



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«ΕΠΑΥΞΗΜΕΝΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΙΣΤΟΣΕΛΙΔΑΣ
ΕΡΓΑΣΤΗΡΙΟΥ ΕΔΑ»**

ΔΗΜΗΤΡΙΟΣ ΓΚΟΓΚΟΣ

Επιβλέπων: Καθηγητής Γεώργιος Σταμούλης



Ιούνιος 2020

THESIS

**«AUGMENTED FUNCTIONALITY OF EDA LAB
WEBPAGE»**

DIMITRIOS GKOGKOS

Supervisor : Professor Georgios Stamoulis

© 2020 Δημήτριος Γκόγκος

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του Προπτυχιακού Διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Ευχαριστίες

Θέλω να ευχαριστήσω μερικούς ανθρώπους σημαντικούς για μένα που χωρίς αυτούς είναι αμφίβολο το αν θα τα είχα καταφέρει να φτάσω στον τερματισμό. Πρώτα απ' όλα να ευχαριστήσω τον κύριο Γιώργο Σταμούλη που ήταν εκεί για μένα από τα αρχαία χρόνια του 2004 που πέρασα στην σχολή και όταν μετά από αμέτρητες ατυχίες και δυσκολίες που με βρήκαν στην πορεία μου βρήκα το σθένος να ξανασπουδάσω στις αρχές του 2017, ήταν εκεί πάντα να με κατευθύνει, να με βοηθάει, να με εμπνέει και να με στηρίζει σαν Καθηγητής και πάνω απ' όλα σαν Άνθρωπος. Εκεί ήταν και την πιο δύσκολη μέρα της ζωής μου. Μου έδωσε τα χέρια του να σηκωθώ όταν έπεσα. Οι λέξεις στο λεξικό δεν φτάνουν για να εκφράσω την ευγνωμοσύνη μου. Να πω απλά ευχαριστώ μέσα από την καρδιά μου. Να ευχαριστήσω θέλω το ίδιο καρδιακά τον κύριο Νέστωρα Ευμορφόπουλο και τον Γιώργο Φλώρο που ήταν και αυτοί δίπλα μου στις πιο δύσκολες μέρες της ζωής μου και δεν με βοηθήσαν μόνο στα μαθήματά μου, αλλά η ψυχική στήριξη που μου χάρισαν απλόχερα είναι πιο πολύτιμη από διαμάντι και έγινε κινητήριος δύναμη για να μπορέσω να συνεχίσω. Θέλω επίσης να ευχαριστήσω τους γονείς μου που ό,τι και να έγινε πάντα πίστευαν σε μένα και δεν με αφήσαν στιγμή να αμφισβητήσω τον εαυτό μου και ήταν και είναι εκεί πάντα σαν προσωπικοί άγγελοι που μου χουν χαριστεί από τη ζωή και ό,τι χρειαστώ δέχομαι από αυτούς σαν δώρο. Φυσικά να ευχαριστήσω τον Θοδωρή Κασιδάκη, που δεν ήταν εκεί μόνο για να κάνουμε τις δοκιμές της εργασίας σε δεύτερη οθόνη και να μου επισημαίνει γενικά σαν δεύτερος παρατηρητής την οπτική του πάνω στο έργο, αλλά η πεντακάθαρη αξιοπρέπεια του είναι έμπνευση που χαρίζει πνοή για την επόμενη μέρα. Τέλος, θέλω να ευχαριστήσω από τα βάθη της ψυχής μου την Μαριλένα Δημακουλέα, που όχι μόνο δεν με αφήνει να πέφτω αλλά με την κρυστάλλινη σοφία της μου μαθαίνει να σκέφτομαι και να είμαι ο Άνθρωπος που είμαι.

Για τον Κούμπα

ΠΕΡΙΛΗΨΗ

Υπάρχουν διάφοροι ρόλοι και ανάλογες ευθύνες για κάθε άτομο ή ομάδα που συνεισφέρει στην σχεδίαση μίας ιστοσελίδας ή γενικά ενός ιστότοπου. Τα πρώτα δύο κεφάλαια της παρούσας εργασίας παρουσιάζουν μια επισκόπηση αυτών των ρόλων και της αντίστοιχης αναγκαίας εργασίας – συνεισφοράς του κάθε ατόμου ή ομάδας που έχει αναλάβει τον αντίστοιχο ρόλο, ώστε να επιτευχθεί όχι μόνο η ολοκλήρωση της σχεδίασης και κατασκευής ενός ιστότοπου αλλά και η λειτουργία του και ο διαμοιρασμός του στο διαδίκτυο. Εκτός των ρόλων – ευθυνών, περιγράφονται οι μηχανισμοί, τα εργαλεία και τα τμήματα που είναι απαραίτητα στην επικοινωνία και την μεταφορά των αρχείων που τον συνιστούν πάνω από το δίκτυο συνήθως μέσω ενός πρωτόκολλου πελάτη – εξυπηρετητή, καθώς και τα επιμέρους στοιχεία που συνιστούν την ανατομία του.

Στη συνέχεια, στο τρίτο κεφάλαιο, περιγράφονται και επισημαίνονται μερικές σημαντικές έννοιες όπως η Σχεδίαση με Ανταπόκριση και πώς η ύπαρξη μίας πληθώρας συσκευών πρέπει να λαμβάνεται υπόψιν στη σχεδίαση ενός ιστότοπου καθώς ανάλογα με το μέγεθος ή την λειτουργικότητα της συσκευής αλλάζει και η εμφάνιση και λειτουργία του. Η κατανόηση αυτών των εννοιών είναι αρκετά σημαντική για την μελέτη του τέταρτου κεφάλαιου, όπου περιγράφεται, μαζί με μια εισαγωγή στις γλώσσες HTML και CSS, η κατασκευή του ιστότοπου του εργαστηρίου με κώδικα HTML και CSS, καθώς τα στοιχεία που αποδίδουν την Ανταπόκριση βρίσκονται σε διάφορα διάσπαρτα σημεία μέσα στον κώδικα.

Όπως και με όλες τις γλώσσες προγραμματισμού, έτσι και με την HTML και τα CSS, ο μόνος τρόπος να μπορέσει κανείς να γράψει σε αυτές τις γλώσσες είναι να πειραματιστεί και να αφιερώσει αρκετές ώρες ώστε να εξοικειωθεί μαζί τους όπως θα έκανε και με μία φυσική γλώσσα όπως τα Ισπανικά ή τα Γαλλικά.

ABSTRACT

There are various roles and responsibilities for each person or group contributing to the design of a webpage or a website in general. The first two chapters of this thesis present an overview of these roles and the equivalent needed work - contribution of each individual or group that is assigned each role, in order to achieve not only the completion of the design and construction of a website but also its operation and sharing on the Internet. In addition to these responsibilities, the mechanisms, tools and necessary parts of communication and transfer of the files that compose it, over the network, usually through a client-server protocol, as well as the components that constitute its anatomy, are also further described.

Subsequently, some important concepts are described and highlighted, such as Responsive Design, and how the existence of a variety of devices should be taken into account in the design of a website, since as the size or functionality of the device changes, so does the appearance and function of the website. Understanding this is rather important for the studying of the fourth chapter, where, along with an introduction to HTML and CSS languages, the construction of the laboratory's website with HTML and CSS code is described, as the elements attributing the website's Responsiveness are located in various scattered parts within the code.

As with all programming languages, so with HTML and CSS, the only way for someone to be able to write in these languages is to experiment and spend several hours in order to get accustomed with them, as he or she would do with a natural language such as Spanish or French.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ.....	10
1.1 Ξεκινώντας με την σχεδίαση ιστοσελίδων	10
1.2 Ρόλοι στη δημιουργία ιστοσελίδων	10
1.2.1 Είδη Σχεδίασης	11
1.2.1.1 Σχεδίαση Εμπειρίας Χρήστη, Αλληλεπίδρασης και Διεπαφής Χρήστη.....	11
1.2.1.2 Οπτική (γραφική) σχεδίαση (Visual (graphic) design).....	14
1.2.2 Δημιουργία κώδικα.....	14
1.2.2.1 Ανάπτυξη frontend (Frontend development).....	14
1.2.2.1.1 Σύνταξη/σήμανση (HTML)	14
1.2.2.1.2 Φύλλα Επικαλυπτόμενου Στυλ (CSS)	15
1.2.2.1.3 Δέσμες ενεργειών JavaScript και DOM (JavaScript and DOM scripting)	15
1.2.2.2 Ανάπτυξη συστημάτων υποστήριξης (Backend development)	17
1.3 Προετοιμασία για τη σχεδίαση ιστοσελίδων	17
1.3.1 Εξοπλισμός	17
1.3.2 Εργαλεία κώδικα	18
1.3.3 Μια ποικιλία προγραμμάτων περιήγησης	18
1.3.4 Διαχείριση Αρχείων και εργαλεία μεταφοράς	18
ΚΕΦΑΛΑΙΟ 2 : ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ ΤΟ ΔΙΑΔΙΚΤΥΟ	20
2.1 Το Internet έναντι του Web	20
2.1.1 Μια μικρή ιστορία του Παγκόσμιου Ιστού	20
2.2 Εξυπηρέτηση Πληροφορίας	21
2.2.1 Χρήσιμη Ορολογία	21
2.2.2 Υπολογιστές και μηχανισμοί που συνθέτουν το διαδίκτυο	22
2.2.2.1 Διευθύνσεις IP και Σύστημα Ονομάτων Τομέων	23

2.2.2.2	Προγράμματα Περιήγησης	23
2.2.2.3	Μηχανισμοί απόδοσης προγράμματος περιήγησης	25
2.2.2.4	Διευθύνσεις Ιστοσελίδων (URLS)	26
2.2.2.4.1	HTTPS, το ασφαλές πρωτόκολλο Web	29
2.3	Η ανατομία μιας ιστοσελίδας	30
2.3.1	Έγγραφα HTML	30
2.3.1.1	Οι εικόνες	32
2.3.2	Προσθέτοντας λίγο Στυλ (CSS)	33
2.3.3	Προσθέτοντας συμπεριφορές με JavaScript	34
2.3.4	Συνθέτοντας τα όλα μαζί	34
2.4	Στατικοί έναντι Δυναμικών Ιστότοπων	37
2.5	Κωδικοί κατάστασης HTTP	37
ΚΕΦΑΛΑΙΟ 3 : ΜΕΡΙΚΕΣ ΣΗΜΑΝΤΙΚΕΣ ΕΝΝΟΙΕΣ		38
3.1	Μια πληθώρα συσκευών	38
3.2	Τήρηση των προτύπων	40
3.3	Προοδευτική Βελτίωση	41
3.4	Σχεδίαση Ιστοσελίδων με Ανταπόκριση (Responsive Web Design)	42
3.5	Προσβασιμότητα	45
ΚΕΦΑΛΑΙΟ 4 : ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ ΤΗΣ ΙΣΤΟΣΕΛΙΔΑΣ ΤΟΥ ΕΡΓΑΣΤΗΡΙΟΥ ΕΔΑ (EDA LAB – ELECTRONICS RESEARCH LAB [ERL])		47
4.1	Μια σχετικά μικρή εισαγωγή	47
4.1.1	Τα στοιχεία του εγγράφου HTML	47
4.1.2	Προσθέτοντας CSS	51
4.2	Η αρχική σελίδα : ιστοσελίδα (καρτέλα) Home	55
4.3	Η ιστοσελίδα People	63
4.4	Η καρτέλα Academics	69
4.4.1	Η ιστοσελίδα Courses	70
4.5	Η καρτέλα Publications	75
4.5.1	Η ιστοσελίδα International Conference Publications	76
4.6	Η καρτέλα Research	79

4.6.1 Η ιστοσελίδα R&D Projects	79
ΚΕΦΑΛΑΙΟ 5 : ΣΥΜΠΕΡΑΣΜΑΤΑ	84
ΒΙΒΛΙΟΓΡΑΦΙΑ	86

Πίνακας Εικόνων

Εικόνα 1 Διάγραμμα περιγράμματος επιφάνειας	13
Εικόνα 2 Διάγραμμα ιστότοπου	13
Εικόνα 3 Τα μέρη μιας διεύθυνσης URL	26
Εικόνα 4 Ορισμένοι διακομιστές εμφανίζουν τα περιεχόμενα του καταλόγου αν το προεπιλεγμένο αρχείο δεν βρεθεί	29
Εικόνα 5 Το πηγαίο αρχείο και το φύλλο στυλ που αποτελούν μια απλή ιστοσελίδα	31
Εικόνα 6 Η μικρή ιστοσελίδα χωρίς CSS (αριστερά) και με CSS (δεξιά)	34
Εικόνα 7 Πώς τα προγράμματα περιήγησης εμφανίζουν ιστοσελίδες	36
Εικόνα 8 Ο Μπραντ Φροστ συνοψίζει επιτυχώς την πραγματικότητα της ποικιλομορφίας των συσκευών	39
Εικόνα 9 Η διάταξη μιας ιστοσελίδας με ανταπόκριση αλλάζει ανάλογα με το μέγεθος του παραθύρου. Πώς φαίνεται στον υπολογιστή (πάνω) και στο κινητό (κάτω αριστερά και κάτω δεξιά)	44
Εικόνα 10 Συμπεριλαμβάνουμε το Bootstrap από ένα CDN (Content Delivery Network - Δίκτυο Παροχής Περιεχομένου)	48
Εικόνα 11 Πώς αλλάζει η προβολή (viewport) στην συσκευή του κινητού ανάλογα με την οθόνη μας αν χρησιμοποιήσουμε το στοιχείο <code><meta name="viewport" content="width=device-width, initial-scale=1.0"></code>	50
Εικόνα 12 Ο κώδικας για το λογότυπο που θα εμφανίζεται όταν προωθείται ο σύνδεσμος του ιστότοπου μέσω εφαρμογών μηνυμάτων	51

Εικόνα 13	Internal CSS (Εσωτερικό φύλλο)	51
Εικόνα 14	Πώς εμφανίζεται το μενού (πάνω) σε οθόνες που δεν «χωράνε» να εμφανιστούν όλες οι καρτέλες του. Πώς εμφανίζεται στην οθόνη του υπολογιστή (κάτω)	53
Εικόνα 15	Κουμπί εναλλαγής του μενού	53
Εικόνα 16	Εισαγωγή γραμματοσειρών από την Google	55
Εικόνα 17	Πρώτη ενότητα της αρχικής σελίδας	56
Εικόνα 18	Κώδικας για το κουτί “ABOUT ERL”	58
Εικόνα 19	Ο διαχωρισμός της ενότητας σε δύο στήλες	59
Εικόνα 20	Ο κώδικας για την πρώτη στήλη	59
Εικόνα 21	Ο κώδικας για την εικονίτσα CERTH	61
Εικόνα 22	Η απόδοση των εικόνων σε μικρές συσκευές (δεξιά) με την χρήση των ιδιοτήτων first, second, third και fourth	62
Εικόνα 23	Χρήση του αναγνωριστικού optionalstuff ορισμένου από το αντίστοιχο CSS	63
Εικόνα 24	Στιγμιότυπο της αλληλεπιδραστικότητας των καρτέλων των ατόμων της ομάδας του εργαστηρίου	64
Εικόνα 25	Η κλάση transition ορισμένη στο εξωτερικό CSS “people.css”	65
Εικόνα 26	Ο κώδικας CSS για το κουτί – «καρτέλα» ατόμου	66
Εικόνα 27	Ο κώδικας HTML για το κουτί – «καρτέλα» ατόμου	67
Εικόνα 28	Ο επιλογέας “hover” που συνεισφέρει σημαντικά στην λειτουργικότητα της μετάβασης	68
Εικόνα 29	Η κλάση “social” και μία ελάχιστη λειτουργικότητα του επιλογέα “hover” σε αυτήν την κλάση	69
Εικόνα 30	Στιγμιότυπο της ιστοσελίδας Courses	71

Εικόνα 31	Ο κώδικας HTML για την δημιουργία ενός αλληλεπιδραστικού κύκλου για το μάθημα Ψηφιακή Σχεδίαση	72
Εικόνα 32	Ο κώδικας CSS για τον ορισμό του μεγάλου κύκλου και της αντίστοιχης μετάβασης	72
Εικόνα 33	Ο κώδικας CSS για τον ορισμό των μικρών κύκλων και της αντίστοιχης μετάβασης	74
Εικόνα 34	Ορισμός της απόστασης των κύκλων σε μικρές και πολύ μικρές οθόνες	75
Εικόνα 35	Στιγμιότυπο της ιστοσελίδας International Conference Publications..	77
Εικόνα 36	Η στοίχιση των χρονολογιών σε μια στήλη	77
Εικόνα 37	Στιγμιότυπο κώδικα HTML της δεξιάς στήλης (των δημοσιεύσεων) ..	78
Εικόνα 38	Η ενσωματωμένη συνάρτηση circle() του CSS	78
Εικόνα 39	Στιγμιότυπο μίας κάρτας της ιστοσελίδας R&D Projects	80
Εικόνα 40	Ορισμός των κλάσεων για την κατασκευή της κάρτας	80
Εικόνα 41	Ο κώδικας HTML για την κατασκευή μίας κάρτας	82
Εικόνα 42	Ορισμός του πλάτους και των ορίων της κάρτας ειδικά για μικρές και πολύ μικρές συσκευές	83

Πίνακας πινάκων

Πίνακας 1	Μηχανισμοί απόδοσης (Rendering Engines)	25
-----------	---	----

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Ξεκινώντας με την σχεδίαση ιστοσελίδων

Το διαδίκτυο υπάρχει εδώ και πάνω από 25 χρόνια, βιώνοντας μια ευφορική πρώιμη εξάπλωση, μια οικονομική κατάρρευση, μια αναγέννηση που βασίζεται στην καινοτομία και συνεχή εξέλιξη στην πορεία. Ένα είναι σίγουρο: το διαδίκτυο ως μέσο επικοινωνίας και εμπορίου είναι εδώ για να μείνει. Όχι μόνο αυτό, έχει βρει το δρόμο του σε συσκευές όπως smartphone, tablet, τηλεοράσεις και πολλές άλλες. Δεν υπήρξαν ποτέ περισσότερες ευκαιρίες για να αξιοποιηθεί η τεχνογνωσία του σχεδιασμού ιστοσελίδων.

Όποια και αν είναι τα κίνητρα, η πρώτη ερώτηση είναι πάντα η ίδια: "Πού μπορώ να ξεκινήσω;" Μπορεί να φαίνεται σαν να υπάρχει ένα βουνό από πράγματα για να μάθεις και δεν είναι εύκολο να ξέρεις από πού να πιαστείς. Αλλά πρέπει να ξεκινήσεις από κάπου. Αυτό το κεφάλαιο παρέχει μια επισκόπηση του αντικειμένου πριν αρχίσω να περιγράψω την κατασκευή ή τα στοιχεία της ιστοσελίδας. Αρχίζει με την εισαγωγή των ρόλων και των ευθυνών που σχετίζεται με τη δημιουργία ιστοσελίδων.

1.2 Ρόλοι στη δημιουργία ιστοσελίδων

Κοιτώντας μια ιστοσελίδα, μπορεί να φανταστεί κανείς την πληθώρα αποφάσεων και τομέων εξειδίκευσης που χρειάστηκε για να χτιστεί. Οι ιστοσελίδες δεν είναι μόνο κώδικας και εικόνες. Συχνά ξεκινούν με ένα επιχειρηματικό σχέδιο ή κάποια άλλη προκαθορισμένη αποστολή. Πριν να τεθούν σε λειτουργία, το περιεχόμενο τους πρέπει να δημιουργηθεί και να οργανωθεί, να εκτελεστεί έρευνα, να γίνει σχεδίαση των ευρύτερων στόχων μέχρι τις καλύτερες λεπτομέρειες, να γραφτεί κώδικας και όλα θα πρέπει να συντονιστούν με όσα συμβαίνουν στον εξυπηρετητή για να καρποφορήσουν. Μεγάλοι, γνωστοί ιστότοποι δημιουργούνται από ομάδες δεκάδων, εκατοντάδων ή ακόμα και χιλιάδων συνεισφερόντων. Υπάρχουν επίσης ιστοσελίδες που δημιουργούνται και διαχειρίζονται από μια ομάδα με λίγα μόνο μέλη. Είναι επίσης απολύτως εφικτό να δημιουργήσεις μια

αξιοσέβαστη ιστοσελίδα με μια ομάδα που να αποτελείται μόνο από σένα. Αυτή είναι η ομορφιά του διαδικτύου.

1.2.1 Είδη Σχεδίασης

Η σχεδίαση ακούγεται αρκετά απλή, αλλά ακόμα και αυτή η απλή απαίτηση έχει χωριστεί σε διάφορες ειδικότητες όσον αφορά τη δημιουργία ιστότοπων. Ακολουθούν μερικές από τις περιγραφές εργασιών που σχετίζονται με το σχεδιασμό ενός ιστότοπου, αλλά πρέπει να θυμόμαστε ότι οι κλάδοι συχνά επικαλύπτονται και ότι το πρόσωπο που καλείται συχνά "σχεδιαστής" είναι υπεύθυνος για περισσότερες από μία (αν όχι για όλες) αυτές τις διεργασίες.

1.2.1.1 Σχεδίαση Εμπειρίας Χρήστη, Αλληλεπίδρασης και Διεπαφής Χρήστη

Συχνά, όταν σκεφτόμαστε τη σχεδίαση, σκεφτόμαστε το πώς κάτι φαίνεται . Στο διαδίκτυο, το πρώτο θέμα του έργου είναι η σχεδίαση του τρόπου λειτουργίας του ιστότοπου. Πριν να επιλέξεις χρώματα και γραμματοσειρές, είναι σημαντικό να προσδιορίσεις τους στόχους του ιστότοπου, τον τρόπο με τον οποίο θα χρησιμοποιηθεί και πώς θα περιηγηθούν σε αυτόν οι επισκέπτες. Αυτές οι εργασίες εμπίπτουν στους τομείς της Σχεδίασης της Εμπειρίας Χρήστη ([User Experience \(UX\) design](#)), της Σχεδίασης Αλληλεπίδρασης ([Interaction Design \(IXD\)](#)) και της Σχεδίασης Διεπαφής Χρήστη ([User Interface \(UI\) design](#)). Υπάρχει μεγάλη αλληλοεπικάλυψη μεταξύ αυτών των ευθυνών και δεν είναι ασυνήθιστο για ένα άτομο ή μια ομάδα να διαχειρίζεται και τις τρεις. Ο σχεδιαστής εμπειρίας χρήστη έχει μια ολιστική άποψη της διαδικασίας σχεδίασης εξασφαλίζοντας ότι όλη η εμπειρία του χρήστη με τον ιστότοπο είναι ευνοϊκή. Η Σχεδίαση της Εμπειρίας Χρήστη ([User Experience \(UX\) design](#)), βασίζεται στη σαφή κατανόηση των χρηστών και των αναγκών τους βάσει παρατηρήσεων και συνεντεύξεων. Σύμφωνα με τον Ντόναλντ Νόρμαν (ο οποίος επινόησε τον όρο), η Σχεδίαση της Εμπειρίας Χρήστη περιλαμβάνει «όλες τις πτυχές της αλληλεπίδρασης του χρήστη με το προϊόν: πώς είναι αντιληπτή, καταρτισμένη και χρησιμοποιούμενη.» Για μια ιστοσελίδα ή εφαρμογή, αυτό περιλαμβάνει την

οπτική σχεδίαση, την διεπαφή χρήστη, την ποιότητα και το μήνυμα του περιεχομένου και ακόμη και τη συνολική απόδοση του ιστότοπου. Ο στόχος του Σχεδιαστή Αλληλεπίδρασης ([Interaction Designer](#)) είναι να κάνει την τοποθεσία πιο εύκολη, αποτελεσματική και απολαυστική για χρήση. Στενά συνδεδεμένη με τη Σχεδίαση Αλληλεπίδρασης είναι η Σχεδίαση Διεπαφής Χρήστη ([User Interface \(UI\) design](#)), που τείνει να εστιάζει πιο πολύ στη λειτουργία οργάνωσης της σελίδας καθώς και στα συγκεκριμένα εργαλεία (κουμπιά, συνδέσμους, μενού, και ούτω καθεξής) που χρησιμοποιούν οι χρήστες για να περιηγηθούν στο περιεχόμενο ή να εκτελέσουν εργασίες.

Τα παρακάτω είναι παραδοτέα που παράγουν οι σχεδιαστές UX, UI ή αλληλεπίδρασης:

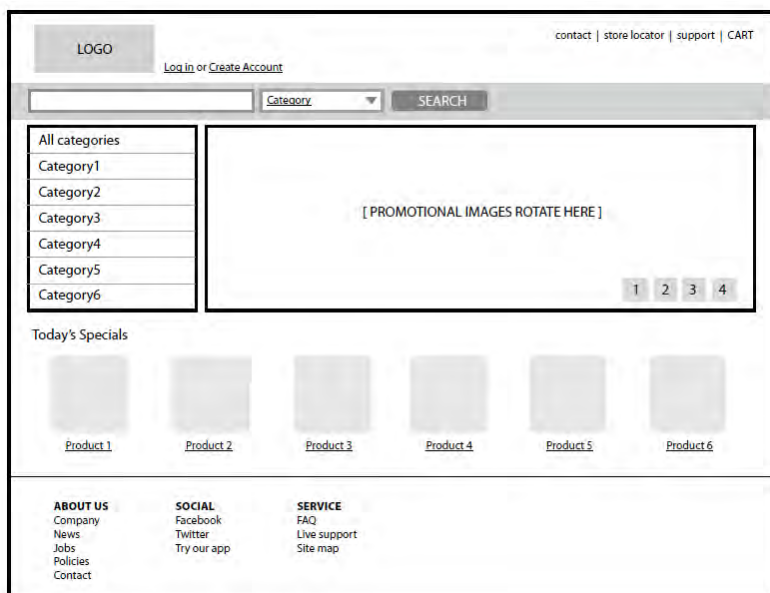
Έρευνα χρηστών και εκθέσεις δοκιμών

Η κατανόηση των αναγκών, των επιθυμιών και των περιορισμών των χρηστών είναι σημαντική για την επιτυχία του σχεδιασμού της ιστοσελίδας ή της εφαρμογής διαδικτύου. Η προσέγγιση της σχεδίασης γύρω από τις ανάγκες του χρήστη αναφέρεται ως Σχεδίαση με Επίκεντρο το Χρήστη ([User-Centered Design \(UCD\)](#)) και είναι κεντρικής σημασίας στη σύγχρονη σχεδίαση ιστοσελίδας. Η σχεδίαση ιστοσελίδων συχνά ξεκινά με την έρευνα των χρηστών, συμπεριλαμβανομένων συνεντεύξεων και παρατηρήσεων, προκειμένου να επιτευχθεί καλύτερη κατανόηση του τρόπου με τον οποίο ο ιστότοπος μπορεί να επιλύσει προβλήματα ή τον τρόπο με τον οποίο θα χρησιμοποιηθεί. Είναι τυπικό για τους σχεδιαστές να κάνουν έναν κύκλο δοκιμών χρήστη στην κάθε φάση της διαδικασίας σχεδίασης, ώστε να εξασφαλίζεται η χρηστικότητα των σχεδίων τους. Εάν οι χρήστες δυσκολεύονται να αντιληφθούν πού θα βρουν κάποιο περιεχόμενο ή πώς να μεταβούν στο επόμενο βήμα μιας διαδικασίας, τότε χρειάζεται επιστροφή στον πίνακα σχεδίασης.

Διαγράμματα περιγράμματος επιφάνειας (*Wireframe diagrams*)

Ένα διάγραμμα περιγράμματος επιφάνειας δείχνει τη δομή μιας ιστοσελίδας χρησιμοποιώντας μόνο περιγράμματα για κάθε τύπο περιεχομένου και γραφικού στοιχείου (Εικόνα 1). Ο σκοπός του διαγράμματος περιγράμματος επιφάνειας είναι να υποδεικνύει τον τρόπο διαίρεσης της επιφάνειας οθόνης και που τοποθετούνται

λειτουργίες και περιεχόμενο όπως περιήγηση, πλαίσια αναζήτησης, φόρμες κ.ο.κ. Χρώματα, γραμματοσειρές και άλλα στοιχεία οπτικής ταυτότητας παραλείπονται σκόπιμα, ώστε να μην αποσπούν την προσοχή από τη δομή της σελίδας. Αυτά τα διαγράμματα συνήθως σχολιάζονται με οδηγίες για το πώς πρέπει να λειτουργούν τα πράγματα, ώστε η ομάδα ανάπτυξης να ξέρει τι να χτίσει.

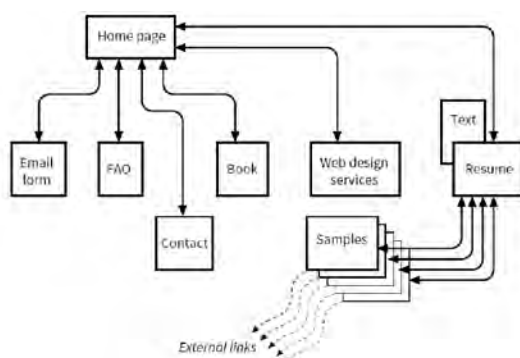


Εικόνα 1. Διάγραμμα περιγράμματος επιφάνειας.

