

Name: Harsh Singh
Roll No: 2315000918
Assignment: 02

1.

users collection db.createCollection("users", { validator: { \$jsonSchema: { bsonType: "object", required: ["name", "email", "password"], properties: { name: { bsonType: "string" }, email: { bsonType: "string", pattern: "^\\S+@\\S+\\.\\S+\$" }, password: { bsonType: "string" } } } } }); db.users.createIndex({ email: 1 }, { unique: true });

products collection db.createCollection("products", { validator: { \$jsonSchema: { bsonType: "object", required: ["title", "description", "price", "category", "stock"], properties: { title: { bsonType: "string" }, description: { bsonType: "string" }, price: { bsonType: "number", minimum: 0 }, category: { bsonType: "string" }, stock: { bsonType: "int", minimum: 0 } } } } });

orders collection db.createCollection("orders", { validator: { \$jsonSchema: { bsonType: "object", required: ["userId", "products", "totalAmount", "orderDate"], properties: { userId: { bsonType: "objectId" }, products: { bsonType: "array", items: { bsonType: "object", required: ["productId", "quantity"], properties: { productId: { bsonType: "objectId" }, quantity: { bsonType: "int", minimum: 1 } } } }, totalAmount: { bsonType: "number", minimum: 0 }, orderDate: { bsonType: "date" } } } } });

reviews collection

db.createCollection("reviews", { validator: { \$jsonSchema: { bsonType: "object", required: ["userId", "productId", "rating", "comment"], properties: { userId: { bsonType: "objectId" }, productId: { bsonType: "objectId" }, rating: { bsonType: "int", minimum: 1, maximum: 5 }, comment: { bsonType: "string" } } } } });

users collection

db.createCollection("users", { validator: { \$jsonSchema: { bsonType: "object", required: ["name", "email", "role"], properties: { name: { bsonType: "string" }, bsonType: "string", pattern: "^\\S+@\\S+\\.\\S+\$" }, role: { bsonType: "string", enum:

2.

```
email: {
```

```
[ "student", "instructor" ] } } } } );  
db.users.createIndex( { email: 1 }, { unique: true } );
```

courses collection

```
db.createCollection("courses", { validator: { $jsonSchema: { bsonType: "object",  
required: ["title", "instructorId", "category", "price", "createdAt"], properties: { title: {  
bsonType: "string" }, instructorId: { bsonType: "objectId" }, category: { bsonType:  
"string" }, price: { bsonType: "number", minimum: 0 }, createdAt: { bsonType: "date"  
}, lessons: { bsonType: "array", items: { bsonType: "object", required: ["title",  
"videoURL", "duration"], properties: { title: { bsonType: "string" }, videoURL: {  
bsonType: "string" }, duration: { bsonType: "number", minimum: 1 } } } } } } } }
```

enrollments collection

```
db.createCollection("enrollments", { validator: { $jsonSchema: { bsonType: "object",
required: ["userId", "courseId", "enrolledAt"], properties: { userId: { bsonType:
"objectId" }, courseId: { bsonType: "objectId" }, enrolledAt: { bsonType: "date" }}}}});
```

3.

email: {

users collection

```
db.createCollection("users", { validator: { $jsonSchema: { bsonType: "object",
required: ["name", "email", "role"], properties: { name: { bsonType: "string" },
bsonType: "string", pattern: "^\\S+@\\S+\\.\\.\\S+$" }, role: { bsonType: "string", enum:
["organizer", "attendee"] }}}});
db.users.createIndex({ email: 1 }, { unique: true });
```

events collection

```
db.createCollection("events", { validator: { $jsonSchema: { bsonType: "object",
required: ["title", "organizerId", "location", "startTime", "endTime", "capacity"],
properties: { title: { bsonType: "string" }, organizerId: { bsonType: "objectId" },
location: { bsonType: "string" }, startTime: { bsonType: "date" }, endTime: { bsonType:
"date" }, capacity: { bsonType: "int", minimum: 1 }}}});
```

bookings collection

```
db.createCollection("bookings", { validator: { $jsonSchema: { bsonType: "object",
required: ["eventId", "attendeId", "bookingDate"], properties: { eventId: { bsonType:
"objectId" }, attendeId: { bsonType: "objectId" }, bookingDate: { bsonType: "date"
}}}});
```

4.

email: {

authors collection

```
db.createCollection("authors", { validator: { $jsonSchema: { bsonType: "object",  
required: ["name", "email", "bio"], properties: { name: { bsonType: "string" },  
bsonType: "string", pattern: "^\\S+@\\S+\\.\\.\\S+$" }, bio: { bsonType: "string" }}}});  
db.authors.createIndex({ email: 1 }, { unique: true });
```

articles collection

```
db.createCollection("articles", { validator: { $jsonSchema: { bsonType: "object",  
required: ["title", "content", "authorId", "tags", "published", "createdAt"], properties:  
{ title: { bsonType: "string" }, content: { bsonType: "string" }, authorId: { bsonType:  
"objectId" }, tags: { bsonType: "array", items: { bsonType: "string" } }, published: {  
bsonType: "bool" }, createdAt: { bsonType: "date" }}}});
```

comments collection

```
db.createCollection("comments", { validator: { $jsonSchema: { bsonType: "object",  
required: ["articleId", "userName", "commentText", "postedAt"], properties: {  
articleId: { bsonType: "objectId" }, userName: { bsonType: "string" }, commentText: {  
bsonType: "string" }, postedAt: { bsonType: "date" }}}});
```

5.

email: {

6.

users collection

```
db.createCollection("users", { validator: { $jsonSchema: { bsonType: "object",
required: ["email", "name", "signupDate"], properties: { email: { bsonType: "string",
pattern: "^\\S+@\\S+\\.\\S+$" }, name: { bsonType: "string" }, signupDate: {
bsonType: "date" }}}});
db.users.createIndex({ email: 1 }, { unique: true });
```

plans collection

```
db.createCollection("plans", { validator: { $jsonSchema: { bsonType: "object",
required: ["name", "price", "features", "billingCycle"], properties: { name: { bsonType:
"string" }, price: { bsonType: "number", minimum: 0 }, features: { bsonType: "array",
items: { bsonType: "string" } }, billingCycle: { bsonType: "string", enum: ["monthly",
"yearly"] }}}});
```

subscriptions collection

```
db.createCollection("subscriptions", { validator: { $jsonSchema: { bsonType: "object",
required: ["userId", "planId", "startDate", "isActive"], properties: { userId: { bsonType:
"objectId" }, planId: { bsonType: "objectId" }, startDate: { bsonType: "date" }, isActive:
{ bsonType: "bool" }}}});
```