



📖 Setup & Usage Guide

Job Application Assistant

Complete Setup & Usage Guide

A step-by-step walkthrough — from installation to your first application

Prepared for Manoj Kumar



What This Tool Does — And Never Does

✓ It DOES

- ✓ Search live job listings via the Adzuna API
- ✓ Score every job 0–100 against your real skills
- ✓ Draft tailored cover letters & resume notes
- ✓ Pre-fill application forms in a visible browser
- ✓ Track status: Saved → Applied → Interview → Offer
- ✓ Generate a daily summary report

✗ It NEVER Does

- ✗ Auto-submit any application, anywhere
- ✗ Log into any account or bypass CAPTCHA/OTP
- ✗ Scrape LinkedIn or Naukri (against their ToS)
- ✗ Fabricate skills, employers, or achievements
- ✗ Click Submit/Apply on your behalf

Step 1 — One-Time Environment Setup

- 1 Open a terminal in the project folder

```
cd job_app_assistant
```

- 2 Create a virtual environment

```
python -m venv venv
```

- 3 Activate it (Windows)

```
venv\Scripts\activate
```

- 4 Install dependencies

```
pip install -r requirements.txt
```

ⓘ One extra step if you'll use the Assisted Apply feature: **playwright install chromium**

Step 2 — Configure Your API Keys

Create a `.env` file

Copy `.env.example` to `.env` in the project folder, then fill in your keys.

```
ADZUNA_APP_ID=your_app_id_here # required
ADZUNA_APP_KEY=your_app_key_here # required
ANTHROPIC_API_KEY=your_key_here # optional
```

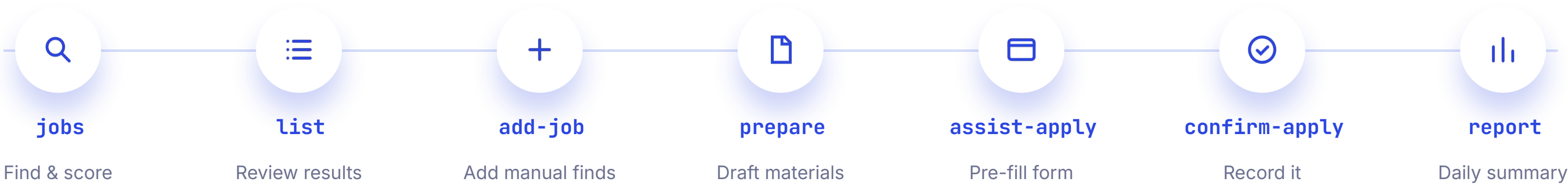
Never commit your real `.env`

It's already excluded via `.gitignore`.

Anthropic key is optional

Cover letters still work fine using built-in templates.

Your Daily Job Search Pipeline



Every step below is covered in detail on the following slides.

Search & Score New Jobs

```
python -m src.cli jobs
```

- ✓ Searches Adzuna for every title in your profile, across every preferred location
- ✓ Scores each result 0–100 against your skills, experience, and location fit
- ✓ Saves everything into data/jobs.csv — safe to open in Excel any time
- ✓ Prints a results table right in the terminal, sorted by match score



Run this once a day

Existing jobs are never duplicated — only new or updated listings are added.

Review Results in Excel

```
python -m src.cli list --min-score 75
```

- ✓ Filters saved jobs by minimum match score directly in the terminal
- ✓ Or open data/jobs.csv in Excel to browse, sort, and filter visually
- ✓ A polished tracker can be generated on request — color-coded score, clickable Apply button, frozen header, autofilter

- Green = 75%+ match
- Yellow = 50–74% match
- Red = below 50%

Focus your time on green rows first.

Add Jobs Found on **LinkedIn / Naukri**

```
python -m src.cli add-job --company "Acme" --title "..."
```

- ✓ LinkedIn and Naukri block automated scraping in their Terms of Service
- ✓ So this tool never touches them directly — you paste in what you find
- ✓ The same scoring, tailoring, and tracking pipeline then runs on it
- ✓ Include the job URL and full description for the most accurate score



Keeps your account safe

No bot behavior on sites that prohibit it.

Draft Your **Application Materials**

```
python -m src.cli prepare <job_id>
```

- ✓ Flags keywords from the job post that are missing in your resume
- ✓ Drafts a tailored cover letter and a short recruiter message
- ✓ Saves everything to prepared_applications/<job_id>/ for you to review
- ✓ Asks for your confirmation before marking the job "Ready to Apply"



Nothing is fabricated

Tailoring only reorders and rephrases what's genuinely on your resume.

Assisted Apply — Browser Pre-fill

```
python -m src.cli assist-apply <job_id>
```

- ✓ Opens the company's application page in a real, visible browser window
- ✓ Best-effort fills name, email, phone, LinkedIn, portfolio & cover letter
- ✓ Leaves resume upload and the Submit button for YOU to handle
- ✓ Refuses outright on linkedin.com / naukri.com, or if a login page is detected



The browser never clicks Submit

You review every field, then apply yourself.

Confirm & Track Your Progress

```
confirm-apply <job_id> → status <job_id> Interview
```

- ✓ confirm-apply records that YOU personally submitted the application
- ✓ It never submits anything itself — it's a note-taking step, after the fact
- ✓ status moves a job through: Saved → Applied → Interview → Offer/Rejected
- ✓ Your full history lives in data/tracker.csv, always editable by hand



Your personal CRM

For the job hunt — nothing more.

Your Daily Summary

```
python -m src.cli report
```

- ✓ Jobs found today, and how many crossed the 75% match threshold
- ✓ Applications prepared vs. actually submitted
- ✓ Interviews in progress and total rejections, at a glance
- ✓ Highlights your single best opportunity, plus any follow-ups now due



A 30-second check-in

Run it first thing to plan your day.

Optional

Automate the **Daily Search**

Windows Task Scheduler can run the search for you every morning, before you even open your laptop.

```
schtasks /create /tn "JobAssistant Daily Search" /tr "cmd /c cd /d C:\...\job_app_assistant && venv\Scripts\python.exe -m src.cli jobs >> logs\scheduled_run.log 2>&1" /sc daily /st 09:00
```



Run Command Prompt as Administrator the first time



Change /st 09:00 to suit your morning routine



All output is logged to logs/scheduled_run.log

Common Setup Mistakes

Wrong folder

Commands fail with "No module named src" — always cd into the project folder first.

venv not activated

Prompt must show (venv) before running any `python -m src.cli` command.

File permission errors

Close jobs.csv in Excel and remove any Read-only flag before re-running jobs.

Pasting job_id wrong

Copy the plain ID only — never include the `< >` angle brackets from examples.

🏠 [A Note on Safety](#)

Every feature in this tool was built around one rule: *it prepares, you decide.*

It never logs in anywhere, never bypasses anti-bot protection, never touches sites that prohibit automation, and never clicks Submit on your behalf. The human — you — always makes the final call before an application goes out.

No auto-submit

No LinkedIn/Naukri automation

No fabricated skills

No account risk

You're Ready to Start

Run `python -m src.cli jobs` each morning, review your matches, and let the tool handle the busywork — you handle the decisions.

```
python -m src.cli jobs
```