



ΤΕΙ Αθήνας
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής ΤΕ
MSc στις Τεχνολογίες Υπολογισμού & Δικτύων
Αθήνα 16/1/2017



Βάσεις Δεδομένων και Web-based Εφαρμογές



Σύντομη τεκμηρίωση εργασίας **με θέμα:**

Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυνητικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται απόψεις και links των αγαπημένων ταινιών τους.

Αλεξοπούλου Αγγελική (16005) & Γαλλιάρης Μιχαήλ (16003)

Καθηγητές μαθήματος:
Σκουρλάς Χρήστος * Γαλιώτου Ελενη

Όλα τα αρχεία της εφαρμογής βρίσκονται στο bitbucket:
https://bitbucket.org/Aaggeliki/projdbawba_aa_cnt16005_mg_cnt16003.git

Περιεχόμενα

Εκφώνηση εργασίας.....	3
Ανάλυση, σχεδίαση και μεθοδολογία.....	4
Σχεδίαση και υλοποίηση της βάσης δεδομένων.....	6
Υλοποίηση του Website.....	9
Παρουσίαση του Website.....	13

Εκφώνηση εργασίας

Βάσεις Δεδομένων και web-based Εφαρμογές

Διδάσκοντες: Ε. Γαλιώτου, Χ. Σκουρλάς Ημερομηνία: 17/11/2016

Άσκηση Πρώτη

Σας ανατίθεται από την εταιρεία σας να σχεδιάσετε και να υλοποιήσετε πιλοτικά έναν ιστότοπο. Η εφαρμογή σας, κατά προτίμηση, θα βασίζεται σε βάση δεδομένων, δηλαδή είναι επιθυμητό να υλοποιήσετε μία εφαρμογή δυναμικών ιστοσελίδων. Η εφαρμογή, η βάση δεδομένων και τα εργαλεία υλοποίησης θα είναι δικής σας επιλογής.

Παραδείγματα ιστοτόπων βασιζόμενων σε συστήματα βάσεων δεδομένων που θα μπορούσατε να υλοποιήσετε είναι: βάση ταινιών, βάση τραγουδιών, βάση αυτοκινήτων, βάση προσωπικού εταιρείας.

Μπορείτε να χρησιμοποιήσετε ως εργαλεία για την κατασκευή φορμών-ιστοσελίδων τα εργαλεία που γνωρίζετε καλύτερα, π.χ., java ή jsp ή php ή ...

Ως προϊόν διαχείρισης της βάσης δεδομένων μπορείτε να χρησιμοποιήσετε MySQL ή Oracle ή SQL Server ή PostgreSQL ή ... Εναλλακτικά, μπορείτε να χρησιμοποιήσετε Joomla ή WordPress ή Drupal ή ...

Στην εφαρμογή σας θα υπάρχει μια αρχική σελίδα (login form) όπου θα δίνεται η δυνατότητα στον χρήστη είτε να συνδεθεί είτε να δημιουργήσει ένα νέο λογαριασμό (account) συμπληρώνοντας σχετική φόρμα (registration form).

Αφού συνδεθεί ο χρήστης θα έχει κάποιες δυνατότητες όπως οι παρακάτω:

1. Εισαγωγή (insert) στοιχείων για αντικείμενα της βάσης. Για παράδειγμα, στην περίπτωση βάσης ταινιών, θα μπορεί να κάνει εισαγωγή νέας ταινίας με στοιχεία όπως θεματική κατηγορία, τίτλο, πρωταγωνιστές, φωτογραφίες, url βίντεο από το youtube. Στην περίπτωση βάσης αυτοκινήτων, θα μπορεί να κάνει εισαγωγή ενός νέου αυτοκινήτου με στοιχεία όπως κατηγορία, κατασκευαστή, url βίντεο από το youtube.
2. Εισαγωγή (insert) κατηγοριών στις οποίες ανήκουν οι ταινίες, τα τραγούδια, τα αυτοκίνητα κ.λπ., ανάλογα με τη βάση που θα υλοποιήσετε.
3. Προβολή των εισαχθέντων στοιχείων. Για παράδειγμα, αν διαλέξατε βάση τραγουδιών η προβολή θα μπορούσε να γίνει με διάφορους τρόπους: α) με αναζήτηση δίδοντας το όνομα του τραγουδιστή σε κατάλληλο text box οπότε θα εμφανίζονται όλα τα τραγούδια του τραγουδιστή, β) επιλέγοντας την κατηγορία οπότε θα εμφανίζονται όλα τα τραγούδια που ανήκουν σε αυτήν κ.λπ.
4. Επίσης, δίπλα στην κατηγορία θα πρέπει να εμφανίζεται ο συνολικός αριθμός αντικειμένων, π.χ., ταινιών, τραγουδιών ή αυτοκινήτων, που ανήκουν σε αυτήν. Ο αριθμός αυτός θα πρέπει να διαμορφώνεται από την εκτέλεση trigger
5. Επεξεργασία των εισαχθέντων στοιχείων (δυνατότητα για update, delete).

Η εργασία είναι προτιμότερο να παραδίδεται τμηματικά ανά εβδομάδα, π.χ., σχεδιασμός βάσης δεδομένων, σχεδιασμός User interface, φόρμα

σύνδεσης/εγγραφής χρήστη, φόρμα(ες) εισαγωγής δεδομένων, φόρμα(ες) αναζήτησης. Ως ημερομηνία υποβολής τελικής λύσης ορίζεται η **10/01/2017**
Η εργασία μπορεί να παραδοθεί ατομικά ή από ομάδες των 2 ατόμων.

