



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
ΗΥ359 – Διαδικτυακός Προγραμματισμός (Χειμερινό εξάμηνο 2022-2023)
Διδάσκων: **Μ. Μουνταντωνάκης**
Θέμα: **Ομαδική Εργασία (project) μαθήματος 2 ατόμων**
Βοηθοί: Όλοι

E-Libraries

Διάρκεια: 09/12/2022 – 24/01/2023

Project 2 ατόμων

Αξία: 25% του τελικού σας βαθμού
(Συνολο: 108 μονάδες)

Εκπαιδευτικοί Στόχοι

- Εξοικείωση με τη διαδικασία σχεδίασης και ανάπτυξης ενός ολοκληρωμένου διαδικτυακού πληροφοριακού συστήματος
- Εξοικείωση με τεχνολογίες πελάτη και διακομιστή (HTML, Javascript, AJAX, JAVA Servlets, CSS, REST APIS, JSP)
- Εξοικείωση με χρήσιμα APIs για οπτικοποίηση
 - ο Χάρτες
 - ο Γραφικές Παραστάσεις
- Τρόποι υποστήριξης διαφορετικών τύπων χρηστών
- Τεκμηρίωση, Έλεγχος

Σύντομη Περιγραφή Εργασίας

Στόχος. Καλείστε να σχεδιάσετε και να αναπτύξετε ένα διαδικτυακό πληροφοριακό σύστημα, ας το ονομάσουμε **E-Libraries**. Στόχος του πληροφοριακού συστήματος είναι να επιτρέψει στους φοιτητές να μπορούν να δανείζονται βιβλία από βιβλιοθήκες που βρίσκονται στην Κρήτη.

Πολύ Σημαντικό: Τα δεδομένα που θα χρησιμοποιηθούν θα πρέπει να είναι φανταστικά. Απαγορεύεται όμως να χρησιμοποιηθούν δεδομένα που περιέχουν υβριστικό, πολιτικό σεξουαλικό και ρατσιστικό περιεχόμενο. Σε τέτοιες περιπτώσεις θα υπάρχουν και οι ανάλογες ποινές (ακόμα και μηδενισμός).

Απαγορεύεται η χρήση κώδικα που δεν έχετε γράψει οι ίδιοι, είτε από το διαδίκτυο είτε από κάποιο συμφοιτητή σας. Σε περίπτωση εντοπισμού αντιγραφής η εργασία θα μηδενίζεται. Αν στην εργασία σας χρησιμοποιήσετε κάτι έτοιμο, πρέπει να το αναφέρεται ρητώς και να εξηγήσετε και το λόγο.

Αναλυτική Περιγραφή της Εργασίας

1. Σενάρια που θα πρέπει να υλοποιηθούν

Σενάριο 1: Ένας χρήστης πρέπει να μπορεί να δει τις πιο κοντινες βιβλιοθήκες που έχουν το βιβλίο που θέλει με βάση την απόσταση από το σπίτι του (ιδανικά και με τη χρήση χάρτη), και να μπορεί να το δανειστεί.

Σενάριο 2: Να μπορεί ένας φοιτητής να δει όλα τα βιβλία ενός συγκεκριμένου είδους.

Σενάριο 3: Ένας χρήστης να μπορεί να δει ποια βιβλία έχει δανειστεί, και τότε τελειώνει η περίοδος δανεισμού.

Σενάριο 4: Ένας υπεύθυνος βιβλιοθήκης να μπορεί να βάζει πληροφορίες για νέα βιβλία, να βάζει τη διαθεσιμότητα της βιβλιοθήκης για ένα βιβλίο, και να διαχειρίζεται τις κρατήσεις.

Σενάριο 5: Ένας φοιτητής να μπορεί να κάνει κριτική για ένα βιβλίο

2. Τύποι Χρηστών

Οι τύποι χρηστών της εφαρμογής είναι:

- Κεντρικός administrator
- Υπεύθυνοι Βιβλιοθήκης
- Εγγεγραμμένοι χρήστες (φοιτητές)
- Επισκέπτες

A. Ο κεντρικός administrator έχει την πλήρη εποπτεία του συστήματος. Έχει τη δυνατότητα διαγραφής χρηστών και μπορεί να δει διάφορα στατιστικά.

