

# Εφαρμογή διαχείρισης

Νίκος Αυγούστης Α.Μ. 10273

## Περιεχόμενα

<b>ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ.....</b>	<b>1</b>
<b>ΠΕΡΙΛΗΨΗ:.....</b>	<b>1</b>
<b>.....</b>	<b>2</b>
<b>ΕΙΣΑΓΩΓΗ:.....</b>	<b>2</b>
<b>ΔΟΜΙΚΑ ΜΕΡΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ:.....</b>	<b>2</b>
<b>ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ:.....</b>	<b>3</b>
<b>-SCREEN SHOTS.....</b>	<b>4</b>
<b>-ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....</b>	<b>8</b>
<b>ΑΝΑΦΟΡΕΣ.....</b>	<b>8</b>

## Περίληψη:

Η εφαρμογή αποτελεί ένα σύστημα διαχείρισης μαθητών και πληρωμών για φροντιστήρια και έχει τις εξής επιλογές:

1. Δημιουργία/Διαγραφή δύο πινάκων στη βάση δεδομένων οι οποίοι θα κρατάν ο πρώτος τα στοιχεία κάθε μαθητή που εγγράφεται στο φροντιστήριο και ο δεύτερος για τον κάθε μαθητή τις μηνιαίες πληρωμές και τυχόν οφειλές.
2. Μια φόρμα εγγραφής νέων μαθητών.
3. Μια φόρμα εισαγωγής μια πληρωμής.
4. Μια σελίδα στην οποία θα εμφανίζονται αυτόματα τα ονόματα μαθητών που έχουν κάποια τρέχουσα οφειλή.
5. Την δυνατότητα αναζήτησης η οποία θα προβάλλει τα αποτελέσματα και θα δίνει την δυνατότητα επεξεργασίας/διαγραφής των στοιχείων ενός μαθητή.

## Εισαγωγή:

Το πρόβλημα όπως δόθηκε από τον καθηγητή είναι το εξής:

Η δημιουργία μιας εφαρμογής διαχείρισης η οποία θα χρησιμοποιεί HTML/CSS και θα έχει τις παρακάτω δυνατότητες:

- 1.Εγκατάσταση των πινάκων στη Β.Δ.
- 2.Εισαγωγή τιμών στον πίνακα από φόρμα με έλεγχο τιμών για τα υποχρεωτικά πεδία.
- 3.Προεπισκόπηση των εγγράφων του πίνακα με χρήση paginator.
- 4.Δυνατοτητα τροποποίησης/διαγραφής των εγγράφων ενσωματωμένη στην προεπισκόπηση με βάση την id κάθε τιμής.
- 6.Δυνατοτητα αναζήτησης.

## Δομικά Μέρη της εφαρμογής:

Η εφαρμογή αποτελείτε από 12 αρχεία .php και 1 αρχείο .css

-Τα αρχεία index.php και front\_page.php είναι τα δυο αρχεία που λειτουργούν ως αρχικές σελίδες το πρώτο όταν δεν υπάρχει πίνακας στη βάση δεδομένων ενώ το δεύτερο όταν υπάρχουν οι πίνακες.

-Τα αρχεία create.php και table\_delete.php δημιουργούν και καταστρέφουν τους πίνακες στη βάση δεδομένων αντίστοιχα.

-Το αρχείο add\_student.php εισάγει στοιχεία στον πίνακα students.

-Το αρχείο add\_payment.php περιέχει τη φόρμα η οποία εισάγει στοιχεία στον πίνακα payments.

-Τα αρχεία update\_payment.php και update\_student.php ενημερώνουν τα είδη υπάρχοντα στοιχεία στους δυο πίνακες.

-Το αρχείο delete1.php διαγράφει στοιχεία από του δυο πίνακες.

-Το αρχείο debt.php με τη βοήθεια του αρχείου pagination\_data.php προβάλλει αυτόματα τους χρήστες που έχουν κάποια οφειλή προς το σύστημα.

-Το αρχείο search.php εκτελεί την αναζήτηση και προβάλλει τα αποτελέσματά σε ξεχωριστή σελίδα.

-Τέλος το αρχείο styles.css περιέχει όλες τις τροποποιήσεις τις οποίες δεν δίνει είδη η χρήση του bootstrap.

