PLANORAMA

Sprint 1 Planning Document

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

Sprint Overview

This sprint will focus on implementing core task management functionalities, including task creation, editing, deletion, and organization features. The team will develop a user-friendly task page where users can view, update, and categorize their tasks efficiently. By incorporating features like drag-and-drop reordering, filtering, and task prioritization, the sprint aims to enhance usability and create a more dynamic and interactive task management experience. These features will ensure that users can not only manage their to-do lists effectively but also customize their workspace to suit their needs.

In addition to task management, this sprint will introduce profile management capabilities, allowing users to personalize their accounts by uploading profile pictures, managing achievements, and adjusting settings such as font size, spacing, and color themes. The settings menu will provide further customization options, including task view preferences and accessibility features. The implementation of account-related features, such as user authentication, login, and deletion, will also be prioritized to establish a secure and seamless user experience.

Key efforts will be directed toward ensuring data persistence and synchronization across sessions, allowing users to maintain their preferences and task data even after logging out. A significant focus will be on implementing robust UI interactions that provide real-time feedback, ensuring smooth task reordering and modifications. Debugging functionalities through unit testing will play a crucial role in identifying and resolving potential issues early in development. The project scope balances frontend UI development with backend logic, ensuring an intuitive experience while maintaining reliable data handling. Challenges such as real-time updates, database consistency, and intuitive design elements will be addressed through structured testing and iterative development, ensuring a stable and well-functioning application by the end of the sprint.

Scrum Master: Sruthi Bhamidipati

Meeting Plan: T/Th 3:30-4:00 pm

Risks and Challenges:

- Data Persistence & Synchronization: Ensuring that user actions persist after logout and across multiple sessions may require careful database handling.
- UI/UX Complexity: Features like drag-and-drop, color coding, and real-time status updates must be visually clear and responsive to avoid confusing interactions.
- Undo & Archival Functionality: Implementing an efficient undo system for task deletions and ensuring seamless archiving of completed tasks without performance issues.
- Task Categorization & Filtering: Developing a flexible filtering system that allows multiple simultaneous criteria (e.g., priority, tags, color) without slowing down performance.
- Security & Account Management: Handling user authentication, ensuring password security, and properly deleting user data upon account removal while maintaining database integrity.
- Testing & Debugging: With multiple interdependent features, extensive unit testing will be necessary to ensure smooth performance and prevent UI or data-related errors.
- Time Constraints: Given the scope of the sprint, balancing feature completion with debugging and optimization may pose a challenge, requiring prioritization of core functionalities.

Current Sprint Detail

User Story #1

As a user, I would like a task page so I can view all my tasks.

#	Description	Estimated Time	Owner
1	Create a background that will hold the tasks.	1 Hr	Vihaan
2	Tasks that are created or removed will show up or disappear from this page, and be updated when edited.	3 Hrs	Vihaan
3	Each task should have sections for its description, priority, and deadline.	2 Hrs	Vihaan
4	Should be buttons that will be used to complete, edit, delete, or create tasks (just UI, not functionality).	1 Hr	Vihaan
5	Debug and test with unit tests.	2 Hrs	Vihaan

Acceptance Criteria:

- Given the user is on the task page, when there are existing tasks, then the user should see a list of tasks, including their descriptions, priorities, and deadlines.
- Given the user has no tasks, when they access the task page, then they should see a message stating "No tasks available."
- Given a task is created elsewhere in the system, when the user is on the task page, then the task list should automatically update to include the new task.
- Given a task is edited elsewhere in the system, when the user is on the task page, then the task list should automatically reflect the changes made to the task.
- Given a task is deleted elsewhere in the system, when the user is on the task page, then the task list should automatically remove the deleted task.
- Given the task page loads, when the UI is displayed, then there should be clear section labels for "Task Description," "Priority," and "Deadline."
- Given the user is viewing their task list, when tasks are present, then each task should be displayed in a structured format (e.g., a table or card layout).
- Given the task list is empty, when the user views the page, then a message should indicate that no tasks are available instead of an empty screen.

