

Υλοποίηση της Εργασίας:

Για την υλοποίηση της εργασίας, επιλέξαμε να αξιοποιήσουμε τις προτεινόμενες τεχνολογίες που περιγράφονται στην εκφώνηση. Συγκεκριμένα:

1. Για τη διαχείριση της βάσης δεδομένων, επιλέξαμε να χρησιμοποιήσουμε τη MySQL.
2. Στον τομέα των χαρτών, επιλέξαμε να χρησιμοποιήσουμε τη βιβλιοθήκη Leaflet.
3. Για τη δημιουργία διαγραμμάτων, επιλέξαμε τη βιβλιοθήκη chart.js.
4. Για την υλοποίηση ενός RESTful API, επιλέξαμε να χρησιμοποιήσουμε το express framework σε συνδυασμό με το Node.js.
5. Όλη η ανάπτυξη της εφαρμογής πραγματοποιήθηκε στο περιβάλλον του vscode.
6. Η επικοινωνία της ομάδας διαμορφώθηκε μέσω του GitHub για τη διαχείριση του κώδικα, ενώ χρησιμοποιήθηκε το Discord για την επικοινωνία.

Διαχείριση Χρηστών και Διαχειριστή:

Σύμφωνα με τις οδηγίες της εκφώνησης, τηρήσαμε τους περιορισμούς σχετικά με την εγγραφή στο σύστημα για τους χρήστες:

1. Δημιουργήσαμε την οντότητα του διαχειριστή (Admin) απευθείας στη βάση δεδομένων.
2. Δημιουργήσαμε έναν πίνακα "users" που περιέχει όλους τους χρήστες, συμπεριλαμβανομένου του διαχειριστή, και προσθέσαμε τα απαραίτητα πεδία για την εγγραφή τους.
3. Προσθέσαμε ένα λογικό πεδίο με το όνομα "is_user", το οποίο μας επιτρέπει να ξεχωρίζουμε τους απλούς χρήστες από τον διαχειριστή.

Σελίδα Χάρτη:

1. Αναπτύξαμε μια σελίδα χαρτών που είναι προσβάσιμη τόσο από τους χρήστες όσο και από τους διαχειριστές.
2. Χρησιμοποιήσαμε τη λειτουργία geolocation του Leaflet για να εμφανίσουμε την τοποθεσία των χρηστών στον χάρτη και δημιουργήσαμε markers που μπορούν να μετακινηθούν.
3. Ομαδοποιήσαμε τα καταστήματα που έχουν ενεργές προσφορές και τα αναπαράστήσαμε με διαφορετικά markers.
4. Προσθέσαμε δυνατότητες αναζήτησης ανά κατάσταση και ανά κατηγορία προϊόντων, παρέχοντας επίσης pop-up παράθυρα με πληροφορίες σχετικά με τα καταστήματα, συμπεριλαμβανομένων των προσφορών και των χαρακτηριστικών τους.
5. Επιπλέον, δημιουργήσαμε μια δυνατότητα υποβολής προσφορών μέσω μιας λίστας επιλογών που περιλαμβάνει κατηγορίες και υποκατηγορίες προϊόντων.
6. Κατανέμαμε δίκαια εικονικά νομίσματα (tokens) με βάση τον μηνιαίο βαθμό των χρηστών.

Σύστημα Tokens:

1. Χρησιμοποιήσαμε το εργαλείο cron για τον υπολογισμό των tokens βάσει των στατιστικών δεδομένων.
2. Αξιολογήσαμε τους χρήστες βάσει του ιστορικού τιμών προϊόντων και της συμπεριφοράς τους.
3. Στη συνέχεια, υπολογίσαμε την αξία του κάθε token βάσει του σκορ του κάθε χρήστη.

