



❏ KOMPLETTE .env DATEI (Zum Ersetzen)

Öffne backend/.env und ersetze den GESAMTEN Inhalt mit diesem hier:

```
# ===== ENVIRONMENT =====
NODE_ENV=development
HOST=localhost
PORT=3000

# ===== FRONTEND CONFIG =====
FRONTEND_URL=http://localhost:5500
ALLOWED_ORIGINS=http://localhost:5500

# ===== DATABASE CONFIG =====
DB_HOST=localhost
DB_PORT=5432
DB_NAME=song_nexus_dev
DB_USER=song_nexus_user
DB_PASSWORD=SuperSicheresPasswort2025!
DB_SSL=false

# ===== AUTH CONFIG =====
JWT_SECRET=dein_super_geheimes_jwt_schluessel_min_32_zeichen_lang_1234567890
JWT_EXPIRE=7d
BCRYPT_ROUNDS=10

# ===== PAYPAL CONFIG =====
PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID_HERE
PAYPAL_CLIENT_SECRET=YOUR_PAYPAL_CLIENT_SECRET_HERE
PAYPAL_MODE=sandbox

# ===== UPLOAD CONFIG =====
UPLOAD_DIR=./uploads
MAX_FILE_SIZE=52428800

# ===== LOGGING CONFIG =====
LOG_LEVEL=debug
LOG_FILE=logs/app.log
```

✓ **Speichern** (Ctrl + S)

📄 PASSWORT SICHERHEIT - Detailliert erklär!

⚠ DAS IST WICHTIG!

Wo ist dein Passwort sicher?

Dein Passwort in der `.env` Datei ist:

Ort	Sicherheit	Grund
Lokal auf deinem PC	✓ SICHER	Nur du hast Zugriff
GitHub	✗ GEFÄHRLICH	Jeder kann dein Passwort sehen!
Produktionsserver	⚠ VORSICHT	Umgebungsvariablen nutzen, nicht Dateien

📄 WIE DU SICHER BLEIBST:

1📄 `.env` NIEMALS in GitHub pushen!

Öffne `backend/.gitignore` und stelle sicher, dass `.env` drin steht:

```
.env
.env.local
.env.*.local
node_modules/
logs/
```

Wenn `.env` nicht dort ist, **FÜGE ES HINZU!**

✓ Dann ist dein Passwort safe!

2📄 Starkes Passwort verwenden

Dein aktuelles Passwort in der `.env`:

```
SuperSicheresPasswort2025!
```

Das ist **OKAY** für Entwicklung, aber:

Anforderung	Dein Passwort	Status
Mindestens 8 Zeichen	26 Zeichen	✓ PASS
Großbuchstaben	S,P	✓ PASS
Kleinbuchstaben	upericheresasswort	✓ PASS
Zahlen	2025	✓ PASS

Anforderung	Dein Passwort	Status
Sonderzeichen	!	✓ PASS

Du kannst es auch zufälliger machen:

```
P@ssw0rd_SongNexus_2025_Secure!
```

Oder noch besser - einen Generator nutzen:

<https://www.random.org/strings/>

3 JWT_SECRET - Das wichtigste Passwort

Das ist das **super kritische** Passwort:

```
dein_super_geheimes_jwt_schluessel_min_32_zeichen_lang_1234567890
```

⚠ **WICHTIG:**

- Muss **mindestens 32 Zeichen** lang sein
- Muss **ZUFÄLLIG** sein (nicht "test123")
- Darf **NIEMALS** geleakt werden

Generiere einen starken JWT Secret:

Öffne PowerShell und führe das aus:

```
$bytes = New-Object byte[] 32
$rng = [System.Security.Cryptography.RNGCryptoServiceProvider]::new()
$rng.GetBytes($bytes)
$jwt = [Convert]::ToBase64String($bytes)
Write-Output $jwt
```

Kopiere die lange Zufallszeichenkette und verwende sie in `.env`:

```
JWT_SECRET=<DEINE_GENERIERTE_ZEICHENKETTE>
```

4 PayPal Credentials - Wo bekommst du die?

1. Gehe zu: <https://developer.paypal.com>
2. Login mit deinem PayPal Account (oder erstelle einen)
3. Gehe zu: **Apps & Credentials**
4. Wähle: **Sandbox** (NICHT Live!)
5. Du siehst:

- **Client ID** → In PAYPAL_CLIENT_ID einfügen
- **Secret** → In PAYPAL_CLIENT_SECRET einfügen

Beispiel nach dem Einfügen:

```
PAYPAL_CLIENT_ID=AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsT  
PAYPAL_CLIENT_SECRET=XyZaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQq
```

✓ **Das ist sicher** - Das sind nur Sandbox (Test) Credentials!

▮ POSTGRESQL SETUP - SCHRITT FÜR SCHRITT

Schritt 1: PowerShell öffnen (WICHTIG: Als Admin!)

