# ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

# Μάρτιος 2019

**Απαλλακτική εργασία μαθήματος**

1. **Στόχος εργασίας**

Σκοπός της εργασίας σας είναι να προγραμματίσετε μία ολοκληρωμένη δυναμική εφαρμογή λογισμικού (web app programming). H συγκεκριμένη εφαρμογή θα αφορά σε ένα σύστημα διαχείρισης θέσεων και κράτησης εισιτηρίων. Το περιεχόμενο της λύσης σας και η τελική της μορφή θα εξαρτάται από το ειδικό θέμα περιεχομένου της εργασίας που θα αποφασίσει η ομάδα σας και το οποίο είναι ελεύθερης επιλογής.

Ο κώδικας που θα παραδώσετε για την εργασία σας θα περιλαμβάνει client-side και server-side programming. Οι γλώσσες/γλώσσες σήμανσης που θα χρησιμοποιήσετε περιλαμβάνουν (αλλά όχι να περιορίζονται) στις: HTML, CSS, JAVASCIPT, PHP.

Η ιστοσελίδα σας θα πρέπει να διαθέτει βάση δεδομένων, π.χ. MySQL, την οποία φυσικά να χρησιμοποιεί/αξιοποιεί αντλώντας/ενημερώνοντας δεδομένα.

Η ιστοσελίδα σας θα πρέπει να είναι «responsive» και να παρέχει ικανοποιητικό «viewing experience» σε ένα σημαντικό εύρος από συσκευές (από υπολογιστές μέχρι κινητά τηλέφωνα).

Είναι επίσης επιθυμητό, η ιστοσελίδα σας να έχει όσο το δυνατόν πιο φιλικό User Interface. Για το λόγο αυτό μπορείτε να αξιοποιήσετε εφέ ή/και δυνατότητες γραφικών από κάποια βιβλιοθήκη (προτείνονται bootstrap & angularJS). Για τη χρωματική αντιμετώπιση της εφαρμογής χρησιμοποιήστε βοήθεια βασικών χρωματικών θεμάτων για σχεδίαση ιστοσελίδων (π.χ. https://color.adobe.com/explore/?filter=most-popular&time=month) .

Για να αξιολογηθεί η εργασία σας με άριστα το 10 θα πρέπει να δημιουργήσετε την ιστοσελίδα που ζητείται χωρίς τη χρήση έτοιμου λογισμικού διαχείρισης και κράτησης θέσεων τρίτων (προγραμματιστών/εταιριών).

1. **Συγκέντρωση πληροφοριών**

Πριν την υλοποίηση της ιστοσελίδας ζητείται να προηγηθεί σύντομη ανάλυση απαιτήσεων και έρευνα για παρόμοιες λύσεις. Έτσι θα καταγράψετε καλύτερα τις απαιτήσεις για τη δημιουργία της ιστοσελίδας σας και ταυτόχρονα θα συλλέξετε και ιδέες ή/και πρόσθετες πληροφορίες για αυτήν.

Η ανάλυση απαιτήσεων σύμφωνα με τις αρχές τεχνολογίας λογισμικού θα πρέπει να περιλαμβάνει σύντομη καταγραφή και ανάλυση των διαδικασιών και τουλάχιστο τις διαδικασίες σε διάγραμμα χρήσης περίπτωση (use case diagram) και διάγραμμα ακολουθιακό (sequence diagram) (συνολικά αναφορά ως 3-5 σελίδες) χρησιμοποιώντας PlantUML scripts (http://plantuml.com/).

1. **Δομή εργασίας**

Η δυναμική ιστοσελίδα που θα υλοποιήσετε θα πρέπει να περιέχει τα ακόλουθα (με μικρές προσαρμογές σε σχέση με το εκάστοτε ειδικό θέμα):

* Ημερολόγιο διαθεσιμότητας και κρατήσεων
* φόρμα κράτησης
* υποσύστημα διαχείρισης χρηστών/ρόλων
* υποσύστημα διαχείρισης χώρων/αιθουσών/ταινιών
* καταγραφή εισόδου στο χώρο/αίθουσα/ταινία

Ανάλογα με το θέμα της εργασίας σας, η ιστοσελίδα σας μπορεί να περιέχει και άλλου είδους πληροφορίες τις οποίες θα υλοποιήσετε εσείς.