User Story #2

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

#	Description	Estimated Time	Owner
1	Make UI for the main task board that allows for easy task creation.	4 Hrs	Sruthi
2	Implement task creation logic and ensure tasks can be added dynamically.	3 Hrs	Sruthi
3	Ensure task UI design is intuitive and responsive.	2 hrs	Sruthi

4	Debug and test task creation functionality with unit tests.	2 Hrs	Sruthi

Acceptance Criteria:

- Given the user is on the task page, when they press the "Create Task" button, then a modal or form should appear with input fields for task details.
- Given that the information fields are implemented correctly, when the user goes to fill them out, then the user should be able to fill out theTask name (required), Description (optional), Due time (required), Priority level (e.g., Low, Medium, High), Color tag (optional), Status (e.g., To-Do, In Progress, Completed)
- Given the user fills out all required fields, when they submit the form, then the task should be: Saved in the system, Displayed immediately in the task list without requiring a page refresh.
- Given that the task creation UI appears, when a user begins to update the new tasks information, then the user will be able to scroll down to reach fields that are below the UI box.
- Given that the information fields are implemented correctly, if the user does not fill out at least the name and time, then the user will get an error message prompting them to fill out those two information fields.
- Given a user tries to create a task with the same name and due time as an existing task, then they should see a warning message informing them of the duplicate but still be allowed to proceed.
- Given the task creation UI is open, when the user presses "Cancel" or clicks outside the form, then the form should close without saving any data.

User Story #3

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

#	Description	Estimated Time	Owner
1	Implement task editing functionality, allowing users to update task details.	4 Hrs	Sruthi
2	Ensure tasks within the database update once edits are submitted.	1 Hrs	Sruthi
3	Debug and test task editing and organization features with unit tests.	2 Hrs	Sruthi

Acceptance Criteria:

- Given the editing functionality is implemented, when a user selects a task to edit, then a window should appear with the task's existing details pre-filled in the corresponding fields (e.g., title, description, priority, time, tags).
- Given a user is editing a task, when they submit the edited task, then the updated task should immediately reflect in the task list on the same page.
- Given a user edits only one aspect (e.g., title, priority, description) of a task, when they submit the changes, then only the updated field should change, while all other fields remain the same.
- Given task edits are saved correctly, when a user logs out and logs back in, then the edited task should retain the updated details.
- Given a user edits a task to match another task's details (e.g., title or description), when they save the edited task, then both tasks should be allowed to coexist as long as they are distinct (i.e., not triggering a duplicate error unless unique fields are affected).
- Given the user attempts to submit an empty required field (e.g., missing title or description), when they attempt to save the task, then they should receive a prompt/error message indicating that the required field is missing.

- Given a user attempts to input invalid data into any field (e.g., entering letters in a time field), when they attempt to save the task, then they should receive an error message prompting them to correct the invalid input.
- Given a user is editing a task, when they edit a field (e.g., priority or tags), then the field's value should be validated to ensure it conforms to the allowable input format (e.g., valid priority levels, tags).
- Given a user is editing a task, when they make changes (e.g., select new priority or update a description), then real-time visual feedback should confirm that the changes are being applied (e.g., field highlights, save button activation).

User Story #4

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

#	Description	Estimated Time	Owner
1	Implement task deletion functionality, allowing users to remove tasks.	4 Hrs	Sruthi
2	Ensure task data is stored in a database and persists across sessions.	3 Hrs	Sruthi
3	Implement undo functionality for accidental deletions.	2 Hrs	Jack
4	Debug and test task deletion and persistence with unit tests.	1 Hr	Jack

Acceptance Criteria:

- Given the user is on the task page, when they click the "Delete" button on a task, then a confirmation prompt should appear asking them to confirm the deletion.
- Given the user confirms the deletion, when they proceed, then the task should be immediately removed from the task list in the UI.
- Given the user deletes a task, when they log out and log back in, then the deleted task should not reappear, ensuring that deletions persist in the database.
- Given the user deletes a task, when they see the confirmation message, then an "Undo" button should be available for them to restore the task.
- Given the user clicks "Undo" within a set time frame (e.g., 10 seconds), then the task should reappear in its original state in the task list.
- Given the undo time frame expires, when the user tries to restore the task, then they should no longer be able to undo the deletion.
- Given the user deletes multiple tasks, when they access the undo function, then only the last 10 deleted tasks should be recoverable.
- Given the user deletes more than 10 tasks, when they try to undo, then older deleted tasks (beyond the last 10) should no longer be available for recovery.

