ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

Σχολή Θετικών Επιστημών Τμήμα Πληροφορικής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Δημιουργία διαδραστικής διαδικτυακής εφαρμογής τροποποίησης, επεξεργασίας και αποθήκευσης κωδίκων VHDL

Εκπόνηση εργασίας: Ρούτσης Κωνσταντίνος Α.Μ.:2113122
Επιβλέπων Καθηγητής: Δαδαλιάρης Αντώνης

ΠΕΡΙΛΗΨΗ

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η δημιουργία διαδραστικής διαδικτυακής εφαρμογής για την τροποποίηση, την επεξεργασία και την αποθήκευση κωδίκων VHDL.Το όνομα της εφαρμογής - Ιστοσελίδας είναι VHDL Companion και αναφέρεται σε χρήστες που θέλουν να ξεκινήσουν να μαθαίνουν VHDL αλλά και σε χρήστες πιο έμπειρους με την γλώσσα οι οποίοι θέλουν να αποθηκεύσουν, να επεξεργαστούν και να τροποποιήσουν τους κωδικές τους με γρήγορο, εύκολο και δυναμικό τρόπο. Αν και αναφέρεται κυρίως σε προγραμματιστές η σχεδίαση της εφαρμογής έγινε με τέτοιο τρόπο ώστε να είναι κατανοητή και ευκολόχρηστη από όλους. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν γλώσσα σήμανσης υπερκειμένου HTML, γλώσσα φύλλου ύφους CSS, καθώς και γλώσσες προγραμματισμού PHP και Javascript. Τέλος χρησιμοποιήθηκε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων MySQL για την αποθήκευση και την ανάκτηση πληροφοριών για τους χρήστες και τα αρχεία τους από μια βάση δεδομένων.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Η παρούσα πτυχιακή εργασία έχει σαν στόχο την ανάπτυξη μιας δικτυακής εφαρμογής για την τροποποίηση , την επεξεργασία και την αποθήκευση κωδίκων VHDL. Πρόκειται για μια δυναμική διαδραστική ιστοσελίδα που απευθύνεται κυρίως σε προγραμματιστές της γλώσσας VHDL η σε χρήστες που επιθυμούν να τη μάθουν τη γλώσσα.

Η εφαρμογή εισάγει τους χρήστες στη VHDL παρέχοντας τους inks που θα τους παραπέμψουν σε μερικά από τα καλύτερα tutorial, παραδείγματα και μαθήματα που υπάρχουν στο διαδίκτυο. Προσφέρει links με βίντεο tutorial, λίστα με βιβλία, καθώς και λίστα με τους σημαντικότερους προσομοιωτές αλλά και άλλου τύπου εφαρμογές που σχετίζονται με τη VHDL. Επιτρέπει στους χρήστες να ανεβάσουν και να αποθηκεύσουν τους κώδικές τους και να έχουν πρόσβαση σε αυτούς με έναν ασφαλή και συνεπή τρόπο. Τέλος παρέχει στους προγραμματιστές λειτουργικότητες για την επεξεργασία και τροποποίηση των κωδίκων τους με ευκολόχρηστο και δυναμικό τρόπο, απαλλάσσοντας τους από ένα πλήθος τυποποιημένων και βαρετών διαδικασιών.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν γλώσσα σήμανσης υπερκειμένου HTML, γλώσσα φύλλου ύφους CSS, καθώς και γλώσσες προγραμματισμού PHP και Javascript. Τέλος χρησιμοποιήθηκε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων MySQL για την αποθήκευση και την ανάκτηση πληροφοριών για τους χρήστες και τα αρχεία τους από μια βάση δεδομένων.

Η γραπτή αναφορά της πτυχιακής εργασίας(το εγχειρίδιο που κρατάτε στα χέρια σας) αποτελείται από τέσσερα κεφάλαια εκτός της εισαγωγής και των συμπερασμάτων

Στο πρώτο κεφάλαιο παραθέτονται γενικές πληροφορίες σχετικά με την επιστήμη της πληροφορικής, το διαδίκτυο, τον παγκόσμιο ιστό και τους ιστότοπους.

Στο δεύτερο κεφάλαιό γίνεται αναφορά στις γλώσσες περιγραφής υλικού με εκτενή αναφορά στη γλώσσα VHDL παραθέτοντας στοιχεία για την ιστορία, τη σχεδίαση και τα πλεονεκτήματα της γλώσσας .Τέλος δίνονται και κάποια παραδείγματα κωδίκων σε VHDL.

Στο τρίτο κεφάλαιο γίνεται περιγραφή των εργαλείων ανάπτυξης λογισμικού. Ουσιαστικά μιλάμε για τα προγράμματα λογισμικού αλλά και τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη του κώδικα. Όλα τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της παρούσας εφαρμογής είναι ανοιχτού κώδικα (open source).

Στο τέταρτο και τελευταίο κεφάλαιο γίνεται αναφορά στον τρόπο σχεδίασης και τη λειτουργικότητα της εφαρμογής. Πιο συγκεκριμένα γίνεται παρουσίαση του τρόπου χειρισμού του προγράμματος από τους χρήστες με απλό και κατανοητό τρόπο. Κάθε κίνηση περιγράφεται γραπτώς, συνοδευόμενη από τα απαραίτητα screenshots της εφαρμογής αλλά και κομμάτια κώδικα της κάθε λειτουργίας.

1.1 Η ΕΠΙΣΤΗΜΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πληροφορική ονομάζεται η επιστήμη του επιτελεστικού λόγου, δηλαδή του λόγου (σε διάφορες μορφές) με τη βοήθεια του οποίου είναι εφικτή η επιτέλεση (εκτέλεση) ενεργειών που προκαλούν μεταβολές στον φυσικό κόσμο ή στον τρόπο λειτουργίας τεχνολογικών συσκευών διατάξεων και συστημάτων. Ειδικότερα, η Πληροφορική είναι θετική και εφαρμοσμένη επιστήμη και ερευνά τις

τεχνολογικές εφαρμογές σε αυτοματοποιημένα υπολογιστικά συστήματα, από τη σκοπιά της σχεδίασης, της ανάπτυξης, της υλοποίησης, της διερεύνησης, της ανάλυσης και της προδιαγραφής τους, την απόκτηση, την εκπροσώπηση, την επεξεργασία, την αποθήκευση, την επικοινωνία και την πρόσβαση στις πληροφορίες.

Τα εν λόγω υπολογιστικά συστήματα συνήθως είναι ηλεκτρονικές και ψηφιακές συσκευές, όμως τυπικά αυτό δεν είναι απαραίτητο αφού έχουν υπάρξει και μηχανικοί ή κβαντικοί υπολογιστές. Καθώς τα δεδομένα εισόδου, τα οποία ένας αλγόριθμος επεξεργάζεται, και τα δεδομένα εξόδου, τα οποία παράγει μετά την επεξεργασία και τη λήξη των υπολογισμών, αποτελούν κωδικοποιημένες πληροφορίες, η πληροφορική μπορεί επίσης να γίνει αντιληπτή ως η επιστήμη που ερευνά θεωρητικές μεθόδους και πρακτικούς μηχανισμούς διαχείρισης πληροφοριών. Η τεχνολογία πληροφοριών και επικοινωνίας άρχισε να λαμβάνει χώρα ευρέως μετά το 1970, με αποτέλεσμα σημαντικές κοινωνικές, οικονομικές και τεχνολογικές αλλαγές σε διεθνές επίπεδο.

Η αυτοματοποιημένη υλοποίηση των μεθόδων της πληροφορικής βασίστηκε από την πρώτη στιγμή στους ηλεκτρονικούς υπολογιστές (Η/Υ). Ωστόσο, αυτή έχει έναν ευρύτερο σκοπό που δεν περιορίζεται σε συγκεκριμένες τεχνολογικές επιλογές. Για παράδειγμα, ο αλγόριθμος της δυαδικής αναζήτησης μπορεί να εφαρμοστεί και σε τηλεφωνικό κατάλογο «χειρωνακτικά», από έναν άνθρωπο χωρίς τη βοήθεια υπολογιστή ο οποίος εκτελεί τους σχετικούς υπολογισμούς με τον νου του, ενώ ένα πρωτόκολλο επικοινωνίας μπορεί να εφαρμοστεί ακόμη και σε σήματα καπνού - όχι μόνο σε τηλεπικοινωνιακά δίκτυα. Αναλόγως με το επίπεδο αφαίρεσης, μπορεί να μελετηθεί είτε ανεξάρτητα από τις τεχνολογικές της συνιστώσες, είτε ως ένα ενιαία με αυτές επιστήμη. Επίσης, με την πληροφορική σχετίζεται και η διερεύνηση φυσικών διεργασιών επεξεργασίας πληροφοριών (βλ. γνωσιακή επιστήμη).

ΠΗΓΗ:https://el.wikipedia.org/wiki/Πληροφορική

1.2 ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ

Το **Διαδίκτυο** (Internet) είναι παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί εκατομμύρια χρήστες καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο.
Το Διαδίκτυο είναι επικοινωνιακό δίκτυο που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιουδήποτε διασυνδεδεμένου υπολογιστή. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και σε πολυάριθμα πρωτόκολλα επικοινωνίας. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τον όρο Διαδίκτυο περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί [μεταγωγή πακέτων] και τη [στοίβα πρωτοκόλλων].

ΠΗΓΗ:https://el.wikipedia.org/wiki/Διαδίκτυο

Η ιστορία του Διαδικτύου ξεκινά με την ανάπτυξη των ηλεκτρονικών υπολογιστών στη δεκαετία του 1950. Οι πρώτες έννοιες του δικτύου ευρείας περιοχής προήλθαν από διάφορα εργαστήρια πληροφορικής στις Ηνωμένες Πολιτείες, το Ηνωμένο Βασίλειο και τη Γαλλία. Το Υπουργείο Άμυνας των ΗΠΑ ανέθετε συμβάσεις έργων ήδη από τη δεκαετία του 1960, όπως για την ανάπτυξη του ARPANET από τους Ρομπέρ Τέιλορ και Λόρενς Ρόμπερτς. Το 1969 στάλθηκε

το πρώτο μήνυμα μέσω του ARPANET, από το εργαστήριο του Καθηγητή πληροφορικής Λέναρντ Κλέινροκ στο Πανεπιστήμιο της Καλιφόρνια, Λος Άντζελες (UCLA) προς το δεύτερο κόμβο του δικτύου στο Ερευνητικό Ίδρυμα του Στάνφορντ (SRI).

Τα δίκτυα μεταγωγής πακέτων όπως το δίκτυο NPL, το ARPANET, το Δίκτυο Μεριτ, το CYCLADES, και το Telenet, αναπτύχθηκαν περί τα τέλη της δεκαετίας του 1960 και τις αρχές της δεκαετίας του 1970 με χρήση μιας ποικιλίας από πρωτόκολλα επικοινωνίας. Ο Ντόναλντ Ντέιβις ήταν ο πρώτος που επέδειξε μεταγωγή πακέτων το 1967, στο Εθνικό Εργαστήριο Φυσικής (NPL) στο Ηνωμένο Βασίλειο. Το ARPANET οδήγησε στην ανάπτυξη των πρωτοκόλλων για διαδικτύωση, όπου πολλαπλά χωριστά δίκτυα θα μπορούσαν να ενταχθούν σε ένα ενιαίο δίκτυο δικτύων.

Η σουίτα πρωτοκόλλου διαδικτύου (TCP/IP) αναπτύχθηκε από τους Ρόμπερτ Καν και Βιντ Σερφ κατά τη δεκαετία του 1970 και αποτέλεσε το πρότυπο πρωτόκολλο διαδικτύου στο ARPANET, ενσωματώνοντας έννοιες από το γαλλικό πρόγραμμα ΚΥΚΛΑΔΕΣ του Λουίς Πουζίν. Στις αρχές της δεκαετίας του 1980 το NSF χρηματοδότησε την ίδρυση των εθνικών κέντρων υπερυπολογιστών σε διάφορα πανεπιστήμια, και το 1986 τους παρείχε διασυνδεσιμότητα με το πρόγραμμα NSFNET. Οι εμπορικοί πάροχοι υπηρεσιών Διαδικτύου άρχισαν να εμφανίζονται περί τα τέλη της δεκαετίας του 1980. Το 1990 το ARPANET αποσύρθηκε. Το 1989-90 σε αρκετές Αμερικανικές πόλεις εμφανίστηκαν οι πρώτες περιορισμένες ιδιωτικές συνδέσεις σε μέρη του Διαδικτύου από επίσημες εμπορικές οντότητες, και το 1995 αποσύρθηκε το NSFNET, μαζί με τους τελευταίους περιορισμούς σχετικά με τη χρήση του Διαδικτύου στην εμπορική κίνηση.

Τη δεκαετία του 1980, οι έρευνες στο CERN της Ελβετίας του Βρετανού επιστήμονα πληροφορικής Τιμ Μπέρνερς Λι, απέδωσαν τον Παγκόσμιο Ιστό, ένα σύστημα διασυνδεδεμένων εγγράφων υπερκειμένου προσπελάσιμο από

οποιοδήποτε κόμβο του διαδικτύου. Από τα μέσα της δεκαετίας του 1990, το Διαδίκτυο είχε έναν επαναστατικό αντίκτυπο στον πολιτισμό, το εμπόριο και την τεχνολογία, όπως την προαγωγή της γρήγορης επικοινωνίας με ηλεκτρονικό ταχυδρομείο, ανταλλαγή άμεσων μηνυμάτων, τηλεφωνικές κλήσεις με VoIP, βιντεοκλήσεις, και του Παγκόσμιου Ιστού με τα διαδικτυακά φόρουμ, τα ιστολόγια, τις εφαρμογές κοινωνικής δικτύωσης και τις αγορές από ηλεκτρονικά καταστήματα.

ΠΗΓΗ:https://el.wikipedia.org/wiki/Ιστορία_του_διαδικτύου

Ο παγκόσμιος ιστός (World Wide Web ή www) είναι ένα ανοιχτό σύστημα διασυνδεδεμένων πληροφοριών και πολυμεσικού περιεχομένου, που επιτρέπει στους χρήστες του Διαδικτύου να αναζητήσουν πληροφορίες μεταβαίνοντας από ένα έγγραφο στο άλλο.

Κάθε δικτυακή δομομονάδα του διαδικτύου αποτελείται από συνδεδεμένους υπολογιστές σε τοπικό επίπεδο, για παράδειγμα το δίκτυο υπολογιστών των κεντρικών γραφείων μιας εταιρείας. Αυτά τα δίκτυα με τη σειρά τους συνδέονται σε ευρύτερα δίκτυα, όπως εθνικά και υπερεθνικά. Το ευρύτερο δίκτυο στον κόσμο λέγεται παγκόσμιος ιστός το οποίο είναι μοναδικό (δηλαδή δεν υπάρχουν παραπάνω από ένα δίκτυα υπολογιστών παγκόσμιας κλίμακας), και συμπεριλαμβάνεται τόσο τα γήινα δίκτυα, όσο και τα δίκτυα των δορυφόρων της και άλλων διαστημικών συσκευών που είναι συνδεδεμένα σε αυτό.

Η τεχνολογία του ιστού καθιστά δυνατή την δημιουργία "υπερκειμένων", μία διασύνδεση δηλαδή πάρα πολλών μη ιεραρχημένων στοιχείων που παλαιότερα ήταν απομονωμένα. Τα στοιχεία αυτά μπορούν να πάρουν και άλλες μορφές πέραν της μορφής του γραπτού κειμένου, όπως εικόνας και ήχου.Ως πληροφοριακό σύστημα παρέχει συγκεκριμένο μοντέλο δεδομένων το οποίο βασίζεται σε κόμβους και υπερσυνδέσμους. Το μοντέλο δεδομένων του παγκόσμιου ιστού παρέχει κόμβους (αγγλ. nodes), άγκυρες (αγγλ. anchors) και συνδέσμους (ή υπερσυνδέσμους) (αγγλ. links ή hyperlinks). Οι κόμβοι είναι

φορείς του περιεχομένου ενώ οι άγκυρες είναι περιοχές του περιεχομένου των κόμβων από όπου ξεκινούν ή καταλήγουν σύνδεσμοι. Οι σύνδεσμοι συνδέουν δύο κόμβους.

ΠΗΓΗ:https://en.wikipedia.org/wiki/World_Wide_Web

1.3 ΙΣΤΟΤΟΠΟΣ

Ένας **ιστότοπος**, **ιστοχώρος** ή **διαδικτυακός τόπος** (*web site*) είναι μία συλλογή από ιστοσελίδες, εικόνες, βίντεο και άλλα ψηφιακά στοιχεία, τα οποία φιλοξενούνται στο ίδιο domain (περιοχή) του Παγκόσμιου Ιστού. Βασίζεται στην υπηρεσία www (*world wide web - παγκόσμιος ιστός*), μια από τις υπηρεσίες που παρέχονται στο Διαδίκτυο, με τη χρησιμοποίηση του πρωτοκόλλου http. Η υπηρεσία αυτή δίνει τη δυνατότητα στους χρήστες του ίντερνετ να δημιουργήσουν οποιουδήποτε είδους περιεχόμενο στις ιστοσελίδες τους. Το σύνολο των ιστοτόπων αποτελεί το world wide web (www).

Κάθε ιστοσελίδα είναι συνήθως της μορφής *www.τίτλος.TLD* που υποδηλώνει ότι βασίζεται στην υπηρεσία www. Το πρόθεμα www δεν είναι αναγκαίο, αλλά χρησιμοποιείται συνήθως σαν το όνομα του διακομιστή (web server) ο οποίος παρέχει την υπηρεσία www. Οι ιστότοποι μπορούν να αποκτήσουν πρόσβαση μέσω ενός δημόσιου δικτύου πρωτοκόλλου Internet (IP), όπως το Internet ή ένα ιδιωτικό τοπικό δίκτυο (LAN), από έναν ενιαίο εντοπιστή πόρων (URL) που προσδιορίζει τον ιστότοπο.Οι ιστότοποι μπορούν να έχουν πολλές λειτουργίες και μπορούν να χρησιμοποιηθούν σε διάφορες μορφές. ένας ιστότοπος μπορεί να είναι ένας προσωπικός ιστότοπος, ένας εταιρικός ιστότοπος για μια εταιρεία, ένας κυβερνητικός ιστότοπος, ένας ιστότοπος οργανισμού κ.λπ. Οι ιστότοποι είναι συνήθως αφιερωμένοι σε ένα συγκεκριμένο θέμα ή σκοπό, από την ψυχαγωγία και την κοινωνική δικτύωση μέχρι την παροχή ειδήσεων και εκπαίδευσης. Όλοι οι ιστοτόποι που είναι προσβάσιμοι από το κοινό συνιστούν το World Wide Web, ενώ ιδιωτικοί ιστότοποι, όπως ένας ιστότοπος εταιρείας για τους

υπαλλήλους του, είναι συνήθως μέρος ενός ενδοδικτύου.Οι ιστοσελίδες, οι οποίες είναι τα δομικά στοιχεία των ιστοτόπων, είναι έγγραφα που τυπικά αποτελούνται σε απλό κείμενο που διανέμονται με οδηγίες μορφοποίησης της Hypertext Markup Language (HTML, XHTML). Μπορούν να ενσωματώνουν στοιχεία από άλλους ιστότοπους με κατάλληλα άγκυρα σήμανσης. Οι ιστοσελίδες προσπελάζονται και μεταφέρονται με το πρωτόκολλο HTTP (Hypertext Transfer Protocol), το οποίο μπορεί προαιρετικά να χρησιμοποιεί κρυπτογράφηση (HTTP Secure, HTTPS) για την παροχή ασφάλειας και ιδιωτικότητας στον χρήστη. Η εφαρμογή του χρήστη, συχνά ένα πρόγραμμα περιήγησης ιστού, καθιστά το περιεχόμενο της σελίδας σύμφωνα με τις οδηγίες σήμανσης HTML σε ένα τερματικό οθόνης.Η υπερσύνδεση μεταξύ ιστοσελίδων μεταδίδει στον αναγνώστη τη δομή του ιστότοπου και καθοδηγεί την πλοήγηση του ιστότοπου, η οποία συχνά αρχίζει με μια αρχική σελίδα που περιέχει έναν κατάλογο του περιεχομένου του ιστότοπου. Ορισμένοι ιστότοποι απαιτούν εγγραφή χρήστη ή συνδρομή για πρόσβαση σε περιεχόμενο. Οι τελικοί χρήστες μπορούν να έχουν πρόσβαση σε ιστότοπους απο μια σειρά συσκευών, συμπεριλαμβανομένων επιτραπέζιων και φορητών υπολογιστών,tablet, smartphones και smart TV.Οι ιστότοποι μπορούν να χωριστούν σε δύο ευρείες κατηγορίες - στατικές και διαδραστικές. Οι διαδραστικοι ιστοτόποι αποτελούν μέρος της κοινότητας ιστότοπων Web 2.0 και επιτρέπουν τη διαδραστικότητα μεταξύ του ιδιοκτήτη του ιστότοπου και των επισκεπτών ή χρηστών.

Οι **διαδραστικοί ιστότοποι** μπορούν να αλληλεπιδράσουν χρησιμοποιώντας φόρμες HTML, αποθηκεύοντας και διαβάζοντας τα cookies του προγράμματος περιήγησης ή δημιουργώντας μια σειρά από σελίδες που αντικατοπτρίζουν το προηγούμενο ιστορικό των κλικ.Επιπλέων η JavaScript ενσωματώνεται στα σύγχρονα προγράμματα περιήγησης ιστού και επιτρέπει στους δημιουργούς ιστότοπων να αποστέλλουν κώδικα στο πρόγραμμα περιήγησης ιστού, το οποίο καθοδηγεί τον χρήστη να τροποποιήσει αλληλεπιδραστικά το περιεχόμενο της σελίδας και να επικοινωνήσει με τον

διακομιστή ιστού, αν χρειαστεί. Η εσωτερική αναπαράσταση του περιεχομένου του προγράμματος περιήγησης είναι γνωστή ως Μοντέλο αντικειμένου εγγράφου (DOM Object Model) και η τεχνική ως Dynamic HTML.

ΠΗΓΗ:https://en.wikipedia.org/wiki/Website

2.1 ΓΛΩΣΣΕΣ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ

Στη μηχανική υπολογιστών, μια **γλώσσα περιγραφής υλικού** hardware description language(HDL) είναι μια εξειδικευμένη γλώσσα υπολογιστών που χρησιμοποιείται για να περιγράψει τη δομή και τη συμπεριφορά των ηλεκτρονικών κυκλωμάτων και πιο συχνά τα κυκλώματα μιας ψηφιακής λογικής.

Μια γλώσσα περιγραφής υλικού επιτρέπει μια ακριβή, επίσημη περιγραφή ενός ηλεκτρονικού κυκλώματος για την αυτοματοποιημένη ανάλυση και προσομοίωση του. Επιτρέπει επίσης τη σύνθεση μιας περιγραφής HDL σε ένα netlist (μια προδιαγραφή των φυσικών ηλεκτρονικών εξαρτημάτων και του τρόπου με τον οποίο συνδέονται μεταξύ τους), τα οποία στη συνέχεια μπορούν να τοποθετηθούν και να δρομολογηθούν για να παράγουν το σύνολο των μασκων που χρησιμοποιούνται για τη δημιουργία ενός ολοκληρωμένου κυκλώματος.