* Ημερολόγιο διαθεσιμότητας και κρατήσεων

Θα υλοποιήσετε ένα ημερολόγιο ή κατάλογο όπου θα εμφανίζεται η διαθεσιμότητα χώρων, η κράτηση εισιτηρίων που έχει πραγματοποιηθεί και ο βαθμός πληρότητας από επισκέπτες που εισήλθαν στην αίθουσα. Τα στοιχεία θα προκύπτουν από δεδομένα αποθηκευμένα στη ΒΔ σας. Πιο συγκεκριμένα, θέλουμε να εμφανίζονται τουλάχιστον κωδικός και περιγραφή της, το ονοματεπώνυμο του χρήστη που έκανε την κράτηση. Το ημερολόγιο είναι επιθυμητό να εμφανίζεται μηνιαία ή/και σε κατάλογο. Αν το επιθυμείτε, μπορεί να προσθέσετε και επιπλέον πληροφορίες.

* Φόρμα κράτησης

Εγγεγραμμένοι χρήστες στο σύστημά σας (με ρόλο δυνατότητας κράτησης) αφότου θα εισάγονται στο σύστημα θα μπορούν να επιλέξουν να κάνουν κράτηση εισιτηρίου σε κάποια ημερομηνία και για συγκεκριμένο ταινία.

Με την καταχώρηση της κράτησης θα είναι δυνατό ο χρήστης να μπορεί να επιλέξει να αποσταλούν τα στοιχεία της κράτησης με email στους προσκεκλημένους (κωδικός, όνομα, περιγραφή, θέση και φωτογραφία αίθουσας) θα παρουσιάζονται στην οθόνη καταγραφής με έμφαση στην ημερομηνία της κράτησης.

* Υποσύστημα διαχείρισης εισιτηρίων

Θα δημιουργήσετε ένα σύστημα διαχείρισης εισιτηρίων. Χρήστης του συστήματός σας που θα έχει ρόλο διαχειριστή εισιτηρίων (θα μπορούν να υπάρχουν πολλοί χρήστες με τον ίδιο ρόλο) θα είναι δυνατό:

* Να εισάγει/τροποποιεί/διαγράφει βασικά στοιχεία ταινίας
* Να εισάγει/τροποποιεί/διαγράφει σε έναν χώρο στοιχεία για το διάστημα διαθεσιμότητας ταινίας από – έως ημερομηνία και ώρα
* Στις ανωτέρω επεξεργασίες θα πρέπει να γίνεται έλεγχος κατά πόσο υπάρχουν κρατήσεις ήδη που επηρεάζονται από αλλαγή της διαθεσιμότητας ή διαγραφή της αίθουσας και να ειδοποιείται ο χρήστης της κράτησης με email για την αλλαγή.
* υποσύστημα διαχείρισης χρηστών

Θα δημιουργήσετε ένα σύστημα διαχείρισης χρηστών συνδυαστικά με την αυθεντικοποίηση στο σύστημά σας.

Η αυθεντικοποίηση στο σύστημά σας θα χρησιμοποιεί το σύστημα αυθεντικοποίησης χρηστών της Google μέσω αυθεντικοποίησης ροής OAuth 2.0 <https://developers.google.com/api-client-library/>

Η εγγραφή και εισαγωγή στο σύστημα θα μπορεί να γίνει και με χρήση ενός τουλάχιστο από τα : third party SSO όπως Google API, Facebook API Linkedin API, github API.

Επίσης να έχετε το δικό σας σύστημα αυθεντικοποίησης.

Επιπλέον διαχειριστής του συστήματός σας θα μπορεί να αναθέτει σε κάθε χρήστη έναν ή και περισσότερους ρόλους. Οι ρόλοι θα καθορίζουν την πρόσβαση στις λειτουργίες όπως αναφέρετε παραπάνω.

* καταγραφή εισόδου στο χώρο/αίθουσα/ταινία

Η εφαρμογή σας θα διαθέτει επιλογή για την καταγραφή ενός επισκέπτη πριν την είσοδό του σε έναν χώρο.

Ο χρήστης θα κάνει καταγραφή των στοιχείων του μέσα από ένα περιβάλλον φιλικό και κατά προτίμηση σχεδιασμένο με προσέγγιση mobile first. Tα στοιχεία του θα περιλαμβάνουν υποχρεωτικά πεδία: κινητό τηλέφωνο, όνομα, επώνυμο, email, και (υποχρεωτικό!) φωτογραφία

Εάν στη συνέχεια το σύστημά σας με βάση το κινητό του τηλέφωνο ή / και το email θα δίνει τη δυνατότητα στον επισκέπτη να κάνει αναζήτηση για να βρει τυχόν κράτηση σε συγκεκριμένη ταινία/αίθουσα/χώρο. Εφόσον βρεθεί αίθουσα-χώρος για το εισιτήριο τότε θα εμφανίζει μήνυμα με καλωσόρισμα (εξατομικευμένο) και τα στοιχεία της κράτησης (κωδικός, όνομα αίθουσας, περιγραφή, τίτλο ταινίας) θα παρουσιάζονται στην οθόνη καταγραφής με έμφαση στη θέση της αίθουσας.

