

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: Τεχνολογίες και προγραμματισμός στον Ιστό 2021-22

ΔΙΔΑΣΚΟΝΤΑΣ: Β. Ζαφείρης

ΕΡΓΑΣΙΑ 2

Ημ/νια παράδοσης: 21/01/2022

(Ομαδική 2 ατόμων)

Δίνεται το Web API μιας υπηρεσίας πρόσβασης στα μαθήματα που προσφέρονται μέσω e-Learning από το ΟΠΑ. Το API δίνει δυνατότητα πρόσβασης στις κατηγορίες των e-Learning μαθημάτων, μέσω αιτήματος GET /categories στο διακομιστή <https://elearning-aueb.herokuapp.com>. Το αίτημα επιστρέφει λίστα με τις κατηγορίες σε μορφή JSON, όπου κάθε κατηγορία χαρακτηρίζεται από τον κωδικό της (id) και το όνομά της (title).

Η υπηρεσία επιτρέπει, επίσης, αναζήτηση μαθημάτων μέσω υποβολής στο διακομιστή των ακόλουθων αιτημάτων:

- GET /courses/search?title=keyword : για αναζήτηση μαθημάτων με βάση μια λέξη keyword που εμφανίζεται στον τίτλο τους,
- GET /courses/search?category=id : για αναζήτηση μαθημάτων με βάση των κωδικό id της κατηγορίας στην οποία ανήκουν.

Η υπηρεσία αναζήτησης επιστρέφει λίστα με μαθήματα σε μορφή JSON, όπου κάθε μάθημα χαρακτηρίζεται από τα γνωρίσματα:

- id: κωδικός μαθήματος,
- title: τίτλος μαθήματος
- category: κωδικός κατηγορίας μαθήματος,
- objectives: κείμενο με μαθησιακούς στόχους του μαθήματος,
- description: σύντομη περιγραφή του περιεχομένου του μαθήματος,
- img: ονομασία αρχείου εικόνας που συνοδεύει την περιγραφή του μαθήματος. Το URI για την πρόσβαση στην εικόνα έχει τη μορφή:
 - ο <https://elearning-aueb.herokuapp.com/static/images/{img}> όπου {img} η ονομασία του αρχείου εικόνας.

Ζητείται η υλοποίηση μιας εφαρμογής ιστού η οποία θα παρέχει τις εξής περιπτώσεις χρήσης:

ΠΧ1. Αναζήτηση μαθημάτων

Η εφαρμογή θα υποστηρίζει την αναζήτηση μαθημάτων με βάση μια λέξη κλειδί που εμφανίζεται στον τίτλο τους. Η αναζήτηση θα είναι υλοποιημένη εξ' ολοκλήρου στον Web Browser αξιοποιώντας το Fetch API για κλήση του Web API. Η εμφάνιση των αποτελεσμάτων αναζήτησης θα γίνεται με δυναμική παραγωγή HTML κώδικα και προσαρτησή του κατάλληλο σημείο της σελίδας αναζήτησης. Η παραγωγή HTML κώδικα θα γίνει, επίσης αξιοποίηση της βιβλιοθήκης [Handlebars](#) και κατάλληλων HTML templates.

Η σελίδα αναζήτησης μαθημάτων θα αποτελεί την αρχική σελίδα της εφαρμογής (`index.html`) και θα περιλαμβάνει κατάλληλο πεδίο κειμένου για πληκτρολόγηση του όρου αναζήτησης και σχετικό κουμπί για υποβολή της αναζήτησης. Η υποβολή του αιτήματος αναζήτησης θα γίνει αποκλειστικά με JavaScript. Η μορφοποίηση και δομή της αρχικής σελίδας μπορεί να βασιστεί σε HTML περιεχόμενο και CSS αρχεία που υλοποιήθηκαν στην Εργασία 1.

ΠΧ2. Λίστα μαθημάτων κάθε κατηγορίας

Η σελίδα αναζήτησης (`index.html`) θα περιλαμβάνει μενού με τις κατηγορίες μαθημάτων. Το μενού θα δημιουργείται δυναμικά κατά την φόρτωση της σελίδας, μετά από λήψη των κατηγοριών μαθημάτων με χρήση του Fetch API.

Κάθε στοιχείο του μενού θα περιλαμβάνει σύνδεσμο προς την ίδια HTML σελίδα όπου θα γίνεται παρουσίαση των μαθημάτων της επιλεγμένης κατηγορίας (`courses.html`). Το URI κάθε συνδέσμου του μενού θα συνοδεύεται από παράμετρο ερωτήματος που θα αντιστοιχεί στον κωδικό κατηγορίας. Συγκεκριμένα η μορφή του URI κάθε στοιχείου του μενού θα είναι `courses.html?category={id}`, όπου `{id}` ο κωδικός της αντίστοιχης κατηγορίας.

