

ResuméX

Transforming the Hiring Lifecycle with Next-Gen AI Features

Problem Statement

- The Challenge: "Build a full-stack AI Enabled HR Evaluation System from scratch that manages the journey from resume upload to a final hiring decision with transparent rationale."
- Hiring new employees is often a hassle, involving manual resume filtering fatigue, multiple interviews, and unconscious bias in selection.
- Recruiters are overwhelmed, skimming resumes in seconds, while candidates are 'ghosted' without feedback. Our website helps ease these difficulties.

Proposed Solution

Resumex is an end to end full stack platform which manages the entire hiring lifecycle

1. Smart Resume Shortlisting: An AI "Gatekeeper" that parses and ranks applicants against the Job description
2. Multi-Modal Assessment Engine: Evaluate a candidate's job-readiness using multiple input modes ,not just resumes or code.
3. Dual-Track Assessment Logic: Hard skills and soft skills are evaluated independently and merged only at the final decision stage to ensure fairness, transparency, and explainability.
4. The Integrity Shield: A webcam-enabled proctoring system to ensure test validity.
5. Explainable AI: A decision engine that generates a text-based rationale for every "Hire/No-Hire" recommendation.

Module 1: Smart Resume Shortlisting

- Parsing Logic: We use pypdf to extract key entities (Skills, Education, Years of Experience) from uploaded PDF/DOCX files and send it to AI for summary description
- Scoring Heuristics: The system cross-references extracted data against the specific Job Description keywords.
- Auto-Ranking: Candidates are automatically bucketed into:
 -  High Match (Direct fit)
 -  Potential (Needs review)
 -  Reject (Misaligned)

Module 2: Multi-Modal Assessment Engine

- Coding Sandbox: A fully functional, browser-based IDE where candidates write code. It executes securely using piston api and checks hidden test cases given by the recruiter
- Scenario-Based MCQs: Questions presenting workplace conflicts to test judgment and ethics
- Text-Based Responses: Open-ended questions analyzed for communication clarity and reasoning ability
- Slider-Based Inputs: Quantitative psychometric self-assessment

Module 3: Dual-Track Assessment Logic

- Track A: Technical Assessment
 - Focus: Hard Skills (Syntax, Domain Knowledge, Efficiency).
 - Measurement: Automated grading using Execution Time (Latency), Code Conciseness (Memory Proxy), and Code Complexity
 - Track B: Psychometric Assessment
 - Focus: Soft Skills (EQ, Resilience, Leadership, Culture Fit).
 - Measurement: Analysis of scenario choices , Text Response and slider inputs.
- Final score: Calculated based on Job Description with AI-generated explanation
(example) $final_score = (0.65 * technical_score) + (0.35 * psychometric_score)$

Module 3: The Integrity Shield (Proctoring)

- Real-Time Monitoring:
 - Visual: Browser-based webcam access detects Multiple Faces or No Face using getUserMedia
 - Environmental: Listeners detect Tab-Switching, Copy-Paste events, and forbidden keyboard shortcuts (Alt+Tab).
- Audit Trail:
 - The system logs every suspicious event with a precise timestamp.
 - Recruiters see a "Trust Score" alongside the candidate's technical score.

Module 4: "Explainable AI" Decision Engine

- Beyond the Score:
 - Instead of just a number, the system outputs a Rationale.
- Automated Decision:
 - Generates a clear "Hire" or "No-Hire" recommendation badge.
- The "Why" (Example Output):
 - *"Candidate scored 92% in Technical Assessment but showed low resilience in psychometric scenarios. Recommended for Senior Individual Contributor role rather than Team Lead."*

Technical Stack

1. Frontend (User Interface)

- **Framework:** **Next.js 14** (App Router) – For server-side rendering and fast performance.
- **Library:** **React.js** – Component-based UI structure.
- **Styling:** **Tailwind CSS** – For responsive and modern utility-first styling.
- **UI Components:** **Shadcn/ui** – For pre-built, accessible components (Cards, Buttons, Inputs).
- **Icons:** **Lucide React** – For the clean icons used in the dashboard.

