SecondHand

Learn physical skills from invisible teachers

Core loop: Expert skill clip → keypoint extraction → **real-time alignment + “ghost” AR overlay** → **instant correction feedback** → optional **voice coach** (NLP) that explains fixes in plain language

---

0) Executive Summary

Most physical-skill learning is broken because it’s taught like language.

- Traditional instruction is verbal: “rotate your wrist,” “keep your elbows in,” “relax your fingers.”
- But the learner’s problem is *visual + motor*: “What do I look like relative to correct form, right now?”

SecondHand turns expert demonstrations into an **invisible coach** that overlays directly onto your body/hands in real time. Instead of interpreting instructions, you **match**.

The product is its own proof:

A judge steps in front of a webcam → a ghost overlay appears → they try to align → the system highlights drift and gives instant correction → their score jumps in seconds.

SecondHand wins because it hits the rare intersection:

- **Demo-first magic** (obvious in 5 seconds)
- **Real utility** (skill learning, training, accessibility, rehab)
- **Hard tech that feels human** (CV + alignment + feedback + voice coaching)

---

1) The Problem: Why this is real (and why most “learning apps” fail)

Physical skills are hard for the same structural reasons across domains (piano, CPR, sports form, sign language, dance, rehab):

1.1 The Mirror Gap

Learners can’t see themselves the way the instructor does.
Even with a mirror, the mapping is cognitively heavy.

1.2 The Timing Gap

Watching an expert video doesn’t solve *phase*: you’re always early/late, faster/slower, missing micro-transitions.

1.3 The Correction Gap

YouTube gives no feedback. Coaches can’t be present 24/7. Most practice is “blind repetition.”

1.4 The Cognitive Overload Gap

Verbal instruction is lossy:

- You translate words → mental model → movement
- But bodies learn best through **visual imitation + immediate correction**

What learners actually need:

- A **visual template** on top of their motion
- A way to know **where** they deviate (which joint/angle/path)
- A way to know **when** they deviate (timing)
- A way to practice with feedback **without** a live coach

SecondHand creates a new category: alignment-based instruction.

2) The Solution: What SecondHand is

2.1 Core concept

SecondHand is an invisible teacher that:

- captures motion in real time (camera)
- extracts pose/hand keypoints (computer vision)
- overlays a semi-transparent expert “ghost”
- computes alignment continuously (spatial + temporal)
- generates intuitive correction cues (visual + voice)

2.2 What it feels like

Instruction becomes *physical alignment*, not interpretation:

- “put your hand inside the ghost hand”
- “match this pose”
- “follow this path”
- “sync to this timing”

SecondHand turns practice into a simple game:

align → feedback → improve → repeat

2.3 The real-time “Voice Coach” (NLP) layer

SecondHand uses voice in two ways that directly strengthen the core product:

A) Voice control (reliability + speed)

- “Start lesson”
- “Loop that part”
- “Slow the ghost”
- “Show my mistakes”
- “Switch to Pack: CPR”
- “Explain what I’m doing wrong”

B) Voice coaching (NLP “translator” from geometry → human cues)
The system converts raw deviations into short, human corrections:

- “Open your fingers”
- “Rotate wrist slightly left”
- “Raise your elbow”
- “You’re rushing the transition—slow down”

This is not a chatbot layer.
It’s an interface layer that makes the correction engine feel like a real teacher.

---

3) Product Modes (User-facing)

Mode A — Follow (live guidance)

- ghost plays in real time
- user matches it live
- errors highlight instantly
- voice coach optionally calls out micro-corrections

Mode B — Practice (ghost fades)

- ghost starts strong, then gradually fades
- user performs from memory
- system scores and pinpoints mistakes
- voice coach summarizes: “Your wrist angle is consistent; your finger spread collapses on beat 3.”

Mode C — Loop & Master (micro-drills)

- choose a 2–6 second “micro-move”
- loop it until alignment stabilizes
- score trend proves improvement

This mode is the highest leverage for a hackathon MVP because it creates a visible “improvement arc” in under a minute.

Mode D — Ask (voice Q&A, super tight)

- user: “What’s wrong?”
- system: generates 1–2 actionable cues, not paragraphs
  This is how you add “NLP” without becoming cringe or vague.

---

4) Skill Packs (4–5 packs in the product plan)

SecondHand ships as **packs** so it can be multi-skill without becoming chaotic.
Each pack has: expert clips + extracted keypoints + segment markers + cue templates + voice scripts.