Μια γλώσσα περιγραφής υλικού μοιάζει πολύ με μια γλώσσα προγραμματισμού όπως η C. Είναι μια περιγραφή κειμένου που αποτελείται από εκφράσεις, δηλώσεις και δομές ελέγχου. Μια σημαντική διαφορά μεταξύ των περισσότερων γλωσσών προγραμματισμού και των HDLs είναι ότι οι HDL περιλαμβάνουν ρητά την έννοια του χρόνου.

Τα συστήματα HDL αποτελούν αναπόσπαστο μέρος των συστημάτων αυτοματισμού ηλεκτρονικής σχεδίασης (EDA), ειδικά για σύνθετα κυκλώματα, όπως ολοκληρωμένα κυκλώματα ειδικά για εφαρμογές, μικροεπεξεργαστές και προγραμματιζόμενες λογικές συσκευές.

Λόγω της εκρηκτικής πολυπλοκότητας των ψηφιακών ηλεκτρονικών κυκλωμάτων από τη δεκαετία του 1970 (βλέπε νόμο του Moore), οι σχεδιαστές κύκλωματων χρειάζονται ψηφιακές λογικές περιγραφές για να εκτελούνται σε υψηλό επίπεδο χωρίς να συνδέονται με μια συγκεκριμένη ηλεκτρονική τεχνολογία.Υπάρχουν δύο κύριες γλώσσες περιγραφής υλικού: VHDL και Verilog.

Οι HDL είναι τυπικές εκφράσεις της δομής των ηλεκτρονικών συστημάτων και της συμπεριφοράς τους με την πάροδο του χρόνου. Όπως και οι ταυτόχρονες γλώσσες προγραμματισμού, η σύνταξη και η σημασιολογία της HDL περιλαμβάνει ρητές εντολές για την έκφραση του ταυτοχρονισμού . Ωστόσο, σε αντίθεση με τις περισσότερες γλώσσες προγραμματισμού λογισμικού, οι HDL περιλαμβάνουν επίσης μια ρητή έννοια του χρόνου, η οποία αποτελεί πρωταρχικό χαρακτηριστικό του υλικού. Γλώσσες των οποίων το μοναδικό χαρακτηριστικό είναι να εκφράζουν συνδεσιμότητα κυκλωμάτων μεταξύ μιας ιεραρχίας μπλοκ κατατάσσονται καταλλήλως στις γλώσσες netlist που χρησιμοποιούνται στον ηλεκτροτεχνικό σχεδιασμό (CAD).Το HDL μπορεί να χρησιμοποιηθεί για την έκφραση αρχιτεκτονικών structural, behavioral η register-transfer-level για την ίδια λειτουργικότητα του κυκλώματος. Στις τελευταίες δύο περιπτώσεις ο συνθέτης αποφασίζει τη διάταξη και αρχιτεκτονική των λογικών πυλών. Οι HDL χρησιμοποιούνται για την εγγραφή εκτελέσιμων προδιαγραφών για το υλικό. Προσομοιώνουν την πρόοδο του χρόνου παρέχοντας στον σχεδιαστή υλικού τη δυνατότητα να μοντελοποιεί ένα κομμάτι του υλικού προτού αυτό δημιουργηθεί φυσικά. Είναι αυτή η δυνατότητα εκτέλεσης που δίνει στις HDL την ψευδαίσθηση ότι είναι γλώσσες προγραμματισμού, όταν αυτές ταξινομούνται με μεγαλύτερη ακρίβεια ως γλώσσες προδιαγραφών ή γλώσσες μοντελοποίησης.Υπάρχουν προσομοιωτές που υποστηρίζουν ψηφιακή και αναλογική μοντελοποίηση και HDL που στοχεύουν στο καθένα ξεχωριστά.

ΠΗΓΗ:https://en.wikipedia.org/wiki/Hardware_description_language

2.2 VHDL

VHDL (VHSIC-HDL) (Very High Speed Integrated Circuit Hardware Description Language) είναι μια γλώσσα περιγραφής υλικού που χρησιμοποιείται στον αυτοματισμό ηλεκτρονικών σχεδιάσεων (electronic design automation) για την περιγραφή ψηφιακών και μεικτών (mixed-signal) συστημάτων, όπως οι συστοιχίες επιτόπια προγραμματιζόμενων πυλών (FPGA) και τα ολοκληρωμένα κυκλώματα.

2.2.1 ΙΣΤΟΡΙΑ

Η VHDL αναπτύχθηκε αρχικά από το Υπουργείο Άμυνας των ΗΠΑ για την τεκμηρίωση των κυκλωμάτων ASIC που χρησιμοποιούσαν οι εταιρείες-προμηθευτές στον εξοπλισμό τους. Η VHDL δημιουργήθηκε δηλαδή σαν εναλλακτική λύση αντί των μεγάλων και πολύπλοκων εγχειριδίων που βασίζονταν στις λεπτομέρειες της εκάστοτε υλοποίησης.

Η ιδέα της προσομοίωσης της τεκμηρίωσης οδήγησε στην ανάπτυξη λογικών προσομοιωτών που μπορούσαν να διαβάσουν τα αρχεία VHDL. Το επόμενο βήμα ήταν η ανάπτυξη εργαλείων λογικής σύνθεσης που διάβαζαν την VHDL και παρήγαγαν έναν ορισμό της φυσικής υλοποίησης του κυκλώματος.

Η αρχική έκδοση της VHDL, που σχεδιάστηκε για να αποτελέσει το πρότυπο 1076-1987 του IEEE, περιλάμβανε αρκετούς τύπους δεδομένων, όπως τους αριθμητικούς ακέραιους (integer) και πραγματικούς (real), τους λογικούς (bit και boolean), χαρακτήρες (character) και χρόνο (time), καθώς και πίνακες από bit (bit_vector) και από character (string).

Ένα πρόβλημα που έμεινε άλυτο σε αυτήν την έκδοση ήταν η χρήση "λογικής πολλαπλών τιμών" ("multi-valued logic"), όπου λαμβάνονται υπόψη το

σθένος οδήγησης ενός σήματος (μηδενικό, ασθενές ή ισχυρό) και οι άγνωστες τιμές. Αυτό απαιτούσε το πρότυπο IEEE 1164, που όριζε τύπους λογικής 9 τιμών: τον βαθμωτό std_ulogic και την ανυσματική έκδοσή του std_ulogic_vector.

Το ανανεωμένο IEEE 1076 του 1993, ανέμεσα σε άλλες αλλαγές, έκανε την σύνταξη πιο συνεπή, επέτρεψε μεγαλύτερη ευελιξία στα ονόματα που χρησιμοποιούνταν, επέκτεινε τον τύπο character για να επιτρέπει τους εκτυπώσιμους χαρακτήρες του ISO-8859-1, πρόσθεσε τον τελεστή xnor.

Μικρές αλλαγές του προτύπου (το 2000 και το 2002) πρόσθεσαν την ιδέα των προστατευμένων τύπων (που είναι παρόμοιοι με την έννοια των κλάσεων της C++) και αφαίρεσαν κάποιους περιορισμούς από τους κανόνες αντιστοίχισης θυρών (port mapping rules).

Εκτός από το πρότυπο IEEE 1164, εμφανίστηκαν πολλά θυγατρικά πρότυπα που επέκτειναν την λειτουργικότητα της γλώσσας. Το πρότυπο IEEE 1076.2 πρόσθεσε καλύτερο χειρισμό τύπων δεδομένων πραγματικών και μιγαδικών αριθμών. Το πρότυπο IEEE 1076.3 εισήγαγε τους τύπους με πρόσημο και χωρίς πρόσημο (signed/unsigned types) για τη διευκόλυνση των αριθμητικών λειτουργιών σε διανύσματα. Το πρότυπο IEEE 1076.1 (γνωστό και ως VHDL-AMS) περιλαμβάνει επίσης επεκτάσεις για τη σχεδίαση αναλογικών και μεικτών κυκλωμάτων.

Κάποια άλλα πρότυπα υποστηρίζουν ευρύτερη χρήση της VHDL, όπως το VITAL (VHDL Initiative Towards ASIC Libraries) και οι επεκτάσεις για σχεδίαση κυκλωμάτων μικροκυμάτων.

Τον Ιούνιο του 2006, η Τεχνική Επιτροπή της VHDL (VHDL Technical Committee) της Accellera (στην οποία είχε ανατεθεί από τον IEEE η εργασία για την επόμενη αναθεώρηση του προτύπου) ενέκρινε το αποκαλούμενο Draft 3.0 της VHDL-2006. Διατηρώντας πλήρη συμβατότητα με τις παλιότερες εκδόσεις, το προτεινόμενο αυτό πρότυπο παρέχει διάφορες επεκτάσεις που διευκολύνουν τη συγγραφή και τη διαχείριση κώδικα σε VHDL. Βασικές αλλαγές είναι η

ενσωμάτωση θυγατρικών προτύπων (1164, 1076.2, 1076.3) στο βασικό πρότυπο 1076, ένα μεγαλύτερο σύνολο τελεστών, πιο ευέλικτη χρήση των εντολών case και generate, ενσωμάτωση του VHPI (διεπαφής προς τις γλώσσες C και C++) και ένα υποσύνολο της PSL (Property Specification Language). Οι αλλαγές αυτές βελτιώνουν την ποιότητα του παραγόμενου κώδικα VHDL, κάνουν τα testbenches πιο ευέλικτα και επιτρέπουν τη χρήση της VHDL σε ένα ευρύτερο πεδίο περιγραφών επιπέδου συστήματος.

Τον Φεβρουάριο του 2008 η Accellera ενέκρινε την VHDL 4.0 (ανεπίσημα γνωστή και ως VHDL 2008), η οποία αντιμετώπιζε πάνω από 90 προβλήματα που είχαν ανακαλυφθεί κατά τη δοκιμαστική περίοδο της έκδοσης 3.0 και περιλαμβάνει εμπλουτισμένους γενικούς τύπους (enhanced generic types). Το 2008 η Accellera διένειμε την VHDL 4.0 στον IEEE προσπαθώντας να την περιλάβει στο πρότυπο IEEE 1076-2008. Το πρότυπο IEEE 1076-2008 της VHDL δημοσιεύτηκε τον Ιανουάριο του 2009.

2.2.2 ΣΧΕΔΙΑΣΗ

Η VHDL συνήθως χρησιμοποιείται για τη συγγραφή μοντέλων σε κείμενο που περιγράφουν ένα λογικό κύκλωμα. Ένα πρόγραμμα σύνθεσης μπορεί να επεξεργαστεί ένα τέτοιο μοντέλο μόνο αν είναι μέρος της λογικής σχεδίασης, επομένως χρησιμοποιείται ένα πρόγραμμα προσομοίωσης για να δοκιμαστεί η λογική σχεδίαση χρησιμοποιώντας μοντέλα προσομοίωσης που αναπαριστούν τα λογικά κυκλώματα που αντιστοιχούν στη σχεδίαση. Η συλλογή αυτή από μοντέλα προσομοίωσης συνήθως αποκαλείται testbench.

Η VHDL έχει δομές (διεργασίες ή processes) που χειρίζονται τον παραλληλισμό που υπάρχει στις σχεδιάσεις υλικού.Η VHDL έχει ισχυρούς τύπους (strongly typed) και δεν κάνει διάκριση μεταξύ κεφαλαίων και μικρών γραμμάτων. Η VHDL επιτρέπει επίσης τη δεικτοδότηση πινάκων σε σειρά από μικρότερο προς μεγαλύτερο δείκτη ή αντίστροφα - και οι δύο συμβάσεις χρησιμοποιούνται στο

υλικό, ενώ στις περισσότερες άλλες γλώσσες προγραμματισμού χρησιμοποιείται μόνο ο πρώτος τρόπος δεικτοδότησης.

Η VHDL έχει δυνατότητες εισόδου και εξόδου σε αρχεία και μπορεί να χρησιμοποιηθεί σαν γλώσσα γενικών καθηκόντων για επεξεργασία κειμένου, αλλά τα αρχεία χρησιμοποιούνται συνήθως από ένα testbench προσομοίωσης για τον ορισμό διεγέρσεων (stimuli), αλληλεπίδραση με τον χρήστη και για τη σύγκριση των ληφθέντων δεδομένων με τα επιθυμητά δεδομένα. Παρόλα αυτά, οι περισσότεροι σχεδιαστές αφήνουν αυτήν την εργασία στον προσομοιωτή.
Για έναν προγραμματιστή χωρίς εμπειρία είναι σχετικά εύκολο να παράγει κώδικα που προσομοιώνεται με επιτυχία αλλά δε μπορεί να παραχθεί σαν πραγματική υλοποίηση, ή είναι πολύ μεγάλος για να χρησιμοποιηθεί στην πράξη.

Η σχεδίαση του υλικού μπορεί να γίνει σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης για VHDL (για υλοποίηση FPGA τέτοια είναι το Xilinx ISE, το Altera Quartus, το Synopsys Synplify και το Mentor Graphics HDL Designer), ώστε να παραχθεί το σχηματικό διάγραμμα RTL του επιθυμητού κυκλώματος. Μετά από αυτό, το παραγόμενο σχηματικό διάγραμμα μπορεί να επαληθευτεί με χρήση λογισμικού προσομοίωσης που δείχνει τις κυματομορφές των εισόδων και των εξόδων του κυκλώματος μετά την δημιουργία του κατάλληλου testbench. Η δημιουργία του σωστού testbench για ένα κύκλωμα ή έναν κώδικα σε VHDL απαιτεί τον σωστό ορισμό των εισόδων.

Όταν ένα μοντέλο σε VHDL μεταφράζεται σε "πύλες και γραμμές" που αντιστοιχίζονται σε μια προγραμματιζόμενη λογική συσκευή όπως ένα CPLD ή ένα FPGA, τότε το πραγματικό υλικό είναι αυτό που ρυθμίζεται και δεν "εκτελείται" ο κώδικας VHDL σε κάποιου τύπου επεξεργαστή.

2.2.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ

Το βασικό πλεονέκτημα της VHDL, όταν αυτή χρησιμοποιείται για σχεδίαση συστημάτων, είναι ότι επιτρέπει την περιγραφή (μοντελοποίηση) και την επαλήθευση (προσομοίωση) του επιθυμητού συστήματος, πριν τα εργαλεία σύνθεσης μεταφράσουν τη σχεδίαση σε πραγματικό υλικό (πύλες και γραμμές).

Ένα άλλο όφελος της VHDL είναι ότι επιτρέπει τον ορισμό ταυτόχρονων συστημάτων (concurrent systems). Η VHDL είναι γλώσσα ροής δεδομένων, σε αντίθεση με τις διαδικαστικές γλώσσες προγραμματισμού όπως η BASIC, η C και η συμβολική γλώσσα, οι οποίες εκτελούνται ακολουθιακά, με κάθε εντολή να ακολουθεί την προηγούμενη.

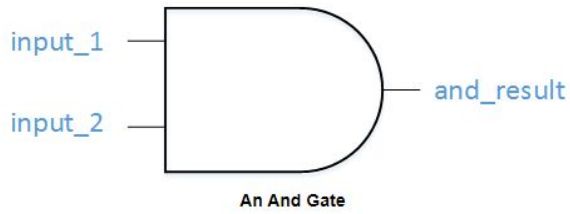
Ένα έργο σε VHDL έχει πολλές εφαρμογές. Ένα μπλοκ υπολογισμού (calculation block) δημιουργείται μια φορά αλλά μπορεί να χρησιμοποιηθεί σε άλλα έργα. Μπορούν επίσης να ρυθμιστούν διάφορες παράμετροι διαμόρφωσης και λειτουργίας του μπλοκ (παράμετροι χωρητικότητας, το μέγεθος της μνήμης, η βάση των στοιχείων (element base), η σύνθεση μπλοκ και η δομή διασύνδεσης).

Ένα έργο σε VHDL είναι επίσης μεταφέρσιμο. Αν έχει δημιουργηθεί για μια βάση στοιχείων, μπορεί να μεταφερθεί σε μια άλλη βάση, για παράδειγμα σε ένα VLSI με διάφορες τεχνολογίες.

ΠΗΓΗ:https://en.wikipedia.org/wiki/VHDL

2.2.4 ΠΑΡΑΔΕΙΓΜΑΤΑ

Μια μικρή εισαγωγή στη VHDL



Η θεμελιώδης μονάδα της VHDL ονομάζεται σήμα. Προς το παρόν, ας υποθέσουμε ότι ένα σήμα μπορεί να είναι είτε 0 είτε 1 (υπάρχουν και άλλες δυνατότητες).Το σήμα ακολουθεί κάποια βασική λογική VHDL:

```
1  signal and_gate : std_logic;
2  and_gate <= input_1 and input_2;
```

Η πρώτη γραμμή κώδικα ορίζει ένα σήμα τύπου std_logic και καλείται and_gate. Το Std_logic είναι ο τύπος που χρησιμοποιείται συχνότερα για τον ορισμό σημάτων, αλλά υπάρχουν και άλλα. Αυτός ο κώδικας θα δημιουργήσει μια πύλη AND με μία έξοδο (and_gate) και 2 εισόδους (input_1 και input_2). Η λέξη-κλειδί "and" είναι δεσμευμένη στη VHDL. Ο χειριστής <= είναι γνωστός ως χειριστής εκχώρησης

.

Τώρα θα αναρωτιέστε από πού προέρχονται οι input_1 και input_2. Οπως το όνομά τους υπονοεί, είναι εισροές στο αρχείο, έτσι πρέπει να δηλωθούν. Οι εισόδοι και οι εξόδοι σε ένα αρχείο ορίζονται σε μια οντότητα. Μια οντότητα περιέχει μια θύρα που ορίζει όλες τις εισόδους και εξόδους σε ένα αρχείο. Ας δημιουργήσουμε μια απλή οντότητα:

```
1  entity example_and is
2    port (
3      input_1    : in  std_logic;
4      input_2    : in  std_logic;
5      and_result : out std_logic
6    );
7  end example_and;
```

Ο παραπάνω κώδικας ορίζει μια οντότητα που ονομάζεται example_and και 3 σήματα, 2 είσοδοι και 1 έξοδος, τα οποία είναι τύπου std_logic. Μια άλλη λέξη-κλειδί της VHDL που είναι απαραίτητη για να τελειώσουμε με τη σχεδίαση είναι η λέξη αρχιτεκτονική. Μια αρχιτεκτονική χρησιμοποιείται για να περιγράψει τη λειτουργικότητα μιας συγκεκριμένης οντότητας. Σκεφτείτε ένα έγγραφο εργασίας: η οντότητα είναι ο πίνακας περιεχομένων και η αρχιτεκτονική είναι το περιεχόμενο. Ας δημιουργήσουμε μια αρχιτεκτονική για αυτήν την οντότητα:

```
architecture rtl of example_and is
  signal and_gate : std_logic;
begin
  and_gate <= input_1 and input_2;
  and_result <= and_gate;
end rtl;
```

Ο παραπάνω κώδικας ορίζει μια αρχιτεκτονική που ονομάζεται rtl της οντότητας example_and. Όλα τα σήματα που χρησιμοποιούνται από την αρχιτεκτονική πρέπει να ορίζονται μεταξύ των λέξεων κλειδιών "is" και "begin". Η πραγματική λογική της αρχιτεκτονικής έρχεται ανάμεσα στις λέξεις-κλειδιά "begin" και "end". Ένα τελευταίο πράγμα που πρέπει ορίσετε είναι ποια βιβλιοθήκη να χρησιμοποιήσετε. Μια βιβλιοθήκη καθορίζει τον τρόπο συμπεριφοράς ορισμένων λέξεων-κλειδιών στο αρχείο σας. Προς το παρόν, θεωρήστε ότι είναι απαραίτητο να έχετε αυτές τις 2 γραμμές στην κορυφή του αρχείου σας:

```
library ieee;
use ieee.std_logic_1164.all;
```

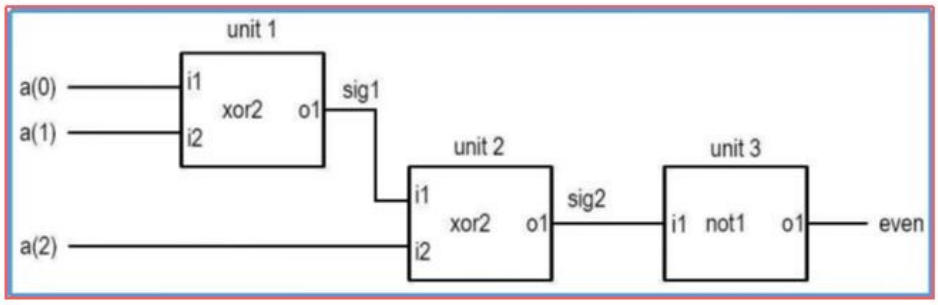
Δημιουργήσατε το πρώτο σας αρχείο VHDL. Μπορείτε να δείτε το ολοκληρωμένο αρχείο εδώ:

```
library ieee;
use ieee.std_logic_1164.all;

entity example_and is
  port (
    input_1    : in  std_logic;
    input_2    : in  std_logic;
    and_result : out std_logic
    );
end example_and;

architecture rtl of example_and is
  signal and_gate : std_logic;
begin
  and_gate   <= input_1 and input_2;
  and_result <= and_gate;
end rtl;
```

ΠΗΓΗ:https://www.nandland.com/vhdl/tutorials/tutorial-introduction-to-vhdl-for-beginners.html

<u>Παραδείγματα διαφορετικών περιγραφών κωδίκων VHDL</u>

Structural Description - Δομική περιγραφή



test.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY xor2 IS
PORT(
   i1, i2: IN STD_LOGIC;
   o1: OUT STD_LOGIC
);
END xor2;

ARCHITECTURE arch OF xor2 IS
BEGIN
   o1 <= i1 xor i2;
END arch;
```

test2.vhd

```
LIBRARY ieee;
USE std_logic_1164.ALL;

ENTITY not1 IS
PORT(
   i1: IN STD_LOGIC;
   o1: OUT STD_LOGIC
);
END not1;