B. Οι Υπεύθυνοι Βιβλιοθήκης κάνουν εγγραφή στο σύστημα και διαχειρίζονται την αντίστοιχη βιβλιοθήκη.

Γ. Οι εγγεγραμμένοι χρήστες (φοιτητές) θα μπορούν να χρησιμοποιήσουν την εφαρμογή και να εισάγουν στοιχεία αρκεί να έχουν εγγραφεί σε αυτήν (όπως η άσκηση 1,2,3).

Δ. Οι απλοί χρήστες θα μπορούν να δουν πληροφορίες σχετικά με το σύστημα, να δουν πληροφορίες για βιβλία (δεν θα μπορούν όμως να κάνουν δανεισμό, κριτική κλπ).

3. Βασικές Λειτουργίες Συστήματος

Παρακάτω συνοψίζονται οι βασικές λειτουργίες του συστήματος. Αυτές οι λειτουργίες θα αναλυθούν με περισσότερες λεπτομέρειες στη συνέχεια.

Λειτουργίες Administrator

- Ξεχωριστό Login
- Προβολή Χρηστών με δυνατότητα διαγραφής
- Προβολή στατιστικών

Λειτουργίες Υπεύθυνου βιβλιοθήκης

- Καταχώρηση πληροφοριών για ένα βιβλίο
- Διαθεσιμότητα της συγκεκριμένης βιβλιοθήκης
- Ενημέρωση/Προβολή κατάστασης δανεισμού βιβλίου (και τέλος δανεισμού)
- Ενημέρωση στοιχείων βιβλιοθήκης.

Λειτουργίες Εγγεγραμμένου Χρήστη

- Ενημέρωση στοιχείων
- Εύρεση και προβολή βιβλίων με βάση διάφορα κριτήρια
- Εύρεση πιο κοντινής βιβλιοθήκης για ένα βιβλίο με βάση την τοποθεσία
- Επιλογή, Προβολή Δανεισμού
- Κριτική βιβλίου (μετά το πέρας του δανεισμού, αν θέλει).

Λειτουργίες Απλού Χρήστη (Επισκέπτη)

- Εύρεση και προβολή βιβλίων (χωρίς τη δυνατότητα δανεισμού)
- Αρχική σελίδα συστήματος

Σε καμία περίπτωση δεν πρέπει ένας εγγεγραμμένος χρήστης να βλέπει τα στοιχεία των άλλων χρηστών (εκτός φυσικά από τον admin).

Παρακάτω παρουσιάζονται οι βασικές λειτουργίες του συστήματος

3.1 Λειτουργίες Administrator (10 %)

Για τη λειτουργία του administrator, μπορείτε να θεωρήσετε ότι είναι ο χρήστης με username admin και password admin12* (δεν υπάρχει στη βάση). Θα πρέπει να του δίνεται η δυνατότητα να συνδεθεί μέσω μιας άλλης σελίδας, που δεν θα είναι γνωστή στο χρήστη, και φυσικά με τα σωστά στοιχεία.

Λειτουργία	Περιγραφή	Βαθμολογία
Σύνδεση με τα σωστά στοιχεία	Θα πρέπει να συνδέεται σε ξεχωριστή σελίδα. Ας πούμε ότι έχει username: admin password: admin12*	1%
Προβολή Χρηστών με δυνατότητα διαγραφής	Θα πρέπει να βλέπει κάθε χρήστη ή υπεύθυνο βιβλιοθήκης και να μπορεί να τον διαγράψει. Δεν είναι απαραίτητο να βλέπει όλα τα στοιχεία του, μπορείτε να βάλετε ας πούμε να βλέπει username όνομα και επίθετο	3%
Προβολή στατιστικών	Να μπορεί να βλέπει τα εξής στατιστικά ιδανικά με χρήση google charts (πχ pie ή bar charts):	6%

	<ul style="list-style-type: none"> • αριθμός βιβλίων ανα βιβλιοθήκη • αριθμός βιβλίων ανα είδος • αριθμός προπτυχιακών/μεταπτυχιακών/διδασκονικών φοιτητών 	
--	---	--

3.2 Λειτουργίες Υπεύθυνου βιβλιοθήκης (25%)

Θα πρέπει να του δίνεται η δυνατότητα να συνδεθεί μέσω μιας άλλης σελίδας, από ότι ο χρήστης. Ουσιαστικά θα χειρίζεται μία βιβλιοθήκη.

