



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Τμήμα Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων

Μάθημα

Τεχνολογίες Δικτύων και Νέφους

Τίτλος Εργασίας

Secure & Trusted Cloud Computing

Ονοματεπώνυμο:

Αριθμός Μητρώου:

Γιακουμιδάκης Κων/νος

321/2011026

Πέππας Κων/νος

321/2011134

Βάνης Αναστάσιος

321/2012017

Χαϊκάλης Νικόλαος

321/2012200

Διδάσκων: Σκιάνης Χ.

Ημερομηνία Παράδοσης: 23/05/2017

Σάμος, Καρλόβασι 83200

Abstract

Σαν θέμα διαλέξαμε το Secure & Trusted Clouds και αποφασίσαμε να σχεδιάσουμε μια cloud πλατφόρμα διαχείρισης φωτογραφιών (portfolio). Πιο αναλυτικά, θέλουμε να δημιουργήσουμε έναν ιστότοπο όπου κάποιος client θα εγγράφεται στην υπηρεσία μας όπου και θα μπορεί να αποθηκεύσει/αρχειοθετήσει τις φωτογραφίες του, καθώς θα του παρέχεται η δυνατότητα να κρυπτογραφήσει τις φωτογραφίες αυτές. Οι πελάτες μας θα μπορούν να συνδεθούν στην υπηρεσία μας ώστε να κατεβάσουν τις φωτογραφίες τους ή να ανεβάσουν νέες χρησιμοποιώντας αυτόματα τις διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης όπου είναι απαραίτητες. Η τεχνολογία που θα χρησιμοποιήσουμε για την δημιουργία του back end της πλατφόρμας είναι php σε συνδυασμό με MySQL για την δημιουργία της βάσης δεδομένων που θα χρειαστούμε. Για το front end θα χρησιμοποιήσουμε html / JavaScript & Bootstrap ώστε το περιβάλλον μας να είναι ωραία προσπελάσιμο σε όλες τις οθόνες.

Cryptography

Αρχικά όταν κάποιος πελάτης εγγράφεται στην υπηρεσία μας θα δημιουργούμε αυτόματα τα εξής κλειδιά:

- Δυο ζευγάρια Private / Public keys.
- Δυο Secret Keys.

Από εδώ και πέρα το πρώτο ζευγάρι θα το αναφέρουμε ως Priv1/Pub1, το δεύτερο ως Priv2/Pub2 και στα συμμετρικά κλειδιά θα αναφερόμαστε ως Sec1, Sec2. Τα Priv1/Pub1 & Sec1 θα τα χρησιμοποιήσουμε για την κρυπτογραφία του password και τα Priv2/Pub2 & Sec2 θα τα χρησιμοποιήσουμε για την κρυπτογραφία των φωτογραφιών. Σκοπός μας είναι να δημιουργήσουμε ένα Hybrid Cryptosystem για την κρυπτογράφηση των passwords και για την κρυπτογράφηση των φωτογραφιών.

- ### Password Encryption

Το password του client κρυπτογραφείται ως εξής:

1. $\text{HashPass} = \text{HashFunction}(\text{password});$
2. Τώρα το αποτέλεσμα από το hashαρισμένο password θα το κρυπτογραφούμε με ένα Secret Key (Sec1) δηλαδή:
 $\text{SecKeyEnPass} = \text{Sec1}(\text{HashPass});$
3. Στην συνέχεια το αποτέλεσμα αυτό (SecKeyEnPass) θα κρυπτογραφείται με το δημόσιο κλειδί Pub1 δηλαδή:
 $\text{EncryptPass} = \text{Pub1}(\text{SecKeyEnPass}).$
Όπου αυτό είναι και το τελικό μας αποτέλεσμα, πλέον είμαστε στο σημείο που αποθηκεύουμε τον κρυπτογραφημένο κωδικό στην βάση μας.
4. Επιπλέον να τονίσουμε πως το Secret Key (Sec1) κρυπτογραφείται με το Pub1 και το κρυπτογράφημα αυτό αποθηκεύτε στην DB δηλαδή:
 $\text{EnSec1} = \text{Pub1}(\text{Sec1});$

- Password Decryption

Όταν κάποιος χρήστης κάνει login στην υπηρεσία μας θα πρέπει να ταυτοποιήσουμε το password του. Η διαδικασία της αποκρυπτογράφησης του password είναι η εξής:

1. Hashαρουμε το password που έδωσε ο client ώστε να συγκρίνουμε αν οι δύο hashes είναι ίδιες. Αν οι δύο hashes είναι ίδιες τότε ο client έδωσε το σωστό password.
 $\text{GivenPasswordHash} = \text{HashFunction}(\text{password});$
2. Με το Priv1 αποκρυπτογραφούμε το EnSec1 που είναι το αποτέλεσμα από το $\text{EnSec1} = \text{Pub1}(\text{Sec1});$ Άρα ως return έχουμε το Sec1.
3. Στην συνέχεια αποκρυπτογραφούμε το EncryptPass που ήταν το αποτέλεσμα της $\text{EncryptPass} = \text{Pub1}(\text{SecKeyEnPass})$ και παίρνουμε ως return το SecKeyEnPass που είναι το αποτέλεσμα από το $\text{SecKeyEnPass} = \text{Sec1}(\text{HashPass});$
4. Σε αυτό εδώ το σημείο χρησιμοποιώντας το Sec1 που αποκρυπτογραφήσαμε προηγούμενος θα το χρησιμοποιήσουμε για να αποκρυπτογραφήσουμε το SecKeyEnPass για να βρούμε το hash password από το σωστό password.
5. Τέλος θα checkαρουμε αν η hash από το βήμα 4.(σωστό hash password) είναι ίδια με την hash από το βήμα 1. (GivenPasswordHash) και αν οι δυο hashes είναι ίδιες τότε ο χρήστης έδωσε το σωστό password.

