Here is a **cleaned-up**, **concise version** of your original **NoSQL (MongoDB-style)** schema for an **Automated Job Requisition & Posting System**, with all unnecessary repetition, verbosity, and extended explanations removed — keeping only what's essential for implementation.

## ✅ Cleaned-Up NoSQL Schema: Automated Job Requisition & Posting System

### 1. Design Principles

* Use **embedding** for 1:N data (e.g., approvals).
* Use **referencing** for large or shared data (e.g., users, AI logs).
* Index for query patterns.
* Enforce **schema validation** (JSON Schema).
* Use **transactions** for multi-document writes (replica set required).

### 2. Collections & Schemas

#### 📁 users

Stores all system users.

{

\_id: ObjectId,

name: string,

email: string, // unique

role: 'manager' | 'system' | 'hr' | 'approver',

department: string,

metadata: { phone: string, title: string },

created\_at: date,

updated\_at: date

}

**Indexes:**

* { email: 1 } (unique)
* { role: 1 }
* { department: 1 }

#### 📁 job\_requisitions

Main document for each job requisition.

{

\_id: ObjectId,

manager\_id: ObjectId, // ref: users.\_id

prompt\_text: string,

extracted: { title: string, department: string, location: string, experience\_level: string },

required\_skills: [string],

description: string,

salary\_range: { min: number, max: number, currency: string },

status: 'initiated' | 'ai\_generated' | 'approved' | 'posted' | 'closed',

approval\_chain: [

{ approver\_id: ObjectId, level: number, status: 'pending'|'approved'|'rejected'|'auto\_approved', remarks: string, acted\_at: date }

],

suggested\_channels: [string],

created\_at: date,

updated\_at: date

}

**Indexes:**

* { manager\_id: 1, status: 1 }
* { 'extracted.title': 'text', 'description': 'text' }
* { created\_at: -1 }

#### 📁 ai\_processing\_logs

Logs for AI stages, inputs, and outputs.

{

\_id: ObjectId,

job\_id: ObjectId, // ref: job\_requisitions.\_id

stage: 'prompt\_parsing' | 'jd\_generation' | 'skill\_prediction' | 'timeline\_prediction',

input\_data: object,

output\_data: object,

confidence: number,

model\_version: string,

processed\_at: date

}

**Indexes:**

* { job\_id: 1 }
* { stage: 1, processed\_at: -1 }

#### 📁 ai\_predictions

Stores AI model predictions per job.

{

\_id: ObjectId,

job\_id: ObjectId,

predicted\_time\_to\_hire\_days: number,

candidate\_availability\_score: number,

suggested\_channels: [string],

feature\_vectors: object,

model\_version: string,

generated\_at: date

}

**Indexes:**

* { job\_id: 1 }
* { generated\_at: -1 }

#### 📁 job\_postings

Tracks external/internal job postings.

{

\_id: ObjectId,

job\_id: ObjectId,

channel: 'linkedin' | 'naukri' | 'indeed' | 'internal',

post\_url: string,

status: 'pending' | 'posted' | 'failed',

external\_id: string,

response\_count: int,

metrics: { clicks: int, views: int, applies: int },

last\_synced\_at: date,

created\_at: date

}

**Indexes:**

* { job\_id: 1, channel: 1 }
* { external\_id: 1 }

#### 📁 skills (optional)

Canonical skill list for normalization and mapping.

{

\_id: ObjectId,

name: string, // unique

aliases: [string],

popularity\_score: number,

created\_at: date

}

**Indexes:**

* { name: 1 } (unique)
* { aliases: 1 }