

ASSIGNMENT

Name: Saransh Varshney

University Roll Number: 2315001986

Assignment Number: 2 (Mongo DB)

1. E-Commerce Store – Product & Orders

```
const user = {  
  username: {type: String ,required: true},  
  email: {type : String ,required: true , unique: true},  
  password: {type: String, required: true, minLength:8,  
    maxLength:20}  
}
```

```
const product = {  
  title: {type: String , required: true},  
  description: {type: String , required: true},  
  price: {type: Number ,  
    required: true ,min: 0},  
  category: {type: String ,  
    required: true},  
  stock: {type: Number ,required: true}  
}
```

```
const order = {  
  userId: { type: mongoose.Schema.Types.ObjectId,  
    ref: 'user', required: true }, products: [
```

```
{
  productId: { type:
    mongoose.Schema.Types.ObjectId, ref: 'product',
    required: true }, quantity: { type: Number, required:
    true, min: 1 }
},
totalAmount: { type: Number,
  required: true, min: 0 },
orderDate: { type: Date,
  default: Date.now }
}
```

```
const review = {
  userId: { type: mongoose.Schema.Types.ObjectId,
    ref: 'user', required: true },
  productId: { type:
    mongoose.Schema.Types.ObjectId, ref: 'product',
    required: true }, rating: { type: Number, required:
    true, min: 1, max: 5 },
  comment: { type: String, required: true }
}
```

```
}
```

2. Online Course Platform – Instructors & Students

```
const user = {  
  name: { type: String, required: true },  
  email: { type: String, required: true, unique: true},  
  role: { type: String, required: true, enum: ['student',  
    'instructor'] }  
}
```

```
const lesson = {  
  title: { type: String,  
    required: true },  
  videoURL: { type: String,  
    required: true },  
  duration: { type: Number, required: true, min: 1 }  
}
```

```
const course = {  
  title: { type: String, required: true },  
  instructorId: { type:
```

```
mongoose.Schema.Types.ObjectId, ref: 'user',  
required: true },
```

```
category: { type: String, required: true },
```

```
price: { type: Number,
```

```
required: true, min: 0 },
```

```
createdAt: { type: Date,
```

```
default: Date.now }, lessons:
```

```
[lesson]
```

```
}
```

```
const enrollment = {
```

```
  studentId: { type:
```

```
    mongoose.Schema.Types.ObjectId, ref: 'user',
```

```
    required: true }, courseId: { type:
```

```
    mongoose.Schema.Types.ObjectId, ref: 'course',
```

```
    required: true }, enrolledAt: { type: Date, default:
```

```
    Date.now }
```

```
}
```

3. Event Booking System – Organizers & Attendees

```
const user = {  
  name: { type: String, required: true },  
  email: { type: String, required: true, unique: true},  
  role: { type: String, required: true, enum: ['organizer',  
    'attendee'] }  
}
```

```
const event = {  
  title: { type: String, required: true },  
  organizerId: { type:  
    mongoose.Schema.Types.ObjectId, ref: 'user',  
    required: true }, location: { type: String, required:  
    true },  
  startTime: { type: Date,  
    required: true }, endTime: {  
    type: Date, required: true  
  },  
  capacity: { type: Number, required: true, min: 1 }
```

```
}
```

```
const booking = {  
  eventId: { type: mongoose.Schema.Types.ObjectId,  
    ref: 'event', required: true }, attendeeId: { type:  
    mongoose.Schema.Types.ObjectId, ref: 'user',  
    required: true }, bookingDate: { type: Date, default:  
    Date.now }  
}
```

4. Blogging Platform – Authors & Articles

```
const author = {  
  name: { type: String, required: true },  
  email: { type: String, required:  
    true, unique: true }, bio: { type:  
    String }  
}
```

```
const article = {
```

```
title: { type: String, required: true },
content: { type: String, required: true },
authorId: { type:
mongoose.Schema.Types.ObjectId, ref: 'author',
required: true }, tags: { type: [String] },
published: { type: Boolean,
required: true }, createdAt: {
type: Date, default: Date.now }
}

const comment = {
  articleId: { type: mongoose.Schema.Types.ObjectId,
ref: 'article', required: true },
  userName: { type: String,
required: true }, commentText: {
type: String, required: true },
  postedAt: { type: Date, default:
Date.now }
}
```


5. Subscription App – Users & Plans

```
const user = {  
  email: { type: String, required:  
    true, unique: true}, name: {  
    type: String, required: true },  
  signupDate: { type: Date, required: true }  
}
```

```
const plan = {  
  name: { type: String, required: true },  
  price: { type: Number, required:  
    true, min: 0 }, features: { type:  
    [String], required: true },  
  billingCycle: { type: String, required: true, enum:  
    ['monthly', 'yearly'] }  
}
```

```
const subscription = {  
  userId:{type:mongoose.Schema.Types.ObjectId,ref:'user',required:true},planId:{type:  
mongoose.Schema.Types.ObjectId, ref: 'plan',  
required:true},startDate:{type:Date,  
required: true },  
isActive: { type: Boolean, required: true }  
}
```