



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΕΠΙΧΕΙΡΗΣΕΩΝ

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

(πρώην Τμήμα Διοίκησης Επιχειρήσεων – Μεσολόγγι)

Πτυχιακή εργασία

«ΔΗΜΙΟΥΡΓΙΑ ΔΥΝΑΜΙΚΗΣ RESPONSIVE
ΙΣΤΟΣΕΛΙΔΑΣ ΓΙΑ ΕΝΑΝ ΜΟΥΣΙΚΟ ΜΕ ΤΗ
ΧΡΗΣΗ JSP/SERVLETS»

Επιβλέπων καθηγητής

Ιωάννης Ντόκας

Ευστάθιος Ευσταθιάδης

Μεσολόγγι 2020

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Διοίκησης Επιχειρήσεων/Μεσολογίου του ΤΕΙ Δυτικής Ελλάδας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	ii
ABSTRACT	ii
1. Εισαγωγή.....	1
1.1 Περιγραφή της ιστοσελίδας.....	1
1.2 Που απευθύνεται	1
2. Δημιουργία ιστότοπου.....	2
2.1 Περιγραφή.....	2
3. Ανάλυση απαιτήσεων ιστότοπου	4
3.1 Τι σημαίνει Responsive	4
4. Τεχνολογίες	9
5. Ανάλυση τεχνολογιών.....	9
5.1 Java Servlets	9
5.2 Java Server Pages (JSP)	12
5.3 Front-end Τεχνολογίες	12
6. Δόμηση κώδικα	14
6.1 Διαχωρισμός κλάσεων.....	14
6.2 Hibernate Framework.....	15
6.3 Φόρμα επικοινωνίας και newsletter	16
6.4 Σύνδεση και Εγγραφή χρήστη.....	19
6.5 Επεξεργασία στοιχείων χρήστη.....	21
6.6 Καταχώριση και επεξεργασία άρθρων.....	24
6.7 Καταχώριση και επεξεργασία εγγραφών μενού.....	28
7. Παρουσίαση ιστότοπου	32
7.1 Περιγραφή.....	32
7.2 Νέος Χρήστης	46
7.3 Σελίδα Χρήστη	48
7.4 Σύστημα διαχείρισης περιεχομένου	50
8. Επίλογος	56
9. Βιβλιογραφία.....	56

ΠΕΡΙΛΗΨΗ

Αντικείμενο της εργασίας αποτελεί η σχεδίαση και η ανάπτυξη μίας δυναμικής responsive ιστοσελίδας για έναν μουσικό, με τη χρήση JSP/Servlets. Η ιστοσελίδα περιλαμβάνει ένα σύστημα διαχείρισης περιεχομένου και έχει ως κύρια λειτουργία την καταχώριση άρθρων και την ενημέρωση στοιχείων που αφορούν τον μουσικό. Πιο αναλυτικά, η ιστοσελίδα παρέχει τη δυνατότητα στον διαχειριστή να κάνει καταχώριση και διαχείριση άρθρων και να ανανεώνει το βασικό μενού πλοήγησης, το οποίο δημιουργείται δυναμικά. Επίσης, δίνει στον χρήστη τη δυνατότητα επικοινωνίας με τον μουσικό μέσω μιας φόρμας αλλά και τη δυνατότητα εγγραφής σε newsletter για άμεση ενημέρωση. Τέλος, οι εγγεγραμμένοι χρήστες της ιστοσελίδας έχουν την δυνατότητα υποβολής σχολίων στα άρθρα της ιστοσελίδας.

Λέξεις κλειδιά: Ιστοσελίδα, Responsive, JSP/Servlets,

ABSTRACT

The purpose of the project is to design and develop a dynamic responsive website for a musician using JSP / Servlets. The site includes a content management system with the primary function of listing articles and updating information related to the musician. In more detail, the site allows the administrator to register and manage articles and to refresh the dynamically created main navigation menu. It also allows the user to communicate with the musician through a form as well as the ability to subscribe to a newsletter for immediate updates regarding the musician. Finally, registered users of the site are able to comment on the articles in the site.

1. Εισαγωγή

1.1 Περιγραφή της ιστοσελίδας

Αντικείμενο της παρούσας πτυχιακής εργασίας αποτελεί η σχεδίαση και η ανάπτυξη μίας δυναμικής responsive ιστοσελίδας για τον μουσικό Σωκράτη Μάλαμα, με τη χρήση JSP/Servlets. Ο συγκεκριμένος καλλιτέχνης, ενώ έχει επίσημη ιστοσελίδα στο Youtube και στο Facebook και ενώ μπορεί κάποιος στο internet να διαβάσει πληθώρα άρθρων για τον ίδιο και τη δουλειά του, δεν έχει προσωπική ιστοσελίδα με συγκεντρωμένη όλη την πληροφορία για τη μουσική του πορεία και τα νέα του.

Η ιστοσελίδα που υλοποιήσαμε περιλαμβάνει ένα σύστημα διαχείρισης περιεχομένου και έχει ως κύρια λειτουργία την καταχώριση άρθρων και την ενημέρωση στοιχείων που αφορούν στον ίδιο και στη δουλειά του.

Πιο αναλυτικά, η ιστοσελίδα παρέχει τη δυνατότητα στον διαχειριστή της να κάνει καταχώριση και διαχείριση άρθρων και να ανανεώνει το βασικό μενού πλοήγησης, το οποίο δημιουργείται δυναμικά αντλώντας πληροφορία από τον αντίστοιχο πίνακα στη βάση δεδομένων. Επίσης, δίνει στον χρήστη τη δυνατότητα επικοινωνίας με τον μουσικό μέσω μιας φόρμας αλλά και τη δυνατότητα εγγραφής σε newsletter για άμεση ενημέρωση. Επιπλέον, οι εγγεγραμμένοι χρήστες της ιστοσελίδας έχουν τη δυνατότητα υποβολής σχολίων στα άρθρα της ιστοσελίδας.

Τέλος, η ιστοσελίδα «Σωκράτης Μάλαμας» περιέχει πληροφορίες για τη βιογραφία του, τη δισκογραφία και τις προγραμματισμένες συναυλίες, βίντεο και φωτογραφίες από τις συναυλίες του καθώς και πληροφορίες για τους συνεργάτες του.

1.2 Που απευθύνεται

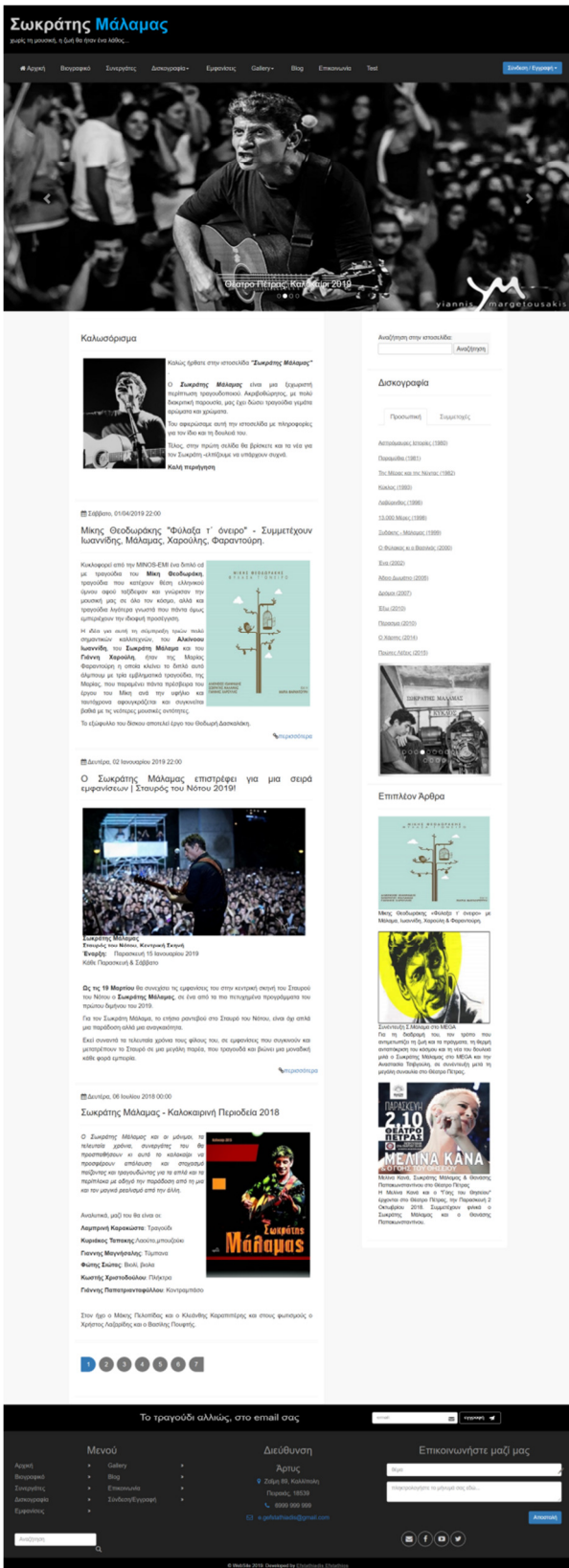
Ο Σωκράτης Μάλαμας είναι ένας από τους πιο γνωστούς Έλληνες συνθέτες και ερμηνευτές και ένα από τα πιο χαρακτηριστικά παραδείγματα του λεγόμενου έντεχνου τραγουδιού. Η ιστοσελίδα «Σωκράτης Μάλαμας» απευθύνεται σε όλους εκείνους που ενδιαφέρονται να μάθουν τα τελευταία νέα σχετικά με τον μουσικό, τη δουλειά του και τους συνεργάτες του.

2. Δημιουργία ιστότοπου

2.1 Περιγραφή

Η ιστοσελίδα «Σωκράτης Μάλαμας» έχει ένα σύντομο μενού με τα απαραίτητα στοιχεία (Βιογραφικό, Συνεργάτες, Δισκογραφία, Εμφανίσεις, Gallery, Blog, Επικοινωνία) για τον μουσικό και απευθύνεται στοχευμένα στα άτομα που αγαπούν τον συγκεκριμένο μουσικό.

Πιο αναλυτικά, στην αρχική σελίδα επιλέχθηκε ένα slider με 4 φωτογραφίες του μουσικού από συναυλίες που έχει πραγματοποιήσει (Εικόνα 2.1.1). Επίσης, στην αρχική σελίδα ο επισκέπτης μπορεί να διαβάσει άρθρα για τη δισκογραφία, τις εκδηλώσεις, τους διαγωνισμούς κ.α που αφορούν το ίδιο τον μουσικό και τους συνεργάτες του.



Εικόνα 2.1.1: Αρχική σελίδα ιστότοπου

Με το μενού «Βιογραφικό» ο ιστότοπος «Σωκράτης Μάλαμας» έχει ως στόχο να ενημερώσει τους επισκέπτες για το βιογραφικό, τη δισκογραφία, τις συνεργασίες και τις εμφανίσεις του μουσικού.

Στο μενού «Συνεργάτες», ο επισκέπτης του ιστότοπου έχει τη δυνατότητα να ενημερωθεί για τους σταθερούς συνεργάτες του καλλιτέχνη, που τον ακολουθούν για πολλά χρόνια. Επίσης, οι επισκέπτες ενημερώνονται πιο αναλυτικά για τη δισκογραφία και τις συναυλίες του, στο μενού «Δισκογραφία» και «Εμφανίσεις» αντίστοιχα. Επίσης, στο μενού «Gallery» ο επισκέπτης μπορεί να δει βίντεο και φωτογραφίες του Σωκράτη Μάλαμα από συναυλίες που έχει πραγματοποιήσει στο παρελθόν.

Ο επισκέπτης που επιθυμεί να επικοινωνήσει με τον μουσικό έχει τη δυνατότητα στο μενού «Επικοινωνία» να αποστείλει φόρμα επικοινωνίας στον διαχειριστή του ιστότοπου καθώς και να εγγραφεί στο newsletter. Τέλος, στην ενότητα «Blog» ο επισκέπτης ενημερώνεται για όλα τα νέα του καλλιτέχνη.

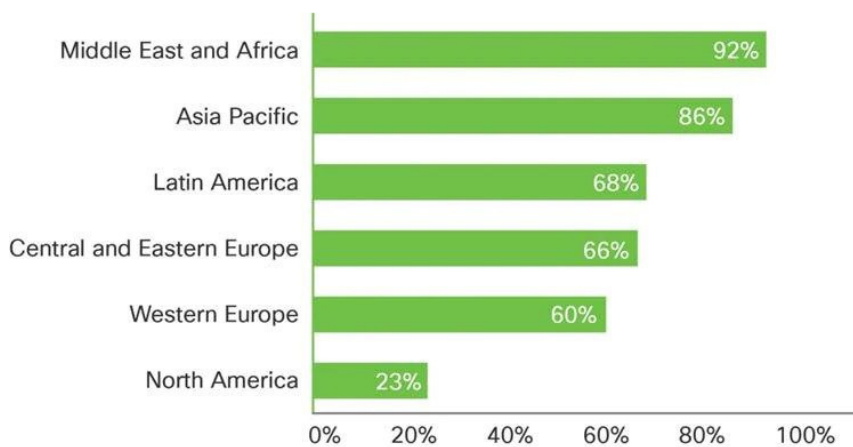
3. Ανάλυση απαιτήσεων ιστότοπου

Αρχικά, ο ιστότοπος, ο οποίος περιλαμβάνει μενού, νέα, φόρμα επικοινωνίας καθώς και ένα σύστημα περιήγησης περιεχομένου, διαθέτει βάση δεδομένων MySQL. Με τη βοήθεια της MySQL γίνονται τα κατάλληλα ερωτήματα και μπορεί ο διαχειριστής να αντλήσει και να ενημερώσει τα δεδομένα του ιστότοπου μέσω του συστήματος διαχείρισης περιεχομένου. Ο ιστότοπος φροντίσαμε να έχει μια ιεραρχική δομή, να είναι ιδιαίτερα φιλικός προς τους χρήστες και δομήσαμε την πληροφορία με τέτοιο τρόπο ώστε ο χρήστης να μπορεί με λίγα κλικ να βρίσκει την πληροφορία που χρειάζεται και να περιηγείται εύκολα σε αυτόν. Επίσης, ο ιστότοπος είναι responsive και προσαρμόσιμος σε ένα σημαντικό εύρος από συσκευές, όπως τα κινητά τηλέφωνα ή τα tablets. Στόχος μας ήταν να πετύχουμε το βέλτιστο αποτέλεσμα τόσο όσον αφορά την αισθητική του ιστότοπου όσο και τη λειτουργικότητά του.

3.1 Τι σημαίνει Responsive

Την τελευταία δεκαετία θεωρείται απαραίτητο σε κάθε ιστοσελίδα στο διαδίκτυο να έχει γίνει σχεδιασμός ο οποίος θα επιτρέπει σε αυτήν να προσαρμόζεται ανάλογα με το πλάτος της οθόνης που χρησιμοποιεί ο χρήστης.

Πιο συγκεκριμένα, το 2015 η Google προσάρμοσε τον αλγόριθμό της μηχανής αναζήτησης έτσι ώστε να «επιβραβεύονται» όσες ιστοσελίδες έχουν ένα προσαρμοζόμενο (responsive) περιεχόμενο, βάσει πάντα του πλάτους της οθόνης της συσκευής [17]. Μεγάλο ρόλο στην αλλαγή που παρουσιάζεται παραπάνω, όπως και σε πολλές άλλες, είναι η ραγδαία εξέλιξη της τεχνολογίας στις κινητές συσκευές αλλά και η δυνατότητα στον χρήστη να έχει πλέον πρόσβαση από κάθε σημείο στο διαδίκτυο. Στην Εικόνα 7.5.1 [8] παρατηρούμε αύξηση κατά 71% χρόνο με τον χρόνο, στην κίνηση δεδομένων μέσω κινητών συσκευών και με ποσοστά να αγγίζουν το 92% για χώρες όπως η Μέση ανατολή και η Αφρική.

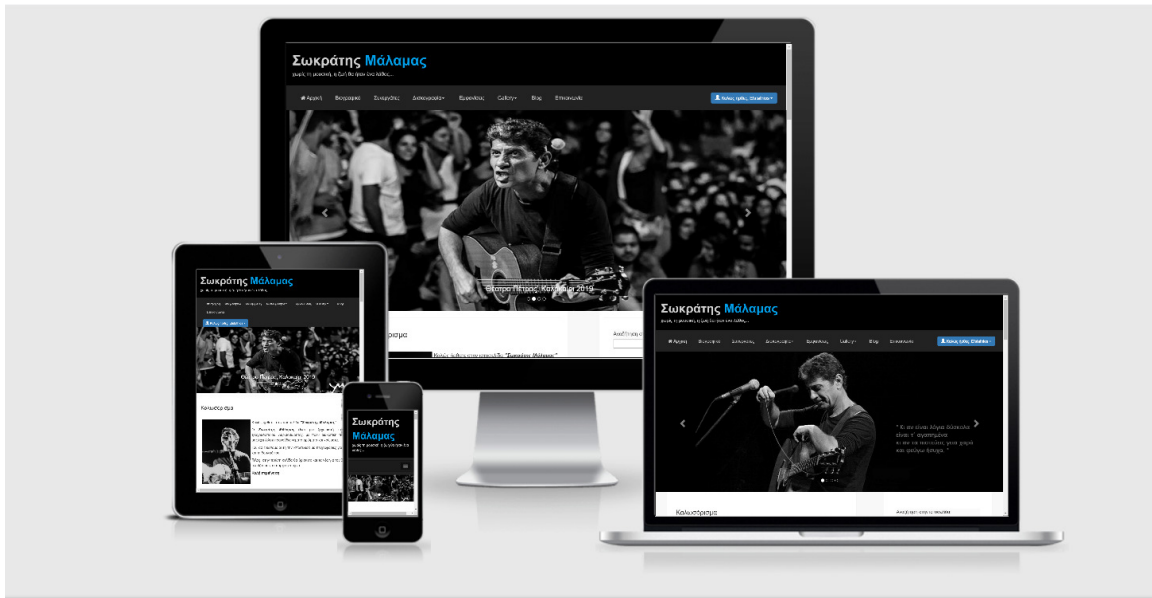


Εικόνα 7.5.1 Αύξηση της μετάδοσης δεδομένων μέσω κινητών συσκευών [8]

Μελετώντας τα παραπάνω δεδομένα είναι ξεκάθαρη η ανάγκη για μια responsive ιστοσελίδα. Για αυτό τον λόγο έχουμε χρησιμοποιήσει και το σύνολο εργαλείων (framework) της Bootstrap στην ιστοσελίδα «Σωκράτης Μάλαμας» το οποίο μας δίνει την δυνατότητα για τη δημιουργία responsive στοιχείων στην σελίδα μας (Εικόνα 7.5.1, 7.5.2, 7.5.3).

Μερικά από τα πλεονεκτήματα μίας responsive σελίδας είναι ότι μπορούμε να δημιουργήσουμε ένα πιο φιλικό προς τον χρήστη περιβάλλον για να περιηγηθεί στην ιστοσελίδα μας, το οποίο θα έχει ως αποτέλεσμα και την αύξηση της επισκεψιμότητας στην ιστοσελίδα μας. Επίσης, θα υπάρχει χαμηλότερος ρυθμός εγκατάλειψης από τον χρήστη με αποτέλεσμα την αύξηση του χρόνου παραμονής και περιήγησης στην ιστοσελίδα μας [16]. Μπορούμε επίσης να μειώσουμε το κόστος κατασκευής και συντήρησης της ιστοσελίδας μας, καθώς δεν θα υπάρξει η ανάγκη για την ανάπτυξη ξεχωριστής εφαρμογής για την παρουσίαση της ιστοσελίδας [7]. Τέλος, θα μπορέσει να επιτευχθεί χαμηλότερος χρόνος

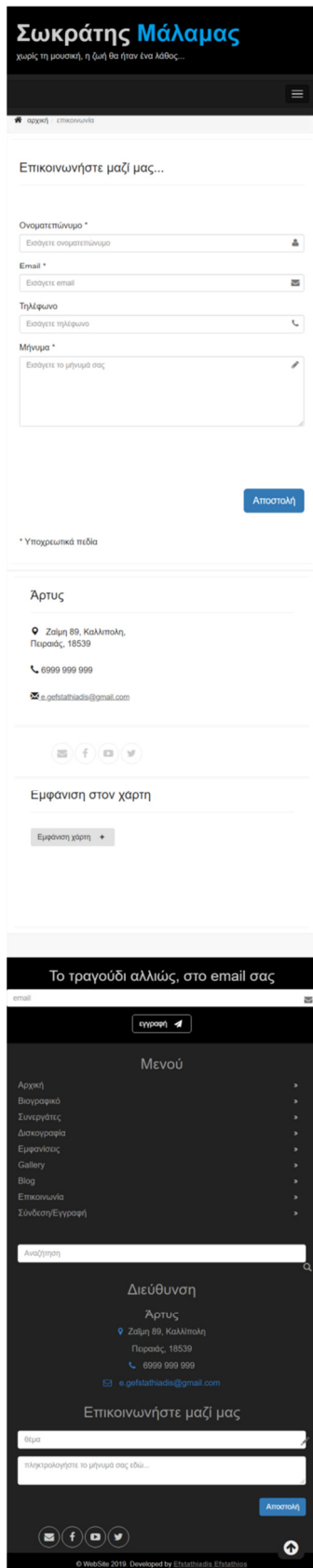
φόρτωσης της ιστοσελίδας καθώς θα φορτώνονται στον χρήστη μόνο τα στοιχεία και οι πληροφορίες που έχουμε ορίσει εμείς ως απαραίτητες σε μια κινητή συσκευή.



Εικόνα 3.1.1: Ο ιστότοπος σε κινητές συσκευές

The screenshot displays a search result for the film "Σουκράτης Μάλαμας" (Sokratis Malamas). The page includes a header with the movie title and a small image. Below the header, there is a section for "Πλοκή/Σύνοψη" (Synopsis) with a small image of the main character. The main content area features a large image of the movie poster and a detailed synopsis in Greek. The synopsis describes the plot involving a young man named Sokratis Malamas who is involved in a complex situation. Below the synopsis, there is a section for "Πληροφορίες" (Information) with a table listing details such as the director (Dimitris Malamas), the year (2010), and the genre (Drama). The page also includes a "Διαμοιρασμοί" (Sharing) section with social media icons and a "Διαμοιρασμοί" (Sharing) section with a list of sharing options. At the bottom, there is a "Τι τραγουδά ολόκληρο, στο email σας" (What you sing, in your email) section with a search bar and a list of songs.

