# Basil - Cultural Cards App 🌸

A React web app for sharing cultural moments through kawaii-illustrated cards with QR code support.

## Features

- 🎨 Beautiful card-based UI with kawaii aesthetic
- 🔐 User authentication (signup/login)
- 📸 Image upload for card illustrations
- 🎵 Music support (Spotify/YouTube URLs or MP3 uploads)
- 🗑️ Delete your own cards
- 📱 Mobile-responsive design
- 🔗 QR code friendly (access cards via `?id=XXX`)
- ⚡ Real-time updates

## Tech Stack

- **Frontend**: React + Vite
- **Styling**: Tailwind CSS
- **Backend**: Supabase (Auth, Database, Storage)
- **Routing**: React Router
- **Icons**: Lucide React

## Setup Instructions

### 1. Create Supabase Project

1. Go to supabase.com
2. Create a new project
3. Wait for the database to be ready
4. Go to Project Settings > API to get your credentials

### 2. Set Up Database

Go to SQL Editor in Supabase and run:

```sql
-- Create cards table
CREATE TABLE cards (
  id UUID DEFAULT gen_random_uuid() PRIMARY KEY,
  user_id UUID REFERENCES auth.users(id) ON DELETE CASCADE,
  title TEXT NOT NULL,
  description TEXT NOT NULL,
  image_url TEXT NOT NULL,
  music_url TEXT,
  music_file_url TEXT,
  category TEXT,
  card_id TEXT UNIQUE NOT NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
  updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

-- Enable RLS
ALTER TABLE cards ENABLE ROW LEVEL SECURITY;

-- Public read access
CREATE POLICY "Anyone can view cards"
  ON cards FOR SELECT
  TO public
  USING (true);

-- Authenticated users can insert
CREATE POLICY "Authenticated users can insert cards"
  ON cards FOR INSERT
  TO authenticated
  WITH CHECK (auth.uid() = user_id);

-- Users can update their own cards
CREATE POLICY "Users can update own cards"
  ON cards FOR UPDATE
  TO authenticated
  USING (auth.uid() = user_id)
  WITH CHECK (auth.uid() = user_id);

-- Users can delete their own cards
CREATE POLICY "Users can delete own cards"
  ON cards FOR DELETE
  TO authenticated
  USING (auth.uid() = user_id);

-- Create indexes
CREATE INDEX cards_card_id_idx ON cards(card_id);
CREATE INDEX cards_user_id_idx ON cards(user_id);
CREATE INDEX cards_category_idx ON cards(category);
```

### 3. Set Up Storage

In Supabase Dashboard → Storage:

1. Create bucket: `card-images` (make it public)
2. Create bucket: `card-music` (make it public)

For each bucket, set up a policy:

```sql
-- Allow public read access
CREATE POLICY "Public Access"
ON storage.objects FOR SELECT
USING ( bucket_id = 'card-images' );

-- Allow authenticated users to upload
CREATE POLICY "Authenticated users can upload"
ON storage.objects FOR INSERT
WITH CHECK (
  bucket_id = 'card-images' AND
  auth.role() = 'authenticated'
);

-- Repeat for 'card-music' bucket
```

### 4. Install Dependencies

```bash
# Clone or create the project
npm install
```

### 5. Environment Variables

Create a `.env` file in the root:

```env
VITE_SUPABASE_URL=https://your-project.supabase.co
VITE_SUPABASE_ANON_KEY=your-anon-key-here
```

### 6. Run Development Server

```bash
```

npm run dev

Visit `http://localhost:5173`

### 7. Build for Production

```bash
npm run build
```

The `dist` folder will contain your production build.

### 8. Deploy to Netlify

#### Option A: Drag & Drop
1. Run `npm run build`
2. Drag the `dist` folder to [Netlify](https://app.netlify.com/drop)
3. Add environment variables in Site Settings

#### Option B: GitHub Integration
1. Push your code to GitHub
2. Connect repository in Netlify
3. Set build command: `npm run build`
4. Set publish directory: `dist`
5. Add environment variables

Important: Add your environment variables in Netlify:
- Go to Site Settings → Environment Variables
- Add `VITE_SUPABASE_URL` and `VITE_SUPABASE_ANON_KEY`

## Usage

### For Users

1. **Sign Up**: Create an account with email/password
2. **Add Cards**: Click "Add Card" to create a new cultural card
   - Upload a kawaii illustration
   - Add title and description
   - Optionally add category and music
3. **View Cards**: Browse all cards on the home page
4. **Delete Cards**: You can only delete cards you created

### QR Code Usage

Each card has a unique ID (e.g., `ABC123`). The URL format is:
```
https://your-domain.netlify.app/?id=ABC123
```

You can:
1. Generate QR codes pointing to these URLs
2. Print them on physical cards
3. When scanned, users see the full card details

### Adding Music

Option 1: URL (Spotify, YouTube, SoundCloud)
- Just paste the URL (e.g., `https://open.spotify.com/track/...`)
- The app will automatically embed the player

Option 2: Upload MP3
- Upload your own MP3 file
- Maximum file size depends on Supabase storage limits

## Project Structure

```
basil/
├── src/
│   ├── components/
│   │   ├── Auth/          # Login & SignUp
│   │   ├── Cards/         # Card display components
│   │   ├── Forms/         # Add card form
│   │   ├── Layout/        # Navbar
│   │   └── Music/         # Audio player
│   ├── hooks/
│   │   ├── useAuth.js     # Authentication logic
│   │   └── useCards.js    # Card CRUD operations
│   ├── lib/
│   │   └── supabase.js    # Supabase client
│   ├── App.jsx
│   ├── main.jsx
│   └── index.css
├── .env
├── package.json
└── README.md
```

## Customization

### Change Colors
Edit `tailwind.config.js` to customize the color scheme. Current theme uses pink/purple.

### Add Categories
Categories are optional text fields. You can add a dropdown by modifying `AddCardForm.jsx`:

```jsx
const CATEGORIES = ['Film', 'Literature', 'Art', 'Music', 'Philosophy'];
```

### Generate QR Codes in App

Install the QR code library:
```bash
npm install qrcode
```

Add a QR code generator component to display QR codes for each card.

## Troubleshooting

### Images not uploading
- Check that storage buckets are created and public
- Verify RLS policies allow authenticated users to insert

### Cards not showing
- Check browser console for errors
- Verify Supabase credentials in `.env`
- Ensure RLS policy allows public SELECT

### Authentication issues
- Check Supabase email settings
- Verify SITE_URL in Supabase auth settings

## Contributing

This is a personal project for friends, but feel free to fork and customize for your own use!

## License

MIT