

Εργασία (ατομική): Κατασκευή απλού REST API και Web Εφαρμογής

Το project είναι προαιρετικό και δίνει ως 4 (+ bonus σε εξαιρετική επίδοση) μονάδες στον τελικό βαθμό, αφορά δε σε όλους όσους χρωστούν το μάθημα. Κάνοντάς το θα αποκτήσετε εμπειρία σε:

- α) δημιουργία αλληλεπιδραστικής web εφαρμογής (με χρήση AJAX κλήσεων)
- β) υλοποίηση και χρήση REST API,
- γ) χρήση Google Maps JavaScript API καθώς και σε χρήση κάποιου PHP framework (Slim, Laravel, κτλ)

Αντικείμενο της εργασίας είναι η δημιουργία μιας αλληλεπιδραστικής web εφαρμογής που τροφοδοτείται με δεδομένα μέσω ενός web API τύπου REST, που επίσης θα πρέπει να κατασκευάσετε.

Θα σας δοθεί το schema μιας σχεσιακής βάσης δεδομένων (σε αρχείο MySQL Workbench) καθώς και δεδομένα σε μορφή SQL κώδικα για να τη γεμίσετε. Θα πρέπει να δημιουργήσετε με αυτά μια λειτουργική σχεσιακή βάση δεδομένων (σε MySQL ή ισοδύναμο RDBMS). Τα δεδομένα είναι μερικό στιγμιότυπο (partial snapshot) της βάσης δεδομένων της εφαρμογής fuelGR. Όλες οι σχέσεις ακεραιότητας ξένων κλειδιών (on update, on delete) είναι σε ρύθμιση CASCADE.

Στην εφαρμογή υπάρχουν δύο ειδών χρήστες, οι ιδιοκτήτες πρατηρίου και οι απλοί καταναλωτές. Χωρίς login οι χρήστες της εφαρμογής απλά βλέπουν τον χάρτη με τα πρατήρια και μπορούν να επιλέξουν καύσιμο. Στο infobox ενός marker βλέπουν τις πληροφορίες του πρατηρίου και τον τιμοκατάλογό του. Μετά από login, οι ιδιοκτήτες πρατηρίου μπορούν επιπλέον να δουν τι παραγγελίες έχουν δεχθεί καθώς και να αλλάξουν τιμή σε κάποιο προϊόν που εμπορεύονται. Οι απλοί χρήστες, μετά από login μπορούν να καταχωρήσουν παραγγελία για κάποιο καύσιμο σε κάποιο πρατήριο.

Η web εφαρμογή θα πρέπει να λαμβάνει δεδομένα (πρατηρίων, τιμών για το καύσιμο επιλογής – default βενζίνη 95, και τιμοκαταλόγου πρατηρίου) και θα στέλνει δεδομένα (login, παραγγελιών) μέσω http κλήσεων στο REST API που θα κατασκευάσετε. Η web εφαρμογή θα πρέπει να έχει τον χάρτη ως κυρίαρχο στοιχείο. Λοιπά στοιχεία που τυχόν απαιτηθούν (π.χ. φόρμα login, υποβολής παραγγελίας, κτλ) μπορεί να υλοποιηθούν σε μορφή modal παραθύρων¹ (πρέπει να τα κλείσεις για να επιστρέψεις στην εφαρμογή) ή σε κάποιο sidebar.

Το REST API στο οποίο θα βασιστεί η εφαρμογή σας θα πρέπει να υποστηρίζει τις ακόλουθες κλήσεις:

1. GET: Λήψη δεδομένων πρατηρίων και τιμών επιλεγμένου καυσίμου.
2. GET: Πλήθος πρατηρίων (ακέραιος), μέγιστη, ελάχιστη και μέση τιμή ανά lt (με 3 δεκαδικά).
3. GET: Λήψη τιμοκαταλόγου ενός πρατηρίου.
4. POST: Login χρήστη (η database έχει ήδη χρήστες).
5. POST: Αποστολή δεδομένων παραγγελίας από πλευράς χρήστη (ποσότητα).
6. GET: Λήψη παραγγελιών πρατηρίου από τον πρατηριούχο
7. PUT: Αλλαγή τιμής σε καύσιμο από πρατηριούχο
8. PUT: Εκτέλεση παραγγελίας από πρατηριούχο
9. DELETE: Διαγραφή παραγγελίας από πρατηριούχο ή από απλό χρήστη (εφόσον δεν έχει εκτελεστεί).

Γενικότερα, το API θα πρέπει να ακολουθεί κατά το δυνατό τις καλές πρακτικές των REST APIs, όπως αυτές παρουσιάστηκαν στις διαλέξεις. Τα δεδομένα θα πρέπει να κωδικοποιηθούν σε JSON ή XML. Ειδικά για το #1 θα πρέπει να υλοποιήσετε λήψη δεδομένων τόσο σε JSON (για την εφαρμογή) όσο και σε XML (για εξάσκηση).

Η εφαρμογή θα πρέπει να ολοκληρωθεί ως και το τέλος του εξαμήνου. Προτείνεται η χρήση του SLIM micro-framework που παρέχει έτοιμο σκελετό για δημιουργία API με βάση standards. Επίσης υπάρχει και έτοιμος stateless μηχανισμός login βασισμένος σε JWT (JSOM Web Tokens).

Παραδοτέα: zip αρχείο με τους κώδικες και report σε MS Word (όχι PDF) με συμπληρωμένο τον ακόλουθο πίνακα, καθώς και screenshots από οθόνες της εφαρμογής που να φαίνονται τα 9 end-points παραπάνω.

¹ Θα εκτιμηθεί ιδιαίτερα η χρήση κάποιου CSS framework (π.χ. Bootstrap, Foundation, κτλ.) που παρέχει έτοιμες τέτοιες δομές. Δεν θα χρειαστεί να το μάθετε όλο! Μόνο τα σημεία που σας αφορούν. Αν δεν χρησιμοποιήσετε CSS framework, προτιμήστε μια διάταξη με sidebar στο οποίο θα εμφανίζετε τα επιπλέον (φόρμα login, παραγγελίας, ελέγχου παραγγελιών, κτλ)

End-point	Επίπεδο Ολοκλήρωσης			
	καθόλου	κάτι έκανα αλλά δεν λειτουργεί	σχεδόν το έκανα αλλά έχει "θεματάκια"	το έκανα ΟΚ
1. GET: Λήψη δεδομένων πρατηρίων και τιμών επιλεγμένου καυσίμου.				
2. GET: Πλήθος πρατηρίων (αέρας), μέγιστη, ελάχιστη και μέση τιμή ανά lt (με 3 δεκαδικά).				
3. GET: Λήψη τιμοκαταλόγου ενός πρατηρίου.				
4. GET: Login χρήστη (η database έχει ήδη χρήστες).				
5. POST: Αποστολή δεδομένων παραγγελίας από πλευράς χρήστη (ποσότητα).				
6. GET: Λήψη παραγγελιών πρατηρίου από τον πρατηριούχο				
7. PUT: Αλλαγή τιμής σε καύσιμο από πρατηριούχο				
8. PUT: Εκτέλεση παραγγελίας από πρατηριούχο				
9. DELETE: Διαγραφή παραγγελίας από πρατηριούχο ή από απλό χρήστη (εφόσον δεν έχει εκτελεστεί).				