Κατά το άνοιγμά της, η σελίδα `courses.html` θα λαμβάνει την παράμετρο ερωτήματος από τη γραμμή διευθύνσεων του browser και θα εκτελεί αίτημα με χρήση του Fetch API για λήψη των μαθημάτων της συγκεκριμένης κατηγορίας. Η πρόσβαση στις παραμέτρους ερωτήματος της γραμμής διευθύνσεων του browser περιγράφεται [εδώ](#).

Το περιεχόμενο της `courses.html` θα παράγεται δυναμικά από τα JSON δεδομένα μαθημάτων και θα περιλαμβάνει αναλυτική περιγραφή των μαθημάτων της κατηγορίας. Το περιεχόμενο κάθε μαθήματος θα περιγράφεται με κατάλληλα σημασιολογικά στοιχεία HTML και θα έχει κατάλληλη διάταξη με χρήση κανόνων CSS. Συγκεκριμένα, θα περιλαμβάνει τον τίτλο, την περιγραφή, τους μαθησιακούς στόχους, καθώς και την εικόνα (`img`) του μαθήματος. Η δυναμική παραγωγή HTML περιεχομένου θα βασιστεί σε Handlebars Templates.

Η σελίδα `courses.html` θα περιλαμβάνει σύνδεσμο προς την `index.html`. Η σχεδιάσή της μπορεί να βασιστεί σε HTML περιεχόμενο και CSS αρχεία που υλοποιήθηκαν στην Εργασία 1.

ΠΧ3. Εγγραφή νέου χρήστη

Η σελίδα αναζήτησης μαθημάτων (`index.html`) θα περιλαμβάνει σύνδεσμο προς φόρμα εγγραφής νέου χρήστη (`register.html`). Η φόρμα θα περιλαμβάνει τα προσωπικά στοιχεία του χρήστη (όνομα, επώνυμο, διεύθυνση, τηλέφωνο, μορφοτικό επίπεδο) καθώς και στοιχεία πρόσβασης (`email`, `password` και επιβεβαίωση του `password`). Όλα τα στοιχεία θα είναι

υποχρεωτικά, ενώ το password θα πρέπει να ικανοποιεί τους εξής κανόνες πολυπλοκότητας: (α) θα πρέπει να περιλαμβάνει τουλάχιστον 8 χαρακτήρες, (β) θα πρέπει να περιλαμβάνει τουλάχιστον ένα αριθμό, ένα γράμμα και ένα από τα ειδικά σύμβολα !@#\$%^&*.

Η επικύρωση της φόρμας θα γίνεται μέσω HTML γνωρισμάτων καθώς και με χρήση του Constraint Validation API. Η υποβολή της φόρμας θα γίνεται με χρήση του Fetch API. Σε περίπτωση που υπάρχει χρήστης με το ίδιο email, η εγγραφή του χρήστη θα ακυρώνεται και θα εμφανίζεται στο χρήστη κατάλληλο μήνυμα σφάλματος. Μετά την επιτυχημένη εγγραφή του χρήστη, η φόρμα θα απενεργοποιείται και θα εμφανίζεται κατάλληλο κείμενο με link (ή κουμπί) για επιστροφή του χρήστη στην αρχική σελίδα (index.html).

Κατά την υποβολή της φόρμας εγγραφής νέου χρήστη, θα καλείται κατάλληλη υπηρεσία ιστού, την οποία καλείστε να υλοποιήσετε στην πλατφόρμα Node.js. Η υλοποίηση του διακομιστή ιστού και της υπηρεσίας εγγραφής χρήστη θα γίνει με χρήση του Node.js API και του Express framework. Ο διακομιστής θα μετατρέπει τα δεδομένα της φόρμας σε JavaScript αντικείμενο που θα ανήκει στην κλάση User και θα αντιπροσωπεύει τα στοιχεία ενός χρήστη. Ο νέος χρήστης θα αποθηκεύεται στην τοπική βάση δεδομένων.