Ανάλυση, σχεδίαση και μεθοδολογία

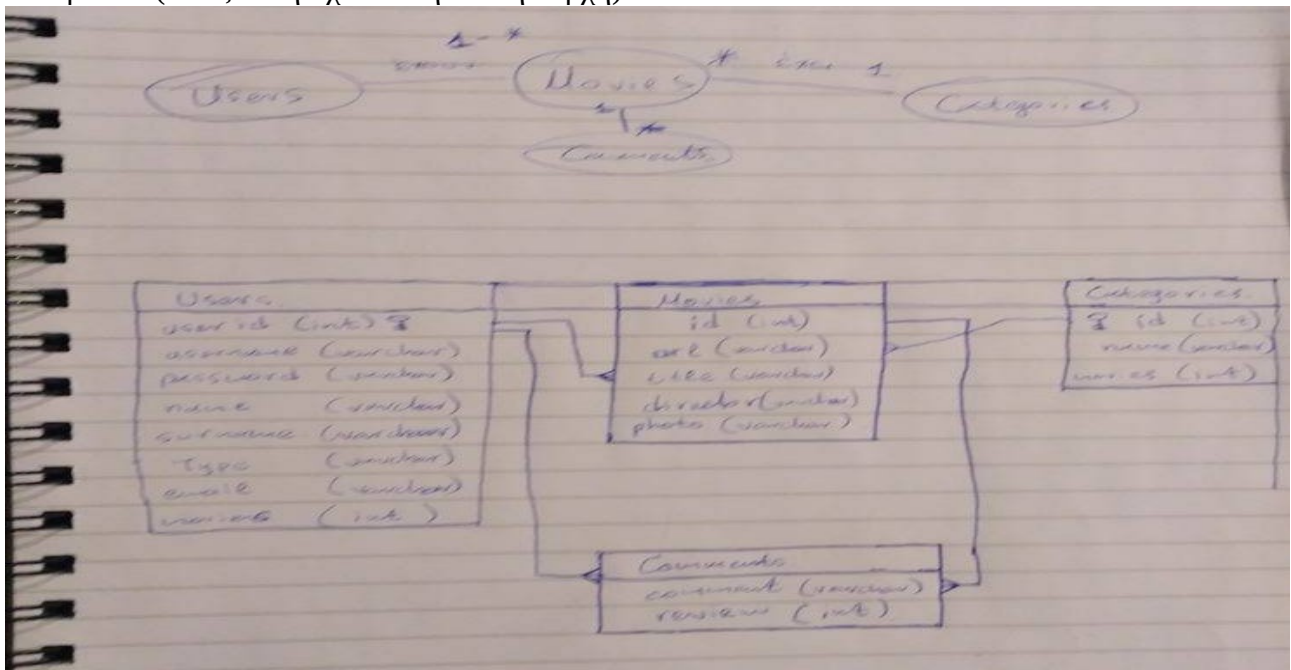
Εισαγωγή

Αρχικά, αυτό που απασχόλησε ήταν να οριστικοποιηθεί το θέμα της εργασίας και να καθοριστούν οι λειτουργίες-δυνατότητες της ιστοσελίδας, ώστε να πληρούνται οι προδιαγραφές της άσκησης. Στην συνέχεια καθορίστηκαν τα “εργαλεία” που ήταν απαραίτητα να χρησιμοποιηθούν, ώστε να υλοποιηθεί η εργασία, τα οποία περιληπτικά είναι τα εξής:

Γλώσσες :	JAVA (JSP & Servlet), HTML, CSS, Javascript, SQL
Πλατφόρμα ανάπτυξης:	Netbeans (IDE 8.2) [maven]
Web container:	Apache Tomcat 8.0.27.0
DBMS:	MySQL (Workbench)
Git:	https://bitbucket.org/

Έπειτα, δημιουργήθηκαν χειρόγραφες σημειώσεις, με όλες τις πιθανές περιπτώσεις χρήσης (με λόγια), όπως επίσης σχεδιάστηκαν ακόμη τα interfaces, τα usecases, η βάση δεδομένων κ.ά της εφαρμογής.

Για παράδειγμα, μια ενδεικτική φωτογραφία από τις χειρόγραφες σημειώσεις που δείχνει την βάση δεδομένων (όπως αυτή σχεδιάστηκε στην αρχή):



Αξίζει να αναφερθεί εισαγωγικά, ότι καθ' όλη τη διάρκεια ανάπτυξης του website, πέρα από την λειτουργικότητα και την σταθερότητα, απασχόλησε εξίσου η **ευχρηστία** και η **φιλικότητα** της σελίδας ως προς τον τελικό χρήστη. π.χ. μεταξύ άλλων, έγινε και η χρήση της javascript για να επιτευχθεί καλύτερη αλληλεπίδραση του χρήστη με το site (έλεγχος λαθών, εμφάνιση μηνυμάτων κ.λπ) και της CSS για να είναι πιο όμορφο το περιβάλλον και πιο φιλικό & γνώσιμο προς τον χρήστη... Ακόμη απασχόλησε σε κάποιο βαθμό και το θέμα της **ασφάλειας** του όλου συστήματος.

Περιπτώσεις χρήσης της ιστοσελίδας

Ο σκοπός του website είναι να δίνει την δυνατότητα σε χρήστες του να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.

Πιο συγκεκριμένα, κάποιος “**επισκέπτης**” έχει τις εξής δυνατότητες μέσα από το website:

- ✓ Να δει τις τελευταίες χρονικά ταινίες που προστέθηκαν
- ✓ Να αναζητήσει ταινίες με βάση κάποια κατηγορία
- ✓ Να αναζητήσει ταινίες με βάση λέξεις κλειδιά (που να σχετίζονται με το τίτλο ή τον σκηνοθέτη)
- ✓ Να παρακολουθήσει κάποια ταινία
- ✓ Να δει πληροφορίες σχετικά με την εργασία
- ✓ Να κάνει Login
- ✓ Να κάνει εγγραφή

Επίσης, κάποιος **εγγεγραμμένος χρήστης**, εκτός από τα παραπάνω, έχει επιπλέον τις δυνατότητες:

- ✓ Να διαχειρίζεται το προφίλ του (Edit, Διαγραφή χρήστη)
- ✓ Να προσθέτει, να τροποποιεί και να αφαιρεί ταινίες (εφόσον τις έχει προσθέσει ο ίδιος)
- ✓ Να κάνει Logout

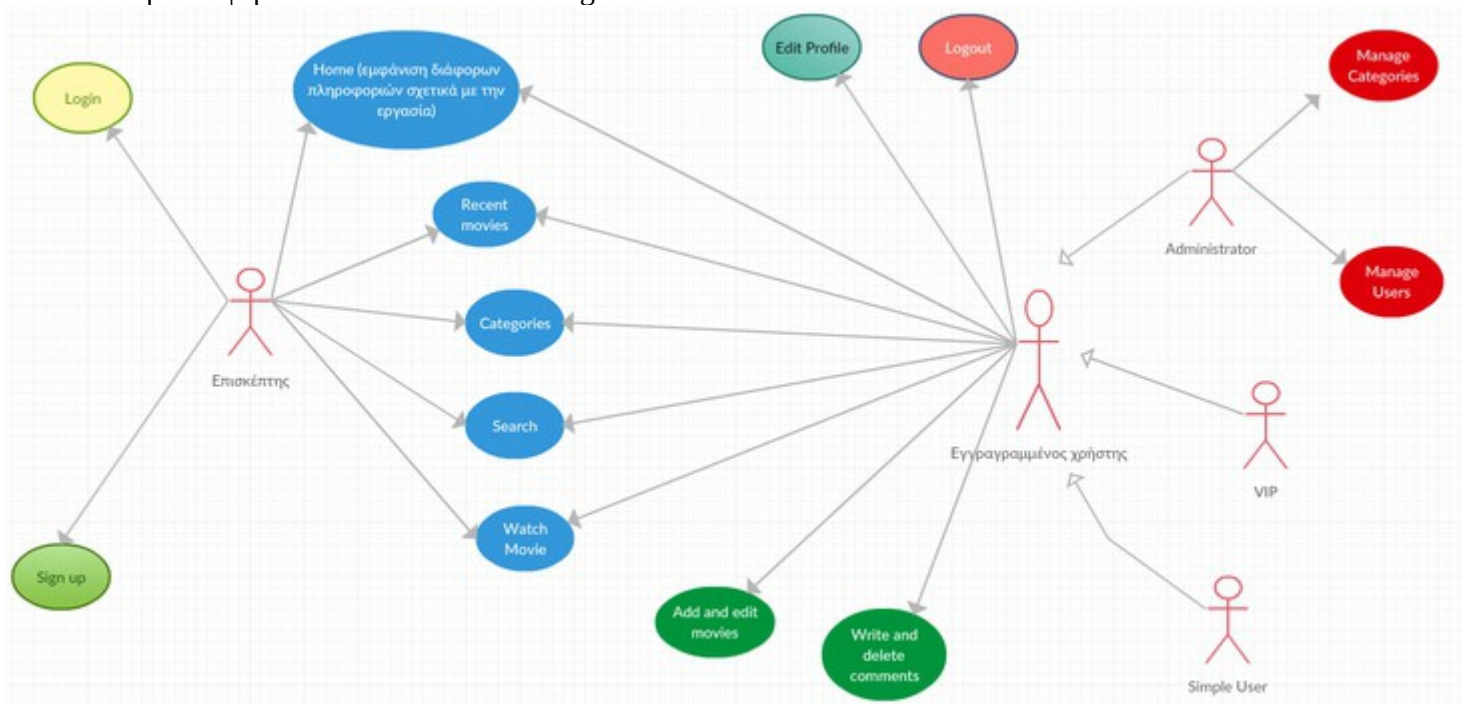
Στην περίπτωση που κάποιος χρήστης είναι τύπου **administrator**, τότε έχει ακόμη 2 δυνατότητες:

- ✓ Να προσθέτει και να αφαιρεί κατηγορίες ταινιών
- ✓ Να διαχειρίζεται όλους τους χρήστες του Συστήματος

Οι εγγεγραμμένοι χρήστες χωρίζονται με βάση το τύπο τους, σε τρεις κατηγορίες:

- Απλοί
- VIP (Έχουν προσφέρει χρήματα για την συντήρηση της ιστοσελίδας... :))
- Administrators (Διαχειριστές της σελίδας)

Τα προαναφερθέντα σε ένα use case diagram:



Σχεδίαση και υλοποίηση της βάσης δεδομένων

Για τις ανάγκες της εφαρμογής, δημιουργήθηκε μια βάση δεδομένων (moviesdb) με 4 λογικά συσχετισμένους πίνακες (users, movies, categories, comments).

