Τμήμα Μηχανικών Πληροφορικής Τ.Ε.Ι Δυτικής Ελλάδος

Πτυχιακή Εργασία με Θέμα:

# Σχεδιασμός και ανάπτυξη εφαρμογής σε κινητά για την βιβλιοθήκη του τμήματος

Εκπωνητές :
Λαμπρακοπούλου Θεοδώρα
Νιώτη Χρυσάνθη

Επιβλέπων Καθηγητής:
Χριστοδούλου Σωτήρης

12 Μαΐου 2016

# Περίληψη

Σκοπός της πτυχιακής εργασίας είναι η δημιουργία μιας webεφαρμογής για τη λειτουργία μίας βιβλιοθήκης. Η εφαρμογή υποστηρίζει την αποθήκευση και παρουσίαση βιβλίων που έχουν καταλογογραφηθεί , προαιρετικά μαζί με το εξώφυλλό τους.Υποστηρίζει την εγγραφή χρηστών (τόσο από το διαχειριστή όσο και από μόνοι τους) και ο κάθε χρήστης μπορεί να δημιουργεί μία λίστα αγαπημένων βιβλίων, καθώς και να βαθμολογεί τα υπάρχοντα βιβλία.Επίσης μπορεί να έχει πρόσβαση στο ιστορικό των δανεισμών του για να βλέπει βιβλία που δανείστηκε στο παρελθόν.Μπορεί επιπλέον να κάνει κράτηση σε κάποιο βιβλίο για να το δανειστεί μόλις θα είναι διαθέσιμο.Ο διαχειριστής διαχειρίζεται τους δανεισμούς, εγκρίνει και απορρίπτει κρατήσεις, και σημειώνει τις επιστροφές των βιβλίων.Επίσης το σύστημα παρέχει προς όλους τους επισκέπτες (είτε ανώνυμοι είτε εγγεγραμμένοι χρήστες) στατιστικά σχετικά με τα τελευταία καταχωρισμένα βιβλία, τα πιο διάσιμα στις λίστες αγαπημένων, καθώς και με τα βιβλία με τις υψηλότερες βαθμολογίες.Τέλος, κατά την επισκεψη στη σελίδα ενός βιβλίου, ο χρήστης έχει πρόσβαση σε μια δυναμική λίστα από άλλα βιβλία που έχουν δανειστεί όσοι δανείστηκαν το συγκεκριμένο βιβλίο.Η υλοποίηση της εφαρμογής έγινε με τη χρήση τεχνολογιών διαδικτύου ώστε ο χρήστης να μην έχει την ανάγκη εγκατάστασης επιπλέον εφαρμογών για την προβολή και διαχείριση της.Το πρόγραμμα δημιουργήθηκε συνδυάζοντας τεχνολογίες ανοιχτού κώδικα(open source), οι οποίες αποτελούν άλλο ένα παράδειγμα για τις δυνατότητες και την χρησιμότητα των open source εφαρμογών. Η υλοποίηση βασίστηκε στη χρήση αντικειμενοστραφούς προγραμματισμού σε συνδυασμό με το σχεσιακό μοντέλο βάσης δεδομένων. Η εφαρμογή θα αναπτυχτεί με την χρήση της PHP ως γλώσσας προγραμματισμού αλλά με συμμετοχή HTML και JavaScript και της MYSQL ως βάση δεδομένων.

## SUMMARY

Aim of final work is the creation of web application for the operation of library. The application supports the storage and presentation of books that has been registered, optionally with their cover. It supports the registration of users ( so much from the administrator what from alone them) and each user it can create a list of beloved books, as well as mark the existing books. Also, it can have access in the background of his lendings in order to sees books that were lented in the past. It can more over mmake reservation in a book in order to lented it soon will be available. The administrator managed the lendings, approves and rejects reservations and marks the returnsof books. Also, the system provides to the all visitors ( or unregistered or registered users ) statistics with regard to the last registered books, the most famous in the lists of lovers, as well as with books with the higher rates. Finally, at the visit in the page of book, the user has access in a dynamic list from other books that have be lented those who were lented the particular book. The concretization of application became with the use of technologies of internet in order that the user does not have the need of installation of additional applications for the projection and the management. The program was created combining technologies of open code, that constitute another example for the possibilities and the usefulness of open source applications. The concretization was based on the use of object oriented programming in combination with the relational database model. The application will be developed with the use PHP of planning but with attendance HTML and JavaScript and MYSQL as database.

## Εισαγωγή

Με την εκμετάλλευση των τεχνολογιών διαδικτύου γίνεται δυνατή η απλούστευση της επαγγελματικής, ακαδημαϊκής και προσωπικής ζωής των ανθρώπων, με πάρα πολλές θετικές συνέπειες στην οργάνωση της καθημερινότητας και στην εξοικονόμηση πόρων.Το μεγάλο πλεονέκτημα του διαδικτύου είναι ότι παρέχει απομακρυσμένη πρόσβαση σε όλων των ειδών τις πληροφορίες χωρίς να χρειάζεται ο χρήστης να χάσει χρόνο, να ξοδέψει χρήματα στη μετακίνηση και να κουραστεί.Δίνει επίσης τη δυνατότητα απλούστευσης πολυάριθμων διεργασιών που στο παρελθόν απαιτούσαν τη φυσική παρουσία των χρηστών σε κάποιο χώρο.Τέλος παρέχει στους κάθε λογής υπεύθυνους λειτουργούς την ευκολία να μηχανοργανώνουν το αντικείμενο της δουλειάς τους, να διατηρούν συγκεντρωμένη την πληροφορία και να απλουστεύουν την αλληλεπίδραση με τους χρήστες, αφού πια η πληροφορία είναι οργανωμένη και διαθέσιμη από το σπίτι του καθένα, χάρη στο διαδίκτυο.

Η λειτουργίας μίας δανειστικής βιβλιοθήκης είναι ένα πολύ χαρακτηριστικό παράδειγμα αυτής της απλούστευσης. Μπορούμε να συγκρίνουμε δεκάδες επιμέρους λειτουργίες που με τη χρήση ενός πληροφοριακού συστήματος έχουν υπεραπλουστευτεί, θα μείνουμε όμως στις πιο βασικές:

- Ο χρήστης δεν χρειάζεται να μεταβεί στο χώρο της βιβλιοθήκης για να αναζητήσει βιβλία που τον ενδιαφέρουν, αφού ο κατάλογος είναι διαθέσιμος στο διαδίκτυο.
- Για το δανεισμό ενός βιβλίου, ο χρήστης θα έπρεπε να πάει ο ίδιος στη βιβλιοθήκη, να εντοπίσει το βιβλίο, να ζητήσει την κράτησή του, να περιμένει όσες μέρες χρειαστεί, και μετά να ξαναπάει προκειμένου να το δανειστεί όταν αυτό θα είναι διαθέσιμο. Τώρα γλιτώνει την πρώτη μετάβαση καθώς η κράτηση γίνεται online.
- Σε όλες τις περιπτώσεις επικοινωνίας μεταξύ του προσωπικού της βιβλιοθήκης και των χρηστών, η πολυδάπανη και χρονοβόρα διαδικασία των τηλεφωνημάτων, τώρα μπορεί να αντικατασταθεί από την αυτοποιημένη αποστολή ενός e-mail. Μόλις ένας χρήστης επιστρέψει ένα βιβλίο, με το πάτημα ενός κουμπιού ο βιβλιοθηκονόμος μπορεί να καταχωρίσει την επιστροφή, και ταυτόχρονα να ενημερώσει τον επόμενο χρήστη που έχει κάνει κράτηση ότι το βιβλίο του είναι διαθέσιμο.
- Κατά την απώλεια ενός βιβλίου, με το πάτημα ενός κουμπιού ο διαχειριστής μπορεί να ενημερώσει τους χρήστες ότι το βιβλίο δεν είναι πια διαθέσιμο, ενώ το βιβλίο θα παραμένει ως καταχώριση στο σύστημα ώστε αν οι χρήστες κρίνουν πως είναι ένα βιβλίο σημαντικό και χρήσιμο, να μπορέσουν να το ξαναπαραγγείλουν.
- Διαδικασίες που μέχρι πριν ήταν πρακτικά αδύνατες τώρα πια είναι εφικτές εύκολα. Παραδείγματος χάριν οι χρήστες της βιβλιοθήκης έχουν τη δυνατότητα να βαθμολογήσουν ένα βιβλίο και να ενημερωθούν για τις βαθμολογίες άλλων χρηστών ώστε να μπορούν να κρίνουν αν το βιβλίο είναι καλό και αν θα τους είναι χρήσιμο.
- Επίσης, ανεξάρτητα από το πληροφοριακό σύστημα της βιβλιοθήκης, με λίγα κλικ ο κάθε χρήστης μπορεί να αναζητήσεις επιπλέον πληροφορίες

στο διαδίκτυο για κάθε βιβλίο που ενδιαφέρεται. Δεν θα ήταν υπερβολή να πούμε ότι όλο το διαδίκτυο, και οι δισεκατομμύρια χρήστες του παγκόσμιου ιστού είναι ουσιαστικά μέλη της βιβλιοθήκης μας, ενώ στην πραγματικότητα αγνοούν καν την ύπαρξή της, αφού οι πληροφορίες και η αξιολόγηση των βιβλίων γίνονται δυνητικά από οποιονδήποτε κάτοικο του πλανήτη έχει διαβάσει από οποιαδήποτε πηγή το συγκεκριμένο βιβλίο!

Η παγκοσμιοποίηση της πληροφορίας, με τη βοήθεια του διαδικτύου, έχει μετατρέψει τον πλανήτη σε ένα τεράστιο ψηφιακό χωριό, όπου όλοι οι χρήστες μπορούν ανά πάσα στιγμή να γνωριστούν μεταξύ τους και να ανταλλάξουν πληροφορίες, κάνοντας τη ζωή τους πιο εύκολη.

Αυτή βέβαια είναι η μία πλευρά του ζητήματος.Από την άλλη, με τη χρήση των νέων τεχνολογιών χάνεται η διαπροσωπική επαφή μεταξύ των χρηστών καθώς και η προσαρμογή στις ανάγκες του καθένα. Για παράδειγμα, στο δικό μας σύστημα, μπορεί ο χρήστης να έχει πρόσβαση στη βαθμολογία ενός βιβλίου, όμως η βαθμολογία προκύπτει ως μέσος όρων βαθμών, και καθώς οι άνθρωποι είναι διαφορετικοί μεταξύ τους, ο καθένας έχει τη δικιά του προσωπικότητα, το δικό του τρόπο σκέψης και μελέτης και τις δικές του ανάγκες, μπορεί ένα βιβλίο με υψηλή βαθμολογία να μην είναι κατάλληλο για κάποιους, ενώ αντίθετα ένα βιβλίο με χαμηλή βαθμολογία να αποτελέσει χρήσιμο οδηγό και πηγή γνώσης.Επιπλέον υπεισέρχεται πάντα ο κίνδυνος οι διάφοροι εκδοτικοί οίκοι να βάλουν υπαλλήλους τους να γράφουν ψευδείς θετικές κριτικές και ψευδείς υψηλές βαθμολογίες στα βιβλία προκειμένου να τα προωθήσουν.

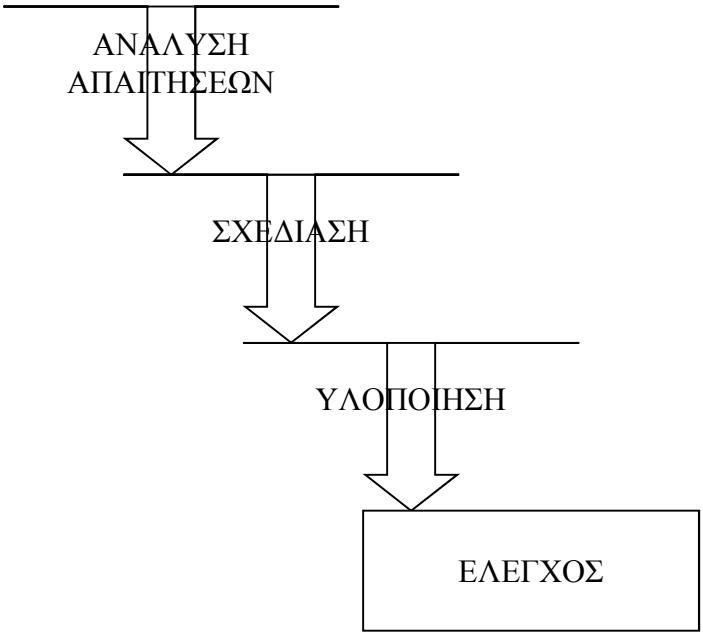
Γίνεται φανερό ότι με τη χρήση των νέων τεχνολογιών οι ζωές των ανθρώπων αλλάζουν. Σίγουρα η αλλαγή αυτή δεν είναι ιδανική, καθώς τίποτα δεν είναι τέλειο, όμως με την ακούραστη δουλειά των μηχανικών, των πληροφορικάριων, των ειδικών τεχνικών σε ζητήματα πληροφορικής, διαδικτύου και προγραμματισμού, ένα μέρος των οποίων προέρχεται και από τους απόφοιτους της δικής μας σχολής, καθίσταται δυνατός ο μετριασμός των αρνητικών

συνεπειων και διαρκής βελτίωση των τεχνολογιών στην υπηρεσία του ανθρώπου.

# 1. Ανάλυση και Σχεδιασμός

## 1.1 Μεθοδολογία

Η μεθοδολογία που θα χρησιμοποιήσουμε για την υλοποίηση του δικτυακού τόπου είναι με βάση το μοντέλο Καταρράκτη (waterfall) όπως φαίνεται και στο Σχήμα 1.




**Σχήμα 1** Μεθοδολογία Ανάπτυξης Εφαρμογής

Στην αρχική φάση γίνεται ανάλυση των απαιτήσεων όσον αφορά τις λειτουργίες που θέλουμε να προσφέρει η συγκεκριμένη εφαρμογή.Γίνεται καθορισμός των χρηστών που πρόκειται να χρησιμοποιήσει την εφαρμογή και καθορίζονται οι λειτουργίες οι οποίες θα είναι διαθέσιμες για κάθε χρήστη.

Στη φάση του σχεδιασμού επιλέγονται εκείνες οι τεχνολογίες που είναι κατάλληλες για την υλοποίηση της εφαρμογής ενώ σχεδιάζεται ο χάρτης πλοήγησης ανάμεσα στις οθόνες της εφαρμογής καθώς επίσης και η βάση δεδομένων που πρόκειται να υλοποιηθεί. Τέλος στη φάση της υλοποίησης χρησιμοποιούνται τα κατάλληλα εργαλεία κατασκευής του site για την κατασκευή του template του site καθώς επίσης και της βάσης δεδομένων, ενώ ταυτόχρονα γίνεται η ενοποίηση των διαφορετικών συστημάτων όπως είναι η βάση δεδομένων με τις δυναμικές σελίδες του website. Τελικά γίνεται έλεγχος των λειτουργιών του website και τυχόν διορθώσεις.

Η σχεδίαση του συστήματος γίνεται με βάση το μοντέλο MVC όπου ουσιαστικά διαχωρίζεται η λειτουργία από τον τρόπο εκτέλεσής της και από την εμφάνιση του αποτελέσματος στην οθόνη του χρήστη/διαχειριστή.

## 1.2 Ανάλυση Απαιτήσεων

Κατά την ανάλυση απαιτήσεων καταγράφονται όλες οι λειτουργίες και υπηρεσίες που πρόκειται να προσφέρει ο δικτυακός τόπος:

- Προβολή λίστας βιβλίων ταξινομημένα από το πιο πρόσφατο στο πιο παλιό
- Προβολή ενός μεμονωμένου βιβλίου
- Προφίλ του χρήστη με ιστορικό δανεισμών, λίστα αγαπημένων και λίστα βιβλίων που έχει βαθμολογήσει
- Σελίδα στατιστικών
- Προσθήκη βιβλίου

- Τροποποίηση βιβλίου
- Απενεργοποίηση βιβλίου (αντί διαγραφής – βλ. παρακάτω)
- Ενεργοποίηση βιβλίου
- Προσθήκη χρήστη
- Απενεργοποίηση χρήστη (αντί διαγραφής – βλ. παρακάτω)
- Αίτημα δανεισμού βιβλίου
- Βαθμολόγηση βιβλίου
- Προσθήκη βιβλίου στη λίστα αγαπημένων του χρήστη
- Αφαίρεση βιβλίου από τη λίστα αγαπημένων του χρήστη
- Έγκριση δανεισμού βιβλίου
- Σύναψη δανεισμού βιβλίου
- Απόρριψη κράτησης βιβλίου
- Επιστροφή βιβλίου

### 1.3 Ομάδες χρηστών

Οι παραπάνω λειτουργίες δεν θα πρέπει να διαθέσιμες σε όλους τους χρήστες.Έτσι υπάρχει μια διαβάθμιση όσον αφορά τα δικαιώματα που θα έχει ο κάθε χρήστης στις παραπάνω λειτουργίες. Έτσι για τις απαιτήσεις του δικτυακού τόπου ορίζονται δυο κατηγορίες χρηστών:

- Επισκέπτες. Πρόκειται για τους χρήστες του Διαδικτύου που επισκέπτονται την ιστοσελίδα χωρίς να είναι μέλη της βιβλιοθήκης. Έχουν δικαίωμα να δούν όλα τα βιβλία, καθώς και να πληροφορηθούν για τα στατιστικά τους. Μπορούν να μάθουν ποια είναι τα πιο διάσημα βιβλία, ποια έχουν τις υψηλότερες βαθμολογίες, ποια προσθέτουν οι περισσότεροι χρήστες στις λίστες αγαπημένων τους.

- Εξουσιοδοτημένοι χρήστες – μέλη της βιβλιοθήκης. Πρόκειται για τους επισκέπτες οι οποίοι είναι εγγεγραμμένοι ως εξουσιοδοτημένοι χρήστες για να μπορούν να υποβάλλουν τα στοιχεία τους για κράτηση και δανεισμό βιβλίων, βαθμολόγηση βιβλίων, διατήρηση λίστας αγαπημένων.
- Διαχειριστές. Έχουν ένα διευρυμένο διαχειριστικό περιβάλλον με περισσότερα δικαιώματα. Μπορούν να κάνουν ότι μπορεί να κάνει οποιοσδήποτε χρήστης αλλά επιπλέον μπορούν να τροποποιήσουν τα βιβλία, να απενεργοποιήσουν όσα έχουν χαθεί ή να τα ξαναενεργοποιήσουν εάν βρεθούν. Μπορούν να διαγράψουν χρήστες από το σύστημα ή να τους ξαναεγγράψουν. Και φυσικά μπορούν να εγκρίνουν κρατήσεις βιβλίων ή να τις απορρίψουν, μπορούν να επικυρώσουν το δανεισμό ενός βιβλίου και να μαρκάρουν την επιστροφή του.
- Διαγραμμένοι (απενεργοποιημένοι) χρήστες. Δεν έχουν κανένα απολύτως δικαίωμα, κατά τη σύνδεση το σύστημα τους ενημερώνει ότι έχουν διαγραφεί και τους αποσυνδέει αυτόματα.

Για τον έλεγχο της πρόσβασης στις υπηρεσίες ανάλογα με τον τύπο χρήστη που επισκέπτεται το site είναι απαραίτητη μια διαδικασία Ταυτοποίησης (login).Με τη διαδικασία αυτή οι χρήστες που έχουν πρόσβαση σε συγκεκριμένες υπηρεσίες του δικτυακού τόπου πέραν των γενικών που είναι προσβάσιμες σε όλους, θα πρέπει να υποβάλλουν ένα όνομα χρήστη και κωδικό.Το σύστημα θα ελέγχει το αν είναι εγγεγραμμένος ο χρήστης στη λίστα εξουσιοδοτημένων χρηστών και ανάλογα με τον τύπο του θα του δίνει και τα αντίστοιχα δικαιώματα πρόσβασης.