User Story #5

As a user, I would like to categorize tasks by priority, tags, or deadlines so that I can make decisions on which order to take the tasks.

#	Description	Estimated Time	Owner
1	Implement a tagging system for tasks, allowing users to	4 Hrs	Jack

	assign priorities and deadlines.		
2	Add filtering and sorting options to the UI to allow users to view tasks by category.	4 Hrs	Jack
3	Ensure task properties persist correctly in the database.	4 Hrs	Jack
4	Debug and test with unit tests.	2 Hrs	Jack

Acceptance Criteria:

- Given that the task categorization system is implemented correctly, when a user assigns a priority or tag, then it should be reflected in the UI.
- Given that filtering and sorting options exist, when a user applies a filter, then only the matching tasks should be displayed.
- Given that task categorization persists in the database, when a user refreshes the page or logs back in, then all assigned priorities, tags, and deadlines should remain as they were.
- Given that multiple filters can be applied, when a user selects more than one filter (e.g., high priority and due this week), then only tasks matching both criteria should be shown.
- Given that task properties are editable, when a user modifies a task's priority, tag, or deadline, then the changes should be immediately reflected in the UI and saved in the database.

User Story #6

As a user, I would like to color code tasks based on categories so that I can quickly identify them.

#	Description	Estimated Time	Owner
1	Add button to task UI that shows a popup to choose task color.	1 Hr	Sruthi
2	Add UI for choosing the color of a task from a predefined set of options.	1 Hr	Sruthi
3	Have task information be updated accordingly and saved to the database.	2 Hrs	Sruthi
4	Allow filtering of tasks based on their color.	2 Hrs	Sruthi
5	Debugging and unit tests.	1 Hrs	Sruthi

Acceptance Criteria:

- Given that the "Choose Color" button is added, when the user presses the button, then a popup should appear, allowing the user to choose a color for the task from a predefined set of options.
- Given that the user selects a color, when they view the task on their task list, then the task should be displayed in the selected color.
- Given that filtering by color is enabled, when the user selects a specific color filter, then only tasks of that color should appear.
- Given that no tasks have the selected color, when the user filters by that color, then a message should appear stating, "No tasks match this color filter."

User Story #7

As a user, I would like to set different statuses for tasks (in-progress, blocked, waiting for review).

#	Description	Estimated Time	Owner
1	Add a button to task UI to allow the user to change the status of the task.	2 Hr	Sruthi
2	Add UI for setting/changing status of a task.	3 Hrs	Jack
3	Have task information be updated accordingly and the information saved to the database.	2 Hrs	Jack
4	Debugging and unit tests.	2 Hrs	Jack

Acceptance Criteria:

- Given the user is on the task board page, when they click the "Set Status" button on a task, then a popup should appear displaying status options (e.g., In Progress, Blocked, Waiting for Review).
- Given the user selects a new status from the popup, when they confirm the change, then the new status should immediately appear on the task (e.g., status label updates, color change).
- Given a user updates a task status, when they refresh the page or log out and log back in, then the task should retain the last selected status.
- Given a task has been deleted in the current session, when the user attempts to change its status, then the action should be blocked, and an error message should notify them that the task no longer exists.
- Given a user deletes a task and then logs back in, when they try to update its status, then the task should not be available for updates.
- Given the status selection popup is open, when the user tries to save without selecting a valid status, then an error message should prompt them to pick an option before proceeding.

User Story #8

As a user, I would like to be able to filter tasks by completion status.

#	Description	Estimated Time	Owner
1	Implement the filter button above tasks.	1 Hr	Jack
2	Save filter status even after the app has been closed or the user has logged out.	1 Hrs	Jack
3	Implement sorting of tasks based on what the user has picked.	3 Hrs	Jack
4	Debugging and unit tests.	2 Hrs	Jack

Acceptance Criteria:

- Given that the filter button is added, when the user clicks the button, then a scrollable UI should appear, allowing them to select a filtering option.
- Given that the user has selected a filter, when they close the app or log out and log back in, then the task list should remain filtered according to their last selected preference.
- Given that tasks have different statuses (e.g., "Completed," "In Progress," "Pending"), when the user selects a status filter, then only tasks with that status should appear.

