



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών  
**ΗΥ359 – Διαδικτυακός Προγραμματισμός (Χειμερινό εξάμηνο 2023-2024)**  
Διδάσκων: **Μ. Μουνταντωνάκης**  
Θέμα: **Ομαδική Εργασία (project) μαθήματος 2 ατόμων**  
Βοηθοί: Όλοι

## Pet-Care

**Διάρκεια: 11/12/2023 – 25/01/2024**

**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 **Pet Care**. Στόχος του πληροφοριακού συστήματος είναι να επιτρέψει σε ιδιοκτήτες κατοικίδιων να αφήνουν τα κατοικίδια τους για φιλοξενία όσο λείπουν.

**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 pet owner μπορεί να δει όλους τους διαθέσιμους keeper και να μπορεί να επιλέξει ποιες μέρες ζητά να γίνει η φιλοξενία.

**Σενάριο 2:** Να μπορεί ένας pet owner να καταχωρήσει πληροφορίες για το κατοικίδιο του

**Σενάριο 3:** Ένας pet keeper να μπορεί να διαχειρίζεται τις κρατήσεις του και να μπορεί να βρίσκει έξτρα πληροφορίες για το κατοικίδιο που θα φιλοξενήσει

**Σενάριο 4:** Ένας pet owner να μπορεί να μιλάει με έναν pet keeper.

**Σενάριο 5:** Ένας pet owner να μπορεί να κάνει κριτική για έναν pet keeper

### 2. Τύποι Χρηστών

Οι τύποι χρηστών της εφαρμογής είναι:

- Κεντρικός administrator
- Pet Owner
- Pet Keeper
- Επισκέπτες

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

**B.** Οι Pet Keepers κάνουν εγγραφή στο σύστημα και διαχειρίζονται τα κατοικίδια και τις κρατήσεις τους

**Γ.** Οι pet owners θα μπορούν να χρησιμοποιήσουν την εφαρμογή και να εισάγουν στοιχεία αρκεί να έχουν εγγραφεί σε αυτήν (όπως η άσκηση 1,2,3), και να διαχειρίζονται κρατήσεις.

**Δ.** Οι απλοί χρήστες θα μπορούν να δουν πληροφορίες σχετικά με το σύστημα, να δουν πληροφορίες για διαθέσιμους keeper (δεν θα μπορούν όμως να κάνουν κρατήσεις 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 Pet Keeper

- Ενημέρωση στοιχείων/Καταχώρηση πληροφοριών για το χώρο φιλοξενίας
- Διαχείριση κρατήσεων
- Ερωτήσεις προς chatgpt για βοήθεια στα κατοικίδια που θα φιλοξενήσουν/φιλοξενούν,
- Μήνυμα προς ιδιοκτήτη
- Προβολή στατιστικών

### Λειτουργίες Εγγεγραμμένου Χρήστη

- Ενημέρωση στοιχείων
- Καταχώρηση κατοικίδιου
- Εύρεση και προβολή pet keepers με βάση διάφορα κριτήρια
- Επιλογή, Προβολή Φιλοξενία
- Κριτική Pet Keeper (μετά το πέρας της φιλοξενίας, αν θέλει).

### Λειτουργίες Απλού Χρήστη (Επισκέπτη)

- Εύρεση και προβολή pet keeper (χωρίς τη δυνατότητα κράτησης)
- Αρχική σελίδα συστήματος

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

**Για λόγους απλότητας θα θεωρήσουμε ότι ο κάθε owner έχει ένα κατοικίδιο και ο κάθε pet keeper φιλοξενεί ένα κατοικίδιο τη φορά.**

**Μπορεί κάποιος σαν bonus να το κάνει έτσι ώστε ο owner να έχει όσα κατοικίδια θέλει και ο keeper να τα φιλοξενεί όλα μαζί.**

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 <b>username:</b> admin <b>password:</b> admin12*	1%
Προβολή Χρηστών με δυνατότητα διαγραφής	Θα πρέπει να βλέπει κάθε χρήστη (pet owner ή keeper) και να μπορεί να τον διαγράψει. Δεν είναι απαραίτητο να βλέπει όλα τα στοιχεία του,	3%

	μπορείτε να βάλετε ας πούμε να βλέπει username όνομα και επίθετο	
Προβολή στατιστικών	<p>Να μπορεί να βλέπει τα εξής στατιστικά ιδανικά με χρήση google charts (πχ pie ή bar charts):</p> <ul style="list-style-type: none"> <li>• αριθμός γατών/σκύλων</li> <li>• χρήματα που έχουν κερδίσει οι keepers και η εφαρμογή (για κάθε finished κράτηση κερδίζει ο keeper 85% της τιμής και η εφαρμογή 15%)</li> <li>• αριθμός pet keepers και pet owners</li> </ul>	6%

### 3.2 Λειτουργίες Pet Keeper (28%)

Θα πρέπει να του δίνεται η δυνατότητα να συνδεθεί μέσω μιας άλλης σελίδας, από ότι ο pet owner. Ουσιαστικά θα χειρίζεται τις φιλοξενίες του.

