

Σχεδίαση Εφαρμογών & Υπηρεσιών Διαδικτύου 2η Εργασία

Μέλη ομάδας:

Καραπιπεράκης Εμμανουήλ A.M:2022201800075

Παπαβασιλόπουλος Ιωάννης A.M:2022201800141

Παπαχρήστου Ιωάννης A.M:2022201800146

Η εργασία είναι επίσης διαθέσιμη στο: <https://websiteformula1.000webhostapp.com>

Περιεχόμενα

SITEMAP.....	3
Αναλυτική περιγραφή.....	4
Λίστα με λειτουργικότητες που υλοποιήσαμε.....	23
Λίστα με λειτουργικότητες που δεν υλοποιήσαμε.....	24
Σημαντικότερες Δυσκολίες.....	25

SITEMAP

index.html

basics.html

more.html

quiz.html

sign-up.php

login.php

Ανεπιτυχής σύνδεση

Επιτυχής σύνδεση

connected/

index.php

basics.php

more.php

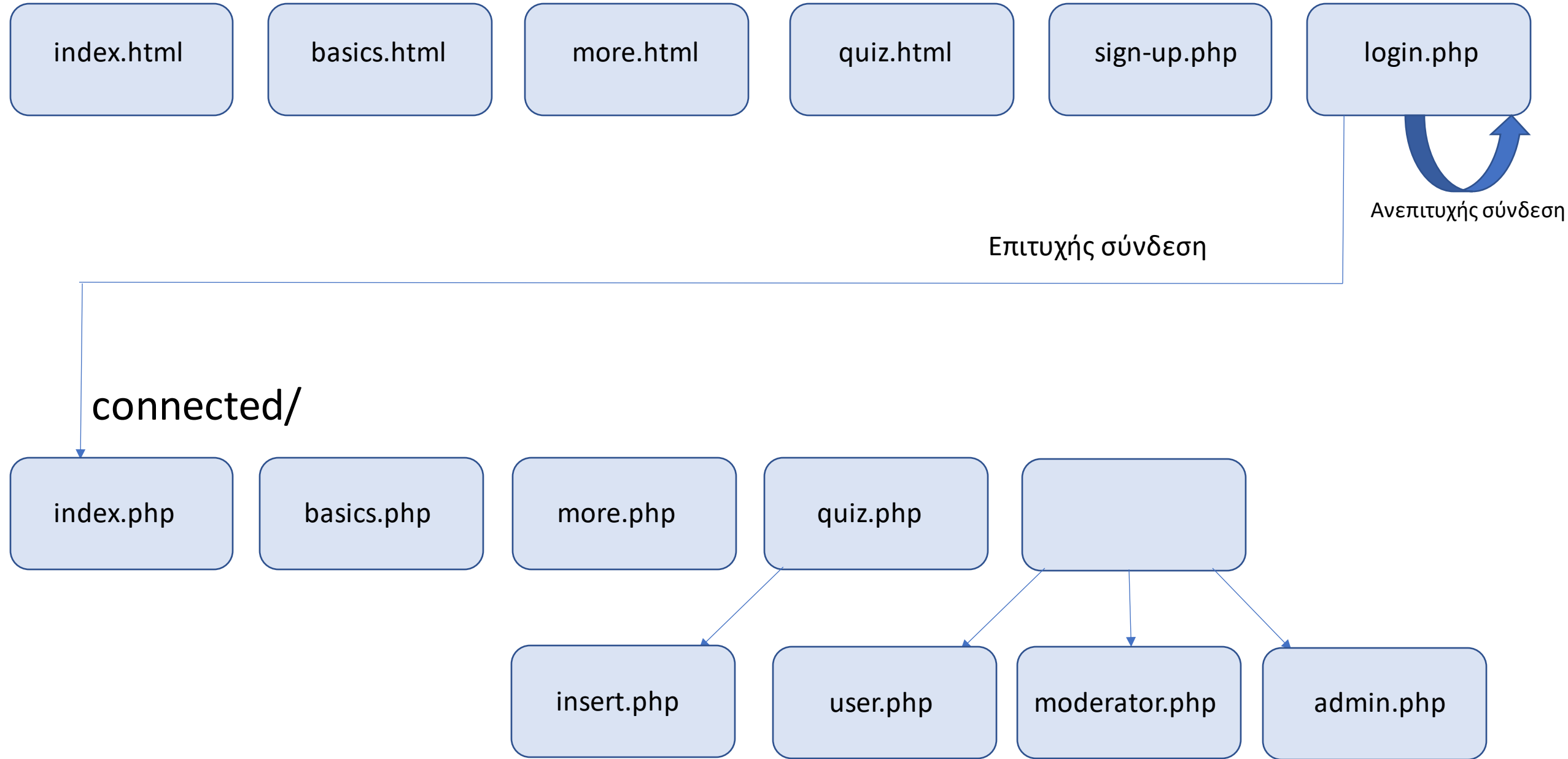
quiz.php

insert.php

user.php

moderator.php

admin.php



Αναλυτική περιγραφή

Το θέμα της συγκεκριμένης ιστοσελίδας είναι **Η Ιστορία της Formula 1**. Το σαιτ τρέχει σε 3 ταχύτητες(desktop,tablet,smartphone).Αρχικά αποτελείται από 6 σελίδες στις οποίες δεν απαιτείται εξουσιοδότηση για πρόσβαση στο περιεχόμενό τους.

Αναλυτικότερα η σελίδα index.html περιλαμβάνει το εισαγωγικό σημείωμα και μια εικόνα με το πρόγραμμα αγώνων της διοργάνωσης. Στην κορυφή βρίσκεται η μπάρα πλοήγησης για τη μετακίνηση του χρήστη στους υπόλοιπους χώρους του σαιτ η οποία είναι fixed και ακολουθεί τη ροή της σελίδας καθώς γίνεται scroll down. Για διαστάσεις κινητών το μενού προσαρμόζεται κατάλληλα και αντικαθίσταται από ένα εικονίδιο στο οποίο αν γίνει hover με το ποντίκι εμφανίζεται το μενού. Ακολουθώντας υπάρχει ένα λευκό header με το θέμα της ιστοσελίδας μας και στην περίπτωση που γίνει κλικ στο εικονίδιο της «Formula 1» έχει ως αποτέλεσμα ο χρήστης να μεταφερθεί ξανά στην αρχική σελίδα. Στο τέλος βρίσκεται το footer όπου υπάρχουν τα στοιχεία της ομάδας μας καθώς και ένας εναλλακτικός τρόπος για να επισκεφτεί ο χρήστης τα μέρη του σαιτ χωρίς να χρειαστεί να επιστρέψει στην αρχή της σελίδας.

Εμφάνιση σε Desktop/tablet:

ΑΡΧΙΚΗ

ΕΙΣΑΓΩΓΗ

ΠΕΡΙΣΣΟΤΕΡΑ

QUIZ

ΕΓΓΡΑΦΗ

ΕΙΣΟΔΟΣ

Η ιστορία της 
Formula 1.

Σκοπός της παρούσας ιστοσελίδας είναι η παρουσίαση της ιστορίας της κορωνίδας του μηχανοκίνητου αθλητισμού.

Χρησιμοποιείτε την παραπάνω μπάρα για να πλοηγηθείτε στον ιστότοπο.


