

Mongodb Schemas

1.customer Schema

```
name: { type: String, required: true },
email: { type: String, required: true, unique: true },
mobile: { type: Number, required: true, unique: true },
password: { type: String, required: true },

ordermobile: {
  type: Number,
  required: true,
  enum: [0],
  default: 0,
},
address: {
  type: String,
  required: true,
  enum: ["add address"],

  default: "add address",
},
```

2.Restaurant Schema

```
restaurantname: { type: String, required: true },
address: { type: String, required: true },
opentime: { type: String, required: true },
closetime: { type: String, required: true },
username: { type: String, required: true, unique: true },
email: { type: String, required: true, unique: true },
mobile: { type: Number, required: true, unique: true },
password: { type: String, required: true },
posterurl: { type: String, required: true },
role: { type: String, default: "restaurant", enum: "restaurant" },
```

3.Food Item Schema

```

    itemname: { type: String, required: true },
    quantity: { type: Number, required: true },
    price: { type: Number, required: true },
    description: { type: String, required: true },
    posterurl: { type: String, required: true },
    restaurant: {
      type: mongoose.Schema.Types.ObjectId,
      ref: "restarurantuser",
    },
  },

```

4.Order Schema

```

customer: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "users",
  required: true,
},
foodItem: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "fooditems",
  required: true,
},
quantity: { type: Number, required: true },

restaurant: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "restarurantusers",
  required: true,
},
addres: {
  type: String,
  required: true,
},
ordermobile: { type: Number, required: true, unique: true },
orderStatus: {
  type: String,
  required: true,
  enum: ["pending", "Accepted", "Rejected"],
  default: "pending",
}

```

```

    },
    AcceptStatus: {
      type: String,
      required: true,
      enum: ["Accept Order", "Order Accepted"],
      default: "Accept Order",
    },
    rejectStatus: {
      type: String,
      required: true,
      enum: ["Reject Order", "Order Rejected"],
      default: "Reject Order",
    },
  },

```

5.Cart item Schema

```

foodItem: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "fooditems",
  required: true,
},
customer: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "users",
  required: true,
},
restaurant: {
  type: mongoose.Schema.Types.ObjectId,
  ref: "restarurantusers",
  required: true,
},
quantity: { type: Number, min: 1, required: true, default: 1 },

```