Λειτουργία	Περιγραφή	Βαθμολογία
Εγγραφή στο σύστημα/ Αλλαγή στοιχείων	Θα πρέπει να μπορεί να γραφτεί στο σύστημα και έπειτα να αλλάζει και τα στοιχεία του αν επιθυμεί (όχι username).	5%
Καταχώρηση πληροφοριών βιβλίου	Θα μπορεί να προσθέτει πληροφορίες για ένα βιβλίο αν δεν υπάρχει ήδη στη βάση (δείτε το REST API της 3ης Σειράς Ασκήσεων). Σε αυτήν την φάση, δεν βάζει ακόμα ότι είναι διαθέσιμο από τη βιβλιοθήκη (δείτε το επόμενο ερώτημα)	5%
Διαθεσιμότητα βιβλίου στη βιβλιοθήκη	Μπορεί να βάζει ότι ένα βιβλίο είναι διαθέσιμο από την βιβλιοθήκη (από αυτά που υπάρχουν στη βάση). Πχ μπορεί μέσα από μία φόρμα να βάζει το ISBN και να πατάει ότι υπάρχει στη βιβλιοθήκη. <u>Θεωρείστε για λόγους απλότητας ότι κάθε βιβλίο υπάρχει σε ένα μόνο αντίτυπο σε κάθε βιβλιοθήκη.</u>	5%
Ενημέρωση κατάστασης δανεισμού βιβλίου	Βλέπει τα request για το δανεισμό βιβλίου και αλλάζει το status. Δείτε και αναλυτικά την ενότητα 3.5.	5%
Προβολή ενεργών δανεισμών	Να βλέπει όλα τα βιβλία που έχουν δανειστεί φοιτητές από τη συγκεκριμένη βιβλιοθήκη (είναι σε κατάσταση borrowed ή returned), και	5%

	πληροφορίες για το χρήστη και να μπορεί ιδανικά να κατεβάζει τη λίστα σε PDF.	
--	---	--

3.3 Λειτουργίες Φοιτητή (35%)

Ο κάθε εγγεγραμμένος φοιτητής θα μπορεί να γραφτεί, σε μια φόρμα που θα περιέχει το όνομα του, ηλικία, φύλο, διεύθυνση, κλπ (ασκήση 1,2, 3).

Επίσης θα πρέπει να μπορεί να βρει βιβλία, διαθέσιμες βιβλιοθήκες για ένα βιβλίο, πχ ταξινομημένες με βάση την πιο κοντινή τοποθεσία. Η ιδανική λύση είναι να μπορεί βλέπει και στο χάρτη που βρίσκεται ένα βιβλίο που μπορεί να δανειστεί. Θα πρέπει να μπορεί να ζητάει ένα δανεισμό, να επιστρέφει το βιβλίο και όταν έχει δανειστεί ένα βιβλίο, να μπορεί να γράφει κριτική. Το σύστημα θα πρέπει να υπενθυμίσει στον χρήστη το τέλος του δανεισμού, για παράδειγμα 3 μέρες πριν.

Λειτουργία	Περιγραφή	Βαθμολογία
Εγγραφή στο σύστημα/ Επεξεργασία στοιχείων	Θα πρέπει να μπορεί να γραφτεί στο σύστημα και να επεξεργάζεται τα στοιχεία του	5%
Εύρεση Βιβλίων- Επέκταση API της 3ης σειράς ασκήσεων	<p>Θα μπορεί να μπορεί να βρίσκει βιβλία εκτός από το είδος και τα fromYear, toYear, και με βάση τα εξής:</p> <ul style="list-style-type: none"> • α) τον τίτλο (πχ με τη χρήση της contains στην SQL) • β) το συγγραφέα (πχ με τη χρήση της contains στην SQL) • γ) τον αριθμό των σελίδων (πχ fromPageNumber, toPageNumber) <p>Μπορεί να γίνει είτε με REST είτε με Servlets.</p> <p>Για κάθε βιβλίο θα βλέπει όλες τις πληροφορίες, αλλά και πιθανές κριτικές που έχουν γίνει και το score τους.</p>	8%