Οι σχέσεις μεταξύ τους:

- Ένας User μπορεί να έχει πολλές ταινίες και μια ταινία να ανήκει σε ένα μόνο user.
- Μια ταινία ανήκει σε μια ή καμία κατηγορία ταινίας (Ενδεικτική λύση, όχι τελείως σωστή...) και σε μια κατηγορία μπορούν να ανήκουν πολλές ταινίες.
- Μια ταινία μπορεί να έχει πολλά σχόλια και ένα σχόλιο ανήκει σε μια μόνο ταινία
- Ένας χρήστης μπορεί να έχει πολλά σχόλια και ένα σχόλιο ανήκει σε ένα μόνο χρήστη.

```
MariaDB [moviesdb]> show tables ;
+-----+
| Tables_in_moviesdb |
+-----+
| categories          |
| comments            |
| movies              |
| users               |
+-----+
4 rows in set (0.00 sec)
```

Ο πίνακας **users** αποθηκεύει όλες τις πληροφορίες που αφορούν κάποιο χρήστη.

Δηλαδή το userID (Πρωτεύον κλειδί, αύξων αριθμός), το username (Υποψ. κλειδί), το password, το name, το surname, το type (Τύπος χρήστη [U,V,A]), το email, τα totalmovies (σύνολο από ταινίες που έχει καταχωρήσει [γεμίζει αυτόματα από trigger]) και το isDeleted (για “λογική” διαγραφή...)

```
MariaDB [moviesdb]> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userID | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(45) | NO | UNI | NULL | |
| password | varchar(45) | NO | | NULL | |
| name | varchar(45) | YES | | NULL | |
| surname | varchar(45) | YES | | NULL | |
| type | varchar(1) | NO | | U | |
| email | varchar(45) | NO | | NULL | |
| totalmovies | int(11) | NO | | 0 | |
| isDeleted | bit(1) | NO | | b'0' | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

Ο πίνακας **categories** αποθηκεύει όλες τις πληροφορίες που αφορούν κάποια κατηγορία ταινίας.

Δηλαδή το id (Πρωτεύον κλειδί, αύξων αριθμός), το name και τα movies (σύνολο από ταινίες που ανήκουν στην συγκεκριμένη κάθε φορά κατηγορία ταινίας [γεμίζει αυτόματα από trigger])

```
MariaDB [moviesdb]> desc categories;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(45) | NO | UNI | NULL | |
| movies | int(11) | YES | | 0 | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Ο πίνακας **movies** αποθηκεύει ενδεικτικά κάποιες πληροφορίες που αφορούν μια ταινία.

Δηλαδή το id (Πρωτεύον κλειδί, αύξων αριθμός), το url (Υποψ. κλειδί), το title, το director, το photo (path φωτογραφίας που δεν χρησιμοποιήθηκε τελικά), το userID (Ξένο κλειδί από τον πίνακα users), το Category (Ξένο κλειδί από τον πίνακα categories), τα totalComments (σύνολο από σχόλια που αφορούν την ταινία [γεμίζει αυτόματα από trigger]) και το cDate (Ημερ. & ώρα καταχώρησης).

```
MariaDB [moviesdb]> desc movies;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRI | NULL | auto_increment |
| url   | varchar(150) | NO | UNI | NULL |
| title | varchar(45) | NO | | NULL |
| director | varchar(45) | YES | | NULL |
| photo | varchar(45) | YES | | NULL |
| userID | int(11) | NO | MUL | NULL |
| Category | int(11) | YES | MUL | NULL |
| totalcomments | int(11) | NO | | 0 |
| cDate | datetime | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

Ο πίνακας **comments** αποθηκεύει ενδεικτικά κάποιες πληροφορίες που αφορούν μια ταινία.

Δηλαδή το userID (Πρωτεύον κλειδί μαζί με το movieID), το movieID (Πρωτεύον κλειδί μαζί με το userID), το review (αριθμός από το 1 μέχρι το 5 για βαθμολογία), το comment (κείμενο σαν κριτική), και το cDate (Ημερομηνία και ώρα καταχώρησης).

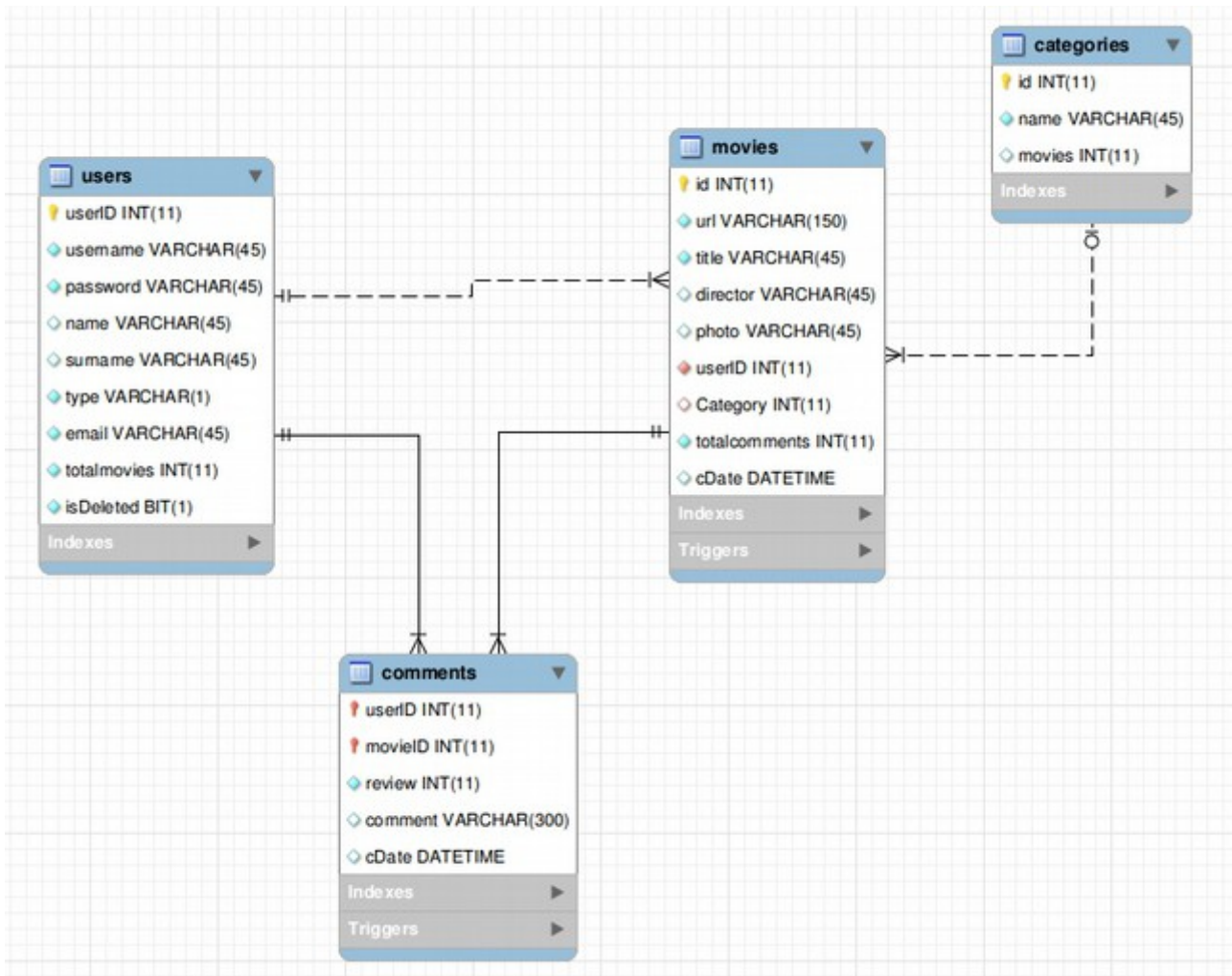
```
MariaDB [moviesdb]> desc comments;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userID | int(11) | NO | PRI | NULL |
| movieID | int(11) | NO | PRI | NULL |
| review | int(11) | NO | | 0 |
| comment | varchar(300) | YES | | NULL |
| cDate | datetime | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ενδεικτικά, μία μόνο από τις υλοποιήσεις των triggers:

```
CREATE TRIGGER moviesdb.insert_movie AFTER INSERT ON movies FOR EACH ROW
UPDATE users, categories
SET users.totalmovies = users.totalmovies + 1,
categories.movies = categories.movies + 1
WHERE users.userID = new.userID AND categories.id = new.Category
```


Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: “Βάσεις Δεδομένων και Web-based Εφαρμογές”

Ένα διάγραμμα της βάσης δεδομένων (που φτιάχτηκε αυτόματα με το workbench):



Να αναφερθεί ότι τα password στον πίνακα users είναι κρυπτογραφημένα με τον αλγόριθμο MD5 και έχουν 32 χαρακτήρες ώστε αν διαρρεύσει το περιεχόμενο της βάσης να είναι δύσκολο να ανακτηθούν οι κωδικοί και άρα με αυτό το τρόπο υπάρχει μεγαλύτερη ασφάλεια στο σύστημα.

```
MariaDB [moviesdb]> SELECT * FROM users ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| userID | username | password | name | surname | type | email | totalmovies | isDeleted |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | aggeliki | 28bcbaa890f1f1d3d416b3d04f0b05c0 | Aggeliki | Alexopoulou | A | agg@tei.gr | 5 |  |
| 2 | michalis | 4fea6c9fb84ece455b16e47bf1c12435 | Michalis | Galliakis | V | mike@yahoo.gr | 4 |  |
| 4 | useruser | 5cc32e366c87c4cb49e4309b75f57d64 | user | user | U | user@user.gr | 2 |  |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```


Υλοποίηση του Website

Οργάνωση αρχείων

Οργάνωση με βάση το πρότυπο MVC:

- ✓ View - JSP (Folder: Web pages)
administration.jsp, categories.jsp, editMovie.jsp, editProfile.jsp, editUser.jsp, index.jsp, login.jsp, logout.jsp, manageCategories.jsp, manageUsers.jsp, newMovie.jsp, playMovie.jsp, profile.jsp, pronunciation.jsp, recentMovies.jsp, search.jsp, signUp.jsp, viewMyMovies.jsp
- ✓ Controllers - Servlets (package: dbawba.controllers)
DeleteCategory.java, DeleteComment.java, DeleteMovie.java, DeleteUser.java, Login.java, NewCategory.java, NewComment.java, NewMovie.java, NewUser.java, UpdateMovie.java, UpdateUser.java
- ✓ Model – Java class (package: dbawba.data)
Category.java, Comment.java, Movie.java, User.java
- ◆ Package: dbawba.tools
CaptchaGenerator.java (Για τη δημιουργία captcha), Debug.java (Εργαλεία για debug), Globals.java (Χρήσιμοι static μέθοδοι), ManageDBConnection.java (Για τη διαχείριση των database connections)

Folders:

- AuxiliaryFiles (Βοηθητικά jsp αρχεία) [displayInfo.jsp, error.jsp]
- CSS (Αρχείο css για το design της σελίδας) [layout.css]
- CommonFiles (Αρχεία jsp που γίνονται include από άλλα jsp αρχεία) [footer.jsp, header.jsp, menu.jsp, viewTableWithMovies.jsp]
- Images (Βρίσκονται όλες οι φωτογραφίες που χρησιμοποιούνται)
- JS (Αρχείο javascript για validate κυρίως) [form-validation.js]
- OtherFiles (Το pdf με την εκφώνηση της άσκησης) [Master_Assignment_1_2016_17.pdf]

Χρήση βάσεων δεδομένων μέσα από το Website

Ενδεικτικά θα παρουσιαστούν κάποια screenshots από τη χρήση της database μέσα από τη Java.

Insert:

```
public boolean insertNewMovieToDB()
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String insert = "INSERT INTO movies " +
        "(url," +
        "title," +
        "director," +
        "photo," +
        "userID," +
        "cDate," +
        "Category)" +
        "VALUES" +
        "(" + url + "," +
        " " + title + "," +
        " " + director + "," +
        " " + photo + "," +
        " " + userID + "," +
        " " + new java.sql.Timestamp(cDate.getTime()) + "," +
        " " + Category + ");" ;

    boolean result = mdbc.executeUpdate(insert, true);
    mdbc.closeDBConnection();

    return result ;
}
```

Update:

```
public boolean updateMovieToDB()
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String update = "UPDATE movies SET " +
        "title = '" + title + "'," +
        " director = '" + director + "'," +
        " photo = '" + photo + "'," +
        " Category = '" + Category + "'" +
        " WHERE id = " + id + " ;";

    boolean result = mdbc.executeUpdate(update, true);
    mdbc.closeDBConnection();

    return result;
}
```

Delete:

```
public boolean deleteMovieFromDB()
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String delete = "DELETE FROM comments WHERE movieID = " + id + " ;";

    mdbc.executeUpdate(delete, true);

    delete = "DELETE FROM movies WHERE id = " + id + " ;";

    boolean result = mdbc.executeUpdate(delete, true);
    mdbc.closeDBConnection();

    return result;
}

//Static methods:
public static boolean delAnyoneMovieFromDB(int mID)
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String delete = "DELETE FROM comments WHERE movieID = " + mID + " ;";

    mdbc.executeUpdate(delete, true);

    delete = "DELETE FROM movies WHERE id = " + mID + " ;";

    boolean result = mdbc.executeUpdate(delete, true);
    mdbc.closeDBConnection();

    return result;
}
```

Select:

```
public static boolean isThereTheURL(String mUrl)
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String select = "SELECT id FROM movies WHERE url = '" + mUrl + "'" ;";
    ResultSet result = mdbc.executeQuery(select, true);

    boolean isEmpty = false ;
    try {
        isEmpty = result.next() ;
    } catch (SQLException ex) {
        Debug.printStackTrace(ex);
    }
    mdbc.closeDBConnection();
    return isEmpty ;
}

public static boolean isThereTheID(int mID)
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String select = "SELECT id FROM movies WHERE id = '" + mID + "'" ;";
    ResultSet result = mdbc.executeQuery(select, true);

    boolean isEmpty = false ;
    try {
        isEmpty = result.next() ;
    } catch (SQLException ex) {
        Debug.printStackTrace(ex);
    }
    mdbc.closeDBConnection();
    return isEmpty ;
}
```

Το “χτίσιμο” της συνθήκης where για το search, όπου με βάση αυτό που έχει συμπληρώσει ο χρήστης, εκτελείται το ανάλογο Select στη βάση:

Search:



Only Title: ☐ | All: ☒ | Only Director: ☐