Πηγή: [1]

Διάγραμμα ιστότοπου (Site diagram)

Ένα διάγραμμα τοποθεσίας υποδεικνύει τη δομή του ιστότοπου ως σύνολο και τον τρόπο με τον οποίο η κάθε μία από τις σελίδες σχετίζονται μεταξύ τους. Η Εικόνα 2 δείχνει ένα πολύ απλό διάγραμμα ιστότοπου. Μερικά διαγράμματα ιστότοπων γεμίζουν ολόκληρους τοίχους.



Εικόνα 2. Διάγραμμα ιστότοπου.

Πηγή: [1]

1.2.1.2 Οπτική (γραφική) σχεδίαση (Visual (graphic) design)

Επειδή το Web είναι ένα οπτικό μέσο, οι ιστοσελίδες απαιτούν προσοχή στην οπτική παρουσίαση. Οι πρώτες εντυπώσεις είναι τα πάντα. Ένας σχεδιαστής γραφικών δημιουργεί την "εμφάνιση και αίσθηση" της τοποθεσίας, λογότυπα, γραφικά, τύπους, χρώματα, διάταξη κ.ο.κ., για να εξασφαλίσει ότι η τοποθεσία θα έχει καλή πρώτη εντύπωση και θα είναι συνεπής με την επωνυμία και το μήνυμα του οργανισμού που αντιπροσωπεύει. Οι σχεδιαστές μπορεί επίσης να είναι υπεύθυνοι για τη δημιουργία ενός οδηγού στυλ που θα περιλαμβάνει τις επιλογές στυλ, όπως γραμματοσειρές, χρώματα και άλλα διακοσμητικά στυλ, ώστε να διατηρηθεί η συνέπεια της τοποθεσίας με την πάροδο του χρόνου.

1.2.2 Δημιουργία κώδικα

Ένα μεγάλο μέρος της διαδικασίας δημιουργίας ιστοσελίδων περιλαμβάνει τη δημιουργία των εγγράφων, των φύλλων στυλ (style sheets), των δεσμών ενεργειών (scripts) και των εικόνων που αποτελούν έναν ιστότοπο καθώς και την αντιμετώπιση προβλημάτων που σχετίζονται με όλα αυτά. Στις εταιρείες σχεδιασμού ιστοσελίδων, η ομάδα που χειρίζεται τη δημιουργία των αρχείων που συνθέτουν την τοποθεσία (ή πρότυπα για σελίδες που συναρμολογούνται δυναμικά) αποκαλείται τμήμα ανάπτυξης ή παραγωγής. Η ανάπτυξη εμπίπτει σε δύο μεγάλες κατηγορίες: ανάπτυξη **frontend** και ανάπτυξη συστημάτων υποστήριξης (**backend**). Για μια ακόμη φορά, αυτά τα καθήκοντα μπορεί να ανατεθούν σε ειδικούς, αλλά μπορεί ένα άτομο ή μια ομάδα να διαχειρίζεται και τις δύο ευθύνες.

1.2.2.1 Ανάπτυξη **frontend** (Frontend development)

Η ανάπτυξη **frontend** αναφέρεται σε οποιαδήποτε πτυχή της διαδικασίας σχεδίασης που εμφανίζεται ή σχετίζεται απευθείας με το πρόγραμμα περιήγησης. Αυτό περιλαμβάνει HTML, CSS και JavaScript.

1.2.2.1.1 Σύνταξη/σήμανση (HTML)

Η σύνταξη είναι η διαδικασία προετοιμασίας του περιεχομένου για παράδοση του στο διαδίκτυο ή πιο συγκεκριμένα, η επισήμανση του περιεχομένου με ετικέτες

HTML που περιγράφουν το περιεχόμενό του και τη λειτουργία του. Η HTML (HyperText Markup Language) είναι η γλώσσα σύνταξης που χρησιμοποιείται για τη δημιουργία εγγράφων ιστοσελίδας. Η τρέχουσα έκδοση είναι η HTML 5.2. Η HTML δεν είναι γλώσσα προγραμματισμού· είναι γλώσσα σήμανσης, που σημαίνει είναι ένα σύστημα για τον προσδιορισμό και την περιγραφή των διαφόρων στοιχείων ενός εγγράφου όπως επικεφαλίδες, παραγράφοι και λίστες. Η σήμανση δηλώνει την υποκείμενη δομή του εγγράφου (μπορεί να θεωρηθεί λεπτομερής, αναγνώσιμη από μηχάνημα, διάρθρωση). Δεν χρειάζονται δεξιότητες προγραμματισμού —μόνο υπομονή και κοινή λογική— για την εγγραφή HTML. Ο καλύτερος τρόπος για να μάθει κανείς HTML είναι να γράψει κάποιες σελίδες με το χέρι.

1.2.2.1.2 Φύλλα Επικαλυπτόμενου Στυλ (CSS)

Ενώ η HTML χρησιμοποιείται για την περιγραφή του περιεχομένου σε μια ιστοσελίδα, τα Φύλλα Επικαλυπτόμενου Στυλ (Cascading StyleSheets (CSS)) περιγράφουν τον τρόπο εμφάνισης αυτού του περιεχομένου. Ο τρόπος που εμφανίζεται η ιστοσελίδα αναφέρεται ως παρουσίασή της. Γραμματοσειρές, χρώματα, εικόνες φόντου, το διάστιχο, η διάταξη σελίδας κ.ο.κ. ελέγχονται με CSS. Μπορεί κανείς ακόμα να προσθέσει ειδικά εφέ και βασική κίνηση εικόνας στη σελίδα του. Η προδιαγραφή CSS παρέχει επίσης μεθόδους για τον έλεγχο του τρόπου με τον οποίο τα έγγραφα θα παρουσιαστούν σε περιβάλλοντα διαφορετικά από κάποιο πρόγραμμα περιήγησης, όπως σε εκτύπωση ή ανάγνωση από συσκευή ανάγνωσης οθόνης. Αν και είναι δυνατή η δημοσίευση ιστοσελίδων μόνο με τη χρήση HTML, πιθανότατα να χρειαστεί κανείς να χρησιμοποιήσει CSS για να μην μείνει περιορισμένος στο προεπιλεγμένο στυλ του προγράμματος περιήγησης. Εάν κάποιος ενδιαφέρεται επαγγελματικά για τη σχεδίαση τοποθεσιών Web, είτε ως σχεδιαστής είτε ως προγραμματιστής, η επάρκεια σε CSS είναι υποχρεωτική.

1.2.2.1.3 Δέσμες ενεργειών JavaScript και DOM (JavaScript and DOM scripting)

Η JavaScript είναι μια γλώσσα δέσμης ενεργειών που προσθέτει αλληλεπίδραση και συμπεριφορές στις ιστοσελίδες, συμπεριλαμβανομένων αυτών είναι οι εξής:

- Έλεγχος καταχωρήσεων φόρμας για έγκυρες καταχωρήσεις

- Ανταλλαγή στυλ για ένα στοιχείο ή για μια ολόκληρη τοποθεσία
- Αυτόματη φόρτωση τροφοδοσιών κύλισης με περισσότερο περιεχόμενο
- Να γίνεται το πρόγραμμα περιήγησης να απομνημονεύει πληροφορίες σχετικά με τους χρήστες
- Δημιουργία γραφικών συστατικών, όπως ενσωματωμένα προγράμματα αναπαραγωγής βίντεο ή είσοδοι ειδικής φόρμας

Υπάρχει επίσης και ο όρος δέσμες ενεργειών DOM που χρησιμοποιείται σε σχέση με την JavaScript. Το DOM σημαίνει Document Object Model (Μοντέλο αντικειμένου εγγράφου) και αναφέρεται στην τυποποιημένη λίστα στοιχείων ιστοσελίδας στα οποία είναι δυνατή η πρόσβαση και ο χειρισμός με χρήση JavaScript (ή άλλη γλώσσα δέσμης ενεργειών). Ενδέχεται επίσης να απαιτείται από τους frontend προγραμματιστές να είναι εξοικειωμένοι με τα JavaScript πλαίσια (όπως React, Bootstrap, Angular και άλλα) που αυτοματοποιούν ένα μεγάλο μέρος της διαδικασίας παραγωγής. Είναι πιθανό να χρειαστεί επίσης να είναι εξοικειωμένοι με το AJAX (που σημαίνει "Ασύγχρονη JavaScript και XML" (Asynchronous JavaScript And XML)), μια τεχνική που χρησιμοποιείται για φόρτωση περιεχομένου στο παρασκήνιο, επιτρέποντας στην σελίδα να ενημερώνεται ομαλά χωρίς επαναφόρτωση (όπως αυτές που ανανεώνονται αυτόματα). Η δημιουργία δεσμών ενεργειών στο Web απαιτεί οπωσδήποτε κάποια παραδοσιακή γνώση προγραμματισμού. Ενώ πολλοί διαδικτυακοί προγραμματιστές έχουν πτυχία στην επιστήμη των υπολογιστών, είναι επίσης συχνό για τους προγραμματιστές να είναι αυτοδίδακτοι. Ωστόσο, αν δεν έχει κανείς εμπειρία με τις γλώσσες προγραμματισμού, η αρχική πορεία μάθησης μπορεί να είναι λίγο απότομη. Αν θέλει κανείς να γίνετε προγραμματιστής ιστοσελίδων για τα προς το ζην, η JavaScript είναι βασική απαίτηση. Οι σχεδιαστές θα επωφεληθούν από την κατανόηση του τι μπορεί να κάνει η JavaScript, αλλά ίσως να μην χρειάζεται να μάθουν να τη γράφουν αν εργάζονται με μια ομάδα ανάπτυξης.

1.2.2.2 Ανάπτυξη συστημάτων υποστήριξης (Backend development)

Οι προγραμματιστές συστημάτων υποστήριξης εστιάζουν στον διακομιστή/εξυπηρετητή (server), συμπεριλαμβανομένων των εφαρμογών και των βάσεων δεδομένων που «τρέχουν» πάνω του. Μπορεί να είναι υπεύθυνοι για την εγκατάσταση και τη ρύθμιση του λογισμικού διακομιστών. Θα πρέπει οπωσδήποτε να γνωρίζουν τουλάχιστον μία και πιθανώς περισσότερες server-side γλώσσες προγραμματισμού όπως PHP, Ruby, .NET (ή ASP.NET), Python ή JSP, για τη δημιουργία εφαρμογών που παρέχουν την λειτουργικότητα που απαιτείται από τον ιστότοπο. Οι εφαρμογές χειρίζονται εργασίες και δυνατότητες όπως επεξεργασία φόρμών, συστήματα διαχείρισης περιεχομένου (Content Management Systems (CMS)), αγορές μέσω διαδικτύου και άλλα. Επιπλέον, οι προγραμματιστές συστημάτων υποστήριξης πρέπει να είναι εξοικειωμένοι με τη διαμόρφωση και τη διατήρηση βάσεων δεδομένων που αποθηκεύουν όλα τα δεδομένα για μια τοποθεσία, όπως το περιεχόμενο που μετατρέπεται σε πρότυπα, λογαριασμούς χρηστών, απογραφές προϊόντων και άλλα. Ορισμένες κοινές γλώσσες βάσεων δεδομένων είναι οι MySQL, Oracle και SQL Server. Η ανάπτυξη του συστήματος υποστήριξης είναι πολύ πιο πέρα από το πεδίο αυτής της εργασίας, αλλά είναι σημαντικό να γνωρίζει κανείς τα είδη των εργασιών που διεκπεραιώνονται σε επίπεδο διακομιστή. Θα πρέπει να γνωρίζει ότι είναι δυνατόν να αποκτήσει λειτουργικότητα όπως τα καλάθια για αγορές (shopping carts), λίστες αλληλογραφίας, κ.ο.κ. ως προσυσκευασμένες λύσεις από την εταιρεία φιλοξενίας της ιστοσελίδας χωρίς να χρειάζεται να το προγραμματίσει από το μηδέν.

1.3 Προετοιμασία για τη σχεδίαση ιστοσελίδων

1.3.1 Εξοπλισμός

Για ένα άνετο περιβάλλον διαδικτυακής ανάπτυξης, χρειάζεται ο ακόλουθος εξοπλισμός: Ένας αξιόπιστος, ενημερωμένος υπολογιστής, μία μεγάλη οθόνη, ένας δεύτερος υπολογιστής για δοκιμή, κινητές συσκευές για δοκιμή και ίσως ένα scanner ή μια κάμερα.

1.3.2 Εργαλεία κώδικα

Αν και μπορεί κανείς να τα βγάλει πέρα με τα απλά προγράμματα επεξεργασίας κειμένου του υπολογιστή, ένας αποκλειστικά επεξεργαστής κώδικα (code editor), κάνει την σύνταξη HTML, CSS και JavaScript πολύ ευκολότερη. Οι εφαρμογές επεξεργασίας κώδικα κατανοούν τη σύνταξη του κώδικα που χρησιμοποιείται για εγγραφή, ώστε να μπορούν να κάνουν πράγματα για εμάς όπως χρωματική κωδικοποίηση, ανίχνευση σφαλμάτων και αυτόματη ολοκλήρωση απλών εργασιών, όπως κλείσιμο ετικετών HTML.

Για την διεκπεραίωση της παρούσας εργασίας, χρησιμοποίησα τον Notepad++ που είναι πολύ απλός και εύχρηστος.

1.3.3 Μια ποικιλία προγραμμάτων περιήγησης

Μια από τις μεγαλύτερες προκλήσεις για τους σχεδιαστές ιστοσελίδων είναι ότι οι ιστότοποι μπορεί να φαίνονται και να συμπεριφέρονται διαφορετικά ανάλογα με το πρόγραμμα περιήγησης. Για το λόγο αυτό, είναι σημαντικό να δοκιμάζουμε τα σχέδιά μας νωρίς και συχνά στο ευρύτερο δυνατό φάσμα προγραμμάτων περιήγησης.

Μερικά πρόγραμμα περιήγησης που χρησιμοποιούνται για δοκιμές είναι τα εξής:

- Chrome ([google.com/chrome](https://www.google.com/chrome))
- Firefox (<https://www.mozilla.org/el/>)
- MS Edge (<https://www.microsoft.com/en-us/edge>, *Windows only*)
- Internet Explorer 9–11 (www.microsoft.com, search “Internet Explorer”, *Windows only*)
- Safari (support.apple.com/downloads/#safari, *Mac only*)
- Opera ([opera.com](https://www.opera.com))

Θα χρειαστεί επίσης να κάνουμε δοκιμή σε μια ποικιλία προγραμμάτων περιήγησης smartphone, συμπεριλαμβανομένου του iOS, Safari, Android και άλλα.

1.3.4 Διαχείριση Αρχείων και εργαλεία μεταφοράς

Ο σχεδιασμός και η ανάπτυξη ιστοσελίδων περιλαμβάνει πολλά μεταφερόμενα αρχεία, ιδιαίτερα από τον υπολογιστή στον οποίο εκτελούμε την εργασία μας στον

υπολογιστή του διακομιστή που φιλοξενεί τον ιστότοπο. Για τη μεταφορά αρχείων στο Internet, χρησιμοποιείται ένα FTP (Πρωτόκολλο Μεταφοράς Αρχείων - [FileTransfer Protocol](#)). Πολλές υπηρεσίες φιλοξενίας προσφέρουν τα δικά τους εργαλεία FTP για την αποστολή των αρχείων στους διακομιστές τους. Πολλά από τα προγράμματα επεξεργασίας κώδικα περιλαμβάνουν επίσης ενσωματωμένη λειτουργικότητα FTP.

Στην διεκπεραίωση της παρούσας εργασίας χρησιμοποίησα το WinSCP για να μεταφέρω τα αρχεία στον server της σχολής.

Μερικά άλλα αυτόνομα πρόγραμμα FTP είναι τα εξής :

- Filezilla (filezilla-project.org)
- Cyberduck (cyberduck.io, Mac και Windows)
- Transmit (panic.com/transmit/, Mac)

ΚΕΦΑΛΑΙΟ 2: ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ ΤΟ ΔΙΑΔΙΚΤΥΟ

2.1 Το Internet έναντι του Web

Είναι σημαντικό να επισημάνω τη διάκριση μεταξύ δύο λέξεων που χρησιμοποιούνται όλο και περισσότερο εναλλακτικά. Το διαδίκτυο (**internet**) είναι ένα διεθνές δίκτυο συνδεδεμένων υπολογιστών. Καμία εταιρεία δεν κατέχει το διαδίκτυο· είναι μια συλλογική προσπάθεια που διέπεται από πρότυπα και κανόνες. Σκοπός της σύνδεσης υπολογιστών μεταξύ τους, φυσικά, είναι η ανταλλαγή πληροφορίας. Υπάρχουν πολλοί τρόποι για να μεταφέρεται πληροφορία μεταξύ υπολογιστών, συμπεριλαμβανομένου του ηλεκτρονικού ταχυδρομείου (POP3/IMAP/SMTP), μεταφορά αρχείων (FTP), secure shell (SSH) και πολλές άλλες εξειδικευμένες λειτουργίες πάνω στις οποίες έχει δημιουργηθεί το διαδίκτυο. Αυτές οι τυποποιημένες μέθοδοι για τη μεταφορά δεδομένων ή εγγράφων πάνω από ένα δίκτυο είναι γνωστά ως πρωτόκολλα (**protocols**).

Ο Παγκόσμιος Ιστός (**Web**) (αρχικά ονομαζόταν World Wide Web, και έτσι το "www" στην τοποθεσία διευθύνσεων) είναι ένας μόνο από τους τρόπους με τους οποίους μπορούν να κοινοποιηθούν πληροφορίες μέσω του διαδικτύου. Είναι μοναδικός στο ότι επιτρέπει τη σύνδεση εγγράφων μεταξύ τους μέσω συνδέσμους υπερκειμένου (**hypertext** links) δημιουργώντας έτσι ένα τεράστιο "δίκτυο" συνδεδεμένων πληροφοριών. Το Web χρησιμοποιεί ένα πρωτόκολλο που ονομάζεται HTTP (HyperText Transfer Protocol). Το ακρωνύμιο HTTP είναι τα πρώτα τέσσερα γράμματα σχεδόν όλων των διευθύνσεων ιστοσελίδων. Το Web είναι υποσύνολο του διαδικτύου. Είναι μόνο ένα από τους τρόπους που μπορεί να μεταφερθεί πληροφορία μεταξύ δικτυωμένων υπολογιστών.

2.1.1 Μια μικρή ιστορία του Παγκόσμιου Ιστού

Το διαδίκτυο γεννήθηκε σε ένα εργαστήριο μοριακής φυσικής (CERN) στη Γενεύη της Ελβετίας, το 1989. Εκεί ένας ειδικός υπολογιστών, ο Τιμ Μπέρνερς-Λι, πρότεινε για πρώτη φορά ένα σύστημα διαχείρισης πληροφοριών που χρησιμοποίησε μια διαδικασία "υπερκειμένου" για τη σύνδεση σχετικών εγγράφων μέσω δικτύου. Αυτός και ο συνεργάτης του, Ρομπέρ Καγιό, δημιούργησαν ένα πρωτότυπο και

αφέθηκαν ελεύθεροι για αναθεώρηση. Για τα πρώτα χρόνια, οι ιστοσελίδες ήταν μόνο κείμενο. Είναι δύσκολο να πιστέψει κανείς ότι το 1992, ο κόσμος είχε μόνο περίπου 50 διαδικτυακούς διακομιστές, συνολικά. Η πραγματική αύξηση τη δημοσιότητας του Διαδικτύου έλαβε χώρα το 1992, όταν το πρώτο πρόγραμμα περιήγησης με γραφικά (NCSA Mosaic) παρουσιάστηκε και το διαδίκτυο βγήκε από το πεδίο της επιστημονικής έρευνας στα μέσα μαζικής ενημέρωσης. Η συνεχιζόμενη ανάπτυξη των διαδικτυακών τεχνολογιών επιβλέπεται από το World Wide Web Consortium (W3C). Αν θέλει κάποιος να εμβαθύνει στην ιστορία του διαδικτύου, μπορεί να δει τα Ιστορικά Αρχεία του W3C στο <https://www.w3.org/History.html>.

2.2 Εξυπηρέτηση Πληροφορίας

2.2.1 Χρήσιμη Ορολογία

Λογισμικό ανοικτού κώδικα

Το λογισμικό ανοικτού κώδικα αναπτύσσεται ως συλλογική προσπάθεια με την πρόθεση δημιουργίας του πηγαίου κώδικα που διατίθεται σε άλλους προγραμματιστές για χρήση και τροποποίηση. Εφαρμογές ανοικτού κώδικα είναι συνήθως ελεύθερα χρήσης.

Πλευρά εξυπηρετητή και πλευρά πελάτη (Server-Side and Client-Side)

Συχνά στη σχεδίαση Web, θα ακούσουμε αναφορές σε εφαρμογές στην "πλευρά πελάτη" ή στην "πλευρά εξυπηρετητή". Αυτοί οι όροι χρησιμοποιούνται για να υποδείξουν ποιο μηχανήμα κάνει την επεξεργασία. Εφαρμογές στην πλευρά του πελάτη εκτελούνται στο μηχανήμα του χρήστη (αναφέρεται επίσης ως **frontend**), ενώ εφαρμογές και λειτουργίες που βρίσκονται στην πλευρά του εξυπηρετητή χρησιμοποιούν την επεξεργαστική ισχύ του υπολογιστή εξυπηρετητή (**backend**, σύστημα υποστήριξης).

Intranets και Extranets

Όταν σκέφτεται κανείς μια ιστοσελίδα, γενικά υποθέτει ότι είναι προσβάσιμη από οποιονδήποτε περιηγείται στο διαδίκτυο. Ωστόσο, πολλοί οργανισμοί επωφελούνται της φοβερής ανταλλαγής πληροφοριών και την συγκεντρωτική δύναμη των ιστοσελίδων για ανταλλαγή πληροφορίας μόνο στο δικό τους δίκτυο. Αυτά τα ειδικά βασιζόμενα στο web δίκτυα ονομάζονται ενδοδίκτυα (**intranets**). Λειτουργούν όπως οι συνήθεις ιστότοποι, αλλά χρησιμοποιούν ειδικές συσκευές ασφαλείας (τείχη προστασίας (**firewalls**) που αποτρέπουν κάποιον από τον «έξω κόσμο» το να τους δει. Τα ενδοδίκτυα έχουν πολλές χρήσεις, όπως την κοινή χρήση πληροφοριών ανθρώπινων πόρων ή την παροχή πρόσβασης στον κατάλογο βάσεων δεδομένων. Ένα **extranet** είναι σαν ένα intranet, αλλά επιτρέπει την πρόσβαση σε επιλεγμένους χρήστες εκτός του οργανισμού. Για παράδειγμα, μια εταιρεία παρέχει στους πελάτες της κωδικούς πρόσβασης που τους επιτρέπει να ελέγχουν την κατάσταση για τις παραγγελίες τους στην βάση δεδομένων παραγγελιών της εταιρείας. Οι κωδικοί πρόσβασης καθορίζουν ποιο κομμάτι των πληροφοριών της εταιρείας είναι προσβάσιμο.

2.2.2 Υπολογιστές και μηχανισμοί που συνθέτουν το διαδίκτυο

Να περιγράψω λίγο περισσότερο τους υπολογιστές που συνθέτουν το διαδίκτυο. Επειδή τα έγγραφα «σερβίρονται» (εξυπηρετούνται) κατόπιν αιτήματος, αυτοί οι υπολογιστές είναι γνωστοί ως servers (εξυπηρετητές/ διακομιστές). Για την ακρίβεια, ο εξυπηρετητής είναι το λογισμικό (όχι ο ίδιος ο υπολογιστής) που επιτρέπει στον υπολογιστή να επικοινωνεί με άλλους υπολογιστές· ωστόσο, είναι κοινό να χρησιμοποιείται η λέξη "server" για αναφορά και στον υπολογιστή. Ο ρόλος του λογισμικού διακομιστή είναι ότι πρέπει να περιμένει μια αίτηση για πληροφορίες και στη συνέχεια να ανακτήσει και να αποστείλει τις πληροφορίες αυτές το συντομότερο δυνατόν. Δεν υπάρχει τίποτα το ιδιαίτερο στους ίδιους τους υπολογιστές. Οποιοδήποτε μηχάνημα, από Unix υψηλής ισχύος ως ένα ταπεινό προσωπικό υπολογιστή μπορεί να κατέχει λογισμικό διακομιστή, το οποίο τα κάνει όλα πραγματικότητα. Για να γίνει ένας υπολογιστής μέρος του Web, πρέπει να εκτελεί ειδικό λογισμικό διακομιστή Web που του επιτρέπει να διαχειρίζεται

συναλλαγές πρωτοκόλλου μεταφοράς HyperText (HyperText Transfer Protocol) . Οι διακομιστές Web καλούνται επίσης διακομιστές HTTP ([HTTP servers](#)). Υπάρχουν πολλές επιλογές λογισμικού διακομιστών αλλά οι δύο πιο δημοφιλείς είναι το Apache (λογισμικό ανοικτού κώδικα) και το Microsoft Internet Information Services (IIS). Το Apache είναι ελεύθερα διαθέσιμο σε υπολογιστές που βασίζονται σε Unix και είναι προεγκατεστημένο σε Mac που εκτελούν macOS. Υπάρχει επίσης μια έκδοση για τα Windows. Το Microsoft IIS αποτελεί μέρος της οικογένειας λύσεων διακομιστών της Microsoft.

2.2.2.1 Διευθύνσεις IP και Σύστημα Ονομάτων Τομέων

Σε κάθε υπολογιστή και συσκευή (δρομολογητή, smartphone, αυτοκίνητο κ.λπ.) που είναι συνδεδεμένος στο διαδίκτυο εκχωρείται μια μοναδική αριθμητική διεύθυνση IP («IP» σημαίνει «Internet Protocol»). Για παράδειγμα, ο υπολογιστής που φιλοξενεί το oreilly.com έχει διεύθυνση IP 199.27.145.64. Όλοι αυτοί οι αριθμοί μπορεί να είναι ζάλη, οπότε ευτυχώς, το Σύστημα Ονομάτων Τομέων ([Domain Name System \(DNS\)](#)) αναπτύχθηκε για να μας επιτρέψει επίσης να αναφερθούμε στον διακομιστή με το όνομα τομέα του, "oreilly.com". Η αριθμητική διεύθυνση IP είναι χρήσιμη για το λογισμικό υπολογιστών, ενώ το όνομα τομέα είναι πιο εύκολο για τους ανθρώπους. Η αντιστοίχιση των ονομάτων τομέα με την αντίστοιχη αριθμητική διεύθυνση IP είναι δουλειά ενός ξεχωριστού διακομιστή DNS. Εάν σκεφτούμε μια διεύθυνση IP ως τηλεφωνικό αριθμό, ο διακομιστής DNS θα είναι το τηλεφωνικό βιβλίο. Είναι δυνατή η ρύθμιση παραμέτρων του διακομιστή Web, ώστε περισσότερα από ένα ονόματα τομέα να αντιστοιχίζονται σε μία διεύθυνση IP, επιτρέποντας σε πολλούς ιστότοπους να μοιράζονται έναν διακομιστή.