ARCHITECTURE arch OF not1 IS
BEGIN
   o1 <= NOT i1;
END arch;
```

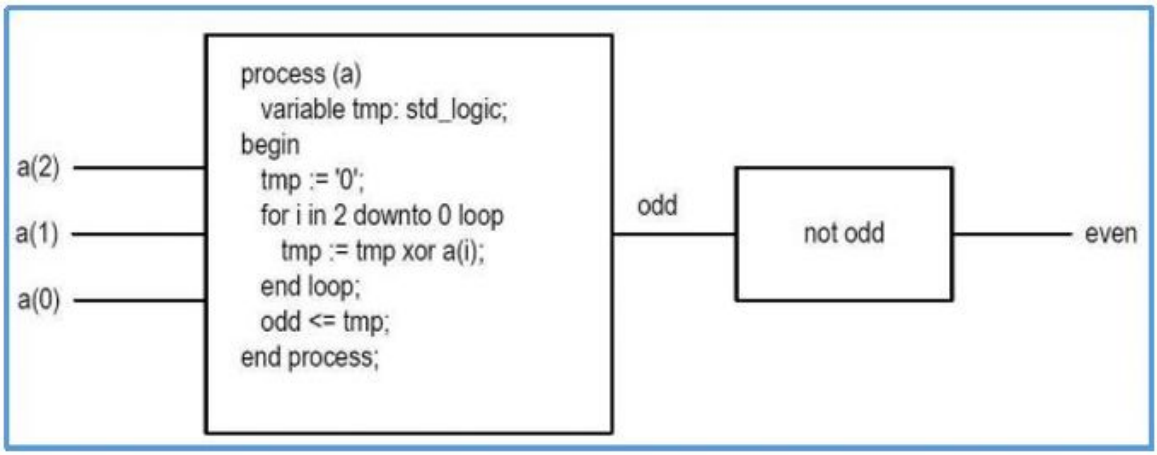
test.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY circ IS
PORT(
   a0, a1, a2: IN STD_LOGIC;
   o1: OUT STD_LOGIC
);
END circ;

ARCHITECTURE str_arch OF circ IS
COMPONENT xor2
PORT(
   i1, i2: IN STD_LOGIC;
   o1: OUT STD_LOGIC
);
```

test.vhd

```
COMPONENT not1
PORT(
   i1: IN STD_LOGIC;
   o1: OUT STD_LOGIC
);
END COMPONENT;

SIGNAL sig1, sig2: STD_LOGIC;

BEGIN
unit1: xor2 PORT MAP(a0, a1, sig1);
unit2: xor2 PORT MAP(a2, sig1, sig2);
unit3: not1 PORT MAP(sig2, o1);
END str_arch;
```

Behavioral Description - Περιγραφή συμπεριφοράς



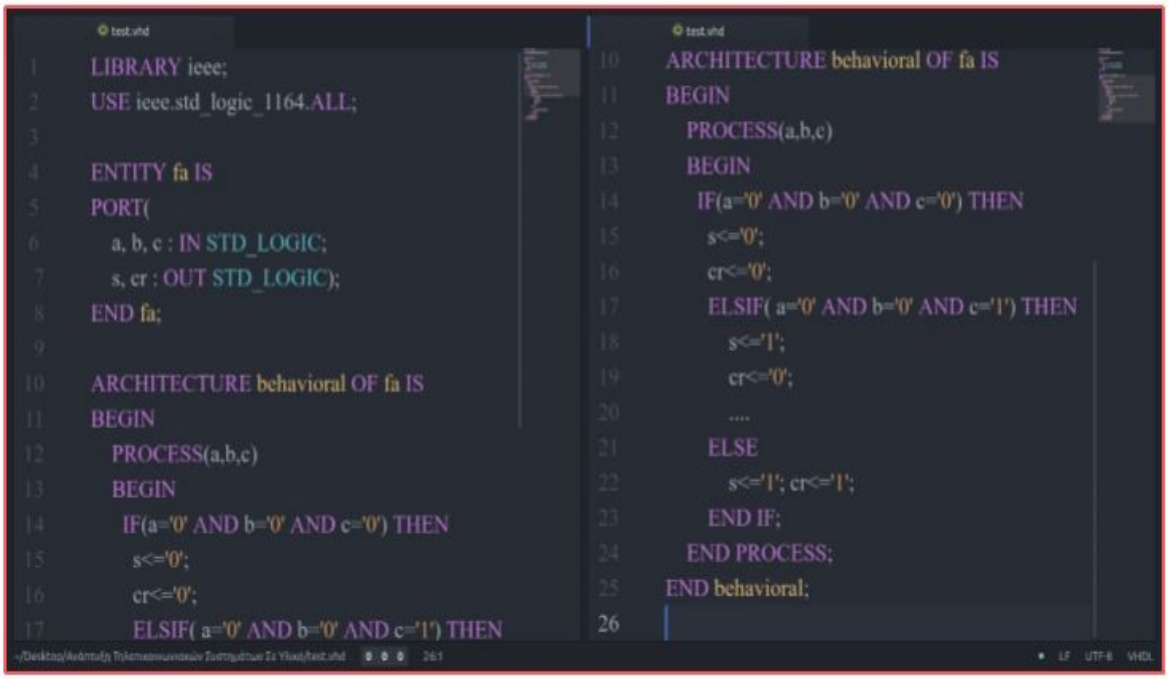

```
test.vhd
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.ALL;
3
4   ENTITY circ IS
5   PORT(
6     a: IN STD_LOGIC_VECTOR(2 DOWNTO 0);
7     o1: OUT STD_LOGIC
8   );
9   END circ;
10
11  ARCHITECTURE beh_arch OF circ IS
12  SIGNAL odd: STD_LOGIC;
13  BEGIN
14    o1 <= NOT odd;
15    PROCESS(a)
16      VARIABLE temp:= '0';
```

```
test.vhd
11  ARCHITECTURE beh_arch OF circ IS
12  SIGNAL odd: STD_LOGIC;
13  BEGIN
14    o1 <= NOT odd;
15    PROCESS(a)
16      VARIABLE temp:= '0';
17      BEGIN
18      tmp:= '0';
19      FOR i IN 2 DOWNTO 0 LOOP
20        tmp:= tmp XOR a(i);
21      END LOOP;
22
23      odd <= tmp;
24    END PROCESS;
25  END beh_arch;
26
```

Dataflow description - Περιγραφή ροής δεδομένων
(Full Adder - Πλήρης Αθροιστής)

```
test.vhd
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.ALL;
3
4   ENTITY fa IS
5   PORT(
6      a, b, c : IN STD_LOGIC;
7      s, cr : OUT STD_LOGIC);
8   END fa;
9
10  ARCHITECTURE dataflow OF fa IS
11  BEGIN
12     s<= a XOR b XOR c;
13     cr<= (a AND b) OR (b AND cin) OR (c AND a);
14  END dataflow;
```

Behavioral Description - Περιγραφή συμπεριφοράς
(Full Adder - Πλήρης Αθροιστής)



```
test.vhd
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.ALL;
3
4   ENTITY fa IS
5   PORT(
6      a, b, c : IN STD_LOGIC;
7      s, cr : OUT STD_LOGIC);
8   END fa;
9
10  ARCHITECTURE behavioral OF fa IS
11  BEGIN
12    PROCESS(a,b,c)
13    BEGIN
14     IF(a='0' AND b='0' AND c='0') THEN
15       s<='0';
16       cr<='0';
17       ELSIF( a='0' AND b='0' AND c='1') THEN
```

```
test.vhd
10  ARCHITECTURE behavioral OF fa IS
11  BEGIN
12    PROCESS(a,b,c)
13    BEGIN
14     IF(a='0' AND b='0' AND c='0') THEN
15       s<='0';
16       cr<='0';
17       ELSIF( a='0' AND b='0' AND c='1') THEN
18          s<='1';
19          cr<='0';
20          ....
21       ELSE
22          s<='1'; cr<='1';
23       END IF;
24    END PROCESS;
25  END behavioral;
26
```

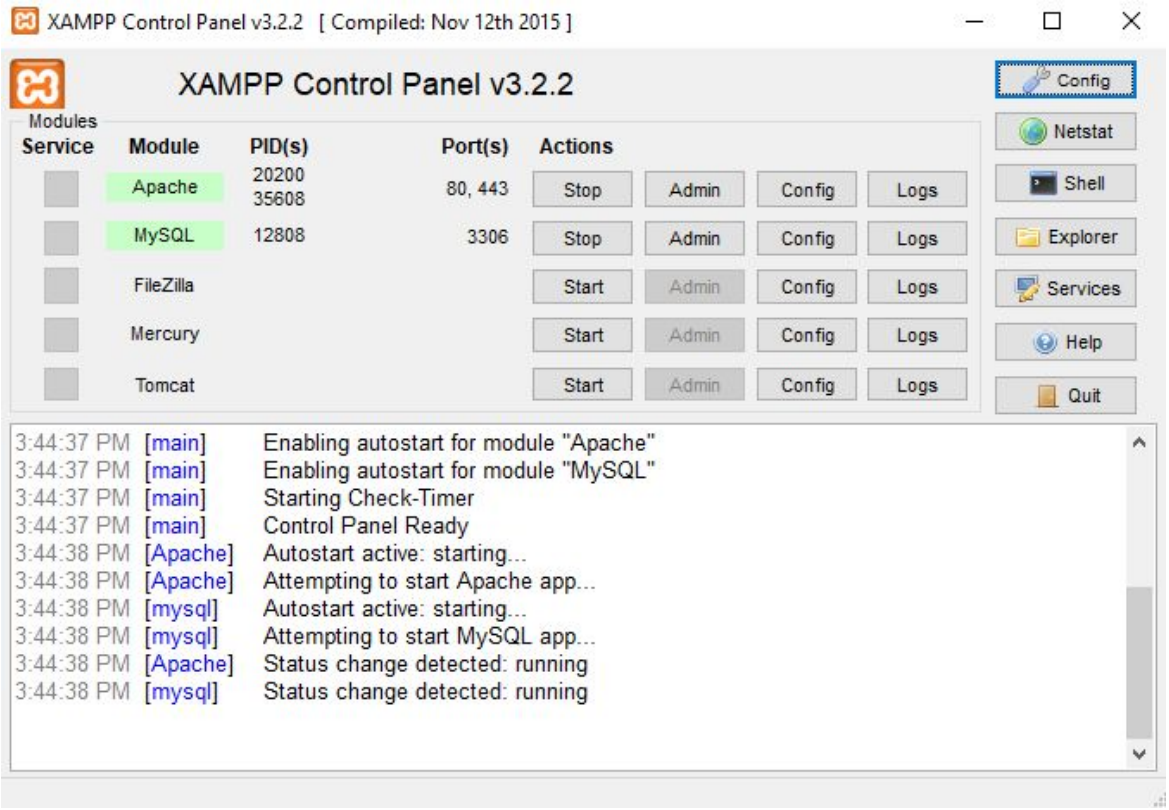
3 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

3.1.1 XAMPP Control Panel

Το **XAMPP Control Panel** είναι μια πλατφόρμα ελεύθερου λογισμικού που αναπτύχθηκε από τους Apache Friends και αποτελείται κυρίως από το Apache HTTP Server, τη βάση δεδομένων MariaDB και διερμηνείς για scripts γραμμένα στις γλώσσες προγραμματισμού PHP και Perl. Το XAMPP ενημερώνεται τακτικά με τις πιο πρόσφατες κυκλοφορίες των Apache, MariaDB, PHP και Perl. Έρχεται επίσης με μια σειρά άλλων modules όπως το OpenSSL, το phpMyAdmin, το MediaWiki, το Joomla, το WordPress και πολλά άλλα. Αυτοτελείς, πολλαπλές εμφανίσεις του XAMPP μπορούν να υπάρχουν σε έναν μόνο υπολογιστή και κάθε δεδομένη στιγμή μπορεί να αντιγραφεί από έναν υπολογιστή στον άλλο. Είναι πολύ εύκολο στη χρήση και στην εγκατάσταση καθώς η όλη διαδικασία γίνεται αυτόματα μέσα από ένα γραφικό περιβάλλον. Για την παρούσα εργασία χρησιμοποιήθηκε η έκδοση XAMPP Version: 7.2.5 για τα Windows home 64-bit. Η έκδοση αυτή περιλαμβάνει :

- Apache 2.4.33
- MariaDB 10.1.32
- PHP 7.2.5 (VC15 X86 32bit thread safe)
- phpMyAdmin 4.8.0.1
- OpenSSL 1.1.0g
- ADOdb 518a
- Mercury Mail Transport System v4.63
- FileZilla FTP Server 0.9.41
- Webalizer 2.23-04
- Strawberry Perl 5.16.1.1 Portable

- Tomcat 7.0.56
- XAMPP Control Panel Version 3.2.2



ΠΗΓΕΣ:https://www.apachefriends.org/index.html
https://en.wikipedia.org/wiki/XAMPP

3.1.2 Apache HTTP Server

Ο **Apache HTTP** γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα

Windows, το Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους. Τον Μάρτιο του 2017, το 49,48% του συνόλου των καταχωρισμένων Ελληνικών τομέων χρησιμοποιούσε το Apache.

ΠΗΓΕΣ:https://en.wikipedia.org/wiki/Apache_HTTP_Server

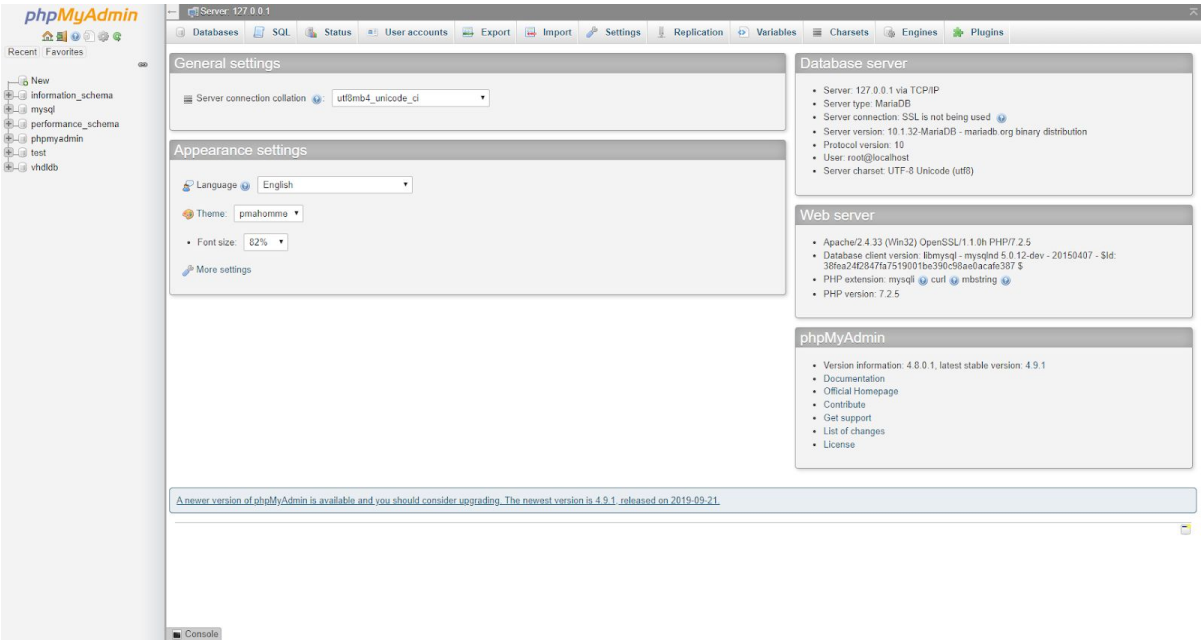
https://httpd.apache.org/

3.1.3 PhpMyAdmin

Το **PhpMyAdmin** είναι ένα εργαλείο ελεύθερου λογισμικού γραμμένο σε PHP, το οποίο προορίζεται για τη διαχείριση της MySQL μέσω του Web. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα λειτουργιών στις υπηρεσίες MySQL και MariaDB. Συχνά χρησιμοποιούμενες λειτουργίες (διαχείριση βάσεων δεδομένων, πίνακες, στήλες, σχέσεις, ευρετήρια, χρήστες, δικαιώματα κ.λπ.)

μπορούν να εκτελεστούν μέσω του περιβάλλοντος χρήστη, ενώ υπάρχει και η δυνατότητα εκτέλεσης απευθείας οποιαδήποτε δήλωσης SQL.Χαρακτηριστικά που παρέχει το πρόγραμμα:

- Διεπαφή ιστού
- Διαχείριση βάσεων δεδομένων MySQL και MariaDB
- Υποστήριξη για τις περισσότερες λειτουργίες MySQL
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV, SQL, XML, PDF (μέσω της βιβλιοθήκης TCPDF), ISO / IEC 26300 - Κείμενο και υπολογιστικό φύλλο OpenDocument, Word, Excel, LaTeX και άλλα
- Διαχείριση πολλών διακομιστών
- Δημιουργία γραφικών PDF της διάταξης βάσης δεδομένων
- Δημιουργία πολύπλοκων ερωτημάτων χρησιμοποιώντας ερωτήσεις ανά παράδειγμα (QBE)
- Αναζήτηση σε παγκόσμιο επίπεδο σε μια βάση δεδομένων ή σε ένα υποσύνολο της
- Μετασχηματισμός αποθηκευμένων δεδομένων σε οποιαδήποτε μορφή χρησιμοποιώντας ένα σύνολο προκαθορισμένων λειτουργιών, όπως την εμφάνιση δεδομένων BLOB ως εικόνα ή link
- Ζωντανά διαγράμματα για την παρακολούθηση της δραστηριότητας διακομιστή MySQL όπως συνδέσεις, διαδικασίες, χρήση CPU / μνήμης κλπ.
- Εργασία με διαφορετικά λειτουργικά συστήματα.



ΠΗΓΕΣ:https://en.wikipedia.org/wiki/PhpMyAdmin

https://www.phpmyadmin.net/

3.2 HTML (Γλώσσα Σήμανσης Υπερκειμένου)

Η **HTML** (αρχικοποίηση του αγγλικού HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990.

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), τους βασισμένους σε χαρακτήρες τύπους δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στο πρόγραμμα περιήγησης (browser) τον τρόπο εμφάνισης της σελίδας.

Στην HTML, το πρόγραμμα Hello world, ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών σεναρίων και γλωσσών σήμανσης, φτιάχνεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στην HTML:

```
<!DOCTYPE html>
<html>
  <head>
```

```
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Τα έγγραφα HTML αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή < και >.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: <ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">περιεχόμενο</ετικέτα>. Μερικά στοιχεία HTML περιγράφονται ως άδεια ή κενά στοιχεία, έχουν τη μορφή <ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια πλάγια «/», η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

ΠΗΓΗ:https://en.wikipedia.org/wiki/HTML

3.3 CSS (Διαδοχικά Φύλλα Ύφους)

Η **CSS** (Cascading Style Sheets – διαδοχικά φύλλα ύφους ή επάλληλα φύλλα ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των

γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης.

Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.
Η CSS έχει μια απλή σύνταξη και χρησιμοποιεί μια σειρά αγγλικών λέξεων-κλειδιών για να καθορίσει τα ονόματα των διαφόρων ιδιοτήτων στυλ. Ένα φύλλο στυλ αποτελείται από μια λίστα κανόνων. Κάθε κανόνας ή σύνολο κανόνων αποτελείται από έναν ή περισσότερους επιλογείς και ένα μπλοκ δήλωσης.

Στη CSS, οι επιλογείς δηλώνουν ποιο μέρος της σήμανσης ισχύει για ένα στυλ, συνδυάζοντας ετικέτες και χαρακτηριστικά στο ίδιο το markup.
Οι επιλογείς μπορεί να ισχύουν για τα ακόλουθα:

- όλα τα στοιχεία συγκεκριμένου τύπου, π.χ. οι κεφαλίδες δευτέρου επιπέδου h2
- στοιχεία που καθορίζονται από το χαρακτηριστικό, και ιδίως:
- id: ένα αναγνωριστικό μοναδικό μέσα στο έγγραφο
- class: ένα αναγνωριστικό που μπορεί να σχολιάσει πολλά στοιχεία σε ένα έγγραφο
- ανάλογα με τον τρόπο με τον οποίο τοποθετούνται σε σχέση με άλλους στο δέντρο εγγράφων.

ΠΗΓΗ:https://en.wikipedia.org/wiki/Cascading_Style_Sheets

3.4 PHP

Η **PHP** (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργαστεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal).

Ο κώδικας PHP μπορεί να εκτελεστεί με μια διασύνδεση γραμμής εντολών (CLI), ενσωματωμένη σε κώδικα HTML ή σε συνδυασμό με διάφορα συστήματα προτύπων ιστού, συστήματα διαχείρισης περιεχομένου ιστού και πλαίσια ιστού. Ο κώδικας PHP συνήθως επεξεργάζεται από έναν διερμηνέα PHP που υλοποιείται ως ενότητα σε ένα web server ή ως εκτελέσιμο CGI (Common Gateway Interface). Ο διακομιστής ιστού εξάγει τα αποτελέσματα του ερμηνευμένου και εκτελεσμένου κώδικα PHP, ο οποίος μπορεί να είναι οποιοσδήποτε τύπος δεδομένων, όπως ο παραγόμενος κώδικας HTML ή δυαδικά δεδομένα εικόνας. Η PHP μπορεί να χρησιμοποιηθεί για πολλές εργασίες προγραμματισμού εκτός του πλαισίου του διαδικτύου, όπως οι αυτόνομες γραφικές εφαρμογές και εφαρμογές ρομποτικού ελέγχου.

Ο πρότυπος διερμηνέας της PHP, που τροφοδοτείται από το Zend Engine, είναι ελεύθερο λογισμικό που εκδίδεται υπό την PHP. Η PHP έχει μεταφερθεί

ευρέως και μπορεί να αναπτυχθεί στους περισσότερους διακομιστές ιστού σε σχεδόν κάθε λειτουργικό σύστημα και πλατφόρμα, δωρεάν.

Η ανάπτυξη της PHP άρχισε το 1994, όταν ο Rasmus Lerdorf έγραψε διάφορα προγράμματα CGI (Common Gateway Interface) στη C τα οποία συνήθιζε να διατηρεί την προσωπική του σελίδα. Τα επέκτεινε να δουλεύουν με φόρμες ιστού και να επικοινωνούν με βάσεις δεδομένων και κάλεσε αυτή την εφαρμογή "Personal Home Page / Forms Interpreter" ή PHP / FI.
Τα ακόλουθα "Hello World!" πρόγραμμα είναι γραμμένο σε κώδικα PHP ενσωματωμένο σε ένα έγγραφο HTML:

```
<!DOCTYPE html>
<html>
 <head>
  <title>PHP "Hello, World!" program</title>
 </head>
 <body>

