AI-Powered Resume Screening System

MongoDB Database Design Document

1. Overview

The system uses MongoDB (via Djongo) for flexible schema storage, particularly for unstructured resume text and JSON screening results.

2. Collections & Schema

2.1 Users (accounts_user)

Stores authentication and role data.

```
{
  "_id": ObjectId("65c1..."),
  "username": "recruiter_jane",
  "email": "jane@company.com",
  "role": "recruiter",
  "is_active": true,
  "date_joined": ISODate("2024-02-11T10:00:00Z")
}
```

2.2 Job Postings (jobs_jobposting)

Stores job requirements.

```
{
  "_id": ObjectId("65c2..."),
  "title": "Senior Python Developer",
  "description": "Looking for a Django expert...",
  "required_skills": "Python, Django, MongoDB",
  "recruiter_id": ObjectId("65c1..."),
  "created_at": ISODate("2024-02-11T12:00:00Z")
}
```

2.3 Resumes (resumes_resume)

Stores file metadata and extracted text.

```
{
  "_id": ObjectId("65c3..."),
  "candidate_name": "John Doe",
  "file": "resumes/john_doe.pdf",
  "parsed_text": "EXPERIENCE: Senior Dev... SKILLS: Python, SQL...",
  "uploaded_at": ISODate("2024-02-11T12:05:00Z")
}
```

2.4 Screening Results (screening_screeningresult)

Stores outcomes of the ML pipeline.

```
{
  "_id": ObjectId("65c4..."),
  "job_id": ObjectId("65c2..."),
  "resume_id": ObjectId("65c3..."),
  "score": 0.85,
```

```
  "skill_match_details": {
    "matched": ["Python", "Django"],
    "missing": ["MongoDB"]
  },
  "created_at": ISODate("2024-02-11T12:06:00Z")
}
```

3. Indexing Strategy

1. users.username (Unique)

2. screening_results.job_id (Ascending)

3. screening_results.score (Descending)

4. jobs.recruiter_id (Hash/Ascending)

4. Django Interaction

Djongo translates Django ORM calls to MongoDB queries. Application handles relation logic (Joins via secondary queries or embedded logic).