



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σύστημα κράτησης εισιτηρίων - Διαδικασίες
αναζήτησης και κράτησης»

Του φοιτητή

Ζερβάκη Θεόδωρου

ΑΜ: 093453

Επιβλέπων

Μιχαήλ Σαλαμπάσης

Καθηγητής

9/6/2020

Σύστημα κράτησης εισιτηρίων - Διαδικασίες αναζήτησης και κράτησης

Κωδικός: 20115

Φοιτητής: Ζερβάκης Θεόδωρος

Επιβλέπων: Μιχαήλ Σαλαμπάσης

Ημερομηνία ανάληψης: 21-3-2020

Ημερομηνία περάτωσης: 9-6-2020

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Θεόδωρου Ζερβάκη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Οι αλλαγές που συντελούνται τα τελευταία χρόνια σε τεχνολογικό, και όχι μόνο, επίπεδο επηρεάζουν πολλούς τομείς της σύγχρονης ζωής. Ένας από αυτούς είναι και το ηλεκτρονικό εμπόριο. Στο πλαίσιο των αλλαγών αυτών δεν είναι δυνατό να νοούμαστε μία κοινωνία χωρίς πρόσβαση σε αγαθά και υπηρεσίες από την άνεση του σπιτιού μας, ειδικά όταν οι τεχνολογίες που στηρίζονται είναι σε πολύ ώριμο στάδιο.

Πολύ συχνά επισκεπτόμαστε ιστοσελίδες που προσφέρουν τέτοιου είδους περιεχόμενο και μπορούμε να πούμε ότι έχουμε διαμορφώσει ο καθένας μια εικόνα για το τι περιμένουμε να δούμε. Μάλιστα εξίσου εύκολα θα εκφέρουμε γνώμη, ακόμα και δημόσια εφόσον μας προτρέπει και η εφαρμογή. Αυτό είναι και ένα σημαντικό μέρος στον κύκλο ανάπτυξης λογισμικού, η ανατροφοδότηση, η οποία είναι βασικό κομμάτι της εξέλιξης και της βελτίωσης της ποιότητας λογισμικού.

Περίληψη

Το αντικείμενο αυτής της εργασίας, όπως μαρτυρά ο τίτλος της είναι η ανάπτυξη των διαδικασιών αναζήτησης και κράτησης ενός διαδικτυακού συστήματος κρατήσεων εισιτηρίων. Συγκεκριμένα έγινε ανάπτυξη λογισμικού τύπου ηλεκτρονικού καταστήματος, βασισμένο σε ανάλογο μοντέλο δεδομένων κράτησης θέσεων και διαχείρισης χρηστών. Το μοντέλο αυτό ορίζει ενδεικτικά τις έννοιες του χρήστη, του γεγονότος, της τοποθεσίας διεξαγωγής, της θέσης και της κράτησης, καθώς και τις ανάλογες συσχετίσεις και περιορισμούς.

Από πλευράς λειτουργικότητας, οι κύριες λειτουργίες του αντιστοιχίζονται σε αυτές ενός ηλεκτρονικού καταστήματος, όπως είναι η σελίδα καταστήματος όπου παρουσιάζονται όλες οι κατηγορίες προϊόντων και παρέχεται πλοήγηση και αναζήτηση. Η σελίδα προϊόντος όπου παρουσιάζεται το προϊόν και δίνεται η δυνατότητα προσθήκης στο καλάθι αγορών. Η σελίδα με το καλάθι αγορών και η σελίδα πληρωμής - ολοκλήρωσης της παραγγελίας. Στην περίπτωση της εργασίας είναι αντίστοιχα, η σελίδα κατηγοριών γεγονότων με δυνατότητες δυναμικής αναζήτησης, η σελίδα παρουσίασης γεγονότος με δυνατότητα κράτησης, σελίδα για το καλάθι αγορών και σελίδα πληρωμής.

Άλλες λειτουργίες είναι η εγγραφή και η είσοδος - ταυτοποίηση χρήστη με τις κατάλληλες λειτουργίες ελέγχου πρόσβασης, η δημιουργία γεγονότος από εγγεγραμμένο και ταυτοποιημένο στην υπηρεσία χρήστη (διαχειριστή), η διαχείριση χρηστών από τον διαχειριστή, η επεξεργασία των πληροφοριών του χρήστη στο σύστημα (σελίδα προφίλ) και η λειτουργία αποσύνδεσης.

Abstract

The subject of this paper, as its title suggests, is the development of search and booking procedures for an online ticket booking system. Specifically, the software was developed as an online store-type software (e-shop) and was based on a similar data model for booking and user management. This model defines the concepts of user, event, location, location and reservation, as well as the relevant correlations and constrains.

In terms of functionality, its main functions correspond to those of an online store, such as the store page where all product categories are presented and navigation and search are provided. The product page where the product is presented and the possibility to add to the shopping cart is given. The shopping cart page and the payment - order completion page. In this case, respectively, the event categories page with dynamic search features, the event presentation page with reservation options, the shopping cart page, and the payment page.

Other functions are registration and login - user authentication with the appropriate access control functions, the creation of an event by a registered and authenticated user, the processing of his information in the system (profile page) and the logout function.

Ευχαριστίες

Ευχαριστώ τον επιβλέπων καθηγητή κ. Μιχάλη Σαλαμπάση, γιατί μου έδωσε την ευκαιρία να ασχοληθώ με ένα ιδιαίτερα ενδιαφέρον και πολύ επίκαιρο θέμα, όπως και για την ουσιαστική συμβολή του στην ανάπτυξη του θέματος αυτού. Τον ευχαριστώ θερμά για τις κρίσιμες παρατηρήσεις του και τις ωφέλιμες κριτικές του κατά την πορεία της επεξεργασίας του θέματος.

Ακόμα ευχαριστώ όλους τους καθηγητές που συμμετείχαν στην εκπαίδευσή μου, καθώς μου έδωσαν τα κατάλληλα κίνητρα και τις απαραίτητες γνώσεις για να φτάσω στο στάδιο παρουσίασης της πτυχιακής μου εργασίας.

Θερμά ευχαριστώ την οικογένειά μου που με στήριξε αυτά τα χρόνια και έχει συνδράμει τα μέγιστα σε όλη τη διάρκεια φοίτησής μου στο τμήμα.

Επίσης ιδιαίτερο ευχαριστώ τους συμφοιτητές μου Μαρία, Κώστα, Γιάννη και Βασίλη για την πολύτιμη στήριξή και συμπαράστασή τους.

Τέλος, θα ήθελα να απευθύνω τις ευχαριστίες μου στα μέλη της Εξεταστικής Επιτροπής, που δέχθηκαν να αξιολογήσουν την παρούσα πτυχιακή εργασία.

Ευρετήριο Περιεχομένων

Πρόλογος.....	3
Περίληψη	4
Abstract	5
Ευχαριστίες	6
Ευρετήριο Περιεχομένων.....	7
Ευρετήριο εικόνων	10
Εισαγωγή.....	11
Κεφάλαιο 1: Το μοντέλο δεδομένων	12
1.1 Εισαγωγή.....	12
1.2 Τι είναι μια διαδικτυακή εφαρμογή.....	12
1.3 Το μοντέλο για το ticket-manager.....	12
1.3.1 Πίνακας Χρήστες (users)	13
1.3.2 Πίνακας Τοποθεσία (venue).....	13
1.3.3 Πίνακας Πόλη (city).....	14
1.3.4 Πίνακας Υποπεριοχή (subarea)	14
1.3.5 Πίνακας Θέση (seat).....	14
1.3.6 Πίνακας Κράτηση (reservation)	14
1.3.7 Πίνακας Γεγονός (event)	14
1.3.8 Πίνακες Τύπος και Οικογένεια Γεγονότος (EventType, FamilyEvent)	14
1.4 Το υπόλοιπο μοντέλο	14
1.4.1 Ρόλοι και Ρόλοι Χρήστη	14
1.4.2 Ταυτοποίηση Χρήστη με Τεκμήριο (token).....	15
1.4.3 Συνδέσεις Χρήστη	15
1.4.4 Εξουσιοδότηση Βάση Αξιώσεων.....	15
1.5 Το σχήμα	16
1.5.1 Τύποι πεδίων	16
1.5.2 Ακεραιότητα: Κανόνες και Περιορισμοί	16
1.6 Επίλογος.....	17
Κεφάλαιο 2: Οι τεχνολογίες υλοποίησης.....	18
2.1 Εισαγωγή.....	18
2.2 Γλώσσα Σήμανσης Υπερκειμένου HTML.....	18
2.3 Διαδοχικά Φύλλα Ύφους CSS.....	19
2.4 Γλώσσες σεναρίου JavaScript και PHP.....	22
2.4.1 Η γλώσσα JavaScript	22
2.4.2 Η βιβλιοθήκη jQuery και η τεχνική Ajax	24
2.4.3 Η γλώσσα PHP	25
2.5 Η δομή της εμφάνισης της εφαρμογής	28

Το μοντέλο δεδομένων

2.6	Η γλώσσα SQL	30
2.7	Ο διακομιστής Apache HTTP	31
2.8	Το σύστημα διαχείρισης βάσης δεδομένων MySQL	31
2.9	Επίλογος.....	32
Κεφάλαιο 3: Ηλεκτρονικό εμπόριο και διαχείριση γεγονότων.....		33
3.1	Εισαγωγή.....	33
3.2	Ορισμός ηλεκτρονικού εμπορίου.....	33
3.3	Πλεονεκτήματα και Μειονεκτήματα	34
3.3.1	Πλεονεκτήματα για τους καταναλωτές	34
3.3.2	Πλεονεκτήματα για τις Επιχειρήσεις	34
3.3.3	Μειονεκτήματα για τους Καταναλωτές.....	35
3.3.4	Μειονεκτήματα για τις Επιχειρήσεις	35
3.4	Κατηγορίες Ηλεκτρονικού Εμπορίου	35
3.4.1	Επιχείρηση προς Επιχείρηση - Business - to - Business (B2B).....	35
3.4.2	Επιχείρηση προς Καταναλωτή - Business – to - Consumer (B2C).....	36
3.4.3	Κράτος με Επιχείρηση – Government – to - Business (G2B)	36
3.4.4	Κράτος με Καταναλωτή – Government-to-Customer (G2C)	36
3.4.5	Άλλες κατηγορίες.....	37
3.5	Ορισμός Ηλεκτρονικού Καταστήματος – e-shop	37
3.6	Διαχείριση γεγονότων	38
3.7	Συστήματα κρατήσεων εισιτηρίων.....	38
3.8	Επίλογος.....	39
Κεφάλαιο 4: Υλοποίηση Ticket Manager		40
4.1	Εισαγωγή.....	40
4.2	Λειτουργία αρχικής σελίδας index.php	40
4.3	Λειτουργίες της σελίδας αναζήτησης select.php.....	41
4.4	Λειτουργίες στην σελίδα κράτησης event.php	44
4.5	Οι λειτουργίες εγγραφής και εισόδου χρήστη	47
4.6	Σελίδες διαχειριστή	48
4.7	Σελίδες λογαριασμού χρήστη, καλάθι και ολοκλήρωση αγοράς	50
4.8	Επίλογος.....	51
Κεφάλαιο 5: Εγχειρίδιο χρήσης ticket manager.....		52
5.1	Εισαγωγή.....	52
5.2	Δημιουργία λογαριασμού και σύνδεση	52
5.3	Αρχική σελίδα και μενού πλοήγησης	55
5.4	Σελίδα λογαριασμού χρήστη και λειτουργία εξόδου.....	55
5.5	Σελίδες διαχειριστή	56
5.6	Σελίδα αναζήτησης.....	58

Το μοντέλο δεδομένων

5.7	Σελίδα κράτησης γεγονότος.....	59
5.8	Σελίδες καλάθι αγορών και πληρωμής	60
5.9	Επίλογος.....	62
	Κεφάλαιο 6: Συμπεράσματα και μελλοντικές επεκτάσεις.....	63
	Βιβλιογραφικές αναφορές	64

Ευρετήριο εικόνων

Εικόνα 1: Το διάγραμμα οντοτήτων συσχετίσεων	15
Εικόνα 2: Η αρχή του eventStyle.css.....	21
Εικόνα 3: Μέρος του w3.css.....	22
Εικόνα 4: Συνάρτηση επαναφοράς πεδίου ημερομηνίας στην select.php	23
Εικόνα 5: Παράθυρο confirm στην cart.php	23
Εικόνα 6: Σενάριο JavaScript στην σελίδα εισόδου χρήστη login.php	23
Εικόνα 7: jQuery για επιλογή κατηγοριών στην select.php.....	24
Εικόνα 8: Σενάριο Ajax με jQuery που εκτελεί την μέθοδο POST ασύγχρονα.....	25
Εικόνα 9: Δημιουργία HTML δυναμικά με χρήση PHP	27
Εικόνα 10: Λειτουργία σύνδεσης με τη βάση δεδομένων σε PHP.....	27
Εικόνα 11: Η εντολή include της PHP	28
Εικόνα 12: Οι πληροφορίες στην κεφαλίδα της header.php	28
Εικόνα 13: Κομμάτι του μενού πλοήγησης στην header.php	29
Εικόνα 14: Μενού χρήστη	29
Εικόνα 15: Μενού διαχειριστή	29
Εικόνα 16: Εκτέλεση εντολής DELETE της SQL με PHP	30
Εικόνα 17: Το περιβάλλον της εφαρμογής MySQL Workbench.....	32
Εικόνα 18: Συνάρτηση lastposted() στην index.php.....	40
Εικόνα 19: Απόδοση τιμών στις μεταβλητές στην do_search.php.....	41
Εικόνα 20: Περιπτώσεις κανενός η ενός υποβληθέν πεδίου στην do_search.php	42
Εικόνα 21: Περιπτώσεις συνδυασμού δύο υποβληθέντων πεδίων στην do_search.php	42
Εικόνα 22: Οι συναρτήσεις στην do_search.php.....	43
Εικόνα 23: Γέμισμα επιλογών πόλης και τύπου στην select.php.....	44
Εικόνα 24: λειτουργίες JS στην event.php	45
Εικόνα 25: Λειτουργίες PHP στην event.php	45
Εικόνα 26: Η σελίδα seatselect.php	46
Εικόνα 27: Η σελίδα do_register.php	46
Εικόνα 28: Λειτουργίες ελέγχου της φόρμας στην do_signup.php.....	47
Εικόνα 29: Μέρος της σελίδας do_login.php.....	48
Εικόνα 30: Το σενάριο JS στην dashboard.php	49
Εικόνα 31: Η σελίδα add_event.php	49
Εικόνα 32: Συνάρτηση πίνακα κρατήσεων και σενάριο ακύρωσης κράτησης στην profile.php.....	50
Εικόνα 33: Το PHP μέρος της checkout.php.....	51
Εικόνα 34: Η σελίδα σύνδεσης login.php.	52
Εικόνα 35: Η σελίδα εγγραφής χρήστη signup.php.....	53
Εικόνα 36: Λάθος εισαγωγή στοιχείων κατά την εγγραφή.....	53
Εικόνα 37: Επιτυχής εγγραφή στο σύστημα.	54
Εικόνα 38: Λάθος εισαγωγή κατά την σύνδεση	54
Εικόνα 39: Η αρχική σελίδα index.php	55
Εικόνα 40: Σελίδα επεξεργασίας λογαριασμού profile.php	56
Εικόνα 41: Επιτυχής αποσύνδεση	56
Εικόνα 42: Η σελίδα dashboard.php.....	57
Εικόνα 43: Μέρος της σελίδας new_event.php	57
Εικόνα 44: Αναζήτηση βάση συνδυασμού κατηγοριών και ημερομηνίας	58
Εικόνα 45: Μέρος της σελίδας κράτησης event.php.....	59
Εικόνα 46: Αναδυόμενο παράθυρο επιλογής θέσης.....	60
Εικόνα 47: Μήνυμα επιτυχούς κράτησης	60
Εικόνα 48: Ακύρωση κράτησης από το καλάθι αγορών	61
Εικόνα 49: Σελίδα checkout πριν την πληρωμή	61
Εικόνα 50: Σελίδα checkout μετά την πληρωμή	62

Εισαγωγή

Το θέμα που πραγματεύεται η πτυχιακή εργασία είναι η ανάπτυξη των διαδικασιών αναζήτησης και κράτησης θέσης ενός υπάρχοντος διαδικτυακού συστήματος διαχείρισης κρατήσεων εισιτηρίων. Το σύστημα είναι βασισμένο σε μοντέλο δεδομένων με διακριτές οντότητες όπως ο χρήστης, το γεγονός, η τοποθεσία, η υποπεριοχή στην τοποθεσία, η θέση και η κράτηση.

Σκοπός είναι να υλοποιηθούν οι έννοιες αυτές και να παρουσιαστούν στον χρήστη με τρόπο απλό και οικείο ενώ παράλληλα οι λειτουργίες του συστήματος να είναι δυναμικές και να καλύπτουν τις απαιτήσεις τις σύγχρονης ηλεκτρονικής αγοράς.

Στα κεφάλαια που ακολουθούν περιγράφεται το μοντέλο δεδομένων και το σχήμα, οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση, τα δομικά στοιχεία της εφαρμογής, καθώς και όλες οι επιμέρους τεχνικές λεπτομέρειες, όπως η ανάλυση του κώδικα υλοποίησης και το εγχειρίδιο χρήσης. Τέλος παρατίθενται τα συμπεράσματα, όπως και πιθανές μελλοντικές βελτιώσεις - ιδέες.

Πιο αναλυτικά, στο Κεφάλαιο 1: γίνεται αναλυτική παρουσίαση του μοντέλου δεδομένων και του σχήματος στα οποία βασίστηκε η εφαρμογή, με παράθεση των οντοτήτων με τους τύπους και τα γνωρίσματά τους. Καθώς αναλύονται και οι περιορισμοί ακεραιότητας του.

Στο Κεφάλαιο 2: παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος σε όλα τα επίπεδα όπως το κομμάτι αλληλεπίδρασης με το χρήστη (front end) και τα τρία μέρη της ραχοκοκαλιάς (backend) του συστήματος: διακομιστής ιστού (web server), εφαρμογή (application) και σύστημα διαχείρισης βάσης δεδομένων (DBMS). Στο κεφάλαιο αυτό θα σχηματίσουμε και μια πρώτη εικόνα για την εμφάνιση της εφαρμογής.

Στο Κεφάλαιο 3: γίνεται εισαγωγή στο ηλεκτρονικό εμπόριο γενικά, όπως και στην ευρύτερη όψη των συστημάτων διαχείρισης γεγονότων. Ακόμα θα δούμε πως ξεκίνησαν και τι εικόνα παρουσιάζουν σήμερα τα ηλεκτρονικά συστήματα κρατήσεων εισιτηρίων.

Στο Κεφάλαιο 4: αναλύεται ο κώδικας των κύριων και επιμέρους λειτουργιών του συστήματος. Τέτοιες λειτουργίες είναι η αναζήτηση, η κράτηση, το καλάθι, οι σελίδες διαχειριστή και οι λειτουργίες εισόδου και εγγραφής στην εφαρμογή.

Στο Κεφάλαιο 5: παρατίθεται το εγχειρίδιο χρήσης (user manual) στο οποίο αναλύονται όλες οι λειτουργίες της εφαρμογής από την πλευρά του απλού χρήστη αλλά και του διαχειριστή συστήματος. Οι βασικότερες λειτουργίες είναι η αναζήτηση γεγονότος και η κράτηση θέσης.

Στο Κεφάλαιο 6: γίνεται σύνοψη των δυνατοτήτων του συστήματος και παρατίθενται σκέψεις για μελλοντικές προσθήκες.

Μετά το πέρας των κεφαλαίων παρατίθενται τα συμπεράσματα που προκύπτουν από την εμπειρία της δημιουργίας των διαδικασιών ενός τέτοιου συστήματος και του τελικού αποτελέσματος.

Κεφάλαιο 1: Το μοντέλο δεδομένων

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα πάρουμε μια ιδέα για το τι είναι μία διαδικτυακή εφαρμογή και στη συνέχεια θα αναλύσουμε το μοντέλο και το σχήμα του θέματος. Περισσότερα για την εφαρμογή και το σύστημα γενικότερα θα εξετάσουμε στο επόμενο κεφάλαιο.

1.2 Τι είναι μια διαδικτυακή εφαρμογή

Στις μέρες μας είναι δύσκολο να φανταστούμε τον προσωπικό μας υπολογιστή ως μεμονωμένη οντότητα όταν όλες σχεδόν οι ενασχολήσεις μας είναι στο διαδίκτυο. Στην πραγματικότητα οι υπολογιστές που δεν είναι μόνιμα στο διαδίκτυο σπανίζουν και συνήθως εξυπηρετούν άλλους σκοπούς όπως για παράδειγμα βιομηχανικούς.

Η μεγαλύτερη μερίδα των ηλεκτρονικών υπολογιστών σήμερα, τόσο οι προσωπικοί όσο και οι επαγγελματικής χρήσης χρησιμοποιούν το διαδίκτυο και τις υπηρεσίες που προσφέρει. Έτσι παρατηρούμε ότι και οι εφαρμογές μεταναστεύουν από το τοπικό περιβάλλον (desktop) στο διαδικτυακό. Για παράδειγμα πριν την κυριαρχία του διαδικτύου το 90% της χρήσης οικιακών υπολογιστών γίνονταν για επεξεργασία κειμένου με προγράμματα όπως MS Word ή για λογιστικά φύλλα με το MS Excel. Αντίστοιχα με την επικράτηση του www (world wide web) αυτές οι εφαρμογές υλοποιήθηκαν και διαδικτυακά και προσφέρουν πολλές δυνατότητες. Ακόμα υπήρξε και ανάπτυξη νέων εφαρμογών όπως των υπηρεσιών της Google όπως το Google Docs που προσφέρει παρόμοιες αλλά πιο απλουστευμένες λειτουργίες.

Οι δύο πιο βασικές και κύριες δυνατότητες των διαδικτυακών εφαρμογών είναι η ευκολία στην αποθήκευση δεδομένων καθώς και στην ευκολία χρήσης αφού είναι ανεξάρτητη από την πλατφόρμα πρόσβασης. Ας δούμε τα παρακάτω παραδείγματα για να καταλάβουμε την αξία αυτών των δυνατοτήτων αντίστοιχα.

Όταν ένα αρχείο αποθηκεύεται τοπικά στον υπολογιστή μας είμαστε υπεύθυνοι εμείς για την τήρηση εφεδρείας (backup). Όταν αποφασίσουμε να αποθηκεύσουμε το αποθηκεύσουμε στο "Cloud", όπως προτείνουν οι περισσότερες εφαρμογές, μεταφέρουμε την αποθήκευσή μας από τον υπολογιστή μας στο νέφος. Αυτό σημαίνει ότι μεταβιβάζουμε την ευθύνη του backup στην εταιρία που μας προσφέρει τη φιλοξενία.

Για τον τρόπο πρόσβασης το όφελος γίνεται προφανές όταν αναλογιστούμε μία εταιρία που χρησιμοποιεί κάποια προγράμματα για να χειρίζεται το απόθεμα της αποθήκης της, να την ενημερώνει για την έγκαιρη αγορά πρώτων υλών κλπ. Αυτές οι εφαρμογές θα έχουν συνήθως σχεδιαστεί για να εκτελούνται είτε σε έναν συγκεκριμένο υπολογιστή είτε θα είναι εξαρτώμενες από το λειτουργικό σύστημα π.χ. Windows. Μια εφαρμογή στο διαδίκτυο όμως προϋποθέτει απλώς την χρήση ενός φυλλομετρητή ιστού (browser). Φυσικά σε αυτήν την περίπτωση πρέπει να εφαρμόζεται και κάποιο πρωτόκολλο ασφαλείας για τον έλεγχο της πρόσβασης από μη εξουσιοδοτημένους χρήστες.

Κατανοώντας τι είναι και ποια τα οφέλη μιας διαδικτυακής εφαρμογής μπορούμε να πούμε ότι στο μέλλον ο αριθμός θα αυξηθεί όσο αυξάνεται και η ζήτηση.

1.3 Το μοντέλο για το ticket-manager

Ένα μοντέλο δεδομένων είναι ένα υψηλού επιπέδου σχέδιο, ή καλύτερα μια αφαιρετική αναπαράσταση μιας βάσης δεδομένων που καθορίζει τι θα παρουσιαστεί σε ένα σχήμα (schema). Το σχήμα θα λέγαμε ότι είναι το προσχέδιο της βάσης δεδομένων. Είναι αυτό που περιγράφει τι στήλες

θα υπάρχουν στην βάση και ποιος θα είναι ο τύπος των δεδομένων καθεμιάς από αυτές. Π.χ. σε μια βάση με πίνακα με την ονομασία υπάλληλος θα υπάρχει μία στήλη «αναγνωριστικό Υπαλλήλου» (Id) η οποία θα είναι τύπου κειμένου και θα έχει μήκος 10 χαρακτήρες, ενώ η στήλη «όνομα Υπαλλήλου» θα είναι τύπου κειμένου με μήκος 45 χαρακτήρες.

Πιο συγκεκριμένα το μοντέλο δεδομένων μπορεί π.χ. να είναι ένα σχεσιακό μοντέλο όπου τα δεδομένα οργανώνονται σε πίνακες, ενώ το σχήμα για το μοντέλο αυτό θα ήταν ένα σύνολο γνωρισμάτων και οι αντιστοιχίσεις τους.

Στην εργασία το μοντέλο που χρησιμοποιήθηκε περιλάμβανε περισσότερες από 15 οντότητες από τις οποίες θα αναπτυχθούν αναλυτικά μόνον όσες χρησιμοποιήθηκαν στην υλοποίηση, ενώ για τις υπόλοιπες θα γίνει μια συνοπτική αναφορά.

1.3.1 Πίνακας Χρήστες (users)

Στην οντότητα αυτή περιγράφεται ο φυσικός χρήστης της εφαρμογής. Αυτή η οντότητα είναι εύκολα κατανοητή και γι' αυτό επιλέχθηκε σαν εισαγωγική. Παρόλα αυτά θα εξετασθούν και πιο ειδικά γνωρίσματα και θα αναλυθεί η σημασία τους.

Εφόσον η οντότητα που περιγράφεται αναφέρεται σε φυσικό πρόσωπο που χρησιμοποιεί μια εφαρμογή, όπως είναι αναμενόμενο έχει όνομα χρήστη για την εφαρμογή, κατακερματισμένο¹ κωδικό πρόσβασης, όνομα και επώνυμο, διεύθυνση ηλεκτρονικού ταχυδρομείου και τηλεφωνικό αριθμό.

Τα χαρακτηριστικά «ομαλοποιημένο όνομα χρήστη» και «ομαλοποιημένο email» (NormalizedUserName, NormalizedEmail) είναι το όνομα χρήστη και το email που ο ίδιος εισήγαγε κατά την εγγραφή του, μετά από ομαλοποίηση, δηλαδή μετατροπή των χαρακτήρων σε πεζούς. Χρησιμοποιούνται για να αποφεύγεται η πολλαπλή εγγραφή χρήστη με την ίδια σειρά χαρακτήρων που διαφέρουν απλώς σε κεφαλαία και πεζά γράμματα.

Κάποια ακόμα χαρακτηριστικά ασφαλείας είναι το SecurityStamp όπου είναι μοναδικό και αλλάζει μόνο εάν συμβεί κάποια αλλαγή π.χ. στον κωδικό πρόσβασης. Το EmailConfirmed και το PhoneNumberConfirmed δείχνουν αν ο χρήστης επιβεβαίωσε το λογαριασμό του μέσω του email ή/και του κινητού τηλεφώνου του, ενώ το TwoFactorEnabled επιβεβαιώνει τη χρήση της μεθόδου διπλής αυθεντικοποίησης π.χ. μέσω εφαρμογής για κινητό. Η παράμετρος ConcurrencyStamp είναι για την αποφυγή λάθους στην παράλληλη επεξεργασία της εγγραφής από τους διαχειριστές του συστήματος.

Τα LockoutEnabled και LockoutEnd δείχνουν αν ο χρήστης έχει αποκλειστεί από την είσοδο στο σύστημα και πότε τελειώνει (ημερομηνία και ώρα) ο αποκλεισμός αντίστοιχα. Το AccessFailedCount μετρά τις αποτυχημένες προσπάθειες σύνδεσης έως ότου φτάσει στο όριο και συνεπώς στον αποκλεισμό του χρήστη από τη σύνδεση στο σύστημα.

1.3.2 Πίνακας Τοποθεσία (venue)

Όπως και στον πίνακα χρήστες έτσι και στον πίνακα τοποθεσία συναντάται το μοναδικό αναγνωριστικό (Id) και το όνομα. Υπάρχει φυσικά η διεύθυνση, ο ταχυδρομικός κώδικας και τηλέφωνο επικοινωνίας. Το «αναγνωριστικό χρήστη» (UserId) μας δείχνει σε ποιον χρήστη ανήκει και είναι σχετικό² με το μοναδικό χαρακτηριστικό Id στον πίνακα χρήστες. Το ImgUrl δείχνει τη διαδρομή για το αρχείο εικόνας π.χ. images/image.jpg, ενώ το CityId είναι σχετικό με τον πίνακα πόλης στην οποία βρίσκεται η τοποθεσία.

¹ Ο κωδικός πρόσβασης ενός χρήστη δεν μπορεί να είναι ορατός στο σύστημα οπότε πριν την αποθήκευση χρησιμοποιείται κρυπτογραφική συνάρτηση κατακερματισμού.

² Βλέπε κεφ. 1.5.2 παρακάτω.

1.3.3 Πίνακας Πόλη (city)

Η πόλη περιγράφεται απλώς από το μοναδικό αναγνωριστικό (Id) και το όνομά της.

1.3.4 Πίνακας Υποπεριοχή (subarea)

Η υποπεριοχή περιγράφεται από το Id, το όνομα της, και μια σύντομη περιγραφή π.χ. από που γίνεται η πρόσβαση σε σχέση με την τοποθεσία. Έχει πλάτος και μήκος και τρία χαρακτηριστικά επάνω, αριστερά και προσανατολισμός, για χρήση ως συντεταγμένες στο επίπεδο. Τελευταίο αλλά σημαντικό είναι το μοναδικό χαρακτηριστικό τοποθεσίας (VenueId) με το οποίο αντιστοιχίζεται η υποπεριοχή σε μια κύρια τοποθεσία.

