AG RUNBOOK — Admin UI → Users Management

Purpose:

This runbook defines a safe, UI-focused implementation for Admin → Users management in a multi-tenant SaaS.

It strictly consumes existing APIs and must not introduce new backend logic.

Global Constraints:

- Do not modify backend APIs
- Do not introduce new permissions
- Do not bypass tenant isolation
- Do not expose audit or internal IDs unnecessarily
- Assume Phase 7.1 and 7.2 APIs already exist

User Management Capabilities:

- List users in tenant
- Invite new user
- Show user status (PENDING, ACTIVE, DISABLED)
- Deactivate user
- Reactivate user
- View assigned roles

Step 1 — Users List Page:

Route:

- /admin/users

Behavior:

- Fetch users using admin users API
- Scope strictly to current tenant
- Paginate results
- Display columns:
- Email
- Status
- Roles
- Created date
- Actions

Step 2 — Invite User UI:

- Button: "Invite User"
- Modal or page with:
- Email input
- Role selection
- Submit via POST /api/admin/users
- Show success confirmation
- Do not display invite token

Step 3 — User Status Actions:

- ACTIVE user:
- Show "Deactivate" action
- DISABLED user:
- Show "Reactivate" action
- PENDING user:
- Show status only
- Optional: resend invite later

Step 4 — Confirmation Guards:
- Require confirmation modal before:
- Deactivate
- Reactivate
- Clearly explain effect:
- Sessions revoked
- Access removed/restored

Step 5 — Error Handling:
- Generic error messages
- Do not expose backend error details
- Handle permission-denied gracefully

Step 6 — Audit Awareness (UI Only):
- Do not display audit logs here
- Actions will automatically generate audit events server-side
- UI must not attempt to write audit data

Success Criteria:
- Admin can manage users without touching auth flows
- UI reflects real user lifecycle states
- No backend behavior is duplicated in UI
- Tenant isolation is preserved

Out of Scope:
- Bulk actions
- CSV upload
- Email preview
- Audit viewer integration
- Cross-tenant administration