- Photographs Encryption

Ο πελάτης όταν θα ανεβάζει μια φωτογραφία θα πρέπει αυτόματα να κρυπτογραφείται. Η διαδικασία κρυπτογράφησης μιας φωτογραφίας θα είναι η εξής:

1. Αρχικά αποκρυπτογραφούμε με το ιδιωτικό κλειδί (Priv2) το αποτέλεσμα της κρυπτογράφησης $EnSec = Pub2(Sec2)$; Όστε να βρούμε το Sec2 αφού το έχουμε κρυπτογραφημένο στην DB, ώστε και κάποιος επιτιθέμενος να κλέψει το EnSec να χρειάζεται το Priv2 για να το βρει.
2. Παίρνουμε την φωτογραφία που ανέβασε ο χρήστης (Photo1) και θα την κρυπτογραφήσουμε με το συμμετρικό κλειδί (Sec2) που αποκρυπτογραφήσαμε στο προηγούμενο στάδιο δηλαδή:
 $EnPhotoSec = Sec2(Photo1)$;
3. Στην συνέχεια το αποτέλεσμα από την παραπάνω κρυπτογράφηση (EnPhotoSec) το παίρνουμε και το κρυπτογραφούμε με το Pub2 δηλαδή:
 $EnPhoto = Pub2(EnPhotoSec)$;
4. Upload στην DB την EnPhoto που είναι η κρυπτογραφημένη φωτογραφία.

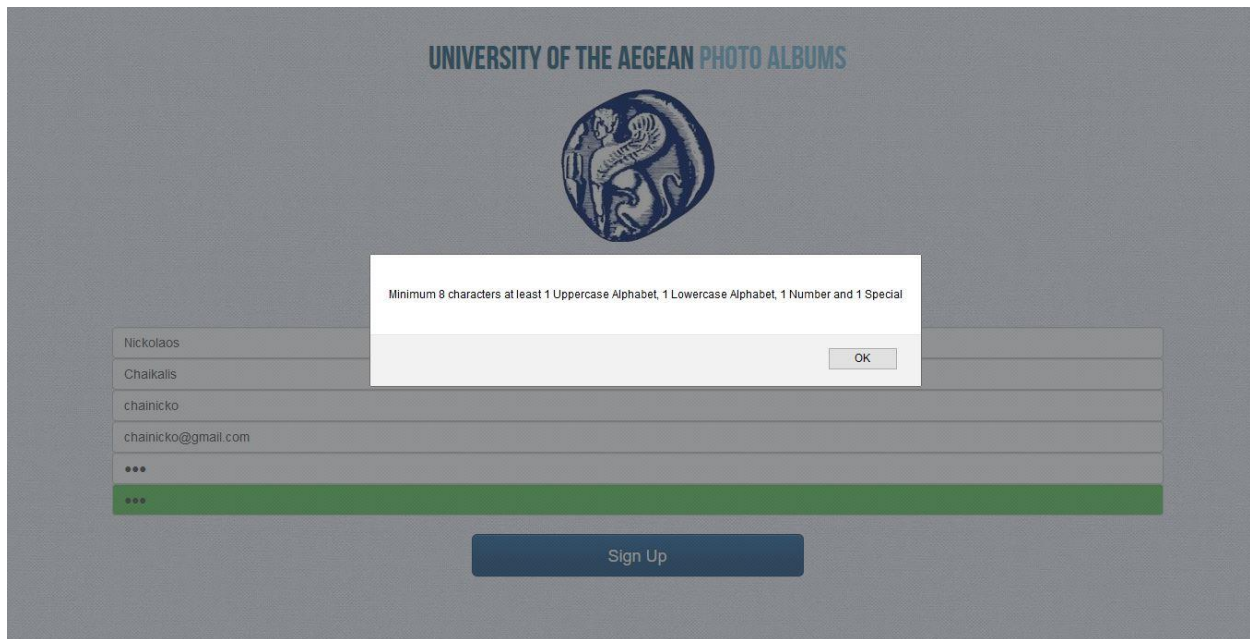
- Photographs Decryption

Βρισκόμαστε στο σημείο όπου ένας πελάτης έχει συνδεθεί στην υπηρεσία μας και θέλει να δει κάποια φωτογραφία. Για να δει κάποια φωτογραφία θα πρέπει να την αποκρυπτογραφήσουμε. Η διαδικασία αποκρυπτογράφησης μια φωτογραφίας είναι η εξής:


1. Με το Priv2 αποκρυπτογραφούμε το EnSec2 που είναι το αποτέλεσμα από το $\text{EnSec2} = \text{Pub2}(\text{Sec2})$; Άρα ως return έχουμε το Sec2.
2. Στην συνέχεια με το Priv2 αποκρυπτογραφούμε το EnPhoto όπου $\text{EnPhoto} = \text{Pub2}(\text{EnPhotoSec})$; Άρα ως έχουμε στα χέρια μας το EnPhotoSec που είναι η κρυπτογραφημένη φωτογραφία μέσω του Secret Key.
3. Στην συνέχεια παίρνουμε το Sec2 που αποκρυπτογραφήσαμε στο βήμα 1 και αποκρυπτογραφούμε το EnPhotoSec, όπου ως return θα πάρουμε την φωτογραφία που ανέβασε ο χρήστης.