Pack 1 — Sign Language (Hands-only)

- best for: accessibility + crystal clarity
- why it works: hands-only, minimal occlusion, no equipment
- demo value: instant “aww” + utility

Pack 2 — CPR Form (Serious utility)

- best for: posture + safety relevance
- tracks: hand placement, elbow lock, shoulder stacking, compression rhythm
- demo value: “this is real life”

Pack 3 — Piano Technique (Air piano / fingering drills)

- best for: complex hand articulation
- tracks: finger curl, wrist stability, posture transitions
- demo value: high skill aura

Pack 4 — Sports Micro-Form (choose one: basketball shot pocket / golf takeaway / tennis serve prep)

- best for: mainstream appeal
- tracks: key checkpoints (elbow position, torso angle, wrist set)
- demo value: familiar and easy for judges to try

Pack 5 — Rehab/PT (range-of-motion + mobility)

- best for: long-term product legitimacy
- tracks: controlled movement ranges, posture correctness
- demo value: serious and sponsor-friendly

Important principle:
For McHacks, you demo **one pack perfectly**, but the product story shows expandability.

---

5) MVP Scope for McHacks 13 (what actually gets built)

Even if the “plan” includes packs, the hackathon build must be ruthless:

MVP = One pack + one perfect interaction

- pick **one pack** (most likely Sign Language or CPR)
- 1–2 expert clips
- 2–3 loopable segments
- flawless ghost overlay + alignment score + drift highlighting
- voice control for loop/explain (optional but high impact)

Winning demo recipe:

- a judge tries it in 10 seconds
- score moves visibly
- the system calls out one correction
- judge fixes it immediately
- room reacts

---

6) System Architecture (Layers)

Think of SecondHand as **8 layers**:

1. **Experience Layer** — UI/UX + demo flow
2. **Capture Layer** — camera + audio capture
3. **Perception Layer** — keypoints for hands/body
4. **Alignment Layer** — spatial + temporal alignment (the “magic”)
5. **Feedback Layer** — ghost rendering + highlighting + scoring
6. **Coach Layer** — voice control + NLP correction generator
7. **Content Layer** — packs, preprocessing, metadata
8. **Infrastructure Layer** — deployment, reliability, analytics

---

7) Detailed Design + Tech Stack (by layer)

7.1 Experience Layer (UI/UX)

Goal: 5-second comprehension. Zero friction.

Key screens

- Landing: choose pack → start camera
- Calibration: “hold hands up” / “stand in frame”
- Session: ghost overlay + score + cues + loop control
- Recap: best attempt + “top 2 fixes”

UI elements

- live camera feed (mirrored option)
- translucent ghost skeleton + optional silhouette
- alignment meter (0–100)
- joint heat map / glow on key joints
- loop bar for micro-drills
- one-line cues (visual + optional voice)

Stack

- Next.js (or Vite + React)
- TailwindCSS
- Canvas/WebGL overlay (Three.js optional; Canvas2D fine for MVP)
- Zustand (or simple React state) for speed

7.2 Capture Layer (Camera + Audio)

Goal: stable low-latency pipeline.

Camera

- browser getUserMedia()
- lock framerate ~30 FPS (best effort)
- mirrored view toggle (important for learning)
- lighting warnings (“too dark”, “backlit”)

Audio

- Web Audio API (mic input)
- used for:
    - voice commands
    - optional rhythm/metronome cues (CPR pack)
    - optional “clap to start” fallback

Stack

- Web APIs only (fast, reliable)

7.3 Perception Layer (Pose/Hands)

Goal: keypoints reliably and fast.

Hackathon recommendation: run inference **in-browser** to kill latency and avoid network failure.

Hands

- MediaPipe Hands (21 keypoints per hand)

Optional body

- MediaPipe Pose (33 keypoints) for CPR/sports/posture

Stack

- MediaPipe (JS)
- fallback: TensorFlow.js

Reliability tricks

- confidence gating (ignore low-confidence points)
- temporal smoothing to reduce jitter
- “occlusion-aware” cues: if tracking drops, prompt user to adjust

---

7.4 Alignment Layer (the “winner engine”)

This is what makes SecondHand not “just overlaying a video.”

You need:

- **Spatial alignment**: ghost sticks to the user correctly
- **Temporal alignment**: guidance feels synchronized, not drifting

7.4.1 Spatial alignment (MVP)

Goal: map expert skeleton to user skeleton coordinates.

