EasyCal Authentication Setup Guide

🎯 Quick Setup Checklist

1. Supabase Configuration

☐ Go to Supabase Dashboard → Authentication → Providers
☐ Enable Google OAuth provider
☐ Add Google OAuth credentials (Client ID + Secret)
☐ Set redirect URLs:

- `http://localhost:3000/auth/callback` (development)

- `https://your-domain.com/auth/callback` (production)

2. Database Setup

☐ Run the SQL setup script in Supabase SQL Editor
☐ Verify tables created: `user_profiles`, `usage_tracking`
☐ Check RLS policies are enabled

3. Install Dependencies

bash

```bash
npm install react-hot-toast
```

4. File Updates Required

☐ Replace `src/components/Auth/LoginModal.jsx` with new AuthModal
☐ Update `src/hooks/useAuth.js` with Google OAuth support
☐ Replace `src/components/Layout/Navbar.jsx` with auth integration
☐ Update `src/app/layout.tsx` to include AuthProvider
☐ Create `src/app/auth/callback/page.tsx` for OAuth callback

5. Environment Variables

☐ Add Supabase credentials to `.env.local`

🧪 Testing the Auth Flow

Email Authentication:

1. Click "Start Free" → "Sign Up"

2. Enter email + password

3. Check email for confirmation link

4. Confirm account and test login

Google OAuth:

1. Click "Start Free" → "Continue with Google"

2. Complete Google authorization

3. Should redirect to `/auth/callback` then `/create`

✅ Expected Behavior

Before Auth:

- Navbar shows "Sign In" and "Start Free" buttons

- No access to user-specific features

After Auth:

- Navbar shows user avatar + dropdown menu

- Dropdown includes: Profile, My Events, Settings, Billing, Sign Out

- "Create Event" button appears in navbar

🔧 User Profile Features

The system automatically:

- Creates user profile on signup

- Tracks monthly usage (free plan limits)

- Stores user preferences and settings

- Links all events to user accounts

🚨 Troubleshooting

Google OAuth not working?

- Check Google OAuth credentials in Supabase

- Verify redirect URLs match exactly

- Ensure Google Cloud Console project is configured

Database errors?

- Run the SQL setup script completely

- Check RLS policies are enabled

- Verify user permissions in Supabase

UI not updating?

- Clear browser cache

- Check browser console for errors

- Verify AuthProvider wraps the app

🎯 Next Steps After Auth

1. **Protected Routes**: Add auth guards to sensitive pages

2. **Usage Limits**: Implement free plan restrictions (5 events/month)

3. **User Dashboard**: Create `/dashboard` page for user's events

4. **Settings Page**: User profile management

5. **Billing Integration**: Stripe/payment processing

🔒 Security Notes

- RLS policies protect user data

- OAuth tokens handled securely by Supabase

- User profiles auto-created on signup

- All user data is isolated by user_id