Screenshots

Κάποιος χρήστης προσπαθεί να κάνει εγγραφή στην υπηρεσία μας με αδύναμο password.



UNIVERSITY OF THE AEGEAN PHOTO ALBUMS



Minimum 8 characters at least 1 Uppercase Alphabet, 1 Lowercase Alphabet, 1 Number and 1 Special

Nickolaos

Chaikalis

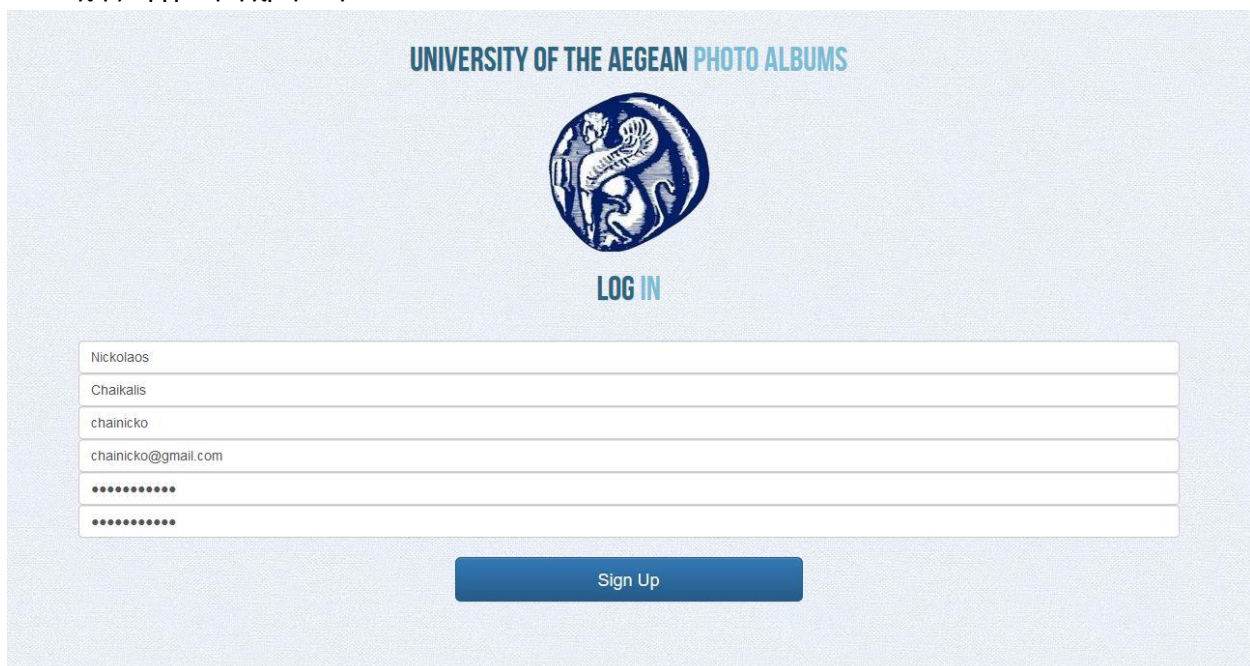
chainicko

chainicko@gmail.com


OK

Sign Up

Επιτυχής εγγραφή χρήστη:



UNIVERSITY OF THE AEGEAN PHOTO ALBUMS



LOG IN

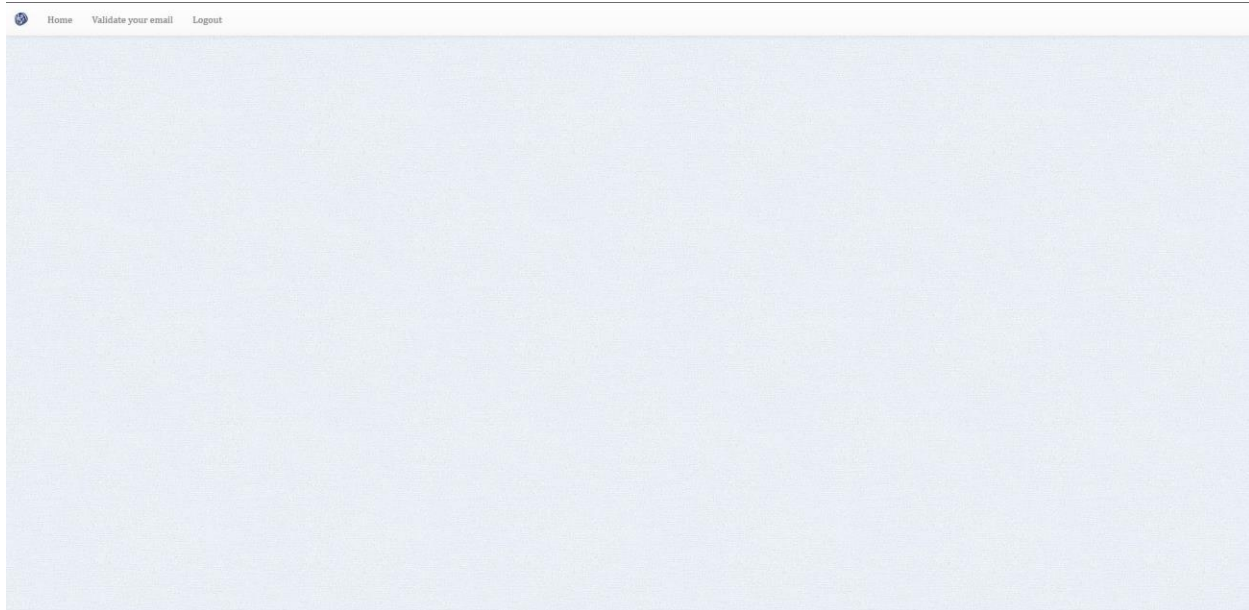
Nickolaos

Chaikalis

chainicko

chainicko@gmail.com

Sign Up



Παρατηρούμε πως δεν μπορούμε να ανεβάσουμε κάποιο φωτογραφία γιατί ο χρήστης δεν έχει πιστοποιήσει την εγγραφή του. Για να πιστοποιήσει ο χρήστης την εγγραφή του, του στέλνουμε ένα token στο email από όπου θα συνδεθεί για να πιστοποιήσει την αυθεντικότητα του.

Το token είναι το εξής:

```
2017-05-18 14:48:32.859.eml x
1 Date: Thu, 18 May 2017 11:48:32 +0000
2 Subject: University of the Aegean registration
3 To: chainicko@gmail.com
4 X-PHP-Originating-Script: 0:SendMail.php
5 From: noreplyUoTARegister@test.test.com
6
7 Welcome aboard !!!! :) To complete your registration click the link below:
8 http://localhost/nefos\_php/Validate.php?token=5ecelaea0872860584fbdae91c2d6437&user=chainicko
9
```

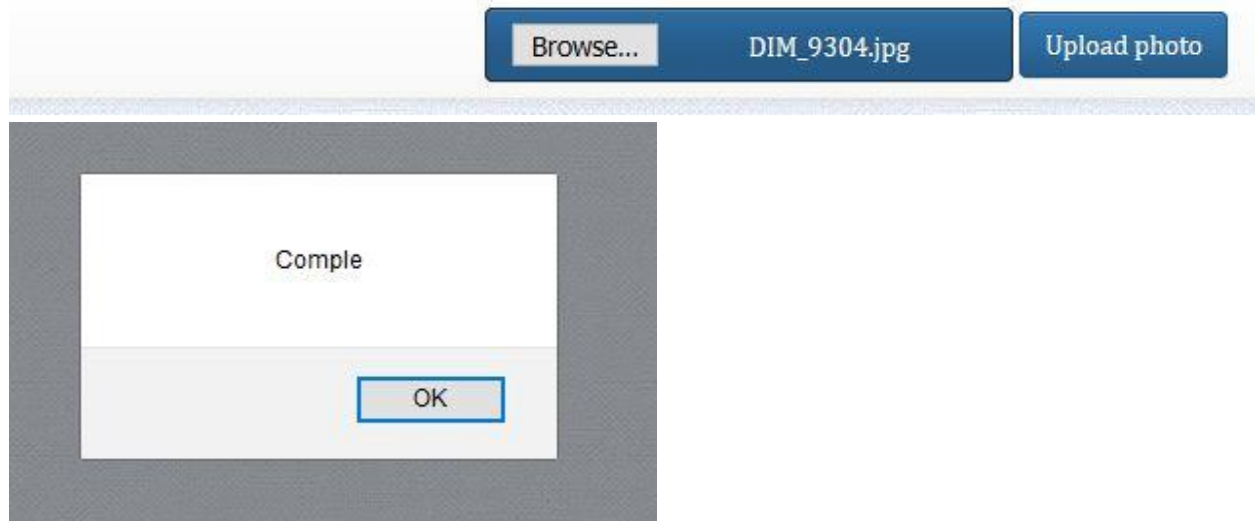
Όπως παρατηρούμε το token μόνο για το Localhost στην περίπτωση που είχαμε κάποιο Host θα βάζαμε το site του Host/Validate.php?token=...

Κάνουμε validate μέσα από το email:



Για να τεσταρουμε το email στο localhost χρησιμοποιήσαμε την εφαρμογή Test Mail Server Tool.

