* **Fixing and cleaning the TitleTesterPro repo (lrkjktnfdFIXED)**
* **Focusing ONLY on backend/core feature stability, scheduler, analytics, and authentication**
* **Removing ALL dead, duplicate, and legacy code**
* **Ensuring 100% working A/B YouTube Title testing, with correct scheduling and analytics**

### **1. Remove ALL Dead Code & Junk Folders**

**Delete:**

* **server/googleAuth.ts, server/adminRoutes.ts, server/adminService.ts, or any other files only used by the old Passport/OAuth flow or deprecated admin**
* **Unused or duplicate dashboard pages (keep only the current, mounted dashboard file)**
* **Any folders/files named assessment, attached\_assets, AI-generated markdowns, legacy scripts, test folders not run in production**

### **2. Unify & Patch Authentication (backend)**

**In server/middleware/auth.ts:**

**Replace the requireAuth middleware with the following so it accepts BOTH sb-access-token (Supabase) and session-token (legacy/demo):**

**typescript**

**CopyEdit**

**// server/middleware/auth.ts**

**import { Request, Response, NextFunction } from 'express';**

**import { supabase } from '../supabaseClient'; // adjust import as needed**

**import { storage } from '../storage'; // adjust as needed**

**export async function requireAuth(req: Request, res: Response, next: NextFunction) {**

**const sbToken = req.cookies['sb-access-token'];**

**const sessionToken = req.cookies['session-token'];**

**if (!sbToken && !sessionToken) {**

**return res.status(401).json({ error: 'Not authenticated' });**

**}**

**try {**

**let dbUser: any = undefined;**

**if (sbToken) {**

**const { data: { user }, error } = await supabase.auth.getUser(sbToken);**

**if (user) dbUser = await storage.getUserByEmail(user.email!);**

**}**

**if (!dbUser && sessionToken) {**

**const session = await storage.getSession(sessionToken);**

**if (session && storage.isValidSession(session.expires)) {**

**dbUser = await storage.getUser(session.userId);**

**}**

**}**

**if (!dbUser) return res.status(404).json({ error: 'User not found' });**

**req.user = dbUser;**

**return next();**

**} catch (error) {**

**console.error('Auth middleware error:', error);**

**return res.status(500).json({ error: 'Authentication check failed' });**

**}**

**}**

**Update ALL backend route files (especially routes/rotation.ts, routes/tests.ts, etc.) so EVERY route requiring authentication uses this unified requireAuth.**

### **3. Patch /api/auth/user Endpoint**

**In server/routes/auth-supabase.ts:**

**Update the /api/auth/user route to handle both tokens, mirroring the middleware above:**

**typescript**

**CopyEdit**

**router.get('/api/auth/user', async (req, res) => {**

**const sbToken = req.cookies['sb-access-token'];**

**const sessionToken = req.cookies['session-token'];**

**if (!sbToken && !sessionToken) {**

**return res.status(401).json({ error: 'Not authenticated' });**

**}**

**try {**

**let dbUser: any = undefined;**

**if (sbToken) {**

**const { data: { user }, error } = await supabase.auth.getUser(sbToken);**

**if (user) dbUser = await storage.getUserByEmail(user.email!);**

**}**

**if (!dbUser && sessionToken) {**

**const session = await storage.getSession(sessionToken);**

**if (session && storage.isValidSession(session.expires)) {**

**dbUser = await storage.getUser(session.userId);**

**}**

**}**

**if (!dbUser) return res.status(401).json({ error: 'Invalid session' });**

**return res.json({ user: dbUser });**

**} catch (error) {**

**console.error('[AUTH-USER] Error verifying user:', error);**

**return res.status(500).json({ error: 'Failed to get user' });**

**}**

**});**

### **4. Token Storage Cleanup**

* **Decide on ONE source for OAuth tokens (recommended: users table).**
* **Remove redundant token fields from any legacy accounts or sessions tables.**
* **Refactor any token retrieval code in the backend (especially in YouTube API calls and analytics polling) to ONLY use the primary storage (users).**
* **(Optional: If needed, add a migration script to remove deprecated fields/tables.)**

