

# Logo Upload Fix - Supabase Storage Integration

---

## Problem Summary

---

The customer logo upload feature was failing with a **500 error** because the original implementation attempted to write logo files to the local filesystem using Node.js `fs` module. This approach doesn't work on Vercel's serverless platform where the filesystem is **read-only** and any writes are **ephemeral** (lost after the serverless function completes).

## Original Issue

```
Failed to upload logo
Server responded with status 500
```

## Solution

---

Migrated from local filesystem storage to **Supabase Storage** (cloud storage), which persists files and works seamlessly with Vercel's serverless architecture.

## Changes Made

---

### 1. New Storage Utility ( `src/app/lib/storage.ts` )

Created a comprehensive Supabase Storage utility with the following features:

- **Automatic Supabase URL detection** from `DATABASE_URL`
- **File upload** with upsert capability (replaces existing files)
- **File deletion** functionality
- **Bucket management** (auto-create if doesn't exist)
- **Public URL generation** for uploaded files

### 2. Updated Logo Upload API ( `src/app/api/admin/customers/[customerId]/logo/route.ts` )

**Before:** Used `fs.writeFile()` and `fs.unlink()` to save/delete files locally

**After:** Uses Supabase Storage APIs for cloud-based file management

Key improvements:

- ☒ Files persist across deployments
- ☒ Works on serverless platforms (Vercel, AWS Lambda, etc.)
- ☒ Better error handling with specific error messages
- ☒ Automatic cleanup of old logos when uploading new ones
- ☒ Scalable storage solution

### 3. Enhanced Frontend Error Handling ( `src/app/admin/components/CustomerManagement.tsx` )

Improved user experience:

- ☒ Detailed error messages shown to users
- ☒ Success confirmation with cloud storage info
- ☒ Better loading states
- ☒ Clear feedback for upload/delete operations

## Setup Requirements

### Required Environment Variables

You need to add the Supabase anonymous key to your environment variables:

#### Option 1: Get from Supabase Dashboard

1. Go to your Supabase project dashboard: <https://supabase.com/dashboard>
2. Navigate to **Settings** → **API**
3. Copy the **anon/public** key (not the service\_role key!)
4. Add to your `.env.local` or Vercel environment variables:

```
# Option A: Public variable (accessible client-side)
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

# OR Option B: Server-only variable
SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

**Note:** The anon key is safe to expose publicly - it's designed for client-side use with Row Level Security (RLS) protecting your data.

#### Option 2: Extract from Connection String

If you can't find the anon key, the code will automatically extract the Supabase project URL from your existing `DATABASE_URL`, but you still need to provide the anon key.

## Supabase Storage Bucket Setup

The code automatically creates a `customer-logos` bucket with these settings:

- **Public access:** Yes (logos need to be viewable)
- **File size limit:** 5MB maximum
- **Allowed MIME types:**
  - `image/jpeg`
  - `image/jpg`
  - `image/png`
  - `image/gif`
  - `image/bmp`

If you want to manually create the bucket:




1. Go to **Storage** in your Supabase dashboard
2. Click **New Bucket**
3. Name it: `customer-logos`

4. Enable **Public bucket**
5. Set file size limit: 5MB
6. Add allowed MIME types (optional)

## Image Requirements

---

### Supported Formats

-  JPEG (.jpg, .jpeg)
-  PNG (.png)
-  GIF (.gif)
-  BMP (.bmp)

### File Size Limit

- **Maximum:** 5MB per image
- **Recommended:** Under 1MB for optimal performance

### Recommended Dimensions

- **Minimum:** 200x200 pixels
- **Recommended:** 400x400 pixels (square logos work best)
- **Maximum:** 2000x2000 pixels

### Best Practices

1. **Use PNG** for logos with transparency
2. **Use JPEG** for photos or complex images
3. **Optimize images** before uploading (use tools like TinyPNG)
4. **Square aspect ratio** is ideal for consistent display
5. **High contrast** logos work best on various backgrounds

## Deployment Steps

---

### 1. Update Dependencies

```
npm install
```

This installs the `@supabase/supabase-js` package that was added.

### 2. Add Environment Variable to Vercel

```
# Using Vercel CLI
vercel env add NEXT_PUBLIC_SUPABASE_ANON_KEY

# Or using Vercel Dashboard
# 1. Go to your project settings
# 2. Navigate to Environment Variables
# 3. Add NEXT_PUBLIC_SUPABASE_ANON_KEY with your anon key
# 4. Select all environments (Production, Preview, Development)
```

### 3. Deploy to Vercel

```
git add .
git commit -m "fix: Migrate logo upload to Supabase Storage for Vercel compatibility"
git push origin main
```

Vercel will automatically deploy the changes.

### 4. Verify Deployment

1. Go to your admin dashboard
2. Navigate to **Customers** tab
3. Click **Logo** button for any customer
4. Upload a test image
5. Verify success message appears
6. Check that logo displays correctly

## Testing the Fix

### Manual Testing Checklist

- ☐ Upload a new logo for a customer
- ☐ Verify success message appears
- ☐ Confirm logo displays in the preview
- ☐ Reload the page and verify logo persists
- ☐ Upload a different logo to replace the existing one
- ☐ Delete a logo
- ☐ Verify error messages for invalid file types
- ☐ Verify error messages for files over 5MB
- ☐ Test with different image formats (JPG, PNG, GIF)

### Expected Behavior

#### ✓ Success Case:

- Upload completes within 2-3 seconds
- Alert shows: "Logo uploaded successfully to cloud storage"
- Logo appears in customer list
- Logo URL stored in database
- File accessible via public URL

#### ✗ Error Cases:

##### 1. Invalid File Type:

- Error: "Invalid file type. Allowed types: .jpg, .jpeg, .png, .gif, .bmp"

##### 2. File Too Large:

- Error: "File too large. Maximum size is 5MB"

##### 3. Missing Supabase Key:

- Error: "Storage service configuration error"
- Details: "Please ensure Supabase storage is properly configured..."

#### 4. Network Error:

- Error: "Network error - please check your connection and try again"

## Troubleshooting

---

### Issue: "Supabase anon key not configured"

#### Solution:

