

## Personal Log for Sprint 3 – Apple Pie Club

Task #	Title	Time Spent	Description
165	Set up client server communication with socket.io	150	Implementing a real-time communication between the frontend (client) and the backend (server) using Socket.io. This will allow the server to emit user status updates to the clients and receive status changes from them, ensuring that all clients are informed of the current user statuses.
166	Implement backend config that tracks user statuses based on their activity	600	Create an object or data structure on the server (like usersStatus) to track each user's current status (e.g., "available", "away", "unavailable") based on their activity. This status will be updated in real-time whenever the user interacts with the application or changes their status.
167	Change the user status automatically upon login and logout	200	Automatically set the logged-in user's status to "available" when they log in and "unavailable" when they log out. This helps reflect the user's presence and availability in the system based on their login/logout activity.
168	Allow user to set its own status to unavailable through button logic and broadcast the change to all current users	200	Implement a feature in the frontend where the user can manually set their status to "unavailable" through a button click. This action will send a signal to the server, updating the user's status for all connected clients.
169	Work on UI for the user status presence feature	60	Design and implement a user interface (UI) component that visually represents the user's current status (e.g., "available", "away", "unavailable") as well as other users statuses. This UI feature should update dynamically based on real-time status changes received from the server.
170	Add a last seen time stamp whenever the status is set to away	250	Add a timestamp to the backend that records the last time a user was marked as "away". When the user switches to an "away" status, the server should store the current time, which can then be displayed to other users as the "last seen" time for that user.

I also worked on documentation for the project on the GitHub organization, in the Sprint planning document, and even in the meeting minutes.