Approach:

- normalize both to a canonical coordinate space
- scale using stable reference lengths:
  - hands: wrist → middle fingertip
  - body: shoulder width / torso length
- rigid transform (translation + scale + optional rotation)
- anchor points:
  - hands: wrist + MCP knuckles
  - body: hips/shoulders depending on pack

Output:

- ghost skeleton positioned over the user with consistent stickiness

Stretch: Procrustes alignment for best-fit mapping (fast and robust)

7.4.2 Temporal alignment (MVP)

You can win without full DTW if feedback is strong.

MVP:

- play expert clip at normal speed
- compute similarity frame-by-frame
- if user lags/leads, show:
  - “slow down” / “speed up”
  - optional ghost slow-down if user is significantly behind
- loop segments for mastery

Stretch:

- phase estimation / DTW-lite:
  - map user motion phase → expert phase
  - ghost “phase-locks” to user tempo (feels like adaptive coaching)

7.5 Feedback Layer (Overlay + Corrections)

Goal: make correction feel physical, not analytical.

Ghost rendering

- expert skeleton (lines/joints)
- optional faint silhouette mask (super “AR” feel)
- directional arrows on joints with biggest drift

Scoring

- per-frame similarity score (0–100)
- smoothed using EMA (avoids UI jitter)
- trendline for loop mode (“attempt 1 → attempt 2 improvement”)

Error highlighting

- joint glow intensity proportional to error magnitude
- show only top-k problem joints (avoid visual spam)

7.6 Coach Layer (Voice + NLP)

This layer exists for ONE reason:

> turn numeric drift into human coaching instantly

It has two subsystems:

A) Voice Commands (fast, reliable)

- “start / stop”
- “loop”
- “next / previous”
- “slow”
- “explain”
- “switch pack”

Implementation options

- browser SpeechRecognition (fast but inconsistent)
- fallback: push-to-talk with a small STT call

B) NLP Coach (short, actionable)

Input:

- error vectors (joint deltas, angle diffs, timing lag)
- pack context (sign language vs CPR)
- optionally a short user question (“what am I doing wrong?”)

Output:

- 1–2 micro-cues max
- never paragraphs
- never moralizing

Two-tier approach (best for hackathons)
Tier 1 (always-on, deterministic): rule-based cue mapping
Tier 2 (optional “wow”): LLM rewrite into natural teacher voice

So even if the LLM fails, the coach still works.

Stack

- deterministic mapping in TypeScript (fast)
- optional LLM via API for phrasing polish (only if stable)
- optional STT:
  - ○ Whisper API (robust) or any reliable speech-to-text
  - ○ for hackathon, keep it minimal

7.7 Content Layer (Packs + Preprocessing)

This layer makes the product scalable.

Each lesson stores:

- expert keypoints per frame (JSON)
- optional reference video
- segment markers (loop zones)
- cue mapping templates (pack-specific)
- difficulty tags

How lessons are created

- record expert clip
- run preprocessing script once:
    - extract keypoints
    - smooth + normalize
    - store as JSON
- ship JSON with the app (fastest, most reliable)

Stretch:

- “Upload-to-Lesson”: user uploads any clip, pipeline generates a new lesson

---

7.8 Infrastructure Layer (Deployment + Reliability)

Hackathon goal: **demo cannot fail**.

MVP infra

- frontend on Vercel/Netlify
- no backend required
- expert keypoints bundled in repo or served via CDN

Optional backend (stretch)

- FastAPI service for:
    - uploads
    - preprocessing jobs
    - storing generated lessons
- storage: Cloudflare R2 / S3
- websockets for real-time job status

Telemetry:

- simple analytics (session starts, completion, average score improvement)
- purely for demo storytelling (not creepy tracking)

---

8) Feature Catalog (What SecondHand becomes)

8.1 Core features (MVP+)

- live ghost overlay with stable alignment
- alignment score + improvement trend
- drift highlighting (top joints)
- loop mode micro-drills
- voice commands (loop, explain)
- one pack fully polished

8.2 Advanced features (stretch)

- adaptive phase-lock ghost
- multi-angle camera support
- ghost fade-out training (“training wheels”)
- replay mode (you vs expert side-by-side)
- auto-detect “hard moments” → generate loops
- accessibility cues: audio tone feedback; colorblind-safe palette
- live remote coach: expert becomes ghost in real time (WebRTC)
- upload-to-lesson pipeline

---

9) Algorithmic Core (technical)

9.1 Keypoint representation

Hands: 21 keypoints/hand
Pose: 33 keypoints (optional)

