AI Engineer Assessment — KidSafe Alphabet Tutor (Vision + Speech)

Build a kid‑friendly, speech‑first learning agent that teaches the English alphabet (A–Z) using a conversational voice interface and optional on‑device vision (webcam) to recognize letters or objects. The agent listens to the child, responds with a playful TTS voice, gives corrective feedback on pronunciation, and adapts to the last three conversational turns. All evaluation will be performed through a visible UI.

# Goals

1) Speech-to-Speech loop: child speaks, agent understands, and talks back naturally.

2) Vision assist (optional but strongly recommended): detect printed letters or common objects and tie them to letters (e.g., “B is for ball”).

3) Phonics-focused: emphasize letter names and sounds, with pronunciation feedback.

4) Adaptive short-term memory: remember the last 3 user/assistant exchanges to personalize prompts (e.g., child’s name, letter of focus, difficulty).

5) Safety & privacy: child-safe content, no ads, no data leaks; provide parental controls.

6) Low latency & streaming: near real-time mic capture, incremental ASR, and streaming TTS.

# UI Requirements (Used for Testing)

Provide a runnable web UI (React/Next.js/Streamlit/etc.). Minimum components:

- Mic Button: push-to-talk and auto VAD mode toggle; live transcription field (streaming if supported).

- Avatar Bubble: animated face or character lip-syncing/animating while speaking.

- Camera Panel: webcam preview with a “Recognize” button; display the detected letter/object + confidence.

- Lesson Card: shows current target letter, sound (“/b/”), example word(s), and a simple activity prompt.

- Progress Strip: badges/stars for “heard correctly,” “needs practice,” and session streak.

- Memory Panel (read-only): last 3 pairs (U/A) and derived settings (name: “Lina”, focus: “B”, difficulty: “easy”).

- Settings (parental gate): mute camera, clear memory, toggle offline/STT model, language/accent (e.g., en‑US, en‑GB), volume, speech rate.

- Status & Errors: non-intrusive toasts (“listening…”, “thinking…”, “speaking…”, “camera blocked”, “network offline”).

# Functional Requirements

Speech & Understanding

- ASR: recognize child speech for short utterances (1–5s); show live transcript; handle mispronunciations.

- Phoneme/Pronunciation Feedback: detect common mispronunciations and provide gentle corrective prompts.

- Intent: detect letter requests (“Teach me B”), answers (“/b/ like ball”), and questions.

- TTS: kid-friendly, warm tone; adjustable speed; stream audio as tokens arrive.

Vision (Strongly Recommended)

- Detect printed capital letters held up to the camera (A–Z).

- (Bonus) Detect common objects and map to starting letter (ball, cat, apple).

Curriculum & Flow

- Core lessons for A–Z with: letter name, phoneme, 2–3 example words, and a micro-activity.

- Micro-activities: repeat-after-me, find-an-object, choose-the-sound (multiple choice via voice), show-the-letter (vision).

- Personalization from last 3 turns (e.g., if child struggled with /th/, suggest “T” or “H” practice next).

Memory (3-turn rolling)

- Maintain last 3 user+assistant pairs per session; latest wins on conflicts.

- Derive simple state: child\_name (if told), current\_letter, difficulty\_hint, last\_mistake.

Safety & Privacy

- COPPA-style posture: no ads, no external sharing; avoid storing voice/video by default.

- Parental gate (e.g., math puzzle) to access Settings.

- Content moderation: block unsafe words; never ask for PII.

- Camera and mic permissions explicit; provide camera-off mode.

Performance

- Target end-to-end < 1.2s first audible response on happy path.

- Graceful degradation: if ASR/TTS/vision unavailable, show clear fallback and continue lesson textually.

# Architecture Expectations

- Frontend: web app with mic + (optional) camera; streaming UI for ASR and TTS; avatar animation tied to speech events.

- Backend: ASR, NLU, policy/lesson engine, TTS; optional on-device models or server inference.

- State: per-session memory buffer (3 pairs) + lightweight profile (current letter/streak).

- Content Store: A–Z lesson JSON with phonics, words, and prompts.

- Telemetry: anonymized event logs (consent-gated) for states: listening, recognized, intent, lesson step, error, latency metrics.

# Acceptance Tests (Performed in the UI)

1) Basic Speech Loop

- Child says: “Teach me B.”

- UI shows streaming transcript; agent replies within ~1.2s: “B says /b/ like ball. Can you say /b/?”

- Award a star when child repeats correctly; visible in Progress Strip.

2) Mispronunciation Coaching

- Tester says “/p/” for “B” on purpose.

- Agent detects mismatch and responds kindly: “Close! Try /b/ — press your lips together and voice it: /b/.” Retry loop visible.

3) Vision Letter Recognition

- Tester shows a printed “C” to camera and clicks Recognize.

- UI shows “Detected: C (0.94)” and agent adapts: “Great! C says /k/ like ‘cat’. Can you find something that starts with C?”

4) 3-Turn Memory Personalization

- Turn 1: “My name is Zain.” (assistant: “Nice to meet you, Zain!”)

- Turn 2: “I want A.” (assistant teaches A)

- Turn 3: “What next?”

- Agent proposes B; Memory Panel displays last 3 pairs + derived state (name: Zain, current\_letter: A→B).

5) Safety & Settings

- Try to open Settings: parental gate prompts and must be solved to proceed.

- Toggle camera off; attempt Recognize; app shows clear “camera disabled” notice.

6) Latency & Resilience

- Simulate offline ASR; app falls back to on-device or text entry with a banner.

- Streaming TTS visibly starts before full text is generated (or progressive status toasts).

# Scoring Rubric (100 pts)

UI/UX for Kids (20)

- Clear, warm visuals; big controls; readable; avatar lip/mouth animation during TTS.

Speech Recognition & Coaching (20)

- Robust to kid speech; actionable feedback; correct phoneme guidance.

Vision Integration (15)

- Accurate printed-letter detection; clean confidence display; activity linkage.

Curriculum & Flow (15)

- Phonics accuracy; varied micro-activities; positive reinforcement; star/sticker system.

Memory & Personalization (10)

- Correct 3-turn buffer; state derivation influences prompts; visible Memory Panel.

Safety & Privacy (10)

- Parental gate; no PII; permissions clarity; camera/mic off modes.

Performance & Streaming (5)

- Sub-1.2s first audible response (happy path) or clear progressive states.

Resilience & DX (5)

- Fallbacks; error toasts; README clarity; reproducible setup.

# Deliverables

- Source code with README (run instructions), Dockerfile, Makefile, .env.example.

- UI includes: Mic button, Avatar, Camera panel, Lesson card, Progress strip, Memory panel, Settings with parental gate.

- A–Z lesson JSON (example words + phonics).

- Demo video (≤5 min) showing all Acceptance Tests in the UI.

- Notes: ASR/TTS choices (cloud/on-device), phoneme detection method, privacy posture, and latency measurements.

# API (Optional)

- `POST /talk`: returns assistant transcript + audio stream URL/chunks; supports `session\_id` and returns updated memory snapshot.

- `POST /vision/letter`: returns detected letter + confidence.

- `GET /lesson/current`: returns current letter and activity.

Evaluators grade via the UI, but API helps modularity.

# Anti-Cheat & Review Notes

- No hardcoded answers; we’ll test multiple letters and intentional mispronunciations.

- We will cover camera on/off, Settings parental gate, and latency perception.

- We will check that memory truly influences the next prompt.