

Εργασία

Πληροφοριακά Συστήματα Παγκόσμιου Ιστού

Α.Π.Θ. Τμήμα Πληροφορικής

Εαρινό Εξάμηνο 2023-2024

Διδάσκοντες: Καθ. Αθηνά Βακάλη, Δρ. Παύλος Σερμπέζης

Υπεύθυνοι εργασίας: Απόστολος Τσιάκαλος (atsiakalos@csd.auth.gr), Παύλος Σερμπέζης (sermpezis@csd.auth.gr)

Σύνοψη εργασίας

Η εργασία αποτελείται από 4 ενότητες όπως περιγράφονται στον παρακάτω πίνακα, με ξεχωριστή αξιολόγηση σε κάθε ενότητα.

Ενότητα	Τίτλος	Μονάδες αξιολόγησης
A	Ανάπτυξη στατικών ιστοσελίδων	1
B	Ανάπτυξη REST API	1
Γ	Ανάπτυξη δυναμικής ιστοσελίδας	1
Συνολικές μονάδες		3

Προθεσμία παράδοσης της εργασίας είναι **03/06/2024**.

Η εκπρόθεσμη παράδοση θα έχει ως αποτέλεσμα την αφαίρεση 10% της συνολικής βαθμολογίας για κάθε μέρα που περνάει, πχ . -10% μέχρι και 04/06/2024, -20% μέχρι και 05/06/2024, κλπ.

Αξιολόγηση: Κάθε ενότητα περιλαμβάνει τις μονάδες βαθμολόγησης της με ανάλυση βαθμολογίας για κάθε επιμέρους κομμάτι. Η αδυναμία εκτέλεσης κάποιου κομματιού βαθμολογείται ως Λάθος. Καθώς η αξιολόγηση θα γίνει με αυτοματοποιημένο τρόπο, δεν επιτρέπεται η χρήση κάποια άλλη βιβλιοθήκης εκτός από αυτές που αναφέρονται ρητώς στις εκφωνήσεις των ενότητων. Σε περίπτωση που κάποιο κομμάτι χρησιμοποιεί κάποια βιβλιοθήκη πέρα από αυτές που ορίζονται στην εκφώνηση θα έχει ως συνέπεια την αδυναμία εκτέλεσης του και άρα την βαθμολόγηση ως Λάθος.

Αρχεία εργασίας: Στο repository <https://github.com/Datalab-AUTH/pspi> δίνονται 11 αρχεία. Θα χρησιμοποιήσετε αποκλειστικά αυτά τα αρχεία για να υλοποιήσετε την εργασία σας. Επεξεργαστείτε και προσθέστε κώδικα μόνο εντός των σημείων που ξεκινούν με **BEGIN CODE**

HERE και τελειώνουν με **END CODE HERE**. Υποβάλετε στο [elearning](#) ένα **zip** αρχείο με όνομα **AEM1-AEM2-AEM3-AEM4.zip** που θα περιλαμβάνει την υλοποίηση σας.

Ο κώδικας θα ελεγχθεί ως προς την πρωτοτυπία του, και κάθε εντοπισμός λογοκλοπής -αντιγραφής κώδικα συνεπάγεται τον μηδενισμό της βαθμολογίας.

Επικοινωνία: Όλη η επικοινωνία σχετικά με απορίες για την εργασία θα γίνεται μέσω του φόρουμ στην σελίδα του μαθήματος στο elearning. Για οτιδήποτε άλλο σχετικά με την εργασία μπορείτε να στείλετε email με το πρόθεμα WIS_23 στους υπεύθυνους της εργασίας (συμπεριλάβετε και τους δύο υπευθυνους της εργασίας σε όλα τα emails σας).

