```
      // e-commerce website
const userSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true }
});

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 }
});

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 }
});

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 }
});

// online course platform

const courseUserSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  role: { type: String, enum: ['student', 'instructor'], required: true }
});

const lessonSchema = new mongoose.Schema({
  title: { type: String, required: true },
  videoURL: { type: String, required: true },
  duration: { type: Number, required: true, min: 1 }
});

const courseSchema = new mongoose.Schema({
  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: [lessonSchema]
});
```

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

// event booking system

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

const eventSchema = new mongoose.Schema({
  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 bookingSchema = new mongoose.Schema({
  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 }
});

// blogging platform

const authorSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true, match: /^[\w.-]+@[\w.-]+\.\w{2,4}$/ },
  bio: { type: String }
});

const articleSchema = new mongoose.Schema({
  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 commentSchema = new mongoose.Schema({
  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 }
});

// subscription app

const subUserSchema = new mongoose.Schema({
  name: { type: String, required: true },
```

```
  email: { type: String, required: true, unique: true, match: /^[\w.-]+@[\w.-]+\.\w{2,4}$/ },
  signupDate: { type: Date, default: Date.now }
});

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

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