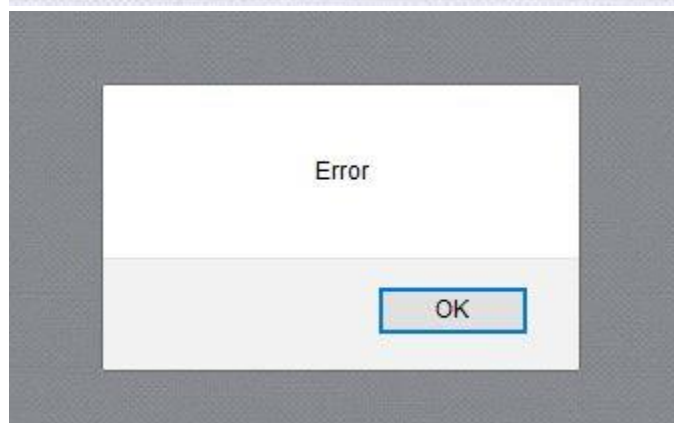
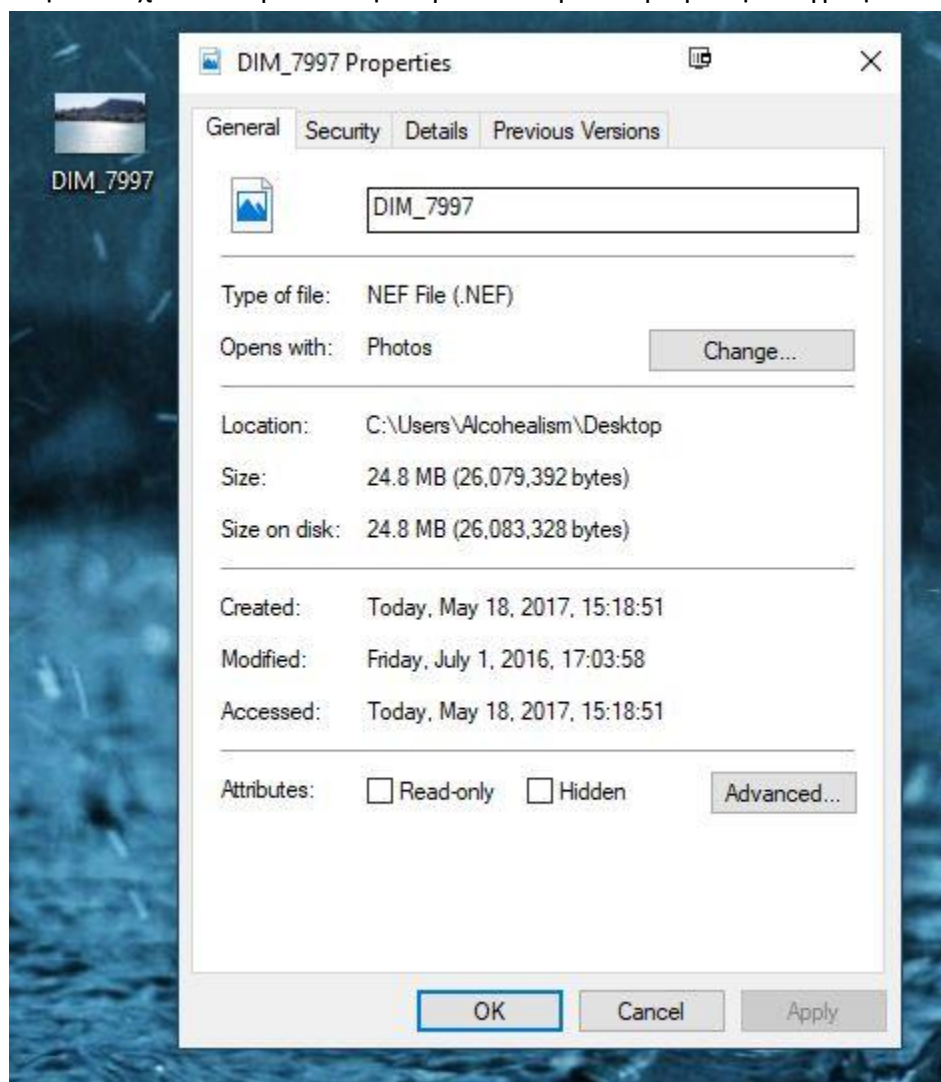
Πλέον μπορούμε να ανεβάσουμε τις φωτογραφίες:



Επιλέξαμε μια φωτογραφία με όνομα DIM_9304.jpg και την ανεβάσαμε στο cloud μας.



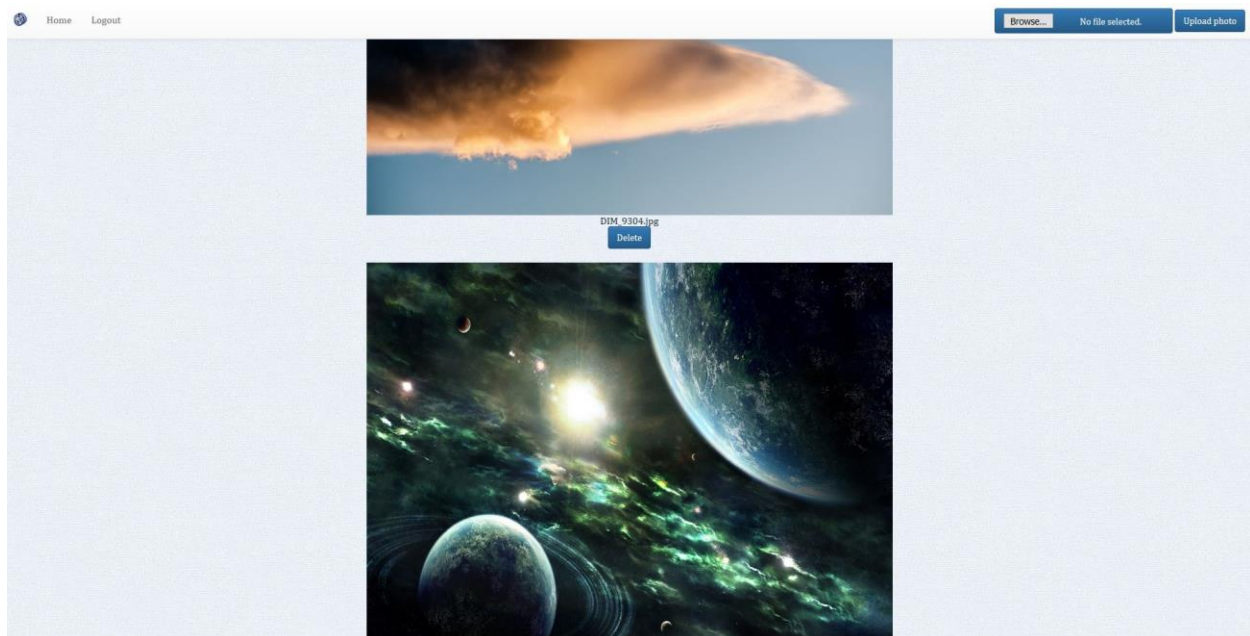
Στην συνέχεια θα προσπαθήσουμε να ανεβάσουμε μια φωτογραφία .nef όπου είναι > 8MB