- Given that no tasks match the selected status, when the user applies the status filter, then a message should appear stating, "No tasks match this status filter."

User Story #9

As a user, I would like to access a settings option that allows me to customize my experience, including enabling dark mode or adjusting the app theme, so that I can work in my preferred visual setting.

#	Description	Estimated Time	Owner
1	Create a general UI for the settings menu with placeholder options.	2 Hrs	Yana
2	Create a settings button on the main task page.	1 Hrs	Yana
3	Sync settings button with UI so that clicking it opens the settings menu.	1 Hrs	Yana
4	Implement at least one functional setting (e.g., toggle dark mode or change task view).	2 Hrs	Yana
5	Create a "Text Preferences" section within settings, including a button that leads to text customization options.	2 Hrs	Yana
6	Ensure user settings persist after closing or logging out.	2 Hrs	Yana
7	Debug and test with unit tests.	1 Hr	Yana

Acceptance Criteria:

- Given the user is on the main task page, when they click the "Settings" button, then a scrollable settings menu should appear.
- Given the settings menu is open, when the user selects dark mode, then the app's background and UI elements should immediately switch to a darker theme.
- Given the settings menu is open, when the user selects a different app theme (e.g., blue, green, or custom colors), then the app's appearance should update accordingly.
- Given the settings menu is open, when the user chooses a different task view (e.g., list view vs. grid view), then the task display should update in real time.
- Given the settings menu is open, when the user clicks "Text Preferences," then a section with font size and spacing options should appear.
- Given a user has customized their settings, when they refresh the page, close the app, or log out and log back in, then their selected settings should remain applied.

User Story #10

As a user, I would like to change the font size and spacing so that I can improve readability based on my preference.

#	Description	Estimated Time	Owner
1	Implement font size options that update all app text when selected.	2 Hrs	Jack

2	Implement text spacing options that update all app text when selected.	2 Hrs	Jack
3	Ensure task titles and descriptions dynamically adjust to the selected font size.	1 Hr	Jack
4	Ensure task lists (grid/list view) adjust their layout based on the selected spacing.	1 Hr	Jack
5	Ensure that large fonts do not cause text to overlap or break UI elements.	1 Hrs	Jack
6	Implement a preview feature so users can see changes before applying them.	2 Hrs	Jack
7	Debug and test with unit tests.	1 Hr	Jack

Acceptance Criteria:

- Given the user is in the Text Preferences menu, when they select a font size option (e.g., Small, Medium, Large), then all text in the app should update in real time, including settings, navigation, and task details.
- Given the user selects a font size, when they view tasks in a list or grid format, then task titles, descriptions, and labels should adjust accordingly without text being cut off.
- Given the user selects a font size, when they create or edit a task, then the input fields should display text in the chosen size.
- Given the font size option is selected, when the user navigates to another page, then the selected font size should remain applied.
- Given the user is in the Text Preferences menu, when they select a text spacing option (e.g., Compact, Default, Wide), then all text in the app should update immediately, including navigation menus and task details.
- Given a user changes text spacing, when they view tasks, then the spacing between tasks should adjust accordingly (Compact → minimal padding, Default → balanced, Wide → increased separation).
- Given a user has changed text spacing, when they navigate between pages, then the new spacing should remain consistent.
- Given a user has selected a font size or spacing option, when they refresh the page, close the app, or log back in, then their preferred settings should be retained.
- Given the user is in the Text Preferences menu, when they adjust font size or spacing, then a live preview should appear before applying changes.
- Given the font and spacing settings are implemented, when new text customization options (e.g., different fonts, line height) need to be added, then they should be easily integrated into the settings menu.

User Story #11

As a user, I would like to be able to change my profile picture so that I can customize my account.