```
String sqlWhereCondition = "" ;
if (!search.equals(""))
{
    String[] arrWords = search.split("\\s+");
    switch(type){
        case "A":
            sqlWhereCondition = " WHERE (Director like LOWER('%" + Globals.fixSQLString(arrWords[0]).toLowerCase() + "%') "+
                                " OR Title like LOWER('%" + Globals.fixSQLString(arrWords[0]).toLowerCase() + "%')) " ;
            for (int i=1;i<arrWords.length;i++)
                sqlWhereCondition += " OR (Director like LOWER('%" + Globals.fixSQLString(arrWords[i]).toLowerCase() + "%') "+
                                    " OR Title like LOWER('%" + Globals.fixSQLString(arrWords[i]).toLowerCase() + "%')) " ;
            break ;
        case "T":
            sqlWhereCondition = " WHERE Title like LOWER('%" + Globals.fixSQLString(arrWords[0]).toLowerCase() + "%') " ;
            for (int i=1;i<arrWords.length;i++)
                sqlWhereCondition += " OR Title like LOWER('%" + Globals.fixSQLString(arrWords[i]).toLowerCase() + "%') " ;
            break ;
        case "D":
            sqlWhereCondition = " WHERE Director like LOWER('%" + Globals.fixSQLString(arrWords[0]).toLowerCase() + "%') " ;
            for (int i=1;i<arrWords.length;i++)
                sqlWhereCondition += " OR Director like LOWER('%" + Globals.fixSQLString(arrWords[i]).toLowerCase() + "%') " ;
            break ;
    }
}
```

Η μέθοδος “selectMoviesFromDB” παίρνει σαν παραμέτρους, τον αριθμό των εγγραφών που θα επιστρέψει, την συνθήκη “where” και τον αριθμό της σελίδας ώστε να υπολογιστεί το offset. Διακρίνεται ότι έχει υπάρξει η ανάγκη να γίνουν Select μέσα σε Select...

```
public static ArrayList<Movie> selectMoviesFromDB(int limit,String sqlWhereCondition,int page)
{
    ManageDBConnection mdbc = new ManageDBConnection() ;

    String select = "SELECT id, url, title, director, " +
                    "photo, userID, Category, totalcomments, cDate, " +
                    "(SELECT username FROM users WHERE users.userID = movies.userID) AS userUsername, " +
                    "(SELECT name FROM categories WHERE categories.id = movies.Category) AS CategoryName " +
                    "FROM movies "+ sqlWhereCondition + " " +
                    " ORDER BY cDate DESC LIMIT "+ limit + " OFFSET "+(page*limit)+" " ;

    ResultSet resultTable = mdbc.executeQuery(select, true);

    ArrayList<Movie> resultWithUsers=new ArrayList<>();

    try {
        while(resultTable.next()) {
            Movie newMovie = new Movie();

            newMovie.setId(resultTable.getInt("id"));
            newMovie.setUrl(resultTable.getString("url"));
            ...
        }
    }
```


Μια άλλη υλοποίηση, με Join αντί για Select μέσα σε Select:

```
public boolean loadMovieFromDB(int mID)
{
    ManageDBConnection mdhc = new ManageDBConnection() ;

    String select = "SELECT m.id, m.url, m.title, m.director, " +
        "m.photo, m.userID, m.Category, m.totalcomments, m.cDate, " +
        "u.username AS userUsername, " +
        "c.name AS CategoryName " +
        "FROM movies m "+
        "INNER JOIN users u ON u.userID = m.userID " +
        "INNER JOIN categories c ON c.id = m.Category " +
        "WHERE m.id = " + mID + " ;" ;

    ResultSet resultTable = mdhc.executeQuery(select, true);

    try {
        if(resultTable.next()) {
            id = resultTable.getInt("id");
            url = resultTable.getString("url");
            title = resultTable.getString("title");
            director = resultTable.getString("director");
        }
    }
}
```

Τέλος, θα παρουσιαστεί ενδεικτικά και η υλοποίηση της λογικής διαγραφής των Users.

```
public boolean deleteUserFromDB()
{
    ManageDBConnection mdhc = new ManageDBConnection() ;

    String delete = "UPDATE users SET isDeleted = 1 WHERE userID = " + userID + " ;";

    boolean result = mdhc.executeUpdate(delete, true);
    mdhc.closeDBConnection();

    return result;
}

//Static methods:
public static boolean delAnyoneUserFromDB(int uID)
{
    ManageDBConnection mdhc = new ManageDBConnection() ;

    String delete = "UPDATE users SET isDeleted = 1 WHERE userID = " + uID + " ;";

    boolean result =mdhc.executeUpdate(delete, true);
    mdhc.closeDBConnection();

