ai_interviewer_prd

AI Interviewer Webapp PRD

TL;DR

A Next.js-based AI interviewer webapp designed to help students and junior professionals practice coding interviews with real-time voice/text interactions and receive automated, actionable feedback. Recruiters can generate unique interview links, assess candidates at scale, and review results via an admin dashboard. The system utilizes Supabase/MongoDB for data, Vapi/ElevenLabs for voice, and n8n for orchestrating question generation, scoring, and feedback delivery.

Goals

Business Goals

- Drive user acquisition among students, job-seekers, and recruiting organizations.
- Increase recruiter engagement by enabling scalable, consistent candidate evaluation.
- Establish a differentiated, sticky product experience that encourages repeat usage.
- Collect anonymized performance data to enable continuous improvement of question sets and scoring systems.
- Facilitate verified candidate testing to position the platform as a trusted assessment tool in tech hiring.

User Goals

- **Students/Juniors**: Practice real coding interviews, receive detailed, actionable feedback, and effortlessly track improvement over time.
- **Recruiters**: Generate and manage assessment links, review structured candidate results with insights, and make hiring decisions faster.
- Share results and achievements easily, boosting user motivation and platform virality.
- Experience a fast, intuitive, and immersive interview simulation across all devices.

Non-Goals

- Manual interview evaluation or human scoring outside of the automated system.
- Standalone text-only chat interview experience (voice and/or text input only as needed for accessibility).
- Expansion to non-technical domains not defined in platform scope (e.g., sales, HR, marketing interviews).

User Stories

Personas

Student/Junior

- As a student, I want to log in and choose a coding domain and difficulty to practice interviews relevant to my learning path.
- As a junior developer, I want to receive instant, actionable feedback on my interview performance so I can identify strengths and weaknesses.
- As a user, I want the option to retry the same interview or select a new one so I can measure my progress.
- As an applicant, I want to share my best interview results to LinkedIn or with recruiters to stand out in job searches.

Recruiter / Company Representative

- As a recruiter, I want to authenticate and access a dashboard where I can create new interview links tailored to roles.
- As a recruiter, I want to distribute unique, expirable interview links to candidates without manual coordination.
- As a company admin, I want to quickly review candidates' responses, scores, and per-question feedback to make quick hiring decisions.
- As a recruiter, I want to compare results across candidates for a given interview or job requisition.

Functional Requirements

Authentication & Authorization (Priority: High)

- **Student/Junior Auth**: Magic Link (Supabase), OAuth (Google/GitHub).
- **Recruiter/Company Auth**: Secure login, with role-based access (admin dashboard access and interview management).

Interview Creation & Parameters (Priority: High)

- Domain and difficulty selection (Python, JS, React, DevOps, etc.; Beginner/Intermediate/Advanced).
- (Optional) Custom question count or time limit.
- Recruiter-initiated interview link generation with expiration controls.

AI Interview Orchestration (Priority: High)

- Front-end triggers parameters for interview, sent to backend (API route).
- Integration with n8n webhook for prompt orchestration:
  - If using Gemini: Generate novel questions via LLM.
  - Else: Pull questions from curated database for standardized assessments.
- Unified data structure: `{question, domain, difficulty, ideal_answer}`.

Interview Experience (Priority: High)

- Voice (Vapi: two-way interaction) or fallback to TTS (ElevenLabs: interviewers' voice, user via text).
- Present questions sequentially; collect user answers via voice or text (accessibility fallback).
- Immediate storage/streaming of responses to backend for reliability.

Scoring & Feedback (Priority: High)

- Automated LLM scoring: Compare user responses to ideal answer using Gemini or OpenAI.
- Rule-based metrics: Keyword match, sentiment analysis, communication, fluency, and completeness.
- Feedback categorized: Accuracy, Communication, Confidence, Completeness, Overall.
- Option for detailed per-question feedback and holistic summary, delivered at end of session.

