

# ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

## ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΜΑΘΗΜΑ: Τεχνολογίες και προγραμματισμός στον Ιστό 2022-23**

**ΔΙΔΑΣΚΟΝΤΑΣ: Β. Ζαφείρης**

### ΕΡΓΑΣΙΑ 2

**Προθεσμία παράδοσης: 15/01/2023**

**(Ομαδική 2 ατόμων)**

Δίνεται το Web API μιας υπηρεσίας πρόσβασης στα προϊόντα του ηλεκτρονικού καταστήματος WikiShop, το οποίο εμπορεύεται προϊόντα για οικονομική ενίσχυση των δραστηριοτήτων της Wikipedia. Το API δίνει δυνατότητα πρόσβασης στις κατηγορίες και υπο-κατηγορίες προϊόντων του καταστήματος καθώς και στα προϊόντα επιλεγμένης κατηγορίας μέσω κατάλληλων HTTP GET αιτημάτων στο διακομιστή <https://wiki-shop.onrender.com>. Συγκεκριμένα, υποστηρίζονται οι ακόλουθες κλήσεις, των οποίων τα αποτελέσματα έχουν τη μορφή JSON:

- GET /categories: για λήψη λίστας με τις κατηγορίες προϊόντων, όπου κάθε κατηγορία χαρακτηρίζεται από τα γνωρίσματα:
  - id: ακέραιος, μοναδικό αναγνωριστικό της κατηγορίας,
  - title: string, τίτλος της κατηγορίας προϊόντων,
  - img\_url: string, URL μιας εικόνας που χαρακτηρίζει την κατηγορία προϊόντων,
- GET /categories/:id/subcategories: για λήψη λίστας με τις υποκατηγορίες μιας συγκεκριμένης κατηγορίας προϊόντων, η οποία προσδιορίζεται στο τμήμα :id του μονοπατιού κλήσης της υπηρεσίας. Κάθε υποκατηγορία χαρακτηρίζεται από τα γνωρίσματα:
  - id: ακέραιος, μοναδικό αναγνωριστικό της υποκατηγορίας,
  - category\_id: ακέραιος, αναγνωριστικό της κατηγορίας στην οποία εντάσσεται η υποκατηγορία,
  - title: τίτλος της υποκατηγορίας,
- GET /categories/:id/products: για λήψη λίστας με τα προϊόντα μιας συγκεκριμένης κατηγορίας, η οποία προσδιορίζεται στο τμήμα :id του μονοπατιού κλήσης της υπηρεσίας. Κάθε προϊόν χαρακτηρίζεται από τα γνωρίσματα:
  - id: ακέραιος, μοναδικό αναγνωριστικό του προϊόντος,
  - title: string, ονομασία του προϊόντος,
  - subcategory\_id: ακέραιος, αναγνωριστικό της υπο-κατηγορίας στην οποία ανήκει το προϊόν,
  - description: string, σύντομη περιγραφή του προϊόντος,
  - cost: ακέραιος, κόστος του προϊόντος σε \$,
  - image: string, URL μιας εικόνας που χαρακτηρίζει το προϊόν.

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

#### ΠΧ1. Πλοήγηση σε κατηγορίες προϊόντων (20%)

Η εφαρμογή θα υποστηρίζει την πλοήγηση του χρήστη στις κατηγορίες προϊόντων και στα προϊόντα της κάθε κατηγορίας. Η πλοήγηση θα είναι υλοποιημένη εξ' ολοκλήρου στον Web Browser αξιοποιώντας το Fetch API για κλήση του WikiShop API. Η εμφάνιση των κατηγοριών προϊόντων καθώς και των προϊόντων κάθε κατηγορίας θα γίνεται με δυναμική παραγωγή HTML κώδικα και προσαρτησή του σε κατάλληλα σημεία των ιστοσελίδων. Η παραγωγή HTML κώδικα θα γίνει με αξιοποίηση της βιβλιοθήκης [Handlebars](#) και κατάλληλων HTML templates.

