

Assignment: AI-Powered Job Tracker with Smart Matching

Time limit: 2-3 days

The Challenge

Build a job tracking system that fetches jobs from an external API, tracks application status intelligently, and uses AI to match jobs with user's resume.

Core Features to Build

1. Job Feed & External Integration

- Fetch jobs from a **public job API** (Adzuna, JSearch on RapidAPI, or mock API)
- Display jobs in a clean feed with: title, company, location, description, job type
- Each job card should have an "Apply" button

Required Filters:

- **Role/Title:** Search by job title
- **Skills:** Multi-select skills (React, Node.js, Python, etc.)
- **Date Posted:** Last 24 hours, Last week, Last month, Any time
- **Job Type:** Full-time, Part-time, Contract, Internship
- **Work Mode:** Remote, Hybrid, On-site
- **Location:** City/region filter
- **Match Score:** High (>70%), Medium (40-70%), All

2. Resume Upload

- At login, prompt user to upload resume (PDF/TXT)
- Single resume - user can replace/update anytime
- Extract and store resume text

3. AI-Powered Job Matching

- **When jobs load**, automatically score each job against user's resume
- Show **match score (0-100%)** on every job card
- Display with color badges: Green (>70%), Yellow (40-70%), Gray (<40%)
- Show "Best Matches" section at top (6-8 highest scoring jobs)

- Explain why it matched (key skills, experience alignment)

4. Smart Application Tracking ★ (Critical Thinking Test)

- When user clicks "Apply", **open job link in new tab**
- **When user returns to your app**, show popup:
 - "Did you apply to [Job Title] at [Company]?"
 - Options: "Yes, Applied" | "No, just browsing" | "Applied Earlier"
- If "Yes", save as "Applied" with timestamp
- User can later update status: Applied → Interview → Offer/Rejected
- Show all applications in a dashboard with timeline with filters with different stages

5. AI Sidebar Assistant

Build a chat interface that answers:

- "Show me remote React jobs"
- "Give me UX jobs requiring Figma"
- "Which jobs have highest match scores?"
- "Find senior roles posted this week"

Also answers product questions:

- "Where do I see my applications?"
- "How do I upload my resume?"
- "How does matching work?"

The AI filters jobs intelligently and explains recommendations.

Technical Stack

Frontend: React (clean UI, responsive design)

Backend: Node.js with Fastify(preferred) or Nodejs

AI: OpenAI/Anthropic/Gemini or any for scoring and chat

Storage: In-memory use redis e.g upstash

Required Deliverables

1. Live Link (MANDATORY)

Deploy and share working URL

2. GitHub Repository (MANDATORY)

- Clean code, good structure
- `.env.example` with required variables
- No API keys committed

3. README.md Must Include:

- a) **Architecture Diagram** Visual diagram showing components, API flow, AI integration, and data structure
 - b) **Setup Instructions** How to run locally, environment variables, prerequisites
 - c) **AI Matching Logic** How you calculate scores, your approach, and efficiency considerations
 - d) **Critical Thinking: Popup Flow** Explain your design decisions, edge cases handled, alternatives considered
 - e) **Scalability** How this handles 100 jobs at once and 10,000 users
 - f) **Tradeoffs** What you'd improve with more time and known limitations
-

What We're Evaluating

Must Have:

- Live deployment working
- GitHub repo accessible
- Architecture diagram in README
- All filters functional
- AI match scores showing
- Smart popup flow working

Looking For:

- Product thinking (does it make sense for job seekers?)
- AI used meaningfully, not as gimmick
- Clean React code and good UX
- Smart backend design
- Independent problem-solving
- Professional UI that looks good

Bonus:

- Exceptional design
 - Performance optimizations
 - Mobile responsive
 - Creative extra features
-

Submission Checklist

- Live link works on mobile + desktop
- GitHub repo is public
- README has architecture diagram
- Filters work
- Match scores visible
- Popup flow works
- AI chat functional
- No secrets in code