Database & Persistence (Priority: High)

- Store UserID/InterviewID, questions, answers, full score breakdown, timestamps.
- Data export/download for recruiter accounts.

Result Display & Post-Interview Actions (Priority: Medium)

- User sees: Total score, detailed per-question feedback, actionable improvement tips.
- Post-interview options: Retry (same/different parameters), share results (LinkedIn, email, unique link).
- Recruiter dashboard: Filter, search, and export candidate results; view linked interviews.

Admin & Security (Priority: Medium)

- Expiration logic for recruiter-generated links.
- Role-based access in dashboard/API endpoints.
- Audit trail of interview access/activity (GDPR-log for compliance).

Future Enhancements (Out of Current Scope)

- Invite system/referrals, in-depth analytics for users, custom company branding, LMS integrations.

User Experience

Entry Point & First-Time User Experience

- Users land on a modern, visually appealing home page highlighting two entry points: "Practice Interview" (for Students/Juniors) and "Recruiter Login".
- Clear onboarding for new users: Guided walkthrough showcases app value, voice interaction tips, and privacy assurances.
- Minimal, mobile-first registration for students (magic link/email or OAuth). Recruiters see a secure login form with option to request company access.

Student/Junior Journey

Step 1: Select domain, difficulty, and optional time/questions.

- Intuitive dropdowns/radio buttons, error checks for invalid selections.
- Accessible: Keyboard and screen-reader friendly.

Step 2: Begin interview in immersive voice or text-driven interface.

- Countdown and prompts; microphone permissions handled gracefully.
- Progress bar shows interview status; option to pause/cancel.
- Automatic save-on-error/disconnect.

Step 3: Complete interview; answers processed instantly.

Step 4: Results dashboard:

- See total score + category breakdown.
- Per-question analysis with actionable, AI-generated tips.
- Clear call-to-action: retry, choose another category, or share results (with preview).

Step 5: Access interview history and progress tracking (optional, by account).

Recruiter/Company Journey

Step 1: Login to recruiter dashboard with secure authentication.

Step 2: Generate new interview (domain, difficulty, parameters), get shareable, expiring link for candidate.

Step 3: Monitor status: See when links are used, interviews started/completed.

Step 4: View candidate submissions by position/link; structured feedback and per-question details.

Step 5: Export/download candidate results and add notes (advanced).

Step 6: Log out, with all data access and role security enforced.

Advanced Features & Edge Cases

- Voice engine fallback: If Vapi fails/unsupported, switch to ElevenLabs TTS or text.
- Session timeouts, abandoned/cancelled interviews: Save user progress and allow resumption.
- Accessibility: High-contrast mode, adjustable audio, screen reader announcements.
- Error dialogs: Clear, friendly messaging for network/API/voice errors, prompt retries.

UI/UX Highlights

- Consistent, accessible color scheme and iconography.
- Responsive design for mobile and desktop.
- Minimal distractions in interview mode; focus on usability and reducing anxiety.

- Clear status feedback and seamless transitions between stages.
- Privacy-by-design: Personal data clearly explained, with options for deletion.

Narrative

Aman, a final-year computer science student, is preparing for technical interviews but finds conventional prep resources unengaging and opaque. On a friend's recommendation, Aman signs up for the AI Interviewer webapp. After a quick onboarding, Aman chooses "Python," selects "Intermediate," and starts an interview. Each question is delivered by a realistic AI voice. Aman responds out loud, emulating a real interview scenario.

At the end, Aman instantly receives detailed feedback on not just accuracy, but also how clearly, confidently, and completely each response was delivered—plus targeted tips on how to improve. Impressed by his score, Aman shares it to LinkedIn and challenges classmates to beat it.

Meanwhile, at a startup hiring for JavaScript devs, recruiter Priya creates unique interview links and sends them to applicants. Within hours, she receives complete, automatically scored results in her dashboard—no manual grading required. Priya filters candidates by score, reviews their communication breakdowns, and fast-tracks top talent for in-person interviews, saving hours of time while improving hiring quality.