    return result;
}
```

Παρουσίαση του Website

Αρχική σελίδα:



Σελίδα που δείχνει την εκφώνηση της άσκησης:



Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: "Βάσεις Δεδομένων και Web-based Εφαρμογές"

Σελίδα για το login:

(Υπάρχει επιλογή "να με θυμάσαι", ώστε με χρήση cookies να θυμάται η σελίδα το login μετά το κλείσιμο του browser)

Σύνδεση :
Username: michalis
Password: *****
Να με θυμάσαι: ☒

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνών (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυναμικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Εργασία για το μάθημα :
"Βάσεις Δεδομένων και Web-based Εφαρμογές".
Φοιτητής:
Αγγελική Αλεξοπούλου

Σελίδα για την εμφάνιση του profile ενός χρήστη:

Username:	michallis
Name:	Michalis
Surname:	Galliakis
Email:	mike@yahoo.gr
Total movies:	4

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνών (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυναμικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται απόψεις και links των αγαπημένων ταινιών τους.
Εργασία για το μάθημα :
"Βάσεις Δεδομένων και Web-based Εφαρμογές".
Φοιτητής:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιιάκης
1/2017

Σελίδα για τη τροποποίηση του profile ενός χρήστη:

Όνομα χρήστη: michalis
Κωδικός:
Επιβεβαίωση κωδικού:
Όνομα: Michalis
Επίθετο: Galliakis
Email: mike@yahoo.gr

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθήνας (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Εργασία για το μάθημα : "Βάσεις Δεδομένων και Web-based Εφαρμογές".

Σελίδα για την εμφάνιση όλων των ταινιών που έχει καταχωρήσει ο συνδεδεμένος χρήστης:

My movies [4]:

Rows: 5

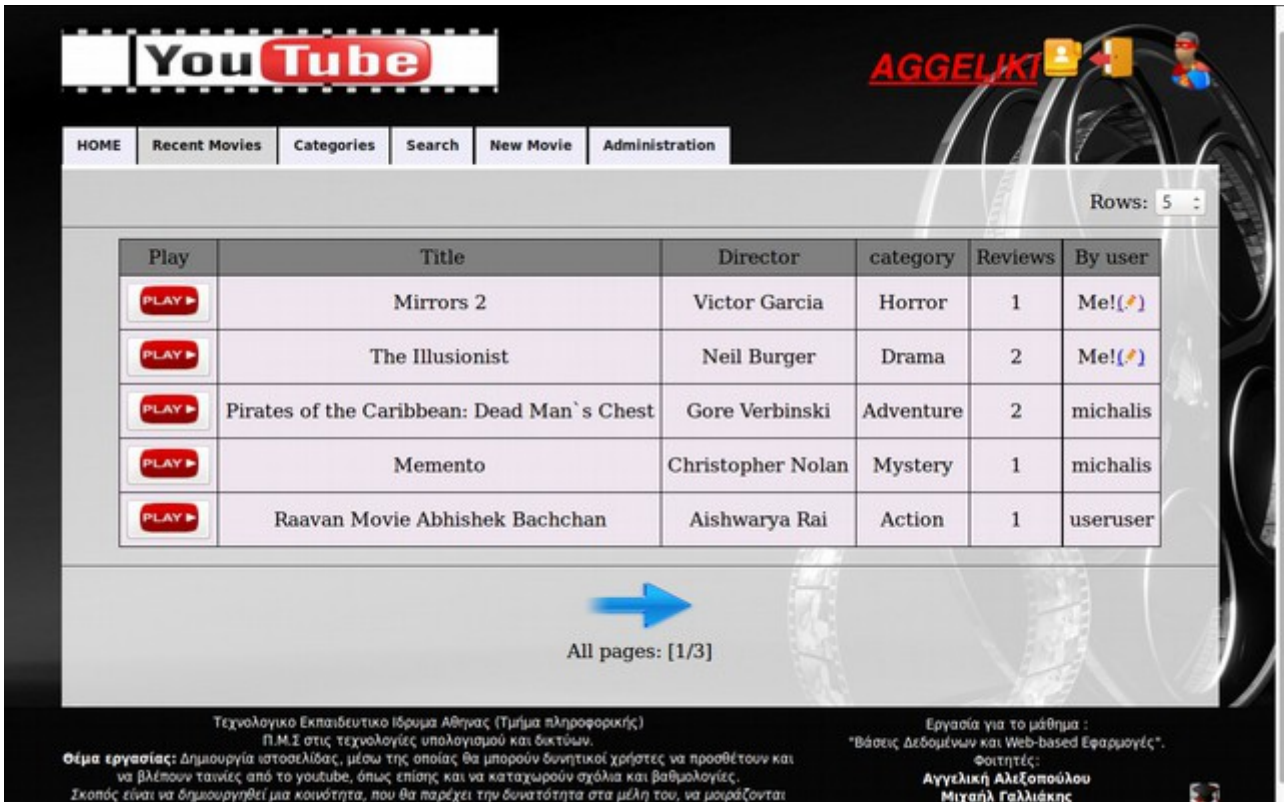
Play	Title	Director	category	Reviews	By user
	Pirates of the Caribbean: Dead Man`s Chest	Gore Verbinski	Adventure	2	Me! (👍)
	Memento	Christopher Nolan	Mystery	1	Me! (👍)
	Anna 2013 Full English `Movie		Comedy	3	Me! (👍)
	Hachi A Dogs Tale (2009)		Comedy	2	Me! (👍)

All pages: [1/1]

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθήνας (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Εργασία για το μάθημα : "Βάσεις Δεδομένων και Web-based Εφαρμογές".
Φοιτητές: Αγγελική Αλεξοπούλου, Μιχαήλ Γαλλιιάκης

Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: "Βάσεις Δεδομένων και Web-based Εφαρμογές"

Σελίδα που εμφανίζει τις τελευταίες χρονικά ταινίες που προστέθηκαν στο website:



Rows: 5 :

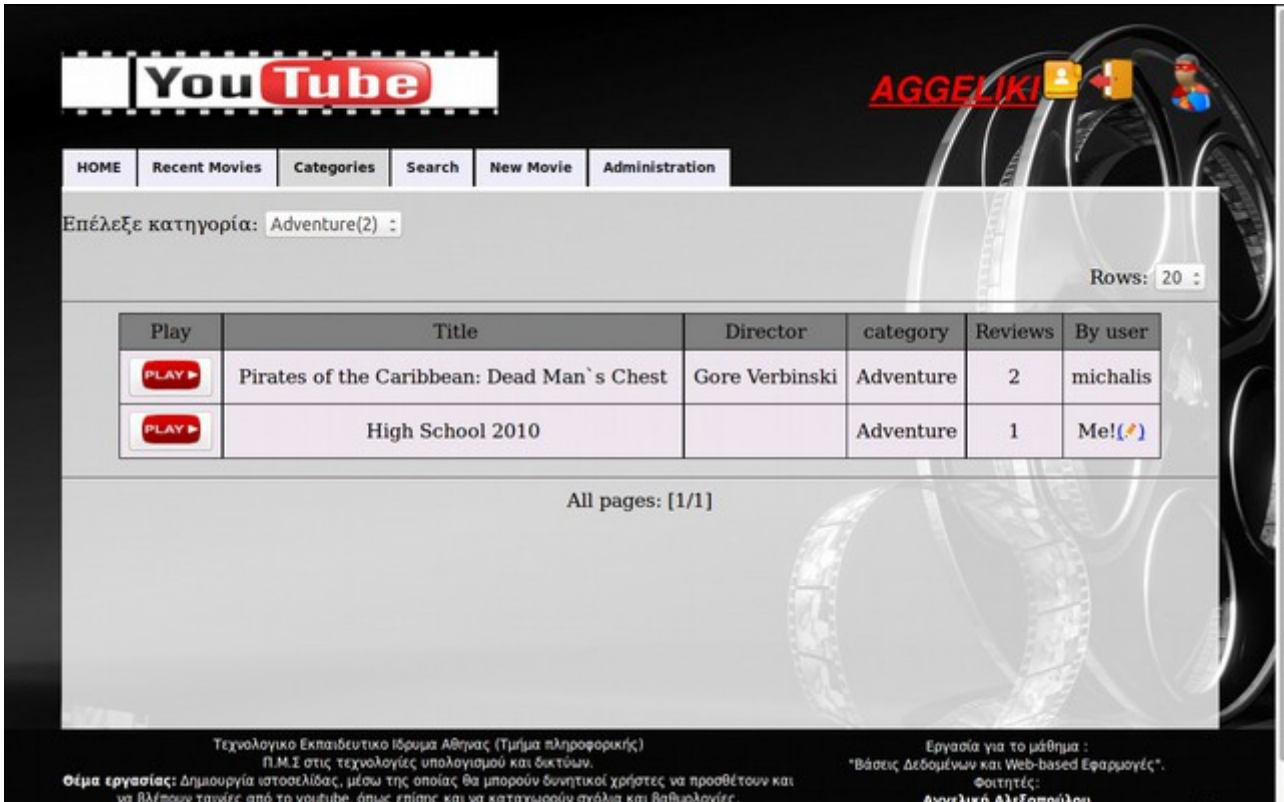
Play	Title	Director	category	Reviews	By user
	Mirrors 2	Victor Garcia	Horror	1	Me! [🔥]
	The Illusionist	Neil Burger	Drama	2	Me! [🔥]
	Pirates of the Caribbean: Dead Man`s Chest	Gore Verbinski	Adventure	2	michalis
	Memento	Christopher Nolan	Mystery	1	michalis
	Raavan Movie Abhishek Bachchan	Aishwarya Rai	Action	1	useruser

All pages: [1/3]

