Dynamic Server Usage Fix - October 26, 2025

Issue Summary

During the Next.js build process, five API routes were generating “Dynamic server usage” errors because they use authentication (via `headers()` function) but Next.js was attempting to statically generate them.

Affected Routes:

1. `/api/analytics/dashboard` - Analytics data endpoint
2. `/api/auditor/history` - Website audit history
3. `/api/crm/stats` - CRM statistics
4. `/api/dashboard/employee-stats` - Employee dashboard statistics
5. `/api/team/workload` - Team workload data

Root Cause

These API routes all use authentication via:
- `getServerSession(authOptions)` from NextAuth
- `getCurrentUser()` from session helpers

Both of these functions internally call the `headers()` function to access request headers for authentication. Next.js was trying to pre-render these routes at build time (static generation), but they require runtime access to request headers, making them inherently dynamic.

Solution

Added `export const dynamic = 'force-dynamic';` to each affected API route file. This configuration explicitly tells Next.js that these routes:
- Cannot be statically generated
- Must be rendered on-demand at runtime
- Require access to request-time data (headers, cookies, etc.)

Files Modified:

1. `/app/api/analytics/dashboard/route.ts`
2. `/app/api/auditor/history/route.ts`
3. `/app/api/crm/stats/route.ts`
4. `/app/api/dashboard/employee-stats/route.ts`
5. `/app/api/team/workload/route.ts`

Changes Made

Each file received the same modification at the top of the file:

```
// Before
import { NextRequest, NextResponse } from 'next/server';
import { getServerSession } from 'next-auth';
import { authOptions } from '@/lib/auth';

export async function GET(req: NextRequest) {
  // ... authentication and logic
}

// After
import { NextRequest, NextResponse } from 'next/server';
import { getServerSession } from 'next-auth';
import { authOptions } from '@/lib/auth';

export const dynamic = 'force-dynamic';  // Added this line

export async function GET(req: NextRequest) {
  // ... authentication and logic
}
```

Verification

After applying the fix:
- TypeScript compilation: No errors
- Production build: Completed successfully
- Static page generation: 154 pages generated successfully
- No Dynamic server usage errors
- All API routes properly configured as dynamic

Technical Details

What is force-dynamic?

The `dynamic = 'force-dynamic'` export is a Next.js route segment configuration that:
- Opts the route out of static optimization
- Forces the route to be rendered dynamically on every request
- Allows use of dynamic APIs like headers(), cookies(), and searchParams

Why This Fix is Necessary

API routes that perform authentication MUST be dynamic because:
1. Authentication requires reading request headers/cookies
2. Different users will get different responses (personalized data)
3. The response cannot be pre-generated at build time
4. Security: Static pages could leak authenticated data

Impact

Positive:
- Build process completes without errors
- API routes function correctly with authentication
- No performance impact (these routes were already dynamic at runtime)
- Clearer code intent - explicitly marks routes as dynamic

None:
- No breaking changes
- No changes to API behavior
- No changes to frontend code needed
- No database schema changes

Related Documentation

- Next.js Dynamic Rendering
- Next.js Route Segment Config
- Next.js API Routes Authentication

Pre-existing Issues (Not Fixed)

The test revealed some pre-existing issues that were not part of this fix:
- Inactive “Services” button on some pages
- Duplicate images on blog pages

These are documented separately and require different fixes.

Status: Complete
Build: Passing
Deployed: Ready for deployment