Progress Report 5

Work Date / Hours Log

Date	Number of Hours	Description of Work Done
Nov 18	3	Implemented backend support for user status management by adding an is_active boolean field to backend user types and Supabase helpers. Updated the deactivate logic to toggle is_active instead of hard-deleting users, and added dedicated admin API functions to deactivate, reactivate, and delete users while keeping status handling consistent.
Nov 19	2.5	Updated the admin frontend user-management views to display each user's is_active status and added filters to quickly show active or inactive users. Wired the new deactivate/reactivate/delete API calls into the UI actions and verified that status changes are correctly reflected after refresh and filtering.
Nov 20	1	Improved the plan-change UX by updating PlanChangeModal to pre-select the user's current plan whenever the modal opens. Ensured the selected plan matches the latest subscription data and confirmed the behavior works across different navigation paths.
Nov 20	1.5	Refined the billing pie chart styling by replacing hardcoded plan-name colors with a dynamic array-based color palette. Ensured the Free plan always uses a neutral gray (#6B7280) while paid plans are automatically assigned vibrant, visually distinct colors, and verified that the legend and chart segments remain readable for different plan combinations.

Description of Work Done

Over this development cycle, I focused on improving both user status management and billing visualization in the Gluu Admin Dashboard. On the user-management side, I introduced an is_active boolean field into the backend types and Supabase helpers, updated the deactivate logic and added a dedicated reactivate endpoint so that user status is consistently controlled through is_active, and extended the admin API with deactivate, reactivate, and delete functions. I then updated the frontend to surface this status by displaying and filtering users by is_active, and enhanced the plan change experience by pre-selecting the current plan when the PlanChangeModal opens. In parallel, I refined the billing UI by replacing hardcoded plan-name colors in the billing pie chart with a dynamic array-based palette, ensuring the Free plan always uses a neutral gray while all other plans are automatically assigned vibrant, visually distinct colors.

Repo Check-in of Implementation Completed

The files/folders I have checked into the repository so far include:

- Implementation/backend/src/routes/admin.ts (branch dev)
- Implementation/backend/src/supabase.ts (branch dev)
- Implementation/backend/src/types.ts (branch dev)
- Implementation/frontend/src/billing/BillingFilters.tsx (branch dev)
- Implementation/frontend/src/billing/BillingKPIs.tsx (branch dev)
- Implementation/frontend/src/billing/FailuresTable.tsx (branch dev)
- Implementation/frontend/src/billing/PlanDistributionPie.tsx (branch dev)
- Implementation/frontend/src/pages/Analytics/Billing.tsx (branch dev)
- Implementation/frontend/src/components/users/PlanChangeModal.tsx (branch dev)
- Implementation/frontend/src/components/users/UserActionsModal.tsx (branch dev)
- Implementation/frontend/src/components/users/UserDetailDrawer.tsx (branch dev)
- Implementation/frontend/src/components/users/UserFilters.tsx (branch dev)
- Implementation/frontend/src/lib/adminApi.ts (branch dev)
- Implementation/frontend/src/lib/mock.ts (branch dev)
- Implementation/frontend/src/pages/Users/UsersList.ts (branch dev)
- Implementation/frontend/src/services/users.ts (branch dev)