1.3.5 Πίνακας Θέση (seat)

Στον πίνακα αυτόν όπως και στους προηγούμενους υπάρχει το Id της θέσης και τα χαρακτηριστικά X και Y που αναπαριστούν γεωγραφικά σημεία στο επίπεδο. Ακόμα υπάρχει η ονομασία της π.χ. αναδιπλούμενη καρέκλα no1, η στήλη «διαθέσιμη» που χρησιμοποιείται ως δείκτης διαθεσιμότητας και το SubAreaId που αντιστοιχεί στην υποπεριοχή που βρίσκεται η θέση.

1.3.6 Πίνακας Κράτηση (reservation)

Στην οντότητα κράτηση έχουμε το χαρακτηριστικό Id, το Id του χρήστη που δείχνει ποιος χρήστης έχει κάνει την κράτηση και το Id της θέσης στην οποία έγινε η κράτηση. Ακολουθούν η ημερομηνία της και η διάρκεια, καθώς και το Id που αντιστοιχεί στο γεγονός – δρώμενο.

1.3.7 Πίνακας Γεγονός (event)

Στον πίνακα αυτόν αναφέρεται το δρώμενο που είναι να λάβει χώρα. Υπάρχει το χαρακτηριστικό διάκρισης Id, το όνομα και η περιγραφή του, όπως και η διαδρομή για το αρχείο εικόνας που μεταφόρτωσε ο εκδότης κατά τη δημιουργία (ImageUrl). Στη συνέχεια ορίζονται οι ημερομηνίες και ώρες έναρξης και λήξης καθώς και η τιμή για κράτηση θέσης. Η στήλη VenueId ορίζει την τοποθεσία που θα λάβει χώρα το δρώμενο. Οι τελευταίες EventTypeId και FamilyEventId δείχνουν αντίστοιχα τον τύπο του γεγονότος π.χ. σεμινάριο, συναυλία κ.α. και την οικογένεια του με την έννοια της ευρύτερης σκοπιάς όπως π.χ. μια παραλλαγή του έργου «Άμλετ» του Σαίξπηρ.

1.3.8 Πίνακες Τύπος και Οικογένεια Γεγονότος (EventType, FamilyEvent)

Στον πίνακα τύπος γεγονότος με τα χαρακτηριστικά Id και όνομα περιγράφεται ουσιαστικά η κατηγορία που ανήκει το δρώμενο π.χ. θεατρική παράσταση, σεμινάριο κ.α.

Για τον πίνακα οικογένεια γεγονότος, που περιγράφεται με τα ίδια χαρακτηριστικά όπως ο τύπος γεγονότος, αναφερθήκαμε ήδη στο υποκεφάλαιο 1.2.7.

1.4 Το υπόλοιπο μοντέλο

Παρά το γεγονός ότι οι παρακάτω οντότητες που περιγράφονται στο ολοκληρωμένο μοντέλο δεν υλοποιήθηκαν, καθώς ξεφεύγουν από τη σκοπιά την εργασίας αυτής, θεώρησα σημαντικό να παρατεθούν έστω επιγραμματικά. Ο λόγος είναι ότι στους πίνακες περιγράφονται εκτεταμένες έννοιες ασφαλείας που αντιπροσωπεύουν την σκοπιά της πραγματικής χρήσης μιας διαδικτυακής εφαρμογής. Ας μην ξεχνάμε ότι στην εποχή της πληροφορίας που διανύουμε όλες οι εφαρμογές που είναι να αναπτυχθούν πρέπει να ξεκινούν με γνώμονα την ασφάλεια.

1.4.1 Ρόλοι και Ρόλοι Χρήστη

Η οντότητα ρόλος περιγράφει το σύνολο των δικαιωμάτων πρόσβασης που έχει ο κάθε χρήστης στο σύστημα. Αναφορικά ένας χρήστης με ρόλο «διαχειριστής συστήματος» θα έχει πρόσβαση σε περισσότερες λειτουργίες της εφαρμογής σε σχέση με έναν χρήστη με τον ρόλο «συνδρομητής». Τέτοιες λειτουργίες μπορεί να είναι η προβολή και διαχείριση των εγγραφών των εγγεγραμμένων χρηστών.

Ο πίνακας ρόλος χρήστη είναι πίνακας αντιστοιχίας του χρήστη με τον ρόλο του στο σύστημα.

1.4.2 Ταυτοποίηση Χρήστη με Τεκμήριο (token)

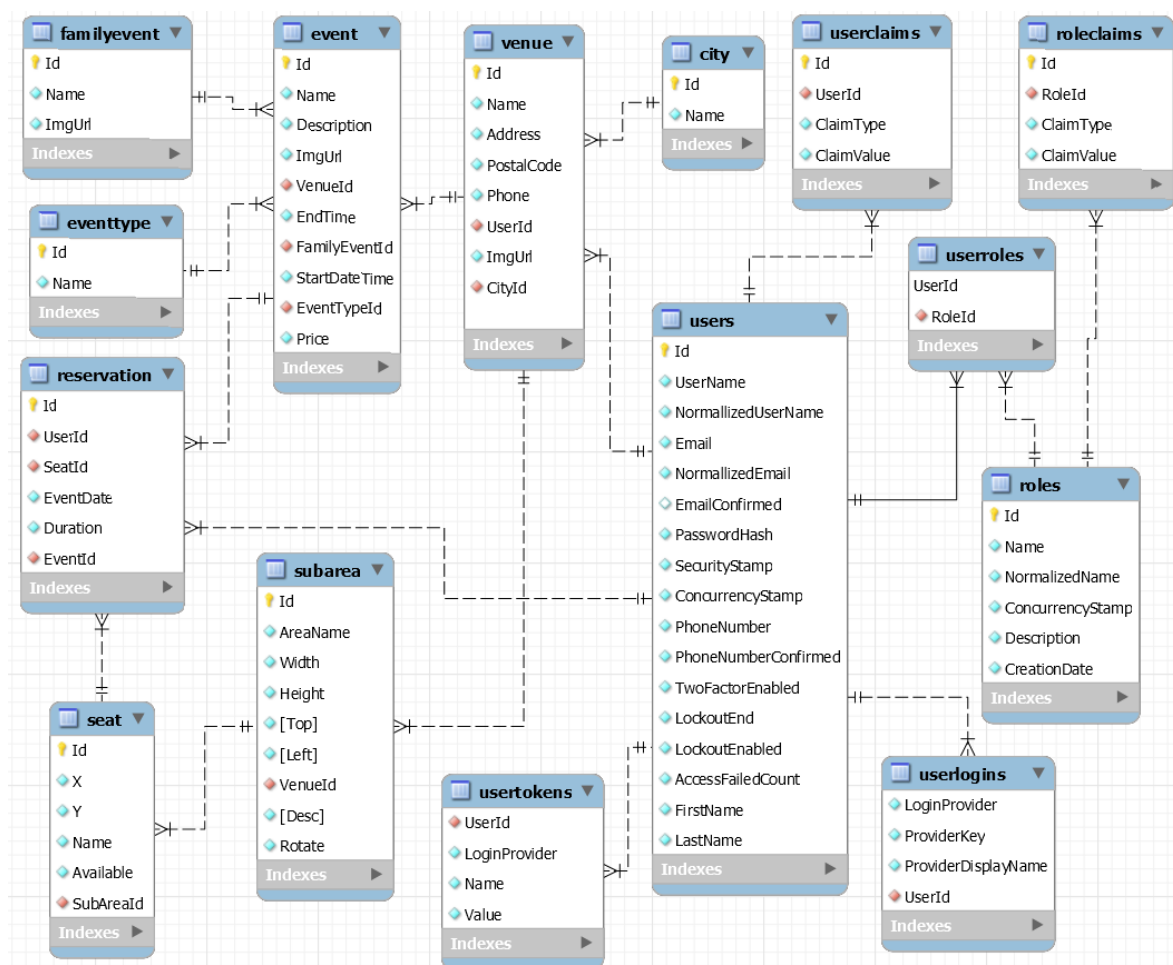
Κατά τη διαδικασία αυτή επισυνάπτεται στην κεφαλίδα ενός HTTP αιτήματος ένα τεκμήριο για αυθεντικοποίηση. Κατά την άφιξη του αιτήματος στον διακομιστή εξετάζεται. Όταν βρεθεί έγκυρο τεκμήριο η αίτηση επιτρέπεται, ενώ στην αντίθετη περίπτωση απορρίπτεται με απάντηση 401 Unauthorized. Συνήθως συναντώνται στα γενικά πλαίσια πρωτοκόλλων αυθεντικοποίησης OAuth 2.0 και OpenID Connect.

1.4.3 Συνδέσεις Χρήστη

Στον πίνακα αυτόν περιγράφονται τα ατομικά χαρακτηριστικά σύνδεσης για τον κάθε χρήστη όπως ποιος είναι ο πάροχος σύνδεσης π.χ. σύνδεση μέσω Facebook, κλειδιά διαπιστευτηρίων παρόχου κ.α.

1.4.4 Εξουσιοδότηση Βάση Αξιώσεων

Όταν δημιουργείται μια ταυτότητα, μπορεί να εκχωρηθούν μία ή περισσότερες αξιώσεις που εκδίδονται από μία αξιόπιστη πηγή. Μια αξίωση είναι ένα ζεύγος τιμής ονόματος που αντιπροσωπεύει τι είναι το υποκείμενο, και όχι τι μπορεί να κάνει το υποκείμενο. Για παράδειγμα, ενδέχεται να έχετε άδεια οδήγησης, που εκδίδεται από τοπική αρχή άδειας οδήγησης. Η άδεια οδήγησης σας φέρει την ημερομηνία γέννησής σας. Σε αυτήν την περίπτωση το όνομα της αξίωσης «ημερομηνία γέννησης», η αξία της αξίωσης θα είναι η ημερομηνία γέννησής σας, για παράδειγμα 8 Ιουνίου 1990 και ο εκδότης θα είναι η αρχή άδειας οδήγησης. Η εξουσιοδότηση βάσει αξιώσεων, στην απλούστερη της, ελέγχει την αξία μιας αξίωσης και επιτρέπει την πρόσβαση σε έναν πόρο βάσει αυτής της τιμής.



Εικόνα 1: Το διάγραμμα οντοτήτων συσχετίσεων

1.5 Το σχήμα

Όπως αναφέρθηκε το σχήμα είναι το προσχέδιο της πραγματικής βάσης δεδομένων που τελικά θα δημιουργηθεί και ακολουθεί τις οδηγίες του μοντέλου. Περιγράφει με λεπτομέρειες τους πίνακες και τα πεδία (στήλες) τους, τους τύπους των πεδίων τους και τις συσχετίσεις ανάμεσά τους. Τα πεδία, οι στήλες δηλαδή, των πινάκων αναφέρθηκαν ήδη παραπάνω οπότε θα αναφερθούμε κυρίως στους τύπους τους, και τις συσχετίσεις ανάμεσα στους πίνακες.

1.5.1 Τύποι πεδίων

Οι τύποι που χρησιμοποιούνται στο σχήμα μας είναι κατά κύριο λόγο οι εξής:

- Αριθμητικοί Τύποι (ακέραιος, δεκαδικός)
- Συμβολοσειρές Χαρακτήρων (varchar, nvarchar)
- Τύποι Χρονικής σήμανσης (timestamp)
- Bit (κατάσταση 0 ή 1)

Αριθμητικοί τύποι είναι οι ακέραιοι (integer), οι δεκαδικοί (decimal) και κινητής υποδιαστολής (float) και χρησιμοποιούνται για πεδία όπως το αναγνωριστικό Id, AccessFailedCount κ.α.

Οι συμβολοσειρές είναι σειρές χαρακτήρων που χρησιμοποιούνται σε στήλες όπως το όνομα χρήστη, η ονομασία ενός γεγονότος και η περιγραφή του. Άλλη χρήση που συναντάμε είναι η τήρηση της διαδρομής και του τύπου αρχείου μίας εικόνας που θέλουμε να αναπαράγουμε (ImageUrl). Οι τύποι τους είναι οι varchar και nvarchar όπου οι διαφορές είναι:

1. στο μέγεθος της μνήμης που δεσμεύεται στο σύστημα για την αποθήκευση της πληροφορίας και
2. το εύρος υποστηριζόμενων χαρακτήρων της πληροφορίας βάση του 1. Π.χ. κωδικοποίηση χαρακτήρων utf-8.

Οι τύποι χρονικής σήμανσης (timestamp) είναι η ακριβής ημερομηνία και ώρα π.χ. έναρξης ή τέλεσης ενός γεγονότος και έχει τη μορφή:

YYYY-MM-DD(κενό διάστημα)HH:ii:SS

Όπου Y = ψηφίο έτους, M = ψηφίο μήνα, D = ψηφίο ημέρας, H = ψηφίο ώρας, i = ψηφίο λεπτών και S = ψηφίο δευτερολέπτων. Η επανάληψη των στοιχείων δείχνει τον αριθμό των ψηφίων πχ έχουμε YYYY = 2019 τότε YY = 19. Η επανάληψη στην ώρα HH σημαίνει μορφή 24 ωρών.

Η τιμές bit παίρνουν κατάσταση 0 ή 1 και χρησιμοποιούνται για ένδειξη διαθεσιμότητας όπως για παράδειγμα: είναι ο χρήστης μπλοκαρισμένος; 0 όχι, 1 είναι μπλοκαρισμένος. Ο κύριος λόγος που προτιμώνται έναντι παραλλαγών είναι η ελάχιστη κατανάλωση μνήμης για την αναπαραγωγή αυτών των δύο καταστάσεων.

1.5.2 Ακεραιότητα: Κανόνες και Περιορισμοί

Οι περιορισμοί είναι ένα πολύ σημαντικό χαρακτηριστικό σε ένα σχεσιακό μοντέλο. Στην πραγματικότητα, το σχεσιακό μοντέλο υποστηρίζει την καλά ορισμένη θεωρία των περιορισμών σε χαρακτηριστικά ή πίνακες. Οι περιορισμοί είναι χρήσιμοι επειδή επιτρέπουν σε έναν σχεδιαστή να καθορίσει τη σημασιολογία των δεδομένων στη βάση δεδομένων. Οι περιορισμοί είναι οι κανόνες που αναγκάζουν τα DBMS³ να ελέγχουν ότι τα δεδομένα ικανοποιούν τη σημασιολογία.

Περιορισμοί ακεραιότητας είναι:

- Ακεραιότητα Οντότητας (Entity Integrity)

Με αυτούς διασφαλίζεται ότι κάθε πίνακας έχει ένα πρωτεύων κλειδί (Primary Key). Ένα πρωτεύων κλειδί είναι μοναδικό αναγνωριστικό που έχει αποδοθεί σε κάθε σειρά του πίνακα, όπως

³ Data Base Management Systems, Συστήματα Διαχείρισης Βάσεων Δεδομένων.

το πεδίο Id που εξετάσαμε, και δεν μπορεί να μην πάρει τιμή. Χρησιμοποιούμε για να μπορούμε να αναγνωρίσουμε την γραμμή στην οποία αναφερόμαστε.

- Ακεραιότητα Αναφοράς (Referential Integrity)

Η ακεραιότητα αναφοράς σημαίνει ότι ένα ξένο κλειδί (Foreign Key) πρέπει να έχει ένα αντίστοιχο πρωτεύον κλειδί. Αυτός ο περιορισμός καθορίζεται μεταξύ δύο πινάκων (γονέας και παιδί). Από αυτό συνεπάγεται ότι η στήλη που φιλοξενεί το ξένο κλειδί (παιδί) πρέπει να είναι του ίδιου τύπου δεδομένων με τη στήλη του πίνακα που φιλοξενεί το πρωτεύον κλειδί (γονέας). Αυτό σημαίνει ότι η αναφορά σε μια σειρά από έναν πίνακα σε έναν άλλο πίνακα πρέπει να είναι έγκυρη.

Ακόμα μπορούν να προστεθούν πρόσθετοι κανόνες ξένου κλειδιού κατά τον καθορισμό της ακεραιότητας αναφοράς, όπως τι πρέπει να γίνεται με τις θυγατρικές σειρές όταν η εγγραφή με το πρωτεύον κλειδί, μέρος του γονέα, διαγράφεται ή τροποποιείται (ενημερώνεται). Ένα παράδειγμα επιπλέον κανόνα είναι ο κανόνας ON DELETE CASCADE όπου ορίζεται πως αν η εγγραφή στον πίνακα γονέα διαγραφεί τότε θα διαγραφεί αυτόματα και η αντιστοιχιζόμενη εγγραφή στον πίνακα παιδί.

1.6 Επίλογος

Στο κεφάλαιο αυτό εξετάστηκαν αναλυτικά το μοντέλο και το σχήμα που αναπαράγεται στην εργασία αυτή και περιγράψαμε τις οντότητες που την απαρτίζουν με τους κανόνες που την διέπουν.

Στο επόμενο κεφάλαιο θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

Κεφάλαιο 2: Οι τεχνολογίες υλοποίησης

2.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια γενική εισαγωγή στις τεχνολογίες που απαιτούνται για την δημιουργία ενός διαδικτυακού τόπου και τα χαρακτηριστικά τους. Στη συνέχεια θα αναλυθούν τα μέρη του συστήματος και οι επιμέρους τεχνολογίες που το απαρτίζουν.

Πιο συγκεκριμένα θα εξετάσουμε τις γλώσσες προγραμματισμού για τα μέρη της προβολής της εφαρμογής, της αλληλεπίδρασης με το χρήστη και τις τεχνολογίες που στηρίζουν αυτές τις λειτουργίες. Χαρακτηριστικά θα αναφερθούμε στις γλώσσες σήμανσης (Markup), φύλλων ύφους (StyleSheet) και σεναρίου (Scripting), γλώσσα ερωτήσεων βάσης δεδομένων (SQL) και στις τεχνολογίες του διακομιστή (server) και του συστήματος διαχείρισης βάσης δεδομένων (DBMS).

2.2 Γλώσσα Σήμανσης Υπερκειμένου HTML

Με τον όρο γλώσσα σήμανσης στην ορολογία των υπολογιστών αναφερόμαστε σε ένα σύστημα σχολιασμού εγγράφων με τέτοιο τρόπο ώστε να είναι συντακτικά διακριτός ο σχολιασμός με το απλό κείμενο. Αυτό σημαίνει πως όταν το έγγραφο αυτό βρίσκεται στην επεξεργασία για προβολή, η γλώσσα σήμανσης αποκρύπτεται και δεν εμφανίζεται στην τελική προβολή, αντιθέτως συμβάλει μόνον στην μορφοποίηση του κειμένου. Μια τέτοια γλώσσα τυπικά περιλαμβάνει διορθώσεις περιεχομένου όπως ορθογραφία, σημεία στίξης κ.α. καθώς και τυπογραφικές οδηγίες όπως ο τρόπος που δηλώνεται μια κεφαλίδα ή πως τονίζουμε (bold) μία λέξη ή φράση.

Η γλώσσα σήμανσης που χρησιμοποιείται στο σύστημα είναι η κύρια γλώσσα σήμανσης των ιστοσελίδων και ονομάζεται HTML από το HyperText Markup Language, Γλώσσα Σήμανσης Υπερκειμένου. Είναι δηλαδή η γλώσσα που έχει σχεδιαστεί για να συντάσσονται έγγραφα που θα προβληθούν σε έναν περιηγητή ιστού (web browser)⁴ και μπορεί να ενισχυθεί με τεχνολογίες όπως διαδοχικά φύλλα ύφους (Cascading Style Sheets, CSS) και με γλώσσες σεναρίου όπως η JavaScript.

Στην πραγματικότητα οι περιηγητές δέχονται ένα έγγραφο HTML από έναν διακομιστή ή από μία τοπική διαδρομή αποθήκευσης και το μετατρέπουν σε πολυμεσική ιστοσελίδα. Η HTML περιγράφει τη δομή της ιστοσελίδας αυτής σημασιολογικά και περιλαμβάνει τις οδηγίες εμφάνισης του εγγράφου.

Τα στοιχεία της είναι τα δομικά συστατικά των σελίδων HTML. Μπορούμε να δομήσουμε εικόνες και άλλα αντικείμενα όπως διαδραστικές φόρμες π.χ. φόρμα επικοινωνίας. Συνεπώς η HTML παρέχει τα μέσα για τη δημιουργία δομημένων εγγράφων υποδηλώνοντας τη δομική σημασιολογία για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, πεδία εισαγωγής κειμένου και άλλα στοιχεία. Τα στοιχεία αυτά ονομάζονται ετικέτες (tags) και περιβάλλονται με γωνιακές αγκύλες, για παράδειγμα οι ετικέτες εικόνας και πεδίου εισαγωγής <input />. Άλλες ετικέτες, που συνήθως λειτουργούν ανά ζεύγη όπως η παράγραφος με αρχή <p> και τέλος </p> παρέχουν πληροφορίες για το κείμενο που περιέχουν και μπορούν να συνδυαστούν με άλλες ετικέτες υποστοιχεία.

Όλες οι σελίδες HTML πρέπει να ξεκινούν με την δήλωση του τύπου του εγγράφου: <!DOCTYPE html>, η οποία ορίζει και τον τρόπο εμφάνισης στο πρόγραμμα περιήγησης (browser). Το ίδιο το έγγραφο αρχίζει με <html> και τελειώνει με </html>. Ακολουθεί το ζεύγος <head> και </head> το οποίο μπορεί να περιλαμβάνει έναν τίτλο για το έγγραφο, διάφορα σενάρια, πληροφορίες δεδομένων (metadata) και πλήθος άλλων στοιχείων. Ο τίτλος που θα εμφανίζεται στο παράθυρο του browser βρίσκεται στις ετικέτες <title> και </title> ενώ τα μεταδεδομένα με το στοιχείο <meta>.

⁴ Αναφέρεται και ως φυλλομετρητής ιστού ή απλά περιηγητής.

Το ορατό τμήμα του εγγράφου περιλαμβάνεται μέσα στις ετικέτες `<body>` και `</body>`. Οι κεφαλίδες ανά επίπεδο ορίζονται με τις ετικέτες `<h1>` έως `<h6>` με πιο σημαντικό τίτλο την `<h1>` και `<h6>` ο λιγότερο σημαντικός. Άλλα σημαντικά στοιχεία είναι τα κουμπιά που ορίζονται με την ετικέτα `<button>` όπως και τα σχόλια τα οποία ξεκινούν με `<!--` και τελειώνουν με `-->`. Τα σχόλια βοηθούν στην τεκμηρίωση του πηγαίου κώδικα HTML και δεν εμφανίζονται από τον περιηγητή.

Ένα βασικό στοιχείο που πρέπει να εξεταστεί ξεχωριστά είναι οι φόρμες συμπλήρωσης στοιχείων. Ξεκινούν με `<form>` και τελειώνουν με `</form>` και χρησιμοποιούνται για να συλλέξουν πληροφορίες από τον χρήστη. Οι πληροφορίες αυτές μπορεί να έχουν την μορφή πεδίων κειμένου (`input type="text"`), πλαισίων ελέγχου (`checkboxes`), κουμπιών επιλογής (`radio buttons`), κουμπιών υποβολής (`submit buttons`) κ.α. και αφορούν προσωπικά στοιχεία, επιλογές του χρήστη κ.λπ.. Κύριο συστατικό της φόρμας είναι το στοιχείο `<input>` και μπορεί να εμφανίζεται με διάφορους τρόπους όπως κειμένου που είδαμε παραπάνω ή ως ένα κουμπί υποβολής των στοιχείων της φόρμας `<input type="submit">` σε έναν χειριστή φόρμας. Ο χειριστής φόρμας είναι συνήθως μία σελίδα στον διακομιστή με ενέργειες για την επεξεργασία των δεδομένων και ορίζεται με το χαρακτηριστικό `<form action="ιστοσελίδα">`. Συνήθως μετά από αυτό ακολουθεί και το χαρακτηριστικό `method` που καθορίζει την μέθοδο HTTP ως GET ή POST κατά την υποβολή των στοιχείων της φόρμας. Η προεπιλεγμένη μέθοδος είναι η GET κατά την οποία τα δεδομένα της φόρμας που θα υποβληθούν θα εμφανιστούν στο πεδίο διεύθυνσης της σελίδας ως ζεύγος ονόματος τιμής. Αντίθετα με την μέθοδο POST δεν εμφανίζεται κανένα δεδομένο και αυτός είναι ο λόγος που ενδείκνυται για φόρμες δεδομένων με προσωπικές ή ευαίσθητες πληροφορίες.

Ακόμα είναι σημαντικό να αναφέρουμε ότι όλα τα στοιχεία της HTML μπορούν να έχουν ιδιότητες, όπως ο τύπος που είδαμε, το μέγεθος και το στυλ εμφάνισης. Το στυλ μπορεί να είναι η γραμματοσειρά, το μέγεθος των γραμμάτων η μορφοποίησή τους, το χρώμα των γραμμάτων κ.λπ..

Στην εργασία οι περισσότερες σελίδες που εμφανίζονται στον χρήστη είναι δομημένες με HTML εκτός αυτών που εφαρμόζουν πιο σύνθετες λειτουργίες. Πιο συγκεκριμένα αυτές βρίσκονται στο γονικό κατάλογο⁵ του διακομιστή και είναι αλφαβητικά οι εξής:

- checkout.php
- footer.php
- header.php
- index.php
- login.php
- new_event.php
- profile.php
- select.php
- signup.php

Από αυτές διακρίνονται οι `header` (κεφαλίδα) και `footer` (υποσέλιδο) οι οποίες συμπεριλαμβάνονται στις υπόλοιπες σελίδες εμφάνισης στην αρχή και στο τέλος αντίστοιχα. Ο λόγος που συμβαίνει αυτό είναι ότι στις δύο αυτές σελίδες έχουν δομηθεί στοιχεία, όπως η κεφαλίδα και το μενού πλοήγησης με την κατάλληλη στοίχιση, και μεταφέρουν ουσιαστικά τις ιδιότητές τους στις σελίδες που τις επικαλούνται. Ακόμα, όπως θα δούμε παρακάτω στην ενότητα 2.5 περιλαμβάνουν και κάποιες αναφορές σε εξωτερικά αρχεία και κοινές λειτουργίες που δεν χρειάζεται να επαναλαμβάνονται σε κάθε σελίδα ξεχωριστά.

2.3 Διαδοχικά Φύλλα Ύφους CSS

Τα διαδοχικά φύλλα ύφους, γνωστά ως `Cascading Style Sheets`, είναι η γλώσσα προγραμματισμού που χρησιμοποιείται για την παρουσίαση ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης όπως

⁵ Βλέπε διακομιστής παρακάτω.

Οι τεχνολογίες υλοποίησης

η HTML. Το CSS ανήκει και αυτό στις ακρογωνιαίες λίθους των τεχνολογιών του διαδικτύου μαζί με την HTML και την JavaScript.

Ο σκοπός του είναι η διαφοροποίηση του περιεχομένου από την παρουσίασή του, συμπεριλαμβανομένων των χρωμάτων, των γραμματοσειρών και της διάταξης. Αυτός ο διαχωρισμός διευκολύνει την πρόσβαση στο περιεχόμενο και παρέχει έλεγχο και ευελιξία στον προσδιορισμό των χαρακτηριστικών παρουσίασης. Επιτρέπει δηλαδή σε πολλές ιστοσελίδες να μοιράζονται την μορφοποίηση που ορίζεται στο σχετικό αρχείο .css, και μειώνει την πολυπλοκότητα και την επανάληψη κώδικα δομικού περιεχομένου.

Η σύνταξή του είναι απλή και περιλαμβάνει κάποιες αγγλικές λέξεις κλειδιά για διάφορες ιδιότητες χαρακτηριστικών. Για παράδειγμα η έκφραση: `div .myClass {color: red;}` δηλώνει πως τα περιεχόμενα ενός διαχωριστή περιεχομένου `div` που ανήκει στην κλάση `myClass` θα έχουν χρώμα κόκκινο. Για να δούμε πιο αναλυτικά τα πλεονεκτήματά του ας δούμε πώς ήταν η σύνταξη μορφοποίησης πριν το CSS.

Τα χαρακτηριστικά παρουσίασης συντάσσονταν μέσα στον κώδικα της γλώσσας σήμανσης, π.χ. η φράση `<h1> Κεφάλαιο 1 </h1>` δηλώνει μία κεφαλίδα πρώτου επιπέδου (`h1`) με χρώμα γραμματοσειράς κόκκινο (`font color="red"`). Χρησιμοποιώντας το CSS στην ίδια γραμμή (`inline`) η φράση γίνεται: `<h1 style="color: red;"> Κεφάλαιο 1 </h1>`. Η αξία του όμως ανεβαίνει όταν η δήλωση γίνει εσωτερικά σε άλλο σημείο του εγγράφου, ή ακόμα καλύτερα σε ένα άλλο αρχείο .css. Η φράση εκεί γίνεται ακόμα πιο απλή.

Εσωτερικά:

```
<style>
h1 {
  color: red;
}
</style>
```

Εξωτερικά:

```
h1 {
  color: red;
}
```

Εδώ βλέπουμε στην σύνταξη της γλώσσας ότι πριν από κάθε μπλοκ αναφέρεται το στοιχείο HTML που θα μορφοποιηθεί και το μπλοκ οριοθετείται με άγκιστρα `{ }`. Μέσα στα άγκιστρα δίνεται κάθε μια ιδιότητα ξεχωριστά, ακολουθεί `άνω – κάτω τελεία` και η τιμή της. Η κάθε φράση τελειώνει με ελληνικό ερωτηματικό. Μάλιστα κατά τον τρίτο τρόπο (εξωτερική δήλωση) όλες οι κεφαλίδες τύπου `1` θα πάρουν χρώμα γραμματοσειράς κόκκινο χωρίς να χρειάζεται να συντάξουμε καμία άλλη δήλωση στο έγγραφο markup παρά την φράση μέσα στην κεφαλίδα `<head> </head>` του HTML εγγράφου:

```
<link href="διαδρομή/για/το/αρχείο.css" rel="stylesheet" type="text/css">
```

Έτσι εάν στη συνέχεια αποφασίσουμε, για παράδειγμα, να αλλάξουμε το χρώμα της γραμματοσειράς σε μαύρο, το μόνο που έχουμε να κάνουμε είναι να ανατρέξουμε στο αρχείο.css και να αλλάξουμε το χρώμα από `"red"` σε `"black"`. Ακόμα, τα HTML στοιχεία για διαμόρφωση μπορούν να χαρακτηρίζονται από ένα μοναδικό χαρακτηριστικό (`id`) ή να ανήκουν σε κάποια κλάση, όπως είδαμε. Στον κώδικα CSS αναφέρονται `#id` και `.class` ενώ κάθε στοιχείο μπορεί να ανήκει σε παραπάνω από μια κλάση π.χ. `<p class="class1 class2">`. Επίσης δίνεται και η επιλογή ομαδοποίησης ετικετών βάζοντας κόμμα ανάμεσά τους π.χ. `h1, h2 { ιδιότητα: τιμή; }`.