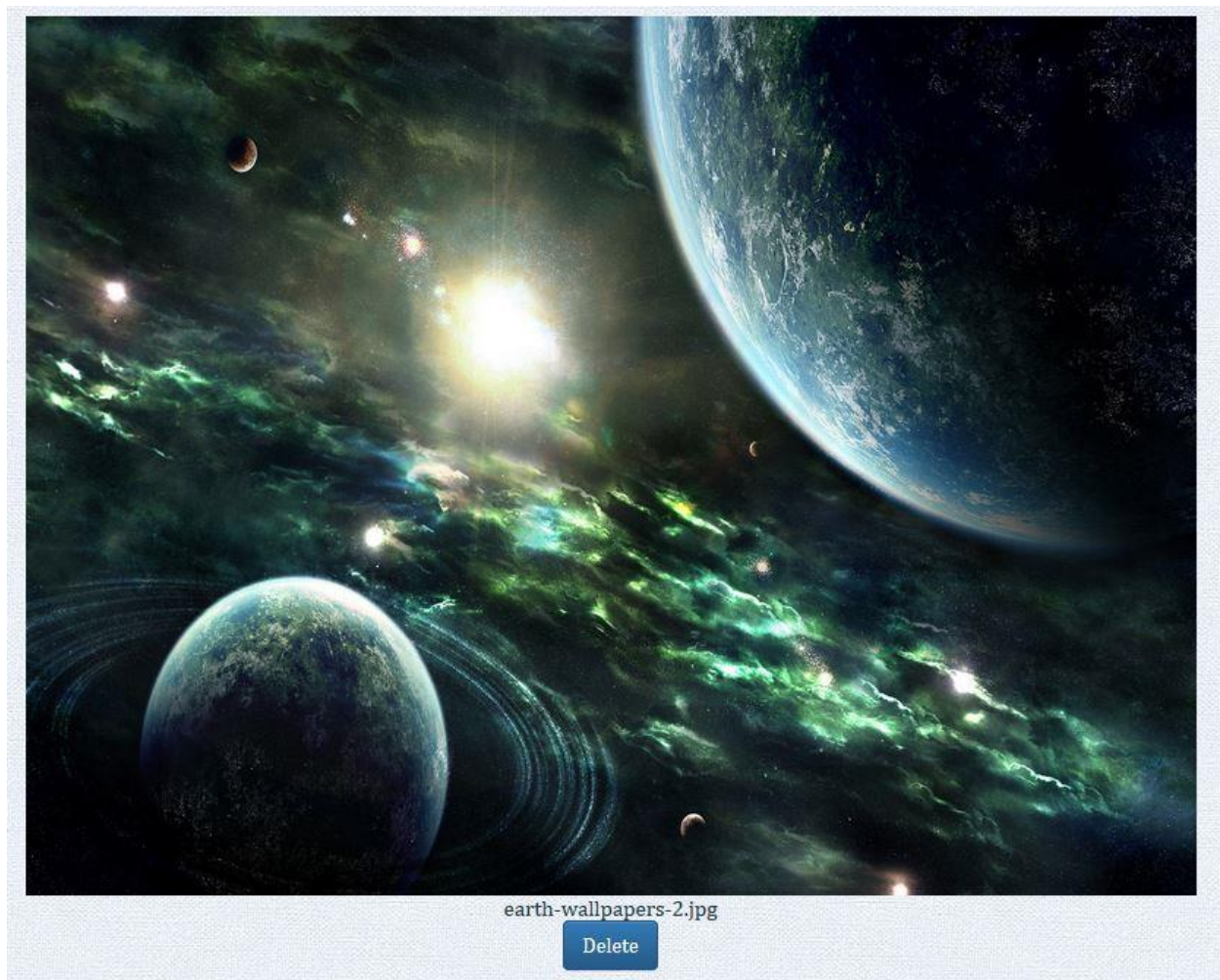
Και βλέπουμε ότι δεν μπορούμε να την ανεβάσουμε αφού το .nef δεν υποστηρίζεται από την εφαρμογή μας. Το ίδιο θα μας εμφάνιζε αν βάζαμε και ένα .png >8.1MB.

Γιατί συγκεκριμένα 8.1MB; Γιατί χρησιμοποιούμε κρυπτογράφηση Private Public Key με τον RSA. Δηλαδή όταν κρυπτογραφούμε κάτι πχ την φωτογραφία με τον RSA η encrypted value πρέπει να είναι μικρότερη από το κλειδί που έχουμε φτιάξει. Αν για παράδειγμα χρησιμοποιούμε 1024-bit key θεωρητικά μπορούμε να κρυπτογραφήσουμε μέχρι 1023. Στην προκειμένη περίπτωση που έχουμε φωτογραφίες βάζουμε ένα μεγάλο κλειδί των 8192-bits. Το κακό με αυτό είναι πως έχουμε μια μικρή καθυστέρηση στο register γιατί πρέπει να δημιουργηθούν τα ζευγάρια.

