

Progress Report #3

1. Work Date / Hours Log

Date	Number of Hours	Description of Work Done
Oct 10	1	Reviewed PostHog documentation and identified integration options for frontend dashboard (funnel embedding vs API).
Oct 13	1.5	Configured PostHog environment variables and tested connection within React (posthog-js initialization).
Oct 14	2	Created dedicated Onboarding Funnel page layout and cleaned up dummy components to prepare for embed.
Oct 15	3.5	Implemented responsive PostHog iframe embed component (FunnelAnalytics.tsx) and tested rendering. Integrated routing (/analytics/funnel) and sidebar navigation entry for the new analytics page.
Oct 16	1	Group meeting with team members. Aligned the midterm deliverables and timeline.
Oct 17	2.5	Completed PostHog integration commit (feature/analytics-funnel-embed), pushed to <code>dev</code> branch and verified deployment build.
Oct 20	3	Implemented initial mock API structure in <code>src/services/mockApi/</code> ; created <code>index.ts</code> and <code>funnel.ts</code> with Promise-based functions returning sample data.
Oct 21	2.5	Expanded mock API modules (<code>user.ts</code> , <code>usage.ts</code>), tested async responses and integrated into frontend fetch logic; committed to dev.
Oct 22	2.5	Finished the progress report and midterm report.
Oct 23	1	Meeting with project owner, shared project progress

2. Description of Implementation Completed

During this reporting period, the PostHog analytics integration was finalized and the mock backend service was developed.

The Onboarding Funnel page now embeds a live PostHog funnel insight through a responsive iframe (FunnelAnalytics.tsx). Routing and navigation were updated to enable direct access from the Analytics sidebar, providing real-time conversion visualization without needing backend data.

Subsequently, a mock API system was built under src/services/mockApi/ to simulate future backend endpoints. Modules (funnel.ts, user.ts, usage.ts) contain Promise-based async functions that return typed dummy data (FunnelStep, User, UsageRecord) to enable continued frontend testing and UI interaction. This implementation mirrors the expected data structure defined in the project guidance and ensures smooth integration once the real backend is available.

All changes were developed in feature branches and merged into **dev** after successful local testing and build verification.

3. Repo Check-in Summary

Date	Branch	Commit Summary
Oct 17	feature/analytics-funnel-embed → dev	feat(analytics): integrate PostHog iframe embed for Onboarding Funnel page
Oct 21	feature/mock-backend-api → dev	feat(mock-api): add simulated backend API services for users, funnel, and usage stats