Εύρεση πιο κοντινής βιβλιοθήκης για ένα βιβλίο με βάση την τοποθεσία	<p>Όταν θα επιλέξει ένα συγκεκριμένο βιβλίο να του δείχνει σε ποιες βιβλιοθήκες είναι διαθέσιμο, και πληροφορίες για κάθε βιβλιοθήκη. Θα πρέπει να είναι ταξινομημένοι, α) με βάση την απόσταση από τη διεύθυνση του σπιτιού του και β) το χρόνο για να φτάσουν εκεί με αυτοκίνητο.</p> <ul style="list-style-type: none"> Υπολογισμός απόστασης μεταξύ συντεταγμένων και απόσταση με αμάξι <ul style="list-style-type: none"> https://rapidapi.com/trueway/api/trueway-matrix/ <ul style="list-style-type: none"> Βάζετε σαν origin τα lat/lon Δείτε παράδειγμα στα βοηθητικά Μπορείτε να χρησιμοποιήσετε και άλλο κώδικα, πχ σε java (δείτε τα βοηθητικά) 	10%
Δανεισμός βιβλίου	Θα μπορεί να δανείζεται ένα βιβλίο αν είναι διαθέσιμο. Θεωρούμε ότι κάθε βιβλίο δανείζεται για 1 μήνα. Όταν το έχει δανειστεί, θα μπορεί να το κάνει return. Δείτε την ενότητα 3.5.	5%
Ειδοποίηση για τέλος δανεισμού	3 μέρες πριν το τέλος του δανεισμού, να του έρχεται ειδοποίηση, πχ να φαινεται στην αρχική σελίδα ή σε ένα tab ειδοποιήσεις. Δε χρειάζεται να γίνεται live με τη χρήση των setTimeout ή setInterval.	3%
Καταχώρηση Κριτικής	Μονο αν έχει τελειώσει ο δανεισμός του βιβλίου (κατάσταση successEnd), να μπορεί να γράφει μία κριτική γι αυτό	4%

3.4 Λειτουργίες Επισκέπτη/ Μη συνδεδεμένου Χρήστη (5%)

Ο επισκέπτης θα μπορεί να βλέπει τα διαθέσιμα βιβλία και βασικές πληροφορίες.

Λειτουργία	Περιγραφή	Βαθμολογία
Εύρεση και προβολή βιβλίων	Θα βλέπει όλα τα διαθέσιμα βιβλία, ανα κατηγορία αλλά δε θα μπορεί να	3%

	δανείζεται ένα βιβλίο, να βλέπει ποιες βιβλιοθήκες το έχουν διαθέσιμο κλπ.	
Πληροφορίες για άλλα site με βιβλία, και links στα πανεπιστημιακά ιδρύματα της κρήτης	Θα παρέχει plugins/links από άλλα sites για να βρίσκει έξτρα πληροφορίες, πχ για βιβλία, και links στα πανεπιστημιακά ιδρύματα της κρήτης. Θα πρέπει να φαίνονται οι πληροφορίες για όλους τους χρήστες. Αυτό μπορεί να υπάρχει σαν footer στο σύστημα και να φαίνεται πάντα.	2%

3.5 Οδηγίες για δανεισμό βιβλίου

Ο χρήστης βλέπει ποιες βιβλιοθήκες έχουν το βιβλίο

- Μέσα από τον πίνακα **BooksInLibraries**, λαμβάνουμε πληροφορίες για το ποιες βιβλιοθήκες έχουν ένα βιβλίο διαθέσιμο (δηλαδή έχουν **available=true**)
- Δείχνουμε τις διαθέσιμες βιβλιοθήκες στο χρήστη
- Μόλις επιλέξει από ποια βιβλιοθήκη θα το δανειστεί γίνονται τα ακόλουθα:
- **Η κατάσταση του συγκεκριμένου αντιτύπου για τη συγκεκριμένη βιβλιοθήκη γίνεται available=false στον πίνακα BooksInLibraries**
- Δημιουργείται ένα νέο entry στον πίνακα **Borrowing** με τα εξής στοιχεία:
 - **bookcopy_id** (από τον πίνακα BooksInLibraries)
 - **user_id** (το id του φοιτητή)
 - **fromDate** (η τωρινή ημερομηνία)
 - **toDate** (σε ένα μήνα από τώρα)
 - **status=requested**
- Οπότε οι άλλοι χρήστες δεν μπορούν να επιλέξουν για δανεισμό αυτό το αντίτυπο από τη συγκεκριμένη βιβλιοθήκη, όσο είναι ενεργός ο δανεισμός