Άλλα χαρακτηριστικά του CSS εκτός των βασικών που είδαμε είναι η ιδιότητα `padding` που ορίζει το διάστημα μεταξύ του αντικειμένου και του περιγράμματος, και η ιδιότητα `margin` που ορίζει το

Οι τεχνολογίες υλοποίησης

διάστημα μεταξύ αντικειμένου έξω από το όριο. Οι ιδιότητες αυτές καθορίζονται είτε ξεχωριστά για τις τέσσερις κατευθύνσεις (padding-top, padding-bottom, padding-left, padding right) είτε με τρεις πάνω + κάτω, αριστερά, δεξιά, είτε ανά δύο πάνω + κάτω, δεξιά + αριστερά, είτε με μια που αφορά όλες τις κατευθύνσεις.

Για την εργασία χρησιμοποιείται κυρίως δήλωση σε εξωτερικά αρχεία αλλά σε ιδιαίτερες περιπτώσεις υπάρχουν και εσωτερικές δηλώσεις. Τα δύο εξωτερικά αρχεία είναι το `eventsStyle.css` και το `w3.css`. Το πρώτο αρχείο έχει συνταχθεί στο χέρι και είναι αυτό που περιλαμβάνει τις οδηγίες για προσαρμοσμένα χαρακτηριστικά όπως ένα μήνυμα λάθους ή επιτυχίας, τη μορφοποίηση κάποιων διαχωριστών περιεχομένου (div) κ.α. Το δεύτερο αρχείο παρέχεται από τον οργανισμό W3Schools⁶ και είναι ουσιαστικά ένα πλαίσιο CSS το οποίο είναι δωρεάν χωρίς να απαιτεί κάποια άδεια χρήσης. Σε αυτό περιλαμβάνονται πολλές κλάσεις μορφοποίησης οι οποίες μπορούν να αποδοθούν σε οποιαδήποτε ετικέτα μπορεί να υποστηρίξει μορφοποίηση, όπως κουμπιά, κεφαλίδες, παράγραφοι και πολλά άλλα.

```
1  * {
2    box-sizing: border-box;
3  }
4  body {
5    font-family: Arial;
6    padding: 10px;
7    background: #f1f1f1;
8  }
9
10 form.userInput {
11   display: inline-block;
12 }
13 p.errorMsg {
14   font-size: 16px;
15   color: red;
16 }
17 p.successMsg {
18   font-size: 16px;
19   color: green;
20 }
21 a.title {
22   color: inherit;
23   text-decoration: inherit;
24 }
25 span.pretext {
26   color: #838383;
27 }
28 input[type=text], input[type=password], input[type=datetime-local], input[type="date"], select
29 {
30   padding: 12px 18px;
31   margin: 5px 0;
32   display: inline-block;
33   border: 1px solid #ccc;
34   border-radius: 4px;
35   box-sizing: border-box;
36   font-size: 16px;
37   transition: 0.3s;
38   min-width: 265px;
39 }
40 input[type=file]
41 {
42   padding: 12px 18px;
43   margin: 5px 0;
44   display: inline-block;
45   border: 1px solid #ccc;
46   border-radius: 4px;
47   box-sizing: border-box;
48   font-size: 12px;
49 }
```

Εικόνα 2: Η αρχή του `eventStyle.css`

⁶ <https://www.w3schools.com/>

```

39 .w3-btn:hover{box-shadow:0 8px 16px 0 rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19)}
40 .w3-btn,.w3-button,.webkit-touch-callout:none,.webkit-user-select:none,.khtml-user-select:none,.moz-user-select:none,.ms-user-select:none,.user-select:none}
41 .w3-disabled,.w3-btn.disabled,.w3-button.disabled{cursor:not-allowed;opacity:0.3;.w3-disabled *,.disabled *(pointer-events:none)}
42 .w3-btn.w3-disabled:hover,.w3-btn.disabled:hover{box-shadow:none}
43 .w3-badge,.w3-tag{background-color:#000;color:#fff;display:inline-block;padding-left:8px;padding-right:8px;text-align:center;.w3-badge{border-radius:50%}}
44 .w3-ul{list-style-type:none;padding:0;margin:0}.w3-ul li{padding:8px 16px;border-bottom:1px solid #ddd}.w3-ul li:last-child{border-bottom:none}
45 .w3-tooltip,.w3-display-container{position:relative}.w3-tooltip .w3-text{display:none}.w3-tooltip:hover .w3-text{display:inline-block}
46 .w3-ripple:active{opacity:0.5}.w3-ripple{transition:opacity 0s}
47 .w3-input{padding:8px;display:block;border:none;border-bottom:1px solid #ccc;width:100%}
48 .w3-select{padding:9px 0;width:100%;border:none;border-bottom:1px solid #ccc}
49 .w3-dropdown-click,.w3-dropdown-hover{position:relative;display:inline-block;cursor:pointer}
50 .w3-dropdown-hover:hover .w3-dropdown-content{display:block}
51 .w3-dropdown-hover:first-child,.w3-dropdown-click:hover{background-color:#ccc;color:#000}
52 .w3-dropdown-click > .w3-button:first-child,.w3-dropdown-hover > .w3-button:first-child{background-color:#ccc;color:#000}
53 .w3-dropdown-content{cursor:auto;color:#000;background-color:#fff;display:none;position:absolute;min-width:160px;margin:0;padding:0;z-index:1}
54 .w3-check,.w3-radio{width:24px;height:24px;position:relative;top:6px}
55 .w3-sidebar{height:100%;width:200px;background-color:#fff;position:fixed!important;z-index:1;overflow:auto}
56 .w3-bar-block .w3-dropdown-hover,.w3-bar-block .w3-dropdown-click{width:100%}
57 .w3-bar-block .w3-dropdown-hover .w3-dropdown-content,.w3-bar-block .w3-dropdown-click .w3-dropdown-content{min-width:100%}
58 .w3-bar-block .w3-dropdown-hover .w3-button,.w3-bar-block .w3-dropdown-click .w3-button{width:100%;text-align:left;padding:8px 16px}
59 .w3-main #main{transition:margin-left 4s}
60 .w3-modal{font-size:24px;display:none;padding-top:100px;position:fixed;left:0;top:0;width:100%;height:100%;overflow:auto;background-color:rgb(0,0,0);background-color:rgba(0,0,0,0.4)}
61 .w3-modal content{margin:auto;background-color:#fff;position:relative;padding:8px 0;width:600px}
62 .w3-bar{width:100%;overflow:hidden}.w3-center .w3-bar{display:inline-block;width:auto}
63 .w3-bar .w3-bar-item{padding:8px 16px;float:left;width:auto;border:none;display:block;outline:0}
64 .w3-bar .w3-dropdown-hover,.w3-bar .w3-dropdown-click{position:static;float:left}
65 .w3-bar .w3-button{white-space:normal}
66 .w3-bar-block .w3-bar-item{width:100%;display:block;padding:8px 16px;text-align:left;border:none;white-space:normal;float:none;outline:0}
67 .w3-bar-block .w3-center .w3-bar-item{text-align:center}.w3-block{display:block;width:100%}
68 .w3-responsive{display:block;overflow-x:auto}
69 .w3-container:after,.w3-container:before,.w3-panel:after,.w3-panel:before,.w3-row:after,.w3-row:before,.w3-row-padding:after,.w3-row-padding:before,
70 .w3-cell-row:before,.w3-cell-row:after,.w3-clear:after,.w3-clear:before,.w3-bar:before,.w3-bar:after{content:"";display:table;clear:both}
71 .w3-col,.w3-half,.w3-third,.w3-twothird,.w3-quarter,.w3-quarter{float:left;width:100%}
72 .w3-col.s1{width:8.33333%}.w3-col.s2{width:16.66666%}.w3-col.s3{width:24.99999%}.w3-col.s4{width:33.33333%}
73 .w3-col.s5{width:41.66666%}.w3-col.s6{width:49.99999%}.w3-col.s7{width:58.33333%}.w3-col.s8{width:66.66666%}
74 .w3-col.s9{width:74.99999%}.w3-col.s10{width:83.33333%}.w3-col.s11{width:91.66666%}.w3-col.s12{width:99.99999%}
75 @media (min-width:601px){.w3-col.m1{width:8.33333%}.w3-col.m2{width:16.66666%}.w3-col.m3,.w3-quarter{width:24.99999%}.w3-col.m4,.w3-third{width:33.33333%}
76 .w3-col.m5{width:41.66666%}.w3-col.m6,.w3-half{width:49.99999%}.w3-col.m7{width:58.33333%}.w3-col.m8,.w3-twothird{width:66.66666%}
77 .w3-col.m9,.w3-threequarter{width:74.99999%}.w3-col.m10{width:83.33333%}.w3-col.m11{width:91.66666%}.w3-col.m12{width:99.99999%}}
78 @media (min-width:993px){.w3-col.l1{width:8.33333%}.w3-col.l2{width:16.66666%}.w3-col.l3{width:24.99999%}.w3-col.l4{width:33.33333%}}

```

Εικόνα 3: Μέρος του w3.css

2.4 Γλώσσες σεναρίου JavaScript και PHP

Μία γλώσσα σεναρίου είναι μια γλώσσα προγραμματισμού που αυτοματοποιεί την εκτέλεση κάποιων εργασιών, σε ένα ειδικό περιβάλλον εκτέλεσης. Οι γλώσσες αυτές συχνά ερμηνεύονται⁷ (interpreted) αντί να μεταγλωττιστούν⁸ (compiled) για λόγους απόδοσης. Τέτοια περιβάλλοντα μπορεί να είναι: εφαρμογές λογισμικού, ιστοσελίδες, όταν γίνεται χρήση από κάποιο κέλυφος (shell) ή από ένα λειτουργικό σύστημα (operating system, OS), η χρήση από ενσωματωμένα συστήματα και από πολλά βιντεοπαιχνίδια.

2.4.1 Η γλώσσα JavaScript

Η συχνά αναφερόμενη και ως JS είναι μία γλώσσα προγραμματισμού σεναρίου που ακολουθεί το πρότυπο ECMAScript και είναι και αυτή μέλος του πυρήνα των γλωσσών προγραμματισμού του διαδικτύου. Είναι η γλώσσα που επιτρέπει την διαδραστικότητα στις ιστοσελίδες και είναι απαραίτητο κομμάτι των διαδικτυακών εφαρμογών. Η πλειοψηφία των ιστοσελίδων χρησιμοποιούν για να ρυθμίσουν την συμπεριφορά της σελίδας στην μεριά του πελάτη (client side), ενώ όλοι οι κύριοι web browsers διαθέτουν ειδικό μηχανισμό για την εκτέλεσή της.

Η λειτουργία της υποστηρίζει διάφορα στυλ προγραμματισμού όπως αντικειμενοστραφής, οδηγούμενο από τα γεγονότα, συναρτησιακό και προστακτικό. Η γλώσσα πάντως καθαυτή δεν διαθέτει λειτουργίες όπως εισόδου/εξόδου (I/O), αποθήκευσης ή απεικόνισης καθώς το περιβάλλον φιλοξενίας είναι αυτό που παρέχει τις λειτουργίες αυτές.

Συντακτικά ο κώδικας JavaScript μιας σελίδας περικλείεται από τις ετικέτες <script> και </script> ή σε παλαιότερες εκδόσεις <script type="text/javascript"> </script>. Βασικά στοιχεία της είναι οι συναρτήσεις (functions) που αποτελούν μπλοκ κώδικα και εκτελούνται όταν κληθούν από το πρόγραμμα. Συντάσσονται με την λέξη κλειδί function ενώ ακολουθεί η ονομασία και παρένθεση που περικλείει τις βασικές παραμέτρους της. Ανάμεσα στα άγκιστρα { } βρίσκεται ο κώδικας που θα εκτελεστεί.

⁷ διαχωρισμός σε υπορουτίνες από τον διερμηνέα και μετατροπή σε γλώσσα χαμηλού επιπέδου.

⁸ μετάφραση ολόκληρου του πηγαιού κώδικα γραμμή προς γραμμή σε γλώσσα χαμηλότερου επιπέδου.

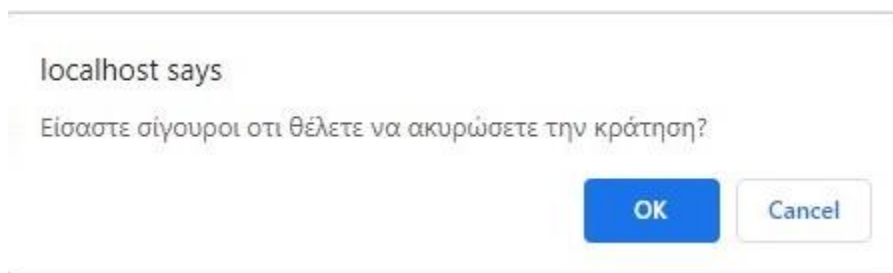
```
52 function resetdate()
53 {
54     document.getElementById("startdate").value="";
55 };
```

Εικόνα 4: Συνάρτηση επαναφοράς πεδίου ημερομηνίας στην select.php

Στην Εικόνα 4 βλέπουμε μια βασική μέθοδο της γλώσσας, την `getElementById()` η οποία ψάχνει να βρει το στοιχείο HTML με το αναγνωριστικό "startdate" ώστε να μπορέσει να τροποποιήσει την τιμή του (value). Η γλώσσα προσφέρει και κάποιες καθολικές συναρτήσεις όπως:

- `Alert()`: Εμφανίζει το επιθυμητό μήνυμα σε ένα παράθυρο, προτρέποντας τον χρήστη να πατήσει "OK".
- `Prompt()`: Εμφανίζει ένα μήνυμα ζητώντας από το χρήστη να γράψει ένα κείμενο και επιστρέφει το κείμενο, εκτός αν ο χρήστης πατήσει Ακύρωση (Cancel) ή το πλήκτρο Esc όπου δεν εμφανίζει τίποτα.
- `Confirm()`: Εμφανίζει ένα μήνυμα και περιμένει από το χρήστη να επιλέξει "OK" ή "Cancel" και επιστρέφει τις τιμές TRUE ή FALSE αντίστοιχα.

Αυτές οι μέθοδοι σταματούν την εκτέλεση ενεργειών και δεν επιτρέπουν στο χρήστη να αλληλοεπιδράσει με την υπόλοιπη σελίδα μέχρι να εξαφανιστεί το αναδυόμενο παράθυρο με το επιθυμητό μήνυμα.



Εικόνα 5: Παράθυρο confirm στην cart.php

```
<script>
// Get the modal
var modal = document.getElementById("myModal");

// Get the button that opens the modal
var tos = document.getElementById("tos");

// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks the button, open the modal
tos.onclick = function() {
    modal.style.display = "block";
}

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
    modal.style.display = "none";
}

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>
```

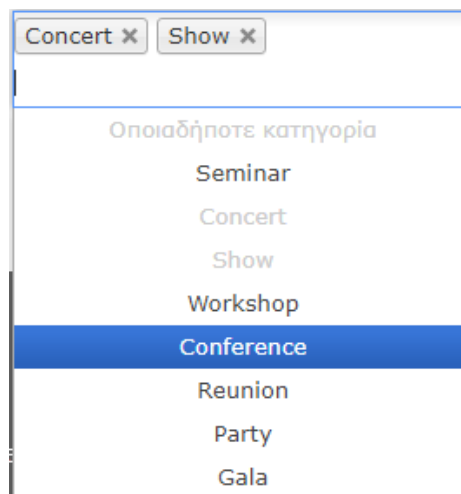
Εικόνα 6: Σενάριο JavaScript στην σελίδα εισόδου χρήστη login.php

Το σενάριο στην Εικόνα 6 αφορά την εμφάνιση και την απόκρυψη ενός παραθύρου τύπου Modal στην σελίδα εισόδου του χρήστη. Συγκεκριμένα είναι το παράθυρο με τους όρους χρήσης τις εφαρμογής στους οποίους ο χρήστης συμφωνεί με την χρήση της. Εδώ διακρίνονται δύο φράσεις, αυτές που ξεκινούν με var «ονομασία» = document.get και αυτές με την μορφή «ονομασία».onclick = function. Στις πρώτες γεμίζουμε κάποιες μεταβλητές (var) με τα αντίστοιχα στοιχεία της γλώσσας σήμανσης HTML. Στη συνέχεια στις επόμενες φράσεις ορίζουμε την διαδικασία που θα ακολουθηθεί εάν γίνει το γεγονός click σε αυτά τα στοιχεία. Συγκεκριμένα το στυλ εμφάνιση του διαχωριστικού div με id="myModal" θα γίνει "block" εφόσον γίνει κλικ στο στοιχείο με id="tos", ενώ θα αλλάξει σε "none" εάν γίνει κλικ στο στοιχείο με κλάση "close" η σε άλλο μέρος του παραθύρου (window).

2.4.2 Η βιβλιοθήκη jQuery και η τεχνική Ajax

Η αυξημένη χρήση της JavaScript από επαγγελματίες στον χώρο του προγραμματισμού πρόσφερε πολλά στην γλώσσα όπως την ανάπτυξη μεγάλης γκάμας βιβλιοθηκών της, την βελτίωση του προγραμματισμού με JS και την χρήση της εκτός προγραμμάτων web. Μια τέτοια βιβλιοθήκη της είναι η jQuery.

Η jQuery είναι η πιο ευρέως χρησιμοποιούμενη βιβλιοθήκη της JavaScript αποδεδειγμένα με ποσοστό 73% ανάμεσα σε 10 εκατομμύρια δημοφιλή web sites⁹ τον Μάιο του 2019. Είναι σχεδιασμένη ώστε να απλοποιεί το DOM (Document Object Model) της HTML, όπως και στο να διαχειρίζεται συμβάντα (γεγονότα), κινούμενα σχέδια CSS και τεχνικές Ajax. Στην εργασία μας το συναντάμε στην σελίδα αναζήτησης στο πεδίο επιλογής κατηγοριών, για την λειτουργία και την εμφάνισή του, όπως φαίνεται στην Εικόνα 7.



Εικόνα 7: jQuery για επιλογή κατηγοριών στην select.php

Η βασική της σύνταξη είναι της μορφής: \$(selector).action() όπου το σύμβολο \$ χρησιμοποιείται για να καθορίσει ή να εκκινήσει το σενάριο jQuery, το "selector" είναι η έκφραση «βρες το στοιχείο» HTML και action() κάνει την ενέργεια. Για παράδειγμα \$("p").hide() όπου αποκρύπτονται όλα τα στοιχεία <p>, \$(".class1").hide() που αποκρύπτονται τα στοιχεία με κλάση "class1" ή \$("#identifier").hide() όπου αποκρύπτεται το στοιχείο με αναγνωριστικό "identifier". Ακόμα η γλώσσα υποστηρίζει και γεγονότα στην σελίδα όπως το πάτημα κλικ (click) σε κάποιο στοιχείο, επιλογή σε κάποιο radio button ή περνώντας τον δείκτη του ποντικιού πάνω (mouseover) από κάποιο στοιχείο.

⁹ <https://en.wikipedia.org/wiki/JQuery>

Άλλα τέτοια γεγονότα άξια αναφοράς είναι η φόρτωση (load) και αλλαγή μεγέθους (resize) της σελίδας, mouseenter και mouseleave για τον δείκτη του ποντικιού, keypress και keydown για το πάτημα πλήκτρων και την υποβολή (submit), την αλλαγή (change), εστίαση (focus) για τις φόρμες. Μια άλλη κοινή χρήση είναι η μέθοδος \$(document).ready() η οποία μας επιτρέπει να εκτελέσουμε μια συνάρτηση αφού το έγγραφο έχει φορτωθεί πλήρως.

Η τεχνική Ajax (Asynchronous JavaScript and XML) είναι η μία από τις πιο βασικές τεχνικές προγραμματισμού σε JS. Χρησιμοποιεί πολλές τεχνολογίες διαδικτύου στην πλευρά του πελάτη για να δημιουργήσει ασύγχρονες διαδικτυακές εφαρμογές. Με αυτήν την τεχνική οι εφαρμογές μπορούν να επικοινωνήσουν με τον διακομιστή στο παρασκήνιο, χωρίς να επηρεαστεί δηλαδή η απεικόνιση στη μεριά του χρήστη ή να αλλάξει η συμπεριφορά της τρέχουσας σελίδας. Έτσι επιτυγχάνεται ανταλλαγή δεδομένων με τον server που μπορεί να αναπαρασταθεί δυναμικά χωρίς να χρειάζεται να ανανεωθεί ολόκληρη η σελίδα. Θα λέγαμε λοιπόν ότι η Ajax δεν είναι κάποια καινούρια τεχνολογία ή διαφορετική γλώσσα αλλά συνδυασμός τεχνολογιών που χρησιμοποιούνται με καινούριο τρόπο.

```
39 $('#search').click(function(){
40     var typeid = $('#typeid').val();
41     var cityid = $('#city').val();
42     var startdate = $('#startdate').val();
43     $.ajax({
44         url:"functions/do_search.php",
45         method:"POST",
46         data:{cityid:cityid, typeid:typeid, startdate:startdate},
47         success:function(data){
48             $('#filterresults').html(data);
49         }
50     });
51 });
```

Εικόνα 8: Σενάριο Ajax με jQuery που εκτελεί την μέθοδο POST ασύγχρονα

Στην Εικόνα 8 το σενάριο ελέγχει εάν πατήθηκε το κουμπί με το χαρακτηριστικό id="search". Εφόσον πατηθεί αποθηκεύονται στις μεταβλητές typeid, cityid και startdate οι κατάλληλες τιμές των στοιχείων με το αντίστοιχο αναγνωριστικό (id). Στη συνέχεια γίνεται μία ασύγχρονη κλήση στην σελίδα do_search.php, με την μέθοδο POST, στην οποία στέλνουμε σαν ορίσματα τις τιμές των παραπάνω μεταβλητών. Οι πληροφορίες επεξεργάζονται στον διακομιστή και, εφόσον το αποτέλεσμα είναι επιτυχές, το περιεχόμενο της απόκρισης ενσωματώνεται στον διαχωριστή περιεχομένου με id="filterresults".

2.4.3 Η γλώσσα PHP

Η γλώσσα PHP είναι μία δημοφιλής γλώσσα σεναρίου γενικού σκοπού η οποία ταιριάζει περισσότερο στον διαδικτυακό προγραμματισμό. Η επεξεργασία του κώδικά της γίνεται συνήθως στον server από έναν διερμηνέα PHP που έχει εγκατασταθεί σαν πρόσθετο σε αυτόν. Σε έναν web server το αποτέλεσμα της ερμηνείας και τις εκτέλεσης του κώδικα, που μπορεί να είναι οποιασδήποτε μορφής όπως π.χ. δημιουργία δεδομένων HTML, θα διαμορφώσει το σύνολο ή μέρος της απόκρισης HTTP. Περισσότερα για το πρωτόκολλο HTTP θα δούμε στην ενότητα 2.7.

Ένα σενάριο PHP αρχίζει με <?php και τελειώνει με ?> και μπορεί να τοποθετηθεί οπουδήποτε μέσα στο έγγραφο. Η εξορισμού επέκταση του αρχείου είναι η .php και συνήθως περιλαμβάνει ανάμικτα στοιχεία HTML και κώδικα σεναρίου PHP. Η σύνταξή της έχει ρίζες στην C, Java και Perl και είναι εύκολη στη μάθηση. Ο κύριος σκοπός της είναι να επιτρέψει στους προγραμματιστές να

γράφουν δυναμικές ιστοσελίδες γρήγορα όμως μπορεί να κάνει πολλά περισσότερα. Εκτός δηλαδή από το να τυπώνει δυναμικά HTML η γλώσσα βρίσκει πολλές εφαρμογές και εκτός διαδικτυακού περιβάλλοντος. Είναι ικανή να λειτουργεί σαν μία standalone γραφική εφαρμογή¹⁰ ή ακόμα και να ελέγξει ένα ρομποτικό drone¹¹. Ο διερμηνέας που παρέχεται από την Zend Engine είναι δωρεάν υπό την άδεια PHP. Στο θέμα μας η γλώσσα αυτή είναι θεμέλιο και χρησιμοποιείται τόσο για να δημιουργήσει περιεχόμενο δυναμικά όσο και για να εκτελέσει πιο εξειδικευμένες λειτουργίες. Μάλιστα όπως είδαμε όλες οι σελίδες (έγγραφα) μας έχουν την επέκταση .php ώστε να μπορούν να λειτουργήσουν στον web browser (client). Μάλιστα έχουν γίνει και οι κατάλληλες ρυθμίσεις στον server ώστε να μπορεί να επεξεργαστεί και να μεταγλωττίσει επιτυχώς τον κώδικα σε HTML που καταλαβαίνει ο client.

Στη σύνταξή της περιλαμβάνονται λέξεις κλειδιά όπως if, else, while, foreach, echo κ.α., συναρτήσεις, κλάσεις και συναρτήσεις που έχει ορίσει ο χρήστης. Τα προηγούμενα δεν είναι case – sensitive σε αντίθεση με τις μεταβλητές οι οποίες δηλώνονται με ξόνομα. Για να αποδώσουμε τιμή χρησιμοποιούμε το = και η φράση τελειώνει με ελληνικό ερωτηματικό, π.χ. \$name = "George"; και \$age = 30; Όπως βλέπουμε, σε αντίθεση με άλλες γλώσσες, η PHP δεν έχει προσδιορισμό τύπου μπροστά από το όνομα καθώς ο τύπος καθορίζεται την στιγμή της απόδοσης της τιμής. Οι τύποι που υποστηρίζει είναι:

- συμβολοσειράς (String)
- αληθείας (boolean)
- ακέραιος αριθμός (integer)
- κινητής υποδιαστολής (float)
- πίνακας (array)
- αντικείμενο (object)
- callable
- επαναληπτικός (iterable)
- πόρος (resource)
- κενό (NULL)

Ένα πλεονέκτημά της είναι οι μεταβλητές συστήματος που ονομάζονται Superglobals όπως η \$_SERVER, \$_GET, \$_POST, \$GLOBALS, \$_FILES, \$_SESSION κ.α. και περιλαμβάνουν όλες τις πληροφορίες του διακομιστή. Ένας κύριος τύπος στοιχείων είναι οι πίνακες οι οποίοι διαθέτουν πάρα πολλές συναρτήσεις όπως η array() που δημιουργεί έναν πίνακα, η ταξινόμηση sort(), συνένωση πινάκων array_merge() και πλήθος άλλων. Βασικές συναρτήσεις της είναι οι συνθήκες if() – elseif() – else, οι συνθήκες επανάληψης for() και while(), οι λειτουργίες επιστροφής τιμής return και return_array() και οι μέθοδοι εκτύπωσης στην οθόνη echo και print(). Τα σχόλια μπορούν να είναι μίας γραμμής και ξεκινάνε με διπλή κάθετο (//), και πολλαπλής γραμμής ξεκινώντας με /* και τελειώνουν με */.

Παρακάτω, στις Εικόνα 9 και Εικόνα 10 φαίνονται οι περιπτώσεις χρήσης της PHP για δημιουργία περιεχομένου και εκτέλεση βασικής λειτουργίας αντίστοιχα.

¹⁰ <https://www.php.net/manual/en/intro-whatcando.php>

¹¹ <https://github.com/jolicode/php-ar-drone>

Οι τεχνολογίες υλοποίησης

```
$output='';

if (empty($_SESSION['cart']))
{
    $output .= '<div class="center w3-padding w3-margin">';
    $output .= '<h2> Δεν έχετε κάνει καμία κράτηση!</h2>';
    $output .= '</div>';
}
else
{
    $cart = $_SESSION['cart'];
    //print_r($cart);
    $output .= '<div class="w3-container w3-padding">';
    $output .= '<h2>Οι κρατήσεις σας: </h2>';
    $output .= '<table id="cart-table" style="width:100%" class="w3-table-all center">';
    $output .= '<tr>';
    $output .= '<th>Id</th>';
    $output .= '<th>Event</th>';
    $output .= '<th>Ημερομηνία</th>';
    $output .= '<th>Τοποθεσία</th>';
    $output .= '<th>Αριθμός θέσης</th>';
    $output .= '<th>Ακύρωση</th>';
    $output .= '</tr>';

    foreach ($cart as $key => $value)
    {
        $output .= '<tr>';
        $output .= '<td>'. $key. '</td>';

        foreach ($value as $row => $data) {
            if ($row != 'EventId') {
                $output .= '<td>'. $data. '</td>';
            }
        }
        $output .= '<td><input id="cancel" type="button" value="X" class="w3-btn w3-red w3-round-small w3-margin-left"></td>';
        $output .= '</tr>';
    }
    $output .= '</tr></table></div><br/>';
    $output .= '<button id="checkout" class="w3-btn w3-teal w3-padding w3-round w3-margin-left">Συνέχεια</button>';
}
echo $output;
```

Εικόνα 9: Δημιουργία HTML δυναμικά με χρήση PHP

```
1 <?php
2
3 // Connection to the database ticket-manager (MySQL).
4 $host = "localhost";
5 $user = "root";
6 $pass = "654321";
7 $dbname = "ticket-manager";
8
9 $dbconn = mysqli_connect($host, $user, $pass, $dbname);
10
11 // Case connection fails.
12 if (!$dbconn) {
13     die("Connection failed: ".mysqli_connect_error());
14 }
15 ?>
```

Εικόνα 10: Λειτουργία σύνδεσης με τη βάση δεδομένων σε PHP

Στην Εικόνα 9 η μεταβλητή \$output παίρνει τιμή βάση της μεταβλητής συστήματος \$_SESSION['cart']. Στην περίπτωση που η τιμή της δεύτερης δεν έχει οριστεί η \$output παίρνει την μορφή διαχωριστή περιεχομένου με κεφαλίδα τύπου <h2>. Εφόσον έχει οριστεί τότε η \$output γίνεται πιο σύνθετη και περιλαμβάνει πίνακα στοιχείων μέσα σε διαχωριστή περιεχομένου. Οι τιμές του πίνακα γεμίζουν ανάλογα με το πόσα στοιχεία είχε ο πίνακας της μεταβλητής \$cart που πήρε τις

τιμές του πίνακα `$_SESSION['cart']`. Στην τελευταία γραμμή, και ενώ έχει διαμορφωθεί ο κώδικας HTML στην `$output`, με την εντολή `echo` επιστρέφουμε το περιεχόμενο της `$output`.