### **5. Repair and Validate Title Rotation System**

**Choose one scheduler system:  
 If Replit is used (no Redis/BullMQ), default to node-cron/setTimeout for all rotations.**

#### **a. Scheduler Setup (Example)**

**Create or update server/scheduler.ts:**

**typescript**

**CopyEdit**

**import cron from 'node-cron';**

**import { storage } from './storage';**

**import { youtubeService } from './youtubeService';**

**const scheduledJobs = new Map<string, cron.ScheduledTask>();**

**export async function startRotationJob(testId: string, intervalMinutes: number) {**

**// Remove existing job if present**

**if (scheduledJobs.has(testId)) {**

**scheduledJobs.get(testId)!.stop();**

**scheduledJobs.delete(testId);**

**}**

**// Schedule new rotation**

**const cronExpr = `\*/${intervalMinutes} \* \* \* \*`;**

**const job = cron.schedule(cronExpr, async () => {**

**await rotateTitle(testId);**

**});**

**scheduledJobs.set(testId, job);**

**}**

**export async function rotateTitle(testId: string) {**

**const test = await storage.getTest(testId);**

**if (!test || test.status !== 'active') return;**

**const titles = await storage.getTitlesByTestId(testId);**

**let currentIndex = test.currentTitleIndex || 0;**

**const nextIndex = (currentIndex + 1) % titles.length;**

**const nextTitle = titles[nextIndex];**

**await youtubeService.updateVideoTitle(test.userId, test.videoId, nextTitle.text);**

**await storage.updateTestCurrentTitle(testId, nextIndex);**

**await storage.logRotationEvent(testId, nextTitle.id, nextTitle.text, new Date(), nextIndex);**

**// (Optional: Collect analytics after switch)**

**// Check for end date or last title to complete test, then stop the job**

**}**

#### **b. On Test Creation/Activation:**

* **When a test is created/activated, call startRotationJob(testId, intervalMinutes) with the correct interval.**
* **When a test is paused, stopped, or deleted, stop and remove the scheduled job.**

#### **c. Clean Up Old Jobs**

* **Periodically iterate scheduledJobs and remove jobs for tests that no longer exist or are completed.**

### **6. Accurate Analytics and CTR Calculation**

* **When polling YouTube analytics, always calculate CTR as:  
   CTR = (clicks / impressions) \* 100**
* **If the API doesn't provide both clicks and impressions, gracefully fallback or hide the metric (do not show impressions as CTR).**
* **Ensure the winner selection for each test uses the correct metric (CTR, views, etc.).**
* **Update dashboard metrics APIs so that only available and accurate data is returned (remove any unsupported counters).**

### **7. Database Migrations for Foreign Keys and Cleanup**

**Add foreign keys with cascading deletes (example for PostgreSQL):**

**sql**

**CopyEdit**

**ALTER TABLE tests**

**ADD CONSTRAINT fk\_user**

**FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE;**

**ALTER TABLE titles**

**ADD CONSTRAINT fk\_test**

**FOREIGN KEY (test\_id) REFERENCES tests(id) ON DELETE CASCADE;**

**-- Remove deprecated/unused tables if present:**

**DROP TABLE IF EXISTS accounts CASCADE;**

**DROP TABLE IF EXISTS sessions CASCADE;**

**Remove any unused fields in users or tests as identified.**

### **8. Final Checklist and Test**

* **Ensure all backend routes (rotation, tests, analytics, dashboard, etc.) are only using the unified authentication method.**
* **Ensure only a single dashboard page/component remains.**
* **Verify that test creation, scheduling, rotation, analytics, and winner selection all work end to end with both Google and demo logins.**
* **Ensure that all dead code, duplicate logic, and assessment/junk folders are gone.**
* **Do NOT change or restyle frontend pages except as required for backend fixes.**

## **End of Master Patch Prompt**

**The AI agent must apply all of the above code and migration changes, run a full clean build, and confirm that the TitleTesterPro backend is production ready and fully functional on Replit.  
 Do not return anything but the code and scripts needed to achieve this.**

**If you require sample code for any utility or storage function (like updateTestCurrentTitle, logRotationEvent, etc.), please specify and generate them as part of the fix.**