

WoozySocial

Pre-Release Progression Roadmap

v1.0 · LIVE

26 February 2026

Important: The report runs against **ispqivmpffpcsquuoofmn** (Bohlale's first test — production DB with real client workspaces). SQL work today was on **adyeceovkhnacaxkymih** (WoozySocial DBMS — dev/test). All DB migrations below must be applied to **ispqivmpffpcsquuoofmn**.

Status Overview

#	Issue	Category	Status
CRIT 1	media_assets RLS disabled	Security	■ Not done
CRIT 2	Always-true RLS on posts / post_comments / notifications	Security	■ Not done
CRIT 3	Instagram (TODAY), TikTok, Twitter posting broken	Platform	■ Not done
CRIT 4	Deploy scheduler + webhook code fixes	Deploy	■ In test-branch
SEC 1	12 DB functions — mutable search_path	Security	■ Not done
SEC 2	Auth leaked password protection disabled	Auth	■ Not done
SEC 3	Auth DB connections set absolute (not %)	Auth	■ Not done
PERF 4.1	63 RLS policies re-evaluate auth.uid() per row	Performance	■ Not done
PERF 4.3	11 unindexed foreign keys	Performance	■ Not done
PERF 4.4	Duplicate index on post_drafts	Performance	■ Not done
INFO	Ayrshare profile keys — bad keys triggering invalid API calls	Fix	■ Done
INFO	8 posts stuck in 'scheduled'	Fix	■ Done

PHASE 1

Tonight — Pre-Launch Blockers

1

Enable RLS on media_assets + add policies

Supabase ispgqivmpffpcsquuofmn → SQL Editor

Enable RLS and add four workspace-scoped policies:

```
ALTER TABLE public.media_assets ENABLE ROW LEVEL SECURITY;

-- Read: workspace members
CREATE POLICY "Workspace members can read media assets"
ON public.media_assets FOR SELECT
USING (
  EXISTS (
    SELECT 1 FROM workspace_members
    WHERE workspace_members.workspace_id = media_assets.workspace_id
    AND workspace_members.user_id = (SELECT auth.uid())
  )
);

-- Insert: uploader only
CREATE POLICY "Users can upload media assets"
ON public.media_assets FOR INSERT
WITH CHECK (uploaded_by = (SELECT auth.uid()));

-- Delete: uploader only
CREATE POLICY "Uploader can delete their assets"
ON public.media_assets FOR DELETE
USING (uploaded_by = (SELECT auth.uid()));

-- Update: workspace members
CREATE POLICY "Workspace members can update media assets"
ON public.media_assets FOR UPDATE
USING (
  EXISTS (
    SELECT 1 FROM workspace_members
    WHERE workspace_members.workspace_id = media_assets.workspace_id
    AND workspace_members.user_id = (SELECT auth.uid())
  )
);
```

2

Fix always-true RLS on posts / post_comments / notifications

Supabase ispgqivmpffpcsquuofmn → SQL Editor

Query exact policy names first, then drop USING (true) / WITH CHECK (true) policies and replace with workspace-scoped versions for **posts**, **post_comments**, and **notifications**.

Query existing policies:

```
SELECT tablename, policyname, cmd, qual, with_check
FROM pg_policies
WHERE tablename IN ('posts','post_comments','notifications')
ORDER BY tablename, policyname;
```

Replace each USING (true) / WITH CHECK (true) policy with a workspace-scoped version after reviewing the query output.

3

Fix platform posting failures (Instagram / TikTok / Twitter)

Ayrshare Dashboard → Platform Connections

Cannot be fixed in code — requires direct Ayrshare dashboard access. For each platform: reconnect OAuth, run a test post, confirm 200 response.

Platform	Failing Posts	Action Required
Instagram	18 (failing TODAY)	Reconnect Instagram OAuth — token likely expired
TikTok	21	Reconnect TikTok OAuth; confirm video-only posts
Twitter/X	13	Verify API tier — free tier blocks 3rd-party posting

Pull `last_error` values from DB to get exact failure messages before reconnecting:

```
SELECT platform, error_message, COUNT(*) as failures
FROM posts
WHERE status = 'failed'
AND created_at > NOW() - INTERVAL '7 days'
GROUP BY platform, error_message
ORDER BY failures DESC;
```

4

Merge test-branch → main and deploy to Vercel

Terminal → Vercel Dashboard

Pending fixes in test-branch: `isInvalidProfileKeyError` + `nullifyWorkspaceProfileKey` + webhook workspace lookup fix.

```
git checkout main
git merge test-branch
git push origin main
```

Then confirm Vercel auto-deploys and scheduler continues returning 200.