#	Description	Estimated Time	Owner
1	Add a generic profile icon.	1 Hrs	Yana
2	Add a profile page UI	2 Hrs	Yana
3	Create a button to change the generic profile icon.	1 Hrs	Yana

4	Implement the ability to upload photo from your device and save that photo as the profile picture.	2 Hrs	Yana
5	Debug and test with unit tests.	2 Hr	Yana

Acceptance Criteria:

- Given the profile icon is present, when the user clicks the icon, then they should be navigated to the profile page.
- Given the user is on the profile page, when they select a photo from their device, then they should be presented with an option to crop or adjust which part of the photo will be displayed as their profile picture.
- Given the user selects and adjusts the photo, when they confirm their selection, then the new profile picture should be saved and displayed on the profile page.
- Given the profile picture is set, when the user logs out and logs back in, then their selected profile picture should persist across sessions.
- Given the user attempts to upload a file that is not an image (e.g., a PDF, text file), when they select this file, then an error message should appear stating that only image files are allowed.
- Given the user attempts to upload an unsupported image format (e.g., BMP, TIFF), when they select this file, then an error message should appear stating that only supported image formats (e.g., JPG, PNG) are allowed.
- Given the user attempts to upload an image that exceeds the size limit (e.g., 10MB), when they select the file, then an error message should appear stating that the image exceeds the maximum size limit.
- Given the user uploads an image, when the upload is successful, then the user should see a confirmation message that their profile picture has been updated.
- Given the user uploads an invalid file or encounters an error, when the upload fails, then an error message should appear detailing the reason for the failure.

User Story #12

As a user, I'd like to be able to navigate to a profile tab where I can see relevant information about my account.

#	Description	Estimated Time	Owner
1	Implement basic profile page structure and navigation.	2 Hrs	Yana
2	Display username and profile photo on profile page.	3 Hrs	Yana
3	Display user achievements such as badges and streaks.	2 Hrs	Yana
4	Add buttons to edit profile information (UI, not functionality).	1 Hr	Yana
4	Add buttons to change what badges or streaks you display (UI, not functionality).	2 Hrs	Yana
6	Debug and unit tests.	2 Hrs	Yana

Acceptance Criteria:

- Given the user is on the profile page, when they view the page, then they should see their username, profile photo, and achievement slots (e.g., badges, streaks).
- Given the user is on the profile page, when they view the page, then they should see a button to edit their profile (e.g., change username, update profile photo).
- Given the user has achievements, when they view their profile, then badges and streaks should be displayed visually. If none, a placeholder message should appear.
- Given the user is on the profile page, when they select which badges or streaks to display, then their profile should update to reflect these changes (UI only).
- Given the user customizes their profile, when they log out and log back in, then their profile settings (photo, achievements) should persist.
- Given the user accesses their profile page on different devices, when they view their profile, then the layout and content should display consistently across all devices.
- Given the user profile is incomplete, when the user views their profile, then placeholders should indicate missing profile info (e.g., "No photo uploaded").

User Story #13

As a user, I would like to create an account with unique log in credentials.

#	Description	Estimated Time	Owner
1	Create general UI that has an account creation button.	5 Hrs	Vihaan
2	Make sure all credentials are unique and link them to each user.	3 Hrs	Vihaan
3	Basic password check to ensure security during password creation.	2 Hrs	Vihaan
4	Create unit tests to debug features and ensure uniqueness.	2 Hrs	Vihaan

Acceptance Criteria:

- Given the user is on the account creation page, when they press the "Create Account" button, then they should be prompted with a new UI to enter their account details (e.g., username, email, password).
- Given the user enters a username, when they submit the form, then the system should check if the username is already taken and show an error message if it is.
- Given the user enters an email, when they submit the form, then the system should check if the email is already in use and show an error message if it is.
- Given the user inputs a password, when they submit it, then the system should provide real-time feedback on password strength (e.g., weak, moderate, strong) and prompt the user if the password is too weak.
- Given the user enters a password and a repeated password, when the passwords do not match, then the system should display an error message prompting the user to enter matching passwords.
- Given the user enters a password, when the password does not meet the required minimum length (e.g., 8 characters), then the system should display an error message instructing the user to increase the password length.
- Given the user enters an email address, when the email is not in a valid format (e.g., missing "@" or domain), then the system should display an error message prompting the user to enter a valid email.
- Given there is a server error or an unexpected issue during account creation, when the user tries to submit the form, then they should receive a generic error message indicating the issue and prompting them to try again.