Αξιολόγηση Χρηστών:

1. Για τον υπολογισμό του σκορ αξιολόγησης χρησιμοποιήσαμε δύο διαφορετικές προσεγγίσεις, μία βασισμένη στο ιστορικό τιμών προϊόντων και μία στη συμπεριφορά των χρηστών. Χρησιμοποιήσαμε triggers στη MySQL για τον υπολογισμό του σκορ κατά την εισαγωγή και ενημέρωση δεδομένων.
2. Για τον υπολογισμό του σκορ βάσει του ιστορικού τιμών προϊόντων, ανακτήσαμε τα δεδομένα από τη βάση δεδομένων και χρησιμοποιήσαμε statements στη JavaScript για την αξιολόγηση. Έτσι, κάθε χρήστης αξιολογήθηκε βάσει του πόσες φορές προτίμησε ή δεν προτίμησε τις τιμές προϊόντων.
3. Για τον υπολογισμό του σκορ βάσει της συμπεριφοράς των χρηστών, χρησιμοποιήσαμε triggers στη MySQL κατά την εισαγωγή (insert) και την ενημέρωση (update) δεδομένων. Με τη χρήση αυτών των triggers, ανατίθενται τα points που αναγράφονται στην εκφώνηση σύμφωνα με τις ενέργειες των χρηστών, δηλαδή το αν προτιμούν ή απορρίπτουν προϊόντα.

Διαχείριση Διαχειριστή:

1. Χρησιμοποιήσαμε Python scripts για την ανάκτηση δεδομένων από το e-katanalotis, αν και υπήρξαν προβλήματα με τα scripts.
2. Τα δεδομένα που ανακτήθηκαν αποθηκεύτηκαν σε ένα αρχείο JSON, και υλοποιήθηκε ένα κουμπί ανανέωσης (upload button) που επιτρέπει την ανανέωση των δεδομένων.
3. Για την ανάκτηση των POIs (Points of Interest), χρησιμοποιήσαμε το Overpass API βάσει των δεδομένων του OpenStreetMap.