 <?php echo '<p>Hello World</p>'; ?>

</body>
</html>
```

Ο διερμηνέας PHP εκτελεί μόνο κώδικα εντος των οριοθετήσεων του. Οτιδήποτε έξω από τις οριοθετήσεις δεν επεξεργάζεται από την PHP, αν και το κείμενο που δεν είναι PHP υπόκειται ακόμη σε δομές ελέγχου που περιγράφονται στον κώδικα PHP. Οι πιο συνηθισμένοι οριοθέτες είναι "<? Php" για ανοιγμα και " ?>" για να κλείσιμο τμημάτων PHP.

Οι μεταβλητές προστίθενται με το σύμβολο του δολαρίου στην αρχη και δεν χρειάζεται να προσδιοριστουν εκ των προτέρων.
Σε αντίθεση με τα ονόματα των functions και των κλάσεων, τα ονόματα μεταβλητών είναι case sensitive. Τα σύμβολα ("") και (') οταν παρεμβάλουν την τιμή μιας μεταβλητής τη μετατρέπουν σε συμβολοσειρά.
Η PHP αντιμετωπίζει τα new lines ως κενά και οι δηλώσεις τερματίζονται με ένα ερωτηματικό. Η PHP έχει τρεις τύπους σύνταξης σχολίων: / * * / σημειώνει μπλοκ

με ενσωματωμένα σχόλια. // ή # χρησιμοποιούνται για σχόλια μιας γραμμής. Η αναφορά echo είναι μια από τις πολλές εντολές που παρέχει η PHP για έξοδο κειμένου.

Όσον αφορά τις λέξεις-κλειδιά και τη σύνταξη, η PHP έχει παρόμοιο στυλ σύνταξης με τη C.Οι συνθήκες if, for, while loops και οι επιστροφές των functions είναι παρόμοιες στη σύνταξη με γλώσσες όπως η C, C ++, C #, Java και Perl.

ΠΗΓΗ:https://en.wikipedia.org/wiki/PHP

Όταν ενσωματωθεί κώδικας PHP σε μια ιστοσελίδα, εκτελούνται οι παρακάτω διαδικασίες:

- Ο Web browser ενός χρήστη κάνει μία http:// αίτηση για μια συγκεκριμένη php Web σελίδα. (πχ.http://www.eclass.uth.gr/index.php)
- Ο Web διακομιστής λαμβάνει την αίτηση για την σελίδα php, ανακαλεί το αρχείο και το περνά στην μηχανή php για επεξεργασία.
- Η μηχανή php αρχίζει την ανάλυση του php κώδικα επικοινωνώντας αν υπάρχει ανάγκη με τον διακομιστή της βάσης δεδομένων.
- Η μηχανή php σταματά την εκτέλεση του προγράμματος επιστρέφοντας στον Web διακομιστή την τελική HTML σελίδα.
- Ο Web διακομιστής περνά την html σελίδα ξανά στο Web browser όπου ο χρήστης μπορεί να διαβάσει το αποτέλεσμα της php σελίδας.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP, η επιλογή της οποίας βασίστηκε στα πλεονεκτήματα που παρουσιάζει:

1.Ανοιχτός Κώδικας

Η PHP είναι Ανοικτού Κώδικα. Αυτό σημαίνει ότι είναι άμεσα διαθέσιμη, και εντελώς δωρεάν! Αν σχεδιάζετε να προσλάβετε προγραμματιστές PHP και αυτοί διαθέτουν ικανοποιητική εμπειρία σε C και Java, τότε θα έχουν εξαιρετική απόδοση και στην PHP. Για να πούμε την αλήθεια, η PHP είναι πραγματικά πολύ

απλούστερη από τις υπόλοιπες και έτσι μπορεί να είστε σίγουρος ότι θα έχετε καταπληκτικά αποτελέσματα.

2.Ανεξάρτητη Πλατφόρμας

Η PHP είναι ανεξάρτητη πλατφόρμα, επομένως δίνει τη δυνατότητα λειτουργίας σε διάφορα λειτουργικά συστήματα. Λειτουργεί άριστα σε πλατφόρμες LINUX, UNIX και Windows. Επίσης, συνδέεται εύκολα με Apache/MySQL.

3.Υποστήριξη

Η PHP, όπως προαναφέρθηκε, είναι πολύ δημοφιλής. Επομένως, υπάρχει πλήθος αναφορών και κατευθυντήριων οδηγιών, οι οποίες είναι διαθέσιμες στο Διαδίκτυο. Μπορεί επίσης να βρει κάποιος ομάδες και φόρουμ υποστήριξης της PHP. Εάν κολλήσετε κάπου, υπάρχει πάντα αρκετή υποστήριξη από διαδικτυακές βιβλιοθήκες για να ανταπεξέλθετε. Αυτός ο πλούτος διαδικτυακής υποστήριξης είναι επίσης διαθέσιμος σε διάφορες γλώσσες.

4.Τεράστια Κοινότητα

Η κοινότητα της PHP είναι μεγάλη. Εξαιτίας της δημοτικότητάς της, υπάρχουν πολλοί προγραμματιστές, συνεργάτες και χρήστες PHP αλλά και εργοδότες που προσφέρουν θέσεις εργασίας στην PHP. Αν ψάχνετε να προσλάβετε προγραμματιστές PHP, τότε σίγουρα δεν είστε μόνος.

5.Ενσωμάτωση

Αυτό που αποτελεί μία ενδιαφέρουσα ιδιότητα, είναι ότι η PHP μπορεί να ενσωματωθεί εύκολα σε HTML. Το γεγονός αυτό καθιστά πολύ εύκολη τη μετατροπή ενός ήδη υπάρχοντος στατικού ιστότοπου, σε έναν δυναμικό ιστότοπο με έντονα χαρακτηριστικά.

6.Σταθερότητα, Ευελιξία και Ταχύτητα

Η Σταθερότητα, η Ευελιξία και η Ταχύτητα αποτελούν τις κύριες ιδιότητες που προσελκύουν τους ιδιοκτήτες επιχειρήσεων να επιλέξουν την PHP.

7.Γρήγορη

Καθώς η ανάπτυξη της PHP είναι πολύ γρήγορη, ο χρόνος εκτέλεσης της εργασίας είναι μικρός. Όταν προσλαμβάνετε προγραμματιστές PHP, η αποδοτικότητά τους σας προσφέρει τη διαδικτυακή λύση που πιθανώς ψάχνατε εδώ και πολύ καιρό.

8.Επεκτάσεις

Η PHP διαθέτει διάφορες επεκτάσεις και είναι άκρως κλιμακώσιμη. Όταν προσλαμβάνετε προγραμματιστές PHP, αυτοί διασφαλίζουν ότι τα αποτελέσματά σας είναι μετρήσιμα. Αυτό σας βοηθά να υπολογίσετε την Επένδυση της Απόδοσής (ROI) και σας δίνει καλύτερη θέση έναντι του ανταγωνισμού.

3.5 JavaScript

Η **JavaScript** (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές.[1] Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).
Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript. Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web).

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη της γλώσσας προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

Αρχικά χρησιμοποιήθηκε κατα κυριο λογο για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού). Παρ όλα αυτά Η χρήση της Javascript στο διακομιστή εμφανίζεται, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML <script type="text/javascript"> και </script>.
Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Hello world!":

```
<script type="text/javascript">
alert('Hello world!');
</script>
```

ΠΗΓΗ:https://en.wikipedia.org/wiki/JavaScript

Το **Ajax** (JavaScript και XML) είναι ένα σύνολο τεχνικών ανάπτυξης ιστού που χρησιμοποιεί πολλές τεχνολογίες ιστού στην πλευρά του πελάτη για τη δημιουργία ασύγχρονων εφαρμογών ιστού. Με το Ajax, οι εφαρμογές ιστού μπορούν να αποστέλλουν και να ανακτούν δεδομένα από έναν διακομιστή ασύγχρονα (στο παρασκήνιο) χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά της υπάρχουσας σελίδας. Με την αποσύνδεση του στρώματος ανταλλαγής δεδομένων από το στρώμα παρουσίασης, το Ajax επιτρέπει στις ιστοσελίδες και, κατ 'επέκταση, στις εφαρμογές ιστού να αλλάζουν δυναμικά το περιεχόμενο χωρίς να χρειαστεί να επαναφορτωθεί ολόκληρη η σελίδα. Στην πράξη, οι σύγχρονες εφαρμογές χρησιμοποιούν συνήθως JSON αντί για XML.

Το Ajax δεν είναι μια ενιαία τεχνολογία, αλλά μια ομάδα τεχνολογιών. Το HTML και το CSS μπορούν να χρησιμοποιηθούν σε συνδυασμό για να επισημάνουν πληροφορίες παρουσιάζοντας τες με στυλ. Η ιστοσελίδα μπορεί στη συνέχεια να τροποποιηθεί από τη JavaScript για να εμφανιστεί δυναμικά και να επιτρέψει στο χρήστη να αλληλεπιδράσει με τις νέες πληροφορίες. Το ενσωματωμένο αντικείμενο XMLHttpRequest ή από το 2017 η νέα λειτουργία "fetch ()" μέσα στη JavaScript χρησιμοποιείται συνήθως για την εκτέλεση του Ajax σε ιστοσελίδες, επιτρέποντας στους ιστότοπους να φορτώνουν περιεχόμενο στην οθόνη χωρίς ανάγκη ανανέωση της σελίδας. Ο Ajax δεν είναι μια νέα τεχνολογία, ή διαφορετική γλώσσα, απλώς υπάρχουσες τεχνολογίες που χρησιμοποιούνται με νέους τρόπους.

ΠΗΓΗ:https://en.wikipedia.org/wiki/Ajax_(programming)

3.6 MySQL

Η **MySQL** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.Η MySQL είναι γραμμένη σε C και C ++ και λειτουργεί σε πολλές πλατφόρμες συστημάτων.
Το σύστημα διαχείρισης MySQL λοιπόν, δίνει την δυνατότητα της αποθήκευσης, αναζήτησης,ταξινόμησης, ομαδοποίησης και ανάκλησης δεδομένων με βάση την γλώσσα ερωτημάτων SQL. Το γεγονός ότι η MySQL είναι σχεσιακή συνεπάγεται ότι η οργάνωση των δεδομένων γίνεται σε διαφορετικούς πίνακες οι οποίοι σχετίζονται μεταξύ τους με κάποιο συγκεκριμένο τρόπο. Η MySQL επιπλέον μπορεί να ελέγχει την πρόσβαση στα δεδομένα, εξασφαλίζοντας έτσι την δυνατότητα η πρόσβαση να γίνεται από διαφορετικούς χρήστες. Κάθε χρήστης έχει συγκεκριμένα δικαιώματα πάνω στις βάσεις δεδομένων τα οποία του τα δίνει η MySQL.

ΠΗΓΕΣ:https://en.wikipedia.org/wiki/MySQL
https://www.mysql.com

Στην παρούσα εφαρμογή χρησιμοποιήθηκε η MySQL, η επιλογή της οποίας βασίστηκε στα πλεονεκτήματα που παρουσιάζει:

1.Απόδοση
Η MySQL είναι αναμφίβολα γρήγορη .Μπορείτε να δείτε την σελίδα δοκιμών στο http://web.mysql.com/benchmark.html . Πολλές από τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό .

2.Χαμηλό κόστος

Η MySQL είναι διαθέσιμη δωρεάν ,με άδεια ανοικτού κώδικα (open source), ή με χαμηλό κόστος ,αν η εμπορική άδεια απαιτείται από την εφαρμογή

3.Ευκολία Χρήσης

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL .Αν κάποιος χρησιμοποιεί ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων ,δεν θα έχει πρόβλημα να προσαρμοστεί σε αυτό .Η MySQL είναι ευκολότερη από παρόμοια προϊόντα .

4.Μεταφερσιμότητα

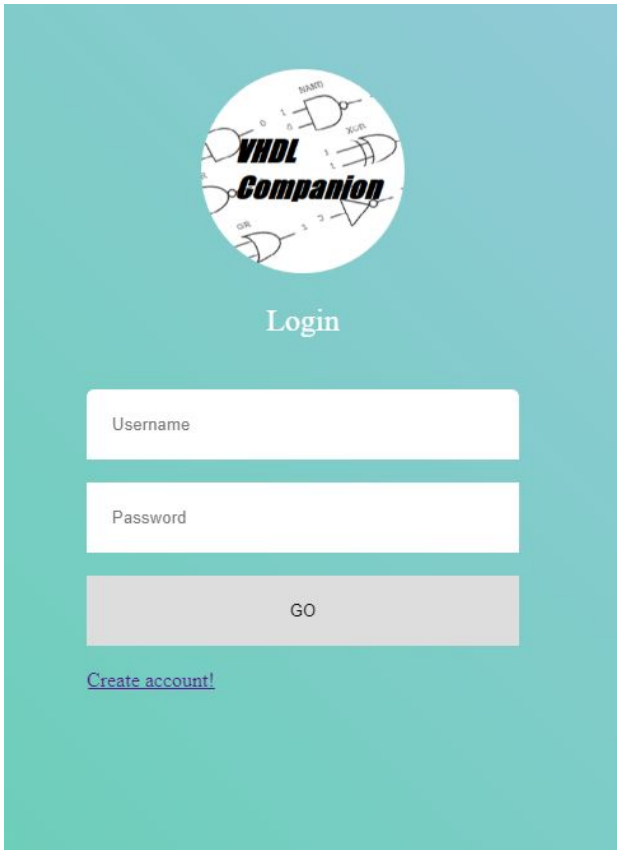
Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα UNIX ,καθώς επίσης και στα Microsoft Windows .

5.Κώδικας Προέλευσης

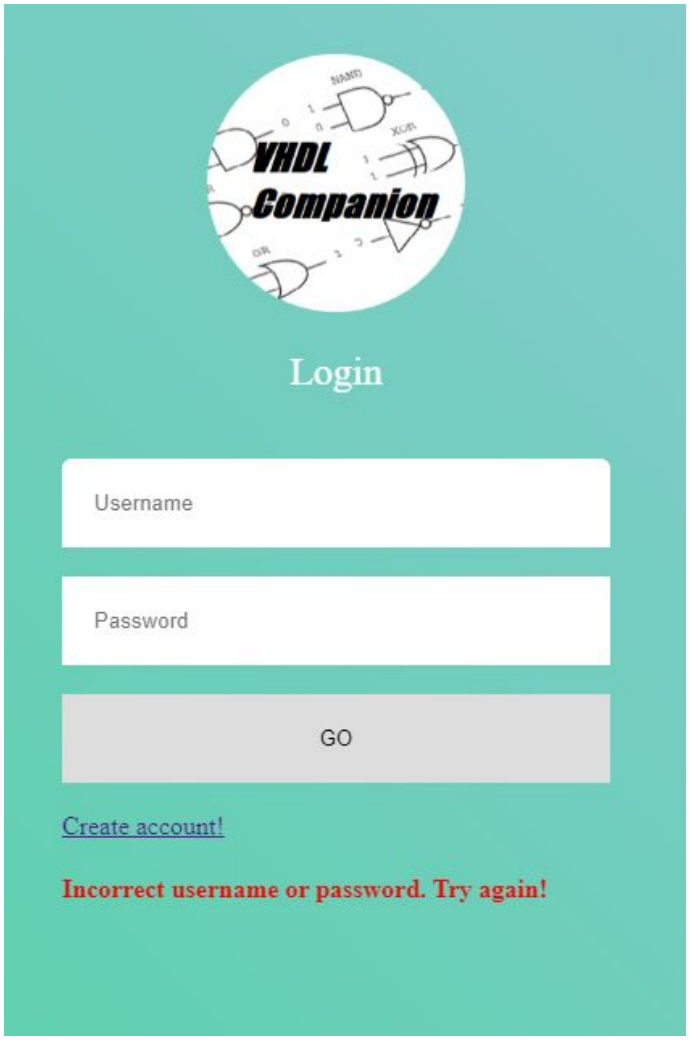
Όπως και με την PHP, ο προγραμματιστής μπορεί να πάρει και να τροποποιήσει τον πηγαίο κώδικα της MySQL, προσαρμόζοντάς τον στις ανάγκες του.

4 ΣΧΕΔΙΑΣΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

4.1 Log in & Register



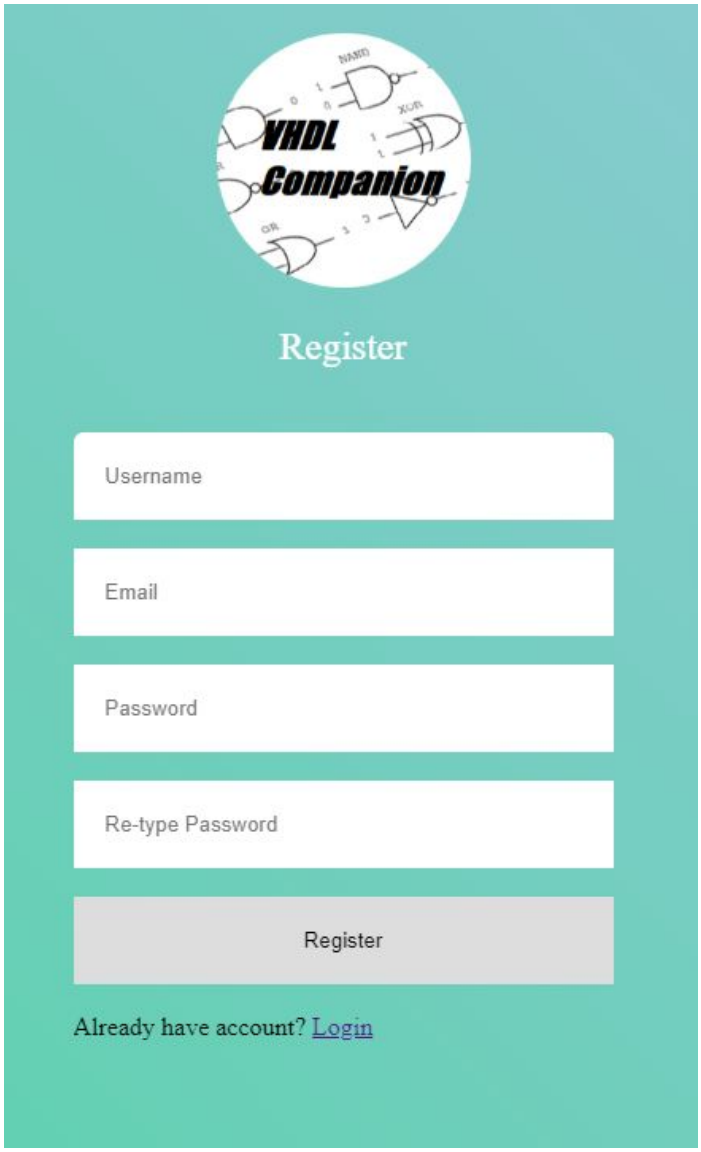
Κατά τη είσοδο του χρήστη στην εφαρμογή η πρώτη σελίδα που παρουσιάζεται είναι η Log in (σελίδα εισαγωγής χρήστη). Σε αυτή ο χρήστης καλείται να συμπληρώσει τα πεδία username και password προκειμένου να συνεχίσει την περιήγηση του στην εφαρμογή. Είναι απαραίτητο να συμπληρωθούν τα πεδία σωστά. Με τα ίδια δηλαδή username και password που έχουν εισαχθεί κατά την εγγραφή του χρήστη. Σε περίπτωση που δεν γίνει αυτό εμφανίζεται μήνυμα λάθους, όπως φαίνεται παρακάτω:



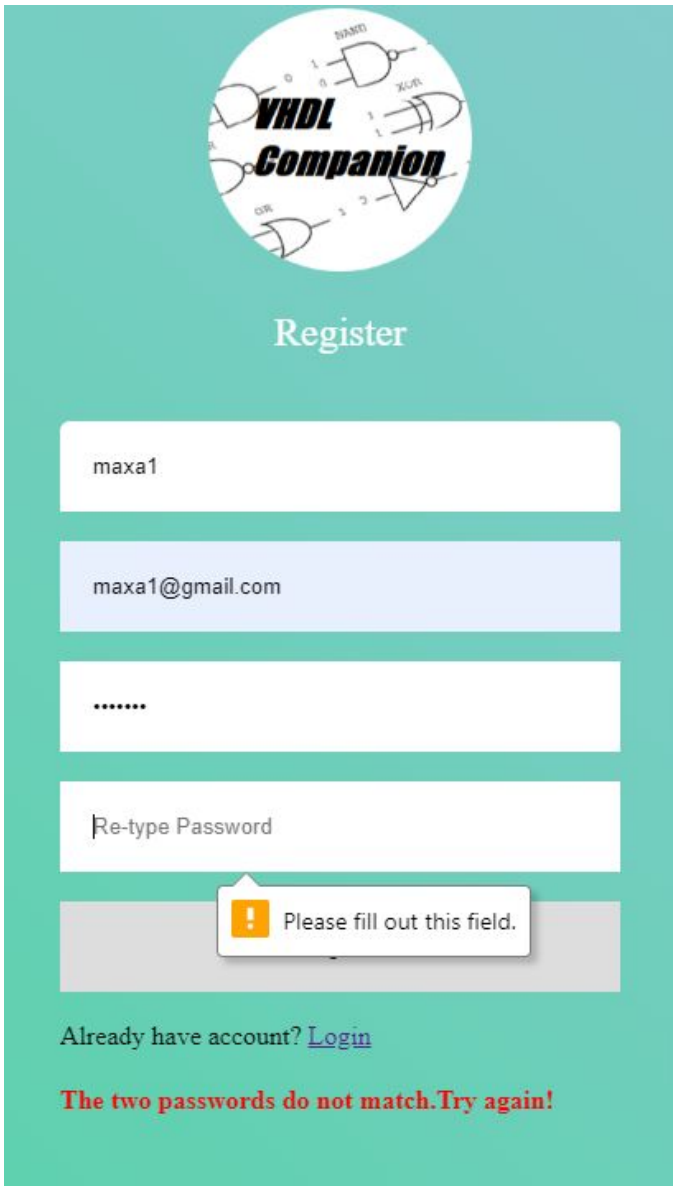
Πληκτρολογώντας σωστά τα κείμενα στα πεδία και πατώντας το κουμπί 'Go' γίνεται άμεση εισαγωγή στο σύστημα και ο χρήστης παραπέμπεται την main page της εφαρμογής.

Αν ο χρήστης δεν διαθέτει λογαριασμό θα πρέπει να πατήσει τον σύνδεσμο 'Create account!' ο οποίος θα τον παραπέμψει στη σελίδα register της εφαρμογής ώστε να δημιουργήσει έναν νέο λογαριασμό

.



Κατά την εγγραφή του ο χρήστης καλείται να συμπληρώσει όλα τα πεδία της φόρμας και να επαναλάβει σωστά τον κωδικό πρόσβασης του για λόγους ασφάλειας. Αν αυτό δεν γίνει η σελίδα βγάζει μήνυμα λάθους και τον παραπέμπει στο να ξαναεισάγει τα στοιχεία από την αρχή.



Επιπλέον το username είναι το κύριο αναγνωριστικό του κάθε χρήστη και πρέπει να είναι μοναδικό. κατά την εγγραφή αν ο χρήστης χρησιμοποιήσει ίδιο username με κάποιον άλλο προηγούμενο η sql θα εμφανίσει μήνυμα λάθους.

Κωδικας για Log in page:

```html
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <link rel="icon" href="xoricon.png">
        <link rel="stylesheet" type="text/css" href="reglog.css">
        <title>Login Page</title>
    </head>
    <body>

        <div class="user">
            <header class="user__header">
                <img src="logocircle.png" height="160" width="160">
                <h2 class="user__title">Login</h2>
            </header>
            <form class="form" method="POST" enctype="multipart/form-data">

                <input class="form__input" placeholder="Username" type="text" name="username" required><br>
                <input class="form__input" placeholder="Password" type="password" name="pass" required><br>
                <input class="btn" type="submit" name="submit" value="GO"><br>
                <a href="register.php">Create account!</a><br>
```

.