Η αρχική σελίδα της εφαρμογής (`index.html`) θα παρουσιάζει τα στοιχεία των κατηγοριών προϊόντων που διαθέτει το κατάστημα, εμφανίζοντας περιγραφή και εικόνα για κάθε κατηγορία με κατάλληλη διάταξη στη σελίδα. Η εικόνα κάθε κατηγορίας θα ενσωματώνει υπερ-σύνδεσμο προς τη σελίδα των προϊόντων της κατηγορίας (`category.html`), όπου θα εμφανίζεται λίστα με τα προϊόντα επιλεγμένης κατηγορίας.

Η λίστα προϊόντων της σελίδας `category.html` θα περιλαμβάνει πληροφορίες για όλα τα προϊόντα της κατηγορίας, εμφανίζοντας για κάθε προϊόν τα στοιχεία: τίτλος, περιγραφή, φωτογραφία και τιμή προϊόντος. Ο κωδικός κατηγορίας θα κωδικοποιείται ως παράμετρος ερωτήματος στα URL της σελίδας `index.html` (πχ. `/category.html?categoryId=1`), έτσι ώστε κατά το άνοιγμα της σελίδας `category.html`, ο κώδικας JavaScript της σελίδας να μπορεί να έχει πρόσβαση στον κωδικό κατηγορίας μέσω των παραμέτρων ερωτήματος της γραμμής διευθύνσεων του Browser (δείτε [εδώ](#) το σχετικό Browser API).

Η μορφοποίηση και δομή των ιστοσελίδων μπορεί να βασιστεί σε HTML περιεχόμενο και CSS αρχεία που υλοποιήθηκαν στην Εργασία 1.

#### ΠΧ2. Φιλτράρισμα προϊόντων βάσει υπο-κατηγορίας (10%)

Εμπλουτίστε την σελίδα `category.html` που παρουσιάζει τα προϊόντα επιλεγμένης κατηγορίας, με λειτουργικότητα για φιλτράρισμα των προϊόντων με βάση την υπο-κατηγορία στην οποία ανήκουν. Συγκεκριμένα, κατά τη φόρτωση της `category.html` θα λαμβάνονται, επίσης, και οι υπο-κατηγορίες της τρέχουσας κατηγορίας με χρήση του WikiShop API. Οι υπο-κατηγορίες θα εμφανίζονται σε κατάλληλο πλευρικό μενού ως λίστα από υπερσυνδέσμους ή radio buttons. Κατά την επιλογή οποιασδήποτε υπο-κατηγορίας, τα προϊόντα θα φιλτράρονται και θα εμφανίζονται μόνο αυτά που ανήκουν στην υπο-κατηγορία (βάσει του γνωρίσματος `subcategory_id` των προϊόντων). Η λίστα επιλογών θα περιλαμβάνει και την επιλογή «Όλα», η οποία θα επιτρέπει εμφάνιση όλων των προϊόντων της κατηγορίας και η οποία θα είναι προεπιλεγμένη κατά τη φόρτωση της σελίδας. Δείτε παρακάτω ένα παράδειγμα με το περιεχόμενο του πλευρικού μενού:

Πατήστε το σύνδεσμο για φιλτράρισμα των προϊόντων ανάλογα με την υπο-κατηγορία τους.

