✅ Web Workflow: Roles & Modules

---

1. Roles

- **Admin**
  - Manages platform, jobs, recruiters, and candidates
  - Access reports & analytics
- **Recruiter (HR)**
  - Posts jobs
  - Views and downloads applications
  - Searches candidates
- **Candidate**
  - Creates profile or uploads resume
  - Searches jobs (our board + external sources via meta-search)
  - Applies for jobs (native + smart apply for external)

✅ Main Workflow

Landing Page

- Menu: *Home* | *Jobs* | *Recruiters* | *Login* | *Register*
- Options:

- Candidate → Register / Login
- Recruiter → Register / Login
- Admin → Login (hidden link)

Candidate Web Workflow

```
Candidate Landing → Register/Login
   ↓
Dashboard:
   - Search Jobs
   - View Applications
   - Update Profile
   ↓
Search Jobs:
   - Filters: Role, Location, Experience, Salary
   - Fetch from:
      → Our Board (DB)
      → External API (Google, aggregators, LinkedIn company feeds)
   - Show Source Badge: [Our Board / LinkedIn / Google]
   ↓
View Job Details → Apply
   - Fill Application Form:
      * Name, Email, Phone
      * Relevant Experience
      * Current & Expected CTC
      * Notice Period
      * Relocation (Yes/No)
      * Resume Upload
      * Custom Questions (mandatory & optional)
   ↓
Confirmation: Application submitted
```

Recruiter (HR) Web Workflow

```
Recruiter Landing → Register/Login
   ↓
Dashboard:
   - Post New Job
   - View Applicants
```

  - Search Candidates
  - Manage Profile
  ↓
Post Job Form:
  * Job Title, Company Name (text box)
  * Location, Salary, Experience Range
  * Job Description
  * Employment Type, Notice Period
  * Add Custom Questions (mandatory toggle)
  ↓
Job Created → Visible on:
  * Our Job Board
  * Google Jobs (via structured data)
  ↓
View Applicants:
  - Table: Candidate name, CTC, Notice Period, Resume
  - Actions:
    * Download Resume (single)
    * Download All (ZIP)
    * Export CSV (with custom responses)
  ↓
Search Candidates:
  * Role, Skills, Experience, Location
  * View Profile → Resume → Shortlist

Admin Web Workflow

Admin Login
  ↓
Dashboard:
  - Recruiter Management (approve/ban)
  - Candidate Management
  - Job Post Management (edit/remove)
  - Reports:
    * Applications per Job
    * Download CSV/Resumes
  - Provider Settings:
    * API Provider Management (add/edit/delete)
    * API Limits & Rotation
  ↓
Analytics:
  * Active Jobs
  * Total Applications

* External Apply Click Tracking

✅ Key Features for Web

- **Dynamic Job Listings**: From DB + external APIs
- **Responsive Design**: Mobile-first for job seekers
- **Structured Data for Google Jobs**: JSON-LD on job details page
- **Bulk Download**: Resumes in ZIP, CSV export
- **Custom Questions Handling**:
    - Store questions in job schema
    - Render dynamically on application form
- **Search System**:
    - Meta-search engine calls:
        - Our DB
        - External providers (Google APIs, aggregators)
- **API Provider Management (Admin)**:
    - Add/edit/remove providers (OpenRouter, Together, ChatGPT)
    - Set token limits
    - Automatic fallback on failures

✅ Visual Workflow (High-Level)

```
graph TD
A[Landing Page] --> B{Login/Register}
```

```
B -->|Candidate| C[Candidate Dashboard]
B -->|Recruiter| D[Recruiter Dashboard]
B -->|Admin| E[Admin Dashboard]

C --> C1[Search Jobs]
C --> C2[Apply for Jobs]
C --> C3[Manage Profile]
C2 --> C4[Upload Resume + Fill Form]

D --> D1[Post New Job]
D --> D2[View Applicants]
D --> D3[Search Candidates]

E --> E1[Manage Recruiters]
E --> E2[Manage Jobs]
E --> E3[Manage Candidates]
E --> E4[Reports & Provider Settings]
```

Here’s the **HR (Recruiter) Bot Chat Flow** designed for **posting jobs, viewing applicants, and managing candidates**. This is structured for both **web and messaging bots (WhatsApp/Telegram)**.

✅ HR Bot Chat Flow

---

0. Greeting & Role Selection

Bot: Hi! I’m Sree 👋

Are you looking for:

1️⃣Job (Candidate)

2️⃣Candidate (Recruiter/HR)

Please type 1 or 2 or tap the button.

- **If HR selects option 2 → go to Step 1.**

---

1. HR Profile Creation (First-time only)

Bot: Great! Let’s set up your Recruiter profile.