9.2 Normalization

- convert pixel coordinates → normalized screen space
- scale normalization:
  - hand: wrist→middle_tip
  - body: shoulder width / hip width
- confidence gating: ignore low-confidence joints

9.3 Similarity metric (per frame)

Compute weighted error:

- positional error per joint
- angle error (finger curl, wrist rotation, elbow lock, etc.)
- time lag penalty (if applicable)
- confidence weighting

Score:

- score = clamp(100 - k * error, 0, 100)
- smooth score with EMA for stability

9.4 Cue mapping (geometry → language)

Map error patterns to cues:

- finger spread small → “open fingers”
- wrist rotated → “rotate wrist left/right”
- elbow bent (CPR) → “lock elbows”
- hand too high/low → “raise/lower hand”
- lagging/leading → “slow down / speed up”

This is the secret: **small rule-mapping makes it feel intelligent**.

9.5 NLP coaching (optional tier)

Take the deterministic cue + context and generate:

- a friendly phrasing
- a single next step
- no lecture

---

10) Hackathon Build Plan (concrete, engineered to win)

Pre-hack (if allowed)

- choose pack (Sign Language or CPR)
- record 1–2 expert clips in excellent lighting
- preprocess keypoints → JSON
- define 2–3 loop segments (“easy/medium/hard”)

Day 1

1. camera + MediaPipe keypoints stable
2. draw user skeleton overlay
3. load expert keypoints and render ghost skeleton (no alignment yet)

Day 2 (morning)

4. implement spatial alignment (anchors + scaling)
5. per-frame similarity score + EMA smoothing
6. drift highlighting (top joints)

Day 2 (afternoon)

7. loop mode (micro-drills) + improvement trend
8. deterministic cue mapping (2–5 cues max)
9. optional voice commands (“loop”, “explain”) with fallback UI buttons

Final 4 hours

10. polish visuals (ghost aesthetic matters)
11. rehearse demo 20+ times
12. bring lighting fix (tiny ring light) for stage reliability

---

11) Demo Script (winning choreography)

0–5 seconds
"This is **SecondHand**. It teaches physical skills without words — by overlaying an expert's motion onto your body."

5–15 seconds
Judge steps up. Camera opens. Ghost appears.

15–35 seconds
Judge attempts. System highlights one mistake.
Voice coach: "Open fingers slightly."
Judge fixes. Score jumps.

35–55 seconds
Loop a 3-second segment. Show improvement trend.

Close
"This scales into packs: sign language, CPR, piano technique, sports form, rehab — anything where form matters."

The point is not "look what we built."
The point is: **the judge felt it work.**

---

12) Risks & Mitigations (honest, but engineered)

Risk: jitter/occlusion breaks the illusion

Mitigation:

- smoothing + confidence gating
- pick clips that avoid self-occlusion
- one controlled lighting setup

Risk: “this is just overlaying a video”

Mitigation:

- make scoring + drift highlights unmistakable
- show instant improvement via loop mode
- make the ghost “stick” precisely (alignment layer must feel magical)

Risk: multi-pack scope creep

Mitigation:

- demo ONE pack perfectly
- packs exist in the plan, not the MVP

Risk: voice layer becomes gimmicky

Mitigation:

- voice control + cue explanation only
- deterministic cues first, LLM phrasing optional
- never long-form chat

---

13) Why SecondHand wins McHacks specifically

SecondHand is one of the rare hacks that wins on:

- **technical difficulty** (real-time CV + alignment + feedback)
- **design** (beautiful and legible)
- **utility** (learning/training/accessibility)
- **stage impact** (judge tries it instantly)
- **expandability** (packs + upload-to-lesson + remote coach)

It is not a “tool.”
It’s a **new interface for learning movement**.

Appendix A — Recommended tech stack (final proposal)

Frontend

- Next.js + React + Tailwind
- Canvas2D or Three.js (only if time)
- Zustand

CV

- MediaPipe Hands (+ Pose if needed)
- smoothing + confidence gating

Audio/Voice

- Web Audio API
- SpeechRecognition or push-to-talk STT
- deterministic cue mapping always-on
- optional LLM for phrasing polish

Backend (optional)

- FastAPI
- R2/S3 storage for uploads
- background job runner (simple queue)

Deployment

- Vercel for frontend
- avoid backend dependence for demo

Appendix B — The “packs” truth

You can *say* 5 packs in the pitch.
You must *build* 1 pack for the win.
That’s how you look visionary without dying in scope.