```
# Add the anon key to your environment variables
vercel env add NEXT_PUBLIC_SUPABASE_ANON_KEY
```

### Issue: "Failed to access or create the storage bucket"

#### Possible Causes:

1. RLS (Row Level Security) policies blocking access
2. Insufficient permissions on the anon key
3. Bucket doesn't exist and auto-creation failed

#### Solution:

1. Go to Supabase Dashboard → Storage
2. Check if `customer-logos` bucket exists
3. If not, create it manually (see setup instructions above)
4. Ensure the bucket is marked as **Public**
5. Check Storage policies:

```
```sql
```

```
- Allow public read access
```

```
CREATE POLICY "Public Access"
```

```
ON storage.objects FOR SELECT
```

```
USING (bucket_id = 'customer-logos');
```

```
- Allow authenticated users to insert/update/delete
```

```
CREATE POLICY "Authenticated users can upload"
```

```
ON storage.objects FOR INSERT
```

```
TO authenticated
```

```
WITH CHECK (bucket_id = 'customer-logos');
```

```
```
```

### Issue: Logo uploads but doesn't display

#### Possible Causes:

1. Bucket is not public
2. CORS issues
3. URL not properly stored in database

#### Solution:

1. Make bucket public in Supabase dashboard
2. Check CORS settings in Supabase (usually auto-configured)
3. Verify `logo_url` column in `customers` table contains the full URL

## Issue: Old logos from filesystem still showing

### Solution:

These were stored locally and won't work on Vercel. Users will need to re-upload logos through the admin dashboard. The old logo URLs in the database will fail to load.

## Migration Notes

---

### For Existing Customers with Logos

If you had customers with logos uploaded using the old filesystem method, those logos are **lost** on Vercel deployments because:

1. Local filesystem on Vercel is ephemeral
2. Files don't persist between deployments
3. Files aren't shared across serverless function instances

### Action Required:

- Re-upload logos for all existing customers through the admin dashboard
- Old logo URLs in the database will need to be replaced

### Database Schema

No database changes required! The `customers.logo_url` column still stores the URL, it just now points to Supabase Storage URLs instead of local file paths.

**Before:** `/customer-logos/customer-5.png`

**After:** `https://images-platform.99static.com//Yl0uDdi6Ya4JwjUE_SPcQH-Ky0k=/0x104:597x701/fit-in/500x500/99designs-contests-attachments/68/68374/attachment_68374811`

## Performance Considerations

---

### Upload Speed

- **Local FS (old):** ~50-100ms (but doesn't work on Vercel)
- **Supabase Storage (new):** ~500-1500ms (depends on file size and network)

### CDN Benefits

Supabase Storage includes CDN benefits:

- Global distribution
- Automatic caching
- Faster loading for end users
- No server load for serving images

## Security

---

### What's Protected

- ☒ Only SUPER\_ADMIN role can upload/delete logos
- ☒ File type validation (images only)
- ☒ File size limit (5MB max)
- ☒ Customer ID validation before upload

## What's Public

- ⚠️ Uploaded logos are publicly accessible (by design)
- ⚠️ Logo URLs can be accessed by anyone with the link
- ⚠️ This is intentional - logos need to display on customer portals

## Best Practices

1. Don't upload sensitive/confidential information as logos
2. Ensure uploaded images are appropriate for public display
3. Monitor storage usage in Supabase dashboard
4. Set up alerts for unusual storage activity

## Cost Implications

---

### Supabase Storage Pricing (as of 2024)

#### Free Tier:

- 1GB storage
- 2GB bandwidth per month

#### Paid Plans:

- \$0.021 per GB of storage per month
- \$0.09 per GB of bandwidth

### Estimated Usage

Assuming 100 customers with logos:

- Average logo size: 100KB
- Total storage: 10MB (negligible)
- Monthly bandwidth: ~1GB (well within free tier)

**Conclusion:** Storage costs are minimal for logo hosting.

## Alternative Solutions

---

If you prefer not to use Supabase Storage, here are alternatives:

### 1. Vercel Blob Storage

```
npm install @vercel/blob
```

- Native Vercel integration
- Similar pricing to Supabase
- Easier setup for Vercel deployments

### 2. AWS S3

```
npm install @aws-sdk/client-s3
```

- Most popular cloud storage
- Pay-as-you-go pricing
- Requires AWS account setup

### 3. Cloudinary

```
npm install cloudinary
```

- Image optimization included
- Automatic transformations
- Higher cost but more features

## Support

---

If you encounter issues:

1. **Check Vercel logs:**

```
bash
vercel logs
```

2. **Check browser console** for client-side errors

3. **Verify environment variables:**

```
bash
vercel env pull
cat .env.local
```

4. **Test locally first:**

```
bash
npm run dev
```

## Summary

---

- ✓ **Fixed:** Logo upload now works on Vercel
- ✓ **Improved:** Better error handling and user feedback
- ✓ **Scalable:** Cloud storage solution ready for production
- ✓ **Secure:** Proper authentication and file validation

**Next Steps:**

1. Add Supabase anon key to environment variables
2. Deploy to Vercel
3. Re-upload logos for existing customers
4. Monitor storage usage in Supabase dashboard