

### 1. User Schema

```
const mongoose = require('mongoose');

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

module.exports = mongoose.model('User', userSchema);
```

### 2. Product 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
  },
  stock: {
    type: Number,
    required: true,
    min: 0
  }
});

module.exports = mongoose.model('Product', productSchema);
```

### 3. Order Schema

```
const orderSchema = new mongoose.Schema({
  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
    }
  }
});

module.exports = mongoose.model('Order', orderSchema);

```

#### 4. Review Schema

```

const reviewSchema = new mongoose.Schema({
  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
  }
});

module.exports = mongoose.model('Review', reviewSchema);

```

#### 5. Course User Schema

```

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

```

#### 6. Course Schema

```

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

```

#### 7. Enrollment Schema

```

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

```

```
});  
module.exports=mongoose.model('Enrollment',enrollmentSchema);
```

## 8. Event User Schema

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

## 9. Event Schema

```
const eventSchema=new mongoose.Schema({  
  title:String,  
  organizerId:{type:mongoose.Schema.Types.ObjectId,ref:'User'},  
  location:String,  
  startTime:Date,  
  endTime:Date,  
  capacity:{type:Number,min:1}  
});  
module.exports=mongoose.model('Event',eventSchema);
```

## 10. Booking Schema

```
const bookingSchema=new mongoose.Schema({  
  eventId:{type:mongoose.Schema.Types.ObjectId,ref:'Event'},  
  attendeeId:{type:mongoose.Schema.Types.ObjectId,ref:'User'},  
  bookingDate:{type:Date,default:Date.now}  
});  
module.exports=mongoose.model('Booking',bookingSchema);
```

## 11. Author Schema

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

## 12. Article Schema

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

## 13. Comment Schema

```
const commentSchema=new mongoose.Schema({  
  articleId:{type:mongoose.Schema.Types.ObjectId,ref:'Article'},  
  userName:String,  
  commentText:String,  
  postedAt:{type:Date,default:Date.now}  
});  
module.exports=mongoose.model('Comment',commentSchema);
```

## 14. Subscription User Schema

```
const userSchema=new mongoose.Schema({  
  email:{type:String,required:true,unique:true},  
  name:String,  
  signupDate:{type:Date,default:Date.now}  
});  
module.exports=mongoose.model('User',userSchema);
```

## 15. Plan Schema

```
const planSchema=new mongoose.Schema({
```

```
name:String,  
price:{type:Number,min:0},  
features:[String],  
billingCycle:{type:String,enum:['monthly','yearly']}  
});  
module.exports=mongoose.model('Plan',planSchema);
```

## 16. Subscription Schema

```
const subscriptionSchema=new mongoose.Schema({  
  userId:{type:mongoose.Schema.Types.ObjectId,ref:'User'},  
  planId:{type:mongoose.Schema.Types.ObjectId,ref:'Plan'},  
  startDate:{type>Date,default>Date.now},  
  isActive:Boolean  
});  
module.exports=mongoose.model('Subscription',subscriptionSchema);
```