Εικόνα 3.1.2: Αρχική σελίδα σε κινητή συσκευή



Εικόνα 3.1.3: Σελίδα επικοινωνία σε κινητή συσκευή

4. Τεχνολογίες

Παρόμοιες ιστοσελίδες μουσικών του λεγόμενου έντεχνου τραγουδιού χρησιμοποιούν εργαλεία διαχείρισης περιεχομένου, όπως το Joomla και το WordPress ή είναι custom και χρησιμοποιούν PHP server-side γλώσσα προγραμματισμού (πχ alkinoos.gr, kotsiras.gr, mikistheodorakis.gr, hainides.gr). Στην ιστοσελίδα «Σωκράτης Μάλαμας» επιλέξαμε να δημιουργήσουμε το δικό μας σύστημα διαχείρισης περιεχομένου και να υλοποιήσουμε τον ιστότοπο με πληθώρα τεχνολογιών που μας επιτρέπουν να αποδώσουμε το περιεχόμενο της ιστοσελίδας χωρίς να περιοριζόμαστε σε έτοιμα templates. Για την παρουσίαση του δικού μας template και του τμήματος που μπορεί ο χρήστης να αλληλοεπιδράσει καθώς και για την προβολή του μέσω φυλλομετρητή, χρησιμοποιήσαμε τις τεχνολογίες HTML, Javascript, CSS, JQuery-(JS framework) και Bootstrap (Front-End). Για τις λειτουργίες που εκτελούνται στο «παρασκήνιο» χρησιμοποιήσαμε τα JSP/Servlets και για την απλοποίηση της διαδικασίας που χρειάζεται η σελίδα μας για να αλληλοεπιδράσει με τη βάση δεδομένων MySQL χρησιμοποιήσαμε το Hibernate Framework (Back-end). Παραδείγματα back-end λειτουργιών στην ιστοσελίδα μας είναι η συμπλήρωση της φόρμα επικοινωνίας, η εγγραφή στο newsletter, η ενημέρωση του μενού και η καταχώριση νέων άρθρων.

5. Ανάλυση τεχνολογιών

5.1 Java Servlets

Η Java είναι μια γλώσσα αντικειμενοστραφούς προγραμματισμού που έχει σχεδιαστεί με σκοπό να έχει όσο το δυνατόν λιγότερες εξαρτήσεις με την εφαρμογή. Ο γενικός σκοπός της γλώσσας είναι οι προγραμματιστές να γράψουν μια φορά τον κώδικα που χρειάζεται και μετά η εφαρμογή/κώδικας να τρέχει σε κάθε μέσο που μπορεί να υποστηρίξει Java χωρίς να χρειαστεί να συνταχθεί ξανά.

Τα Java Servlets τα οποία χρησιμοποιούμε και στην παρούσα εργασία, είναι η απάντηση της Java στις δυναμικές σελίδες που δημιουργούνται μέσω CGI σεναρίων και τα οποία εκτελούνταν ως μια ξεχωριστή διαδικασία στην πλευρά του εξυπηρετητή ιστού. Τα Java Servlets είναι κλάσεις οι οποίες μπορούν να λειτουργήσουν στην μεριά του εξυπηρετητή και

να χρησιμοποιηθούν ως μια βοηθητική γλώσσα η οποία θα μπορέσει να δημιουργήσει το περιεχόμενο μιας σελίδας δυναμικά, βάσει πάντα των παραμέτρων/αιτήσεων που δέχτηκε από τον φυλλομετρητή-πελάτη.

Σκοπός της ιστοσελίδας «Σωκράτης Μάλαμας» είναι η δημιουργία σελίδων που παράγονται δυναμικά σύμφωνα με τα αιτήματα του χρήστη που γίνονται με τη βοήθεια ενός φυλλομετρητή. Απαραίτητη προϋπόθεση για να λειτουργήσει σωστά μία Servlet κλάση είναι ο εξυπηρετητής ιστού ή εφαρμογών να έχει ενσωματωμένο το Servlet Container. Στην παρούσα εργασία τον ρόλο ενός servlet container τον έχει αναλάβει ο Apache Tomcat [11]. Μέσω των servlets τα οποία ορίζονται στο πακέτο javax.servlet μπορούν να δηλωθούν τρεις βασικές μέθοδοι που αφορούν τον κύκλο ζωής ενός servlet – init (), service() και destroy(). Εφαρμόζονται από κάθε servlet (καθορισμένο σε SDK ή αυτοπροσδιορισμένο) και χρησιμοποιούνται σε συγκεκριμένες χρονικές στιγμές από τον εξυπηρετητή.

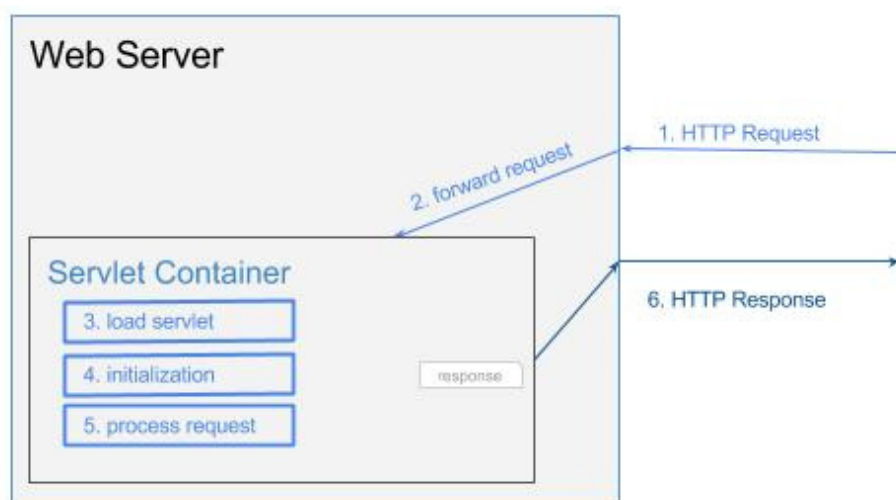
1. Η μέθοδος init() ενεργοποιείται κατά το στάδιο της αρχικοποίησης του κύκλου ζωής του servlet. Δημιουργείται ένα αντικείμενο που υλοποιεί τη διεπαφή javax.servlet.ServletConfig, η οποία επιτρέπει στο servlet να έχει πρόσβαση στις παραμέτρους αρχικοποίησης από την εφαρμογή ιστού.
2. Η μέθοδος service() ενεργοποιείται όταν η αίτηση από τον χρήστη έχει περάσει το στάδιο της αρχικοποίησής της. Κάθε αίτηση εξυπηρετείται στο δικό της ξεχωριστό νήμα. Το web container καλεί τη μέθοδο service() του servlet για κάθε αίτημα ενώ παράλληλα η ίδια μέθοδος καθορίζει το είδος της αίτησης που γίνεται και την αποστέλλει στην κατάλληλη μέθοδο του προγράμματος για να την χειριστεί.
3. Τέλος η μέθοδος destroy() ενεργοποιείται όταν το αντικείμενο του servlet πρέπει να καταστραφεί. Με αυτό τον τρόπο απελευθερώνονται όσοι πόροι είχαν χρησιμοποιηθεί κατά την επεξεργασία του αιτήματος.

Από τον κύκλο ζωής του servlet μπορούμε να καταλάβουμε ότι οι κλάσεις του φορτώνονται δυναμικά σε έναν container με τη βοήθεια ενός φορτωτή κλάσης. Κάθε αίτημα επεξεργάζεται στο δικό του νήμα και ένα αντικείμενο servlet μπορεί να εξυπηρετήσει πολλαπλά νήματα ταυτόχρονα. Όταν δεν χρησιμοποιείται πλέον θα θεωρηθεί ως ένα απρόσιτο αντικείμενο, και ο αποκομιστής απορριμμάτων (garbage collector) της JVM (Java Virtual Machine) θα το συλλέξει.

Όπως κάθε πρόγραμμα Java έτσι και το servlet λειτουργεί μέσα σε ένα JVM. Ο λόγος που είναι χρήσιμο ένα servlet container είναι για να αντιμετωπισθούν οι πολυπλοκότητες των HTTP αιτήσεων. Το servlet container είναι υπεύθυνο για τη δημιουργία, την εκτέλεση και την καταστροφή των Servlets.

Παρακάτω αναλύεται και ο τρόπος με τον οποίο ένας εξυπηρετητής ιστού και ένα servlet container επεξεργάζονται ένα αίτημα [10]:

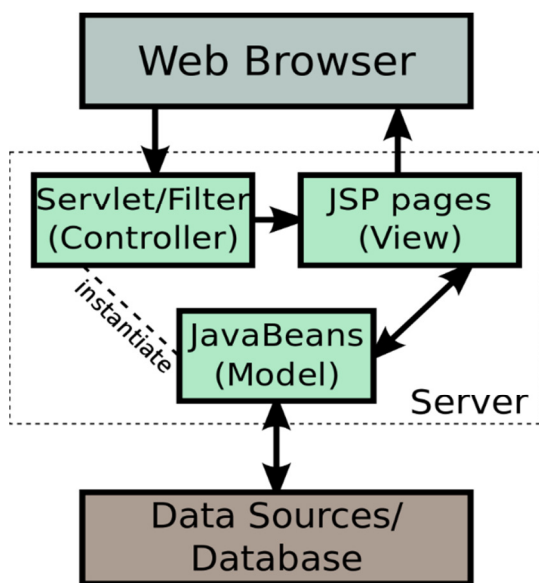
1. Ο εξυπηρετητής ιστού λαμβάνει ένα αίτημα HTTP.
2. Ο εξυπηρετητής ιστού προωθεί το αίτημα στο servlet container
3. Το servlet καλείται δυναμικά και φορτώνεται στο χώρο διεύθυνσης του container, αν δεν βρίσκεται μέσα σε αυτό.
4. Το Container καλεί τη μέθοδο `init()` του servlet για την αρχικοποίηση (που ενεργοποιείται μία φορά όταν το servlet φορτώνεται για πρώτη φορά).
5. Το Container καλεί τη μέθοδο `service()` του εξυπηρετητή για να επεξεργαστεί το αίτημα HTTP, δηλαδή να διαβάσει τα δεδομένα που έχουν σταλεί ως αίτημα και να δομίσει μία απάντηση. Το servlet παραμένει στο χώρο διευθύνσεων του container και μπορεί να επεξεργαστεί άλλες αιτήσεις HTTP.
6. Τέλος, ο εξυπηρετητής ιστού μπορεί να επιστρέψει τα δυναμικά δημιουργούμενα αποτελέσματα στη σωστή θέση.



Εικόνα 5.1.1 : Επεξεργασία αιτήματος πελάτη. [10]

5.2 Java Server Pages (JSP)

Τα JSP είναι αρχεία της Java που έρχονται για να καλύψουν τα κενά που μπορούν να δημιουργηθούν από τα Servlets ως προς την απεικόνιση μιας δυναμικής σελίδας. Με τα JSP μπορούμε να ενσωματώσουμε Java μέσα σε HTML. Με αυτό τον τρόπο τα Servlets είναι υπεύθυνα για την επεξεργασία των αιτημάτων που θα γίνουν από τον πελάτη και τα JSP αρχεία είναι υπεύθυνα να επιστρέψουν και να εμφανίσουν τα αποτελέσματα. Με τον παραπάνω διαχωρισμό των λειτουργιών επιτυγχάνουμε έναν ευανάγνωστο κώδικα και αποφεύγουμε την συνεχόμενη επανάληψη την μεθόδου `out.println` για να εμφανίσουμε το περιεχόμενο της σελίδας [3].



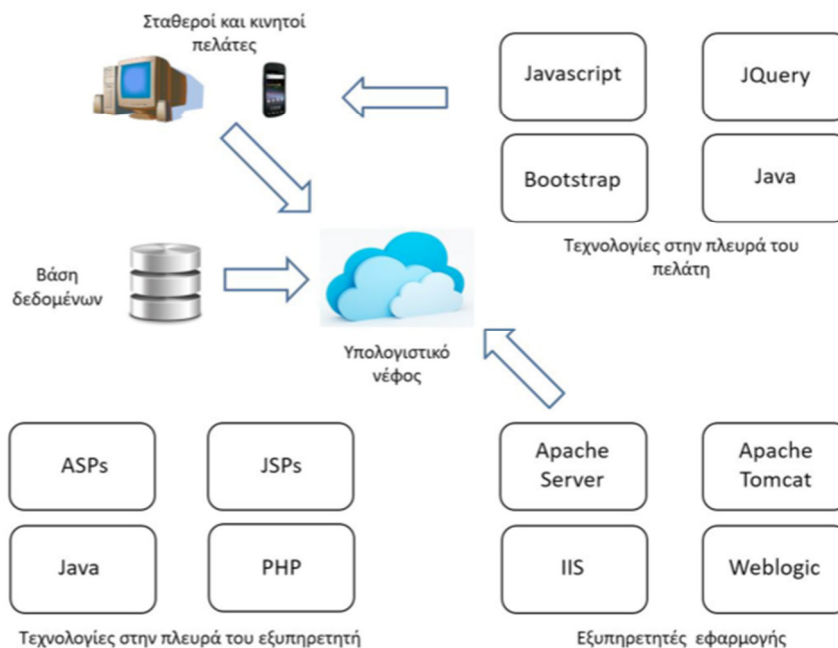
Εικόνα 5.2.1 : Μοντέλο αρχιτεκτονικής ενός JSP [12]

5.3 Front-end Τεχνολογίες

Η HTML (HyperText Markup Language), η CSS (Cascading Style Sheets) και η Javascript είναι κώδικες που εκτελούνται από την πλευρά του πελάτη (client-side). Με αυτόν τον τρόπο μπορούμε να δημιουργήσουμε έναν κώδικα που μπορεί να εκτελεστεί στην πλευρά του πελάτη με τη βοήθεια ενός φυλλομετρητή (browser) έτσι ώστε να μην είναι απαραίτητη η επεξεργασία από την πλευρά του εξυπηρετητή. Πιο συγκεκριμένα [5]:

- HTML: Είναι τα θεμελιώδη δομικά στοιχεία ενός προγραμματισμού ιστού και είναι υπεύθυνα για τη διάταξη του περιεχομένου μιας ιστοσελίδας.

- CSS: Παρέχει τον τρόπο σχεδιασμού των γραφικών στοιχείων που βοηθούν σε μια πιο ελκυστική εμφάνιση της ιστοσελίδας μας.
- Javascript: Είναι μια γλώσσα προγραμματισμού για τον ιστό όπου μπορεί να ενημερώσει και να αλλάξει τον τρόπο παρουσίασης ενός HTML στοιχείου και ενός CSS. Επίσης μπορεί να χρησιμοποιηθεί για να υπολογίσει, να χειριστεί και να επικυρώσει τα δεδομένα που μπορεί να λάβει από τον χρήστη. Ένα κλασικό παράδειγμα είναι ο έλεγχος των δεδομένων που έχει εισάγει ο χρήστης σε μια φόρμα επικοινωνίας.
- JQuery Είναι μια βιβλιοθήκη της Javascript που έχει σχεδιαστεί για να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην μεριά του πελάτη και παρέχει τη δυνατότητα στον προγραμματιστή να έχει το ίδιο αποτέλεσμα όπως μια Javascript αλλά με λιγότερες γραμμές κώδικα. Η JQuery προσφέρει στον προγραμματιστή τις παρακάτω δυνατότητες: Χειρισμό των στοιχείων HTML / DOM, μεθόδους συμβάντων HTML εφέ και κινούμενα σχέδια, και βοηθητικά προγράμματα AJAX.



Εικόνα 5.3.1 : Front-end και Back-end Τεχνολογίες [3]

6. Δόμηση κώδικα

6.1 Διαχωρισμός κλάσεων

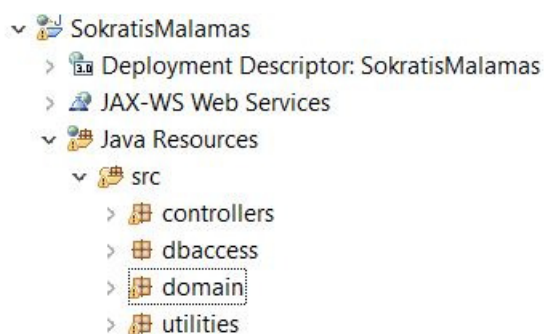
Στην ιστοσελίδα «Σωκράτης Μάλαμας» οι κλάσεις χωρίστηκαν στα παρακάτω πακέτα (Εικόνα 6.1.1):

Στο πρώτο πακέτο με όνομα `controllers` έχουμε δημιουργήσει τα Servlets που είναι υπεύθυνα για την επεξεργασία των αιτημάτων του χρήστη. Επιπλέον, έχουμε δημιουργήσει κάποιες κλάσεις Java οι οποίες περιλαμβάνουν όλες εκείνες τις συναρτήσεις που είναι απαραίτητες για τη διαχείριση των αντικειμένων του domain model.

Στο δεύτερο πακέτο με όνομα `dbaccess` έχουμε δημιουργήσει την κλάση της αρχικοποίησης της βάσης (`initialize.java`) η οποία όταν ξεκινήσει θα γεμίσει τη βάση με δεδομένα που έχουμε ορίσει. Στο ίδιο πακέτο θα βρούμε και την κλάση `JPAUtil.java` η οποία είναι υπεύθυνη για την σύνδεση με τη βάση δεδομένων.

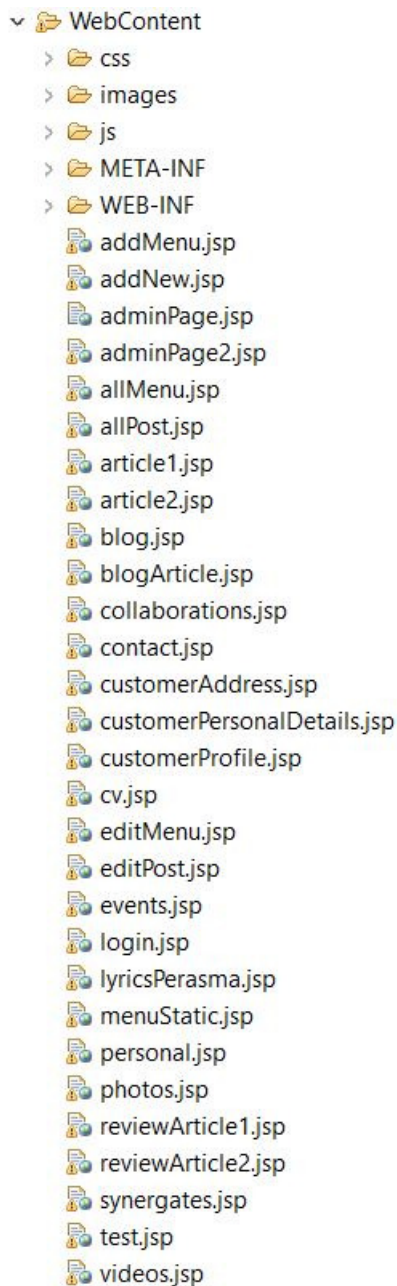
Στο τρίτο πακέτο με όνομα `domain` έχουμε δημιουργήσει τις κλάσεις οι οποίες είναι υπεύθυνες για τη δημιουργία των πινάκων της βάσης μας. Έχουμε τον πίνακα `Customer` στον οποίο αποθηκεύονται οι χρήστες του ιστότοπου, τον πίνακα `Address` που αποθηκεύονται τα στοιχεία διεύθυνσης των χρηστών, ο πίνακας `News` με όλα τα άρθρα του ιστότοπου και τέλος ο πίνακας `Menu` όπου περιλαμβάνει τις εγγραφές του μενού.

Στο τέταρτο πακέτο με όνομα `utilities` έχουμε δημιουργήσει τις κλάσεις οι οποίες είναι υπεύθυνες για κάποιες από τις λειτουργίες που υπάρχουν στον ιστότοπο, όπως για παράδειγμα η δημιουργία και η αποστολή mail στους χρήστες αλλά και η δημιουργία και αποστολή του κωδικού χρήστη.



Εικόνα 6.1.1: Διαχωρισμός κλάσεων

Τέλος, μέσα στον φάκελο WebContent (Εικόνα 6.1.2) βλέπουμε όλες τις σελίδες που είναι υπεύθυνες για τη δημιουργία της ιστοσελίδας και αφορούν το Front-end . Επίσης, βλέπουμε τον φάκελο με όλες τις φωτογραφίες που υπάρχουν στη σελίδα μας αλλά και τα αρχεία με τους κώδικες CSS και Javascript που χρησιμοποιούμε.



Εικόνα 6.1.2: Οι σελίδες του ιστότοπου

6.2 Hibernate Framework

Το Hibernate framework που χρησιμοποιήθηκε στην υλοποίηση της σελίδας, είναι ένα εργαλείο που σκοπός του είναι να συνδέσει τα αντικείμενα που δημιουργούνται από τον

αντικειμενοστραφή κώδικα, με τους αντίστοιχους πίνακες μέσα στη βάση δεδομένων μας. Η παραπάνω διαδικασία μπορεί να επιτευχθεί με την προσθήκη επιπρόσθετων πληροφοριών (metadata) όπου θα περιγράφουν την αντιστοιχία που υπάρχει μεταξύ των αντικειμένων και πινάκων μέσα στην βάση.

6.3 Φόρμα επικοινωνίας και newsletter

Ο χρήστης αφού έχει συμπληρώσει τη φόρμα επικοινωνίας και πατήσει το κουμπί «Αποστολή» τότε έχουμε τις παρακάτω λειτουργίες (Εικόνα 6.3.1):

```

2
3 <%@ page import="javax.mail.MessagingException"%>
4
5<%
6     String message = null;
7     String status = null;
8     if (request.getParameter("submit") != null) {
9         JavaEmail javaEmail = new JavaEmail();
10        javaEmail.setMailServerProperties();
11        String emailSubject = "Contact Form";
12        String emailBody = "";
13        /*if (request.getParameter("frstname") != null) {
14            emailBody = "Sender Name: " + request.getParameter("frstname")
15                + "<br>";
16        }*/
17        if (request.getParameter("fullname") != null) {
18            emailBody = "Sender Fullname: " + request.getParameter("fullname")
19                + "<br>"+ "<br>";
20        }
21        if (request.getParameter("email_cntc") != null) {
22            emailBody = emailBody + "Sender Email: "
23                + request.getParameter("email_cntc") + "<br>"+ "<br>";
24        }
25        if (request.getParameter("phone_cntc") != null) {
26            emailBody = emailBody + "Sender Phone: "
27                + request.getParameter("phone_cntc") + "<br>"+ "<br>";
28        }
29        if (request.getParameter("message_cntc") != null) {
30            emailBody = emailBody + "Message: " + request.getParameter("message_cntc")
31                + "<br>"+ "<br>";
32        }
33        javaEmail.createEmailMessage(emailSubject, emailBody);
34        try {
35            javaEmail.sendEmail();
36            status = "success";
37            message = "Το μήνυμά σας εστάλη επιτυχώς!";
38        } catch (MessagingException me) {
39            status = "error";
40            message = "Λάθος στην αποστολή του μηνύματος!";
41        }
42    }

```

Εικόνα 6.3.1: Λειτουργία κουμπιού «Αποστολή» φόρμας επικοινωνίας

Όπως βλέπουμε στον παραπάνω κώδικα, δημιουργείται ένα νέο αντικείμενο της κλάσης JavaEmail. Η κλάση JavaEmail βρίσκεται μέσα στο πακέτο utilities.

