-- File: progymdb.sql

-- Path: ./progymdb.sql

-- ===========================

-- CREATE DATABASE

-- ===========================

-- Create the main database for the ProGym application if it doesn't already exist,

-- using UTF-8 character set for full Unicode support.

CREATE DATABASE IF NOT EXISTS progymdb

CHARACTER SET utf8mb4

COLLATE utf8mb4\_unicode\_ci;

-- Switch the connection to the ProGym database for subsequent operations

USE progymdb;

-- ===========================

-- USERS TABLE

-- ===========================

-- Stores each user's account details and profile information.

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique user identifier

username VARCHAR(255) NOT NULL UNIQUE, -- Login name, must be unique

email VARCHAR(255) NOT NULL UNIQUE, -- User email, must be unique

password VARCHAR(255) NOT NULL, -- Hashed password for authentication

role ENUM('user', 'premium') NOT NULL DEFAULT 'user', -- Access level

height\_cm INT, -- User height in centimeters

weight\_kg INT, -- User weight in kilograms

age INT, -- User age in years

gender ENUM('male', 'female', 'other'), -- User gender

goal ENUM('cutting', 'bulking', 'maintenance'), -- Fitness goal

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP -- Record creation timestamp

);

-- ===========================

-- WORKOUTS TABLE

-- ===========================

-- Logs individual workout entries for progress tracking.

CREATE TABLE IF NOT EXISTS workouts (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique workout entry ID

user\_id INT NOT NULL, -- Which user performed the workout

category VARCHAR(50) NOT NULL, -- Workout category (e.g., chest, legs)

exercise VARCHAR(100) NOT NULL, -- Name of the exercise performed

sets INT NOT NULL, -- Number of sets completed

reps INT NOT NULL, -- Number of repetitions per set

weight DECIMAL(5,2) NOT NULL, -- Weight used, in kilograms

date DATE NOT NULL, -- Date of the workout

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Entry creation timestamp

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Link to users table

);

-- ===========================

-- MEALS TABLE

-- ===========================

-- Records daily meals for calorie and macro nutrient tracking.

CREATE TABLE IF NOT EXISTS meals (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique meal entry ID

user\_id INT NOT NULL, -- Which user logged the meal

date DATE NOT NULL, -- Date of the meal

calories INT, -- Total calories consumed

protein INT, -- Protein in grams

carbs INT, -- Carbohydrates in grams

fat INT, -- Fat in grams

meal\_type ENUM('breakfast', 'lunch', 'dinner') DEFAULT 'lunch', -- Meal category

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Entry creation timestamp

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Link to users table

);

-- ===========================

-- WATER INTAKE TABLE

-- ===========================

-- Tracks daily hydration levels by total water consumed.

CREATE TABLE IF NOT EXISTS water\_intake (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique water intake entry ID

user\_id INT NOT NULL, -- Which user logged the intake

date DATE NOT NULL, -- Date of water consumption

total\_ml INT NOT NULL, -- Total water consumed in milliliters

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Entry creation timestamp

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Link to users table

);

-- ===========================

-- FOODS TABLE

-- ===========================

-- Master list of foods for meal planning and calorie lookup.

CREATE TABLE IF NOT EXISTS foods (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique food item ID

name VARCHAR(100) NOT NULL, -- Common name of the food

calories FLOAT NOT NULL, -- Calories per 100g

protein\_per\_100g FLOAT NOT NULL, -- Protein per 100g

carbs\_per\_100g FLOAT NOT NULL, -- Carbs per 100g

fat\_per\_100g FLOAT NOT NULL -- Fat per 100g

);

-- ===========================

-- MEAL PLANS TABLE

-- ===========================

-- Stores metadata for autogenerated meal plans.

CREATE TABLE IF NOT EXISTS meal\_plans (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique meal plan ID

user\_id INT NOT NULL, -- Owner of the meal plan

meals\_per\_day INT NOT NULL, -- Number of meals planned per day

total\_calories INT, -- Total daily calories in the plan

total\_protein FLOAT, -- Total daily protein

total\_carbs FLOAT, -- Total daily carbs

total\_fat FLOAT, -- Total daily fat

goal ENUM('cutting', 'bulking', 'maintenance'), -- User's fitness goal for the plan

date DATE NOT NULL, -- Date the plan was generated

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Creation timestamp

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Link to users table

);

-- ===========================

-- MEAL PLAN ITEMS TABLE

-- ===========================

-- Details which foods and quantities make up each meal in a plan.

CREATE TABLE IF NOT EXISTS meal\_plan\_items (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique item entry ID

meal\_plan\_id INT NOT NULL, -- Which meal plan this item belongs to

meal\_number INT NOT NULL, -- Sequence number of the meal (e.g., 1 for breakfast)

food\_id INT NOT NULL, -- Which food item is used

quantity\_grams INT NOT NULL, -- Quantity of food in grams

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Entry creation timestamp

FOREIGN KEY (meal\_plan\_id) REFERENCES meal\_plans(id) ON DELETE CASCADE, -- Link to meal\_plans

FOREIGN KEY (food\_id) REFERENCES foods(id) ON DELETE CASCADE -- Link to foods table

);

-- ===========================

-- GYMBOT SESSIONS TABLE

-- ===========================

-- Tracks individual chat sessions with the GymBot feature.

CREATE TABLE IF NOT EXISTS gymbot\_sessions (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique session ID

user\_id INT NOT NULL, -- Which user started the session

title VARCHAR(255) DEFAULT 'Untitled Chat', -- User-assigned session title

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Session start timestamp

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Link to users table

);

-- ===========================

-- GYMBOT MESSAGES TABLE

-- ===========================

-- Stores each message exchanged between user and GymBot within a session.

CREATE TABLE IF NOT EXISTS gymbot\_messages (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unique message ID

session\_id INT NOT NULL, -- Which session this message belongs to

sender ENUM('user', 'bot') NOT NULL, -- Who sent the message

text TEXT NOT NULL, -- Content of the message

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Message timestamp

FOREIGN KEY (session\_id) REFERENCES gymbot\_sessions(id) ON DELETE CASCADE -- Link to sessions

);