

# CoachGPT Pro - Full Backend Documentation

## CoachGPT Pro - Full Backend Documentation (Final Version)

### 1. Project Overview

CoachGPT Pro Backend is a Node.js + TypeScript server managing:

- User Authentication (Register, Login, Delete)
- Workout Plan Management (Generate, Swap, Add, Delete exercises)
- Exercise Database Management (Get all exercises)
- Action History (Track plan changes)
- Database: PostgreSQL

Architecture: Microservice style - Auth Service, Plan Service, Exercise Service.

### 2. Database Structure

Tables:

- users (id, email, password)
- workout\_plans (id, user\_id, goal, days\_per\_week, plan\_data, created\_at)
- plan\_actions (id, plan\_id, action\_type, old\_exercise, new\_exercise, week\_number, day\_name, created\_at)
- exercises (id, name, muscle\_group, equipment, difficulty, substitutes)

### 3. Authentication (Auth Controller)

Functions:

- registerUser: Create a new user with hashed password
- loginUser: Authenticate user and return JWT token
- deleteUser: Delete a user account by ID

### 4. Workout Plans (Plan Controller)

Functions:

- generatePlan: Create personalized 4-week plan
- getPlanByUser: Fetch latest plan by user ID
- getPlanById: Fetch plan by ID
- swapExercise: Swap exercise (whole program or specific week)

# CoachGPT Pro - Full Backend Documentation

- addExerciseToPlan: Add new exercise to a day
- deleteExerciseFromPlan: Remove exercise from a day
- deletePlan: Delete plan
- getPlanActions: Retrieve action history

## 5. Exercises (Exercise Controller)

Functions:

- getAllExercises: Fetch all exercises from database

## 6. Utilities (Utils)

Functions:

- isValidUUID
- findWeek
- findDayByMuscleGroup
- exerciseExistsInDay
- addExerciseToDay
- deleteExerciseFromDay

## 7. API Endpoints

(Auth, Plan, Exercise)

Auth Endpoints:

POST /auth/register

POST /auth/login

DELETE /auth/delete/:userId

Plan Endpoints:

POST /plan/generate

GET /plan/user/:userId

GET /plan/planId/:planId

PATCH /plan/:planId/swap-exercise

# CoachGPT Pro - Full Backend Documentation

PATCH /plan/:planId/add-exercise

PATCH /plan/:planId/delete-exercise

DELETE /plan/:planId

GET /plan/:planId/actions

Exercise Endpoints:

GET /exercises

Body Parameters (if needed):

- POST /auth/register:

```
{ "name": "John", "email": "john@example.com", "password": "123456" }
```

- POST /auth/login:

```
{ "email": "john@example.com", "password": "123456" }
```

- POST /plan/generate:

```
{ "userId": "UUID", "goal": "hypertrophy", "daysPerWeek": 4, "difficultyLevel": "intermediate" }
```

- PATCH /plan/:planId/swap-exercise:

```
{ "currentExercise": "Push-Up", "newExercise": "Incline Push-Up", "weekNumber": 2 }
```

- PATCH /plan/:planId/add-exercise:

```
{ "weekNumber": 2, "muscleGroup": "chest", "newExercise": "Cable Crossover" }
```

- PATCH /plan/:planId/delete-exercise:

```
{ "weekNumber": 2, "muscleGroup": "chest", "exerciseToDelete": "Push-Up" }
```

## 8. Security Notes

- Use JWT\_SECRET
- Hash passwords with bcrypt
- Use helmet/cors

## 9. Future Improvements

- Protect routes with authentication
- Add pagination
- Admin dashboard for exercise management

# CoachGPT Pro - Full Backend Documentation

## 10. Final Note

CoachGPT Pro backend is complete and production ready!