



Sprint 1 Retrospective

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

What went well?

Generally, the project successfully implemented a comprehensive task management system with a responsive UI, real-time updates, and persistent data storage. Key features, including task creation, editing, deletion, categorization, and filtering, were effectively integrated, allowing users to manage tasks efficiently. User account functionalities, such as registration, login, and profile customization, were completed, along with settings for theme customization, dark mode, and text preferences, enhancing personalization. Task prioritization, status tracking, and color coding improved organization, while interactive UI elements like modals, dropdowns, and confirmation prompts enhanced usability. The system maintained data persistence across sessions, ensuring a smooth and intuitive user experience.

User Story #1: As a user, I would like a task page so I can view all my tasks.

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

Completed:

Created a basic task page which included three columns for the task's description, priority, and deadline. Added a button to create tasks at the bottom of the page. When later user stories were also implemented, this page was then able to be updated to include any tasks that were added and was updated any time tasks were deleted or edited.

User Story #2: As a user, I would like to be able to create tasks.

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	3 Hrs	Sruthi

	added dynamically.		
3	Ensure task UI design is intuitive and responsive.	2 hrs	Sruthi

Completed:

The task creation feature is completed with a responsive UI. Users can open a modal overlay with the “Create Task” button to input task details, including name, description, due time, priority, color tag, and status. Tasks are dynamically added without page refresh, and required fields (task name and due time) trigger error messages if left blank. Duplicate tasks prompt a warning but can still be created. The modal overlay supports scrolling if needed, and users can cancel without saving by clicking “Cancel” or clicking outside the overlay.

User Story #3: As a user, I would like to be able to edit tasks.

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

Completed:

The task editing feature is complete with a responsive modal overlay. When a user selects a task to edit, the existing details appear pre-filled, allowing changes to any field. Only the modified fields update upon submission, while others remain unchanged. Edits are saved instantly without a page refresh and persist after logging out and back in. Required fields must be filled, and missing values trigger an error message. Data validation ensures that fields accept only appropriate input types when editing – deadlines must be valid date values, and dropdowns enforce proper selection for priority, color, and status.

User Story #4: As a user, I would like to be able to delete tasks.

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

Completed:

The task deletion feature is implemented, allowing users to remove tasks easily. When a user clicks the “Delete” button, a confirmation prompt appears to prevent accidental deletions. Upon confirmation, the task is immediately removed from the UI and no longer appears in the task list. Deletions persist across sessions, ensuring that removed tasks do not reappear after logging out and back in.

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.

1	Implement a tagging system for tasks, allowing users to assign priorities and deadlines.	4 Hrs	Jack
---	--	-------	------

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
---	---	-------	------

Completed:

The task filtering by category is complete, allowing users to sort by different categories. When attempting to sort the user is presented with a dropdown menu that will allow them to select their option for filtering. This filtering works with multiple filters applied at once and will only show the tasks that meet the criteria the user specifies. Users are also able to assign deadlines and priorities to each task. When the user attempts to create a task or edit a task they are present with the option to change these criteria. These properties also persist across session and are saved within the database based on the user

User Story #6: As a user, I would like to color code tasks based on categories so that I can quickly identify them.

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
---	--	-------	--------

Completed:

The task color-coding feature is implemented, allowing users to categorize tasks visually. A color dropdown, defaulting to “No color,” lets users select from five predefined colors. Once selected, the task is displayed with the chosen color as its background in the task list. Tasks can also be filtered by color, showing only tasks of the selected color. If no tasks match the selected filter, a message appears stating, “No tasks match.”

User Story #7: As a user, I would like to set different statuses for tasks (in-progress, blocked, waiting for review).

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
---	---	-------	------

Completed:

The task status feature is implemented, allowing users to track progress efficiently. A status dropdown, defaulting to “To-Do,” enables users to select from three predefined options: “To-Do,” “In Progress,” and “Complete.” The selected status is displayed on the task and updates immediately upon selection. Changes persist across sessions, ensuring that tasks retain their latest status even after logging out and back in. As the status is selected through a dropdown, a value is always chosen before saving.

User Story #8: As a user, I would like to be able to filter tasks by completion status.

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
---	---	-------	------

Completed:

The filter dropdown as a general concept was added along with the filtering for status. This filtering allows for a user to choose what type of status they would like to filter their tasks by. This also works with the other filters that were implemented so a user can filter by multiple things at once. If the user logs out or refreshes, these properties persist for the user.

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.

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,	2 Hrs	Yana
---	--	-------	------

	including a button that leads to text customization options.		
--	--	--	--

6	Ensure user settings persist after closing or logging out.	2 Hrs	Yana
---	--	-------	------

Completed:

Completed tasks include creating a general UI for the settings menu with placeholder options, adding a settings button on the main task page, and syncing the button to open the settings menu. Dark mode functionality was implemented, allowing users to toggle between light and dark themes. A “Text Preferences” section was also added, including a button that leads to text customization options. Users can now access the settings menu by clicking the “Settings” button on the main task page, which opens a scrollable menu. When dark mode is selected, the app’s background and UI elements switch to a darker theme immediately. Users can also change the app theme (e.g., blue, green, or custom colors), and the interface updates accordingly. Additionally, task view customization is functional, allowing users to switch between list and grid views in real time. Clicking “Text Preferences” opens a section with font size and spacing options.