Λειτουργία	Περιγραφή	Βαθμολογία
<b>Εγγραφή στο σύστημα/Αλλαγή στοιχείων</b>	Θα πρέπει να μπορεί να γραφτεί στο σύστημα και έπειτα να αλλάζει και τα στοιχεία του αν επιθυμεί (όχι username, email).	5%
<b>Διαχείριση Κρατήσεων</b>	Δείτε την ενότητα 3.5	7%
<b>Ερωτήσεις προς chatgpt για συγκεκριμένα κατοικίδια</b>	Χρησιμοποιώντας το CHATGPT ( <a href="https://rollbar.com/blog/how-to-use-chatgpt-api-with-java/">https://rollbar.com/blog/how-to-use-chatgpt-api-with-java/</a> ), ο pet keeper να μπορεί να στείλει ερωτήσεις και να παίρνει απαντήσεις. Πρέπει να έχετε κάποιες default επιλογές, όπως: Πως να φροντίσω ένα [dog/cat], Δώσε μου πληροφορίες για τη ράτσα [dog/cat] (όπου εκεί μπορεί κάποιος να γράψει μία ράτσα σκύλου/γάτας κλπ). Ιδανικά θα μπορεί να στέλνει αυτές τις ερωτήσεις για ένα ζώο που η κατάσταση της κράτησης είναι accepted. Όμως μπορείτε να το υλοποιήσετε και ξεχωριστά αν θέλετε.	10%

<b>Ανανέωση προφίλ με βάση τις φιλοξενίες που έχει κάνει</b>	<p>Να φαίνονται στο προφίλ του οι εξής πληροφορίες: αριθμός κρατήσεων που έχει κάνει, συνολικές μέρες φιλοξενίας, κριτικές κλπ. Αφορά μόνο περιπτώσεις που είναι finished</p> <p>Δε χρειάζεται να είναι μέσω charts, μπορεί απλά να είναι σε ένα table.</p>	3%
<b>Μήνυμα σε ιδιοκτήτη</b>	Όσο η κατάσταση είναι accepted, μπορεί να μιλήσει με τον ιδιοκτήτη μέσω μηνυμάτων.	3%

### 3.3 Λειτουργίες Pet Owner (32%)

Ο pet owner θα μπορεί να προσθέσει ένα κατοικίδιο, να βρίσκει pet keeper, να αναζητά φιλοξενία για το κατοικίδιο του, να μιλάει και να κάνει κριτική σε keepers.

Λειτουργία	Περιγραφή	Βαθμολογία
<b>Εγγραφή στο σύστημα/Επεξεργασία στοιχείων</b>	Θα πρέπει να μπορεί να γραφτεί στο σύστημα και να επεξεργάζεται τα στοιχεία του	5%
<b>Εισαγωγή στοιχείων για κατοικίδια</b>	Όπως το rest api της τρίτης σειράς ασκήσεων. Μπορεί να γίνει και με servlets.	5%
<b>Εύρεση Pet Keepers , ιδανικά και με βάση την τοποθεσία</b>	<p>Θα μπορεί να δει όλους τους διαθέσιμους pet keepers (θεωρούμε ότι είναι όσοι δέχονται το είδος του κατοικίδιου του χρήστη, και δεν έχουν κάποιο booking σε κατάσταση requested ή accepted)</p> <p>Σαν έξτρα, θα μπορεί να τους δει ταξινομημένους, α) με βάση την απόσταση από τη διεύθυνση του σπιτιού του, β) το χρόνο για να φτάσουν εκεί με αυτοκίνητο, γ) την τιμή ανα βράδυ φιλοξενίας.</p> <ul style="list-style-type: none"> <li>Υπολογισμός απόστασης μεταξύ συντεταγμένων και απόσταση με αμάξι <ul style="list-style-type: none"> <li><a href="https://rapidapi.com/trueway/api/trueway-matrix/">https://rapidapi.com/trueway/api/trueway-matrix/</a></li> </ul> </li> </ul>	8%

