

# Complete Supabase Security & Authentication Fix

NOTE: Documentation has been consolidated into [README.md](#). See the main README for the canonical guide. Original content preserved below.

## Overview

This guide fixes all RLS (Row Level Security) issues, authentication problems, and sets up a complete profile management system.

## Step-by-Step Deployment

### 1. Run Database Security Scripts

Execute these scripts in your Supabase SQL Editor in order:

```
```bash
```

#### 1. Fix RLS Security (CRITICAL)

[scripts/07-fix-rls-security.sql](#)

#### 2. Fix Authentication Triggers

[scripts/08-fix-auth-triggers.sql](#)

#### 3. Create Storage Bucket for Avatars

[scripts/09-create-storage-bucket.sql](#)

#### 4. Update Profiles Table (if needed)

[scripts/05-update-profiles-table.sql](#)

#### 5. Fix Duplicate Profiles (if any exist)

[scripts/06-fix-duplicate-profiles.sql](#) ```

## 2. Verify Environment Variables

Ensure your [.env.local](#) file has:

```
```env
NEXT_PUBLIC_SUPABASE_URL=your_supabase_url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your_supabase_anon_key
```

SUPABASE\_SERVICE\_ROLE\_KEY=your\_service\_role\_key ````

### 3. Configure Supabase Auth Settings

In your Supabase Dashboard → Authentication → Settings:

- **Site URL:** `http://localhost:3000` (development) or your production URL
- **Redirect URLs:** Add your callback URLs
- **Email Templates:** Customize if needed
- **Enable Email Confirmations:** Recommended for production

### 4. Test the Security Setup

Run these verification queries in Supabase SQL Editor:

```
```sql -- Check RLS is enabled on all tables
SELECT schemaname, tablename, rowsecurity
FROM pg_tables
WHERE schemaname = 'public' AND rowsecurity = true;
```

```
-- Check policies exist
SELECT schemaname, tablename, policymame, permissive, roles, cmd, qual
FROM pg_policies
WHERE schemaname = 'public'; ````
```

## ✓ What's Fixed

### 🔒 Security Issues Resolved:

- **✓ RLS Enabled** on all tables
- **✓ Proper Policies** for data access control
- **✓ User Isolation** - users can only access their own data
- **✓ Public Data** properly secured (products, categories)
- **✓ Social Features** with appropriate permissions

### 🔓 Authentication Fixed:

- **✓ Auto Profile Creation** for new users
- **✓ Profile Updates** sync with auth changes
- **✓ Proper Cleanup** on user deletion
- **✓ Error Handling** for auth failures

### 👤 Profile Management:

- **✓ Complete Profile Fields** (name, address, bio, etc.)
- **✓ Profile Picture Upload** with validation
- **✓ Real-time Updates** without page refresh
- **✓ Persistent Data** across login sessions
- **✓ Error Recovery** with retry mechanisms

## 💡 Testing Checklist

Authentication Tests:

- User registration creates profile automatically

- Login loads existing profile data
- Profile updates save to database
- Profile picture upload works
- Logout clears session properly

## Security Tests:

- Users can't access other users' data
- Public data (products) is accessible to all
- Private data requires authentication
- RLS policies block unauthorized access

## Profile Management Tests:

- All form fields save correctly
- Address information persists
- Profile picture uploads and displays
- Error states show properly
- Loading states work correctly

## Troubleshooting

### Common Issues:

#### 1. "RLS policy violation"

- Run the RLS fix script: `scripts/07-fix-rls-security.sql`

#### 2. "Profile not found"

- Run the auth triggers script: `scripts/08-fix-auth-triggers.sql`

#### 3. "Storage bucket not found"

- Run the storage setup script: `scripts/09-create-storage-bucket.sql`

#### 4. "Multiple rows returned"

- Run the duplicate fix script: `scripts/06-fix-duplicate-profiles.sql`

### Debug Commands:

```
``sql -- Check user's profile SELECT * FROM profiles WHERE id = auth.uid();
```

```
-- Check RLS status SELECT tablename, rowsecurity FROM pg_tables WHERE schemaname = 'public';
```

```
-- Check policies SELECT * FROM pg_policies WHERE schemaname = 'public'; ``
```

## Production Deployment

1. **Update Environment Variables** for production URLs
2. **Run All Scripts** in production Supabase

3. **Test Authentication Flow** thoroughly
4. **Verify RLS Policies** are working
5. **Monitor Error Logs** for any issues

## **Support**

If you encounter issues:

1. Check the Supabase logs in your dashboard
  2. Verify all scripts ran successfully
  3. Test with a fresh user account
  4. Check browser console for client-side errors
- 

 **Your 3D Outfit Builder now has enterprise-grade security and authentication!**