2. Backend (API & Logic)

- **Framework:** **FastAPI** (Python) – High-performance, async web framework.
- **Server:** **Uvicorn** – ASGI server to run the Python application.
- **ORM:** **SQLAlchemy** – To manage database interactions using Python classes instead of raw SQL.
- **Data Validation:** **Pydantic** – Used for validating request data (like UserSync and JobCreate schemas).

3. Database & Storage

- **Database:** **Supabase (PostgreSQL)** – The primary relational database used to store candidates, jobs, and assessments
- **File Storage:** Initially configured to save PDFs to a local uploads/ folder, with plans to move to cloud buckets.

4. Artificial Intelligence (AI)

- **LLM Model:** **Google Gemini 2.0 Flash** – Used for summary review of resumes and generating rationale.
Rule Engine → Score Breakdown → LLM Explanation
- **PDF Parsing:** **pypdf** – Library used to extract raw text from uploaded PDF resumes.
- **getUserMedia** - used to capture image every 30s during assessment for multiple face

5. Code Execution Engine (Sandbox)

- **Service:** **Piston API** - An external API used to compile and run candidate code safely

6. Authentication

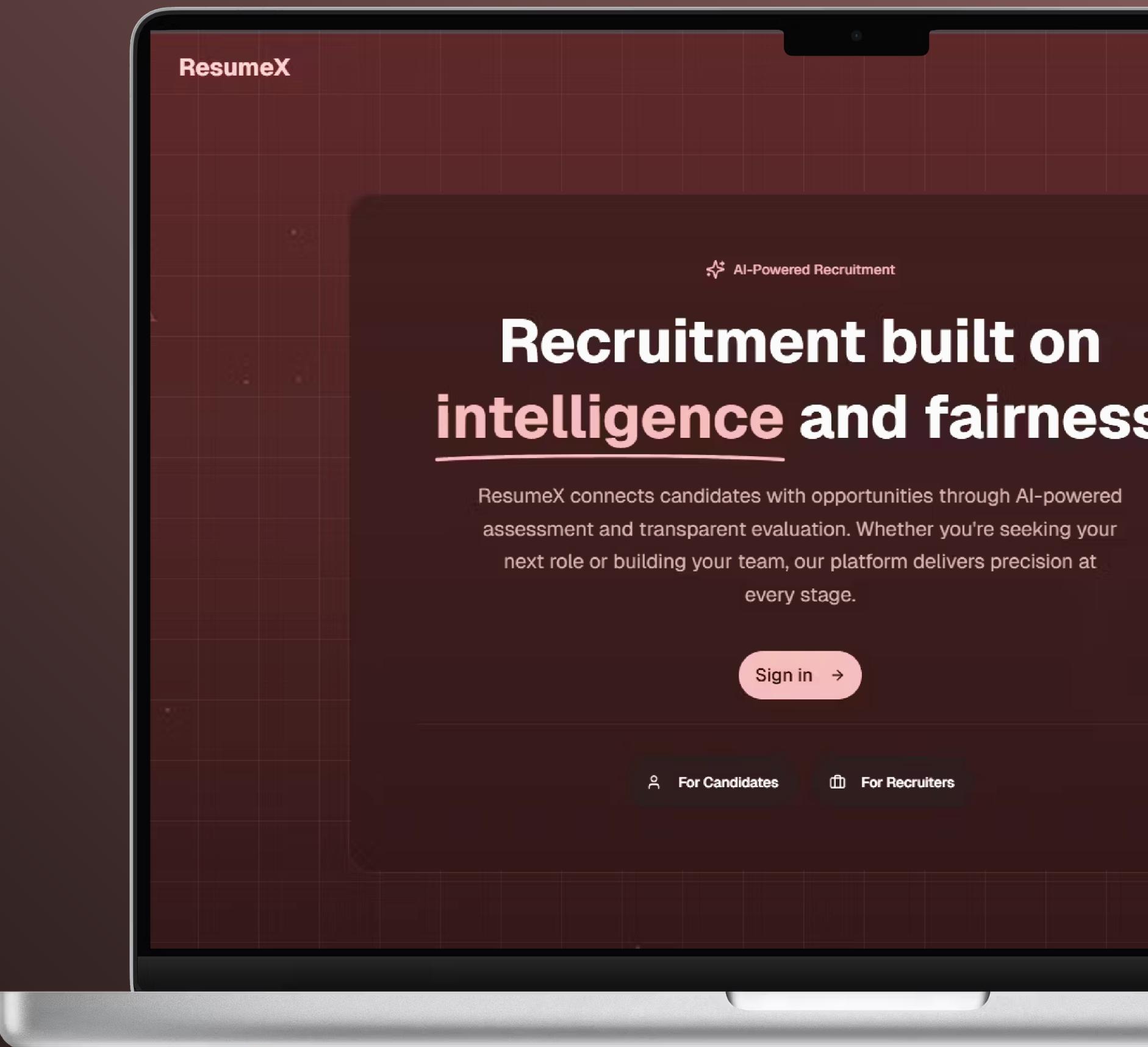
- **Service:** **Clerk** (implied by UserSync logic) – Handles user login/signup, while the backend syncs this identity to the internal database.

