FRs	Category	Sub Category	FR ID	Requirements
FRs	Security	Authentication	FR-AUTH-01	System SHALL support email/password authentication via Supabase Auth
FRs	Security	Authentication	FR-AUTH-02	System SHALL support Magic Link (passwordless) authentication
FRs	Security	Authentication	FR-AUTH-03	System SHALL support Google OAuth for simplified signup/login
FRs	Security	Authentication	FR-AUTH-04	System SHALL issue JWT tokens containing user_id and email
FRs	Security	Authentication	FR-AUTH-05	System SHALL map authenticated users to tenants via tenant_users table
FRs	Security	Authentication	FR-AUTH-06	System SHALL support users belonging to multiple tenants with different roles
FRs	Security	Authentication	FR-AUTH-07	System SHALL provide tenant selector when user has multiple tenant memberships
FRs	Security	Authentication	FR-AUTH-08	System architecture SHALL support future SAML SSO without modification
FRs	Security	Security Layer	FR-SEC-01	RLS enabled on ALL tables with tenant_id
FRs	Security	Security Layer	FR-SEC-02	set_tenant_context(tenant_id, user_id, role) function available
FRs	Security	Security Layer	FR-SEC-03	Context propagates via PostgreSQL session settings
FRs	Security	Security Layer	FR-SEC-04	audit_log captures INSERT/UPDATE/DELETE on core tables
FRs	Security	Security Layer	FR-SEC-05	Audit log is append-only (no UPDATE/DELETE allowed)
FRs	Security	Security Layer	FR-SEC-06	Tenants can only read their own audit records
FRs	Security	Cycle State Management	FR-CYC-01	Each tenant has exactly one organization_cycle_state record
FRs	Security	Cycle State Management	FR-CYC-02	Tracks current stage: discovery → audit → gap_analysis → ideation → planning → execution
FRs	Security	Cycle State Management	FR-CYC-03	Tracks stage status: not_started, in_progress, blocked, completed
FRs	Security	Cycle State Management	FR-CYC-04	advance_cycle_stage() function moves to next stage
FRs	Security	Cycle State Management	FR-CYC-05	Only owner/admin roles can advance stage
FRs	Security	Cycle State Management	FR-CYC-06	Stage changes logged to activity stream
FRs	Security	User Presence	FR-PRE-01	Track which users are currently online per tenant
FRs	Security	User Presence	FR-PRE-02	Track current view/resource each user is on
FRs	Security	User Presence	FR-PRE-03	Heartbeat updates presence (30-second interval)
FRs	Security	User Presence	FR-PRE-04	Users offline after 2 minutes without heartbeat
FRs	Security	User Presence	FR-PRE-05	API returns list of active users for tenant
FRs	Security	Record Locking	FR-LOCK-01	Users can acquire lock on dataset (type + id)
FRs	Security	Record Locking	FR-LOCK-02	Locks expire after 30 minutes
FRs	Security	Record Locking	FR-LOCK-03	Lock acquisition fails if already locked by another user
FRs	Security	Record Locking	FR-LOCK-04	Lock holder name returned when acquisition fails
FRs	Security	Record Locking	FR-LOCK-05	User can release lock explicitly
FRs	Security	Record Locking	FR-LOCK-06	Expired locks auto-released on next acquisition attempt
FRs	Security	Activity Stream	FR-ACT-01	Log actor (user/agent/system), action, target for key events
FRs	Security	Activity Stream	FR-ACT-02	Human-readable summary for each activity
FRs	Security	Activity Stream	FR-ACT-03	Highlight flag for dashboard-worthy events
FRs	Security	Activity Stream	FR-ACT-04	Activity feed API returns recent items (paginated)
FRs	Security	Activity Stream	FR-ACT-05	Tenants can only see their own activity
FRs	Security	Tech Database	TR-DB-01	All new tables follow RLS pattern: {table}_tenant + {table}_service policies
FRs	Security	Tech Database	TR-DB-02	All tables use tenant_id = current_setting('app.current_tenant_id', true)::UUID
FRs	Security	Tech Database	TR-DB-03	Indexes on tenant_id + time for all queryable tables
FRs	Security	Tech Database	TR-DB-04	JSONB for flexible fields (health_indicators, details)
FRs	Security	Tech API	TR-API-01	All endpoints call set_tenant_context() first
FRs	Security	Tech API	TR-API-02	Presence heartbeat: POST /api/presence
FRs	Security	Tech API	TR-API-03	Lock management: POST/DELETE /api/locks
FRs	Security	Tech API	TR-API-04	Cycle state: GET /api/cycle, POST /api/cycle/advance
FRs	Security	Tech API	TR-API-05	Activity feed: GET /api/activity
User Stories	Security	User Stories Authentiicate	US-A01	As a new user, I want to sign up with email/password so I can access the platform
User Stories	Security	User Stories Authentiicate	US-A02	As a user, I want to sign in with Google so I don't need another password
User Stories	Security	User Stories Authentiicate	US-A03	As a user, I want to use Magic Link so I can sign in without remembering a password

User Stories	Security	User Stories Authentiicate	US-A04	As a user with multiple organizations, I want to switch between them easily
User Stories	Security	User Stories Authentiicate	US-A05	As an enterprise admin, I want my team to use our company SSO (future capability)
User Stories	Security	User Stories Security	US-S01	As Platform Owner, I want tenant data isolated at DB level so app bugs can't leak data