1. **Template**

Μπορείτε να δημιουργήσετε το δικό σας template. Εναλλακτικά, μπορείτε να βασιστείτε σε κάποιο υπάρχον template το οποίο ανταποκρίνεται στις ανάγκες τις ιστοσελίδας σας. Ωστόσο το template θα πρέπει να κάνει χρήση του Bootstrap Framework (http://getbootstrap.com/).

1. **Σύστημα**

Το σύστημα προτείνεται να βασίζεται σε framework PHP αρχιτεκτονικής MVC με framework όπως το Yii (<https://www.dailyrazor.com/blog/best-php-frameworks/>). Αναφορικά με την πρόσβαση σε δεδομένα του backend θα εφαρμόσετε ένα επίπεδο αφαίρεσης πρόσβασης δεδομένων Data Access Layer ώστε να οργανώσετε, ομαδοποιήσετε και βελτιώσετε την πρόσβαση στα δεδομένα της ΒΔ (προτείνεται το DAL doctrine για PHP <https://www.doctrine-project.org/projects/dbal.html> ή αντίστοιχο). Μπορείτε να χρησιμοποιήσετε ενναλακτικές προσεγγίσεις για τη γλώσσα προγραμματισμού και το framework.

Εναλλακτικά μπορείτε να μην εφαρμόσετε php framework αλλά απλή php όπου στην περίπτωση αυτή η εργασία σας θα λαμβάνει άριστα 9/10.

Στόχος του συστήματος είναι να ικανοποιεί τις λειτουργικές προδιαγραφές που περιγράφονται (και ό,τι παραπάνω επιλέξετε) και να περιλαμβάνει οργάνωση στον κώδικα που θα προγραμματίσετε.

1. **Τρόπος παράδοσης**

Για την παράδοση της τελικής εργασίας, θα προσκομίσετε σε CD το αποτέλεσμα και θα τοποθετήσετε σε ένα αρχείο zip τα ακόλουθα:

* Αρχεία php
* Αρχεία html, css
* Αρχεία πολυμέσων, εικόνων, κ.λπ.
* Αρχεία Javascript
* Αρχεία άλλων γλωσσών/βιβλιοθηκών που τυχόν χρησιμοποιήσατε
* Βάση δεδομένων με τους πίνακες και τα δεδομένα (τη ΒΔ θα πρέπει να την έχετε κάνει export)
* Αναφορά

Η εργασία θα υλοποιείται από 2 έως 3 άτομα σε ομάδα η οποία θα πρέπει να επιλέξει ειδικότερο αντικείμενο για την υλοποίηση του ανωτέρω:

* **Ειδικό θέμα εργασίας ομάδας:** Τα εισιτήρια και οι κρατήσεις μπορούν να είναι για ταινίες σινεμά, θέατρα, χώρο στάθμευσης, συναυλίες και όπου αλλού γίνεται χρήση εισιτηρίων και κράτησης κλπ.

1. **Τεχνολογικές Εναλλακτικές Υλοποίησης**

Είναι δυνατό να υποβάλλετε εργασία με εναλλακτικές τεχνολογίες υλοποίησης στα επιμέρους στοιχεία εφόσον τηρηθεί η αντίστοιχη αρχιτεκτονική προσέγγιση. (π.χ. Java, Python, .NET C#, postgress, MS SQL Srv, Oracle, firebase κλπ).

1. **Συνδυαστική Εργασία με το μάθημα Τεχνολογία Εξατομικευμένου Λογισμικού.**

Είναι δυνατό να υποβάλλετε εναλλακτικά συνδυαστική εργασία με το μάθημα τεχνολογία εξατομικευμένου λογισμικού. Στο πλαίσιο αυτό θα πρέπει να καλύπτονται:

(α) οι λειτουργικές και λοιπές προδιαγραφές του μαθήματος τεχνολογία εξατομικευμένου λογισμικού και

(β) ταυτόχρονα να τηρούνται οι τεχνολογικές προδιαγραφές της παρούσας εργασίας καθώς και

(γ) να αναπτυχθούν αντίστοιχης έκτασης υποσυστήματα κατ’ αναλογία με την παρούσα εργασία.