- ☒ Όλα
- ☐ DrinkWear
- ☐ Pens
- ☐ Stickers
- ☐ Water Bottle

**Σημείωση:** Η λήψη των προϊόντων της κατηγορίας θα γίνεται μόνο μια φορά κατά την φόρτωση της σελίδας `category.html` με χρήση του Fetch API. Κατά την επιλογή ενός φίλτρου θα προσαρμόζεται η λίστα προϊόντων προς εμφάνιση και θα παράγεται κατάλληλο HTML περιεχόμενο χωρίς υποβολή επιπλέον HTTP αιτημάτων για λήψη των προϊόντων. Προς διευκόλυνσή σας, συνίσταται η κωδικοποίηση τόσο στα radio inputs όσο και στα HTML στοιχεία που περιγράφουν τα προϊόντα κατάλληλης πληροφορίας όπως (`product-id`, `subcategory-id` κτλ) με χρήση HTML γνωρισμάτων `data-*` (πχ. `data-product-id`). Τα γνωρίσματα αυτά είναι προσβάσιμα μέσω JS από ένα στοιχείο HTML μέσω του γνωρίσματος `dataset` (πχ. `element.dataset.productId`).

### ΠΧ3. Προσθήκη προϊόντων στο καλάθι (30%)

#### Ταυτοποίηση χρήστη

Προϋπόθεση για την προσθήκη προϊόντων στο καλάθι ενός χρήστη είναι η επιτυχής ταυτοποίησή του (`login`) από το σύστημα. Η ταυτοποίηση θα γίνεται μέσω HTML φόρμας (**Φόρμα Ταυτοποίησης**), η οποία θα εμφανίζεται σε κατάλληλη περιοχή της σελίδας `category.html` (πχ. στην πάνω δεξιά πλευρά της σελίδας) και θα λαμβάνει ως είσοδο το `username` και `password` του χρήστη. Η υποβολή της φόρμας θα γίνεται μέσω του Fetch API και θα περιλαμβάνει κλήση κατάλληλης υπηρεσίας ιστού (**Login Service – LS**).

Υλοποιήστε την υπηρεσία LS ώστε να καλείται με χρήση κατάλληλης HTTP μεθόδου, σε κατάλληλα επιλεγμένο URL και να λαμβάνει από τον Πελάτη τα απαραίτητα στοιχεία για την ταυτοποίηση του χρήστη (`username`, `password`). Η υπηρεσία θα επιστρέφει κατάλληλο κωδικό απόκρισης, σε περίπτωση επιτυχούς ταυτοποίησης, καθώς και αντικείμενο JSON με ένα μοναδικό αναγνωστικό σύνδεσης που θα έχει τη μορφή:

```
{ "sessionId": "1b9d6bcd-bbfd-4b2d-9b5d-ab8dfbbd4bed" }
```

Ο μοναδικός κωδικός θα παράγεται με τη βοήθεια της βιβλιοθήκης [uuid](#). Ανάλογα με τον κωδικό απόκρισης θα εμφανίζεται στο χρήστη κατάλληλο μήνυμα στην περιοχή της Φόρμας Ταυτοποίησης.

Για λόγους απλούστευσης της υλοποίησης, σε περίπτωση επαναφόρτωσης της σελίδας `category.html` η κατάσταση ταυτοποίησης του χρήστη θα χάνεται και θα απαιτείται εκ' νέου ταυτοποίησή του προκειμένου να μπορεί να προσθέσει προϊόντα στο καλάθι του.

#### Προσθήκη προϊόντος στο καλάθι

Εμπλουτίστε τη σελίδα `category.html` με δυνατότητα προσθήκης προϊόντων στο καλάθι αγορών του χρήστη. Συγκεκριμένα, κάθε εμφανιζόμενο προϊόν στη σελίδα θα περιλαμβάνει κουμπί (πχ. κουμπί **Αγορά**) για κλήση κατάλληλης υπηρεσίας ιστού (**Cart Item Service - CIS**),

η οποία θα προσθέτει ένα τεμάχιο του προϊόντος στο καλάθι αγορών του τρέχοντος χρήστη. Αν πατηθεί ξανά το κουμπί «**Αγορά**» για προϊόν που υπάρχει ήδη στο καλάθι του χρήστη, η ποσότητα του προϊόντος στο καλάθι θα αυξάνεται κατά ένα.