ΠΧ4. Ταυτοποίηση και προβολή του προφίλ χρήστη

Η σελίδα αναζήτησης μαθημάτων (index.html) θα περιλαμβάνει σύνδεσμο προς φόρμα ταυτοποίησης και προβολής του προφίλ ενός χρήστη (profile.html). Η σελίδα profile.html θα περιλαμβάνει φόρμα εισαγωγής του username (email) και password ενός χρήστη και κουμπί υποβολής των στοιχείων ταυτοποίησης. Το πάτημα του κουμπιού θα στέλνει αίτημα ταυτοποίησης του χρήστη στον διακομιστή Node.js με χρήση του Fetch API. Ακριβώς μετά την επιτυχή ταυτοποίηση του χρήστη, θα υποβάλλεται δεύτερο αίτημα στο διακομιστή για λήψη του προφίλ του χρήστη (στοιχεία που καταχώρισε κατά την εγγραφή του), χωρίς κάποια ενέργεια του χρήστη. Μετά τη λήψη τους, τα στοιχεία του χρήστη θα εμφανίζονται σε κατάλληλη περιοχή της σελίδας profile.html. Η δυναμική παραγωγή του HTML περιεχομένου θα γίνεται με Handlebars Templates.

Η ταυτοποίηση του χρήστη και η λήψη του προφίλ του θα υλοποιηθούν ως ξεχωριστές υπηρεσίες στον διακομιστή ιστού Node.js, με χρήση του Express framework.

Περιορισμοί/λεπτομέρειες υλοποίησης

- **Αρχιτεκτονική:** η εφαρμογή θα βασίζεται στην αρχιτεκτονική Πελάτη-Διακομιστή. Το τμήμα Πελάτη θα υλοποιηθεί ως μια σειρά από HTML σελίδες που θα ενσωματώνουν κώδικα JavaScript. Το τμήμα διακομιστή θα υλοποιηθεί ως μια Node.js εφαρμογή, η οποία θα καλείται από τον Πελάτη μέσω υπερσυνδέσμων ή HTTP αιτημάτων με χρήση του Fetch API.
- **Βιβλιοθήκες/frameworks:** Επιτρέπεται η χρήση του Fetch API και της βιβλιοθήκης Handlebars για την υλοποίηση του Πελάτη. Η υλοποίηση του διακομιστή μπορεί να βασιστεί στο Node.js API και στο Express framework. Δεν επιτρέπεται η χρήση άλλων εξωτερικών βιβλιοθηκών/frameworks.

- **Βάση δεδομένων:** δεν απαιτείται η αποθήκευση των δεδομένων σε πραγματική βάση δεδομένων. Τα δεδομένα θα διατηρούνται μόνο κατά τη διάρκεια λειτουργίας του εξυπηρετητή και θα αποθηκεύονται σε κατάλληλες λίστες, τις οποίες θα διαχειρίζονται αντικείμενα DAO (Data Access Objects). Τα DAO αντικείμενα θα υποστηρίζουν μεθόδους αποθήκευσης και αναζήτησης χρηστών. Σε περίπτωση που ενσωματωθεί πραγματική βάση δεδομένων και γίνει κατάλληλη υλοποίηση των DAO αντικειμένων θα υπάρξει bonus 10%.
- **Debugging:** Για πιο αποδοτική δοκιμή και εκσφαλμάτωση της εφαρμογής, προτείνεται η χρήση του εργαλείου nodemon. Το nodemon ξεκινά τον διακομιστή της εφαρμογής και φορτώνει τη νέα έκδοση του κώδικά του, μετά από κάθε αλλαγή.
- **Web API:** η λειτουργικότητα του τμήματος διακομιστή της εφαρμογής σας θα παρέχεται με τη μορφή υπηρεσιών ιστού. Η σχεδίαση των υπηρεσιών ιστού θα πρέπει να γίνει με κατάλληλη επιλογή URIs και σωστή χρήση των HTTP methods και HTTP κωδικών απόκρισης.
- **Παρουσίαση:** Η παρουσίαση της εφαρμογής θα βασιστεί σε σημασιολογικά στοιχεία HTML και κατάλληλους κανόνες CSS.

Βαθμολόγηση

Η κατανομή της βαθμολογίας, αναλύεται ως εξής:

- PX1 (20%)
- PX2 (20%)
- PX3 (20%)
- PX4 (20%)
- Χρήση σημασιολογικών στοιχείων HTML και κανόνων CSS (10%)
- Σχεδίαση υπηρεσιών με κατάλληλα URIs και σωστή χρήση των HTTP methods και κωδικών απόκρισης (5%)
- Καλή σχεδίαση κώδικα με χρήση συναρτήσεων και κλάσεων (5%)
- Ενσωμάτωση πραγματικής βάσης δεδομένων (10%)

Καλή δουλειά!