Please share:

- Full Name

- Company Name

- Official Email (OTP optional)

- Phone Number

- Role / Department

- Location

- Preferred Contact (WhatsApp / Email / Both)

Bot Confirmation:

Bot: ✅ Profile created successfully!

What would you like to do now?

1️⃣Post a Job

2️⃣Search Candidates

3️⃣View Applicants

4️⃣Update Profile

2. Main Menu (Post-registration)

Bot: Hi [HR Name]! 👋 What would you like to do?

1️⃣Post a Job

2️⃣Search Candidates

3️⃣View Applicants

4️⃣Update Profile

3. Post a Job

HR: Post a Job

Bot: Let’s create your job post! Please share:

- Job Title

- Skills (comma-separated)

- Min & Max Experience

- Location

- Salary Range (optional)

- Job Description (optional)

Do you want to add:

- Remote/Hybrid/Onsite option? (Yes/No)

- Employment type (Full-time, Contract)?

- Notice Period preference? (Immediate, 15/30/60 days)

- Custom Questions for Applicants? (Yes/No)

Custom Questions Flow

Bot: Please enter your custom questions one by one.

Type “done” when finished.

Example: “Why do you think you are a good fit?”

[Mandatory? Yes/No]

Bot Confirmation:

Bot: ✅ Job posted successfully!

Job ID: J1234

Would you like to:

1️⃣Share with candidates

2️⃣Wait for applications

3️⃣Both

---

4. View Applicants

HR: View Applicants

Bot: Which job do you want to see applicants for?

[Show list of HR’s jobs with IDs and titles]

HR selects → Bot fetches applications:

- Candidate Name

- Experience

- Current CTC / Expected CTC

- Notice Period

- Resume link

- Custom Question responses

Options:

[Download CSV] [Download All Resumes] [View More]

---

5. Search Candidates

HR: Search Candidates

Bot: Please provide:

- Role/Job Title

- Skills

- Experience Range

- Location

- Salary Range (optional)

- Notice Period (optional)

Bot: 🔍 Searching database...

Found X candidates. Showing top 3:

- Candidate Name (masked email/phone)

- Experience & Skills

- Location

Options: [View Resume] [Shortlist] [Contact Now]

---

6. Update Profile

Bot: You can update:

- Name

- Company Name

- Email

- Phone

- Location

- Preferred Contact

✅ Additional Features

- **Bulk Actions**: In “View Applicants,” HR can download all resumes in ZIP or CSV with custom question responses.

- **Notifications**: Bot sends alert when:

    - New applicant applies

    - Candidate responds to outreach

- **Analytics (Optional)**: HR can type “My Jobs Report” → Bot sends application counts per job.

✅ Fallback Messages

- If HR sends incomplete info:
  Bot: ⚠️ I didn’t get that. Please provide [field name].
- If no applicants:

Bot: No applications yet for this job. I’ll notify you when someone applies.

Candidate Chat Flow (with Internet-wide Job Search)

0) Entry

- **User**: “Search Jobs”
- **Bot**: “Great! Tell me a Role/Skill.”
  Optional quick filters: **Location**, **Experience (yrs)**, **Salary (range)**
  Buttons: *Skip / Use my profile / Change filters*

1) Meta-Search & Aggregation (behind the scenes)

- Bot triggers **Job Meta-Search Engine** → queries multiple **Source Adapters** in parallel:
  - Search APIs (e.g., Google Programmable Search, approved SERP providers)
  - Job aggregators that accept feeds
  - Company career APIs/feeds (Greenhouse, Lever, Workday, etc., where available)
  - LinkedIn: *only via official/approved endpoints* (no scraping)
- **LLM Enrichment Layer** (via **Provider Manager**: OpenRouter/Together/ChatGPT/etc.) normalizes snippets into a common **JobUnified** shape.
- **Deduping + Ranking** → top results returned.

2) Results Presentation (clearly marked sources)

- **Bot**: “Found 10 jobs! Showing top 3.”
- Each card shows:

  - **Job Title — Company**
  - **Location • Posted X days ago • Match %**
  - **Badges**: `Our Board` / `Google` / `LinkedIn` / `Company Site` / `Aggregator`
  - **Provider note** (if fallback kicked in): `via Together (fallback from OpenRouter)`
  - Buttons: **[Details] [Apply] [Save]**
- **Actions**:
  - **Details** → expanded description, skills, salary (if available), canonical source URL.
  - **Apply**:
    - If **Our Board** → open **native application** form.
    - If **External** → **Smart Apply**: prefill as possible, then redirect to the canonical apply URL (record click + state).
  - **Save** → Bookmark in user's profile.