**UPDATED
2021**

R1



26-28 MAR
BAHRAIN
SAKHIR

R2



16-18 APR
ITALY
IMOLA

R3



30 APR-02 MAY
PORTUGAL
PORTIMAO

R4



07-09 MAY
SPAIN
BARCELONA

R5



20-23 MAY
MONACO
MONACO

R6



04-06 JUN
AZERBAIJAN
BAKU

R7



18-20 JUN
FRANCE
LECASTELLET

R13



03-05 SEPT
NETHERLANDS
ZANDVOORT

R14



10-12 SEPT
ITALY
MONZA

R15



24-26 SEPT
RUSSIA
SOCHI

R16



01-03 OCT
SINGAPORE
SINGAPORE

R17



08-10 OCT
JAPAN
SUZUKA

R18



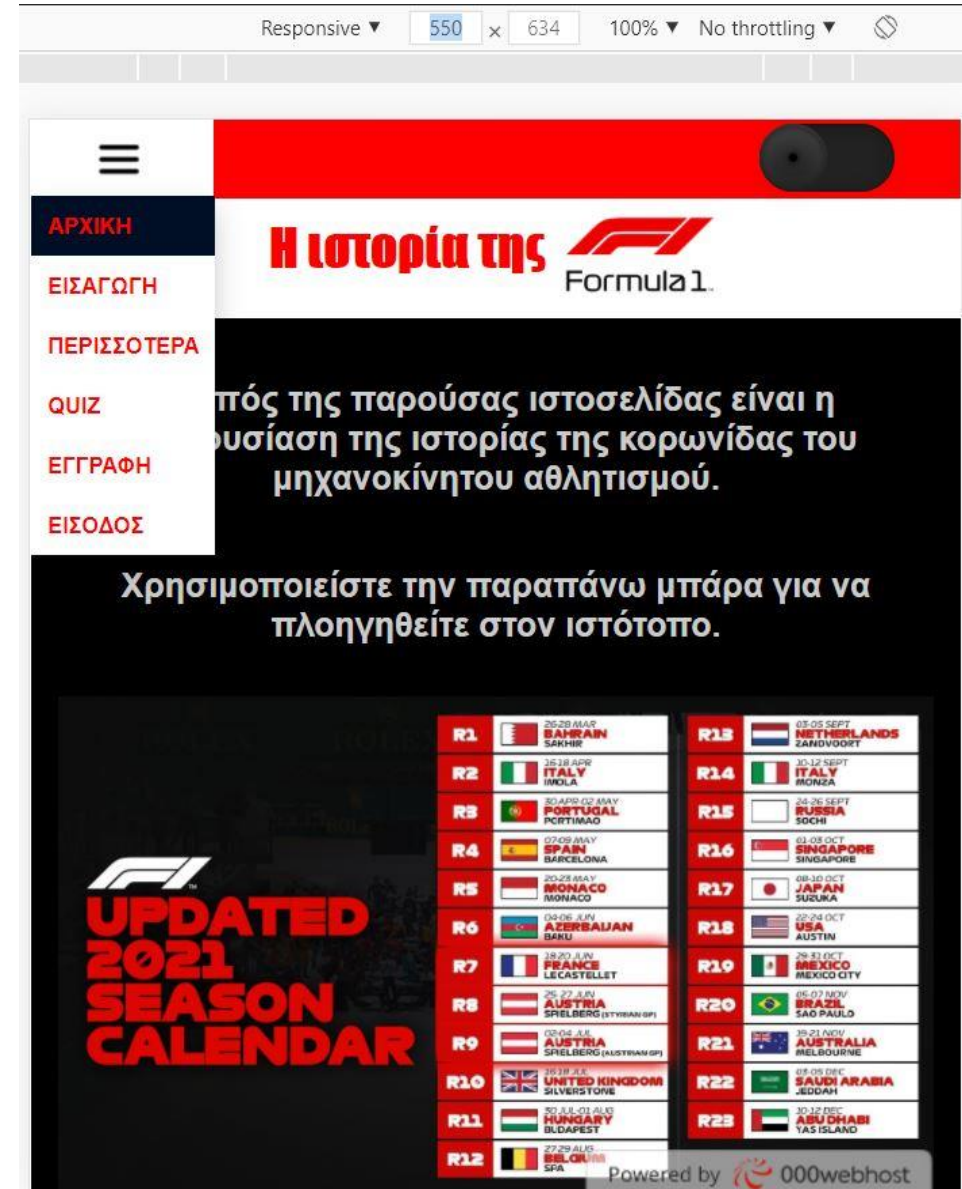
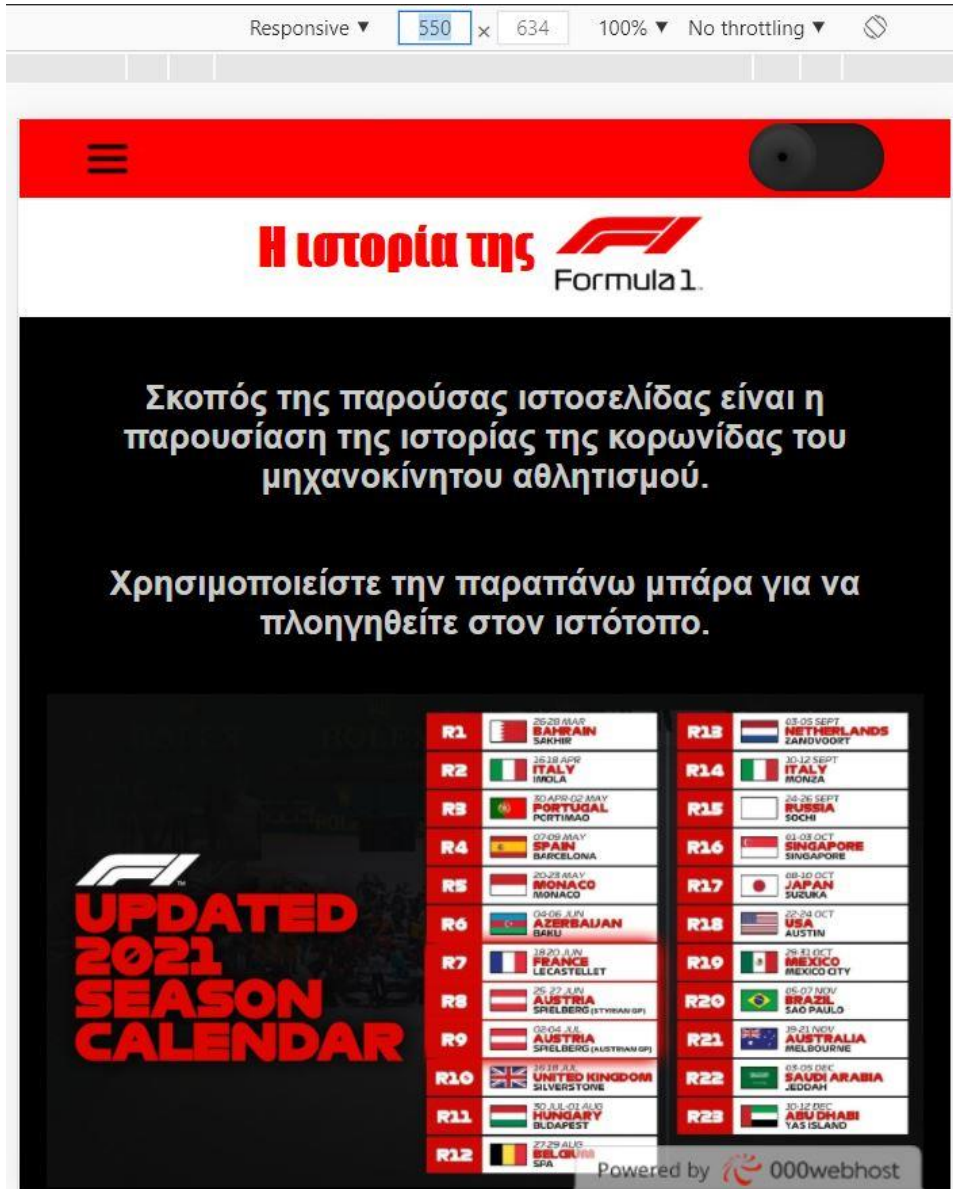
22-24 OCT
USA
AUSTIN

R19



29-31 OCT
MEXICO
MEXICO CITY

Εμφάνιση σε smartphone:



Επίσης προσθέσαμε ένα «dark-light mode» κουμπί το οποίο όταν πατηθεί μετατρέπει το background σε λευκό και το περιεχόμενο σε κόκκινο. Για να το πετύχουμε αυτό κατασκευάσαμε ένα checkbox και μέσω css βελτιώσαμε την εμφάνισή του. Το css το εμπνευστήκαμε από αυτό το βίντεο: <https://www.youtube.com/watch?v=p58K05YD98A&t=0s> και στη συνέχεια δημιουργήσαμε έναν event listener ο οποίος έπαιρνε κάθε φορά την τρέχουσα κατάσταση του checkbox(checked,unchecked) και άλλαζε το background σε λευκό και σε όσα tags είχαν name = "color" άλλαζε το χρώμα της γραμματοσειράς.