Ένας δανεισμός έχει τις εξής καταστάσεις:

- **requested**: το έχει ζητήσει κάποιος φοιτητής
 - μόλις το επιλέγει
- **borrowed**: ο υπεύθυνος βιβλιοθήκης (κάποια άλλη στιγμή) κάνει accept τον δανεισμό, ο δανεισμός ξεκινάει
- **returned**: ο χρήστης πατάει ότι επέστρεψε το βιβλίο (κάποια άλλη στιγμή)
- **successEnd**: ο υπεύθυνος βιβλιοθήκης επιβεβαιώνει (κάποια άλλη στιγμή) ότι επιστράφηκε το βιβλίο και ο δανεισμός τελειώνει
 - Τότε αλλάζει αυτόματα και στον πίνακα **BooksInLibraries**, για το συγκεκριμένο **book_id** την τιμή **available=true**.

- Μόλις τελειώσει ο δανεισμός οι άλλοι χρήστες μπορούν να επιλέξουν για δανεισμό αυτό το αντίτυπο, και θα γίνει η αντίστοιχη διαδικασία
- Μπορείτε να θεωρήσετε ότι δεν δανείζεται ξανά ο ίδιος χρήστης το ίδιο αντίτυπο.

Ο φοιτητής θα μπορεί

- να βλέπει τους δανεισμούς που έχει κάνει στο παρελθόν αλλά και τους ενεργούς δανεισμούς (και πόσες μέρες μένουν μέχρι την επιστροφή)
- Θα μπορεί να κάνει κριτική σε ένα βιβλίο αν το έχει δανειστεί στο παρελθόν και η κατάσταση του είναι `successEnd`
- Θα του έρχεται ειδοποίηση 3 μέρες πριν το τέλος του δανεισμού.

Ο υπεύθυνος βιβλιοθήκης θα μπορεί

- να βλέπει τους δανεισμούς που έχουν γίνει στο παρελθόν
- τους ενεργούς δανεισμούς (και πόσες μέρες μένουν μέχρι την επιστροφή), και να τους κατεβάζει σε PDF. Στον κάθε δανεισμό καλό είναι να φαίνεται το ISBN, ο τίτλος του βιβλίου, η κατάσταση δανεισμού, το όνοματεπώνυμο του χρήστη, πανεπιστήμιο, email και τηλέφωνο.

Δεν υπάρχει περιορισμός για το πως θα φτιαχτούν τα παραπάνω, πχ JAVA Servets, REST APIs, κλπ

3.6 Αναφορά που περιγράφει τη σχεδίαση - Ποιότητα Σχεδίασης-Κώδικας (15%)

Είναι σημαντικό πρώτα να σχεδιάσετε το παραπάνω σύστημα. Σκοπός της σχεδίασης είναι η αποσαφήνιση των παρεχόμενων λειτουργιών και η αποτύπωση των σχεδιαστικών αποφάσεων που θα χρησιμοποιηθούν κατά την υλοποίηση. Η σχεδίαση σας θα πρέπει τουλάχιστον να περιλαμβάνει:

1. τις κλάσεις και τις μεθόδους που χρησιμοποιήσατε
2. τη γενικότερη αρχιτεκτονική του συστήματος. Το σχέδιο σας είναι καλό να απαντάει σε ερωτήματα της μορφής:
 - Πόσα, ποια και με τι αρμοδιότητα servlets χρησιμοποιήσατε (η ότι άλλη τεχνολογία χρησιμοποιήσατε)
 - Πόσα, ποια και με τι αρμοδιότητα JavaScript Libraries/html/jsp χρησιμοποιήσατε
 - Πόσα, ποια και με τι αρμοδιότητα css χρησιμοποιήσατε
 - Τι κάνατε στην πλευρά του client με javascript
 - Ποια είναι τα Ajax τμήματα της εφαρμογής σας;
 - Που χρησιμοποιήσατε REST requests?
 - Ποια APIs χρησιμοποιήσατε;
 - Αν χρειάστηκε να κάνετε αλλαγές/προσθήκες στο σχήμα της βάσης

Το σύστημα πρέπει να είναι διαδικτυακό. Οποιαδήποτε συσκευή εφοδιασμένη με ένα φυλλομετρητή του Ιστού (Web browser) πρέπει να μπορεί να χρησιμοποιήσει την εφαρμογή.

