📦Final MongoDB Database Design (Industry-Ready)

This document contains the **complete and final database design** for your blog platform. You can download, store, or share this as your single source of truth.

🕗Project Scope Covered

- Admin panel (write/edit/delete blogs)
- Public blog reading
- User comments & moderation
- Likes
- Google login + normal login
- Buy-Me-Coffee support model
- Future subscriptions
- User restrictions & quotas
- Audit logging (admin actions)

✂️users Collection

Single collection for **users + admins**

```
{
  "_id": "ObjectId",
  "name": "John Doe",
  "email": "john@gmail.com",

  "authProvider": "LOCAL",
  "password": "hashed_password",

  "role": "USER",
  "isActive": true,

  "restrictions": {
    "commenting": {
      "isBlocked": false,
      "reason": null,
      "blockedUntil": null
    }
  },
  "blogQuota": {
    "monthlyLimit": 1,
    "usedThisMonth": 0
  },

  "createdAt": "ISODate",
  "updatedAt": "ISODate"
}
```

Notes

- `role` : USER | ADMIN
- Google users → password = null
- Admins are just users with role = ADMIN

✒️ blogs Collection

```
{
  "_id": "ObjectId",
  "title": "How Spring Boot Works",
  "content": "Full blog content",

  "authorId": "ObjectId",

  "status": "PUBLISHED",
  "likeCount": 0,

  "isDeleted": false,
  "deletedBy": null,
  "deletedAt": null,

  "createdAt": "ISODate",
  "updatedAt": "ISODate"
}
```

Notes

- status: DRAFT | PUBLISHED
- Soft delete supported
- `likeCount` stored for performance

🛏️comments Collection

```
{
  "_id": "ObjectId",
  "blogId": "ObjectId",
  "userId": "ObjectId",

  "content": "Nice article!",

  "status": "APPROVED",
  "moderatedBy": null,
  "moderatedAt": null,
  "moderationReason": null,

  "createdAt": "ISODate"
}
```

Notes

- status: PENDING | APPROVED | REJECTED
- Admin moderation supported

🌂likes Collection

```
{
  "_id": "ObjectId",
  "userId": "ObjectId",
  "blogId": "ObjectId",
  "createdAt": "ISODate"
}
```

Index (Important)

```
{ userId: 1, blogId: 1 } // unique
```

🌡️ payments Collection

```
{
  "_id": "ObjectId",
  "userId": "ObjectId",

  "type": "SUPPORT",
  "amount": 199,
  "currency": "INR",

  "provider": "RAZORPAY",
  "status": "SUCCESS",

  "createdAt": "ISODate"
}
```

Notes

- type: SUPPORT | SUBSCRIPTION
- Enables Buy-Me-Coffee & future plans

🎒audit_logs Collection

```
{
  "_id": "ObjectId",
  "adminId": "ObjectId",

  "action": "BLOCK_COMMENT",
  "entityType": "COMMENT",
  "entityId": "ObjectId",

  "targetUserId": "ObjectId",
  "reason": "Violation of rules",

  "createdAt": "ISODate"
}
```

Notes

- Tracks all admin actions
- Important for security & debugging

🔗Relationships Overview

```
User ─┬─ Blogs
      ├─ Comments
```

```
         ├─ Likes
         └─ Payments

Admin ── Audit Logs
```

🔗Why This Design Is Industry-Level

- Clean & scalable
- No unnecessary collections
- Supports future monetization
- Easy to map to Spring Boot
- Easy to explain in interviews

🗜️Ready for Next Steps

You can now: 1. Map this to Spring Boot entities 2. Create REST APIs 3. Add Spring Security + JWT 4. Dockerize & deploy

This is the FINAL version. You can safely build your entire backend on top of this.