Η βάση δεδομένων τώρα περιέχει δυο πίνακες students και payments οι οποίοι είναι συνδεδεμένοι μεταξύ τους. Ο students έχει μια θέση για int(4), τρεις θέσεις για varchar(80), μια θέση για bigint(15) και μια θέση για date. Ο πίνακας payments έχει τρεις θέσεις για double(10,2) και δυο θέσεις για int(4). Συνδέονται μέσω των δυο int τιμών τους uid και id.

## Χρήση της εφαρμογής:

-Queries:

**-“create table students(id int(4) auto\_increment,fname varchar(80),lname varchar(80),contact bigint(15), class varchar(80),enr\_date date,primary key(id));”**

**“create table payments(pid int(4) auto\_increment,uid int(4),to\_be\_paid double(10,2),paid double(10,2),debt double(10,2),pay\_date date,primary key(pid), constraint mk foreign key(uid) references students(id) on delete cascade on update cascade)”**

Το παραπάνω query δημιουργεί και συνδέει τους δυο πίνακες στη βάση μας. Μερικές εξηγήσεις για τους τύπους δεδομένων: Στην μεταβλητή contact χρησιμοποιώ bigint(15) για να αποθηκεύει τηλέφωνα τα οποία αποτελούνται από 10 ψηφία χωρίς κωδικούς και περισσότερα από 10 αν συμπεριλάβουμε και τους κωδικούς. Τα debt,paid,to\_be\_paid ως χρηματικά ποσά δηλώνονται σαν πραγματικοί με δυο στοιχεία μετά την υποδιαστολή.

**-“drop table if exists payments;”**

**“drop table if exists students”**

Διαγράφει τους πίνακες μας ξεκινώντας από τον payments ο οποίος είναι εξαρτημένος από τον students και δεν επιτρέπει την διαγραφή του students πρώτα για αυτό και η συγκεκριμένη σειρά.

**-“insert into students values (‘,?,?,?,?’)”**

Εισάγει στον πίνακα students στοιχεία τα οποία θα δοσουμε μετά.

**-“insert into payments values (‘,?,?,?,?’)”**

Όμοια με πριν αλλά για τον πίνακα payments.

**“delete from students where id=“.\$\_POST[‘id’]**

Διαγράφει από τον πίνακα students το στοιχείο με id ίσο με την τιμή που κουβαλάει η POST και τις αντίστοιχες εγγραφές από τον άλλο πίνακα αφού είναι συνδεδεμένη με περιορισμούς cascade.

**-“update payments set debt=“.\$\_POST[‘debt’].”’,paid=“.\$\_POST[‘paid’].”’,pay\_date=“.\$\_POST[‘pdate’].” ’where pid= “.\$\_POST[‘id’];**

Ενημερώνει την εγγραφή στον πίνακα payments για τις τιμές που δίνουμε με την POST για το συγκεκριμένο pid.

**-“select \* from payments where pid=“.\$\_GET[‘pid’];**

Διαλέγει και μας δείχνει όλα τα στοιχεία του πίνακα με συγκεκριμένο pid για να χρησιμοποιήσουμε στο πάνω query.

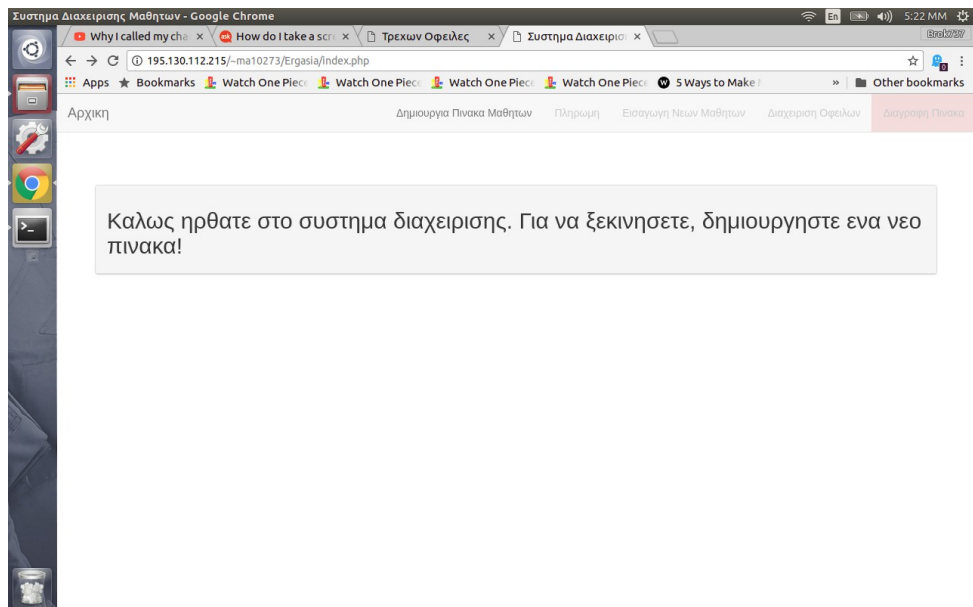
Αντίστοιχα υπάρχουν query για την ενημέρωση του πίνακα students.