```php
                <?php

                $servername = "localhost";

                if(isset($_POST['submit'])){

                    session_start();
                    $username = $_POST['username'];
                    $pass = $_POST['pass'];

                    $pass = md5($pass);
                    $sql = "SELECT * FROM users WHERE username = '{$username}' and pass = '{$pass}'";
                    try{
                        $conn = new PDO("mysql:host=$servername;dbname=vhdldb", "root", "");
                        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                        #echo "connected successfully";
                        $stmt = $conn->prepare($sql);
                        $stmt->execute();
                        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);

                        if(empty($result)){
                            echo "<h4 style='color: red'>Incorrect username or password. Try again!</h4>";
                        }else{
                            #print_r($result);
                            $_SESSION['message'] = "You are now logged in!";
                            $_SESSION['username'] = $username;
                            echo $_SESSION['username']." ".$_SESSION['message'];
                            header("location:main.php");
                        }
                    }catch(PDOException $e){
                        echo "Connection failed: ".$e->getMessage();
                    }
                }
                ?>
            </form>
        </div>
    </body>
</html>
```

Κωδικας για Register page:

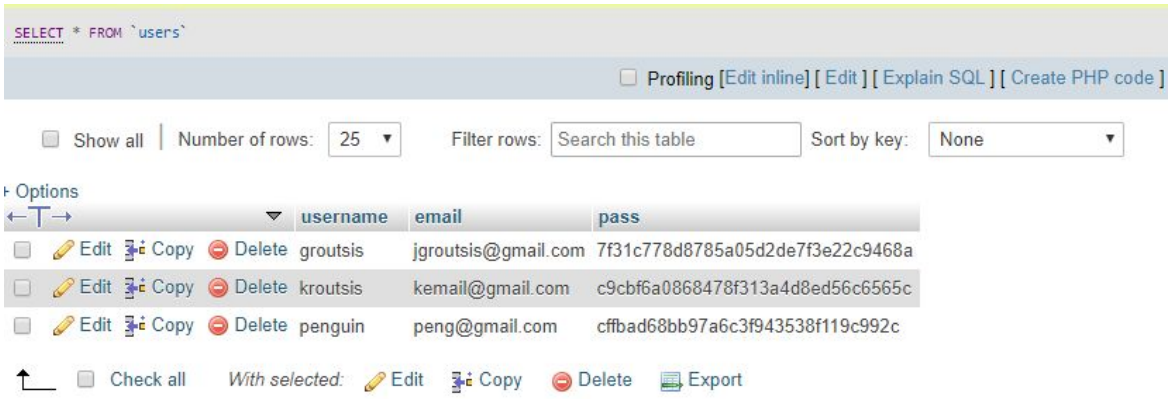
```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <link rel="icon" href="xoricon.png">
        <link rel="stylesheet" type="text/css" href="reglog.css">
        <title>Register Page</title>
    </head>
    <body>
        <div class="user">
            <header class="user__header">
                <img src="logocircle.png" height="160" width="160">
                <h2 class="user__title">Register</h2>
            </header>
            <form class="form" method="POST" enctype="multipart/form-data">

                <input class="form__input" placeholder="Username" type="text" name="username" required><br>
                <input class="form__input" placeholder="Email" type="email" name="email" required><br>
                <input class="form__input" placeholder="Password" type="password" name="pass" required><br>
                <input class="form__input" placeholder="Re-type Password" type="password" name="pass2" required><br>
                <input class="btn" type="submit" name="register" value="Register"><br>
                Already have account? <a href="login.php">Login</a>
```

```
            <?php

            $servername = "localhost";

            if(isset($_POST['register'])){

                session_start();
                $username = $_POST['username'];
                $email = $_POST['email'];
                $pass = $_POST['pass'];
                $pass2 = $_POST['pass2'];

                if($pass == $pass2){
                    $pass = md5($pass); #encrypt password
                    #$query = "SELECT * FROM users WHERE username = '{$username}';";
                    $sql = "INSERT INTO users(username, email, pass)VALUES('{$username}', '{$email}', '{$pass}');";
                    try{
                        $conn = new PDO("mysql:host=$servername;dbname=vhdldb", "root", "");
                        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                        #echo "connected successfully";
                        $conn->exec($sql);

                        $_SESSION['message'] = "You are now logged in!";
                        $_SESSION['username'] = $username;

                        header("location:main.php");

                    }catch(PDOException $e){
                        echo "<br>Connection failed: ".$e->getMessage();
                    }
                }else{
                    $_SESSION['message'] = "<h4 style='color: red'>The two passwords do not match.Try again!</h4>";
                    echo $_SESSION['message'];
                }
            }
            ?>
            </form>

        </div>

    </body>
</html>
```

Στη σχεδίαση μου κατά την εισαγωγή χρήστη τα στοιχεία του εισάγονται και στη βάση δεδομένων στο table users με τρόπο που φαίνεται στον παραπάνω κώδικά.

Το όνομα χρήστη (username) αποτελεί primary key το μαιλ (email) δηλώνεται με τύπο email και στη βάση δεδομένων και στη φόρμα και το password (pass) κρυπτογραφείται με md5 encryption αυτόματα από την php.

Table users απο τη βαση δεδομενων vhdldb:



4.2 Index - Main Page

Όταν η είσοδος του χρήστη στο σύστημα εγκριθεί η login ή η register page θα τον παραπέμψει στη main page της εφαρμογής.

Η main page είναι το μέρος όπου όλες οι λειτουργίες της εφαρμογής λαμβάνουν χώρα. Τα πρώτα πράγματα που βλέπει κάποιος είναι το λογότυπο, την ημερομηνία, έναν ευγενικό χαιρετισμό προς το πρόσωπο του από τη μεριά της

εφαρμογής, το μενού, έναν σύνδεσμο Log out για έξοδο από την εφαρμογή, έναν σύνδεσμο Upload Files για ανέβασμα αρχείων καθώς και τις λειτουργίες που διαθέτει η εφαρμογή, άλλα απενεργοποιημένες αφού δεν υπάρχουν ακόμα αρχεία για να επεξεργαστούν.
Αν ο χρήστης προσπαθήσει να μπει στην εφαρμογή με μη νόμιμο τρόπο, όχι δηλαδή μέσω login η register αλλα για παράδειγμα γράφοντας το URL μιας υποσελίδας της εφαρμογής στον browser, η εφαρμογή τον στέλνει στη σελίδα log in. Όταν εισέλθει σωστά, η εφαρμογή δημιουργεί μια $_SESSION μεταβλητή με το username του και τη χρησιμοποιεί έως ότου κάνει log out.
Το λογότυπο της σελίδας λειτουργεί σαν κουμπί ανανέωσης.

Κώδικας των στοιχείων που εμφανίζονται πάνω από το μενού:

```
<?php session_start();
if(!isset($_SESSION['username'])){header("Location:Login.php");}else{ $username = $_SESSION['username'];}?>
<div class="top">
    <img class="logo__main" src="logocircle.png" height="130" width="130" onClick="location.reload();" >
    <div class="outer">

    <span class="user__name" id="datetime" ></span>
    <h2 class="user__name">Hello <?php echo $username; ?> !</h2>

    <div class="user__in">
        <a href="logout.php">Log out</a><br>
        <a href="upload.php" target="popup" onclick="window.open('upload.php','popup','width=600,height=400');
    </div>
```

Κώδικας εμφάνισης της ημερομηνίας:

```
var dt = new Date();
document.getElementById("datetime").innerHTML = dt.toLocaleDateString();
```

4.2.1 Menu

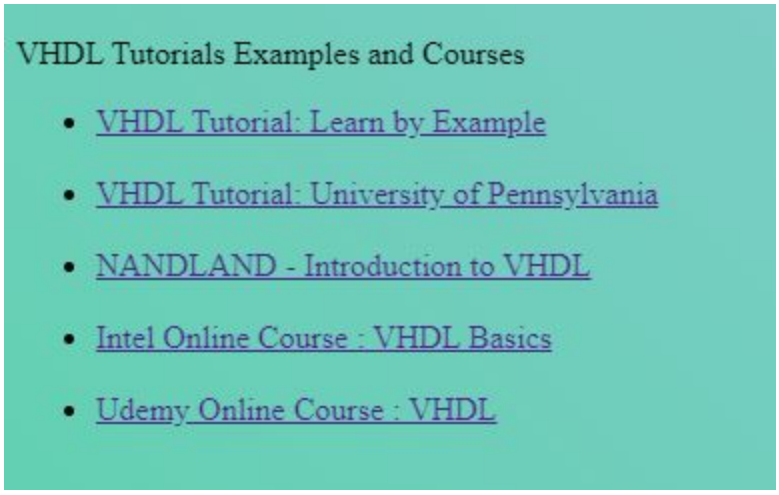
Στο menu της εφαρμογής μπορεί κανείς να βρει χρήσιμες πληροφορίες σχετικά με την γλώσσα προγραμματισμού VHDL , μαθήματα και παραδείγματα , μαθήματα σε μορφή βίντεο, βιβλία για την εκμάθηση της γλώσσας, προγράμματα και εφαρμογές σχετικές με τη VHDL, και πληροφορίες σχετικές με την εφαρμογή.

1. VHDL Wiki

Το πρώτο κουμπί του μενού παραπέμπει τον χρήστη στην ιστοσελίδα: https://en.wikipedia.org/wiki/VHDL που είναι το άρθρο της wikipedia για την VHDL.

2. Tutorials & Examples

Το δεύτερο κουμπί του μενού παρουσιάζει στον χρήστη μια λίστα με links που περιέχει παραδείγματα και μαθήματα για την VHDL.

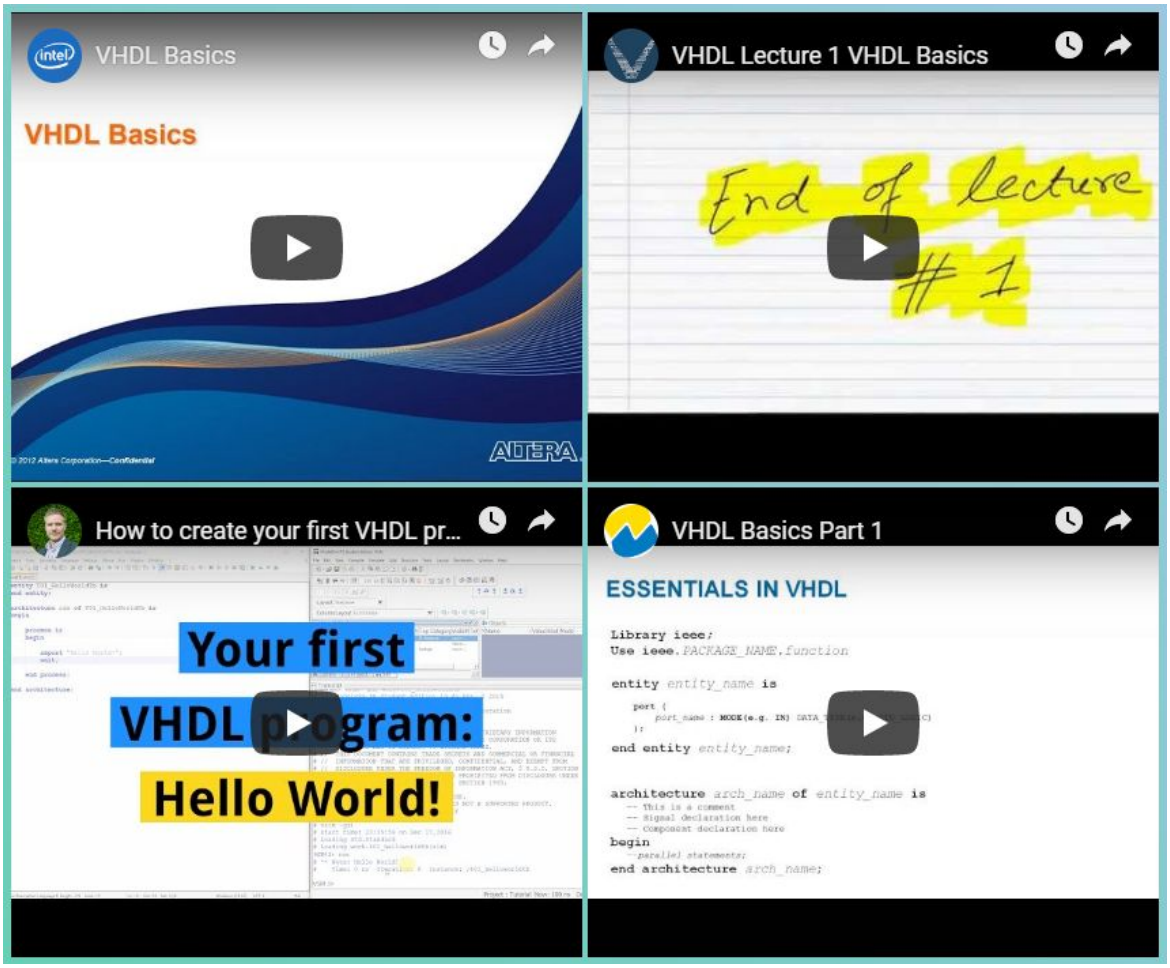


Τα links είναι τα εξής:

- http://esd.cs.ucr.edu/labs/tutorial/
- https://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
- https://www.nandland.com/vhdl/tutorials/tutorial-introduction-to-vhdl-for-beginners.html
- https://www.intel.com/content/www/us/en/programmable/support/training/course/ohdl1110.html
- https://www.udemy.com/topic/vhdl/

3. Video Tutorials

Το τρίτο κουμπί του μενού παρουσιάζει στον χρήστη 4 tutorial - μαθήματα σε μορφή βίντεο για την εκμάθηση VHDL.



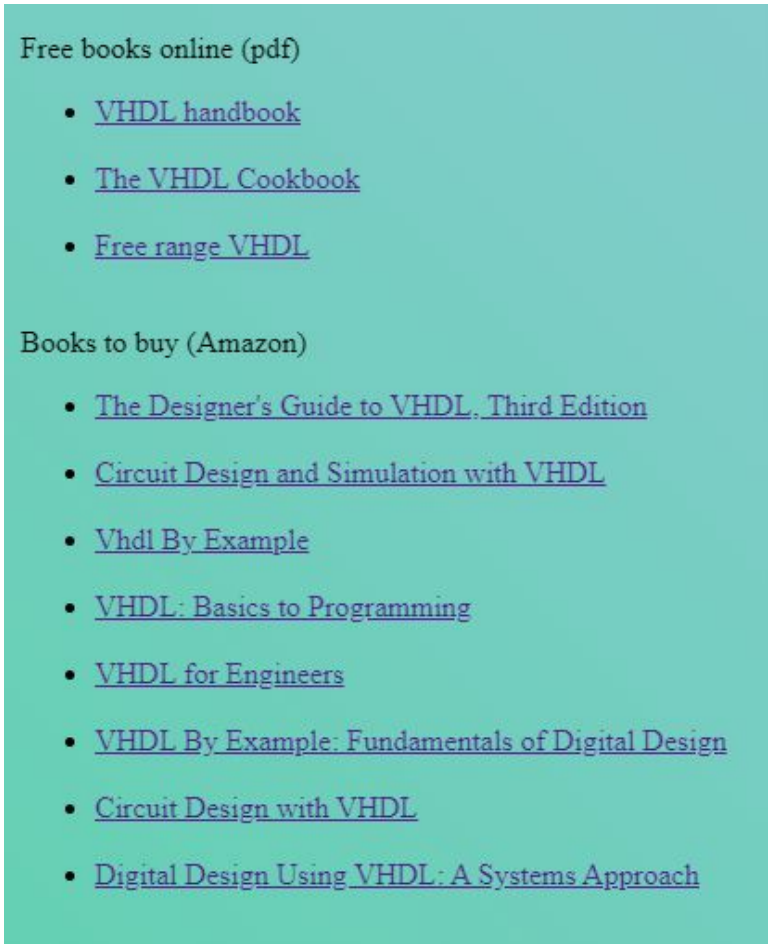
Τα βίντεο είναι από την Ιστοσελίδα Youtube και υπάρχει η δυνατότητα προβολής τους στην ίδια την εφαρμογή (και σε πλήρη οθόνη) αλλά και η παραπομπή από αυτά στην αντίστοιχη σελίδα του Youtube.

Τα links των βίντεο είναι τα εξής:

- https://www.youtube.com/embed/zm-RA6BsYmc
- https://www.youtube.com/embed/BDq8-QDXmek
- https://www.youtube.com/embed/h4ZXge1BE80
- https://www.youtube.com/embed/vXF0yDeQ-Ms

4. VHDL Books

Το τέταρτο κουμπί του μενού παρουσιάζει στον χρήστη δύο λίστες με βιβλία για τη γλώσσα.



Η πρώτη λίστα περιέχει links με δωρεάν βιβλία που είναι άμεσα διαθέσιμα σε μορφή pdf. Η δεύτερη λίστα περιέχει links με βιβλία που είναι διαθέσιμα για αγορά μέσα από την ιστοσελίδα του Amazon.

Τα links των βιβλίων είναι τα εξής:

- https://www.csee.umbc.edu/portal/help/VHDL/VHDL-Handbook.pdf
- https://www.ics.uci.edu/~alexv/154/VHDL-Cookbook.pdf
- http://freerangefactory.org/pdf/df344hdh4h8kjfh3500ft2/free_range_vhdl.pdf

- https://www.amazon.com/Designers-Guide-Edition-Systems-Silicon/dp/0120887851
- https://www.amazon.com/Circuit-Design-Simulation-VHDL-Press/dp/0262014335
- https://www.amazon.com/Vhdl-Example-Blaine-Readler/dp/0983497354
- https://www.amazon.com/VHDL-Basics-Programming-Gaganpreet-Kaur-ebook/dp/B00BIZS378
- https://www.amazon.com/VHDL-Engineers-Kenneth-L-Short/dp/0131424785
- https://www.amazon.com/VHDL-Example-Fundamentals-Digital-Design/dp/0982497059
- https://www.amazon.com/Circuit-Design-VHDL-Volnei-Pedroni/dp/0262162245
- https://www.amazon.com/Digital-Design-Using-VHDL-Approach/dp/1107098866

5. Implementations & Simulators

Το πέμπτο κουμπί του μενού παρουσιάζει στον χρήστη δύο λίστες με Implementations and Simulators για την VHDL

Η πρώτη λίστα περιέχει open source προγράμματα διαθέσιμα δωρεάν ενώ η δεύτερη λίστα commercial προγράμματα.

Τα links των προγραμμάτων είναι τα εξής:

- http://ghdl.free.fr/index.html
- https://github.com/nickg/nvc
- http://freehdl.seul.org/
- https://www.aldec.com/en/products/fpga_simulation/active-hdl
- https://www.cadence.com/content/cadence-www/global/en_US/home/tools/system-design-and-verification/simulation-and-testbench-verification/incisive-enterprise-simulator.html
- https://www.mentor.com/products/fpga/model/

6. About

Το πέμπτο κουμπί του μενού παρουσιάζει στον χρήστη πληροφορίες σχετικά με την εφαρμογή και τον δημιουργό της.Οι πληροφορίες αυτές είναι ίδιες με ένα μέρος της εισαγωγής της γραπτής αναφοράς της πτυχιακής εργασίας

Τα κουμπιά του μενού έχουν λειτουργία on - off και τα αποτελέσματα τους εμφανίζονται ανάμεσα στο μενού και τη διπλή γραμμή που χωρίζει το πάνω μέρος της σελίδας από τις λειτουργίες και τα αρχεία.

Κώδικας παρουσίασης του μενού:

```
<ul class="menu">
    <!-- <input type="submit" name="correct" value="Correct"><br> -->
    <li><input type="button" id="vhdlwiki" onclick="window.open('https://en.wikipedia.org/wiki/VHDL');" value="VHDL Wiki"/>
    <li><input type="button" id="vhdltutorials" onclick="showDiv()" value="Tutorials & Examples"/></li>
    <li><input type="button" id="vhdlvideo" onclick="showDiv1()" value="Video Tutorials"/></li>
    <!--<input type="button" id="vhdllessons" onclick="showDiv2()" value="VHDL Lessons"/>-->
    <li><input type="button" id="vhdlbooks" onclick="showDiv2()" value="VHDL Books"/></li>
    <li><input type="button" id="vhdlsimulators" onclick="showDiv3()" value="Implementations & Simulators"/></li>
    <li><input type="button" id="about" value="About"/></li>
</ul>
<div id="tutorials" style="display:none">
<br>VHDL Tutorials Examples and Courses<br>
<ul>
</div>
<div id="simulators" style="display:none">
<br>Open Source Implementations and Simulators<br>
<ul>
Commercial Implementations and Simulators<br>
<ul>
</div>
<div id="books" style="display:none">
<br>Free books online (pdf)<br>
<ul>
Books to buy (Amazon)<br>
<ul>
</div>
```

Κώδικας για τη λειτουργία on - off των κουμπιών του μενού:

```
function showDiv() {
    var z = document.getElementById("tutorials");
    if (z.style.display === "none") {
        z.style.display = "block";
    } else {
        z.style.display = "none";
    }
}
```

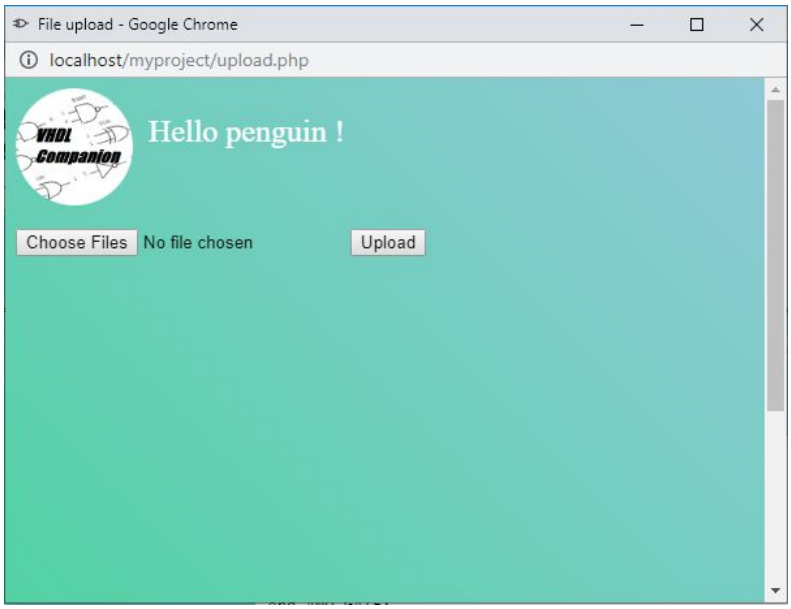
4.2.2 Log out

Ο σύνδεσμος Log out διαγράφει τη μεταβλητή $_SESSION που είχε δημιουργηθεί κατά την εισαγωγή του χρήστη στην εφαρμογή και τον παραπέμπει στη σελίδα Log in.

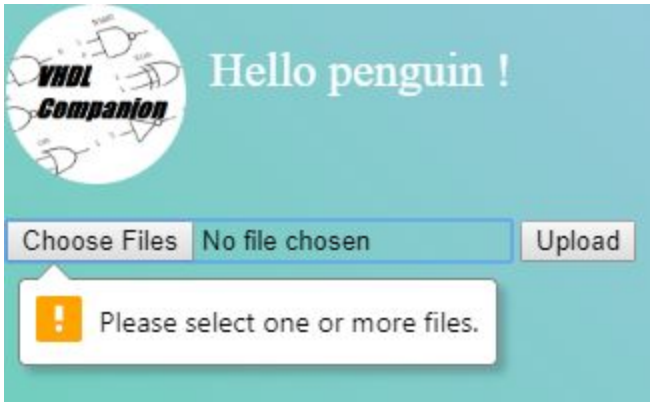
```php
<?php

    session_start();
    session_unset();
    session_destroy();

    header("location:login.php");

    exit();

