

Setup Instructions for Print Area Configuration System



Quick Start Guide

This guide will help you set up and test the complete Print Area Configuration system with Supabase backend persistence.

Prerequisites

Before you begin, make sure you have:

- Node.js (v16 or higher) installed
 - A Supabase account (create one at <https://supabase.com> if you don't have one)
 - Git installed on your system
-

Step 1: Clone and Install Dependencies

```
# Clone the repository (if you haven't already)
git clone https://github.com/PGifts2025/site.git
cd site

# Install dependencies
npm install
```

Step 2: Supabase Project Setup

2.1 Create a Supabase Project

1. Go to <https://supabase.com>
2. Sign in or create an account
3. Click "New Project"
4. Fill in the project details:
 - **Name:** `promo-gifts` (or any name you prefer)
 - **Database Password:** Create a strong password (save it somewhere safe)
 - **Region:** Choose the closest region to you
5. Click "Create new project" and wait for it to be ready (this takes ~2 minutes)

2.2 Get Your API Credentials

1. In your Supabase project dashboard, click on the **Settings** icon (gear icon) in the left sidebar
2. Click on **API** in the settings menu

3. You'll see two important values:
 - **Project URL** (looks like: `https://xxxxx.supabase.co`)
 - **anon/public key** (a long JWT token)
4. Keep this tab open - you'll need these values in the next step

Step 3: Configure Environment Variables

The project already has a `.env.example` file with the Supabase credentials pre-configured. You need to create a `.env` file:

```
# In the project root directory
cp .env.example .env
```

If you're using your own Supabase project (not the one in `.env.example`), open the `.env` file and update it with your credentials:

```
VITE_SUPABASE_URL=your_project_url_here
VITE_SUPABASE_ANON_KEY=your_anon_key_here
```

Important Notes:

- Environment variable names **must** start with `VITE_` for Vite to expose them to the browser
- After changing environment variables, you must restart the development server

Step 4: Set Up the Database

4.1 Run the Database Schema

1. In your Supabase project dashboard, click on **SQL Editor** in the left sidebar
2. Click **New Query**
3. Open the `supabase/schema.sql` file from this project
4. Copy the entire content of the file
5. Paste it into the SQL Editor in Supabase
6. Click **Run** (or press Cmd/Ctrl + Enter)

You should see a success message. This creates:

- `product_templates` table
- `print_areas` table
- All necessary indexes and triggers
- Row Level Security (RLS) policies
- Helper functions for admin checks
- A view for complete product configurations

4.2 Set Up Storage Bucket

1. In Supabase dashboard, click on **Storage** in the left sidebar
2. Click **Create a new bucket**
3. Enter the bucket name: `product-templates`

4. Check **Public bucket** (this allows product template images to be publicly accessible)
5. Click **Create bucket**

The storage bucket is now ready for template image uploads!

Step 5: Create Admin Users

The system requires users to have admin privileges to access the Print Area Configuration panel.

Method 1: During Signup (Recommended for Testing)

1. Start your development server:

```
bash
npm run dev
```

2. Open your browser and go to `http://localhost:3000`
3. Sign up with a test email (e.g., `admin@test.com`) and password
4. Go to Supabase Dashboard → **Authentication** → **Users**
5. Find your newly created user and click on them
6. Go to the **SQL Editor** and run this query (replace `USER_ID` with your user's ID):

```
UPDATE auth.users
SET raw_user_meta_data = jsonb_set(
  COALESCE(raw_user_meta_data, '{}')::jsonb,
  '{is_admin}',
  'true'::jsonb
)
WHERE id = 'USER_ID_HERE';
```

Method 2: Using Supabase Dashboard UI (Alternative)

1. Go to **Authentication** → **Users** in Supabase dashboard
2. Click on your user
3. Scroll down to **Raw User Meta Data**
4. Click **Edit**
5. Add this JSON:

```
json
{
  "is_admin": true
}
```

6. Click **Save**

Note: You may need to sign out and sign in again for the admin status to take effect.

Step 6: Run the Application

```
# Make sure you're in the project directory
cd site

# Start the development server
npm run dev
```


The application will start at **http://localhost:3000**

Step 7: Testing the System

7.1 Test Authentication

1. Open <http://localhost:3000> in your browser
2. Click **Sign In** in the header
3. Sign in with the admin user you created
4. You should see your email in the header after signing in

7.2 Test Print Area Configuration

1. While signed in as an admin, navigate to the **Design Tool** page:
 - Go to <http://localhost:3000/enhanced-designer>
 - Or navigate through the UI
2. Click on the  **Settings** icon or **Print Area Admin** button
3. You should see the **Print Area Configuration** modal
4. Select a product (e.g., "Tote Bag")
5. Try the following features:
 - **Upload Template**: Upload a new product template image
 - **Add Print Area**: Add a new print area
 - **Drag and Resize**: Move and resize the blue print area rectangles
 - **Configure Print Area**: Select a print area and modify its properties
 - **Save Configuration**: Save your changes to the database
6. Check the browser console for any errors or success messages