2.2.2.2 Προγράμματα Περιήγησης

Ο διακομιστής εκτελεί την εξυπηρέτηση, αλλά τι γίνεται με το άλλο μισό της εξίσωσης; Το λογισμικό που κάνει την αίτηση ονομάζεται πελάτης ([client](#)). Οι

χρήστες χρησιμοποιούν προγράμματα περιήγησης για επιτραπέζιους υπολογιστές, προγράμματα περιήγησης για φορητές συσκευές και άλλες βοηθητικές τεχνολογίες (όπως προγράμματα ανάγνωσης οθόνης) ως πελάτες για πρόσβαση σε έγγραφα στο Web. Ο διακομιστής επιστρέφει τα έγγραφα στο πρόγραμμα περιήγησης (αναφέρεται επίσης ως [user agent](#)) για εμφάνιση. Οι αιτήσεις και οι απαντήσεις διεκπεραιώνονται μέσω του πρωτοκόλλου HTTP, το οποίο ανέφερα νωρίτερα. Αν και μιλάμε για "έγγραφα", το HTTP μπορεί να χρησιμοποιηθεί για μεταφορά εικόνων, ταινιών, αρχείων ήχου, δεδομένων, δεσμών ενεργειών και όλων των άλλων πόρων του Web που συνιστούν συνήθως ιστότοπους και εφαρμογές. Είναι συνήθες να βλέπουμε ένα πρόγραμμα περιήγησης ως παράθυρο σε μια οθόνη υπολογιστή με μια ιστοσελίδα που εμφανίζεται σε αυτό. Αυτά είναι γνωστά ως γραφικά προγράμματα περιήγησης ή προγράμματα περιήγησης επιφάνειας εργασίας και για μεγάλο χρονικό διάστημα, ήταν το μοναδικό μέσω προβολής μέσω Web που κυκλοφορούσε. Τα πιο δημοφιλή προγράμματα περιήγησης είναι το Edge και ο Internet Explorer για Windows, το Chrome, το Firefox και το Safari, και το Opera και το Vivaldi που είναι τα πιο πρόσφατα. Αυτές τις ημέρες, ωστόσο, περισσότερο από το ήμισυ της «κίνησης» στο Παγκόσμιο Ιστό προέρχεται από προγράμματα περιήγησης για κινητές συσκευές, smartphones και tablets, όπως Safari σε iOS, Android και προγράμματα περιήγησης Chrome σε συσκευές Android, Opera Mini, και πολυάριθμες άλλες προεπιλογές και εγκατεστημένα προγράμματα περιήγησης για κινητές συσκευές (Η διεύθυνση en.wikipedia.org/wiki/Mobile_browser περιέχει την πλήρη λίστα). Η πλοήγηση στο Web σε μια οθόνη αφής είναι ο νέος κανόνας. Είναι επίσης σημαντικό να έχουμε υπόψη μας εναλλακτικές διαδικτυακές εμπειρίες. Χρήστες με μειωμένη όραση μπορούν να ακούνε μια ιστοσελίδα μέσω ενός προγράμματος ανάγνωσης οθόνης (ή απλά μπορούν να κάνουν το κείμενό τους πολύ μεγάλο). Οι χρήστες με περιορισμένη ικανότητα μετακίνησης μπορούν να χρησιμοποιούν βοηθητική τεχνολογία όπως joystick ή φωνητικές εντολές για πρόσβαση σε συνδέσμους και για να εισάγουν περιεχόμενο. Οι τοποθεσίες που δημιουργούμε πρέπει να είναι προσβάσιμες και να μπορούν να χρησιμοποιηθούν από όλους τους χρήστες, ανεξάρτητα από την εμπειρία περιήγησης τους. Το διαδίκτυο βρίσκει επίσης το δρόμο του σε έξυπνες τηλεοράσεις και συστήματα παιχνιδιών, όπου οι χρήστες έχουν πρόσβαση στις

ιστοσελίδες με τηλεχειριστήρια ή χειριστήρια κονσόλων παιχνιδιών. Η πραγματικότητα είναι ότι οι σελίδες μπορεί να έχουν διαφορετική εμφάνιση και απόδοση από πρόγραμμα περιήγησης σε πρόγραμμα περιήγησης. Αυτό οφείλεται στην ποικίλη υποστήριξη για τεχνολογίες Web, διαφορετικές δυνατότητες συσκευών και τη δυνατότητα των χρηστών να ορίζουν τις δικές τους προτιμήσεις περιήγησης. Είναι η πιο απαιτητική πτυχή του σχεδιασμού και της ανάπτυξης για το μέσο μας.

2.2.2.3 Μηχανισμοί απόδοσης προγράμματος περιήγησης

Το πρόγραμμα που είναι υπεύθυνο για τη μετατροπή HTML και CSS σε αυτό που βλέπουμε στην οθόνη λέγεται μηχανισμός απόδοσης (**rendering engine**). Η αλλιώς μηχανισμός προγράμματος περιήγησης (**browser engine**) ή μηχανισμός διάταξης (**layout engine**). Τα προγράμματα περιήγησης που χρησιμοποιούμε σε επιτραπέζιους υπολογιστές και φορητές συσκευές δημιουργούνται από μηχανισμός απόδοσης καθώς και άλλον κώδικα που χρησιμοποιείται για τις διεπαφές χρηστών και τη λειτουργικότητα. Διάφορα προγράμματα περιήγησης συχνά μοιράζονται έναν μηχανισμό απόδοσης: για παράδειγμα, ο μηχανισμός Blink παρέχει λειτουργικότητα στο Chrome, το Opera και μια ποικιλία από Android προγράμματα περιήγησης. Ο Πίνακας 1 εμφανίζει τους μηχανισμούς απόδοσης που χρησιμοποιούνται από τα πιο δημοφιλή προγράμματα περιήγησης Web σήμερα.

Browser	Rendering engine
Chrome 28+	Blink (forked from WebKit)
Firefox (all)	Gecko (except Firefox for iOS, which uses WebKit)
Safari and Safari iOS (all)	WebKit
Internet Explorer 4–11	Trident
MS Edge (all)	EdgeHTML (forked from Trident)
Opera 15+	Blink (forked from WebKit)

Πίνακας 1. Μηχανισμοί Απόδοσης (Rendering Engines)

Πηγή: [1]

2.2.2.4 Διευθύνσεις Ιστοσελίδων (URLS)

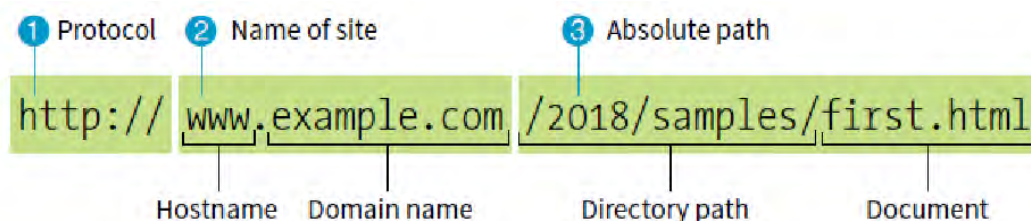
Κάθε ιστοσελίδα και πόρος στο διαδίκτυο έχει τη δική του ειδική διεύθυνση που ονομάζεται URL, που σημαίνει Ενιαίος Εντοπιστής Πόρων ([Uniform Resource Locator](#)). Οι διευθύνσεις Ιστοσελίδων είναι πλήρως ενσωματωμένες στη σύγχρονη γλώσσα.

Διευθύνσεις URL έναντι διευθύνσεων URI

Η W3C και η κοινότητα ανάπτυξης απομακρύνεται από τον όρο URL (Ενιαίος Εντοπιστής Πόρων) και κινείται προς τον πιο γενικό και τεχνικά ακριβές όρο URI (Uniform Resource Identifier - Ενιαίος Αναγνωριστής Πόρων). Είναι πιο συνήθες, ωστόσο, να ακούσουμε για τη διεύθυνση URL. Η διαφορά ανάμεσα στο URL και το URI είναι η εξής: μια διεύθυνση URL είναι ένας τύπος URI που προσδιορίζει τον πόρο από τη θέση του (το L(= Location) στη διεύθυνση URL) στο δίκτυο. Ο άλλος τύπος URI είναι ένας URN (Uniform Resource Name) που προσδιορίζει τον πόρο με βάση το όνομα ή τον χώρο ονομάτων (το N(=Name) στο URN). Οι διευθύνσεις URL είναι υποσύνολο του URI και οι όροι συχνά χρησιμοποιούνται εναλλακτικά. Ορισμένες διευθύνσεις URL είναι σύντομες και ευκολοδιάβαστες. Άλλες μπορεί να μοιάζουν με τρελές χορδές χαρακτήρων διαχωρισμένες με τελείες και καθέτους, αλλά κάθε τμήμα έχει ένα συγκεκριμένο σκοπό

Τα μέρη μίας διεύθυνσης URL

Μια πλήρης διεύθυνση URL αποτελείται γενικά από τρία στοιχεία: το πρωτόκολλο, το όνομα τοποθεσίας και την απόλυτη διαδρομή προς το έγγραφο ή τον πόρο, όπως εμφανίζεται στην Εικόνα 3.



Εικόνα 3. Τα μέρη μιας διεύθυνσης URL

Πηγή: [1]

1 http://

Το πρώτο πράγμα που κάνει η διεύθυνση URL είναι να ορίσει το πρωτόκολλο το οποίο θα χρησιμοποιηθεί για τη συγκεκριμένη μεταφορά. Τα γράμματα "HTTP" ενημερώνουν το διακομιστή για τη χρήση του HyperText Transfer Protocol ή αλλιώς για "να μπει σε λειτουργία Web". Μια διεύθυνση URL μπορεί επίσης να αρχίζει με https://, το οποίο θα εξηγήσω λίγο παρακάτω.

2 www.example.com

Το επόμενο τμήμα της διεύθυνσης URL προσδιορίζει την τοποθεσία Web με το όνομα τομέα της. Σε αυτό το παράδειγμα, το όνομα τομέα είναι το "example.com." Το τμήμα "www." στην αρχή της διεύθυνσης είναι το συγκεκριμένο όνομα του κεντρικού υπολογιστή σε αυτόν τον τομέα. Το όνομα του υπολογιστή φιλοξενίας, "www" έχει γίνει σύμβαση, αλλά δεν είναι κανόνας. Στην πραγματικότητα, μερικές φορές το όνομα του κεντρικού υπολογιστή μπορεί να παραλειφθεί. Επίσης, μπορεί να υπάρχουν περισσότερες από μία ιστοσελίδες σε έναν τομέα (ονομάζονται υποτομείς). Για παράδειγμα, μπορεί να υπάρξει και "development.example.com, "clients.example.com" κ.ο.κ.

3 /2018/samples/first.html

Αυτή είναι η απόλυτη διαδρομή μέσω καταλόγων στο διακομιστή προς το ζητούμενο έγγραφο HTML, "first.html". Οι λέξεις που διαχωρίζονται με καθέτους είναι τα ονόματα καταλόγων, ξεκινώντας με το ριζικό κατάλογο (root directory) του κεντρικού υπολογιστή (όπως υποδεικνύεται από το αρχικό /). Επειδή το διαδίκτυο περιελάμβανε αρχικά υπολογιστές που λειτουργούσαν το λειτουργικό σύστημα Unix, ο τρέχων τρόπος ακολουθίας ενεργειών ακολουθεί κανόνες και συμβάσεις Unix, εξ ου και τα «/» ως διαχωρισμός ονομάτων καταλόγων.

Για να συνοψιστεί η διεύθυνση URL στην Εικόνα 3 : Θέλω να χρησιμοποιήσω το πρωτόκολλο HTTP για σύνδεση με διακομιστή Web, στο διαδίκτυο, που ονομάζεται

"www.example.com" και να ζητήσω το έγγραφο "first.html", που βρίσκεται στον κατάλογο "samples", ο οποίος βρίσκεται στον κατάλογο "2018".

Απλοποιημένα URLs

Προφανώς , δεν είναι όλες οι διευθύνσεις URLs με τέτοιο μήκος. Για να μπω στον ιστότοπο της σχολής ΗΜΜΥ, φτάνει να πληκτρολογήσω "e-ce.uth.gr" αντί για το "https://www.e-ce.uth.gr/".

Παραλείποντας το πρωτόκολλο

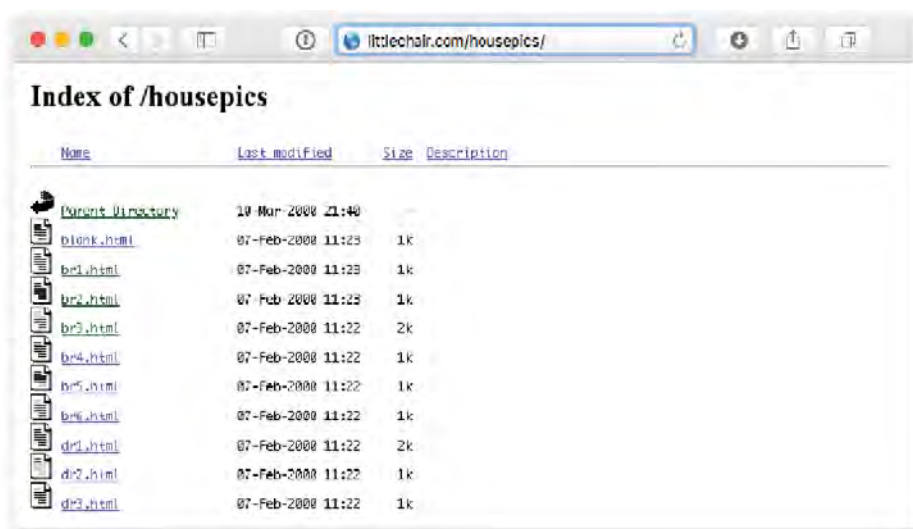
Επειδή σχεδόν όλες οι ιστοσελίδες χρησιμοποιούν το HyperText Transfer Protocol, το κομμάτι http:// συχνά υπονοείται. Αυτό συμβαίνει όταν τα ονόματα ιστότοπων διαφημίζονται στον τύπο ή στην τηλεόραση, για να μπορούμε να θυμόμαστε εύκολα τη διεύθυνση URL. Επιπλέον, τα προγράμματα περιήγησης έχουν προγραμματιστεί για αυτόματη προσθήκη του http:// ως ευκολία για να παραλείψουμε ορισμένα πατήματα πλήκτρων. Μπορεί να φαίνεται σαν να το μην το συμπεριλαμβάνουμε, αλλά αποστέλλεται στο διακομιστή στο παρασκήνιο.

Υποδεικνύοντας σε προεπιλεγμένα αρχεία

Πολλές διευθύνσεις δεν περιλαμβάνουν κάποιο όνομα αρχείου, αλλά απλά παραπέμπουν σε κατάλογο . Όπως: <http://inf-server.inf.uth.gr/~digogos> . Όταν ένας διακομιστής λαμβάνει ένα αίτημα για ένα όνομα καταλόγου αντί για ένα συγκεκριμένο αρχείο, αναζητά σε αυτόν τον κατάλογο ένα προεπιλεγμένο έγγραφο, το οποίο συνήθως ονομάζεται "index.html" (ευρετήριο). Έτσι, όταν κάποιος πληκτρολογεί την προηγούμενη διεύθυνση URL στο πρόγραμμα περιήγησης του, θα δει αυτό: <http://inf-server.inf.uth.gr/~digogos/index.html>. Το όνομα του προεπιλεγμένου αρχείου (το οποίο αναφέρεται επίσης ως αρχείο ευρετηρίου) ενδέχεται να ποικίλει και εξαρτάται από τον τρόπο ρύθμισης παραμέτρων του διακομιστή. Σε αυτό το παράδειγμα, ονομάζεται "index.html", αλλά ορισμένοι διακομιστές χρησιμοποιούν το όνομα αρχείου "default.html". Εάν ο ιστότοπος χρησιμοποιεί προγραμματισμό πλευρά διακομιστή για τη δημιουργία ιστοσελίδων, το αρχείο ευρετηρίου μπορεί να ονομαστεί "index.php" ή "default.aspx". Κάτι άλλο που πρέπει να παρατηρήσουμε είναι ότι στο πρώτο παράδειγμα, η αρχική

διεύθυνση URL δεν να έχει μια τελική κάθετο που να δείχνει ότι ήταν ένας κατάλογος. Εάν η κάθετος παραλειφθεί, ο διακομιστής ελέγχει αν η αίτηση είναι αρχείο ή κατάλογος. Αν είναι κατάλογος, ο διακομιστής ζητά από το πρόγραμμα περιήγησης να στείλει την αίτηση ξανά με κάθετο. Στο τέλος, η κάθετος περιλαμβάνεται στους καταλόγους, ακόμη και αν δεν περιλαμβάνεται την πρώτη φορά που έχει εισαχθεί.

Το αρχείο ευρετηρίου είναι επίσης χρήσιμο για την ασφάλεια. Ορισμένοι διακομιστές (ανάλογα με την ρύθμιση παραμέτρων) εμφανίζουν τα περιεχόμενα του καταλόγου αν το προεπιλεγμένο αρχείο δεν βρεθεί. Η Εικόνα 4 δείχνει πώς εκτίθενται τα έγγραφα στον κατάλογο "housepics" λόγω έλλειψης προεπιλεγμένου αρχείου. Ένας τρόπος να προλαμβάνουμε την αναζήτηση στα αρχεία μας είναι να βεβαιωθούμε ότι υπάρχει ένα αρχείο ευρετηρίου σε κάθε κατάλογο. Ο διαχειριστής του διακομιστή μπορεί επίσης να προσθέσει άλλες προφυλάξεις για να αποτρέψει το να εμφανίζονται οι καταλόγοι μας στο πρόγραμμα περιήγησης.



Εικόνα 4. Ορισμένοι διακομιστές εμφανίζουν τα περιεχόμενα του καταλόγου αν το προεπιλεγμένο αρχείο δεν βρεθεί.

Πηγή: [1]

2.2.2.4.1 HTTPS, το ασφαλές πρωτόκολλο Web

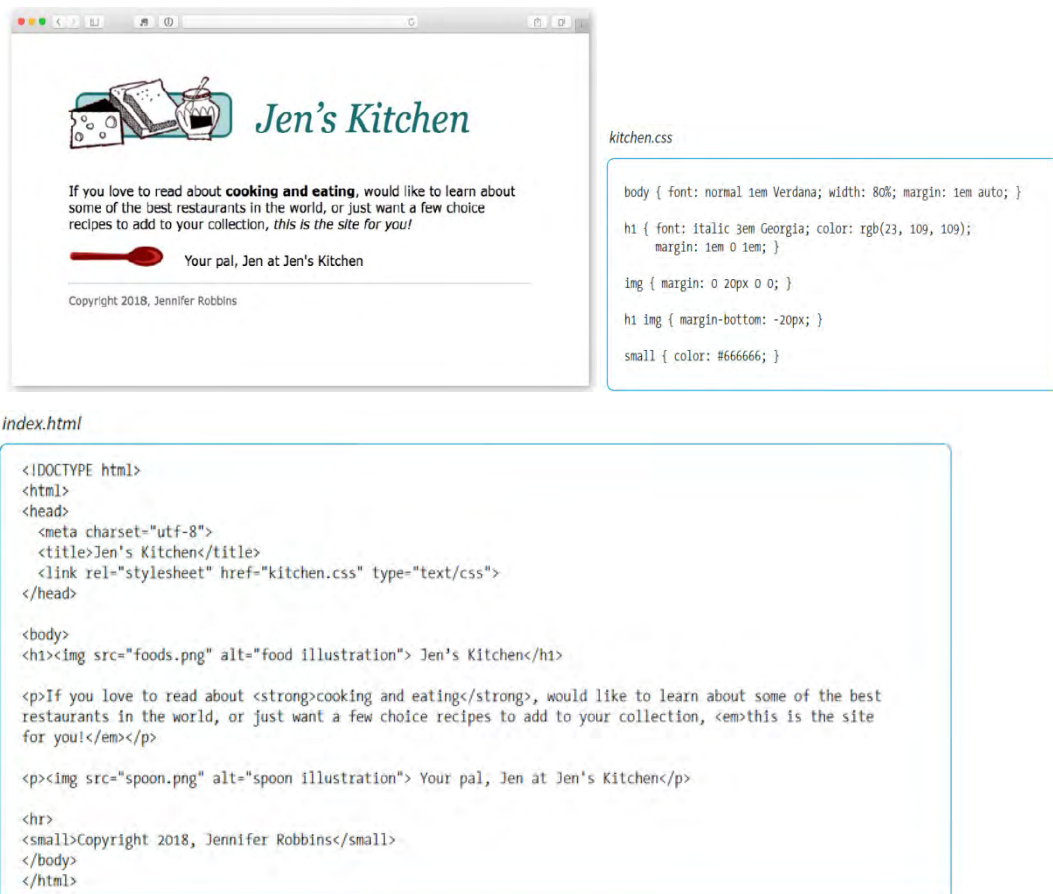
Αν κοιτάξουμε τη γραμμή διευθύνσεων σε ηλεκτρονικές αγορές ή στη χρήση ιστότοπων τραπεζικών υπηρεσιών ή ακόμα και στην διεύθυνση της ιστοσελίδας του τμήματος ΗΜΜΥ, θα παρατηρήσουμε ότι χρησιμοποιούν πρωτόκολλο HTTPS. HTTPS, όπου το "S" συμβολίζει το "secure"(ασφαλές), είναι μια τροποποίηση του HTTP που κρυπτογραφεί πληροφορίες φόρμας όταν αποστέλλονται μεταξύ του χρήστη του πελάτη και τον διακομιστή. Οποιαδήποτε ιστοσελίδα που περιέχει πεδία φόρμας που δέχονται κείμενο (όπως μια μπάρα αναζήτησης ή μια σύνδεση (login) πρέπει να χρησιμοποιεί HTTPS. Περίπου το 60% των ιστοσελίδων χρησιμοποιούν HTTPS και για καλό λόγο. Όχι μόνο είναι καλή ιδέα για την διατήρηση της ασφάλειας των δεδομένων του χρήστη στη μεταφορά, αλλά η Google προχωρά στη μετάβαση στο HTTPS με επίσης ορισμένα σοβαρά κίνητρα. Εάν έχουμε μια τοποθεσία που δέχεται εισαγωγή κειμένου και δεν χρησιμοποιήσουμε το HTTPS, η τοποθεσία μας δεν θα βρίσκεται στα πρώτα σχετικά αποτελέσματα αναζήτησης της Google. Επιπλέον, στο Chrome, αυτές οι ιστοσελίδες λαμβάνουν την σήμανση "Μη ασφαλές" στο επάνω μέρος του προγράμματος περιήγησης. Το HTTPS συνεργάζεται με ένα άλλο πρωτόκολλο, το SSL (Secure Socket Layer, Επίπεδο Ασφαλής Υποδοχής), το οποίο πρέπει να ενεργοποιηθεί στο διακομιστή για ασφαλείς συναλλαγές. Οι εταιρείες υποδοχής έχουν επιλογές για την ενεργοποίηση SSL, συχνά δωρεάν. Πρέπει να λάβουμε υπόψιν ότι το HTTPS προστατεύει τα δεδομένα φόρμας κατά την αποστολή τους στο διακομιστή, αλλά δεν κάνει τίποτα για να κάνει τοποθεσία "ασφαλή από εισβολείς".

2.3 Η ανατομία μιας ιστοσελίδας

Είμαστε όλοι εξοικειωμένοι με το πώς μοιάζουν οι ιστοσελίδες στο παράθυρο του προγράμματος περιήγησης, αλλά τι συμβαίνει "κάτω από το καπό"; Στην Εικόνα 5, βλέπουμε μια μικρή ιστοσελίδα όπως εμφανίζεται σε ένα γραφικό πρόγραμμα περιήγησης. Αν και το βλέπουμε ως μία συνεκτική σελίδα, στην πραγματικότητα είναι συναρμολόγηση τεσσάρων ξεχωριστών αρχείων: ένα έγγραφο HTML (index.html), ένα φύλλο στυλ (kitchen.css) και δύο γραφικά-εικόνες (food.png και spoon.png). Το έγγραφο HTML εκτελεί την προβολή.

2.3.1 Έγγραφο HTML

Η προσθήκη περιγραφικών ετικετών (tags) σε ένα έγγραφο κειμένου είναι γνωστή ως "σήμανση" του εγγράφου. Όπως ανέφερα στην υποενότητα 1.2.1.1 οι ιστοσελίδες χρησιμοποιούν μια γλώσσα σήμανσης που ονομάζεται **HyperText Markup Language (Γλώσσα Σήμανσης Υπερκειμένου)** ή HTML, η οποία δημιουργήθηκε ειδικά για έγγραφα με συνδέσεις υπερκειμένου. Η HTML ορίζει δεκάδες στοιχεία κειμένου που συνιστούν τα έγγραφα όπως επικεφαλίδες, παράγραφοι, κείμενο με έμφαση και, φυσικά, συνδέσμους. Υπάρχουν επίσης στοιχεία που προσθέτουν πληροφορίες σχετικά με το έγγραφο (όπως ο τίτλος του), μέσα όπως εικόνες και βίντεο και γραφικά συστατικά για φόρμες εισόδου, για να αναφέρω μερικά. Η έκδοση HTML που χρησιμοποιούμε σήμερα είναι η HTML5. Υπάρχουν διάφορες εκδόσεις HTML από τη γέννησή της το 1989, και ορισμένες εξακολουθούν να χρησιμοποιούνται σήμερα.



Εικόνα 5. Το πηγαίο αρχείο και το φύλλο στυλ που αποτελούν μια απλή ιστοσελίδα.

Πηγή: [1]

Μια μικρή εισαγωγή στη σήμανση HTML

Μπορούμε να διαβάσουμε το έγγραφο HTML στην Εικόνα 6 και να το συγκρίνουμε με το αποτέλεσμα του προγράμματος περιήγησης. Είναι εύκολο να δούμε πώς τα στοιχεία που σημειώνονται ως ετικέτες HTML (HTML tags) στο έγγραφο προέλευσης αντιστοιχούν στα στοιχεία που εμφανίζονται στο παράθυρο του προγράμματος περιήγησης.

Πρώτον, θα παρατηρήσουμε ότι το κείμενο μέσα σε παρενθέσεις (για παράδειγμα, <body> και) δεν εμφανίζεται στην τελική σελίδα. Το πρόγραμμα περιήγησης εμφανίζει μόνο ό,τι είναι ανάμεσα στα tag — το περιεχόμενο του στοιχείου. Η σήμανση είναι κρυφή. Η ετικέτα παρέχει το όνομα του στοιχείου HTML — συνήθως μια συντομογραφία όπως "h1" για "επικεφαλίδα(header) επιπέδου 1" ή "em" για "κείμενο με έμφαση(emphasized text)".

Δεύτερον, θα δούμε ότι τα περισσότερα HTML tags εμφανίζονται σε ζεύγη γύρω από το περιεχόμενο του στοιχείου. Στο έγγραφό μας HTML, το <h1> δηλώνει ότι το κείμενο που ακολουθεί πρέπει να είναι επικεφαλίδα πρώτου επιπέδου· το </h1> δηλώνει το τέλος της επικεφαλίδας. Ορισμένα στοιχεία, τα οποία ονομάζονται «κενά» στοιχεία, δεν έχουν περιεχόμενο. Στο παράδειγμα, το tag <hr> υποδεικνύει ένα κενό στοιχείο που λέει στο πρόγραμμα περιήγησης να "εισάγει εδώ έναν οριζόντιο κανόνα" ως θεματικό διαχωριστικό. Όταν το πρόγραμμα περιήγησης συναντήσει ανοικτή αγκύλη (<), λαμβάνει υπόψη ότι όλοι οι ακόλουθοι χαρακτήρες αποτελούν μέρος της σήμανσης μέχρι να βρεθεί η δεξιά αγκύλη (>). Παρομοίως, λαμβάνει υπόψιν ότι όλο το περιεχόμενο μετά από το άνοιγμα του tag <h1> είναι μια επικεφαλίδα μέχρι να συναντήσει το κλείσιμο του tag </h1>. Αυτός είναι ο τρόπος με τον οποίο το πρόγραμμα περιήγησης αναλύει το έγγραφο HTML. Η κατανόηση αυτής της μεθόδου του προγράμματος περιήγησης είναι χρήσιμη κατά την αντιμετώπιση προβλημάτων εγγράφων HTML.

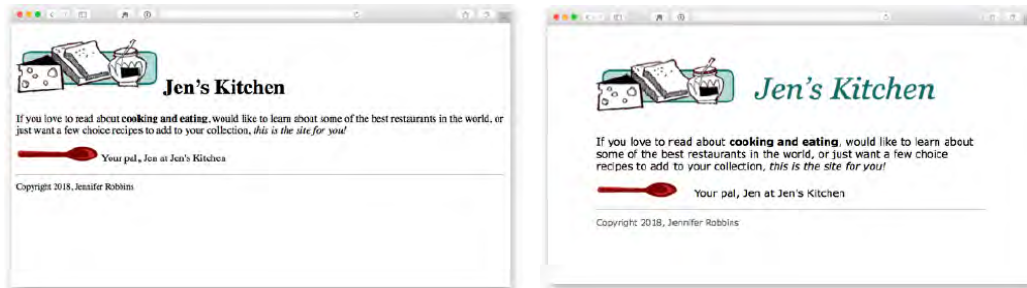
2.3.1.1 Οι εικόνες

Προφανώς, δεν υπάρχουν εικόνες στο ίδιο το αρχείο HTML, οπότε πώς μπορούμε να τις βλέπουμε στην τελική σελίδα; Κάθε εικόνα είναι ένα ξεχωριστό αρχείο. Οι

εικόνες τοποθετούνται στη ροή του κειμένου με το στοιχείο εικόνας HTML (`img`), το οποίο ενημερώνει το πρόγραμμα περιήγησης που θα βρει το γραφικό (με τη διεύθυνση URL του). Όταν το πρόγραμμα περιήγησης δει το στοιχείο `img`, κάνει μια άλλη αίτηση στο διακομιστή για το αρχείο εικόνας και, στη συνέχεια, το τοποθετεί στη ροή περιεχομένου. Το πρόγραμμα περιήγησης στέλνει επίσης αιτήσεις στο διακομιστή για φύλλα στυλ (CSS) (όπως το `kitchen.css`), αρχεία JavaScript (`.js`) και άλλα ενσωματωμένα μέσα, όπως ήχο και βίντεο. Το λογισμικό του προγράμματος περιήγησης (ή, πιο συγκεκριμένα, η μηχανή απόδοσης) συνενώνει τα ξεχωριστά κομμάτια μαζί στην τελική σελίδα. Η συναρμολόγηση της σελίδας γενικά γίνεται σε μια στιγμή, οπότε φαίνεται σαν ολόκληρη η σελίδα να φορτώνεται ταυτόχρονα. Για τις αργές συνδέσεις ή αν η σελίδα περιλαμβάνει τεράστια γραφικά ή αρχεία πολυμέσων, η διαδικασία συγκρότησης μπορεί να είναι πιο εμφανής, καθώς οι εικόνες φορτώνονται πιο αργά από το κείμενο.