7. Deployment (Planned)

- **Frontend Hosting:** **Vercel** (Optimized for Next.js).
- **Backend Hosting:** **Render** (Great for Python FastAPI services).

User Experience

1. Clear, distinct pathways for Recruiters and Candidates right from the start. No confusion.
2. Explains the working process of candidates and recruiters in a simplified way
3. Built with Shadcn/ui & Tailwind CSS for a clean, professional, and distraction-free interface.
4. Users can easily log in with Google, LinkedIn using Clerk



Candidate Experience

1. A single, unified dashboard where users manage everything: Applications, Interviews, and Resume data. No clutter.
2. Jobs: users can apply and search for new jobs
3. Interviews: shows all the interview's progress you have applied and if you are selected for taking up assessment
4. My Resume: users can post their resume and get recommended for job opportunities based on their resume
5. Step-by-step progression with real time progress indicators
6. Visual warnings on tab switch or copy paste , Minimal intrusion with clear system feedback

The screenshot shows the ResumeX candidate experience dashboard. At the top left is the logo 'ResumeX'. To its right is a search bar with the placeholder 'Search candidates, jobs, or analytics...'. On the far right are three icons: a red circle with a '2', a gear, and a user profile.

The main content area has a dark background. At the top right is a red button labeled 'Find New Jobs'. Below it is a greeting 'Hello, _pravin_!?! 🙌' and a message: 'You are all set to land your next dream job. Here is your daily digest.' In the center, there's a dark rounded rectangle containing the text 'All caught up!' and 'No pending actions. Why not apply to more roles?' with a checkmark icon.

Below this are three cards: 'Total Applications' (2), 'Assessments Taken' (0), and 'Profile Visibility' (High). The 'Profile Visibility' card includes the note 'Your resume is optimized.'

At the bottom, there are sections for 'Recommended for you' (listing a 'Senior Backend Engineer' position at Google) and 'Prep Resources' (listing 'Practice Python Algorithms').

Recruiter Experience

The screenshot shows the Recruiter Dashboard interface. At the top, there's a navigation bar with 'ResumeX' logo, search bar, and notification icons. Below it is a sidebar with 'Dashboard', 'Candidates', 'Analytics', and 'Recruitment Board' options. The main area is titled 'Recruiter Dashboard' with the subtitle 'Manage your job postings and candidates.' It displays two job postings: 'test' (test - Remote, Full-time, View Applicants) and 'test #2' (hmm - Remote, Full-time, View Applicants). There's also a 'Global Analytics' button and a 'Post New Job' button.