Στην Εικόνα 10 οι μεταβλητές `$host`, `$user`, `$pass`, `$dbname` παίρνουν ως τιμές τις πληροφορίες για την επίτευξη σύνδεσης με την βάση δεδομένων. Έπειτα καλείται η μέθοδος `mysqli_connect()` της PHP για να πραγματοποιήσει την σύνδεση. Η συνθήκη στο τέλος έχει χαρακτήρα ελέγχου (debugging), δηλαδή μας ενημερώνει εφόσον κάτι πάει στραβά με την διαδικασία σύνδεσης προβάλλοντας το αντίστοιχο μήνυμα λάθους.

2.5 Η δομή της εμφάνισης της εφαρμογής

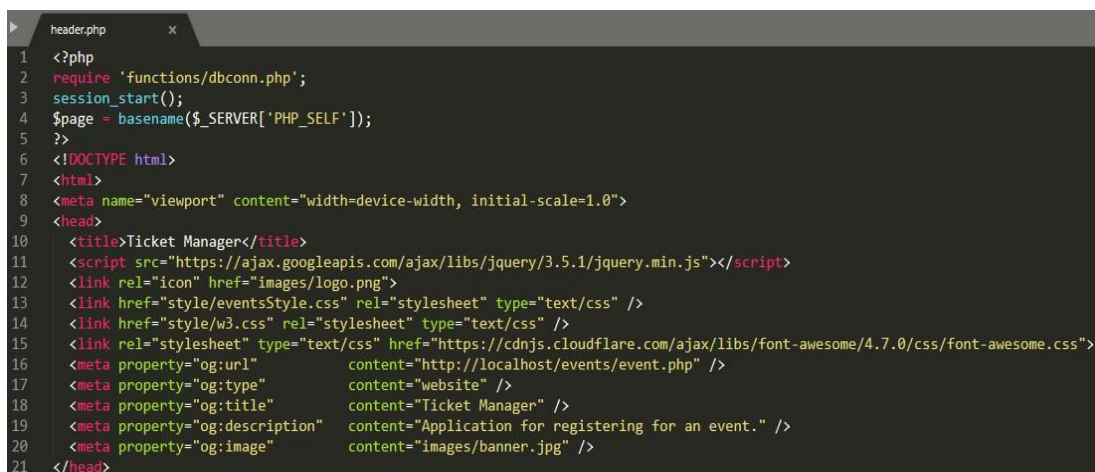
Η εφαρμογή μας συνδυάζει τις παραπάνω γλώσσες για την μορφοποίηση του front end, δηλαδή την διεπαφή με το χρήστη όσο και για να εκτελέσει λειτουργίες στο backend. Η δόμηση της εμφάνισης ξεκινά, όπως αναφέραμε στην ενότητα 2.2, από την σελίδα `header.php`. Όπως φανερώνει και το όνομά της περιέχει τα στοιχεία σχηματισμού της κεφαλίδας αλλά και του αριστερού μενού πλοήγησης (navigation menu) τα οποία πρέπει να περιλαμβάνονται σε κάθε σελίδα. Για να το πετύχουμε αυτό, αντί να γράφουμε σε κάθε σελίδα τον ίδιο κώδικα, χρησιμοποιούμε σε κάθε σελίδα εμφάνισης την εντολή της PHP “include”.



```
login.php
1 <?php
2 include 'header.php';
3 ?>
```

Εικόνα 11: Η εντολή include της PHP

Η χρήση της `header` όμως δεν είναι μόνο να προσαρμόζει τα στοιχεία εμφάνισης της κεφαλίδας και του μενού που περιλαμβάνει, αλλά να μεταφέρει και πληροφορίες μεταξύ των σελίδων που την επικαλούνται. Πληροφορίες σαν αυτές είναι οι πληροφορίες συνόδου (session), σύνδεσης με τη βάση ή αναφορές σε εξωτερικά αρχεία CSS και την βιβλιοθήκη jQuery. Ακόμα περιλαμβάνει μεταδεδομένα για την καλύτερη αναγνώριση του ιστότοπου από τις μηχανές αναζήτησης υλοποιώντας το πρωτόκολλο Open Graph.



```
header.php
1 <?php
2 require 'functions/dbconn.php';
3 session_start();
4 $page = basename($_SERVER['PHP_SELF']);
5 ?>
6 <!DOCTYPE html>
7 <html>
8 <meta name="viewport" content="width=device-width, initial-scale=1.0">
9 <head>
10 <title>Ticket Manager</title>
11 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
12 <link rel="icon" href="images/logo.png">
13 <link href="style/eventsStyle.css" rel="stylesheet" type="text/css" />
14 <link href="style/w3.css" rel="stylesheet" type="text/css" />
15 <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.css">
16 <meta property="og:url" content="http://localhost/events/event.php" />
17 <meta property="og:type" content="website" />
18 <meta property="og:title" content="Ticket Manager" />
19 <meta property="og:description" content="Application for registering for an event." />
20 <meta property="og:image" content="images/banner.jpg" />
21 </head>
```

Εικόνα 12: Οι πληροφορίες στην κεφαλίδα της `header.php`

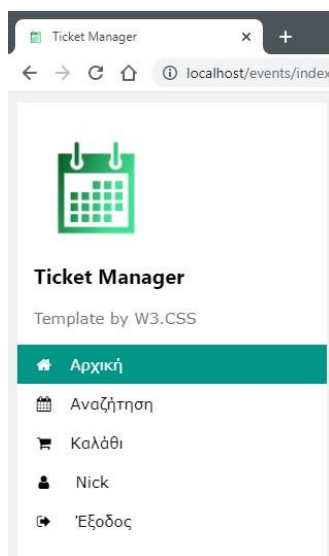
Οι τεχνολογίες υλοποίησης

```
23 <!-- Sidebar/menu -->
24 <nav class="w3-sidebar w3-collapse w3-white w3-animate-left" style="z-index:3;width:300px;" id="sidebar"><br>
25 <div class="w3-container">
26 <a href="#" onclick="w3_close()" class="w3-hide-large w3-right w3-jumbo w3-padding w3-hover-grey" title="close menu">
27 <i class="fa fa-remove"></i>
28 </a>
29 
30 <h2><i>Ticket Manager</i></h2>
31 <p class="w3-text-grey">Template by W3.CSS</p>
32 </div>
33 <div class="w3-bar-block">
34 <a href="index.php" onclick="w3_close()" class="w3-bar-item w3-button w3-padding <?php if($page=="index.php"){echo 'w3-teal';}>?><i class="fa fa-home fa-fw w3-margin-right"></i>Αρχική</a>
35 <a href="select.php" id="select.php" onclick="w3_close()" class="w3-bar-item w3-button w3-padding <?php if($page=="select.php"){echo 'w3-teal';}>?><i class="fa fa-calendar fa-fw w3-margin-right"></i>Αναζήτηση</a>
36 <?php
37 {
38 if (isset($_SESSION['userid']) && $_SESSION['userid'] == '1')
39 {
40 $addevent = ' <a href="new_event.php" onclick="w3_close()" class="w3-bar-item w3-button w3-padding ' ;
41 $adminpanel = ' <a href="dashboard.php" onclick="w3_close()" class="w3-bar-item w3-button w3-padding ' ;
42 if($page == 'new_event.php')
43 {
44 $addevent = ' w3-teal'><i class="fa fa-plus fa-fw w3-margin-right"></i>Προσθήκη</a>;
45 $adminpanel = ' ><i class="fa fa-heartbeat fa-fw w3-margin-right"></i>Dashboard</a>;
46 }
47 elseif($page == 'dashboard.php')
48 {
49 $addevent = ' ><i class="fa fa-plus fa-fw w3-margin-right"></i>Προσθήκη</a>;
50 $adminpanel = ' w3-teal'><i class="fa fa-heartbeat fa-fw w3-margin-right"></i>Dashboard</a>;
51 }
52 else
53 {
54 $addevent = ' ><i class="fa fa-plus fa-fw w3-margin-right"></i>Προσθήκη</a>;
55 $adminpanel = ' ><i class="fa fa-heartbeat fa-fw w3-margin-right"></i>Dashboard</a>;
56 }
57 echo $adminpanel $addevent;
58 }
59 >>
```

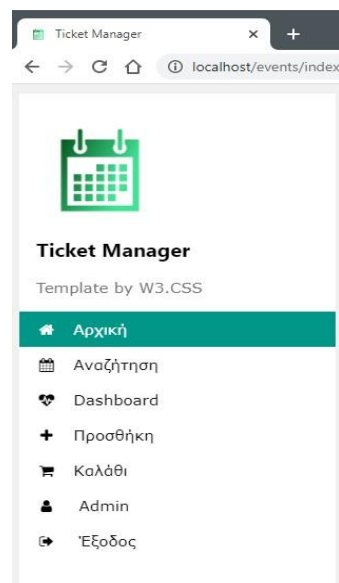
Εικόνα 13: Κομμάτι του μενού πλοήγησης στην header.php

Στην Εικόνα 13 διακρίνουμε την ανάμιξη της PHP ανάμεσα στον κώδικα της HTML. Σε περιπτώσεις όπως αυτές των γραμμών 34 και 35 στην Εικόνα 13, βλέπουμε την επέμβαση της PHP για να αποδώσει μία κλάση στα χαρακτηριστικά του συνδέσμου `<a>`. Συγκεκριμένα η λειτουργία του σεναρίου είναι να αποδίδει ή όχι την κλάση χρωματισμού “w3-teal” στον κάθε σύνδεσμο εφόσον είναι σχετικός με την τρέχουσα σελίδα. Στις Εικόνα 14 και Εικόνα 15 παρακάτω φαίνεται και η λειτουργία αυτή.

Στις γραμμές 36 έως 58 της Εικόνα 13, η επέμβαση της PHP έχει σκοπό την εμφάνιση ή την απόκρυψη κάποιων συνδέσμων σύμφωνα με το αν έχει γίνει σύνδεση, και τον ρόλο που έχει ο συνδεδεμένος χρήστης στο σύστημα. Ειδικότερα εξετάζει την περίπτωση για το αν θα εμφανίσει τους συνδέσμους για το κομμάτι διαχείρισης εφόσον ο χρήστης είναι διαχειριστής. Αυτό επιτυγχάνεται με την δυναμική δημιουργία των δύο συνδέσμων σε μορφή HTML (μεταβλητές `$addevent` και `$dashboard`) και την εκτύπωσή τους με την εντολή “echo” στο κατάλληλο σημείο της λίστας. Μάλιστα στην ίδια συνθήκη εφαρμόζεται και ο ανάλογος χρωματισμός όπως παραπάνω για την ευκολία στην πλοήγηση.



Εικόνα 14: Μενού χρήστη



Εικόνα 15: Μενού διαχειριστή

2.6 Η γλώσσα SQL

Η γλώσσα SQL (Structured Query Language) είναι γλώσσα που επιτρέπει την πρόσβαση και την επεξεργασία δεδομένων σε μία βάση δεδομένων (ΒΔ). Πιο αναλυτικά οι λειτουργίες που μπορεί να εκτελέσει είναι:

- να ανακτήσει δεδομένα από μια βάση δεδομένων (SELECT ... FROM)
- να εισαγάγει εγγραφές σε μια βάση δεδομένων (INSET INTO ... VALUES)
- να ενημερώσει τις εγγραφές σε μια βάση δεδομένων (UPDATE .. SET)
- να διαγράψει εγγραφές από μια βάση δεδομένων (DELETE FROM)
- να δημιουργήσει νέες βάσεις δεδομένων (CREATE DATABASE)
- να δημιουργήσει νέους πίνακες σε μια βάση δεδομένων (CREATE TABLE)
- να διαγράψει ολόκληρους πίνακες (DROP TABLE)
- να δημιουργήσει αποθηκευμένες διαδικασίες σε μια βάση δεδομένων (CREATE PROCEDURE)
- να δημιουργήσει προβολές σε μια βάση δεδομένων (CREATE VIEW)
- να τροποποιήσει πίνακες, διαδικασίες και προβολές (ALTER TABLE)

Η SQL έγινε πρότυπο του Αμερικανικού Ινστιτούτου Εθνικών Προτύπων (ANSI) το 1986 και του Διεθνή Οργανισμού Τυποποίησης (ISO) έναν χρόνο αργότερα. Παρότι προτυποποιημένη σήμερα υπάρχουν πολλές εκδόσεις της, όλες όμως υποστηρίζουν τις βασικές της λειτουργίες όπως SELECT, UPDATE, DELETE, INSERT και WHERE.

Άλλες εντολές άξιες αναφοράς είναι:

- Η εντολή FROM που χρησιμοποιείται για τον καθορισμό του πίνακα κατά την επιλογή ή διαγραφή στοιχείων από αυτόν.
- Η εντολή WHERE η οποία φιλτράρει ένα σύνολο αποτελεσμάτων ώστε να περιλαμβάνει μόνο εγγραφές που πληρούν καθορισμένη προϋπόθεση.

Βασικό στοιχείο είναι το σύμβολο αστερίσκος (*) το οποίο χρησιμοποιείται για να αναφερθούμε σε όλα τα πεδία ενός πίνακα. Άλλες δυνατότητες είναι οι συνδυαστικές λέξεις (AND, OR, NOT, LIKE κ.α.), οι κατηγοριοποιήσεις των εγγραφών (ORDER BY, GROUP BY) και οι έτοιμες συναρτήσεις που εντάσσονται σε τέσσερις κατηγορίες (String Functions, Numeric Functions, Date functions, Advanced Functions).

Σύμφωνα με το w3schools για να δημιουργήσει κάποιος έναν ιστότοπο που εμφανίζει δεδομένα από μια βάση δεδομένων, θα χρειαστεί:

- ένα πρόγραμμα βάσης δεδομένων RDBMS (π.χ. MS Access, SQL Server, MySQL)
- να χρησιμοποιήσει μια γλώσσα σεναρίου στο διακομιστή, όπως PHP ή ASP
- να χρησιμοποιεί την SQL για να λαμβάνει τα δεδομένα που θέλει
- να χρησιμοποιήσει HTML / CSS για το στυλ της σελίδας.

Από τα παραπάνω λοιπόν μένει να δούμε την βάση δεδομένων και τον διακομιστή.

```
$sql = "DELETE FROM reservation WHERE SeatId='".$seatid.'";  
$result = mysqli_query($dbconn,$sql);
```

Εικόνα 16: Εκτέλεση εντολής DELETE της SQL με PHP

2.7 Ο διακομιστής Apache HTTP

Ως εξυπηρετητής ιστού ορίζεται ένα πρόγραμμα λογισμικού, ή ένα σύστημα υλικού που τρέχει το λογισμικό αυτό, το οποίο ικανοποιεί αιτήματα πελατών στο διαδίκτυο. Πρωταρχική του λειτουργία είναι να αποθηκεύει, να επεξεργάζεται και να παραδίδει ιστοσελίδες σε ένα πρόγραμμα πελάτη όπως ένας web browser. Γενικά μπορεί να φιλοξενεί από μία έως πολλές ιστοσελίδες και επεξεργάζεται τα αιτήματα που λαμβάνει μέσω HTTP ή άλλων πρωτοκόλλων. Οι περισσότερες ιστοσελίδες είναι έγγραφα HTML τα οποία μπορεί, όπως είδαμε, να περιλαμβάνουν εικόνες, CSS και σενάρια εκτός από περιεχόμενο (κείμενο). Άλλες χρήσεις του είναι η ενσωμάτωση του σε συσκευές όπως εκτυπωτές, δρομολογητές και κάμερες web ώστε να μπορούν να διαχειρίζονται απλώς από ένα πρόγραμμα πελάτη, το οποίο είναι και ενσωματωμένο σε κάθε λειτουργικό σύστημα.

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (world wide web). Είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, κυρίως γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation). Ο Apache μπορεί να εγκατασταθεί και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης βάσης δεδομένων π.χ. Oracle, MySQL, όπως έγινε και στο θέμα μας.

Σύμφωνα με την τελευταία καταμέτρηση¹² στις 26 Μαΐου 2020 από την netcraft ο Apache παραμένει στην 1^η θέση των διακομιστών διαδικτύου παγκοσμίως, με πάνω από 52 εκατομμύρια ιστοσελίδες με ποσοστό 27% της αγοράς. Ο Apache περιλαμβάνει υποστήριξη για γλώσσες σεναρίου στην πλευρά του διακομιστή (server-side scripting) συμπεριλαμβανομένης και της PHP. Για την εργασία χρησιμοποιήθηκε η έκδοση 2.4 που είναι και η πιο πρόσφατη.

2.8 Το σύστημα διαχείρισης βάσης δεδομένων MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων (Relational Database Management System) και είναι δωρεάν υπό την άδεια της GNU General Public License. Ξεκίνησε από την Σουηδική εταιρία MySQL AB η οποία εξαγοράστηκε από την Sun Microsystems και στη συνέχεια από την Oracle Corporation μέχρι σήμερα.

Μία βάση δεδομένων γενικά από έναν ή περισσότερους πίνακες. Κάθε πίνακας είναι μία συλλογή σχετικών καταχωρήσεων. Προσδιορίζεται από ένα όνομα και περιέχει εγγραφές (σειρές) με δεδομένα και πεδία (στήλες) με συγκεκριμένες πληροφορίες. Ένας πίνακας μπορεί να έχει μόνο ένα πρωτεύων κλειδί (PRIMARY KEY), το οποίο μπορεί να αποτελείται από ένα μόνο ή από πολλαπλά πεδία. Ο περιορισμός PRIMARY KEY αναγνωρίζει με μοναδικό τρόπο κάθε εγγραφή σε έναν πίνακα. Ο περιορισμός FOREIGN KEY είναι ένα κλειδί που χρησιμοποιείται για την σύνδεση δύο πινάκων. Ορίζεται ως ξένο – δευτερεύων κλειδί και είναι πεδίο (ή συλλογή πεδίων) σε έναν πίνακα που αναφέρεται στο πρωτεύων κλειδί σε έναν άλλο πίνακα. Οι βασικές τις λειτουργίες (Create Retrieve Update Delete, CRUD) στα δεδομένα εκτελούνται από την μηχανή InnoDB η οποία υποστηρίζει συναλλαγές (transactions) και περιορισμό ξένου κλειδιού. Ένα από τα πιο δυνατά χαρακτηριστικά της είναι η επαναφορά από βλάβη ή άλλο απρόβλεπτο τερματισμό του συστήματος βασιζόμενη στα αρχεία καταγραφής της.

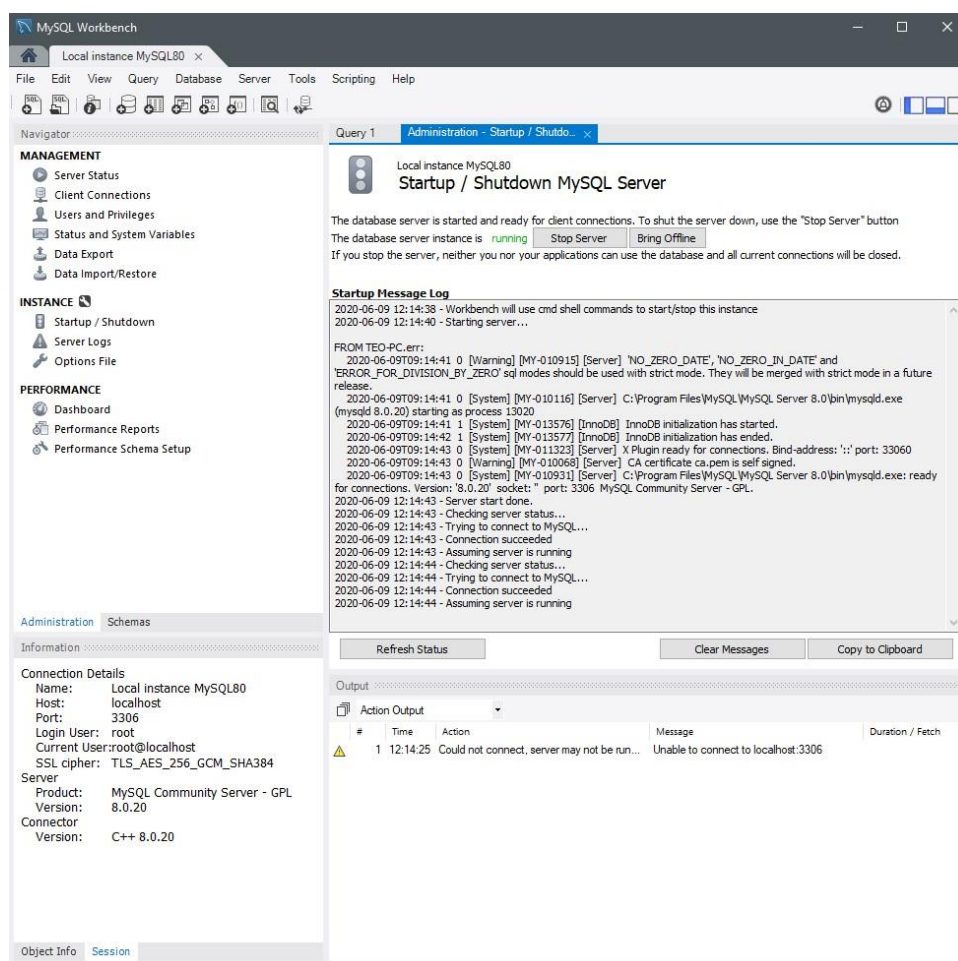
Εξορισμού η MySQL δεν διαθέτει γραφικό περιβάλλον αλλά η διαχείριση γίνεται μέσω της γραμμής εντολών (command line) του συστήματος που την φιλοξενεί. Για την διαχείριση στην εργασία χρησιμοποιήθηκε το πρόγραμμα διαχείρισης MySQL Workbench 8.0 το οποίο διαθέτει γραφικό περιβάλλον για να δουλέψουμε με την MySQL. Οι κύριοι τομείς λειτουργίας του είναι:

¹² <https://news.netcraft.com/archives/category/web-server-survey/>

Οι τεχνολογίες υλοποίησης

- Ανάπτυξη και διαχείριση SQL: Επιτρέπει την δημιουργία και τους τρόπους σύνδεσης με μια βάση δεδομένων και τη δυνατότητα να εκτελούμε query μέσω του ενσωματωμένου συντάκτη SQL.
- Μοντελοποίηση δεδομένων: Επιτρέπει την δημιουργία μοντέλων από το σχήμα της βάσης με αντίστροφη μηχανική (reverse engineering) και περιέχει συντάκτη πινάκων με τον οποίο μπορούμε να διαχειριστούμε πίνακες, όψεις, ρουτίνες κ.α.
- Διαχείριση database server
- Μεταφορά δεδομένων: Επιτρέπει την μετανάστευση από MS SQL Server, SQLite, PostgreSQL και άλλα σχεσιακά συστήματα.

Όπως είδαμε το ΣΔΒΔ είναι αναπόσπαστο κομμάτι του διαδικτυακού προγραμματισμού και συμπληρώνει το υποσύστημα γνωστό ως LAMP (Linux, Apache, MySQL, Perl/PHP/Python) ή WAMP εάν τρέχει σε περιβάλλον Windows. Χρήσεις του συναντάμε σε πολλές διαδικτυακές εφαρμογές που είναι οδηγούμενες από την βάση όπως Drupal, Joomla και Wordpress. Ακόμα συναντάται σε πολλές δημοφιλείς ιστοσελίδες όπως το Facebook, το Flickr, το Twitter και το YouTube.



Εικόνα 17: Το περιβάλλον της εφαρμογής MySQL Workbench

2.9 Επίλογος

Στο κεφάλαιο αυτό αναλύσαμε τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του θέματος. Είδαμε τις γλώσσες προγραμματισμού και τα υποσυστήματα που συνθέσαμε για την ανάπτυξη του θέματος. Επίσης ρίξαμε μία πρώτη ματιά στην εμφάνισή της εφαρμογής. Στο επόμενο κεφάλαιο θα κάνουμε μία εισαγωγή στο ηλεκτρονικό εμπόριο και θα εξετάσουμε την εξέλιξη των συστημάτων κρατήσεων εισιτηρίων.

Κεφάλαιο 3: Ηλεκτρονικό εμπόριο και διαχείριση γεγονότων

3.1 Εισαγωγή

Το ηλεκτρονικό εμπόριο στις μέρες μας είναι αναπόσπαστο κομμάτι των επιχειρήσεων, όλων των μεγεθών. Οι περισσότεροι νέοι είναι εξοικειωμένοι με την αναζήτηση και αγορά προϊόντων και υπηρεσιών μέσω του παγκόσμιου ιστού. Βάση στοιχείων της ΕΛΣΤΑΤ και συγκεκριμένα στο άρθρο «Η Ελλάδα με Αριθμούς»¹³ το ποσοστό του πληθυσμού που χρησιμοποίησε οποτεδήποτε το διαδίκτυο και πραγματοποίησε ηλεκτρονικές αγορές από το 2015 έως το 2019 έχει αυξηθεί κατά 7,7%.

3.2 Ορισμός ηλεκτρονικού εμπορίου

Στις μέρες μας το ηλεκτρονικό εμπόριο ή η-εμπόριο (Electronic Commerce ή e-commerce αντίστοιχα, ή EC) αποτελεί αναπόσπαστο κομμάτι του παγκόσμιου εμπορίου. Ωστόσο, η συγκεκριμένη έννοια δεν έχει ακόμη αποκτήσει έναν αποκλειστικό ορισμό, γεγονός που οφείλεται ενδεχομένως στη ραγδαία ανάπτυξη της τεχνολογίας. Είναι λογικό, στην πρώιμη εμφάνισή του, τη δεκαετία του '70, ο όρος να αφορούσε μόνο τα συστήματα ηλεκτρονικής μεταφοράς χρηματικών πόρων (Electronic Funds Transfer-EFT) μεταξύ τραπεζών, μέσα από ασφαλή ιδιωτικά δίκτυα, και όχι για παράδειγμα την ηλεκτρονική επικοινωνία ή την παροχή υπηρεσιών. Ωστόσο, παρά τις διαφορές που παρατηρούνται κατά καιρούς στους ορισμούς του, κανένας δεν δείχνει να αναιρεί τον άλλον. Πιο συγκεκριμένα, ορισμένοι από τους ορισμούς αυτούς είναι:

Το ηλεκτρονικό εμπόριο ορίζεται ως η διαδικασία της αγοράς και πώλησης ή ανταλλαγής αγαθών, υπηρεσιών και πληροφοριών μέσω δικτύων υπολογιστών, συμπεριλαμβανομένου και του διαδικτύου [2].

Το ηλεκτρονικό εμπόριο είναι η αλληλεπίδραση μεταξύ των συστημάτων επικοινωνίας, των συστημάτων διαχείρισης δεδομένων και των συστημάτων ασφάλειας, εξαιτίας των οποίων επιτυγχάνεται η ανταλλαγή εμπορικών πληροφοριών σχετικά με την πώληση προϊόντων ή υπηρεσιών που είναι διαθέσιμα [3]. Το ηλεκτρονικό εμπόριο αναφέρεται σε όλους τους τύπους ηλεκτρονικών συναλλαγών μεταξύ επιχειρήσεων και εμπλεκόμενων εκτός επιχείρησης, είτε πρόκειται για οικονομικές συναλλαγές, είτε για ανταλλαγές πληροφοριών, είτε για άλλες υπηρεσίες [5].

Μία έννοια που συγγέεται συχνά με το ηλεκτρονικό εμπόριο είναι το ψηφιακό / ηλεκτρονικό επιχειρείν (digital / electronic business).Για κάποιους η έννοια αυτή θεωρείται ως ένας ευρύτερος ορισμός του ηλεκτρονικού εμπορίου και το αντικαθιστά. Ενώ για άλλους, η έννοια αυτή διαφέρει και θεωρείται ως συμπλήρωμα των δραστηριοτήτων του η-εμπορίου. Δύο αντίστοιχα παραδείγματα ορισμών του η-επιχειρείν είναι :

Από τη μία πλευρά, σύμφωνα με τον Παπαθανασίου,(2011, σελ. 97), ο όρος ηλεκτρονικό επιχειρείν για μια επιχείρηση ή έναν οργανισμό περιλαμβάνει ολόκληρο τον κύκλο των οργανωσιακών διαδικασιών, οι οποίες εκτελούνται ψηφιακά, είτε αυτές είναι εσωτερικού είτε εξωτερικού χαρακτήρα και επιπλέον περιλαμβάνει και αυτό το ίδιο το ηλεκτρονικό εμπόριο, καθώς και ολόκληρη την πληροφοριακή υποδομή.

Από την άλλη, όπως ορίζει ο Chaffey, (2016, σελ.14) το ηλεκτρονικό επιχειρείν, διαφέρει από το η-εμπόριο, καθώς είναι ο τρόπος με τον οποίο οι επιχειρήσεις εφαρμόζουν την ψηφιακή τεχνολογία και τα ψηφιακά μέσα για να αυξήσουν την ανταγωνιστικότητά τους, βελτιστοποιώντας τις εσωτερικές διαδικασίες τους στα διαδικτυακά και παραδοσιακά κανάλια προμήθειας και διάθεσης στην αγορά.

¹³ https://www.statistics.gr/documents/20181/1515741/GreeceInFigures_2020Q1_GR.pdf

3.3 Πλεονεκτήματα και Μειονεκτήματα

Ένα τέτοιο φαινόμενο όπως αυτό του ηλεκτρονικού εμπορίου, που έχει λάβει παγκόσμιες διαστάσεις, προσφέρει πολλά οφέλη σε καταναλωτές και επιχειρήσεις, αλλά εγκυμονεί και κινδύνους. Σε αυτό το κεφάλαιο θα παρουσιαστούν τόσο τα υπέρ, όσο και τα κατά του, τα οποία λόγω της συνεχόμενης εξέλιξής του είναι λογικό με την πάροδο του χρόνου να μεταβάλλονται.