Μέσα στην JavaEmail δημιουργήσαμε τη συνάρτηση setMailServerProperties() που ορίζει τον τρόπο επικοινωνίας μεταξύ του mail server, τη συνάρτηση createEmailMessage() που είναι υπεύθυνη για τη δημιουργία του μηνύματος και τη συνάρτηση sendEmail() που είναι υπεύθυνη για την αποστολή της φόρμας επικοινωνίας στον διαχειριστή του ιστότοπου.(Εικόνα 6.3.2)

```
--
14 public class JavaEmail {
15
16     Properties emailProperties;
17     Session mailSession;
18     MimeMessage emailMessage;
19
20     String emailHost = "smtp.gmail.com";|
21     String emailPort = "587";// gmail's smtp port
22     String fromUser = "e.gefstathiadis@gmail.com";// your gmail id
23     String fromUserEmailPassword = " ";
24     String[] toEmails = { "e.gefstathiadis@gmail.com" };
25     String fromEmail = "info@sokratismalamas.com";
26
27 public void setMailServerProperties() {
28     emailProperties = System.getProperties();
29     emailProperties.put("mail.smtp.host", emailHost);
30     emailProperties.put("mail.smtp.port", emailPort);
31     emailProperties.put("mail.smtp.auth", "true");
32     emailProperties.put("mail.smtp.starttls.enable", "true");
33     emailProperties.put("mail.smtp.ssl.trust", "smtp.gmail.com");
34 }
35
36 public void createEmailMessage(String emailSubject, String emailBody)
37     throws AddressException, MessagingException, IOException {
38     mailSession = Session.getDefaultInstance(emailProperties, null);
39     emailMessage = new MimeMessage(mailSession);
40     for (int i = 0; i < toEmails.length; i++) {
41         emailMessage.addRecipient(Message.RecipientType.TO,
42             new InternetAddress(toEmails[i]));
43     }
44     emailMessage.setSubject(emailSubject);
45     emailMessage.setFrom(new InternetAddress("info@sokratismalamas.com",
46         "Sokratismalamas"));
47
48     emailMessage.setContent(emailBody, "text/html");// for a html email
49     // emailMessage.setText(emailBody);// for a text email
50
51 }
--
```

```

52
53 public void sendEmail() throws AddressException, MessagingException {
54     Transport transport = mailSession.getTransport("smtp");
55     transport.connect(emailHost, fromUser, fromUserEmailPassword);
56     transport.sendMessage(emailMessage, emailMessage.getAllRecipients());
57     transport.close();
58 }
59 }

```

Εικόνα 6.3.2: Οι συναρτήσεις `setMailServerProperties()`, `createEmailMessage ()` και `sendEmail()`

Για την αποστολή του newsletter χρησιμοποιούμε το Servlet `ContactControl.java` (Εικόνα 6.3.3) το οποίο εμπεριέχει τη μέθοδο `doPost`. Στην `doPost`, αφού έχουμε εισάγει (`import`) την κλάση `JavaEmail` από το πακέτο `utilities` (Εικόνα 6.3.4), δημιουργούμε το αντικείμενο `javaEmail` και του αναθέτουμε τις ιδιότητες που χρειάζεται για να γίνει η δημιουργία του μηνύματος επιβεβαίωσης και η αποστολή του στο email του χρήστη. Εφόσον η διαδικασία γίνει με επιτυχία, θα εμφανιστεί στον χρήστη το μήνυμα «Σας ευχαριστούμε για την εγγραφή!» και η φόρμα θα αποσταλεί.

```

protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {

    String footerEmail = request.getParameter("footer_email").toString();
    if (footerEmail != null && !footerEmail.isEmpty()) {

        try {
            // EmailProvider provider = new JMailEmailProvider();
            JavaEmail javaEmail = new JavaEmail();
            javaEmail.setMailServerProperties();

            String recipientsEmails = "";
            String msgBody = "";
            String msgSubject = "Sokratis Malamas Newsletter Registration";
            msgBody = ""
                + "Thank you for subscribing to our newsletter."
                + "\n"
                + "\n"
                + " You have been successfully added to our mailing list, keeping you up-to-date with our latest news."
                + "\n" + "\n";

            // provider.sendEmail(msgSubject, msgBody,
            // request.getParameter("footer_email").toString());
            javaEmail.createEmailMessage(msgSubject, msgBody);
            javaEmail.sendEmail();
        } catch (Exception e) {
            e.printStackTrace();
        }

        JOptionPane.showMessageDialog(null,
            "Σας ευχαριστούμε για την εγγραφή!");

        response.sendRedirect("http://localhost:8080/SokratisMalamas/faces/login.jsp");
        // JPAUtil.getSessionFactory().getCurrentSession().close();

    }

    else
        response.sendRedirect("http://localhost:8080/SokratisMalamas/faces/login.jsp");
}

```

Εικόνα 6.3.3: Η συνάρτηση `doPost`

```
import utilities.JavaEmail;
```

Εικόνα 6.3.4: Εισαγωγή της κλάσης JavaEmail από το πακέτο utilities.

6.4 Σύνδεση και Εγγραφή χρήστη

Το Servlet που είναι υπεύθυνο για την εγγραφή νέου χρήστη είναι το CreateAccountControll. Η κλάση CreateAccountControll περιέχει τη συνάρτηση doPost. Μέσα στη συνάρτηση doPost δημιουργείται ένα νέο αντικείμενο με την ονομασία customer στο οποίο του ανατίθενται οι τιμές που έχει δώσει ο χρήστης στην φόρμα εγγραφής (Εικόνα 6.4.1).

```
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    try {
        Customer customer = new Customer(new Address(
            request.getParameter("addrstreet"),
            request.getParameter("addrnumber"),
            request.getParameter("zipcode"),
            request.getParameter("city"),
            request.getParameter("country")),
            request.getParameter("firstname"),
            request.getParameter("lastname"),
            request.getParameter("email"),
            request.getParameter("telephone"),
            request.getParameter("username"), "user");
```

Εικόνα 6.4.1: Δημιουργία νέου χρήστη και ανάθεση τιμών από τη φόρμα εγγραφής

Στη συνέχεια για να μπορέσουμε να χρησιμοποιήσουμε τη συνάρτηση insertCustomer() η οποία είναι υπεύθυνη για την αποθήκευση των στοιχείων του χρήστη στον πίνακα Customers, δημιουργείται ένα νέο αντικείμενο της κλάσης CustomerMgmController. (Εικόνα 6.4.2)

```
CustomerMgmController cm = new CustomerMgmController();
cm.insertCustomer(customer);
```

Εικόνα 6.4.2: Αποθήκευση στοιχείων με χρήση της συνάρτησης insertCustomer().

Η κλάση η οποία είναι υπεύθυνη για τη δημιουργία τυχαίου κωδικού πρόσβασης για τον χρήστη, είναι η SendCodeRegistrationService που βρίσκεται στο πακέτο utilities. Για να ολοκληρωθεί η διαδικασία εγγραφής του χρήστη δημιουργούμε ένα νέο αντικείμενο της κλάσης SendCodeRegistrationService και χρησιμοποιούμε τη συνάρτηση CodeAsDefaultPassword() (Εικόνα 6.4.3).

```
SendCodeRegistrationService notificationService = new SendCodeRegistrationService();
notificationService.CodeAsDefaultPassword(customer);
```

Εικόνα 6.4.3: Δημιουργία τυχαίου κωδικού πρόσβασης.

Τέλος, αφού ο χρήστης ολοκληρώσει τη διαδικασία εγγραφής ανακατευθύνεται στην αρχική σελίδα (Εικόνα 6.4.4).

```
response.sendRedirect("http://localhost:8080/SokratisMalamas/faces/login.jsp");
JPAUtil.getSessionFactory().getCurrentSession().close();
```

Εικόνα 6.4.4: Ανακατεύθυνση χρήστη στην αρχική σελίδα.

Το Servlet που είναι υπεύθυνο για τη σύνδεση του χρήστη είναι το LoginControll το οποίο βρίσκεται στο πακέτο controllers. Μέσα στην κλάση LoginControll υπάρχει η αντίστοιχη συνάρτηση doPost στην οποία γίνεται έλεγχος για τα δεδομένα που έβαλε ο χρήστης στα πεδία της φόρμας σύνδεσης (Εικόνα 6.4.5). Δημιουργούμε ένα νέο αντικείμενο της κλάσης CustomerMgmController για να μπορούμε να χρησιμοποιήσουμε τη μέθοδο authenticateCustomer() η οποία είναι υπεύθυνη για τον έλεγχο των δεδομένων ενός εγγεγραμμένου χρήστη με τη βάση.

```
if (request.getParameter("SignIn") != null) {

    String username = request.getParameter("username");
    String password = request.getParameter("password");

    CustomerMgmController cm = new CustomerMgmController();
    Customer c = cm.authenticateCustomer(username, password);
```

Εικόνα 6.4.5: Έλεγχος δεδομένων εγγεγραμμένου χρήστη.

Στη συνέχεια γίνεται έλεγχος για τα δικαιώματα του χρήστη. Όπως φαίνεται στον παρακάτω κώδικα, το αντικείμενο που δημιουργήσαμε πριν (c) θα χρησιμοποιήσει τη συνάρτηση getUerrights() η οποία ελέγχει τα δικαιώματα του χρήστη που έχουν οριστεί στον πίνακα customers μέσα στη βάση δεδομένων (Εικόνα 6.4.6). Αν ο χρήστης είναι ο διαχειριστής τότε ανακατευθύνει στη σελίδα allPost ενώ εάν είναι απλός χρήστης ανακατευθύνει στη σελίδα customerPersonalDetails.


```

if (c.getUserrights().equals("admin")) {
    // session.setAttribute("getAlert", "Yes");//
    request.getRequestDispatcher("/allPost.jsp").forward(
        request, response);
} else {
    // session.setAttribute("getAlert", "Yes");//
    request.getRequestDispatcher("/customerPersonalDetails.jsp")
        .forward(request, response);
}

```

Εικόνα 6.4.6: Έλεγχος δικαιωμάτων χρήστη.

Στην περίπτωση που κάποιο από τα στοιχεία παραπάνω είναι λάθος τότε εμφανίζεται το μήνυμα «Λάθος όνομα χρήστη ή/και κωδικός πρόσβασης» και ο χρήστης ανακατευθύνεται στην αρχική σελίδα (login) (Εικόνα 6.4.7).

```

else {
    JOptionPane.showMessageDialog(null,
        "Λάθος όνομα χρήστη ή/και κωδικός πρόσβασης.",
        "Login failed", JOptionPane.ERROR_MESSAGE);
    request.getRequestDispatcher("/login.jsp").forward(request,
        response);
}

```

Εικόνα 6.4.7: Μήνυμα λάθους και ανακατεύθυνση στην αρχική σελίδα.

6.5 Επεξεργασία στοιχείων χρήστη

Ο εγγεγραμμένος χρήστης αφού κάνει είσοδο στον ιστότοπο (και τα δικαιώματα που έχει είναι «απλός χρήστης») ανακατευθύνεται στη σελίδα customerPersonalDetails. Σε αυτή τη σελίδα υπάρχουν τρεις επιλογές :

1. Επεξεργασία των στοιχείων του χρήστη
2. Επεξεργασία διευθύνσεων
3. Επικοινωνία

Όταν ο χρήστης επιλέξει την επεξεργασία των στοιχείων του τότε ανακατευθύνεται στη σελίδα customerProfile. Η συγκεκριμένη σελίδα χρησιμοποιεί το Servlet με το όνομα CustomerMgmControll. Μέσα στη συνάρτηση doPost δηλώνουμε πρώτα μια μεταβλητή strAction (Εικόνα 6.5.1) η οποία θα κάνει τον έλεγχο ανάλογα με τη λειτουργία υποβολής που έχουμε δηλώσει στα κουμπιά της κάθε φόρμας.

```
HttpSession session=request.getSession();
String strAction=request.getParameter("action");
```

Εικόνα 6.5.1: Δήλωση μεταβλητής strAction.

Ακολουθεί ο έλεγχος βάσει του strAction που θα ελέγξει ποιο κουμπί υποβολής θα πατήσει ο χρήστης. Δημιουργούμε ένα νέο αντικείμενο της κλάσης CustomerMgmController για να μπορέσουμε να χρησιμοποιήσουμε τις συναρτήσεις findCustomerById() και updateCustomer(). Στη γραμμή 47 ορίζουμε μια μεταβλητή customer τύπου Customer. Χρησιμοποιούμε την μεταβλητή customer για να μπορέσουμε να τραβήξουμε τα νέα στοιχεία που έχει δώσει ο χρήστης στα πεδία της φόρμας. Στη συνέχεια χρησιμοποιώντας τη συνάρτηση updateCustomer() ενημερώνεται με τα νέα στοιχεία ο πίνακας της βάσης δεδομένων (πίνακας Customers) και αναθέτονται στο υπάρχον session τα νέα στοιχεία (Εικόνα 6.5.2)

```
42
43     if (strAction.equals("SavePro")) {
44
45         try {
46             CustomerMgmController cm = new CustomerMgmController();
47             Customer customer;
48             customer = cm.findCustomerById(Integer.parseInt(request
49                 .getParameter("id")));
50             customer.setUsername(request.getParameter("userfield")
51                 .toString());
52             customer.setPassword(request.getParameter("passfield")
53                 .toString());
54             customer.setTelephone(request.getParameter("telefield")
55                 .toString());
56             customer.setEmail(request.getParameter("mailfield").toString());
57             customer.setLastName(request.getParameter("lstnamefield")
58                 .toString());
59
60             cm.updateCustomer(customer);
61             RequestDispatcher rd = null;
62             rd = (request.getRequestDispatcher("/customerProfile.jsp"));
63             session.setAttribute("user", customer.getFirstName());
64             session.setAttribute("username", customer.getUsername());
65             session.setAttribute("password", customer.getPassword());
66             session.setAttribute("telephone", customer.getTelephone());
67             session.setAttribute("email", customer.getEmail());
68             session.setAttribute("id", customer.getId());
69
70             session.setAttribute("lastname", customer.getLastName());
71
72
```

Εικόνα 6.5.2: Συναρτήσεις στο Servlet CustomerMgmCotroll.

Όταν ο χρήστης επιλέξει την επεξεργασία των διευθύνσεών του, ανακατευθύνεται στη σελίδα `customerAddress` και η διαδικασία είναι ίδια με τη διαδικασία που περιγράψαμε παραπάνω. Οι διαφορές είναι στα πεδία που συμπληρώνει ο χρήστης όπως επίσης και στον έλεγχο της μεταβλητής `strAction` (Εικόνα 6.5.3). Επίσης, πρέπει να σημειωθεί ότι η διεύθυνση του χρήστη έχει ξεχωριστό πίνακα μέσα στη βάση (πίνακας `Address`) για να μπορούμε να διαχειριζόμαστε καλύτερα τις διευθύνσεις.

```

79         else if (strAction.equals("SaveAddr")) {
80
81             try {
82
83                 CustomerMgmController cm = new CustomerMgmController();
84                 Customer customer;
85                 customer = cm.findCustomerById(Integer.parseInt(request
86                     .getParameter("id")));
87                 customer.getAddress().setAddressStreet(
88                     request.getParameter("addrstreet").toString());
89                 customer.getAddress().setAddressNumber(
90                     request.getParameter("addrnumber").toString());
91                 customer.getAddress().setAddressZipCode(
92                     request.getParameter("zipcode").toString());
93                 customer.getAddress().setCity(
94                     request.getParameter("city").toString());
95                 customer.getAddress().setCountry(
96                     request.getParameter("country").toString());
97
98                 customer.setLastName(request.getParameter("lstnamefield")
99                     .toString());
100
101                 cm.updateCustomer(customer);
102                 RequestDispatcher rd = null;
103                 rd = (request.getRequestDispatcher("/customerAddress.jsp"));
104                 session.setAttribute("user", customer.getFirstName());
105                 System.out.println(customer.getAddress().getAddressZipCode());
106                 session.setAttribute("addressStreet", customer.getAddress()
107                     .getAddressStreet());
108                 session.setAttribute("addressNumber", customer.getAddress()
109                     .getAddressNumber());
110                 session.setAttribute("addressZipCode", customer.getAddress()
111                     .getAddressZipCode());
112                 session.setAttribute("country", customer.getAddress()
113                     .getCountry());
114                 session.setAttribute("city", customer.getAddress().getCity());
115
116                 session.setAttribute("id", customer.getId());
117
118                 session.setAttribute("lastname", customer.getLastName());

```

Εικόνα 6.5.3: Servlet CustomerMgmCotroll – Επεξεργασία διευθύνσεων.

6.6 Καταχώριση και επεξεργασία άρθρων

Τη δυνατότητα για καταχώριση και επεξεργασία άρθρων έχει μόνο ο διαχειριστής της σελίδας. Αφού ο διαχειριστής κάνει είσοδο στην σελίδα με τον κωδικό του, θα εμφανιστεί η σελίδα allPost. Σε αυτή την σελίδα ο διαχειριστής βλέπει τα ήδη υπάρχοντα άρθρα τα οποία είναι αποθηκευμένα στη βάση και μπορεί να επιλέξει «Edit» ή «Delete».

Σε περίπτωση που επιλέξει “Edit” τότε μεταφέρεται στον ακόλουθο σύνδεσμο «.../NewMgmControl?action=edit&id=\${p.id}» όπου p.id είναι το id του άρθρου που υπάρχει στον πίνακα News μέσα στη βάση. Το Servlet που είναι υπεύθυνο γι’ αυτή τη διαδικασία είναι το NewMgmControl. Μέσα σε αυτό, λοιπόν, θα βρούμε τη συνάρτηση doGet στην οποία γίνεται έλεγχος για να διαχωριστούν οι λειτουργίες. Με αυτό τον τρόπο έχουμε τη μεταβλητή action που ελέγχει ποια λειτουργία επέλεξε ο χρήστης (Εικόνα 6.6.1).

```
HttpSession session=request.getSession();
String action=request.getParameter("action");
```

Εικόνα 6.6.1: Servlet NewMgmControl

Δημιουργούμε το αντικείμενο της κλάσης NewsMgmController και της κλάσης News για να χρησιμοποιήσουμε τις αντίστοιχες συναρτήσεις (Εικόνα 6.6.2).

```
NewsMgmController nm = new NewsMgmController();
News news=new News();
```

Εικόνα 6.6.2: Δημιουργία αντικειμένων.

Στη συνέχεια κάνουμε τον έλεγχο της επιλογής του χρήστη (Εικόνα 6.6.3).

```
else if (action.equals("edit")){
    Integer id=Integer.valueOf(request.getParameter("id"));
```

Εικόνα 6.6.3: Έλεγχος επιλογής χρήστη.

Για να μας εμφανίσει το άρθρο με το συγκεκριμένο id στο οποίο πάτησε ο χρήστης χρησιμοποιούμε τη συνάρτηση findNewsById(). Αυτή η συνάρτηση υπάρχει μέσα στην κλάση NewsMgmController. Τέλος, αναθέτει στο session τα πεδία του άρθρου από τη βάση με το συγκεκριμένο Id (Εικόνα 6.6.4). Οπότε έχουμε τη δυνατότητα να χρησιμοποιήσουμε τις πληροφορίες και σε άλλες σελίδες.

```
106
107     try {
108         news = nm.findNewsById(id);
109     } catch (NotLegalStateException e) {
110         // TODO Auto-generated catch block
111         e.printStackTrace();
112     }
113
114     /*news.setId(id);
115     news.setTitle(request.getParameter("title").toString());
116     news.setDescription(request.getParameter("description").toString());
117     news.setDetail(request.getParameter("detail").toString());
118     news.setCategory(request.getParameter("category").toString());
119     news.setImage(request.getParameter("image").toString());*/
120
121
122     //nm.updateNews(news);
123     RequestDispatcher rd = null;
124     rd = (request.getRequestDispatcher("/editPost.jsp"));
125     session.setAttribute("id", news.getId());
126     session.setAttribute("title", news.getTitle());
127     session.setAttribute("description", news.getDescription());
128     session.setAttribute("detail", news.getDetail());
129     session.setAttribute("category", news.getCategory());
130     session.setAttribute("image", news.getImage());
131
```

Εικόνα 6.6.4:Επιστροφή επιλεγμένου άρθρου.

Τέλος, όταν ο χρήστης έχει κάνει τις αλλαγές στο άρθρο, αναλαμβάνει πλέον η `doPost` που υπάρχει στο ίδιο Servlet (`NewMgmControl`). Χρησιμοποιώντας τη συνάρτηση `updateNews()` ενημερώνονται πλέον τα νέα στοιχεία στη βάση, σύμφωνα με αυτά που έβαλε ο χρήστης στα αντίστοιχα πεδία (Τίτλος, Περιγραφή, Άρθρο, Κατηγορία, Φωτογραφία). Αφού όλα έχουν γίνει με επιτυχία ο διαχειριστής ανακατευθύνεται στην σελίδα `allPost` (Εικόνα 6.6.5).

```

if (action.equals("edit")){

    try {
        NewsMgmController nc = new NewsMgmController();
        News news=new News();

        news = nc.findNewsById(Integer.parseInt(request.getParameter("id")));

        news.setDate(dateTemp);
        news.setTitle(request.getParameter("title").toString());
        news.setDescription(request.getParameter("description").toString());
        news.setDetail(request.getParameter("detail").toString());
        news.setCategory(request.getParameter("category").toString());
        news.setImage(request.getParameter("image").toString());

        nc.updateNews(news);

        session.setAttribute("id", news.getId());
        session.setAttribute("date", news.getDate());
        session.setAttribute("title", news.getTitle());
        session.setAttribute("description", news.getDescription());
        session.setAttribute("detail", news.getDetail());
        session.setAttribute("category", news.getCategory());
        session.setAttribute("image", news.getImage());

        RequestDispatcher rd = null;
        rd = (request.getRequestDispatcher("/allPost.jsp"));

        rd.forward(request, response);
    }
    catch (NumberFormatException | NotLegalStateException e) {
        e.printStackTrace();
    }
}

```

Εικόνα 6.6.5: Συνάρτηση doPost.

Στην περίπτωση τώρα που ο χρήστης επιλέξει Delete μεταφέρεται στον ακόλουθο σύνδεσμο «[../NewMgmControl?action=delete&id=\\${p.id}](#)» όπου p.id είναι το id του άρθρου που υπάρχει στον πίνακα News μέσα στη βάση. Το Servlet που είναι υπεύθυνο γι' αυτή τη διαδικασία είναι το NewMgmControl. Μέσα σε αυτό, θα βρούμε στη συνάρτηση doGet τη διαδικασία που δημιουργεί ένα παράθυρο διαλόγου με δυο επιλογές (Ναι & Όχι), που ρωτάει τον χρήστη αν θέλει να διαγράψει το άρθρο. Σε περίπτωση που ο χρήστης επιλέξει την επιλογή «Όχι» τότε το παράθυρο θα εξαφανιστεί και καμία ενέργεια δεν θα γίνει. Αν επιλέξει «Ναι» τότε χρησιμοποιούμε το id του άρθρου και καλώντας τη συνάρτηση deleteNews() η οποία βρίσκεται μέσα στην κλάση NewsMgmController, γίνεται η διαγραφή του άρθρου από τη βάση (Εικόνα 6.6.6)

```

if (action.equals("delete")){

    int answer = JOptionPane.showConfirmDialog(null, "Το άρθρο θα διαγραφεί. Θέλετε να συνεχίσετε;", "Warning",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
    if (answer == JOptionPane.NO_OPTION) {

        JOptionPane.getRootFrame().dispose();
        /*JFrame parent = new JFrame();
        JOptionPane.showMessageDialog(parent, "Printing complete");*/

    } if (answer == JOptionPane.YES_OPTION) {
        System.out.println("Yes button clicked");

        Integer id=Integer.valueOf(request.getParameter("id"));
        nm.deleteNews(id);

        RequestDispatcher rd = null;
        rd = (request.getRequestDispatcher("/allPost.jsp"));
        rd.forward(request, response);

    }else if (answer == JOptionPane.CLOSED_OPTION) {
        System.out.println("JOptionPane closed");
        //System.exit(0);
        JOptionPane.getRootFrame().dispose();
    }
}

```

Εικόνα 6.6.6: Διαγραφή επιλεγμένου άρθρου.