3.7 Χρήση APIs Χαρτών και Οπτικοποίησης, REST, AJAX (10%)

Θα πρέπει να χρησιμοποιηθούν χάρτες, API για οπτικοποίηση, REST Request και AJAX, σε κάποια μέρη της εργασίας. Όπως έχει αναφερθεί και στα παραπάνω βήματα, ενδεικτικά μπορούν να χρησιμοποιηθούν:

- Χάρτες για τις βιβλιοθήκες (μέσω OpenStreetMaps ή GoogleMaps)
- Οπτικοποίηση στατιστικών (μέσω Google Charts)
- REST Requests για βιβλία, και ότι άλλο θέλετε.
- AJAX για μεταφορές τιμών από/προς server

3.8 Bonus - Έξτρα Λειτουργικότητα (8%)

3.8.1. Bonus 1. GiveAway (5%)

Κλήρωση Δώρων GiveAway	<p>Η εφαρμογή να κάνει κλήρωση κάθε μήνα για τους φοιτητές της εφαρμογής. Για κάθε βιβλίο που έχει δανειστεί στο παρελθόν ο κάθε χρήστης θα κερδίζει 1 συμμετοχή, αν έχει κάνει και κριτική θα κερδίζει +2 συμμετοχές. Τα δώρα θα είναι δύο κάθε μήνα, ενώ δε γίνεται ένας χρήστης να πάρει πάνω από 1 δώρο το μήνα.</p> <ul style="list-style-type: none">• 1ο δώρο: Συλλεκτική έκδοση Σειράς Harry Potter• 2ο δώρο : Εκτυπωτής HP <p>Οι νικητές θα βγαίνουν με ανακοίνωση από τον admin (δείτε πίνακα announcements)</p> <p>Οι ανακοινώσεις θα μπορούν να φαίνονται είτε στην αρχική είτε σε μία ξεχωριστή σελίδα.</p>	
------------------------	---	--

3.8.1. Bonus 2. Μία έξτρα Λειτουργικότητα (3%)

Σκεφτείτε και υλοποιήστε μία έξτρα λειτουργικότητα. Μπορείτε να χρησιμοποιήσετε και δεδομένα από APIs όπως <https://rapidapi.com/roftcomp-laGmBwlWlm/api/hapi-books> .

3.7 Διευκρινήσεις

- Σας έχει δοθεί η βάση και κάποια αρχικά στοιχεία, για να είναι εφικτά κάποια επιθυμητά σενάρια, χωρίς να χρειαστεί να χάνεται πολύ χρόνο στην εισαγωγή δεδομένων. Μπορείτε να τα επεξεργαστείτε όλα αυτά όπως σας βολεύει. Θα πρέπει όμως να υποστηρίζεται και η εισαγωγή νέων δεδομένων, για να δουλέψουν αρκετά από τα ζητούμενα της εργασίας
- Στο επόμενο κεφάλαιο δίνονται links για κώδικα για ότι χρειάζεστε (πχ να φτιάχνετε pdf μέσω java και για πολλά άλλα).

- (SOS) Δείτε το tutorial για την βάση και το σχετικό κώδικα

4. Σύνδεσμοι για Τεχνικά Θέματα

4.1 Βασικές Πληροφορίες για website

Ενδεικτικοί σύνδεσμοι ακολουθούν. Γενικά όμως να χρησιμοποιείτε μηχανές αναζήτησης για ό,τι απορία/πρόβλημα έχετε.