**-“ select students.\*, payments.\* from students right join payments on students.id = payments.uid and payments.to\_be\_paid != payments.paid where students.id is not null order by students.id limit \$start, \$per\_page ”**

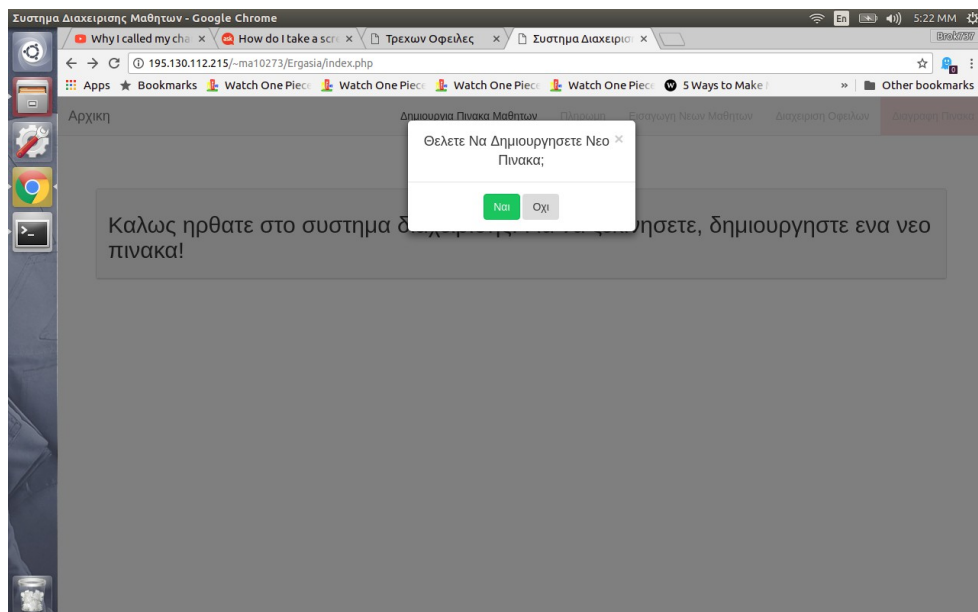
Το παραπάνω τρομακτικό query κάνει ένα πολύ απλό αλλά βολικό πράγμα, βρίσκει τα στοιχεία στους δυο πίνακες για τα οποία ισχύει id=uid και τα paid != to\_be\_paid δηλαδή στους δυο πίνακες βρίσκει ποιος μαθητής έχει πληρώσει λιγότερα από ότι ήταν να πληρώσει και επίσης το id να μην είναι null και κολλάει τα στοιχεία του δεύτερου πίνακα στα δεξιά του πρώτου και τα κατατάσσει ως προς το id και τα προβάλλει με κάποιο τρόπο που εξυπηρετεί το pagination.

