

Full stack code assessment

Functional requirements - taco challenge



Imagine I'm a user who lives in Cape Town and I'm feeling like a mexican party
Thinking of taco and tequila pairing but all I've got is just that bottle of Olmeca
It's raining outside so I don't feel like stepping out
I go to some food ordering website and just type *Taco in Cape Town* in the search box

After I click search, the list of all the restaurants that have tacos comes up. Meaning they either have tacos as separate menu items or they have a Taco menu category.

<Search keyword> restaurants in **<location>** for you:

- <Restaurant Name 1> - <Suburb 1> - rated # <Rank> overall
- <Restaurant Name 1> - <Suburb 2> - rated # <Rank> overall
- <Restaurant Name 2> - <Suburb 3> - rated # <Rank> overall

The list should be sorted in the following way:

- Relevance - those who have more taco items on the menu - ie by the # of occurrences
e.g. Beef Taco, Chicken Taco, Vegan Taco in the menu itself or items under the Taco menu category come first
- The Rank field (which is part of the "rated as # ... " line on the mockup)

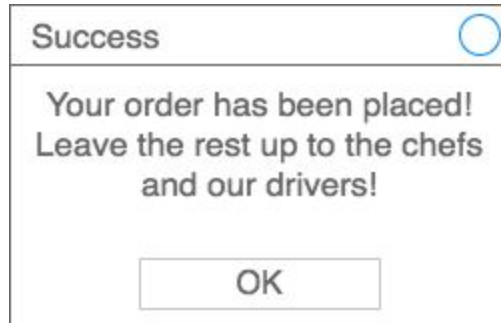
The list should be displayed in hierarchical order, having restaurant name + suburb first, category (if any) on the sub level and the menu items (**only** the relevant ones) under the category

The menu items must have a checkbox next to them allowing me to mark the ones I like (can be from different restaurants) so I can order them all by hitting that big Order Now button.

Please see this mockup below for the example:

The mockup shows a web browser window with a tab labeled 'Feed me now' and a URL bar containing 'https://localhost'. The main content area features a search bar with the text 'Taco in Cape Town' and a magnifying glass icon. Below the search bar, a heading reads '**Taco** restaurants in **Cape Town** we found for you:'. There are two restaurant listings. The first listing is for 'El Burro - Gardens - rated #1 overall', preceded by a placeholder 'IMG'. It lists three menu items: 'Classic Chicken Taco - R99' (unchecked), 'Crispy Battered Fish Taco - R120' (unchecked), and 'Mexican Chilli Beef Taco - R105' (checked). The second listing is for 'Fat Cactus - Woodstock - rated #3 overall', also preceded by a placeholder 'IMG'. It lists two menu items: 'Chipotle Steak - R135' (unchecked) and 'Falafel - R70' (checked). At the bottom center, there is a large green button labeled 'Order - R175'.

Clicking the **Order Now** button must trigger a backend call and confirm the order placement in the case of successful response.



The rest you leave to us and the chefs!

Implementation

- You will **not** be judged on graphic design and colors - we've got good designers of our own here!
- Frontend framework - you may use any client-side technology you want. However, if you're comfortable in the technology we use (React) then it would be great. You can use either a SPA approach (i.e. REST API plus JS) or a server-side approach (e.g. ASP.NET MVC)
- You can use off-the-shelf libraries and frameworks to speed up your progress
- Backend - .Net Core - how would you design it? which RESTful APIs can you see here?
- Unit / integration tests are not required, but will get you bonus points
- Database - please use the attached SampleData.json as your data set. Additionally add the ability to use any another data source (SQL Server, Postgres, Mongo etc) based on the app's configuration (you do not have to implement this secondary data source or populate it with any data)
- All the code must be committed to a git repository, and the progress should be committed regularly. Create a public repo on Github or on another service, and send us the link as soon as you start
- As you're working on the project, be sure to send us any questions you may have. If you're stuck with something, also let us know - we'll be happy to point you in the right direction, and we'll make ourselves available to respond timeously