3.3.1 Πλεονεκτήματα για τους καταναλωτές

Στην πρώτη υποενότητα προβάλλονται τα πλεονεκτήματα του η-εμπορίου που σχετίζονται με τους καταναλωτές. Το η-εμπόριο λοιπόν :

- Προσφέρει τη δυνατότητα πραγματοποίησης συναλλαγών ανά πάσα στιγμή, αφού οι συναλλαγές πραγματοποιούνται 24 ώρες την ημέρα και 7 ημέρες την εβδομάδα, 365 μέρες το χρόνο. Συνεπώς, ο καταναλωτής έχει την ευχέρεια να κάνει τις αγορές του οποιαδήποτε ώρα τον βολεύει μέσα στην ημέρα, χωρίς να περιορίζεται στο καθιερωμένο ωράριο των φυσικών καταστημάτων.
- Εκμηδενίζει τις αποστάσεις, καθώς οι συναλλαγές πραγματοποιούνται άμεσα, ανεξάρτητα από γεωγραφικά όρια και την απόσταση που χωρίζει την επιχείρηση από τον καταναλωτή. Αυτό το γεγονός εξοικονομεί στο χρήστη χρόνο αλλά και χρήματα που θα αφορούσαν τη μετακίνηση του στο φυσικό κατάστημα.
- Προφέρει σφαιρική πληροφόρηση, διότι οι καταναλωτές έχουν την ευκαιρία να πραγματοποιήσουν συγκρίσεις τόσο στις τιμές όσο και στην ποιότητα των προϊόντων που επιθυμούν να αγοράσουν.
- Κάνει τις συναλλαγές να είναι γρήγορες και άμεσες. Από τη στιγμή που ολοκληρώνεται η παραγγελία, σε 1-2 ημέρες συνήθως ο καταναλωτής έχει παραλάβει το προϊόν του ή διαφορετικά ενημερώνεται για την διαθεσιμότητα του προϊόντος και για τον χρόνο παραλαβής

3.3.2 Πλεονεκτήματα για τις Επιχειρήσεις

Στην δεύτερη υποενότητα παρουσιάζονται τα οφέλη του η-εμπορίου που καρπώνονται οι επιχειρήσεις. Το η-εμπόριο σε αυτές :

- Προσφέρει επέκταση των γεωγραφικών ορίων των συναλλαγών τους και την ένταξή τους στην παγκόσμια αγορά. Έτσι διευρύνει το καταναλωτικό τους κοινό και αυξάνει κατά συνέπεια τα χρηματικά τους κέρδη.
- Με τις νέες τεχνολογίες που συνεχώς αναδύονται, κάνει το περιβάλλον μεταξύ των επιχειρήσεων περισσότερο ανταγωνιστικό. Έτσι κάθε εταιρία που υιοθετεί και αξιοποιεί πρώτη τις δυνατότητες αυτές, έχει προβάδισμα στο χώρο της αγοράς.
- Συμβάλλει στη διαμόρφωση στρατηγικών και μάρκετινγκ με βάση τις προτιμήσεις των καταναλωτών, ώστε να παρέχουν εξειδικευμένα προϊόντα και υπηρεσίες.
- Βελτιώνει τη λειτουργία μιας επιχείρησης σε τομείς όπως είναι η εσωτερική οργάνωση, και βοηθάει στην αποφυγή σφαλμάτων, απωλειών και φθορών που μπορούν να προκύψουν ή να μη γίνουν αντιληπτά από τον άνθρωπο.
- Βοηθά στην ταχύτερη παράδοση προϊόντων, ειδικά των ψηφιακών και στην καλύτερη εξυπηρέτηση πελατών, αφού αυτοματοποιεί πολλές διαδικασίες και επικοινωνίες.
- Βελτιώνει την συνεργασία, την αποδοτικότητα και την παραγωγικότητα της επιχείρησης. Κάθε επικοινωνία που είναι απαραίτητη για μια εμπορική συναλλαγή μεταξύ εταιρίας και προμηθευτή κοστίζει και διαρκεί λιγότερο.

3.3.3 Μειονεκτήματα για τους Καταναλωτές

Παρά τα θετικά του στοιχεία, όπως αναφέρθηκε στην αρχή της ενότητας, το ηλεκτρονικό εμπόριο ενέχει και αρκετούς κινδύνους. Από την μεριά των καταναλωτών, οι κίνδυνοι αυτοί είναι οι εξής :

- Δεν υπάρχει η δυνατότητα της φυσικής επαφής, της δοκιμής του προϊόντος, του ελέγχου ποιότητας και χρηστικότητας από αυτούς. Οι παράγοντες αυτοί επηρεάζουν την πρόθεση του χρήστη να αγοράσει ένα προϊόν ειδικά όταν αυτό αφορά είδη ένδυσης, υπόδησης, καλλυντικά κλπ.
- Απαιτείται η γνώση χρήσης ηλεκτρονικού υπολογιστή και του διαδικτύου. Συνεπώς, πολλές κατηγορίες ανθρώπων αποκλείονται από το η-εμπόριο. Ακόμη, συχνά απαιτείται για τις πληρωμές η κατοχή πιστωτικής κάρτας, που δεν διαθέτουν όλοι.
- Η έλλειψη ασφάλειας στις ηλεκτρονικές πληρωμές είναι ένα πολύ σημαντικό θέμα. Δεν είναι λίγες οι περιπτώσεις εξαπάτησης των καταναλωτών, από ηλεκτρονικά καταστήματα που λαμβάνουν πληρωμές χωρίς να αποστείλουν τελικά το προϊόν.
- Στον τομέα της ηλεκτρονικής ασφάλειας ανήκει και η υποκλοπή στοιχείων των χρηστών, που φτάνουν στα χέρια κακόβουλων ατόμων, τα οποία μπορούν να τα εκμεταλλευτούν προς όφελός τους με τη μέθοδο της πλαστοπροσωπίας.
- Η αποξένωση των καταναλωτών από τον συνάνθρωπο, καθώς η ευκολία της αγοράς χωρίς την απαίτηση της κοινωνικής συναναστροφής σε συνεχόμενη βάση καθιστούν τους χρήστες κοινωνικά αποκομμένους.

3.3.4 Μειονεκτήματα για τις Επιχειρήσεις

Σχετικά με τους εταιρικούς οργανισμούς, τα κατά του η-εμπορίου που καλούνται να αντιμετωπίσουν είναι :

- Τον ενδοιασμό των καταναλωτών να στείλουν τα στοιχεία της πιστωτικής τους κάρτας για να πραγματοποιήσουν αγορές, καθώς δεν εμπιστεύονται τις ηλεκτρονικές αγορές.
- Το αυξημένο κόστος για τη μεταφορά προϊόντων που είναι αρκετά μεγάλα και βαριά.
- Το κόστος δημιουργίας, συντήρησης και βελτιστοποίησης του ηλεκτρονικού καταστήματος.

3.4 Κατηγορίες Ηλεκτρονικού Εμπορίου

Το ηλεκτρονικό εμπόριο συναντάται σε ποικίλες μορφές. Κάθε μία από τις μορφές αυτές χαρακτηρίζεται από τον ρόλο που κατέχει ο πωλητής, αλλά και ο αγοραστής στη διαδικασία της ανταλλαγής αγαθών. Τέσσερις από τις βασικές κατηγορίες του είναι οι εξής:

3.4.1 Επιχείρηση προς Επιχείρηση - Business - to - Business (B2B)

Η εμφάνιση του ηλεκτρονικού εμπορίου ως μέσο επιχειρηματικής δραστηριότητας έχει δημιουργήσει ένα περιβάλλον στο οποίο οι ανάγκες και οι προσδοκίες των επιχειρήσεων, των πελατών και των καταναλωτών αλλάζουν και εξελίσσονται ταχύτατα.

Το γεγονός αυτό, δημιούργησε έναν επιπλέον ανταγωνισμό μεταξύ των επιχειρήσεων, που απαιτεί από αυτές να βρίσκονται σε εγρήγορση, προκειμένου να μπορέσουν να καρπωθούν τα πολύπλευρα πλεονεκτήματα που τους προσφέρει.

Τα δύο είδη ηλεκτρονικού εμπορίου που χρήζει να ακολουθούν τις μεταβολές αυτές, είναι το λεγόμενο B2B (Business -to -Business) και το αναφερόμενο ως B2C (Business –to Consumer). Το δεύτερο είδος παρουσιάζεται στην αμέσως επόμενη υποενότητα.

Το πρώτο, αφορά συναλλαγές μεταξύ εταιρειών και χαρακτηρίζεται ως το ηλεκτρονικό χονδρικό εμπόριο. Αναφέρεται στο ηλεκτρονικό δίκτυο που χρησιμοποιούν οι επιχειρήσεις για τον εφοδιασμό και τη διανομή προμηθειών. Ορισμένα από τα οφέλη που προσφέρει το δίκτυο αυτό, είναι ο

αποτελεσματικότερος έλεγχος των αποθεμάτων, η δυνατότητα ολοκληρωμένης και άμεσης πληροφόρησης σχετικά με τα προσφερόμενα προϊόντα, μέσα από μια ιστοσελίδα ή το ηλεκτρονικό ταχυδρομείο, η πραγματοποίηση ηλεκτρονικών πληρωμών και η γενικότερη διευκόλυνση των διαδικαστικών ροών.

Ένα παράδειγμα αυτής της κατηγορίας, είναι οι συναλλαγές που γίνονται για να επιτευχθεί η ολοκληρωτική κατασκευή ενός μοντέλου ηλεκτρονικού υπολογιστή. Οι κατασκευάστριες εταιρείες που θα εξάγουν μαζικά το τελικό προϊόν έχουν συνεργασίες με εταιρείες που τους προμηθεύουν με τα κατάλληλα εξαρτήματα (διάφορα τσιπάκια, πλακέτες όπως είναι η μητρική, επεξεργαστές, κάρτες γραφικών κ.α.) και το κατάλληλο λογισμικό.

Σε αυτή την κατηγορία ηλεκτρονικού εμπορίου, συναντάται και η τεχνολογία της ηλεκτρονικής ανταλλαγής δεδομένων (Electronic Data Interchange - EDI), η οποία αντικατέστησε την παραδοσιακή επικοινωνία, μέσω χρήσιμων για τις επιχειρήσεις εντύπων, με την ηλεκτρονική ανταλλαγή των επιχειρηματικών αυτών εγγράφων. Με τη μετάβαση αυτή, οι επιχειρήσεις απολαμβάνουν σημαντικά οφέλη, όπως είναι το μειωμένο κόστος, η αυξημένη ταχύτητα επεξεργασίας, η μείωση ,ενδεχομένων, ανθρώπινων σφαλμάτων και οι βελτιωμένες σχέσεις με τους επιχειρηματικούς τους εταίρους.

3.4.2 Επιχείρηση προς Καταναλωτή - Business – to - Consumer (B2C)

Η δεύτερη κατηγορία ηλεκτρονικού εμπορίου εστιάζει στις λιανικές πωλήσεις προϊόντων και υπηρεσιών και είναι εξίσου σημαντική με την προαναφερθείσα (B2B). Πραγματοποιείται μεταξύ επιχείρησης και καταναλωτή, για το λόγω αυτό αναφέρεται ως B2C (Business – to Consumer).

Σήμερα ο καταναλωτής έχει πρόσβαση σε μια τεράστια ποικιλία προϊόντων σε δικτυακά καταστήματα, βλέπει, επιλέγει, αν επιθυμεί μπορεί ενίοτε να δοκιμάσει τα προϊόντα (μέσω ειδικών προγραμμάτων), ανακαλύπτει προϊόντα τα οποία δεν θα μπορούσε να βρει εύκολα στη χώρα του, συγκρίνει τιμές και τέλος αγοράζει. Κι όλα αυτά χωρίς να βγει από το σπίτι του, κερδίζοντας πολύτιμο χρόνο και κόπο. Χρησιμοποιώντας το διαδίκτυο οι κατασκευαστές μπορούν να έρθουν σε άμεση επαφή με τους πελάτες τους χωρίς την παρουσία ενδιάμεσων [4].

Μία εφαρμογή αυτού του είδους εμπορίου αποτελεί η οργάνωση ενός ταξιδιού. Ο καταναλωτής μέσα από μία ιστοσελίδα, μπορεί με μερικά μόνο κλικ να δει εικόνες των προορισμών που τον ενδιαφέρουν, να αναζητήσει τα διαθέσιμα καταλύματα της κάθε περιοχής, να συγκρίνει τις κριτικές και τις τιμές τους και να αποφασίσει. Πολλοί είναι οι ιστότοποι που δίνουν την ευκαιρία στον χρήστη να επιλέξει και τον τρόπο μετακίνησής του προς τον επιθυμητό προορισμό.

3.4.3 Κράτος με Επιχείρηση – Government – to - Business (G2B)

Σε μία τρίτη μορφή το ηλεκτρονικό εμπόριο συνδέεται με τις κυβερνήσεις που επιδιώκουν να αυτοματοποιήσουν τις αλληλεπιδράσεις τους με τις επιχειρήσεις. Παρόλο που αυτή η κατηγορία καλείται διακυβέρνηση-προς-επιχείρηση (G2B), η σχέση λειτουργεί με δύο τρόπους: κυβέρνηση-προς-επιχείρηση και επιχείρηση-προς-κυβέρνηση. Έτσι, το G2B αναφέρεται σε δραστηριότητες όπου η κυβέρνηση πωλεί προϊόντα σε επιχειρήσεις ή παρέχει στις επιχειρήσεις υπηρεσίες και αντίστροφα [2].

Οι συναλλαγές των δημόσιων οργανισμών με τις επιχειρήσεις αφορούν συνήθως περιπτώσεις φορολογίας, εισαγωγές-εξαγωγές μέσω τελωνείων, δημόσιες προμήθειες, προηγμένες ηλεκτρονικές υπηρεσίες κ.α..

3.4.4 Κράτος με Καταναλωτή – Government-to-Customer (G2C)

Όπως ορίζουν στο βιβλίο τους οι Turban E. et al. (2015, σελ. 211) η συγκεκριμένη μορφή περιλαμβάνει όλες τις αλληλεπιδράσεις μεταξύ μιας κυβέρνησης και των πολιτών της που πραγματοποιούνται ηλεκτρονικά. Το G2C περιλαμβάνει δεκάδες διαφορετικές πρωτοβουλίες. Η βασική ιδέα είναι η δυνατότητα των πολιτών να αλληλεπιδρούν ηλεκτρονικά με την κυβέρνηση από οπουδήποτε και ανά πάσα στιγμή.

Οι εφαρμογές G2C επιτρέπουν στους πολίτες να κάνουν ερωτήσεις σε κυβερνητικούς οργανισμούς και να λαμβάνουν απαντήσεις, να πληρώνουν φόρους, να λαμβάνουν πληρωμές και έγγραφα που χρειάζονται και να προγραμματίζουν συνεντεύξεις εργασίας όπως και ιατρικά ραντεβού. Πιο συγκεκριμένα, σε πολλές πολιτείες των ΗΠΑ, οι κάτοικοι μπορούν να ανανεώσουν τις άδειες οδήγησης, να πληρώσουν εισιτήρια μετακίνησης και να κάνουν δοκιμές οδήγησης διαδικτυακά.

Οι κυβερνήσεις μπορούν επίσης να διαδίδουν πληροφορίες στο Διαδίκτυο, να παρέχουν κατάρτιση, να βοηθούν τους πολίτες να βρουν απασχόληση, να διαχειρίζονται έρευνες και πολλά ακόμη.

3.4.5 Άλλες κατηγορίες

Άλλες μορφές που μπορεί να έχει το η-εμπόριο είναι συνοπτικά:

- Επιχείρηση με Εργαζομένους – Business -to – Employee (B2E)

Η κατηγορία αυτή σχετίζεται με τους υπαλλήλους της επιχείρησης και λαμβάνει χώρα μέσα από ενδοδίκτυα (Intranet) που διαθέτει η εταιρία. Όσοι είναι μέρος της επιχείρησης έχουν πρόσβαση σε πληροφορίες μιας κοινής βάσης δεδομένων, γεγονός που ωφελεί τον συντονισμό τους και την καλύτερη ενδο-επικοινωνία. Παράδειγμα αυτού του είδους η-εμπορίου αποτελεί η τηλεκπαίδευση. Με αυτόν τον τρόπο οι επιχειρήσεις κρατούν ενημέρους τους υπαλλήλους τους για τις νέες τεχνικές ή τεχνολογίες που προκύπτουν κατά καιρούς και τους εκπαιδεύουν στο πώς να τις χρησιμοποιούν, ώστε να παράγουν τα μέγιστα επιθυμητά αποτελέσματα που θα συμβάλλουν εντέλει στην συνολική ανάπτυξη της εταιρίας.

- Συναλλαγές μεταξύ κρατών – Government – to - Government (G2G)

Η μορφή αυτή αφορά είτε δραστηριότητες ηλεκτρονικού εμπορίου που εκτελούνται στο πλαίσιο μιας κυβέρνησης, ενός έθνους, είτε δραστηριότητες ηλεκτρονικού εμπορίου που διεξάγονται μεταξύ κυβερνήσεων δύο ή περισσότερων κρατών, συμπεριλαμβανομένης της παροχής εξωτερικής βοήθειας.

- Συναλλαγές μεταξύ Καταναλωτών – Consumer – to - Consumer (C2C)

Το είδος αυτό, περιλαμβάνει τις συναλλαγές που γίνονται αποκλειστικά μεταξύ των καταναλωτών. Γενικά αναφέρεται σε πωλήσεις που πραγματοποιούνται άμεσα από καταναλωτές σε άλλους καταναλωτές, όπως αυτοκίνητα, περιουσιακά στοιχεία κτλ.

Σύγχρονο παράδειγμα αποτελεί το Marketplace του Facebook, το οποίο παρουσιάζεται ως ένα εύχρηστο και πρακτικό μέσο για τον χρήστη, ώστε να αγοράζει και να πουλάει προϊόντα στην περιοχή του.

Τέλος, τονίζεται ότι όλες οι κατηγορίες που αναπτύχθηκαν στην ενότητα 3.4 δεν είναι απαραίτητο να υιοθετούνται από τους συμμετέχοντες ρόλους μόνο ως μεμονωμένες, καθώς μπορούν να συνδυαστούν αναμεταξύ τους και να καλύψουν ένα μεγαλύτερο εύρος απαιτήσεων και αναγκών.

3.5 Ορισμός Ηλεκτρονικού Καταστήματος – e-shop

Οι Turban E. et al. (2015, σελ. 61) σημειώνουν στο βιβλίο τους, ότι ο όρος ηλεκτρονικό κατάστημα (e-shop / webstore / storefront) αναφέρεται στον ιστότοπο μιας εταιρείας ή ενός μεμονωμένου πωλητή, μέσω του οποίου πωλούνται προϊόντα και υπηρεσίες. Ενδέχεται να στοχεύει σε μια βιομηχανία, μια τοποθεσία ή μια εξειδικευμένη αγορά. Μπορεί να ανήκει σε έναν κατασκευαστή ή σε έναν έμπορο λιανικής, σε άτομα που πωλούν προϊόντα από το σπίτι τους ή σε άλλους τύπους επιχείρησης.

Ένα ηλεκτρονικό κατάστημα περιλαμβάνει εργαλεία γνωστά ως εμπορικό λογισμικό, που είναι απαραίτητα για τη διεξαγωγή διαδικτυακών πωλήσεων. Τα πιο κοινά εργαλεία είναι ένας

ηλεκτρονικός κατάλογος και μία μηχανή αναζήτησης που βοηθούν τους καταναλωτές να βρουν τα καταχωρημένα προϊόντα. Ένα ηλεκτρονικό καλάθι αγορών για την συγκέντρωση των αντικειμένων ωστόσο μεταβεί για την αγορά των αντικειμένων σε μία πύλη πληρωμής. Και τέλος, ένα κέντρο αποστολής όπου πραγματοποιούνται οι ρυθμίσεις αποστολής και η εξυπηρέτηση πελατών, η οποία περιλαμβάνει πληροφορίες σχετικά με το προϊόν και την εγγύηση του.

Αξίζει να σημειωθεί, ότι εκτός από τα μεμονωμένα ηλεκτρονικά καταστήματα, οι καταναλωτές μπορούν να ψωνίζουν και σε ηλεκτρονικά εμπορικά κέντρα (e-Malls). Η έννοια των e-Malls είναι παρόμοια με αυτή των φυσικών πολυκαταστημάτων, αφού αποτελούν μια ηλεκτρονική σελίδα αγορών όπου πολλά καταστήματα παρουσιάζουν τα προϊόντα τους.

3.6 Διαχείριση γεγονότων

Με τον όρο διαχείριση γεγονότος ορίζεται η διαδικασία της δημιουργίας και της δημόσιας προβολής γεγονότων όπως φεστιβάλ, συναυλίες, συνέδρια, επίσημα πάρτι κ.α. δρόμενα. Σήμερα παρέχονται τέτοια συστήματα όλων των μεγεθών που εξυπηρετούν από τους Ολυμπιακούς Αγώνες έως επαγγελματικές συναντήσεις μικρών επιχειρήσεων. Μάλιστα υπάρχουν και ακαδημαϊκά ιδρύματα τα οποία προσφέρουν πιστοποίηση αλλά και προπτυχιακά και μεταπτυχιακά προγράμματα πάνω στο αντικείμενο.

Ενδιαφέρον όμως παρουσιάζουν πρόσφατες δημοσιεύσεις¹⁴ όπου εντάσσουν τον κλάδο στην κατηγορία καριέρας με τους έχοντες το περισσότερο στρες. Αυτό οφείλεται σε πολλούς παράγοντες όπως η πολυπλοκότητα, ο εκτεταμένος όγκος γνώσεων και το ταχέως αναπτυσσόμενο περιβάλλον.

3.7 Συστήματα κρατήσεων εισιτηρίων

Τα συστήματα κράτησης εισιτηρίων με υπολογιστή, ή αλλιώς κεντρικά συστήματα κρατήσεων (CRS), είναι ηλεκτρονικά συστήματα που χρησιμοποιούνται για την αποθήκευση και ανάκτηση πληροφοριών και τη διεξαγωγή συναλλαγών που σχετίζονται με αεροπορικά ταξίδια, ξενοδοχεία, ενοικιάσεις αυτοκινήτων ή άλλες δραστηριότητες.

Αρχικά σχεδιάστηκαν και λειτουργούσαν από αεροπορικές εταιρείες, όμως επεκτάθηκαν αργότερα για χρήση και από ταξιδιωτικά γραφεία. Μάλιστα στις μέρες μας οι εταιρίες χρησιμοποιούν εξωτερικούς συνεργάτες για να χειριστούν τις κρατήσεις και την αγορά εισιτηρίων μέσω παγκόσμιων συστημάτων διανομής (global distribution systems, GDS) όπου φιλοξενούνται πολλές εταιρίες ταυτόχρονα.

Η προέλευσή τους ξεκινά από την American Airlines μόλις από το 1946 με το πειραματικό ηλεκτρομηχανικό σύστημα Reservisor¹⁵. Η επόμενη έκδοσή του που προσέφερε προσωρινή αποθήκευση βασισμένη σε μαγνητικό τύμπανο αποδείχθηκε επιτυχημένη και σύντομα άρχισε να χρησιμοποιείται από περισσότερες αεροπορικές εταιρίες καθώς και από άλλες όπως η Goodyear για την διαχείριση των αποθεμάτων της.

Το κύριο πρόβλημά τους ήταν η απομακρυσμένη πρόσβαση καθώς η διαδικασία δεν ήταν αυτόματη. Η ομάδα ελέγχου του Reservisor έπρεπε να διασταυρώνει τηλεφωνικά τις διαθέσιμες θέσεις με τον πράκτορα εισιτηρίων και να του μεταβιβάσει τα αποτελέσματα. Η πρώτη ιδέα ενός αυτοματοποιημένου συστήματος κρατήσεων εμφανίστηκε το 1959 και το SABRE (Semi-automated Business Research Environment) έκανε την εμφάνισή του ένα χρόνο αργότερα. Μέχρι τη στιγμή που

¹⁴ <https://www.cbsnews.com/news/10-most-and-least-stressful-jobs-in-america/>
<https://www.citmagazine.com/article/1449482/events-coordinator-fifth-stressful-job-why-matters>

¹⁵ <https://en.wikipedia.org/wiki/Reservisor>

το δίκτυο ολοκληρώθηκε τον Δεκέμβριο του 1964, ήταν το μεγαλύτερο σύστημα επεξεργασίας πολιτικών δεδομένων στον κόσμο.

Σήμερα η μοντέρνα μορφή των GDS είναι αυτή μέσω διαδικτύου όπου στις εφαρμογές ο ενδιαφερόμενος μπορεί να επιλέξει ανάμεσα σε μεγάλη γκάμα υπηρεσιών κρατήσεων και μπορεί να πληρώσει ηλεκτρονικά. Ένα τέτοιο σύστημα στην Ελλάδα είναι το viva.gr το οποίο παρέχει ολοκληρωμένο πακέτο υπηρεσιών όπως ακτοπλοϊκά εισιτήρια, εισιτήρια για συναυλίες, θεατρικές παραστάσεις κ.α.

3.8 Επίλογος

Όπως είδαμε σε αυτό το κεφάλαιο, ο τομέας του ηλεκτρονικού εμπορίου έχει συναντήσει μεγάλη πρόοδο και αυτοματοποιείται όλο και περισσότερο όσο εξελίσσεται και η τεχνολογία. Τελειώνοντας το κεφάλαιο αυτό ολοκληρώνουμε και την εικόνα μας για το θεωρητικό μέρος της εργασίας. Έχουμε δει μέχρι στιγμής τις γλώσσες και τις τεχνολογίες στις οποίες βασίστηκε το `ticket manager` καθώς και τι σημαίνει η διαχείριση κράτησης εισιτηρίων και τους τρόπους που πραγματοποιείται. Στο κεφάλαιο που ακολουθεί θα εξετάσουμε αναλυτικά όλες τις λειτουργίες του συστήματος που δεν έχουμε δει ως τώρα και θα μελετήσουμε τον κώδικά τους.

Κεφάλαιο 4: Υλοποίηση Ticket Manager

4.1 Εισαγωγή

Προκειμένου να υλοποιηθούν οι διαδικασίες του συστήματος κρατήσεων της παρούσας εργασίας ελήφθησαν υπόψη οι προαναφερθείσες τεχνολογίες όπως και ρεαλιστικά παραδείγματα σύγχρονων συστημάτων τέτοιου τύπου ώστε να είναι όσο το δυνατόν πιο κοντά σε αυτά. Στο Κεφάλαιο 1: είδαμε αναλυτικά τα στοιχεία και τις οντότητες που απαρτίζουν το σύστημά μας. Ενώ στο Κεφάλαιο 2: μελετώντας τις τεχνολογίες υλοποίησης είδαμε επιμέρους λειτουργίες της εφαρμογής όπως την σύνδεση με την βάση δεδομένων, την δομή της εμφάνισης και κάποιες λειτουργίες των γλωσσών JavaScript και PHP. Στο κεφάλαιο αυτό θα εξετάσουμε τις πιο ειδικές λειτουργίες του συστήματος όπως η αναζήτηση και η κράτηση.

4.2 Λειτουργία αρχικής σελίδας index.php

Η αρχική σελίδα διαθέτει μια συνάρτηση εμφάνισης γεγονότων τα οποία έπονται της παρούσας ημερομηνίας. Για την επίτευξη του αποτελέσματος αρχικά παίρνουμε την τρέχουσα ημερομηνία κατά την φόρτωση της σελίδας και στη συνέχεια εκτελούμε ερωτήματα SELECT στην βάση δεδομένων. Το πρώτο ερώτημα αφορά τον πίνακα event όπου ζητάμε τα 6 δρόμενα (LIMIT 6) με ημερομηνία >= της παρούσας (μεταβλητή \$now). Στην συνέχεια για κάθε αποτέλεσμα (εντολή while) ανακτούμε από τον πίνακα venue το όνομα τις τοποθεσίας που θα λάβει χώρα το γεγονός και στην ίδια επανάληψη συντάσσουμε την δόμηση της εμφάνισης στην μεταβλητή \$output. Τελικά επιστρέφεται η τιμή της μεταβλητής \$output με την εντολή return της PHP στο σημείο όπου κλήθηκε η συνάρτηση, εντός του διαχωριστή div class="w3-row-padding".

```

17 <!-- Fetching last posted events. -->
18 <div class="w3-row-padding">
19   <?php echo lastposted(); ?>
20 </div>
21
22 <?php }
23
24 function lastposted()
25 {
26   global $dbconn, $row, $sql, $stmt, $result, $now;
27   $output = '';
28   $sql = "SELECT * FROM event WHERE StartDateTime>='".$now.'" ORDER BY StartDateTime ASC LIMIT 6";
29   $result = mysqli_query($dbconn, $sql);
30
31   while ($row = mysqli_fetch_assoc($result))
32   {
33     $sql = "SELECT * FROM venue WHERE Id = '".$row['VenueId']."'";
34     $result1 = mysqli_query($dbconn, $sql);
35     $row1 = mysqli_fetch_array($result1);
36
37     // Get event start date and time.
38     $sqlstartdate = strtotime($row['StartDateTime']);
39     $getstartdate = date('Y-m-d', $sqlstartdate);
40     $startdate = date("d M Y", $sqlstartdate);
41     $starttime = date("H:i", $sqlstartdate);
42
43     $output .= '<div class="w3-third w3-container w3-margin-bottom">';
44     $output .= '<div class="w3-card w3-padding-small">';
45     $output .= '<a class="title" href="event.php?eventId=' . $row['Id'] . '&familyeventId=' . $row['FamilyEventId'] . '&date=' . $getstartdate . '>';
46     $output .= '<div class="w3-display-container">';
47     $output .= '<b>' . $row["Name"] . '</b></h4>';
50     $output .= '<h5><i class="fa fa-globe" aria-hidden="true"></i> ' . $row1["Name"] . '</h5>';
51     $output .= '<h5><i class="fa fa-calendar" aria-hidden="true"></i> ' . $startdate . '</h5>';
52     $output .= '<h5><i class="fa fa-clock-o" aria-hidden="true"></i> ' . $starttime . '</h5><br/>';
53     $output .= '</a></div>';
54     $output .= '</div>';
55   }
56   return $output;
57 }
58 ?>

```

Εικόνα 18: Συνάρτηση lastposted() στην index.php

4.3 Λειτουργίες της σελίδας αναζήτησης select.php

Στην ενότητα 2.4 μελετήσαμε τις λειτουργίες της επαναφοράς ημερομηνίας και της αναζήτησης με jQuery. Στην δεύτερη λειτουργία τα δεδομένα των πεδίων εισαγωγής μεταφέρονται στην σελίδα λειτουργίας do_search.php στον κατάλογο functions. Σε αυτόν τον κατάλογο φιλοξενούνται οι σελίδες επεξεργασίας στον διακομιστή. Στην σελίδα do_search υπάρχουν οι συνθήκες παραμετροποίησης των μεταβλητών που υποβλήθηκαν και είναι: το αναγνωριστικό πόλης (cityid), το αναγνωριστικό τύπου (typeid) το οποίο έχει μορφή πίνακα, και η ημερομηνία εκκίνησης (startdate). Η μεταβλητή \$typecount μετρά τον αριθμό των επιλεγμένων τύπων.

```

10  /* Get City Id if is set. Else set to empty String */
11  if (isset($_POST['cityid']) && $_POST['cityid'] != '')
12  {
13      $cityid = $_POST['cityid'];
14  }
15  else
16  {
17      $cityid = '';
18  }
19  /* Get Event Type Id if is set. Else set to empty String */
20  if (isset($_POST['typeid']) && $_POST['typeid'] != '')
21  {
22      $typeid = $_POST['typeid'];
23      $typecount = count($typeid);
24  }
25  else
26  {
27      $typeid = '';
28  }
29  /* Get event start date and time. If set */
30  if (isset($_POST['startdate']) && $_POST['startdate'] != '')
31  {
32      $startdate = $_POST['startdate'];
33  }
34  else
35  {
36      $startdate = '';
37  }

```

Εικόνα 19: Απόδοση τιμών στις μεταβλητές στην do_search.php

Στη συνέχεια ακολουθούν οι συνθήκες ελέγχου και εκτέλεσης ερωτήματος στην βάση κατάλληλα. Οι περιπτώσεις που καλύπτονται είναι οκτώ όσες δηλαδή και ο συνδυασμός των τριών περιπτώσεων υποβολής πεδίων (2³). Αυτές οι περιπτώσεις ενδεικτικά είναι:

- Κανένα υποβληθέν πεδίο (1)
- Ένα υποβληθέν πεδίο (3)
- Συνδυασμός όλων των πεδίων εφόσον υποβλήθηκαν (4)

Κατά την πρώτη περίπτωση το ερώτημα προς την βάση και συγκεκριμένα τον πίνακα event είναι απλό και περιλαμβάνει τα 50 αποτελέσματα από την παρούσα ημερομηνία και έπειτα. Στην δεύτερη περίπτωση τα ερωτήματα βασίζονται στο ποιο πεδίο είναι συμπληρωμένο και βάση αυτού ορίζονται τα ερωτήματα. Η πιο σύνθετη περίπτωση είναι αυτή των τύπων γεγονότων όπου ανάλογα τον αριθμό των στοιχείων διαμορφώνεται και το όριο των αποτελεσμάτων για λόγους απόδοσης του συστήματος. Πιο συγκεκριμένα εάν επιλέχθηκε μόνο μία κατηγορία τα αποτελέσματα δεν θα έχουν όριο 50, εάν επιλέχθηκαν έως τρεις κατηγορίες τα αποτελέσματα θα είναι έως 18 για την κάθε κατηγορία, ενώ στην περίπτωση τεσσάρων και περισσότερων κατηγοριών τα αποτελέσματα δεν ξεπερνούν τα 8 για κάθε κατηγορία.

