Build a simple **Gym Workout Tracker** application where a **trainer** can log and manage workout activities for gym **members** (clients). The focus is on the trainer logging sessions for different members.

Core Entities & Fields:

- **Member** (simple representation of a gym client):
    - ID (auto-generated)
    - Name
    - Join Date (or optional fields like age/gender if time allows)
- **Workout Log** (the main tracked item):
    - ID (auto-generated)
    - Member ID (foreign key/reference to Member)
    - Date (when the session happened)
    - Exercise Name (e.g., 'Bench Press', 'Squats', 'Treadmill')
    - Sets (integer)
    - Reps (integer or per set — can be simple single value or string like '10/8/6')
    - Weight (in kg, optional/nullable)
    - Notes (optional text, e.g., 'good form', 'PR today')

Requirements:

- **Database (MySQL)**:
    - Create two tables: members and workout_logs (with foreign key from workout_logs to members).
    - Add some sample members (3–5) via SQL or in code for testing.
    - Ensure basic constraints (e.g., non-negative sets/reps, date validation if possible).
- **Backend (Java Spring Boot)**:
    - Use Spring Data JPA for entities/repositories.
    - Create RESTful API endpoints (focus on trainer perspective):
        - GET /api/members → list all members

        - GET /api/workouts?memberId={id} → get all workout logs for a specific member (sorted by date descending)
        - GET /api/workouts/{id} → get single log
        - POST /api/workouts → create new log (include memberId in request body)
        - PUT /api/workouts/{id} → update existing log
        - DELETE /api/workouts/{id}
    - Add basic validation (e.g., @NotBlank on exercise name, @Positive on sets/reps).
    - Optional: Simple endpoint GET /api/workouts/recent → last 10 logs across all members (for trainer dashboard feel).
- **Frontend (React)**:
    - Simple single-page app with tabs or sections:

1. Member list (dropdown or selectable cards) to choose which member's workouts to view/log.

2. Form to add a new workout log (select member, date picker/default today, inputs for exercise/sets/reps/weight/notes).

3. Table or list showing the selected member's workout history (columns: date, exercise, sets × reps × weight, notes; edit/delete buttons per row).

    - Use useState + useEffect to fetch members on load, fetch workouts when member selected.
    - Use Axios/Fetch for API calls.
    - Show loading/error states, basic success messages (e.g., "Workout logged!").
    - Minimal styling — focus on functionality.
- **Integration & Flow**:
    - Trainer selects a member → sees their past workouts → can add/edit/delete logs.
    - Ensure frontend communicates correctly with backend (handle CORS if needed).
    - Run both apps locally.

- **Time Limit & Priorities**:
    - You have **60 minutes**.
    - **Must-have**: CRUD for workout logs tied to members, member selection in UI, end-to-end working flow.
    - **Nice-to-have (if ahead of time)**: Sort/filter workouts by date/exercise, simple summary (e.g., total sessions per member), or pre-filled exercise suggestions (hardcoded array).

Think aloud, explain your design choices (e.g., why separate member table, how you handle foreign keys). Use any boilerplate we provide (Spring Boot starter + React create-app skeleton)."