Nawallah DB Schema:

Collections and Fields:

1. **System Admin**
2. **Restaurant Admin**
3. **NGOs**
4. **Customers**
5. **Restaurants**
6. **Food Items**
7. **Orders**
8. **Donations**

MongoDB Schema

1. System Admin

```
{
  "_id": ObjectId,
  "email": String,
  "password": String,
  "name": String,
  "role": { type: String, default: "admin" }
}
```

2. Restaurant Admin

```
{
  "_id": ObjectId,
  "email": String,
  "password": String,
  "name": String,
  "restaurantId": ObjectId, // Reference to Restaurants
  "role": { type: String, default: "restaurant_admin" }
}
```

3. NGOs

```
{
  "_id": ObjectId,
  "email": String,
  "password": String,
  "name": String,
  "location": {
    "type": { type: String, enum: ['Point'], required: true },
    "coordinates": [Number]
  },
  "role": { type: String, default: "ngo" }
}
```

4. Customers

```
{
  "_id": ObjectId,
  "email": String,
  "password": String,
  "name": String,
  "location": {
    "type": { type: String, enum: ['Point'], required: true },
    "coordinates": [Number]
  },
  "googleId": String, // For Google Sign-in
  "role": { type: String, default: "customer" }
}
```

5. Restaurants

```
{
  "_id": ObjectId,
  "name": String,
```

```
 "location": {
   "type": { type: String, enum: ['Point'], required: true },
   "coordinates": [Number]
 },
 "profile": {
   "address": String,
   "phone": String,
   "description": String,
   "image": String
 },
 "menu": [
   {
     "foodItemId": ObjectId, // Reference to Food Items
     "price": Number,
     "availabilityStatus": Boolean,
     "isDelivery": Boolean,
     "isDonation": Boolean
   }
 ]
}
```

6. Food Items

```
{
  "_id": ObjectId,
  "name": String,
  "description": String,
  "image": String,
  "category": String
}
```

7. Orders

```
{
 "_id": ObjectId,
 "customerId": ObjectId, // Reference to Customers
 "restaurantId": ObjectId, // Reference to Restaurants
 "items": [
  {
   "foodItemId": ObjectId, // Reference to Food Items
   "quantity": Number,
   "price": Number
  }
 ],
 "totalPrice": Number,
 "status": { type: String, enum: ["Pending", "Confirmed", "Cancelled", "Delivered"] },
 "createdAt": { type: Date, default: Date.now }
}
```

8. Donations

```
{
 "_id": ObjectId,
 "restaurantId": ObjectId, // Reference to Restaurants
 "ngoId": ObjectId, // Reference to NGOs
 "items": [
  {
   "foodItemId": ObjectId, // Reference to Food Items
   "quantity": Number
  }
 ],
 "status": { type: String, enum: ["Pending", "Accepted", "Rejected", "Delivered"] },
 "deliveryOption": Boolean,
```

"createdAt": { type: Date, default: Date.now }

}

Indexing and Relationships

- **Indexing Locations:** For queries related to location (e.g., finding nearby restaurants or customers), create 2dsphere indexes on the location fields in the Restaurants, NGOs, and Customers collections.
- **Referential Integrity:** Use ObjectId references to ensure relationships between entities (e.g., restaurantId in Restaurant Admins, foodItemId in Menu, etc.).
- **Authentication & Authorization:** Ensure password fields are hashed and salted. Use middleware or services for authentication (e.g., JWT) and authorization to handle roles and permissions.