2.3.2 Προσθέτοντας λίγο Στυλ (CSS)

Υπάρχει ένα τελευταίο βασικό συστατικό αυτής της μικρής ιστοσελίδας. Κοντά στο επάνω μέρος του εγγράφου HTML υπάρχει ένα στοιχείο σύνδεσης στο οποίο παραπέμπει στο αρχείο CSS, `kitchen.css`. Αυτό το φύλλο στυλ περιλαμβάνει μερικές γραμμές οδηγιών για τον τρόπο εμφάνισης της σελίδας στο πρόγραμμα περιήγησης. Αυτές είναι οδηγίες στυλ που έχουν συνταχθεί σύμφωνα με τους κανόνες των επικαλυπτόμενων φύλλων στυλ (CSS). Το CSS επιτρέπει στους σχεδιαστές να προσθέτουν οδηγίες οπτικού στυλ (γνωστές ως παρουσίαση του εγγράφου) στο κείμενο με σήμανση (δομή του εγγράφου, στην ορολογία σχεδίασης). Η Εικόνα 6 δείχνει την ιστοσελίδα χωρίς CSS (αριστερά) και με CSS (δεξιά). Τα προγράμματα περιήγησης διαθέτουν προεπιλεγμένα στυλ για κάθε στοιχείο HTML που υποστηρίζουν, επομένως, εάν ένα έγγραφο HTML δεν διαθέτει προσαρμοσμένο CSS, το πρόγραμμα περιήγησης θα χρησιμοποιήσει το δικό του. Αυτό φαίνεται στην Εικόνα 6 στα αριστερά. Ακόμη και λίγοι κανόνες στυλ μπορούν να επιφέρουν σημαντικές βελτιώσεις στην εμφάνιση μιας σελίδας.



Εικόνα 6. Η μικρή ιστοσελίδα χωρίς CSS (αριστερά) και με CSS (δεξιά)

Πηγή: [1]

2.3.3 Προσθέτοντας συμπεριφορές με JavaScript

Για να κάνουμε τα στοιχεία στην ιστοσελίδα να κάνουν κάτι, χρησιμοποιούμε μια γλώσσα δέσμης ενεργειών που ονομάζεται JavaScript. Ενώ η HTML παρέχει τη δομή και το CSS αλλάζει τον τρόπο με τον οποίο τα πράγματα φαίνονται, η JavaScript προσθέτει ένα στοιχείο συμπεριφοράς που ελέγχει τον τρόπο με τον οποίο λειτουργούν τα πράγματα. Οι δέσμες ενεργειών μπορεί να είναι αυτόνομα αρχεία στο διακομιστή (με το επίθημα .js) ή γραμμένα μέσα στο έγγραφο HTML. Μπορεί να ενεργοποιηθούν για άμεση λειτουργία όταν η σελίδα φορτώνεται ή να ενεργοποιούνται από κάτι που κάνει ο χρήστης, όπως ένα κλικ ή όταν περνάει το δείκτη του ποντικιού πάνω από ένα στοιχείο ή όταν εισάγει κάτι σε ένα πεδίο φόρμας. Για την διεκπεραίωση της παρούσας εργασίας και το χτίσιμο της σελίδας του εργαστηρίου ERL δεν πρόβηκα καθόλου στη συγγραφή Javascript. Μόνο HTML και CSS. Επίσης χρησιμοποίησα Bootstrap, το οποίο είναι ο πιο δημοφιλής πόρος έτοιμων πλαισίων CSS για την ανάπτυξη ιστοσελίδων με ανταπόκριση (responsive websites) για σύνδεση και προβολή των ιστοσελίδων μέσω κινητών συσκευών. Το Bootstrap χρησιμοποιεί επίσης έτοιμους πόρους Javascript τους οποίους μπορούμε να εισάγουμε (plugin) στην ιστοσελίδα μας.

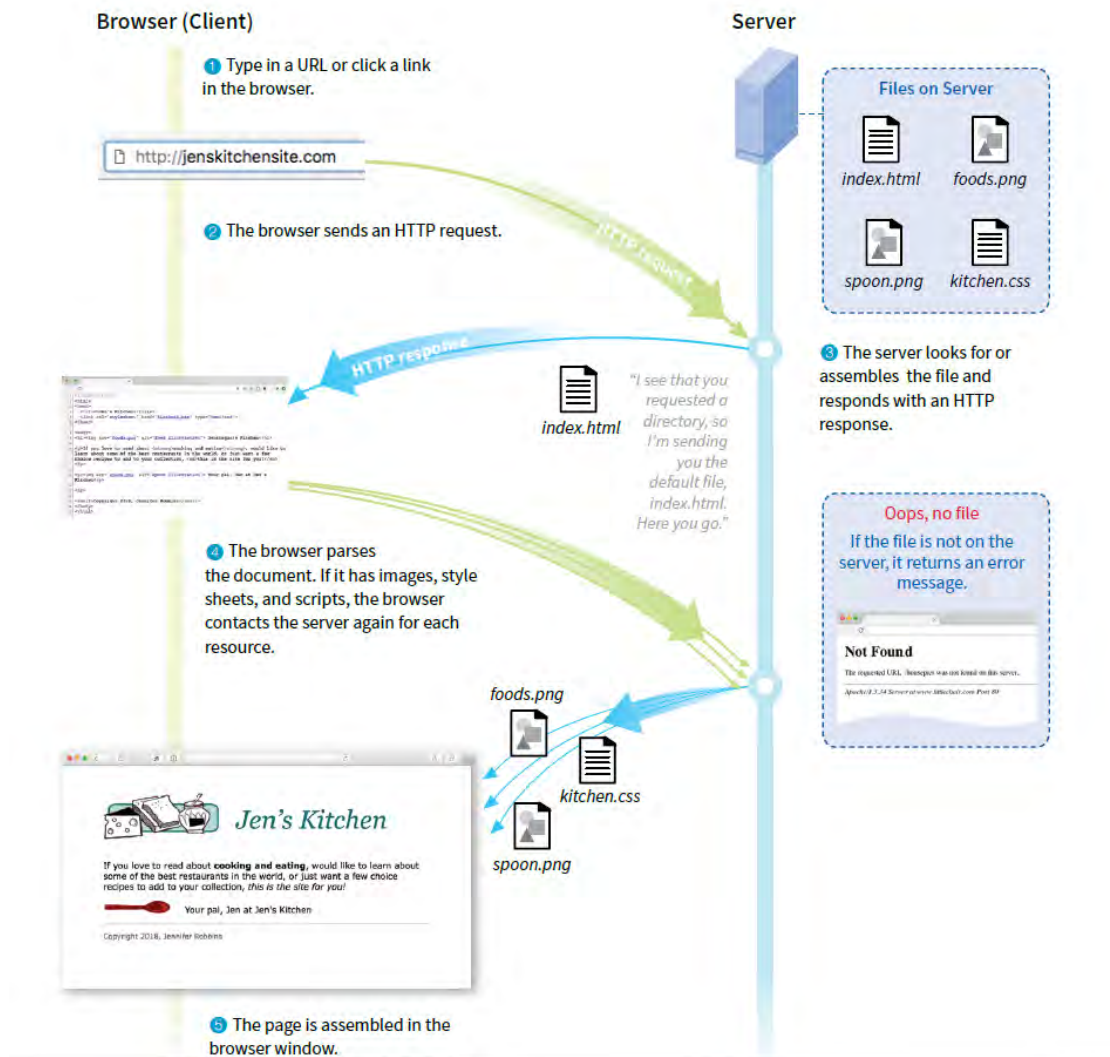
2.3.4 Συνθέτοντας τα όλα μαζί

Για να συνοψίσουμε την εισαγωγή στον τρόπο λειτουργίας του διαδικτύου, ας καταγράψουμε μία τυπική ροή συμβάντων που συμβαίνει σε κάθε ιστοσελίδα που εμφανίζεται στην οθόνη (Εικόνα 7).

1. Κάνουμε αίτηση για μια ιστοσελίδα είτε πληκτρολογώντας τη διεύθυνση URL της (για παράδειγμα, <http://jenskitchensite.com>) απευθείας στο πρόγραμμα περιήγησης ή κάνοντας κλικ σε μια σύνδεση σε μια ιστοσελίδα. Η διεύθυνση URL περιέχει τις πληροφορίες που απαιτούνται για τον προορισμό ενός συγκεκριμένου εγγράφου σε έναν συγκεκριμένο διακομιστή Web στο Internet. Στην περίπτωση αυτή, παραπέμπει στο προεπιλεγμένο αρχείο (index.html) στον αρχικό κατάλογο.
2. Το πρόγραμμα περιήγησης που χρησιμοποιούμε στέλνει ένα αίτημα HTTP στο διακομιστή που καθορίζεται στο URL και ζητά το συγκεκριμένο αρχείο. Η αίτηση περιλαμβάνει επίσης πληροφορίες σχετικά με τις γλώσσες που μπορεί να διαβάσει ο χρήστης και τους τύπους αρχείων που το πρόγραμμα περιήγησης του μπορεί να δεχτεί. Εάν η διεύθυνση URL καθορίζει έναν κατάλογο (όχι ένα αρχείο), είναι το ίδιο με το να κάνουμε αίτηση του προεπιλεγμένου αρχείου σε αυτόν τον κατάλογο
3. Ο διακομιστής αναζητά το ζητούμενο αρχείο και εκδίδει μια απόκριση HTTP με τη μορφή μιας κεφαλίδας HTTP([HTTP header](#)) Η κεφαλίδα περιλαμβάνει πληροφορίες σχετικά με το αρχείο, όπως η ημερομηνία τελευταίας τροποποίησης, το μήκος του αρχείου και ο τύπος περιεχομένου του (για παράδειγμα, ένα αρχείο .html έχει τον τύπο περιεχομένου "text/html").
 - α. Εάν δεν είναι δυνατή η εύρεση της σελίδας, ο διακομιστής επιστρέφει ένα μήνυμα σφάλματος. Το μήνυμα συνήθως λέει "404 δεν βρέθηκε(404 Not Found)", αν και ενδέχεται να παρέχονται πιο φιλόξενα μηνύματα σφάλματος. Επίσης είναι δυνατοί και άλλοι τύποι σφαλμάτων (γράψω λίγο παρακάτω για αυτό στην υποενότητα "Κωδικοί κατάστασης HTTP").
 - β. Εάν βρεθεί το έγγραφο, ο διακομιστής ανακτά το ζητούμενο αρχείο και το «απαντά» στο πρόγραμμα περιήγησης. Εάν η τοποθεσία είναι δυναμική, ο διακομιστής συγκροτεί τη σελίδα από τα αποθηκευμένα δεδομένα πριν την απάντηση τους στο πρόγραμμα περιήγησης.
4. Το πρόγραμμα περιήγησης αναλύει το έγγραφο HTML. Εάν η σελίδα περιέχει εικόνες (υποδεικνύεται από το στοιχείο HTML `img`) ή άλλους εξωτερικούς πόρους, όπως δέσμες ενεργειών ή φύλλα στυλ, το πρόγραμμα περιήγησης

επικοινωνεί ξανά με το διακομιστή για να ζητήσει κάθε πόρο που καθορίζεται στη σήμανση.

5. Το πρόγραμμα περιήγησης εισάγει κάθε εικόνα στη ροή του εγγράφου όπου υποδεικνύεται από το στοιχείο `img`, εφαρμόζει στυλ και εκτελεί δέσμες ενεργειών. Και έτσι η συναρμολογημένη ιστοσελίδα εμφανίζεται για τη δική μας οπτική απόλαυση.



Εικόνα 7. Πώς τα προγράμματα περιήγησης εμφανίζουν ιστοσελίδες.

Πηγή: [1]

Θα πρέπει να σημειώσω ότι απεικονίστηκε εδώ ένα παραδοσιακό και απλοποιημένο σενάριο για να δείξω το πώς δημιουργούνται οι ιστοσελίδες. Αυτές τις μέρες, είναι συνηθισμένο οι ιστοσελίδες να δημιουργηθούν από συστήματα διαχείρισης περιεχομένου (content management systems CMS) που διατηρούν

περιεχόμενο σε βάσεις δεδομένων και χρησιμοποιούν πρότυπα (templates) για τη συναρμολόγηση των δεδομένων σε σελίδες πάρα πολύ γρήγορα. Στην περίπτωση αυτή, στο βήμα 3β, υπάρχει μια πιο περίπλοκη διαδικασία συναρμολόγησης του αρχείου από διάφορα μέρη αντί για την παράδοση ενός υπάρχοντος αρχείου. Δεν θα αναφερθώ όμως άλλο σε αυτό.

2.4 Στατικοί έναντι Δυναμικών Ιστότοπων

Οι στατικοί ιστότοποι ([static websites](#)) αποτελούνται από αρχεία HTML με σταθερό περιεχόμενο που εμφανίζουν τις ίδιες πληροφορίες για κάθε επισκέπτη. Με άλλα λόγια, κάθε σελίδα που βλέπουμε στο πρόγραμμα περιήγησης είναι προβολή ενός αρχείου HTML στο διακομιστή. Η σελίδα του εργαστηρίου ERL είναι ένας στατικός ιστότοπος. Αντίθετα, οι δυναμικοί ιστότοποι ([dynamic websites](#)) δημιουργούνται με σύστημα υποστήριξης προγραμματισμού όπως PHP ή ASP. Κάθε σελίδα δημιουργείται άμεσα από την εφαρμογή. Δυναμικές τοποθεσίες προσπελάσουν περιεχόμενο και δεδομένα από μία βάση δεδομένων και οι τελικές σελίδες μπορεί να προσαρμόζονται για κάθε χρήστη. Για εξαιρετικά μεγάλους ιστότοπους με εκατοντάδες ή χιλιάδες ιστοσελίδες, η ρύθμιση και διατήρηση ενός δυναμικού ιστότοπου χρειάζεται σημαντικά λιγότερη εργασία από τη δημιουργία και αποθήκευση κάθε ιστοσελίδας ξεχωριστά ως ένα στατικό έγγραφο HTML.

2.5 Κωδικοί κατάστασης HTTP

Οι διακομιστές δίνουν κωδικούς κατάστασης ως απόκριση σε αιτήματα προγραμμάτων περιήγησης. Ο πλήρης κατάλογος κωδικών κατάστασης είναι αρκετά μεγάλος (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>), αλλά εδώ είναι μερικές συχνές απαντήσεις:

- 200 OK
- 301 Moved Permanently (Μεταφέρθηκε Μόνιμα)
- 302 Moved Temporarily (Μεταφέρθηκε Προσωρινά)
- 404 Not Found (Δεν βρέθηκε)
- 410 Gone (no longer available) (Πλέον μη διαθέσιμη)
- 500 Internal Server Error (Εσωτερικό σφάλμα διακομιστή)

ΚΕΦΑΛΑΙΟ 3: ΜΕΡΙΚΕΣ ΣΗΜΑΝΤΙΚΕΣ ΕΝΝΟΙΕΣ

Καθώς το web εξελίσσεται και ο αριθμός των συσκευών από τις οποίες θα αποκτούμε πρόσβαση σε αυτό αυξάνεται εκθετικά, το έργο των σχεδιαστών ιστότοπων και των προγραμματιστών γίνεται αρκετά πιο περίπλοκο. Θέλω να παρουσιάσω μερικές σημαντικές έννοιες που κάθε σχεδιαστής ιστοσελίδων πρέπει να γνωρίζει. Η καρδιά του θέματος είναι ότι ένας σχεδιαστής ιστοσελίδων, ποτέ δεν ξέρει ακριβώς τον τρόπο προβολής των σελίδων που δημιουργεί. Δεν ξέρει ποιος από τα δεκάδες προγράμματα περιήγησης μπορεί να χρησιμοποιούνται, είτε αν αυτά βρίσκονται σε επιτραπέζιο υπολογιστή είτε βρίσκονται σε κάποια κινητή συσκευή, πόσο μεγάλο θα είναι το παράθυρο του προγράμματος περιήγησης, ποιες γραμματοσειρές έχει εγκατεστημένες, πόσο γρήγορα είναι ενεργοποιημένη μια λειτουργία όπως JavaScript, πόσο γρήγορη είναι η σύνδεση στο διαδίκτυο, αν οι σελίδες διαβάζονται από συσκευή ανάγνωσης οθόνης, και ούτω καθεξής. Οι σημαντικές έννοιες στο παρόν κεφάλαιο είναι κυρίως μέθοδοι αντιμετώπισης του αναπόφευκτου στοιχείου του Αγνώστου στο μέσο μας. Θα διατηρήσω τις περιγραφές σύντομες και αρκετά μη-τεχνικές. Ο στόχος μου είναι να υπάρχει μια βασική κατανόηση του τι σημαίνουν όροι όπως "προοδευτική βελτίωση" όταν τους συναντά κανείς σε μαθήματα αργότερα. Έχουν γραφτεί πολλά εξαιρετικά άρθρα και βιβλία για καθένα από τα αυτά τα θέματα και τις σχετικές τεχνικές παραγωγής τους.

3.1 Μια πληθώρα συσκευών

Μέχρι το 2007, θα μπορούσε κάποιος να είναι σχετικά σίγουρος ότι οι χρήστες επισκέπτονταν τους ιστότοπους ενώ κάθονται στα γραφεία τους, κοιτώντας μια μεγάλη οθόνη, χρησιμοποιώντας μια σχετικά γρήγορη σύνδεση στο Internet. Ο κόσμος είχε σχεδόν κατασταλάξει στα 960 pixels ως ένα καλό πλάτος ιστοσελίδας με βάση το πιο κοινό μέγεθος οθόνης. Τότε, η μεγαλύτερη ανησυχία ήταν η αντιμετώπιση των δώδεκα περίπου προγραμμάτων περιήγησης επιφάνειας εργασίας και η υπερπήδηση μερικών επιπλέον εμποδίων για την υποστήριξη ιδιότυπων παλαιών εκδόσεων του Internet Explorer. Αν και μπορούσαμε να έχουμε πρόσβαση σε ιστοσελίδες και περιεχόμενο Web σε κινητά τηλέφωνα πριν

από το 2007, η εισαγωγή των smartphones, iPhone και Android , όπως και ταχύτερων δικτύων προανήγγειλαν μια τεράστια αλλαγή στο πώς, στο πότε, και στο πού θα κάνουμε την περιήγηση στο διαδίκτυο. Από τότε, είδαμε την εισαγωγή πολλών τηλεφώνων και tablet κάθε διαφορετικής διάστασης, καθώς και προγράμματα περιήγησης στο Web σε τηλεοράσεις, συστήματα παιχνιδιών και άλλες συσκευές. Και η ποικιλομορφία απλά θα συνεχίσει να αυξάνει. Ο ειδικός σε θέματα σχεδιασμού ιστοσελίδων για κινητά, Μπραντ Φροστ (Brad Frost - <https://bradfrost.com/>) το συνοψίζει επιτυχώς στην Εικόνα 8.



Εικόνα 8. Ο Μπραντ Φροστ συνοψίζει επιτυχώς την πραγματικότητα της ποικιλομορφίας των συσκευών

Πηγή: [1]

Η πρόκληση του σχεδιασμού για όλες αυτές τις συσκευές ξεπερνά την αντιμετώπιση των διαφορετικών μεγεθών οθόνης. Υπάρχει μεγάλη διαφορά μεταξύ της χρήσης ενός ιστότοπου πάνω από μία γρήγορα ευρυζωνική σύνδεση και μέσω ενός αργού δικτύου κινητών. Οι σχεδιαστές πρέπει να αντισταθούν στην πραγματοποίηση παραδοχών σχετικά με την ταχύτητα δικτύου και το περιβάλλον περιήγησης με βάση το μέγεθος οθόνης. Το ότι είναι μικρή η οθόνη δεν σημαίνει ότι είναι αργή σύνδεση ή ότι το άτομο βιάζεται. Δεν είναι ασυνήθιστο να περιηγείται κανείς εύκολα στο διαδίκτυο από ένα smartphone με μία σταθερή σύνδεση WiFi. Τα iPad με μεγαλύτερες οθόνες υψηλής ανάλυσης μπορούν να έχουν πρόσβαση στο Internet με συνδέσεις 4G και σύντομα με συνδέσεις 5G. Για πολλούς ιστότοπους σήμερα, περισσότεροι άνθρωποι έχουν πρόσβαση σε αυτούς μέσω των κινητών συσκευών τους παρά από έναν επιτραπέζιο υπολογιστή. Ήδη πολλοί

χρησιμοποιούν τα κινητά τους τηλέφωνα ως το μόνο μέσο πρόσβασης στο διαδίκτυο. Αυτό σημαίνει ότι είναι κρίσιμη η ανάλογη σωστή σχεδίαση και λειτουργικότητα. Έχουν γίνει τεράστια βήματα για να παρέχεται μια υπηρεσία ευχάριστης εμπειρίας στους χρήστες συσκευών χειρός και η τεχνολογία που στοχεύει στις ανάγκες τους. Ο τρόπος που βλέπουμε το σχέδιό μας όταν εργαζόμαστε σε έναν επιτραπέζιο υπολογιστή δεν είναι ίδιος για όλους. Κάποιοι θα το δουν πολύ μικρότερο. Μερικοί θα το δουν να φορτώνεται αργά. Κάποιοι μπορεί να το βλέπουν σε μια τηλεόραση. Όλα οι σχεδιαστές ιστοσελίδων πρέπει να τα λάβουν όλα αυτά υπόψη.

Ιστός για κινητές συσκευές (Mobile Web)

Μπορεί να ακούσουμε να χρησιμοποιείται ο όρος "Mobile Web", αλλά η αλήθεια είναι ότι δεν υπάρχει Mobile Web όπως δεν υπάρχει υπάρχει Desktop Web (Ιστός για επιτραπέζιους υπολογιστές) ή Tablet Web, κ.ο.κ. Υπάρχει μόνο ο (Παγκόσμιος) Ιστός, και μπορεί να έχουν πρόσβαση σε αυτόν σχεδόν όλοι από κάθε είδους συσκευή.

3.2 Τήρηση των προτύπων

Μια καλή αρχή για να αντιμετωπίσουμε αυτή την πολυμορφία είναι η τήρηση των προτύπων που τεκμηριώνονται από το World Wide Web Consortium (W3C - w3.org/standards - Κοινοπραξία Παγκόσμιου Ιστού). Το να τηρήσουμε τα πρότυπα Web είναι το κύριο εργαλείο για να διασφαλίσουμε τη συνέπεια του ιστότοπού μας με όλα τα τυπικά – συμβατικά προγράμματα περιήγησης. Μας βοηθά επίσης να καταστήσουμε το περιεχόμενό μας συμβατό με τις τεχνολογίες Web και τις δυνατότητες των προγραμμάτων περιήγησης που συνεχώς εξελίσσονται. Η συμμόρφωση προς τα πρότυπα μπορεί να φαίνεται σαν να μην έχει νόημα, αλλά αυτό που συνέβη στο παρελθόν είναι ότι όλοι, συμπεριλαμβανομένων των κατασκευαστών προγραμμάτων περιήγησης, «έπαιξαν» με ανευθυνότητα με το HTML και το scripting. Το τίμημα ήταν να έχουμε υλοποιήσεις μη συμβατές με τα

πρόγραμματα περιήγησης και την ανάγκη δημιουργίας ισότοπων δύο ή παραπάνω φορών έτσι ώστε να λειτουργούν για όλους.

3.3 Προοδευτική Βελτίωση

Με μια πληθώρα προγραμμάτων περιήγησης έρχεται και μια πληθώρα επιπέδων υποστήριξης για τα πρότυπα Web. Στην πραγματικότητα, κανένα πρόγραμμα περιήγησης δεν έχει εφαρμόσει όλα τα πρότυπα 100%, και υπάρχουν πάντα νέες τεχνολογίες που σιγά σιγά κερδίζουν έδαφος. Επιπλέον, οι χρήστες μπορούν να ορίσουν τις δικές τους προτιμήσεις για το πρόγραμμα περιήγησης, έτσι ώστε να έχουν ένα πρόγραμμα περιήγησης που υποστηρίζει JavaScript αλλά έχουν επιλέξει να το απενεργοποιήσουν. Το θέμα εδώ είναι ότι αντιμετωπίζουμε ένα ευρύ φάσμα δυνατοτήτων του προγράμματος περιήγησης—από μία βασική υποστήριξη HTML έως το κάθε επιπλέον στοιχείο. Η προοδευτική βελτίωση είναι μια στρατηγική για την αντιμετώπιση των άγνωστων δυνατοτήτων του προγράμματος περιήγησης. Όταν σχεδιάζουμε με προοδευτική βελτίωση, ξεκινάμε με μια βασική εμπειρία που καθιστά το περιεχόμενο ή την βασική λειτουργικότητα διαθέσιμη ακόμη και στα πιο στοιχειώδη προγράμματα περιήγησης ή βοηθητικές συσκευές. Από εκεί και πέρα, μπορούμε να χρησιμοποιήσουμε περισσότερες προηγμένες δυνατότητες για τα προγράμματα περιήγησης τις οποίες αυτά μπορούν να χειριστούν. Μπορούμε για παράδειγμα να βελτιώσουμε μια ιστοσελίδα με μερικά εφέ , όπως κινούμενα σχέδια ή αναδίπλωση κειμένου γύρω από εικόνες σε ενδιαφέροντα σχήματα, που βελτιώνουν την εμπειρία για χρήστες με πιο προηγμένα προγράμματα περιήγησης, αλλά δεν είναι πραγματικά κρίσιμα για την επωνυμία ή το μήνυμα της ιστοσελίδας. Η προοδευτική βελτίωση είναι μια προσέγγιση που αφορά όλες τις πτυχές της σχεδίασης και δημιουργίας ιστοσελίδας συμπεριλαμβανομένης του HTML, του CSS, και του JavaScript.

Στρατηγική σύνταξης

Όταν ένα έγγραφο HTML είναι γραμμένο με λογική σειρά και τα στοιχεία του επισημαίνονται με ουσιαστικό τρόπο, μπορεί να χρησιμοποιηθεί από το μεγαλύτερο δυνατό εύρος περιβαλλόντων περιήγησης, συμπεριλαμβανομένων των

παλαιότερων προγραμμάτων περιήγησης, μελλοντικών προγραμμάτων περιήγησης, και κινητών και βοηθητικών συσκευών. Μπορεί να μην φαίνεται ακριβώς το ίδιο αλλά το σημαντικό είναι ότι το περιεχόμενό θα είναι διαθέσιμο. Επίσης διασφαλίζει ότι οι μηχανές αναζήτησης, όπως η Google, θα ταξινομήσουν σωστά το περιεχόμενο. Ένα καθαρό έγγραφο HTML με τα στοιχεία του που περιγράφονται με ακρίβεια είναι η βάση για την προσβασιμότητα.

Στρατηγική Στυλ

Μπορούμε να δημιουργήσουμε διάφορα επίπεδα εμπειρίας απλά εκμεταλλευόμενοι τον τρόπο με τον οποίο τα προγράμματα περιήγησης αναλύουν τους κανόνες φύλλου στυλ (CSS). Μπορούμε π.χ. να γράψουμε έναν κανόνα στυλ που κάνει κόκκινο το φόντο ενός στοιχείου αλλά επίσης να συμπεριλάβουμε ένα στυλ που του παρέχει ένα μοντέρνο ντεγκραντέ (μια σταδιακή, βαθμιαία αλλαγή από ένα χρώμα σε άλλο) για προγράμματα περιήγησης που γνωρίζουν πώς να αποδίδουν ντεγκραντέ. Ή μπορούμε να χρησιμοποιήσουμε έναν προηγμένο επιλογέα CSS για να αποδώσουμε συγκεκριμένα στυλ μόνο σε προηγμένα προγράμματα περιήγησης. Η γνώση ότι τα προγράμματα περιήγησης απλώς αγνοούν τις ιδιότητες και τους κανόνες που δεν καταλαβαίνουν μας δίνει το ελεύθερο για καινοτομία χωρίς να «γονατίζουμε» παλαιότερα προγράμματα περιήγησης. Απλά πρέπει να έχουμε υπόψη μας τον ορισμό στυλ της βασικής εμπειρίας πρώτα και, στη συνέχεια, να προσθέτουμε βελτιώσεις, αφού πληρούνται οι ελάχιστες απαιτήσεις.

