Cheferyone

As an application, we believe that Cheferyone deserves the coolness points firstly due to the problem it is trying to solve. We live in an unprecedented time with COVID-19, and while it shuts down some markets, it also opens the opportunity for a new one to emerge. In our case, we want to help the home chefs who might require a streamlined way of promoting what they can offer. It also gives customers new options, as they might get the chance to try out new dishes or cuisines which might have not been mainstream otherwise.

In terms of the technical implementation, we have implemented several features which distinguish us from run-of-the-mill projects. To name a few: Algolia integration, personalised recommendation, and geolocation features.

As a start, we have integrated Algolia to help with our search feature. With Algolia, our search feature is more complex than a simple substring match in the database. Firstly, we can return entries which have a match in either the menu ‘name’ or ‘description’ field. Moreover, due to Algolia’s search algorithm, we can even return matches when users made typos in their query! For example, typing ‘chicne’ as a query still returns matches for which the menu has ‘Chicken’ in their name or description. Combined with the tagging feature, we can have advanced search which improves the discoverability of the chefs. These will serve us well in creating more powerful searches should we expand the application even further as we are able to configure more search options.

Another thing about our application is that we have also implemented a recommendation algorithm that is catered to the different users. While it is not as sophisticated as Machine Learning algorithms, it is still sure to produce meaningful results. Not sure what menu to get? Feeling a bit adventurous? We’ve got you covered! Let our application do the hard work of looking for what suits your taste buds best. How cool is that?

On top of that, we have also integrated Geolocation features. We understand that some users might not be keen on travelling all the way from the east to west of Singapore just to collect their food. That is why we also give users a listing of menus which are near to them to reduce the hassle of travelling, especially during a pandemic.

In terms of UI/UX, we have implemented the API endpoints with pagination for the frontend in mind. When we have a list of items, not all of the items will be displayed to the user directly, this gives them some breathing room and they won't be overwhelmed. This also gives a better user experience, as loading a few items rather than a long list will give a better response time and gives the impression that the application is fast. With this, we are able to implement infinite scrolling as well.

A factor that makes Cheferyone usable for public use is the fact that we have 2 interfaces, one each for the appropriate user. For example, a customer might make an order to the home chef and have paid for it. In the restaurant side, this same order will be available for viewing by the home chef, with an entirely different set of actions available. The chef might choose to confirm this order, an option that is not available for the customer.

As an extension to this, we have also implemented a notification system. Customers can subscribe to their favourite home chefs and get notified for any new menu that this home chef might prepare in the future. Customers need not worry about needing to constantly check the home chef page for existing menus with a new pre-order date or for the creation of a new menu entirely. Never miss-out on your favourite home chefs anymore!

Finally, to ensure reliability of the application, we have implemented both client-side and server-side validations in our application to handle erroneous cases. For example, a preorder's start date must come earlier before the preorder's end date. In the frontend, the 'confirm' button is disabled when the user tries to do otherwise, and if for some reason the user manages to send an erroneous request to the server, then the backend will also have additional checks to prevent the request from being processed.

All in all, for the reasons elaborated above, we believe that Cheferyone will provide existing home chefs with a solid platform to start a business from the comfort of their homes. Gone are the days when they have to tirelessly advertise on social media to gain a customer following. With Cheferyone, we believe that everyone can be a chef

Interested? Try the application here: https://cheferyone.herokuapp.com/