- Given the user has entered valid information (unique email, valid password), when they submit the form, then they should be redirected to the login page with a success message indicating the account was created successfully.
- Given the user is creating an account, when they input a password, then the password should be securely hashed before it is stored in the database.
- Given the user interacts with the account creation form, when the form is submitted, then unit tests should validate that all validations (username, email, password) are properly executed and errors are displayed when necessary.

User Story #14
As a user, I would like to log in to my account when I return.

#	Description	Estimated Time	Owner
1	Create general login button on the same general UI as the creation button.	1 Hrs	Vihaan
2	Allow for data to be connected to unique credentials so users can log into multiple devices.	4 Hrs	Vihaan
3	Create unit tests to debug features and ensure uniqueness.	2 Hrs	Vihaan

Acceptance Criteria:

- Given the login UI is implemented, when the user accesses the login page, then they should see fields for username/email and password with a "Log In" button.
- Given the user inputs valid credentials, when they press "Log In", then they should be redirected to the task dashboard or relevant page.
- Given the user inputs an incorrect username/email, when they press "Log In", then they should see an error message indicating invalid username/email.
- Given the user inputs the correct username/email but an incorrect password, when they press "Log In", then they should see an error message indicating incorrect password.
- Given the user logs in on one device, when they log in on another device with the same credentials, then they should access the same account data across both devices.
- Given the user has forgotten their password, when they click the "Forgot Password" link, then they should be directed to the password reset page.

User Story #15
As a user, I would like to be able to delete my account if I no longer want to use the app.

#	Description	Estimated Time	Owner
1	Delete account button within user settings.	1 Hrs	Vihaan
2	Manage data that is removed from the database after account deletion properly.	3 Hrs	Vihaan
3	Create unit tests to debug features and ensure uniqueness.	2 Hrs	Vihaan

Acceptance Criteria:

- Given that a user is on the Settings page, when they scroll to the "Account Management" section, then they should see a clearly labeled "Delete Account" button.

- Given that a user clicks the "Delete Account" button, when they do so, then a confirmation popup should appear with a message explaining that all data will be permanently deleted and asking them to confirm the action.
- Given that a user is on the confirmation popup, when they select "Confirm Deletion", then their account and all associated data should be permanently removed from the database.
- Given that an account has been deleted, when another user tries to register using the same credentials (e.g., email, username), then they should be allowed to do so since the previous account no longer exists.
- Given that an account has been deleted, when the user tries to log in using their old credentials, then they should receive an error message stating that the account does not exist.

Backlog

Functional

Task Management

1. ~~As a user, I would like a task page so I can view all my tasks.~~
2. ~~As a user, I would like to be able to create tasks.~~
3. ~~As a user, I would like to be able to edit tasks.~~
4. ~~As a user, I would like to be able to delete tasks.~~
5. As a user, I would like to be able to create subtasks within a task.
6. As a user, I would like to drag and drop tasks so that I can easily visualize my task order.
7. ~~As a user, I would like to categorize tasks by priority, tags, or deadlines so that I can make decisions on which order to take the tasks.~~
8. ~~As a user, I would like to color code tasks based on categories so that I can quickly identify them.~~
9. As a user, I would like to set default task templates so that I can quickly create repetitive tasks.
10. ~~As a user, I would like to set different statuses for tasks (in-progress, blocked, waiting for review).~~
11. ~~As a user, I would like to be able to filter tasks by completion status.~~
12. ~~As a user, I would like to archive completed tasks so that I can keep my workspace clutter-free.~~
13. As a user, I would like to be able to roll over incomplete tasks to the following day, either automatically or manually.
14. As a user, I would like to be able to set task dependencies to organize connected tasks better.
15. As a user, I would like to schedule tasks for a future start date so that I can plan ahead without cluttering my current task list.
16. As a user, I would like to receive smart suggestions for breaking up long tasks into smaller subtasks

Goal-Setting & Progress Tracking

17. 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.
18. As a user, I would like to track undesirable habits/tasks so that I can check progress on if I'm lowering how unproductive I am (if time allows).