3.4 Σχεδίαση Ιστοσελίδων με Ανταπόκριση (Responsive Web Design)

Από προεπιλογή, τα περισσότερα προγράμματα περιήγησης σε μικρές συσκευές, όπως smartphone και tablet συρρικνώνουν μια ιστοσελίδα για προσαρμογή στην οθόνη και παρέχουν μηχανισμούς για ζουμ και κίνηση στη σελίδα. Αν και τεχνικά λειτουργεί, δεν είναι μία εξαιρετική εμπειρία. Το κείμενο είναι πολύ μικρό για ανάγνωση, οι σύνδεσμοι είναι πολύ μικροί για να τους πατήσουμε και όλο αυτό το ζουμ και η κίνηση αποσπά την προσοχή. Η Σχεδίαση Ιστοσελίδων με Ανταπόκριση ([Responsive Web Design](#) (RWD)) είναι μια στρατηγική για την παροχή κατάλληλων

διατάξεων σε συσκευές που βασίζονται στο μέγεθος της προβολής (παράθυρο προγράμματος περιήγησης). Το κλειδί για τη Σχεδίαση Ιστοσελίδων με Ανταπόκριση είναι ένα μοναδικό έγγραφο HTML (με μία διεύθυνση URL) για όλες τις συσκευές, αλλά εφαρμογή διαφορετικού CSS με βάση το μέγεθος της οθόνης για να παρέχεται η βελτιστοποιημένη διαμόρφωση για τη συγκεκριμένη συσκευή. Για παράδειγμα, όταν η σελίδα προβάλλεται σε smartphone, εμφανίζεται σε μία στήλη με μεγάλους συνδέσμους για εύκολο πάτημα. Αλλά όταν αυτή η ίδια σελίδα προβάλλεται σε πρόγραμμα περιήγησης μεγάλου μεγέθους, το περιεχόμενο αναδιατάσσεται σε πολλές στήλες με παραδοσιακά στοιχεία περιήγησης. Η Σχεδίαση Ιστοσελίδων με Ανταπόκριση έχει γίνει ένα από τα κύρια εργαλεία που χρησιμοποιούμε για να αντιμετωπίσουμε το άγνωστο μέγεθος προβολής. Η Εικόνα 9 δείχνει πώς φαίνεται, ένα κομμάτι της αρχικής σελίδας που έχτισα (του εργαστηρίου ERL) με Responsive Web Design, στον υπολογιστή και πώς φαίνεται στο κινητό. Για το κινητό έχω πάρει δύο στιγμιότυπα γιατί η συσκευή δείχνει («χωράει να δείξει») μόνο το ένα στιγμιότυπο. Απλά παρέθεσα και τα δύο στιγμιότυπα για να μπορέσουμε να το συγκρίνουμε με την πάνω εικόνα. Ασφαλώς κάνοντας scroll την ιστοσελίδα προς τα κάτω θα συναντήσουμε και την υπόλοιπη και θα θούμε και το δεύτερο στιγμιότυπο.

Αν ανοίξουμε τον ιστότοπο στο πρόγραμμα περιήγησης και, στη συνέχεια αυξομειώσουμε το μέγεθος του παραθύρου θα δούμε πώς αλλάζει η διάταξη με βάση το μέγεθος του παραθύρου. Η Σχεδίαση Ιστοσελίδων με Ανταπόκριση βοηθάει σε θέματα διάταξης, αλλά δεν είναι λύση σε όλες τις προκλήσεις σχεδίασης ιστοσελίδων για φορητές συσκευές. Είναι γεγονός είναι ότι για την παροχή της καλύτερης εμπειρίας στους χρήστες και την επιλεγμένη συσκευή τους ενδέχεται να απαιτούνται βελτιστοποιήσεις πέρα από την προσαρμογή της εμφάνισης και της αίσθησης. Μπορούμε να αντιμετωπίσουμε καλύτερα ορισμένα προβλήματα χρησιμοποιώντας το διακομιστή για την ανίχνευση της συσκευής και των δυνατοτήτων του και, στη συνέχεια, να πάρουμε αποφάσεις για το τι θα σταλεί πίσω.



Εικόνα 9. Η διάταξη μιας

ιστοσελίδας με ανταπόκριση αλλάζει ανάλογα με το μέγεθος του παραθύρου. Πώς φαίνεται στον υπολογιστή (πάνω) και στο κινητό (κάτω αριστερά και κάτω δεξιά).

Για ορισμένες τοποθεσίες και υπηρεσίες, ίσως είναι προτιμότερο να κατασκευάσουμε έναν ξεχωριστό ιστότοπο για κινητές συσκευές ("Ιστότοπος M-dot" – ονομάστηκαν έτσι επειδή οι διευθύνσεις URLs τους ξεκινούν με το "m." ή το "mobile." π.χ. <https://m.youtube.com>) με προσαρμοσμένο περιβάλλον εργασίας που εκμεταλλεύεται τις δυνατότητες της κινητής συσκευής, όπως την γεωγραφική της θέση (geolocation). Στην διεκπεραίωση της παρούσας εργασίας δεν έκανα κάτι τέτοιο και χρησιμοποίησα απλά Σχεδίαση με Ανταπόκριση.

Αν και η Σχεδίαση με Ανταπόκριση δεν θα διορθώσει τα πάντα, αποτελεί σημαντικό μέρος της λύσης για την παροχή ικανοποιητικής εμπειρίας σε μία μεγάλη ποικιλία προγραμμάτων περιήγησης.

3.5 Προσβασιμότητα

Υπάρχει ένας πολύ μεγάλος αριθμός προγραμμάτων περιήγησης που χρησιμοποιούνται σήμερα, αλλά μέχρι στιγμής, έχω απευθυνθεί μόνο σε οπτικά προγράμματα περιήγησης που ελέγχονται με δείκτες ποντικιού ή ακροδάχτυλα. Είναι σημαντικό, ωστόσο, να έχουμε υπόψη μας ότι οι άνθρωποι έχουν πρόσβαση στο διαδίκτυο με πολλούς διαφορετικούς τρόπους — με πληκτρολόγιο, ποντίκι, φωνητικές εντολές, συσκευές ανάγνωσης οθόνης, παροχή Braille, μεγεθυτικά στοιχεία, joystick, ποδόπληκτρα, κ.ο.κ. Οι σχεδιαστές ιστοσελίδων πρέπει να δημιουργούν σελίδες με τρόπο που να δημιουργεί όσο το δυνατόν λιγότερους φραγμούς στην πρόσβαση σε πληροφορίες, ανεξάρτητα από την ικανότητα του χρήστη και τη συσκευή που χρησιμοποιείται για την πρόσβαση στο διαδίκτυο. Με άλλα λόγια, πρέπει να σχεδιάσουν ανάλογα, με στόχο την προσβασιμότητα. Αν και προορίζονται για χρήστες με αναπηρίες, όπως η περιορισμένη όραση ή κινητικότητα, οι τεχνικές και οι στρατηγικές που αναπτύσσονται για την προσβασιμότητα ωφελούν επίσης άλλους χρήστες που δεν μπορούν να έχουν τη βέλτιστη εμπειρία περιήγησης. Οι προσβάσιμοι ιστότοποι καταχωρούνται επίσης αποτελεσματικότερα στα ευρετήρια μηχανών αναζήτησης όπως η Google. Η πρόσβαση στον ιστότοπο αξίζει την επιπλέον προσπάθεια. Υπάρχουν τέσσερις ευρείες κατηγορίες αναπηριών που επηρεάζουν την αλληλεπίδραση των ατόμων με τους υπολογιστές τους και τις πληροφορίες που τους αφορούν: Προβλήματα όρασης, προβλήματα ακοής, περιορισμένη κινητικότητα και νοητικές δυσλειτουργίες.

Η W3C ξεκίνησε την Πρωτοβουλία Προσβασιμότητας στο Web ([Web Accessibility Initiative \(WAI\)](#)) για να αντιμετωπίσει την ανάγκη για να γίνει το Web διαθέσιμο για όλους. Η ιστοσελίδα WAI (www.w3.org/WAI) είναι εξαιρετικό σημείο εκκίνησης για περισσότερες πληροφορίες σχετικά με την προσβασιμότητα στο Web. Μια άλλη προσπάθεια της W3C είναι οι προδιαγραφές WAI-ARIA ([Accessible Rich Internet Applications](#) - Προσβάσιμες εφαρμογές Εμπλουτισμένου Διαδικτύου), που αφορούν την προσβασιμότητα εφαρμογών Web που περιλαμβάνουν δυναμικό περιεχόμενο, δέσμες ενεργειών και στοιχεία περιβάλλοντος εργασίας για προχωρημένους που είναι ιδιαίτερα πολύπλοκα σε βοηθητικές συσκευές. Η ARIA ορίζει έναν αριθμό

ρόλων για περιεχόμενο και γραφικά συστατικά που οι συντάκτες μπορούν να εφαρμόσουν ρητά χρησιμοποιώντας το αντίστοιχο χαρακτηριστικό ρόλου (**role attribute**). Οι ρόλοι περιλαμβάνουν γραμμή μενού, γραμμή προόδου, χρονόμετρο και συμβουλές εργαλείων (tooltip), για να αναφέρω μόνο μερικούς. Η ολοκληρωμένη λίστα ρόλων βρίσκεται στη διεύθυνση <https://www.w3.org/TR/wai-aria-1.1/>.

ΚΕΦΑΛΑΙΟ 4 : ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ ΤΗΣ ΙΣΤΟΣΕΛΙΔΑΣ ΤΟΥ ΕΡΓΑΣΤΗΡΙΟΥ EDA (EDA LAB – ELECTRONICS RESEARCH LAB [ERL])

4.1 Μια σχετικά μικρή εισαγωγή

Ανοίγουμε τον επεξεργαστή κώδικα (code editor) – π.χ. Notepad++. Δημιουργούμε ένα νέο έγγραφο. Για να «καταλάβει» η μηχανή απόδοσης του κάθε προγράμματος περιήγησης ότι πρόκειται για ένα έγγραφο HTML αποθηκεύουμε το έγγραφο με κατάληξη “.html”. Αυτό είναι καλύτερα να το κάνουμε εξαρχής πριν να αρχίσουμε να γράφουμε τον κώδικα ώστε να «καταλάβει» ο editor ότι πρόκειται για ένα έγγραφο HTML. Έτσι θα μπορέσει να στοιχίσει το περιεχόμενο και να αντιστοιχίσει κάθε «άνοιγμα» στοιχείου με το αντίστοιχο «κλείσιμο» του, π.χ. το “<head>” με το “</head>” αντίστοιχα. Αυτό είναι πολύ σημαντικό για την αντιμετώπιση προβλημάτων και την εύρεση “bug” , όταν χρειαστεί στον κώδικα μας. Το πρώτο έγγραφο HTML το οποίο θα περιλαμβάνει τον κώδικα για την απόδοση της αρχικής σελίδας, συνήθως το ονομάζουμε “index.html”. Κάνουμε το ίδιο για τα έγγραφα CSS . Δηλαδή δημιουργούμε ένα έγγραφο και το αποθηκεύουμε με κατάληξη “.css”. Στη συνέχεια μπορούμε να επεξεργαστούμε το έγγραφο και να αρχίσουμε να γράφουμε τον κώδικά μας.

4.1.1 Τα στοιχεία του εγγράφου HTML

Είναι αναγκαία πέντε βασικά στοιχεία στην σύνταξη του κειμένου HTML. Το <!DOCTYPE html>, το <html> , το <head> και το <body> και το <title>. Επίσης δύο πολύ σημαντικά χαρακτηριστικά της σήμανσης HTML είναι η κλάση <class> και το χαρακτηριστικό <a href ..>.

Αρκετά σημαντικά είναι και τα σχόλια τα οποία μπορούμε να συμπεριλάβουμε ανάμεσα στους χαρακτήρες <!--και --> . Τα σχόλια δεν λαμβάνονται υπόψιν από τους μηχανισμοί απόδοσης και τα γράφουμε για να διευκολυνόμαστε εμείς οι ίδιοι και να θυμόμαστε σημαντικές λεπτομέρειες στην σύνταξη του κώδικά μας.

Το στοιχείο `<!DOCTYPE html>` είναι απαραίτητο για να καταλαβαίνει το πρόγραμμα περιήγησης μας ότι πρόκειται για ένα έγγραφο της έκδοσης HTML 5. Χρειάζεται μόνο μία φορά στην αρχή του εγγράφου και για αυτό δεν χρειάζεται το «άνοιγμά» και το «κλείσιμό» του όπως όλα τα υπόλοιπα στοιχεία.

Το στοιχείο `<html>` περιλαμβάνει όλα τα υπόλοιπα στοιχεία του εγγράφου. Καθορίζει ολόκληρο το έγγραφο. Όλα τα υπόλοιπα στοιχεία δηλαδή βρίσκονται ανάμεσα στο `<html>` και στο `</html>`.

Το στοιχείο `<head>` περιλαμβάνει όλα τα στοιχεία κεφαλίδας : `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, και `<base>`. Επίσης στην κεφαλίδα `<head>` περιλαμβάνουμε όλες τις απαραίτητες προσθέσεις (plugins) για να μπορέσουν να λειτουργήσουν το πλαίσια CSS και Javascript του Bootstrap όπως φαίνεται στην Εικόνα 10.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
```

Εικόνα 10. Συμπεριλαμβάνουμε το Bootstrap από ένα CDN (Content Delivery Network - Δίκτυο Παροχής Περιεχομένου)

Το στοιχείο `<title>` ορίζει τον τίτλο του εγγράφου και απαιτείται σε όλα τα έγγραφα HTML. Στην ιστοσελίδα ERL :

```
<!--Title-->
<title> Electronics Research Lab</title>
```

Το στοιχείο `<style>` χρησιμοποιείται για τον ορισμό πληροφοριών στυλ για την συγκεκριμένη σελίδα HTML. Με άλλα λόγια περιλαμβάνει κώδικα CSS για την ιστοσελίδα. Αυτός είναι ένας από τους τρεις τρόπους για να συμπεριλάβουμε κώδικα CSS στην ιστοσελίδα μας , τους οποίους θα περιγράψω λίγο παρακάτω.

Το στοιχείο <link> χρησιμοποιείται για τη σύνδεση με εξωτερικά φύλλα στυλ. Αυτός είναι και ο δεύτερος τρόπος για να συμπεριλάβουμε κώδικα CSS στην ιστοσελίδα μας.

Π.χ. <link rel="stylesheet" href="people.css">

Το χαρακτηριστικό (attribute) href χρησιμοποιείται πάντα για να δημιουργήσουμε έναν σύνδεσμο, άσχετα αν αυτός είναι σύνδεσμος για CSS ή σύνδεσμος για μία άλλη ιστοσελίδα ή γενικά για κάποιο αρχείο. Σε περίπτωση που ο σύνδεσμος παραπέμπει στην μετάβαση σε κάποια ιστοσελίδα ή αρχείο, είτε εσωτερικά στον ιστότοπο, είτε εξωτερικά, χρησιμοποιείται πάντα σε συνδυασμό με το χαρακτηριστικό <a>. Π.χ. ο σύνδεσμος για την ιστοσελίδα People.html είναι ο εξής:

```
<a href="http://inf-server.inf.uth.gr/~digogos/EDA_HTML/People.html">People</a>
```

όπου αυτό που υπάρχει ανάμεσα στο στοιχείο <a ...> και το κλείσιμό του, , είναι η ονομασία του συνδέσμου. Βάζουμε δηλαδή μία ετικέτα στον σύνδεσμο αν δεν θέλουμε να εμφανίζεται όλη η διαδρομή στον server για το αντίστοιχο αρχείο. Στο συγκεκριμένο παράδειγμα είναι η λέξη People. Να αναφέρω ότι αν η τιμή "href" δεν υπάρχει και γράψουμε " href="#" " τότε ο σύνδεσμος είναι κενός και πατώντας πάνω του γίνεται ανακατεύθυνση στην ίδια την σελίδα που βρισκόμαστε. Επίσης, αν θέλουμε να πατάμε πάνω στον σύνδεσμο και η ιστοσελίδα που αντιστοιχεί στον σύνδεσμο να ανοίγει σε νέα καρτέλα, πρέπει να ενσωματώσουμε στο χαρακτηριστικό <a> την ιδιότητα " target="_blank" ".

Το στοιχείο <meta> χρησιμοποιείται για τον καθορισμό του συνόλου χαρακτήρων που χρησιμοποιείται, της περιγραφής σελίδας, των λέξεων-κλειδιών, του συντάκτη και άλλων μεταδεδομένων. Τα μεταδεδομένα χρησιμοποιούνται από προγράμματα περιήγησης (τρόπος εμφάνισης περιεχομένου), από μηχανισμούς αναζήτησης (λέξεις-κλειδιά) και άλλες υπηρεσίες Web.

Π.χ. Ορισμός του συνόλου χαρακτήρων που χρησιμοποιείται

```
<meta charset="UTF-8">
```

Π.χ. Ορισμός μια περιγραφής της ιστοσελίδας μας:

```
<meta name="description" content="EDA Lab">
```

Το στοιχείο <meta> είναι επίσης πολύ σημαντικό για να ρυθμίσουμε την προβολή (viewport) της ιστοσελίδας μας. Η προβολή είναι η ορατή περιοχή του χρήστη μιας ιστοσελίδας. Ποικίλλει ανάλογα με τη συσκευή και θα είναι μικρότερη σε ένα κινητό τηλέφωνο από ό,τι σε μια οθόνη υπολογιστή. Αυτή η λειτουργία έχει να κάνει ασφαλώς με το Responsiveness (Ανταπόκριση) που περιγράψα στην Ενότητα 3.4. Είναι το βασικότερο στοιχείο Ανταπόκρισης και θα πρέπει να συμπεριλάβουμε την επόμενη γραμμή κώδικα σε όλες τις ιστοσελίδες μας.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Η Εικόνα 11 δείχνει πώς αλλάζει η προβολή (-τα δύο στιγμιότυπα αριστερά-) όταν συμπεριλάβουμε αυτήν την γραμμή κώδικα και πώς δεν αλλάζει αν δεν την συμπεριλάβουμε(-το στιγμιότυπο δεξιά-) ανάλογα με το μέγεθος της οθόνης μας.



Εικόνα 11 . Πώς αλλάζει η προβολή (viewport) στην συσκευή του κινητού ανάλογα με την οθόνη μας αν χρησιμοποιήσουμε το στοιχείο `<meta name="viewport" content="width=device-width, initial-scale=1.0">` .

Μία ακόμα λειτουργία του στοιχείου <meta> έχει επίσης ενδιαφέρον, η οποία θέτει την εικόνα – λογότυπο (logo) που θα εμφανίζεται μαζί με τον σύνδεσμο (link) του ιστότοπου όταν κάποιος μοιράζεται – προωθεί (share) τον σύνδεσμο μέσω εφαρμογών μηνυμάτων όπως το viber, το messenger κ.ά. Η εικόνα 12 δείχνει τον απαραίτητο κώδικα.

```
<meta property="og:image"
content="http://inf-server.inf.uth.gr/~digogos/EDA_images/blue_glowing.jpg" />
```

Εικόνα 12 . Ο κώδικας για το λογότυπο που θα εμφανίζεται όταν προωθείται ο σύνδεσμος του ιστότοπου μέσω εφαρμογών μηνυμάτων.

Το στοιχείο `<script>` χρησιμοποιείται για τον ορισμό JavaScripts στην πλευρά του πελάτη.

Το στοιχείο `<base>` καθορίζει τη βασική διεύθυνση URL και τον βασικό προορισμό για όλες τις σχετικές διευθύνσεις URL σε μια σελίδα. Δεν είναι απαραίτητο να το συμπεριλάβουμε.

Το στοιχείο `<body>` καθορίζει το σώμα του κειμένου . Σε αυτό γράφουμε όλον τον υπόλοιπο απαραίτητο κώδικα για την παρουσίαση του εγγράφου. Μπορεί να περιλαμβάνει παραγράφους `<p>`, επικεφαλίδες `<h1>` έως `<h6>` (το 1 σημαίνει τη μεγαλύτερη επικεφαλίδα και το 6 την μικρότερη) , στοιχεία `<div>` για να χωρίσουμε τη σελίδα , μας σε διάφορα μέρη , ετικέτες (tags) ul για να ορίσουμε λίστες , και πολλά άλλα στοιχεία που μπορεί να περιλαμβάνει ένα έγγραφο HTML.

Το χαρακτηριστικό `<class>` χρησιμοποιείται για τον καθορισμό ίσων στυλ για στοιχεία με το ίδιο όνομα κλάσης. Θα αναφερθώ σε αυτό το χαρακτηριστικό αρκετές φορές στη συνέχεια.

4.1.2 Προσθέτοντας CSS

Πριν να μιλήσω για τους τρόπους που μπορούμε να συμπεριλάβουμε CSS στην ιστοσελίδα μας να αναφέρω ότι τα σχόλια στα Επικαλυπτόμενα Φύλλα γράφονται λίγο διαφορετικά από τα σχόλια στα έγγραφα HTML , δηλαδή ως εξής:

```
/* ..... */
```

Ανάμεσα στους χαρακτήρες `/*` και `*/` γράφουμε ό,τι θέλουμε και όπως και στη σήμανση HTML ό,τι υπάρχει μέσα σε σχόλιο δεν λαμβάνεται υπόψιν από τους μηχανισμούς απόδοσης.

Όπως ανέφερα παραπάνω υπάρχουν τρεις τρόποι για να συμπεριλάβουμε CSS στην ιστοσελίδα μας: External CSS, Internal CSS, Inline CSS.

External CSS (Εξωτερικό φύλλο): Δημιουργούμε ένα ξεχωριστό αρχείο CSS, όπως π.χ. το “people.css” και χρησιμοποιούμε το στοιχείο <link> για τη σύνδεση με το εξωτερικό φύλλο στυλ που βρίσκεται αποθηκευμένο σε κάποιο κατάλογο σε κάποια διαδρομή στον server.

```
<link rel="stylesheet" type="text/css" href="http://inf-server.inf.uth.gr/~digogos/EDA_CSS/people.css">
```

Θα παραθέσω λίγο αργότερα κάποια στιγμιότυπα του συγκεκριμένου αρχείου όταν θα περιγράψω την καρτέλα People της ιστοσελίδας του ERL.

Internal CSS (Εσωτερικό φύλλο): Ένα εσωτερικό φύλλο στυλ μπορεί να χρησιμοποιηθεί αν μία σελίδα HTML έχει ένα μοναδικό στυλ. Γράφεται μέσα στο ίδιο το έγγραφο html και ορίζεται μέσα στο στοιχείο <style> όπως φαίνεται στην Εικόνα 13. Είναι προτιμότερο να συμπεριλάβουμε το στοιχείο <style> μέσα στο στοιχείο κεφαλίδας <head> αλλά μπορούμε να το συμπεριλάβουμε και ανάμεσα στο <head> και το <body> ή και μέσα στο <body>. Για παράδειγμα το εσωτερικό φύλλο που χρησιμοποίησα ίδιο σε όλα τα αρχεία html του ιστότοπου του εργαστηρίου γιατί περιλαμβάνουνε ίδια αυτά τα δύο συγκεκριμένα χαρακτηριστικά είναι το εξής :

```
<!--CSS-->
<style>

/*for icon-bar : Menu appears as an icon-- for mobile device */
.icon-bar {
width: 90px;
background-color: DodgerBlue;
}

/*for optional images : images with id=optionalstuff will not appear in devices with
width less than 991px */
@media (max-width:991px) {
img#optionalstuff {
display: none;
}
}
</style>
```

Εικόνα 13. Internal CSS (Εσωτερικό φύλλο)

Το πρώτο μέρος δημιουργεί ένα στοιχείο “icon-bar” το οποίο είναι μία γραμμή (μπάρα) πλάτους 90 pixel και χρώματος DodgerBlue. Χρησιμοποιώ αυτή τη μπάρα

όταν παρακάτω στο έγγραφο δημιουργώ τη μπάρα περιήγησης – μενού (Navigation bar) ώστε να εμφανίζονται τρεις μπάρες – χρησιμοποίησα τρία “icon-bar” – σε οθόνες που δεν είναι δυνατόν να εμφανίζουν ολόκληρο το μενού και όλες τις καρτέλες του. Αυτό φαίνεται στην Εικόνα 14.



Εικόνα 14. Πώς εμφανίζεται το μενού (πάνω) σε οθόνες που δεν «χωράνε» να εμφανιστούν όλες οι καρτέλες του. Πώς εμφανίζεται στην οθόνη του υπολογιστή (κάτω).

Αυτός ο μικρός κώδικας CSS βέβαια ορίζει απλά μόνο ένα στοιχείο. Μία μπάρα πλάτους 90 pixel. Για να αποδώσω λειτουργικότητα τέτοια ώστε οι τρεις μπάρες να γίνουν ένα κουμπί (button), το οποίο θα εμφανίζεται σε μικρότερες οθόνες και όταν το πατάς τότε θα εμφανίζεται το μενού έπρεπε να συμπεριλάβω τον εξής κώδικα (Εικόνα 15):

```
<!-- Brand and toggle get grouped for better mobile display -->
<div class="navbar-header">

  <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
    <span class="sr-only">Toggle navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </button>
  <a class="navbar-brand" href="#">ERL</a>
</div>
```

Εικόνα 15. Κουμπί εναλλαγής του μενού

Το στοιχείο <button> δίνει τη λειτουργικότητα ενός κουμπιού που κάνει collapse – καταρρέει – όταν το πατάμε. Αυτό φαίνεται από την κλάση

class="navbar-toggle collapsed" του κουμπιού.

Η ετικέτα είναι ένας ενσωματωμένος περιέκτης που χρησιμοποιείται για την επισήμανση ενός τμήματος κειμένου ή ενός τμήματος ενός εγγράφου.

Η κλάση `class="sr-only"` χρησιμοποιείται για την απόκρυψη πληροφοριών που προορίζονται μόνο για προγράμματα ανάγνωσης οθόνης από τη διάταξη της σελίδας που αποδίδεται. Έτσι αποδίδεται και η αντίστοιχη λειτουργικότητα που περιέγραψα νωρίτερα.

Δεν θα αναφερθώ άλλο στο μενού γιατί μπορεί κανείς πολύ εύκολα να ανατρέξει στην ιστοσελίδα https://www.w3schools.com/bootstrap/bootstrap_navbar.asp όπου βρίσκεται έτοιμος ο κώδικας για τα διάφορα navigation bars του bootstrap. Να αναφέρω μόνο ότι στην γενική κλάση `<nav class="navbar navbar-default">` που χρειάζεται για να δημιουργήσουμε το navigation bar και μέσα της γράφουμε όλο τον υπόλοιπο απαραίτητο κώδικα, εάν δεν χρησιμοποιήσουμε την επιλογή `navbar-default` ή οποιαδήποτε άλλη επιλογή, δηλαδή εάν χρησιμοποιήσουμε απλά την κλάση `<nav class="navbar">`, τότε το μενού σχηματίζεται χωρίς κάποιο χρώμα. Έχει χρώμα λευκό. Αυτή την κλάση χρησιμοποίησα στο μενού της σελίδας του εργαστηρίου. Τα γράμματα των λέξεων που συνιστούν τις καρτέλες του μενού είναι μπλε γιατί κάθε λέξη – καρτέλα είναι σύνδεσμος και το προεπιλεγμένο χρώμα των συνδέσμων είναι μπλε.

Το δεύτερο μέρος του συγκεκριμένου εσωτερικού CSS (Εικόνα 13) δηλώνει ότι οποιαδήποτε εικόνα (`img`) μέσα στο έγγραφο έχει το αναγνωριστικό `“optionalstuff”` (`id="optionalstuff"`) (προαιρετικό) δεν θα εμφανίζεται σε οθόνες με πλάτος μικρότερο από 991px. Αυτό γίνεται χρησιμοποιώντας την επιλογή `“display: none;”`. Το χρησιμοποίησα για την ενότητα Contact όπου το ένα από τα δύο ρομποτάκια (το δεύτερο) δεν εμφανίζεται σε μικρότερες οθόνες. Στις δοκιμές εμφάνισης που έκανα, αν εμφανιζόταν, τα γράμματα στην αντίστοιχη ενότητα θα φαινότουσαν πολύ μικρά.

Inline CSS (Ενσωματωμένο CSS) : Ένα ενσωματωμένο στυλ μπορεί να χρησιμοποιηθεί για την εφαρμογή ενός μοναδικού στυλ για ένα μεμονωμένο στοιχείο. Για να χρησιμοποιήσουμε ενσωματωμένα στυλ, προσθέτουμε την ιδιότητα στυλ στο αντίστοιχο στοιχείο. Το χαρακτηριστικό `“style”` μπορεί να περιέχει οποιαδήποτε ιδιότητα CSS. Για παράδειγμα ενσωματώνοντας στην κεφαλίδα `“h1”`

το χαρακτηριστικό “style” όπως φαίνεται εδώ, επιλέγουμε για την “h1” μία γραμματοσειρά διαφορετική από την προεπιλεγμένη. Το ότι συμπεριέλαβα τρεις γραμματοσειρές αντί για μία σημαίνει ότι εάν το πρόγραμμα περιήγησης ή καλύτερα ο μηχανισμός απόδοσης δεν μπορέσει να βρει και να χρησιμοποιήσει την πρώτη, θα χρησιμοποιήσει την δεύτερη. Αν δεν μπορέσει να χρησιμοποιήσει ούτε την δεύτερη, θα χρησιμοποιήσει την τρίτη, αλλιώς θα χρησιμοποιήσει την προεπιλεγμένη γραμματοσειρά.

```
<h1 style = "font-family:courier,arial,Helvetica;" class="display-1">.
```

Να αναφέρω εδώ ότι μπορούμε να εισάγουμε στην κεφαλίδα <head> ειδικές γραμματοσειρές , εκτός από τις ήδη κοινότητες υπάρχουσες, από την Google, και να τις χρησιμοποιήσουμε με τον ίδιο τρόπο που χρησιμοποιούμε οποιαδήποτε άλλη γραμματοσειρά σε οποιοδήποτε στοιχείο του εγγράφου μας. Για παράδειγμα στην αρχική σελίδα του ERL εισάγω και χρησιμοποιώ τις εξής γραμματοσειρές όπως φαίνεται στην Εικόνα 16: Sofia, Volkorn, Cinzel .

