

Junior Front-End Developer Assessment

Το YummyGreek χρειάζεται την βοήθεια σου!

To Challenge

Σε αυτό το challenge, καλείσαι να βοηθήσεις ένα νέο Ελληνικό εστιατόριο με όνομα "YummyGreek.", να διαχειριστεί καλύτερα τις καθημερινές του διαδικασίες.

Λόγω της ιδιαιτερότητας του, το επόμενο διάστημα περιμένει τους διπλάσιους πελάτες και μάλιστα είναι από τα λίγα εστιατόρια στον κόσμο με το βραβείο "Χωρίς Αποκλεισμούς". Αυτό δημιουργεί την ανάγκη για ύπαρξη διάφορων φίλτρων.

Ο κύριος στόχος του challenge είναι να δημιουργηθεί μια νέα responsive εφαρμογή που θα επιτρέπει στους πελάτες να παραγγείλουν το δείπνο τους, χρησιμοποιώντας το tablet που θα βρίσκεται σε κάθε τραπέζι του εστιατορίου και τέλος να αφήνουν και την κριτική τους.

Περιγραφή

Το εστιατόριο θέλει να προσφέρει μια ιδιαίτερη εμπειρία στους πελάτες του και αυτό αρχίζει με το που καθίσουν στο τραπέζι του. Οι πελάτες θα πρέπει να βλέπουν το μενού του εστιατορίου και με εύκολο τρόπο να μπορούν να διαβάσουν πληροφορίες για κάθε ένα από τα στοιχεία του μενου.

Επιπλέον θα πρέπει να μπορούν να φιλτράρουν το μενού καθώς ένας πελάτης μπορεί να είναι Vegan ή να έχει αλλεργίες ή να είναι διαβητικός.

Όταν ο πελάτης αποφασίσει τι θέλει, θα πρέπει να μπορεί να το προσθέσει στο καλάθι και στο τέλος να γίνει η παραγγελία.

Technical Stack

- Επίλεξε όποιο JS framework θέλεις (π.χ., Vue3, Angular, React).
- Επίλεξε όποιο CSS framework θέλεις (π.χ., Tailwind, Bootstrap, Vuetify).

Προτείνουμε την δημιουργία SPA σε Vue3 με χρήση του Tailwind για CSS και του Pinia για την διαχείριση του state της εφαρμογής.

Αν έχεις εμπειρία σε κάτι διαφορετικό, έχεις το ελεύθερο να τα επιλέξεις. Σημασία έχει να δημιουργήσεις το καλύτερο δυνατό αποτέλεσμα, συνάρτηση των γνώσεών σου αλλά και των απαιτήσεων που ακολουθούν.

Λειτουργικές Απαιτήσεις

Απαραίτητη Λειτουργικότητα

Η εφαρμογή θα πρέπει να έχει τα παρακάτω χαρακτηριστικά:

1. Προβολή όλων των πιάτων του εστιατορίου.
2. Προσθήκη φίλτρων για τα πιάτα.
 - a. Φίλτρο για Διαιτητικές Προτιμήσεις (Dietary Preferences):
 - i. Χορτοφαγικό (Vegan)
 - ii. Χωρίς Γλουτένη (Gluten-Free)
 - iii. Χωρίς Λακτόζη (Lactose-Free)
 - iv. Χωρίς Καρυκεύματα (Nut-Free)
 - b. Φίλτρο για Εύρος Τιμής (Price Range):
 - i. Κάτω από 10€ (Under €10)
 - ii. €10 - €20 (€10 - €20)
 - iii. €20 - €30 (€20 - €30)
 - iv. €30 - €40 (€30 - €40)
 - v. Άνω των €40 (Over €40)
 - c. Φίλτρο για Αλλεργίες (Allergens):
 - i. Περιέχει Ξηρούς Καρπούς (Contains Nuts)
 - ii. Περιέχει Γλουτένη (Contains Gluten)
 - iii. Περιέχει Λακτόζη (Contains Lactose)

- iv. Περιέχει Άλλα Αλλεργιογόνα (Other Allergens)
 - v. Χωρίς Αλλεργιογόνα (Allergen-Free)
3. Δυνατότητα προβολής ενός πιάτου.
 4. Δυνατότητα παραγγελίας ενός πιάτου.
 5. Δυνατότητα προβολής της παραγγελίας σου.