Ενότητα Α - Ανάπτυξη στατικών ιστοσελίδων

Στην ενότητα Α καλείστε να αναπτύξετε δύο στατικές σελίδες, την homepage.html και την products.html, που είναι ορατές σε έναν χρήστη, από τη δομή, το περιεχόμενο, μέχρι τη μορφοποίηση και τη βασική αλληλεπίδραση με μενού, κουμπιά και τα υπόλοιπα διαδραστικά στοιχεία. Οι τεχνολογίες που απαιτούνται για τη συγκεκριμένη υλοποίηση είναι η HTML5 και CSS. Οι σελίδες αυτές περιγράφονται αναλυτικά στο παράρτημα Α. Για τη δημιουργία τους μπορείτε να χρησιμοποιήσετε έτοιμα συστατικά **μόνο** από τη βιβλιοθήκη [Bootstrap](#).

Κάθε σελίδα που θα παραδώσετε πρέπει να έχει την δομή σύμφωνα με τις εικόνες που δίνονται στο παράρτημα Α. Κάθε σελίδα περιλαμβάνει σε κόκκινο πλαίσιο (βλ. παραρτημα Α) τα στοιχεία που θα αναπτύξετε, δηλ. Header 1, Frame 1, κτλ. Κάθε κόκκινο πλαίσιο βαθμολογείται ξεχωριστά σύμφωνα με τις μονάδες που αναγράφονται στην επιμέρους βαθμολογία.

Ανάλυση βαθμολογία			
homepage.html		products.html	
Κόκκινο πλαίσιο	Βαθμολογία	Κόκκινο πλαίσιο	Βαθμολογία
Header 1 & Footer 1	0.1	Header 2 & Footer 2	0.1
Frame 1	0.1	Frame 4	0.2
Frame 2	0.2	Frame 5	0.2
Frame 3	0.1		

Ενότητα B - Ανάπτυξη REST API

Στην ενότητα B καλείστε να αναπτύξετε ένα Flask REST API, που περιλαμβάνει όλη την αλληλεπίδραση ενός server με μία βάση δεδομένων για την αναζήτηση και παρουσίαση αποθηκευμένων δεδομένων, εισαγωγή νέων δεδομένων αλλά και τη δυναμική δημιουργία περιεχομένου, χρησιμοποιώντας Python 3.10. Ως βάση δεδομένων θα χρησιμοποιήσετε τη MongoDB (ανήκει στην κατηγορία NoSQL βάσεων δεδομένων). Η δημιουργία του API θα γίνει σε ένα αρχείο με όνομα `app.py`, χρησιμοποιώντας **μόνο και αποκλειστικά** τις βιβλιοθήκες: [Flask](#), [Flask-PyMongo](#), [Flask-Cors](#), [numpy](#), [selenium](#)

Τα **endpoints** που θα αναπτύξετε είναι:

- **/search**: Αναπτύξτε ένα GET request που θα δέχεται μία παράμετρο με όνομα name με τύπο string και θα αναζητάτε ένα προϊόν στη βάση με βάση το όνομα και θα σας επιστρέφει το προϊόν ή τα προϊόντα. Σε περίπτωση που βρει περισσότερα από ένα προϊόν τότε τα προϊόντα που θα επιστρέφει θα πρέπει να είναι κατά φθίνουσα σειρά ταξινομημένα με βάση τη τιμή. Το request θα επιστρέφει μία λίστα με όλα τα προϊόντα που βρήκε. Η λίστα θα περιέχει JSON objects με schema `{"id": string, "name": string, "production_year": number, "price": number, "color": number, "size": number}`. Αν δεν βρει κανένα προϊόν τότε το endpoint θα επιστρέφει κενή λίστα. Για παράδειγμα αν αναζητήσετε ένα προϊόν με όνομα Paper, τότε η αναζήτηση θα πρέπει να επιστρέφει όλα τα προϊόντα που έχουν τη λέξη Paper στο όνομα, π.χ. Paper A4, Paper A3. Ωστόσο αν αναζητήσετε ένα προϊόν με όνομα Paper A3, τότε η αναζήτηση πρέπει να επιστρέφει μόνο αυτό το προϊόν.

