

## **First Wattanajungroj**

### **Posthog Progress Report for Progress Report #2**

My current work involves research and experimentation on integrating PostHog into our React/Vite frontend, utilizing US clouds to power analytics for the Gluu Admin Dashboard, with a focus on onboarding, engagement, content uploads, and moderation. The SDK initializes at app start with autocapture and masked inputs; I then gate session replay so nothing records before authentication. Replay is explicitly started after a successful login, and stopped or reset on logout. To avoid touching many files, I added a “zero-surgery” fetch wrapper that watches successful API calls and emits events such as `login_success`, `onboarding_step_completed` (for `/onboarding/stepX`), `item_uploaded` (on `POST /items`), and `item_updated` (on `PUT /items/:id`). For rapid funneling without code, I also created PostHog “Actions” from autocaptured clicks (e.g., the Google login button) and tied them to an Onboarding Funnel I’m building (`sign_up_started` → `email_verified` → `profile_completed` → `first_upload_completed`) with time-to-convert enabled; pageview trends are already live, and the funnel/usage insights are set up to be embedded into the internal admin page via PostHog’s Share→Embed, so the dashboard can ship quickly with zero maintenance. I resolved setup issues (region mismatch, TypeScript option for session recording, a missing react-is dependency, and a localStorage JSON parse error) and verified ingestion with Live events and controlled session replays that begin only after login. Some of the frontend webpage was not fully complete, it might be struggle when I am doing my experiment.