3) Candidate Apply (native form)

- Fields (your spec):
  - Name, Email, Phone
  - *Years of relevant experience*
  - Current CTC, Expected CTC
  - Notice period
  - **Ready to relocate?** (Yes/No)
  - **Upload resume** (PDF/DOC)

    - **Has current offers?** (Yes/No)
    - **Best time to contact** (text, max 100)
    - **Preferred mode** (Call / WhatsApp / Email)
    - **Custom questions** from the job (mandatory/optional; all stored)
- **Bot**: “✅ Application sent!”
    - Store under **Job ID**
    - Log provenance (source) if job originated externally
    - Email/WhatsApp confirmations (configurable)

System Architecture (what to build)

Components

1. **Chat Orchestrator**
    - Manages conversation state and intents (Search → Results → Apply).
    - Calls Aggregation Engine and Provider Manager.
2. **Job Meta-Search Aggregation Engine**
    - **Query Planner**: builds per-source queries from role/filters.
    - **Source Adapters** (parallel):
        - Google/approved SERP → result snippets (respect TOS).
        - Aggregators (Jooble, Jora, Adzuna, etc.) via feeds or APIs.
        - Company career APIs/feeds (Greenhouse, Lever, Workday; RSS/Atom where offered).

        - **Your own job board** (first-class).
    - **LLM Enrichment** (via Provider Manager): extract title/company/location/desc/salary/post date/apply URL.
    - **Normalizer → JobUnified**.
    - **Deduper** (canonical URL/host + title + company + fuzzy text).
    - **Ranker** (relevance to candidate profile & filters, recency, source trust).
    - **Cache** for recent queries (per role+filters).

3. **Provider Manager (LLM)**
    - **Providers**: OpenRouter, Together, ChatGPT, others.
    - **Config**: add/edit/delete providers, **per-provider limits** (RPM/TPM, daily caps).
    - **Rotation**: weighted-round-robin; **automatic failover** on timeout/5xx/quality drop.
    - **Policy**: model selection by task (extraction, summarization, classification).
    - **Observability**: success rate, latency, token usage, cost guardrails.
4. **Application Router**
    - If job origin = **Our Board** → native apply.
    - If **External** → prefill (where allowed), **redirect to canonical** apply link (log click).
5. **Storage**
    - **jobs_unified** (search results, normalized)
    - **jobs_source_raw** (raw payloads + source metadata)
    - **applications** (ties candidate → Job ID, stores custom Q&A, resume path)
    - **providers** (config), **provider_metrics**, **search_logs**
    - **bookmarks** (saved jobs), **clicks** (attribution)

6. **Admin Console**
    - Provider CRUD & limits
    - Health/usage charts
    - Search quality dashboards (dedupe rate, n-sources/job, CTR)
    - Compliance flags (robots/state, source TOS notes)

Flow Diagrams

A) Candidate Search → Apply (high level)

```
graph TD
A[Candidate: Search Jobs] --> B[Enter Role/Skill + Filters]
B --> C{Cache Hit?}
C -- Yes --> R[Return Cached Top N]
C -- No --> D[Aggregation Engine: Query Planner]
D --> E[Parallel Source Adapters]
E --> F[LLM Enrichment via Provider Manager]
F --> G[Normalize → JobUnified]
G --> H[Dedupe + Cluster]
H --> I[Rank + Cache]
I --> R[Return Top N]
R --> J[Bot Shows 3 with Source Badges]
J --> K{Apply?}
K -- Details --> L[Expand Card]
K -- Save --> M[Bookmark]
K -- Apply --> N{Origin?}
N -- Our Board --> O[Native Application Form]
N -- External --> P[Smart Prefill + Redirect]
O --> Q[Store Application + Files]
P --> Q
Q --> S[Bot: ✅ Application Sent!]
```

B) Provider Rotation & Fallback

```
sequenceDiagram
participant Orchestrator
participant ProviderMgr
participant P1 as Provider A (OpenRouter)
participant P2 as Provider B (Together)
participant P3 as Provider C (ChatGPT)
Orchestrator->>ProviderMgr: extract_fields(snippets)
ProviderMgr->>P1: call model (task=extraction)
P1-->>ProviderMgr: timeout / 5xx
ProviderMgr->>P2: retry with backoff
P2-->>ProviderMgr: success + quality score
ProviderMgr-->>Orchestrator: result + meta {provider: B, fallback_from: A}
Note over Orchestrator,ProviderMgr: UI badge: "via Together (fallback from OpenRouter)"
```

Key Behaviors (dev-ready)

Source Adapters (examples)

- **Google Programmable Search / approved SERP** → returns snippet + URL.
- **Aggregators** (opt-in, free feeds where possible) → XML/JSON.
- **Company Career Systems** → Greenhouse/Lever Boards JSON, RSS/Atom feeds (if exposed).
- **LinkedIn** → only via official partnerships/APIs allowed for your account (otherwise treat as a *share link*, not a “job slot”).
  Never scrape against TOS. Respect robots.txt and rate limits.