Προσοχή: Στο collection της βάσης που θα έχετε τα προϊόντα πρέπει να φτιάξετε έναν Index στο πεδίο name με τύπο text (θα εξηγηθεί στο εργαστήριο για τις NoSQL βάσεις δεδομένων).

- **/add-product**: Αναπτύξτε ένα POST request που θα έχει body ένα JSON object με schema `{"id": string, "name": string, "production_year": number, "price": number, "color": number, "size": number}` και θα προσθέτει ένα προϊόν στη βάση. Σε περίπτωση που υπάρχει προϊόν με το ίδιο όνομα στη βάση τότε θα πρέπει να ενημερώνει την τιμή, τη χρονιά παραγωγής, το χρώμα και το μέγεθος του.

Προσοχή:

- Υπάρχουν 3 χρώματα, κόκκινο με κωδικό 1, κίτρινο με κωδικό 2 και μπλε με κωδικό 3. Στο JSON που θα στέλνετε θα πρέπει στο πεδίο color να περνάτε ένα από τρεις κωδικούς χρωμάτων.
- Υπάρχουν 4 μεγέθη, small με κωδικό 1, medium με κωδικό 2, large με κωδικό 3 και extra large με κωδικό 4. Στο JSON που θα στέλνετε θα πρέπει στο πεδίο size να περνάτε ένα από τέσσερις κωδικούς μεγέθων.
- **/content-based-filtering**: Αναπτύξτε ένα POST request που θα έχει body ένα JSON object με schema `{"id": string, "name": string, "production_year": number, "price": number, "color": number, "size": number}` και θα υλοποιεί τον αλγόριθμο *Content Based Filtering* (ο αλγόριθμος υπάρχει στις διαφάνειες της θεωρίας). Ως συνάρτηση ομοιότητας θα χρησιμοποιήσετε cosine similarity (ενδεικτικά μπορείτε να χρησιμοποιήσετε μεθόδους της βιβλιοθήκης numpy). Το request θα επιστρέφει μία λίστα με τα ονόματα των

προϊόντων που έχουν ομοιότητα πάνω από 70%. Αν δεν υπάρχει κανένα προϊόν με ομοιότητα πάνω από 70% τότε θα επιστρέφει κενή λίστα.

- **/crawler**: Αναπτύξτε ένα GET request που θα δέχεται μία παράμετρο με όνομα semester με τύπο int και θα συλλέγει όλα τα μαθήματα του εξαμήνου που έχει δοθεί στην αντίστοιχη παράμετρο από το παρακάτω URL:

<https://qa.auth.gr/el/x/studyguide/600000438/current>

Πρακτικά, θα φτιάξετε έναν crawler (υπάρχει αναφορά στις διαφάνειες της θεωρίας), χρησιμοποιώντας τη βιβλιοθήκη selenium, που θα επιστρέφει μία λίστα με τα ονόματα των μαθημάτων. Για παράδειγμα, περάσουμε ως παράμετρο το 2 τότε το endpoint θα πρέπει να επιστρέψει την παρακάτω λίστα:

["ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ ΙΙ", "ΠΙΘΑΝΟΤΗΤΕΣ & ΣΤΑΤΙΣΤΙΚΗ", "ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ", "ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ", "ΘΕΩΡΙΑ ΥΠΟΛΟΓΙΣΜΟΥ"]

Η πρόσβαση μεταξύ Flask και MongoDB θα γίνει **αποκλειστικά** με την βιβλιοθήκη Flask-PyMongo. Για να αποφύγετε προβλήματα σχετικά με την πολιτική CORS, θα χρειαστεί να ενεργοποιήσετε την CORS policy στο Flask API με την βιβλιοθήκη Flask-Cors. Το Flask API πρέπει να τρέχει στην IP 127.0.0.1 (localhost) και στη PORT 5000, δηλ. <http://127.0.0.1:5000>. Κάθε endpoint βαθμολογείται ξεχωριστά σύμφωνα με τις μονάδες που αναγράφονται στην επιμέρους βαθμολογία.