[ΑΡΧΙΚΗ](#)[ΕΙΣΑΓΩΓΗ](#)[ΠΕΡΙΣΣΟΤΕΡΑ](#)[QUIZ](#)[ΕΓΓΡΑΦΗ](#)[ΕΙΣΟΔΟΣ](#)

Η ιστορία της Formula 1.

Σκοπός της παρούσας ιστοσελίδας είναι η παρουσίαση της ιστορίας της κορωνίδας του μηχανοκίνητου αθλητισμού.

Χρησιμοποιείτε την παραπάνω μπάρα για να πλοηγηθείτε στον ιστότοπο.



UPDATED 2021

R1		26-28 MAR BAHRAIN SAKHIR
R2		16-18 APR ITALY IMOLA
R3		30 APR-02 MAY PORTUGAL PORTIMAO
R4		07-09 MAY SPAIN BARCELONA
R5		20-23 MAY MONACO MONACO
R6		04-06 JUN AZERBAIJAN BAKU
R7		18-20 JUN FRANCE CATALUNYA
R8		29-31 JUL UNITED KINGDOM SILVERSTONE
R9		29-31 AUG HUNGARY BUDAPEST
R10		29-31 OCT MEXICO CULIACAN
R11		22-24 OCT USA AUSTIN
R12		08-10 OCT JAPAN SUZUKA
R13		01-03 OCT SINGAPORE SINGAPORE
R14		24-26 SEPT RUSSIA SOCHI
R15		10-12 SEPT ITALY MONZA
R16		03-05 SEPT NETHERLANDS ZANDVOORT

Η ιστορία της



Σκοπός της παρούσας ιστοσελίδας είναι η παρουσίαση της ιστορίας της κορωνίδας του μηχανοκίνητου αθλητισμού.

Χρησιμοποιείτε την παραπάνω μπάρα για να πλοηγηθείτε στον ιστότοπο.



UPDATED 2021 SEASON CALENDAR

R1		26-28 MAR BAHRAIN SAKHIR
R2		16-18 APR ITALY IMOLA
R3		30 APR-02 MAY PORTUGAL PORTIMAO
R4		07-09 MAY SPAIN BARCELONA
R5		20-23 MAY MONACO MONACO
R6		04-06 JUN AZERBAIJAN BAKU
R7		18-20 JUN FRANCE CATALUNYA
R8		29-31 JUL UNITED KINGDOM SILVERSTONE
R9		29-31 AUG HUNGARY BUDAPEST
R10		29-31 OCT MEXICO CULIACAN
R11		22-24 OCT USA AUSTIN
R12		08-10 OCT JAPAN SUZUKA
R13		01-03 OCT SINGAPORE SINGAPORE
R14		24-26 SEPT RUSSIA SOCHI
R15		10-12 SEPT ITALY MONZA
R16		03-05 SEPT NETHERLANDS ZANDVOORT
R17		08-10 OCT JAPAN SUZUKA
R18		22-24 OCT USA AUSTIN
R19		29-31 OCT MEXICO CULIACAN
R20		05-07 NOV BRAZIL SAO PAULO
R21		19-21 NOV AUSTRALIA MELBOURNE
R22		05-07 DEC SAUDI ARABIA JEDDAH
R23		10-12 DEC ABU DHABI YAS Viceroy

Powered by  000webhost

Στις επόμενες 2 σελίδες(basics.html, more.html) οι λειτουργικότητες παραμένουν ίδιες με την αρχική σελίδα. Με διαφορά πως προσθέσαμε ένα css στις εικόνες σε περίπτωση που γίνει hover πάνω σε αυτές. Το css το βρήκαμε έτοιμο εδώ: <https://codepen.io/nxworld/pen/ZYNOBZ>

Στη συνέχεια έχουμε τη σελίδα quiz.html η οποία δημιουργήθηκε με βάση την 1η εργασία. Αποτελείται από διάφορες ερωτήσεις βασισμένες στις 2 προηγούμενες θεματικές ενότητες. Εάν ο χρήστης απαντήσει στις ερωτήσεις του quiz και πατήσει το κουμπί submit τότε δίπλα στις σωστές απαντήσεις του θα εμφανιστεί ένα πράσινο «τικ» που υποδυκνύει πως είναι σωστή, ή ένα κόκκινο «x» το οποίο δηλώνει πως είναι λάθος. Επίσης εάν αφήσει κενή μια επιλογή τύπου μονής επιλογής, θα εμφανιστεί ένα πράσινο «τικ» δίπλα στη σωστή απάντηση. Τέλος θα εμφανιστεί και ένα label με τα αποτελέσματα του χρήστη στο quiz

Η λογική του quiz έγινε μέσω javascript. Μέσα στη φόρμα βάλαμε id και name στα απαραίτητα πεδία και μέσα απο το javascript function **results()** το οποίο το καλούσαμε με το πάτημα του κουμπιού submit γινόντουσαν οι ελέγχοι αν είναι σωστή κάποια απάντηση και εάν ήταν αυξανόταν ένας counter ο οποίος δήλωνε πόσες σωστές απαντήσεις βρήκε ο χρήστης.

Τέλος υπάρχει ένα κουμπί try again όπου καθαρίζει τη φόρμα.

Η ιστορία της Formula 1.

Ποιο έτος ξεκίνησε επίσημα το πρωτάθλημα της Formula 1;

- ☒ 1950 ✓
- ☐ 1960
- ☐ 1970

Ποιος είναι ο πιο πετυχημένος βρετανός πιλότος στην ιστορία του αθλήματος;

- ☐ Nigel Mansell
- ☒ Lewis Hamilton ✓
- ☐ Jackie Stewart

Ο Sebastian Vettel κέρδισε όλα του τα πρωταθλήματα με την ομάδα της Red Bull Racing.

- ☒ Σωστό ✓
- ☐ Λάθος

Η Ferrari είναι η στατιστικά πιο πετυχημένη ομάδα.

- ☒ Σωστό ✓
- ☐ Λάθος

Ο Sebastian Vettel είναι Έλληνας.

- ☐ Σωστό
- ☒ Λάθος ✓

Ποιοι οδηγοί έχουν ίδιο αριθμό πρωταθλημάτων;

- ☐ Fernando Alonso
- ☒ Michael Schumacher ✓
- ☒ Lewis Hamilton ✓

Ο είναι ο πιο πετυχημένος Γερμανός πιλότος



Η βρετανική ομάδα της κυριάρχησε την δεκαετία του 90.



Ο Ayrton Senna κέρδισε πρωταθλήματα.



Submit

Try again

Results: 9/9

Η επόμενη σελίδα είναι η **sign-up.php** μέσω της οποίας ένας επισκέπτης έχει τη δυνατότητα να εγγραφεί αποκτώντας δικαιώματα "user" τα οποία μπορούν να αλλάξουν εάν το επιτρέψει κάποιος διαχειριστής.

Η λογική που χρησιμοποιήσαμε εδώ έχει ως εξής. Αρχικά όπως είναι φανερό χρειαζόμαστε να κατασκευάσουμε ένα table στη βάση δεδομένων μας για να αποθηκεύσουμε τα στοιχεία του χρήστη που επιθυμεί να εγγραφεί.

