user clicks the delete button underneath the desired template and a window appears asking if they are sure they want to delete the template. The user then selects ok.
   Observed output: The task template is removed from the list of templates.
   Acceptable output: yes no
7. Description: Verify that the favorite feature is present and working for tasks so the user can keep their favorite templates at the top of the list.
   Input: The user creates 5 templates and tries to favorite the one they want at the top.
   Observed output: There is no favorite button so the user is unable to accomplish this goal.
   Acceptable output: yes no
8. Description: Verify that the most recent template button is working so that the user can quickly create another task from that template they just used.
   Input: The user opens the templates menu and creates a task from one of their templates. The user then attempts to click the most recent button and have that task be created again from the template.
   Observed output: There is no most recent button so the user is unable to accomplish this goal.
   Acceptable output: yes no
9. Description: Verify that templates and tasks created from templates persist across sessions.
   Input: The user creates 3 templates and chooses the first template to create 3 more tasks with. The user then refreshes the page.
   Observed output: After the refresh the 3 templates and the 3 tasks created with one of them are gone.
   Acceptable output: yes no