Success Metrics

User-Centric Metrics

- Number of completed student interviews per week.
- User feedback scores on clarity and helpfulness of AI feedback.
- Repeat usage rate (% of users going through multiple interviews).
- Result shares (to LinkedIn/email).

Business Metrics

- Recruiter account signups and active dashboard usage.
- Number of candidate assessments initiated and completed by recruiters.
- Adoption velocity among target companies/universities.

Technical Metrics

- Median latency from interview completion to feedback display (<8 seconds).
- API uptime (>99.5%).
- Voice/Text answer transcription accuracy.

Tracking Plan

- Interview started/abandoned/completed events.
- Parameter selection logs (domain, difficulty).

- Feedback view/click events.
- Result sharing actions.
- Recruiter login, link creation, and candidate view events.

Technical Considerations

Technical Needs

- **Frontend**: Next.js app in `/app`—modern, accessible, real-time interaction components.
- **API Layer**: `/api` for routing: Vapi/ElevenLabs, supabase auth, data endpoints, recruiter dashboard APIs.
- **n8n Integration**: Webhook orchestration for interview flows, question/feedback generation, scoring triggers in `/ai`.
- **Data Layer**: MongoDB or Supabase, schema in `/db`. Store users, interview data, scoring metrics, recruiter organizations.

Integration Points

- **Vapi**: Two-way voice.
- **ElevenLabs**: Text-to-speech fallback.
- **Supabase/MongoDB**: Authentication and persistent storage.
- **n8n**: All AI logic, scoring, question/feedback prompt calls.

Data Storage & Privacy

- Interviews and results linked to UUIDs for privacy; minimal PII required.
- Clear user data deletion workflows and GDPR compliance.
- Audio responses securely stored and auto-deleted as per retention policies.

Scalability & Performance

- Stateless API design for scalable deployments.
- Anticipate 1000+ concurrent user sessions; voice streaming to be load-tested.
- Lazy-loading of large lists (dashboard, results).

Potential Challenges

- Real-time voice interaction stability in low-bandwidth scenarios.
- Robustness of AI scoring and feedback (edge cases, adversarial answers).
- Ensuring accessibility for all users.
- Secure handling of recruiter-generated links and authentication tokens.

Milestones & Sequencing

Project Estimate

Medium: 3–4 weeks for initial MVP covering both core user types; consider staged rollout of recruiter dashboard.

Team Size & Composition

Small team (2–3 people maximum):

- 1 full-stack developer (Next.js, API, DB)
- 1 product/design/copy (wireframes, UI/UX, AI prompt tuning)
- Optionally, 1 part-time QA/support/infra (phase-based)

Suggested Phases

1. Foundation & Prototype (1 week)

- **Deliverables**: Core Next.js app with auth, question selection, basic interview flow (voice or text).
- **Dependencies**: Supabase/Mongo, Vapi/ElevenLabs API access, n8n basic workflows.

2. Core MVP Launch (1.5 weeks)

- **Deliverables**: End-to-end AI interview (voice), scoring, and feedback. Result persistence and per-user progress display.
- **Dependencies**: All prior.

3. Recruiter Dashboard (1 week, parallelizable)

- **Deliverables**: Admin login, interview link generation, result display, link expiration logic.
- **Dependencies**: MVP backend APIs.

4. Rapid Iteration & Sharing (0.5 weeks)

- **Deliverables**: Result sharing, mobile polish, onboarding flows, error handling, and accessibility upgrades.
- **Dependencies**: Early user feedback from MVP phase.

5. Future Enhancements (Post-MVP, as scheduled)

- **Deliverables**: Advanced analytics, progress tracking, company branding, referral system.
- **Dependencies**: Usage metrics, partner/recruiter feedback.