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	profpic	varchar(200)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	name	varchar(15)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	3	surname	varchar(25)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	4	email	varchar(28)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	5	password	varchar(20)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	6	dob	date			Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	7	gender	varchar(8)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	8	username	varchar(15)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	9	role	varchar(20)	utf8_unicode_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

Επειδή το username είναι μοναδικό, το ορίσαμε ως πρωτεύον κλειδί. Έτσι όλες οι μελλοντικές αλλαγές που θα γίνουν για κάποιον χρήστη θα γίνονται με βάση το username του. Πριν σταλθούν όμως τα δεδομένα από τη φόρμα στο php αρχείο όπου θα γίνει η εκτέλεση του query για να μπουν στη βάση τα περνάμε πρώτα από ένα javascript function έτσι ώστε να κάνουμε όσο το δυνατόν περισσότερους ελέγχους για την εγκυρότητα των δεδομένων που έδωσε ο χρήστης.

Με παρόμοια λογική με το quiz δώσαμε κατάλληλα name στα πεδία που θέλουμε να εξετάσουμε. Έτσι το πρώτο πράγμα που κάνουμε στο function μας είναι να περάσουμε το value των πεδίων σε μεταβλητές, κάπως έτσι δηλαδή:

name = document.forma.name.value;

Με αυτό τον τρόπο παίρνουμε από τη φόρμα με name = forma, από input με name = name το value του, το οποίο θα είναι το όνομα που μόλις έδωσε ο χρήστης. Εκτελούμε τα αντίστοιχα βήματα για όλα τα πεδία προς συμπλήρωση της φόρμας.

Επίσης έχουμε ένα flag το οποίο λαμβάνει integer τιμές και μας ενημερώνει πότε είναι όλα OK για να στείλουμε τα δεδομένα μας στο php αρχείο για να περαστούν στη βάση. Όταν δηλαδή λάβει τη τιμή flag == 7 όπου 7 είναι τα πεδία προς συμπλήρωση τότε θα σταλθούν τα δεδομένα.

Οι ενδεικτικοί ελέγχοι που βάλαμε είναι οι εξής:

- Τα text fields name,surname,username δεν μπορούν να είναι άδεια.(εξετάζαμε το μέγεθος του string αν είναι μηδέν)
- Η δομή του email πρέπει να είναι έγκυρη.(χρησιμοποιήσαμε έτοιμο regex που βρήκαμε online)
- Ο κωδικός πρέπει να περιέχει τουλάχιστον 8 χαρακτήρες.(εξετάζαμε αν το μέγεθος του string είναι >= 8)
- Οι κωδικοί πρέπει να ταιριάζουν.(το περιεχόμενο στα 2 password fields να είναι ίσο)
- Η ημερομηνία πρέπει να είναι έγκυρη.
- Το username ΔΕΝ πρέπει να χρησιμοποιείται ήδη από κάποιον άλλο χρήστη.(ασύγχρονος έλεγχος με χρήση AJAX)

Αν τηρούνται όλες αυτές οι προϋποθέσεις(flag ==7, κάθε φορά που κάποιος περιορισμός τηρείται το counter αυξάνεται κατά 1 αλλιώς μειώνεται) είμαστε έτοιμοι να στείλουμε τα δεδομένα μας.

Αλλιώς θα εμφανιστεί μήνυμα σφάλματος κάτω από το αντίστοιχο πεδίο, με χρήση της εντολής innerHTML απο το javascript function. Επίσης το board του αντίστοιχου input field θα γίνει κόκκινο και το background θα ενημερωθεί κατάλληλα με ένα εικονίδιο σφάλματος αλλιώς θα γίνει λευκό και θα ενημερωθεί με ένα εικονίδιο επιτυχίας.

Στην επιλογή φωτογραφίας γίνεται έλεγχος ώστε ο χρήστης να επιλέξει αρχείο εικόνας αλλά δεν είναι υποχρεωμένος να προσθέσει εικόνα προφίλ. Αν δεν επιλέξει κάποια εικόνα θα λάβει την by default φωτογραφία που φαίνεται στη φόρμα.

Επίσης στα input πεδία προσθέσαμε και maxlength που συμβαδίζει με το μέγεθος που έχουμε δώσει στη βάση έτσι ώστε ο χρήστης να μη βάλει κάτι που να μη χωράει στη βάση.

Sign-Up



Αλλαγή φωτογραφίας προφίλ:

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχ

Όνομα:

Επώνυμο:

Όνομα Χρήστη:

Email:

Κωδικός Πρόσβασης:

Επιβεβαίωση Κωδικού:

Ημερομηνία Γέννησης:

Ημέρα Μήνας Έτος

Φύλο:

- ☒ Άνδρας
☐ Γυναίκα

Ήδη Εγγεγραμμένος:

Δημιουργία Λογαριασμού



Αλλαγή φωτογραφίας προφίλ:

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχ

Όνομα:

Κώστας

Επώνυμο:

Παπαδόπουλος

Όνομα Χρήστη:

Empty Field

Empty Field

Email:

kostas@

Invalid email

Κωδικός Πρόσβασης:

Password must contain at least 8 characters

Επιβεβαίωση Κωδικού:

Ημερομηνία Γέννησης:

16 Αυγ 1985

Φύλο:

- ☒ Άνδρας
☐ Γυναίκα

Ήδη Εγγεγραμμένος:

Δημιουργία Λογαριασμού

Ο ιστότοπος websiteformula1.000webhostapp.com λείπει

Success

OK



Αλλαγή φωτογραφίας προφίλ:

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχ

Όνομα:

Κώστας

Επώνυμο:

Παπαδόπουλος

Όνομα Χρήστη:

kostas Pap

Email:

kostas@gmail.com

Κωδικός Πρόσβασης:

Επιβεβαίωση Κωδικού:

Ημερομηνία Γέννησης:

16 Αυγ 1985

Φύλο:

- ☒ Άνδρας
☐ Γυναίκα

Ήδη Εγγεγραμμένος:

Δημιουργία Λογαριασμού

Στην πρώτη εικόνα φαίνεται η φόρμα δημιουργίας λογαριασμού πριν πατηθεί το κουμπί Δημιουργία Λογαριασμού. Όπως φαίνεται στη 2η εικόνα δεν έχει συμπληρωθεί το username, έχει δοθεί μη έγκυρο email και ο κωδικός αποτελείται από 7 χαρακτήρες. Έτσι εμφανίζονται κατάλληλα μηνύματα και τα δεδομένα δεν φεύγουν ακόμα. Στην τρίτη εικόνα δίνονται έγκυρα στοιχεία και έχουμε προσθέσει ένα ενδεικτικό alert όπου μας ενημερώνει πως ο λογαριασμός δημιουργήθηκε με επιτυχία

Τώρα που είμαστε έτοιμοι να στείλουμε τα δεδομένα(flag == 7) δημιουργούμε μια καινούργια μεταβλητή τύπου formData(), κάνουμε append όλες τις πλέον έγκυρες μεταβλητές κάπως έτσι:

data.append("u",username);

Με αυτόν τον τρόπο "περνάμε την τιμή της μεταβλητής username στη μεταβλητή "u" την οποία στη συνέχεια θα λάβει το php αρχείο. Ακολούθως δημιουργούμε ένα καινούργιο XMLHttpRequest() δίνουμε την εντολή **"xhr.open('POST', "send.php");"** η οποία δηλώνει πως θα στείλουμε τα δεδομένα μας στο αρχείο "send.php" με τη μέθοδο "POST". Τέλος δίνουμε την εντολή **"xhr.send(data);"** και τα δεδομένα που έχουμε αποθηκεύσει στο data στέλνονται στο αρχείο send.php

Η λογική που ακολουθούμε στο αρχείο send.php είναι η εξής:

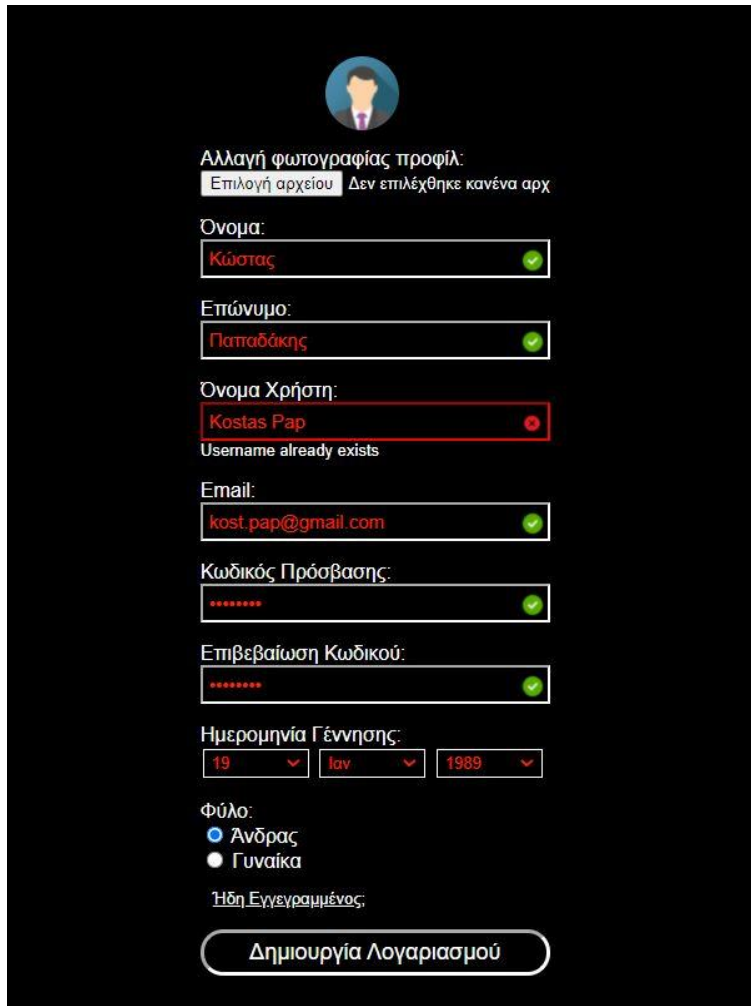
Πρώτο βήμα είναι η σύνδεση με τη βάση δεδομένων μας δίνοντας τα κατάλληλα \$servername,\$username,\$password,\$database Στη συνέχεια με τις ακόλουθες εντολές:

```
$n1 = $_POST["n"];  
$n2 = $_POST["s"];  
$n3 = $_POST["u"];  
$n4 = $_POST["e"];  
$n5 = $_POST["p"];  
$n6 = $_POST["d"];  
$n7 = $_POST["m"];  
$n8 = $_POST["y"];  
$n9 = $_POST["g"];
```

Αποθηκεύουμε στις μεταβλητές \$n1 - \$n9 τα δεδομένα που στείλαμε με μέθοδο POST από το προηγούμενο javascript function.

Και στη συνέχεια καλούμε ένα query INSERT με το οποίο βάζουμε τα δεδομένα στη βάση. Εδώ δεν υπήρχε κάποια ιδιαίτερη δυσκολία πέρα από τη διαχείριση των φωτογραφιών. Για τις εικόνες κατασκευάσαμε έναν φάκελο με όνομα upload όπου αποθηκεύεται το αρχείο της εικόνας και στη βάση δεδομένων αποθηκεύεται μόνο το path για την εικόνα. Επίσης όπως προαναφέραμε εάν δεν έχει δοθεί κάποια εικόνα τότε μέσα από ένα απλό if statement περνάμε στη βάση δεδομένων το μονοπάτι upload/default.png για μια εικόνα που έχουμε προσθέσει από μόνοι μας στον φάκελο upload.

Όλα αυτά λειτουργούν ομαλά σε περίπτωση που δίνεται πάντα διαφορετικό username. Όμως για να αποφύγουμε την περίπτωση να δοθεί υπάρχον username μέσα στο προηγούμενο javascript function καλούμε ασύγχρονα μέσω AJAX ένα αρχείο **"uesrname_check.php"** και εξετάζει τον αριθμό εμφανίσεων του username που δόθηκε από τον χρήστη μέσω της εντολής **"mysqli_num_rows(\$select)"** όπου \$select είναι ένα query το οποίο ψάχνει στη βάση δεδομένων των χρηστών να βρει username αντίστοιχα με αυτό που έδωσε. Εάν υπάρχει ήδη αποτρέπει την κλήση του send.php που προαναφέραμε και εκτυπώνει ανάλογο μήνυμα σφάλματος κάτω από το πεδίο username.



Αλλαγή φωτογραφίας προφίλ:
Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο

Όνομα:
Κώστας ✓

Επώνυμο:
Παπαδόκης ✓

Όνομα Χρήστη:
Kostas Pap ✗
Username already exists

Email:
kost.pap@gmail.com ✓

Κωδικός Πρόσβασης:
***** ✓

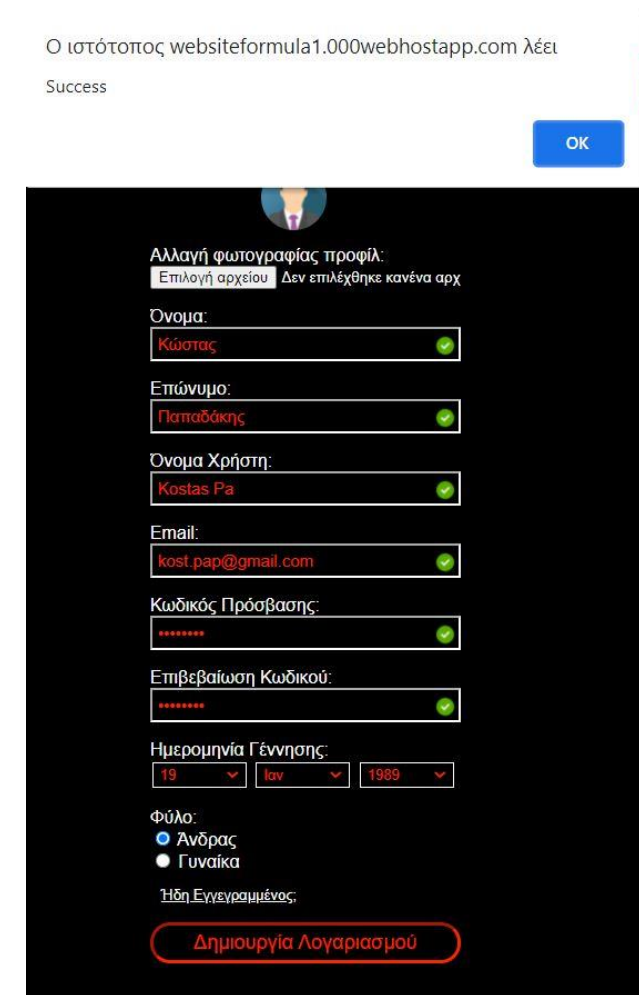
Επιβεβαίωση Κωδικού:
***** ✓

Ημερομηνία Γέννησης:
19 Jan 1989

Φύλο:
☒ Άνδρας
☐ Γυναίκα

Ήδη Εγγεγραμμένος:

Δημιουργία Λογαριασμού



Ο ιστότοπος websiteformula1.000webhostapp.com λέει
Success

OK

Αλλαγή φωτογραφίας προφίλ:
Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο

Όνομα:
Κώστας ✓

Επώνυμο:
Παπαδόκης ✓

Όνομα Χρήστη:
Kostas Pa ✓

Email:
kost.pap@gmail.com ✓

Κωδικός Πρόσβασης:
***** ✓

Επιβεβαίωση Κωδικού:
***** ✓

Ημερομηνία Γέννησης:
19 Jan 1989

Φύλο:
☒ Άνδρας
☐ Γυναίκα

Ήδη Εγγεγραμμένος:

Δημιουργία Λογαριασμού

Επόμενη και τελευταία σελίδα που είναι διαθέσιμη για έναν επισκέπτη είναι η φόρμα σύνδεσης όπου ο επισκέπτης καλείται να συμπληρώσει ένα username field και ένα password field σύμφωνα με τα στοιχεία που έδωσε κατά την εγγραφή του. Τα στοιχεία της φόρμας θα σταλθούν στο αρχείο **"login_check.php"** που βρίσκεται μέσα στον φάκελο **connected**. Στον φάκελο **connected** βρίσκονται αντίστοιχα αρχεία με τη διαφορά ότι απευθύνονται σε συνδεδεμένους χρήστες και η προβολή του περιεχομένου της κάθε σελίδας διαφοροποιείται ανάλογα με το ποιος είναι συνδεδεμένος (admin, user, moderator). Να σημειωθεί πως στην αρχή κάθε αρχείου που βρίσκεται μέσα στο **connected** γίνεται έλεγχος στο role του session έτσι ώστε να αποτραπεί η πρόσβαση σε μη εξουσιοδοτημένο χρήστη. Για παράδειγμα αν ο χρήστης πληκτρολογήσει το ακόλουθο URL <https://websiteformula1.000webhostapp.com/connected/index.php> ενώ δεν έχει συνδεθεί έτσι ώστε να του αποδοθεί το role του από τη βάση δεδομένων, τότε γίνεται redirect στο login.php.