Στην περίπτωση που ο χρήστης επιλέξει να κάνει προσθήκη νέου άρθρου τότε θα μεταφερθεί στη σελίδα addNew. Το Servlet το οποίο είναι υπεύθυνο για τη δημιουργία ενός νέου άρθρου είναι το CreateArticleControl. Σε αυτό θα βρούμε τη συνάρτηση doPost μέσα στην οποία δημιουργείται ένα αντικείμενο της κλάσης News και δημιουργείται το άρθρο σύμφωνα με τα στοιχεία που έχει συμπληρώσει ο χρήστης στον Editor (Τίτλος, Περιγραφή, Άρθρο, Κατηγορία, Εικόνα.). Στη συνέχεια δημιουργείται το αντικείμενο της κλάσης NewsMgmController για να μπορέσουμε να χρησιμοποιήσουμε τη συνάρτηση insertNews() η οποία είναι υπεύθυνη για την εισαγωγή του νέου άρθρου στη βάση δεδομένων. Τέλος, αφού όλα έχουν γίνει με επιτυχία, ο χρήστης θα επιστρέψει στη σελίδα allPost (Εικόνα 6.6.7)

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    try {
        News news = new News(request.getParameter("title"), new Date(), request.getParameter("description"),
            request.getParameter("detail"), request.getParameter("category"), request.getParameter("image"));
        NewsMgmController mn = new NewsMgmController();
        mn.insertNews(news);

        response.sendRedirect("http://localhost:8080/SokratisMalamas/faces/allPost.jsp");
        JPAUtil.getSessionFactory().getCurrentSession().close();
    } catch (NotLegalStateException e) {
        e.printStackTrace();
    }
}

```

Εικόνα 6.6.7:Εισαγωγή νέου άρθρου.

6.7 Καταχώριση και επεξεργασία εγγραφών μενού

Τη δυνατότητα για την καταχώριση και την επεξεργασία στοιχείων στο μενού πλοήγησης έχει μόνο ο διαχειριστής της σελίδας. Αφού ο διαχειριστής κάνει είσοδο στη σελίδα με τον κωδικό του, από το μενού του διαχειριστή έχει τις εξής επιλογές:

1. Προβολή όλων
2. Προσθήκη Μενού.

Στην περίπτωση που επιλέξει την «Προβολή όλων» τότε θα μεταφερθεί στη σελίδα allMenu. Σε αυτή την σελίδα θα εμφανιστούν όλες οι εγγραφές του μενού. Υπάρχουν οι επιλογές της επεξεργασίας του μενού και της διαγραφής του. Όταν ο διαχειριστής επιλέξει την επεξεργασία μιας εγγραφής θα μεταφερθεί στη σελίδα «../MenuMgmControl?action=editMenu&id=\${p.id}» όπου p.id είναι το id της εγγραφής του μενού, στον πίνακα menu μέσα στη βάση δεδομένων. Το Servlet που είναι υπεύθυνο γι' αυτή τη διαδικασία είναι το MenuMgmControl. Μέσα στο Servlet θα βρούμε τη συνάρτηση doGet στην οποία γίνεται έλεγχος για να διαχωριστούν οι λειτουργίες που θα επιλέξει ο χρήστης. Έτσι δημιουργήσαμε την μεταβλητή action που ελέγχει ποια λειτουργία επέλεξε ο χρήστης (Εικόνα 6.7.1).

```

HttpSession session=request.getSession();
String action=request.getParameter("action");

```

Εικόνα 6.7.1:Servlet MenuMgmControl

Δημιουργούμε το αντικείμενο της κλάσης MenuMgmController και της κλάσης Menu για να χρησιμοποιήσουμε τις αντίστοιχες συναρτήσεις (Εικόνα 6.7.2).


```
MenuMgmController mc = new MenuMgmController();
Menu menu=new Menu();
```

Εικόνα 6.7.2: Δημιουργία αντικειμένων MenuMgmController & Menu.

Στη συνέχεια κάνουμε τον έλεγχο της επιλογής του χρήστη. (Εικόνα 6.7.3)

```
else if (action.equals("editMenu")){
    Integer id=Integer.valueOf(request.getParameter("id"));
```

Εικόνα 6.7.3: Έλεγχος επιλογής χρήστη.

Στη συνέχεια για να μας εμφανίσει την εγγραφή του μενού με το συγκεκριμένο id στο οποίο πάτησε ο χρήστης χρησιμοποιούμε τη συνάρτηση findMenuById(). Αυτή η συνάρτηση υπάρχει μέσα στην κλάση MenuMgmController. Τέλος, αναθέτει στο session τα πεδία του μενού από τη βάση με το συγκεκριμένο Id (Εικόνα 6.7.4). Οπότε έχουμε τη δυνατότητα να χρησιμοποιήσουμε τις πληροφορίες και σε άλλες σελίδες.

```
else if (action.equals("editMenu")){
    Integer id=Integer.valueOf(request.getParameter("id"));

    try {
        menu = mc.findMenuById(id);
    } catch (NotLegalStateException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    RequestDispatcher rd = null;
    rd = (request.getRequestDispatcher("/editMenu.jsp"));
    session.setAttribute("id", menu.getId());
    session.setAttribute("name", menu.getName());
    session.setAttribute("parentId", menu.getParentId());
    session.setAttribute("url", menu.getUrl());
    session.setAttribute("userrights", menu.getUserrights());

    rd.forward(request, response);
```

Εικόνα 6.7.4: Έλεγχος επιλογής χρήστη & ανάθεση τιμών.

Τέλος, όταν ο χρήστης έχει κάνει τις αλλαγές που επιθυμεί στο μενού, αναλαμβάνει πλέον η doPost που υπάρχει στο ίδιο Servlet (MenuMgmControl). Χρησιμοποιώντας τη συνάρτηση

updateMenu() ενημερώνονται πλέον τα νέα στοιχεία στη βάση δεδομένων, σύμφωνα με αυτά που έβαλε ο χρήστης στα αντίστοιχα πεδία. Αφού όλα έχουν γίνει με επιτυχία ο διαχειριστής ανακατευθύνεται στη σελίδα /allMenu (Εικόνα 6.7.5).

```

if (action.equals("updateMenu")){

    try {
        MenuMgmController mm = new MenuMgmController();
        Menu menu=new Menu();

        menu = mm.findMenuById(Integer.parseInt(request.getParameter("id")));

        menu.setName(request.getParameter("name").toString());
        menu.setParentId(request.getParameter("parentId")=="" ? 0 : Integer.parseInt(request.getParameter("parentId")));
        menu.setUrl(request.getParameter("url").toString());
        menu.setUserrights(request.getParameter("userrights").toString());

        //Integer.parseInt(request.getParameter("parentId"))
        mm.updateMenu(menu);

        session.setAttribute("id", menu.getId());
        session.setAttribute("name", menu.getName());
        session.setAttribute("parentId", menu.getParentId());
        session.setAttribute("url", menu.getUrl());
        session.setAttribute("userrights", menu.getUserrights());

        RequestDispatcher rd = null;
        rd = (request.getRequestDispatcher("/allMenu.jsp"));

        rd.forward(request, response);
    }
}

```

Εικόνα 6.7.5: Συνάρτηση doPost

Στην περίπτωση που ο χρήστης επιλέξει Delete μεταφέρεται στον ακόλουθο σύνδεσμο «.../MenuMgmControl?action=deleteMenu&id=\${p.id}» όπου p.id είναι το id της εγγραφής του μενού που υπάρχει στον πίνακα «Menu» μέσα στη βάση δεδομένων. Το Servlet που είναι υπεύθυνο για τη διαδικασία της διαγραφής είναι το MenuMgmControl. Μέσα σε αυτό λοιπόν θα βρούμε στη συνάρτηση doGet, τη διαδικασία που δημιουργεί ένα παράθυρο διαλόγου με δυο επιλογές (Ναι & Όχι) που ρωτάει τον χρήστη αν θέλει να διαγράψει την εγγραφή του μενού. Σε περίπτωση που ο χρήστης επιλέξει την επιλογή «Όχι» τότε το παράθυρο θα εξαφανιστεί και καμία ενέργεια δεν θα γίνει. Αν επιλέξει «Ναι» τότε χρησιμοποιούμε το id του μενού και καλώντας τη συνάρτηση deleteMenu() η οποία βρίσκεται μέσα στην κλάση MenuMgmController, γίνεται η διαγραφή της εγγραφής του μενού από τη βάση δεδομένων.(Εικόνα 6.7.6)

```

if (action.equals("deleteMenu")){

    int answer = JOptionPane.showConfirmDialog(null, "Το μενού θα διαγραφεί. Θέλετε να συνεχίσετε;", "Warning",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
    if (answer == JOptionPane.NO_OPTION) {

        JOptionPane.getRootFrame().dispose();

    } if (answer == JOptionPane.YES_OPTION) {
        System.out.println("Yes button clicked");
        Integer id=Integer.valueOf(request.getParameter("id"));
        mc.deleteMenu(id);

        RequestDispatcher rd = null;
        rd = (request.getRequestDispatcher("/allMenu.jsp"));
        rd.forward(request, response);

    }else if (answer == JOptionPane.CLOSED_OPTION) {
        System.out.println("JOptionPane closed");
        //System.exit(0);
        JOptionPane.getRootFrame().dispose();
    }
}

```

Εικόνα 6.7.5: Διαγραφή εγγραφής μενού.

Στην περίπτωση τώρα που ο χρήστης επιλέξει να κάνει προσθήκη νέας εγγραφής μενού τότε θα μεταφερθεί στη σελίδα addMenu. Το Servlet το οποίο είναι υπεύθυνο για τη δημιουργία μίας νέας εγγραφής είναι το CreateMenuControl. Σε αυτό θα βρούμε τη συνάρτηση doPost μέσα στην οποία δημιουργείται ένα αντικείμενο της κλάσης Menu και δημιουργείται η εγγραφή του μενού σύμφωνα με όσα έχει συμπληρώσει ο χρήστης στα πεδία (Μενού, Περιγραφή, Κωδικός συσχέτισης, Url). Στη συνέχεια δημιουργείται το αντικείμενο της κλάσης MenuMgmController για να μπορέσουμε να χρησιμοποιήσουμε τη συνάρτηση insertMenu() η οποία είναι υπεύθυνη για την εισαγωγή της νέας εγγραφής μενού στη βάση δεδομένων. Τέλος, αφού όλα έχουν γίνει με επιτυχία, ο χρήστης θα επιστρέψει στη σελίδα allMenu (Εικόνα 6.7.6)

```
^/  
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    try {  
  
        Menu menu = new Menu(request.getParameter("name"),  
            request.getParameter("parentId")=="" ? 0 : Integer.parseInt(request.getParameter("parentId")),  
            request.getParameter("url"), request.getParameter("userrights"));  
        MenuMgmController mc = new MenuMgmController();  
        mc.insertMenu(menu);  
  