Παραθέτουμε παράδειγμα των δεδομένων που χρησιμοποιήσαμε:

```
In [17]: import json
file_path = r'C:\Users\stell\OneDrive\Desktop\file.json'
with open(file_path, "w") as json_file:
    json.dump(old_data, json_file)

In [ ]:

In [15]: import random

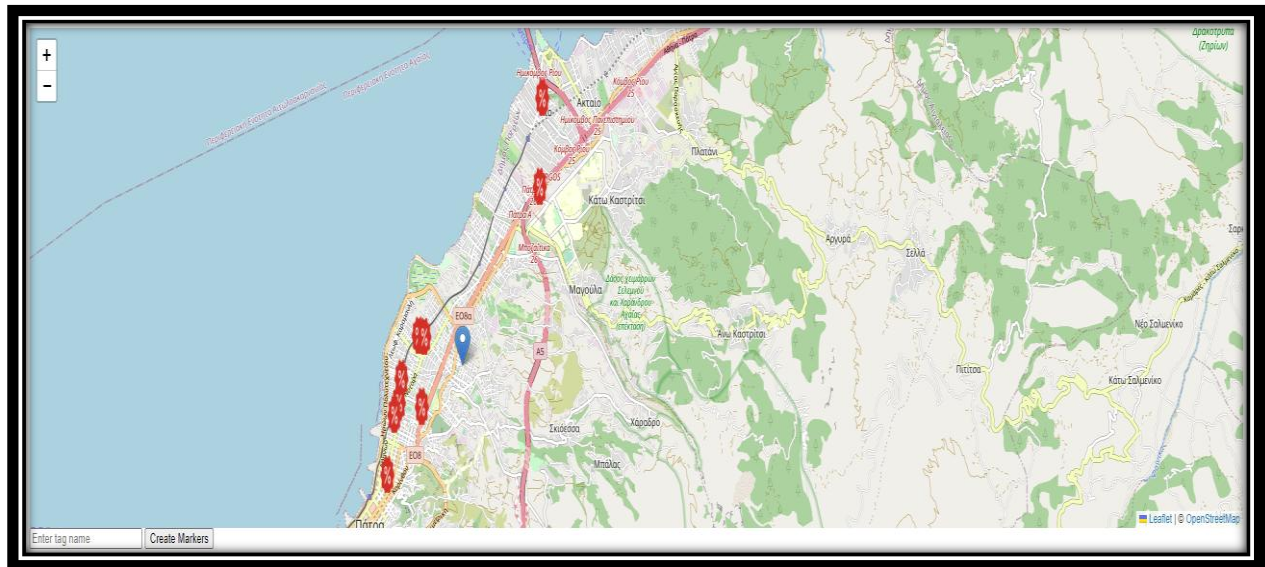
dates = [
    "2023-08-24", "2023-08-25", "2023-08-26", "2023-08-27", "2023-08-28", "2023-08-29", "2023-08-30", "2023-08-31",
    "2023-09-1", "2023-09-2", "2023-09-3", "2023-09-4", "2023-09-5", "2023-09-6", "2023-09-7", "2023-09-8",
    "2023-09-9", "2023-09-10", "2023-09-11", "2023-09-12", "2023-09-13", "2023-09-14", "2023-09-15", "2023-09-16", "2023-09-17",
    "2023-09-18", "2023-09-19", "2023-09-20"
]

def random_price_generator(min_val, max_val):
    return round(random.uniform(min_val, max_val), 2)

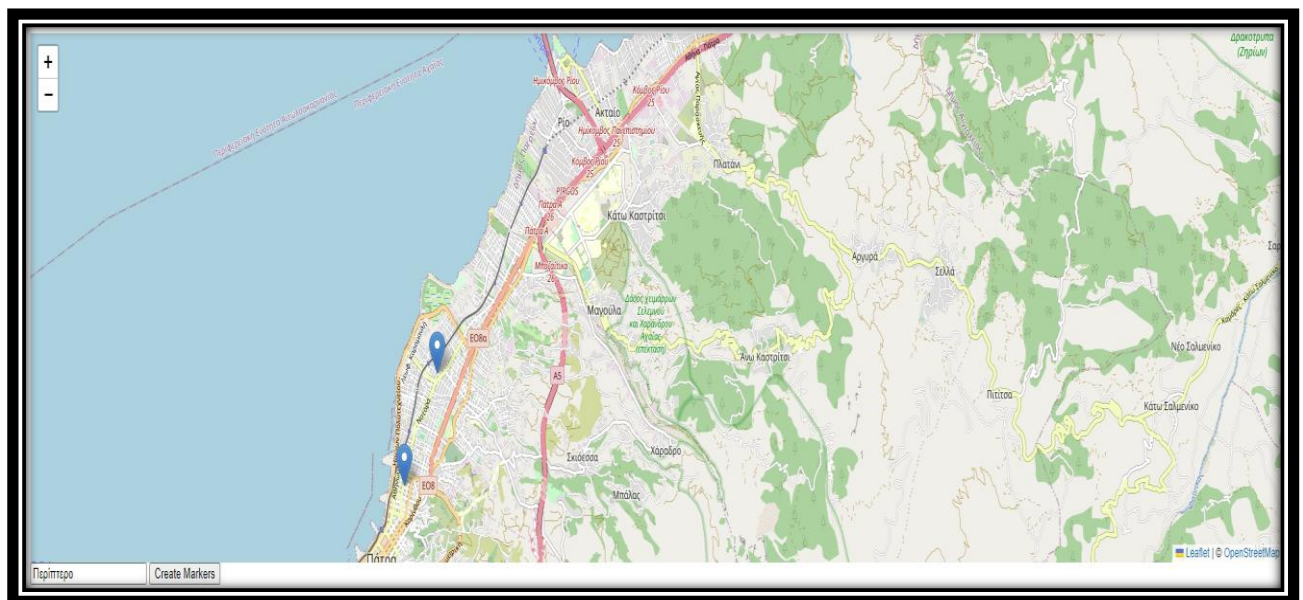
random_data1 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data2 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data3 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data4 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data5 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data6 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data7 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data8 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]
random_data9 = [{'date': date, 'price': str(random_price_generator(0.6, 1.6))} for date in dates]

In [16]: old_data = {
    'fetch_date': '2023-08-15',
    'products': [
        {
            'id': '1',
            'name': 'Εύρηκα Σκόνη Antikalk 54γρ',
            'category': 'Καθαριότητα',
            'subcategory': 'Απορρυπαντικά',
            'prices': random_data1
        },
        {
            'id': '2',
            'name': 'Palmolive Υγρό Πιάτων Regular 500ml',
            'category': 'Καθαριότητα',
            'subcategory': 'Απορρυπαντικά',
            'prices': random_data2
        },
        {
            'id': '3',
            'name': 'Palmolive Υγρό Πιάτων Λεμόνι 500ml',
            'category': 'Καθαριότητα',
            'subcategory': 'Απορρυπαντικά',
            'prices': random_data3
        },
        {
            'id': '4',
            'name': 'Svelto Gel Υγρό Πιάτων Λεμόνι 500ml',
            'category': 'Καθαριότητα',
            'subcategory': 'Απορρυπαντικά',
            'prices': random_data4
        }
    ]
}
```