Επομένως και στο login_check.php η λογική που ακολουθήσαμε είναι παρόμοια. Γίνεται ένα query όπου κάνει `SELECT *` από το table των χρηστών, αποδίδεται στο SESSION ο ρόλος του χρήστη που προσπαθεί να συνδεθεί καθώς και το username του και στη συνέχεια γίνεται direct στην τοποθεσία **connected/index.php**.

Αν ο χρήστης δώσει έγκυρα στοιχεία, θα αποδοθεί ρόλος και username στο session και η ανακατεύθυνση στο **connected/index.php** θα γίνει με επιτυχία.

Αν δώσει όμως μη έγκυρα στοιχεία, τότε το query θα αποτύχει, το session δε θα λάβει ρόλο και username του συνδεδεμένου και στην προσπάθεια να μεταβεί στη τοποθεσία **connected/login.php** θα γίνει ο πρώτος έλεγχος μετά το `start session()` και εφόσον δεν έχει δοθεί ρόλος θα γίνει redirect στην πλατφόρμα σύνδεσης.

Με την είσοδο του χρήστη πλέον οι σελίδες διαφοροποιούνται. Όπως φαίνεται πλέον δεν υπάρχει στο μενού επιλογή για εγγραφή και σύνδεση αλλά υπάρχει επιλογή ΡΥΘΜΙΣΕΙΣ και ΑΠΟΣΥΝΔΕΣΗ.

Μέσα στο φάκελο connected/ οι σελίδες index.php, basics.php, more.php έχουν το ίδιο περιεχόμενο με πριν. Σε αντίθεση με πριν όμως το quiz φαίνεται διαφορετικό.

Πιο συγκεκριμένα πλέον ο συνδεδεμένος χρήστης έχει τη δυνατότητα να πάρει μέρος σε 3 διαφορετικά quiz. Υπάρχει ένα combo box από το οποίο γίνεται η επιλογή της δυσκολίας και έχει ως εξής:

Default: Υπάρχουν 9 standar ερωτήσεις, ο χρήστης μπορεί να διενεργήσει το quiz, να πατήσει το κουμπί submit και λίγο πριν το κουμπί submit υπάρχει ένα label όπου εμφανίζεται με τη μορφή ποσοστού η βαθμολογία του στο quiz. Οι σωστές και οι λανθασμένες απαντήσεις δεν φαίνονται για να γίνει πιο ενδιαφέρον. Ο χρήστης είναι σε θέση να δει το ιστορικό των quiz του οποιαδήποτε χρονική στιγμή πηγαίνοντας στις ρυθμίσεις του λογαριασμού του, στις οποίες θα αναφερθούμε αναλυτικότερα στη συνέχεια. Για να επιτύχουμε την αποθήκευση των αποτελεσμάτων αρχικά κατασκευάσαμε ένα table στη βάση δεδομένων μας με όνομα **quiz_stats** και πεδία:

- **Time:** Η πλήρης ημερομηνία που υπέβαλε το quiz.(χρησιμοποιήσαμε μεθόδους date της javascript)
- **Results:** Τα αποτελέσματα σε ποσοστά του quiz. $((count/size)*100$ για να πάρουμε ποσοστο επιτυχίας)
- **Username:** Το username του χρήστη που διενέργησε το quiz, προκειμένου να γίνει η σύνδεση με τη βάση δεδομένων.
- **Difficulty:** Το επιλεγμένο επίπεδο δυσκολίας του quiz.

Για την προβολή του κουιζ με βάση το επίπεδο δυσκολίας κατασκευάσαμε 3 divs και κάθε φορά που επιλέγαμε ένα επίπεδο δυσκολίας κρύβαμε τα άλλα 2 div μέσα από ένα javascript function που έπερνε ως όρισμα την κατάσταση του select και προβάλαμε μόνο αυτό που ταίριαζε με το επιλεγμένο επίπεδο δυσκολίας.

Τέλος υπάρχει ένα κουμπί με το οποίο ο χρήστης μπορεί να προτείνει την προσθήκη κάποιας ερώτησης, η οποία θα γίνει αποδεκτή από κάποιον moderator ή admin στη συνέχεια και θα προσθεθεί στο αποθετήριο ερωτήσεων. Επομένως εάν ο χρήστης κάνει κλικ στο κουμπί "+" τότε θα ανοίξει το παρακάτω παράθυρο **insert.php**

Προσθήκη ερώτησης

Μονής επιλογής ▾

Εκφώνηση

Σωστή Απάντηση

Λάθος Απάντηση 1

Λάθος Απάντηση 2

Δυσκολία(easy,hard)

Submit

Προσθήκη ερώτησης

Μονής επιλογής ▾

Εκφώνηση

Σωστή Απάντηση

Λάθος Απάντηση 1

Λάθος Απάντηση 2

Δυσκολία(easy,hard)

Submit

Your question has been successfully submitted

Ο χρήστης επιλέγει τον τύπο της ερώτησης, συμπληρώνει τα πεδία σύμφωνα με τις οδηγίες που έχουν και πατάει το κουμπί submit. Στη συνέχεια θα καταλήξουν σε ένα javascript function, ένα μήνυμα επιτυχίας υποβολής της ερώτησης θα εμφανιστεί και το javascript function θα στείλει τα δεδομένα σε ένα αρχείο **temp_quiz.php** το οποίο θα τα βάλει μέσα σε ένα table στη βάση δεδομένων που δημιουργήσαμε με όνομα Temp_Questions και πεδία:

- **quest:** εκφώνηση
- **type:** τύπος ερώτησης(single, multiple etc)
- **correct_ans:** Σωστή απάντηση
- **ans1:** Δεύτερη απάντηση(κάποιες φορές σωστή, κάποιες λάθος ανάλογα τον τύπο της ερώτησης)
- **ans2:** Το ίδιο με την ans1
- **difficulty:** επίπεδο δυσκολίας.
- **id:** πρωτεύον κλειδί, μοναδικό για κάθε ερώτηση έτσι ώστε να μπορούμε να τις διαχειριστούμε.

Αυτές οι ερωτήσεις θα ληφθούν στη συνέχεια από τα ανώτερα στρώματα είτε θα αποριθφούν, είτε θα γίνουν τροποποιήσεις είτε θα γίνουν αποδεκτές στο αποθετήριο ερωτήσεων.

Να σημειωθεί πως για να ξέρουμε τι τύπος ερώτησης είναι έχουμε προσθέσει ένα κρυφό πεδίο αμέσως μετά τη δυσκολία όπου ο χρήστης δεν μπορεί να πειράξει, το οποίο έχει name = "<τύπο δυσκολίας>" και έτσι τον περνάμε στο javascript function μαζί με τα υπόλοιπα πεδία.

Επόμενο επίπεδο δυσκολίας είναι το easy.

Σε αυτή την περίπτωση, εάν ο χρήστης επιλέξει easy έχουμε ένα function για το checkbox το οποίο διαμορφώνει τη σελίδα με τις κατάλληλες ερωτήσεις ανάλογα τη δυσκολία. Η λογική που ακολουθήσαμε είναι η εξής:

Πάντα θα υπάρχουν 5 ερωτήσεις, 3 single(είτε ΣΛ, είτε με 3 επιλογές όπου όμως μόνο η μία θα είναι σωστή), 1 πολλαπλής επιλογής και 1 συμπλήρωσης κενού.

