**Overview**

The Cornucopia App is a restaurant finder application that helps users discover new restaurants in the Cincinnati area. The app consists of a front-end and a back end, with the back end being responsible for managing the restaurant data, handling user requests, and communicating with external APIs.

The back end of the Cornucopia App will be built using Node.js and the Express.js framework. The data will be stored in a MongoDB database and accessed using the Mongoose library. The app will use the Google Maps API to provide location and map-related services.

**Architecture**

The back end of the Cornucopia App will consist of the following components:

**Server**

The server will be implemented using Node.js and the Express.js framework. It will handle incoming requests from the client, process them, and return the appropriate response. The server will use middleware to perform tasks such as logging, error handling, and authentication.

The server will also implement a RESTful API, which will allow the client to interact with the server using HTTP requests. The API will be designed to be intuitive and easy to use, with well-defined endpoints and response formats.

**Database**

The data for the Cornucopia App will be stored in a MongoDB database. The data will include restaurant information such as name, location, cuisine type, and reviews. The Mongoose library will be used to access the database and define the schema for the data.

Mongoose is an ORM (Object-Relational Mapping) library that provides a simple and intuitive interface for working with MongoDB. It allows developers to define models for their data, which can then be used to perform CRUD (Create, Read, Update, Delete) operations on the database.

**APIs**

The Cornucopia App will use the Google Maps API to provide location and map-related services. The app will use the Places API to search for restaurants based on the user's location and provide additional information such as photos and reviews.

**Endpoints**

The following endpoints will be implemented in the Cornucopia App:

**GET /API/restaurants**

This endpoint will return to a list of restaurants based on the user's location. The endpoint will accept query parameters such as location, radius, and cuisine type to filter the results. The response will include basic information about each restaurant such as name, address, and cuisine type.

Example request:

GET /api/restaurants?location=Cincinnati&radius=10&cuisine=Italian

Example response:

{

"results": [

{

"name": "Pizzelii Brick Oven Pizza",

"address": "7639 Wooster Pike, Cincinnati, OH 45227",

"cuisine": "Italian"

},

{

"name": "Maggiano's Little Italy",

"address": "7875 Montgomery Rd, Cincinnati, OH 45236",

"cuisine": "Italian"

}

]

}

**GET /api/restaurants/:id**

This endpoint will return detailed information about a specific restaurant. The response will include information such as name, address, phone number, website, cuisine type, photos, and reviews.

Example request:

GET /api/restaurants/12345

Example response:

{

"name": "Pizzelii Brick Oven Pizza",

"address": "7639 Wooster Pike, Cincinnati, OH 45227",

"phone": "(513) 271-2333",

"website": "https://www.pizzelliipizza.com/",

"cuisine": "Italian",

"photos": [

"https://www.example.com/photo1.jpg",