This screenshot shows a candidate profile for '_pravin_!?' (pravinleein@gmail.com). The profile includes a red circular icon with a white 'P'. A 'Review Pending' button is visible. The profile section contains an 'AI Analysis' summary for Sangeetha K, detailing her experience as a Software Test Engineer and her skills in various programming languages and frameworks. The 'Detected Skills' section lists Java, ASP.NET Framework, Entity Framework, ADO.NET, MVC, C#, HTML, CSS, JavaScript, JQuery, W, and others. The 'Documents' section shows a PDF file named 'Resume.pdf' (Original Application) with a 'View' button. At the bottom, there are 'Reject Candidate' and 'Accept for Interview' buttons.

- A neat simple dashboard where the user could see all the details of the jobs posted
- Users can view the AI analysis of candidate's resume and can either manually or automatically select candidates for interviews
- Users can create assessments as they wish and get notified if any candidate is involved in malpractice
- Interview analysis are automated with text-based justification by our system
- Users can pick the best candidates recommended by our system

Candidate workflow

The image illustrates the candidate workflow through three main stages:

- Explore Jobs:** The user is searching for roles. They find two job postings: "test" (Full-time, Remote, \$100k - \$120k) and "test #2" (Remote, \$100k - \$120k). Both are marked as "Applied". Red arrows point from these cards to the "My Applications" section.
- My Applications:** The user reviews their applications. They see the same two jobs: "test" (Interview Selected) and "test #2" (Review Pending). Red arrows point from these cards to the "Technical Assessment" section.
- Technical Assessment:** The user takes a "Two Sum" assessment in Python. The code provided is:

```
1 # Write your solution here
2 print('Hello World')
```

. They have the option to "Run" or "Stop". Below the code editor are "Reject Candidate" and "Accept for Interview" buttons. Red arrows point from this section to the "Recruiter Dashboard" section.

Recruiter workflow:

- Recruiter Dashboard:** The user manages job postings and candidates. They see two job postings: "test" (test - Remote, Full-time) and "test #2" (hmm - Remote, Full-time). Red arrows point from these cards to the "Job Dashboard" section.
- Job Dashboard:** The user views the candidate pipeline. A specific candidate, "pravin_!?", is highlighted. Red arrows point from this candidate to the "AI Analysis" and "Assessment Configuration" sections.
- AI Analysis:** The user reviews the AI analysis for candidate "pravin_!?", which includes detected skills like Java, ASP.NET Framework, Entity Framework, ADO.NET, MVC, C#, HTML, CSS, JavaScript, jQuery, and WCF.
- Assessment Configuration:** The user configures an assessment titled "Technical Assessment" with a duration of 60 minutes. They add a question titled "Two Sum" with the instruction: "Given an array of integers 'nums' and an integer 'target', return indices of the two numbers such that they add up to 'target'. nums = [2,7,11,15], target = 9". They click "Save Assessment".
- Recruitment Analytics:** Global insights across all job postings. Key metrics shown are Total Candidates (2), Average Assessment Score (10 / 10), Placement Rate (12.5%), Top Performers (#1 pravin_!?, test, 10 / 10), and Recent Applications (pravin_!?, Applied for test; pravin_!?, Applied for test #2).

***zoom in to view the screenshots clearly**

System Architecture

1. AI Resume Parsing

- Reads text from uploaded PDFs using "pypdf".
- Sends text to AI to generate a "Smart Summary" and extract "Technical Skills" as JSON.
- Includes error handling to ensure data is captured even if the AI fails.

