MelodyMind Backend Setup Guide

This guide documents the step-by-step process for setting up the backend of MelodyMind, including database verification, Next.js project initialization, and Prisma ORM integration.

1. Verify Database Connection

```
# Enter Postgres shell inside container
docker exec -it melodymind-db psql -U melody -d melodymind

# Run test query
SELECT 1;

# Enable pgvector
CREATE EXTENSION IF NOT EXISTS vector;
\dx
```

2. Initialize Next.js App (Frontend + API)

```
# From project root (melodymind/)
npx create-next-app@latest frontend

# Choose: TypeScript + App Router
```

3. Add Prisma ORM

```
cd frontend

# Install Prisma
npm install @prisma/client
npm install --save-dev prisma

# Initialize Prisma
npx prisma init
```

4. Configure Database URL

```
# In frontend/.env
DATABASE_URL="postgresql://melody:melodypass@localhost:5432/melodymind"
```

5. Define Prisma Schema

```
datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

generator client {
  provider = "prisma-client-js"
}

model User {
  id        Int      @id @default(autoincrement())
  email     String   @unique
  password  String
  createdAt DateTime @default(now())
  texts     Text[]
  audios    Audio[]
}

model Text {
  id        Int      @id @default(autoincrement())
```

```
  title     String
  raw       String
  createdAt DateTime @default(now())
  user      User     @relation(fields: [userId], references: [id])
  userId    Int
}

model Audio {
  id        Int      @id @default(autoincrement())
  title     String
  url       String
  durationS Int
  createdAt DateTime @default(now())
  user      User     @relation(fields: [userId], references: [id])
  userId    Int
}
```

6. Apply Migration

```
# Run from frontend/ directory
npx prisma migrate dev --name init
```

7. Test Prisma in Next.js

```
// frontend/lib/prisma.ts
import { PrismaClient } from '@prisma/client';

const globalForPrisma = global as unknown as { prisma: PrismaClient };

export const prisma =
  globalForPrisma.prisma ||
  new PrismaClient({
    log: ['query'],
  });

if (process.env.NODE_ENV !== 'production') globalForPrisma.prisma = prisma;
// frontend/app/api/users/route.ts
import { NextResponse } from "next/server";
import { prisma } from "@/lib/prisma";

export async function GET() {
  const users = await prisma.user.findMany();
  return NextResponse.json(users);
}
```

8. Run and Test

```
cd frontend
npm run dev

# Open browser at:
http://localhost:3000/api/users   # should return []
```