Επομένως θα κάνουμε μια επανάληψη 3 φορές και κάθε φορά θα κάνουμε SELECT μια τυχαία εκφώνηση και τις απαντήσεις της από το αποθετήριο ερωτήσεων τύπου single.

Για να λάβουμε μια τυχαία ερώτηση χρησιμοποιήσαμε το query: **`$sql = "SELECT * FROM question_repos WHERE difficulty = 'easy' AND type='single' ORDER BY RAND();"`**;

Με αυτό τον τρόπο σε κάθε επανάληψη καλούμε μια ερώτηση με τυχαία σειρά από το αποθετήριο ερωτήσεων. Επόμενο πράγμα που κάναμε ήταν να τυχαιοποιήσουμε όσο μπορούμε και τις υποψήφιες απαντήσεις. Εναλλακτικά κάθε φορά θα καλούσαμε μια τυχαία ερώτηση από τη βάση δεδομένων αλλά η σωστή απάντηση θα ήταν πάντα η πρώτη. Οπότε δημιουργήσαμε μια τυχαία μεταβλητή **`r=rand(1-9)`** για να πάρουμε ένα τυχαίο αριθμό από 1-9 και κάναμε την πράξη **`rmod3`** και ανάλογα το αποτέλεσμα άλλαζε η σειρά των απαντήσεων.

Επίσης για να δούμε εάν μια απάντηση τύπου single είναι ΣΛ εξετάζουμε εάν η 2η και η 3η απάντηση είναι η ίδια. Εάν δεν είναι πρόκειται για τύπου single με 3 υποψήφιες απαντήσεις, αλλιώς πρόκειται για ΣΛ και έτσι κρύβουμε την 3η απάντηση.

Ακολουθώντας κάνουμε ένα παρόμοιο 4ο select για να πάρουμε μια τυχαία ερώτηση τύπου multiple choice όπως φαίνεται στο ακόλουθο query: **`$sql = "SELECT * FROM question_repos WHERE difficulty = 'easy' AND type='multiple' ORDER BY RAND();"`**;

Τέλος πήραμε μια 5η και τελευταία ερώτηση τύπου label όπου πρόκειται για κενό προς συμπλήρωση δίνοντας το παρακάτω query:

`$sql = "SELECT * FROM question_repos WHERE difficulty = 'easy' AND type='label' ORDER BY RAND();"`;

Στη συνέχεια όπως και με το επίπεδο δυσκολίας default τα αποτελέσματα περνιούνται σε ένα javascript, γίνονται οι απαραίτητοι έλεγχοι, εμφανίζονται τα αποτελέσματα στην οθόνη και περνιούνται τα στατιστικά του συνδεδεμένου χρήστη στη βάση δεδομένων.

Το επίπεδο δυσκολίας hard λειτουργεί ακριβώς με τον ίδιο τρόπο απλά παίρνουμε ερωτήσεις δυσκολίας hard από το αποθετήριο.

Επόμενη σελίδα που υλοποιήσαμε είναι οι ρυθμίσεις όπως φαίνεται στο μενού πλοήγησης.

Η λογική είναι πως υπάρχουν 3 διαφορετικά αρχεία που ακούν στο όνομα ρυθμίσεις ανάλογα το ρόλο του συνδεδεμένου χρήστη. Έτσι σε όλα τα αρχεία του φάκελου connected/ πίσω από το κουμπί ΡΥΘΜΙΣΕΙΣ έχουμε βάλει ένα javascript function το οποίο ανάλογα με το ρόλο του τρέχοντος session ανοίγει κάποιο διαφορετικό αρχείο:

- **user.php** εάν ο ρόλος είναι user.
- **moderator.php** εάν ο ρόλος είναι moderator.
- **admin.php** εάν ο ρόλος είναι admin.

Στο user.php ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία του με τον εξής τρόπο:

Μπορεί να επιλέξει κάποια άλλη φωτογραφία και να κάνει κλικ στο εικονίδιο με το μολύβι για να γίνουν οι αλλαγές(απαιτείται ανανέωση της σελίδας)

Η λογική είναι η εξής, στο κλικ πάνω στο μολυβι, γίνεται κλήση ενός javascript function, στέλνουμε τα δεδομένα σε ένα αρχείο

../photo_update.php το οποίο ενημερώνει κατάλληλα το φάκελο **upload** όπου βρίσκονται οι ανεβασμένες εικόνες και στη συνέχεια ενημερώνει κατάλληλα και το path της εικόνας στη βάση δεδομένων. Εάν υπάρχει ήδη επιλεγμένη εικόνα και γίνει κλικ στο μολύβι, τότε η φωτογραφία default.png που έχουμε προαναφέρει θα πάρει τη θέση της.

Παρόμοια λογική υπάρχει και στα υπόλοιπα πεδία προς επεξεργασία, με κλικ πάνω στο μολύβι καλείται η αντίστοιχη συνάρτηση javascript και στη συνέχεια το κατάλληλο php αρχείο ανάμεσα στα:

- name_update.php
- surname_update.php
- email_update.php
- password_update.php
- dob_update.php
- gender_update.php

Επίσης λάβαμε ορισμένες περιπτώσεις, συγκεκριμένα τις ίδες με το sign-up όσον αφορά την αλλαγή των στοιχείων, έτσι ώστε να εμφανίζεται μήνυμα σφάλματος σε λάθος είσοδο.

Τέλος για τον απλό χρήστη εμφανίζονται και τα αποτελέσματα στα quiz που έχει διενεργήσει στο παρελθόν.

Εάν είναι συνδεδεμένος κάποιος moderator, οι ρυθμίσεις του περιέχουν τα ίδια πράγματα με του απλού χρήστη συν τις ερωτήσεις που έχουν σταλθεί για επεξεργασία,απόριψη ή αποδοχή. Επίσης υπάρχει και το αποθετήριο των ερωτήσεων.

Στις ερωτήσεις η λογική που ακολουθήσαμε είναι η εξής:

Κάνουμε επαναληπτικά ένα SELECT από την αντίστοιχη βάση και εμφανίζουμε σε ένα πίνακα την εκφώνηση, τον τύπο της ερώτησης, τη σωστή απάντηση, τη 2η και την 3η απάντηση και το επίπεδο δυσκολίας. Όλα αυτά είναι τύπου **input type = text**, επομένως ο moderator ή ο admin μπορούν να επεξεργαστούν με αυτό τον τρόπο όλα τα πεδία απλά κάνοντας κλικ μέσα. Ακολουθώς σε κάθε ερώτηση υπάρχουν 3 κουμπιά.

- **Decline:** Απόριψη της συγκεκριμένης ερώτησης.
- **Change:** Επιβεβαίωση των αλλαγών ώστε να μην χαθούν μετά από ένα refresh.
- **Accept:** Αποδοχή της ερώτησης και εισαγωγή στο αποθετήριο ερωτήσεων.

Κάθε κουμπί αντιστοιχεί σε μια javascript συνάρτηση, από την οποία καλείται το αντίστοιχο php αρχείο για ενημέρωση των βάσεων μας.