Time-Tracking & Productivity Insights

19. As a user, I would like to log time spent on tasks so that I can keep track of progress.
20. As a user, I would like to be able to visualize how my time is spent across different categories (if time allows).
21. 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.
22. 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.
23. As a user, I would like to compare my productivity across different weeks/months to track my long-term progress.
24. As a user, I would like to check time spent on breaks so that I can maintain a healthy balance.
25. 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.
26. As a user, I would like to receive productivity tips based on my task history (if time allows).

Gamification & Team Features

27. As a user, I would like to see a streak tracker to visualize my consistent task completion.
28. ~~As a user, I'd like to be able to navigate to a profile tab where I can see relevant information about my account.~~
29. As a user, I would like to be able to share or display my streak or other object on my profile.
30. As a team member, I would like to view a productivity leaderboard to foster a sense of competition and teamwork.
31. As a team member, I would like to send and receive kudos for completed tasks to encourage engagement.
32. As a team leader, I would like to assign weighted points to tasks to reflect their complexity.
33. As a user, I would like to be able to search for other users so that I can use other functionality like teams.
34. As a user, I would like to be able to be a contributor (leader or member) for multiple unique teams.

35. As a user, I would like to be able to access a unique team page for each of the teams that I am a part of to see team tasks and contributors.
36. As a team leader, I would like to create a “team” so that I can manage group tasks.
37. As a team leader, I would like to be able to add users to a team.
38. As a team leader, I would like to be able to assign tasks.
39. As a team leader, I would like to set permissions for different team members so that I can control who edits tasks.
40. As a team leader, I would like to be able to view team members’ activity.
41. As a team leader, I would like to track the time my members are spending on tasks.
42. As a team member, I would like to be able to add tasks.
43. As a team member, I would like to be able to claim tasks.
44. As a team member, I would like to mark team tasks as completed.
45. As a team member, I would like to leave comments on tasks so that I can communicate updates.
46. As a team member, I would like to tag people so that they can be notified when a task needs their attention.

User Preferences & Accessibility

47. ~~As a user, I would like to create an account with unique log in credentials.~~
48. ~~As a user, I would like to log in to my account when I return.~~
49. As a user, I'd like my data to be synced to my account so that signing in somewhere new shows me my tasks.
50. As a user, I would like to be able to use multiple devices so that I can be productive and check my tasks no matter my situation.
51. ~~As a user, I would like to access a settings option that allows me to customize my experience, including enabling dark mode or adjusting the app theme, so that I can work in my preferred visual setting.~~
52. ~~As a user, I would like to change the font size and spacing so that I can improve readability based on my preference.~~
53. As a user, I would like to choose what view I am in (calendar, timeline, etc.) so that it can be in my most preferred state.
54. As a user, I would like to set focus mode where notifications are muted for deep working sessions.
55. As a user, I would like to receive notifications so that I can be reminded of important tasks and stay productive.
56. As a user, I would like to be able to change my display name for different teams to best suit each team.
57. As a user, I would like to be able to change my profile picture so that I can customize my account.
58. ~~As a user, I would like to be able to delete my account if I no longer want to use the app.~~

Integrations & Data Export

59. As a user, I would like to integrate Planorama with third-party tools like Google Calendar or Slack for streamlined workflow (if time allows).
60. As a user, I would like to export my task data in CSV or PDF format for external analysis (if time allows).

Non-Functional

1. The platform must ensure tasks and insights load within 500ms for an optimal user experience while supporting up to 10,000 simultaneous users with minimal latency.
2. The architecture should be scalable with cloud-based solutions to accommodate a growing user base and new features.
3. A responsive design must ensure compatibility across desktop and mobile devices, while an intuitive and consistent user interface minimizes the learning curve.
4. To ensure data security, sensitive user information must be protected with end-to-end encryption, preventing unauthorized access. Additionally, all user actions should be authenticated using OAuth 2.0 standards.
5. The platform must maintain 99.9% uptime, ensuring high availability and reliability for users. Any downtime should be limited to scheduled maintenance periods, minimizing disruptions and optimizing system performance.