```
<!-- Imported Fonts-->
<link href='https://fonts.googleapis.com/css?family=Sofia' rel='stylesheet'>
<link href="https://fonts.googleapis.com/css2?family=Vollkorn&display=swap" rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Cinzel&display=swap" rel="stylesheet">
```

Εικόνα 16. Εισαγωγή γραμματοσειρών από την Google

4.2 Η αρχική σελίδα : ιστοσελίδα (καρτέλα) Home

Η αρχική σελίδα αποτελείται από το έγγραφο “index.html”. Δεν χρησιμοποίησα για αυτήν την σελίδα εξωτερικό CSS αλλά το εσωτερικό CSS που χρησιμοποίησα ίδιο και σε όλες τις άλλες σελίδες και περιέγραψα νωρίτερα καθώς και ενσωματωμένο CSS όπου χρειάστηκε. Έχω παραθέσει ήδη νωρίτερα μερικά στιγμιότυπα αυτής της σελίδας στις Εικόνες 9 και 11.

Μετά τον κώδικα που χρειάζεται για το Navigation Bar, ο οποίος είναι ίδιος και σε όλες τις υπόλοιπες ιστοσελίδες του ιστότοπου, υπάρχει ο κώδικας για την παρουσίαση της υπόλοιπης σελίδας. Η στρατηγική που ακολούθησα είναι να

χωρίσω (divide) την σελίδα σε τέσσερις ενότητες χρησιμοποιώντας την γενική ετικέτα <div> . Το στιγμιότυπο της πρώτης ενότητας φαίνεται στην Εικόνα 17.

```
<!------- Deivide page to sections-- ----->
<!-------Section 1 : Welcome----->
<div id="section1" class="jumbotron text-center"
  style="background-image: url('http://inf-server.inf.uth.gr/~digogos/EDA_images/fantasy.jpg');
  margin-bottom: 0px;" >
  <br><br><br><br><br><br><br>
  <h1 style = "font-family:courier,arial,Helvetica;" class="display-1">
  <strong> <font color="white"> Welcome to Electronics Research Lab </font> </strong>
  </h1>

  <h2 style = "font-family:Sofia,arial,Helvetica;" class="display-1">
  <b> <font size="70" color="white"> Feel Free to Look Around </font> </b>
  <br><br><br><br><br>
  </h2>
</div>
<!------- End Section 1----->
```

Εικόνα 17. Πρώτη ενότητα της αρχικής σελίδας

Η πρώτη ενότητα παρόλο που είναι ιδιαίτερα απλή περιέχει λίγα σημαντικά στοιχεία που προσδίδουν στυλ στην παρουσίασή της. Στην αρχική και γενική ετικέτα <div> που μέσα της περιέχει όλον τον κώδικα της συγκεκριμένης ενότητας, έδωσα το αναγνωριστικό “id = section1”, σε περίπτωση που χρειαστεί να αναφερθώ σε αυτήν με κάποιον εσωτερικό σύνδεσμο ή για κάποιον άλλον λόγο όπως περιέγραψα νωρίτερα στο παράδειγμα της προαιρετικής εικόνας. Παρόμοια στις επόμενες ενότητες έδωσα τα αναγνωριστικά 2,3 και 4.

Ενδιαφέρον έχουν μερικά στοιχεία όπως :

Η κλάση “ class="jumbotron text-center" “ δεσμεύει ένα πλαίσιο με στρογγυλεμένες γωνίες και μεγεθύνει τα μεγέθη γραμματοσειράς του κειμένου μέσα σε αυτό (jumbotron) καθώς και αποδίδει την εμφάνιση οποιουδήποτε κειμένου περιέχει η ενότητα 1 στο κέντρο της οθόνης (text-center). Αυτό γίνεται γιατί συμπεριλαμβάνουμε την κλάση στην γενική ετικέτα της ενότητας 1. Αν την συμπεριλάβαναμε μόνο σε κάποιο στοιχείο εσωτερικά της ενότητας τότε θα απέδιδε αυτήν την εμφάνιση μόνο στο συγκεκριμένο στοιχείο.

Το ίδιο ισχύει και για την εικόνα φόντου της ενότητας. Αυτός ο πολύ συγκεκριμένος τρόπος για να προσδώσουμε εικόνα φόντου, σε κάποια ενότητα ή σε κάποιο στοιχείο ή αν θέλουμε σε ολόκληρη την ιστοσελίδα, είναι που

χρησιμοποιείται μόνο για εικόνες φόντου. Δηλαδή τοποθετούμε στο αντίστοιχο στοιχείο τον κώδικα “ style="background-image: url('.....') ” όπου ως διεύθυνση url τοποθετούμε την διαδρομή όπου βρίσκεται η εικόνα στον server. Για την προσθήκη γενικά εικόνων όπου θέλουμε στην ιστοσελίδα μας, χρησιμοποιείται ένας διαφορετικός τρόπος με την ετικέτα . Θα αναφερθώ λίγο αργότερα σε αυτό.

Η επιλογή “margin-bottom: 0px;” (margin σημαίνει περιθώριο) χρειάζεται ώστε να μην απέχει καθόλου η ενότητα 1 από την ενότητα 2, να μην υπάρχει δηλαδή αισθητό κενό ανάμεσα τους που από προεπιλογή θα έχει χρώμα λευκό.

Η ετικέτα
 σημαίνει καινούρια γραμμή (line break). Χρησιμοποίησα αρκετά
 ώστε να εμφανίζεται το κείμενο όσο παρακάτω ήθελα. Αντί να κάνω αυτό θα μπορούσα να χρησιμοποιήσω για “style” στο αμέσως επόμενο στοιχείο που είναι η κεφαλίδα <h1> κάποιο αντίστοιχο “margin” όπως περιέγραψα. Εδώ θα χρειαζόταν “margin-top:...px;”.

Η ετικέτα είναι για να προσδώσουμε έμφαση στο κείμενο που βρίσκεται ανάμεσα στο άνοιγμα και το κλείσιμό του. Στην συγκεκριμένη περίπτωση στην κεφαλίδα <h1>.

Παρόμοια η ετικέτα που χρησιμοποιώ στην κεφαλίδα <h2> είναι για να κάνω έντονα τα γράμματα της (bold) .

Το στοιχείο είναι για να προσδώσουμε λευκό χρώμα στην γραμματοσειρά καθώς το προεπιλεγμένο χρώμα είναι μαύρο.

Αυτά είναι λίγα αρκετά απλά στοιχεία αλλά απλά τα περιέγραψα γιατί χρησιμοποιώ ίδια η παρόμοια τους οπουδήποτε στον κώδικα οποιασδήποτε καρτέλας και δεν θα χρειαστεί να ξανααναφερθώ σε αυτά.

Στο τέλος της ενότητας 1 χρησιμοποίησα αρκετές ετικέτες
 για νέες γραμμές. Είναι ένα τέχνασμα που χρησιμοποίησα και σε άλλα μέρη του ιστότοπου ώστε να «επεκτείνω» την προβολή της εικόνας φόντου της συγκεκριμένης ενότητας. Αν προσέθετα κάποιο περιθώριο (margin) όπως προηγουμένως δεν θα απέδιδε την ίδια λειτουργικότητα – εμφάνιση καθώς απλά θα παρατεινόταν με λευκό χρώμα η επόμενη ενότητα και δεν θα επεκτεινόταν η προβολή της εικόνας φόντου.

Η ενότητα 2 έχει λίγο περισσότερο ενδιαφέρον. Το αρχικό στοιχείο <div> είναι παρόμοιο. Να αναφέρω μόνο την ιδιότητα “background-size: cover;” του χαρακτηριστικού style η οποία προσδιορίζει το μέγεθος της εικόνας φόντου. Η επιλογή “cover” αλλάζει το μέγεθος της εικόνας φόντου για να καλύψει ολόκληρη την ενότητα, ακόμα και αν πρέπει να επεκτείνει την εικόνα ή να αποκόψει λίγο από τα άκρα της. Ιδιαίτερο ενδιαφέρον σε αυτήν την ενότητα έχει πάλι το Responsiveness. Η κλάση “.container” (περιέκτης) παρέχει έναν περιέκτη σταθερού πλάτους που ανταποκρίνεται στο μέγεθος της οθόνης μας. Αυτήν είναι μία γενική κλάση Ανταπόκρισης και χρειάζεται να την συμπεριλάβουμε ώστε να αποδοθεί λειτουργία στις υπόλοιπες ιδιότητες των στοιχείων που θα συμπεριλάβουμε μέσα στον περιέκτη και που θα περιγράψω σε λίγο.

Να περιγράψω όμως λίγο συνοπτικά την κεφαλίδα <h1> γιατί είναι μια κεφαλίδα η οποία περικλείεται από ένα μπλε περίβλημα όπως φαίνεται στην Εικόνα 11 (το κουτί “ABOUT ERL”) και φαίνεται σαν να βρίσκεται μέσα σε ένα κουτί. Ο κώδικας φαίνεται στην Εικόνα 18.

```
<h1 style = "font-family: 'Cinzel', serif; border: 1px solid #1E90FF;
border-radius: 4px;
padding: 5px;
width: 150px;float: right;margin-right: 200px;"
align="center" class= "display-1" >
<font color=" DeepSkyBlue"> ABOUT ERL </font>
</h1>
```

Εικόνα 18. Κώδικας για το κουτί “ABOUT ERL”

Η ιδιότητα “border: 1px solid #1E90FF;” ορίζει και δίνει μορφή στο περίβλημα το οποίο παίρνει το χρώμα #1E90FF που είναι η συγκεκριμένη απόχρωση του μπλε. Η ιδιότητα “width: 150px;” ορίζει το πλάτος του περιβλήματος. Χωρίς αυτήν θα αποδιδόταν το προεπιλεγμένο πλάτος το περιβλήματος που είναι μεγαλύτερο από 150 pixel. Οι υπόλοιπες ιδιότητες όπως το “padding” , το “margin” και το “float” είναι για την τοποθέτηση του στο συγκεκριμένο σημείο της ιστοσελίδας, πού δηλαδή θα εμφανίζεται. Επειδή χρειάζεται κανείς να «παίξει» και να τροποποιήσει

ανάλογα αυτές τις ιδιότητες και τα ανάλογα ρίξels για να το τοποθετήσει εκεί που ακριβώς θέλει, δεν έχει ιδιαίτερο νόημα να σταθώ σε αυτές.

Ας αρχίσω να περιγράψω το κομμάτι που έχει το μεγαλύτερο ενδιαφέρον. Όπως περιέγραψα νωρίτερα χρησιμοποίησα έναν περιέκτη (“container”) που ευθύνεται για την Ανταπόκριση της ενότητας. Μέσα στον περιέκτη έχω δύο κλάσεις που ο συνδυασμός τους χωρίζει την ενότητα 2 και άρα το πώς φαίνεται αυτή η ενότητα σε δύο στήλες όπως δείχνει η Εικόνα 19.



Εικόνα 19. Ο διαχωρισμός της ενότητας σε δύο στήλες

Κάθε στήλη μπορεί να περιέχει οτιδήποτε εμείς της προσδώσουμε. Ο κώδικας για την πρώτη στήλη φαίνεται στην Εικόνα 20. Ο κώδικας για την δεύτερη στήλη είναι παρόμοιος και έχει την ίδια λειτουργικότητα – απόδοση.

```

<!-- 1st column -->
<div class="col-lg-5 col-md-5 col-sm-10 col-xs-10">
  

  <h2 style = "font-family:Vollkorn,arial,Helvetica;" >Who we are</h2>
  <h4 style = "font-family:Vollkorn,arial,Helvetica;" >
    The ErL Laboratory is part of the Department of Electrical and
    Computer Engineering at University of Thessaly. The ErL focuses
    on addressing the challenges presented by the rapid increase in
    size and complexity of the integrated circuits.It explores methodologies
    and techniques for the development of algorithms and techniques for Electronic
    Design Automation (EDA) tools that target the modeling, simulation, optimization
    and verification of low-power, deep-submicron VLSI circuits.
  </h4>
</div>
<!-- End : 1st column -->

```

Εικόνα 20. Ο κώδικας για την πρώτη στήλη

Το μεγαλύτερο ενδιαφέρον εδώ το έχει πάλι η Ανταπόκριση της ενότητας. Είναι ένα χαρακτηριστικό παράδειγμα πλέγματος του Bootstrap. Το αρχικό στοιχείο `<div>` περιέχει την απαραίτητη κλάση η οποία είναι απλή. Το “lg” που δηλώνει large αποδίδει την εμφάνιση της στήλης σε μεγάλες οθόνες (οθόνες υπολογιστή), το “md” που δηλώνει medium , σε οθόνες μεσαίες μεγέθους (μικρότερες οθόνες υπολογιστή), το “sm” που δηλώνει small, σε μικρές οθόνες (τάμπλετς) και το “xs” που δηλώνει extra small αποδίδει την εμφάνιση της στήλης σε πολύ μικρές οθόνες (τηλέφωνα). Το μέγεθος της ενότητας – οθόνης χωρίζεται πάντα σύμφωνα με το σύστημα πλέγματος του Bootstrap σε 12 στήλες – υποενότητες . Άρα εδώ το “lg-5” και το “md-5” σημαίνει ότι αυτή η στήλη θα καταλαμβάνει σχεδόν το μισό (το μισό είναι 6) μέρος της οθόνης όταν η ιστοσελίδα προβάλλεται σε μεγάλου ή μεσαίου μεγέθους οθόνες υπολογιστή αντίστοιχα. Στην ίδια αναλογία το “sm-10” και το “xs-10” σημαίνει ότι αυτή η στήλη θα καταλαμβάνει σχεδόν ολόκληρη την οθόνη όταν η ιστοσελίδα προβάλλεται σε μικρού ή πολύ μικρού μεγέθους οθόνες αντίστοιχα. Δεν χρησιμοποίησα 6 (το μισό) ή 12 (ολόκληρο) γιατί στους ελέγχους που έκανα απλά εμφανιζόταν πιο όμορφο έτσι.

Στη συνέχεια τοποθέτησα τα στοιχεία της στήλης τα οποία εμφανίζονται με την σειρά το ένα κάτω από το άλλο. Πρώτα είναι μια εικόνα με Ανταπόκριση (“img-responsive”) η οποία βρίσκεται και σε περίβλημα μικρογραφίας (“img-thumbnail”). Το περίβλημα μικρογραφίας απλά αποδίδει από προεπιλογή ένα λευκό περίβλημα κάποιων συγκεκριμένων pixels που αν θέλουμε μπορούμε να το τροποποιήσουμε. Έχει παρόμοια λειτουργικότητα με το «χειροκίνητα» φτιαγμένο περίβλημα που περιέγραψα νωρίτερα στην Εικόνα 18. Ο κώδικας για την εικόνα όπως και για όλες τις εικόνες εκτός από τις εικόνες φόντου είναι και ο χαρακτηριστικός τρόπος για να εισάγουμε εικόνα στην ιστοσελίδα μας. Αν θέλουμε να δώσουμε ανταπόκριση ή να της δώσουμε περίβλημα μικρογραφίας όπως έκανα εδώ ή να την στρογγυλοποιήσουμε χρησιμοποιούμε την αντίστοιχη κλάση που δίνεται έτοιμη από το Bootstrap ή γράφουμε τον δικό μας απαραίτητο κώδικα CSS για να προσδώσουμε ό,τι εμφάνιση θέλουμε. Η ιδιότητα “alt” (alternative = εναλλακτικά) - alt="blue_glowing" – είναι για την περίπτωση που δεν μπορεί να φορτώσει η ίδια η

εικόνα για κάποιο λόγο. Τότε θα εμφανίζεται οτιδήποτε κείμενο εμείς προσδιορίσουμε ως “alt”.

Τα αμέσως επόμενα στοιχεία είναι δύο κεφαλίδες για τις οποίες δεν χρειάζεται να μπω σε λεπτομέρειες.

Με ακριβώς παρόμοιο τρόπο είναι φτιαγμένη και η δεύτερη στήλη της ενότητας μόνο που αντι για “lg-5” και “md-5” και “sm-10” και “xs-10” χρησιμοποίησα “lg-6” και “md-6” και “sm-12” και “xs-12” αντίστοιχα. Πάλι για λόγους ομορφιάς και παρουσίασης και κανέναν άλλον λόγο. Κατά τα άλλα η λογική και αρχιτεκτονική σχεδίασης είναι ίδια.

Η Τρίτη ενότητα της αρχικής σελίδας περιέχει ξανά μια εικόνα φόντου, μία κεφαλίδα και τέσσερις εικόνες. Θα δώσω απλά μία σύντομα περιγραφή για μία από τις τέσσερις εικόνες γιατί έχει ένα μικρό ενδιαφέρον. Οι υπόλοιπες είναι ορισμένες με τον ίδιο σχεδόν τρόπο. Ο κώδικας φαίνεται στην Εικόνα 21.

```
<!-- 1st column : CERTH-->
<div class="col-lg-4 col-md-4 col-sm-4-first col-xs-4-first" >
<a href="https://www.certh.gr/" target="_blank">
  
</a>
</div>
<!-- End : 1st column : CERTH-- -->
```

Εικόνα 21 . Ο κώδικας για την εικονίτσα CERTH

Είναι μία από τις τέσσερις στήλες (- τέσσερις εικόνες -) ενός περιέκτη. Το ενδιαφέρον εδώ είναι η ιδιότητα “first” για μικρές και πολύ μικρές συσκευές που προσδίδει στην εικόνα την ιδιότητα να εμφανίζεται πρώτη , πάνω από τις άλλες σε τέτοιου μεγέθους συσκευές. Χωρίς αυτήν την ιδιότητα και λόγο του ότι η εικόνα έχει προσδιορισμένο μέγεθος όπως φαίνεται στο χαρακτηριστικό style λίγο παρακάτω στον κώδικα, άρα δεν είναι εικόνα με ανταπόκριση, που σημαίνει ότι δεν αλλάζει ανάλογα με το μέγεθος της οθόνης, οι εικόνες θα εμφανιζόταν στην σειρά η μία μετά την άλλη όπως κάνουνε και σε μεγάλες οθόνες. Στην ίδια λογική ,

χρησιμοποίησα τις επιλογές second , third και fourth στις υπόλοιπες τρεις εικόνες αντίστοιχα. Η απόδοση της ενότητας σε μεγάλη οθόνη και ένα μέρος αυτής σε μικρή οθόνη φαίνεται στην Εικόνα 22.



Εικόνα 22. Η απόδοση των εικόνων σε μικρές συσκευές (δεξιά) με την χρήση των ιδιοτήτων first,second, third και fourth

Η τέταρτη ενότητα της αρχικής σελίδας είναι ίδια και σε όλες τις υπόλοιπες ιστοσελίδες του ιστότοπου του εργαστηρίου. Όπως περιέγραφα στην υποενότητα 4.1.2 χρειάστηκε να προσδιορίσω στο <head> ένα συγκεκριμένο Εσωτερικό CSS το οποίο δηλώνει ότι οποιαδήποτε εικόνα (“img”) μέσα στο έγγραφο έχει το αναγνωριστικό “optionalstuff” (id="optionalstuff") (προαιρετικό) δεν θα εμφανίζεται σε οθόνες με πλάτος μικρότερο από 991px. Αυτό γίνεται χρησιμοποιώντας την επιλογή “display: none;”. Η λογική σύνταξης της ενότητας είναι ίδια με τις δύο προηγούμενες, πάλι κάποιες στήλες που καταλαμβάνουν κάποιο μέρος της οθόνης και οι ιδιότητες “first”, “second” και “third” όπου χρειάζονται. Παραθέτω μόνο τον κώδικα (Εικόνα 23 – πάνω) για την τρίτη στήλη που στην συγκεκριμένη περίπτωση είναι μόνο μια εικόνα. Στην συγκεκριμένη περίπτωση η ιδιότητα “third” δεν είναι απαραίτητη γιατί ούτως ή άλλως η εικόνα δεν εμφανίζεται σε μικρές και πολύ μικρές συσκευές λόγω του αναγνωριστικού της “optionalstuff” και το πώς αυτό προσδιορίζεται από το αντίστοιχο CSS. Η πρόσθεση της ιδιότητας “third” όμως δεν μας ενοχλεί και είναι καλό να υπάρχει σε περίπτωση που θέλουμε να ανατρέξουμε στον κώδικα και να κάνουμε κάποια τροποποίηση όπως να αφαιρέσουμε το συγκεκριμένο αναγνωριστικό. Η Εικόνα 23 δείχνει την

απόδοση της ενότητας στον υπολογιστή και την απόδοση της στο κινητό όπου «αποκρύπτεται» το ρομποτάκι που βρίσκεται δεξιά.

```
<div class="col-lg-3 col-md-3 col-sm-12-third col-xs-12-third ">  
    
</div>
```



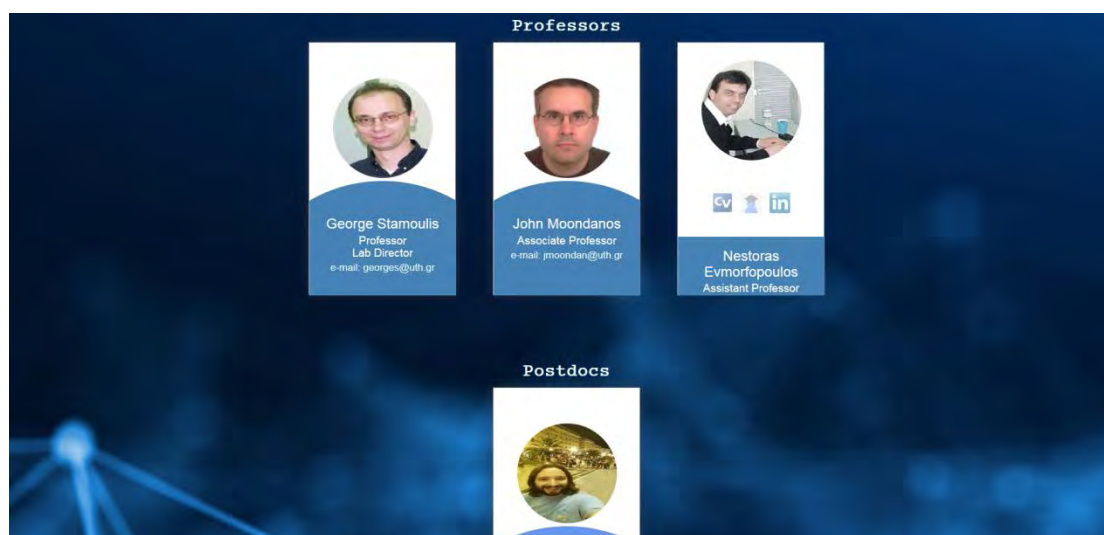
Εικόνα 23. Χρήση του αναγνωριστικού optionalstuff ορισμένου από το αντίστοιχο CSS

Να αναφέρω επίσης απλά την χρήση των “glyphicon” όπως το εικονίδιο τηλεφώνου ή το εικονίδιο αλληλογραφίας (γράμμα) που χρησιμοποιώ στο μέρος με τα στοιχεία επικοινωνίας και που είναι απλά πολύ μικρά εικονίδια και περιλαμβάνονται (260 εικονίδια) στο Bootstrap. Για παράδειγμα το εικονίδιο του τηλεφώνου προσδιορίζεται από τον κώδικα “ “ και μπορούμε απλά να τον χρησιμοποιήσουμε όπου θέλουμε για να τοποθετήσουμε τον εικονίδιο όπου μας αρέσει ή μας βολεύει.

4.3 Η ιστοσελίδα (καρτέλα) People

Στην ιστοσελίδα People έχει περισσότερο ενδιαφέρον ο κώδικας CSS (εξωτερικό φύλλο) που ορίζει τις «αλληλεπιδραστικές» καρτέλες καθεμία από τις οποίες περιέχει την φωτογραφία, τους συνδέσμους και τις πληροφορίες για το αντίστοιχο άτομο της ομάδας του εργαστηρίου, οπότε θα επικεντρωθώ σε αυτόν περισσότερο και θα αναφερθώ μόνο στον κώδικα HTML που είναι απλά χρήση των κλάσεων που

ορίζονται στο CSS καθώς και χρήση παραπλήσιου κώδικα όσων χρησιμοποιήσα στην αρχική ιστοσελίδα. Έχω ήδη περιγράψει στην υποενότητα 4.1.2 πώς εισάγουμε εξωτερικό CSS σε ένα έγγραφο HTML. Παραθέτω ένα στιγμιότυπο της ιστοσελίδας στην Εικόνα 24 για να παρουσιάσω τις καρτέλες. Η αλληλεπιδραστικότητα των καρτέλων ενεργοποιείται με το πάσασμα του δείκτη του ποντικιού «πάνω» από την καρτέλα. Στο στιγμιότυπο της Εικόνας 24 φαίνεται αυτήν η αλληλεπιδραστικότητα στην τρίτη σε σειρά καρτέλα (του κύριου Νέστωρα Ευμορφόπουλου). Για να τραβήξω το στιγμιότυπο έκανα ακριβώς αυτό, πάσασα το δείκτη του ποντικιού «πάνω» από την τρίτη καρτέλα. Οι υπόλοιπες καρτέλες φαίνονται πως εμφανίζονται πριν την συγκεκριμένη αλληλεπίδραση. Όλες οι καρτέλες των καθηγητών είναι ίδιες, οπότε ο κώδικας CSS είναι ένας (ίδιος) για όλες αυτές τις καρτέλες, αφού στο CSS ορίζουμε γενικές κλάσεις που μπορούμε να χρησιμοποιήσουμε όπου θέλουμε, και ο κώδικας HTML είναι απλά αντιγραφή της πρώτης καρτέλας που έφτιαξα, απλά αλλάζουν οι διευθύνσεις των συνδέσμων και των εικόνων και οι πληροφορίες (όνομα και email) για κάθε άτομο. Το στιγμιότυπο βέβαια δεν δείχνει ολόκληρη την ιστοσελίδα που συνεχίζει προς τα κάτω και έχει αντίστοιχα την ενότητα των μεταδιδακτορικών, μετά των διδακτορικών και τελευταία την ενότητα των μεταπτυχιακών φοιτητών.



Εικόνα 24. Στιγμιότυπο της αλληλεπιδραστικότητας των καρτέλων των ατόμων της ομάδας του εργαστηρίου

Ο κώδικας CSS είναι αρκετές γραμμές οπότε θα περιγράψω μόνο μερικά σημαντικά κομμάτια. Η πρώτη ενδιαφέρουσα κλάση παρότι η πιο μικρή είναι η κλάση transition που φαίνεται στην Εικόνα 25.