Οθόνη #1 - Προβολή όλων των πιάτων.

Στη σελίδα αυτή, πρέπει να υπάρχει μια λίστα με όλα τα διαθέσιμα πιάτα που προσφέρει το εστιατόριο. Σε κάθε πιάτο, πρέπει να υπάρχει μια επιλογή για μετάβαση στην οθόνη προβολής του πιάτου, καθώς και μια επιλογή για προσθήκη στο καλάθι.

Οθόνη #2 - Προβολή των πληροφοριών ενός πιάτου.

Στη σελίδα αυτή, υπάρχουν όλες οι πληροφορίες για ένα πιάτο. Σε αυτό το σημείο ο χρήστης θα πρέπει να μπορεί να πάει ένα βήμα πίσω (στην οθόνη #1), καθώς και να προχωρήσει με την προσθήκη του πιάτου στο καλάθι.

Οθόνη #3 - Προβολή της παραγγελίας

Στη σελίδα αυτή, θα μπορεί να δει την παραγγελία που έχει κάνει και να στείλει την τελική του παραγγελία στο εστιατόριο. Δεν θα χρησιμοποιηθεί κάποιο endpoint αλλά ο κώδικας πρέπει να είναι σε σημείο που αν σου δοθεί, να μπορεί να υλοποιηθεί με όσο το δυνατόν λιγότερες αλλαγές.

Claim your Bonus

Η εφαρμογή θα πρέπει να είναι responsive.

Ακολούθησε τα best practices βάση των official documentations.

- Η εφαρμογή θα χρησιμοποιηθεί τόσο σε laptop όσο και σε tablet. Η δυνατότητα να είναι fully responsive είναι plus!.
- Ακολούθησε το Style Guide που θα βρεις στα official documentations του framework που θα επιλέξεις (π.χ. για το Vue3 δες [αυτό το Style Guide](#)).
- Μπορείς να χρησιμοποιήσεις Unit Test για όλα τα κρίσιμα σημεία της εφαρμογής χρησιμοποιώντας το testing framework της επιλογής σου (π.χ. Για το Vue3 δες [αυτό το Unit Test Documentation](#)).
- Χρήση του State Management.

API

Για τα dummy δεδομένα θα χρησιμοποιήσεις το αρχείο **db.json**, Εκεί θα βρεις όλα τα πιάτα του εστιατορίου. Μπορείς να το χρησιμοποιήσεις όπως θέλεις, ωστόσο, ο πιο απλός τρόπος είναι δημιουργώντας έναν json server.

Μάθε περισσότερα εδώ: <https://medium.com/codingthesmartway-com-blog/create-a-rest-api-with-json-server-36da8680136d>

Τα endpoints που θα χρησιμοποιηθούν θα πρέπει να είναι για να προβολή όλων των πιάτων το:

GET /dishes

και για την προβολή ενός πιάτου το:

GET /dishes/<id>

Στην εφαρμογή και στο README αρχείο θα πρέπει να υπάρχουν οδηγίες για το που και πως μπορούμε να τα αλλάξουμε αν αυτό χρειαστεί.

Κριτήρια αξιολόγησης

Η λύση σου θα αξιολογηθεί κριτηρίων κάποια από τα οποία είναι:

- Λειτουργικότητα
- Ποιότητα Κώδικα
- Σχεδιασμός του UI/UX
- Responsiveness

Παράδοση του project

Τα αρχεία που θα πρέπει να παραδοθούν, είναι **ΜΟΝΟ το frontend** της εφαρμογής. Κάθε αρχείο που σχετίζεται με το backend, **δεν** θα λαμβάνεται υπόψη. Υπενθυμίζουμε ότι στο API αναφέρεται πως θα πρέπει τα endpoints να είναι configurable.

Το project θα πρέπει να παραδοθεί ως εξής:

- Δημιούργησε ένα repository στο Github με όλα τα απαραίτητα αρχεία για να είναι λειτουργικό.
- Φτιάξε ένα README αρχείο που να καθοδηγεί πως μπορούμε να τρέξουμε την εφαρμογή.

Καλή επιτυχία!