Υλοποιήστε την υπηρεσία CIS, ώστε να καλείται με χρήση κατάλληλης HTTP μεθόδου, σε κατάλληλα επιλεγμένο URL και να λαμβάνει από τον Πελάτη τα απαραίτητα στοιχεία (κωδικός προϊόντος, username, sessionId) για προσθήκη ενός συγκεκριμένου προϊόντος στο καλάθι αγορών του αντίστοιχου χρήστη. Η υπηρεσία θα ελέγχει αν το συγκεκριμένο username αντιστοιχεί στο αναγνωριστικό σύνδεσης sessionId, προκειμένου να αποφευχθεί η προσθήκη προϊόντων από μη ταυτοποιημένους χρήστες. Η υπηρεσία θα επιστρέφει κατάλληλο κωδικό απόκρισης ανάλογα με την έκβασή της.

Καθώς η επιτυχής ταυτοποίηση του χρήστη είναι προϋπόθεση για προσθήκη προϊόντων στο καλάθι, σε περίπτωση που ο χρήστης δεν είναι ταυτοποιημένος, το πάτημα του κουμπιού **Αγορά** θα εμφανίζει μήνυμα σφάλματος «Παρακαλώ συνδεθείτε για προσθήκη προϊόντων στο καλάθι».

### **Εμφάνιση πλήθους προϊόντων καλαθιού**

Το συνολικό πλήθος προϊόντων που είναι διαθέσιμα στο καλάθι αγορών του τρέχοντος χρήστη θα εμφανίζεται σε κατάλληλη περιοχή της σελίδας `category.html` (περιοχή Καλάθι Αγορών στο εξής) και θα ανανεώνεται με την προσθήκη επιπλέον προϊόντων. Σε περίπτωση επαναφόρτωσης της σελίδας `category.html` το πλήθος προϊόντων στο καλάθι του χρήστη θα λαμβάνεται από το διακομιστή αμέσως μετά την επιτυχή ταυτοποίηση του χρήστη.

Η λήψη του πλήθους προϊόντων θα γίνεται με κλήση κατάλληλης υπηρεσίας ιστού (**Cart Size Service – CSS**). Υλοποιήστε την υπηρεσία CSS, ώστε να καλείται με χρήση κατάλληλης HTTP μεθόδου, σε κατάλληλα επιλεγμένο URL και να λαμβάνει από τον Πελάτη τα απαραίτητα στοιχεία για την λήψη του μεγέθους του καλαθιού αγορών ενός επιλεγμένου χρήστη (username, sessionId). Κατά την επιτυχή εκτέλεσή της θα επιστρέφει αντικείμενο JSON της μορφής { "size": 10 }, όπου η ιδιότητα size θα αντιστοιχεί στο πλήθος προϊόντων στο καλάθι του χρήστη.

### **Παραδοχές**

Κατά την υλοποίηση των υπηρεσιών ιστού CIS, LS και CSS βασιστείτε στις εξής παραδοχές:

- η υλοποίηση των παραπάνω υπηρεσιών CIS, LS και CSS θα γίνει στην πλατφόρμα Node.js με χρήση του express framework,
- ο διακομιστής θα διατηρεί ένα καλάθι αγορών για κάθε χρήστη, το οποίο θα διατηρείται κατά την πλοήγηση του χρήστη στις σελίδες της εφαρμογής,
- τα καλάθια αγορών των χρηστών θα χάνονται κατά την επανεκκίνηση του διακομιστή,
  - τα καλάθια θα διατηρούνται μόνο στην περίπτωση που ο διακομιστής έχει πρόσβαση σε πραγματική βάση δεδομένων (bonus λειτουργικότητα),
- κατά την εκκίνηση της εφαρμογής, θα δημιουργείται αυτόματα μια σειρά από χρήστες με συγκεκριμένα username και passwords, προκειμένου να είναι δυνατή η επίδειξη λειτουργικότητας της εφαρμογής,