## -Screen shots

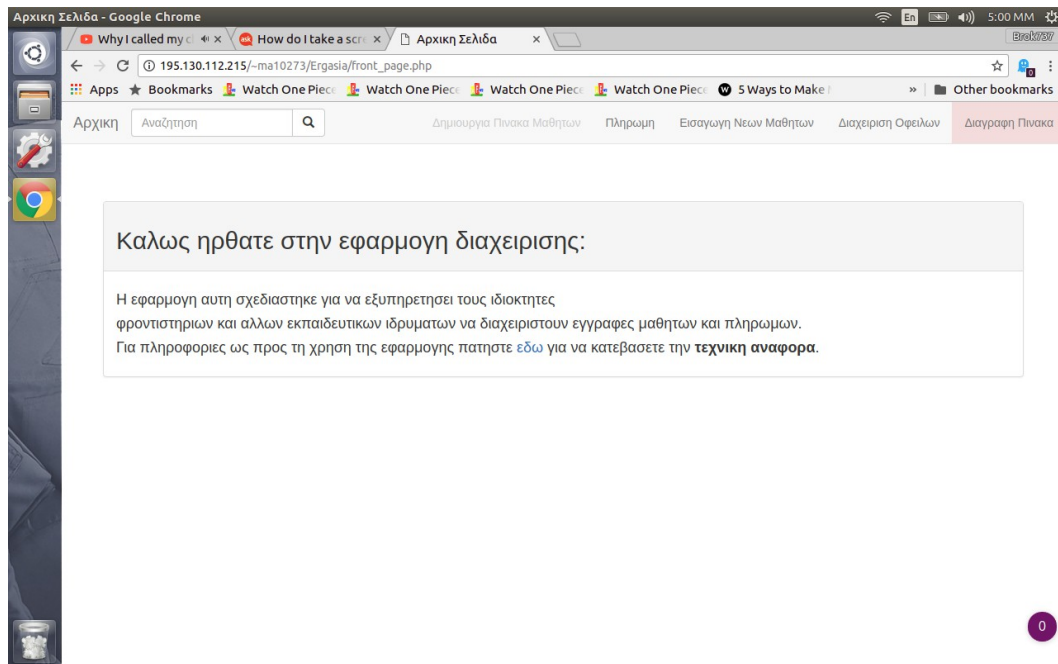
-Αρχική σελίδα πριν την δημιουργία των πινάκων.



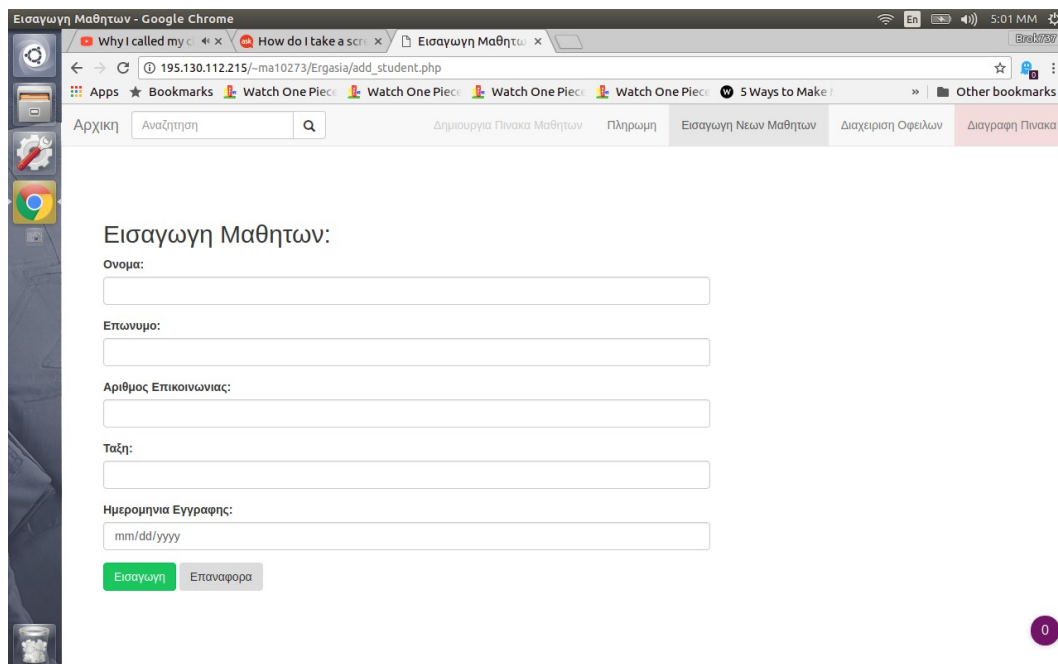
-Πατώντας το πλήκτρο για τη δημιουργία των πινάκων, Πατώντας ναι δημιουργείτε και αλλάζουμε οθόνη ενώ με το όχι παραμένουμε στην ίδια οθόνη.



-Αρχική οθόνη μετά τη δημιουργία πινάκων με όλες τις δυνατές επιλογές της εφαρμογής.



-Προσθέτοντας νέο μαθητή.



-Προσθέτοντας πληρωμή.