  
        response.sendRedirect("http://localhost:8080/SokratisMalamas/faces/allMenu.jsp");  
        JPAUtil.getSessionFactory().getCurrentSession().close();  
    } catch (NotLegalStateException e) {  
        e.printStackTrace();  
    }  
}
```

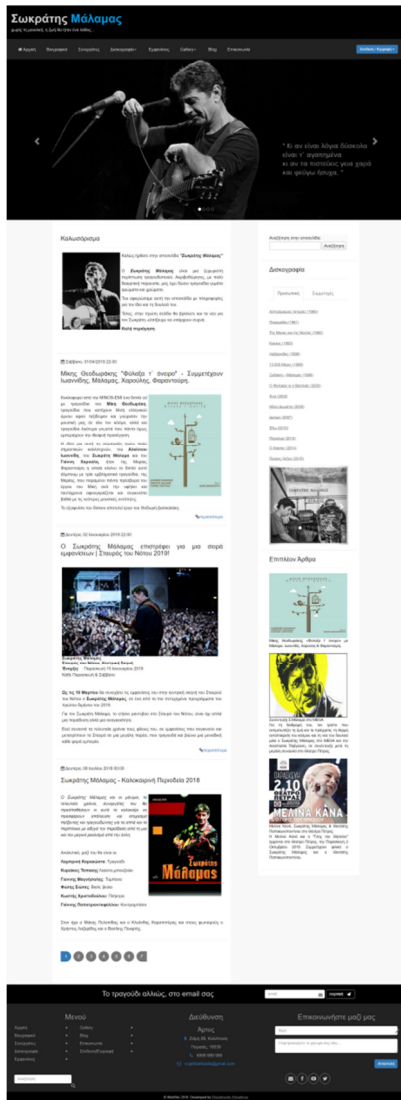
Εικόνα 6.7.6: Προσθήκη νέας εγγραφής μενού.

7. Παρουσίαση ιστότοπου

7.1 Περιγραφή

Η δυναμική ιστοσελίδα «Σωκράτης Μάλαμας» έχει ένα σύντομο μενού με τις βασικές πληροφορίες για τον καλλιτέχνη και τους συνεργάτες του. Το μενού παράγεται δυναμικά, προκύπτει δηλαδή από αποθηκευμένα δεδομένα στη βάση δεδομένων.

Στην αρχική σελίδα επιλέχθηκε ένα slider με 4 φωτογραφίες του μουσικού από συναυλίες που έχει πραγματοποιήσει (Εικόνα 7.1.1). Επίσης, στην αρχική σελίδα του ιστότοπου ο επισκέπτης μπορεί να διαβάσει μία λίστα από άρθρα για τη δισκογραφία, τις εκδηλώσεις, τους διαγωνισμούς κ.α που αφορούν το ίδιο τον μουσικό και τους συνεργάτες του. Τα άρθρα περιλαμβάνουν ημερομηνία δημοσίευσης του άρθρου, τίτλο, περιγραφή και έναν σύνδεσμο που οδηγεί στη σελίδα με όλη την περιγραφή του άρθρου.



Εικόνα 7.1.1: Αρχική σελίδα ιστότοπου

Με το μενού «Βιογραφικό» ο ιστότοπος «Σωκράτης Μάλαμας» έχει ως στόχο να ενημερώσει τους επισκέπτες για το βιογραφικό, τη δισκογραφία, τις συνεργασίες και τις εμφανίσεις του μουσικού (Εικόνα 7.1.2).

Σωκράτης Μάλαμας
 χωρίς τη μουσική, η ζωή θα ήταν ένα λάθος...

Αρχική Βιογραφικό Συνεργάτες Δισκογραφία Εμφανίσεις Gallery Blog Επικοινωνία [Συνθήκη Έγγραφης](#)

Βιογραφικό

Ο **Σωκράτης Μάλαμας** είναι τραγουδιστής, τραγουδιστής και κηρύττης. Είναι ένας από τους πιο γνωστούς Έλληνες συνθέτες και ερμηνευτές και ένα από τα πιο χαρακτηριστικά παραδείγματα του λεγόμενου έντεχνου τραγουδιού.

Γεννήθηκε στις 29 Σεπτεμβρίου 1957 στη Σικά της Χαλκιδικής. Λόγω εργασίας των γονιών του τον έβρισαν να βρεθεί στη Στουβρόλη της Γερμανίας μέχρι το εφτάβελτο του χρόνου. Στη συνέχεια επέστρεψε στη Θεσσαλονίκη το μουσικό που του έκανε δωρο ο πατέρας του αποτίκταιτο το εργατό του κέρμα για τη μουσική.

Στα δεκατέτα το αγόρασε την πρώτη του κιθάρα ως επιβράβευση για τις επιδόσεις του στα σχολικά μαθήματα και στα δεκατέτα του παρακολούθησε μαθήματα κηρύξης στο Μακεδονικό Οδείο Θεσσαλονίκης. Μετά την αποφοίτησή του από το Λύκειο, επέστρεψε και πάλι στη Γερμανία για να παρακολουθήσει μαθήματα μουσικής στο Οδείο της Στουβρόλης. Εξ μιας δύσκολης περιόδου της ζωής του αποφασίζει να μεταπηδήσει να εκπαιδευτεί και να πιάσει να δουλέψει στο Εθνικό Οδείο Αθηνών, μαθηρίζοντας όλη στην Βασιλική Αρματούρα και στον Νάση Μανουρά. Από τα 23 του αρχίζει να επιδεικνύει τον συνθετικό του ενδιαφέροντα, τραγουδώντας ως δίσκος κηρύξης, καθώς και ως τραγουδιστής σε λαϊκά μαγαζιά. Πριν ξεκινήσει την προσωπική του πορεία, υπηρέτησε κηρύττης στην αμφοτέρω του Νίκου Παπαδόπουλου.

Η πρώτη απόπειρα που κάνει στην ελληνική δισκογραφία δεν ατύχησε με επιτυχία καθώς η δισκογραφική εταιρεία Lyra αγόρασε το υλικό του με την απολογία ότι δεν ήταν έμπιστος. Συνέχισε να παίζει μουσική σε λαϊκά μαγαζιά, προσέχοντας και κάποια δικά του τραγούδια στο πρόγραμμα που παρουσίαζε, μέχρι να αναφέρει ότι ο ίδιος ήταν ο δημιουργός. Μέχρι τη στιγμή που ο Νίκος Παπαδόπουλος τον άκουσε και διέκρινε το ταλέντο του, όπως και τον έπεισε να μετα για ηχογράφηση στο στούντιο και να ετοιμάσει την πρώτη του δίσκο, με τίτλο «Αποφραγμένες Ιστορίες» το 1989. Σημανική η πορεία του καλλιτέχνη έγινε ιδιαίτερα γνωστή και φημισμένη από το κοινό.

Έχει συνεργαστεί στη διάρκεια της πορείας του με διάφορους καλλιτέχνες, όπως ο συνθέτης Νίκος Ξυδάκης, η τραγουδίστρια Μάρια Αλεξίου, ο τραγουδιστής και συνθέτης Αλέξανδρος Ιωνάκης, η τραγουδίστρια Μελίνα Κανά, οι τραγουδοποιοί Ορέφης Παράσης και Θανάσης Παπακωνσταντίνου. Με συνεργασίες όπως ο Γιάννης Τσατσόπουλος, ο Μανώλης Ρασούλης, ο Γιώργος Αθανασίου, ο Άλκης Ακτσός, η Φιλιττή Λαμπρινή ή Πηνελόπη Καρτερού, ο Θεόδωρος Γιάννης, ο Μιχάλης Γκενός, ο Οδυσσεύς Ιωάννης και ο Θανάσης Παπακωνσταντίνου. Επίσης, έχει αρκετές συμμετοχές σε άλλα έργα καλλιτεχνικά, όπως ο Λουδοβίκος των Αψβούργων, ο Διονύσιος Τσαρνιάς, ο Μάλαμας Παπαδόπουλος, η Μαρία Ουράνη, ο Β.Δ. Ρασούλης και η Θεοδώρα (των Active Member), η Ελευθερία Αρβανιτίκη, η Μαρία Ζουράρη, ο Νίκος Χαλκιόπουλος, και ο Χαλκίδης.

Δισκογραφία

Ο πρώτος προσωπικός δίσκος κυκλοφόρησε το 1989 και έχει τίτλο «**Αποφραγμένες Ιστορίες**». Πρόκειται για τραγούδια που έχουν γραφτεί τα προηγούμενα χρόνια σε μουσική και στίχους του ίδιου του Σωκράτη, ενώ την εισαγωγή και την πρόλογο ανέλαβε ο Νίκος Παπαδόπουλος. Ο δίσκος αυτός αποτελείται κυρίως από τραγούδια με φραγμένες ιστορίες.

Ο επόμενος δίσκος με τίτλο «**Τα Παραμύθια**» έρχεται το 1991, έχει το λαϊκό κηρύττη και την εισαγωγή αυτή τη φορά έχει αναλάβει ο Σωκράτης. Σ' αυτόν συμμετέχουν η Μελίνα Κανά, ο Ν. Παπαδόπουλος και η Ολγα Δερσιντή.