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά τα δικαιώματα όλων των χρηστών:

| Λειτουργία | Επισκέπτης | Χρήστης | Διαχειριστής |
|---|---|---|---|
| Προβολή λίστας βιβλίων | X | X | X |
| Προβολή μεμονωμένου βιβλίου | X | X | X |
| Προφίλ του χρήστη | | X | X |
| Σελίδα στατιστικών | X | X | X |
| Προσθήκη βιβλίου | | | X |
| Τροποποίηση βιβλίου | | | X |
| Απενεργοποίηση βιβλίου | | | X |
| Ενεργοποίηση βιβλίου | | | X |
| Προσθήκη χρήστη - εγγραφή | X | | X |
| Απενεργοποίηση χρήστη | | | X |
| Αίτημα δανεισμού βιβλίου | | X | X |
| Βαθμολόγηση βιβλίου | | X | X |
| Προσθήκη βιβλίου στη λίστα αγαπημένων | | X | X |
| Αφαίρεση βιβλίου από τη λίστα αγαπημένων | | X | X |
| Έγκριση δανεισμού βιβλίου | | | X |
| Σύναψη δανεισμού βιβλίου | | | X |
| Απόρριψη κράτησης βιβλίου | | X | X |
| Επιστροφή βιβλίου | | | X |

### 1.4 Διασύνδεση με το σύστημα του ΤΕΙ

Το σύστημα σχεδιάζεται με τέτοιον τρόπο ώστε η διασύνδεσή του με το πληροφοριακό σύστημα των χρηστών του ΤΕΙ καθώς και με το υπάρχον σύστημα της βιβλιοθήκης να είναι εύκολη.

Για το λόγο αυτό επιλέχθηκαν κάποιες λύσεις στην υλοποίηση που θα περιορίζουν όσο το δυνατόν τον πλεονασμό της πληροφορίας:

- Οι χρήστες δεν έχουν username, συνδέονται με το e-mail τους: Καθώς ο βασικός τρόπος πιστοποίησης ενός χρήστη σε μια πανεπιστημιακή πληροφοριακή υπηρεσία είναι το πανεπιστημιακό του e-mail, ακολουθήσαμε αυτή την οδό και στο σύστημα της βιβλιοθήκης.
- Δεν αποθηκεύεται πουθενά πληροφορία σχετικά με το ονοματεπώνυμο του χρήστη, το τμήμα, την ημερομηνία εγγραφής του κλπ. Η πληροφορία

αυτή είναι αποθηκευμένη στο κεντρικό πληροφοριακό σύστημα του ΤΕΙ, και κατά τη διασύνδεση του συστήματος θα υπάρχει πρόσβαση σε όσα μέρη της χρειάζεται.

### Λεπτομέρειες πάνω στη λειτουργικότητα

Για λόγους ακεραιότητας των δεδομένων, κανένας χρήστης και κανένα βιβλίο δεν μπορεί να διαγραφεί από το σύστημα. Προφανώς, διαγραφή μπορεί να γίνει χειροκίνητα στη βάση, αλλά τέτοια λειτουργία δεν συνίσταται, δεν προβλέπεται, και δεν υποστηρίζεται από το interface.

Αντί της διαγραφής, μπορεί να μαρκαριστεί από το διαχειριστή ως «απενεργοποιημένο(ς)», το οποίο πρακτικά σημαίνει ότι:

- Ο user δεν μπορεί να κάνει login (άρα δεν μπορεί να προχωρήσει σε καμία από τις διαδικασίες)
- Το book δεν μπορεί να ζητηθεί για δανεισμό. Ωστόσο εμφανίζεται κανονικά στη λίστα, με την ένδειξη «απωλεσθέν» και οι χρήστες μπορούν να το βαθμολογήσουν ή να το προσθέσουν στα αγαπημένα τους.

Με τον τρόπο αυτό, καθίσταται πιο εύκολη η διασύνδεση του συστήματος με το κεντρικό πληροφοριακό σύστημα της βιβλιοθήκης του ΤΕΙ. Εφόσον η πληροφορία για την ύπαρξη ή τη διαγραφή ενός βιβλίου γίνεται στο κεντρικό σύστημα, το δικό μας σύστημα είναι απλώς «παθητικός δέκτης» αυτής της πληροφορίας.

## 1.5 Σχεδίαση

Το στάδιο αυτό περιλαμβάνει την σχεδίαση της βάσης δεδομένων και του συστήματος διεπαφής καθώς επίσης και τον καθορισμό των εργαλείων που πρόκειται να χρησιμοποιηθούν στην φάση της υλοποίησης.

### Σχεδίαση Βάσης Δεδομένων

Οι πληροφορίες καταχωρούνται στη βάση δεδομένων “library” της MySQL. Οι βασικοί πίνακες είναι οι users, books, favorites, lend και rates. Ένας διαχειριστής του δικτυακού τόπου είναι καταχωρημένος στον πίνακα users.

| user | book | favorites | rate | lend |
|---|---|---|---|---|
| id | id | id | id | id |
| mail | title | uid (@user.id) | uid (@user.id) | uid (@user.id) |
| password | author | bid (@book.id) | bid (@book.id) | bid (@book.id) |
| created (date) | isbn | created (date) | created (date) | status |
| deactivated | publisher | | rate | created (date) |
| isadmin | image | | | approved (date) |
| | created (date) | | | lent (date) |
| | deactivated | | | expire (date) |
| | | | | returned (date) |

## Επεξήγηση πινάκων και πεδίων

### *1) Πίνακας user: Φιλοξενεί τους χρήστες του συστήματος*

- Id: σειριακός αριθμός, ταυτοποιητικό στοιχείο για τη βάση
- Mail: mail address του χρήστη που θα χρησιμεύει και για το login του (δεν χρειάζεται εξτρα username)
- Password

- Created: Ημερομηνία δημιουργίας του χρήστη
- Deactivated: Boolean τιμή (int 0 ή 1) που δείχνει αν ο χρήστης έχει απενεργοποιηθεί (αντί της διαγραφής, οι χρήστες απενεργοποιούνται ώστε τα δεδομένα που συνδέονται με αυτούς να μη χάσουν την ακεραιότητά τους)
- Isadmin: Boolean τιμή (int 0 ή 1) που δείχνει εάν ο χρήστης έχει πρόσβαση στις διαχειριστικές σελίδες και εάν μπορεί να εκκινήσει διαχειριστικές διαδικασίες.

### *2) Πίνακας book: Φιλοξενεί τα βιβλία*

- Id: σειριακός αριθμός, ταυτοποιητικό στοιχείο για τη βάση
- Title: varchar255, τίτλος του βιβλίου
- Author: varchar 255, συγγραφέας
- ISBN: varchar20
- Publisher: varchar255, εκδοτικός οίκος
- Image: varchar500, link προς εικόνα εξωφύλλου
- Created: Ημερομηνία δημιουργίας του βιβλίου
- Deactivated: Boolean τιμή (int 0 ή 1) που δείχνει αν το βιβλίο έχει απενεργοποιηθεί (αντί της διαγραφής, τα βιβλία απενεργοποιούνται ώστε τα δεδομένα που συνδέονται με αυτά να μη χάσουν την ακεραιότητά τους)

### *3) Πίνακας favorites: Φιλοξενεί τα αγαπημένα βιβλία των χρηστών*

- Id: σειριακός αριθμός, ταυτοποιητικό στοιχείο για τη βάση
- Uid: foreign key, δείχνει στο id του χρήστη
- Bid: foreign key, δείχνει στο id του βιβλίου
- Created: Ημερομηνία που το βιβλίο προστέθηκε στα αγαπημένα

Επεξήγηση ρόλου: Όταν ο χρήστης με id = 123 προσθέσει στα αγαπημένα του το βιβλίο με id = 456, θα προστεθεί στον πίνακα favorites μία νέα καταχώριση, με τις αντίστοιχες τιμές.

Έτσι, μπορεί να γίνει αναζήτηση αγαπημένων καταχωρίσεων, είτε με βάση έναν συγκεκριμένο χρήστη (προβολή αγαπημένων του χρήστη Χ) είτε με βάση ένα συγκεκριμένο βιβλίο (προβολή των χρηστών που έχουν το βιβλίο Ζ στα αγαπημένα τους)

### *4) Πίνακας rates: Φιλοξενεί τις βαθμολογίες που έχουν βάλει οι χρήστες στα βιβλία*

- Id: σειριακός αριθμός, ταυτοποιητικό στοιχείο για τη βάση
- Uid: foreign key, δείχνει στο id του χρήστη
- Bid: foreign key, δείχνει στο id του βιβλίου
- Created: Ημερομηνία που καταχωρήστηκε η βαθμολογία
- Rate: ο βαθμός (int 0-10)

Επεξήγηση ρόλου: Όταν ο χρήστης με id = 123 βαθμολογήσει το βιβλίο με id = 456, βάζοντας βαθμό 8, θα προστεθεί στον πίνακα rates μία νέα καταχώριση, με τις αντίστοιχες τιμές.

Έτσι, μπορεί να γίνει αναζήτηση βαθμολογιών, είτε με βάση έναν συγκεκριμένο χρήστη (προβολή βιβλίων που έχουν βαθμολογηθεί από το χρήστη Χ, μαζί με τους βαθμούς τους) είτε με βάση ένα συγκεκριμένο βιβλίο (προβολή των χρηστών που έχουν βαθμολογήσει το βιβλίο Ζ, μαζί με τους βαθμούς). Επίσης είναι δυνατή και η αναζήτηση βιβλίων βάσει βαθμολογίας (π.χ. «Δείξε μου όλα τα βιβλία που οι χρήστες έχουν βαθμολογήσει με 10»)

***5) Πίνακας lend: Είναι ο βασικός πίνακας δανεισμών, από τον οποίο προκύπτει η κατάσταση κάθε βιβλίου.***

- Id: σειριακός αριθμός, ταυτοποιητικό στοιχείο για τη βάση
- Uid: foreign key, δείχνει στο id του χρήστη
- Bid: foreign key, δείχνει στο id του βιβλίου
- Status: Κατάσταση στην οποία βρίσκεται το βιβλίο (αριθμός)
- Created: Ημερομηνία που καταχωρήστηκε η βαθμολογία
- Approved: Ημερομηνία έγκρισης της κράτησης
- Lent: Ημερομηνία δανεισμού
- Expire: Ημερομηνία λήξης δανεισμού
- Returned: Ημερομηνία επιστροφής

Επεξήγηση ρόλου: Όταν ένας χρήστης ζητήσει ένα βιβλίο, εισάγεται μία καταχώρηση που δείχνει στο χρήστη και στο βιβλίο, με status 0 και προφανώς created date.

Εάν ο διαχειριστής απορρίψει την κράτηση, η καταχώρηση διαγράφεται.

Εάν ο διαχειριστής αποδεχτεί την κράτηση, το status γίνεται 1, προστίθεται η ημερομηνία approved, και ο χρήστης παίρνει e-mail πως το βιβλίο του είναι διαθέσιμο για δανεισμό.

Όταν ο διαχειριστής προχωρήσει με το δανεισμό του βιβλίου από το χρήστη που έκανε την κράτηση, το status γίνεται 2, και προστίθεται η ημερομηνία lent, καθώς και η ημερομηνία expire που είναι η ημερομηνία που πρέπει να επιστραφεί για να μην επιβληθεί πρόστιμο καθυστέρησης.

Τέλος, όταν το βιβλίο επιστραφεί, το status γίνεται 3 και σημειώνεται η ημερομηνία επιστροφής returned.

Από τον πίνακα αυτό, με τα κατάλληκα SQL queries, αντλούνται όλες οι πληροφορίες σχετικά με την κατάσταση και τη διαθεσιμότητα κάθε βιβλίου.

### Λειτουργίες χρήστη

Ένας χρήστης βλέπει ένα βιβλίο στις εξής καταστάσεις:

1) Μη διαθέσιμο προς κράτηση
    - Κάποιος άλλος έχει κάνει κράτηση και είναι ο επόμενος στη σειρά για να το δανειστεί: Η ουρά της κράτησης έχει χωρητικότητα μόνο για 1 θέση. (Lend object @book status 0 ή 1)
    - Ο χρήστης έχει ήδη 3 βιβλία δανεισμένα ή κρατημένα. (3 Lend objects @user status 0 ή 1 ή 2)
2) Κρατημένο, διαθέσιμο προς δανεισμό.
    - Lend object @user&book status 1
3) Διαθέσιμο προς κράτηση
    - Σε κάθε άλλη περίπτωση

### Διαχειριστικές λειτουργίες

Ο admin βλέπει τη λίστα των κρατημένων βιβλίων, και μπορεί να εγκρίνει ή να απορρίψει μια κράτηση.

Έγκριση κράτησης: Lend object status σε 1 (approved). Στο σημείο αυτό αποστέλλεται e-mail προς το χρήστη.Στη συνέχεια μπορεί να περάσει σε σύναψη δανεισμού - Lend object status σε 2 (lent)

Απόρριψη κράτησης: Διαγραφή του Lend object.

Επίσης ο admin βλέπει τη λίστα των δανεισμένων βιβλίων και μπορεί να σημειώσει την επιστροφή ενός βιβλίου - Lend object status σε 3 (returned).

Σε κάθε περίπτωση, ασκείται εξαντλητικός έλεγχος ώστε να μη χαθεί η συνοχή των δεδομένων.

Θα υλοποιηθεί μία επιπλέον σελίδα, εκτός προδιαγραφών, που θα παρουσιάζει status reports για την ακεραιότητα των δεδομένων με live ελέγχους:

Α) Δεν υπάρχουν ορφανές παραγγελίες βιβλίων.

Β) Δεν υπάρχουν ορφανοί δανεισμοί βιβλίων.

Γ) Δεν υπάρχει «μπλοκαρισμένο» βιβλίο (π.χ. με περισσότερες από μία κρατήσεις)

Δ) Δεν υπάρχουν ορφανές βαθμολογίες.

Ε) Δεν υπάρχουν ορφανά «αγαπημένα».

### 1.6 Use cases

Use case δανεισμού:

- Κράτηση βιβλίου (χρήστης): Δημιουργείται ένα Lend object με status 0.
- Έγκριση κράτησης (admin): Lend object status = 1.
- Ο χρήστης ενημερώνεται για την έγκριση της κράτησης.
- Σύναψη δανεισμού (admin): Lend object status = 2.
- Επιστροφή βιβλίου (admin): Lend object status = 3.

Use case απόρριψης κράτησης:

- Κράτηση βιβλίου (χρήστης): Δημιουργείται νέο Lend object με status 0.
- Απόρριψη κράτησης (admin): Διαγραφή του Lend object.

Use case απόρριψης κράτησης που έχει εγκριθεί:

- Κράτηση βιβλίου (χρήστης): Δημιουργείται νέο Lend object με status 0.
- Έγκριση κράτησης (admin): Lend object status = 1

- Απόρριψη κράτησης (admin): Διαγραφή του Lend object.

Use case ακύρωσης κράτησης:

- Κράτηση βιβλίου (χρήστης): Δημιουργείται νέο Lend object με status 0.
- Ακύρωση κράτησης (χρήστης): Διαγραφή του Lend object.

Use case ακύρωσης κράτησης που είχε εγκριθεί:

- Κράτηση βιβλίου (χρήστης): Δημιουργείται νέο Lend object με status 0.
- Έγκριση κράτησης (admin): Lend object status = 1
- Ακύρωση κράτησης (χρήστης): Διαγραφή του Lend object.

## 1.7Προγραμματισμός σε επίπεδο Client

### HTML 5 - HyperText Markup Language

Η HTML (HyperText Markup Language) είναι η βασική γλώσσα που χρησιμοποιείται στον Παγκόσμιο Ιστό για την περιγραφή της δομής και της μορφής του περιεχομένου ενός εγγράφου.Οι φυλλομετρητές (web browsers) μεταφράζουν τη γλώσσα αυτή έτσι ώστε να παρουσιάσουν στο χρήστη το περιεχόμενο του εγγράφου με τον τρόπο αναπαράστασης που περιγράφεται από τη γλώσσα.

Η περιγραφή των περιεχομένων μιας ιστοσελίδας γίνεται με τη χρήση των tags.Σε συνδυασμό με τα Cascading Style Sheets (CSS) ο συντάκτης μιας ιστοσελίδας μπορεί να προσδιορίσει πως τα στοιχεία του εγγράφου θα εμφανιστούν παρακάμπτοντας τις προεπιλογές ενός browser.

Οι βασικές αρχές της HTML στην έκδοση 5 είναι οι εξής:

- Διαχωρισμός του περιεχομένου από τον τρόπο παρουσίασης μέσω των style sheets. Αποτελεί τη βασική αρχή σχεδίασης Web περιεχομένου. Η HTML5 κάνει σαφή διαχωρισμό της δομή από τον τρόπο παρουσίασης του περιεχομένου για την καλύτερη και αποδοτικότερη δημιουργία ιστοσελίδων. Αυτό επιτυγχάνεται με τη χρήση των cascade style sheets (CSS).
- Προσβασιμότητα και Διεθνής Προτυποποίηση. Στην HTML 5 βασική αρχή θεωρείται η πρόσβαση στο περιεχόμενο για άτομα που χρησιμοποιούν ειδικούς browsers είτε λόγω μειωμένων ικανοτήτων ή λόγω έλλειψης τηλεπικοινωνιακής υποδομής. Επίσης είναι σημαντική η υποστήριξη κωδικοποιήσεων για όλες τις γλώσσες.
- Αποδοτικότερη μετάφραση των εγγράφων Web. Στην HTML 5 προστέθηκαν αρκετά στοιχεία για την καλύτερη και αποδοτικότερη μετάφραση των εντολών περιγραφής του περιεχομένου.
- Απλοποίηση πληροφοριών doctype. Στην HTML 5δεν δηλώνεται το Document Type Definition (DTD): Strict, Transitional, και Frameset όπως γίνονταν στην HTML 4.

### CSS - Cascading Style Sheets

Τα Cascading Style Sheets (CSS) είναι ένας μηχανισμός για τη μορφοποίηση του περιεχομένου των εγγράφων που είναι γραμμένα σε HTML ή XML, εφαρμόζοντας στυλ μορφοποίησης σε τύπους στοιχείων ή κλάσεων καθορισμένων από τον συντάκτη του εγγράφου ή συγκεκριμένες περιπτώσεις αυτών των στοιχειών.

Τα Stylesheets μπορούν να χρησιμοποιηθούν για να ορίσουν τον τρόπο εμφάνισης ενός ολόκληρου δικτυακού τόπου με ενιαίο τρόπο.Τα CSS υποστηρίχτηκαν από το W3C έτσι ώστε η μορφοποίηση των HTML σελίδων να στηρίζεται σε stylesheets έτσι να γίνεται διαχωρισμός του περιεχομένου από τον

τρόπο παρουσίασης, συμβάλλοντας έτσι σε ένα πιο απλό και σωστά δομημένο Παγκόσμιο Ιστό.

Για την εφαρμογή επιλέχθηκε η χρήση του Bootstrap Framework. Το Bootstrap framework δημιουργήθηκε από τους ιδρυτές του Twitter και δόθηκε σαν opensource λύση στην παγκόσμια κοινότητα ώστε να λειτουργεί σαν base theme για την ανάπτυξη responsive, mobile-first ιστοσελίδων και εφαρμογών. Εμπεριέχει ένα προκατασκευασμένο grid το οποίο ο χρήστης μπορεί να τροποποιήσει χωρίς να χρειάζεται να ξαναγράψει από την αρχή όλους τους κανόνες της CSS.

### JavaScript

Η JavaScript αρχικά δημιουργήθηκε από τη Netscape για να επιτρέπει στις σελίδες που αποκωδικοποιούνται από τον browser Navigator 2.x να έχουν αλληλεπίδραση με τον χρήστη.Το όνομα αν και παραπέμπει στη γλώσσα της Sun Microsystems, τη Java, έχει λίγα κοινά. Αρχικά προοριζόταν για προγραμματισμό σε επίπεδο server side αλλά η υποστήριξή της από τους browsers την έκανε πολύ δημοφιλή στον προγραμματισμό σε επίπεδο client. Αρχικά υποστηριζόταν από τον Netscape Communicator και στη συνέχεια από τον Microsoft Internet Explorer.Σήμερα με την προτυποποίημένη έκδοση της, την ECMA-262 ECMAScript είναι πλέον η standard γλώσσα προγραμματισμού που συναντάμε ενσωματωμένη σε HTML κώδικα.

Η JavaScript είναι μια απλή και ελαφριά όσον αφορά την απαίτηση πόρων γλώσσα η οποία χρησιμοποιεί διερμηνευτή για την μετάφρασής της και υποστηρίζεται από όλες τις πλατφόρμες. Έχει στοιχεία αντικειμενοστραφούς προγραμματισμού και δίνει τη δυνατότητα δημιουργίας εφέ και αλληλεπίδρασης με το χρήστη των ιστοσελίδων .