```
/* For the transition when hovering */  
.transition {  
  transition: .7s cubic-bezier(.3, 0, 0, 1.3)  
}
```

Εικόνα 25. Η κλάση transition ορισμένη στο εξωτερικό CSS “people.css”

Η πρώτη ιδιότητα που εδώ είναι “.7s” (0.7 sec) δηλώνει πόσο διαρκέσει αυτήν η αλληλεπίδραση, δηλαδή μέσα σε πόσο χρόνο θα γίνει η μετάβαση από τη μία κατάσταση εμφάνισης της καρτέλας στην άλλη. Η ιδιότητα “cubic-bezier” μας επιτρέπει να ορίζουμε τις τιμές της λειτουργικότητας της μετάβασης. Το πρώτο και το τρίτο στοιχείο, (εδώ “.3” και “0” αντίστοιχα) ορίζει το πόσο «απότομη» θα είναι η μετάβαση από την μία κατάσταση στην άλλη. Όσο μεγαλύτερες αυτές οι τιμές τόσο πιο απότομη και η μετάβαση. Είναι δύσκολο κανείς να καταλάβει ακριβώς αυτήν την λειτουργικότητα αν δεν πειραματιστεί απλά με διάφορες τιμές για να ελέγχει το αποτέλεσμα. Το δεύτερο και το τέταρτο στοιχείο δείχνει πόσο πάνω και κάτω μεταφέρονται τα στοιχεία της καρτέλας (εδώ “0” και “1.3” αντίστοιχα) , που εδώ είναι η υποενότητα με τη φωτογραφία και τους συνδέσμους και η μπλε υποενότητα αντίστοιχα, κατά τη διάρκεια της μετάβασης (δηλαδή αυτά τα συγκεκριμένα 0.7 sec που δήλωσα ως την πρώτη αρχική ιδιότητα του transition).

Δεν έχει και ιδιαίτερο νόημα να αναλύσω λεπτομερώς την κάθε μικρή επιμέρους λειτουργικότητα της κάθε κλάσης και του αντίστοιχου στοιχείου της, πρώτον γιατί αυτό θα χρειαζόταν ένα βιβλίο χιλίων σελίδων τουλάχιστον, αν ήθελα να περιγράψω κάθε ιδιότητα του κώδικα CSS που χρησιμοποίησα για κάθε ιστοσελίδα, αλλά και δεύτερον, γιατί, όπως ανέφερα νωρίτερα, η μόνη λύση για να καταλάβει κανείς αυτές τις λεπτομερείς λειτουργίες είναι να πειραματιστεί με τις τιμές τους. Για αυτό το λόγο θα περιγράψω απλά με αφαιρετικό τρόπο γενικά τη χρήση και λειτουργικότητα ορισμένων κλάσεων και στοιχείων ώστε να μπορέσει να δημιουργηθεί μια γενική εικόνα της λειτουργικότητας. Οποιοσδήποτε θέλει να ρίξει

μια ματιά σε βάθος στις λειτουργικότητες των κλάσεων και των στοιχείων πρέπει απλά να ανατρέξει και να επισκεφτεί τον ιστότοπο <https://www.w3schools.com/> που είναι ένας εξαιρετικός ιστότοπος όσον αφορά την σύνταξη HTML και CSS και περιέχει όλες τις κλάσεις και τις ιδιότητες τους, τα στοιχεία και όλες τις ετικέτες και τα χαρακτηριστικά καθώς και έτοιμα παραδείγματα με κώδικα έτοιμο για χρήση. Επειδή το CSS έχει πραγματικά εκατοντάδες χιλιάδες «τεχνάσματα», αυτό που πρέπει να κάνει κανείς για να αποδώσει την επιθυμητή λειτουργικότητα στην ιστοσελίδα του είναι να πειραματιστεί με τις κλάσεις και τα στοιχεία και τις περισσότερες φορές να παραλλάξει τον έτοιμο κώδικα όπως αυτός επιθυμεί. Δεν χρειάζεται επί το πλείστον να γεννήσουμε κώδικα από την αρχή όπως δεν χρειάζεται να ξαναανακαλύψουμε τον τροχό.

Οπότε παραθέτω μερικά κομμάτια του κώδικα CSS του αρχείου “people.css” που έγραψα για να πετύχω την λειτουργικότητα – εμφάνιση των καρτελών των ατόμων.

Στην Εικόνα 26 φαίνεται ο κώδικας CSS για τον προσδιορισμό του κουτιού, που είναι και η «καρτέλα» του κάθε ατόμου της ομάδας του εργαστηρίου, στην αρχική του κατάσταση πριν την κίνηση που ενεργοποιεί την αλληλεπίδραση. Στο πάνω μέρος φαίνεται η ετικέτα “li” της κλάσης “one” που ορίζει ένα τέτοιο κουτί. Κάτω αριστερά φαίνεται η υποκλάση “wrapper” (περιτύλιγμα) της ετικέτας “li” της κλάσης “one” την οποία χρησιμοποίησα για «περιτύλιγμα» δηλαδή για να συμπεριλάβω μέσα σε αυτήν τα περιεχόμενα του κουτιού. Για αυτό έχει και ορισμένο μέγεθος το ίδιο με το κουτί. Κάτω δεξιά φαίνεται η ετικέτα “span” της ετικέτας “li” της κλάσης “one” που χρησιμοποίησα για το μπλε (ή άλλο χρώμα σε άλλα κουτιά – καρτέλες) πεδίο μέσα στον wrapper.

```

/* For the box */
.one > li {
    background: #FFFFFF;
    border: 1px solid rgba(150,150,150,0.29);
    -moz-box-shadow: 0px 0px 18px 0px rgba(103,85,85,0.39);
    box-shadow: 0px 0px 18px 0px rgba(103,85,85,0.39);
    width: 250px;
    height: 430px;
    display: inline-block;
    margin: 0 30px 30px;
    text-align: center;
}

/* To wrap contents
inside the box --
Same width and height*/
.one > li .wrapper {
    overflow: hidden;
    position: absolute;
    width: 250px;
    height: 430px;
}

/* For the SteelBlue
field inside the
wrapper (Professors) */
.one > li span {
    border-radius: 50%;
    height: 500px;
    width: 500px;
    background: #4682B4;
    display: block;
    position: absolute;
    bottom: -305px;
    left: -125px;
}

```

Εικόνα 26. Ο κώδικας CSS για το κουτί – «καρτέλα» ατόμου

Οπότε , στον κώδικα μου HTML, για να φτιάξω ένα τέτοιο κουτί έπρεπε απλά να ανοίξω μια ετικέτα “li”, μέσα της να ανοίξω μια κλάση “wrapper” και μέσα στην κλάση “wrapper” να χρησιμοποιήσω μαζί με ό,τι άλλο χρειάστηκε μια ετικέτα “span”, όπως φαίνεται στο στιγμιότυπο της Εικόνας 27. Έχω ήδη ανοίξει πριν από όλα αυτά μία κλάση “one” που δεν φαίνεται στο συγκεκριμένο στιγμιότυπο.

```

<!-- Start: George Stamoulis -->
<li class="transition">
    <div class="wrapper"> 
        <ul class="social">
            <li class="transition"><a href="http://inf-server.inf.uth.gr/~digogc
                target="_blank">
                <a href="https://www.e-ce.uth.gr/department/i
                target="_blank">
                <a href="https://www.linkedin.com/in/gestamol
                target="_blank">
                
            <h3>George Stamoulis
                <em>Professor <br>
                Lab Director <br>
                <em1><font color=" Azure">e-mail: georges@uth.gr</font> </em1>
                </em>
            </h3>
        </span>
    </div>
</li>
<!-- End: George Stamoulis -->

```

Εικόνα 27. Ο κώδικας HTML για το κουτί – «καρτέλα» ατόμου

Όπως φαίνεται στην Εικόνα 27 όλα «λαμβάνουν χώρα» μέσα σε μία κλάση “transition” την βασική λειτουργία της οποίας περιέγραψα νωρίτερα. Δεν είναι αρκετή όμως η κλάση “transition” για να προσδιορίσει την μετάβαση από την μία κατάσταση – εμφάνιση στην άλλη. Μόνο με αυτήν, δεν γίνεται τίποτα. Είναι αναγκαία η χρήση του επιλογέα “hover” (πέρασμα από πάνω) για να ορίσει ακριβώς αυτήν την λειτουργικότητα της μετάβασης όταν περνάμε τον δείκτη του ποντικιού πάνω από το κουτί. Αυτό φαίνεται στον κώδικα CSS της Εικόνας 28.

```
.one > li:hover img {
  margin-top: 30px;
}
.one > li:hover {
  -moz-box-shadow: 0px 0px 5px 0px rgba(103,85,85,0.25);
  box-shadow: 0px 0px 5px 0px rgba(103,85,85,0.25);
}
.one > li:hover span {
  border-radius: 0;
  -webkit-border-radius: 0;
  -moz-border-radius: 0;
  bottom: -400px;
}
.one > li span:hover {
  background: #4682B4;
  opacity: 0.9;
}
```

Εικόνα 28. Ο επιλογέας “hover” που συνεισφέρει σημαντικά στην λειτουργικότητα της μετάβασης

Όπως φαίνεται στην Εικόνα 27 στον κώδικα HTML χρειάστηκε και μία ξεχωριστή κλάση “social” για να προσδιορίσω γενικά την εμφάνιση και τοποθέτηση μέσα στο κουτί των εικόνων του LinkedIn, του scholar και του CV καθώς και μία μικρή λειτουργικότητα επιπλέον μετάβασης. Εκτός των άλλων, που μεταξύ τους είναι και ο προσδιορισμός του μεγέθους των στοιχείων και εικόνων (35 pixels) όπως φαίνεται στην τελευταία ετικέτα (“img”) της Εικόνας 29, καθώς και η τοποθέτηση τους μέσα στο κουτί με την χρήση ιδιοτήτων όπως το “padding”, το “display”, και το “margin”, χρησιμοποίησα και εδώ τον επιλογέα “hover”. Αν και εδώ ο επιλογέας “hover” το μόνο που κάνει είναι να μειώνει την αδιαφάνεια της εικονίτσας με την χρήση της ιδιότητας “opacity”, που σημαίνει ότι αφού έχουμε περάσει τον δείκτη του ποντικιού πάνω από το κουτί και έχουν εμφανιστεί πλέον οι συγκεκριμένες εικόνες, όταν τώρα περάσουμε τον δείκτη πάνω από μία από τις μικρές εικονίτσες, αυτήν θα «χρωματιστεί» ελαφρά με ένα «διαφανές» λευκό χρώμα. Πάλι θα φαίνεται η εικόνα αλλά αυτή η λειτουργία είναι απλά για να μας επισημαίνει ότι βρισκόμαστε σε εκείνο το σημείο και πάνω από τη συγκεκριμένη εικονίτσα, με τον

δείκτη του ποντικιού μας. Να αναφέρω απλά ότι οι εικόνες είναι ταυτόχρονα και σύνδεσμοι (links) κάτι που γίνεται πολύ εύκολα συμπεριλαμβάνοντας ανάμεσα στο άνοιγμα και στο κλείσιμο μιας ετικέτας <a> την εικόνα() που θέλουμε.

```
/*For CV and LinkedIn links...*/
.one .social {
  list-style: none;
  text-align: center;
  margin: 10px 0 0 0;
  padding: 10px;
}
.one .social li {
  opacity: 0;
  display: inline-block;
  padding: 5px;
}
.one > li:hover .social li {
  opacity: 0.5;
}
.one .social li a {
  cursor: pointer;
  display: block;
}
.one .social li a:hover {
  opacity: 0.5;
}
.one .social li img {
  width: 35px;
}
```

Εικόνα 29. Η κλάση “social” και μία ελάχιστη λειτουργικότητα του επιλογέα “hover” σε αυτήν την κλάση

Τα υπόλοιπα, όπως φαίνεται στην Εικόνα 27, είναι απλή σύνταξη κώδικα HTML χωρίς κάτι το ιδιαίτερο. Απλή χρήση των κλάσεων που προσδιορίστηκαν στον κώδικα CSS. Να επισημάνω πρώτον ότι δεν είναι αυτός ολόκληρος ο κώδικας CSS, γιατί χρησιμοποίησα ετικέτες με άλλα χρώματα π.χ. για τις καρτέλες των μεταπτυχιακών φοιτητών, οπότε χρειάστηκε ένας αντίστοιχος κώδικας CSS με προσδιορισμό άλλου χρώματος, και δεύτερον ότι όλος ο κώδικας CSS ασφαλώς βρίσκεται σε ένα ενιαίο αρχείο το οποίο εισήγαγα στο αρχείο HTML.

4.4 Η καρτέλα Academics

Η καρτέλα Academics περιέχει προς το παρόν μόνο την υποκαρτέλα – ιστοσελίδα Courses. Η ομάδα του εργαστηρίου με ενημέρωσε να αφήσω κενή την ιστοσελίδα Thesis Proposals, που είναι και αυτήν υποκαρτέλα της καρτέλας Academics, εως ότου συλλέξουν τις απαραίτητες πληροφορίες που χρειάζεται να περιέχει. Τότε θα ενημερωθεί και αυτήν η ιστοσελίδα.

4.4.1 Η ιστοσελίδα Courses

Η ιστοσελίδα Courses αποτελείται κι αυτή από μία εικόνα φόντου στην κεντρική της ενότητα και μία κεφαλίδα <h1>. Το ενδιαφέρον εδώ είναι οι κύκλοι καθένας από τους οποίους αντιστοιχεί σε ένα μάθημα και έχει μέσα του μία εικόνα και στο πέρασμα του δείκτη του ποντικιού πάνω από τον κύκλο εμφανίζονται δύο μικρά λευκά κυκλάκια και το περίβλημα του μεγάλου κύκλου όπως και των μικρών κύκλων χρωματίζεται πράσινο. Έχω βάλει τρεις συνδέσμους σε κάθε κύκλο : Έναν στο κείμενο μέσα στον κύκλο, το οποίο αντιστοιχεί στον κωδικό και την ονομασία του κάθε μαθήματος, και έναν σύνδεσμο σε καθένα από αυτά τα δύο μικρά κυκλάκια που εμφανίζονται. Κάθε σύνδεσμος από τους τρεις αντιστοιχεί στην ίδια διεύθυνση, απλά το έκανα αυτό ώστε σχεδόν όπου και να πατήσει κανείς, στον κύκλο και τα κυκλάκια, να μπορεί μεταβεί στην ιστοσελίδα του αντίστοιχου μαθήματος Αυτό βέβαια ήταν εύκολο να γίνει, αφού πρώτα έφτιαξα τον κύκλο. Όπως και στην καρτέλα People, έτσι κι εδώ, ο κώδικας για τον κύκλο είναι ένας (ορισμένος στο αρχείο “courses.css”) και στον κώδικα HTML κάθε κύκλος – μάθημα είναι σχεδόν αντιγραφή του πρώτου που χρειάστηκε να φτιάξω.

Ένα στιγμιότυπο της ιστοσελίδας φαίνεται στην Εικόνα 30. Έχω περάσει τον δείκτη του ποντικιού πάνω από το πρώτο μάθημα πάνω αριστερά, την στιγμή που τράβηξα το στιγμιότυπο, ώστε να φαίνεται αυτήν η επιπλέον λειτουργικότητα με τα δύο κυκλάκια και το πράσινο περίβλημα. Θα παραθέσω πρώτα τον κώδικα HTML (Εικόνα 31) που χρειάστηκε για το πρώτο μάθημα που εμφανίζεται πάνω αριστερά και είναι σχεδόν ίδιος για όλα τα υπόλοιπα μαθήματα. Οι μικροαλλαγές που χρειάστηκαν σε κάποια άλλα μαθήματα είναι απλά μερικές (επιπλέον) αλλαγές γραμμών (
) γιατί είχαν μεγαλύτερο όνομα και χρειαζόταν αυτήν η τροποποίηση για να υπάρχει μία σχετική ομοιομορφία εμφάνισης. Να επισημάνω ότι οι κύκλοι δεν έχουν δεδομένο προσδιορισμένο μέγεθος και το μέγεθος τους αυξομειώνεται ανάλογα με το κείμενο που περιέχουν όπως φαίνεται στην Εικόνα 30.



Εικόνα 30. Στιγμιότυπο της ιστοσελίδας Courses

Πρώτα , να αναφέρω ότι όπως ίσως είναι σχεδόν προφανές από την εμφάνιση της ιστοσελίδας χρησιμοποίησα πάλι την απαραίτητη σχεδίαση με Ανταπόκριση (Responsiveness). Χρησιμοποίησα έναν περιέκτη (container) και έφτιαξα μέσα του τέσσερις σειρές (rows) που η κάθε μία περιέχει τέσσερις στήλες, μία στήλη για κάθε μάθημα. Κάθε στήλη στη γενική της κλάση <div> έχει τις απαραίτητες τιμές ώστε να επιτυγχάνεται η Ανταπόκριση. Ακολουθεί παρόμοια σχεδίαση με αυτήν που περιέγραψα για την ενότητα 2 της αρχικής σελίδας στην ενότητα 4.2. Ένα μέρος της πρώτης σειράς και της πρώτης στήλης της πρώτης σειράς φαίνεται στον κώδικα για το μάθημα της Ψηφιακής Σχεδίασης στην Εικόνα 31. Επίσης κάθε κύκλος έχει μία δική του εικόνα φόντου μέσα του όπως φαίνεται στο στιγμιότυπο της Εικόνας 30.

Τα υπόλοιπα δεν είναι τίποτα άλλο από την χρήση των κλάσεων και των ετικετών που όρισα στο εξωτερικό CSS ("courses.css") και συνδυασμός αυτών ώστε να εμφανίζεται το αποτέλεσμα ενός κύκλου για το κάθε μάθημα. Δηλαδή τα απλά βήματα είναι τα εξής : Ανοίγω μία κλάση "circle", μέσα της ανοίγω έναν "wrapper" ώστε να τον χρησιμοποιήσω για το τι θα συμπεριλάβω μέσα στον κύκλο και μέσα στον "wrapper" βάζω ότι χρειάζεται. Μια εικόνα φόντου, το κείμενο – ονομασία του μαθήματος και μιά επιπλέον κλάση για να εμφανίζονται τα δύο μικρά κυκλάκια όταν κάνουμε hover δηλαδή περνάμε τον δείκτη του ποντικιού «πάνω» από τον

μεγάλο κύκλο. Και ασφαλώς τους συνδέσμους ενσωματωμένους στο κείμενο και στα μικρά κυκλάκια όπως περιέγραψα νωρίτερα.

```
<!--Start : Container-->
<div class="container" >
  <!-- Start : Row 1-->
  <div class="row" style="margin-top:50px;">
    <!-- Course 1: Digital Design-->
    <div class="col-lg-3 col-md-3 col-sm-6 col-xs-12" >
      <div class="circle">
        <div class="wrapper"
          style="background-image: url('http://inf-server.inf.ut
          background-size: contain;")>
          <a href="https://www.e-ce.uth.gr/studies/undergraduate
          target="_blank">
            <h2 style="color:Azure;" ><br>
              ECE 119 <br>
              Digital Design
            <br><br><br>
            </h2>
          </a>
          <ul class="circles">
            <li><a href="https://www.e-ce.uth.gr/studies/under
            target="_blank" ></a></li>
            <li><a href="https://www.e-ce.uth.gr/studies/under
            target="_blank" ></a></li>
          </ul>
        </div>
      </div>
    </div>
  <!-- End: Course 1 : Digital Design-->
```

Εικόνα 31. Ο κώδικας HTML για την δημιουργία ενός αλληλεπιδραστικού κύκλου για το μάθημα Ψηφιακή Σχεδίαση

Να περιγράψω τώρα των κώδικα CSS που χρειάστηκε για να αποδοθούν οι συγκεκριμένες λειτουργικότητες γιατί χωρίς αυτόν δεν γίνεται σχεδόν τίποτα. Ο κώδικας HTML δείχνει απλά την σχεδίαση οργάνωσης και σύνταξης – σύνδεσης των στοιχείων και ετικετών. Παραθέτω στις Εικόνες 32 – 33 τρία στιγμιότυπα του κώδικα, τα οποία συνιστούν σχεδόν ολόκληρο τον κώδικα του αρχείου courses.css”.

```
.circle{
  text-align: center;
}
.circle .wrapper{
  width: 100%;
  height: 100%;
  border-radius: 50%;
  border: 5px solid #d7d7d7;
  position: relative;
  margin: 0 auto;
  transition: all 0.5s ease 0s;
}
.circle:hover .wrapper{
  border-color: #1fc5b6;
}
.circle .wrapper h2{
  width: 100%;
  height: 100%;
  border-radius: 100%;
}
```

Εικόνα 32. Ο κώδικας CSS για τον ορισμό του μεγάλου κύκλου και της αντίστοιχης μετάβασης

Στην Εικόνα 32 φαίνεται ο κώδικας για τον ορισμό των κλάσεων που χρησιμοποιήσα για να φτιάξω τον μεγάλο κύκλο. Είναι αρκετά απλός. Η κλάση “circle” ορίζει απλά ένα στοιχείο – όποιο στοιχείο χρησιμοποιεί αυτήν την κλάση ασφαλώς – με κείμενο στοιχισμένο στο κέντρο. Η κλάση “wrapper” είναι πιο ενδιαφέρουσα γιατί ορίζει ένα στοιχείο με πλάτος και ύψος 100% που σημαίνει ότι αυξομειώνεται ανάλογα με τα περιεχόμενα του. Η ιδιότητα “border” ορίζει το λευκό αρχικά περίβλημα του κύκλου. Οι ιδιότητες που έχουν να κάνουν με την τοποθέτηση του περιεχομένου μέσα στον κύκλο όπως το “margin” και το “position” είναι σχετικά προφανείς. Η ιδιότητα “transition” χρειάζεται ασφαλώς για την μετάβαση από την αρχική κατάσταση στην επόμενη, η οποία ενεργοποιείται κάνοντας “hover” τον δείκτη του ποντικιού στον κύκλο. Έτσι, η κλάση “.circle:hover .wrapper” δείχνει πώς το περίβλημα χρωματίζεται όταν κάνουμε “hover” και αυτήν είναι και η μόνη λειτουργικότητα του επιλογέα “hover” όπως φαίνεται στο εσωτερικό της κλάσης, αφού το μόνο που κάνω είναι να χρωματίζω το περίβλημα (“border-color”) πράσινο (συγκεκριμένη απόχρωση πράσινου: #1fc5b6).

Αρκετό ενδιαφέρον έχουν και τα κυκλάκια τα οποία εμφανίζονται μόνο όταν κάνουμε hover (Εικόνα 33). Για να το δείξω αυτό στον κώδικα, απομόνωσα στην κλάση “circles” τις δύο ιδιότητες που χρειάζονται για να μην εμφανίζονται πριν το “hover” τα κυκλάκια, την “opacity: 0;” και την “transform: scale(0);”. Ο μηχανισμός απόδοσης βέβαια δεν διαχωρίζει τα κενά γραμμών που εγώ έβαλα και βλέπει τον κώδικα σαν ένα ενιαίο κείμενο που διαχωρίζεται μόνο από τις αγκύλες «{», «}» και τα ελληνικά ερωτηματικά «;». Ο συνδυασμός αυτών των δύο ιδιοτήτων (“opacity” και “scale”) και των αντίστοιχων τιμών τους είναι ένα έξυπνο τέχνασμα για να μην εμφανίζονται καθόλου τα κυκλάκια πριν το “hover”. Και η τιμή 0 του “opacity” και η τιμή 0 του “scale”. Χρειάζονται και τα δύο, ο συνδυασμός τους δηλαδή, για να έχουμε αυτήν την επιθυμητή απόδοση. Έτσι, στην κλάση “.circle:hover .circles” έπρεπε απλά να αλλάξω τις τιμές και των δύο ιδιοτήτων σε 1, ώστε κατά το “hover” να εμφανίζονται τα δύο κυκλάκια. Ενδιαφέρον έχει και η ιδιότητα “animation”, αλλά την λειτουργικότητα που αυτήν αποδίδει μπορεί κανείς να την δει μόνο αν επισκεφτεί την ιστοσελίδα γιατί αυτήν δεν μπορεί να αποδοθεί στο στιγμιότυπο της Εικόνας 30. Είναι μία επιπλέον «κίνηση» στην διάρκεια της μετάβασης

("transition"). Αυτά ήταν για την εμφάνιση των δύο μικρών κύκλων από την πρότερη «αόρατη» μορφή τους.

```
.circle .circles{
  margin: 0;
  padding: 0;
  list-style: none;
  width: 100%;
  position: absolute;
  top: 30%;

  opacity: 0;
  transform: scale(0);

  transition: all 0.5s ease 0s;
}
.circle:hover .circles{
  opacity: 1;
  transform: scale(1);
  animation: 0.5s ease 0s normal 1 bounce;
}
.circle .circles li{
  display: inline-block;
  border-radius: 50%;
}
.circle .circles li{
  display: inline-block;
  border-radius: 50%;
}
.circle .circles li:first-child{
  margin-left: -30px;
  float: left;
}
.circle .circles li:last-child{
  margin-right: -30px;
  float: right;
}
.circle .circles li a{
  display: block;
  width: 50px;
  height: 50px;
  line-height: 47px;
  border-radius: 50%;
  background-color: #fff;
  font-size: 25px;
  color: #1fc5b6;
  border: 3px solid #1fc5b6;
}
```

Εικόνα 33. Ο κώδικας CSS για τον ορισμό των μικρών κύκλων και της αντίστοιχης μετάβασης.

Στην Εικόνα 33 δεξιά φαίνονται οι κλάσεις και ετικέτες που χρειάστηκε να ορίσω ώστε να επιτύχω το μέγεθος, τον χρωματισμό και την τοποθέτηση των δύο μικρών κύκλων στα σημεία που ήθελα να εμφανίζονται μετά το "hover". Είναι σχετικά προφανείς και όχι ιδιαίτερα πολύπλοκες. Μερικές ιδιότητες από αυτές που εμφανίζονται δεν χρειάζονται, όπως π.χ. η ιδιότητα "font-size" στην ετικέτα <a>, απλά καθώς δημιουργούσα των κώδικα σκεφτόμουν να βάλω κείμενο μέσα στα κυκλάκια. Έτσι θα είχε χρησιμότητα και η συγκεκριμένη ιδιότητα.

Απλά να περιγράψω λίγο τον επιλογέα "first-child" και τον επιλογέα "second-child" που αυτά εμφανίζονται σε συνδυασμό με το που τοποθέτησα την κάθε ετικέτα "li" μέσα στην κλάση "circles" (δηλαδή τον «γονέα» τους) στον κώδικα HTML. Δηλαδή το πρώτο παιδί ("first-child"), εμφανίζεται εκεί που εμφανίζεται (ορισμένο με το "margin-left: -30px;" και το "float: left;") γιατί, όπως παρατηρούμε στον κώδικα HTML, στην κλάση "circles" που είναι ο γονέας της ετικέτας "li", η συγκεκριμένη ετικέτα "li" είναι το πρώτο στοιχείο που χρησιμοποιείται (και «ανοίγεται»), το πρώτο δηλαδή παιδί της κλάσης "circles". Με άλλα λόγια, το πρώτο στοιχείο της γονικής κλάσης εμφανίζεται εκεί που το ορίζω με τις ιδιότητες

“margin” και “float”. Το ίδιο ισχύει και για το δεύτερο παιδί. Αν δεν όριζα αυτήν την επιλογή πρώτου και δεύτερου παιδιού αντίστοιχα και προσέθετα ένα στοιχείο “li” όπως έκανα, τότε ένα κυκλάκι απλά θα εμφανιζόταν στο κέντρο του μεγάλου κύκλου. Και κάθε στοιχείο “li” που θα ήταν παιδί της κλάσης “circles” θα εμφανιζόταν κι αυτό στο κέντρο. Όσα κι αν ήταν τα παιδιά της. Το ένα πάνω στο άλλο δηλαδή.

Να προσθέσω κάτι τελευταίο για την ιστοσελίδα Courses που έχει ένα μικρό ενδιαφέρον και καθορίζει την απόσταση που θα έχει ο κάθε κύκλος από τον άλλον σε προβολή μικρών και πολύ μικρών συσκευών, όπου θα εμφανίζονται ο ένας πάνω από τον άλλον (σε πολύ μικρές συσκευές) και δύο κύκλοι πάνω από άλλους δύο κύκλους (σε μικρές συσκευές) εξαιτίας του σχεδιασμού ανταπόκρισης που περιέγραψα σχεδόν στην αρχή αυτής της ενότητας και φαίνεται στην τρίτη σειρά του κώδικα στην Εικόνα 31. Η απόσταση μεταξύ των επιπέδων των κύκλων ορίζεται πολύ απλά με την ιδιότητα “margin-bottom” όπως φαίνεται στην Εικόνα 34.