Ανάλυση βαθμολογίας	
Endpoint	Βαθμολογία
/search	0.25
/add-product	0.25
/content-based-filtering	0.25
/crawler	0.25

Σημείωση: Από το αρχείο requirements.txt μπορείτε εύκολα να κάνετε εγκατάσταση τις βιβλιοθήκες που απαιτούνται για το REST API.

Ενότητα Γ - Ανάπτυξη δυναμικής ιστοσελίδας

Στην ενότητα Γ καλείστε να συνδέσετε την υλοποίησή σας στην ενότητα Α και Β χρησιμοποιώντας JavaScript, ώστε να αναπτύξετε μία δυναμική ιστοσελίδα. Η JavaScript επιτρέπει τη δημιουργία αλληλεπιδράσεων μεταξύ μιας στατικής σελίδας HTML και ενός REST API. Πιο συγκεκριμένα θα χρειαστεί να υλοποιήσετε τις παρακάτω αλληλεπιδράσεις σε ένα αρχείο JavaScript με όνομα products.js:

- Αλληλεπίδραση 1: Όταν ο χρήστης αναζητήσει ένα προϊόν (Frame 4 - εικόνα 2), η ιστοσελίδα πρέπει να κάνει ένα GET request στο endpoint /search που αναπτύξατε στην

ενότητα Β χρησιμοποιώντας το κείμενο που υπάρχει στο πλαίσιο αναζήτησης. Όταν ολοκληρωθεί το request τα αποτελέσματα της αναζήτησης πρέπει να εμφανίζονται στο πίνακα που υπάρχει στην products.html σελίδα (Frame 4 - εικόνα 2). Σε κάθε νέα αναζήτηση τα αποτελέσματα της προηγούμενης πρέπει να σβήνονται από τον πίνακα.

- Αλληλεπίδραση 2: Όταν ο χρήστης προσθέσει ένα προϊόν (Frame 5 - εικόνα 2), η ιστοσελίδα πρέπει να κάνει ένα POST request στο endpoint /add-product που αναπτύξατε στην ενότητα Β χρησιμοποιώντας τα δεδομένα της φόρμας. Όταν ολοκληρωθεί το request τότε τα δεδομένα της φόρμας πρέπει να διαγράφονται και να εμφανίζεται ένα JavaScript alert με κείμενο "OK".

Προσοχή: Για το POST request θα πρέπει να ορίσετε το "Content-Type" header σε "application/json".