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.

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
---	--	-------	------

Completed:

The option to select what font size, spacing, and fonts based on the user preferences was completed. This allows the user to select from a variety of allowed options that can customize their task planning and give better usability. When the user selects their options it shows what it looks like on the settings page before giving them the option to save those preferences. When the user goes back to the main task

page or any other page on the app, they will see that the font settings have been updated accordingly. If the user were to log off and then log back in, those settings will persist across the sessions.

User Story #11: As a user, I would like to be able to change my profile picture so that I can customize my account.

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

Completed:

Clicking the profile icon successfully navigates users to the profile page. Users can upload a photo, and the system automatically crops it to fit. They cannot upload non-image files (e.g., PDFs, text files) as well as unsupported image formats (e.g., BMP, TIFF). They are notified if the image exceeds the maximum file size (e.g., 10MB).

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.

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
5	Add buttons to change what badges or streaks you display (UI, not functionality).	2 Hrs	Yana

Completed:

Users can navigate to the profile page and view their username, profile photo, and achievement slots (e.g., badges, streaks). A button is available to edit profile details, such as changing the username or updating the profile photo. If the user has achievements, badges and streaks are displayed visually; otherwise, a placeholder message appears. Users can select which badges or streaks to display, and the profile updates accordingly (UI only). The profile page layout and content display consistently across different devices.

User Story #13: As a user, I would like to create an account with unique log in credentials.

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
---	---	-------	--------

Completed:

Users can click a button to access a create account page where they can input a username, email, and password for their new account. If the username or email are already taken, they will not be able to make an account. It also checks to make sure each field is not too short or too long, makes sure the email contains an at and a period, and makes sure the password has a number, capital and lowercase letter, and special character. Any issues will be displayed to the user when they try to submit invalid information.

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

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
---	--	-------	--------

Completed:

After creating an account and logging out, the user can log back in by navigating to the log in page and typing in their username and password. If the information is correct, it will redirect them to their tasks page and it will populate with any tasks they previously added on their account. If their information is incorrect, it will simply tell the user their username or password is incorrect.

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.

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
---	--	-------	--------

Completed:

Within the settings page, the user can click a delete account button which will prompt the user to type in their password and click a confirmation. If the password is incorrect, it will not delete their account, but if it is correct all of their data, including any tasks added, username, email, and password will be deleted. This will allow for new users to make accounts using the deleted username.

What did not go well?

Some aspects of the project faced challenges due to backend dependencies, data persistence issues, and the complexity of certain functionalities. Some features were implemented but not fully functional across sessions, while others required additional integration

that delayed their completion. Certain tasks took longer than expected due to technical constraints, and some planned functionalities were not merged into the main system in time. Despite these challenges, most core features were successfully developed.

User Story #4: As a user, I would like to be able to delete tasks.

3	Implement undo functionality for accidental deletions.	2 Hrs	Jack
---	--	-------	------

Not completed:

The functionality for users being able to undo their last 10 deleted tasks has not been added to main yet, because they do not persist across sessions. Also the 10 seconds to decide when a user presses delete has not been finished yet either.

User Story #11: As a user, I would like to be able to change my profile picture so that I can customize my account.

4	Implement the ability to upload photo from your device and save that photo as the profile picture.	2 Hrs	Yana
---	--	-------	------

Not completed:

Users can upload a photo and it follows the ACs in the story but the uploaded profile picture does not persist across sessions yet, as backend functionality to save a user was not set up. Since it is now ready, saving the profile will be completed within this week as part of a story.

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

Not completed:

The “Forgot Password” criteria was not completed on time because it required integrating email communication to send password reset links to users, making it more complex than other criteria.

How should we improve?

During Sprint 1, one of the main challenges we faced was not being intentional enough with our acceptance criteria. In some cases, we wrote down criteria that were either too vague or didn’t fully align with the task, leading to confusion and inefficiencies. This sometimes resulted in contradictory requirements or extra time spent clarifying expectations. In other cases, we had to adjust features to make more sense, even if they didn’t strictly follow the original criteria. Moving forward in Sprints 2 and 3, we plan to be more deliberate in writing clear and specific acceptance criteria to avoid misunderstandings and keep development on track.

Another area we want to improve is how we plan and prioritize tasks each week. In Sprint 1, we didn’t always take dependencies into account when deciding on work, which sometimes left team members blocked while waiting for other tasks to be completed. To improve efficiency, we plan to be more mindful of dependencies and prioritize foundational tasks earlier in the sprint. This should help streamline our workflow and reduce delays in getting work done.