PHASE 2**Within 24–48 Hours After Release**

5

Fix 12 DB function `search_paths` (SEC 1)

Supabase `ispqivmpffpcsuuofmn` → SQL Editor (as a migration)

Single migration script targeting all 12 vulnerable functions. Run as a tracked migration, not ad-hoc SQL.

```
-- Repeat for each of the 12 functions
ALTER FUNCTION handle_new_user() SET search_path = public, pg_catalog;
ALTER FUNCTION update_workspace_stats() SET search_path = public, pg_catalog;
-- ... (query pg_proc to get full list)

-- Find all affected functions:
SELECT proname, pronamespace::regnamespace
FROM pg_proc
WHERE pronamespace = 'public'::regnamespace
AND proisdef = false
AND NOT EXISTS (
  SELECT 1 FROM pg_options_to_table(proconfig)
  WHERE option_name = 'search_path'
);
```

6

Enable Auth Leaked Password Protection (SEC 2)

Supabase Dashboard → Authentication → Providers → Email

One toggle in the Supabase dashboard — zero code changes required.

Navigate to: **Authentication** → **Providers** → **Email** → **Advanced settings** → Enable *Leaked Password Protection*.

7

Switch Auth DB connections to percentage-based (SEC 3)

Supabase Dashboard → Project Settings → Database

Navigate to: **Project Settings** → **Database** → **Connection pooling** and switch from absolute count to percentage-based allocation.

8

Wrap auth.uid() in (SELECT ...) across 63 RLS policies (PERF 4.1)

Supabase ispqivmpffpcsquuoofmn → migration

Prevents PostgreSQL re-evaluating auth.uid() on every row scan. Update all 63 affected policies from:

```
USING (auth.uid() = user_id)
```

to:

```
USING ((SELECT auth.uid()) = user_id)
```

Find affected policies: *SELECT policyname, tablename, qual FROM pg_policies WHERE qual LIKE '%auth.uid()%' AND qual NOT LIKE '%(select auth.uid())%';*

9

Add covering indexes for 11 unindexed FK columns (PERF 4.3)

Supabase ispqivmpffpcsquuoofmn → migration

Add covering indexes for 11 unindexed FK columns:

```
CREATE INDEX ON agency_workspace_provisions(invitation_id);
CREATE INDEX ON agency_workspace_provisions(workspace_member_id);
CREATE INDEX ON inbox_read_status(last_read_message_id);
CREATE INDEX ON inbox_webhook_events(workspace_id);
CREATE INDEX ON login_tokens(workspace_id);
CREATE INDEX ON media_assets(uploaded_by);
CREATE INDEX ON notifications(invitation_id);
CREATE INDEX ON notifications(post_id);
CREATE INDEX ON post_comments(workspace_id);
CREATE INDEX ON posts(created_by);
CREATE INDEX ON short_links(user_id);
```

10

Drop duplicate index on post_drafts (PERF 4.4)

Supabase ispqivmpffpcsquuoofmn → migration

Drop the duplicate index identified on post_drafts:

```
DROP INDEX IF EXISTS post_drafts_workspace_id_idx;
```

Task	Description
Consolidate 48 permissive policies	Merge duplicate RLS policies per table into single policies
Clean up 39 unused indexes	Drop after monitoring live query patterns for 1–2 weeks
User-facing platform error messages	Show clear reasons to users when TikTok/Instagram/Twitter fail
Post failure alerting	Notify workspace owners when posts fail repeatedly (beyond admin alerts already in place)
Stuck post monitoring	Alert if a post stays in 'scheduled' more than 15 min past its time

Recommended Tackle Order — Right Now

CRIT 1	media_assets RLS	Write + run migration now — takes ~2 minutes. Highest security risk in production.
CRIT 2	Always-true policies	Query exact policy names from ispqivmpffpcsquuoofmn first, then replace them. Requires 2 SQL queries.
CRIT 3	Platform reconnections	You do this in the Ayrshare dashboard. Pull last_error values from DB first to get specific failure messages.
CRIT 4	Deploy	Merge test-branch → main when CRIT 1 + 2 are done and platform connections confirmed.

This document was generated as a pre-release progression tracker. All DB migrations target production project ispqivmpffpcsquuoofmn. Dev/test project adyeceovkhnacaxkymih does not need these migrations. Generated: 26 February 2026