Spartans AI Node.JS Test Task - Pizza place simulator (backend)

Build a web app with the following parameters:

1. The app must be written in **Node.JS** and DB must be **MongoDB**

2. You can use NPM packages (only – no yarn or other package managers), but keep in mind that we want to see your coding abilities

3. You must use Git and use it as you would in a real project (commit, branch, etc.)

4. Import attached JSON to the separate collection in DB (**do not edit it**)

5. Use 'replay all' to send an email which contains a link to GitHub repo, and any additional info if needed (which scripts to run, config files locations, etc.) within the **36 hours from receiving the test.**

REST API

API consists of two parts:

1. Public endpoints

2. Admin endpoints

Public area:

1. Pizza creator:

- Endpoint for pizza(s) creation where user can add any ingredient (see attached JSON) to their pizza(s)
- Users can add ingredients only once per pizza.
- For successful order, the user needs to leave their info also (first name, last name, address, phone number)
- Return the wait time and place in the queue (see backend section below)

2. Users can cancel orders at any time while they wait.

3. Users can check if their pizza is finished.

4. Recent orders (from all users) that other users can order

Admin area can only be accessed by logging in and needs to have the following:

1. Top 5 ingredients ordered

2. Total money earned

3. Total (app) working time

4. Work time since the last start

5. Order history (ingredients, price, contact details, timestamps)

Backend

Chief can make only one pizza at the time.

The time needed to make one depends on the size of pizza + each ingredient modifier in ms (same goes for pricing):

Small - 1s - 200$

Medium - 2s - 400$

Large - 3s - 600$

As soon as pizza is complete the chief will announce that pizza is finished (simple console log will do, but if you can implement something better it would be a big plus).

Pizza place will take up to 15 orders at the same time.

Any order after this number will be kindly rejected until the queue is not freed up.

If the app crashes or if a user stops the app, when the app starts again it needs to continue from the previous state. That includes orders and contacts history, time left for pizza being made, and order queue.

There is no UI for this task.

Keep performance and (potential) scalability in mind.

Good luck! 🤖😄