SPRINT 3

Issue #	User Story Title/Task Title	Description/Associated Tasks	Difficulty Level	Priority	Risk	Assign	Sprint	Status
78	**US.12**	As a registered user, I want to have access to the default channels created by the administrator, so that I can see the conversation with all registered users.	5	MED	MED	Elliot	3	Complete
84	US.12 - Task 23	Ensure that current users gain access to default channels	2	LOW	LOW	Elliot	3	Complete
85	US.12 - Task 24	Give all users permission to access default channels upon account creation	2	MED	MED	Elliot	3	Complete
90	**US.13**	As a registered user, I want to create private channels, so that I can have discussions with specific people	8	LOW	LOW	Elliot Ulysse	3	Complete
88	US.13 - Task 25	Create an endpoint to allow registered users to create private channels Ensure the creator is automatically added as admin	2	LOW	LOW	Elliot	3	Complete
89	US.13 - Task 26	Modify backend logic to ensure only invited users can access private channels Implement error handling for unauthorized access attempts	3	MED	LOW	Elliot	3	Complete
90	US.13 - Task 27	Add a "Create Private Channel" button in the UI Create a form for users to enter the channel name and invite members	2	LOW	LOW	Elliot Ulysse	3	Complete
91	US.13 - Task 28	Allow channel creators to invite registered users Ensure invited users receive access to the private channel	1	LOW	LOW	Elliot	3	Complete
79	**US.14**	As a user, I want the ability to leave channels or request access to private channels, so that I can remove myself from getting messages or getting access to another	8	LOW	LOW	Elliot Ulysse	3	Complete
92	US.14 - Task 29	Implement "Leave Channel" Feature (Backend) - Ensure users who leave are removed from the channel's member list - Prevent users from sending/receiving messages after leaving	2	MED	LOW	Elliot	3	Complete
93	US.14 - Task 30	Implement "Leave Channel UI (Frontend) - Add a "Leave Channel" button in the UI - Update UI to reflect when a user has left a channel - Display a confirmation message before leaving	2	MED	LOW	Ulysse	3	Complete
94	US.14 - Task 31	Implement "Request to Join Private Channel" Feature (Backend) - Create an API endpoint to allow users to send a join request - Store pending requests in the database - Notify the private channel creator of pending join requests	2	LOW	LOW	Elliot	3	Complete
95	US.14 - Task 32	Implement "Approve or Deny Join Request Feature (Backend) - Create an API endpoint to allow channel creators to to approve or deny join requests - Automatically add approved users to the private channel's member list - Send a notification to users when their request is accepted/denied	2	LOW	LOW	Elliot	3	Complete
96	US.14 - Task 33	Implement UI for Requestion & Approving Join Requests (Frontend) - Add a "Request to Join" button for private channels in the UI - Add "Approve" and "Deny" buttons for channel creators to manage requests	2	LOW	LOW	Elliot	3	Complete
80	**US.15**	As a user, I want to see a list of all users on the dashboard, so that I can know that user's status	11	HIGH	LOW	Yenita	3	Complete
97	US.15 - Task 34	Implement User Presence Tracking (Backend) - Create a database field to store user presence status (online, offline, away) - Implement WebSockets or polling to track real-time user activity - Set users to "away" after a certain period of inactivity - Update the status to "offline" when a user logs out	5	HIGH	LOW	Yenita	3	Complete
98	US.15 - Task 35	Fetch & Display User List on Dashboard (Backend) - Create an API endpoint to fetch a list of all registered users - Ensure the API includes the user's status (online, offline, away) - Optimize the API response to minimize performance issues	2	HIGH	LOW	Yenita	3	Complete
99	US.15 - Task 36	Display User Presence on the Dashboard UI (Frontend) - Design and implement a "User List" section on the dashboard - Fetch and display all registered users with their presence status - Use different indicators (e.g. green = online, yellow = away, gray = offline).	5	LOW	LOW	Yenita	3	Complete
100	US.15 - Task 37	Implement Real-Time Status Updates (Frontend) - Connect WebSocket or API polling to update user status dynamicall - Ensure status changes reflect immediately without refreshing the page - Handle edge cases (e.g. unexpected disconnections, session timeouts)	3	LOW	MED	Yenita	3	Complete