7.3 Verify Database Persistence

1. After saving a configuration, refresh the page
2. Open the Print Area Configuration modal again
3. Select the same product
4. Your configuration should be loaded from the database

You can also verify in Supabase:

- Go to **Table Editor** → `product_templates`
- You should see your saved product
- Go to **Table Editor** → `print_areas`
- You should see the print areas for your product

Troubleshooting

Issue: “Cannot read properties of undefined”

Solution: Make sure:

- Your `.env` file exists and has valid credentials
- Environment variables are prefixed with `VITE_`
- You’ve restarted the dev server after creating/modifying `.env`

Issue: “Only administrators can save configurations”

Solution:

- Verify you’ve set `is_admin: true` in your user’s metadata
- Sign out and sign in again to refresh the session
- Check the browser console for admin check results

Issue: Template images not loading

Solution:

- Verify the `product-templates` storage bucket exists
- Make sure the bucket is set to **public**
- Check that the image URL in the database is correct

Issue: Database errors when saving

Solution:

- Verify you’ve run the complete `supabase/schema.sql` script
- Check that RLS policies are set up correctly
- Look at the browser console and network tab for specific error messages

Issue: Mock authentication still active

Solution:

- Check that `VITE_SUPABASE_URL` in `.env` is a valid Supabase URL (not the placeholder)
- Restart the development server
- Clear browser localStorage: Open browser console and run `localStorage.clear()`

System Architecture

Backend (Supabase)

- **Authentication:** Managed by Supabase Auth
- **Database:** PostgreSQL with two main tables:
 - `product_templates` : Stores product configurations
 - `print_areas` : Stores print area definitions for each product
- **Storage:** Supabase Storage bucket for template images
- **Security:** Row Level Security (RLS) policies ensure only admins can modify data

Frontend (React + Vite)

- **UI:** React components with Tailwind CSS
- **Canvas:** Fabric.js for interactive print area editing

- **State Management:** React Context API for authentication
- **API Layer:** `src/services/supabaseService.js` handles all database operations

Data Flow

1. User signs in → Supabase Auth verifies credentials
2. Admin check → System queries user metadata for `is_admin` flag
3. Load product → Fetch from Supabase or fallback to `products.json`
4. Edit print areas → Changes tracked in React state
5. Save configuration → Batch update sent to Supabase
6. Template upload → Image uploaded to Supabase Storage, URL saved to database

Project Structure

```

site/
├── src/
│   ├── components/
│   │   ├── AuthProvider.jsx      # Authentication context
│   │   ├── HeaderBar.jsx        # Main navigation
│   │   └── PrintAreaAdmin.jsx    # Print area configuration UI
│   ├── pages/
│   │   ├── EnhancedDesigner.jsx  # Main designer page
│   │   └── ...
│   ├── services/
│   │   └── supabaseService.js    # All Supabase operations
│   ├── config/
│   │   └── supabase.js          # Supabase client configuration
│   ├── data/
│   │   └── products.json        # Product definitions (fallback)
│   ├── utils/
│   │   └── mockAuth.js          # Mock auth for testing
│   └── supabase/
│       └── schema.sql            # Database schema
├── public/
│   └── templates/                # Product template images
├── .env.example                  # Example environment variables
├── .env                          # Your environment variables (create this)
└── package.json                  # Dependencies and scripts

```

Next Steps

After successfully testing the system:

1. **Configure products:** Add more products using the Print Area Admin
2. **Upload templates:** Replace placeholder images with real product photos
3. **Test design flow:** Try creating designs with the configured print areas
4. **Deploy:** When ready, deploy to production (Vercel, Netlify, etc.)

Support

If you encounter issues:

1. Check the browser console for error messages
2. Check the Supabase dashboard logs
3. Review this document for troubleshooting steps
4. Review the code comments in key files:
 - `src/services/supabaseService.js`
 - `src/components/PrintAreaAdmin.jsx`
 - `supabase/schema.sql`

Additional Resources

- [Supabase Documentation](https://supabase.com/docs) (<https://supabase.com/docs>)
- [Vite Documentation](https://vitejs.dev/) (<https://vitejs.dev/>)
- [React Documentation](https://react.dev/) (<https://react.dev/>)
- [Fabric.js Documentation](http://fabricjs.com/docs/) (<http://fabricjs.com/docs/>)
- [Supabase Auth Guide](https://supabase.com/docs/guides/auth) (<https://supabase.com/docs/guides/auth>)
- [Supabase Storage Guide](https://supabase.com/docs/guides/storage) (<https://supabase.com/docs/guides/storage>)

Happy Coding! 🎉👨‍💻