	<ul style="list-style-type: none"> <li>• Βάζετε σαν origin τα lat/lon</li> <li>• Δείτε παράδειγμα στα βοηθητικά</li> <li>• Μπορείτε να χρησιμοποιήσετε και άλλο κώδικα, πχ σε java (δείτε τα βοηθητικά)</li> </ul>	
<b>Διαχείριση Κράτησης</b>	Δείτε την ενότητα 3.5	7%
<b>Μήνυμα σε pet keeper</b>	Όσο η κατάσταση είναι accepted, μπορεί να μιλήσει με τον pet keeper μέσω μηνυμάτων.	3%
<b>Καταχώρηση Κριτικής</b>	Μονο αν έχει τελειώσει η φιλοξενία (κατάσταση finished), να μπορεί να γράφει μία κριτική για τον pet keeper	4%

### 3.4 Λειτουργίες Επισκέπτη/ Μη συνδεδεμένου Χρήστη (5%)

Ο επισκέπτης θα μπορεί να βλέπει τους keepers και βασικές πληροφορίες.

Λειτουργία	Περιγραφή	Βαθμολογία
<b>Εύρεση και προβολή keepers</b>	Θα βλέπει όλους τους keepers που υπάρχουν χωρίς να μπορεί να κάνει κράτηση.	3%
<b>Πληροφορίες για άλλα site με κατοικίδια, και links</b>	Θα παρέχει plugins/links από άλλα sites για να βρίσκει έξτρα πληροφορίες, πχ για κατοικίδια, και links σε παρόμοια site. 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 φιλοξενίες- Διαχείριση Κράτησης

Αρχικά θεωρούμε τα εξής:

- Ένας pet owner μπορεί να δει όλους τους διαθέσιμους keepers που πληρούν τις προϋποθέσεις, πχ αν έχει γάτα θα δει αυτούς που δέχονται γάτες.

- Ο pet keeper είναι διαθέσιμος για όλες τις μελλοντικές ημερομηνίες, εκτός αν υπάρχει ήδη στη βάση μία κράτηση σε κατάσταση requested ή accepted.
- Ο pet owner μπορεί να στείλει request μόνο σε ένα keeper, αν το απορρίψει τότε μπορεί να στείλει σε κάποιον άλλο αν θέλει.

Οι καταστάσεις μιας κράτησης είναι οι εξής:

- **requested:** έχει επιλέξει κάποιος χρήστης έναν keeper για συγκεκριμένες ημερομηνίες
- **accepted:** το έχει δεχτεί ο pet keeper
- **rejected:** το έχει αρνηθεί ο pet keeper, δεν μπορεί να αλλάξει αυτή η κατάσταση
- **finished:** Όταν πατήσει ο pet owner ότι η φιλοξενία έγινε επιτυχώς, τότε να τελειώνει η φιλοξενία. Δεν χρειάζεται να κάνουμε κάποιο ιδιαίτερο έλεγχο σε αυτό, θεωρούμε ότι ο οποιαδήποτε στιγμή μπορεί να θεωρήσει μία φιλοξενία done **(και ως είναι σε μελλοντική ημερομηνία, για λόγους απλότητας)**

#### Ο pet owner θα μπορεί

- να βλέπει τους διαθέσιμους pet keepers που πληρούν τις προϋποθέσεις, βλέποντας και τη συνολική τιμή που θα του κοστίσει
- να κάνει κράτηση, βάζοντας αν θέλει και κάποιες έξτρα πληροφορίες, αρκεί να επιλέξει ένα χρονικό διάστημα στο μέλλον
- να πατάει finished όταν τελειώνει η φιλοξενία
- να στέλνει μηνύματα με ένα keeper όταν η κατάσταση είναι accepted

#### Ο pet keeper θα μπορεί

- να δέχεται/απορρίπτει κρατήσεις
- να βλέπει τις κρατήσεις του και πληροφορίες για αυτήν (ποια ζώα θα φιλοξενήσει κλπ).
- να μιλάει με έναν ιδιοκτήτη που έχει κανονιστεί κράτηση μέσω μηνυμάτων (κατάσταση accepted)

Όταν η κατάσταση αλλάξει σε **finished**, θεωρούμε την εξής διαδικασία

- Να αλλάζουν τα στατιστικά του pet keeper
  - Αριθμός φιλοξενιών
  - Συνολικές μέρες φιλοξενίας
- Ο χρήστης να μπορεί να κάνει κριτική στον pet keeper

**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 (μέσω OpenStreetMaps ή GoogleMaps)
- Οπτικοποίηση στατιστικών (μέσω Google Charts)
- REST Requests
- CHATGPT API για πληροφορίες
- AJAX για μεταφορές τιμών από/προς server