Normalized Job Shape (fields only;)

- `job_id` (generated; stable via URL hash)
- `title`, `company`, `locations[]`, `description_short`, `skills[]`
- `salary_min/max/currency` (if present)

- `posted_at` (ISO), `source_name`, `source_url`, `apply_url`
- `origin_type` = `OUR_BOARD` | `AGGREGATOR` | `COMPANY_SITE` | `GOOGLE` | `LINKEDIN`
- `dedupe_group_id`, `rank_score`

Dedupe (recommended signals)

- Canonical URL (normalized)
- Host + normalized title + company (lowercased, stopword-stripped)
- Fuzzy text similarity (SimHash/MinHash on description)
- Cluster; keep highest-rank representative; list alternates under “Also from: …”

Ranking (example factors)

- **Relevance** to query and/or candidate profile (skills match, experience fit)
- **Freshness** (posted_at decay)
- **Source quality** (your board > official career APIs > aggregators > generic SERP)
- **Completeness** (has salary, apply_url, location precision)
- **Engagement priors** (historical CTR, applies)

Provider Manager (LLM) Policies

- **Routing**:
  - Extraction/structuring → fast, cost-efficient models
  - Summaries/classification → quality-biased
- **Limits**: per-provider RPM/TPM/day caps, hard stop + graceful degrade

- **Failover**: timeout/5xx/quality<τ → next provider (exponential backoff)

- **Marking**: return `{provider, model, fallback_from, attempt_count, latency_ms}` for UI badges & logs

UI: “Clear Marking”

- Each result card shows a **Source badge** and **Provider note** when fallbacks used:
    - Example: `Source: Company Site • via Together (fallback from OpenRouter)`

- “Apply” button shows **where you’ll apply**:
    - `Apply on Our Board` or `Apply on Company Site` / `Apply on LinkedIn`

Apply Flow (native vs external)

- **Native (Our Board)** → your full application form + custom questions; store under **Job ID**, files to secure storage, CSV/ZIP bulk export available to recruiters/admin.

- **External**:
    - Try **Smart Prefill** (only where allowed by the destination; otherwise no prefill).
    - Always **redirect to canonical apply URL**.
    - Log **click** → **external application** with `job_id`, `candidate_id`, timestamp.
    - Offer **“Mark as Applied”** so candidates can track status in your system.

Admin / Recruiter Views (essentials)

- **Admin**

  - Provider management: add/edit/delete, limits, weights
  - Health: success rate by provider, fallback rate, latency, cost caps
  - Compliance panel: source integration modes, robots status, blocklist
  - Search quality: #results/query, dedupe %, source mix, CTR
- **Recruiter**
  - Job posts (your board)
  - Applications per job, CSV/resume bulk export
  - Saved candidates & referral tracking
  - Visibility into external apply clicks per job

Compliance, Safety, and Ops

- **Compliance**
  - Use **official APIs/feeds** only; no scraping where prohibited.
  - Respect **robots.txt**, rate limits, platform branding rules.
  - Keep **source attribution** visible to users.
- **Privacy & Consent**
  - Show consent text before Smart Apply or external redirects.
  - Store only what’s necessary; encrypt PII at rest/in transit.
- **Reliability**
  - Timeouts per source; circuit breakers on repeated failures.
  - Retries with jitter; graceful degradation to cached results.

    - Observability: structured logs, trace IDs across providers.

- **Quality**
    - Continuous eval set: measure extraction accuracy per provider.
    - Auto-fallback if a provider’s quality drops below threshold τ.

What your devs build next (, just tasks)

1. **Provider Manager**: config store, rotation, limits, health, metrics, fallback logic.
2. **Aggregation Engine**: query planner, adapters (Google/aggregators/company feeds/your board), normalizer, dedupe, rank, cache.
3. **Chat Orchestrator**: search intent → results → details → apply (native/external).
4. **Application Router**: native form vs external redirect; logging.
5. **UI Badges & Marking**: source + provider/fallback labels on every result card.
6. **Admin Console**: provider CRUD + dashboards; compliance panel.
7. **Storage**: `jobs_unified`, `jobs_source_raw`, `applications`, `providers`, `provider_metrics`, `search_logs`, `bookmarks`, `clicks`.