Υλοποίηση Ticket Manager

```
39 // Filter search.
40 // case 0 - 0 - 0
41 if ($cityid == '' && empty($typeid) && $startdate == '') {
42     $sql = "SELECT * FROM event WHERE StartDateTime>='". $now.'" LIMIT 50";
43     $result = mysqli_query($dbconn, $sql);
44     while ($row = mysqli_fetch_array($result))
45     {
46         $filteredarray[] = $row;
47     }
48     prepareoutput($filteredarray);
49     echo $output;
50 }
51 // case 0 - 0 - 1
52 if ($cityid == '' && empty($typeid) && $startdate != '') {
53     $sql = "SELECT * FROM event WHERE StartDateTime>='". $startdate.'" LIMIT 50";
54     $result = mysqli_query($dbconn, $sql);
55     while ($row = mysqli_fetch_array($result))
56     {
57         $filteredarray[] = $row;
58     }
59     prepareoutput($filteredarray);
60     echo $output;
61 }
62 //case 0 - 1 - 0
63 if ($cityid == '' && !empty($typeid) && $startdate == '') {
64     foreach ($typeid as $key => $value) {
65         if ($typecount < 2) {
66             $sql = "SELECT * FROM event WHERE (EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 50";
67         }
68         elseif ($typecount < 4) {
69             $sql = "SELECT * FROM event WHERE (EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 18";
70         }
71         else {
72             $sql = "SELECT * FROM event WHERE (EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 8";
73         }
74         $result = mysqli_query($dbconn, $sql);
75         while ($row = mysqli_fetch_array($result))
76         {
77             $filteredarray[] = $row;
78         }
79     }
80     prepareoutput($filteredarray);
81     echo $output;
```

Εικόνα 20: Περιπτώσεις κανενός η ενός υποβληθέν πεδίου στην do_search.php

```
123 //case 1 - 0 - 1
124 if ($cityid != '' && empty($typeid) && $startdate != '') {
125     $sql = "SELECT * FROM venue WHERE CityId = '". $cityid ."'";
126     $result = mysqli_query($dbconn, $sql);
127     while($row = mysqli_fetch_array($result))
128     {
129         $sql = "SELECT * FROM event WHERE (VenueId = '". $row['Id'].'" AND StartDateTime >= '". $startdate.'" ) LIMIT 50";
130         $result1 = mysqli_query($dbconn, $sql);
131         while ($row1 = mysqli_fetch_array($result1))
132         {
133             $filteredarray[] = $row1;
134         }
135     }
136     prepareoutput($filteredarray);
137     echo $output;
138 }
139 // case 1 - 1 - 0
140 if ($cityid != '' && !empty($typeid) && $startdate == '')
141 {
142     $sql = "SELECT * FROM venue WHERE CityId = '". $cityid ."'";
143     $result = mysqli_query($dbconn, $sql);
144     while($row = mysqli_fetch_array($result))
145     {
146         $venueid = $row['Id'];
147         foreach ($typeid as $key => $value)
148         {
149             if ($typecount < 2) {
150                 $sql = "SELECT * FROM event WHERE (VenueId = '". $venueid.'" AND EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 50";
151             }
152             elseif ($typecount < 4) {
153                 $sql = "SELECT * FROM event WHERE (VenueId = '". $venueid.'" AND EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 18";
154             }
155             else {
156                 $sql = "SELECT * FROM event WHERE (VenueId = '". $venueid.'" AND EventTypeId = '". $value.'" AND StartDateTime >= '". $now.'" ) LIMIT 8";
157             }
158             $result1 = mysqli_query($dbconn, $sql);
159             while ($row1 = mysqli_fetch_array($result1))
160             {
161                 $filteredarray[] = $row1;
162             }
163         }
164     }
165 }
166 prepareoutput($filteredarray);
167 echo $output;
168 }
```

Εικόνα 21: Περιπτώσεις συνδυασμού δύο υποβληθέντων πεδίων στην do_search.php

Υλοποίηση Ticket Manager

Ακόμα η ίδια σελίδα περιλαμβάνει και τρεις αλληλεξαρτώμενες συναρτήσεις. Αυτές είναι η `prepareoutput()`, η `orderbydate()` και η `output()`. Κατά την πρώτη και εναρκτήρια συνάρτηση ελέγχεται εάν τα ερωτήματα στην βάση φέρουν η όχι αποτελέσματα. Στην περίπτωση που δεν επεστράφησαν αποτελέσματα καλείτε η συνάρτηση `output()` με κενό όρισμα και τυπώνει το κατάλληλο μήνυμα. Στην περίπτωση ύπαρξης αποτελέσματος καλείται η συνάρτηση `orderbydate()` για τον πίνακα αποτελεσμάτων η οποία ταξινομεί τα αποτελέσματα ανά αύξουσα σειρά ημερομηνίας. Αυτό επιτυγχάνεται μέσω της μεθόδου `usort()` της PHP. Στη συνέχεια καλείτε επαναληπτικά η `output()` για κάθε στοιχείο του πίνακα, με όρισμα την παρούσα γραμμή, όπου συντάσσεται η μεταβλητή `$output` κατάλληλα για να δημιουργηθεί η κάρτα παρουσίασης του γεγονότος.

```
200 function prepareoutput($filteredarray)
201 {
202     if (empty($filteredarray))
203     {
204         output("");
205     }
206     else
207     {
208         usort($filteredarray, 'orderbydate');
209         foreach ($filteredarray as $data)
210         {
211             output($data);
212         }
213     }
214 }
215
216 function orderbydate($a, $b)
217 {
218     return $a['StartDate'] <=> $b['StartDate'];
219 }
220
221 function output($qrow)
222 {
223     global $output, $dbconn;
224     if (empty($qrow))
225     {
226         // $output .= '<div class="w3-third w3-container w3-margin-bottom">';
227         $output .= '<div class="center w3-padding">';
228         $output .= '<p class="w3-teal w3-xxlarge">Sorry!<br/>';
229         $output .= 'No results matching your criteria...</p>';
230         $output .= '</div>';
231     }
232     else
233     {
234         $sql = "SELECT * FROM venue WHERE Id = '" . $qrow['VenueId'] . "'";
235         $result1 = mysqli_query($dbconn, $sql);
236         if (!$result1)
237         {
238             printf("Error: %s\n", mysqli_error($dbconn));
239             exit();
240         }
241         $row1 = mysqli_fetch_array($result1);
242
243         // Get event start date and time.
244         $sqlstartdatetime = strtotime($qrow['StartDateTime']);
245         $startdate = date("d M Y", $sqlstartdatetime);
246         $getstartdate = date("Y-m-d", $sqlstartdatetime);
247         $starttime = date("H:i", $sqlstartdatetime);
248
249         $output .= '<div class="w3-third w3-container w3-margin-bottom">';
250         $output .= '<div class="w3-card w3-padding-small">';
251         $output .= '<a class="title" href="event.php?eventid=' . $qrow['Id'] . '&familyeventid=' . $qrow['FamilyEventId'] . '&date=' . $getstartdate . "'>';
252         $output .= '<div class="w3-display-container">';
253         $output .= '';
254         $output .= '</div>';
255         $output .= '<h4 style="overflow:hidden;white-space:nowrap;"><b>' . $qrow["Name"] . '</b></h4>';
256         $output .= '<h5><i class="fa fa-globe" aria-hidden="true"></i> ' . $row1["Name"] . '</h5>';
257         $output .= '<h5><i class="fa fa-calendar" aria-hidden="true"></i> ' . $startdate . '</h5>';
258         $output .= '<h5><i class="fa fa-clock-o" aria-hidden="true"></i> ' . $starttime . '</h5><br/>';
259         $output .= '</a></div>';
260         $output .= '</div>';
261     }
262 }
```

Εικόνα 22: Οι συναρτήσεις στην `do_search.php`

Το τελικό αποτέλεσμα που θα επιστραφεί στην `select.php` βρίσκεται στην μεταβλητή `$output` και επιστρέφεται με την μέθοδο `echo` στο τέλος κάθε συνθήκης. Όπως αναφέραμε η κλήση στην `select.php` γίνεται ασύγχρονα οπότε το μόνο μέρος που θα επηρεαστεί μετά την απόκριση είναι τα περιεχόμενα του διαχωριστή με αναγνωριστικό `"filterresults"` το οποίο θα γεμίσει με τα αποτελέσματα. Άλλες λειτουργίες της είναι το γέμισμα των στοιχείων τύπου `select` της HTML με επιλογές (`options`). Αυτό συμβαίνει με τα κατάλληλα ερωτήματα SQL ως προς τους πίνακες `city` και `eventtype` με τις συναρτήσεις `fillcities()` και `filltypes()` αντίστοιχα.

```

60 function fillcities()
61 {
62     global $dbconn;
63     $output = '';
64     $sql = "SELECT * FROM city";
65     $result = mysqli_query($dbconn, $sql);
66     while($row = mysqli_fetch_array($result))
67     {
68         $output .= '<option value="'. $row["Id"]. '">' . $row["Name"] . '</option>';
69     }
70     return $output;
71 }
72 function filltypes()
73 {
74     global $dbconn;
75     $output = '';
76     $sql = "SELECT * FROM eventtype";
77     $result = mysqli_query($dbconn, $sql);
78     while($row = mysqli_fetch_array($result))
79     {
80         $output .= '<option value="'. $row["Id"]. '">' . $row["Name"] . '</option>';
81     }
82     return $output;
83 }

```

Εικόνα 23: Γέμισμα επιλογών πόλης και τύπου στην select.php

4.4 Λειτουργίες στην σελίδα κράτησης event.php

Η σελίδα κράτησης αποτελεί μια από τις σελίδες με τις περισσότερες λειτουργίες και για τον λόγο αυτό αποτελείται εξολοκλήρου από κώδικα PHP, εκτός της λειτουργίας αναδυόμενου παραθύρου που είναι συνδυασμός HTML και JavaScript. Αυτή η λειτουργία είναι παρόμοια με αυτήν που είδαμε στην υποενότητα 2.4.1 και την Εικόνα 6. Η λειτουργία διαμόρφωσης του αναδυόμενου παραθύρου (popup) κράτησης θέσης makemodal() και η λειτουργία προσθήκης στο καλάθι addtocart() είναι από τις πιο βασικές λειτουργίες της σελίδας. Κατά την πρώτη ελέγχεται αρχικά εάν υπάρχει το παράθυρο σαν οντότητα και το αφαιρεί από τα περιεχόμενα της σελίδας ώστε να το δημιουργήσει εκ νέου κατάλληλα. Στο ίδιο σενάριο γίνεται μία σύγχρονη κλήση στην seatselect.php, στον κατάλογο functions, ώστε να επιστραφούν οι λεπτομέρειες για την επιλεγμένη κράτηση και να ενσωματωθούν στο αναδυόμενο παράθυρο. Κατά την λειτουργία προσθήκης στο καλάθι η συνάρτηση δέχεται ως όρισμα την σειρά του πίνακα αποτελεσμάτων στην οποία έγινε το κλικ στο κουμπί «ΚΡΑΤΗΣΗ». Στη συνέχεια γίνεται μία ασύγχρονη κλήση στην σελίδα do_register.php, που βρίσκεται και αυτή στον κατάλογο functions, ώστε να πραγματοποιηθεί η κράτηση θέσης, μετά την επιλογή θέσης με το κουμπί «Επιλογή» του παραθύρου. Τελικά εφόσον το αποτέλεσμα ήταν επιτυχές εμφανίζεται παράθυρο τύπου alert με κατάλληλο μήνυμα και η σελίδα event.php επαναφορτώνει.

Οι λειτουργίες PHP της σελίδας ξεκινούν στην αρχή της και συνεχίζουν στο τέλος με τις κατάλληλες συναρτήσεις. Αρχικά γίνεται ο έλεγχος πρόσβασης, δηλαδή βεβαιωνόμαστε ότι έχει εισέλθει ο χρήστης στο σύστημα και έπειτα ότι έχουν οριστεί όλα τα απαραίτητα πεδία της μεθόδου GET. Εφόσον κάτι πάει στραβά γίνεται ανακατεύθυνση στην αρχική σελίδα. Αλλιώς ξεκινά η δόμηση των περιεχομένων της που περιλαμβάνει το κομμάτι οπτικής εμφάνισης και στη συνέχεια τον πίνακα αποτελεσμάτων ερωτήσεων query σε μορφή πίνακα. Για την διαμόρφωση αυτή χρησιμοποιούνται οι συναρτήσεις makeheader() και output().

Υλοποίηση Ticket Manager

```
85 <script>
86 function makemodal($eventid, $date, $eventname, $venue)
87 {
88     var eventid = $eventid;
89     var date = $date;
90     var eventname = $eventname;
91     var venue = $venue;
92     var exmodal = document.getElementById('myModal');
93     var modaldiv = document.getElementById('w3-main');
94     modaldiv.removeChild(exmodal);
95     var modal = '<div id="myModal" class="modal"><div class="modal-content">';
96     modal += '<div class="modal-header">';
97     modal += '<span class="close">&times;</span><h2>Επιλογή θέσης</h2>';
98     modal += '</div><div class="modal-body"><p>'+eventid+' - '+eventname+' - '+date+' - '+venue+'</p>';
99
100     $.ajax({
101         url: 'functions/seatsselect.php',
102         type: "POST",
103         data: {eventid:eventid},
104         async:false,
105         success: function (data) {
106             modal +=data;
107         }
108     });
109     modal += '</div><div class="modal-footer"><h3>Για οποιοδήποτε πρόβλημα με την κρατηση σας επικοινωνήστε στο email: support@ticketmanager.com</h3>';
110     modal += '</div></div>';
111     document.body.innerHTML += modal;
112 }
113 </script>
114 <script>
115 function addtocart(i)
116 {
117     var seatid = $('#seat'+i).val();
118     var eventid = $('#eventid').val();
119     $.ajax({
120         url:"functions/do_register.php",
121         method:"POST",
122         data:{seatid:seatid,eventid:eventid},
123         success:function(data){
124             alert("Η κράτηση θέσης ήταν επιτυχής. Επισκευτείτε το καλάθι για να ολοκληρώσετε την αγορά σας.");
125             location.reload();
126         }
127     });
128 }
129 </script>
```

Εικόνα 24: λειτουργίες JS στην event.php

```
127 function makeheader($qrow)
128 {
129     global $output, $familyeventid, $selecteddate, $dbconn;
130     $output .= '<div class="center w3-container w3-margin">';
131     $output .= '<h1> '.$qrow['Name'].' </h1>';
132     $output .= '';
133     $output .= '</div>';
134     $outputdate = date('d-m-Y', strtotime($selecteddate));
135     $output .= '<div class="center w3-container w3-margin">';
136     $output .= '<h5>Βλέπετε αποτελέσματα για "'.$outputdate.'" και μετά.</h5>';
137     $output .= '</div>';
138 }
139
140 function output($qrow)
141 {
142     global $output, $eventid, $familyeventid, $dbconn, $selecteddate;
143     $sqlstartdatetime = strtotime($qrow['StartDateTime']);
144     $startdate = date("D d M Y", $sqlstartdatetime);
145     $starttime = date("H:i", $sqlstartdatetime);
146     $rowdate = date('Y-m-d', $sqlstartdatetime);
147
148     if ($rowdate == $selecteddate && $qrow['Id'] == $eventid)
149     {
150         $output .= '<tr style="border:2px solid black">';
151     }
152     else
153     {
154         $output .= '<tr>';
155     }
156
157     $output .= '<td> '.$qrow['Id'].'</td>';
158     $sql = "SELECT Name FROM venue WHERE Id='".$qrow['VenueId']."'";
159     $result1 = mysqli_query($dbconn, $sql);
160     $row1 = mysqli_fetch_assoc($result1);
161
162     $output .= '<td> '.$startdate.' ';
163     $output .= '<td> '.$starttime.'</td>';
164     $output .= '<td> '.$qrow['Name'].'</td>';
165     $output .= '<td> '.$row1['Name'].'</td>';
166     if ($qrow['Price'] > 0)
167     {
168         $output .= '<td> '.$qrow['Price'].'</td>';
169     }
170     else
171     {
172         $output .= '<td>Free</td>';
173     }
174
175     $output .= '<td><input id="seatreservation" type="button" value="ΚΡΑΤΗΣΗ" class="w3-btn w3-green w3-btn w3-round-small w3-margin-left"></input></td>';
176     $output .= '</tr>';
177 }
```

Εικόνα 25: Λειτουργίες PHP στην event.php

Υλοποίηση Ticket Manager

Οι βοηθητική λειτουργία στην σελίδα `seatsselect.php` αναλαμβάνει να διαμορφώσει το περιεχόμενο του παραθύρου κράτησης θέσης. Για να το πετύχει συσχετίζει το γεγονός με τις επιλεγμένες τοποθεσίες και υποτοποθεσίες με κατάλληλα ερωτήματα SQL. Αρχικά εξετάζει για ποιο γεγονός έγινε η κράτηση και ενημερώνει την μεταβλητή `$venue` με την τοποθεσία. Στη συνέχεια μέσω της τοποθεσίας τυπώνει τα κατάλληλα στοιχεία όπως όνομα υποπεριοχής και περιγραφή της, και τις διαθέσιμες θέσεις σε κάθε μια από αυτές. Στη συνέχεια φορτώνει τον κώδικα εμφάνισης στην μεταβλητή `$output` που επιστρέφεται με την εντολή `echo`.

```
seatsselect.php
1 <?php
2 session_start();
3 require 'dbconn.php';
4
5 $eventid = $_POST['eventid'];
6 $output = '';
7
8 $sql = "SELECT * FROM event WHERE Id='".$eventid."'";
9 $result = mysqli_query($dbconn, $sql);
10 $row = mysqli_fetch_assoc($result);
11
12 $venueid = $row['VenueId'];
13 $i = 0;
14
15 $sql = "SELECT * FROM subarea WHERE VenueId='".$venueid."'";
16 $result = mysqli_query($dbconn, $sql);
17 while($row = mysqli_fetch_assoc($result))
18 {
19     $sql = "SELECT * FROM seat WHERE SubAreaId = '" . $row['Id'] . "'";
20     $result1 = mysqli_query($dbconn, $sql);
21     $output .= '<div class="w3-container w3-margin">';
22     $output .= '<p>'. $row['AreaName'] . ' (' . $row['Desc'] . ')</p>';
23     $output .= '<div class="w3-container">';
24     $output .= '<select id="seat"'. $i . ">";
25     while($row1 = mysqli_fetch_array($result1))
26     {
27         if ($row1['Available']!=1) {
28             $output .= '<option value="' . $row1["Id"] . '">' . $row1["Name"] . '</option>';
29         }
30     }
31     $output .= '</select>';
32     $output .= '<button class="w3-btn w3-teal w3-btn w3-padding w3-round w3-margin-left" id="seat-submit" onclick="addtocart(' . $i . ')>Επιλογή</button>';
33     $output .= '</div></div>';
34     $output .= '<input id="eventid" style="display: none;" value="' . $eventid . '"></input>';
35     $i++;
36 }
37
38 echo $output;
```

Εικόνα 26: Η σελίδα `seatsselect.php`

Η λειτουργία της άλλης βοηθητικής σελίδας `do_search.php` είναι διπλή. Αρχικά δημιουργεί ή γεμίζει το καλάθι αγορών ανάλογα με το αν έχει ήδη στοιχεία η όχι, ενώ στη συνέχεια εκτελεί τις λειτουργίες κράτησης. Οι τελευταίες είναι η ενημέρωση της διαθέσιμης θέσης ως μη διαθέσιμη στον πίνακα θέσεων (εντολή `UPDATE`) και έπειτα η εισαγωγή νέας εγγραφή κράτησης στον πίνακα κρατήσεων (εντολή `INSERT`).

```
do_register.php
1 <?php
2 session_start();
3 require 'dbconn.php';
4
5 $seatid = $_POST['seatid'];
6 $eventid = $_POST['eventid'];
7 $userid = $_SESSION['userid'];
8
9 $sql = "SELECT * FROM event WHERE Id='".$eventid."'";
10 $result = mysqli_query($dbconn, $sql);
11 $row = mysqli_fetch_assoc($result);
12 $eventname = $row['Name'];
13 $eventdate = $row['StartDateTime'];
14
15 $sql = "SELECT * FROM venue WHERE Id='".$row['VenueId']."'";
16 $result1 = mysqli_query($dbconn, $sql);
17 $row1 = mysqli_fetch_assoc($result1);
18 $venueName = $row1['Name'];
19
20 $cartArray = array(array('EventId'=>$eventid, 'Event'=>$eventname, 'Date'=>$eventdate, 'Location'=>$venueName, 'SeatId'=>$seatid));
21
22 if (empty($_SESSION['cart']))
23 {
24     $_SESSION['cart'] = $cartArray;
25 }
26 else
27 {
28     $_SESSION['cart'] = array_merge($_SESSION['cart'], $cartArray);
29 }
30
31 $sql = "UPDATE seat SET Available = 0 WHERE Id='".$seatid."'";
32 $result = mysqli_query($dbconn, $sql);
33
34 $sql = "INSERT INTO reservation (UserId, SeatId, EventDate, Duration, EventId)
35 VALUES ('" . $userid . "','" . $seatid . "','" . $eventdate . "','" . $eventid . "')";
36 $result = mysqli_query($dbconn, $sql);
37 ?>
```

Εικόνα 27: Η σελίδα `do_register.php`

4.5 Οι λειτουργίες εγγραφής και εισόδου χρήστη

Οι λειτουργίες στις σελίδες αυτές είναι να επεξεργάζονται τις πληροφορίες που έδωσε ο χρήστης μέσω των πεδίων των φορμών στις login.php και signup.php. Κατά την εγγραφή, δηλαδή εφόσον πατηθεί το κουμπί «Εγγραφή» στην σελίδα signup.php, η σελίδα do_signup δέχεται και εξετάζει τα δεδομένα με κατάλληλους ελέγχους. Αυτοί είναι:

- Έλεγχος κενών πεδίων
- Έλεγχος email και ονόματος χρήστη (έγκυρη διεύθυνση, λατινικά και αριθμοί)
- Οι προηγούμενοι (email και username) ξεχωριστά
- Κωδικός και επαλήθευση.

Σε κάθε περίπτωση όταν δεν ικανοποιηθούν οι συνθήκες καλείται η συνάρτηση errorredirect() με ορίσματα τις κατάλληλες παραμέτρους. Αυτή η συνάρτηση μας επιστρέφει στην σελίδα εγγραφής ορίζοντας με την μέθοδο GET τους τύπους σφάλματος εγγραφής. Η signup.php εμφανίζει κατάλληλο μήνυμα ανάλογα τα ορίσματα. Στη περίπτωση ορθής εισαγωγής στην φόρμα ξεκινούν οι έλεγχοι για διπλοεγγραφές όπως είδαμε στην ενότητα 1.3.1. με την ικανοποίηση των συνθηκών ξεκινά η διαδικασία της δημιουργίας χρήστη στον πίνακα users. Αφού προετοιμάσουμε τον κωδικό με την συνάρτηση κατακερματισμού της PHP password_hash() παίρνουμε την τιμή του τελευταίου κατά σειρά αναγνωριστικού χρήστη (Id) για να μπορέσουμε να το αυξήσουμε κατά 1. Αυτό θα γινόταν αυτόματα μέσω της SQL εάν είχαμε επιλέξει για τύπο της στήλης Id ακέραιο αριθμό και είχαμε θέσει εξορισμού την δυνατότητα Auto Increment. Επειδή όμως η στήλη είναι τύπου varchar είμαστε υποχρεωμένοι να ακολουθήσουμε την παραπάνω διαδικασία ώστε να αυξήσουμε στην μεριά της εφαρμογής το αναγνωριστικό. Έπειτα εισάγουμε την εγγραφή με την εντολή INSERT και εξετάζουμε το αποτέλεσμα για να κατευθύνουμε τον χρήστη ανάλογα.

```

6 // Access control.
7 if (isset($_POST['signup-submit'])) {
8
9     require 'dbconn.php';
10    // Variables initialization.
11    $username = $_POST['username'];
12    $fname = $_POST['fname'];
13    $lname = $_POST['lname'];
14    $phone = $_POST['phone'];
15    $mail = $_POST['mail'];
16    $password = $_POST['pwd'];
17    $password_repeat = $_POST['pwd-repeat'];
18
19    $normalizeduname = strtolower($username);
20    $normalizedmail = strtolower($mail);
21
22    // Empty fields validation - error "emptyfields" - GET uname, mail.
23    if (empty($username) || empty($mail) || empty($password) || empty($password_repeat))
24    {
25        $params = 'emptyfields&uname=' . $username . '&mail=' . $mail;
26        errorRedirect($params);
27    }
28    // Both mail and uname validation - error "invalidunamemail".
29    elseif (!filter_var($mail, FILTER_VALIDATE_EMAIL) && !preg_match("/^[a-zA-Z0-9]*$/", $username))
30    {
31        errorRedirect("invalidunamemail");
32    }
33    // Email validation - error "invalidmail" - GET uname.
34    elseif (!filter_var($mail, FILTER_VALIDATE_EMAIL))
35    {
36        $params = 'invalidmail&uname=' . $username;
37        errorRedirect($params);
38    }
39    // Username validation - error "invaliduname" - GET mail.
40    elseif (!preg_match("/^[a-zA-Z0-9]*$/", $username))
41    {
42        $params = 'invaliduname&mail=' . $mail;
43        errorRedirect($params);
44    }
45    // Password match check - error "passwordcheck" - GET uname, mail.
46    elseif ($password !== $password_repeat)
47    {
48        $params = 'passwordcheck&uname=' . $username . '&mail=' . $mail;
49        errorRedirect($params);
50    }

```

Εικόνα 28: Λειτουργίες ελέγχου της φόρμας στην do_signup.php

Κατά την περίπτωση της σύνδεσης ακολουθείται ίδιο μοτίβο. Η φόρμα στην login.php στέλνει τις πληροφορίες στην do_login.php η οποία εκτελεί αρχικά συνθήκη ελέγχου κενών πεδίων. Έπειτα εξετάζει εάν υπάρχει το όνομα χρήστη ή το email στον πίνακα users, εάν έχει επιβεβαιωθεί το email του χρήστη και αν ο κωδικός είναι σωστός. Αν οι συνθήκες ικανοποιηθούν γίνεται έναρξη συνεδρίας και ορίζονται οι μεταβλητές συνόδου userid και username. Σε αντίθετη περίπτωση γίνεται επαναφορά στην σελίδα login.php με κατάλληλα ορίσματα GET.

```
$sql = "SELECT * FROM users WHERE UserName=? OR Email=?";
$stmt = mysqli_stmt_init($dbconn);
// Check the connection to database.
if (!mysqli_stmt_prepare($stmt, $sql))
{
    errorRedirect('sqlerror');
}
// Case no connection error.
else
{
    // Query the database for the credentials provided.
    mysqli_stmt_bind_param($stmt,"ss", $mailuname, $mailuname);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);
    // Check if user exists.
    if ($row = mysqli_fetch_assoc($result))
    {
        // Check if user has verified his account.
        if ($row['EmailConfirmed'] == 0)
        {
            $params = 'mailnotconfirmed&uname='.$mailuname;
            errorRedirect($params);
        }
        // Check if the password is correct. Redirect with error.
        $password_check = password_verify($password, $row['PasswordHash']);
        if ($password_check == false)
        {
            $params = 'wrongpwd&uname='.$mailuname;
            errorRedirect($params);
        }
        // Case password is correct, start the session. Redirect to homepage.
        else if ($password_check == true)
        {
            session_start();
            $_SESSION['userid'] = $row['Id'];
            $_SESSION['username'] = $row['UserName'];

            exit(header('Location: /events/index.php?login=success'));
        }
    }
    // Case no user or email found. Redirect with error.
    else
    {
        errorRedirect("nouser");
    }
}
```

Εικόνα 29: Μέρος της σελίδας do_login.php

4.6 Σελίδες διαχειριστή

Στη σελίδα dashboard.php εμφανίζεται ο πίνακας με τους χρήστες και ο πίνακας με όλες τις κρατήσεις που έχουν γίνει, με τη δυνατότητα για διαγραφή της κάθε εγγραφής. Η PHP χρησιμοποιείται για να δημιουργήσει το περιεχόμενο όπως έχουμε δει και η JavaScript ελέγχει την λειτουργία της διαγραφής από τους πίνακες users και reservation . Κατά τις λειτουργίες αυτές καλείται η delete_user.php όπου εκτελεί τις κατάλληλες ενέργειες για την διαγραφή της εγγραφής χρήστη στην βάση και η cancel_register.php για την διαγραφή της επιθυμητής κράτησης αντίστοιχα.

