db.createCollection("users", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["name", "email", "password"],

properties: {

name: {

bsonType: "string",

description: "must be a string and is required"

},

email: {

bsonType: "string",

description: "must be a string and is required"

},

password: {

bsonType: "string",

description: "must be a string and is required"

},

role: {

bsonType: "string",

enum: ["user", "admin", "editor"],

description: "must be one of the enum values"

},

avatar: {

bsonType: "string",

description: "must be a string"

},

avatarPublicId: {

bsonType: "string",

description: "must be a string"

},

bio: {

bsonType: "string",

description: "must be a string"

},

isVerified: {

bsonType: "bool",

description: "must be a boolean"

},

socialLinks: {

bsonType: "object",

properties: {

twitter: { bsonType: "string" },

facebook: { bsonType: "string" },

instagram: { bsonType: "string" },

linkedin: { bsonType: "string" },

github: { bsonType: "string" }

}

},

notificationSettings: {

bsonType: "object",

properties: {

email: { bsonType: "bool" },

push: { bsonType: "bool" },

marketing: { bsonType: "bool" }

}

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on email

db.users.createIndex({ "email": 1 }, { unique: true });

db.createCollection("blogs", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["title", "slug", "content", "author"],

properties: {

title: {

bsonType: "string",

description: "must be a string and is required"

},

slug: {

bsonType: "string",

description: "must be a string and is required"

},

content: {

bsonType: "string",

description: "must be a string and is required"

},

excerpt: {

bsonType: "string",

description: "must be a string"

},

author: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

category: {

bsonType: "objectId",

description: "must be an objectId"

},

tags: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

imageUrl: {

bsonType: "string",

description: "must be a string"

},

imagePublicId: {

bsonType: "string",

description: "must be a string"

},

imageAlt: {

bsonType: "string",

description: "must be a string"

},

status: {

bsonType: "string",

enum: ["draft", "published", "archived"],

description: "must be one of the enum values"

},

likes: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

views: {

bsonType: "int",

description: "must be an integer"

},

readTime: {

bsonType: "int",

description: "must be an integer"

},

comments: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

seo: {

bsonType: "objectId",

description: "must be an objectId"

},

accessibility: {

bsonType: "object",

properties: {

hasAltText: { bsonType: "bool" },

hasStructuredContent: { bsonType: "bool" },

readabilityScore: { bsonType: "int" }

}

},

featured: {

bsonType: "bool",

description: "must be a boolean"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on slug

db.blogs.createIndex({ "slug": 1 }, { unique: true });

db.createCollection("comments", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["content", "author", "blog"],

properties: {

content: {

bsonType: "string",

description: "must be a string and is required"

},

author: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

blog: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

parent: {

bsonType: "objectId",

description: "must be an objectId"

},

replies: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

likes: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

isEdited: {

bsonType: "bool",

description: "must be a boolean"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create indexes for faster queries

db.comments.createIndex({ "blog": 1 });

db.comments.createIndex({ "author": 1 });

db.createCollection("categories", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["name", "slug"],

properties: {

name: {

bsonType: "string",

description: "must be a string and is required"

},

slug: {

bsonType: "string",

description: "must be a string and is required"

},

description: {

bsonType: "string",

description: "must be a string"

},

imageUrl: {

bsonType: "string",

description: "must be a string"

},

imagePublicId: {

bsonType: "string",

description: "must be a string"

},

color: {

bsonType: "string",

description: "must be a string"

},

featured: {

bsonType: "bool",

description: "must be a boolean"

},

parent: {

bsonType: "objectId",

description: "must be an objectId"

},

blogCount: {

bsonType: "int",

description: "must be an integer"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on slug

db.categories.createIndex({ "slug": 1 }, { unique: true });

db.createCollection("tags", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["name"],

properties: {

name: {

bsonType: "string",

description: "must be a string and is required"

},

displayName: {

bsonType: "string",

description: "must be a string"

},

description: {

bsonType: "string",

description: "must be a string"

},

count: {

bsonType: "int",

description: "must be an integer"

},

featured: {

bsonType: "bool",

description: "must be a boolean"

