

AI Virtual Coach – Lovable Frontend PRD & Backend API Contract

PART A — FRONTEND-FOCUSED PRD

1. Purpose

This document defines **only** the scope, screens, states, and assumptions needed for Lovable.dev to generate a **Flutter frontend** for the AI Virtual Coach MVP.

This is a **research demo / graduation project**, not a consumer fitness app.

2. Product Scope (Frontend Only)

In Scope:

- Camera-based workout recording (single exercise per session)
- Session lifecycle management (record → process → results)
- Text-only AI feedback visualization
- Error and retry handling

Out of Scope:

- User accounts / authentication
- Workout history
- Real-time feedback during exercise
- Video storage or playback
- Audio feedback
- Payments or subscriptions

3. Core User Flow

```
Home → Camera Setup → Recording → Processing → Results → End Session
```

The user performs **one exercise per session**.

4. Screens & Responsibilities

4.1 Home Screen

- App title and short description
 - "Start Workout" CTA
 - Optional disclaimer (research demo)
 - Style: Athletic, gym-inspired look, dark/light contrast, bold typography
-

4.2 Camera Setup Screen

- Camera preview (front or side view)
 - Instruction text ("Place phone X meters away")
 - View selection toggle (Front / Side)
 - "Start Recording" button
 - Style: Fitness/gym aesthetic, clear icons, motivating visuals
-

4.3 Recording Screen

- Live camera preview
- Recording indicator (timer or red dot)
- Minimal UI to avoid distraction
- "Finish Exercise" button
- Style: Coach-style interface, energetic colors, clear recording cues

Important:

- No pose visualization overlays
 - No video saved locally
-

4.4 Processing Screen

- Loading animation
 - Text: "Analyzing your exercise form"
 - Disable navigation
 - Style: Athletic theme, dynamic loading visuals, motivational text
-

4.5 Results / Feedback Screen

- Detected exercise name
- Overall form score (aggregated)
- Breakdown of 5 aspect scores (0-10)
- Textual feedback paragraphs

- "Retry Exercise" button
 - "End Session" button
 - Style: Gym/coaching theme, clear metrics display, motivational cues, bold typography
-

4.6 Error / Retry Screen

Displayed when:

- No reps detected
- Backend error
- Invalid input

Actions:

- Retry session
 - Return to Home
 - Style: Athletic style, clear warnings, energetic call-to-actions
-

5. UI & UX Guidelines

- Athletic, gym-inspired design
 - Bold typography, high contrast, energetic color accents
 - Motivational cues where appropriate
 - Clear loading and error states
 - Portrait orientation only
-

6. Frontend Technical Assumptions

- Built with **Flutter**
- Camera access available
- Internet connection available
- Backend APIs already exist
- Pose extraction handled locally or abstracted

Lovable **must not** implement pose extraction logic — only manage recording lifecycle and API calls.

PART B — BACKEND API SCHEMA (ASSUMED BY FRONTEND)

1. Session Submission

POST /api/session/analyze

Request Body:

```
{  
  "exercise_view": "front" || "side",  
  "pose_sequence": [[[x,y,z], ... 33 landmarks], ...],  
  "metadata": {  
    "fps": 30,  
    "device": "mobile"  
  }  
}
```

2. Successful Response

```
{  
  "exercise": "squat",  
  "reps_detected": 12,  
  "scores": {  
    "aspect_1": 7.5,  
    "aspect_2": 8.0,  
    "aspect_3": 6.5,  
    "aspect_4": 7.0,  
    "aspect_5": 8.2  
  },  
  "overall_score": 7.4,  
  "feedback": [  
    "Your squat depth was consistent across reps.",  
    "Maintain a more upright torso to reduce forward lean.",  
    "Knee alignment was generally good, with slight inward movement on later  
    reps."  
  ]  
}
```

3. Error Response

```
{  
  "error_code": "NO_REPS_DETECTED",  
  "message": "No valid repetitions were detected. Please try again."  
}
```

4. Frontend Error Handling Rules

- Show retry screen on non-200 responses
 - Display backend error message verbatim
 - Allow user to restart session
-

FINAL NOTES

- This document intentionally hides AI implementation details
- Backend remains modular and replaceable
- Frontend is future-ready for real-time feedback extensions
- UI should be **athletic, gym-inspired, and coach-style** rather than academic style