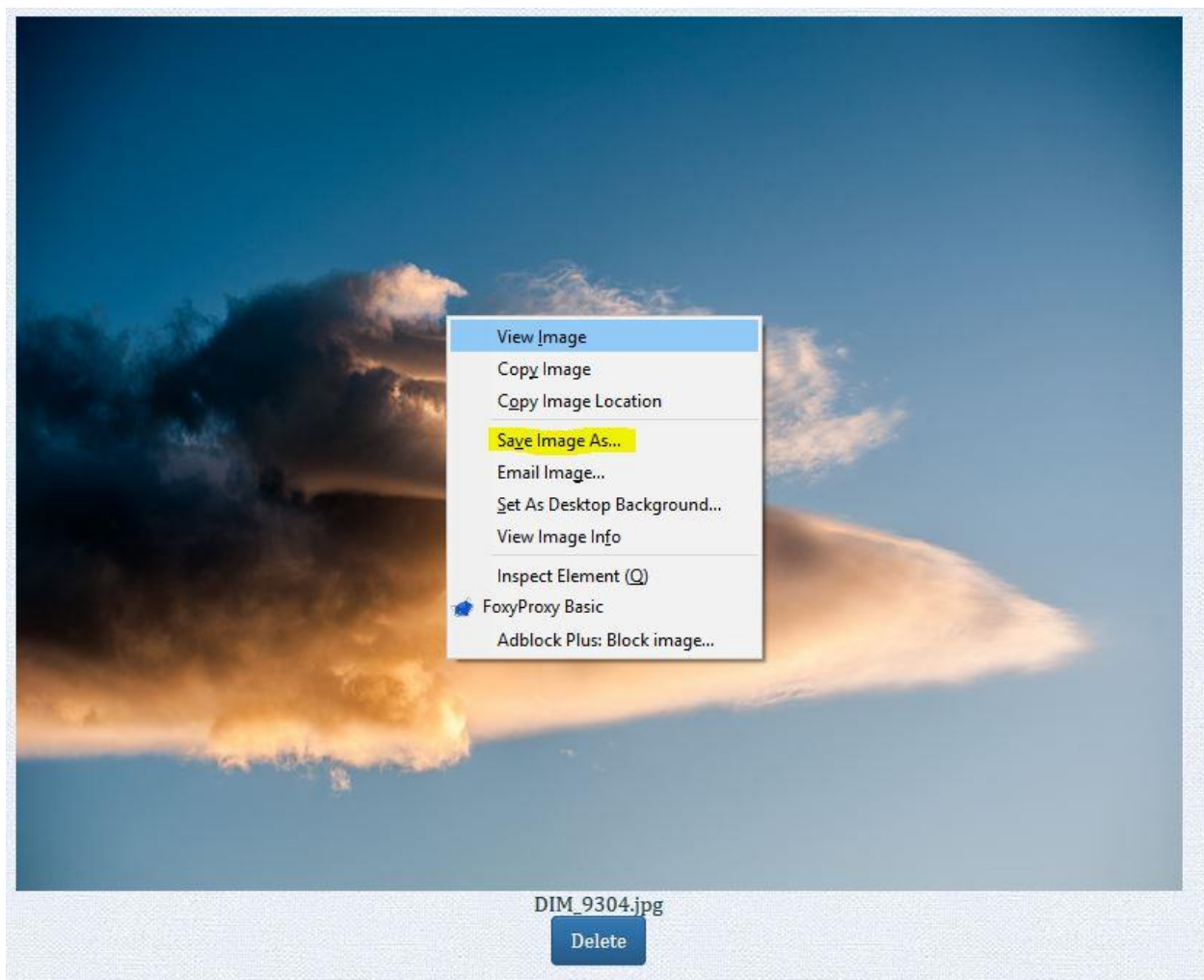
Upload άλλης φωτογραφίας:



Τώρα θα κάνουμε delete της φωτογραφίας:



Για save: left click->Save Image as:



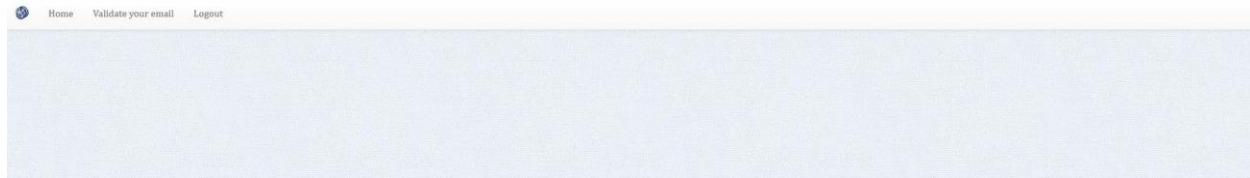
Όπως είχαμε προαναφερθεί έχουμε 2 Βάσεις Δεδομένων για λόγους ασφάλειας. Κανονικά μια βάση θα έτρεχε στον Host μαζί με την σελίδα και η άλλη κάπου άλλου αλλά λόγω του ότι είμαστε σε localhost τρέχουν όλα στο ίδιο μηχάνημα:



Χρήστες της βάσης μας, ο ένας είναι validated ο άλλος όχι:

+ Options													
← T →													
		id	email	password	name	surname	username	SecKey1	PubKey1	SecKey2	PubKey2	token	validate
<input type="checkbox"/>	Edit Copy Delete	14	chainicko@gmail.com	GhQGP9WCouB7BIS1JHqhgSX+Drmk+SZ20ToqthxNCO9Bkg/QZg...	Nickolaos	Chaikalis	chainicko	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 32 B]	1
<input type="checkbox"/>	Edit Copy Delete	15	kostpep@yahoo.gr	aIa6LR0uBExdDa5IQWVGZ83z3Og5+RdZolb/2pF-eg55/b0gqsK...	Kostas	Peppas	kostpep	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 32 B]	0

Ο χρήστης που δεν είναι validate στο σύστημα δεν μπορεί να ανεβάσει φωτογραφίες:

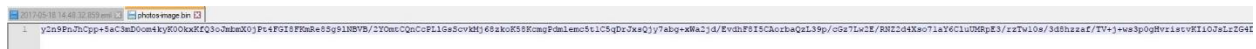


Ο πίνακας με τις φωτογραφίες:

+ Options						
← T →						
		id	user_fk	name	image	
<input type="checkbox"/>	Edit Copy Delete	155	14	DIM_9304.jpg	[BLOB - 8.4 MiB]	

Αν δεν ήταν κρυπτογραφημένες θα βλέπαμε τον τύπο του αρχείου πχ .jpg

Αν κατεβάσουμε την φωτογραφία από την DB βλέπουμε αυτό:



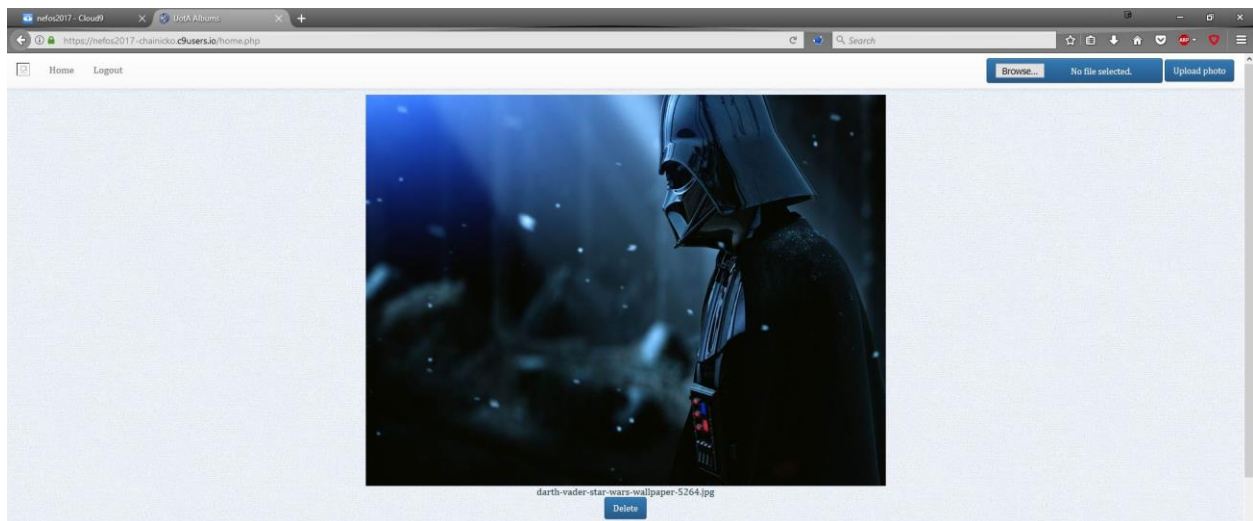
Άρα κάποιος εισβολέας θα καταφέρει να πάρει μια κρυπτογραφημένη φωτογραφία.

Σε αυτό το screenshot βλέπουμε τα passwords των 2 user παρατηρούμε πως το κρυπτογράφημα είναι διαφορετικό. Λογικό αφού χρησιμοποιούμε διαφορετικά ζεύγη Private Public & Secret Keys. Τα passwords είναι και τα δύο τα ίδια αν γράψουμε Olympri@kos7 και για τους 2 users θα μπορούμε στο account.



Έχουμε ανεβάσει το project στο <https://c9.io/> ώστε να είναι 100% cloud. Το project δεν είναι διαθέσιμο σε όλο το κοινό καθώς το c9.io είναι μια πλατφόρμα για να γράφουμε κώδικα (συγγραφή κώδικα στο cloud) αλλά μας παρέχει και όλες τις δυνατότητες που μας παρέχει και ένας host απλά το Link δεν είναι ορατό σε άτομα που δεν είναι συνεργάτες στο Project.

Τέλος βλέπουμε μερικά screenshots από την πλατφόρμας μας να τρέχει live:



Data Base:

The screenshot shows the phpMyAdmin web interface. On the left, a sidebar displays a database structure with folders for 'information_schema', 'myq', 'nefosdb', 'photos', 'users', 'nefosprivatekeysdb', 'performance_schema', and 'phpmyadmin'. The main area shows a SQL query: `SELECT * FROM 'users' LIMIT 0, 30`. Below the query, a table of results is displayed with columns: id, email, password, name, surname, username, SecKey1, PubKey1, SecKey2, PubKey2, token, and validate. The first row of data is visible.

id	email	password	name	surname	username	SecKey1	PubKey1	SecKey2	PubKey2	token	validate
17	chainicko@gmail.com	YUdQZPkdUAz4eeVX4ho7zsfQn9Y767Ffg683FE20X...	Nickolaos	Chaikalis	chainicko	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 1.3 KiB]	[BLOB - 1.5 KiB]	[BLOB - 32 B]	1

Τέλος