},

creator: {

bsonType: "objectId",

description: "must be an objectId"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on name

db.tags.createIndex({ "name": 1 }, { unique: true });

db.createCollection("notifications", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["recipient", "type", "message"],

properties: {

recipient: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

sender: {

bsonType: "objectId",

description: "must be an objectId"

},

type: {

bsonType: "string",

enum: ["like", "comment", "follow", "mention", "system", "blog\_published"],

description: "must be one of the enum values and is required"

},

message: {

bsonType: "string",

description: "must be a string and is required"

},

read: {

bsonType: "bool",

description: "must be a boolean"

},

relatedBlog: {

bsonType: "objectId",

description: "must be an objectId"

},

relatedComment: {

bsonType: "objectId",

description: "must be an objectId"

},

link: {

bsonType: "string",

description: "must be a string"

},

createdAt: { bsonType: "date" }

}

}

}

});

// Create index for faster queries

db.notifications.createIndex({ "recipient": 1 });

db.notifications.createIndex({ "read": 1 });

db.createCollection("userpreferences", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["user"],

properties: {

user: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

theme: {

bsonType: "string",

enum: ["light", "dark", "system"],

description: "must be one of the enum values"

},

fontSize: {

bsonType: "string",

enum: ["small", "medium", "large"],

description: "must be one of the enum values"

},

reducedMotion: {

bsonType: "bool",

description: "must be a boolean"

},

highContrast: {

bsonType: "bool",

description: "must be a boolean"

},

emailNotifications: {

bsonType: "object",

properties: {

comments: { bsonType: "bool" },

likes: { bsonType: "bool" },

follows: { bsonType: "bool" },

mentions: { bsonType: "bool" },

newsletter: { bsonType: "bool" },

blogUpdates: { bsonType: "bool" }

}

},

pushNotifications: {

bsonType: "object",

properties: {

comments: { bsonType: "bool" },

likes: { bsonType: "bool" },

follows: { bsonType: "bool" },

mentions: { bsonType: "bool" },

blogUpdates: { bsonType: "bool" }

}

},

preferredCategories: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

preferredTags: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

language: {

bsonType: "string",

description: "must be a string"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on user

db.userpreferences.createIndex({ "user": 1 }, { unique: true });

db.createCollection("searchhistory", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["user", "query"],

properties: {

user: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

query: {

bsonType: "string",

description: "must be a string and is required"

},

filters: {

bsonType: "object",

properties: {

category: { bsonType: "objectId" },

tags: {

bsonType: "array",

items: {

bsonType: "objectId"

}

},

author: { bsonType: "objectId" },

dateRange: {

bsonType: "object",

properties: {

from: { bsonType: "date" },

to: { bsonType: "date" }

}

},

sortBy: { bsonType: "string" }

}

},

resultCount: {

bsonType: "int",

description: "must be an integer"

},

createdAt: { bsonType: "date" }

}

}

}

});

// Create index for faster queries

db.searchhistory.createIndex({ "user": 1 });

db.searchhistory.createIndex({ "query": 1 });

db.createCollection("seodata", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["blog", "title", "description"],

properties: {

blog: {

bsonType: "objectId",

description: "must be an objectId and is required"

},

title: {

bsonType: "string",

description: "must be a string and is required"

},

description: {

bsonType: "string",

description: "must be a string and is required"

},

keywords: {

bsonType: "array",

items: {

bsonType: "string"

}

},

canonicalUrl: {

bsonType: "string",

description: "must be a string"

},

ogTitle: {

bsonType: "string",

description: "must be a string"

},

ogDescription: {

bsonType: "string",

description: "must be a string"

},

ogImage: {

bsonType: "string",

description: "must be a string"

},

ogImageAlt: {

bsonType: "string",

description: "must be a string"

},

twitterTitle: {

bsonType: "string",

description: "must be a string"

},

twitterDescription: {

bsonType: "string",

description: "must be a string"

},

twitterImage: {

bsonType: "string",

description: "must be a string"

},

twitterImageAlt: {

bsonType: "string",

description: "must be a string"

},

structuredData: {

bsonType: "string",

description: "must be a string"

},

createdAt: { bsonType: "date" },

updatedAt: { bsonType: "date" }

}

}

}

});

// Create unique index on blog

db.seodata.createIndex({ "blog": 1 }, { unique: true });