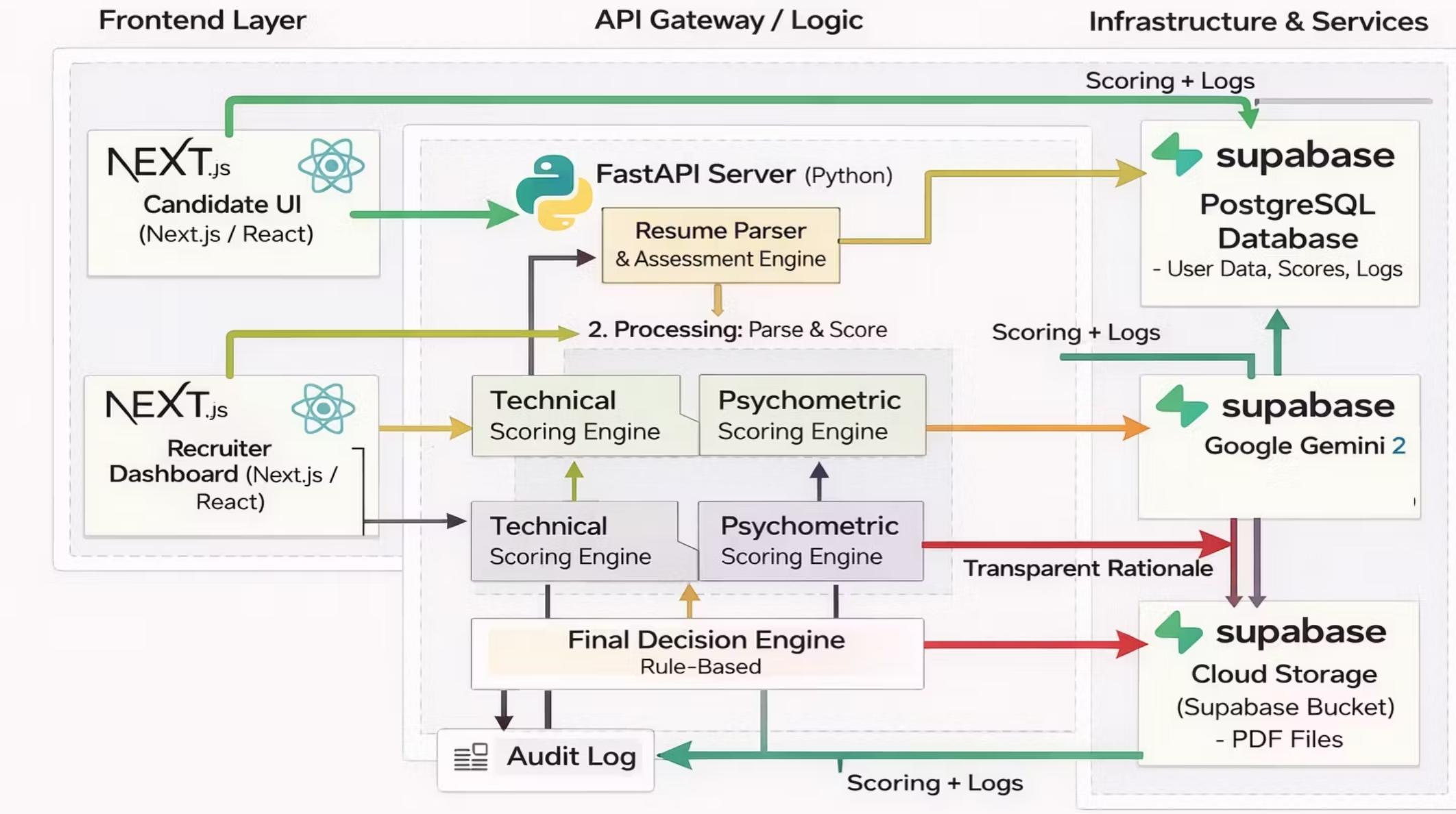
2. Job Application Handling

- Stores the resume file securely.
- Creates/Updates the Candidate profile in the database with the new AI-parsed data.
- Generates an Application record connecting the user to the specific job.

3. Recruiter Dashboard Data

- Joins data from Applications, Candidates, and Jobs tables into one view.
- Merges the candidate's technical Assessment Score (from the coding sandbox) so recruiters see everything in one row.

ResumeX System Architecture



4. Utilities

- **Tracking:** Lets candidates view their application history and status.
- **Controls:** Allows recruiters to update status i.e. "Accepted/Rejected" and prevents duplicate applications.

*the tech implement might change but logic reminds the same

Why Resumex?

We aim to build a shared ecosystem where hiring and job searching are unified, transparent, and data driven, leveraging emerging AI technologies to ensure fair and explainable decisions.

Meet Us

Pravin I (RA2411004010680) Team Leader

Farhaan (RA2411004010731)

Adarsh S (RA2412703010025)