#### ΠΧ4. Προβολή καλαθιού προϊόντων (20%)

Προσθέστε στην περιοχή «Καλάθι αγορών» κατάλληλο σύνδεσμο, ο οποίος θα οδηγεί στη σελίδα `cart.html`, η οποία θα προβάλλει το περιεχόμενο του καλαθιού αγορών ενός συγκεκριμένου χρήστη. Ο προσδιορισμός του χρήστη θα γίνεται με κατάλληλες παραμέτρους ερωτήματος στο URL της σελίδας (πχ. `cart.html?username=bzafiris&sessionId=1b9d6bcd-bbfd-4b2d-9b5d-ab8dfbbd4bed`). Κατά τη φόρτωση της `cart.html`, θα λαμβάνονται οι τιμές των παραμέτρων ερωτήματος με χρήση JavaScript κώδικα (αντίστοιχα με την ΠΧ 1) και θα υποβάλλεται αίτημα στο διακομιστή για λήψη του περιεχομένου του καλαθιού προϊόντων του χρήστη.

Τα στοιχεία του καλαθιού προϊόντων θα εμφανίζονται σε μορφή πίνακα, όπου κάθε γραμμή θα αντιστοιχεί σε ένα προϊόν του καλαθιού και θα περιλαμβάνει τα δεδομένα: *όνομα προϊόντος*, *κόστος προϊόντος*, *ποσότητα προϊόντος*. Ακριβώς κάτω από τον πίνακα, θα εμφανίζεται το συνολικό κόστος του καλαθιού προϊόντων. Η παραγωγή του αντίστοιχου HTML περιεχομένου θα γίνει με χρήση Handlebars templates.

Η λήψη του περιεχομένου του καλαθιού προϊόντων θα γίνεται με κλήση υπηρεσίας ιστού (**Cart Retrieval Service – CRS**). Υλοποιήστε την CRS ώστε να καλείται με χρήση κατάλληλης HTTP μεθόδου, σε κατάλληλα επιλεγμένο URL και να λαμβάνει από τον Πελάτη τα απαραίτητα στοιχεία (`username`, `sessionId`) για προβολή του καλαθιού αγορών. Η υπηρεσία θα ελέγχει αν το συγκεκριμένο `username` αντιστοιχεί στο αναγνωριστικό σύνδεσης `sessionId`, προκειμένου να αποφευχθεί η πρόσβαση στο καλάθι αγορών από μη εξουσιοδοτημένους χρήστες. Σε περίπτωση επιτυχούς εκτέλεσης της υπηρεσίας θα επιστρέφεται το καλάθι αγορών σε μορφή JSON, που θα έχει τη μορφή:

```
{
  "cartItems": [
    {"title": "Globe hoodie (men)", "cost": 42, "quantity": 2},
    {"title": "Globe hoodie (women)", "cost": 40, "quantity": 2}
  ],
  "totalCost": 164
}
```

#### 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 θα βασιστεί στο Node.js API και στο [express](#) framework. Δεν επιτρέπεται η χρήση άλλων εξωτερικών βιβλιοθηκών/frameworks.
  - ο η παραγωγή μοναδικών κωδικών κατά την επιτυχή ταυτοποίηση του χρήστη θα γίνεται με χρήση του NPM module [uuid](#).
- **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 καθώς και καλαθιών αγορών. 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.
- **Οργάνωση κώδικα:** Μπορείτε να βασιστείτε σε ένα ήδη υπάρχον project για γρήγορη εκκίνηση της ανάπτυξης της εφαρμογής σας (<https://github.com/bzafiris/infl65-webdev-project>).

### Βαθμολόγηση

Η κατανομή της βαθμολογίας, αναλύεται ως εξής:

- PX1 (20%)
- PX2 (10%)
- PX3 (30%)
- 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 (bonus 10%)

**Καλή δουλειά!**