Πληρωμή - Google Chrome

Why I called my c... x How do I take a scr... x Πληρωμή x

195.130.112.215/~ma10273/Ergasia/add\_payment.php

Apps Bookmarks Watch One Piece Watch One Piece Watch One Piece Watch One Piece 5 Ways to Make Other bookmarks

Αρχική Αναζήτηση Δημιουργία Πινάκα Μαθητών Πληρωμή Εισαγωγή Νέων Μαθητών Διαχείριση Οφειλών Διαγραφή Πινάκα

### Πληρωμή:

Όνομα:

Επωνυμο:

Ποσο Πληρώμης:

Διδακτρα:

Ημερομηνια:

0

-Εμφάνιση όλων των εκκρεμών οφειλών προς εμάς.

Τρεχων Οφειλες - Google Chrome

Why I called my cha... x How do I take a scr... x Τρεχων Οφειλες x Τρεχων Οφειλες x

195.130.112.215/~ma10273/Ergasia/debt.php#

Apps Bookmarks Watch One Piece Watch One Piece Watch One Piece Watch One Piece 5 Ways to Make Other bookmarks

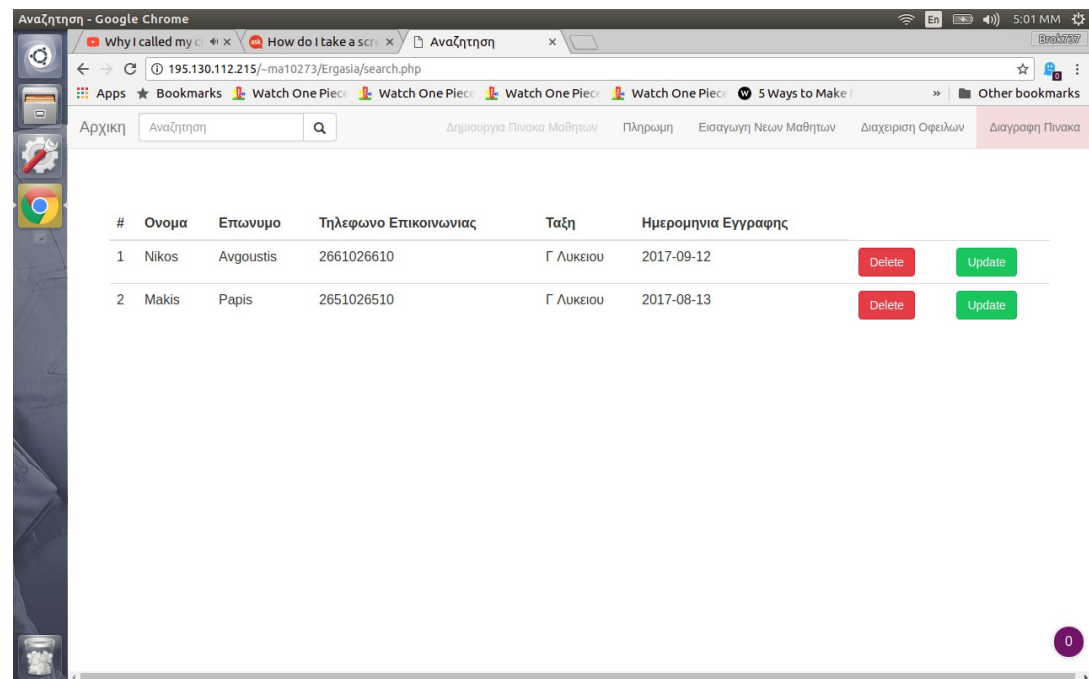
Αρχική Αναζήτηση Δημιουργία Πινάκα Μαθητών Πληρωμή Εισαγωγή Νέων Μαθητών Διαχείριση Οφειλών Διαγραφή Πινάκα