Η βασική JavaScript περιέχει σύνολα αντικειμένων όπως τύπους Array, Date, και Math, και επίσης βασικά στοιχεία όπως εντολές, τελεστές πράξεων και δομές.Η Core JavaScript αποτελεί τη βάση για την γλώσσα σε επίπεδο client side και

server side. Η Client-side έκδοση της JavaScript περιέχει επίσης στοιχεία για την διαχείριση αντικειμένων που βρίσκονται σε μια ιστοσελίδα μέσω του προτύπου DOM (Document Object Model).

Μερικές από τις δυνατότητες της JavaScript οι ακόλουθες:

- Χρησιμοποιείται για να προσθέσει αλληλεπιδραστικότητα των χρηστών με τις ιστοσελίδες χωρίς να επιφορτίζεται ο server.
- Μπορεί να γράψει και να διαβάσει cookies
- Χρησιμοποιείται για τη δημιουργία εργαλείων εκτέλεσης υπολογισμών ενσωματωμένων στις ιστοσελίδες.
- Μπορεί να χρησιμοποιηθεί για τη δημιουργία ιστοσελίδων on-the-fly χωρίς τη παρέμβαση του server.
- Έχει τη δυνατότητα αναγνώρισης του browser, του Λειτουργικού Συστήματος ή του μεγέθους της οθόνης του client.
- Χρησιμοποιείται πολύ συχνά για την επικύρωση των δεδομένων εισόδου σε μια φόρμα.

## 1.8Προγραμματισμός σε επίπεδο Server

Πρόκειται για τεχνολογίες προγραμματισμού και ανάπτυξης εφαρμογών οι οποίες εκτελούνται στην μεριά του web server πριν το περιεχόμενο αποσταλεί στον web browser του τελικού χρήστη.

- PHP. Η PHP είναι μια διαδεδομένη γλώσσα script που εκτελείται σε επίπεδο server και χρησιμοποιείται στη δημιουργία ισοτόπων δυναμικού περιεχομένου. Είναι γλώσσα ανοικτού κώδικα.

- MySQL. Η MySQL είναι ένα ισχυρό Σύστημα Διαχείρισης Βάσεων Δεδομένων. Πρόκειται για λογισμικό ανοικτού κώδικα και η διαχείριση των βάσεων γίνεται με τη γλώσσα Structured Query Language (SQL). Συνδυάζεται συνήθως με PHP και τη υποστηρίζεται από όλα τα

λειτουργικά συστήματα. Βασικός ανταγωνιστής είναι ο SQL Server της Microsoft με αρκετά υψηλό κόστος εγκατάστασης και λειτουργίας.

Πιο αναλυτικά οι πιο διαδεδομένες τεχνολογίες ανάπτυξης δυναμικών εφαρμογών σε επίπεδο Server είναι οι παρακάτω.

### PHP

Οι ιστοσελίδες περιέχουν κώδικα γραμμένο σε γλώσσα HTML (Hyper Text Markup Language).Σε μια Web εφαρμογή όμως εκτός των περιεχομένων, (server pages) οι ιστοσελίδες περιέχουν ενσωματωμένο και εκτελέσιμο κώδικα ο οποίος εκτελείται στον Server χωρίς να είναι ορατός στον τελικό χρήστη.Τέτοιες σελίδες είναι οι PHP σελίδες (Hypertext Preprocessor).Η PHP ξεκίνησε αρχικά σαν μια σύντομη έκδοση της Perl από τον Rasmus Lerdorf το 1994. Δανείστηκε στοιχεία από τη C, τη Java και την Perl και αναπτύχθηκε έτσι ώστε να μπορεί να ενσωματωθεί σε αρχεία HTML με επέκταση ".php", ".php3", ή ".phtml".

Βασικό της χαρακτηριστικό είναι ότι οι σελίδες αυτές σχεδιάζονται δυναμικά ανάλογα με την εκτέλεση του κώδικα. Τα βασικά χαρακτηριστικά των δυναμικών PHP σελίδων είναι τα εξής:

- Είναι πολύ εύκολη η εκμάθηση της PHP
- Υποστηρίζει πολλές πλατφόρμες (Windows, Linux, Unix, κα)
- Υπάρχει συμβατότητα με σχεδόν όλους τους servers (Apache, IIS, κα)
- Παρέχει εύκολη συνδεσιμότητα με Βάσεις Δεδομένων όπως MySQL, Oracle, Sybase, PostgreSQL, Generic ODBC κα.
- Ανήκει στην κατηγορία του Λογισμικού Ανοικτού Κώδικα (Open Source software – OSS).
- Συνεργάζεται με την επίσης Ανοικτού Κώδικα βάση Δεδομένων MySQL.
- Η χρήση είναι δωρεάν.
- Ο προγραμματισμός σε PHP είναι οικείος σε προγραμματιστές C, Perl και Java.

### Βάση Δεδομένων MySQL

Βασικό συστατικό μιας web εφαρμογής είναι μια βάση δεδομένων για την καταχώρηση, συντήρηση και προβολή πληροφοριών στους χρήστες.Στην πλευρά του server υπάρχει ένα σύστημα Διαχείρισης Βάσης Δεδομένων συνήθως Σχεσιακής (Relational Database System - RDBMS) όπου καταχωρούνται τα δεδομένα.Ανάλογα με τις ενέργειες και τις αιτήσεις του χρήστη, ο server επικοινωνεί με το σύστημα διαχείρισης της βάσης δεδομένων εκτελώντας ερωτήματα στη γλώσσα SQL.

Το σύστημα διαχείρισης της Βάσης Δεδομένων απαντάει σε αυτά τα ερωτήματα του server είτε αποστέλλοντας τα δεδομένα που προέκυψαν σαν αποτελέσματα των ερωτημάτων ή εκτελώντας κάποια εισαγωγή ή διαγραφή δεδομένων.Η επικοινωνία μεταξύ Server εφαρμογής και Βάσης Δεδομένων γίνεται με τη χρήση οδηγών (Database Connectivity drivers).Στο Σχήμα 4 φαίνεται η διαδικασία σύνδεσης της Βάσης Δεδομένων με την υπόλοιπη Client – Server εφαρμογή.

Στη συγκεκριμένη εφαρμογή χρησιμοποιείται η MySQL.Η MySQL είναι ένα Σύστημα Διαχείρισης Σχεσιακής Βάσης Δεδομένων και περιέχει και έναν μικρό server της βάσης.Αναπτύχθηκε σαν μια εφαρμογή της γλώσσας SQL από την TcX. Είναι αρκετά σταθερό σύστημα και πολύ ευέλικτο.Υποστηρίζει όλες τις λειτουργίες και τους τύπους δεδομένων της standard. Τα πιο σημαντικά χαρακτηριστικά της MySQL είναι τα ακόλουθα:

- Η MySQL ανήκει στο λογισμικό Ανοικτού Κώδικα (Open Source).
- Είναι γρήγορη και υποστηρίζει multi-thread και πολυχρηστικό περιβάλλον.
- Υποστηρίζει τη standard SQL.
- Υποστηρίζει ποικίλες πλατφόρμες.

- Η χρήση της είναι δωρεάν.

### PHP – MySQL

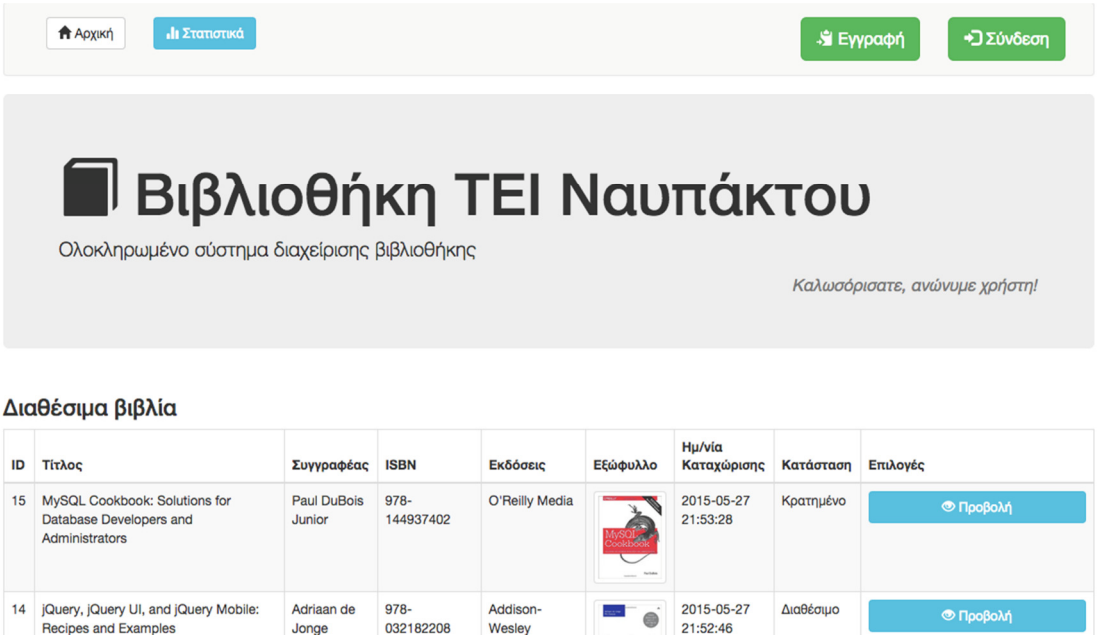
Ο συνδυασμός της γλώσσας PHP και της ΒΔ MySQL έχει σαν βασικό πλεονέκτημα ότι υποστηρίζεται από σχεδόν όλες τις πλατφόρμες.Έχουν κοινά χαρακτηριστικά όπως είναι το γεγονός ότι ανήκουν και οι δύο στις εφαρμογές Ανοικτού Κώδικα και τα δικαιώματα χρήσης τους είναι δωρεάν.Εξαιτίας των κοινών αυτών χαρακτηριστικών, έχουν αναπτυχθεί Web Servers που υποστηρίζουν τα δύο αυτά λογισμικά και την άμεση συνδεσιμότητα μεταξύ τους.

Η βασική λειτουργία του συνδυασμού των δύο τεχνολογιών είναι η εξής: Ένας δυναμικός δικτυακός τόπος αποτελείται από PHP σελίδες. Η λειτουργικότητα που παρέχουν οι σελίδες αυτές στο χρήστη στηρίζεται στον εκτελέσιμο κώδικα που είναι ενσωματωμένος.Οι δυναμικές σελίδες PHP περιέχουν κώδικα ο οποίος εκτελείται στον server.Ο κώδικας αυτός εκτελεί ερωτήματα σε SQL τα οποία μεταβιβάζονται μέσω του ειδικού driver της MySQL στη βάση MySQL.Ανάλογα με την αίτηση του χρήστη μπορεί να γίνει μια καταχώριση, τροποποίηση ή διαγραφή δεδομένων στη Βάση.Επίσης ο χρήστης μπορεί να αιτηθεί την ανάκτηση κάποιας πληροφορίας.Η αίτηση μεταβιβάζεται στη Βάση Δεδομένων και τα αποτελέσματα επιστρέφουν στο Web Server.Στη συνέχεια τα δεδομένα χρησιμοποιούνται στη δημιουργία της σελίδας που τελικά αποστέλλεται στο χρήστη και του προβάλει το περιεχόμενο που ζήτησε.Το περιεχόμενο παρουσιάζεται στο χρήστη από τον αντίστοιχο browser.

# 2. Στάδια υλοποίησης

## 2.1 Αρχική σελίδα – index.php

Στην αρχική σελίδα παρουσιάζεται η λίστα των διαθέσιμων βιβλίων καθώς και τα κουμπιά για τις βασικές λειτουργίες που προσφέρονται στον ανώνυμο χρήστη:

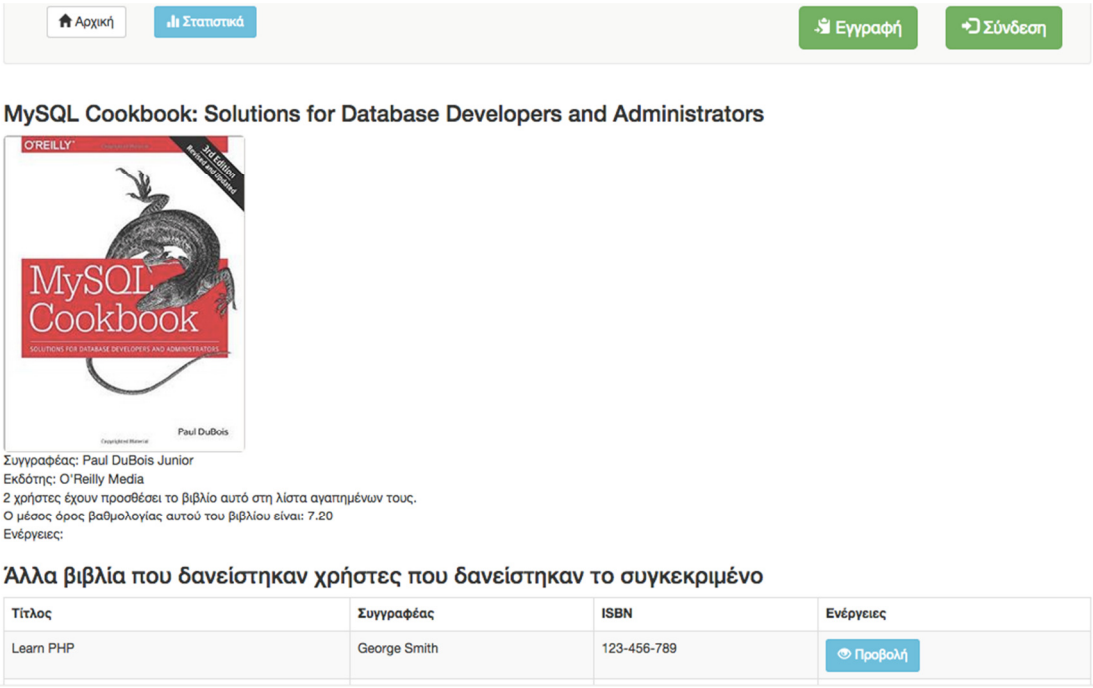



```php
<?php
/*** begin the session ***/
session_start();
include_once 'helpers.php';

?>
<!DOCTYPE html>
<html lang="el">
<head>
<title>Βιβλιοθήκη ΤΕΙ Ναυπάκτου</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'messages.php'; ?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">

</div>

<div class="row">
```

```
<div class="jumbotron">
<h1><span class="glyphicon glyphicon-book"></span>Βιβλιοθήκη TEI Ναυπάκτου</h1>
<p>Ολοκληρωμένο σύστημα διαχείρισης βιβλιοθήκης</p>
<?php include_once 'user_banner.php'; ?>
</div>
</div>

<div class="row">
<h3>Διαθέσιμα βιβλία</h3>
</div>

<div class="row">

<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Εκδόσεις</th>
<th>Εξώφυλλο</th>
<th>Ημ/νία Καταχώρισης</th>
<th>Κατάσταση</th>
<th>Επιλογές</th>
</tr>
</thead>
<tbody>
<?php
get_homepage_list();
?>
</tbody>
</table>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

## 2.2 view.php – Προβολή βιβλίου

Προβολή πληροφοριών για το βιβλίο που επέλεξε ο χρήστης:




```php
<?php
/*** begin the session ***/
session_start();
require_once 'helpers.php';

$id = null;
if ( !empty($_GET['id'])) {
$id = $_REQUEST['id'];
}

if ( null==$id ) {
  header("Location: index.php");
} else {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "SELECT * FROM books where id = ?";
$q = $pdo->prepare($sql);
$q->execute(array($id));
$data = $q->fetch(PDO::FETCH_ASSOC);
Database::disconnect();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
<title><?php echo $data['title']; ?></title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">

<div class="span10 offset1">
```

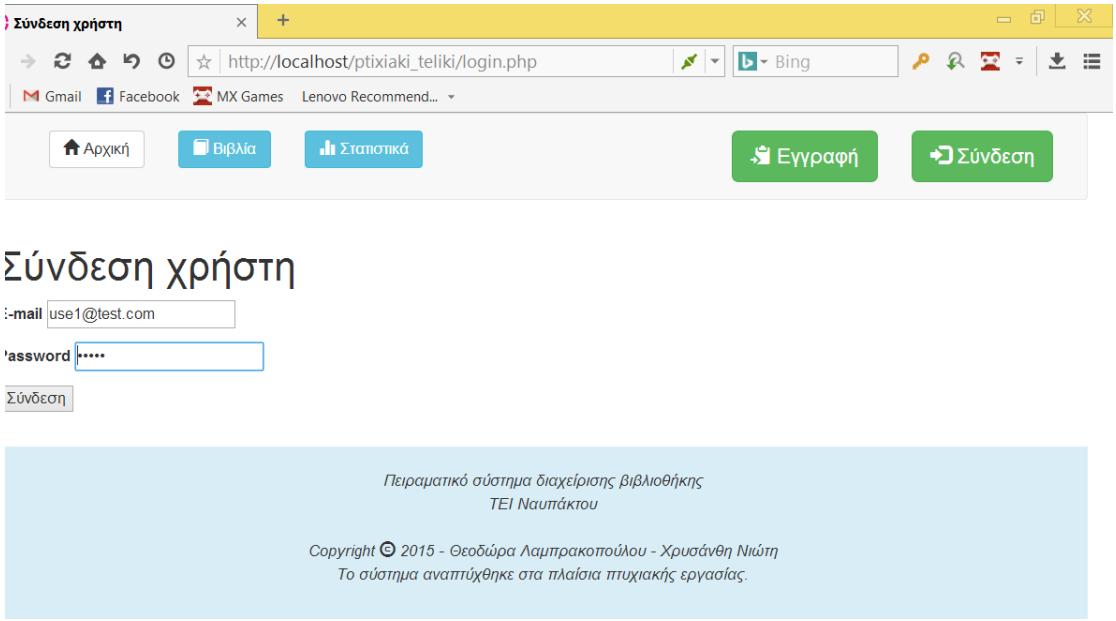
```
<div class="row">
<?php

      include_once 'messages.php';

?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>



<div class="row">
<h3><?php echo $data['title']; ?></h3>
</div>

<?php

        echo '<div class="row">';
echo '  <div>';
echo ($data["image"]!='-') ? '<img class="img-rounded" src="'. $data['image'] . '" />' : 'Εξώφυλλο μη διαθέσιμο.';
echo '  </div>';
echo '</div>';

?>

<div class="row">
<div>Συγγραφέας: <?php echo $data['author']; ?></div>
</div>

<div class="row">
<div>Εκδότης: <?php echo $data['publisher']; ?></div>
</div>

<div class="row">
<div><?php print how_many_favored_this_book($id); ?>χρήστες έχουν προσθέσει το βιβλίο αυτό στη λίστα αγαπημένων τους.</div>
</div>

<div class="row">
<div>Ο μέσος όρος βαθμολογίας αυτού του βιβλίου είναι: <?php echo get_book_average_mark($id)?number_format(get_book_average_mark($id),2):'-'; ?></div>
</div>

<?php
      if (is_user_logged_in()) {
get_rating_block($id);
}
?>

<div class="row">
<div>Ενέργειες: <?php get_book_inside_actions($data['id']); ?></div>
</div>

<div class="row">
```

```
<h3>Άλλα βιβλία που δανείστηκαν χρήστες που δανείστηκαν το συγκεκριμένο</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php get_similar_books($id); ?>
</tbody>
</table>
</div>

</div>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

## 2.3 login.php – Σύνδεση χρήστη

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

/*** check if the users is already logged in ***/
if (isset( $_SESSION['user_id'] )) {
$message = 'Είστε ήδη συνδεδεμένος, παρακαλείσθε να αποσυνδεθείτε εάν θέλετε να συνδεθείτε σαν διαφορετικός χρήστης.';
set_error_message($message, 'index.php', 'danger');
return;
}
?>

<!DOCTYPE html>
<html lang="el">
<head>
<title>Σύνδεση χρήστη</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
```

```
<h1>Σύνδεση χρήστη</h1>
<form action="login_submit.php" method="post">
<fieldset>
<p>
<label for="mail">E-mail</label>
<input type="text" id="mail" name="mail" value="" maxlength="20" />
</p>
<p>
<label for="password">Password</label>
<input type="password" id="password" name="password" value="" maxlength="20" />
</p>
<p>
<input type="submit" value="Σύνδεση" />
</p>
</fieldset>
</form>
</div>
<?php get_footer(); ?>
</div>
</body>
</html>
```



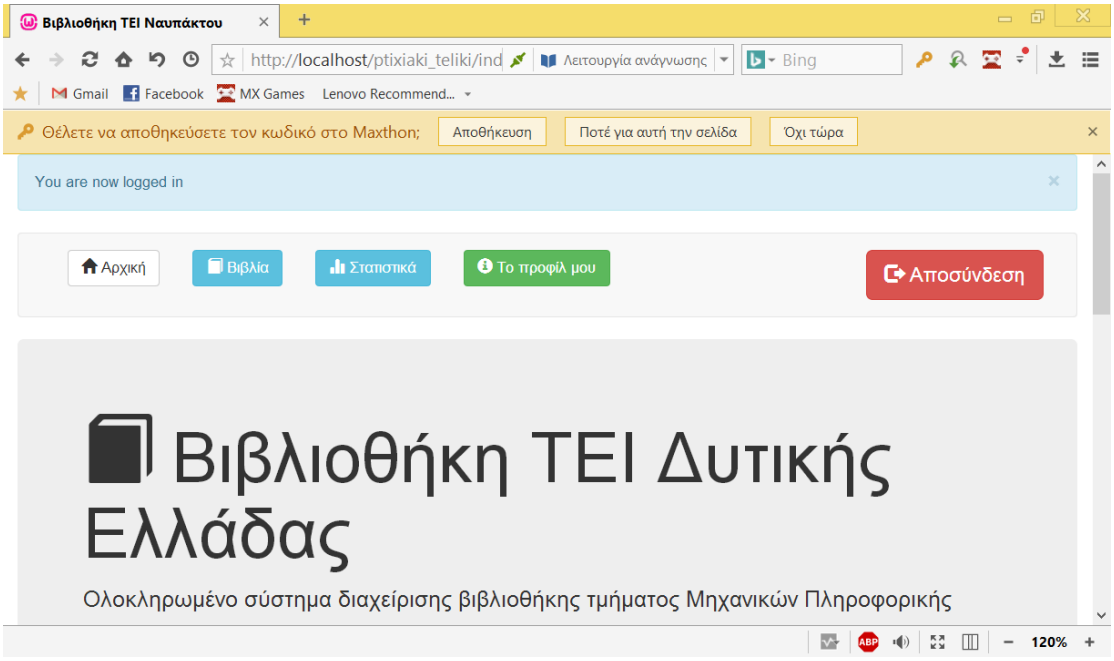

## 2.4 login_submit.php

Αφού ο χρήστης πληκτρολογήσει username και password και πατήσει το κουμπί «σύνδεση», γίνεται κλήση στην παρακάτω σελίδα που ελέγχει εάν όντως υπάρχει τέτοιος χρήστης στη βάση:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

/*** check if the users is already logged in ***/
if (isset( $_SESSION['user_id'] )) {
```

```
$message = 'Είστε ήδη συνδεδεμένος.';
set_error_message($message, 'index.php', 'danger');
}

/*** check that both the username, password have been submitted ***/
if(!isset( $_POST['mail'], $_POST['password'])) {
$message = 'Παρακαλούμε εισάγετε e-mail και κωδικό.';
set_error_message($message, 'index.php', 'danger');
}

else {
try {
/*** if we are here the data is valid and we can insert it into database ***/
$mail = filter_var($_POST['mail'], FILTER_SANITIZE_STRING);
$password = filter_var($_POST['password'], FILTER_SANITIZE_STRING);

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("SELECT id, mail, password, isadmin, deactivated FROM users
                    WHERE mail = :mail AND password = :password");

/*** bind the parameters ***/
$stmt->bindParam(':mail', $mail, PDO::PARAM_STR);
$stmt->bindParam(':password', $password, PDO::PARAM_STR, 40);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();
$result1 = $result[0];
$user_id = $result1['id'];
$user_isadmin = $result1['isadmin'];

/*** if we have no result then fail boat ***/
if ($user_id == FALSE) {
$message = 'Αποτυχία σύνδεσης.';
set_error_message($message, 'index.php', 'danger');
return;
}
/*** if we do have a result, all is well ***/
else {

if ($result1['deactivated'] == 1) {
$message = 'Ο λογαριασμός σας έχει απενεργοποιηθεί, παρακαλούμε επικοινωνήστε με τον
διαχειριστή.';
set_error_message($message, 'index.php', 'danger');
return;
}

/*** set the session user_id variable ***/
$_SESSION['user_id'] = $user_id;
$_SESSION['user_isadmin'] = $user_isadmin;

/*** tell the user we are logged in ***/
$message = 'You are now logged in';
set_error_message($message, 'index.php', 'info');
return;
}
```

```
  }

catch(Exception $e) {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων';
set_error_message($message, 'index.php', 'danger');
}
}
```




## 2.5 logout.php

Για την αποσύνδεση του χρήστη από το σύστημα:

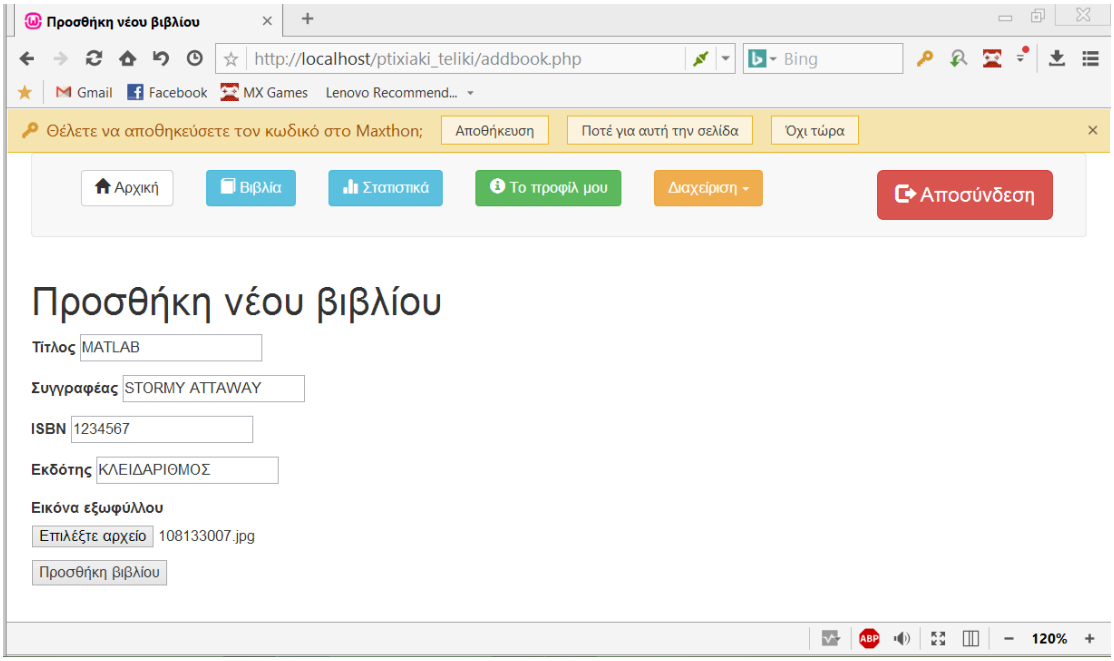
```
<?php
// Begin the session
session_start();
include_once 'helpers.php';

// Unset all of the session variables.
session_unset();

// Destroy the session.
session_destroy();

$message = "Επιτυχής αποσύνδεση.";
set_error_message($message, 'index.php');
```

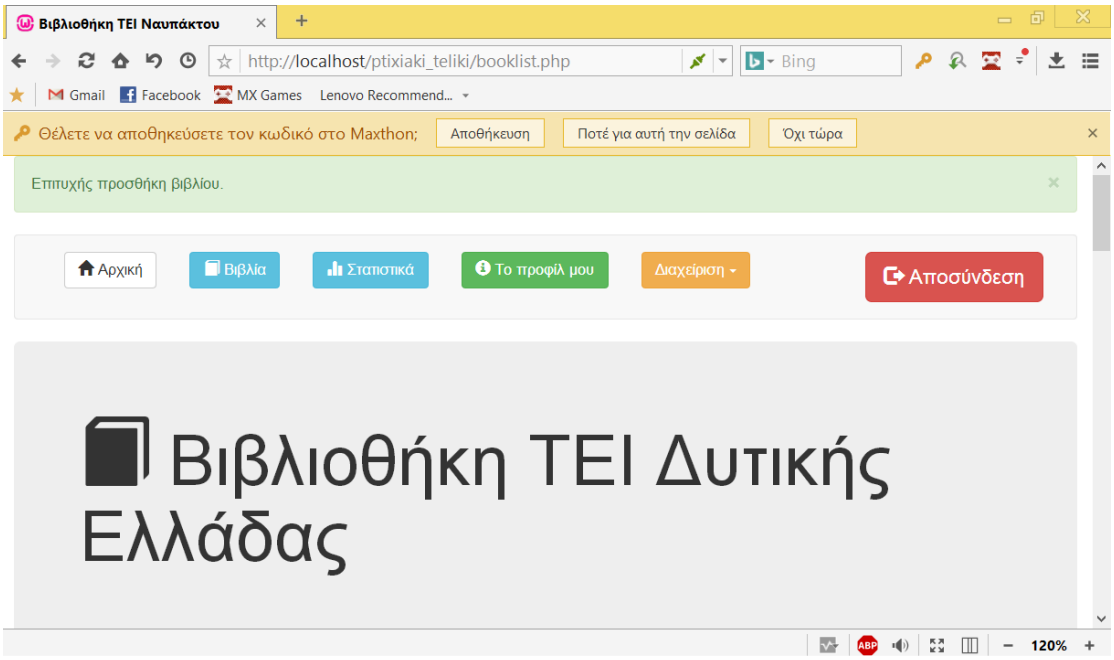
## 2.6 activate_book.php

Ενεργοποίηση βιβλίου από τον διαχειριστή:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';
```

```php
if (!is_user_logged_in()) {
set_error_message('Παρακαλείσθε να συνδεθείτε.');
return;
}

if (!is_user_admin()) {
set_error_message('Μόνο οι διαχειριστές έχουν δικαίωμα διαχείρισης των βιβλίων.');
return;
}

/*** first check a user id has been provided ***/
if(!isset($_GET['bid'])) {
$message = 'Θα πρέπει να επιλέξετε ένα βιβλίο για να ενεργοποιήσετε.';
set_error_message($message);
return;
}

$bid = $_GET['bid'];

try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE books SET deactivated = 0 WHERE id = (:id )");

/*** bind the parameters ***/
$stmt->bindParam(':id', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Το βιβλίο έχει ενεργοποιηθεί επιτυχώς!';
set_admin_message($message, 'index.php');
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Caught exception: ' . $e->getMessage() . "\n";
}
```



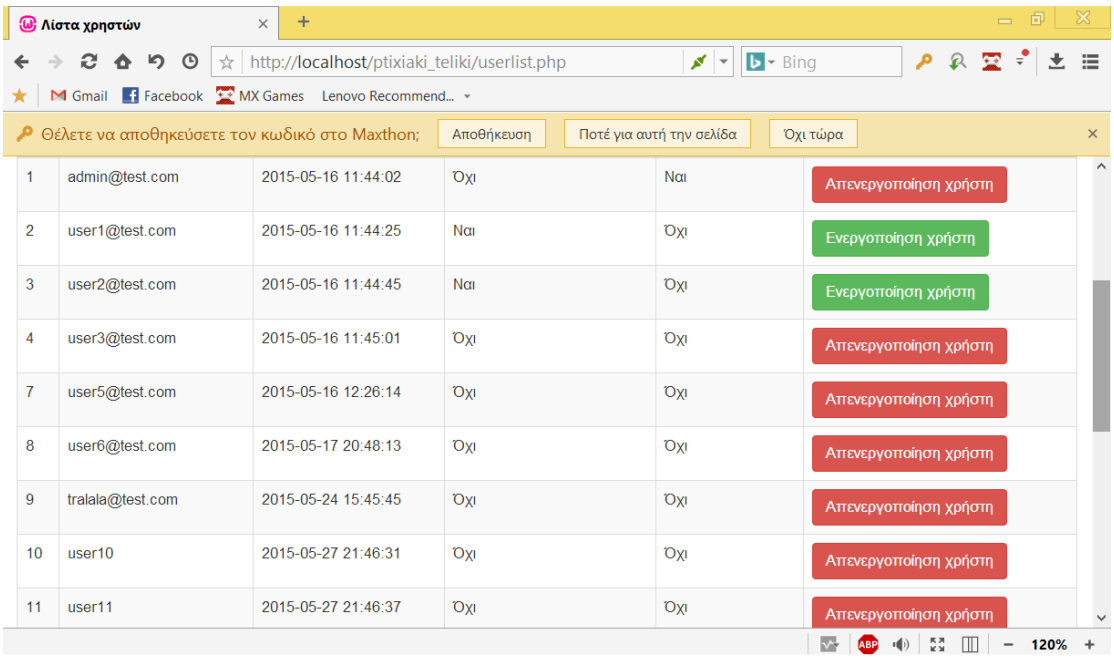




## 2.7 activate_user.php

Ενεργοποίηση χρήστη από το διαχειριστή:

```php
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Παρακαλείσθε να συνδεθείτε.');
return;
}

if (!is_user_admin()) {
```

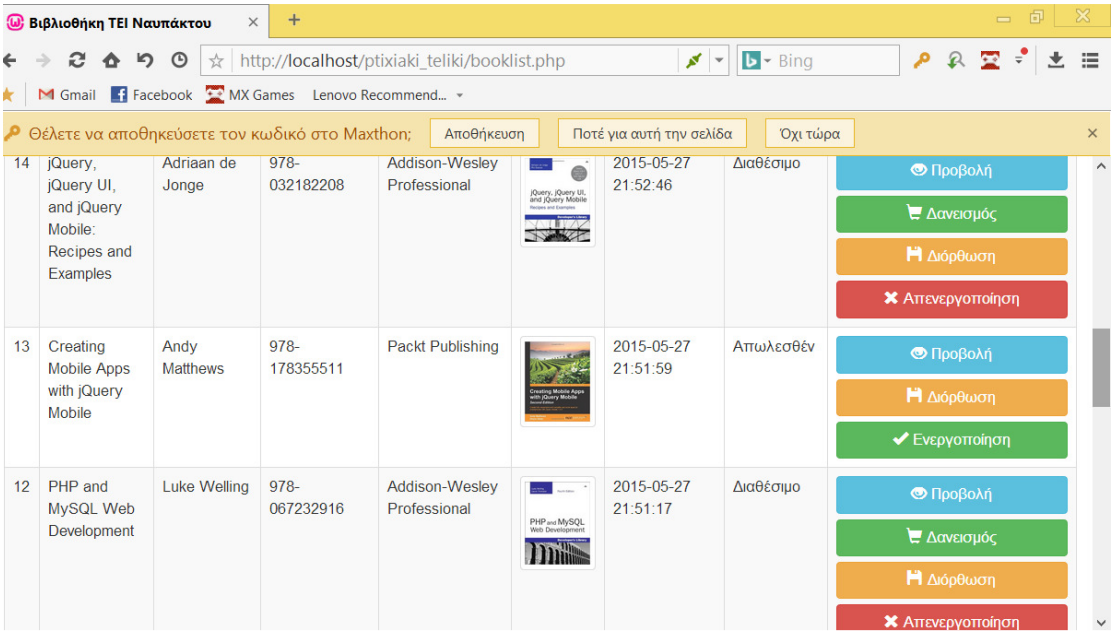
```
set_error_message('Μόνο οι διαχειριστές έχουν πρόσβαση στη λίστα χρηστών.');
return;
}

/*** first check a user id has been provided ***/
if(!isset($_GET['uid'])) {
$message = 'Θα πρέπει να επιλέξετε έναν χρήστη για να ενεργοποιήσετε.';
set_error_message($message);
return;
}

$uid = $_GET['uid'];

try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE users SET deactivated = 0 WHERE id = (:id )");

/*** bind the parameters ***/
$stmt->bindParam(':id', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Ο χρήστης έχει ενεργοποιηθεί επιτυχώς!';
set_admin_message($message, 'userlist.php');
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Caught exception: ' . $e->getMessage() . "\n";
}
```



## 2.8 deactivate_book.php

Απενεργοποίηση βιβλίου από το διαχειριστή:

```php
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Παρακαλείσθε να συνδεθείτε.');
return;
}

if (!is_user_admin()) {
set_error_message('Μόνο οι διαχειριστές έχουν πρόσβαση στη διαχείριση βιβλίων.');
return;
}

/*** first check a book id has been provided ***/
if(!isset($_GET['bid'])) {
$message = 'Θα πρέπει να επιλέξετε ένα βιβλίο για να απενεργοποιήσετε.';
set_error_message($message);
return;
}

$bid = $_GET['bid'];

try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE books SET deactivated = 1 WHERE id = (:id )");

/*** bind the parameters ***/
$stmt->bindParam(':id', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Το βιβλίο έχει απενεργοποιηθεί επιτυχώς!';
set_admin_message($message, 'index.php');
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Caught exception: ' . $e->getMessage() . "\n";
}
```

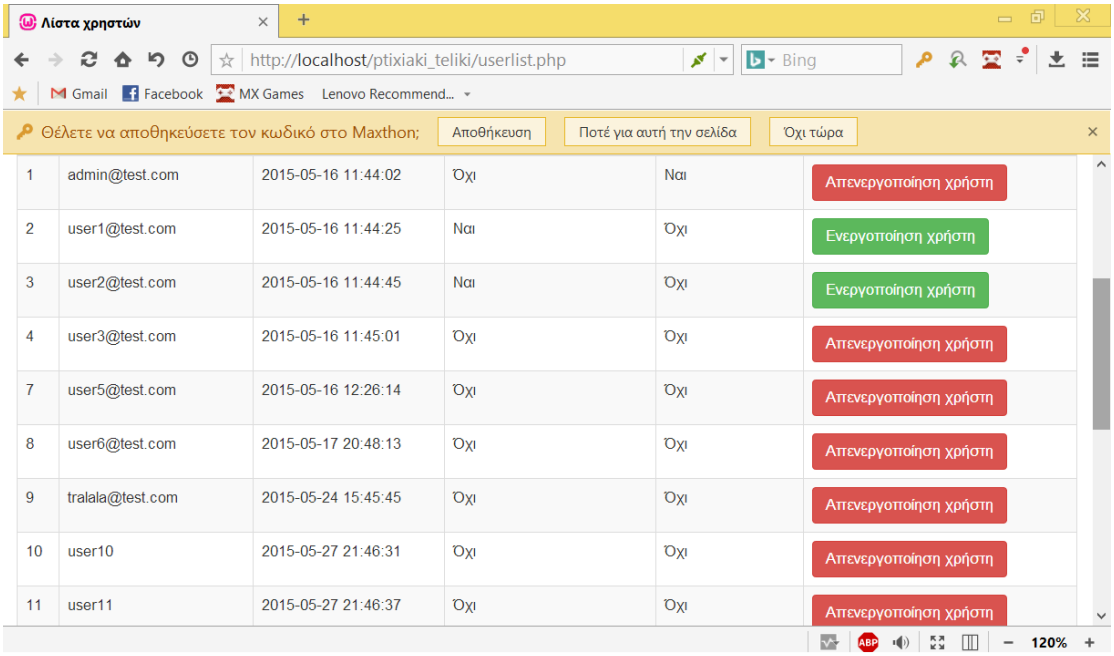



## 2.9 deactivate_user.php

Απενεργοποίηση χρήστη από τον διαχειριστή:

```php
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Παρακαλείσθε να συνδεθείτε.');
return;
}

if (!is_user_admin()) {
set_error_message('Μόνο οι διαχειριστές έχουν πρόσβαση στη λίστα χρηστών.');
return;
}

/*** first check a user id has been provided ***/
if(!isset($_GET['uid'])) {
$message = 'Θα πρέπει να επιλέξετε έναν χρήστη για να απενεργοποιήσετε.';
set_error_message($message);
return;
}

$uid = $_GET['uid'];

if($uid == 1) {
$message = 'Δεν επιτρέπεται να απενεργοποιήσετε τον διαχειριστή!';
set_admin_message($message, 'userlist.php');
return;
}

try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE users SET deactivated = 1 WHERE id = (:id )");
```

```php
/*** bind the parameters ***/
$stmt->bindParam(':id', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Ο χρήστης έχει απενεργοποιηθεί επιτυχώς!';
set_admin_message($message, 'userlist.php');
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Caught exception: ' . $e->getMessage() . "\n";
}
```

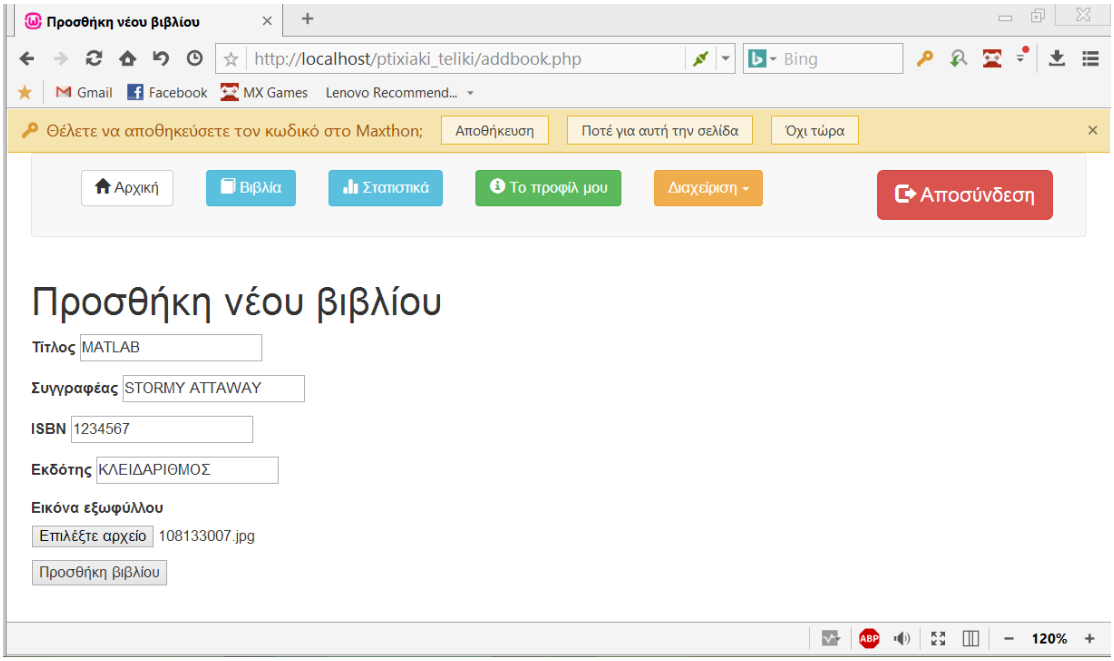



## 2.10 addbook.php

Προσθήκη βιβλίου:

```php
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if(!is_user_admin()) {
set_error_message('Μόνο οι διαχειριστές μπορούν να προσθέσουν νέα βιβλία.');
return;
}
?>

<!DOCTYPE html>
<html lang="el">
<head>
<title>Προσθήκη νέου βιβλίου</title>
```

```
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'messages.php'; ?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
<h1>Προσθήκη νέου βιβλίου</h1>
<form action="addbook_submit.php" method="post" enctype="multipart/form-data">
<fieldset>
<p>
<label for="title">Τίτλος</label>
<input type="text" id="title" name="title" value="" />
</p>
<p>
<label for="author">Συγγραφέας</label>
<input type="text" id="author" name="author" value="" />
</p>
<p>
<label for="isbn">ISBN</label>
<input type="text" id="isbn" name="isbn" value="" />
</p>
<p>
<label for="publisher">Εκδότης</label>
<input type="text" id="publisher" name="publisher" value="" />
</p>
<p>
<label for="cover">Εικόνα εξωφύλλου</label>
<input type="file" name="cover" id="cover">
</p>

<p>
<input type="submit" value="Προσθήκη βιβλίου" />
</p>
</fieldset>
</form>
</div>
</div>
</body>
</html>
```




## 2.11 addbook_submit.php

Έχοντας συμπληρώσει όλα τα απαραίτητα πεδία, ο διαχειριστής πατάει "προσθήκη βιβλίου" και καλείται η ακόλουθη σελίδα:

```php
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

/*** first check that both the username, password and form token have been sent ***/
if(!isset($_POST['title']) || !isset($_POST['author']) || !isset($_POST['publisher']) ||
!isset($_POST['isbn']) ||
  !($_POST['title']) || !($_POST['author']) || !($_POST['publisher']) || !($_POST['isbn'])) {
set_error_message('Παρακαλείσθε να συμπληρώσετε όλα τα απαραίτητα πεδία.', 'addbook.php');
return;
}
else {
/*** if we are here the data is valid and we can insert it into database ***/
$title = filter_var($_POST['title'], FILTER_SANITIZE_STRING);
$author = filter_var($_POST['author'], FILTER_SANITIZE_STRING);
$isbn = filter_var($_POST['isbn'], FILTER_SANITIZE_STRING);
$publisher = filter_var($_POST['publisher'], FILTER_SANITIZE_STRING);

if ($_FILES["cover"]["name"] != '') {
$target_dir = "covers/";
$target_file = $target_dir . basename($_FILES["cover"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
$check = getimagesize($_FILES["cover"]["tmp_name"]);
if($check !== false) {
$uploadOk = 1;
} else {
$message = 'Το αρχείο που επιλέξατε δεν είναι συμβατή εικόνα.';
```

```
$uploadOk = 0;
set_error_message($message);
}
    }
// Check if file already exists
if (file_exists($target_file)) {
$message = 'Υπάρχει ήδη αρχείο με αυτό το όνομα. Παρακαλείσθε να μετονομάσετε το αρχείο.';
$uploadOk = 0;
set_error_message($message);
}
// Allow certain file formats
if($imageFileType != "jpg" &&$imageFileType != "png" &&$imageFileType != "jpeg"
&&$imageFileType != "gif" ) {
$message = 'Μόνο αρχεία τύπου JPG, JPEG, PNG & GIF επιτρέπονται.';
$uploadOk = 0;
set_error_message($message);
}

if ($uploadOk == 0) {
$message = "Σφάλμα στην αποθήκευση του αρχείου.";
set_error_message($message);
// if everything is ok, try to upload file
} else {
if (move_uploaded_file($_FILES["cover"]["tmp_name"], $target_file)) {
$message = "Το αρχείο " . basename( $_FILES["cover"]["name"]) . " έχει αποθηκευτεί.";
} else {
$message = "Σφάλμα στην αποθήκευση του αρχείου.";
set_error_message($message);
}
    }
  }
else {
$target_file = '-';
}

include_once 'classes/Database.php';
try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("
        INSERT
          INTO books (title, author, isbn, publisher, image )
          VALUES (:title, :author, :isbn, :publisher, :image )
    ");

/*** bind the parameters ***/
$stmt->bindParam(':title', $title, PDO::PARAM_STR);
$stmt->bindParam(':author', $author, PDO::PARAM_STR);
$stmt->bindParam(':isbn', $isbn, PDO::PARAM_STR);
$stmt->bindParam(':publisher', $publisher, PDO::PARAM_STR);
$stmt->bindParam(':image', $target_file, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Επιτυχής προσθήκη βιβλίου.';
set_error_message($message);
```

```
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων."';
}
}
```

## 2.12 addtofavorites.php

Όταν ένας χρήστης επιλέξει του κουμπί με το αστεράκι, η διαδικασία προσθαφαίρεσης ενός βιβλίου στη λίστα αγαπημένων του ξεκινάει.
Το σύστημα είναι έξυπνο, καθώς πρώτα αναζητά εάν το συγκεκριμένο βιβλίο είναι ήδη στη λίστα αγαπημένων του συγκεκριμένου χρήστη.Εάν όχι, το προσθέτει.Εάν ναι, το αφαιρεί.

Έτσι, οι διαδικασίες προσθήκης και αφαίρεσης στη λίστα αγαπημένων ενοποιούνται σε μία:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Παρακαλείστε θα συνδεθείτε.');
return;
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['bid'])) {
$message = 'Θα πρέπει να επιλέξετε ένα βιβλίο για να το προσθέσετε στα αγαπημένα σας.';
set_error_message($message);
return;
}

$bid = $_GET['bid'];
$uid = $_SESSION['user_id'];

$is_this_book_in_favorites = is_this_book_in_favorites($bid, $uid);

if ($is_this_book_in_favorites) {
// book is already in favorites, so remove it
try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("DELETE FROM favorites WHERE id = (:bid )");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $is_this_book_in_favorites, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
```

```
$message = 'Το βιβλίο αφαιρέθηκε από τα αγαπημένα σας.';
set_error_message($message, 'view.php?id=' . $bid);
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων.';
}
}
else {
// add the book to favorites
try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("INSERT INTO favorites (bid, uid ) VALUES (:bid, :uid )");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR, 40);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Το βιβλίο προστέθηκε στα αγαπημένα σας.';
set_error_message($message, 'view.php?id=' . $bid);
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων.';
}
}
```

## 2.13 adduser.php

Προσθήκη χρήστη, είτε από το διαχειριστή είτε μέσω ατομικής εγγραφής.

Όπως αναλύθηκε και πιο πάνω, εφόσον αναμένεται το σύστημα να διασυνδέεται με το κεντρικό σύστημα βιβλιοθήκης του ΤΕΙ, η υλοποίηση της εγγραφής έγινε με απλό τρόπο, αφού κανονικά η εγγραφή θα είναι ευθύνη του κεντρικού NOC.

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';
?>

<!DOCTYPE html>
<html lang="el">
<head>
<title>Προσθήκη νέου χρήστη</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
```

```
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'messages.php'; ?>
</div>
<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
<h1>Προσθήκη νέου χρήστη</h1>
<form action="adduser_submit.php" method="post">
<fieldset>
<p>
<label for="mail">E-mail</label>
<input type="text" id="mail" name="mail" value="" maxlength="20" />
</p>
<p>
<label for="password">Password</label>
<input type="password" id="password" name="password" value="" maxlength="20" />
</p>
<p>
<input type="submit" value="Εγγραφή" />
</p>
</fieldset>
</form>
</div>
<?php get_footer(); ?>
</div>
</body>
</html>
```

## 2.14 adduser_submit.php

Μετά τη συμπλήρωση του e-mail και του επιθυμητού password καλείται η ακόλουθη σελίδα που πραγματοποιεί την εγγραφή:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

/*** first check that both the username, password and form token have been sent ***/
if(!isset( $_POST['mail'], $_POST['password']) || !$_POST['mail'] || !$_POST['password']) {
$message = 'Παρακαλούμε εισάγετε ένα έγκυρο e-mail και έναν έγκυρο κωδικό.';
set_error_message($message, 'adduser.php');
}

else {
/*** if we are here the data is valid and we can insert it into database ***/
$mail = filter_var($_POST['mail'], FILTER_SANITIZE_STRING);
$password = filter_var($_POST['password'], FILTER_SANITIZE_STRING);

try {
```

```
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("INSERT INTO users (mail, password ) VALUES (:mail, :password )");

/*** bind the parameters ***/
$stmt->bindParam(':mail', $mail, PDO::PARAM_STR);
$stmt->bindParam(':password', $password, PDO::PARAM_STR, 40);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Η εγγραφή ολοκληρώθηκε.';
set_error_message($message);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Υπάρχει ήδη χρήστης με αυτό το e-mail!';
set_error_message($message);
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων.';
set_error_message($message);
}
  }
}
```

## 2.15 admin.php – Σελίδα διαχείρισης δανεισμών

Η σελίδα αυτή αποτελεί το “στρατηγείο” του διαχειριστή, καθώς από αυτήν ελέγχει όλα τα βιβλία που έχουν κρατηθεί, όσα είναι δανεισμένα, και πραγματοποιεί τις διαχειριστικές ενέργειες έγκρισης, απόρριψης και επιστροφής:

```
<?php
/*** begin the session ***/
session_start();
include_once 'helpers.php';
if (!is_user_admin()) {
set_error_message('Only administrators can access admin page.');
return;
}

?>
<!DOCTYPE html>
<html lang="el">
<head>
<title>Διαχείριση δανεισμών</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
```

```
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'admin_messages.php'; ?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
<h1>Διαχείριση συστήματος δανεισμών</h1>
</div>

<div class="row">
<h2>Αιτήσεις κράτησης</h2>

<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Χρήστης</th>
<th>Ημ/νία κράτησης</th>
<th>Επιλογές</th>
</tr>
</thead>
<tbody>
<?php
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
  books.title,
  books.author,
  books.isbn,
  users.mail,
  lend.created
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 0
ORDER BY lend.created ASC
        ';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';

echo '<td>';
echo is_this_book_in_favorites($row['id'], $_SESSION['user_id']) ? '<span class="glyphicon
glyphicon-star"></span> :'';
echo $row['title'];
echo '</td>';

echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
```

```
echo '<td>'. $row['mail'] . '</td>';
echo '<td>'. $row['created'] . '</td>';
echo '<td width=250>';
get_admin_actions_asked($row['id']);
echo '</td>';
echo '</tr>';
}
        Database::disconnect();
?>
</tbody>
</table>
</div>

<div class="row">
<h2>Κρατημένα βιβλία (αναμένεται δανεισμός)</h2>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Χρήστης</th>
<th>Ημ/νία αποδοχής κράτησης</th>
<th>Επιλογές</th>
</tr>
</thead>
<tbody>
<?php
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
  books.title,
  books.author,
  books.isbn,
  users.mail,
  lend.approved
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 1
ORDER BY lend.approved ASC
      ';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';

echo '<td>';
echo is_this_book_in_favorites($row['id'], $_SESSION['user_id']) ? '<span class="glyphicon
glyphicon-star"></span> :'';
echo $row['title'] . '</td>';

echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['mail'] . '</td>';
echo '<td>'. $row['approved'] . '</td>';
echo '<td width=250>';
get_admin_actions_approved($row['id']);
echo '</td>';
```

```
echo '</tr>';
}
    Database::disconnect();
?>
</tbody>
</table>
</div>

<div class="row">
<h2>Δανεισμένα βιβλία</h2>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Χρήστης</th>
<th>Ημ/νία δανεισμού</th>
<th>Επιλογές</th>
</tr>
</thead>
<tbody>
<?php
$pdo = Database::connect();
$sql = '
  SELECT
    lend.id,
    books.title,
    books.author,
    books.isbn,
    users.mail,
    lend.lent
  FROM books
  JOIN lend on lend.bid = books.id
  JOIN users on lend.uid = users.id
  WHERE lend.status = 2
  ORDER BY lend.lent ASC
  ';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';

echo '<td>';
echo is_this_book_in_favorites($row['id'], $_SESSION['user_id']) ? '<span class="glyphicon
glyphicon-star"></span> :'';
echo $row['title'] . '</td>';

echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['mail'] . '</td>';
echo '<td>'. $row['lent'] . '</td>';
echo '<td width=250>';
get_admin_actions_lent($row['id']);
echo '</td>';
echo '</tr>';
}
    Database::disconnect();
?>
</tbody>
```

```
</table>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

## 2.16 approve_ask.php – Έγκριση κράτησης βιβλίου

Με το πάτημα του αντίστοιχου κουμπιού, φορτώνεται η ακόλουθη σελίδα.Ο διαχειριστής έχει επιλέξει να αποδεχτεί την κράτηση. Ο χρήστης ενημερώνεται ότι το βιβλίο του είναι διαθέσιμο και μπορεί να περάσει από τη βιβλιοθήκη για να το δανειστεί:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_admin()) {
set_error_message('Only administrators can access admin page.');
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['lid'])) {
$message = 'Σφάλμα.';
set_admin_message($message);
return;
}

$lid = $_GET['lid'];
$status = '1';

try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE lend SET status = (:status ), approved = now() WHERE id = (:lid
)");

/*** bind the parameters ***/
$stmt->bindParam(':lid', $lid, PDO::PARAM_STR);
$stmt->bindParam(':status', $status, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Αποδοχή κράτησης - αναμένεται ο δανεισμός από το φοιτητή';
set_admin_message($message);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
```

```
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
}
```

## 2.17 askbook.php – Αίτημα κράτησης βιβλίου

Με το πάτημα του αντίστοιχου κουμπιού ο χρήστης καταθέτει αίτημα για την κράτηση του βιβλίου:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Please login first.');
return;
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['bid'])) {
$message = 'You have to select a book to ask for lend.';
set_error_message($message);
return;
}

$bid = $_GET['bid'];
$uid = $_SESSION['user_id'];
$status = 0;

try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("
        INSERT
        INTO lend (uid, bid, status )
        VALUES (:uid, :bid, :status )
                ");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);
$stmt->bindParam(':status', $status, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Book asked for lend';
set_error_message($message);
```

```
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
}
```

## 2.18 deny_ask.php – Απόρριψη κράτησης

Ο διαχειριστής έχει το δικαίωμα να απορρίψει την κράτηση ενός βιβλίου. Με το πάτημα του αντίστοιχου κουμπιού φορτώνεται η αντίστοιχη σελίδα:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_admin()) {
set_error_message('Only administrators can access admin page.');
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['lid'])) {
$message = 'Σφάλμα.';
set_admin_message($message);
return;
}

$lid = $_GET['lid'];


try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("DELETE FROM lend WHERE id = (:lid )");

/*** bind the parameters ***/
$stmt->bindParam(':lid', $lid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Κράτηση απορρίφθηκε';
set_admin_message($message);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
```

```
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
}
```

## 2.19 lendbook.php – Σύναψη δανεισμού

Ο χρήστης έχει προσέλθει στο χώρο της βιβλιοθήκης για να δανειστεί το βιβλίο. Μαρκάρεται το βιβλίο ως δανεισμένο και ενεργοποιείται η σύναψη δανεισμού:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_admin()) {
set_error_message('Only administrators can access admin page.');
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['lid'])) {
$message = 'Σφάλμα.';
set_admin_message($message);
return;
}

$lid = $_GET['lid'];
$status = '2';

try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE lend SET status = (:status ), lent = now() WHERE id = (:lid )");

/*** bind the parameters ***/
$stmt->bindParam(':lid', $lid, PDO::PARAM_STR);
$stmt->bindParam(':status', $status, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Ολοκλήρωση δανεισμού βιβλίου';
set_admin_message($message);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
```

```
$message = 'We are unable to process your request. Please try again later"';
}
}
```

## 2.20 markbook.php – Βαθμολογία βιβλίου

Με την επιλογή βαθμολογίας και το πάτημα του αντίστοιχου κουμπιού, καταχωρίζεται μια βαθμολογία του χρήστη για ένα συγκεκριμένο βιβλίο. Το σύστημα είναι σχεδιασμένο ώστε, εάν δεν υπάρχει βαθμολογία να την αποθηκεύει, ενώ αν υπάρχει ήδη να την τροποποιεί:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_logged_in()) {
set_error_message('Please login first.');
return;
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_POST['bid'])) {
$message = 'You have to select a book to add to mark.';
set_error_message($message);
return;
}

$bid = $_POST['bid'];
$mark = $_POST['mark'];
$uid = $_SESSION['user_id'];


// Create a strong safety net for the mark number.
$safety_array = array();
foreach (range(1, 10) as $number) {
$safety_array[] = $number;
}
if (!in_array($mark, $safety_array)) {
$message = 'Choose a mark between 1 and 10.';
set_error_message($message);
return;
}


$has_user_marked_this_book = has_user_marked_this_book($bid, $uid);


if ($has_user_marked_this_book) {
// book is already marked, update the mark
try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE rates SET rate = (:rate ) WHERE id = (:id )");
```

```
/*** bind the parameters ***/
$stmt->bindParam(':rate', $mark, PDO::PARAM_STR);
$stmt->bindParam(':id', $has_user_marked_this_book, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Book marked succesfully!';
set_error_message($message, 'view.php?id=' . $bid);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
  }
}
else {
// book not marked, add a new mark
try {

$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("INSERT INTO rates (bid, uid, rate ) VALUES (:bid, :uid, :rate )");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);
$stmt->bindParam(':rate', $mark, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Book marked succesfully!';
set_error_message($message, 'view.php?id=' . $bid);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
  }
}
```

## 2.21 profile.php – Σελίδα προφίλ του χρήστη

Στη σελίδα αυτή ο χρήστης πληροφορείται για το ιστορικό των δανεισμών του, για τα αγαπημένα του βιβλία, και για τις βαθμολογίες του:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if(!isset($_SESSION['user_id'])) {
$message = 'You must be logged in to access a profile page';
set_error_message($message);
return;
}
?>

<!DOCTYPE html>
<html lang="el">
<head>
<title>Το προφίλ μου</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">


<div class="row">
<?php

  include_once 'messages.php';

?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
<h3>Το προφίλ μου</h3>
</div>

<div class="row">
<h3>Βιβλία σε κράτηση</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Ημ/νία κράτησης</th>
</tr>
</thead>
<tbody>
```

```
<?php
get_user_asked($_SESSION['user_id']);
?>
</tbody>
</table>

</div>
<div class="row">
<h3>Δανεισμένα βιβλία</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Ημ/νία δανεισμού</th>
</tr>
</thead>
<tbody>
<?php

get_user_currently_lent($_SESSION['user_id']);

?>
</tbody>
</table>
</div>

<div class="row">
<h3>Ιστορικό δανεισμών</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Ημ/νία δανεισμού</th>
<th>Ημ/νία επιστροφής</th>
</tr>
</thead>
<tbody>
<?php
get_user_history($_SESSION['user_id']);
?>
</tbody>
</table>
</div>

<div class="row">
<h3>Τα αγαπημένα μου βιβλία</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Αφαίρεση από αγαπημένα</th>
</tr>
</thead>
<tbody>
```

```
<?php

get_user_favorites($_SESSION['user_id']);

?>
</tbody>
</table>
</div>
<div class="row">
<h3>Οι βαθμολογίες μου</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Βαθμολογία</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php
get_user_marks($_SESSION['user_id']);

?>
</tbody>
</table>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

## 2.22 returnbook.php – Επιστροφή βιβλίου

Με το πάτημα του αντίστοιχου κουμπιού από τον διαχειριστή, καταχωρίζεται η επιστροφή ενός βιβλίου:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if (!is_user_admin()) {
set_error_message('Only administrators can access admin page.');
}

/*** first check a book id has been asked for favorites ***/
if(!isset($_GET['lid'])) {
$message = 'Σφάλμα.';
set_admin_message($message);
return;
}

$lid = $_GET['lid'];
$status = '3';

try {
```

```
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("UPDATE lend SET status = (:status ), returned = now() WHERE id = (:lid )");

/*** bind the parameters ***/
$stmt->bindParam(':lid', $lid, PDO::PARAM_STR);
$stmt->bindParam(':status', $status, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Βιβλίο επεστράφη';
set_admin_message($message);
}
catch(Exception $e)   {
/*** check if the username already exists ***/
if( $e->getCode() == 23000) {
$message = 'Username already exists';
}
else {
/*** if we are here, something has gone wrong with the database ***/
$message = 'We are unable to process your request. Please try again later"';
}
}
```

## 2.23 statistics.php – Σελίδα στατιστικών

Σελίδα στατιστικών με τα πιο διάσημα βιβλία. Η σελίδα αυτή είναι προσβάσιμη από όλους:

```
<?php
/*** begin the session ***/
session_start();
include_once 'helpers.php';
?>
<!DOCTYPE html>
<html lang="el">
<head>
<title>Στατιστικά Βιβλίων</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">
<h3>Statistics</h3>
```

```
</div>

<div class="row">
<h3>Τα βιβλία με τις 5 υψηλότερες βαθμολογίες</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Μ.Ο. Βαθμολογίας</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php
get_most_rated_books();
?>
</tbody>
</table>
</div>

<div class="row">
<h3>Τα 5 πιο αγαπημένα βιβλία</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Αγαπημένο (αριθμός χρηστών)</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php
get_most_favored_books();
?>
</tbody>
</table>
</div>

<div class="row">
<h3>Τα 5 περισσότερο δανεισμένα βιβλία</h3>
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Δανεισμοί</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php
get_most_lent_books();
?>
</tbody>
</table>
```

```
</div>

<?php get_footer(); ?>

</div><!-- /container -->
</body>
</html>
```

## 2.24 updatebook.php – Τροποποίηση στοιχείων βιβλίου

Από τη σελίδα αυτή ο διαχειριστής μπορεί να αλλάξει κάποια στοιχεία του βιβλίου, εάν βρεθεί πως έγινε λάθος στην καταχώριση:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

if(!is_user_admin()) {
set_error_message('Μόνο οι διαχειριστές μπορούν να διορθώσουν βιβλία.');
return;
}
?>

<!DOCTYPE html>
<html lang="el">
<head>
<title>Προσθήκη νέου βιβλίου</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'messages.php'; ?>
</div>

<?php
$bid = $_GET['bid'];
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "SELECT * FROM books where id = ?";
$q = $pdo->prepare($sql);
$q->execute(array($bid));
$data = $q->fetch(PDO::FETCH_ASSOC);
Database::disconnect();
?>

<div class="row">
<?php get_navbar(); ?>
</div>
```

```
<div class="row">
<h1>Διόρθωση στοιχείων βιβλίου</h1>
<form action="updatebook_submit.php" method="post" enctype="multipart/form-data">
<fieldset>
<p>
<label for="title">Τίτλος</label>
<input type="text" id="title" name="title" value="<?php print $data['title']; ?>" />
</p>
<p>
<label for="author">Συγγραφέας</label>
<input type="text" id="author" name="author" value="<?php print $data['author']; ?>" />
</p>
<p>
<label for="isbn">ISBN</label>
<input type="text" id="isbn" name="isbn" value="<?php print $data['isbn']; ?>" />
</p>
<p>
<label for="publisher">Εκδότης</label>
<input type="text" id="publisher" name="publisher" value="<?php print $data['publisher']; ?>" />
</p>

<p>
<input type="hidden" id="bid" name="bid" value="<?php print $bid; ?>" />
<input type="submit" value="Αποθήκευση βιβλίου" />
</p>
</fieldset>
</form>
</div>
</div>
</body>
</html>
```

## 2.25 updatebook_submit.php

Μετά την τροποποίηση των στοιχείων του βιβλίου, τα νέα στοιχεία αποθηκεύονται στη βάση:

```
<?php
/*** begin our session ***/
session_start();
include_once 'helpers.php';

/*** first check that both the username, password and form token have been sent ***/
if(!isset($_POST['title']) || !isset($_POST['author']) || !isset($_POST['publisher']) ||
!isset($_POST['isbn']) ||
  !($_POST['title']) || !($_POST['author']) || !($_POST['publisher']) || !($_POST['isbn'])) {
set_error_message('Παρακαλείσθε να συμπληρώσετε όλα τα απαραίτητα πεδία.', 'addbook.php');
return;
}
else {
/*** if we are here the data is valid and we can insert it into database ***/
$title = filter_var($_POST['title'], FILTER_SANITIZE_STRING);
$author = filter_var($_POST['author'], FILTER_SANITIZE_STRING);
$isbn = filter_var($_POST['isbn'], FILTER_SANITIZE_STRING);
$publisher = filter_var($_POST['publisher'], FILTER_SANITIZE_STRING);
$bid = $_POST['bid'];
```

```
try {
$pdo = Database::connect();
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $pdo->prepare("
UPDATE books
  SET
    title = '" . $title . "',
    author = '" . $author . "',
    isbn = '" . $isbn . "',
    publisher = '" . $publisher . "'
  WHERE id = '" . $bid . "'
    ");

/*** bind the parameters ***/
$stmt->bindParam(':id', $bid, PDO::PARAM_STR);
$stmt->bindParam(':title', $title, PDO::PARAM_STR);
$stmt->bindParam(':author', $author, PDO::PARAM_STR);
$stmt->bindParam(':isbn', $isbn, PDO::PARAM_STR);
$stmt->bindParam(':publisher', $publisher, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

Database::disconnect();

/*** if all is done, say thanks ***/
$message = 'Επιτυχής αποθήκευση βιβλίου.';
set_error_message($message);
}
catch(Exception $e)   {
/*** if we are here, something has gone wrong with the database ***/
$message = 'Σφάλμα στη βάση δεδομένων.';
}
}
```

## 2.26 userlist.php – Λίστα χρηστών

Ο διαχειριστής έχει πρόσβαση σε μια συγκεντρωτική λίστα των εγγεγραμμένων χρηστών:

```
<?php
/*** begin the session ***/
session_start();
include_once 'helpers.php';
if (!is_user_admin()) {
set_error_message('Only administrators can access user list page.');
return;
}
?>
<!DOCTYPE html>
<html lang="el">
<head>
<title>Λίστα χρηστών</title>
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
```

```
</head>

<body>
<div class="container">

<div class="row">
<?php get_navbar(); ?>
</div>


<div class="row">
<?php
   include_once 'user_banner.php';
?>
</div>
<div class="row">
<?php
   include_once 'messages.php';
?>
</div>
<div class="row">
<h3>User list</h3>
</div>
<div class="row">

<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>mail</th>
<th>Ημ/νία εγγραφής</th>
<th>Απενεργοποιημένος?</th>
<th>Διαχειριστής?</th>
<th>Ενέργειες</th>
</tr>
</thead>
<tbody>
<?php
get_user_list();
?>
</tbody>
</table>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

## 2.27 admin_messages.php

Το αρχείο αυτό δεν εμφανίζεται αυτόνομα. Εμπεριέχει προσωρινά μηνύματα του συστήματος προς τον διαχειριστή και καλείται από τις διαχειριστικές σελίδες με τη μέθοδο “include”:

```php
<?php

if (!isset($_SESSION['message'])) {
return;
}

$message = $_SESSION['message'];

if (isset($_SESSION['message_type'])) {
$message_type = $_SESSION['message_type'];
print '<div class="alert alert-' . $message_type . ' alert-dismissible" role="alert">' .
$_SESSION['message'] . '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button></div>';
unset($_SESSION['message']);
unset($_SESSION['message_type']);
}
else {
print '<div class="alert alert-success alert-dismissible" role="alert">' . $_SESSION['message'] .
'<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button></div>';
unset($_SESSION['message']);
}
```

## 2.28 messages.php

Αντίστοιχα με τα μηνύματα του διαχειριστή, ισχύει και με τα μηνύματα του απλού χρήστη:

```php
<?php

if (!isset($_SESSION['message'])) {
return;
}

$message = $_SESSION['message'];

if (isset($_SESSION['message_type'])) {
$message_type = $_SESSION['message_type'];
print '<div class="alert alert-' . $message_type . ' alert-dismissible" role="alert">' .
$_SESSION['message'] . '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button></div>';
unset($_SESSION['message']);
unset($_SESSION['message_type']);
}
else {
print '<div class="alert alert-success alert-dismissible" role="alert">' . $_SESSION['message'] .
'<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button></div>';
unset($_SESSION['message']);
}
```

## 2.29 user_banner.php

Όπως με τα μηνύματα χρηστών και διαχειριστών, ένα banner αποτελεί σημείο αναφοράς για κάθε χρήστη κατά τη διάρκεια της περιήγησής του:

```php
<?php

include_once 'classes/Database.php';


if(!isset($_SESSION['user_id'])) {
$message = 'Καλωσόρισατε, ανώνυμε χρήστη!';
print '<h4 class="text-muted text-right"><em>' . $message . '</em></h4>';
return;
}
else {
$user_mail = get_user_name_from_id($_SESSION['user_id']);
$message = 'Καλωσορίσατε! Είστε ο χρήστης ' . $user_mail;
print '<h4 class="text-muted text-right"><em>' . $message . '</em></h4>';
}
```

## 2.30 helpers.php

Στο αρχείο helpers.php, το οποίο φορτώνεται στην αρχή κάθε επιμέρους σελίδας, υπάρχουν οι συναρτήσεις που εκτελούν όλες τις διεργασίες:

```php
<?php
include_once 'classes/Database.php';

function get_user_name_from_id($id) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("SELECT id, mail FROM users
                    WHERE id = :id");

/*** bind the parameters ***/
$stmt->bindParam(':id', $id, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();
$result1 = $result[0];
$user_mail = $result1['mail'];
return $user_mail;
}

function is_user_logged_in() {
if(isset($_SESSION['user_id'])) {
return TRUE;
}
```

```
else {
return FALSE;
}
}

function whoami() {
return $_SESSION['user_id'];
}

function is_user_admin() {
if(isset($_SESSION['user_isadmin'])) {
return TRUE;
}
else {
return FALSE;
}
}

function get_book_actions($id) {
echo '<a class="btn btn-info btn-block" href="view.php?id=' . $id . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a>';
if (is_user_logged_in()) {

if (!is_this_book_lent($id) && !is_book_deactivated($id)) {
echo '<a class="btn btn-success btn-block" href="askbook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-shopping-cart"></span> Δανεισμός</a>';
}

if (is_user_admin()) {
echo '<a class="btn btn-warning btn-block" href="updatebook.php?bid=' . $id . '"><span
class="glyphicon glyphicon-floppy-disk"></span> Διόρθωση</a>';

if (is_book_deactivated($id)) {
echo '<a class="btn btn-success btn-block" href="activate_book.php?bid=' . $id . '"><span
class="glyphicon glyphicon-ok"></span> Ενεργοποίηση</a>';
}
else {
echo '<a class="btn btn-danger btn-block" href="deactivate_book.php?bid=' . $id . '"><span
class="glyphicon glyphicon-remove"></span> Απενεργοποίηση</a>';
}

    }
  }
}

function get_book_inside_actions($id) {
if (is_user_logged_in()) {
if (!is_this_book_lent($id) && !is_book_deactivated($id)) {
echo '<a class="btn btn-success" href="askbook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-shopping-cart"></span> Δανεισμός</a>';
}
echo '<a class="btn btn-success" href="addtofavorites.php?bid=' . $id . '"><span class="glyphicon
glyphicon-star"></span> Αγαπημένο</a>';
if (is_user_admin()) {

echo '<a class="btn btn-warning" href="updatebook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-floppy-disk"></span> Διόρθωση</a>';


if (is_book_deactivated($id)) {
```

```
echo '<a class="btn btn-success" href="activate_book.php?bid=' . $id . '"><span class="glyphicon
glyphicon-ok"></span> Ενεργοποίηση</a>';
}
else {
echo '<a class="btn btn-danger" href="deactivate_book.php?bid=' . $id . '"><span class="glyphicon
glyphicon-remove"></span> Απενεργοποίηση</a>';
}
    }
  }
}

function get_admin_actions_asked($bid) {
echo '<a class="btn btn-success" href="approve_ask.php?lid=' . $bid . '">Αποδοχή κράτησης</a>';
echo '<a class="btn btn-success" href="deny_ask.php?lid=' . $bid . '">Απόρριψη κράτησης</a>';
}

function get_admin_actions_approved($bid) {
echo '<a class="btn btn-success" href="lendbook.php?lid=' . $bid . '">Δανεισμός</a>';
echo '<a class="btn btn-success" href="deny_ask.php?lid=' . $bid . '">Απόρριψη κράτησης</a>';
}

function get_admin_actions_lent($bid) {
echo '<a class="btn btn-success" href="returnbook.php?lid=' . $bid . '">Επιστροφή</a>';
}

function set_error_message($message, $location = NULL, $message_type = NULL) {
$_SESSION['message'] = $message;
if($message_type) {
$_SESSION['message_type'] = $message_type;
}

if ($location) {
$filename = explode("?", $location);
if (file_exists($filename[0])) {
     header("Location: " . $location);
return;
}
  }
  header("Location: index.php");
return;
}

function set_admin_message($message, $location = NULL, $message_type = NULL) {
$_SESSION['message'] = $message;
if($message_type) {
$_SESSION['message_type'] = $message_type;
}

if ($location) {
$filename = explode("?", $location);
if (file_exists($filename[0])) {
     header("Location: " . $location);
return;
}
  }

  header("Location: admin.php");
return;
}
```

```
function is_this_book_in_favorites($bid, $uid) {

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid
                    FROM favorites
                    WHERE bid = :bid
                    AND uid = :uid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}

function has_user_marked_this_book($bid, $uid) {

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid, rate
                    FROM rates
                    WHERE bid = :bid
                    AND uid = :uid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}
```

```php
function get_book_average_mark($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT AVG(rate) as avg
                    FROM rates
                    WHERE bid = :bid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
return $result[0]['avg'];
}

}

function is_this_book_lent($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid
                    FROM lend
                    WHERE bid = :bid
                    AND status IN (0, 1, 2)");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}

function get_rating_block($bid) {
echo '<form action="markbook.php" method="post">';
echo '        <fieldset>';
echo '          <p>';
echo '            <label for="mark">Βαθμός:</label>';
```

```
echo '            <select name="mark">';
echo '              <option value="1">1</option>';
echo '              <option value="2">2</option>';
echo '              <option value="3">3</option>';
echo '              <option value="4">4</option>';
echo '              <option value="5">5</option>';
echo '              <option value="6">6</option>';
echo '              <option value="7">7</option>';
echo '              <option value="8">8</option>';
echo '              <option value="9">9</option>';
echo '              <option value="10">10</option>';
echo '            </select>';
echo '            <input type="hidden" name="bid" value="' . $bid . '" />';
echo '            <input type="submit" value="Βαθμολόγηση" />';
echo '          </p>';
echo '        </fieldset>';
echo '      </form>';
}

function get_book_status_number($bid) {

if (is_book_deactivated($bid)) {
return '-2';
}

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid, status
                    FROM lend
                    WHERE bid = :bid
                    ORDER BY created DESC
                    LIMIT 1
                    ");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return '-1';
}
else {
$result1 = $result[0];
return $result1['status'];
}
}

function get_book_status_text($bid) {

$status = get_book_status_number($bid);
switch ($status) {
case '-2':
return 'Απωλεσθέν';
```

```
break;
case '-1':
return 'Διαθέσιμο';
break;
case '0':
return 'Κρατημένο';
break;
case '1':
return 'Κρατημένο';
break;
case '2':
return 'Δανεισμένο';
break;
default:
return 'Διαθέσιμο';
break;
}
}

function is_book_deactivated($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, deactivated
                    FROM books
                    WHERE id = :bid
                    ");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();


if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
if ($result1['deactivated']) {
return TRUE;
}
return FALSE;
}
}

function get_user_favorites($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  books.id,
 books.title,
 books.author,
 books.isbn
FROM books
```

```
JOIN favorites on favorites.bid = books.id
JOIN users on favorites.uid = users.id
WHERE favorites.uid = ' . $uid;
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td><a class="btn btn-success" href="addtofavorites.php?bid=' . $row['id'] . '"><span
class="glyphicon glyphicon-star"></span></a></td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_user_asked($uid) {
$pdo = Database::connect();
$sql = '
SELECT
 lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.created
FROM
 books
JOIN lend ON lend.bid = books.id
JOIN users ON lend.uid = users.id
WHERE
 lend.status IN (0, 1)
AND users.id =' . $uid;

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['created'] . '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_currently_lent($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.lent
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 2
AND users.id = ' . $uid . '
```

```
ORDER BY lend.lent ASC
';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['lent'] . '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_history($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.lent,
 lend.returned
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 3
AND users.id = ' . $uid . '
ORDER BY lend.lent ASC
';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['lent'] . '</td>';
echo '<td>'. $row['returned'] . '</td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_user_marks($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  books.id,
 books.title,
 books.author,
 books.isbn,
 rates.rate
FROM books
JOIN rates on rates.bid = books.id
JOIN users on rates.uid = users.id
WHERE rates.uid = ' . $uid;
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
```

```
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['rate'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['id'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_homepage_list() {
$pdo = Database::connect();
$sql = 'SELECT * FROM books ORDER BY id DESC';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';

echo '<td>';
if (isset($_SESSION['user_id'])) {
echo is_this_book_in_favorites($row['id'], $_SESSION['user_id']) ? '<span class="glyphicon
glyphicon-star"></span> ':'';
}
echo $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['publisher'] . '</td>';
echo ($row['image']!='-') ? '<td><img class="img-thumbnail" width="100" src="'. $row['image'] . '"
/></td>' : '<td>Εξώφυλλο μη διαθέσιμο</td>';
echo '<td>'. $row['created'] . '</td>';
echo '<td>'. get_book_status_text($row['id']) . '</td>';
echo '<td width=250>';
get_book_actions($row['id']);
echo '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_similar_books($bid) {
$pdo = Database::connect();
$sql = '
SELECT DISTINCT
 books.id,
 books.title,
 books.author,
 books.image,
 books.isbn
FROM
 lend
JOIN books ON books.id = lend.bid
WHERE
 lend.bid != ' . $bid . '
AND lend.uid IN (
 SELECT
  lend2.uid
 FROM
  lend AS lend2
 WHERE
```

```
  lend2.bid = ' . $bid . '
)
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['id'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_most_favored_books() {
$pdo = Database::connect();
$sql = '
SELECT
 count(bid) AS count,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 favorites
JOIN books ON books.id = bid
GROUP BY
 bid
ORDER BY
 count DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['count'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_most_rated_books() {
$pdo = Database::connect();
$sql = '
SELECT
 AVG(rate) AS average,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 rates
```

```
JOIN books ON books.id = bid
GROUP BY
 bid
ORDER BY
 average DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['average'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_most_lent_books() {
$pdo = Database::connect();
$sql = '
SELECT
 COUNT(lend.id) AS times,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 lend
JOIN books ON books.id = bid
GROUP BY
 bid
ORDER BY
 times DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['times'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_list() {
$pdo = Database::connect();
$sql = 'SELECT
 id,
 mail,
```

```
 created,
 deactivated,
 isadmin
FROM
 users';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';
echo '<td>'. $row['mail'] . '</td>';
echo '<td>'. $row['created'] . '</td>';

echo '<td>';
echo $row['deactivated'] ? 'Ναι' : 'Οχι';
echo '</td>';

echo '<td>';
echo $row['isadmin'] ? 'Ναι' : 'Οχι';
echo '</td>';

echo '<td width=250>';
get_user_actions($row['id'], $row['deactivated']);
echo '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_actions($uid, $isdeactivated) {
if ($isdeactivated) {
echo '<a class="btn btn-success" href="activate_user.php?uid=' . $uid . '">Ενεργοποίηση χρήστη</a>';
}
else {
echo '<a class="btn btn-danger" href="deactivate_user.php?uid=' . $uid . '">Απενεργοποίηση
χρήστη</a>';
}
}

function how_many_favored_this_book($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT
                      COUNT(uid) as count
                    FROM
                      favorites
                    WHERE bid = :bid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return '0';
```

```
}
else {
return $result[0]['count'];
}

}

function get_navbar() {
print <<<END
<nav class="navbar navbar-default">
<div class="container-fluid">
<!-- Brand and toggle get grouped for better mobile display -->
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-
example-navbar-collapse-1">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
<ul class="nav navbar-nav">
<li><a href="index.php"><button class="btn btn-default "><span class="glyphicon glyphicon-
home"></span> Αρχική</button></a></li>
<li><a href="statistics.php"><button class="btn btn-info "><span class="glyphicon glyphicon-
stats"></span> Στατιστικά</button></a></li>

END;
if (is_user_logged_in()) {
print '<li><a href="profile.php"><button class="btn btn-success "><span class="glyphicon glyphicon-
info-sign"></span> Το προφίλ μου</button></a></li>';
if (is_user_admin()) {
print <<<END4
<li>
<div class="btn-group" style="padding: 15px 15px;">
<button type="button" class="btn btn-warning dropdown-toggle" data-toggle="dropdown" aria-
expanded="false">
        Διαχείριση <span class="caret"></span>
</button>
<ul class="dropdown-menu" role="menu">
<li><a href="admin.php"><span class="glyphicon glyphicon-envelope"></span> Αιτήματα
δανεισμού</a></li>
<li class="divider"></li>
<li><a href="userlist.php"><span class="glyphicon glyphicon-th-list"></span> Λίστα
χρηστών</a></li>
<li><a href="adduser.php"><span class="glyphicon glyphicon-user"></span> Προσθήκη νέου
χρήστη</a></li>
<li><a href="addbook.php"><span class="glyphicon glyphicon-book"></span> Προσθήκη νέου
βιβλίου</a></li>
</ul>
</div>
</li>



END4;
```

```
}
  }
print <<<END3
</ul>

<ul class="nav navbar-nav navbar-right">
END3;
get_login_logout_register_button();
print <<<END2

</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
END2;
}

function get_login_logout_register_button() {
if (is_user_logged_in()) {
print '<li><a href="logout.php"><button class="btn btn-danger btn-lg"><span class="glyphicon
glyphicon-log-out"></span> Αποσύνδεση</button></a></li>';
}
else {
print '<li><a href="adduser.php"><button class="btn btn-success btn-lg"><span class="glyphicon
glyphicon-copy"></span> Εγγραφή</button></a></li>';
print '<li><a href="login.php"><button class="btn btn-success btn-lg"><span class="glyphicon
glyphicon-log-in"></span> Σύνδεση</button></a></li>';
}
}

function get_footer() {
print <<?php
include_once 'classes/Database.php';

function get_user_name_from_id($id) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("SELECT id, mail FROM users
                    WHERE id = :id");

/*** bind the parameters ***/
$stmt->bindParam(':id', $id, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();
$result1 = $result[0];
$user_mail = $result1['mail'];
return $user_mail;
}

function is_user_logged_in() {
if(isset($_SESSION['user_id'])) {
return TRUE;
}
else {
return FALSE;
```

```
}
}

function whoami() {
return $_SESSION['user_id'];
}

function is_user_admin() {
if(isset($_SESSION['user_isadmin'])) {
return TRUE;
}
else {
return FALSE;
}
}

function get_book_actions($id) {
echo '<a class="btn btn-info btn-block" href="view.php?id=' . $id . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a>';
if (is_user_logged_in()) {

if (!is_this_book_lent($id) && !is_book_deactivated($id)) {
echo '<a class="btn btn-success btn-block" href="askbook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-shopping-cart"></span> Δανεισμός</a>';
}
//echo '<a class="btn btn-success" href="addtofavorites.php?bid=' . $id . '"><span class="glyphicon
glyphicon-star"></span></a>';
if (is_user_admin()) {
echo '<a class="btn btn-warning btn-block" href="updatebook.php?bid=' . $id . '"><span
class="glyphicon glyphicon-floppy-disk"></span> Διόρθωση</a>';

if (is_book_deactivated($id)) {
echo '<a class="btn btn-success btn-block" href="activate_book.php?bid=' . $id . '"><span
class="glyphicon glyphicon-ok"></span> Ενεργοποίηση</a>';
}
else {
echo '<a class="btn btn-danger btn-block" href="deactivate_book.php?bid=' . $id . '"><span
class="glyphicon glyphicon-remove"></span> Απενεργοποίηση</a>';
}

    }
  }
}

function get_book_inside_actions($id) {
if (is_user_logged_in()) {
if (!is_this_book_lent($id) && !is_book_deactivated($id)) {
echo '<a class="btn btn-success" href="askbook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-shopping-cart"></span> Δανεισμός</a>';
}
echo '<a class="btn btn-success" href="addtofavorites.php?bid=' . $id . '"><span class="glyphicon
glyphicon-star"></span> Αγαπημένο</a>';
if (is_user_admin()) {

echo '<a class="btn btn-warning" href="updatebook.php?bid=' . $id . '"><span class="glyphicon
glyphicon-floppy-disk"></span> Διόρθωση</a>';


if (is_book_deactivated($id)) {
echo '<a class="btn btn-success" href="activate_book.php?bid=' . $id . '"><span class="glyphicon
```

```
glyphicon-ok"></span> Ενεργοποίηση</a>';
}
else {
echo '<a class="btn btn-danger" href="deactivate_book.php?bid=' . $id . '"><span class="glyphicon
glyphicon-remove"></span> Απενεργοποίηση</a>';
}
   }
  }
}

function get_admin_actions_asked($bid) {
echo '<a class="btn btn-success" href="approve_ask.php?lid=' . $bid . '">Αποδοχή κράτησης</a>';
echo '<a class="btn btn-success" href="deny_ask.php?lid=' . $bid . '">Απόρριψη κράτησης</a>';
}

function get_admin_actions_approved($bid) {
echo '<a class="btn btn-success" href="lendbook.php?lid=' . $bid . '">Δανεισμός</a>';
echo '<a class="btn btn-success" href="deny_ask.php?lid=' . $bid . '">Απόρριψη κράτησης</a>';
}

function get_admin_actions_lent($bid) {
echo '<a class="btn btn-success" href="returnbook.php?lid=' . $bid . '">Επιστροφή</a>';
}

function set_error_message($message, $location = NULL, $message_type= NULL) {
$_SESSION['message'] = $message;
if($message_type) {
$_SESSION['message_type'] = $message_type;
}

if ($location) {
$filename = explode("?", $location);
if (file_exists($filename[0])) {
     header("Location: " . $location);
return;
}
  }
  header("Location: booklist.php");
return;
}

function set_admin_message($message, $location = NULL, $message_type = NULL) {
$_SESSION['message'] = $message;
if($message_type) {
$_SESSION['message_type'] = $message_type;
}

if ($location) {
$filename = explode("?", $location);
if (file_exists($filename[0])) {
     header("Location: " . $location);
return;
}
  }

  header("Location: admin.php");
return;
}

function is_this_book_in_favorites($bid, $uid) {
```

```
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid
                    FROM favorites
                    WHERE bid = :bid
                    AND uid = :uid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}

function has_user_marked_this_book($bid, $uid) {

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid, rate
                    FROM rates
                    WHERE bid = :bid
                    AND uid = :uid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);
$stmt->bindParam(':uid', $uid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}

function get_book_average_mark($bid) {
```

```
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT AVG(rate) as avg
                    FROM rates
                    WHERE bid = :bid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
return $result[0]['avg'];
}

}

function is_this_book_lent($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid
                    FROM lend
                    WHERE bid = :bid
                    AND status IN (0, 1, 2)");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
return $result1['id'];
}
}

function get_rating_block($bid) {
echo '<form action="markbook.php" method="post">';
echo '        <fieldset>';
echo '          <p>';
echo '            <label for="mark">Βαθμός:</label>';
echo '            <select name="mark">';
```

```
echo '              <option value="1">1</option>';
echo '              <option value="2">2</option>';
echo '              <option value="3">3</option>';
echo '              <option value="4">4</option>';
echo '              <option value="5">5</option>';
echo '              <option value="6">6</option>';
echo '              <option value="7">7</option>';
echo '              <option value="8">8</option>';
echo '              <option value="9">9</option>';
echo '              <option value="10">10</option>';
echo '            </select>';
echo '            <input type="hidden" name="bid" value="' . $bid . '" />';
echo '            <input type="submit" value="Βαθμολόγηση" />';
echo '          </p>';
echo '        </fieldset>';
echo '      </form>';
}

function get_book_status_number($bid) {

if (is_book_deactivated($bid)) {
return '-2';
}

$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, bid, uid, status
                    FROM lend
                    WHERE bid = :bid
                    ORDER BY created DESC
                    LIMIT 1
                    ");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return '-1';
}
else {
$result1 = $result[0];
return $result1['status'];
}
}

function get_book_status_text($bid) {

$status = get_book_status_number($bid);
switch ($status) {
case '-2':
return 'Απωλεσθέν';
break;
```

```
case '-1':
return 'Διαθέσιμο';
break;
case '0':
return 'Κρατημένο';
break;
case '1':
return 'Κρατημένο';
break;
case '2':
return 'Δανεισμένο';
break;
default:
return 'Διαθέσιμο';
break;
}
}

function is_book_deactivated($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                    SELECT id, deactivated
                    FROM books
                    WHERE id = :bid
                    ");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();


if (empty($result)) {
return FALSE;
}
else {
$result1 = $result[0];
if ($result1['deactivated']) {
return TRUE;
}
return FALSE;
}
}

function get_user_favorites($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  books.id,
 books.title,
 books.author,
 books.isbn
FROM books
JOIN favorites on favorites.bid = books.id
```

```
JOIN users on favorites.uid = users.id
WHERE favorites.uid = ' . $uid;
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td><a class="btn btn-success" href="addtofavorites.php?bid=' . $row['id'] . '"><span
class="glyphicon glyphicon-star"></span></a></td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_user_asked($uid) {
$pdo = Database::connect();
$sql = '
SELECT
 lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.created
FROM
 books
JOIN lend ON lend.bid = books.id
JOIN users ON lend.uid = users.id
WHERE
 lend.status IN (0, 1)
AND users.id =' . $uid;

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['created'] . '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_currently_lent($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.lent
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 2
AND users.id = ' . $uid . '
ORDER BY lend.lent ASC
```

```
';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['lent'] . '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_history($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  lend.id,
 books.title,
 books.author,
 books.isbn,
 users.mail,
 lend.lent,
 lend.returned
FROM books
JOIN lend on lend.bid = books.id
JOIN users on lend.uid = users.id
WHERE lend.status = 3
AND users.id = ' . $uid . '
ORDER BY lend.lent ASC
';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['lent'] . '</td>';
echo '<td>'. $row['returned'] . '</td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_user_marks($uid) {
$pdo = Database::connect();
$sql = '
SELECT
  books.id,
 books.title,
 books.author,
 books.isbn,
 rates.rate
FROM books
JOIN rates on rates.bid = books.id
JOIN users on rates.uid = users.id
WHERE rates.uid = ' . $uid;
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
```

```
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['rate'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['id'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_homepage_list() {
$pdo = Database::connect();
$sql = 'SELECT * FROM books ORDER BY id DESC';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';

echo '<td>';
if (isset($_SESSION['user_id'])) {
echo is_this_book_in_favorites($row['id'], $_SESSION['user_id']) ? '<span class="glyphicon
glyphicon-star"></span> ':'';
}
echo $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['publisher'] . '</td>';
echo ($row['image']!='-') ? '<td><img class="img-thumbnail" width="100" src="'. $row['image'] . '"
/></td>' : '<td>Εξώφυλλο μη διαθέσιμο</td>';
echo '<td>'. $row['created'] . '</td>';
echo '<td>'. get_book_status_text($row['id']) . '</td>';
echo '<td width=250>';
get_book_actions($row['id']);
echo '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_similar_books($bid) {
$pdo = Database::connect();
$sql = '
SELECT DISTINCT
 books.id,
 books.title,
 books.author,
 books.image,
 books.isbn
FROM
 lend
JOIN books ON books.id = lend.bid
WHERE
 lend.bid != ' . $bid . '
AND lend.uid IN (
 SELECT
  lend2.uid
 FROM
  lend AS lend2
 WHERE
  lend2.bid = ' . $bid . '
```

```
)
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['id'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_most_favored_books() {
$pdo = Database::connect();
$sql = '
SELECT
count(bid) AS count,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 favorites
JOIN books ON books.id = bid
GROUP BY
 bid
ORDER BY
 count DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['count'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();
}

function get_most_rated_books() {
$pdo = Database::connect();
$sql = '
SELECT
AVG(rate) AS average,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 rates
JOIN books ON books.id = bid
```

```
GROUP BY
 bid
ORDER BY
 average DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['average'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_most_lent_books() {
$pdo = Database::connect();
$sql = '
SELECT
COUNT(lend.id) AS times,
 bid,
 books.title,
 books.author,
 books.isbn
FROM
 lend
JOIN books ON books.id = bid
GROUP BY
 bid
ORDER BY
 times DESC
LIMIT 5
';

foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['title'] . '</td>';
echo '<td>'. $row['author'] . '</td>';
echo '<td>'. $row['isbn'] . '</td>';
echo '<td>'. $row['times'] . '</td>';
echo '<td><a class="btn btn-info" href="view.php?id=' . $row['bid'] . '"><span class="glyphicon
glyphicon-eye-open"></span> Προβολή</a></td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_list() {
$pdo = Database::connect();
$sql = 'SELECT
 id,
 mail,
 created,
```

```
 deactivated,
 isadmin
FROM
 users';
foreach ($pdo->query($sql) as $row) {
echo '<tr>';
echo '<td>'. $row['id'] . '</td>';
echo '<td>'. $row['mail'] . '</td>';
echo '<td>'. $row['created'] . '</td>';

echo '<td>';
echo $row['deactivated'] ? 'Ναι' : 'Όχι';
echo '</td>';

echo '<td>';
echo $row['isadmin'] ? 'Ναι' : 'Όχι';
echo '</td>';

echo '<td width=250>';
get_user_actions($row['id'], $row['deactivated']);
echo '</td>';
echo '</tr>';
}
  Database::disconnect();

}

function get_user_actions($uid, $isdeactivated) {
if ($isdeactivated) {
echo '<a class="btn btn-success" href="activate_user.php?uid=' . $uid . '">Ενεργοποίηση χρήστη</a>';
}
else {
echo '<a class="btn btn-danger" href="deactivate_user.php?uid=' . $uid . '">Απενεργοποίηση
χρήστη</a>';
}
}

function how_many_favored_this_book($bid) {
$dbh = Database::connect();
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
/*** prepare the select statement ***/
$stmt = $dbh->prepare("
                SELECT
COUNT(uid) as count
                FROM
                  favorites
                WHERE bid = :bid");

/*** bind the parameters ***/
$stmt->bindParam(':bid', $bid, PDO::PARAM_STR);

/*** execute the prepared statement ***/
$stmt->execute();

/*** check for a result ***/
$result = $stmt->fetchAll();

if (empty($result)) {
return '0';
}
```

```
else {
return $result[0]['count'];
}

}

function get_navbar() {
print <<<END
<nav class="navbar navbar-default">
<div class="container-fluid">
<!-- Brand and toggle get grouped for better mobile display -->
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-
example-navbar-collapse-1">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
<ul class="nav navbar-nav">
<li><a href="index.php"><button class="btn btn-default "><span class="glyphicon glyphicon-
home"></span> Αρχική</button></a></li>
<li><a href="booklist.php"><button class="btn btn-info "><span class="glyphicon glyphicon-
book"></span> Βιβλία</button></a></li>
<li><a href="statistics.php"><button class="btn btn-info "><span class="glyphicon glyphicon-
stats"></span> Στατιστικά</button></a></li>

END;
if (is_user_logged_in()) {
print '<li><a href="profile.php"><button class="btn btn-success "><span class="glyphicon glyphicon-
info-sign"></span> Το προφίλ μου</button></a></li>';
if (is_user_admin()) {
print <<<END4
<li>
<div class="btn-group" style="padding: 15px 15px;">
<button type="button" class="btn btn-warning dropdown-toggle" data-toggle="dropdown" aria-
expanded="false">
            Διαχείριση <span class="caret"></span>
</button>
<ul class="dropdown-menu" role="menu">
<li><a href="admin.php"><span class="glyphicon glyphicon-envelope"></span> Αιτήματα
δανεισμού</a></li>
<li class="divider"></li>
<li><a href="userlist.php"><span class="glyphicon glyphicon-th-list"></span> Λίστα
χρηστών</a></li>
<li><a href="adduser.php"><span class="glyphicon glyphicon-user"></span> Προσθήκη νέου
χρήστη</a></li>
<li><a href="addbook.php"><span class="glyphicon glyphicon-book"></span> Προσθήκη νέου
βιβλίου</a></li>
</ul>
</div>
</li>



END4;
```

```
}
  }
print <<<END3
</ul>

<ul class="nav navbar-nav navbar-right">
END3;
get_login_logout_register_button();
print <<<END2

</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
END2;
}

function get_login_logout_register_button() {
if (is_user_logged_in()) {
print '<li><a href="logout.php"><button class="btn btn-danger btn-lg"><span class="glyphicon
glyphicon-log-out"></span> Αποσύνδεση</button></a></li>';
}
else {
print '<li><a href="adduser.php"><button class="btn btn-success btn-lg"><span class="glyphicon
glyphicon-copy"></span> Εγγραφή</button></a></li>';
print '<li><a href="login.php"><button class="btn btn-success btn-lg"><span class="glyphicon
glyphicon-log-in"></span> Σύνδεση</button></a></li>';
}
}

function get_footer() {
print <<<ENDfooter
<div class="row" style="margin-top: 20px;"><footer class="footer bg-info" style="padding: 20px;">
<p class="text-center">
<em>Πειραματικό σύστημα διαχείρισης βιβλιοθήκης<br />
  TEI Ναυπάκτου<br /><br />

  Copyright <span class="glyphicon glyphicon-copyright-mark"></span> 2015 - Δώρα
Λαμπρακοπούλου<br />
  Το σύστημα αναπτύχθηκε στα πλαίσια πτυχιακής εργασίας.</em>
</p>
</footer></div>
ENDfooter;
}
```

## 2.31 booklist.php

```
<?php
/*** begin the session ***/
session_start();
include_once 'helpers.php';

?>
<!DOCTYPE html>
<html lang="el">
<head>
<title>Βιβλιοθήκη TEI Ναυπάκτου</title>
```

```
<meta charset="utf-8">
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>

<body>
<div class="container">
<div class="row">
<?php include_once 'messages.php'; ?>
</div>

<div class="row">
<?php get_navbar(); ?>
</div>

<div class="row">

</div>

<div class="row">
<div class="jumbotron">
<h1><span class="glyphicon glyphicon-book"></span>Βιβλιοθήκη TEI Ναυπάκτου</h1>
</div>
</div>

<div class="row">
<h3>Διαθέσιμα βιβλία</h3>
</div>

<div class="row">

<table class="table table-striped table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Τίτλος</th>
<th>Συγγραφέας</th>
<th>ISBN</th>
<th>Εκδόσεις</th>
<th>Εξώφυλλο</th>
<th>Ημ/νία Καταχώρισης</th>
<th>Κατάσταση</th>
<th>Επιλογές</th>
</tr>
</thead>
<tbody>
<?php
get_homepage_list();
?>
</tbody>
</table>
</div>
<?php get_footer(); ?>
</div><!-- /container -->
</body>
</html>
```

### 2.32 classes/Database.php

Στο συγκεκριμένο αρχείο υπάρχουν οι πληροφορίες για τη βάση δεδομένων, το όνομα χρήστη και τον κωδικό.

Η υλοποίηση έγινε με βάση το Singleton design pattern του αντικειμενοστρεφή προγραμματισμού: Η κλάση Database ελέγχει εάν υπάρχει αρχικοποιημένο αντικείμενο. Εάν ναι, επιστρέφει το ίδιο. Εάν όχι, αρχικοποιεί μόνη της ένα. Με τον τρόπο αυτό, μπορούμε να εξοικονομήσουμε πόρους και να αποφύγουμε υπερβολική υπερφόρτωση του web server:

```
<?php
class Database {
private static $dbName = library;
private static $dbHost = 'localhost' ;
private static $dbUsername = 'root';
private static $dbUserPassword = 'root';

private static $cont  = null;

public function __construct() {
die('Init function is not allowed');
}

public static function connect() {
// One connection through whole application
if ( null == self::$cont ) {
try {
self::$cont =  new PDO( "mysql:host=".self::$dbHost.";"."dbname=".self::$dbName,
self::$dbUsername, self::$dbUserPassword);
}
catch(PDOException $e) {
die($e->getMessage());
}
    }
return self::$cont;
}

public static function disconnect() {
self::$cont = null;
}
}
```

# 3.Web Server

Η εγκατάσταση και ο έλεγχος της εφαρμογής έγινε με τη χρήση του λογισμικού ανοικτού κώδικα WAMP (Windows – MySQL – PHP – Apache). Το WAMP είναι ένα πακέτο πολλών εργαλείων και το όνομά του είναι ένα αρκτικόλεξο που σχηματίζεται από τα αρχικά του λειτουργικού συστήματος των Microsoft Windows και τις κύριες συνιστώσες του πακέτου: Apache , MySQL και PHP. Ο Apache είναι ένας πολύ ισχυρός και από τους πιο διαδεδομένους web server .Η MySQL είναι όπως είδαμε ένα λογισμικό διαχείρισης βάσεων δεδομένων η οποία σε συνδυασμό με τη γλώσσα PHP δίνει τεράστιες δυνατότητες ανάπτυξης δυναμικών ιστοσελίδων. Άλλα προγράμματα που επίσης περιλαμβάνονται είναι το phpMyAdmin το οποίο παρέχει μια γραφική διεπαφή στο διαχειριστή της βάσης δεδομένων MySQL, ή τις εναλλακτικές γλώσσες προγραμματισμού Python και Perl.

Ο Apache HTTP Server, που συνήθως αναφέρεται ως Apache είναι ένας web server. Πρόκειται για λογισμικό που έχει βασικό ρόλο στην παροχή του περιεχομένου των ιστοσελίδων στις client εφαρμογές που είναι οι web browsers. Η πλειοψηφία των διακομιστών web που χρησιμοποιούν Apache λειτουργούν κάτω από ένα λειτουργικό σύστημα τύπου UNIX. Όμως την προϋπόθεση αυτή ήρθε να αλλάξει το WAMP το οποίο χρησιμοποιεί μια έκδοση του Apache για Windows.

## 3.1 Δημιουργία Βάσης Δεδομένων με PHPMyAdmin

Όπως ήδη αναφέραμε στο πακέτο WAMP συμπεριλαμβάνεται και ένα webbased εργαλείο διαχείρισης βάσεων δεδομένων της MySQL. Το εργαλείο αυτό είναι το PHPMyAdmin.Πρόκειται όπως είπαμε για ένα userinterface που επιτρέπει την κατασκευή αλλά και τροποποίηση μιας βάσης δεδομένων σε MySQL.Στην προκειμένη περίπτωση η βάση δεδομένων ονομάζεται library.

## 4.2 library.sql

Παρατίθεται αρχείο sql που περιέχει όλη τη δομή της βάσης δεδομένων, καθώς και μερικές πειραματικές καταχωρίσεις:

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

-- ----------------------------
--  Table structure for `books`
-- ----------------------------
DROP TABLE IF EXISTS `books`;
CREATE TABLE `books` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
`title` varchar(255) NOT NULL,
`author` varchar(255) DEFAULT NULL,
`isbn` varchar(13) DEFAULT NULL,
`publisher` varchar(255) DEFAULT NULL,
`image` varchar(511) DEFAULT NULL,
`created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`deactivated` int(11) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;

-- ----------------------------
--  Records of `books`
-- ----------------------------
BEGIN;
INSERT INTO `books` VALUES ('1', 'Learn PHP', 'George Smith', '123-456-789', 'Ekdoseis
EPE', '-', '2015-05-16 13:58:46', '0'), ('2', 'Object Oriented Programming', 'Kim Jung Chai',
'123456782254', 'Ekdoseis EPE', '-', '2015-05-23 18:26:57', null), ('8', 'Java for dummies',
'Nickolas Tried', '544582254', 'Ekdoseis EPE', '-', '2015-05-23 18:49:03', null), ('9', 'Software
Engineering', 'George Smith', '9579405738', 'Ekdoseis EPE', '-', '2015-05-23 18:49:22', null),
('10', 'Advanced Mathematics', 'John Steinbeck', '983445782254', 'OReily', '-', '2015-05-26
20:12:09', '0'), ('11', 'Modern PHP: New Features and Good Practices', 'Josh Lockhart', '978-
149190501', 'O&#39;Reilly Media', 'covers/book101.jpg', '2015-05-27 21:50:26', null), ('12', 'PHP
and MySQL Web Development', 'Luke Welling', '978-067232916', 'Addison-Wesley
Professional', 'covers/book102.jpg', '2015-05-27 21:51:17', null), ('13', 'Creating Mobile Apps
with jQuery Mobile ', 'Andy Matthews', '978-178355511', 'Packt Publishing',
'covers/book103.jpg', '2015-05-27 21:51:59', '1'), ('14', 'jQuery, jQuery UI, and jQuery Mobile:
Recipes and Examples', 'Adriaan de Jonge', '978-032182208', 'Addison-Wesley Professional',
'covers/book104.jpg', '2015-05-27 21:52:46', '0'), ('15', 'MySQL Cookbook: Solutions for
Database Developers and Administrators', 'Paul DuBois Junior', '978-144937402', 'O&#39;Reilly
Media', 'covers/book105.jpg', '2015-05-27 21:53:28', '0');
COMMIT;

-- ----------------------------
--  Table structure for `favorites`
-- ----------------------------
DROP TABLE IF EXISTS `favorites`;
CREATE TABLE `favorites` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
`uid` int(11) NOT NULL,
`bid` int(11) NOT NULL,
`created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id`),
KEY `uid@favorites` (`uid`) USING BTREE,
KEY `bid@favorites` (`bid`) USING BTREE,
CONSTRAINT `bid@favorites` FOREIGN KEY (`bid`) REFERENCES `books` (`id`),
CONSTRAINT `uid@favorites` FOREIGN KEY (`uid`) REFERENCES `users` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=45 DEFAULT CHARSET=utf8;
```

```
-- ----------------------------
--  Records of `favorites`
-- ----------------------------
BEGIN;
INSERT INTO `favorites` VALUES ('11', '2', '8', '2015-05-23 22:25:50'), ('12', '2', '2', '2015-05-23 22:25:52'), ('13', '4', '9', '2015-05-23 22:26:10'), ('14', '4', '1', '2015-05-23 22:26:12'), ('16', '7', '1', '2015-05-23 22:26:50'), ('17', '7', '2', '2015-05-23 22:26:52'), ('18', '8', '8', '2015-05-23 22:27:04'), ('21', '3', '2', '2015-05-24 15:39:32'), ('22', '10', '9', '2015-05-27 21:47:06'), ('23', '10', '10', '2015-05-27 21:47:06'), ('24', '10', '1', '2015-05-27 21:47:08'), ('25', '11', '10', '2015-05-27 21:47:40'), ('26', '11', '8', '2015-05-27 21:47:41'), ('27', '11', '1', '2015-05-27 21:47:42'), ('28', '11', '14', '2015-05-27 21:53:39'), ('29', '11', '15', '2015-05-27 21:53:40'), ('32', '12', '8', '2015-05-27 21:53:53'), ('33', '12', '12', '2015-05-27 21:53:55'), ('34', '13', '9', '2015-05-27 21:54:09'), ('35', '13', '14', '2015-05-27 21:54:10'), ('37', '13', '12', '2015-05-27 21:54:14'), ('38', '13', '11', '2015-05-27 21:54:14'), ('41', '12', '14', '2015-05-27 22:45:30'), ('43', '1', '15', '2015-05-29 21:00:25');
COMMIT;

-- ----------------------------
--  Table structure for `lend`
-- ----------------------------
DROP TABLE IF EXISTS `lend`;
CREATE TABLE `lend` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
`uid` int(11) NOT NULL,
`bid` int(11) NOT NULL,
`status` int(11) NOT NULL,
`created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`approved` timestamp NULL DEFAULT NULL,
`lent` timestamp NULL DEFAULT NULL,
`expire` timestamp NULL DEFAULT NULL,
`returned` timestamp NULL DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `uid@lend` (`uid`) USING BTREE,
KEY `bid@lend` (`bid`) USING BTREE,
CONSTRAINT `bid@lend` FOREIGN KEY (`bid`) REFERENCES `books` (`id`),
CONSTRAINT `uid@lend` FOREIGN KEY (`uid`) REFERENCES `users` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=31 DEFAULT CHARSET=utf8;

-- ----------------------------
--  Records of `lend`
-- ----------------------------
BEGIN;
INSERT INTO `lend` VALUES ('8', '3', '8', '3', '2015-05-26 23:47:05', '2015-05-27 00:25:07', '2015-05-27 00:25:19', null, '2015-05-27 00:25:44'), ('9', '2', '1', '3', '2015-05-27 00:34:26', '2015-05-27 00:34:32', '2015-05-27 00:34:33', null, '2015-05-27 00:34:35'), ('10', '2', '8', '3', '2015-05-27 00:35:46', '2015-05-27 00:35:50', '2015-05-27 00:35:51', null, '2015-05-27 00:35:53'), ('11', '1', '9', '3', '2015-05-27 00:36:27', '2015-05-27 00:36:30', '2015-05-27 00:36:32', null, '2015-05-27 00:36:33'), ('12', '3', '2', '3', '2015-05-27 00:44:33', '2015-05-27 01:03:49', '2015-05-27 01:03:54', null, '2015-05-27 01:03:57'), ('13', '1', '8', '3', '2015-05-27 00:45:42', '2015-05-27 01:03:50', '2015-05-27 01:03:55', null, '2015-05-27 01:03:58'), ('14', '1', '1', '3', '2015-05-27 00:45:58', '2015-05-27 01:03:51', '2015-05-27 01:03:56', null, '2015-05-27 01:03:58'), ('15', '10', '9', '3', '2015-05-27 21:47:04', '2015-05-27 21:47:24', '2015-05-27 21:47:26', null, '2015-05-27 21:47:30'), ('16', '10', '2', '3', '2015-05-27 21:47:05', '2015-05-27 21:47:27', '2015-05-27 21:47:28', null, '2015-05-27 21:47:31'), ('17', '12', '13', '3', '2015-05-27 21:53:57', '2015-05-27 21:54:27', '2015-05-27 21:54:34', null, '2015-05-27 21:54:51'), ('18', '12', '9', '3', '2015-05-27 21:53:58', '2015-05-27 21:54:28', '2015-05-27 21:54:35', null, '2015-05-27 21:54:53'), ('19', '13', '14', '3', '2015-05-27 21:54:17', '2015-05-27 21:54:29', '2015-05-27 21:54:33', null, '2015-05-27 21:54:51'), ('20', '13', '2', '3', '2015-05-27 21:54:20', '2015-05-27 21:54:30', '2015-05-27 21:54:35', null, '2015-05-27 21:54:52'), ('21', '13', '8', '3', '2015-05-27 21:54:22', '2015-05-27 21:54:31', '2015-05-27 21:54:36', null, '2015-05-27 21:54:54'), ('22', '12', '14', '3', '2015-05-27 22:45:41', '2015-05-27 22:46:24', '2015-05-27 22:46:26', null, '2015-05-27 22:46:28'), ('23', '1', '15', '3', '2015-05-27 23:17:15', '2015-05-27
```

```
23:17:19', '2015-05-27 23:17:20', null, '2015-05-27 23:17:21');
COMMIT;

-- ----------------------------
--  Table structure for `rates`
-- ----------------------------
DROP TABLE IF EXISTS `rates`;
CREATE TABLE `rates` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
`uid` int(11) NOT NULL,
`bid` int(11) NOT NULL,
`created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`rate` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `uid@rates` (`uid`),
KEY `bid@rates` (`bid`),
CONSTRAINT `bid@rates` FOREIGN KEY (`bid`) REFERENCES `books` (`id`),
CONSTRAINT `uid@rates` FOREIGN KEY (`uid`) REFERENCES `users` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=50 DEFAULT CHARSET=utf8;

-- ----------------------------
--  Records of `rates`
-- ----------------------------
BEGIN;
INSERT INTO `rates` VALUES ('1', '3', '9', '2015-05-23 20:03:37', '4'), ('2', '3', '8', '2015-05-23
20:08:58', '7'), ('3', '2', '9', '2015-05-23 22:29:50', '10'), ('4', '2', '8', '2015-05-23 22:29:54', '8'),
('5', '2', '2', '2015-05-23 22:29:58', '4'), ('6', '2', '1', '2015-05-23 22:30:04', '7'), ('7', '3', '2', '2015-
05-23 22:30:36', '4'), ('8', '3', '1', '2015-05-23 22:30:40', '10'), ('9', '4', '9', '2015-05-23 22:31:03',
'6'), ('10', '4', '8', '2015-05-23 22:31:06', '4'), ('11', '4', '2', '2015-05-23 22:31:09', '10'), ('12', '4',
'1', '2015-05-23 22:31:12', '9'), ('13', '7', '9', '2015-05-23 22:32:36', '1'), ('14', '7', '8', '2015-05-23
22:32:38', '9'), ('15', '7', '2', '2015-05-23 22:32:41', '6'), ('16', '7', '1', '2015-05-23 22:32:44', '1'),
('17', '10', '15', '2015-05-27 22:29:12', '9'), ('18', '10', '14', '2015-05-27 22:29:15', '6'), ('19', '10',
'13', '2015-05-27 22:29:17', '8'), ('20', '10', '12', '2015-05-27 22:29:19', '1'), ('21', '10', '11', '2015-
05-27 22:29:22', '2'), ('22', '10', '10', '2015-05-27 22:29:24', '9'), ('23', '10', '9', '2015-05-27
22:29:26', '5'), ('24', '10', '8', '2015-05-27 22:29:29', '1'), ('25', '10', '2', '2015-05-27 22:29:32',
'10'), ('26', '11', '1', '2015-05-27 22:29:54', '10'), ('27', '11', '2', '2015-05-27 22:29:56', '7'), ('28',
'11', '8', '2015-05-27 22:29:59', '9'), ('29', '11', '9', '2015-05-27 22:30:01', '5'), ('30', '11', '10',
'2015-05-27 22:30:04', '8'), ('31', '11', '11', '2015-05-27 22:30:06', '7'), ('32', '11', '12', '2015-05-27
22:30:09', '8'), ('33', '11', '13', '2015-05-27 22:30:12', '7'), ('34', '11', '14', '2015-05-27 22:30:14',
'2'), ('35', '11', '15', '2015-05-27 22:30:16', '7'), ('36', '12', '15', '2015-05-27 22:30:37', '8'), ('37',
'12', '14', '2015-05-27 22:30:39', '4'), ('38', '12', '13', '2015-05-27 22:30:41', '5'), ('39', '12', '12',
'2015-05-27 22:30:43', '2'), ('40', '12', '11', '2015-05-27 22:30:45', '3'), ('41', '12', '10', '2015-05-27
22:30:47', '2'), ('42', '12', '9', '2015-05-27 22:30:49', '4'), ('43', '12', '8', '2015-05-27 22:30:52', '7'),
('44', '12', '2', '2015-05-27 22:30:54', '2'), ('45', '12', '1', '2015-05-27 22:30:57', '3'), ('48', '1', '15',
'2015-06-01 17:35:49', '9'), ('49', '7', '15', '2015-06-01 18:11:29', '3');
COMMIT;

-- ----------------------------
--  Table structure for `users`
-- ----------------------------
DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
`mail` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`deactivated` int(11) DEFAULT NULL,
`isadmin` int(11) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8;
```

```
-- ----------------------------
--  Records of `users`
-- ----------------------------
BEGIN;
INSERT INTO `users` VALUES ('1', 'admin@test.com', 'adminpassword', '2015-05-16 11:44:02',
null, '1'), ('2', 'user1@test.com', 'user', '2015-05-16 11:44:25', '0', null), ('3', 'user2@test.com',
'user', '2015-05-16 11:44:45', '1', null), ('4', 'user3@test.com', 'user', '2015-05-16 11:45:01', '0',
null), ('7', 'user5@test.com', 'user', '2015-05-16 12:26:14', '0', null), ('8', 'user6@test.com', 'user',
'2015-05-17 20:48:13', '0', null), ('9', 'tralala@test.com', 'tralala', '2015-05-24 15:45:45', '0', null),
('10', 'user10', 'user10', '2015-05-27 21:46:31', null, null), ('11', 'user11', 'user11', '2015-05-27
21:46:37', null, null), ('12', 'user12', 'user12', '2015-05-27 21:46:43', null, null), ('13', 'user13',
'user13', '2015-05-27 21:46:48', null, null);
COMMIT;

SET FOREIGN_KEY_CHECKS = 1;
```

## Βιβλιογραφία


(php)

http://php.net

http://www.techteam.gr/wiki/PHP

http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html

http://web.archive.org/web/20130728125152/http://itc.conversationsnetwork.org/shows/detail58.html

http://www.w3schools.com/php/default.asp

(apache)

http://www.apache.org
http://www.webdevelopersnotes.com/basics/what_is_web_server.php

http://tools.ietf.org/html/rfc2616
http://httpd.apache.org/ABOUT_APACHE.html
http://www.ohloh.net/p/apache
http://en.wikipedia.org/wiki/Apache_HTTP_Server
http://www.webopedia.com/TERM/A/Apache_Web_server.html

http://www.modulehosting.com/apache.html
http://www.ntchosting.com/apache-web-server.html
http://www.apache.com/

(javascript)

http://www.freestuff.gr/forums/viewtopic.php?t=21175
https://developer.mozilla.org/en-US/docs/Web/JavaScript
http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf
http://www.ecmascript.org/es4/spec/overview.pdf
http://read.uberflip.com/i/113144/44

(sql)

http://en.wikipedia.org/wiki/MySQL
http://www.mysql.com/
http://www.sqlcourse.com/index.html
http://coursesweb.net/php-mysql/download_l2

(css)
http://el.wikipedia.org/wiki/CSS
http://www.wlearn.gr/index.php/home-css-83
http://en.wikipedia.org/wiki/Cascading_Style_Sheets
http://www.w3.org/Style/Examples/011/firstcss