

-->

```
const users = new mongoose.Schema(
  {
    name:{type:String, required:true},
    email:{type:String, required:true, unique:true},
    password:{type:String, required:true},
  }
);

const products = new mongoose.Schema(
  {
    title:{type:String, required:true},
    description:{type:String, required:true},
    price:{type:Number, required:true},
    category:{type:String, required:true},
    stock:{type:Number, required:true},
  }
);

const orders = new mongoose.Schema(
  {
    userid:{type: mongoose.Schema.ObjectId, ref:'users', required:true},
    productids:{type:String, required:true},
    total_amount:{type:Number, required:true},
    orderDate:{type>Date, required:true},
  }
);

const reviews = new mongoose.Schema(
  {
    userid:{type: mongoose.Schema.ObjectId, ref:'users', required:true},
    productid:{type: mongoose.Schema.ObjectId, ref:'products', required:true},
    rating:{type:Number, min:1,max:5},
    comment:{type:String, required:true},
  }
);
```

2. -->

```
const users = new mongoose.Schema(
  {
    name:{ type:String, required: true},
    email:{ type:String, unique:true,required: true},
    role:{ type:String,enum:['Student','Instructor'] ,required: true}
  }
);
```

```
const courses = new mongoose.Schema(
  {
    title:{ type:String, required: true},
    instructorid:{ type:String, required: true, unique:true},
    category:{ type:String, required: true},
    price:{ type:Number, required: true, min:0},
    createdAt:{ type:Date, required: true}
  }
);
```

```
const enrollments = new mongoose.Schema(
  {
    title:{ type:String, required: true},
    videoURL:{ type:String, required: true},
    duration:{ type:Number, required: true}
  }
);
```

```
const lessons = new mongoose.Schema(
  {
    duration:{type:Number,min:1}
  }
);
```

3. -->

```
const users = new mongoose.Schema(
  {
    name: { type:String, required: true},
    email: { type:String, required: true, unique:true},
    role: { type:String,enum:['organizer','attendee'], required: true},
  }
);
```

```
const events = new mongoose.Schema(
  {
    title: { type:String, required: true},
    organizerId: { type:String, required: true},
    location: { type:String, required: true},
    startTime: { type:Date, required: true},
    endTime: { type:Date, required: true},
    capacity: { type:Number, required: true, min:0},
  }
);
```

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

```
    eventId: { type:String, required: true},  
    attendeeId: { type:String, required: true},  
    bookingDate: { type:Date, required: true}  
  }  
);
```