Το 1992 ο Σωκράτης Μάλαμας κατανοεί συνθέτουμε και μελοποιούμε στίχους του Γιάννη Τσατσόπουλου και του Μανώλη Ρασούλη για τον δίσκο «**Της μέρας και την νύχτας**» στον οποίο ερμηνεύει η Μελίνα Κανά. Μόσα απτάου το οδ έγραψε το «Να βάλω το μετάνιο» σε στίχους του Γ. Τσατσόπουλου το οποίο σημάδεψε το έντεχνο της δεκαετίας του '90 και έγινε ένα απ' τα πιο πολυαγαπημένα τραγούδια του κόσμου.

Έργα δημοφιλής ένα έντεχνο προσωπικό όφρα και με την ανδραπόψη και την "γάλα" που τον χαρακτηρίζει, ο Σωκράτης παρουσιάζει το 1993 τον δίσκο «**Αδελφάκια**» που αποτελεί σημαντικό σταθμό στην πορεία του Έλληνα δημοφιλούς και την ανάμεσα στα δυο, καταλαμβάνοντας ο ίδιος αυτή τη φορά την εμπέδωση της παραγωγής και της εισαγωγής καταρτίζοντας να αφήσει επίσης με τα τραγούδια "Το πάγω", "Ποιήματα Απλά", και "Φέλιξος χριστός". Τους στίχους υδία υπηρέτησε ο Σ. Μάλαμας, Ανθούλα Αθανασίου, Γ. Τσατσόπουλος, Γιώργος Αθανασίου και Θανάσης Παπακωνσταντίνου. Ο Σωκράτης έχει πια καθιερωθεί στην ελληνική μουσική σκηνή ως ένας από τους κορυφαίους στο είδος του. Το Φεβρουάριο του 1996 κυκλοφόρησε ο πέμπτος δίσκος με τίτλο «**Αδελφάκια**» αρχικά, που όμως για νεανούς Άλκης Ακτσός αφιέρωσε στο ερωτικό από το όνομα του καλλιτέχνη. Το κομμάτι "Το παιδί μες την πλατφό", "Το γράμμα", "Ο κήπος" και "Ένας σκαφός" αφιέρωσε από τον κόσμο και τραγουδώντας τους στίχους. Ο δίσκος παρέμεινε τραγουδώντας σε στίχους και μουσική του ίδιου ενώ δύο απτάου υπηρέτησε ο Γ. Αθανασίου και Οδυσσεύς Ιωάννης.

Δύο χρόνια αργότερα, τον Φεβρουάριο του 1998, κυκλοφόρησε ο νέος δίσκος και έχει τίτλο «**13000 μέρες**». Ο τίτλος είναι εμπνευσμένος από ένα τραγούδι που έγραψε ο Σωκράτης όταν ήταν 35 ετών έχοντας μετρήσει τα χρόνια της ζωής του έως τότε σε ημέρες. Μοναδική στιγμή στο δίσκο αποτελεί η μελοποίηση του ποίματος του Κ.Π. Καβάφη «**Αλεξάνδρος του 1903**».

Με έρχεται συνέχεια είναι στον επόμενο δίσκο το 1999, όπου ο Νίκος Ξυδάκης εμπέδωσε τον Σωκράτη Μάλαμας ως ερμηνευτή των τραγουδιών που συνέθεσε ο ίδιος σε στίχους των Μιχάλη Γκενός και Θεόδωρο Γιάννη. Πρόκειται για μια μοναδική μουσική συνάντηση που έγινε προς τα έξη δύο το μετάνιο του Έλληνα ερμηνευτή.

Το 2000 έρχεται μια νέα δουλειά με τίτλο «**Ο Φύλακας κι ο Βασίλειος**» στον οποίο οι αλλαγές είναι εμφανείς καθώς περιέχονται τόσο λαϊκά κομμάτια όσο και τραγούδια με αλκιστικό όφρα που θυμίζουν κάτι από ποπ. Θεωρούνται πάντα να είναι ουσιαστική επέλαση, ο Σωκράτης δίνει δύο τραγούδια που στην Μελίνα Κανά και άλλα δύο στον Μανώλη Αδάμη για να τα ερμηνεύσουν. Η "Πραγματοποίηση", το "Καμένο ρούχο", το "Τίποτα δε χάνεται" και πολλά ακόμα δημοφιλής γίνονται πρώτης επίσημης και το κοινό του μεγάλου τραγουδοποιοί θεωρούνται κατά ποπ.

Στα ήλια του 2002 ο Σωκράτης Μάλαμας τομά για μια ακόμη φορά, και θέλουμε να πραγματοποιήσει μια μεγάλη του επιβίβαση, ηχογραφεί δεκατέτα κομμάτια για το νέο του δίσκο «**Ένα**» χρησιμοποιώντας δύο κηρύττες και κατά περίπτωση ένα πιάνο. Τον Μάρτιο του 2004 συμμετείχε ερμηνεύοντας έξι τραγούδια στο άλμπουμ της Μαρίας Ουράνη «**Της νύχτας η μεσημέρι μαλλιά**» και το 2005 παρουσίασε το νέο του υλικό με τίτλο «**Αδελφάκια**».

Δύο χρόνια αργότερα έρχεται ο «**Αδελφάκια**» με δεκατέτα ολοκληρωμένα τραγούδια που μας ταξιδεύουν με έναν μοναδικό τρόπο, ενώ η Μάρια Αλεξίου αφιέρωσε τίτλους από αυτό. Ο τελευταίος δίσκος του ανεπανάλητου Σωκράτη Μάλαμας κυκλοφόρησε το 2012, έχει τίτλο «**ΕΣΤΩ**» και είναι το πρώτο live CD και DVD στην δισκογραφία του.

Συνεργάτες-Εμφανίσεις

Λαϊκός ή έντεχνος, στην διάρκεια της ποικίλης καριέρας του ο Σωκράτης Μάλαμας με το κλειστό πιάνο, το σπιθαρό προς το κοινό, την ειλικρίνεια, την μετριοφροσύνη και το ασπύβητο του χαρακτήρα που έχει καταφέρει να συνεργαστεί με ποιά μεγάλα ονόματα του ελληνικού ποπμουσικού όπως ο Θανάσης Παπακωνσταντίνου, ο Διονύσιος Τσαρνιάς, ο Λουδοβίκος των Αψβούργων, ο Μάλαμας Παπαδόπουλος, ο Άλκης Ακτσός, η Μάρια Αλεξίου, ο Ορέφης Παράσης, η Μελίνα Κανά αλλά και οι Active Member. Οι εμφανίσεις του δεν σταματούν ποτέ, το χειμώνα σε μικρούς σχετικά χώρους και εναλλακτικές μουσικές σκηνές, το καλοκαίρι σε αλχιστικές συναυλίες σε ολόκληρη την Ελλάδα παρέραι το κοινό σε ένα μουσικό ταξίδι προκαλώντας δυνατά συναίσθημα και αγάπες της φωνής όλων μέσα από τις μουσικές και τα τραγούδια του.

Το τραγούδι αλλιώς, στο email σας

Μενού: Αρχική, Βιογραφικό, Συνεργάτες, Δισκογραφία, Εμφανίσεις, Gallery, Blog, Επικοινωνία, Συνθήκη Έγγραφης

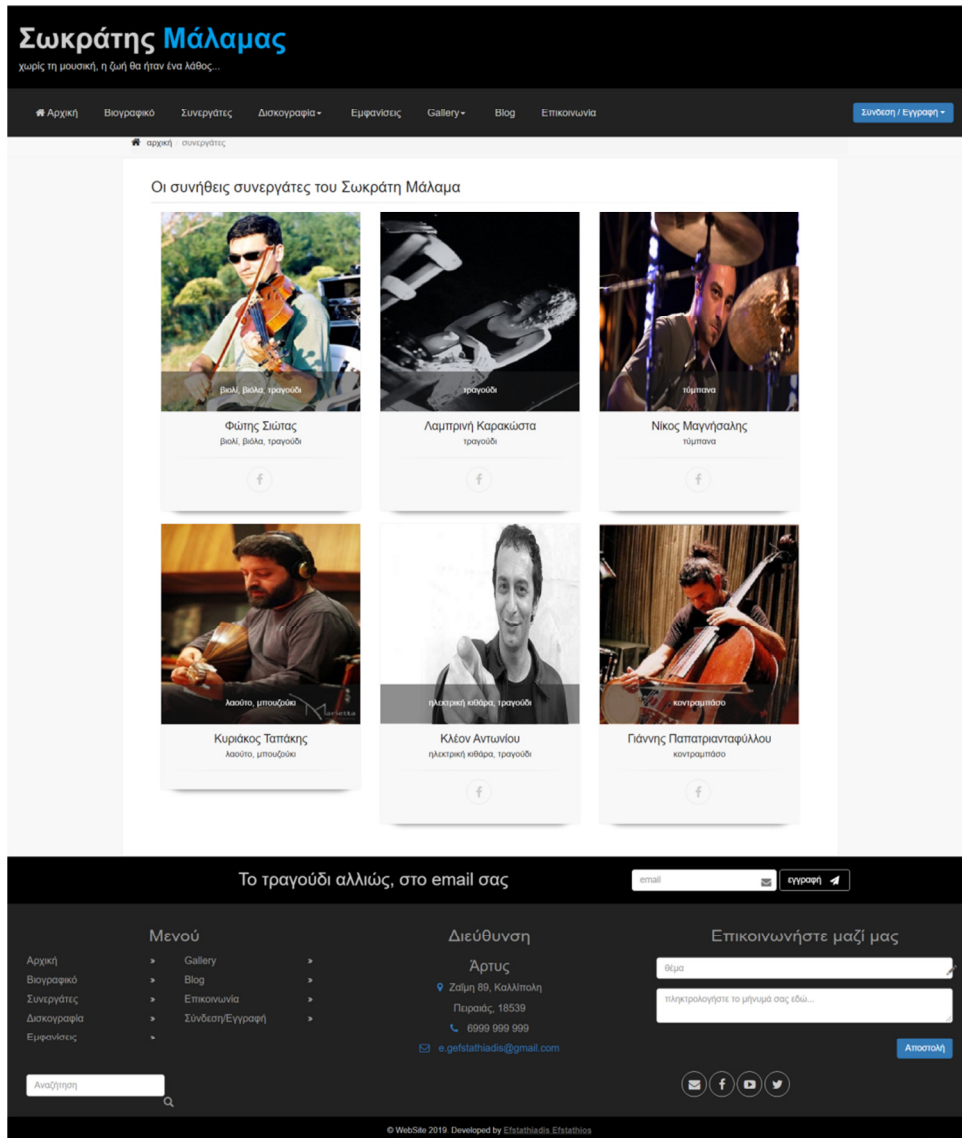
Διεύθυνση: Αρχική, Ζώμη 99, Χαλκιδική, Πάρους, 18019, 6909 999 999, info@melinakanal.gr

Επικοινωνήστε μαζί μας: Email, Πηγαίνετε στην ιστοσελίδα μου...

© Melinda 2019. Designed by Eusstathios Eustathios

Εικόνα 7.1.2: Σελίδα Βιογραφικό

Στο μενού «Συνεργάτες», ο επισκέπτης του ιστότοπου έχει τη δυνατότητα ενημερωθεί για τους σταθερούς συνεργάτες του καλλιτέχνη (Εικόνα 7.1.3).



Εικόνα 7.1.3: Σελίδα Συνεργάτες

Επίσης, οι επισκέπτες θα ενημερώνονται πιο αναλυτικά για τη δισκογραφία του καλλιτέχνη στο μενού «Δισκογραφία» (Εικόνα 7.1.4) και για τις συναυλίες του, στο μενού «Εμφανίσεις» (Εικόνα 7.1.5).


Σωκράτης Μάλαμας
χωρίς τη μουσική, η ζωή θα ήταν ένα λάθος...

