



## UX Flows and Page-API Mapping

The user experience is defined for candidates, employers and admins. Below are narrative flow descriptions followed by a page-API mapping table.

### Candidate Journey

1. **Home/Landing Page** → Presents the platform value proposition, search bar and links to sign up or browse jobs.
2. **Jobs Listing/Search Page** → Candidates can enter keywords and apply filters (country, remote, visa, relocation, category). They view paginated job cards with high-level details and eligibility tags.
3. **Job Detail Page** → Displays full job description, eligibility flags and company overview. Offers an "Apply" button.
4. **Sign Up / Login** → Candidates register or sign in to proceed. Registration collects email, password and role.
5. **Profile Completion** → After initial sign-up, candidates fill in their profile (name, country, summary, skills, experience, education) and upload a resume.
6. **Apply to Job** → The candidate writes a cover letter and submits the application. The system confirms submission and triggers an email notification.
7. **Candidate Dashboard** → Candidates see a summary of their profile status and a list of their applications with current statuses (applied, under review, shortlisted, hired, rejected).
8. **Resources / Guides** → Static pages offering visa information, career guides and skill development resources.

Empty states are handled with helpful messages such as "No jobs match your search" or "You haven't applied to any jobs yet." Loading states use spinners or skeleton cards; error states show contextual alerts (e.g., "Failed to load jobs, please try again").

### Employer Journey

1. **Employer Landing Page** → Describes the benefits of hiring African talent and invites employers to register.
2. **Employer Sign Up / Login** → Employers register or sign in.
3. **Company Profile** → Employers create or update their company information (name, industry, country, description, website, logo).
4. **Employer Dashboard** → Displays a list of their jobs with status indicators (pending, active, closed) and the number of applications per job. A call-to-action encourages posting the first job if none exist.
5. **Post Job** → A form for entering job details, required skills, eligibility flags and application deadline. Submissions enter a pending state until approved by an admin.
6. **View Applications** → Employers view applications per job, see candidate profiles and resumes, and update application statuses (under review, shortlisted, hired, rejected).
7. **Employer Support / Guidance** → Resource pages on inclusive hiring and best practices.

Empty states include “You haven’t posted any jobs yet.” Error states handle missing profile information or failed submissions. Loading states appear when fetching jobs or applications.

## Admin Journey

1. **Admin Login** → Admins authenticate via the same login form but have the `admin` role.
2. **Admin Dashboard** → Displays metrics (pending jobs count, active users) and shortcuts to moderation tools.
3. **Moderate Jobs** → List of pending jobs with options to approve or reject. Admins can view job details and edit if necessary.
4. **Manage Users** → List all users; admins can deactivate accounts.
5. **Content Management** → Manage static pages and FAQ content (may rely on a simple CMS or database entries).

Admin empty states include “No pending jobs.” Error states capture permission issues and invalid actions. Loading states appear when loading lists.

## Page → API Mapping

Page / Component	Purpose	Data Required	API Endpoints Consumed
<b>Home / Landing</b>	Introduce the platform, show featured jobs and resources, provide entry points for candidates and employers.	Featured jobs (optional); static content about mission.	<code>GET /jobs</code> (with filters for featured or newest); no auth required.
<b>Jobs Listing / Search</b>	Allow candidates to browse and search jobs with filters and pagination.	List of jobs matching query; counts for filters.	<code>GET /jobs?</code> <code>q=&amp;country=&amp;remote=&amp;visa=&amp;relocation=&amp;category=&amp;page=&amp;page_size=</code>

Page / Component	Purpose	Data Required	API Endpoints Consumed
<b>Job Detail</b>	Show complete job information, eligibility flags, company details, and allow applying if signed in.	Job details; company profile snippet.	<code>GET /jobs/{jobId}</code> ; (optional) <code>GET /companies/{companyId}</code> for extended company info.
<b>Candidate Sign Up</b>	Create candidate account.	Email, password.	<code>POST /auth/register</code> (role= <code>candidate</code> ).
<b>Candidate Login</b>	Authenticate candidate.	Email, password.	<code>POST /auth/login</code> .
<b>Profile Completion</b>	Collect candidate personal and professional details; upload resume.	Candidate profile data; resume file.	<code>PUT /candidates/me</code> ; <code>POST /candidates/me/resume/upload-url</code> ; <code>POST /candidates/me/resume</code> with key.
<b>Apply to Job</b>	Submit application with cover letter.	Cover letter; job ID.	<code>POST /jobs/{jobId}/applications</code> .
<b>Candidate Dashboard</b>	Summarise candidate profile; list applications and statuses.	Candidate profile; applications list.	<code>GET /candidates/me</code> ; <code>GET /applications/me</code> .
<b>Resources / Guides</b>	Provide helpful content on visas, relocation, skill development.	Static text and media.	Served from CMS or static files; no dynamic API.

Page / Component	Purpose	Data Required	API Endpoints Consumed
<b>Employer Sign Up</b>	Create employer account.	Email, password.	<code>POST /auth/register</code> ( <code>role=employer</code> ).
<b>Employer Login</b>	Authenticate employer.	Email, password.	<code>POST /auth/login</code> .
<b>Company Profile</b>	View/update company details; upload logo.	Company info; logo file.	<code>GET /companies/me</code> ; <code>PUT /companies/me</code> ; <code>POST /companies/me/upload-url</code> → S3; <code>POST /companies/me/logo</code> with key.
<b>Employer Dashboard</b>	Show list of jobs posted by employer and high-level stats.	Employer's jobs and application counts.	<code>GET /jobs?mine=true</code> (custom filter to fetch employer's jobs); <code>GET /{jobId}/applications</code> to count.
<b>Post Job</b>	Enter job details and eligibility flags.	Job title, description, skills, country or remote, visa, relocation, category, deadline.	<code>POST /jobs</code> .
<b>View Applications</b>	View candidates who applied to a job; change status.	Applications list; candidate summaries; resumes (pre-signed URL).	<code>GET /jobs/{jobId}/applications</code> ; <code>PATCH /applications/{applicationId}</code> ; <code>GET /candidates/{id}/resume</code> (via pre-signed URL).
<b>Employer Guidance / Africa-Friendly Hiring</b>	Provide best practices for inclusive hiring.	Static content.	Static or CMS; no dynamic API.
<b>Admin Login</b>	Authenticate admin.	Email, password.	<code>POST /auth/login</code> .

Page / Component	Purpose	Data Required	API Endpoints Consumed
<b>Admin Dashboard</b>	Summarise site metrics; list pending jobs.	Counts of users, jobs, applications; pending jobs.	<code>GET /admin/jobs/pending</code> ; <code>GET /admin/users</code> ; additional analytical endpoints.
<b>Moderate Jobs</b>	Approve or reject pending jobs.	Pending jobs list; job details.	<code>GET /admin/jobs/pending</code> ; <code>PATCH /admin/jobs/{jobId}/approve</code> ; <code>PATCH /admin/jobs/{jobId}/reject</code> .
<b>Manage Users</b>	View and deactivate users.	User list.	<code>GET /admin/users</code> ; <code>PATCH /admin/users/{id}/deactivate</code> .
<b>Content Management</b>	Edit static pages and FAQ content.	Page content.	Custom admin endpoints to update resource pages (e.g., <code>PUT /admin/pages/{slug}</code> ).

This mapping ensures each SEO page has a clear counterpart in the product and consumes the appropriate API endpoint(s) defined in the backend specification.

---