- ❖ Στη συνέχεια επισημαίνουμε κάποια screenshots τα οποία αποτελούν τη βασική δομή και τα πιο σημαντικά σημεία της εργασίας:



1) Markers καταστημάτων προσφορών



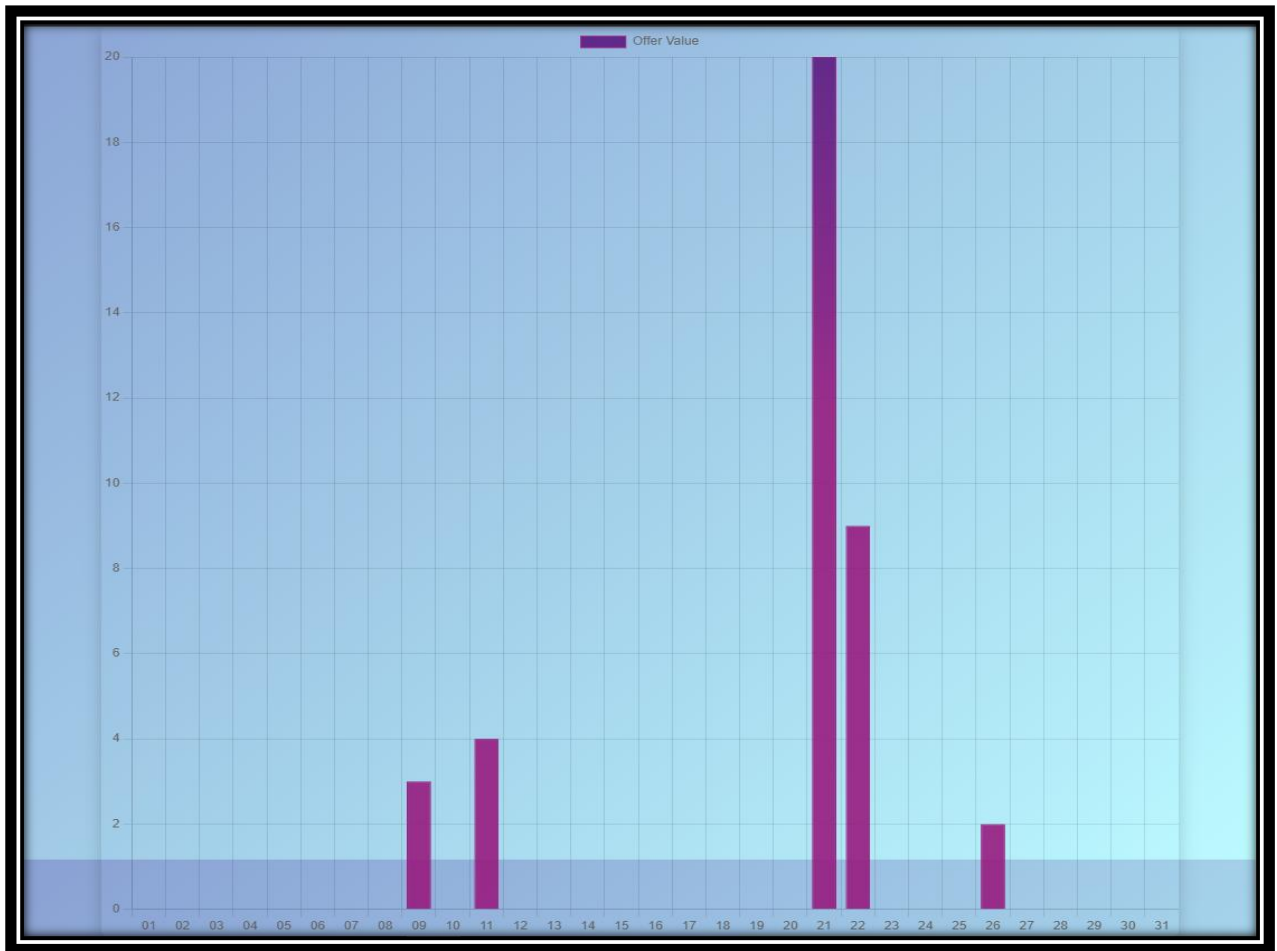
2) Markers αναζήτησης καταστήματος χωρίς ενεργές προσφορές



