

Προγραμματισμός & Συστήματα στον Παγκόσμιο Ιστό **Εργαστηριακή Άσκηση Ακαδημαϊκού Έτους 2019-2020**

Συστήματα πληθοπορισμού καταγραφής και αξιοποίησης των
δραστηριοτήτων των χρηστών

Θεόδωρος Συμεωνίδης 1064870

Θεόδωρος Τωμαδάκης 1047073

Κωδικός υποβολής

MedzcEphSFdvY2ZQUTU2V1FIN0MyUWVWNmNTb3B6aVRPVlhxVEpsdXptYz06Ov4Mf2iC16
dNKiVeH47SloE=

Τεχνολογίες που χρησιμοποιήθηκαν

Η υλοποίηση έγινε σε MERN stack. Δηλαδή ως βάση δεδομένων χρησιμοποιήθηκε η MongoDB (η χρήση της αιτιολογείται παρακάτω), ως backend ExpressJS, ως frontend ReactJS και ως runtime environment για την javascript NodeJS. Σχετικά με τις τεχνολογίες που απαιτείται η χρήση τους, χρησιμοποιήθηκαν με wrappers σε React components, η React-Bootstrap για το CSS, και η react-chartjs-2 για το Chart.js. Χωρίς wrappers χρησιμοποιήθηκαν η Leaflet και η leaflet-heatmap.js. Για την AJAX επικοινωνία χρησιμοποιήθηκε η βιβλιοθήκη Axios. Για την γεωγραφική επεξεργασία των δεδομένων στο client side χρησιμοποιήθηκε η βιβλιοθήκη turf.js. Επίσης χρησιμοποιήθηκαν πολλές ακόμα βιβλιοθήκες τις οποίες μπορείτε να βρείτε στα αρχεία package.json.

Διαδικασία ανάπτυξης

Σχετικά με την χρήση MongoDB, αρχικά έγινε προσπάθεια να χρησιμοποιηθεί PostgreSQL με τη χρήση της βιβλιοθήκης PostGIS και πράγματι γράφτηκε όλο το data storage και retrieval σε PLPGSQL προτού διαπιστωθεί ότι δεν λειτουργεί καλά για την περίπτωση μας. Η λογική που ακολουθήθηκε και πιστέψαμε ότι θα λειτουργήσει ήταν ότι στην βάση δεδομένων θα αποθηκεύονται όλα τα σημεία του LocationHistory.json για τον κάθε χρήστη σε ειδικούς τύπους δεδομένων που προσφέρει το PostGIS και όταν ο διαχειριστής έβλεπε μια συγκεκριμένη περιοχή του χάρτη θα προσκομίζονταν μόνο τα σημεία που ανήκανε σε αυτή την περιοχή, αυτό πιστεύαμε θα διευκολυνόταν και από τα ειδικά indexes για γεωγραφικά δεδομένα (KD-trees κτλ.) που προσφέρει η βιβλιοθήκη. Ωστόσο αυτή η προσπάθεια μας ναυάγησε καθώς η εισαγωγή όλων των δεδομένων για κάθε χρήστη (περίπου 200.000 inserts για ένα τυπικό αρχείο δραστηριοτήτων) ήταν απαγορευτική και απαιτούσε περίπου 5-10 λεπτά καθώς οι SQL βάσεις δεδομένων κάνουν συγχρονισμένα inserts. Αυτό μας οδήγησε αρχικά να κάνουμε δειγματοληψία των δεδομένων στο πριν σταλούν στο backend και ακόμα να γίνει η χρήση NoSQL βάσης δεδομένων όπως η MongoDB.

Η υλοποίηση της βάσης δεδομένων σε PostgreSQL μπορεί να βρεθεί στο αρχείο postgresql-storage.sql και η υλοποίηση του backend σε PostgreSQL (δεν συμπεριλαμβάνεται στο παραδοτέο).

Χρήση GeoJSON

Η MongoDB περιέχει στους default τύπους της υποστήριξη για GeoJSON δεδομένα και αποφασίσαμε να τους χρησιμοποιήσουμε καθώς είναι ένα πρότυπο που βοήθησε στην επικοινωνία μεταξύ βιβλιοθηκών (π.χ. με την turf.js). Το αρχείο LocationHistory.json δεν είναι ωστόσο σύμφωνο με το πρότυπο αυτό. Έτσι χρειάζεται μόλις το επιλέξει ο χρήστης για ανέβασμα να το περάσουμε από τη φάση της προεπεξεργασίας. Από εκεί και πέρα τα δεδομένα δεν απαντώνται ποτέ στην αρχική τους μορφή όπως βρίσκονται στο αρχείο δραστηριοτήτων. Ωστόσο δεν χάνεται κανένα από τα αρχικά δεδομένα παρά μόνο η δομή τους αλλάζει. Μετά το στάδιο της προεπεξεργασίας οποιαδήποτε πράξη γίνεται στα δεδομένα συμμορφώνεται στο πρότυπο GeoJSON.

Μοντέλο βάσης δεδομένων

users			...
_id	string	NN	
name	string	NN	
email	string	NN	
password	string	NN	
type	string	NN	
__v	double	NN	
activities {}	users.activities	NN	
_id	objectId	NN	
type	string	NN	
features [{}]	users.activities.features	NN	
_id	objectId	NN	
type	string	NN	
geometry {}	users.activities.features.geometry	NN	
coordinates []	double	NN	
_id	objectId	NN	
type	string	NN	
properties {}	users.activities.features.properties	NN	
_id	objectId	NN	
timestamp	date	NN	
type	string	NN	
confidence	double	NN	
velocity	null		
accuracy	null		
altitude	null		
lastUpload	date	NN	
ecoScore {}	users.ecoScore	NN	
history []	double	NN	
overall	double	NN	

Ουσιαστικά παρατηρούμε ότι κάθε χρήστης έχει ένα GeoJSON αντικείμενο activities και εκεί αποθηκεύονται όλες οι πληροφορίες από το αρχείο δραστηριοτήτων του. Επίσης περιέχει κάποια ακόμα πεδία που έχουν σχέση με τα στατιστικά των δραστηριοτήτων (lastUpload, ecoScore) του και ενημερώνονται κάθε φορά που ανεβάζει κάποιο αρχείο.

Εκτέλεση

Για την εκτέλεση θα πρέπει να υπάρχει εγκατεστημένη η τελευταία έκδοση της NodeJS. Στην συνέχεια εκτελείτε τα παρακάτω βήματα σε bash shell (το προεπιλεγμένο shell για τα περισσότερα λειτουργικά συστήματα που βασίζονται στο Linux).

1. Μετάβαση στο φάκελο `src` και εκτέλεση της εντολής `npm install && npm start`
2. Μετάβαση στο φάκελο `src/client` και εκτέλεση της εντολής `npm install && npm start`