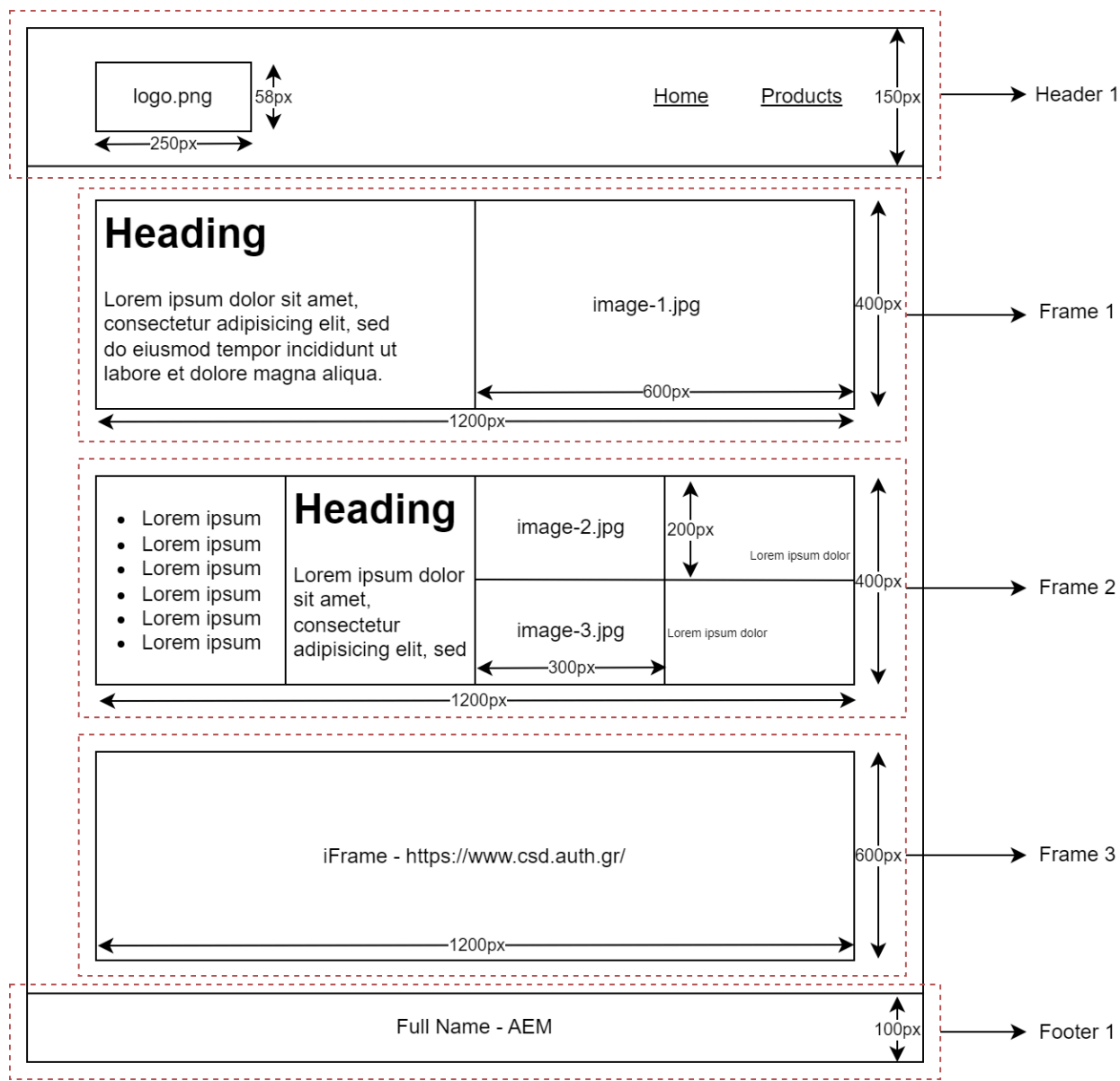
Κάθε αλληλεπίδραση βαθμολογείται ξεχωριστά σύμφωνα με τις μονάδες που αναγράφονται στην επιμέρους βαθμολογία.

Ανάλυση βαθμολογίας	
Αλληλεπίδραση	Βαθμολογία
Αλληλεπίδραση 1	0.5
Αλληλεπίδραση 2	0.5

