

# Inlyne Release Summary

**Team Members:** Katarina, Taiga, Alex, Jason, Jivaj, Manuel

**Date:** June 3, 2025

## Introduction

The final release of Inlyne reflects a strong realization of our initial project goals, while also expanding beyond them in several key areas. Originally envisioned as a platform to unify disparate sources of development knowledge, such as documentation, notes, and online research, Inlyne now functions as an integrated, cross-platform tool that streamlines the way developers manage and access context during coding. The product supports real-time synchronization between a web-based editor and a VSCode extension, user authentication, markdown-based documentation with media support, and early-stage collaborative editing. These features collectively enable developers to consolidate fragmented resources into a single, accessible workspace, significantly reducing context-switching and improving overall productivity.

## Key User Stories:

As a user, I want to access and edit documents from a shared link in both the website and a VSCode extension, so I can seamlessly work across platforms with real-time synchronization.

### Acceptance Criteria:

1. Users can generate and access unique shareable links.
2. Documents open in both the web app and VSCode extension.
3. Changes in one platform are reflected in real time on the other.
4. Consistent document state is maintained across all active sessions.

As a user, I want to create an account to manage, save, and organize multiple documents, with auto saving and session persistence.

### Acceptance Criteria:

1. Users can sign up, log in/out, and stay authenticated across sessions.

2. A dashboard displays all user-owned or shared documents.
3. Documents are autosaved and tied to the user's account.
4. Users can manage permissions (view/edit) for each document.

As a user, I want to include rich content like images, code, videos, and complex formatting in my documents to better convey information.

**Acceptance Criteria:**

1. Editor supports embedded images, code blocks, videos, and graphs.
2. Nested lists (at least 3 levels), checkboxes, and diagrams are supported.
3. Users can insert anchors for linking to specific document sections.
4. Export functionality includes at least one format (PDF, Markdown, JSON).

As a user, I want to collaborate on documents in real time with others and see who is currently viewing or editing them.

**Acceptance Criteria:**

1. Real-time edits are reflected instantly for all collaborators.
2. Presence indicators (e.g., avatars or names) show active users.
3. Concurrent edits are merged without data loss or conflict.
4. Users can invite others to collaborate via email or link.

As a user, I want a modern and visually appealing interface with intuitive navigation, so I enjoy using the platform and can work efficiently.

**Acceptance Criteria:**

1. Landing page, dashboard, and editor have a clean, cohesive design.
2. Interface supports light/dark mode toggle with saved preference.
3. Dashboard supports filtering or organizing documents for easy access.

## Known Problems:

### Known Issues and Limitations

#### 1. Real-Time Collaboration:

- Most of the CRDT integration is complete, but the final logic for seamless synchronization is still pending.
- **Known Bug:** For guest users (not logged in), real-time updates do not sync immediately across sessions. Changes are only reflected after the server writes them to the database and the page is refreshed. However, real-time syncing between logged-in clients is functioning correctly, which covers our primary use case.

## **2. Duplicate requests sent to server**

- There are some duplicate prints indicating the client is sending multiple requests

## **Product Backlog:**

### **Presence Indicators**

- Display active collaborators in real-time.
- Status indicators (e.g., "typing", "viewing") to improve user awareness and collaboration.

### **Real-Time Collaboration Improvements**

- Finalize CRDT (Conflict-Free Replicated Data Type) logic integration using Yjs.
- Resolve guest client syncing issue: Currently, guest updates are delayed until server persistence and manual refresh.
- Improve guest experience to match logged-in client functionality.
- Implement credit/permission system to manage collaborative editing roles and data syncing reliability (e.g., owner vs. guest access behavior).

### **Rich Media Support Expansion**

While basic media support has been implemented, several enhanced features are planned:

- Checkboxes (interactive to-do lists)
- Sticky notes (movable and editable note elements)
- Custom icons or labels for media categorization