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνas (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυνhτικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται

Εργασία για το μάθημα :
"Βάσεις Δεδομένων και Web-based Εφαρμογές".
Φοιτητές:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιάκης

Σελίδα που εμφανίζει τις ταινίες μιας συγκεκριμένης κατηγορίας:
(Εμφανίζεται και το πλήθος των ταινιών της ίδιας κατηγορίας με βάση το trigger)



HOME Recent Movies Categories Search New Movie Administration

Επέλεξε κατηγορία: Adventure(2) :

Rows: 20 :

Play	Title	Director	category	Reviews	By user
	Pirates of the Caribbean: Dead Man`s Chest	Gore Verbinski	Adventure	2	michalis
	High School 2010		Adventure	1	Me! [🔥]

All pages: [1/1]

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνas (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυνhτικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται

Εργασία για το μάθημα :
"Βάσεις Δεδομένων και Web-based Εφαρμογές".
Φοιτητές:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιάκης

Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: “Βάσεις Δεδομένων και Web-based Εφαρμογές”

Σελίδα που δίνει την δυνατότητα στο χρήστη να κάνει αναζήτηση ταινίας:
(Γίνεται match αν έστω και ένα “λεκτικό” στα κριτήρια υπάρχει σε κάποια εγγραφή...)

USERUSER

HOME Recent Movies Categories Search New Movie Administration

Search: pirates the

Only Title: | All: | Only Director:

Rows: 10

Play	Title	Director	category	Reviews	By user
	The Illusionist	Neil Burger	Drama	2	aggeliki
	Pirates of the Caribbean: Dead Man`s Chest	Gore Verbinski	Adventure	2	michalis
	The Croods		Action	1	Me[!]

All pages: [1/1]

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνas (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυναμικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.

Εργασία για το μάθημα :
“Βάσεις Δεδομένων και Web-based Εφαρμογές”.
Φοιτητής:
Αγγελική Αλεξοπούλου

Σελίδα για να προσθέτει κάποιος εγγεγραμμένος χρήστης κάποια ταινία:

MICHALIS

HOME Recent Movies Categories Search New Movie Administration

Καταχώρηση Νέας ταινίας:

Δώσε Url:

Url

Δώσε το όνομα της ταινίας:

Title

Δώσε το όνομα του σκηνοθέτη:

Director

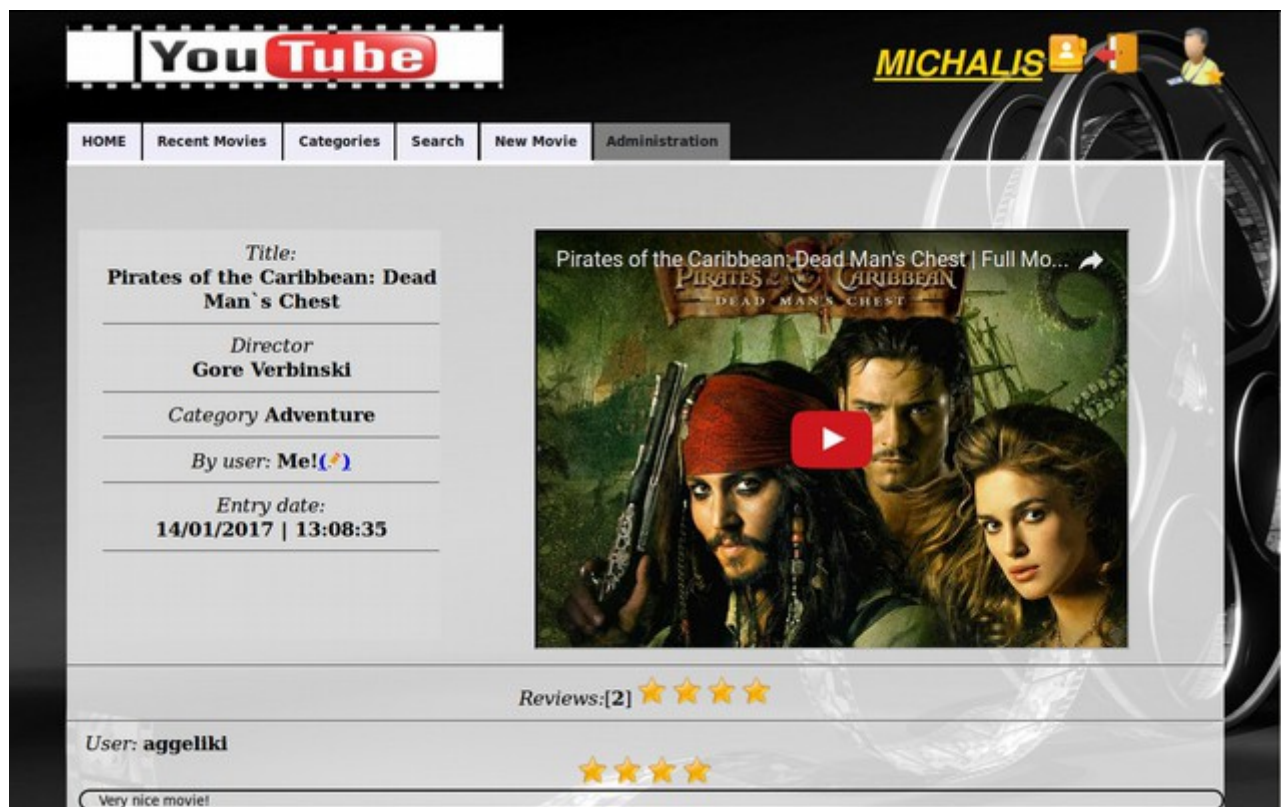
Επέλεξε κατηγορία

Mystery

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθηνas (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυναμικοί χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται

Εργασία για το μάθημα :
“Βάσεις Δεδομένων και Web-based Εφαρμογές”.
Φοιτητής:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιάρης

Σελίδα για την προβολή κάποια ταινίας:



Σελίδα για την αλλαγή των στοιχείων κάποια ταινίας:



Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: “Βάσεις Δεδομένων και Web-based Εφαρμογές”

Σελίδα για την διαχείριση της ιστοσελίδας: (Πρόσβαση μόνο από Administrators:)