Θέμα	Σύνδεσμος
Links σε UOC	www.uoc.gr
Links σε TUC	www.tuc.gr
Links σε HELMEPA	www.helmepa.gr
Link για βιβλία	https://www.abebooks.com/
Άλλα links	Όπως στην άσκηση 1 μπορείτε να βάλετε links σε facebook, twitter κλπ

4.2 Τεχνικές Πληροφορίες για website

Θέμα	Σύνδεσμοι
Χρήση Οπτικοποιήσεων	https://developers.google.com/chart
Χρήση Χάρτη	<ul style="list-style-type: none"> • Δείτε την άσκηση 2 • https://developers.google.com/maps (Θέλουν εγγραφή με χρεωστική, αν και παρέχει αρκετές δωρεάν επιλογές)
Χρήση REST API	https://restfulapi.net/create-rest-apis-with-jax-rs/ Δείτε και Tutorial μαθήματος
Χρήση calendar	https://medium.com/@codenblog/best-html5-calendar-305705d10af
Δημιουργία PDF	https://www.baeldung.com/java-pdf-creation

Υπολογισμός απόστασης μεταξύ συντεταγμένων και απόσταση με αμάξι	https://rapidapi.com/trueway/api/trueway-matrix/
Υπολογισμός Απόστασης μεταξύ δύο σημείων	https://www.geeksforgeeks.org/program-distance-two-points-earth/ για Java και Javascript
Αποστολή email	https://www.tutorialspoint.com/java/java_sending_email.htm

Παράδειγμα <https://rapidapi.com/trueway/api/trueway-matrix/> (ΒΑΛΤΕ ΤΟ ΔΙΚΟ ΣΑΣ KEY)

- Βάζετε στο origin τα lat/lon μίας διεύθυνσης συγκεκριμένης
- Βάζετε στο destination τα lat/lon πολλών διευθύνσεων χωρισμένες με κόμμα
- Λαμβάνετε ως output ένα JSON που για κάθε διαδρομή επιστρέφει τα μέτρα της απόστασης και το χρόνο σε seconds.

Στο από κάτω παράδειγμα, έχουμε βάλει

- σαν origin τα lat/lon της διεύθυνσης Evans 80, Heraklion
- σαν destination τα lat/lon των 62 Martyrwn 80, Heraklion και Venizelou 80, Gazi

```
const data = null;
const xhr = new XMLHttpRequest();
xhr.withCredentials = true;
xhr.addEventListener("readystatechange", function () {
  if (this.readyState === this.DONE) {
    console.log(this.responseText);
  }
});

xhr.open("GET",
  "https://trueway-matrix.p.rapidapi.com/CalculateDrivingMatrix?origins=35.335120%2C25.134450&destinations=35.335770%2C25.118920%3B35.329600%2C25.081010");
xhr.setRequestHeader("x-rapidapi-host", "trueway-matrix.p.rapidapi.com");
xhr.setRequestHeader("x-rapidapi-key", "KEY");
xhr.send(data);
```

Σαν αποτέλεσμα μας επιστρέφει τα εξής. Συγκεκριμένα μας λέει ότι:

- η διαδρομή 0 (Έβανς 80 - 62 μαρτύρων 80) είναι 1806 μέτρα (1.806 χιλιόμετρα) και ότι χρειάζεται κάποιος για να πάει εκεί με αμάξι 340 δευτερόλεπτα (5.6 λεπτά)
- η διαδρομή 1 (Έβανς 80 - Βενιζέλου 80, Γάζι) είναι 5658 μέτρα (5.658 χιλιόμετρα) και ότι χρειάζεται κάποιος για να πάει εκεί με αμάξι 816 δευτερόλεπτα (13.6 λεπτά)

```
{2 items
  "distances": [1 item
    0: [2 items
      0: 1806
      1: 5658
```

```

    ]
  ]
  "durations": [1 item
    0: [2 items
      0: 340
      1: 816
    ]
  ]
}

```

5. Βήματα Έργου - Παράδοση

Τα επιθυμητά βήματα είναι η σχεδίαση και η υλοποίηση του συστήματος με τη σειρά που προαναφέρθηκε. Παρόλα αυτά θα υπάρξει μία παράδοση για το project,

Τα **παραδοτέα** είναι τα εξής:

- Μια αναφορά που να καταγράφει τη σχεδίαση σας, το πώς λειτουργεί το σύστημα, ποια είναι τα συστατικά του, αν είναι εύκολα επεκτάσιμο,
- Ο κώδικας με το Σύστημα με σύντομες οδηγίες εγκατάστασης, υποστηριζόμενης λειτουργικότητας και χρήσης.