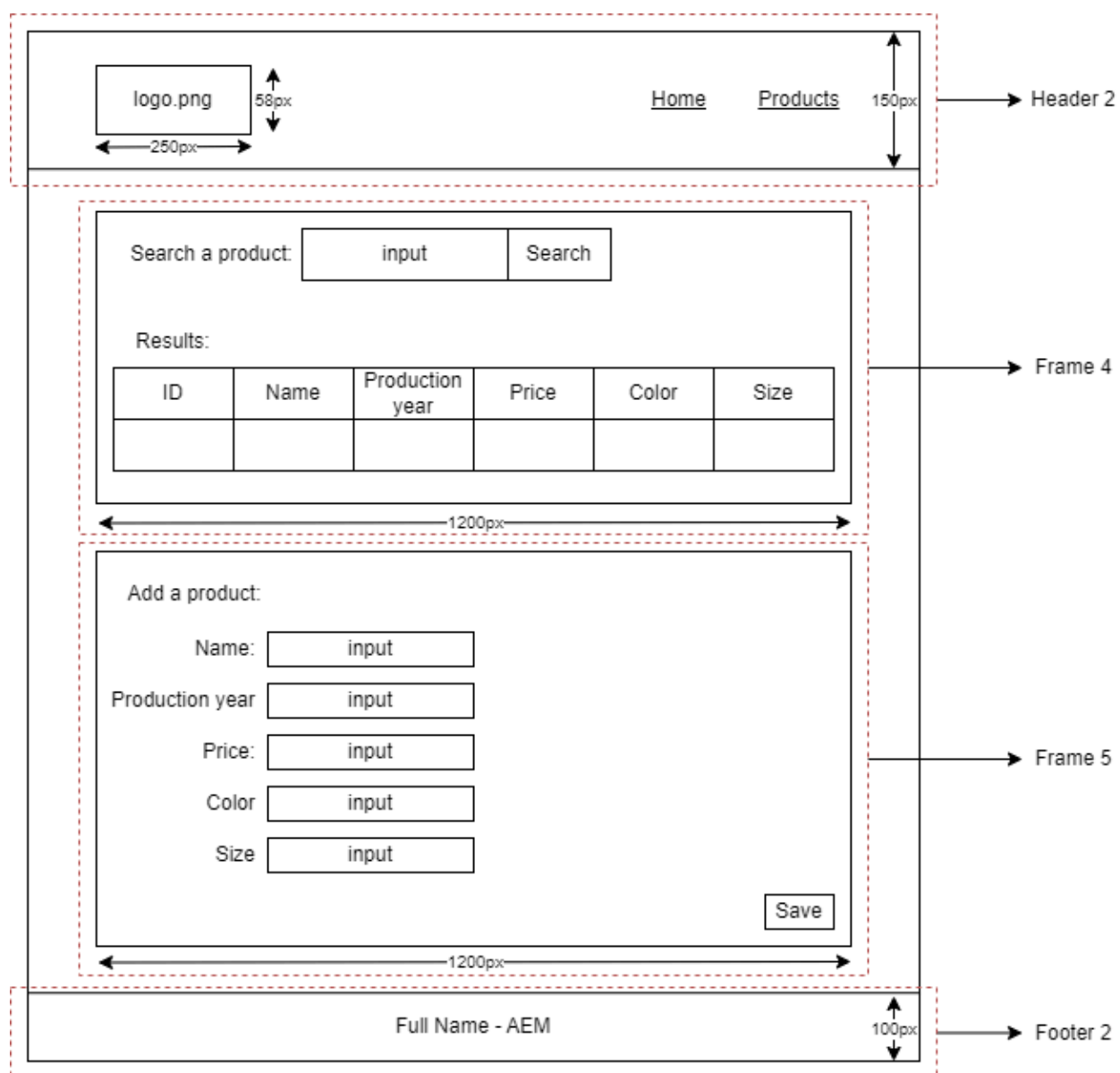
Παράρτημα Α

Η 1η σελίδα που θα αναπτύξετε είναι η homepage.html (εικόνα 1) και η 2η είναι η products.html (εικόνα 2). Ο χρήστης θα πρέπει να μπορεί να εναλλάσσεται μεταξύ των δύο σελίδων χρησιμοποιώντας το μενού. Αν κάνει κλικ στον υπερσύνδεσμο Home τότε θα πηγαίνει στη homepage.html και αν κάνει κλικ στον υπερσύνδεσμο Products τότε θα πηγαίνει στη products.html. Στη σελίδα products.html στο Frame 4 (εικόνα 2) ο χρήστης πρέπει να αναζητά ένα προϊόν στο πλαίσιο αναζήτησης και να μπορεί να βλέπει τα αποτελέσματα στον πίνακα που περιλαμβάνει. Επίσης στο Frame 5 (εικόνα 2) ο χρήστης θα πρέπει να μπορεί να εισάγει ένα καινούργιο προϊόν.

