Nodejs, Mongodb

- Express.js, Create endpoints

Method	Function
POST	- User Login
POST	- User Logout
GET	- search by user firstName and/or lastName (sortable)
GET	- get user by id
POST	- create user
PUT	- update user
POST	- add address to user
GET	- search orders (by any field(s) in the document) (sortable)
GET	- get order by id
POST	- create order (only logged in users can create an order)
GET	- search items (by any field(s) in the document) (sortable)
GET	- get item by id
POST	- create an Item
PUT	- Update an item
DELETE	- Delete an item
GET	- search restaurants (by any field(s) in the document)
GET	- get restaurant by id
POST	- create a restaurant
PUT	- Update a restaurant

- Mongoose.js, Create schema

  (minimum fields to collect, add more as you see fit to complete the task)
  UserSchema
  firstName: String
  lastName: String
  email: String
  password: String - to store a password follow best practices
  addresses: [AddressSchema] - list of user's addresses
  createdAt: Date
  isActive: Boolean by default true

  AddressSchema
  street: String

city: String
state: String
zip: String
country: String

TokenSchema
token: string
user: id of user

ItemSchema
name: String
description: String
price: number
dietaryRestrictions: [String]
isActive: Boolean by default true

RestaurantSchema
name: String
description: String
geo: follow best practices
items: items sold at the restaurant
isActive: Boolean by default true

OrderSchema
user: ID - who ordered food
restaurant - From which location
createdAt: Date
isPaid: Boolean by default false
totalAmount: Int
Items: [item id]

Analytics
Write endpoints that will return

1) Number of orders, min, max and average totalAmount per hour filtered by restaurant
2) Total sales per day per restaurant
3) Most popular item by restaurant
4) Most popular item per user filtered by restaurant

- Axios, Write a test to test all endpoints, use lib
- Use async/await syntax, no callback

Bonus, revoke token after 5 minutes of inactivity. (inactivity defined as no api request)
Bonus deploy your app with a custom domain to be accessible via http