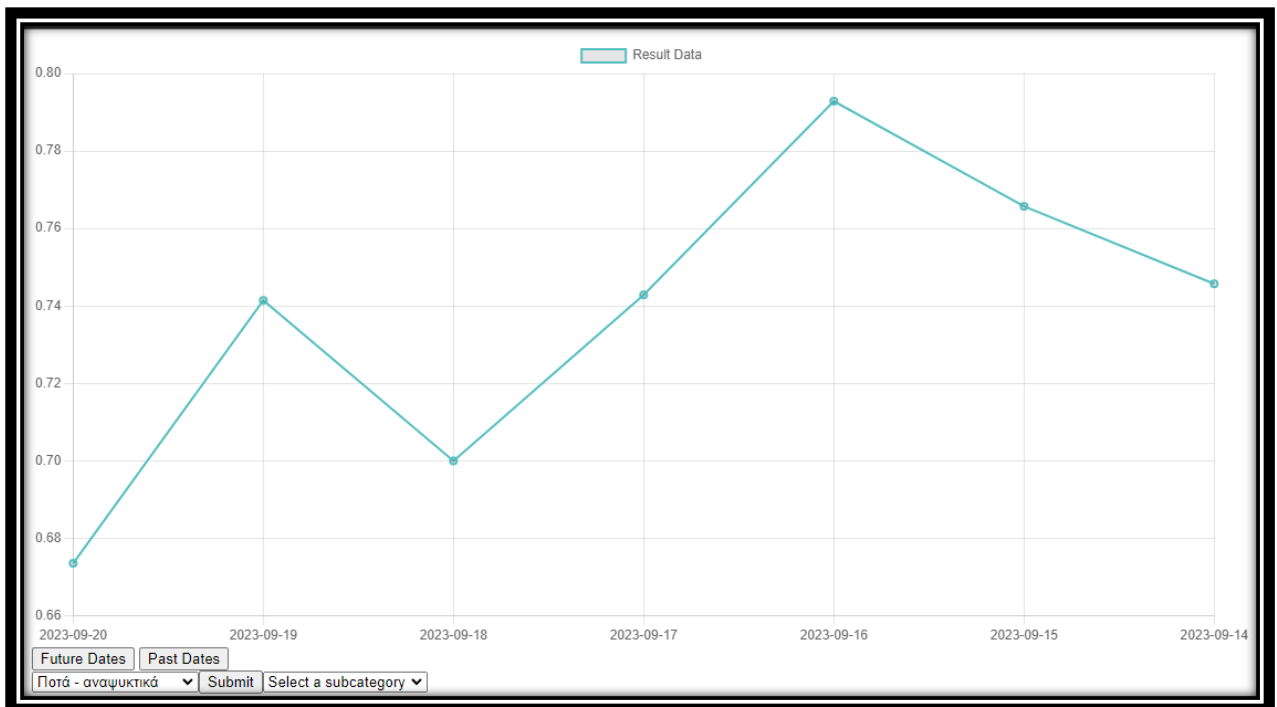
3) *Navigation των γραφημάτων του admin*