- [Αρχική](#)
- [Βιογραφικό](#)
- [Συνεργάτες](#)
- [Δισκογραφία](#)
- [Εμφανίσεις](#)
- [Gallery](#)
- [Blog](#)
- [Επικοινωνία](#)


[Σύνθεση / Έγγραφο](#)

🏠 Αρχική [Δισκογραφία](#) [Προσωπική](#)


Προσωπική Δισκογραφία




Πρωίτες Λέξεις
2015 - Feelgood Records
[τραγούδια](#)




Ο Χάρτης
2014 - Minos-EMI
[τραγούδια](#)




Το Άδικο Δωμάτιο Αφήγηση II
2014 - Minos-EMI, Άρτις
[τραγούδια](#)




Πέρασμα
2010 - Lyra
[τραγούδια](#)




Έγω
2010 - Lyra
[τραγούδια](#)




Δρόμοι
2007 - Lyra
[τραγούδια](#)




Ζωντανή Ηχογράφηση Από Το Θέατρο Του Λαοφίλητου
2006 - Lyra
[τραγούδια](#)




Το Άδικο Δωμάτιο
2005 - Lyra
[τραγούδια](#)




Ένα
2002 - Lyra
[τραγούδια](#)




Ο Φύλακας και ο Βασιλιάς
2000 - Lyra
[τραγούδια](#)




Νίκος Σουλίκας Σωκράτης Μάλαμας
1999 - Lyra
[τραγούδια](#)




13.000 Μέρες
1998 - Lyra
[τραγούδια](#)




Λαβύρινθος
1996 - Lyra
[τραγούδια](#)




Κίχλος
1993 - Lyra
[τραγούδια](#)



Της Μίρας και της Νύχτας
1992 - Lyra
[τραγούδια](#)



Παραμύθια
1991 - Lyra
[τραγούδια](#)



Ασπράμαρες Ιστορίες
1989 - Lyra
[τραγούδια](#)

προηγούμενα < > επόμενα

Το τραγούδι αλλιώς, στο email σας [εγγραφή](#)

Μενού

- [Αρχική](#) >
- [Βιογραφικό](#) >
- [Συνεργάτες](#) >
- [Δισκογραφία](#) >
- [Εμφανίσεις](#) >

Διεύθυνση

Άρτις

Ζωή 89, Καλλιμολαίη
Πειραιάς, 18539
☎ 0999 999 999
✉ a.gelstathadis@gmail.com

Επικοινωνήστε μαζί μας

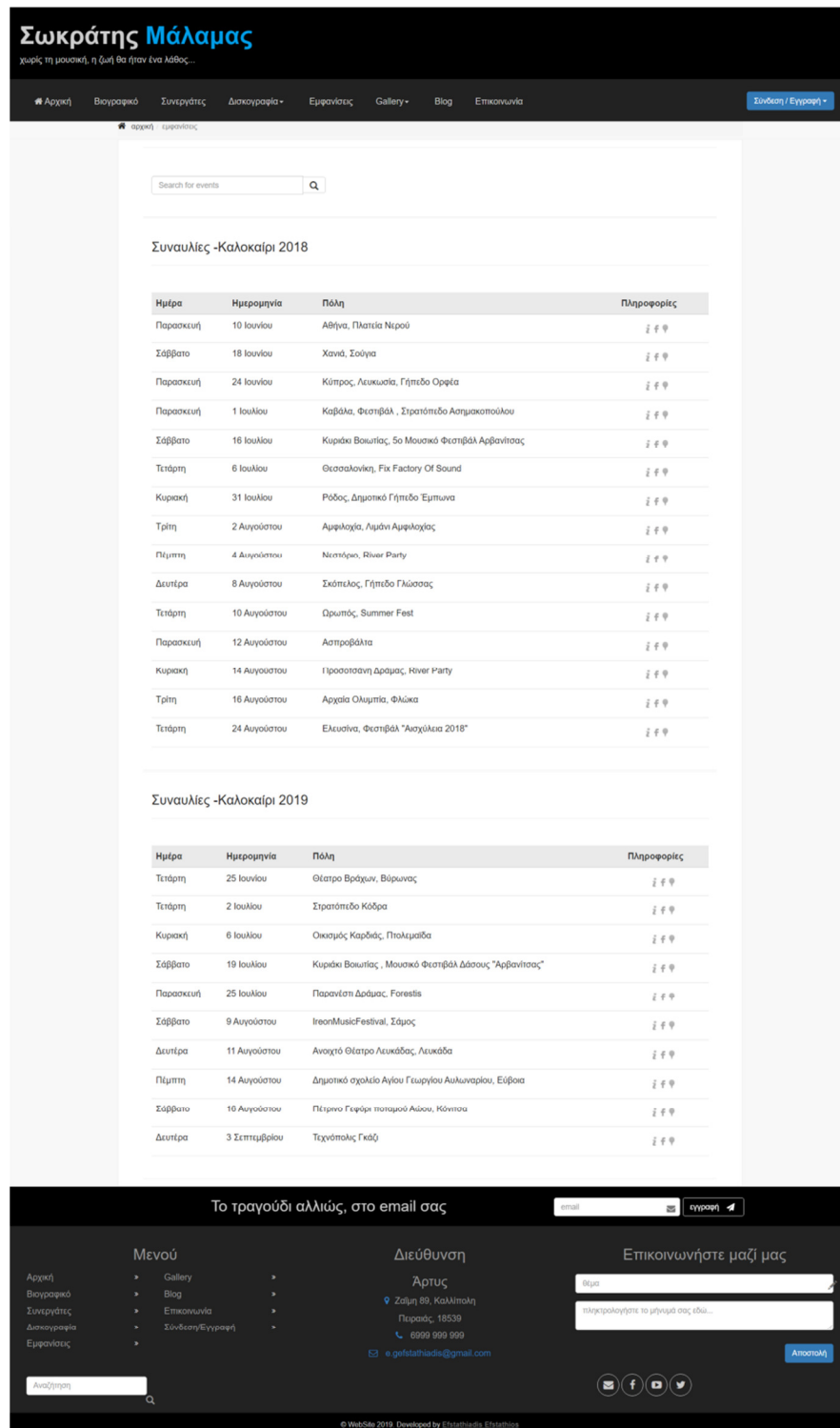
[Αποστολή](#)

Αναζήτηση

📧 📘 📺 🐦

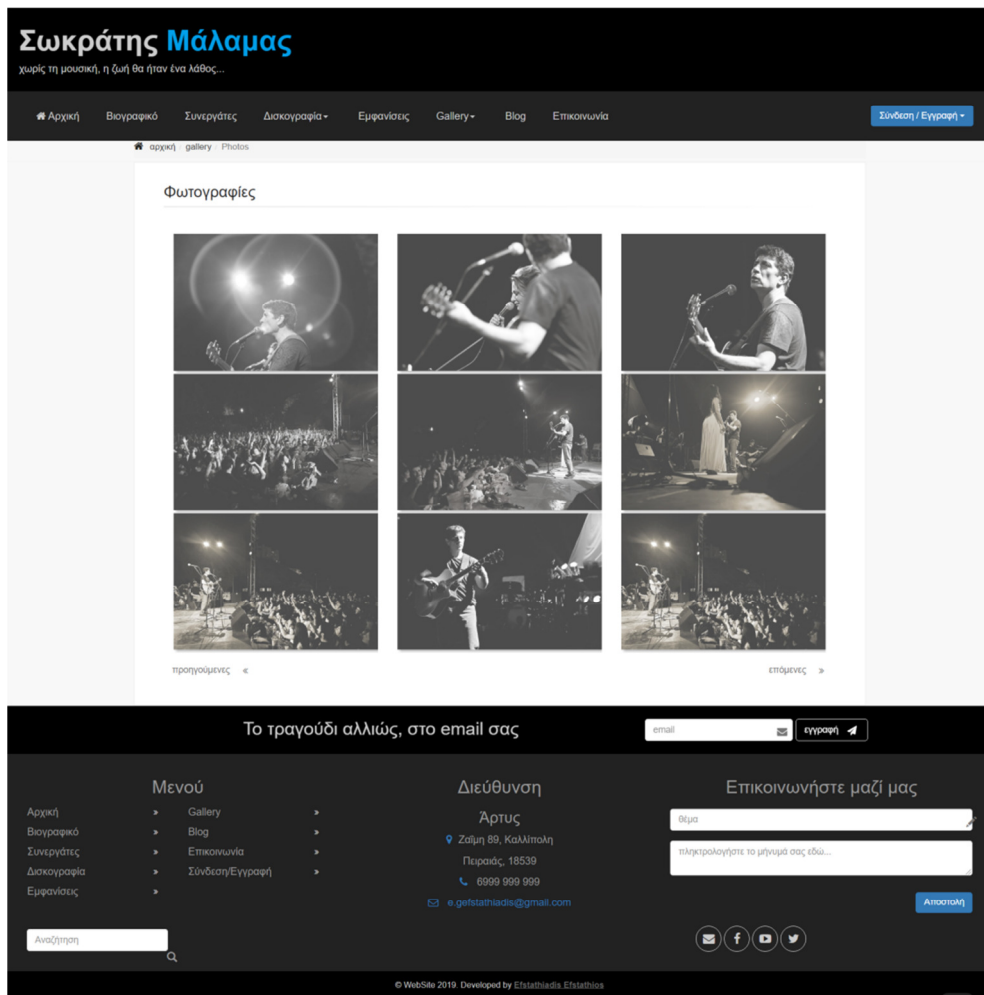
© WebSite 2019. Developed by Efstathidis Efstathios

Εικόνα 7.1.4: Σελίδα Δισκογραφία

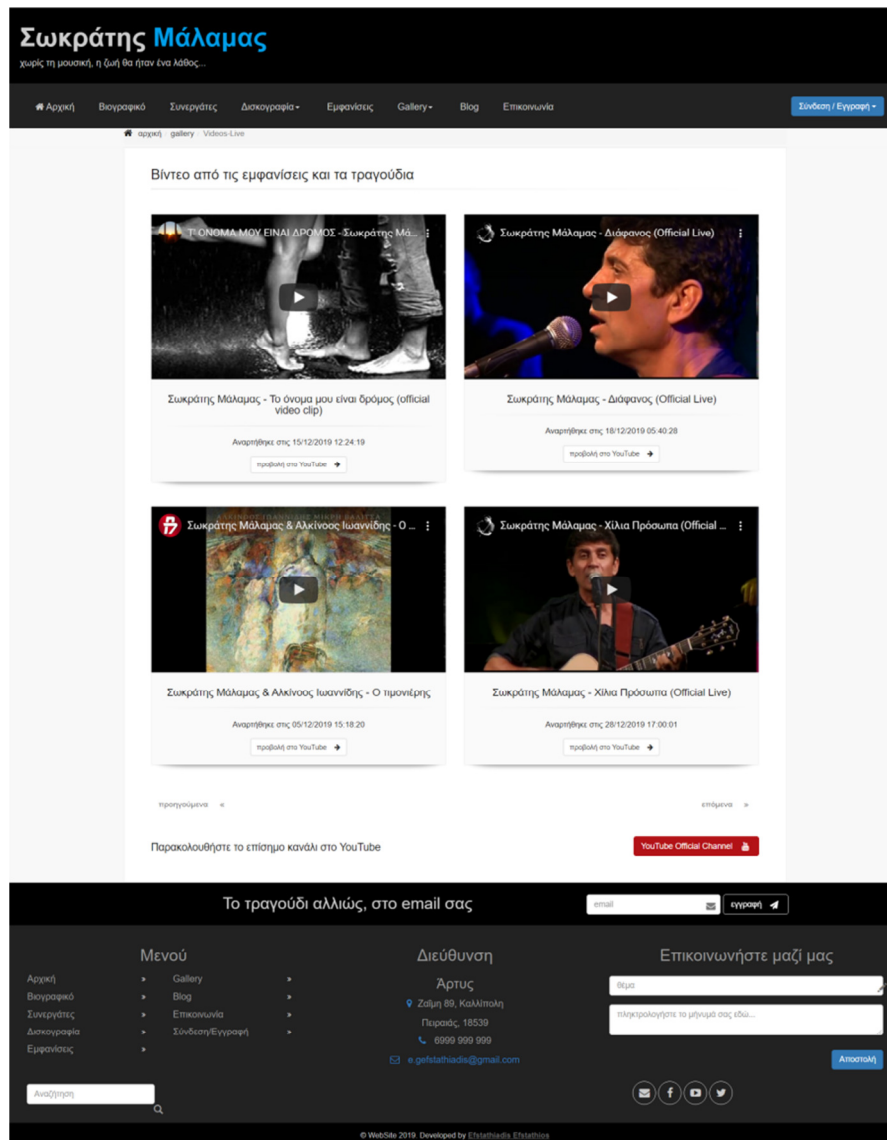


Εικόνα 7.1.5: Σελίδα Συναυλίες

Επίσης, στο μενού «Gallery» ο επισκέπτης μπορεί να δει βίντεο και φωτογραφίες του Σωκράτη Μάλαμα από συναυλίες που έχει πραγματοποιήσει στο παρελθόν (Εικόνα 7.1.6, Εικόνα 7.1.7).

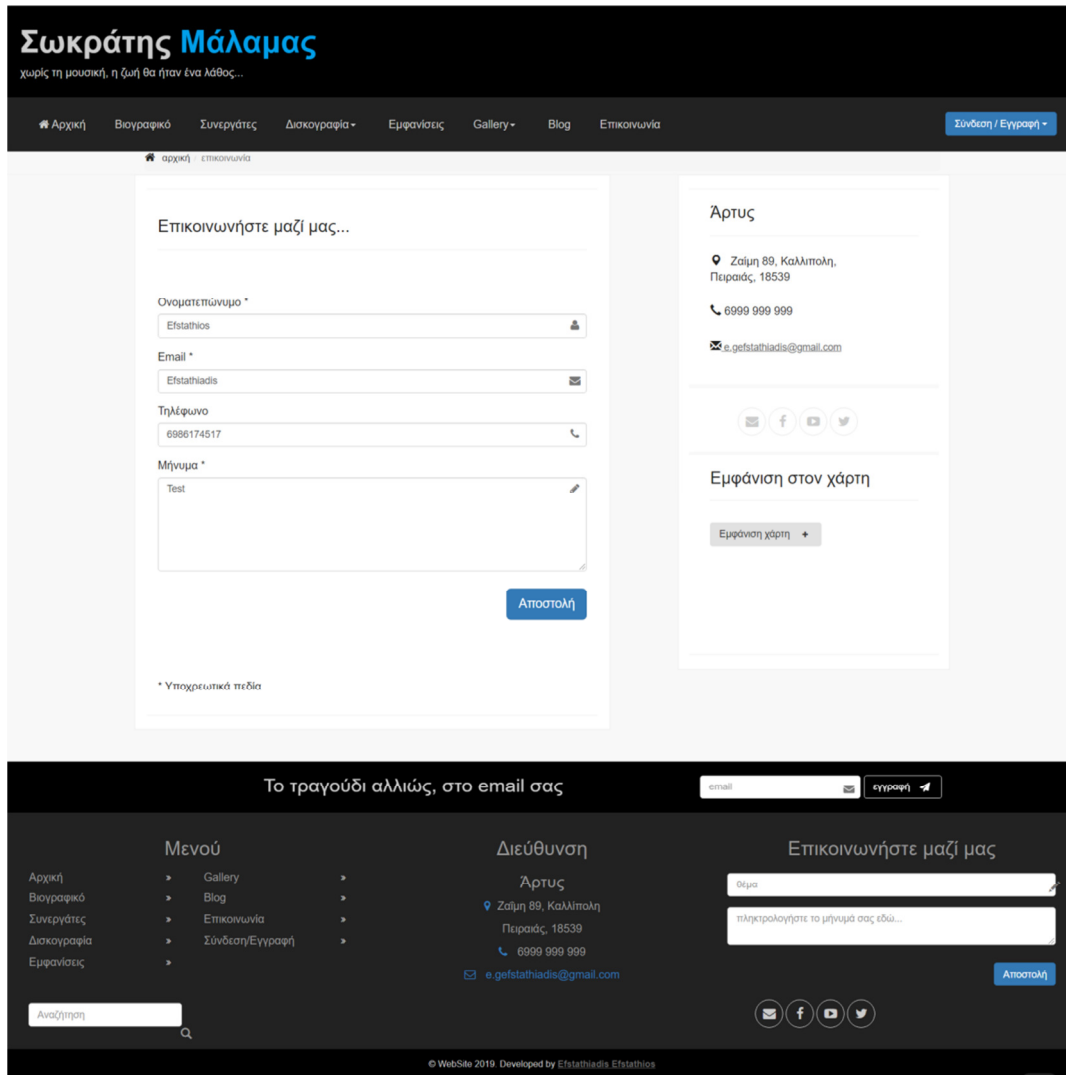


Εικόνα 7.1.6: Σελίδα Φωτογραφίες

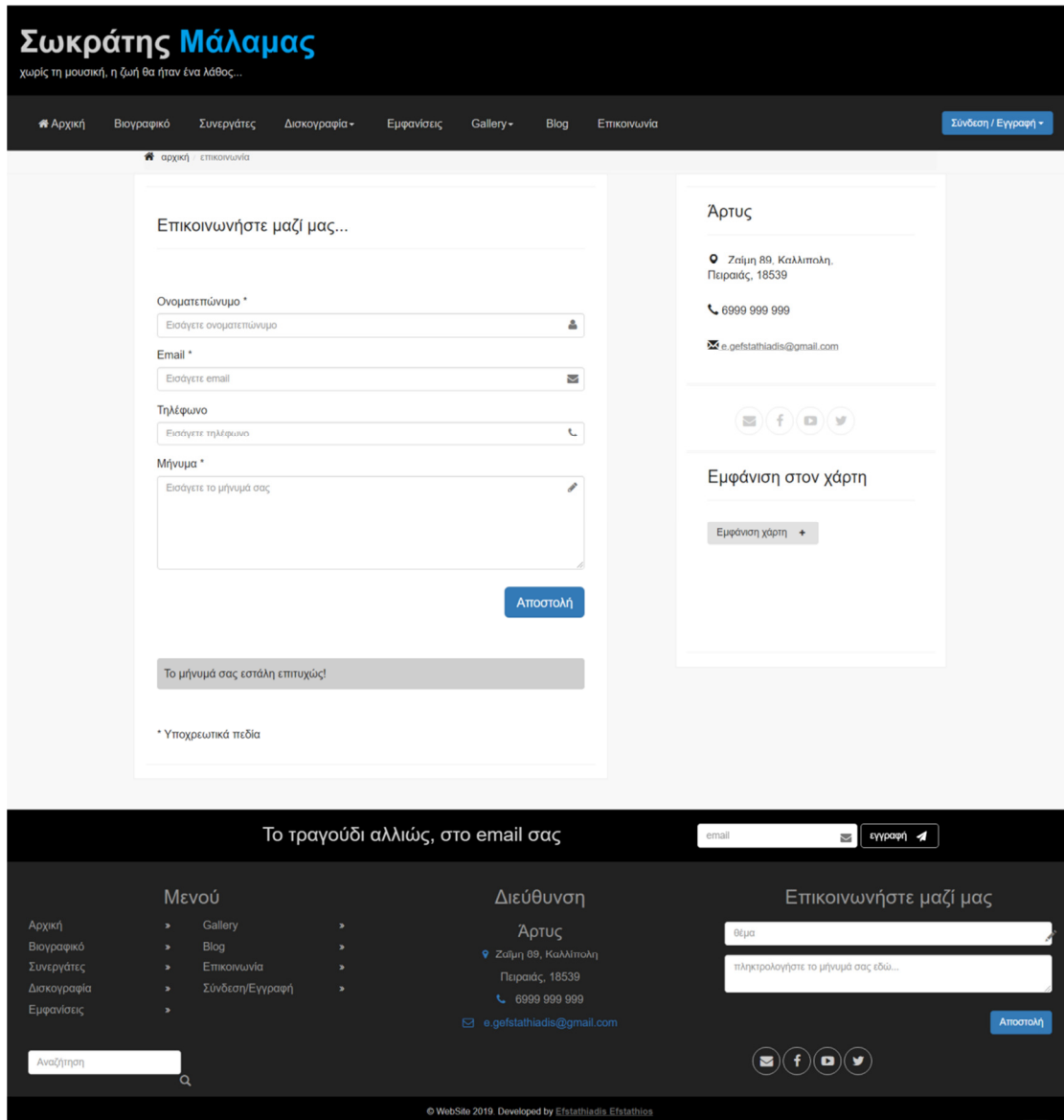


Εικόνα 7.1.7: Σελίδα Βίντεο

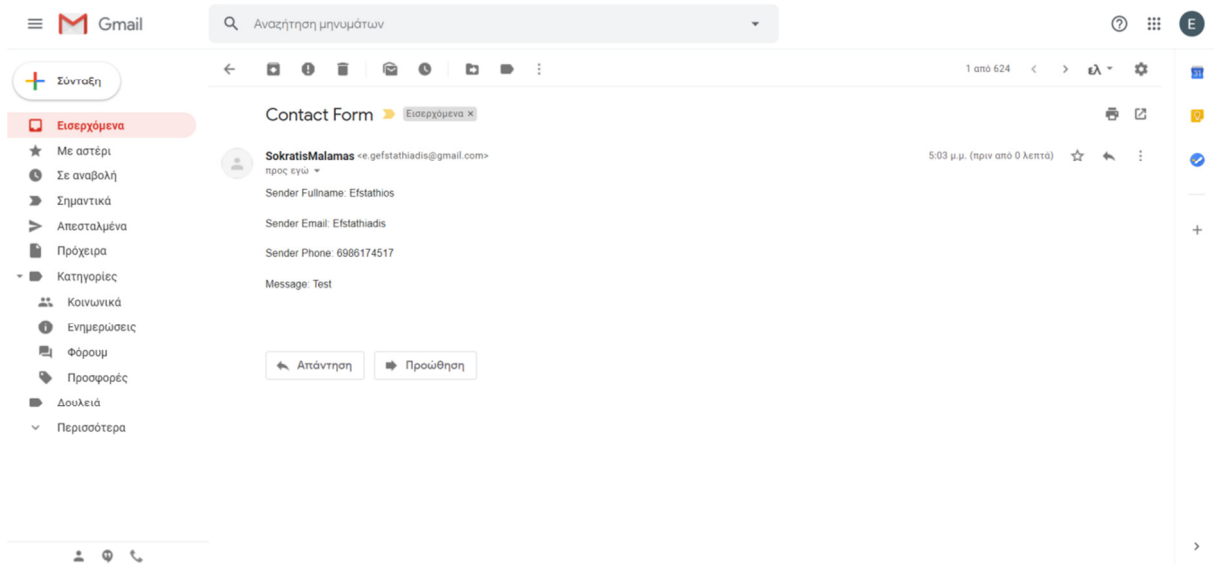
Ο επισκέπτης που επιθυμεί να επικοινωνήσει με τον μουσικό έχει τη δυνατότητα στο μενού «Επικοινωνία» να αποστείλει φόρμα επικοινωνίας. Η φόρμα επικοινωνίας έχει τα πεδία «Όνομα», «Επώνυμο», «Email», «Τηλέφωνο», «Μήνυμα» (Εικόνες 7.1.8α, 7.1.8β, 7.1.8γ).



Εικόνα 7.1.8α: Σελίδα Επικοινωνία

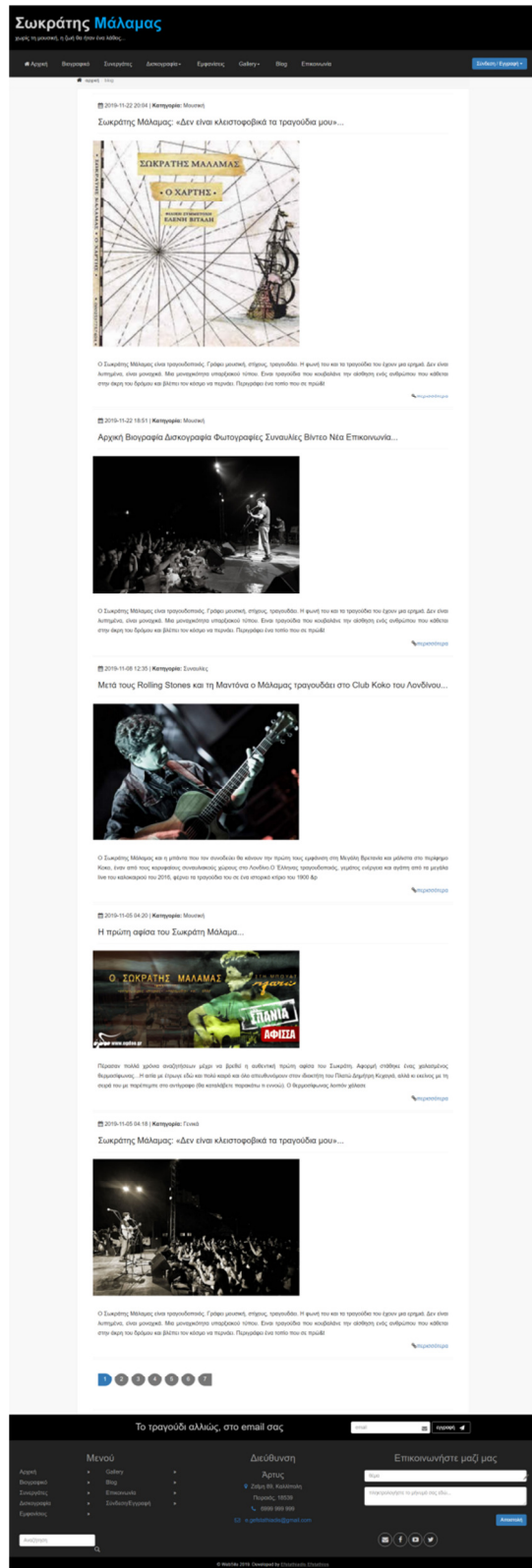


Εικόνα 7.1.8β: Σελίδα Επικοινωνία – Μήνυμα αποστολής

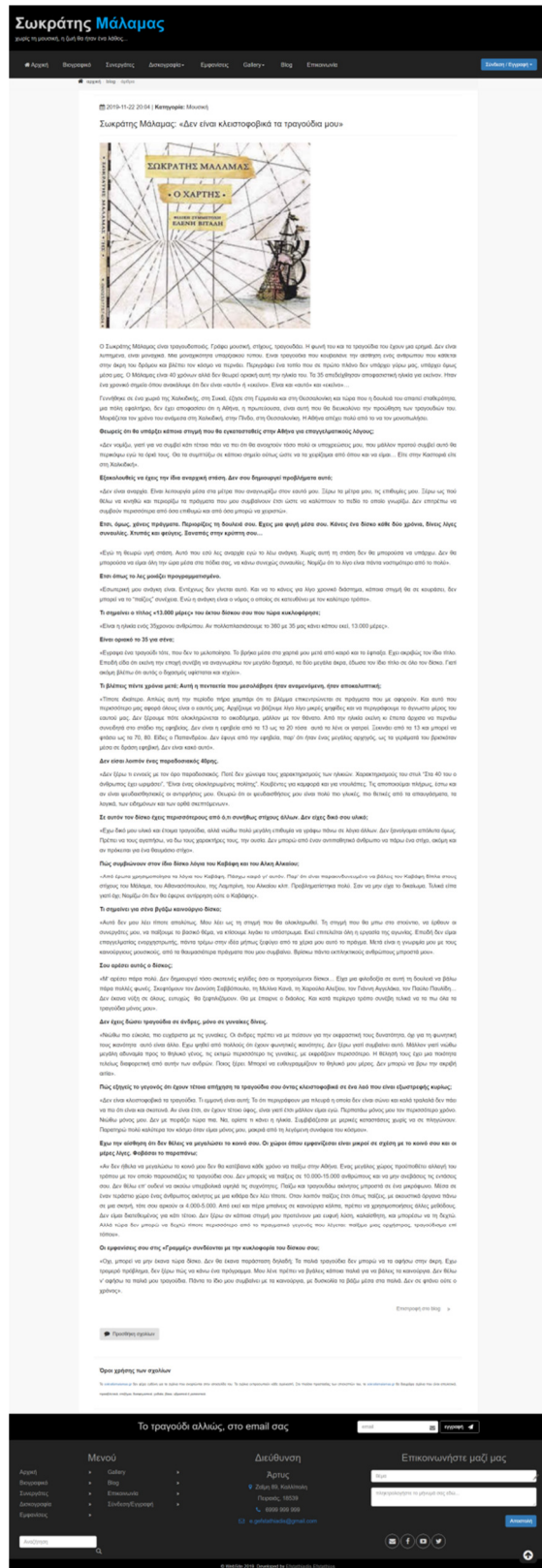


Εικόνα 7.1.8γ: Σελίδα Επικοινωνία - Μήνυμα παραλήπτη

Επίσης, στην ενότητα «Blog» ο επισκέπτης ενημερώνεται για όλα τα νέα του καλλιτέχνη (Εικόνες 7.1.9).

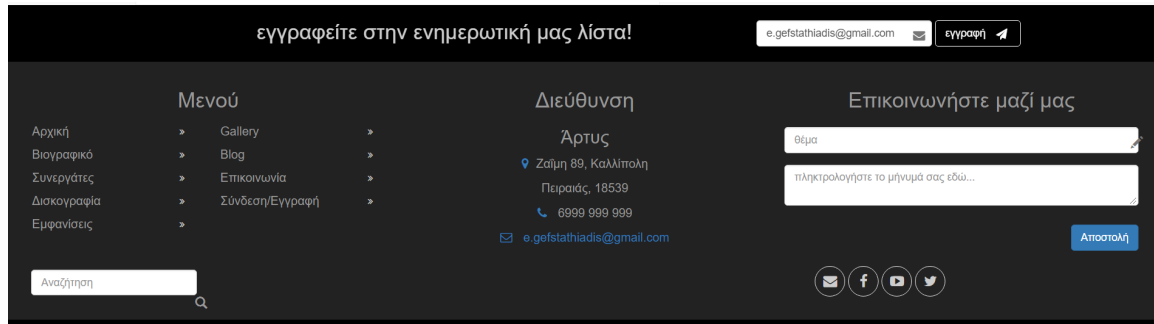


Εικόνα 7.1.9: Σελίδα Blog

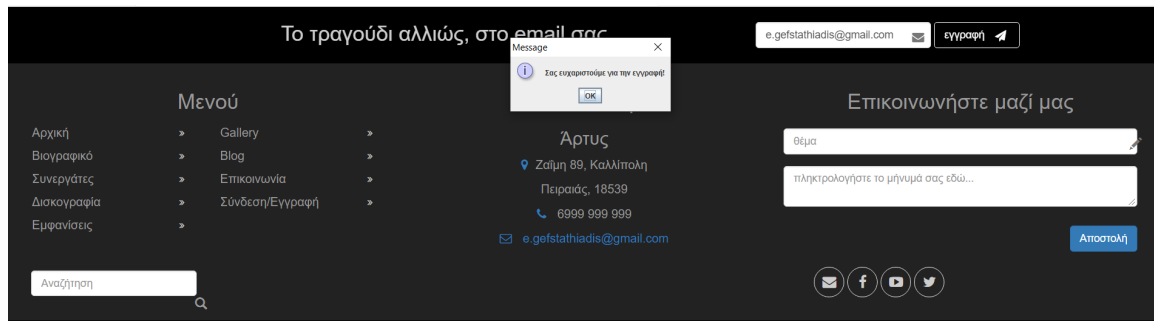


Εικόνα 7.1.9: Σελίδα Άρθρου

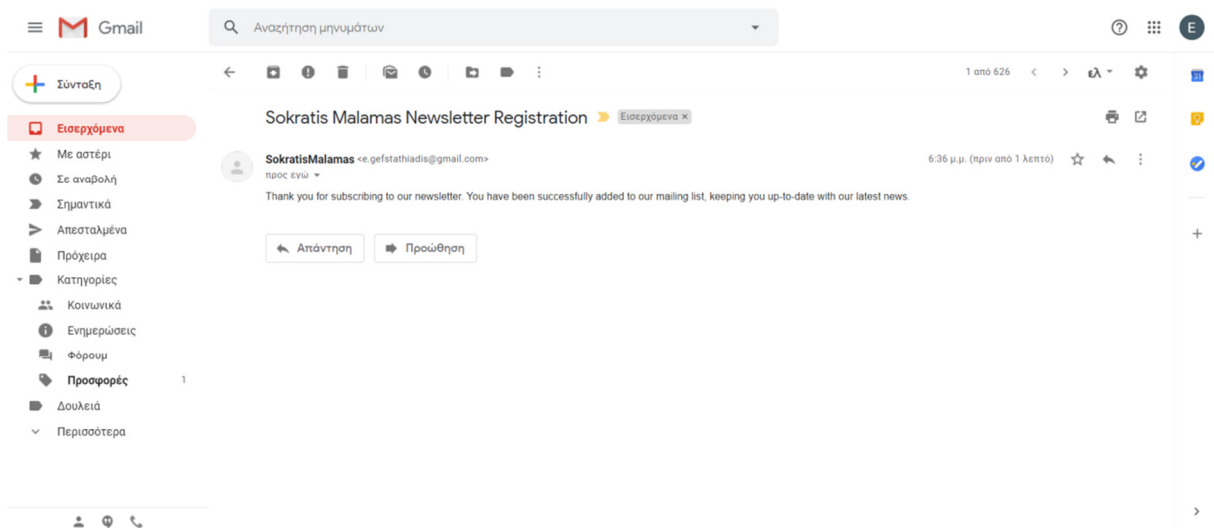
Τέλος, έχει τη δυνατότητα εγγραφής στο newsletter (Εικόνας 7.1.9α, Εικόνα 7.1.9β, Εικόνα 7.1.9γ) .



Εικόνα 7.1.9α: Εγγραφή Newsletter



Εικόνα 7.1.9β: Εγγραφή Newsletter



Εικόνα 7.1.9γ: Εγγραφή Newsletter

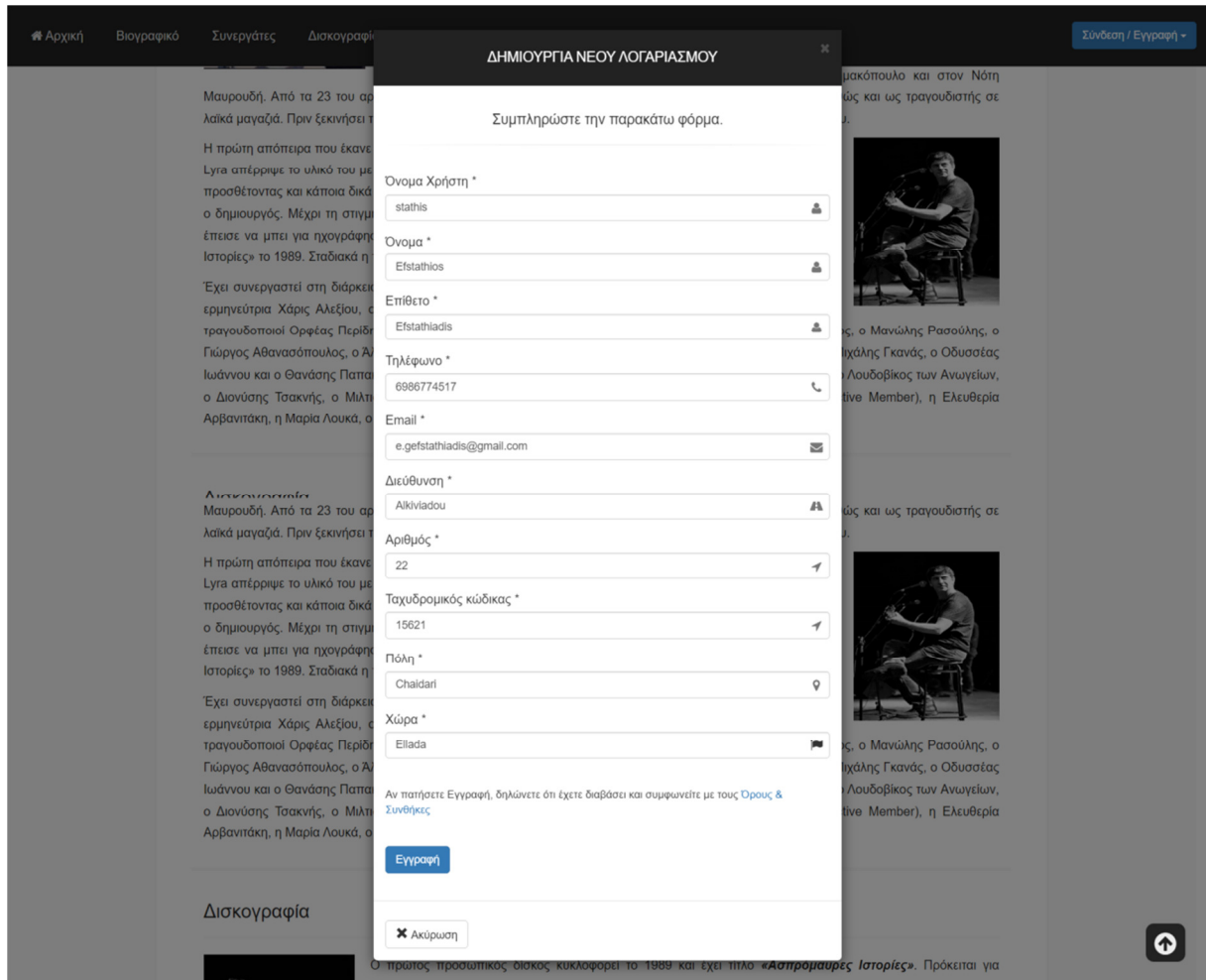
7.2 Νέος Χρήστης

Στην ιστοσελίδα «Σωκράτης Μάλαμας» υπάρχουν δύο ειδών χρήστες, ο διαχειριστής και οι απλοί χρήστες. Στην Εικόνα 7.2.1 παρουσιάζουμε τα στοιχεία που πρέπει να συμπληρώσει ένας νέος χρήστης.

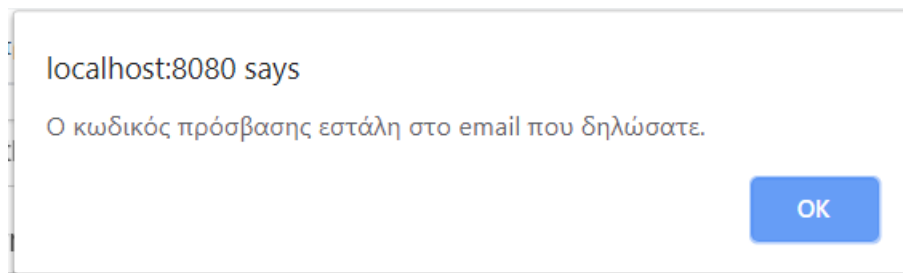
The image shows a web browser window with a modal form titled "ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΛΟΓΑΡΙΑΣΜΟΥ" (Create New Account). The form is overlaid on a background page that includes a navigation menu (Αρχική, Βιογραφικό, Συνεργάτες, Δισκογραφία) and a "Σύνδεση / Εγγραφή" button. The form itself contains the following fields and elements:

- Header: "Συμπληρώστε την παρακάτω φόρμα."
- Fields (all marked as required with an asterisk):
 - Όνομα Χρήστη * (Input: "Εισάγετε όνομα χρήστη")
 - Όνομα * (Input: "Εισάγετε όνομα", with a tooltip: "Please fill out this field.")
 - Επίθετο * (Input: "Εισάγετε επίθετο")
 - Τηλέφωνο * (Input: "Εισάγετε τηλέφωνο")
 - Email * (Input: "Εισάγετε e-mail")
 - Διεύθυνση * (Input: "Εισάγετε διεύθυνση")
 - Αριθμός * (Input: "Εισάγετε αριθμό")
 - Ταχυδρομικός κώδικας * (Input: "Εισάγετε ταχυδρομικό κώδικα")
 - Πόλη * (Input: "Εισάγετε πόλη")
 - Χώρα * (Input: "Εισάγετε χώρα")
- Terms and Conditions: "Αν πατήσετε Εγγραφή, δηλώνετε ότι έχετε διαβάσει και συμφωνείτε με τους Όρους & Συνθήκες"
- Buttons: "Εγγραφή" (blue) and "Ακύρωση" (grey)

Εικόνα 7.2.1: Δημιουργία νέου χρήστη- Έλεγχος



Εικόνα 7.2.2: Δημιουργία νέου χρήστη- Συμπλήρωση στοιχείων



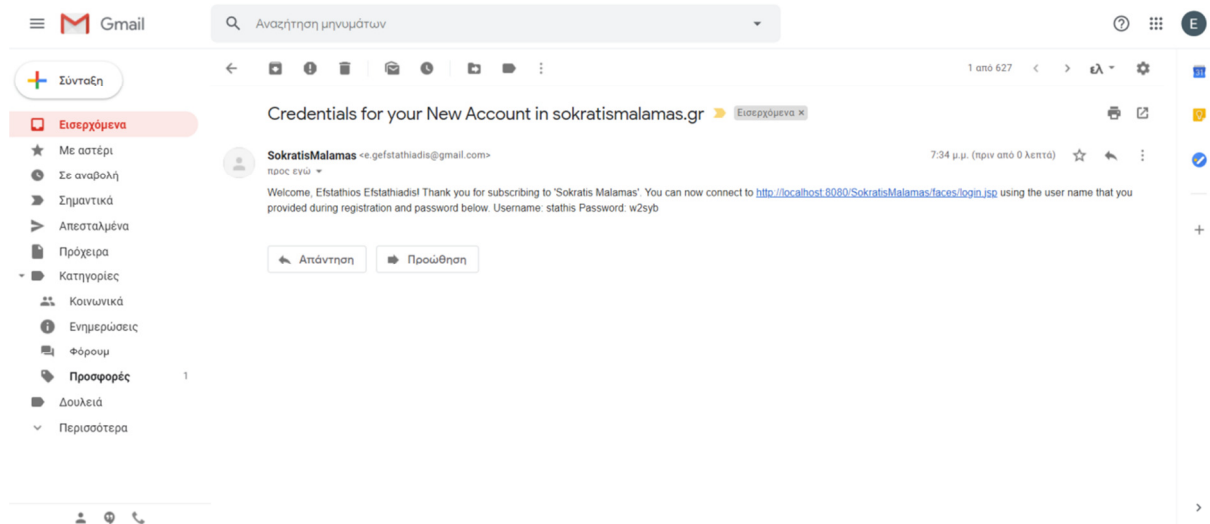
Εικόνα 7.2.3: Δημιουργία νέου χρήστη- Ενημερωτικό μήνυμα

Στην Εικόνα 7.2.1 βλέπουμε ότι υπάρχουν έλεγχοι και δεν επιτρέπεται η μη συμπλήρωση πεδίου. Όταν η διαδικασία της δημιουργίας νέου χρήστη ολοκληρωθεί, όπως φαίνεται και στην Εικόνα 7.2.4, δημιουργείται μία νέα εγγραφή στη βάση δεδομένων.

5	145	Test	12345	Test	Test	FovwJ	e.gefstathiadis@gmail.com	test	test	FovwJ	123456789	test12	user
