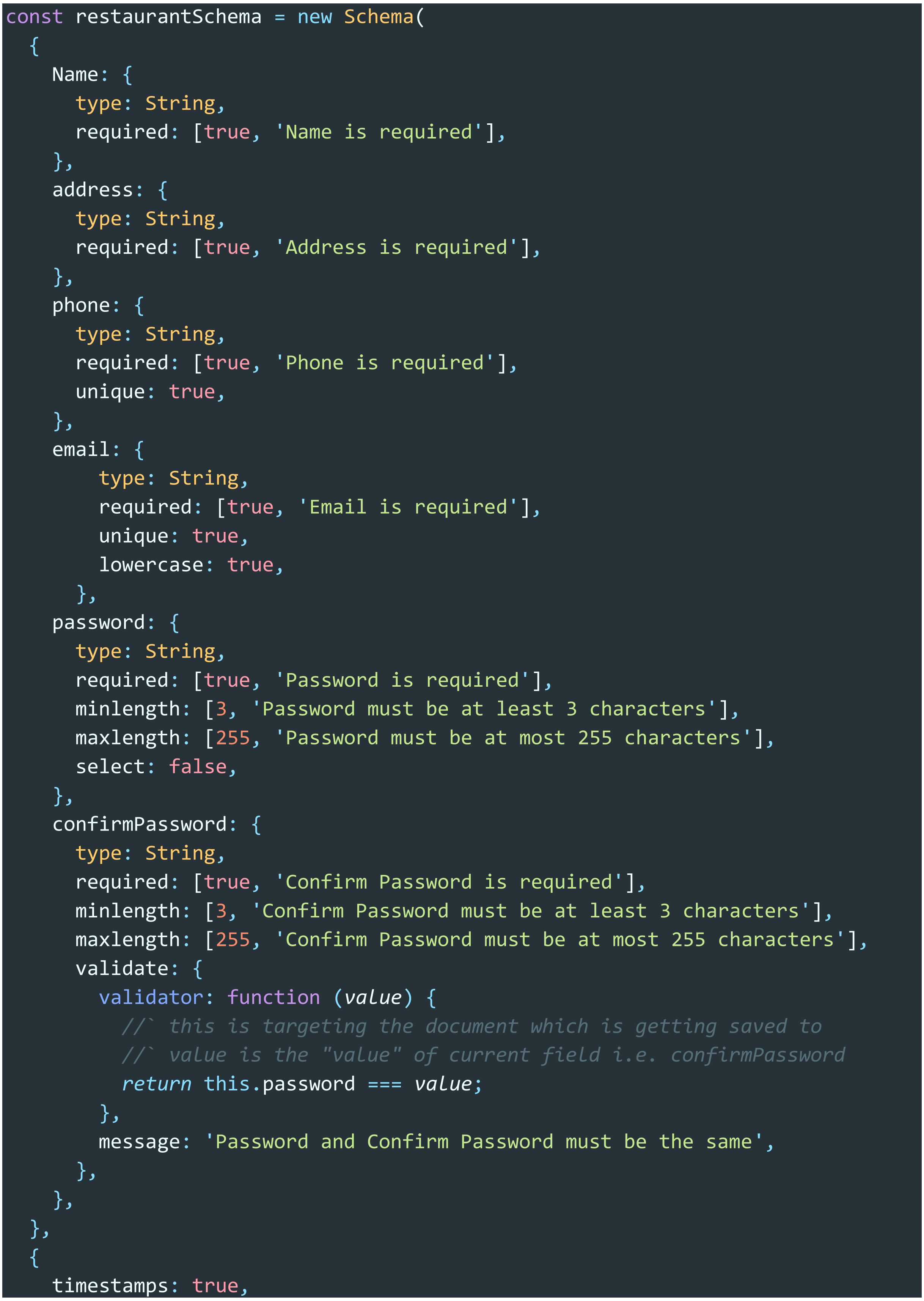
Databases for Food Application

For User information:

```
const userSchema = new Schema(
  {
    Name: {
      type: String,
      required: [true, 'Name is required'],
    },
    email: {
      type: String,
      required: [true, 'Email is required'],
      unique: true,
      lowercase: true,
    },
    phone: {
      type: String,
      required: [true, 'Phone is required'],
      unique: true,
    },
    password: {
      type: String,
      required: [true, 'Password is required'],
      minlength: [3, 'Password must be at least 3 characters'],
      maxlength: [255, 'Password must be at most 255 characters'],
      select: false,
    },
    confirmPassword: {
      type: String,
      required: [true, 'Confirm Password is required'],
      minlength: [3, 'Confirm Password must be at least 3 characters'],
      maxlength: [255, 'Confirm Password must be at most 255 characters'],
      validate: {
        validator: function (value) {
          //` this is targeting the document which is getting saved to
          //` value is the "value" of current field i.e. confirmPassword
          return this.password === value;
        },
        message: 'Password and Confirm Password must be the same',
      },
    },
  },
  {
    timestamps: true,
    collection: 'user',
  }
);
```

For Restaurant:



```
const restaurantSchema = new Schema(
  {
    Name: {
      type: String,
      required: [true, 'Name is required'],
    },
    address: {
      type: String,
      required: [true, 'Address is required'],
    },
    phone: {
      type: String,
      required: [true, 'Phone is required'],
      unique: true,
    },
    email: {
        type: String,
        required: [true, 'Email is required'],
        unique: true,
        lowercase: true,
      },
    password: {
      type: String,
      required: [true, 'Password is required'],
      minlength: [3, 'Password must be at least 3 characters'],
      maxlength: [255, 'Password must be at most 255 characters'],
      select: false,
    },
    confirmPassword: {
      type: String,
      required: [true, 'Confirm Password is required'],
      minlength: [3, 'Confirm Password must be at least 3 characters'],
      maxlength: [255, 'Confirm Password must be at most 255 characters'],
      validate: {
        validator: function (value) {
          //` this is targeting the document which is getting saved to
          //` value is the "value" of current field i.e. confirmPassword
          return this.password === value;
        },
        message: 'Password and Confirm Password must be the same',
      },
    },
  },
  {
    timestamps: true,
    collection: 'restaurant',
  }
);
```

For food items:

```
const itemSchema = new Schema(
  {
    restaurantemail: {
        type: String,
        required: [true, 'restaurantemail is required'],
        lowercase: true,
      },
    item: {
      type: String,
      required: [true, 'item is required'],
    },
    price: {
        type: String,
        required: [true, 'price is required']
    }
  },
  {
    timestamps: true,
    collection: 'fooditems',
  }
);
```

For order:

```
const orderSchema = new Schema(
  {
    email: {    //user email
      type: String,
      required: [true, 'Email is required'],
    },
    item: {
      type: String,
      required: [true, 'item is required'],
    },
    quantity: {
        type: Number,
        required: [true, 'quantity is required']
    },
    price: {
        type: String,
        required: [true, 'price is required']
    },
    process: {
        type: String,
        required: [true, 'process is required']    // [completed]
    }
  },
  {
    timestamps: true,
    collection: 'orders',
  }
);
```