?>
```

4.2.3 Upload Files

Όταν ο χρήστης πατήσει τον σύνδεσμο Upload Files ο browser ανοίγει ένα μικρό pop up παράθυρο.



Όταν ο χρήστης πατώντας την επιλογή Choose Files μπορεί να επιλέξει ένα η και παραπάνω αρχεία για ανέβασμα. Αν πατήσει το κουμπί Upload χωρίς να έχει επιλέξει αρχεία εμφανίζεται προειδοποιητικό μήνυμα.



Όταν τα αρχεία επιλεχθούν από τον χρήστη και στη συνέχεια πατήσει το κουμπί Upload, το σύστημα ελέγχει το μέγεθός τους, τον τύπο format τους και το αν υπάρχουν ήδη ανεβασμένα στο σύστημα αρχεία με το ίδιο όνομα.Οι επιτρεπόμενοι τύποι για τα αρχεία είναι οι εξής: '.txt' '.vhd' και '.vhdl'. Αν ο χρήστης έχει κάνει νέο λογαριασμό και αποθηκεύει πρώτη φορά αρχεία, το σύστημα δημιουργεί ένα νέο directory με το username του και αποθηκεύει εκεί τα αρχεία του. Αν ο χρήστης έχει ξανααποθηκεύσει αρχεία το σύστημα βρίσκει το σωστό directory και αποθηκεύει εκεί να νέα αρχεία. Ταυτόχρονα ενημερώνεται και η βάση δεδομένων με τα νέα αρχεία. Όταν η διαδικασία αυτή τελειώσει το pop up παράθυρο κλείνει και η main page ανανεώνεται και εμφανίζει να νέα αρχεία.

Ο κώδικας της Upload:

```php
<?php session_start();
if(!isset($_SESSION['username'])){header("Location:Login.php");}else{ $username = $_SESSION['username'];}?>

<img class="logo__main" src="logocircle.png" height="90" width="90">
<h2 class="user__name">  Hello <?php echo $username; ?> !</h2>
<!--<a href="logout.php">Log out</a>-->
<br><br><br><br><br><br>
<form method="POST" enctype="multipart/form-data" id="certform">
    <input type="file" name="files[]" required multiple>
    <input type="submit" name="upload" value="Upload" onclick="closeSelf();"><br>
</form>
```

```php
<?php

if((isset($_FILES['files']))&&(isset($_POST['upload']))){

    $servername = "localhost";
    $myfile = $_FILES['files'];
    $filecount = count($myfile["name"]);
    $uploaded = array();
    $allowed = array('txt','vhd','vhdl');

    for($i=0; $i<$filecount; $i++){

        $name = $myfile['name'][$i];
        $type = $myfile['type'][$i];
        $size = $myfile['size'][$i];
        $tmp_name = $myfile['tmp_name'][$i];
        $error = $myfile['error'][$i];

        $format = explode('.',$name);
        $format = strtolower(end($format));

        if(in_array($format, $allowed)){
            if($error === 0){
                if($size <= 2097152){

                    #$newname = $name.uniqid('',FALSE).".".$format;
                    if(!file_exists('uploads/'.$username.'/')){
                        mkdir('uploads/'.$username.'/', 0777, TRUE);
                    }
                    $destination = 'C:/xampp/htdocs/myproject/uploads/'.$username.'/'.$name;
```

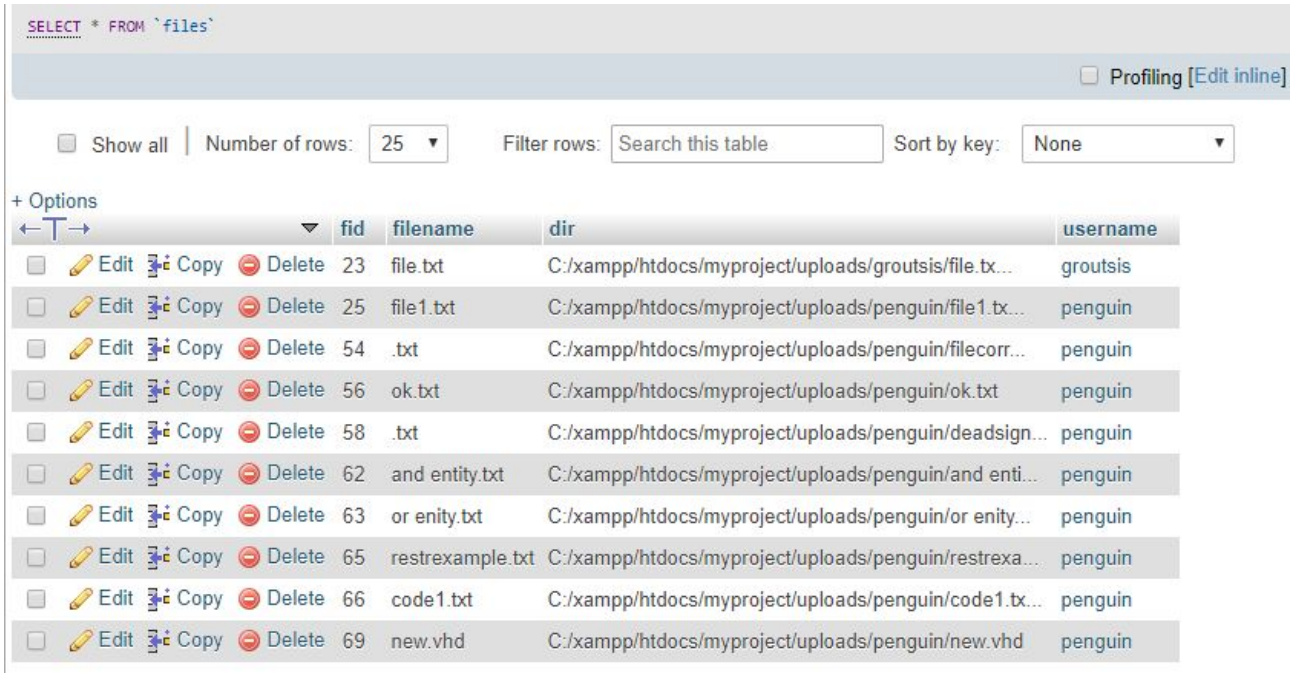
```
                    if(move_uploaded_file($tmp_name,$destination)){
                        $uploaded[$i] = $destination;
                        $sql = "INSERT INTO files(filename, dir, username)VALUES('{$name}', '{$destination}'
                        try{
                            $conn = new PDO("mysql:host=$servername;dbname=vhdldb", "root", "");
                            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                            #echo "connected successfully";
                            $conn->exec($sql);
                            $_SESSION['message'] = "You have uploaded your files!";

                            header("location:main.php");

                        }catch(PDOException $e){
                            echo "Connection failed: ".$e->getMessage();
                        }
                        echo "Upload of {$name} succed!<br>";
                    }else{
                        echo "Upload failed! File {$name} failed to upload!<br>";
                    }
                }else{
                    echo "Upload failed! File {$name} is too big!<br>";
                }
            }else{
                echo "Upload failed! There was an error uploading {$name} file!<br>";
            }
        }else{
            echo "Upload failed! File {$name} has an incompatible type!<br>";
        }
    }
    if(!empty($uploaded)){
        print_r($uploaded);
    }
}
?>
```

```
<script>

    $(document).ready(function () {
        $('#certform').ajaxForm(function () {
            window.opener.location.reload(false);
            window.close();
        });
    });

</script>
```

Στη σχεδίαση μου κατά το ανέβασμα αρχείων, αυτά εισάγονται και στη βάση δεδομένων στο table files με τρόπο που φαίνεται στον παραπάνω κώδικά.
Το fid αποτελεί primary key και είναι ένα μοναδικό id που δίνεται αυτόματα από τη βάση.Επιπλεων αποθηκεύονται το όνομα του αρχείου (filename) , το directory του (dir) και το όνομα του χρήστη (username) που αποθήκευσε το αρχείο.Το

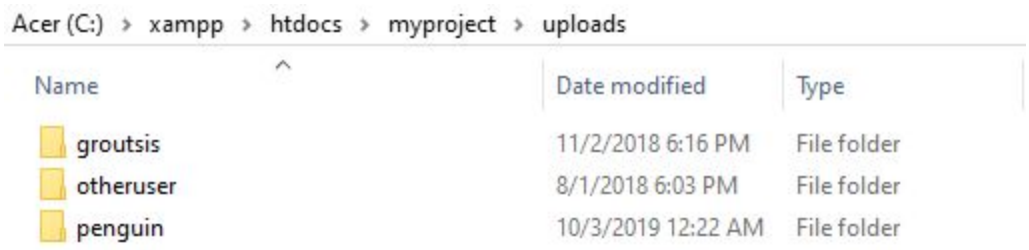
username είναι foreign key και συνδέει τον πίνακα αρχείων (files) με αυτόν των χρηστών (users).

Table files απο τη βαση δεδομενων vhdldb:



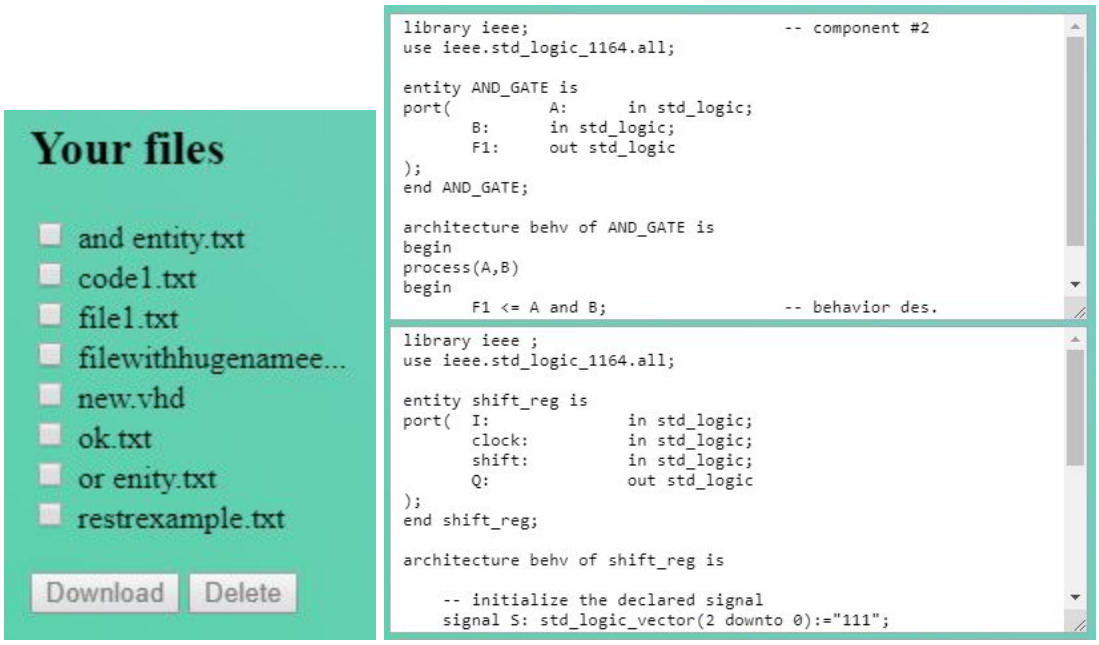
	fid	filename	dir	username
Edit Copy Delete	23	file.txt	C:/xampp/htdocs/myproject/uploads/groutsis/file.tx...	groutsis
Edit Copy Delete	25	file1.txt	C:/xampp/htdocs/myproject/uploads/penguin/file1.tx...	penguin
Edit Copy Delete	54	.txt	C:/xampp/htdocs/myproject/uploads/penguin/filecorr...	penguin
Edit Copy Delete	56	ok.txt	C:/xampp/htdocs/myproject/uploads/penguin/ok.txt	penguin
Edit Copy Delete	58	.txt	C:/xampp/htdocs/myproject/uploads/penguin/deadsign...	penguin
Edit Copy Delete	62	and entity.txt	C:/xampp/htdocs/myproject/uploads/penguin/and enti...	penguin
Edit Copy Delete	63	or enity.txt	C:/xampp/htdocs/myproject/uploads/penguin/or enity...	penguin
Edit Copy Delete	65	restrexample.txt	C:/xampp/htdocs/myproject/uploads/penguin/restrexa...	penguin
Edit Copy Delete	66	code1.txt	C:/xampp/htdocs/myproject/uploads/penguin/code1.tx...	penguin
Edit Copy Delete	69	new.vhd	C:/xampp/htdocs/myproject/uploads/penguin/new.vhd	penguin

Το Directory αποθήκευσης των αρχείων των χρηστών:

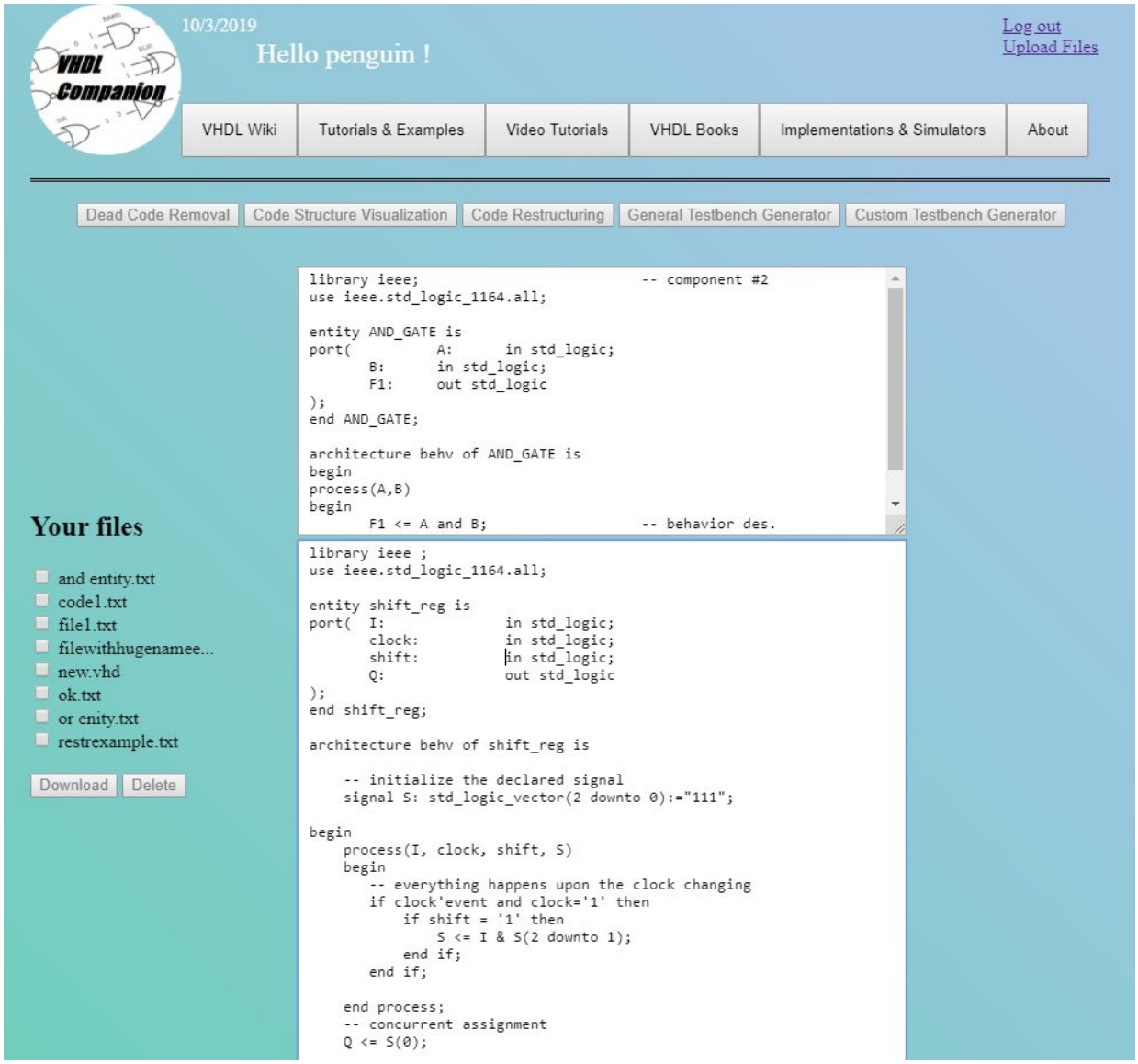


4.2.4 ΑΡΧΕΙΑ

Μετά το ανέβασμα αρχείων η εφαρμογή εμφανίζει τα ονόματα των αρχείων αλλά και το περιεχόμενο τους, σαν checkboxes και textareas αντίστοιχα.



Όταν το όνομα ενός αρχείου είναι μεγάλο συμπληρώνεται με αποσιωπητικά αλλά όταν ο χρήστης πάει τον δείκτη πάνω στο αρχείο το όνομα αυτού εμφανίζεται ολόκληρο. Επιπλέον όταν ο χρήστης κάνει κλικ πάνω στο textarea ενός αρχείου αυτό μεγαλώνει προκείμενου να είναι πιο ευανάγνωστο.


10/3/2019
Hello penguin !
Log out
Upload Files
VHDL Wiki
Tutorials & Examples
Video Tutorials
VHDL Books
Implementations & Simulators
About
Dead Code Removal
Code Structure Visualization
Code Restructuring
General Testbench Generator
Custom Testbench Generator
Your files
and entity.txt
code1.txt
file1.txt
filewithhugenamee...
new.vhd
ok.txt
or enity.txt
restrexample.txt
Download
Delete
library ieee; -- component #2
use ieee.std_logic_1164.all;
entity AND_GATE is
port(A: in std_logic;
B: in std_logic;
F1: out std_logic
);
end AND_GATE;
architecture behv of AND_GATE is
begin
process(A,B)
begin
F1 <= A and B; -- behavior des.
library ieee ;
use ieee.std_logic_1164.all;
entity shift_reg is
port(I: in std_logic;
clock: in std_logic;
shift: in std_logic;
Q: out std_logic
);
end shift_reg;
architecture behv of shift_reg is
-- initialize the declared signal
signal S: std_logic_vector(2 downto 0):="111";
begin
process(I, clock, shift, S)
begin
-- everything happens upon the clock changing
if clock'event and clock='1' then
if shift = '1' then
S <= I & S(2 downto 1);
end if;
end if;
end process;
-- concurrent assignment
Q <= S(0);

Ο κώδικας για τα checkboxes:

```
<div class="checklist" style="width: 190px; display: table-cell;">
<?php
$path = 'C:/xampp/htdocs/myproject/uploads/'.$username.'/';

$dir_handle = @opendir($path) or die("Unable to open $path");
echo "<h2>Your files</h2>";
while ($file = readdir($dir_handle)) {
    if($file == "." || $file == ".." ) {
      continue;
    }
    // Prints a checkbox after each file.
    // The checkboxes are received by the form handler as an array called $files[]
    // holding the filenames of the files .
    $tmpname = mb_strimwidth($file, 0, 20, "...");
    echo "<label title='$file'><input type='checkbox' title='$file'
    class='files' id='chkbx' name='files[]' value='$file'/> $tmpname <label><br>";
}
?>
<br>
<input type="button" id="download" value="Download"/>
<input type="button" id="delete" value="Delete"/>
</div>
```

Ο κώδικας για τα textareas:

```
$i = 0;
$dir_handle = @opendir($path) or die("Unable to open $path");
while ($file = readdir($dir_handle)) {
    if($file == "." || $file == ".." ) {
      continue;
    }
    $i++;

    $file_content[$i] = implode(" ",showfile($path.$file));
    $classcontainer[$i] = 'cc_'.$i;
    $txtclass[$i] = 'sb_'.$i;
    $savebtn[$i] = 'sb_'.$i;
    //echo "$classname";

    echo "<div class='classcontainer[$i]' style='position: relative;'>";
    echo "<textarea rows='15' cols='50' style='width: 75%; height: 100%;
    box-sizing: border-box;' spellcheck='false' id='txtar' class='$txtclass[$i]'
    name='files'> $file_content[$i] </textarea>";
    echo "<button type='button' class='save_btn' name='btn' style='display:none;
    position: absolute; bottom: 5px; left: 0px;' id='$savebtn[$i]'
    onClick='replyclk(this.id)'>Save Changes</button>";
    echo "</div>";
}

function showfile($file){
    $line_num = 0;
    $file = fopen($file, "r") or exit("Unable to open file!");

    while(!feof($file)) {
        $line[$line_num] = fgets($file);
        $line_num ++;
    }
    return $line;
    fclose($file);
}
$j = 0;
foreach($txtclass as $tc){
    $j++;
    echo "<script> $('.$tc').on('keyup',function(){ $('#$savebtn[$j]').show();}); </script>";

}
```

```
$('textarea#txtar').focus(function () {
    $(this).animate({ height: "40em" }, 500);
});
```

4.3 ΛΕΙΤΟΥΡΓΙΕΣ

Όταν κανένα αρχείο δεν είναι επιλεγμένο καμιά λειτουργιά της εφαρμογής δεν είναι δυνατή. Όταν επιλεγεί ένα αρχείο οι λειτουργίες Download, Delete, Dead Code Removal, Code Restructuring, General Testbench Generator και Custom

Testbench Generator ενεργοποιούνται. Η Code Structure Visualization παραμένει ανενεργή. Όταν επιλεγούν πάνω από ένα αρχεία η λειτουργιά Code Structure Visualization ενεργοποιείται ενώ η Custom Testbench Generator απενεργοποιείται. Όλες οι υπόλοιπες παραμένουν ενεργοποιημενες και ανεπηρέαστες.

Ο κώδικας με τον οποίο επιτυγχάνεται η παραπάνω λειτουργία:

```
$('input:checkbox').click(function() {
    if ($(this).is(':checked')) {

        $('#download').prop("disabled", false);
        $('#delete').prop("disabled", false);
        $('#dead_code_removal').prop("disabled", false);
        $('#code_restructuring').prop("disabled", false);
        $('#g_testbench_generator').prop("disabled", false);
        $('#c_testbench_generator').prop("disabled", false);

        if ($('.files').filter(':checked').length > 1){
            $('#code_structure_visualization').prop("disabled", false);
            $('#c_testbench_generator').prop("disabled", true);
        }
        if ($('.files').filter(':checked').length == 1){
            $('#c_testbench_generator').prop("disabled", false);
        }
    } else {
        if ($('.files').filter(':checked').length < 1){
            $('#download').attr('disabled',true);
            $('#delete').attr('disabled',true);
            $('#code_structure_visualization').attr("disabled", true);
            $('#dead_code_removal').attr("disabled", true);
            $('#code_restructuring').attr("disabled", true);
            $('#g_testbench_generator').attr("disabled", true);
            $('#c_testbench_generator').attr("disabled", true);
        }
        if ($('.files').filter(':checked').length < 2){
            $('#code_structure_visualization').attr("disabled", true);
        }
        if ($('.files').filter(':checked').length == 1){
            $('#c_testbench_generator').attr("disabled", false);
        }
    }
});
```

4.3.1 Download & Delete

Αφού ένα ή παραπάνω αρχεία έχουν επιλεγεί, όταν ο χρήστης πατήσει το κουμπί Download, αυτά θα συμπιεστούν και θα κατέβουν σαν αρχείο με μορφή zip. Η εφαρμογή θα βγάλει ειδοποίηση: "Your file(s) has been downloaded!". Όταν ο χρήστης πατήσει το κουμπί Delete η εφαρμογή θα ρωτήσει: "You want to delete these file(s)?" για επιβεβαίωση, αν στη συνέχεια πατήσει OK η εφαρμογή θα διαγράψει οριστικά τα αρχεία από το directory του χρήστη και θα σβήσει την καταχώριση τους από τη βάση δεδομένων. Η σελίδα στη συνέχεια θα ανανεωθεί και το αρχείο δεν θα φαίνεται πια. Αν πατήσει Cancel η σελίδα απλά θα ανανεωθεί χωρίς τίποτα να διαγραφεί.

Ο κώδικας της Download:

```php
<?php

session_start();
if(!isset($_SESSION['username'])){header("Location:Login.php");}
else{ $username = $_SESSION['username'];}
$path = 'C:/xampp/htdocs/myproject/uploads/'.$username.'/';

$files_arr = array();
if(isset($_POST['files'])){
    foreach($_POST['files'] as $filename) {
        $filepath = $path.$filename;
        array_push($files_arr,$filepath);
    }
}

//print_r($files_arr);

header('Content-Type: application/zip');
header('Content-disposition: attachment; filename="'.$zipname.'"');
header('Content-Length: ' . filesize($zipname));

$zipname = time().".zip";
$zip = new ZipArchive;


if($zip->open($zipname, ZipArchive::CREATE)!== TRUE){
    echo " Sorry ZIP creation failed at this time";
}
foreach ($files_arr as $file) {
    $zip->addFile($file);
}
$zip->close();


readfile($zipname);
//unlink($zipname)
```

```
$('#download').prop("disabled", true);
$('#download').click(function() {
    $.ajax({
        url: "download.php",
        type: "post",
        data: $('.files:checked').serialize(),

        success: function(data) {
            //$('#feedback').replaceWith(data);
            alert("Your file(s) has been downloaded!");
        }
    });
});
```

Ο κώδικας της Delete:

```
<?php

session_start();
if(!isset($_SESSION['username'])){header("Location:Login.php");}
else{ $username = $_SESSION['username'];}
$path = 'C:/xampp/htdocs/myproject/uploads/'.$username.'/';

$files_arr = array();
if(isset($_POST['files'])){
    foreach($_POST['files'] as $filename) {
        $filepath = $path.$filename;
        array_push($files_arr,$filepath);
    }
}

foreach ($files_arr as $file_name){
    if(!unlink($file_name)){
        echo "Error while deleting your file";
    }else{
        $sql = "DELETE FROM files WHERE dir = '{$file_name}' AND username = '{$username}';";
        try{
            $conn = new PDO("mysql:host=localhost;dbname=vhdldb", "root", "");
            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $conn->exec($sql);
            $_SESSION['message'] = "You have deleted your files!";
            #header('Location: '.$_SERVER['PHP_SELF']);
            #header("location:main.php");
        }catch(PDOException $e){
            echo "Update database failed: ".$e->getMessage();
        }
    }
}
?>
```

```
$('#delete').prop("disabled", true);
$('#delete').click(function() {
    var c = confirm("You want to delete these file(s)?");
    if (c == true){
        $.ajax({
            url: "delete.php",
            type: "post",
            data: $('.files:checked').serialize(),

            success: function(data) {
                location.reload();
            }
        });
    }else{
        location.reload();
    }
});
```

*Στην παρουσίαση των παρακάτω λειτουργικοτήτων δεν παραθέτονται κώδικες λόγο του εκτενούς τους μεγέθους.

4.3.2 Dead Code Removal

Τα σήματα είναι τα κύρια αντικείμενα περιγραφής ενός συστήματος υλικού και είναι ισοδύναμα με τα "καλώδια" αυτού. Αντιπροσωπεύουν κανάλια επικοινωνίας μεταξύ των παράλληλων δηλώσεων των προδιαγραφών του συστήματος. Τα σήματα και οι συναφείς μηχανισμοί της VHDL (όπως οι εντολές εκχώρησης σήματος, η λειτουργία ανάλυσης, οι καθυστερήσεις κλπ.) χρησιμοποιούνται για να προσομοιωθούν εγγενείς λειτουργίες υλικού όπως η ταυτόχρονη κίνηση, ο αδρανειακός χαρακτήρας της μετάδοσης σήματος κλπ.
Τα σήματα στον κώδικα δηλώνονται μέσα στην αρχιτεκτονική. Πολλές φορες σήματα που οι χρήστες δηλώνουν μπορεί να μην χρησιμοποιούνται από την αρχιτεκτονική. Αυτά τα σήματα παραμένουν σαν νεκρός κώδικας χωρίς όμως να τον επηρεάζουν.

Η λειτουργία Dead Code Removal βρίσκει αυτά τα σήματα, αν υπάρχουν, και τα αφαιρεί από την σχεδίαση-κώδικα. Στη συνέχεια τυπώνει στην οθόνη τα ονόματα αυτών των σημάτων καθώς και την σειρά στην οποία τα βρήκε μέσα στον κώδικα. Ο νέος κώδικας αποθηκεύεται στα αρχεία με το υπάρχον όνομα προσθέτοντας την κατάληξη restructured και γίνεται διαθέσιμο για download.

```
library ieee;
use ieee.std_logic_1164.all;
use work.all;

entity comb_ckt is
port(   input1: in std_logic;
    input2: in std_logic;
    input3: in std_logic;
    output: out std_logic
);
end comb_ckt;

architecture struct of comb_ckt is

    component AND_GATE is
    port(   A:  in std_logic;
            B:  in std_logic;
            F1: out std_logic
    );
    end component;

    component OR_GATE is
    port(   X:  in std_logic;
            Y:  in std_logic;
            F2: out std_logic
    );
    end component;

    signal wire,wire1: std_logic;
    signal wire2: std_logic;

begin

    -- use sign "=>" to clarify the pin mapping

    Gate1: AND_GATE port map (A=>input1, B=>input2, F1=>wire);
    Gate2: OR_GATE port map (X=>wire, Y=>input3, F2=>output);

end struct;
```

Στον παραπάνω κώδικα ο χρήστης έχει δηλώσει τρία σήματα 'wire1','wire2' και 'wire' εκ των οποίων μόνο το ένα χρησιμοποιείται. Η εφαρμογή τα βρίσκει και τα εκτυπώνει στην οθόνη μαζί με τη σειρά στην οποία τα βρήκε όπως φαίνεται παρακάτω.

Unused signals found on file:C:/xampp/htdocs/myproject/uploads/penguin/new.vhd
Array ([1] => Array ([name] => wire1 [line_num] => 29) [2] => Array ([name] => wire2 [line_num] => 30))

Click here to download Reload Page

Μετα τη διαγραφή τους ο νέος κώδικας είναι ο εξής:

```
library ieee;
use ieee.std_logic_1164.all;
use work.all;

entity comb_ckt is
port(   input1: in std_logic;
    input2: in std_logic;
    input3: in std_logic;
    output: out std_logic
);
end comb_ckt;

architecture struct of comb_ckt is

    component AND_GATE is
    port(   A:  in std_logic;
            B:  in std_logic;
            F1: out std_logic
    );
    end component;

    component OR_GATE is
    port(   X:  in std_logic;
            Y:  in std_logic;
            F2: out std_logic
    );
    end component;

    signal wire: std_logic;

begin

    -- use sign "=>" to clarify the pin mapping

    Gate1: AND_GATE port map (A=>input1, B=>input2, F1=>wire);
    Gate2: OR_GATE port map (X=>wire, Y=>input3, F2=>output);

end struct;
```

4.3.3 Code Structure Visualization

Στις σχεδιάσεις- κώδικες δομικής περιγραφής Structural Description υπάρχει η ανάγκη οπτικοποίησης της δομής και παρουσίασής της με εύκολο και κατανοητό τρόπο. Όσο πολυπλοκότερη γίνεται μια δομική περιγραφή η ανάγκη

αναπαράστασής της με γραφικό τρόπο κρίνεται απαραίτητη. Η λειτουργία Code Structure Visualization κάνει ακριβώς αυτό. Ο χρήστης καλείται να επιλέξει τους κώδικες δομικής περιγραφής του λογικού του κυκλώματος και με το πάτημα του κουμπιού η εφαρμογή τους παρουσιάζει σαν δενδρική δομή με τον κώδικα του top level του κυκλώματος να αποτελεί πάτερα και τους κώδικες των κυκλωμάτων components να αποτελούν τα παιδιά της δενδρικής δομής.

Παράδειγμα κωδίκων δομικής περιγραφής.

Κώδικες Components (παιδιά):

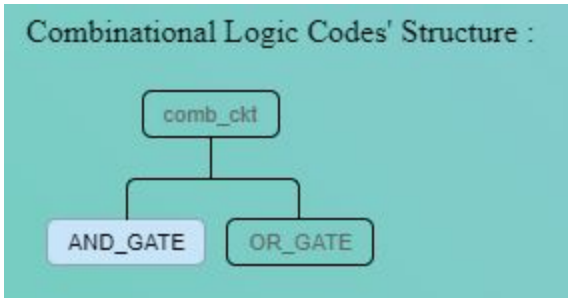
```
library ieee;                -- component #1
use ieee.std_logic_1164.all;

entity OR_GATE is
port(   X:  in std_logic;
    Y:  in std_logic;
    F2: out std_logic
);
end OR_GATE;

architecture behv of OR_GATE is
begin
process(X,Y)
begin
    F2 <= X or Y;            -- behavior des.
end process;
end behv;
```

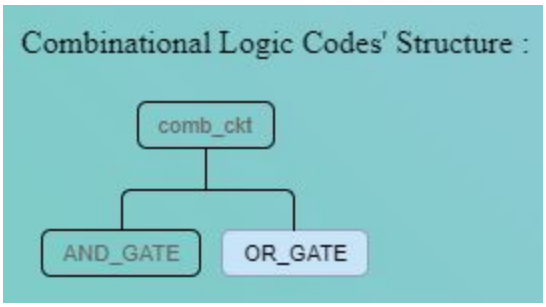
```
library ieee;                -- component #2
use ieee.std_logic_1164.all;

entity AND_GATE is
port(   A:  in std_logic;
    B:  in std_logic;
    F1: out std_logic
);
end AND_GATE;

architecture behv of AND_GATE is
begin
process(A,B)
begin
    F1 <= A and B;           -- behavior des
end process;
end behv;
```

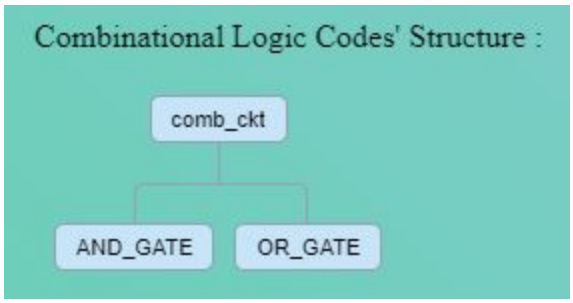
Κώδικας top level (πατέρας):

```
library ieee;               -- top level circuit
use ieee.std_logic_1164.all;
use work.all;

entity comb_ckt is
port(   input1: in std_logic;
    input2: in std_logic;
    input3: in std_logic;
    output: out std_logic
);
end comb_ckt;

architecture struct of comb_ckt is

    component AND_GATE is       -- as entity of AND_GATE
    port(   A:  in std_logic;
            B:  in std_logic;
            F1: out std_logic
    );
    end component;

    component OR_GATE is        -- as entity of OR_GATE
    port(   X:  in std_logic;
            Y:  in std_logic;
            F2: out std_logic
    );
    end component;

    signal wire: std_logic;     -- signal just like wire

begin

    -- use sign "=>" to clarify the pin mapping

    Gate1: AND_GATE port map (A=>input1, B=>input2, F1=>wire);
    Gate2: OR_GATE port map (X=>wire, Y=>input3, F2=>output);

end struct;
```

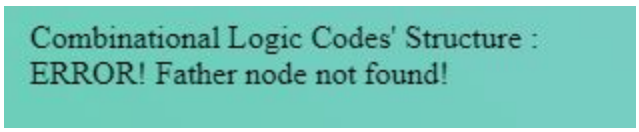
Αποτέλεσμα εφαρμογής (δενδρικη δομη):







Όταν ο κώδικας πατέρας η κάποιος από τους κώδικες παιδιά λείπει η εφαρμογή βγάζει μήνυμα λάθους.



*Ο css κώδικας που χρησιμοποιήθηκε για την δενδρική δομή είναι από:http://thecodeplayer.com/walkthrough/css3-family-tree

4.3.4 Code Restructuring

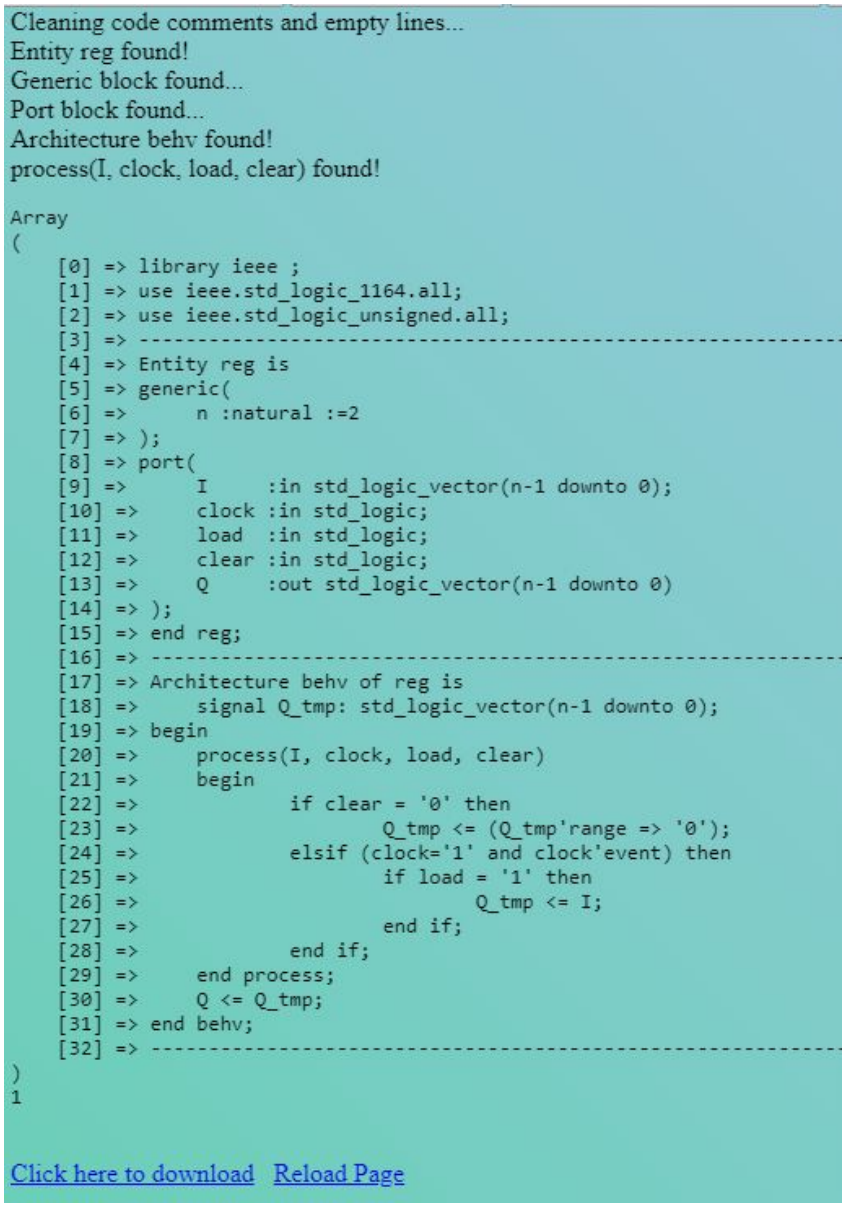
Όλοι οι κώδικες πρέπει να είναι καθαροί καλογραμμένοι και ευανάγνωστοι. Οι κακογραμμένοι κώδικες είναι πολλές φορές δύσκολο να διαβαστούν ακόμη και από τον ίδιο τον προγραμματιστή ο οποίος τους έγραψε και σχεδόν αδύνατο να διαβαστούν από τρίτους, πόσο μάλλον αν αυτοί δεν είναι προγραμματιστές η δεν

ξέρουν τη συγκεκριμένη γλώσσα. Ένας κώδικας δεν μπορεί να θεωρηθεί καλός μόνο αν είναι λειτουργικός και κάνει τη δουλειά του. Είναι σημαντικό να είναι δομημένος και καλογραμμένος για να θεωρηθεί καλός. Η λειτουργιά Code Restructuring αναδομεί και ωραιοποιεί τους κώδικες που επιλέγει ο χρήστης κάνοντάς τους πιο καθαρογραμμένους και ευανάγνωστους.

Παράδειγμα κώδικα που χρειάζεται αναδόμηση-ωραιοποίηση:

```
library ieee ;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;


entity reg is

generic(n: natural :=2);
port(   I:  in std_logic_vector(n-1 downto 0);
    clock:in std_logic;
load:   in std_logic;
     clear:in std_logic;
        Q:  out std_logic_vector(n-1 downto 0));
end reg;



architecture behv of reg is

signal Q_tmp: std_logic_vector(n-1 downto 0);
begin

    process(I, clock, load, clear)
    begin
if clear = '0' then
            -- use 'range in signal assigment
Q_tmp <= (Q_tmp'range => '0');
    elsif (clock='1' and clock'event) then
    if load = '1' then
    Q_tmp <= I;
       end if;
    end if;

    end process;

    -- concurrent statement
Q <= Q_tmp;

end behv;
```

Αποτέλεσμα εφαρμογής:



Cleaning code comments and empty lines...
Entity reg found!
Generic block found...
Port block found...
Architecture behv found!
process(I, clock, load, clear) found!

```
Array
(
    [0] => library ieee ;
    [1] => use ieee.std_logic_1164.all;
    [2] => use ieee.std_logic_unsigned.all;
    [3] => ----------------------------------------------------------------
    [4] => Entity reg is
    [5] => generic(
    [6] =>      n :natural :=2
    [7] => );
    [8] => port(
    [9] =>      I     :in std_logic_vector(n-1 downto 0);
    [10] =>     clock :in std_logic;
    [11] =>     load  :in std_logic;
    [12] =>     clear :in std_logic;
    [13] =>     Q     :out std_logic_vector(n-1 downto 0)
    [14] => );
    [15] => end reg;
    [16] => ----------------------------------------------------------------
    [17] => Architecture behv of reg is
    [18] =>     signal Q_tmp: std_logic_vector(n-1 downto 0);
    [19] => begin
    [20] =>     process(I, clock, load, clear)
    [21] =>     begin
    [22] =>             if clear = '0' then
    [23] =>                     Q_tmp <= (Q_tmp'range => '0');
    [24] =>             elsif (clock='1' and clock'event) then
    [25] =>                     if load = '1' then
    [26] =>                             Q_tmp <= I;
    [27] =>                     end if;
    [28] =>             end if;
    [29] =>     end process;
    [30] =>     Q <= Q_tmp;
    [31] => end behv;
    [32] => ----------------------------------------------------------------
)
1
```

Click here to download Reload Page

Η λειτουργιά κάνει τα εξής:

- Καθαρίζει σχόλια, άδειες γραμμές, περιττά κενά(spaces) και tabs.
- Βρίσκει το entity του κώδικα.
- Βρίσκει τα generic και port block του κώδικα και τα στοιχίζει όπως φαίνεται στο παραπάνω παράδειγμα

- Βρίσκει την αρχιτεκτονική του κώδικα και στοιχίζει τις δομές που Βρίσκονται μέσα σε αυτή
- Βρίσκει τα process της αρχιτεκτονικής στοιχίζει τα στοιχεία και τοποθετεί με σωστό τρόπο τις ένθετες γραμμές των If Statement, Case Statement,
- When Statement, Loop Statement και συνδυασμό αυτών
- Βάζει '-------' γραμμές- σχόλια ανάμεσα στα διαφορετικά κομμάτια του κώδικα

```
library ieee ;
 use ieee.std_logic_1164.all;
 use ieee.std_logic_unsigned.all;
 ------------------------------------------------
 Entity reg is
 generic(
    n :natural :=2
 );
 port(
    I     :in std_logic_vector(n-1 downto 0);
    clock :in std_logic;
    load  :in std_logic;
    clear :in std_logic;
    Q     :out std_logic_vector(n-1 downto 0)
 );
 end reg;
 ------------------------------------------------
 Architecture behv of reg is
    signal Q_tmp: std_logic_vector(n-1 downto 0);
 begin
    process(I, clock, load, clear)
    begin
        if clear = '0' then
            Q_tmp <= (Q_tmp'range => '0');
        elsif (clock='1' and clock'event) then
            if load = '1' then
                Q_tmp <= I;
            end if;
        end if;
    end process;
    Q <= Q_tmp;
 end behv;
 ------------------------------------------------
```

4.3.5 General Testbench Generator

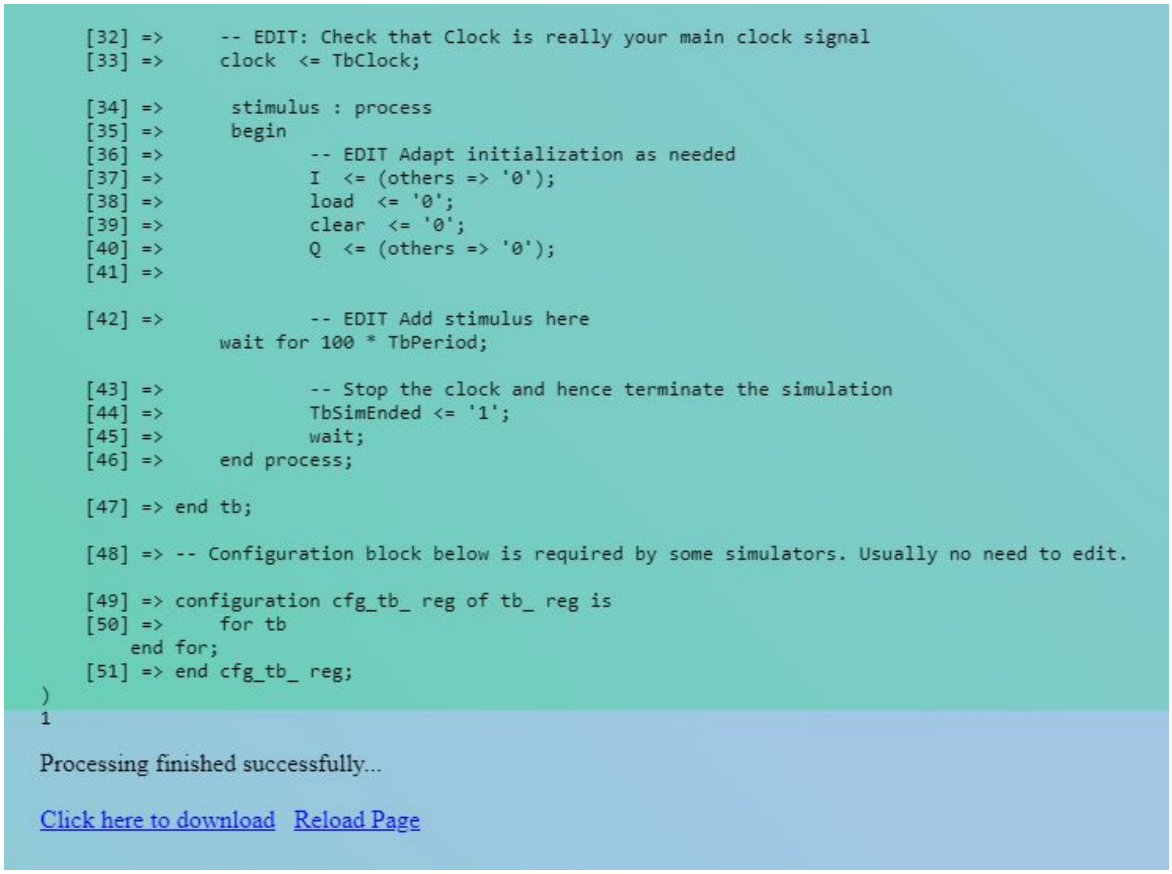
Ένα testbench είναι ένας κώδικας VHDL που παίζει το ρόλο ενός πλήρους περιβάλλοντος προσομοίωσης για το αναλυόμενο σύστημα DUT (design under

test) η σχέδιο υπό δοκιμή.Το testbench καθορίζει ερεθίσματα στις εισόδους του DUT συνήθως ως κυματομορφές δίνοντας σήματα inputs και επιτρέπει τον έλεγχο της λειτουργικότητας και των εξόδων του μέσα σε έναν προσομοιωτή.
Η συγγραφή του δηλωτικού μέρους του testbench είναι μια αυτοματοποιημένη διαδικασία, στάνταρ για τις περισσότερες σχεδιάσεις-κώδικες. Η λειτουργιά General Testbench Generator λύνει τα χέρια του προγραμματιστή κάνοντας τη συγγραφή του δηλωτικού μέρους αυτόματα για τους κώδικες που αυτός έχει επιλέξει. Στη συνέχεια τους τυπώνει στην οθόνη, τους αποθηκεύει προσθέτοντάς στο όνομα του αρχείου την κατάληξη g_testbench και τους κάνει διαθέσιμους για Download. Τέλος ο προγραμματιστής καλείται να προσθέσει τον κώδικα με τα ερεθίσματα (stimuly) στον testbench αλλά και να τροποποιήσει τον κώδικα με όποιον τρόπο επιθυμεί.
Θα χρησιμοποιήσουμε τον παρακάτω κώδικα σαν παράδειγμα για τη δημιουργία ενός testbench αρχείου:

```
library ieee ;
 use ieee.std_logic_1164.all;
 use ieee.std_logic_unsigned.all;
 ---------------------------------------------------
 Entity reg is
 generic(
    n :natural :=2
 );
 port(
    I     :in std_logic_vector(n-1 downto 0);
    clock :in std_logic;
    load  :in std_logic;
    clear :in std_logic;
    Q     :out std_logic_vector(n-1 downto 0)
 );
 end reg;
 ---------------------------------------------------
 Architecture behv of reg is
    signal Q_tmp: std_logic_vector(n-1 downto 0);
 begin
    process(I, clock, load, clear)
    begin
        if clear = '0' then
            Q_tmp <= (Q_tmp'range => '0');
        elsif (clock='1' and clock'event) then
            if load = '1' then
                Q_tmp <= I;
            end if;
        end if;
    end process;
    Q <= Q_tmp;
 end behv;
 ---------------------------------------------------
```

Αποτέλεσμα εφαρμογής:

```
Processing...
Cleaning code comments...
Entity block found...
Port block found...

Entity : reg
Clock :clock
Could not guess reset!

Array
(
    [0] => library ieee;
use ieee.std_logic_1164.all;

    [1] => entirty tb_ reg is
end tb_ reg;

    [2] => architecture tb of tb_ reg is

    [3] => component  reg
    [4] =>      port (
    [6] =>              I :in std_logic_vector(n-1 downto 0);
    [7] =>              clock :in std_logic;
    [8] =>              load :in std_logic;
    [9] =>              clear :in std_logic;
    [10] =>             Q :out std_logic_vector(n-1 downto 0)
    [11] =>     );
    [12] => end component;

    [13] =>     signal I :in std_logic_vector(n-1 downto 0);
    [14] =>     signal clock :in std_logic;
    [15] =>     signal load :in std_logic;
    [16] =>     signal clear :in std_logic;
    [17] =>     signal Q :out std_logic_vector(n-1 downto 0);
    [18] =>
        constant TbPeriod : time := 100 ns; -- EDIT Put right period here
    [19] =>     signal TbClock : std_logic := '0';
    [20] =>     signal TbSimEnded : std_logic := '0';

    [21] => begin

    [22] =>      dut :  reg
    [23] =>      port map (
    [25] =>             I =>I ,
    [26] =>             clock =>clock ,
    [27] =>             load =>load ,
    [28] =>             clear =>clear ,
    [29] =>             Q =>Q
    [30] =>     );

    [31] =>     -- Clock generation
        TbClock <= not TbClock after TbPeriod/2 when TbSimEnded /= '1' else '0';
```



```
    [32] =>     -- EDIT: Check that Clock is really your main clock signal
    [33] =>     clock  <= TbClock;

    [34] =>      stimulus : process
    [35] =>      begin
    [36] =>             -- EDIT Adapt initialization as needed
    [37] =>             I  <= (others => '0');
    [38] =>             load  <= '0';
    [39] =>             clear  <= '0';
    [40] =>             Q  <= (others => '0');
    [41] =>

    [42] =>             -- EDIT Add stimulus here
                wait for 100 * TbPeriod;

    [43] =>             -- Stop the clock and hence terminate the simulation
    [44] =>             TbSimEnded <= '1';
    [45] =>             wait;
    [46] =>     end process;

    [47] => end tb;

    [48] => -- Configuration block below is required by some simulators. Usually no need to edit.

    [49] => configuration cfg_tb_ reg of tb_ reg is
    [50] =>     for tb
        end for;
    [51] => end cfg_tb_ reg;
)
1
```

Processing finished successfully...

Click here to download Reload Page

Η λειτουργιά κάνει τα εξής:

- Καθαρίζει σχόλια, άδειες γραμμές, περιττά κενά(spaces) και tabs.
- Βρίσκει το entity του κώδικα.
- Βρίσκει το port block του κώδικα
- Αποθηκεύει το όνομα του entity
- Αποθηκεύει τα στοιχεία του port block και τους τύπους τους
- Ψάχνει για στοιχεία Clock και Reset και αν τα βρεί τα αποθηκεύει αν οχι ενημερώνει τον χρήστη
- Χρησιμοποιεί τα στοιχεία που σύλλεξε και με βάση ένα πρότυπο γράφει τον κώδικα Testbench
- Ενημερώνει αν η διαδικασία τελείωσε επιτυχώς

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entirty tb_ reg is
end tb_ reg;

architecture tb of tb_ reg is

component  reg
    port (
        I :in std_logic_vector(n-1 downto 0);
        clock :in std_logic;
        load :in std_logic;
        clear :in std_logic;
        Q :out std_logic_vector(n-1 downto 0)
    );
end component;

    signal I :in std_logic_vector(n-1 downto 0);
    signal clock :in std_logic;
    signal load :in std_logic;
    signal clear :in std_logic;
    signal Q :out std_logic_vector(n-1 downto 0);

    constant TbPeriod : time := 100 ns; -- EDIT Put right period here
    signal TbClock : std_logic := '0';
    signal TbSimEnded : std_logic := '0';

begin

     dut :  reg
     port map (
        I =>I ,
        clock =>clock ,
        load =>load ,
        clear =>clear ,
        Q =>Q
    );


    -- Clock generation
    TbClock <= not TbClock after TbPeriod/2 when TbSimEnded /= '1' else '0';

    -- EDIT: Check that Clock is really your main clock signal
    clock  <= TbClock;

     stimulus : process
     begin
        -- EDIT Adapt initialization as needed
        I  <= (others => '0');
        load  <= '0';
        clear  <= '0';
        Q  <= (others => '0');


        -- EDIT Add stimulus here
        wait for 100 * TbPeriod;

        -- Stop the clock and hence terminate the simulation
        TbSimEnded <= '1';
        wait;
    end process;

end tb;

-- Configuration block below is required by some simulators. Usually no need to edit

configuration cfg_tb_ reg of tb_ reg is
    for tb
    end for;
end cfg_tb_ reg;
```

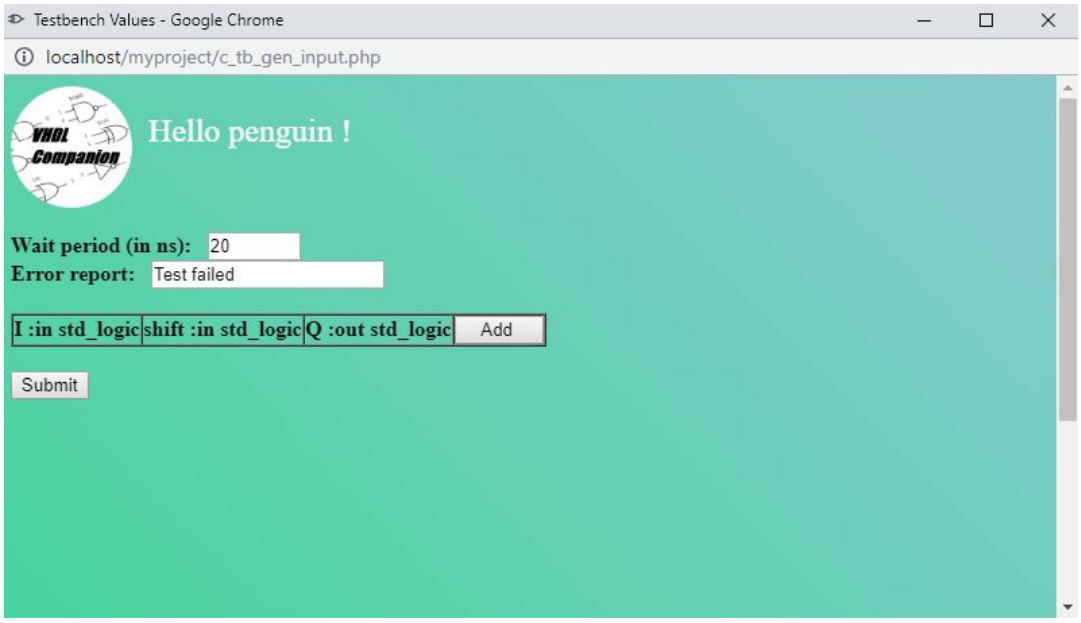
4.3.6 Custom Testbench Generator

Η λειτουργία Custom Testbench Generator είναι παρόμοια με αυτή του General Testbench Generator με τη διαφορά ότι σε αυτή ο χρήστης καλείται να εισάγει και τα ερεθίσματα (stimuly) του testbench. Δίνει τις τιμές inputs που θέλει να χρησιμοποιηθούν, τις τιμές outputs που αναμένει, τον χρόνο αναμονής (wait period) που θέλει στη σχεδίασή του και την αναφορά (report) που θέλει να εμφανιστεί σε περίπτωση που κάποιο τεστ αποτύχει. Η εφαρμογή παίρνει τις τιμές αυτές, δημιουργεί τα κατάλληλα asserts και τα τοποθετεί στον κώδικα, δημιουργώντας έτσι ένα ολοκληρωμένο testbench.

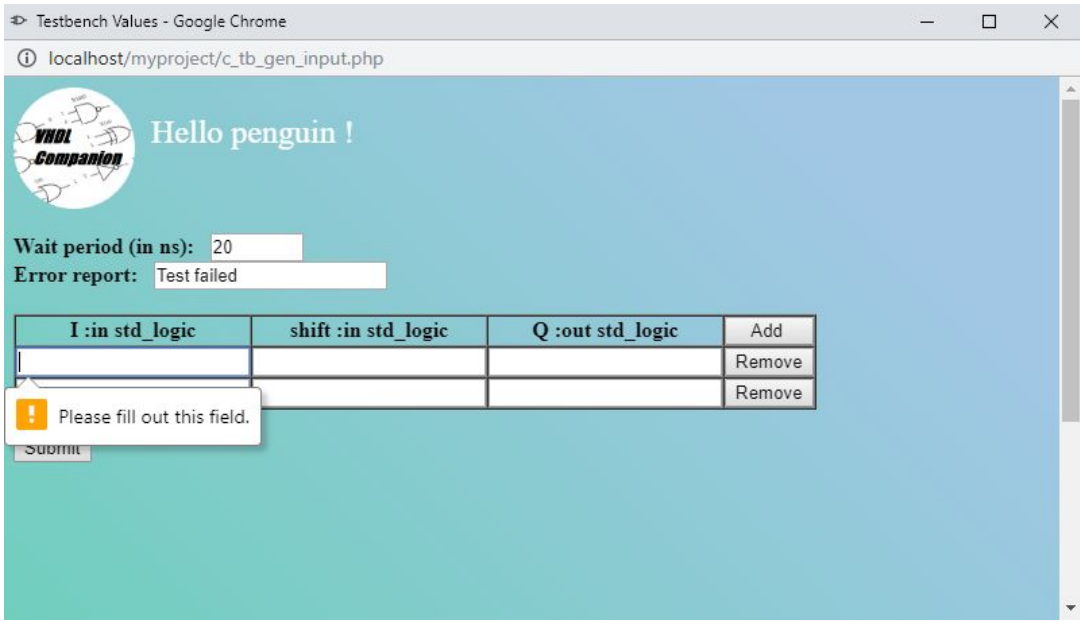
Θα χρησιμοποιήσουμε τον παρακάτω κώδικα σαν παράδειγμα για τη δημιουργία ενός testbench αρχείου:

```
library ieee;
use ieee.std_logic_1164.all;
----------------------------------------------------
entity shift_reg is
port(
        I:      in std_logic;
    clock:      in std_logic;
    shift:      in std_logic;
        Q:      out std_logic
);
end shift_reg;
----------------------------------------------------
architecture behv of shift_reg is

    -- initialize the declared signal
    signal S: std_logic_vector(2 downto 0):="111";

begin

    process(I, clock, shift, S)
    begin

    -- everything happens upon the clock changing
    if clock'event and clock='1' then
        if shift = '1' then
        S <= I & S(2 downto 1);
        end if;
    end if;

    end process;

    -- concurrent assignment
    Q <= S(0);

end behv;
```

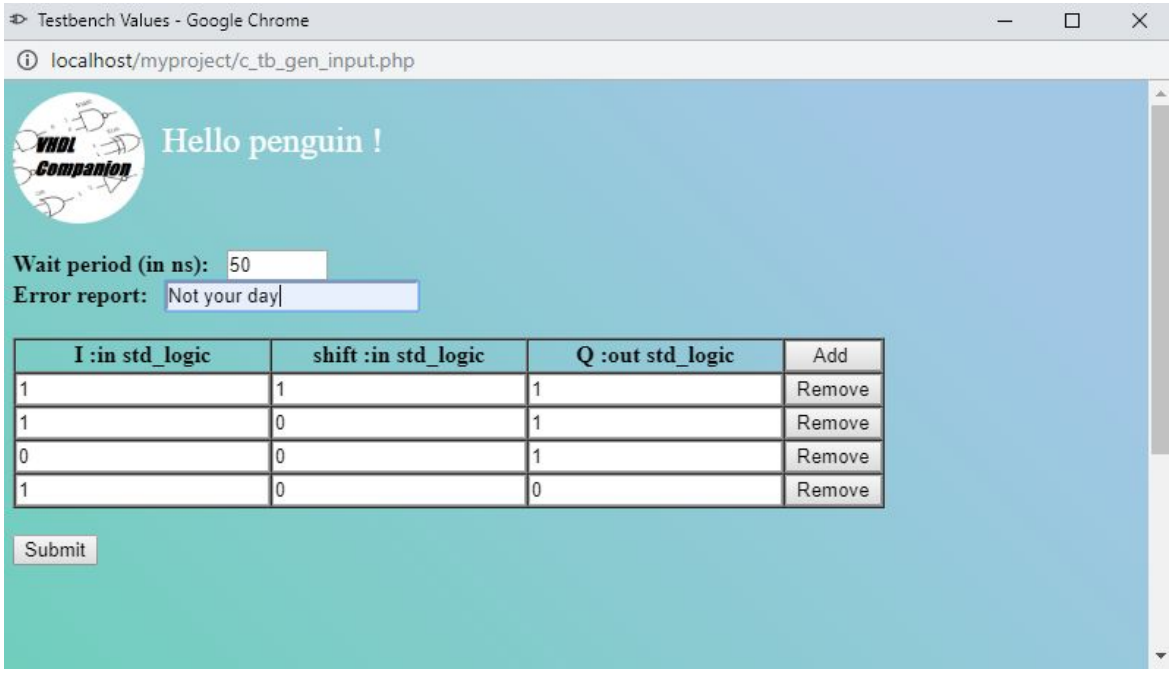
Όταν ο χρήστης επιλέξει το αρχείο του και πατήσει το κουμπί Custom Testbench Generator ένα pop up παράθυρο εμφανίζεται.



Όταν πατήσει το κουμπί Add ο πίνακας μεγαλώνει και νέα κελιά εμφανίζονται. Ο χρήστης καλείται να συμπληρώσει όλα τα κελιά που έχει εισάγει στον πίνακα προκειμένου να προχωρήσει. Όταν θέλει να διαγράψει κάποια σειρά, μπορεί να το κάνει πατώντας το κουμπί Remove.



Κάθε σειρά του πίνακα αποτελεί και ένα τεστ στο testbench.



Όταν ο πίνακας συμπληρωθεί (και προαιρετικά αλλάξουν τα υπόλοιπα πεδία) με το πάτημα του κουμπιού Submit το testbench αρχείο δημιουργείται, αποθηκεύεται και γίνεται διαθέσιμο για κατέβασμα.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entirty tb_ shift_reg is
end tb_ shift_reg;

architecture tb of tb_ shift_reg is

component  shift_reg
    port (
        I :in std_logic;
        clock :in std_logic;
        shift :in std_logic
    );
end component;

    signal I :in std_logic;
    signal ;
    signal shift :in std_logic;

    constant TbPeriod : time := 50 ns; -- EDIT Put right period here
    signal TbClock : std_logic := '0';
    signal TbSimEnded : std_logic := '0';

begin

     dut :  shift_reg
     port map (
        I =>I ,
        clock =>clock ,
        shift =>shift
    );

    -- Clock generation
    TbClock <= not TbClock after TbPeriod/2 when TbSimEnded /= '1' else '0';

    -- EDIT: Check that Clock is really your main clock signal
    clock  <= TbClock;

     stimulus : process
     begin
        -- EDIT Adapt initialization as needed
        I  <= '0';
        shift  <= '0';
```

Η λειτουργία Custom Testbench Generator κάνει τα ίδια πράγματα με την General Testbench Generator με τη διαφορά ότι συλλέγει στοιχεία και από τον κώδικα VHDL αλλά και από τη φόρμα που συμπληρώνει ο χρήστης, δημιουργώντας έτσι

έναν ολοκληρωμένο Testbench κώδικα και όχι απλώς ένα προσχέδιο κώδικα που χρειάζεται επιπλέον επεξεργασία.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η συγγραφή μιας τέτοιας εργασίας αναγκάζει τον φοιτητή να ανέβει level και να γίνει προγραμματιστής. Το κάθε εμπόδιο που συναντά κάποιος κατά τον προγραμματισμό μια διαδραστικής δικτυακής εφαρμογής αποτελεί ευκαιρία να μάθει κάτι καινούριο. Με κάθε νέα λειτουργικότητα που προσθέτει, καινούρια εμπόδια εμφανίζονται και η προσπάθεια που καταβάλει για να τα ξεπεράσει, μαθαίνει κάθε φορά στον προγραμματιστή και κάτι καινούριο. Αυτό συμβαίνει έως ότου φέρει την εργασία εις πέρας. Για παράδειγμα, πριν την ενασχόληση μου με την εργασία αυτή δεν είχα ακούσει καν για την τεχνολογία Ajax, αλλά προκειμένου να δημιουργήσω την διαδραστική σελίδα αναγκάστηκα να την δουλέψω, να την μάθω και τέλος να αντιληφθώ την αξία και την προσφορά της στον κόσμο της τεχνολογίας και του προγραμματισμού.Το ίδιο ισχύει και για τα υπόλοιπα εργαλεία, τεχνολογίες και γλώσσες προγραμματισμού που χρησιμοποίησα(αν και τα χα ακούσει). Επίσης κατάλαβα ότι το front end κομμάτι ανάπτυξης μιας εφαρμογής δεν μου ταιριάζει.Και το κατάλαβα κάνοντας μια εργασία που δεν απαιτούνταν να ασχοληθώ με αυτό. Τέλος η συγγραφή της εργασίας αυτής με βοήθησε να καταλάβω την αξία που έχουν ιστοσελίδες όπως το stackoverflow και το Quora για έναν προγραμματιστή αλλά και γενικότερα η αναζήτηση πληροφοριών στο ίντερνετ.

ΙΣΤΟΓΡΑΦΙΑ & ΒΙΒΛΙΟΓΡΑΦΙΑ

Για τη συγγραφή της γραπτής αναφοράς της πτυχιακής εργασίας δεν χρησιμοποιήθηκε βιβλιογραφία.

Παρακάτω παραθέτονται οι ιστοσελίδες από τις οποίες αντλήθηκαν πληροφορίες για την συγγραφή της εργασίας με τη σειρά που εμφανίζονται σε αυτή:

- https://el.wikipedia.org/wiki/Πληροφορική
- https://el.wikipedia.org/wiki/Διαδίκτυο
- https://el.wikipedia.org/wiki/Ιστορία_του_διαδικτύου
- https://en.wikipedia.org/wiki/World_Wide_Web
- https://en.wikipedia.org/wiki/Website
- https://en.wikipedia.org/wiki/Hardware_description_language
- https://en.wikipedia.org/wiki/VHDL
- https://www.nandland.com/vhdl/tutorials/tutorial-introduction-to-vhdl-for-beginners.html
- https://www.apachefriends.org/index.html
- https://en.wikipedia.org/wiki/XAMPP
- https://en.wikipedia.org/wiki/Apache_HTTP_Server
- https://httpd.apache.org/
- https://en.wikipedia.org/wiki/PhpMyAdmin
- https://www.phpmyadmin.net/
- https://en.wikipedia.org/wiki/HTML
- https://en.wikipedia.org/wiki/PHP

- https://en.wikipedia.org/wiki/JavaScript
- https://en.wikipedia.org/wiki/Ajax_(programming)
- https://en.wikipedia.org/wiki/MySQL
- https://www.mysql.com/

Άλλες χρήσιμες ιστοσελίδες:

- https://www.w3schools.com/
- https://stackoverflow.com/
- http://esd.cs.ucr.edu/labs/tutorial/