Στο αποθετήριο ερωτήσεων υπάρχουν οι ερωτήσεις όπου εμφανίζονται στα quiz. Η εμφάνισή τους γίνεται επαναληπτικά σε ένα table κάνοντας select τα πεδία του table **question_repos** . Επίσης υπάρχει ένα κουμπί delete με το οποίο καλούμε ένα javascript function και από εκεί το αντίστοιχο php αρχείο όπου εκτελείται το query DELETE

Τέλος στο admin.php εμφανίζονται όλα τα προηγούμενα που προαναφέραμε μαζί με ένα admin panel από το οποίο οι διαχειριστές μπορούν να βλέπουν τους εγγεγραμένους χρήστες, να διαγράφουν κάποιον άλλο χρήστη και να αλλάζουν το ρόλο κάποιου ανάμεσα σε user, moderator, admin. Επίσης προσθέσαμε έναν super-admin που βρίσκεται πάνω από όλους και κανείς δεν μπορεί να τον διαγράψει ή να του αλλάξει ρόλο με σκοπό την αποφυγή ανεπιθύμητων καταστάσεων. Για την αλλαγή ρόλου υπάρχει ένα combo box όπου μέσω ενός function ενημερώνεται το αντίστοιχο **change.php** αρχείο το οποίο με τη σειρά του εκτελεί το query UPDATE στο table με τους χρήστες σύμφωνα με το username που βρίσκεται στην ίδια γραμμή με το κουμπί καθώς και το role που επιλέχθηκε στο combo box. Για το delete αντίστοιχα καλείται συνάρτηση, η οποία με τη σειρά της καλεί το **delete.php** όπου εκτελεί το DELETE query για να σβήσουμε το χρήστη από τη βάση με τους άλλους χρήστες σύμφωνα με το μοναδικό username του. Επίσης εκτελούμε επιπλέον ένα query για να σβήσουμε τα στατιστικά του χρήστη από τη βάση του quiz, καθώς αν δεν τα σβήναμε και κάποιος άλλος χρήστης επέλεγε ως username κάποιο που χρησιμοποιήθηκε στο παρελθόν, θα μπορούσε να δει τα αποτελέσματα στο quiz του προηγούμενου κατόχου

Τέλος αν ο χρήστης πατήσει το κουμπί "ΑΠΟΣΥΝΔΕΣΗ" θα γίνει direct στη φόρμα σύνδεσης στο **../login.php** και το session θα τερματιστεί

Ενδεικτικά έχουμε προσθέσει μερικούς χρήστες όπως φαίνεται παρακάτω και μπορεί να πραγματοποιηθεί σύνδεση δίνοντας τα στοιχεία τους.

ADMIN PANEL										
profil_pic	username	name	surname	email	password	dob	gender	role	Change_role	Delete
	dit18075	Εμμανουήλ	Καραπιπεράκης	dit18075@go.uop.gr	admin	1970-01-01	Ανδρας	admin	<div>Change</div>	<div>delete</div>
	dit18141	Ιωάννης	Παπαβασιλόπουλος	dit18141@go.uop.gr	admin	1970-01-01	Ανδρας	admin	<div>Change</div>	<div>delete</div>
	dit18146	Ιωάννης	Παπαχρήστου	dit18146@go.uop.gr	admin	1970-01-01	Ανδρας	admin	<div>Change</div>	<div>delete</div>
	Julia	Ιουλία	Αλεξανδράτου	Julia@gmail.com	12345678	1980-01-01	Γυναίκα	moderator	<div>Change</div>	<div>delete</div>
	ntsel	Νικόλαος	Τσελίκας	ntsel@xxx.gr	admin	1980-01-01	Ανδρας	super-admin	<div>Change</div>	<div>delete</div>

Λίστα με λειτουργικότητες που υλοποιήσαμε

- Διαχωρισμός χρήστη με βάση το ρόλο του σε 4 κατηγορίες:
 - 1)Επισκέπτης-Μη εγγεγραμμένος
 - 2)απλός εγγεγραμμένος χρήστης
 - 3)moderator
 - 4)admin
- Λειτουργία σαιτ σε 3 ταχύτητες(desktop,tablet,smartphone)
- Πρόσβαση μη εγγεγραμμένου χρήστη στο περιεχόμενο της σελίδας
- Διενέργεια quiz ως επισκέπτης με προβολή αποτελεσμάτων
- Δημιουργία νέου Λογαριασμού μαζί με απαραίτητους ελέγχους
- Σύνδεση σε υπάρχον λογαριασμό
- Διενέργεια quiz ως συνδεδεμένος χρήστης με δυνατότητα επιλογής δυσκολίας
- Προσθήκη ερώτησης(όχι στο αποθετήριο)
- Redirect εάν κάποιος μη εξουσιοδοτημένος χρήστης προσπαθήσει να αποκτήσει πρόσβαση κάπου που δεν επιτρέπεται.
- Προβολή αποτελεσμάτων προηγούμενων τεστ
- Επεξεργασία προφίλ
- Διαχείριση υπάρχοντων προφίλ από τους διαχειριστές(αλλαγή ρόλου, διαγραφή)
- Διαχείριση ερωτήσεων & προσθήκη αυτών στο αποθετήριο
- Διαγραφή ερωτήσεων από το αποθετήριο
- Τερματισμός session κατά την Αποσύνδεση

Λίστα με λειτουργικότητες που δεν υλοποιήσαμε

- Σε περίπτωση εσφαλμένων στοιχείων σύνδεσης στο login.php θέλαμε μέσω AJAX να ενημερώνουμε το χρήστη με μήνυμα σφάλματος αντί να κάνουμε redirect.
- Ίσως χρειαζόταν να βάλουμε περισσότερους ελέγχους στη φόρμα Δημιουργίας Λογαριασμού για να καλύψουμε περισσότερες περιπτώσεις σφάλματος
- Επειδή αρχικά βάζαμε τα javascript functions μέσα στο html αρχείο, στο τέλος που θελήσαμε να τα περάσουμε σε ξεχωριστά αρχεία δημιουργήθηκαν προβλήματα. Επομένως τα αφήσαμε μέσα στο HTML.
- Αν κάποιος χρήστης επιλέξει ως εικόνα ένα αρχείο με όνομα που ήδη υπάρχει στο φάκελο των εικόνων , τότε η προηγούμενη εικόνα θα αντικατασταθεί με τη νέα, δημιουργώντας σύγχυση στο χρήστη που θα δει την εικόνα του να αλλάζει χωρίς να έχει πραγματοποιήσει κάποια ενέργεια.

Σημαντικότερες Δυσκολίες

Αρχικά δυσκολευτήκαμε στο πέρασμα της φωτογραφίας του χρήστη στη βάση δεδομένων καθώς προσπαθούσαμε να την περάσουμε ως τύπο BLOB. Εν τέλει το πρόβλημα λύθηκε επιλέγοντας μια μέθοδο που αναφέρθηκε σε μια διάλεξη σύμφωνα με την οποία θα φτιάχναμε ένα folder στο server όπου θα ανέβαινε το αρχείο της εικόνας και στη βάση απλά θα περνούσαμε το path. Έτσι γλυτώσαμε αρκετό χώρο που θα δεσμεύαμε στη βάση εάν δηλώναμε την εικόνα ως τύπο LONGBLOB.

Στη φόρμα σύνδεσης μας πήρε αρκετό χρόνο μέχρι να υλοποιήσουμε όλους τους ελέγχους(ασύγχρονους και μη) πριν σταλθούν τα δεδομένα στη βάση αλλά τελικά τα καταφέραμε.

Επίσης αρχικά δυσκολευτήκαμε στην κατανόηση των sessions και τη διαμόρφωση του περιεχομένου ανάλογα το ρόλο του συνδεδεμένου χρήστη. Εν τέλει καταφέραμε να τερματίζουμε το session κατά την αποσύνδεση και να διακρίνουμε τους χρήστες σύμφωνα με τους ρόλους τους.

Δυσκολευτήκαμε στο quiz όσον αφορά τις ερωτήσεις που θα έστελναν οι χρήστες και το αποθετήριο καθώς αρχικά σκεφτόμασταν να υλοποιήσουμε και τις 2 λειτουργικότητες σε έναν πίνακα στη βάση. Εν τέλει δημιουργήσαμε 2 διαφορετικούς πίνακες και υλοποιήσαμε και τις 2 λειτουργικότητες.

Τέλος δυσκολευτήκαμε στο κομμάτι τυχαιοποίησης των ερωτήσεων στο κουίζ ανάλογα με την επιλεγμένη δυσκολία. Εν τέλει βρήκαμε πως μπορούμε να πραγματοποιήσουμε "τυχαιοποιημένο" SELECT από τη βάση και το πρόβλημα λύθηκε