

# MONGODB ASSIGNMENT 2

## SOLUTIONS

### 1. E-Commerce Store – Product & Orders

- **users collection**

```
db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "email", "password"],
      properties: {
        name: { bsonType: "string" },
        email: {
          bsonType: "string",
          pattern: "^.+@.+\\.\\..+$"
        },
        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: "double", 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: "double", 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" }
      }
    }
  }
});
```

## 2. Online Course Platform – Instructors & Students

- **users collection**

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

- **courses collection (with embedded lessons)**

```
db.createCollection("courses", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["title", "instructorId", "category", "price", "createdAt", "lessons"],
      properties: {
        title: { bsonType: "string" },
        instructorId: { bsonType: "objectId" },
        category: { bsonType: "string" },
        price: { bsonType: "double", minimum: 0 },
        createdAt: { bsonType: "date" },
        lessons: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["title", "videoURL", "duration"],
            properties: {
              title: { bsonType: "string" },
              videoURL: { bsonType: "string" },
              duration: { bsonType: "double", 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. Event Booking System – Organizers & Attendees

- **users collection**

```

db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "email", "role"],
      properties: {
        name: { bsonType: "string" },
        email: {
          bsonType: "string",
          pattern: "^.+@.+\\.\\..+$"
        },
        role: {
          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", "attendeed", "bookingDate"],
      properties: {
        eventId: { bsonType: "objectId" },
        attendeed: { bsonType: "objectId" },
        bookingDate: { bsonType: "date" }
      }
    }
  }
});

```

## 4. Blogging Platform – Authors & Articles

- **authors collection**

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

- **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. Subscription App – Users & Plans

- **users collection**

```

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

```

- **plans collection**

```

db.createCollection("plans", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "price", "features", "billingCycle"],
      properties: {
        name: { bsonType: "string" },
        price: { bsonType: "double", minimum: 0 },
        features: {
          bsonType: "array",
          items: { bsonType: "string" }
        },
        billingCycle: {

```

```
        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" }
      }
    }
  }
});
```

---