User Stories	Security	User Stories Security	US-S02	As Tenant Admin, I want audit logs so I can track who changed what
User Stories	Security	User Stories Cycle management	US-C01	As Tenant Admin, I want to see current cycle stage so team knows what phase we're in
User Stories	Security	User Stories Cycle management	US-C02	As Tenant Admin, I want to advance stages so we progress through the AI Visibility journey
User Stories	Security	User Stories Cycle management	US-C03	As Team Member, I want to see health indicators so I know if we're on track
User Stories	Security	Collaboration	US-L01	As Analyst, I want to see who's online so I know who's available
FR Stories	Security	Collaboration	US-L02	As Analyst, I want to lock a gap analysis while editing so others don't overwrite my work
FR Stories	Security	Collaboration	US-L03	As Team Member, I want to see recent activity so I know what's happened
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Scenario: Email/password signup
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Given a new user with valid email
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	When user submits signup form
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Then account is created
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	And confirmation email is sent
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Scenario: Google OAuth login
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Given Google OAuth is enabled
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	When user clicks "Sign in with Google"
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Then user is redirected to Google
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	And upon success, JWT is issued
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Scenario: Multi-tenant user login
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Given user belongs to Tenant A and Tenant B
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	When user logs in
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Then tenant selector is displayed
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	And selecting Tenant A sets context to Tenant A
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Scenario: Single-tenant user login
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Given user belongs to only Tenant A
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	When user logs in
FR Accept Criteria	Security	Security Acceptance Criteria	Authentication Acceptance Criteria	Then user is automatically directed to Tenant A dashboard
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Scenario: Cross-tenant isolation
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Given User A is in Tenant A
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	When User A queries any table with Tenant B's ID
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Then zero rows are returned
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Scenario: Audit log immutability
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Given an audit_log record exists
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	When any user attempts UPDATE or DELETE
FR Accept Criteria	Security	Security Acceptance Criteria	Security Acceptance Criteria	Then the operation is blocked
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Scenario: Stage advancement
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Given tenant is in "audit" stage
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	And user has "admin" role
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	When user calls advance_cycle_stage()
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Then stage becomes "gap_analysis"

FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	And activity is logged with is_highlight=true
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Scenario: Non-admin cannot advance
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Given user has "member" role
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	When user calls advance_cycle_stage()
FR Accept Criteria	Security	Security Acceptance Criteria	Cycle State Acceptance Criteria	Then operation fails with "Permission denied"
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Scenario: Lock acquisition
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Given no lock exists on gap_analysis:123
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	When User A requests lock
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Then lock is granted to User A
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Scenario: Lock conflict
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Given User A holds lock on gap_analysis:123
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	When User B requests same lock
FR Accept Criteria	Security	Security Acceptance Criteria	Presence & Locking Acceptance Criteria	Then User B receives "Locked by User A"