Υλοποίηση Ticket Manager

```
14 <script>
15 $('user').click(function(e){
16     if (confirm('Είσαστε σίγουροι οτι θέλετε να διαγράψετε αυτόν τον χρήστη?')) {
17         var row = $(this).closest("tr")[0];
18         var cells = row.cells;
19
20         var userid = cells[0].textContent;
21         $(this).closest('tr').remove();
22         $.ajax({
23             url: 'functions/delete_user.php',
24             type: "POST",
25             data: {userid:userid},
26             success: function (data) {
27                 alert('Ο χρήστης διαγράφηκε επιτυχώς! '+data);
28                 location.reload();
29             }
30         });
31     }
32 });
33 $('reservation').click(function(e){
34     if (confirm('Είσαστε σίγουροι οτι θέλετε να ακυρώσετε αυτήν την κράτηση?')) {
35         var row = $(this).closest("tr")[0];
36         var cells = row.cells;
37
38         var seatid = cells[2].textContent;
39         $(this).closest('tr').remove();
40         $.ajax({
41             url: 'functions/cancel_registration.php',
42             type: "POST",
43             data: {seatid:seatid},
44             success: function (data) {
45                 alert('Η κράτηση ακυρώθηκε επιτυχώς! '+data);
46                 location.reload();
47             }
48         });
49     }
50 });
51 </script>
```

Εικόνα 30:Το σενάριο JS στην dashboard.php

Στην σελίδα new_event.php παρέχεται στον διαχειριστή η φόρμα εισαγωγής ενός γεγονότος στην βάση. Η φόρμα είναι σε HTML ενώ οι επιμέρους επεμβάσεις της PHP είναι για να συμπληρωθούν οι επιλογές στα αντικείμενα <select> της HTML. Η μέθοδος εκτελεί τις απαραίτητες ενέργειες για την δημιουργία εγγραφής.

```
6 if (isset($_POST['add-event-submit']))
7 {
8     require 'dbconn.php';
9
10    $uid = $_SESSION["userid"];
11    // Get the user input.
12    $title = $_POST['title'];
13    $description = $_POST['description'];
14    $typeid = $_POST['type'];
15    $venueid = $_POST['venue'];
16    $startdatetime = $_POST['meeting-start-time'];
17    $enddatetime = $_POST['meeting-end-time'];
18    $startdatetime = str_replace('T', '', $startdatetime);
19    $startdatetime = $startdatetime."00";
20    $enddatetime = str_replace('T', '', $enddatetime);
21    $enddatetime = $enddatetime."00";
22    $familyevent = $_POST['family-event'];
23    $price = $_POST['price'];
24
25    // File upload handler.
26    $banner = basename($_FILES['banner']['name']);
27    $uploaddir = realpath($_SERVER["DOCUMENT_ROOT"].'/events/images');
28    $uploadfile = $uploaddir ."\\". $banner;
29    move_uploaded_file($_FILES['banner']['tmp_name'], $uploadfile);
30
31    echo "<p>title: " . $title . " description: " . $description . " type id: " . $typeid . " venue id: " . $venueid . " start: " . $startdatetime . " end: " . $enddatetime . " family: " . $familyevent .
32    " price: " . $price . "</p>";
33    // Insert Query setup.
34    $sql = "INSERT INTO event (Name, Description, ImgUrl, StartDateTime, EndTime, VenueId, EventTypeId, FamilyEventId, Price) VALUES ('$title', '$description', '$banner', '$startdatetime', '$enddatetime', '$venueid', '$typeid', '$familyevent', '$price')";
35    $success = mysqli_query($dbconn,$sql);
36    if ($success)
37    {
38        exit(header('Location: /events/new_event.php?insert=success'));
39    }
40    // Case query was not successful.
41    else
42    {
43        exit(header('Location: /events/new_event.php?insert=fail'));
44    }
45 }
46 else
47 {
48    exit(header('Location: /events/index.php'));
49 }
50 ?>
```

Εικόνα 31: Η σελίδα add_event.php

4.7 Σελίδες λογαριασμού χρήστη, καλάθι και ολοκλήρωση αγοράς

Στην σελίδα λογαριασμού, που φέρει το Username του εκάστοτε χρήστη, ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία του λογαριασμού του καθώς και τον πίνακα με τις κρατήσεις του. Εφόσον έχει κάνει κάποιες κρατήσεις, αυτές θα εμφανίζονται με μορφή πίνακα με την δυνατότητα διαγραφής. Την διαγραφή και σε αυτήν την περίπτωση πραγματοποιεί η `cancel_registration.php`. Στην περίπτωση μη υπάρχουσας κράτησης το σύστημα ενημερώνει με κατάλληλο μήνυμα.

Για την σελίδα καλάθι αναφερθήκαμε στο κομμάτι της εμφάνισης και είδαμε τον κώδικα στην Εικόνα 9. Το άλλο κομμάτι είναι αυτό των σεναρίων ακύρωσης μίας θέσης και μεταφοράς στην σελίδα πληρωμής. Κατά το πρώτο εφόσον πατηθεί κάποιο κουμπί ακύρωσης στον πίνακα με τις κρατήσεις θα εμφανιστεί ένα παράθυρο τύπου `confirm` (Εικόνα 5) και στην περίπτωση επιβεβαίωσης θα γίνει αποστολή των επιθυμητών στοιχείων στην `cancel_registration.php` η οποία αναλαμβάνει την ακύρωση της κράτησης. Στο δεύτερο σενάριο ελέγχουμε αν πατήθηκε το κουμπί «Συνέχεια» για ανακατεύθυνση στην σελίδα πληρωμής.

Η σελίδα πληρωμής περιέχει τα στοιχεία κράτησης και τα εμφανίζει σε μορφή πίνακα. Κάτω από τον πίνακα είναι το τελικό ποσό πληρωμής και ακολουθεί η φόρμα πληρωμής. Η φόρμα δεν λειτουργεί, όμως για να προσομοιώσουμε μια επιτυχής κράτηση δόθηκε λειτουργία στο κουμπί πληρωμής. Η λειτουργία του είναι να μας ανακατευθύνει στην ίδια σελίδα με κατάλληλο όρισμα POST που κάνει την ιστοσελίδα να δείξει μήνυμα επιτυχούς πληρωμής.

```

77 function reservationstable()
78 {
79     global $output, $dbconn;
80
81     $sql = "SELECT * FROM reservation WHERE UserId='".$$_SESSION['userid']."'";
82     $result = mysqli_query($dbconn, $sql);
83     if(mysqli_num_rows($result)==0)
84     {
85         $output .= "<div class='center'><p class='successMsg'>Ο πίνακας κρατήσεων είναι άδειος. Οι κρατήσεις σας θα εμφανίζονται εδώ.</p></div>";
86     }
87     else
88     {
89         $output .= '<div class="w3-container w3-padding">';
90         $output .= '<h2>Οι κρατήσεις μου: </h2>';
91         $output .= '<table id="reservations-table" style="width:100%" class="w3-table-all center">';
92         $output .= '<tr>';
93         $output .= '<th>Αριθμός κράτησης</th>';
94         $output .= '<th>Αριθμός θέσης</th>';
95         $output .= '<th>Ημερομηνία</th>';
96         $output .= '<th>Αναγνωριστικό γεγονότος</th>';
97         $output .= '<th>Ακύρωση</th>';
98         $output .= '<th>Εκτύπωση</th>';
99         $output .= '</tr>';
100        while ($row = mysqli_fetch_assoc($result))
101        {
102            {
103                $output .= '<tr>';
104                $output .= '<td> $row['Id'] . '</td>';
105                $output .= '<td> $row['SeatId'] . '</td>';
106                $output .= '<td> $row['EventDate'] . '</td>';
107                $output .= '<td> $row['EventId'] . '</td>';
108            }
109            $output .= '<td><input id="cancel" type="button" value="X" class="w3-btn w3-red w3-round-small w3-margin-left reservation"></td>';
110            $output .= '<td><input id="print" type="button" value="Print" class="w3-btn w3-blue w3-round-small" onclick="window.print();return false;"></td>';
111        }
112    }
113    $output .= '</table></div>';
114    echo $output;
115 }
116 >>
117 <script>
118 $( ".reservation" ).click(function(e){
119     if (confirm('Είστε σίγουροι ότι θέλετε να ακυρώσετε αυτήν την κράτηση?<br>Είναι πιθανόν να υπάρχουν χρεώσεις!')) {
120         var row = $(this).closest("tr")[0];
121         var cells = row.cells;
122
123         var seatid = cells[2].textContent;
124         $(this).closest("tr").remove();
125         $.ajax({
126             url: 'functions/cancel_registration.php',
127             type: "POST",
128             data: {seatid:seatid},
129             success: function (data) {
130                 alert("Η κράτηση ακυρώθηκε επιτυχώς! "+data);
131                 location.reload();
132             }
133         });
134     }
135 });
136 </script>

```

Εικόνα 32: Συνάρτηση πίνακα κρατήσεων και σενάριο ακύρωσης κράτησης στην `profile.php`

```

1 <?php
2 require 'functions/dbconn.php';
3 include 'header.php';
4
5 $output = '';
6 $sum = 0;
7 $cart = $_SESSION['cart'];
8 //print_r($cart);
9 $output .= '<div class="w3-container w3-padding">';
10 $output .= '<h2>Order Details: </h2>';
11 $output .= '<table id="cart-table" style="width:100%" class="w3-table-all center">';
12 $output .= '<tr>';
13 $output .= '<th>Id</th>';
14 $output .= '<th>Reference Id</th>';
15 $output .= '<th>Event</th>';
16 $output .= '<th>Date</th>';
17 $output .= '<th>Venue</th>';
18 $output .= '<th>Seat No</th>';
19 $output .= '<th>Print</th>';
20 $output .= '</tr>';
21
22 foreach ($cart as $key => $value)
23 {
24     $output .= '<tr>';
25     $output .= '<td>'. $key. '</td>';
26
27     foreach ($value as $row => $data) {
28         if ($row == 'EventId') {
29             $sql = "SELECT Price FROM event WHERE Id='".$data."'";
30             $result = mysqli_query($dbconn,$sql);
31             $row = mysqli_fetch_assoc($result);
32             $price = $row['Price'];
33             $sum = $sum + $price;
34         }
35         $output .= '<td>'. $data. '</td>';
36     }
37
38     if (isset($_POST['paymentconfirmation']))
39     {
40         $output .= '<td><input id="cancel" type="button" value="Print" class="w3-btn w3-blue w3-round-small"></td>';
41     }
42     else
43     {
44         $output .= '<td><input id="cancel" type="button" value="Print" class="w3-btn w3-blue w3-round-small" disabled></td>';
45     }
46     $output .= '</tr>';
47 }
48 $output .= '</table></div>';
49 $output .= '<div class="center w3-container w3-padding w3-margin w3-border">';
50 if (isset($_POST['paymentconfirmation']))
51 {
52     $output .= '<div class="w3-tag w3-xxlarge w3-green">Επιτυχής πληρωμή!</div>';
53 }
54 else
55 {
56     $output .= '<h2>Order Total: </h2>';
57     $output .= '<h2 class="w3-margin-left">€ '. $sum. '</h2>';
58 }
59 $output .= '</div>';
60 echo $output;
61
62 ?>

```

Εικόνα 33: Το PHP μέρος της checkout.php

4.8 Επίλογος

Στο κεφάλαιο αυτό αναλύσαμε τον κώδικα στις κύριες λειτουργίες της εφαρμογής και ολοκληρώσαμε την εικόνα για το σύστημα. Στο κεφάλαιο που ακολουθεί θα δούμε το εγχειρίδιο χρήσης για χρήστη και διαχειριστή, και θα δούμε στην πράξη τα όσα μελετήθηκαν στις ενότητες αυτού του κεφαλαίου.

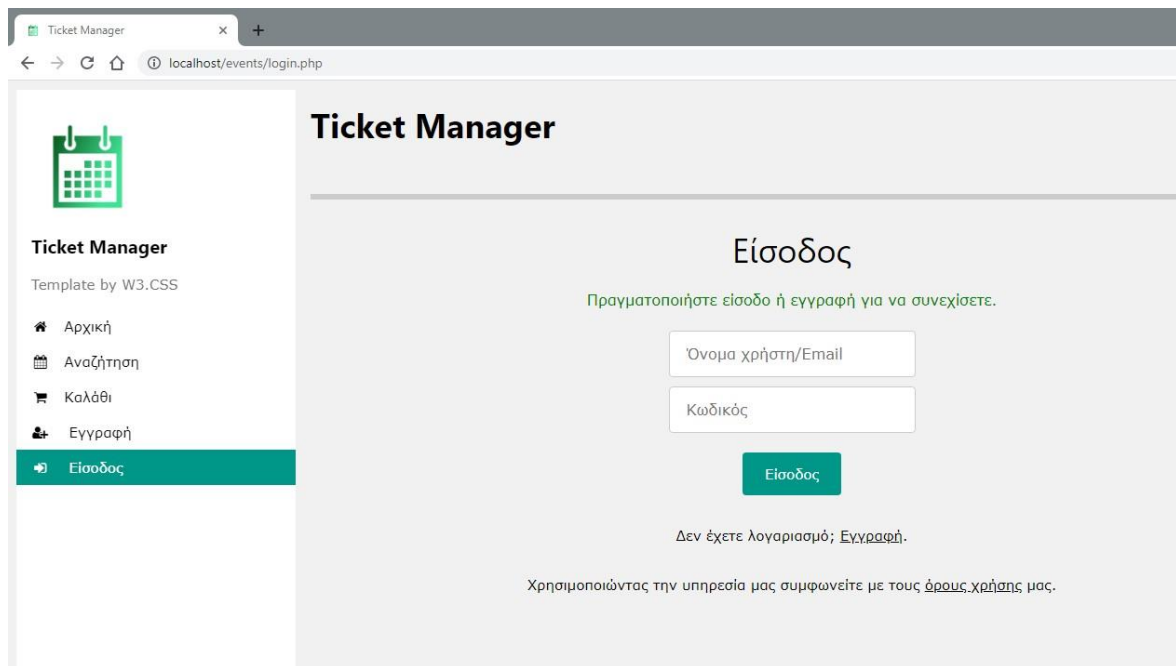
Κεφάλαιο 5: Εγχειρίδιο χρήσης ticket manager

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εξετάσουμε τις σελίδες της εφαρμογής καθώς και τις λειτουργίες της. Ήδη στην υποενότητα 2.4.4 είδαμε το μενού πλοήγησης (navigation menu) βάση του οποίου περιηγείται ο χρήστης και ο διαχειριστής του συστήματος στην εφαρμογή. Παρακάτω θα δούμε αρχικά τις βασικές λειτουργίες δημιουργίας λογαριασμού (signup) και σύνδεσης χρήστη (login). Έπειτα θα δούμε τις σελίδες που διατίθενται με πρόσβαση από τον απλό χρήστη και αυτές με πρόσβαση ως διαχειριστής του συστήματος και θα αναλύσουμε τις δυνατότητές τους.

5.2 Δημιουργία λογαριασμού και σύνδεση

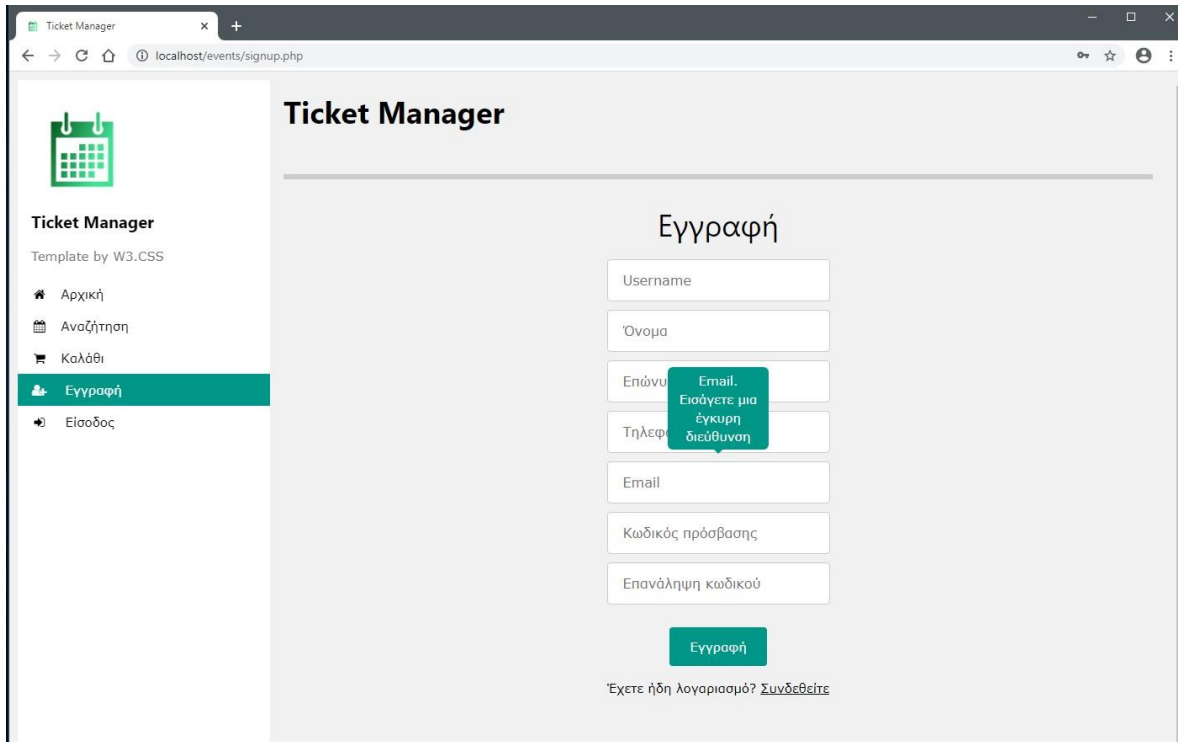
Πληκτρολογώντας την διεύθυνση της εφαρμογής σε έναν περιηγητή (browser) μεταφερόμαστε αυτόματα στην σελίδα σύνδεσης χρήστη login.php. Στη σελίδα αυτή το σύστημα μας ενημερώνει για τη διαδικασία που πρέπει να ακολουθήσουμε για να αποκτήσουμε πρόσβαση στις λειτουργίες της εφαρμογής.



Εικόνα 34: Η σελίδα σύνδεσης login.php.

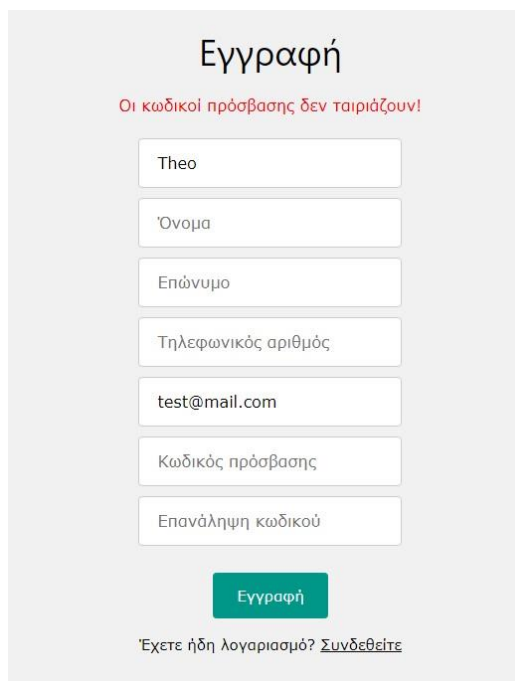
Εφόσον επισκεπτόμαστε για πρώτη φορά την εφαρμογή είναι απαραίτητο να δημιουργήσουμε έναν λογαριασμό για να συνεχίσουμε. Για να δημιουργήσουμε λογαριασμό μπορούμε να πάμε από το μενού πλοήγησης αριστερά και να επιλέξουμε «Εγγραφή» ή να ακολουθήσουμε τον σύνδεσμο κάτω από το κουμπί «Είσοδος».

Όποιον τρόπο και να επιλέξουμε μεταφερόμαστε στην σελίδα δημιουργίας λογαριασμού χρήστη όπου εμφανίζονται τα πεδία που πρέπει να εισάγουμε για τη δημιουργία. Όλα τα πεδία διαθέτουν αναδυόμενη συμβουλή (tooltip) για την διευκόλυνσή κατά την εισαγωγή. Για παράδειγμα το όνομα χρήστη πρέπει να είναι μόνο λατινικοί χαρακτήρες και αριθμοί και η διεύθυνση email να είναι έγκυρη.



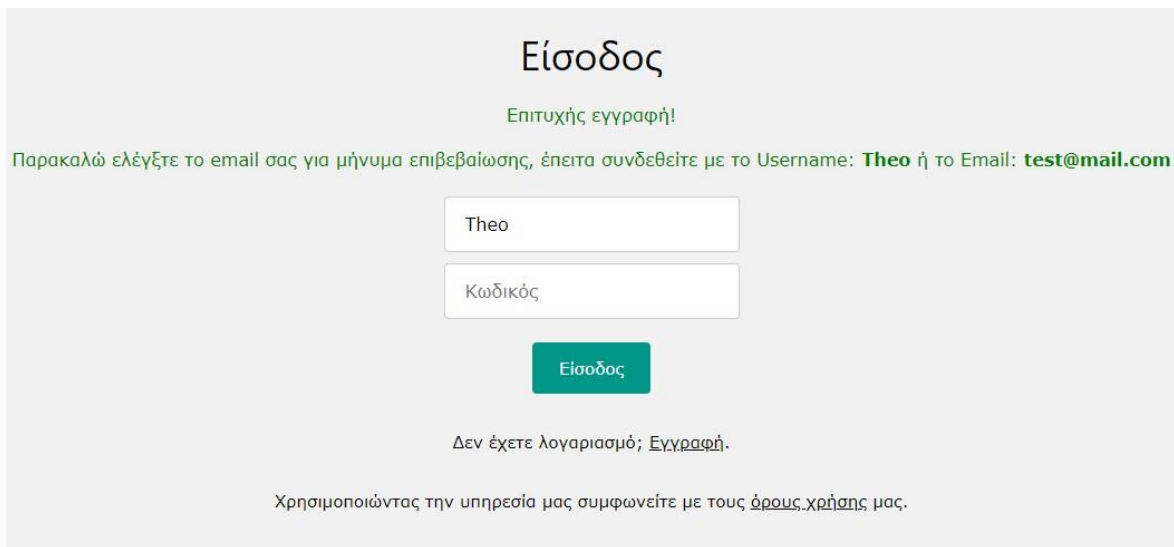
Εικόνα 35: Η σελίδα εγγραφής χρήστη signup.php.

Αφού συμπληρώσουμε τα πεδία πατάμε το κουμπί «Εγγραφή» στο τέλος της φόρμας για να ολοκληρώσουμε την διαδικασία. Εάν κατά την συμπλήρωση των στοιχείων κάνουμε κάποιο λάθος, όπως για παράδειγμα λάθος στα πεδία «κωδικός πρόσβασης» και «επανάληψης κωδικού», το σύστημα θα μας ενημερώσει κατάλληλα.



Εικόνα 36: Λάθος εισαγωγή στοιχείων κατά την εγγραφή.

Στην περίπτωση ορθής εισαγωγής στοιχείων μεταφερόμαστε αυτόματα στη σελίδα σύνδεσης όπου το μήνυμα προτροπής έχει αλλάξει και μας δίνει τις απαραίτητες οδηγίες για την σύνδεσή μας.

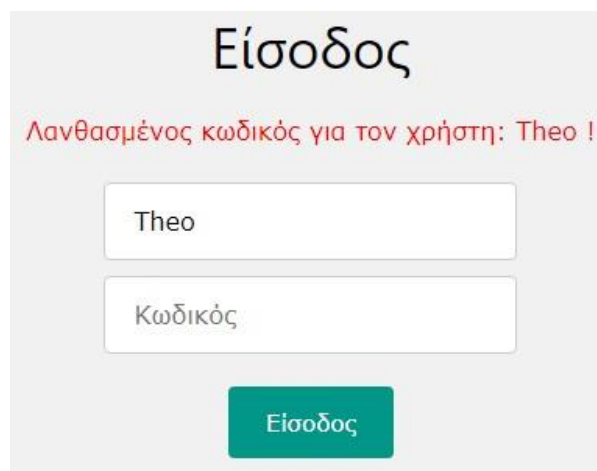


The screenshot shows a login page titled "Είσοδος" (Login). Below the title is the message "Επιτυχής εγγραφή!" (Successful registration!). A green instruction text reads: "Παρακαλώ ελέγξτε το email σας για μήνυμα επιβεβαίωσης, έπειτα συνδεθείτε με το Username: **Theo** ή το Email: **test@mail.com**". There are two input fields: the first contains "Theo" and the second is labeled "Κωδικός" (Password). A green "Είσοδος" (Login) button is positioned below the fields. At the bottom, there is a link: "Δεν έχετε λογαριασμό; [Εγγραφή](#)." and a footer note: "Χρησιμοποιώντας την υπηρεσία μας συμφωνείτε με τους [όρους χρήσης](#) μας."

Εικόνα 37: Επιτυχής εγγραφή στο σύστημα.

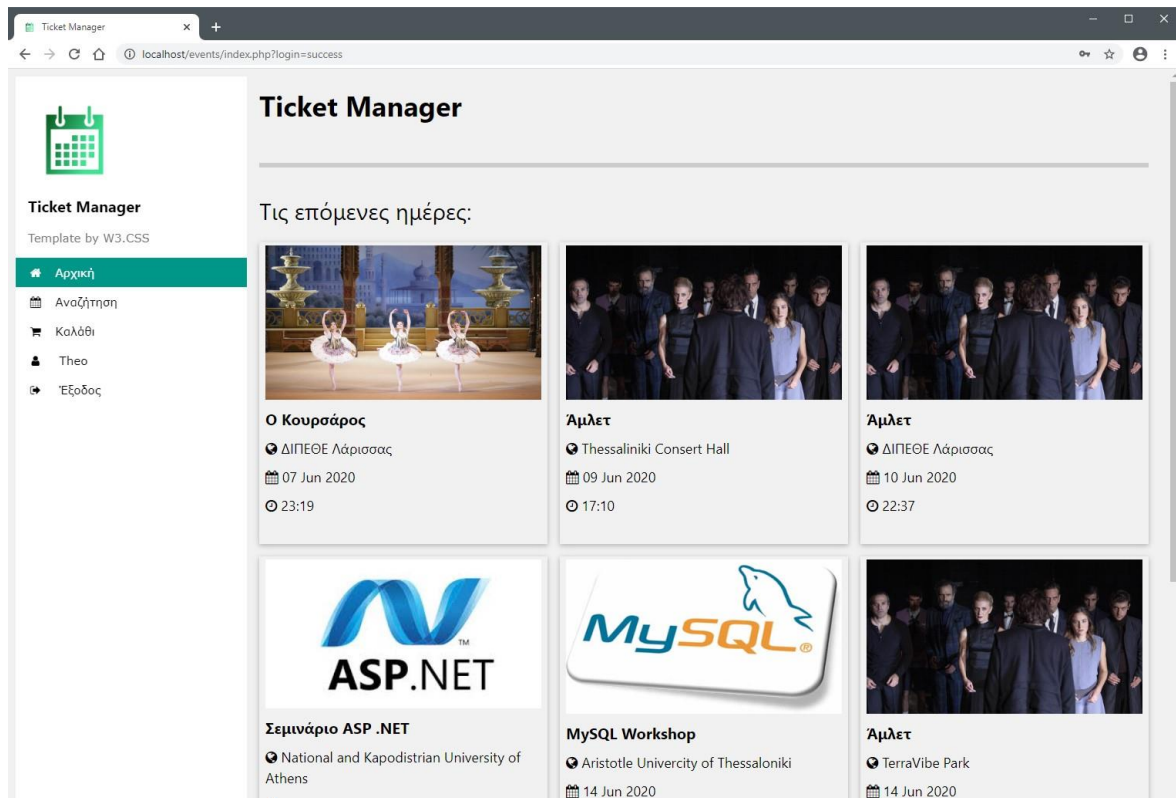
Η λειτουργία email δεν έχει υλοποιηθεί καθώς ξεφεύγει από την σκοπιά της εργασίας αυτής. Πάραυτα όπως σχολιάσαμε και στην ενότητα 1.3 είναι ορθό να περιλαμβάνουμε διάφορους ελέγχους για να αποφύγουμε κακόβουλη χρήση που μπορεί να οδηγήσει στην κατάρρευση του συστήματος. Το σύστημά μας υλοποιεί τους περιορισμούς που είδαμε στην υποενότητα 1.3.1.

Αφού υποθέσουμε ότι επιβεβαιώσαμε τον λογαριασμό μας μπορούμε να συνδεθούμε είτε με το email είτε με το όνομα χρήστη που εισάγαμε κατά την εγγραφή. Όπως είδαμε και παραπάνω στην περίπτωση που κάνουμε κάποιο λάθος στην εισαγωγή των στοιχείων θα λάβουμε αντίστοιχη προειδοποίηση. Στην περίπτωση ορθής εισαγωγής μεταφερόμαστε στην αρχική σελίδα της εφαρμογής index.php.



The screenshot shows the same login page as in Figure 37, but with an error message in red text: "Λανθασμένος κωδικός για τον χρήστη: Theo !". The input fields and the "Είσοδος" button remain the same.

Εικόνα 38: Λάθος εισαγωγή κατά την σύνδεση



Εικόνα 39: Η αρχική σελίδα index.php

5.3 Αρχική σελίδα και μενού πλοήγησης

Όπως βλέπουμε στην Εικόνα 39 στην αρχική σελίδα εμφανίζονται έξι γεγονότα, σε μορφή κάρτας, τα οποία θα λάβουν χώρα τις επόμενες ημέρες. Τα γεγονότα αυτά είναι από όλες τις κατηγορίες και είναι ταξινομημένα βάση ημερομηνίας έναρξης.