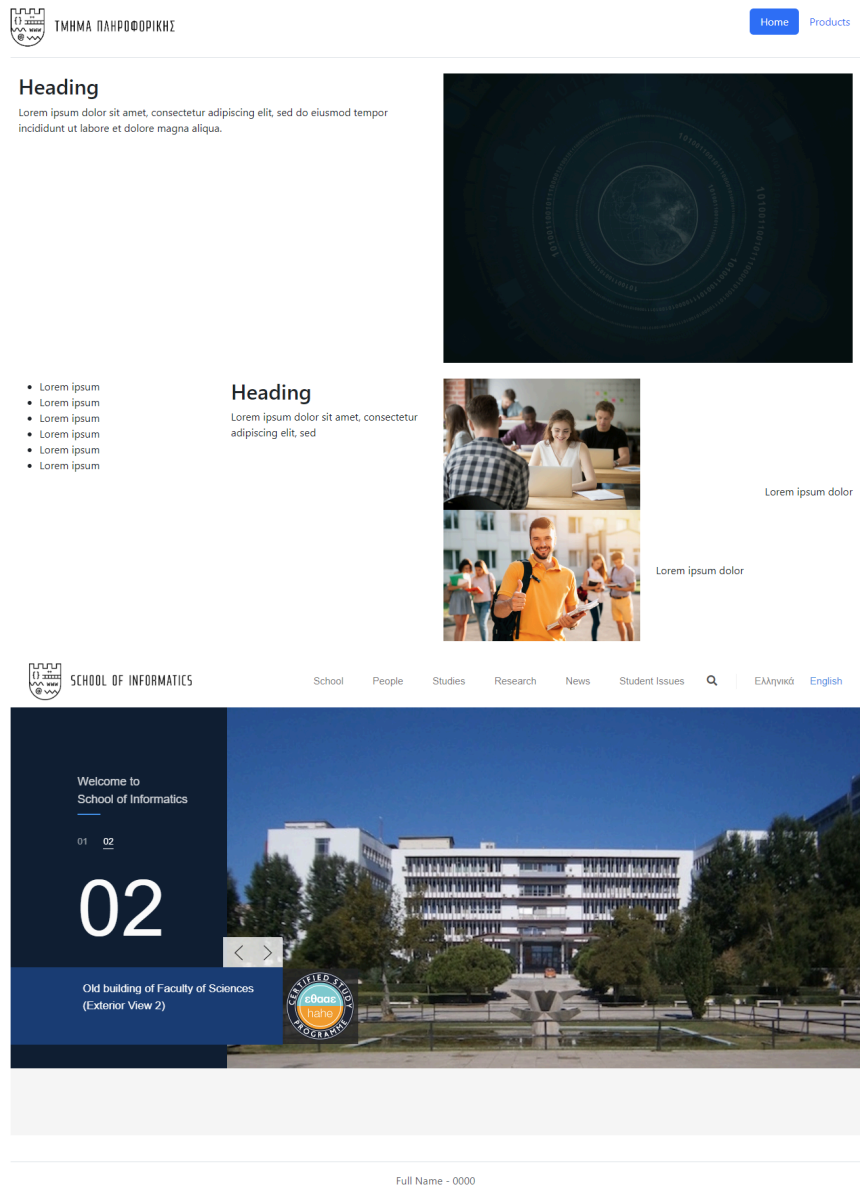
Όλες οι εικόνες προέρχονται από την ιστοσελίδα του τμήματος Πληροφορικής του ΑΠΘ.




Εικόνα 1. homepage.html



Εικόνα 2. products.html



Εικόνα 3. Παράδειγμα homepage.html



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

[Home](#)[Products](#)

Search a product

Results

ID	Name	Production year	Price	Color	Size
----	------	-----------------	-------	-------	------

Add a product

Name

Production year

Price

Color

Size

Full Name - 0000

Εικόνα 4. Παράδειγμα products.html