

Assignment-2 (Schema Creation)

1. E-Commerce Schema

```
const usersSchema=new mongoose.Schema({  
  name:{type:String,required:true},  
  email:{type:String,required:true,unique:true,match:/^.+@.+\.+$/},  
  password:{type:String,required:true,minlength:6}  
});
```

a. Users Schema

```
const productSchema=new mongoose.Schema({  
  title:{type:String,required:true},  
  description:{type:String,required:true},  
  price:{type:Number,required:true,min:0},  
  category:{type:String,required:true,enum:['Electronics','Clothing','Books','Other']},  
  stock:{type:Number,required:true,min:0}  
});
```

b. Product Schema

c. Order Schema

```
const orderSchema=new mongoose.Schema({  
  ⚡userId:{type:mongoose.Schema.ObjectId,ref:'User',required:true},  
  items:[{  
    productId:{type:mongoose.Schema.ObjectId,ref:'Product',required:true},  
    quantity:{type:Number,required:true,min:1}  
  }],  
  totalAmount:{type:Number,required:true,min:0},  
  orderDate:{type>Date,required:true}  
});
```

d. Review Schema

```
const reviewSchema=new mongoose.Schema({
  userId:{type:mongoose.Schema.ObjectId,ref: 'User',required:true},
  productId:{type:mongoose.Schema.ObjectId,ref: 'Product',required:true},
  rating:{type:Number,required:true,min:1,max:5},
  comment:{type:String,required:true}
});
```

2. Online Course Platform

a. User Schema

```
const userSchema=new mongoose.Schema({
  name:{type:String,required:true},
  email:{type:String,required:true,unique:true,match:/^.+@.+\.+$/},
  role:{type:String,required:true,enum:['student','instructor']}
});
```

b. Course (Lessons embedded)

```
const courseSchema=new mongoose.Schema({
  title:{type:String,required:true},
  instructorId:{type:mongoose.Schema.ObjectId,ref: 'User',required:true},
  category:{type:String,required:true},
  price:{type:Number,required:true,min:0},
  createdAt:{type>Date, default:Date.now},
  lessons:[{
    title:{type:String,required:true},
    videoURL:{type:String,required:true},
    duration:{type:Number,required:true,min:1}
  }]
});
```

c. Enrollments Schema

```
const enrollmentSchema=new mongoose.Schema({
  studentId:{type:mongoose.Schema.ObjectId,ref: 'User',required:true},
  courseId:{type:mongoose.Schema.ObjectId,ref: 'Course',required:true},
  enrolledAt:{type>Date, default:Date.now}
});
```

3. Event Booking system

a. User

```
const userSchema=new Schema({
  name:{type:String,required:true},
  email:{type:String,required:true,unique:true,match:/^.+@.+\.+$/},
  role:{type:String,required:true,enum:['organizer','attendee']}
});
```

b. Event

```
const eventSchema=new Schema({
  title:{type:String,required:true},
  organizerId:{type:mongoose.Schema.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}
});
```

c. Booking

```
const bookingSchema=new Schema({
  eventId:{type:mongoose.Schema.ObjectId,ref:'Event',required:true},
  attendeeId:{type:mongoose.Schema.ObjectId,ref:'User',required:true},
  bookingDate:{type:Date,default:Date.now}
});
```

4. Blogging Platform

a. Author

```
const authorSchema=new Schema({
  name:{type:String,required:true},
  email:{type:String,required:true,unique:true,match:/^.+@.+\.+$/},
  bio:{type:String}
});
```

b. Article

```
const articleSchema=new Schema({
  title:{type:String,required:true},
  content:{type:String,required:true},
  authorId:{type:mongoose.Schema.ObjectId,ref:'Author',required:true},
  tags:[{type:String}],
  published:{type:Boolean,required:true},
  createdAt:{type:Date,default:Date.now}
});
```

c. Comment

```
const commentSchema=new Schema({
  articleId:{type:mongoose.Schema.ObjectId,ref:'Article',required:true},
  userName:{type:String,required:true},
  commentText:{type:String,required:true},
  postedAt:{type>Date,default>Date.now}
});
```

5. Subscription App

a. User

```
const userSchema=new Schema({
  email:{type:String,required:true,unique:true,match:/^.+@.+\.+$/},
  name:{type:String,required:true},
  signupDate:{type>Date,default>Date.now}
});
```

b. Plan

```
const planSchema=new Schema({
  name:{type:String,required:true},
  price:{type:Number,required:true,min:0},
  features:[{type:String}],
  billingCycle:{type:String,required:true,enum:['monthly','yearly']}
});
```

c. Subscription

```
const subscriptionSchema=new Schema({
  userId:{type:mongoose.Schema.ObjectId,ref:'User',required:true},
  planId:{type:mongoose.Schema.ObjectId,ref:'Plan',required:true},
  startDate:{type>Date,required:true},
  isActive:{type:Boolean,default:true}
});
```