### 3.8 Bonus - Έξτρα Λειτουργικότητα (8%)

#### 3.8.1. Bonus 1. GiveAway (4%)

Κλήρωση Δώρων GiveAway	<p>Η εφαρμογή μέσω του admin να κάνει κλήρωση κάθε μήνα για τους pet owners της εφαρμογής. Για κάθε μέρα φιλοξενίας 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"><li>• 1ο δώρο: Δωροεπιταγή 100 ευρώ σε pet shop</li><li>• 2ο δώρο: Δωρεάν καφέ σε cat cafe</li></ul>	
------------------------	--	--



	Οι ανακοινώσεις θα μπορούν να φαίνονται είτε στην αρχική είτε σε μία ξεχωριστή σελίδα. (μπορείτε να φτιάξετε ένα έξτρα table στη βάση σας).	
--	---	--

### 3.8.1. Bonus 2. Τροποποιήστε το πρόγραμμα για να έχει ο κάθε χρήστης πάνω από 1 κατοικίδιο (4%)

Τροποποιήστε το πρόγραμμα για να έχει ο κάθε χρήστης πάνω από 1 κατοικίδιο, και όταν θα επιλέγει keeper, τότε ο keeper θα πρέπει να φιλοξενεί όλα τα κατοικίδια του.

### 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
Σελίδα για Κατοικίδια	<a href="https://www.i-pet.gr/">https://www.i-pet.gr/</a>
Σελίδα για Κατοικίδια	<a href="https://allaboutpets.gr/katoikidia-zoa-gia-mesa-sto-spiti/">https://allaboutpets.gr/katoikidia-zoa-gia-mesa-sto-spiti/</a>
Άλλα links	Όπως στην άσκηση 1 μπορείτε να βάλετε links σε facebook, twitter κλπ

### 4.2 Τεχνικές Πληροφορίες για website

Θέμα	Σύνδεσμοι
------	-----------

Χρήση Οπτικοποιήσεων	<a href="https://developers.google.com/chart">https://developers.google.com/chart</a>
Χρήση Χάρτη	<ul style="list-style-type: none"> <li>• Δείτε την άσκηση 2</li> <li>• <a href="https://developers.google.com/maps">https://developers.google.com/maps</a> (Θέλουν εγγραφή με χρεωστική, αν και παρέχει αρκετές δωρεάν επιλογές)</li> </ul>
Χρήση REST API	<a href="https://restfulapi.net/create-rest-apis-with-jax-rs/">https://restfulapi.net/create-rest-apis-with-jax-rs/</a> Δείτε και Tutorial μαθήματος
Χρήση calendar	<a href="https://medium.com/@codenblog/best-html5-calendar-305705d10af">https://medium.com/@codenblog/best-html5-calendar-305705d10af</a>
Δημιουργία PDF	<a href="https://www.baeldung.com/java-pdf-creation">https://www.baeldung.com/java-pdf-creation</a>
Υπολογισμός απόστασης μεταξύ συντεταγμένων και απόσταση με αμάξι	<a href="https://rapidapi.com/trueway/api/trueway-matrix/">https://rapidapi.com/trueway/api/trueway-matrix/</a>
Υπολογισμός Απόστασης μεταξύ δύο σημείων	<a href="https://www.geeksforgeeks.org/program-distance-two-points-earth/">https://www.geeksforgeeks.org/program-distance-two-points-earth/</a> για Java και Javascript
Αποστολή email	<a href="https://www.tutorialspoint.com/java/java_sending_email.htm">https://www.tutorialspoint.com/java/java_sending_email.htm</a>

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%
Pet Keeper	28%
Pet Owner	32%
Επισκέπτης	5%
Σωστή Χρήση REST και AJAX	5%

Σωστή Χρήση Maps και Οπτικοποίησης	5%
Έξτρα Λειτουργικότητα ( <b>Bonus</b> )	8%
<b>Σύνολο</b>	<b>108%</b>

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

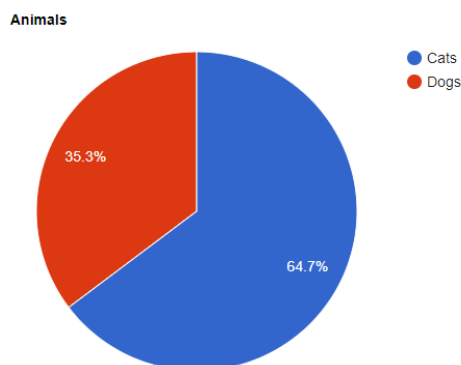
Θα πρέπει να παραδώσει ένα άτομο από κάθε ομάδα στο elearn. Παραδώστε ό,τι έχετε κάνει στη μορφή ενός συμπίεσμένου αρχείου με όνομα teamX.zip (όπου X ο αριθμός ομάδας) με δομή φακέλων της μορφής:

```
/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***