Στην κάρτα – σύνδεσμο βλέπουμε τον τίτλο με τονισμένη γραμματοσειρά και τις πληροφορίες τόπος διεξαγωγής (venue), ημερομηνία και ώρα έναρξης. Ο σύνδεσμος μας οδηγεί στην σελίδα παρουσίασης γεγονότος στην οποία θα αναφερθούμε παρακάτω.

Ακόμα παρατηρούμε οτι το μενού πλοήγησης έχει αλλάξει. Βλέπουμε ότι στην θέση της εγγραφής έχει μπει ο σύνδεσμος για την σελίδα επεξεργασίας λογαριασμού χρήστη και φέρει το όνομα χρήστη μας: "Theo". Ενώ ο σύνδεσμος «Είσοδος» έχει αντικατασταθεί από τον σύνδεσμο «Έξοδος» ο οποίος πραγματοποιεί έξοδο από την εφαρμογή και καταστρέφει τα στοιχεία συνόδου. Στην περίπτωση που ο χρήστης έχει εισέλθει στην εφαρμογή ως διαχειριστής το μενού γίνεται όπως είδαμε στην Εικόνα 15.

5.4 Σελίδα λογαριασμού χρήστη και λειτουργία εξόδου

Στη σελίδα λογαριασμού του ο χρήστης έχει την δυνατότητα να δει και να επεξεργαστεί τα στοιχεία που έδωσε κατά την εγγραφή του. Αυτά είναι το όνομα χρήστη, ο κωδικός πρόσβασης, το ονοματεπώνυμο και η διεύθυνση email. Ακόμα παρέχεται και η λειτουργία διαχείρισης των κρατήσεων εφόσον υπάρχουν. Για την διαδικασία επεξεργασίας του λογαριασμού και κρατήσεων το σύστημα μας ενημερώνει κατάλληλα.

Ο σύνδεσμος «Έξοδος» στο μενού αριστερά εκτελεί τις απαραίτητες λειτουργίες εξόδου όπως η καταστροφή των στοιχείων συνεδρίας και μας ανακατευθύνει στην σελίδα εισόδου (login.php) με κατάλληλο μήνυμα.

Ticket Manager

Template by W3.CSS

- Αρχική
- Αναζήτηση
- Καλάθι
- Theo**
- Έξοδος

Για να ενημερώσετε μόνο τις πληροφορίες σας αφήστε τα πεδία αλλαγής κωδικού κενά!

Username:

First Name:

Last Name:

Email:

Change Password:

Repeat New Password:

Οι κρατήσεις μου:

Αριθμός κράτησης	Αριθμός θέσης	Ημερομηνία	Αναγνωριστικό γεγονόςτος	Ακύρωση	Εκτύπωση
45	25	2020-10-21 15:14:34	141	<input type="button" value="X"/>	<input type="button" value="Print"/>
53	2	2020-09-26 05:53:53	96	<input type="button" value="X"/>	<input type="button" value="Print"/>

Εικόνα 40: Σελίδα επεξεργασίας λογαριασμού profile.php

Είσοδος

Επιτυχής αποσύνδεση!

Εικόνα 41: Επιτυχής αποσύνδεση

5.5 Σελίδες διαχειριστή

Παραπάνω είδαμε τους δύο συνδέσμους του μενού πλοήγησης κατά την σύνδεση ως διαχειριστής της εφαρμογής. Αυτοί οδηγούν στις σελίδες του πίνακα ελέγχου του διαχειριστή, την dashboard.php και την σελίδα προσθήκης γεγονότος new_event.php. Στην σελίδα dashboard ο διαχειριστής μπορεί να δει πληροφορίες για τους χρήστες στη βάση καθώς και να διαγράψει μία εγγραφή. Ενώ στη σελίδα προσθήκης γεγονότος μπορεί να προσθέσει μία νέα εγγραφή στον πίνακα event της βάσης.

Ticket Manager

Template by W3.CSS

- Αρχική
- Αναζήτηση
- Dashboard**
- Προσθήκη
- Καλάθι
- Admin
- Έξοδος

Εγγεγραμμένοι Χρήστες:

Id	Username	Email	FName	LName	Mail Confirmation	Delete User
1	Admin	test@test.com	Θοδωρής	Ζερβάκης	TRUE	X
2	ABC	ABC@ABC.COM	ABCD	DEFG	FALSE	X
4	Nick	test@testmail.com	Νίκος	Παπαδόπουλος	TRUE	X
5	Theo	test@mail.com	Θοδωρής	Ζερβάκης	TRUE	X
6	John	john@example.com	john	johnson	TRUE	X
7	test	test@example.com	George	Georgopoulos	TRUE	X

Κρατήσεις:

Ticket Id	User Id	Seat Id	Event Date	Event Id	Cancel Reservation
45	5	25	2020-10-21 15:14:34	141	X
48	4	13	2021-04-01 03:01:14	97	X
49	4	15	2021-04-01 03:01:14	97	X
52	7	30	2020-08-11 20:01:23	75	X

Εικόνα 42: Η σελίδα dashboard.php

Ticket Manager

Template by W3.CSS

- Αρχική
- Αναζήτηση
- Dashboard
- Προσθήκη**
- Καλάθι
- Admin
- Έξοδος

Create Event

Title

Description

Select Type

Select Venue

Starting Date and Time

Εικόνα 43: Μέρος της σελίδας new_event.php

5.6 Σελίδα αναζήτησης

Στην σελίδα αναζήτησης ο χρήστης έχει την δυνατότητα για αναζήτηση δρώμενων σύμφωνα με:

- την ή τις κατηγορίες γεγονότων,
- τον τόπο διεξαγωγής και
- την ημερομηνία έναρξης.

Ο χρήστης μπορεί να επιλέξει να κάνει την αναζήτηση με τρεις τρόπους:

- I. χωρίς να συμπληρώσει κανένα πεδίο
- II. συμπληρώνοντας ένα μόνο πεδίο της αρεσκείας του
- III. κάνοντας συνδυασμό ανάμεσα στα διαθέσιμα πεδία.

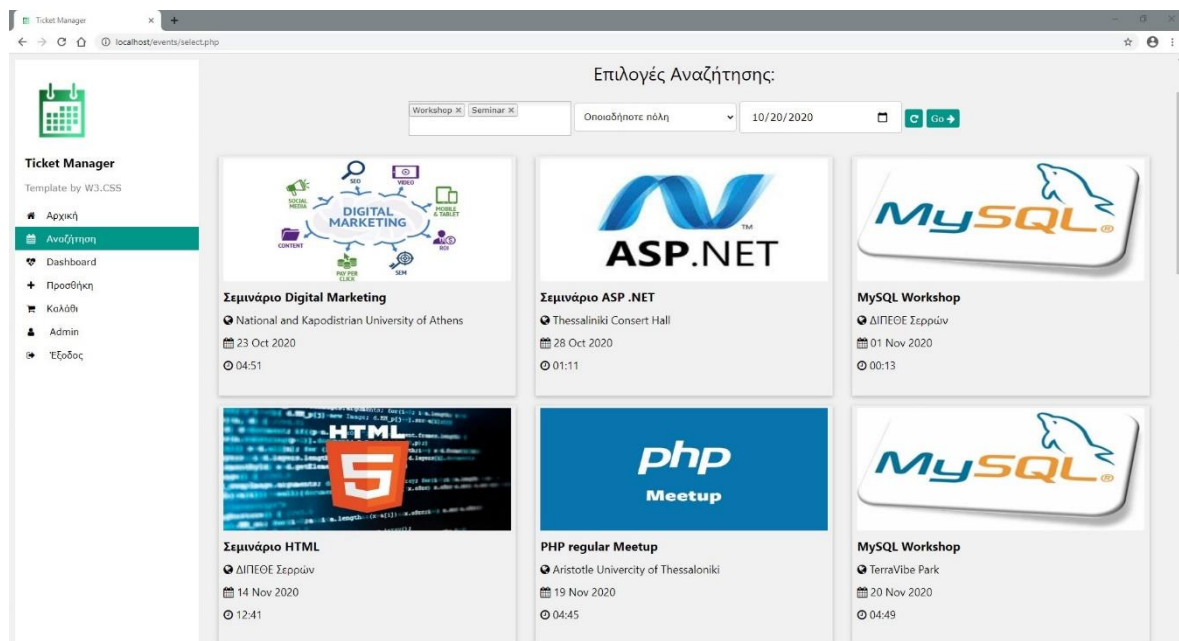
Για όλες της περιπτώσεις η αναζήτηση ολοκληρώνεται αφού πατήσουμε το κουμπί “Go” και ο μέγιστος αριθμός αποτελεσμάτων στην απόκριση του server είναι πενήντα έξι (56). Η ταξινόμησή τους είναι αύξουσα βάση της ημερομηνίας έναρξης

Κατά την πρώτη περίπτωση ο χρήστης πατάει απλώς το κουμπί αναζήτησης “Go” και βλέπει αποτελέσματα για όλες τις κατηγορίες γεγονότων, σε όλες τις πόλεις, που θα λάβουν χώρα από την στιγμή που έγινε η αναζήτηση και έπειτα.

Στην δεύτερη περίπτωση τα αποτελέσματα φιλτράρονται βάση του πεδίου που επιλέχθηκε και τα υπόλοιπα πεδία αντιμετωπίζονται όπως στην πρώτη περίπτωση.

Εάν η επιλογή συνδυάζει περισσότερα πεδία τότε τα αποτελέσματα φιλτράρονται βάση των συμπληρωμένων πεδίων και αν υπάρχουν κενά αντιμετωπίζονται όπως στην πρώτη περίπτωση.

Το κουμπί με την ένδειξη επαναφοράς αριστερά του “Go” επαναφέρει την ημερομηνία στο αντίστοιχο πεδίο.



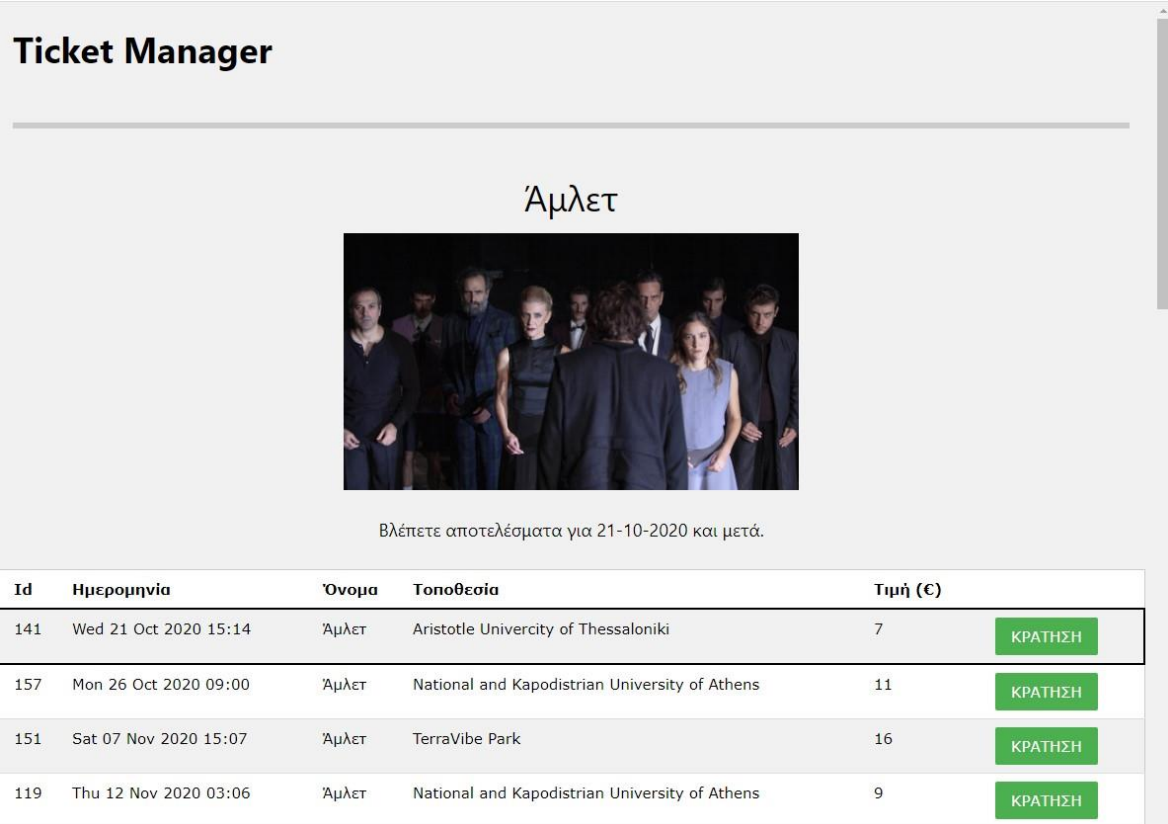
Εικόνα 44: Αναζήτηση βάση συνδυασμού κατηγοριών και ημερομηνίας

5.7 Σελίδα κράτησης γεγονότος

Τα αποτελέσματα που είδαμε στη σελίδα αναζήτησης, όπως και στην αρχική σελίδα είναι τύπου συνδέσμου. Πατώντας στην κάρτα – σύνδεσμο μεταφερόμαστε στην σελίδα κράτησης (εμφάνισης) γεγονότος event.php.

Στην σελίδα γεγονότος μπορούμε να δούμε έναν πίνακα με τις πληροφορίες του ίδιου γεγονότος σε διάφορες τοποθεσίες όπως η ημερομηνία, η τοποθεσία, η τιμή κράτησης κ.α. Τα αποτελέσματα είναι ταξινομημένα και εδώ βάση της ημερομηνίας έναρξης ενώ ο μέγιστος αριθμός τους είναι πενήντα. Ως ημερομηνία εκκίνησης για τις προτάσεις σε άλλες τοποθεσίες είναι οι ημερομηνίες από το επιλεγμένο γεγονός και έπειτα όπως αναφέρεται και στο σχετικό μήνυμα.

Η επιλογή που μας οδήγησε στην σελίδα κρατήσεων βρίσκεται στην πρώτη γραμμή του πίνακα αποτελεσμάτων και έχει τονισμένο περίγραμμα για να διακρίνεται.



Ticket Manager

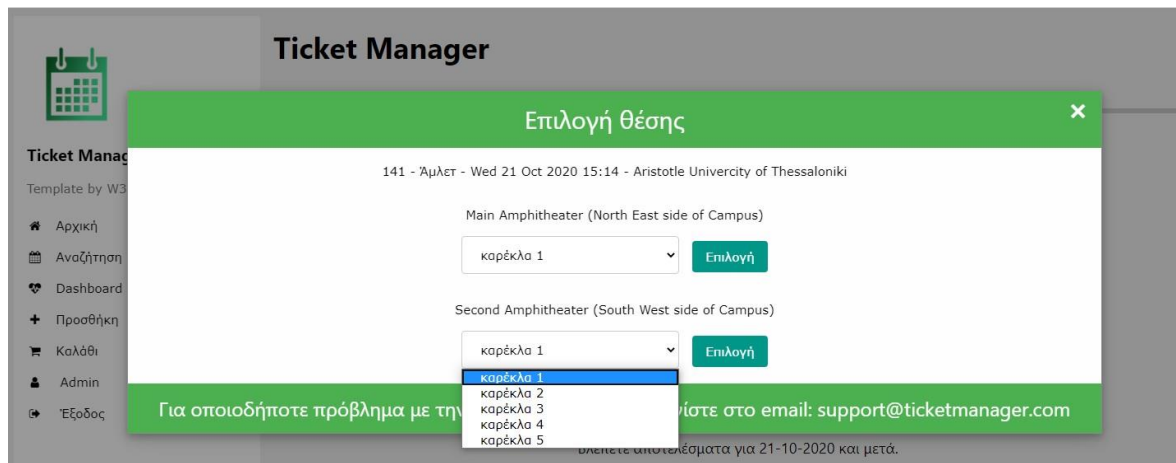
Άμλετ

Βλέπετε αποτελέσματα για 21-10-2020 και μετά.

Id	Ημερομηνία	Όνομα	Τοποθεσία	Τιμή (€)	
141	Wed 21 Oct 2020 15:14	Άμλετ	Aristotle University of Thessaloniki	7	ΚΡΑΤΗΣΗ
157	Mon 26 Oct 2020 09:00	Άμλετ	National and Kapodistrian University of Athens	11	ΚΡΑΤΗΣΗ
151	Sat 07 Nov 2020 15:07	Άμλετ	TerraVibe Park	16	ΚΡΑΤΗΣΗ
119	Thu 12 Nov 2020 03:06	Άμλετ	National and Kapodistrian University of Athens	9	ΚΡΑΤΗΣΗ

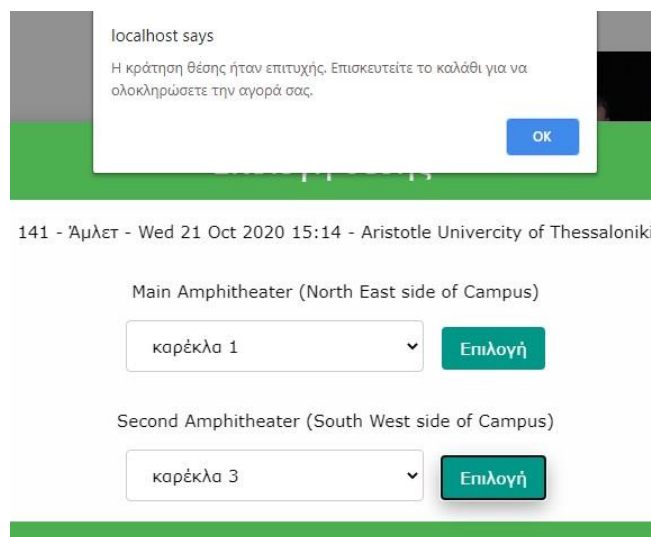
Εικόνα 45: Μέρος της σελίδας κράτησης event.php

Πατώντας ένα από τα κουμπιά «ΚΡΑΤΗΣΗ» στον πίνακα με των αποτελεσμάτων ανοίγει το αναδυόμενο παράθυρο επιλογής θέσης. Το παράθυρο αυτό περιλαμβάνει τις πληροφορίες για το γεγονός και πληροφορίες για την υποπεριοχή στην τοποθεσία διεξαγωγής. Μας δίνει την δυνατότητα να επιλέξουμε την θέση που επιθυμούμε από τα διαθέσιμα dropdown κάτω από την υποπεριοχή.



Εικόνα 46: Αναδυόμενο παράθυρο επιλογής θέσης

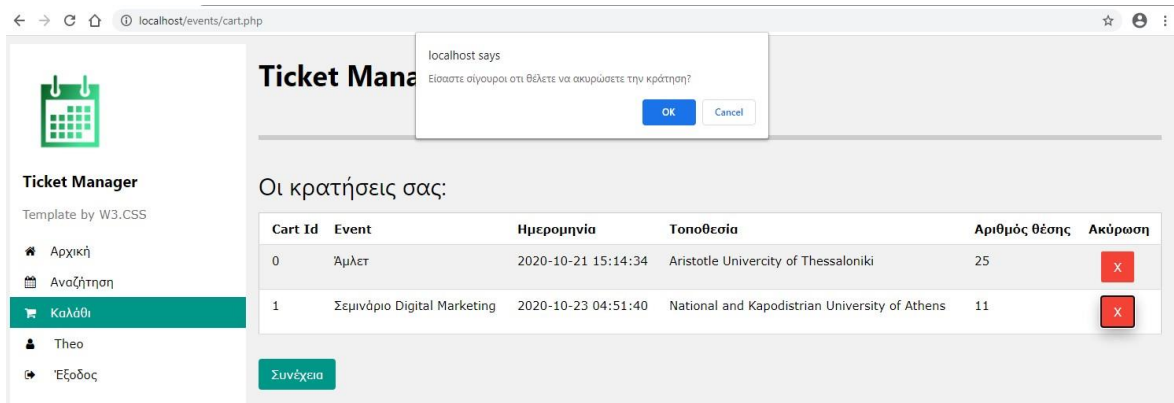
Αφού επιλέξουμε μια θέση πατάμε το κουμπί «Επιλογή» στα δεξιά της και ενημερωνόμαστε από το σύστημα αναλόγως. Στη συνέχεια μπορούμε να κάνουμε κάποια άλλη κράτηση για το ίδιο γεγονός ή να επιστρέψουμε στην αναζήτηση και να επιλέξουμε κάποιο άλλο.



Εικόνα 47: Μήνυμα επιτυχούς κράτησης

5.8 Σελίδες καλάθι αγορών και πληρωμής

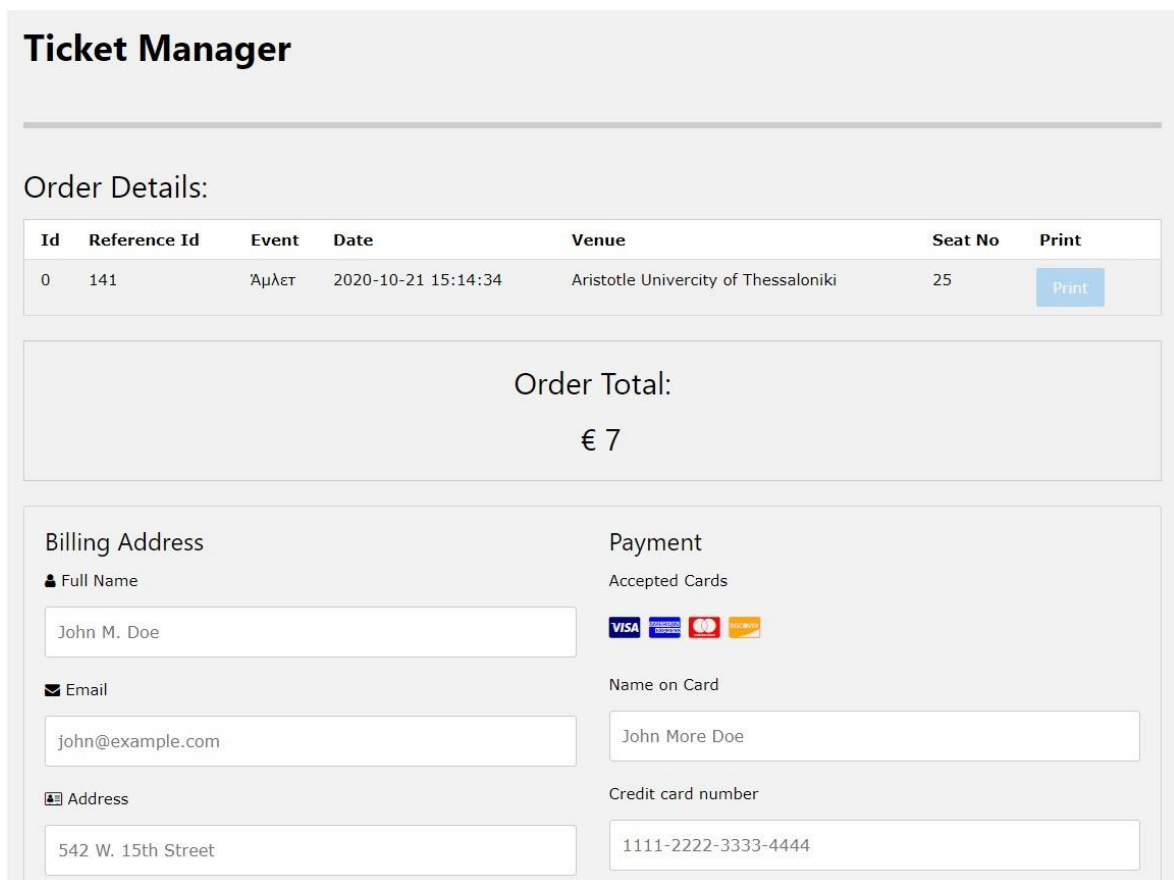
Υποθέτοντας ότι χρειαζόμαστε ακόμα μία θέση, όπως για παράδειγμα για μια άλλη ημερομηνία ή τοποθεσία ακολουθούμε πάλι την παραπάνω διαδικασία. Αφού ολοκληρώσουμε την διαδικασία μπορούμε να επιλέξουμε από το μενού αριστερά τον σύνδεσμο «Καλάθι» με τον οποίο θα μεταφερθούμε στην αντίστοιχη σελίδα cart.php. Στην σελίδα αυτή βλέπουμε τον πίνακα κρατήσεων εισιτηρίων με χαρακτηριστικά όπως όνομα, ημερομηνία, αριθμό θέσης κ.α. Επίσης μας δίνεται η δυνατότητα να αφαιρέσουμε κάποια κράτηση, που αποφασίσαμε ότι δεν χρειαζόμαστε πιά. Αυτό επιτυγχάνεται πατώντας το αντίστοιχο κόκκινο κουμπί με την ένδειξη X στην στήλη «Ακύρωση».



Εικόνα 48: Ακύρωση κράτησης από το καλάθι αγορών

Εφόσον καταλήξαμε στην επιλογή μας πατάμε το κουμπί «Συνέχεια» για να μεταφερθούμε στην σελίδα ολοκλήρωσης της παραγγελίας.

Στην σελίδα αυτή αναφέρεται το συνολικό ποσό που θα πληρώσουμε για την κράτησή μας και εμφανίζεται η φόρμα συμπλήρωσης στοιχείων πληρωμής. Παρατηρούμε ότι πριν από την πληρωμή το κουμπί εκτύπωσης του εισιτηρίου δεν επιτρέπεται να πατηθεί. Αφού ολοκληρώσουμε την πληρωμή το σύστημα μας ειδοποιεί κατάλληλα και το κουμπί "Print" ενεργοποιείται για να μπορούμε να τυπώσουμε τα εισιτήρια.



Εικόνα 49: Σελίδα checkout πριν την πληρωμή

Ticket Manager

Order Details:

Id	Reference Id	Event	Date	Venue	Seat No	Print
0	141	Άμλετ	2020-10-21 15:14:34	Aristotle University of Thessaloniki	25	Print

Επιτυχής πληρωμή!

Εικόνα 50: Σελίδα checkout μετά την πληρωμή

5.9 Επίλογος

Στο κεφάλαιο αυτό ανατρέξαμε σε όλες τις σελίδες της εφαρμογής και εξετάσαμε αναλυτικά τις περιπτώσεις χρήσης της. Στο κεφάλαιο που ακολουθεί παρατίθενται τα συμπεράσματα που προκύπτουν από την συγγραφή της εργασίας και την σχεδίαση της εφαρμογής, ποιες δυσκολίες αντιμετωπίστηκαν, καθώς και μελλοντικές επεκτάσεις.

Κεφάλαιο 6: Συμπεράσματα και μελλοντικές επεκτάσεις

Στην παρούσα πτυχιακή εργασία, με αφορμή την συνεχή ανάπτυξη του ηλεκτρονικού εμπορίου, αποκτήθηκαν σημαντικές γνώσεις για τον σχεδιασμό και την ανάπτυξη ενός διαδικτυακού συστήματος. Συγκεκριμένα με τη βοήθεια των βασικών εργαλείων του διαδικτύου όπως η γλώσσα HTML, τα CSS, τις γλώσσες σεναρίου JavaScript και PHP αλλά και το κομμάτι του backend με τεχνολογίες Apache και MySQL, δημιουργήθηκε το Ticket Manager που διαθέτει τις βασικές λειτουργίες και παροχές ενός διαδικτυακού συστήματος κρατήσεων εισιτηρίων.

Επιλέγοντας τις PHP σελίδες για συγγραφή κώδικα, ο προγραμματισμός γίνεται εύκολος σε σχέση με άλλες γλώσσες καθώς είναι συμβατή με τα πιο γνωστά συστήματα διακομιστών και διαχείρισης βάσεων δεδομένων. Ένα ακόμα όφελος συνολικά του συστήματος είναι ότι βασίζεται πλήρως σε open source τεχνολογίες και το κόστος ανάπτυξης του είναι μηδενικό.

Ωστόσο, κάποιες δυσκολίες που αντιμετωπίστηκαν είναι στο κομμάτι διαχείρισης ημερομηνίας όπου η μορφή έπρεπε να αλλάζει κατάλληλα κάθε φορά για εμφάνιση στον χρήστη ή για σύγκριση με εγγραφή στην βάση. Άλλη μια δυσκολία αντιμετωπίστηκε στην ασύγχρονη αλλαγή στοιχείων περιεχομένου HTML κατ' επανάληψη, χωρίς ανανέωση της σελίδας, όπου αντικαταστάθηκε με υποχρεωτική ανανέωση για ομαλή λειτουργία.

Κάποιες περεταίρω δυνατότητες που θα ήταν ενδιαφέρον να υλοποιηθούν σε τυχόν επέκταση της εφαρμογής αυτής, είναι η προσθήκη λίστας ετικετών οι οποίες θα διευκολύνουν την πλοήγηση και θα δώσουν περισσότερη έμφαση στην αναζήτηση γεγονότων. Ακόμα η ύπαρξη προτεινόμενων γεγονότων βάση προηγούμενων προτιμήσεων του χρήστη και η δυνατότητα αξιολόγησης γεγονότος από κάποιον που ήδη έχει παρακολουθήσει το δρώμενο θα έκαναν την εφαρμογή πιο προσιτή και αξιόπιστη. Επίσης ένας μηχανισμός για την ανάδειξη κορυφαίων δρώμενων θα ήταν χρήσιμος.

Βιβλιογραφικές αναφορές

Βιβλία

- [1] Randy Connolly, Ricardo Hoar, *Προγραμματισμός για το Web*. (2015). Εκδόσεις Μ. Γκιούρδας
- [2] Turban. E., King. D., Lee. J. K., Liang, T. P., and Turban, D.C., (2015), *Electronic Commerce: A Managerial and Social Networks Perspective*. 8th Ed., Publisher: Springer International Publishing.
- [3] Nanehkaran, Y., *An Introduction To Electronic Commerce*, (2013) International Journal Of Scientific & Technology Research, ISSN 2277-8616, 190-193
- [4] Ελευθέριος Αθ. Παπαθανασίου, *Εισαγωγή στις Νέες Τεχνολογίες και το Ηλεκτρονικό Εμπόριο*. (2011)
- [5] Chaffey, D., *Ψηφιακές επιχειρήσεις και Ηλεκτρονικό εμπόριο*, (Π. Αρκουδέας, Μεταφ.) (2016). Αθήνα: Κλειδάριθμος.

Internet sites

- [6] <https://repository.kallipos.gr/>
- [7] <https://sites.google.com/site/adiseteithe/home>
- [8] <https://www.wikipedia.org/>
- [9] <https://www.php.net/>
- [10] <https://www.w3schools.com/>
- [11] <https://www.apache.org/>

Βοηθήματα

- [12] https://www.tutorialspoint.com/execute_sql_online.php