### Τρεχων Οφειλες:

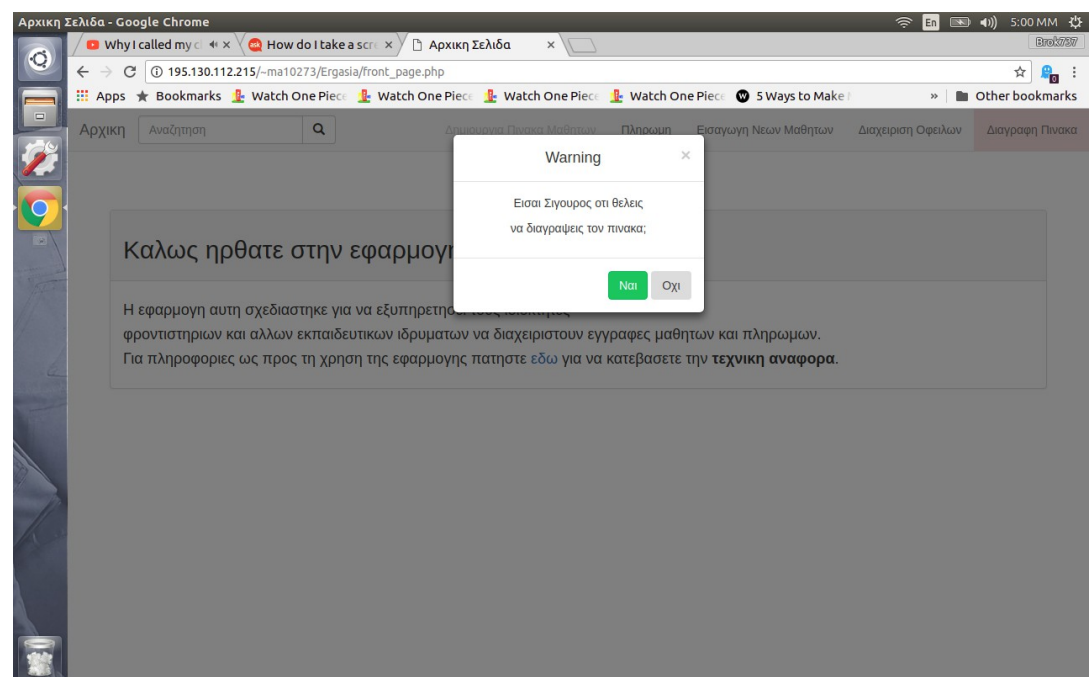
#	Όνομα	Επωνυμο	Τηλεφωνο Επικοινωνιας	Χρεως	Ημερομηνια Πληρωμης	
1	Nikos	Avgoustis	2661026610	20.00	2017-09-12	<input type="button" value="Update"/>
1	Nikos	Avgoustis	2661026610	17.00	2017-09-22	<input type="button" value="Update"/>
1	Nikos	Avgoustis	2661026610	17.00	2017-09-22	<input type="button" value="Update"/>
2	Makis	Papis	2651026510	25.00	2017-09-10	<input type="button" value="Update"/>
2	Makis	Papis	2651026510	25.00	2017-09-10	<input type="button" value="Update"/>

1 2 3

-Αναζήτηση.



-Επιλογή διαγράψης υπάρχον πίνακα.



-Ενημέρωση στοιχείων μαθητή.

195.130.112.215/~ma10273/Ergasia/update\_student.php?id=1 - Google Chrome

195.130.112.215/~ma10273/Ergasia/update\_student.php?id=1

Αρχική Αναζήτηση Δημιουργία Πινάκων Μαθητών Πληρωμή Εισαγωγή Νέων Μαθητών Διαχείριση Οφειλών Διαγραφή Πινάκων

Id: 1

First Name: Nikos

Last Name: Avgoustis

Contact Number: 2661026610

Class: Γ

Enroll Date: 09/09/2017

Update Reset

## -Μελλοντικές επεκτάσεις.

Η εφαρμογή θα μπορούσε να εμπλουτιστεί με την δυνατότητα εγκατάστασης πολλών διαφορετικών πινάκων πχ για κάθε σχολική χρονιά, ένα σύστημα ταυτοποίησης χρηστών έτσι ώστε πολλά ιδρύματα να την χρησιμοποιούν το καθένα με τη δική του βάση δεδομένων.

## Αναφορές

Ahmed, S., n.d. *Sarfraz Ahmed Blog.* [Online]

Available at: <https://sarfraznawaz.wordpress.com/>

Stackoverflow, n.d. *stackoverflow.* [Online]

Available at: <https://www.stackoverflow.com>

W3schools, n.d. *W3schools.* [Online]

Available at: <https://www.w3schools.com>

Κοντογιάννης, Σ., n.d. *Καλυψω.* [Online]

Available at: <https://spooky.math.uoi.gr/files/index.html?dir=L2RpcmVjdG9yeS9Db3Vyc2VzL0Jhc2Vpc19EZWRvbWVub25fa19BREUvVGhlb3JpYQ==>