Αναλυτική Βαθμολόγηση Για το project

Ενότητα	Βαθμολόγηση
Σχεδίαση - Αναφορά	15%
Administrator	10%
Υπεύθυνος βιβλιοθήκης	25%
Φοιτητής	35%
Επισκέπτης	5%
Σωστή Χρήση REST και AJAX	5%
Σωστή Χρήση Maps και Οπτικοποίησης	5%
Έξτρα Λειτουργικότητα (Bonus)	8%
Σύνολο	108%

Τα ποσοστά που αναφέρονται παραπάνω για κάθε περίπτωση χρήσης, αντιστοιχούν στο μέγιστο βαθμό που μπορεί να λάβει κάποιος αν καλύψει πλήρως και άριστα την αντίστοιχη λειτουργικότητα. Αυτό προφανώς περιλαμβάνει τα δικαιώματα χρήσης (βάσει των τύπων των χρηστών).

5.3 Τρόποι Παράδοσης

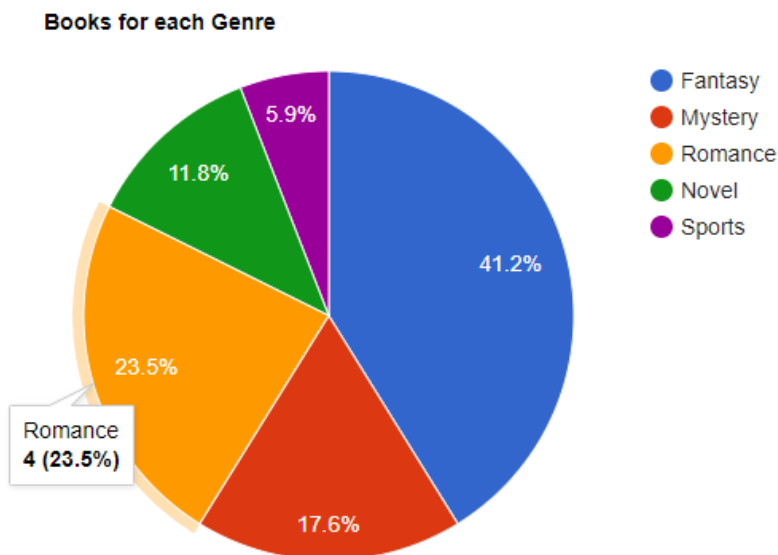
Θα πρέπει να παραδώσει ένα άτομο από κάθε ομάδα στο elearn. Παραδώστε ό,τι έχετε κάνει στη μορφή ενός συμπίεσμένου αρχείου με όνομα teamX.zip (όπου X ο αριθμός ομάδας) με δομή φακέλων της μορφής:

```
/teamInfo
  membersInfo (Ονόματα, AM και emails των μελών της ομάδας)
/documents
  /installationInstructions
  //Με ό,τι πληροφορία χρειάζεται για να εγκατασταθεί και να τρέξει
  /documentation
    /To report που σας ζητείται
    /Περιγραφή του τι τρέχει.
  Επίσης βάλτε ένα doc που να έχει τον πίνακα που εμφανίζεται στην ενότητα βαθμολόγηση
  όπου δίπλα σε κάθε γραμμή γράψτε τι έχετε κάνει
/software
  /webproject
```

7. Ενδεικτικές Οπτικοποιήσεις

Παράδειγμα

- <https://developers.google.com/chart>



8. Βάση δεδομένων

Η βάση δεδομένων και η εξήγηση της έχει δοθεί στο elearn

Τελικό ζητούμενο είναι το σύστημα να είναι χρήσιμο. Άρα με αυτό το σκεπτικό να αποφασίζετε για το πώς θα σχεδιάσετε και θα υλοποιήσετε κάτι. Δεν χρειάζεται να ρωτάτε τους βοηθούς για τον τρόπο εκπλήρωσης μιας απαίτησης. Θεωρείστε ότι εσείς είστε και οι τελικοί χρήστες και το σύστημα πρέπει να είναι τέτοιο ώστε να σας ικανοποιεί. Για τεχνικά ερωτήματα, πρώτα να χρησιμοποιείτε το google και μετά το forum.

Καλή Εργασία