# Technical Specification: VBT Demo React App (No Auth)

## High-level summary

Build a React single-page demo where a user records or uploads a side-view barbell video, the app detects/tracks the bar end frame-by-frame, computes mean & peak concentric velocity, overlays the bar path and velocity box on the video, and allows save/delete of analyzed videos.

## Reference repos / resources

- VBT Barbell Tracker (OpenCV/Python): https://github.com/kostecky/VBT-Barbell-Tracker
- Roboflow Barbell Tracking Model:
https://universe.roboflow.com/vjeko-tracking/barbell-tracker/model/5
- Roboflow Supervision: https://github.com/roboflow/supervision
- KLT Barbell Auto-tracking (validated study):
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0263224
- OpenCV Optical Flow (KLT): https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html

## Architecture & Flow

- Frontend: React SPA (no backend auth).
- Video capture/upload → Frame extraction → Detection/tracking (Roboflow OR KLT optical-flow) → Calibration (plate diameter/manual) → Velocity computation → Overlay → Save/Delete.

## UI Spec

- CameraRecorder (record via MediaRecorder).
- Uploader (upload mp4/webm).
- VideoPlayerWithOverlay (draw bar path + bounding box + velocities).
- AnalysisPanel (show mean/peak).
- SavedList (IndexedDB save/delete).

## Tasks for Replit Agent

1. Scaffold React app (Vite+TS). Add ESLint/Prettier.
2. CameraRecorder + Uploader.
3. Frame extraction (requestVideoFrameCallback).
4. Tracking pipeline:
- Option A: Roboflow detection + Supervision ByteTrack.
- Option B: OpenCV.js KLT optical-flow (Shi-Tomasi corners + calcOpticalFlowPyrLK).
5. Calibration: auto (plate detection) or manual (user line).
6. Velocity math: central difference, identify concentric phase, compute mean & peak.
7. Overlay rendering: bar path + bounding box + floating card with mean/peak.
8. Save/delete analyzed videos (IndexedDB).
9. Export analyzed video (canvas + MediaRecorder or ffmpeg.wasm).

## File structure

/src/components: CameraRecorder.tsx, Uploader.tsx, VideoPlayerWithOverlay.tsx, AnalysisPanel.tsx, SavedList.tsx
/src/workers: cv-worker.ts (OpenCV.js), inference-worker.ts (Roboflow)
/src/utils: calibration.ts, velocity.ts, klt.ts, roboflow.ts

## Acceptance Criteria

- Record/upload video works.
- Analysis produces tracked points and overlays bar path.
- Mean/peak concentric velocity shown in UI.
- Export video with overlay works.
- Save/delete analyzed videos persists via IndexedDB.

## Notes

- Camera must be side-on for accuracy.
- Calibration with known plate diameter required for metric conversion.
- Demo: offline KLT pipeline as default. Roboflow optional with API key.