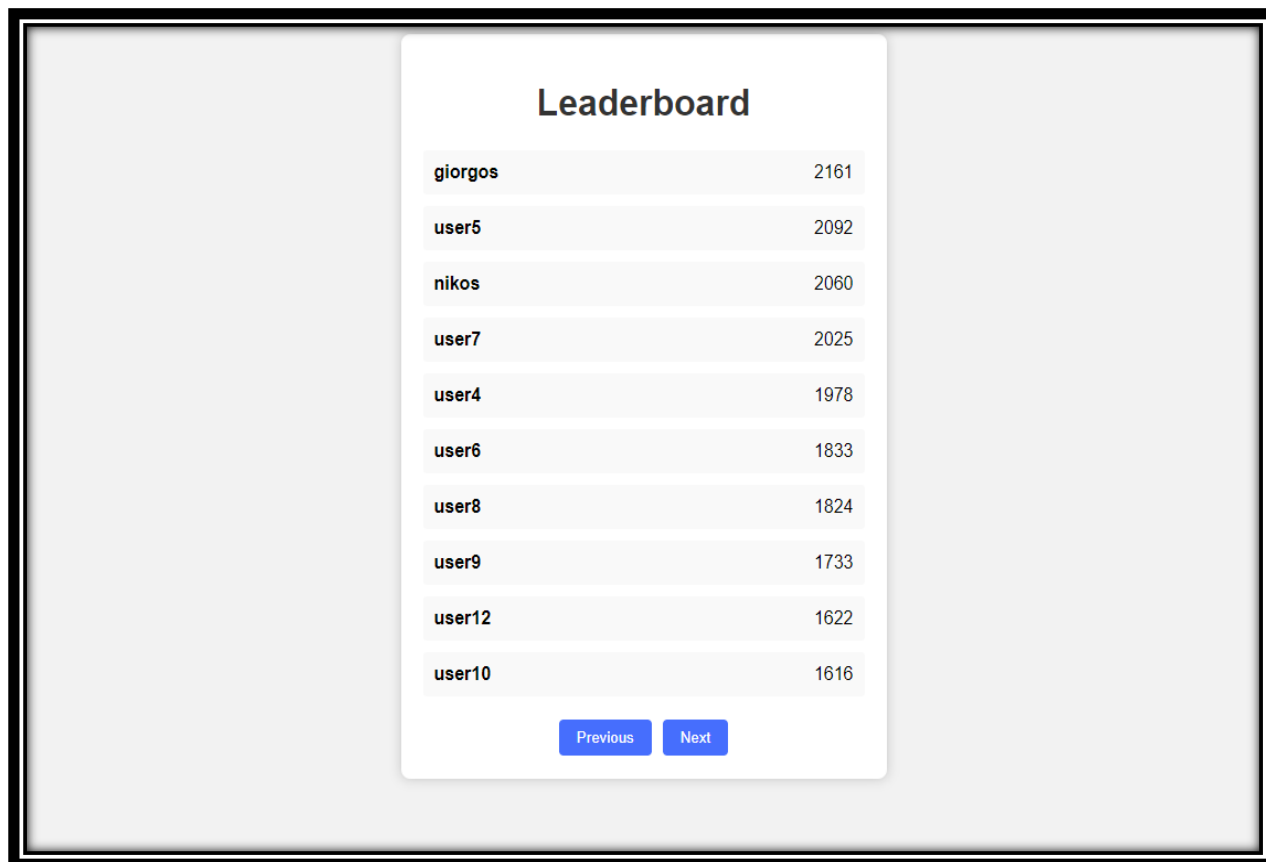
4) *Navigation των γραφημάτων του admin*



5) Chart ερωτήματος 3α Admin για τον μήνα Αύγουστο 2023



6) chart ερωτήματος 3β admin με επιλεγμένη μόνο γενική κατηγορία και δυνατότητα πλοήγησης σε προηγούμενες ημερομηνίες



7) Απεικόνιση LeaderBoard με σελιδοποίηση ανά 10