81	**US.16**	As a user, I want to see the last seen timestamp of other users, so that I know when they were last online	11	LOW	LOW	Yenita	3	Complete
101	US.16 - Task 38	Implement Last Seen Tracking (Backend) - Modify the database schema to store the last seen timestamp for each user - Update the backend logic to record the timestamp when a user logs out of go - Ensure the timestamp updates only when the user transitions from online to c	2	LOW	LOW	Yenita	3	Complete

102	US.16 - Task 39	Create API to Fetch Last Seen Data (Backend) - Create an API endpoint to retrieve a user's last seen timestamp. - Ensure efficient querying to avoid performance issues - Implement security measures to prevent unauthorized access to last seen da	1	MED	LOW	Yenita	3	Complete
103	US.16 - Task 40	DIsplay Last Seen Timestamp in UI (Frontend) - Design the UI to display last seen timestamps next to each user - Format the timestamp in a human-readable format (e.g., "Last seen: 2 hours - Ensure users who are currently online displat "Online" instead of a timestam	2	LOW	LOW	Yenita	3	Complete
104	US.16 - Task 41	Implement Real-Time Updates for Last Seen (Frontend) - Fetch the last seen data periodically or via WebSockets - Update the UI dynamically when the last seen status changes - Handle edge cases (e.g. when a user never logged in before)	2	LOW	LOW	Yenita	3	Complete
82	**US.17**	As a user, I want to use emojis in my messages, so that I can enhance my messages	8	LOW	LOW	Alexandre	3	Complete
105	US.17 - Task 42	Add Emoji Support to Messages (Backend) - Ensure the database supports storing emojis in messages (UTF-8 encoding) - Modify the message-processing lofic to handle emojis correctly - Update the API to allow sending and retrieving messages with emojis	3	LOW	LOW	Alexandre	3	Complete
106	US.17 - Task 43	Implement Emoji Picker in Message Input (Frontend) - Integrate an emoji picker into the message input field - Ensure users can search and select emojis - Display a preview of the selected emoji before sending	2	LOW	LOW	Alexandre	3	Complete
107	US.17 - Task 44	Display Emojis Correctly in Messages (Frontend) - Ensure emojis are rendered properly in sent and received messages - Adjust UI styling to support inline emojis without breaking text formatting - Handle cases where users copy-paste emojis from external sources	3	LOW	LOW	Alexandre	3	Complete
83	**US.18**	As a user, I want to be able to quote messages, so that I can reply to specific messages	5	LOW	LOW	Paria	3	Complete
108	US.18 - Task 45	Modift Database & API to Support Quoted Messages (Backend) - Update the database schema to support quoted messages (store the original message ID in replies) - Modify the message-sending API to allow quoting a message and storing the reference - Ensure the API retrieves and includes quoted messages when fetching conversations	2	LOW	LOW	Paria	3	Complete
109	US.18 - Task 46	Implement UI for Selecting & Quoting Messages (Frontend) - Add a "Quote" option in the message context menu (right-click or long press) - Ensure the quoted message appears in the message input before sending - Display the quotes message in a distinct style above the reply	2	MED	LOW	Paria	3	Complete
111	US.18 - Task 47	Render Quoted Messages in Chat (Frontend) - Ensure quoted messages are displayed correctly in conversation threads - Format the quoted message to show the sender's name and a preview of the original message - Ensure long quoted messages are truncated for readability, with an option to expand	3	LOW	LOW	Paria	3	Complete
120	**US.19**	As a developer, I want to be able to test the features from Sprint 2 in the CI pipeline, so that I can ensure that all features are functional upon new merges	8	MED	LOW	Zachary	3	Complete
121	US.19 - Task 48	Build Tests for Messages and Channels Actions (Backend) - Setup acceptance tests for API routes built in Sprint 2 using Jest	2	LOW	LOW	Zachary	3	Complete
122	US.19 - Task 49	Build Tests for User Account Actions (Backend) - Setup acceptance tests for API routes built in Sprint 2 using Jest	2	LOW	LOW	Zachary	3	Complete
123	US.19 - Task 50	Build Tests for Messages and Channels Actions (Frontend) - Setup acceptance tests for frontend (E2E) components built in Sprint 2 using Cypress	2	LOW	LOW	Zachary	3	Complete
124	US.19 - Task 51	Build Tests for User Account Actions (Frontend) - Setup acceptance tests for frontend (E2E) components built in Sprint 2 using Cypress	2	LOW	LOW	Zachary	3	Complete