

# ASSIGNMENT 2

## MongoDB

-By Yuvraj Singh

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

```
const user = {
  username: {type: String ,require: true},
  email: {type : String ,require: true , unique: true},
  password: {type: String, require: true, minLenght:8, maxLenght:20}
}

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

const order = {
  userId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', require: true },
  products: [
    {
      productId: { type: mongoose.Schema.Types.ObjectId, ref: 'product', require: true },
      quantity: { type: Number, require: true, min: 1 }
```

```
    }  
  ],  
  totalAmount: { type: Number, require: true, min: 0 },  
  orderDate: { type: Date, default: Date.now }  
}  
  
const review = {  
  userId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', require: true },  
  productId: { type: mongoose.Schema.Types.ObjectId, ref: 'product', require: true },  
  rating: { type: Number, require: true, min: 1, max: 5 },  
  comment: { type: String, require: true }  
}
```

## 2. Online Course Platform – Instructors & Students

```
const user = {  
  name: { type: String, require: true },  
  email: { type: String, require: true, unique: true },  
  role: { type: String, require: true, enum: ['student', 'instructor'] }  
}  
  
const lesson = {  
  title: { type: String, require: true },  
  videoURL: { type: String, require: true },  
  duration: { type: Number, require: true, min: 1 }  
}  
  
const course = {  
  title: { type: String, require: true },  
  instructorId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', require: true },  
}
```

```
category: { type: String, require: true },
price: { type: Number, require: true, min: 0 },
createdAt: { type: Date, default: Date.now },
lessons: [lesson]
}
```

```
const enrollment = {
  studentId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', require: true },
  courseId: { type: mongoose.Schema.Types.ObjectId, ref: 'course', require: true },
  enrolledAt: { type: Date, default: Date.now }
}
```

### 3. Event Booking System – Organizers & Attendees

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

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

```
const booking = {
```

```
eventId: { type: mongoose.Schema.Types.ObjectId, ref: 'event', require: true },
attendeeld: { type: mongoose.Schema.Types.ObjectId, ref: 'user', require: true },
bookingDate: { type: Date, default: Date.now }
}
```

## 4. Blogging Platform – Authors & Articles

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

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

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

## 5. Subscription App – Users & Plans

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

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

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