```
@media screen and (max-width: 990px) {  
  .circle{ margin-bottom: 20px; }  
}
```

Εικόνα 34. Ορισμός της απόστασης των κύκλων σε μικρές και πολύ μικρές οθόνες.

4.5 Η καρτέλα Publications

Η καρτέλα Publications αποτελείται από πέντε υποκαρτέλες – ιστοσελίδες. Την International Conference Publications για δημοσιεύσεις της ομάδας του εργαστηρίου σε διεθνή συνέδρια, την Journals για δημοσιεύσεις σε επιστημονικά περιοδικά, την Posters in International Conferences για ανάρτηση παρουσιάσεων έρευνας σε επιστημονικά συνέδρια, την Patents για Πατέντες και την Technical Reports για Τεχνικές Αναφορές. Επειδή το γενικό θέμα είναι το ίδιο για όλες, οι δημοσιεύσεις, όλες οι σελίδες είναι σχεδιασμένες πανομοιότυπα, απλά αλλάζουν τα περιεχόμενα, οι χρονολογίες, το κείμενο και οι εικόνες. Το εξωτερικό αρχείο CSS

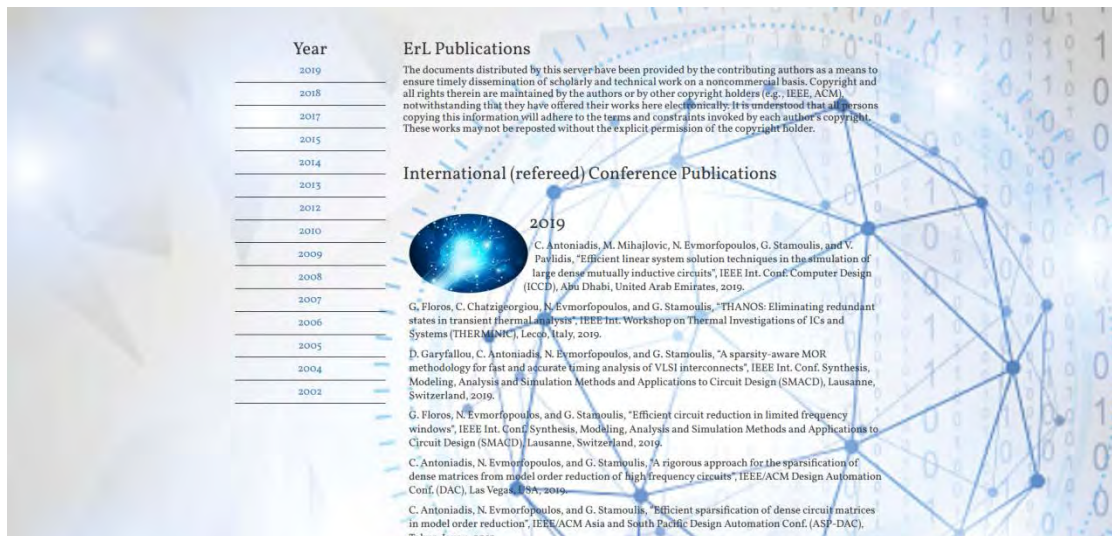
(Publications.css) είναι ένα (το ίδιο) για όλες. Έχουν ακόμα και την ίδια εικόνα φόντου. Οπότε στην παρούσα ενότητα θα παρουσιάσω μόνο την πρώτη.

4.5.1 Η ιστοσελίδα International Conference Publications

Η Εικόνα 35 δείχνει ένα στιγμιότυπο της ιστοσελίδας, όπως αυτή φαίνεται περίπου στην αρχή. Η σχεδίαση της είναι σχετικά απλή. Η ιστοσελίδα χωρίζεται σε δύο στήλες, πάλι με την ανάλογη σχεδίαση με Ανταπόκριση. Οι δύο στήλες ασφαλώς βρίσκονται μέσα σε έναν περιέκτη (container). Η πρώτη στήλη (η αριστερή) αποτελείται από τις χρονολογίες δημοσιεύσεων. Κάθε χρονολογία είναι και ένας σύνδεσμος εσωτερικά στην σελίδα που παραπέμπει και μεταβαίνει τον χρήστη στο σημείο της σελίδας όπου βρίσκονται οι δημοσιεύσεις της συγκεκριμένης χρονολογίας. Αυτό γίνεται πολύ εύκολα. Για παράδειγμα ο εσωτερικός σύνδεσμος για τη μετάβαση στο σημείο όπου παρατίθενται οι δημοσιεύσεις για το 2014 λειτουργεί ως εξής :

Η γραμμή – λέξη 2014 στη στήλη των χρονολογιών γράφεται με την γραμμή κώδικα “2014 <hr>” (μέσα σε μία κεφαλίδα). Έτσι η λέξη 2014 είναι τώρα και ένας εσωτερικός σύνδεσμος που παραπέμπει σε κάποιο στοιχείο της ιστοσελίδας που έχει δηλωμένο ως αναγνωριστικό της το 2014, δηλαδή έχει “id = 2014”. Αυτό το στοιχείο είναι μια κεφαλίδα “<h2>” που βρίσκεται στη δεξιά στήλη στην υποενότητα όπου παρατίθενται οι δημοσιεύσεις για το 2014.

Η δεξιά στήλη συνεχίζει προς τα κάτω με ένα ίδιο μοτίβο, όπου αποτελείται από δημοσιεύσεις προηγούμενων χρόνων, το κείμενο για κάθε μία από τις οποίες ξεκινά, όπως το κείμενο για τη χρονιά 2019 που φαίνεται στην Εικόνα 35, και στοιχίζεται δεξιά από μία κυκλική εικόνα με μία προσαρμογή ώστε εως ότου το μέρος που καταλαμβάνει η εικόνα τελειώσει, αυτό να καμπυλώνεται δίπλα της.



Εικόνα 35. Στιγμιότυπο της ιστοσελίδας International Conference Publications

Ο κώδικας HTML δεν είναι καθόλου πολύπλοκος και στην ουσία ο κώδικας (για τη δεξιά στήλη) που χρειάζεται για κάθε χρονιά είναι απλά αντιγραφή του πρώτου, μόνο που αλλάζει η εικόνα και το κείμενο. Ο κώδικας για την αριστερή στήλη, με τις χρονιές δεν χρειάζεται περαιτέρω επεξήγηση από αυτήν που ήδη έδωσα εως τώρα. Τον παραθέτω στην Εικόνα 36.

```

<!-- 1st column : Year's -->
<div class="col-lg-3 col-md-3 col-sm-10 col-xs-10 text-center">
  <h2 style = "font-family:Vollkorn,arial,Helvetica;" >Year</h2>
  <h4 style = "font-family:Vollkorn,arial,Helvetica;" >
    <a href="#2019">2019 </a> <hr>
    <a href="#2018">2018 </a> <hr>
    <a href="#2017">2017 </a> <hr>
    <a href="#2015">2015 </a> <hr>
    <a href="#2014">2014 </a> <hr>
    <a href="#2013">2013 </a> <hr>
    <a href="#2012">2012 </a> <hr>
    <a href="#2010">2010 </a> <hr>
    <a href="#2009">2009 </a> <hr>
    <a href="#2008">2008 </a> <hr>
    <a href="#2007">2007 </a> <hr>
    <a href="#2006">2006 </a> <hr>
    <a href="#2005">2005 </a> <hr>
    <a href="#2004">2004 </a> <hr>
    <a href="#2002">2002 </a> <hr>
  </h4>
</div>
<!-- End : 1st column : Year's -->

```

Εικόνα 36. Η στοίχιση των χρονολογιών σε μια στήλη

Στην Εικόνα 37 παραθέτω ένα στιγμιότυπο του κώδικα HTML για τη χρονιά 2019 (δεξιά στήλη). Όπως περιέγραψα προηγουμένως, ο κώδικας για κάθε επόμενη χρονιά βασίζεται στο ίδιο ακριβώς μοτίβο. Παραλείπω την αρχή του κώδικα που χρειάζεται για την δεξιά στήλη που είναι ασφαλώς ο ορισμός των τιμών της στήλης

με σκοπό την Ανταπόκριση και μία κεφαλίδα “<h4>” που περιέχει το αρχικό κείμενο πριν από την παράθεση των δημοσιεύσεων. Πιο συγκεκριμένα ο κώδικας της Εικόνας 37 είναι για την πρώτη δημοσίευση της χρονιάς 2019 που φαίνεται στην Εικόνα 35. Όπως φαίνεται στον κώδικα το κείμενο για την δημοσίευση είναι μέσα σε μια κεφαλίδα “<h4>”. Κάθε επόμενη δημοσίευση της ίδιας χρονιάς είναι ακόμα μια κεφαλίδα “<h4>” σε σειρά κάτω από την πρώτη. Όπως ανέφερα προηγουμένως η σχεδίαση της ιστοσελίδας είναι σχετικά απλή και δεν πιστεύω ότι ο κώδικας HTML χρειάζεται περαιτέρω επεξήγηση.

```
<div id="circle-shape">
  
  <h2 id="2019" style = "font-family:Vollkorn,arial,Helvetica;"> 2019</h2>
  <h4 style = "font-family:Vollkorn,arial,Helvetica;">
    C. Antoniadis, M. Mihajlovic, N. Evmorfopoulos, G. Stamoulis, and V. Pavlidis,
    "Efficient linear system solution techniques in the simulation of large dense
    mutually inductive circuits", IEEE Int. Conf. Computer Design (ICCD), Abu Dhabi,
    United Arab Emirates, 2019.
  </h4>
```

Εικόνα 37. Στιγμιότυπο κώδικα HTML της δεξιάς στήλης (των δημοσιεύσεων)

Ο κώδικας CSS για το αναγνωριστικό “circle-shape” και την κλάση “curve” που χρησιμοποιώ όπως δείχνει η Εικόνα 37 είναι και αυτός ιδιαίτερα απλός και αρκετά μικρός και τον παραθέτω στην Εικόνα 38.

```
#circle-shape {
  margin: 1rem;
}

#circle-shape h4 {
  line-height: 1.3;
}

#circle-shape .curve {
  width: 25%; height: auto;
  min-width: 150px;
  float: left;
  margin-right: 9rem;
  border-radius: 100%;
  shape-outside: circle();
}
```

Εικόνα 38. Η ενσωματωμένη συνάρτηση circle() του CSS

Πιστεύω ότι είναι όλες σχεδόν οι ιδιότητες σχετικά προφανείς και τις περισσότερες τις έχω ήδη περιγράψει ή αναφέρει σε προηγούμενες ενότητες και περιγραφές

κώδικα. Αυτή που είναι νέα και έχει το ιδιαίτερο ενδιαφέρον εδώ, είναι η ενσωματωμένη συνάρτηση (inbuilt function) “circle()” του CSS, που χρησιμοποιείται για τη δημιουργία αιωρούμενου κειμένου γύρω από την κυκλική εικόνα και αποδίδει ακριβώς αυτήν την εμφάνιση που περιέγραψα νωρίτερα. Πρέπει να συνδυαστεί με την ιδιότητα “shape-outside” για να αποδοθεί η συγκεκριμένη λειτουργικότητά της για να οριστεί το κυκλικό σχήμα. Όπως ανέφερα είναι ενσωματωμένη συνάρτηση και δεν χρειάζεται να την προγραμματίσουμε ούτε χρίζει περαιτέρω επεξήγηση. Απλά την χρησιμοποιούμε. Να αναφέρω μόνο ότι όπως ακριβώς ορίζουμε αυτό το κυκλικό σχήμα εμφάνισης με την ιδιότητα “shape-outside” μπορούμε να ορίσουμε και άλλα σχήματα – εμφανίσεις εκτός του κυκλικού και να χρησιμοποιήσουμε αντί της συνάρτησης “circle()”, τις συναρτήσεις “inset()” για ορθογώνιο σχήμα, “ellipse()” για σχήμα έλλειψης και τη συνάρτηση “polygon()” για να σχηματίσουμε ό,τι σχήμα επιθυμούμε.

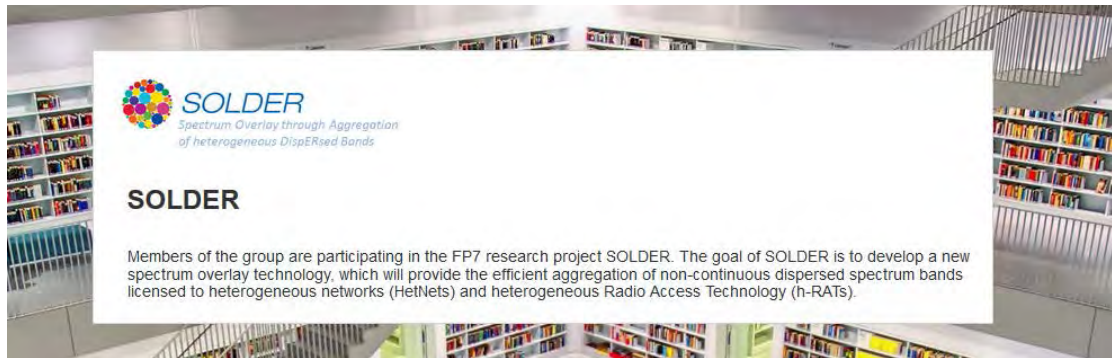
4.6 Η καρτέλα Research

Η καρτέλα Research περιλαμβάνει προς το παρόν την ιστοσελίδα R&D Projects. Η ομάδα του εργαστηρίου θα με ενημερώσει σχετικά για να ανανεώσουμε την προς το παρόν κενή καρτέλα Collaborations.

4.6.1 Η ιστοσελίδα R&D Projects

Η ιστοσελίδα R&D Projects έχει μία κεντρική ενότητα (με id="section1") πάνω από την γενική ενότητα “Contact” που είναι ίδια σε όλες τις ιστοσελίδες του ιστότοπου. Αυτή έχει κλασικά μία εικόνα φόντου και μια επικεφαλίδα. Το ιδιαίτερο στοιχείο είναι εδώ οι δύο (προς το παρόν) «κάρτες». Κάθε κάρτα αντιστοιχεί και σε ένα συγκεκριμένο Έργο Έρευνας και Ανάπτυξης. Παραθέτω στην Εικόνα 39 ένα στιγμιότυπο ενός μέρους της ιστοσελίδας που περιλαμβάνει την πρώτη κάρτα. Ενδιαφέρον έχει και εδώ ο εξωτερικός κώδικας CSS (“research”.css). Η σύνταξη HTML είναι απλά χρήση των ορισμένων κλάσεων σε αυτό το αρχείο. Οπότε θα ξεκινήσω περιγράφοντας το εξωτερικό CSS. Στην Εικόνα 40 φαίνονται οι κλάσεις που όρισα για την κατασκευή της κάρτας. Όπως και στα παραδείγματα των

προηγούμενων ενοτήτων (καρτέλες – κύκλοι – κυκλικό σχήμα) ο κώδικας της πρώτης κάρτας είναι το «πρότυπο» το οποίο απλά αντέγραφα για την επόμενη κάρτα , απλά αλλάζει η εικόνα και το περιεχόμενο κειμένου.



Εικόνα 39. Στιγμιότυπο μίας κάρτας της ιστοσελίδας R&D Projects

```
.card
{
  background:#fff;
  width:800px;
  padding:0;
  margin-bottom:50px;
  position:relative;
  border:1px solid #ddd
}
.card .card-section
{
  background:#fff;
  box-shadow:0px 0px 81px 0 rgba(47,47,47,0.1);
}
.card .card-section-image
{
  width:50%;
}
.card .card-section-image img
{
  margin-left:30px;
  margin-top:20px;
  width:60%; height:auto;
  transition:all .5s ease-in-out;
}
.card .card-section:hover .card-section-image img
{
  transform:scale(1.2) rotate(0deg) }

```

Εικόνα 40. Ορισμός των κλάσεων για την κατασκευή της κάρτας

Η πρώτη και γενική κλάση “card” ορίζει το μέγεθος (width) , το χρώμα (background), το μέγεθος και χρώμα του περιβλήματος (border) και την απόσταση (margin-bottom) της κάρτας από την επόμενη της. Μετά από όσα εως τώρα έχω περιγράψει δεν πιστεύω ότι χρειάζεται περαιτέρω εξήγηση.

Η κλάση “card-section” είναι αυτή που περιέχει μέσα της τα στοιχεία της κάρτας, όπως ακριβώς έκανε η κλάση “wrapper” στις ιστοσελίδες People και Courses για τις

καρτέλες και τους κύκλους αντίστοιχα. Η ιδιότητα “box-shadow” δεν είναι εμφανώς ορατή στην ιστοσελίδα γιατί η σκιά χρώματος που προσδίδει αυτήν η ιδιότητα απορροφάται από την εικόνα φόντου. Απλά είχα σχεδιάσει την κάρτα προτού να ξέρω τι φόντο θα χρησιμοποιήσω στην συγκεκριμένη ενότητα.

Η κλάση “card-section-image” είναι για την εικόνα της κάρτας. Ορίζει το σχετικό μέγεθος που θα καταλαμβάνει η ενότητα που θα περιέχει την εικόνα (logo) σε κάθε κάρτα. Μετά από πειράματα κατέληξα σε “width : 50%”.

Η ετικέτα “img” ορίζει την ίδια την εικόνα που βρίσκεται μέσα στην ενότητα της εικόνας της κάρτας. Όλες σχεδόν οι ιδιότητες είναι αρκετά προφανείς : Η τοποθέτηση – τα όρια της εικόνας μέσα σε αυτήν την ενότητα (margin-left και margin-top) και το σχετικό μέγεθος της (width και height). Όπως προηγουμένως και εδώ κατέληξα σε αυτές τις τιμές μετά από αρκετούς πειραματισμούς ώστε το αποτέλεσμα να φαίνεται όμορφο.

Ενδιαφέρον εδώ έχει ή όχι και τόσο προφανής ιδιότητα “transform: scale(1.2) rotate(0deg)” (στον ορισμό της επιλογής “hover”) που σε συνδυασμό με την ιδιότητα της μετάβασης (“transition”) της ετικέτας “img” που έχω αναφέρει και περιγράψει λίγο στην ενότητα 4.3, προσδίδουν στην εικόνα της κάρτας μία επιθυμητή αλληλεπιδραστικότητα. Να αναφέρω ότι χωρίς την ιδιότητα transition η μετάβαση θα γινόταν εντελώς «απότομα» και το αποτέλεσμα στο ανθρώπινο μάτι δεν θα ήταν όμορφο. Η ιδιότητα “transform” που δηλώνει μετασχηματισμό, μετασχηματίζει την εικόνα σε κλίμακα 1.2, δηλαδή κατά το πέρασμα του δείκτη του ποντικιού από πάνω από την κάρτα, η εικόνα μεγαλώνει κατά 1.2 φορές. Αυτό βέβαια χρειάζεται τον επιλογέα “hover” όπως και στις καρτέλες και τους κύκλους των προηγούμενων ιστοσελίδων. Δοκίμασα να λάβω ένα στιγμιότυπο της κατάστασης μετά την μετάβασή της αλλά δεν είναι εμφανής γι αυτό και δεν παραθέτω την εικόνα. Πρέπει κάποιος να επισκεφτεί την ιστοσελίδα για να παρατηρήσει αυτόν τον μετασχηματισμό. Η επιλογή “rotate(0deg)” δεν κάνει ουσιαστικά τίποτα στο συγκεκριμένο παράδειγμα γιατί η τιμή της είναι μηδέν (0). Οποιαδήποτε άλλη τιμή δώσουμε σε αυτήν την επιλογή, θα αποδώσει στην διεργασία μετάβασης μία περιστροφή, ανάλογα με την τιμή αυτή της γωνίας.

Αυτά όσον αφορά το τμήμα της εικόνας και την ίδια την εικόνα. Το υπόλοιπο της κάρτας είναι μια ενότητα όπου απλά παρατίθεται κείμενο. Ο κώδικας CSS είναι ορισμός των ορίων του κειμένου, της στοίχισης του και της εμφάνισης του και δεν βρίσκω ιδιαίτερο νόημα να τον παραθέσω εδώ. Επειδή παραθέτω στην Εικόνα 41 τον κώδικα HTML για την δημιουργία μίας κάρτας να αναφέρω απλά ότι κώδικας CSS για το κείμενο είναι απλά μία υποκλάση της κλάσης “card”, που ονόμασα “card-desc” και μέσα της περιέχει ακόμα μία κλάση “card-info”. Μέσα τους όπως ανέφερα είναι ορισμένα τα χαρακτηριστικά του κειμένου. Θα μπορούσε εύκολα να επιτευχθεί το ίδιο αποτέλεσμα με εσωτερικό CSS για το συγκεκριμένη ενότητα.

```

<div class="card text-left" >
  <div class="card-section">
    <div class="card-section-image">
      <a href="https://www.isi.gr/project/spectrum-overlay-through-aggregation"
      target="_blank">
        
      </a>
    </div>
    <div class="card-desc">
      <div class="title">
        <h3> <strong>SOLDER</strong> </h3>
      </div>
      <div class="card-info">
        <h5>
          Members of the group are participating in the FP7 research project SOLDER. The goal of SOLDER is to develop a new spectrum overlay technology which will provide the efficient aggregation of non-continuous spectrum bands licensed to heterogeneous networks (HetNets) using 4G Radio Access Technology (4G-RATs).
        </h5>
      </div>
    </div>
  </div>
</div>

```

Εικόνα 41. Ο κώδικας HTML για την κατασκευή μίας κάρτας

Δεν πιστεύω ότι ο κώδικας HTML χρειάζεται περαιτέρω περιγραφή. Είναι ιδιαίτερα προφανής. Παρόμοιες λειτουργικότητες κώδικα έχω ήδη περιγράψει σε προηγούμενες ενότητες.

Αυτό που θέλω να επισημάνω είναι ένα επιπλέον κομμάτι CSS που χρειάστηκε ώστε η κάρτα να μην έχει δεδομένο – ορισμένο πλάτος 800 pixels όταν η ιστοσελίδα

προβάλλεται σε μικρές και πολύ μικρές συσκευές, γιατί αυτό θα είχε ως αποτέλεσμα να βλέπουμε ένα μικρό κομμάτι της κάρτας στο κινητό μας και η υπολοιπή κάρτα να εκτείνονταν δεξιά και αριστερά και να πρέπει να κυλίσουμε την σελίδα δεξιά και αριστερά για να την δούμε. Αυτό «χαλούσε» την καλαίσθητη εμφάνιση της ιστοσελίδας. Έτσι, όρισα ξανά την γενική κλάση της κάρτας (“card”) για τέτοιες συσκευές. Η τροποποιήσεις που χρειάστηκε να κάνω ήταν να θέσω το πλάτος της κάρτας σε αυτόματο (width:auto) και να θεσω και τα όρια (margin-left και margin-right) τα οποία θα απέχει η κάρτα από την οθόνη της συσκευής, όπως φαίνεται στην Εικόνα 42. Έτσι το πλάτος της κάρτας ρυθμίζεται ανάλογα με το μέγεθος της οθόνης για τέτοιες συσκευές.

```
@media only screen and (max-width: 900px) {  
  .card  
  {  
    width:auto;  
    margin-left:20px; margin-right:20px;  
    background:#fff; padding:0;  
    margin-bottom:50px;  
    position:relative;  
    border:1px solid #ddd  
  }  
}
```

Εικόνα 42. Ορισμός του πλάτους και των ορίων της κάρτας ειδικά για μικρές και πολύ μικρές συσκευές

Ο λόγος που δεν έθεσα αρχικά και γενικά την τιμή του πλάτους σε “auto” στην γενική κλάση “card”, ήταν γιατί έτσι η κάρτα όντως θα ανταποκρινόταν σε αυτήν την ιδιότητα και η κάρτα θα καταλάμβανε σχεδόν ολόκληρο το πλάτος της ιστοσελίδας. Θα φαινόταν δηλαδή πάρα πολύ μεγάλη.

ΚΕΦΑΛΑΙΟ 5 : ΣΥΜΠΕΡΑΣΜΑΤΑ

Το Διαδίκτυο (Internet) και ο Παγκόσμιος Ιστός (World Wide Web), παρότι άμεσα συνδεδεμένα μεταξύ τους είναι δύο διακριτά πεδία – σύνολα της επιστήμης των ηλεκτρονικών υπολογιστών και των δικτύων. Ο Παγκόσμιος Ιστός που αποτελείται από ιστότοπους και ξεχωριστές ιστοσελίδες είναι στην ουσία ένα υποσύνολο του Διαδικτύου. Για να λειτουργήσει το Διαδίκτυο και να επικοινωνήσουν ηλεκτρονικοί υπολογιστές και γενικά (ηλεκτρονικές) συσκευές μεταξύ τους έχουν οριστεί διάφορα πρότυπα, πρωτόκολλα και μοντέλα επικοινωνίας ώστε να επιτυγχάνεται η μεταφορά πληροφορίας και δεδομένων από το ένα σημείο στο άλλο ή μεταξύ πολλών σημείων ταυτόχρονα ή παράλληλα, σύγχρονα ή ολικά ασύγχρονα. Εδώ η λέξη σημείο μπορεί να σημαίνει έναν διακεκριμένο υπολογιστή σε κάποιο μέρος του πλανήτη ή έναν κόμβο στον οποίον συνδέονται αρκετοί υπολογιστές που αποτελεί ένα εσωτερικό δίκτυο σε κάποιο άλλο μέρος του πλανήτη είτε σημαίνει απλά κάποιον δρομολογητή μεταξύ δύο σημείων – υπολογιστών που προσπαθούν να επικοινωνήσουν μεταξύ τους. Και βέβαια υπάρχει και το φυσικό επίπεδο μετάδοσης της πληροφορίας, όπου η πληροφορία μετατρέπεται σε ηλεκτρομαγνητικά σήματα και μεταφέρεται μέσα από καλώδια, δορυφόρους ή άλλα μέσα μετάδοσης. Αυτό βέβαια είναι πέραν από τα όρια της παρούσας διπλωματικής εργασίας. Έτσι, καθώς το διαδίκτυο, σε ένα αφαιρετικό και διαφανές επίπεδο είναι η σύνδεση και επικοινωνία διάφορων υπολογιστών μεταξύ τους με σκοπό την ανταλλαγή πληροφορίας, ο Παγκόσμιος Ιστός αποτελείται από ιστότοπους ή και μοναδικές ιστοσελίδες, οι οποίες συνήθως είναι προσβάσιμες από τον καθένα και είναι «αναρτημένες» σε κάποια τοποθεσία. Η «ανάρτηση» αυτήν μπορεί να φαίνεται απλή αφού το μόνο που χρειάζεται για να επισκεφτεί ο χρήστης την ιστοσελίδα είναι η διεύθυνση URL της, αλλά από πίσω ή καλύτερα από «κάτω» διεκπεραιώνεται μια λειτουργικότητα που περιλαμβάνει τον «σεβασμό» στα πρωτόκολλα μεταφοράς καθώς και επανειλημμένες προσπάθειες επικοινωνίας από σημείο σε σημείο μέσα από το δίκτυο για να ληφθεί το αρχείο που τελικά περιέχει την πληροφορία η οποία χρησιμεύει για να εμφανιστεί η ιστοσελίδα στο πρόγραμμα περιήγησης που χρησιμοποιεί ο χρήστης, αλλά και αρκετούς άλλους επιπλέον μηχανισμούς . Το αρχείο αυτό δεν είναι αναρτημένο σε κάποια αόρατη

τοποθεσία όπου βρίσκονται όλες οι τοποθεσίες ιστοσελίδων, αλλά βρίσκεται σε έναν ή περισσότερους διακεκριμένους υπολογιστές που στην γενική περίπτωση ονομάζονται εξυπηρετητές. Το αρχείο αυτό δεν είναι τίποτα άλλο από κώδικας που τις περισσότερες φορές είναι γραμμένος με το χέρι. Όλα τα στοιχεία που αποτελούν μία ιστοσελίδα, από το κείμενο μέχρι τα γραφικά στοιχεία όπως εικόνες και την «κίνηση» διάφορων στοιχείων κατά την εμπειρία περιήγησης είναι αποτέλεσμα απόδοσης κώδικα. Τα προγράμματα περιήγησης περιέχουν ενσωματωμένους μηχανισμούς απόδοσης για να μπορέσει ο κώδικας να μετατραπεί τελικά σε κάποια επιθυμητή εμφάνιση. Χρειάζεται η απαραίτητη σχεδίαση της ιστοσελίδας, από τον τρόπο με τον οποίο θέλουμε να γίνεται τελικά αυτή η απόδοση από άποψη εμφάνισης και γραφικότητας έως τον τρόπο με τον οποίο συνδέεται με συνδέσμους είτε εσωτερικά με άλλες σελίδες που ανήκουν στον ίδιο ιστότοπο ή με άλλες εξωτερικές ιστοσελίδες. Υπάρχουν βέβαια έτοιμες εφαρμογές και εργαλεία για τα οποία συνήθως κανείς πρέπει να πληρώσει και μέσω των οποίων η σύνθεση ή σχεδίαση μίας ιστοσελίδας γίνεται αρκετά πιο εύκολα, μετακινώντας για παράδειγμα και τοποθετώντας μια εικόνα στο σημείο όπου θέλουμε να εμφανίζεται μια εικόνα στην ιστοσελίδα μας χωρίς να γράψουμε τον απαραίτητο κώδικα. Αυτό βέβαια αντιστοιχεί πάλι σε κώδικα, παρότι η διεργασία γίνεται αυτόματα, η συγκεκριμένη εφαρμογή δηλαδή παράγει αυτόματα τον κώδικα τον οποίον χρειάζεται για να αποδοθεί η συγκεκριμένη λειτουργικότητα. Ακόμα και αυτές οι εφαρμογές όμως χρίζουν την απαραίτητη εξοικείωση μαζί τους και δεν είναι απλά παιχνιδάκια. Ασφαλώς για την παρούσα εργασία δεν χρησιμοποίησα καμία τέτοια εφαρμογή γιατί κάτι τέτοιο δεν θε είχε και κανένα νόημα, καθώς τότε δεν θα περιέγραφα την σύνταξη κώδικα αλλά την λειτουργικότητα της εφαρμογής. Κλείνοντας, να αναφέρω ότι η σύνταξη κώδικα, παρότι υπάρχουν έτοιμες συναρτήσεις, κλάσεις και στοιχεία, που ασφαλώς δεν χρειάζεται κανένας να δημιουργήσει από το μηδέν, χρειάζεται ιδιαίτερη αφιέρωση χρόνου και πρέπει κανείς να πειραματιστεί αρκετά για να μπορέσει να επιτύχει ακόμα και μία ελάχιστη επιθυμητή απόδοση στην ιστοσελίδα του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. N. Robbins , *Learning Web Design, A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, Sebastopol, California, O'Reilly Media, Inc., 2018.
- [2] N. Cooper, *Build Your Own Website*, San Francisco, California, No Starch Press, Inc., 2014.
- [3] "w3schools.com," [Online]. Available: <https://www.w3schools.com/>
- [4] "getbootstrap.com," [Online]. Available: <https://getbootstrap.com/>
- [5] "developer.mozilla.org/en-US/docs/Web," [Online]. Available : <https://developer.mozilla.org/en-US/docs/Web>
<https://developers.google.com/web/tools/chrome-devtools/>
- [6] "developers.google.com/web/tools/chrome-devtools," [Online]. Available: <https://developers.google.com/web/tools/chrome-devtools/>
- [7] "w3docs.com ," [Online]. Available: <https://www.w3docs.com/>
- [8] "geeksforgeeks.org," [Online]. Available: <https://www.geeksforgeeks.org/>
- [9] "stackoverflow.com," [Online]. Available: <https://stackoverflow.com/>
- [10] "css-tricks.com ," [Online]. Available: <https://css-tricks.com/>
- [11] "bootsnipp.com," [Online]. Available: <https://bootsnipp.com/>
- [12] "coderwall.com," [Online]. Available: <https://coderwall.com/>