1. Drücke: Windows + X
2. Wähle: **Terminal (Administrator)** oder **PowerShell (Administrator)**
3. Ein Fenster öffnet sich

Schritt 2: PostgreSQL Verbindung

Tippe in das PowerShell Fenster:

```
psql -U postgres
```

Drücke Enter

Es fragt dich nach dem Passwort:

- Das ist das Passwort, das du beim PostgreSQL Installation gesetzt hast
- **Tippe es ein** (du siehst keine Zeichen, das ist normal!)
- Drücke Enter

Expected Output:

```
psql (17.0)  
Type "help" for help.  
  
postgres=#
```

✓ Du bist jetzt in PostgreSQL!

Schritt 3: Datenbank erstellen

Tippe **Wort für Wort** (mit Semikolon am Ende!):

```
CREATE DATABASE song_nexus_dev;
```

Drücke Enter

Expected Output:

```
CREATE DATABASE  
postgres=#
```

✓ Datenbank erstellt!

Schritt 4: Benutzer erstellen

Tippe:

```
CREATE USER song_nexus_user WITH PASSWORD 'SuperSicheresPasswort2025!';
```

Drücke Enter

WICHTIG: Das Passwort hier muss **EXAKT** so sein wie in deiner `.env` Datei:

```
DB_PASSWORD=SuperSicheresPasswort2025!
```

Expected Output:

```
CREATE ROLE  
postgres=#
```

✓ Benutzer erstellt!

Schritt 5: Berechtigungen vergeben

Tippe:

```
GRANT ALL PRIVILEGES ON DATABASE song_nexus_dev TO song_nexus_user;
```

Drücke Enter

Expected Output:

```
GRANT
postgres=#
```

✓ Berechtigungen vergeben!

Schritt 6: PostgreSQL verlassen

Tippe:

```
\q
```

Drücke Enter

Du bist jetzt zurück im PowerShell Terminal.

✓ **PostgreSQL Setup fertig!**

SCHEMA IN DATABASE LADEN

Im PowerShell Terminal (nicht in psql!):

```
cd C:\Users\sebas\Desktop\SongSeite\backend
```

Drücke Enter

Jetzt lade das Schema:

```
psql -U postgres -d song_nexus_dev -f PHASE3_DATABASE_SCHEMA.sql
```

Drücke Enter

Es fragt nach dem PostgreSQL Passwort:

- **Tippe dein PostgreSQL Passwort ein**
- Drücke Enter

Expected Output:

```
CREATE TABLE
CREATE INDEX
CREATE TABLE
...
CREATE VIEW
CREATE OR REPLACE VIEW
```

✓ **Datenbank Schema ist geladen!**

📄 PASSWORT SICHERHEIT - ZUSAMMENFASSUNG

Was du beachten musst:

Punkt	✓ SICHER	✗ UNSICHER
.env in GitHub	Nicht pushen (in .gitignore)	Pushen = Passwort leak
Passwort komplexität	Groß + Klein + Zahlen + Sonderzeichen	Einfach (123456)
Passwort speichern	In .env (lokal)	In Code hardcoded
PostgreSQL Passwort	SuperSicheresPasswort2025!	admin
JWT_SECRET	Zufällig generiert, 32+ Zeichen	jwt123
PayPal Sandbox	Test Credentials	Live Credentials

DAS MUSST DU TESTEN:

Nach dem PostgreSQL Setup, öffne PowerShell und tippe:

```
psql -U song_nexus_user -d song_nexus_dev -W
```

Drücke Enter

Es fragt: Password for user song_nexus_user:

Tippe: SuperSicheresPasswort2025!

Drücke Enter

Expected Output:

```
psql (17.0)
Type "help" for help.

song_nexus_dev=>
```

✓ **Perfekt! Du kannst dich mit dem neuen Benutzer anmelden!**

Verlasse mit:

```
\q
```

❏ KRITISCHE PASSWÖRTER - REFERENCE

Damit du nicht verwirrt wirst, hier sind all deine Passwörter:

Passwort	Wo	Wert	Zweck
PostgreSQL Admin	Windows PostgreSQL Installation	??? (Was hast du gesetzt?)	Admin für PostgreSQL
DB_PASSWORD	.env Datei	SuperSicheresPasswort2025!	Passwort für song_nexus_user
JWT_SECRET	.env Datei	dein_super_geheimes...	Für JWT Tokens
PayPal Client ID	.env Datei	Sandbox ID	PayPal Test
PayPal Secret	.env Datei	Sandbox Secret	PayPal Test

⚠ SICHERHEIT CHECKLISTE

ENTWICKLUNG (Das ist jetzt)

- ☐ .env NICHT in Git pushen
- ☐ Passwort ist komplex
- ☐ PayPal ist Sandbox Mode
- ☐ JWT_SECRET ist zufällig (32+ Zeichen)

SPÄTER (Vor Production):

- ☐ .env Datei auf Server kopieren (via SSH)
- ☐ Passwörter wechseln (neue generieren)
- ☐ PayPal auf LIVE umstellen
- ☐ HTTPS aktivieren
- ☐ Database Backups aufsetzen

❏ NÄCHSTER SCHRITT

Wenn alles fertig:

```
cd backend
npm start
```

Du solltest sehen:

- ✓ Server running on http://localhost:3000
- ✓ Database connected

Dann ist Phase 3 ✓ LIVE!

Fragen zur Sicherheit? Sag mir Bescheid! ❏