6	22	Alkiviadou	15621	Chaidari	Ellada	w2syb	e.gefstathiadis@gmail.com	Efstathios	Efstathiadis	w2syb	6986774517	stathis	user

Εικόνα 7.2.4: Δημιουργία χρήστη στη βάση δεδομένων

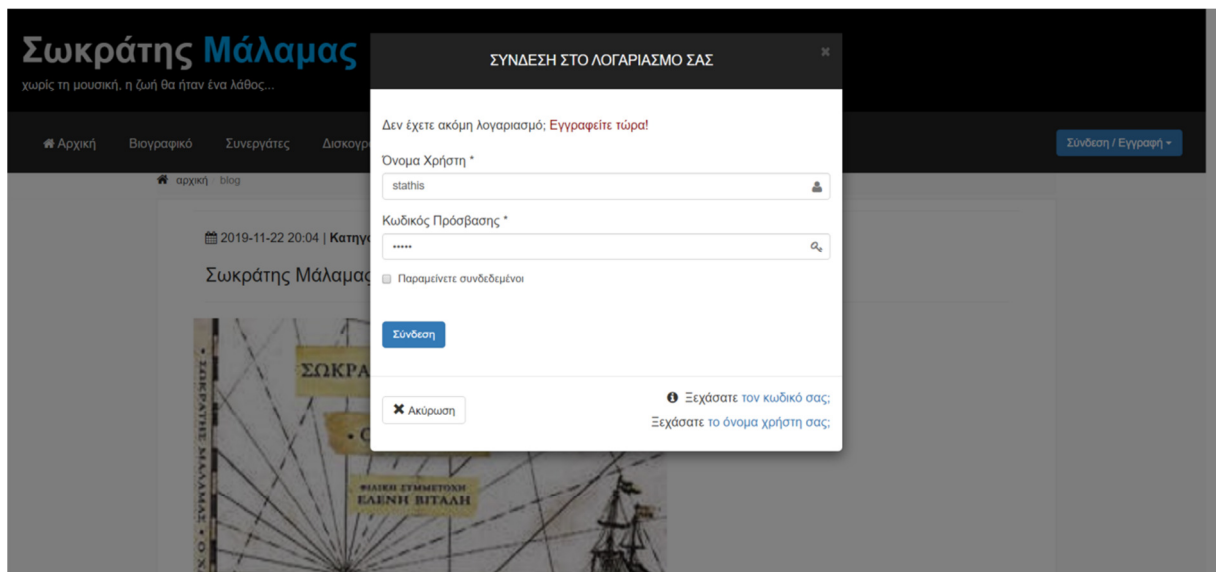
Στη συνέχεια αποστέλλεται email με τον κωδικό πρόσβασης που έχει δημιουργηθεί τυχαία στον νέο χρήστη (Εικόνες 7.2.5).



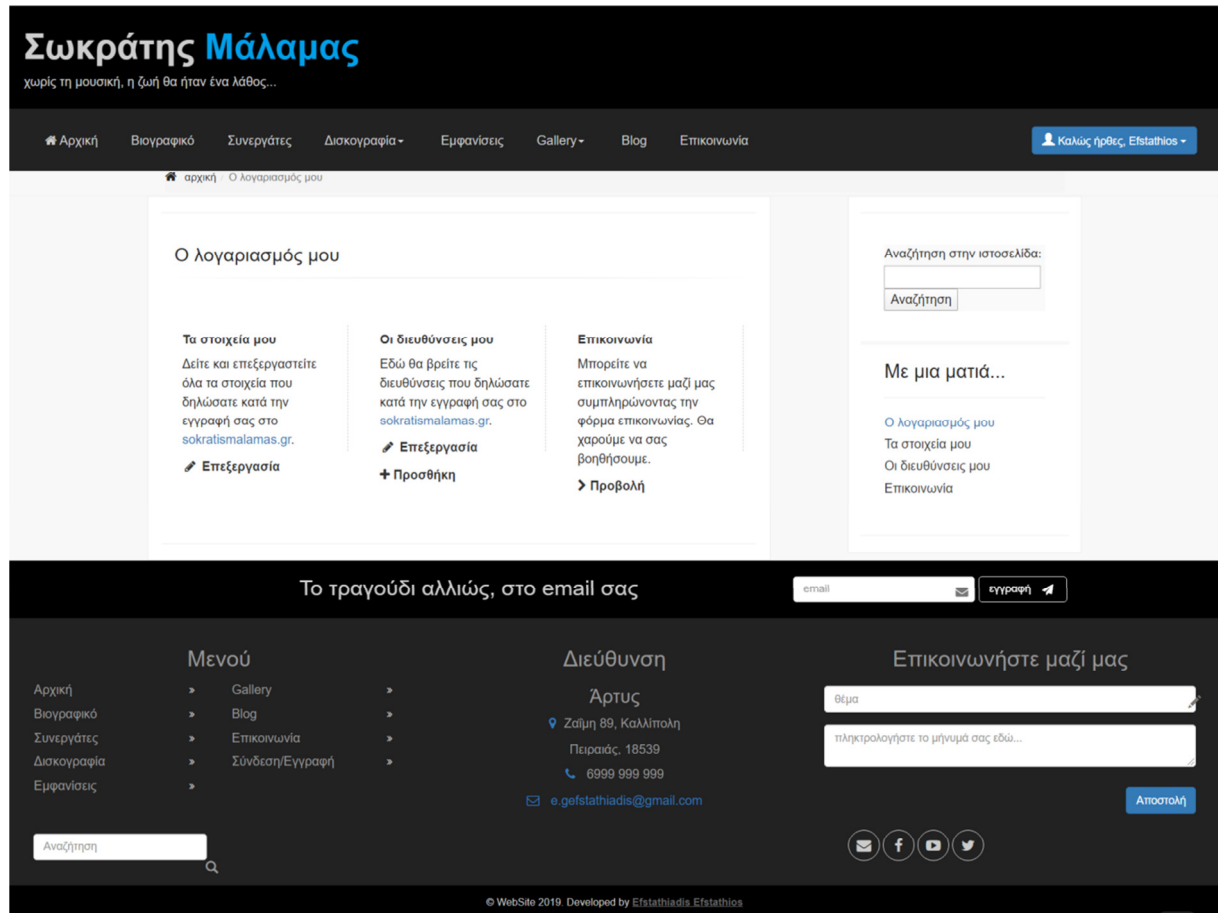
Εικόνα 7.2.5 : Κωδικός πρόσβασης νέου χρήστη

7.3 Σελίδα Χρήστη

Ο χρήστης αφού λάβει με email τον κωδικό πρόσβασης, μπορεί να κάνει είσοδο στην ιστοσελίδα (Εικόνα 7.3.1).

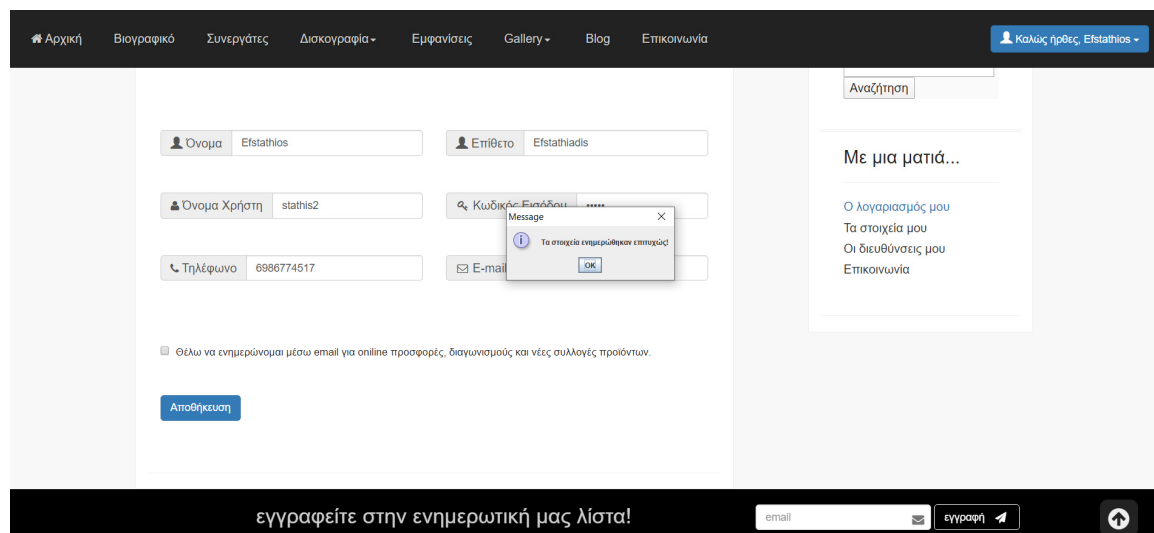


Εικόνα 7.3.1 : Εισαγωγή νέου χρήστη

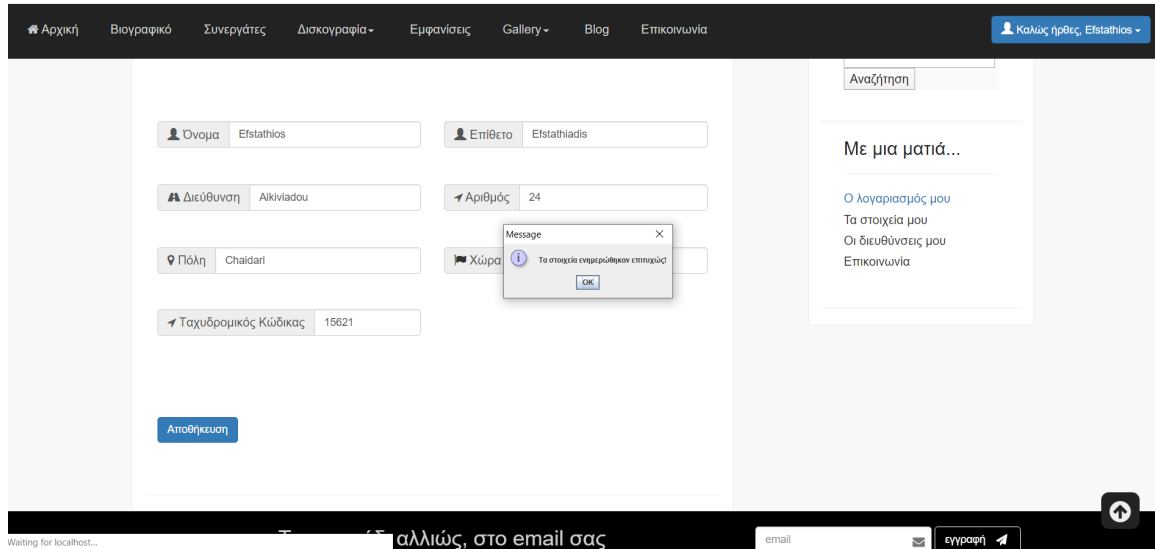


Εικόνα 7.3.2 : Σελίδα νέου χρήστη

Ο εγγεγραμμένος χρήστης έχει τη δυνατότητα να επεξεργαστεί τα προσωπικά του δεδομένα όπως τα προσωπικά στοιχεία και τη διεύθυνση που έχει δηλώσει κατά την εγγραφή του (Εικόνα 7.3.3, Εικόνα 7.3.4).

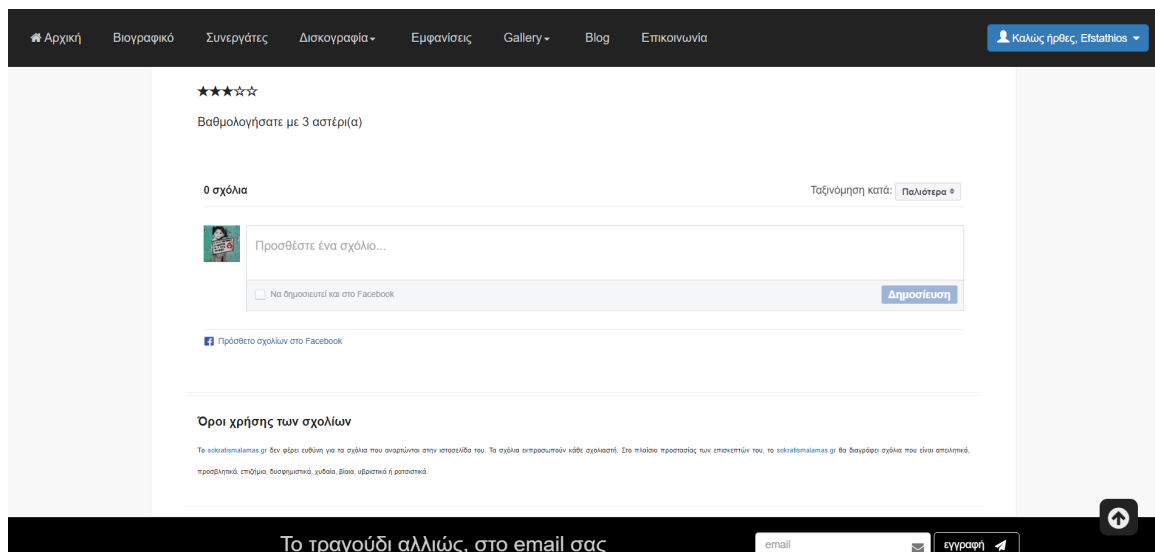


Εικόνα 7.3.3 : Αλλαγή ονόματος νέου χρήστη



Εικόνα 7.3.4 : Αλλαγή αριθμού διεύθυνσης

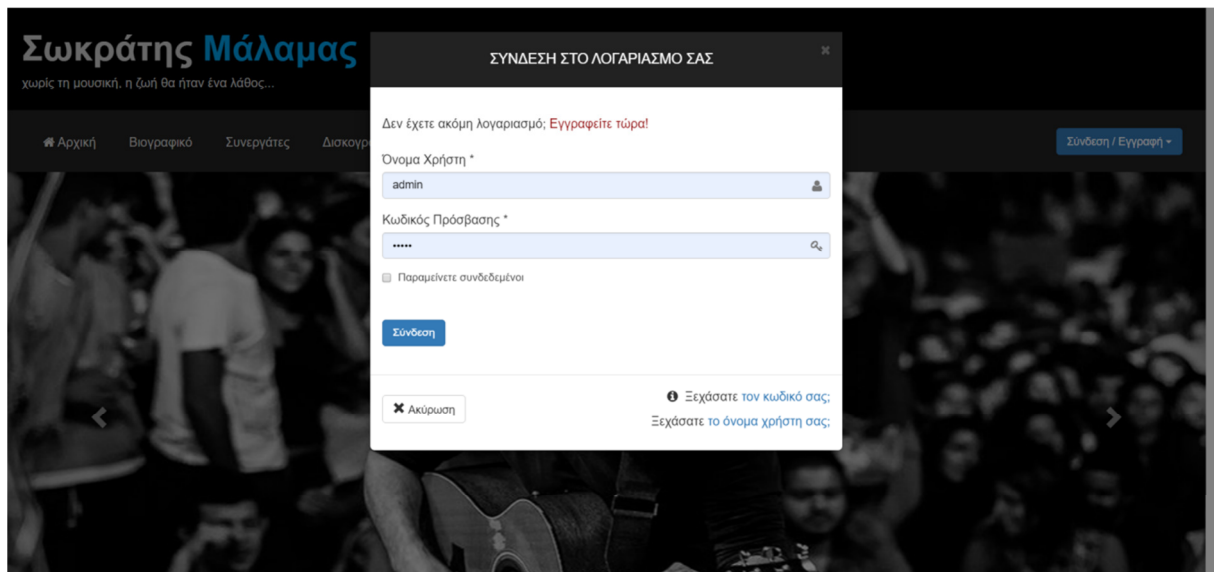
Τέλος, ο εγγεγραμμένος χρήστης έχει τη δυνατότητα να σχολιάσει τα άρθρα της ιστοσελίδας (Εικόνα 7.3.5).



Εικόνα 7.3.5: Σχόλιο άρθρου

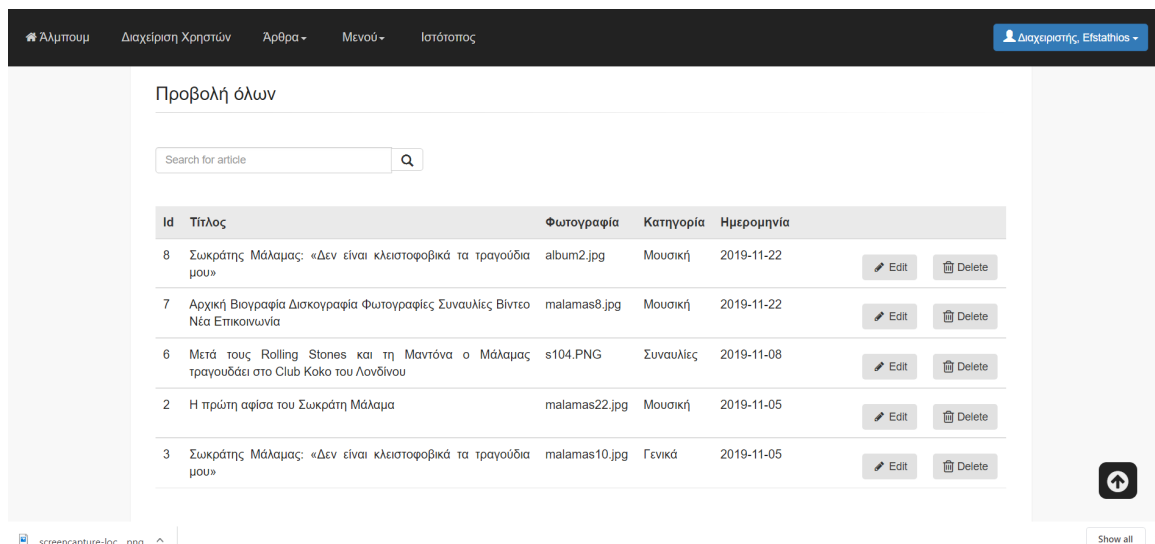
7.4 Σύστημα διαχείρισης περιεχομένου

Στον ιστότοπο «Σωκράτης Μάλαμας» έχει υλοποιηθεί σύστημα διαχείρισης περιεχομένου. Στο σύστημα διαχείρισης περιεχομένου έχει πρόσβαση μόνο ο διαχειριστής του ιστότοπου και μπορεί να εισέλθει με συγκεκριμένο όνομα χρήστη και κωδικό χρήστη (Εικόνα 7.4.1).

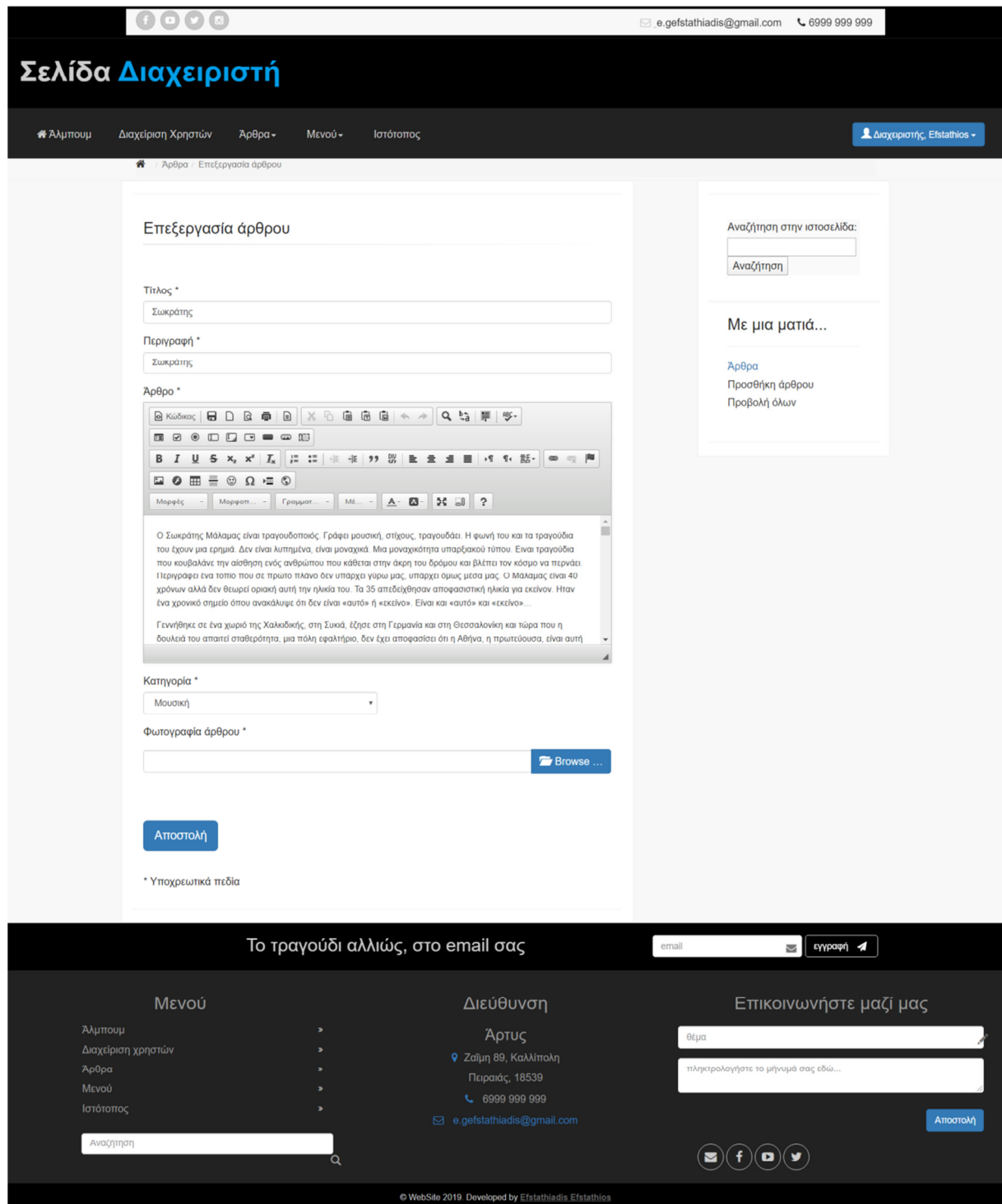


Εικόνα 7.4.1: Είσοδος στο σύστημα διαχείρισης περιεχομένου

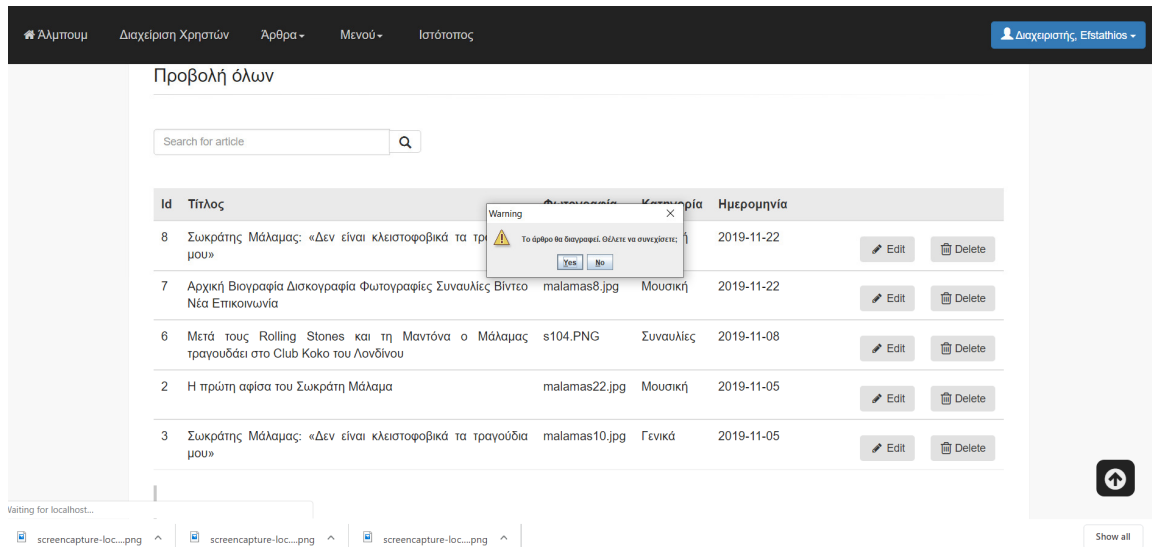
Ο διαχειριστής μέσα από το σύστημα μπορεί να επεξεργαστεί τις εγγραφές του μενού και τα νέα της ιστοσελίδας. Πιο συγκεκριμένα, μπορεί να προσθέσει ένα άρθρο, να το επεξεργαστεί, να το διαγράψει καθώς και να το προβάλει σε μία λίστα με όλα τα άρθρα (Εικόνες 7.4.2 , 7.4.3, 7.4.4).



Εικόνα 7.4.2: Όλα τα άρθρα

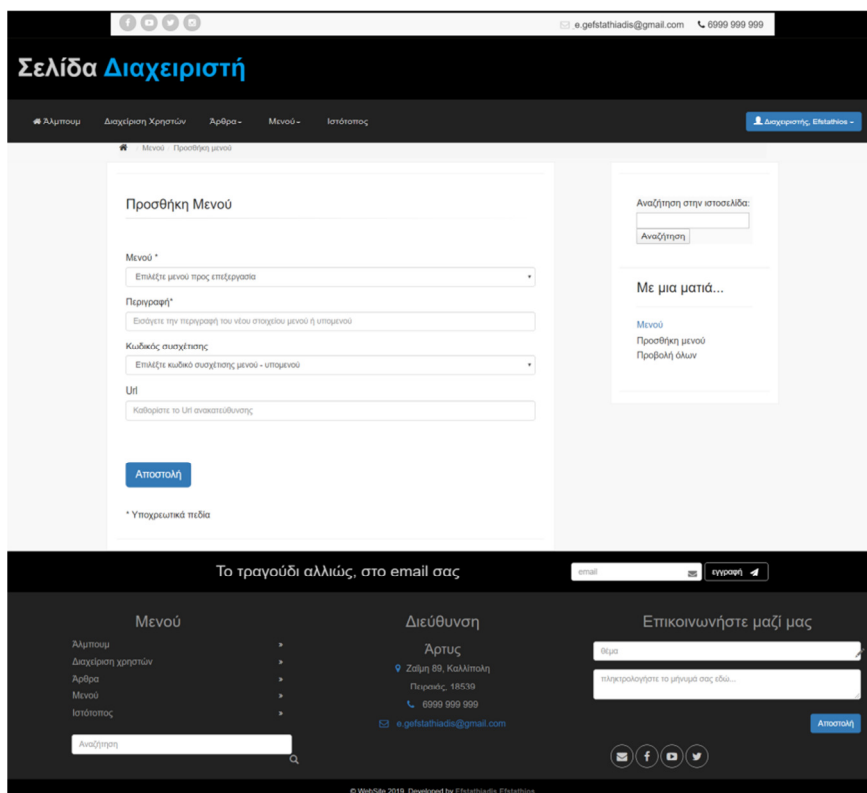


Εικόνα 7.4.3: Επεξεργασία υπάρχοντος άρθρου

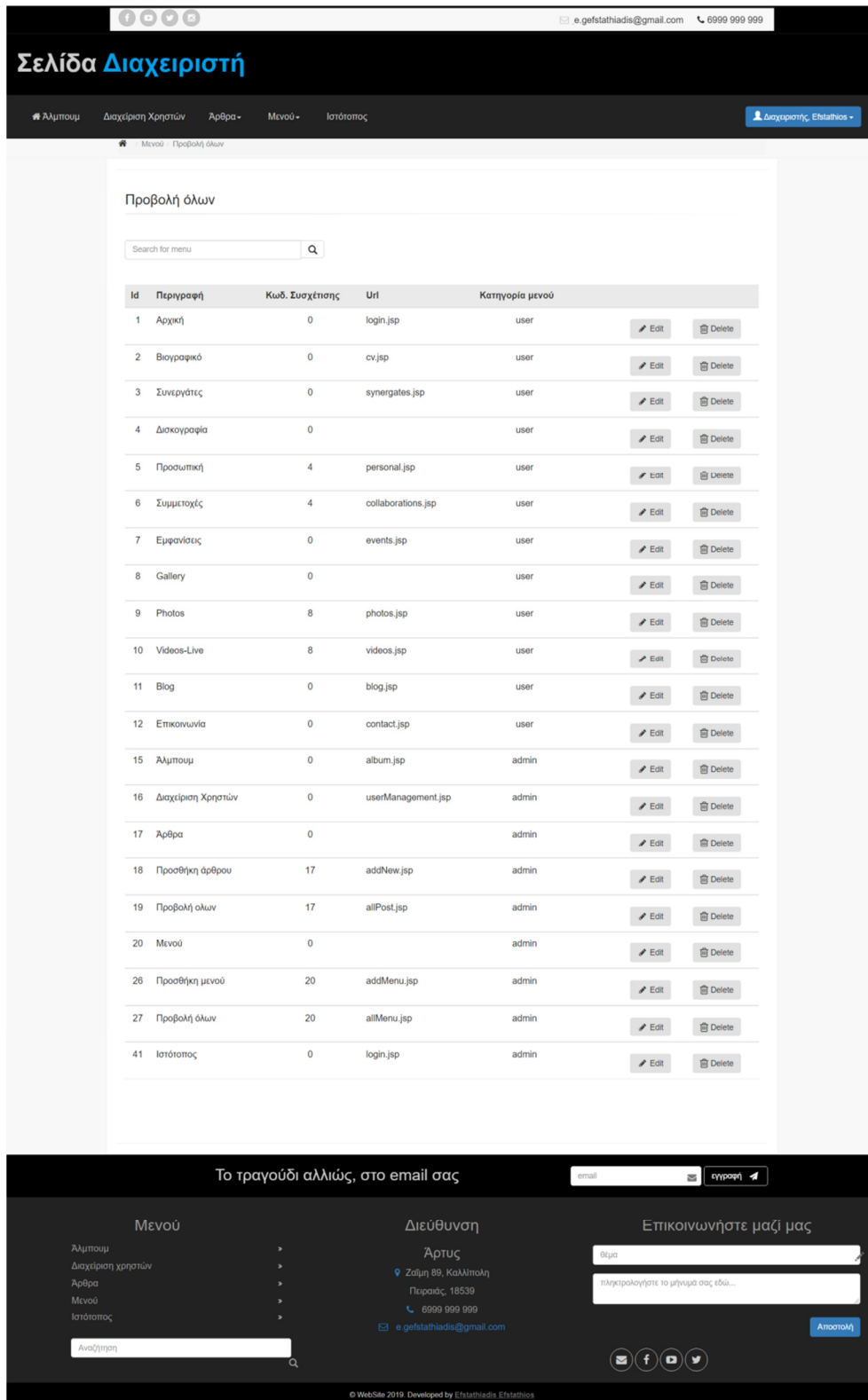


Εικόνα 7.4.4: Διαγραφή υπάρχοντος άρθρου

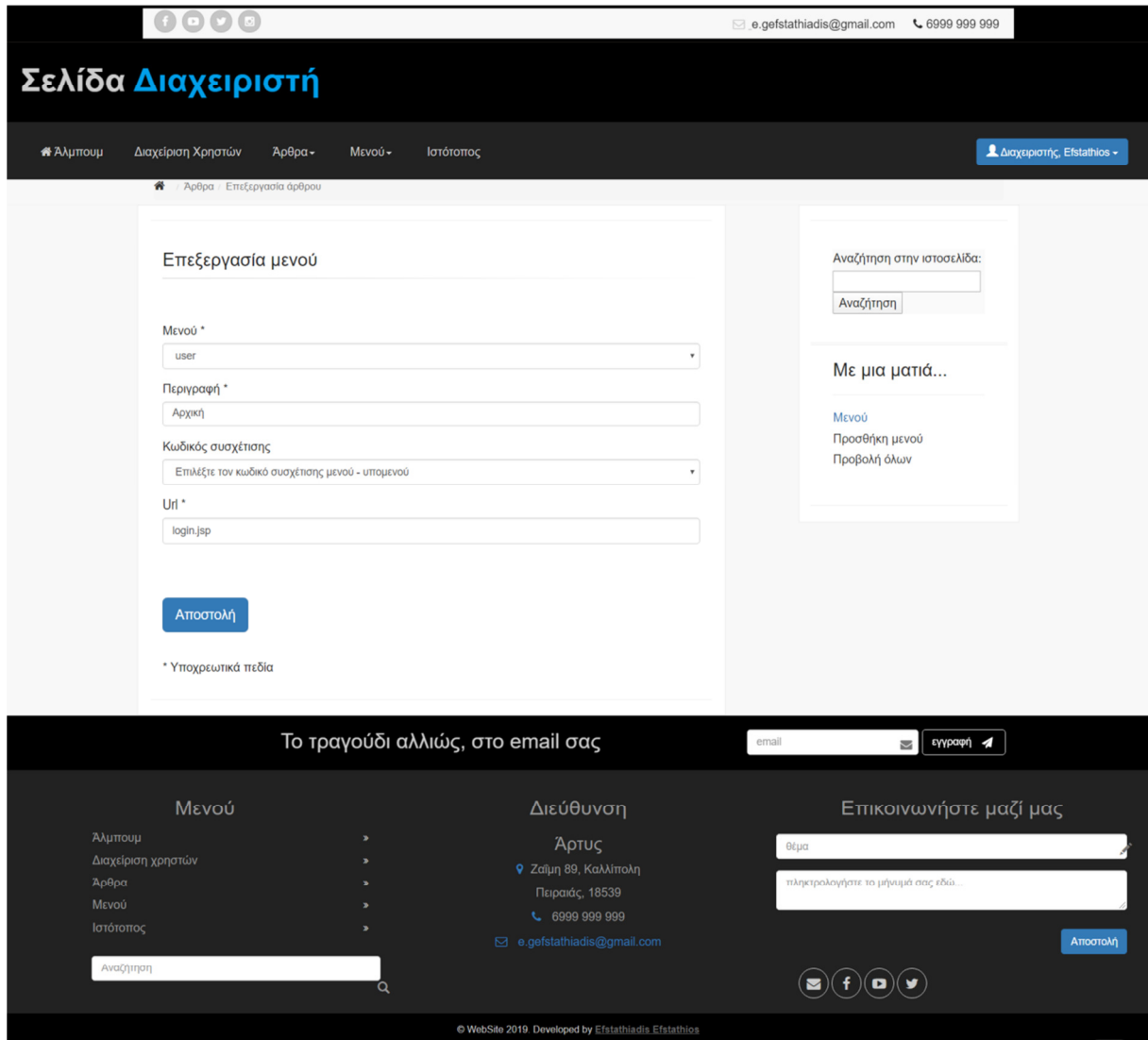
Τέλος, ο διαχειριστής από το σύστημα διαχείρισης περιεχομένου έχει τη δυνατότητα να προσθέσει μία νέα εγγραφή στο μενού, να διαγράψει ή να επεξεργαστεί μία υπάρχουσα εγγραφή (Εικόνες 7.4.5, 7.4.6, 7.4.7, 7.4.8).



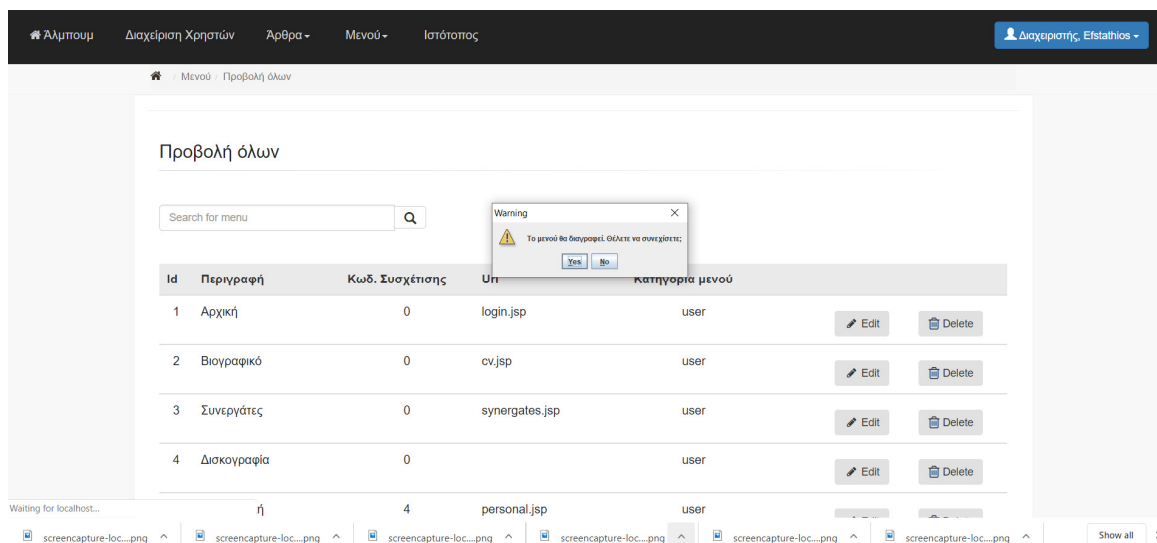
Εικόνα 7.4.5: Προσθήκη νέας εγγραφής στο μενού



Εικόνα 7.4.6: Προβολή όλων των εγγραφών του μενού



Εικόνα 7.4.6: Επεξεργασία υπάρχουσας εγγραφής μενού



Εικόνα 7.4.7: Διαγραφή υπάρχουσας εγγραφής μενού

8. Επίλογος

Συνοψίζοντας, στην παραπάνω εργασία περιγράψαμε την ανάλυση, τον σχεδιασμό και την υλοποίηση μιας δυναμικής responsive ιστοσελίδας ενός μουσικού ,η οποία περιλαμβάνει και ένα συστήματα διαχείρισης περιεχομένου. Το σύστημα διαχείρισης περιεχομένου δίνει τη δυνατότητα στον διαχειριστή να επεξεργαστεί τις εγγραφές του μενού στην ιστοσελίδα καθώς και τα άρθρα που αφορούν τον μουσικό.

Για τη συγκεκριμένη ιστοσελίδα ωστόσο υπάρχουν δυνατότητες βελτίωσης και επέκτασης, οι οποίες αφορούν τόσο το σύστημα διαχείρισης περιεχομένου όσο και τις λειτουργίες του χρήστη. Πιο συγκεκριμένα :

1. Θα δίνεται η δυνατότητα στους εγγεγραμμένους χρήστες να κάνουν αγορά ενός CD μέσω της ιστοσελίδας, όπως επίσης και αγορά εμπορευμάτων που αφορούν τον μουσικό.
2. Ο διαχειριστής θα έχει τη δυνατότητα επεξεργασίας παραγγελιών καθώς και την έκδοση στατιστικών που αφορούν τις πωλήσεις ,μέσω τις ιστοσελίδας. Μέσω του συστήματος διαχείρισης περιεχομένου θα μπορεί, επίσης, να γίνεται διαχείριση των άλμπουμ και των πελατών της ιστοσελίδας.

9. Βιβλιογραφία

1. “Εφαρμογές Διαδικτυακού Προγραμματισμού με JAVA”, Ιωάννης- Χρήστος Παναγιωτόπουλος, Αθήνα 2010
2. “Servlets και σελίδες διακομιστή Java : τεχνολογίες πυρήνα” υπό Hall, Marty, 1962-, Brown, Larry, 1963-, Τσιλογιάννης, Δημήτρης., Σκουλαρικής, Φώτης.
3. Πληροφοριακά Συστήματα στο Διαδίκτυο, Χ. Δουληγέρης, Μητρόπουλος Σαράντης 2015 [ηλεκτρ. βιβλίο]
4. “Εισαγωγή στη γλώσσα προγραμματισμού JAVA”, Εργαστήριο τεχνολογίας πολυμέσων ΕΜΠ
5. <http://www.w3schools.com/>, HTML, CSS, BOOTSTRAP, JAVASCRIPT
6. <https://www.codecademy.com/>
7. <https://developers.google.com/search/mobile-sites>
8. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html#_Toc1455212
9. <https://www.javatpoint.com/hibernate-tutorial>
10. <https://dzone.com/articles/what-servlet-container>
11. https://en.wikipedia.org/wiki/Apache_Tomcat
12. https://en.wikipedia.org/wiki/JavaServer_Pages
13. <http://stackoverflow.com/>
14. Wikipedia
15. <https://repository.kallipos.gr/>
16. <https://www.tipsandtricks-hq.com/top-12-reasons-your-website-needs-to-be-responsive-to-grow-your-business-7514>
17. https://en.wikipedia.org/wiki/Responsive_web_design
18. <https://hibernate.org/orm/>
19. <https://getbootstrap.com/>