MongoDB Data Modeling –

# Collections & Schema Design

## users Collection

Stores user information for both employers and job seekers.

Sample Schema:

{  
 \_id: ObjectId,  
 username: String,  
 email: String,  
 role: String, // "employer" or "job\_seeker"  
 profile: {  
 fullName: String,  
 location: String,  
 skills: [String], // for job seekers  
 companyName: String, // for employers  
 resumeUrl: String // for job seekers  
 }  
}

## jobs Collection

Contains job postings created by employers.

Sample Schema:

{  
 \_id: ObjectId,  
 title: String,  
 company: String,  
 location: String,  
 jobType: String, // "Full-time", "Internship", etc.  
 skillsRequired: [String],  
 employerId: ObjectId, // Reference to users.\_id  
 createdAt: Date  
}

## applications Collection

Represents applications submitted by job seekers to job posts.

Sample Schema:

{  
 \_id: ObjectId,  
 jobId: ObjectId, // Reference to jobs.\_id  
 jobSeekerId: ObjectId, // Reference to users.\_id  
 resumeUrl: String,  
 status: String, // "Pending", "Shortlisted", "Rejected", etc.  
 testScore: Number,  
 submittedAt: Date,  
 interviewScheduled: Boolean,  
 interviewDate: Date  
}

## analytics Collection

Holds statistics for each job post for reporting and insights.

Sample Schema:

{  
 \_id: ObjectId,  
 jobId: ObjectId, // Reference to jobs.\_id  
 employerId: ObjectId, // Reference to users.\_id  
 views: Number,  
 applications: Number,  
 shortlisted: Number,  
 interviewScheduled: Number,  
 hired: Number,  
 generatedAt: Date  
}

# Relationships Between Collections

- users.\_id → Referenced by jobs.employerId and applications.jobSeekerId  
- jobs.\_id → Referenced by applications.jobId and analytics.jobId  
- Each job post can have many applications  
- Each employer can have many jobs and analytics records