Σελίδα για την διαχείριση των κατηγοριών:



Σελίδα για την διαχείριση των χρηστών:

Username	Name	Surname	Type	Total movies	Edit User	Delete User
aggeliki	Aggeliki	Alexopoulou		5		
michalis	Michalis	Galliakis		4		
useruser	user	user		2		

Τεχνολογικό Εκπαιδευτικό Ιδρυμα Αθήνας (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυνάμικοι χρήστες να προσθέτουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται απόψεις και links των αγαπημένων ταινιών τους.

Εργασία για το μάθημα :
“Βάσεις Δεδομένων και Web-based Εφαρμογές”.
Φοιτητές:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιιάκης
1/2017

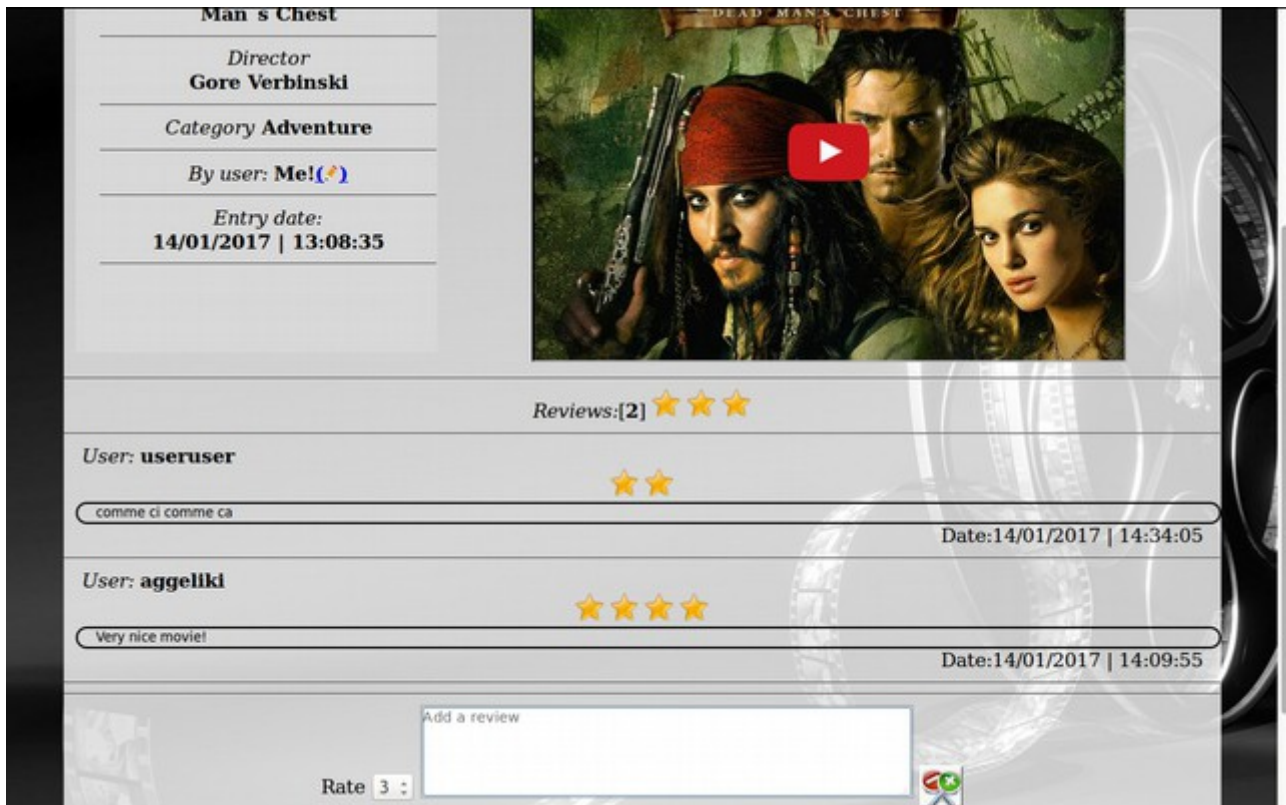
Σελίδα για το edit ενός χρήστη από κάποιον administrator:

(Ο administrator, μεταξύ άλλων, μπορεί και να αλλάξει το τύπο κάπου χρήστη)

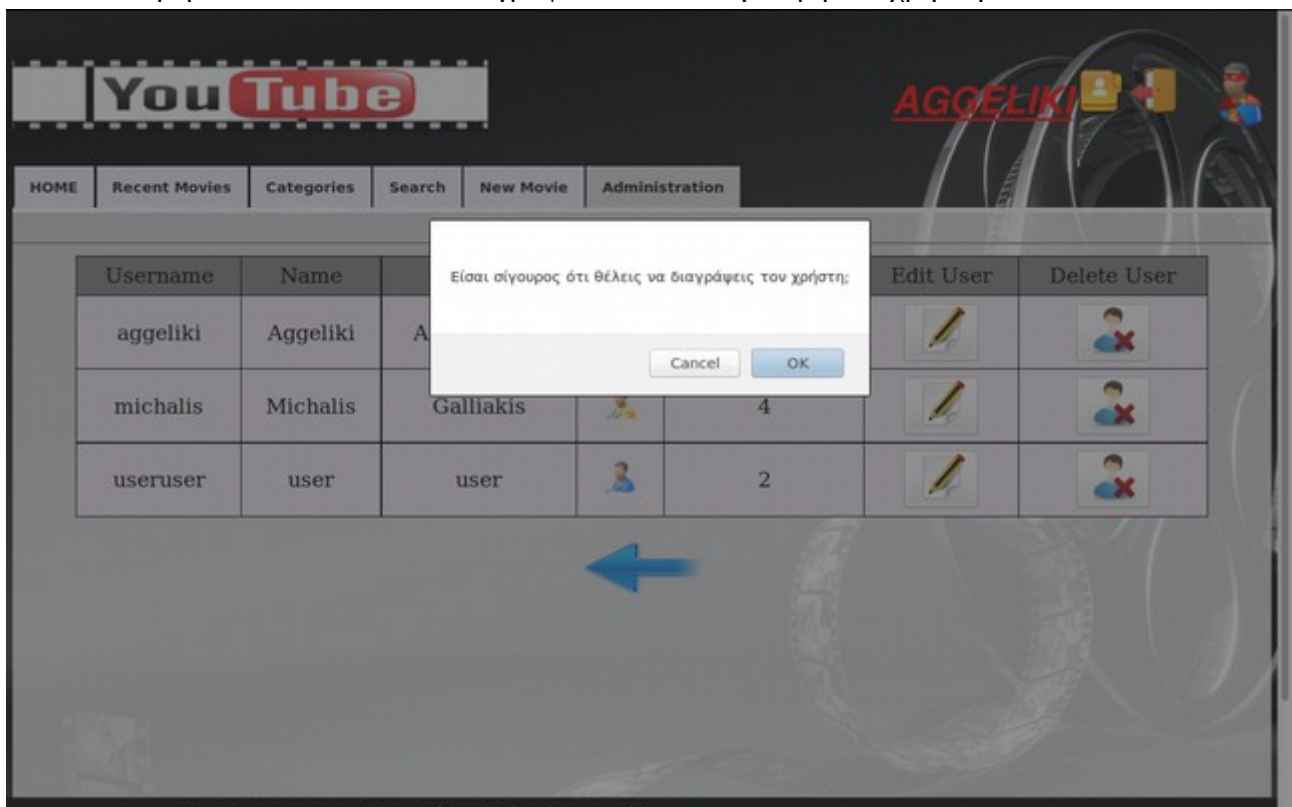
Όνομα χρήστη: useruser
Κωδικός:
.....
Επιβεβαίωση κωδικού:
.....
Όνομα:
user
Επίθετο:
user
Email:
user@user.gr
User: ☒ *** VIP: ☐ *** Admin: ☐

Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: “Βάσεις Δεδομένων και Web-based Εφαρμογές”

Ενδεικτικά, κάποια σχόλια και ο τρόπος εισαγωγής σχόλιου, μιας ταινίας από τη σελίδα προβολής:



Ενδεικτικά, ένα μήνυμα που εμφανίζεται σε κάποιον administrator (με την βοήθεια της javascript) ώστε να επιβεβαιώσει ότι θέλει να διαγράψει κάποιον συγκεκριμένο χρήστη:



Σύντομη τεκμηρίωση 1^{ης} εργασίας για το μάθημα: “Βάσεις Δεδομένων και Web-based Εφαρμογές”

Σελίδα που επιτρέπει σε κάποιον “επισκέπτη” να κάνει εγγραφή:

(Χρησιμοποιείται captcha για την αναγνώριση κάποιου φυσικού προσώπου ότι δεν είναι robot)

Δημιουργία Νέου λογαριασμού χρήστη:

Όνομα χρήστη:
Username

Κωδικός:
Password

Επιβεβαίωση κωδικού:
Confirm Password

Όνομα:
Name

Επίθετο:
Surname

Email:
Email

9 2 3 7 6

Already registered ?
Login

Τεχνολογία Εκπαιδευτικό Ίδρυμα Αθηνών (Τμήμα πληροφορικής)
Π.Μ.Σ στις τεχνολογίες υπολογισμού και δικτύων.
Θέμα εργασίας: Δημιουργία ιστοσελίδας, μέσω της οποίας θα μπορούν δυναμικοί χρήστες να προσέχουν και να βλέπουν ταινίες από το youtube, όπως επίσης και να καταχωρούν σχόλια και βαθμολογίες.
Σκοπός είναι να δημιουργηθεί μια κοινότητα, που θα παρέχει την δυνατότητα στα μέλη του, να μοιράζονται απόψεις και links των αγαπημένων ταινιών τους.

Εργασία για το μάθημα :
“Βάσεις Δεδομένων και Web-based Εφαρμογές”.
Φοιτητές:
Αγγελική Αλεξοπούλου
Μιχαήλ Γαλλιάρης
200317

Ενδεικτικά κάποια μηνύματα βοήθειας προς τους χρήστες:

Σύνδεση :

Μη έγκυρο username ή password!

Username:

Password:

Να με θυμάσαι: ☐

Login

Don't log in

Δημιουργία Νέου λογαριασμού χρήστη:

Το username υπάρχει ήδη! Αλλάξτε το.

Όνομα χρήστη:
michalis

Κωδικός:

Δημιουργία Νέου λογαριασμού χρήστη:

Δεν συμπληρώσατε σωστό captcha.

Όνομα χρήστη:

Ακόμη, ένα ενδεικτικό μήνυμα ενημέρωσης (με μπλε χρώμα) του χρήστη:

