



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Μάθημα: Δικτυοκεντρικά Πληροφοριακά Συστήματα

Επιβλέπων καθηγητής: Κυρία Κούφη Βασιλική

Θέμα Εργασίας:

Ανάπτυξη εφαρμογής στο Android Studio

Φοιτητής:

Παπακωνσταντίνου Αντώνης

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή.....	3
Κεφάλαιο 2: Περιγραφή συστήματος.....	4
Κεφάλαιο 3: Υλοποίηση Συστήματος.....	8
Κεφάλαιο 4: Εγχειρίδιο Διαχειριστή.....	31
Κεφάλαιο 5: Εγχειρίδιο Χρήστη	34

Κεφάλαιο 1: Εισαγωγή

Η παρούσα εργασία αφορά την ανάπτυξη ενός δικτυοκεντρικού πληροφοριακού συστήματος για τη διαχείριση χρηστών και κρατήσεων ενός γυμναστηρίου, χρησιμοποιώντας τη γλώσσα προγραμματισμού Python και το πλαίσιο Flask για τη δημιουργία του web application. Το σύστημα αποτελείται από δύο βασικά υποσυστήματα:

- 1) Το διαχειριστικό.
- 2) Το σύστημα χρηστών.

Στο διαχειριστικό σύστημα, οι διαχειριστές έχουν τη δυνατότητα να διαχειρίζονται αιτήματα εγγραφής χρηστών, να επεξεργάζονται τα στοιχεία τους, και να αναθέτουν ρόλους. Οι ρόλοι είναι 2. Χρήστης ή Διαχειριστής. Επιπλέον, μπορούν να διαχειρίζονται τα δομικά στοιχεία του γυμναστηρίου, όπως τους γυμναστές και τα προγράμματα, καθώς και να αναρτούν ανακοινώσεις και προσφορές.

Από την πλευρά του χρήστη, το σύστημα επιτρέπει την περιήγηση στις υπηρεσίες του γυμναστηρίου, την πραγματοποίηση κρατήσεων για προγράμματα και υπηρεσίες, την προβολή ιστορικού κρατήσεων, καθώς και την ενημέρωση για νέα και ανακοινώσεις. Η εγγραφή στο σύστημα απαιτεί την υποβολή σχετικής αίτησης, η οποία εγκρίνεται από τον διαχειριστή, και στη συνέχεια ο χρήστης αποκτά πρόσβαση στις αντίστοιχες λειτουργίες του συστήματος.

Το UI του συστήματος σχεδιάστηκε χρησιμοποιώντας HTML, με το styling να επιτυγχάνεται μέσω CSS και Bootstrap, προσφέροντας έτσι ένα όμορφο και λειτουργικό περιβάλλον χρήσης. Το σύστημα έχει αναπτυχθεί με σκοπό να διευκολύνει τόσο τη διαχείριση του γυμναστηρίου όσο και την εμπειρία των χρηστών, προσφέροντας ένα οργανωμένο, ευέλικτο και εύχρηστο περιβάλλον.

Κεφάλαιο 2: Περιγραφή συστήματος

Η εργασία αυτή αφορά την ανάπτυξη ενός δικτυοκεντρικού πληροφοριακού συστήματος, σχεδιασμένου για τη διαχείριση χρηστών και κρατήσεων ενός γυμναστηρίου. Το σύστημα στοχεύει στην αποτελεσματική διαχείριση των πελατών και των υπηρεσιών που προσφέρονται, διευκολύνοντας τη λειτουργία του γυμναστηρίου και βελτιώνοντας την εμπειρία των χρηστών.

Για την υλοποίηση του συστήματος χρησιμοποιήσα τις τεχνολογίες Python και Flask, ενώ η βάση δεδομένων υποστηρίζεται από το MongoDB. Το περιβάλλον χρήστη (UI) σχεδιάστηκε με HTML και διαμορφώθηκε με CSS αλλά και Bootstrap.

Υποσυστήματα και Λειτουργίες

Το σύστημα χωρίζεται σε δύο κύρια υποσυστήματα:

1. Το **διαχειριστικό σύστημα**.
2. Το **σύστημα χρηστών**.

1. Διαχειριστικό Σύστημα

1. Διαχείριση Αιτημάτων Εγγραφής:

- Ο διαχειριστής του συστήματος έχει την δυνατότητα να εγκρίνει ή να απορρίπτει αιτήματα εγγραφής νέων χρηστών στο σύστημα. Κάθε χρήστης που επιθυμεί να χρησιμοποιήσει τις υπηρεσίες του γυμναστηρίου πρέπει πρώτα να εγγραφεί, παρέχοντας προσωπικά στοιχεία όπως όνομα, επώνυμο, διεύθυνση, email κ.λπ. Ύστερα, πρέπει να περιμένει έγκριση από τον διαχειριστή.

2. Διαχείριση Χρηστών Συστήματος:

- Ο διαχειριστής μπορεί να επεξεργάζεται τα στοιχεία των χρηστών (Create, Read, Update, Delete - CRUD operations) και να τους αναθέτει ρόλους, όπως απλός χρήστης ή διαχειριστής. Αυτή η λειτουργία επιτρέπει την οργάνωση των χρηστών και τη διασφάλιση ότι ο κάθε χρήστης έχει πρόσβαση στις κατάλληλες λειτουργίες του συστήματος.

3. Διαχείριση Δομικών Στοιχείων του Συστήματος:

- Ο διαχειριστής έχει πρόσβαση και ευθύνη για τη διαχείριση όλων των υπηρεσιών που προσφέρει το γυμναστήριο. Αυτό περιλαμβάνει την επεξεργασία των δεδομένων που αφορούν τους γυμναστές, τα προσφερόμενα προγράμματα (π.χ. pilates, ενδυνάμωση, πισίνα), και τον προγραμματισμό των ομαδικών προγραμμάτων. Για κάθε πρόγραμμα μπορεί να καθορίζεται η μέγιστη χωρητικότητα ατόμων.

4. Διαχείριση Ανακοινώσεων και Προσφορών:

- Ο διαχειριστής μπορεί να αναρτά, να τροποποιεί και να διαγράφει ανακοινώσεις και προσφορές, οι οποίες είναι ορατές μόνο στους εγγεγραμμένους χρήστες του συστήματος.

2. Σύστημα Χρηστών

1. Περιήγηση στις Υπηρεσίες του Γυμναστηρίου:

- Οι χρήστες μπορούν να περιηγηθούν στις διαθέσιμες υπηρεσίες του γυμναστηρίου χωρίς να είναι εγγεγραμμένοι ή να έχουν εισέλθει στο σύστημα. Αυτό παρέχει στους χρήστες μια γενική εικόνα των υπηρεσιών που προσφέρονται.

2. Κράτηση Προϊόντος ή Υπηρεσίας:

- Οι χρήστες που είναι εγγεγραμμένοι και έχουν εισέλθει στο σύστημα μπορούν να κάνουν κρατήσεις για ατομικά ή ομαδικά προγράμματα. Η διαδικασία περιλαμβάνει την αναζήτηση διαθεσιμότητας για το επιθυμητό πρόγραμμα και την ημερομηνία. Εάν υπάρχει διαθέσιμη θέση, ο χρήστης μπορεί να κλείσει ραντεβού. Οι χρήστες μπορούν να ακυρώσουν το ραντεβού τους έως και 2 ώρες πριν την έναρξη, με περιορισμούς στην περίπτωση επαναλαμβανόμενων ακυρώσεων.

3. Ιστορικό Κρατήσεων:

- ο Οι χρήστες έχουν τη δυνατότητα να δουν το ιστορικό των κρατήσεών τους, με αναλυτική λίστα όλων των κρατήσεων που έχουν πραγματοποιήσει.

4. Νέα και Ανακοινώσεις:

- ο Οι χρήστες μπορούν να ενημερώνονται για νέα και ανακοινώσεις που έχουν αναρτηθεί από το γυμναστήριο.

Backend: Το σύστημα αναπτύχθηκε με χρήση της Python και του Flask, που προσφέρει ένα ευέλικτο και ελαφρύ framework για την ανάπτυξη web εφαρμογών. Το Flask επιλέχθηκε λόγω της απλότητας και της δυνατότητας για γρήγορη ανάπτυξη εφαρμογών που απαιτούν βασικές web λειτουργίες.

Database: Η βάση δεδομένων είναι MongoDB, η οποία επιτρέπει την αποθήκευση δεδομένων με ευέλικτο και αποδοτικό τρόπο. Το MongoDB είναι κατάλληλο για εφαρμογές όπου οι σχέσεις μεταξύ δεδομένων δεν είναι αυστηρά δομημένες, παρέχοντας δυνατότητα για αποθήκευση και ανάκτηση δεδομένων με γρήγορη ταχύτητα.

Frontend: Το περιβάλλον χρήστη (UI) δημιουργήθηκε χρησιμοποιώντας HTML για τη δομή, CSS για τη μορφοποίηση, και το framework Bootstrap. Το Bootstrap προσφέρει έτοιμα stylings που επιταχύνουν την ανάπτυξη και εξασφαλίζουν ευχάριστο οπτικό αποτέλεσμα.

Το σύστημα περιλαμβάνει μηχανισμούς ασφαλείας που διασφαλίζουν την προστασία των προσωπικών δεδομένων των χρηστών και τη σωστή διαχείριση των κρατήσεων. Οι κωδικοί πρόσβασης αποθηκεύονται κρυπτογραφημένοι, και η επικοινωνία με το σύστημα γίνεται μέσω ασφαλών πρωτοκόλλων (όπως SSL). Οι διαφορετικοί ρόλοι των χρηστών καθορίζουν τα επίπεδα πρόσβασης στις διάφορες λειτουργίες του συστήματος.

Τέλος, το σύστημα έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι εύκολα επεκτάσιμο και προσαρμόσιμο σε μελλοντικές ανάγκες. Με τη χρήση του MongoDB, μπορεί να διαχειριστεί μεγάλο όγκο δεδομένων και χρηστών, ενώ η αρχιτεκτονική του βασίζεται σε καλά καθορισμένα API που επιτρέπουν την εύκολη ενσωμάτωση νέων λειτουργιών ή την επέκταση των υπαρχόντων.

Συνολικά, το σύστημα προσφέρει ένα ολοκληρωμένο και λειτουργικό περιβάλλον για τη διαχείριση των χρηστών και των υπηρεσιών ενός γυμναστηρίου, με έμφαση στην ευκολία χρήσης, την ασφάλεια και την επεκτασιμότητα.

Κεφάλαιο 3: Υλοποίηση Συστήματος

Η εφαρμογή υλοποιήθηκε χρησιμοποιώντας HTML, CSS, Bootstrap για το frontend, και Python με το Flask για τη διαχείριση των APIs και την αλληλεπίδραση με τη βάση δεδομένων MongoDB. Το backend επεξεργάζεται αιτήσεις HTTP και αλληλοεπιδρά με τη βάση δεδομένων μέσω της βιβλιοθήκης PyMongo.

Για την εκτέλεση της εφαρμογής, απαιτείται η χρήση του XAMPP ως τοπικού διακομιστή ανάπτυξης, ο οποίος διαχειρίζεται τις συνδέσεις της εφαρμογής με τη βάση δεδομένων MongoDB.

Επιπλέον, η ανάπτυξη και εκτέλεση της εφαρμογής πραγματοποιούνται σε ένα εικονικό περιβάλλον (virtual environment) για τη διαχείριση των εξαρτήσεων της Python, διασφαλίζοντας ότι οι απαιτούμενες βιβλιοθήκες και εκδόσεις είναι διαθέσιμες και απομονωμένες από άλλες εφαρμογές. Για όλα αυτά θα δοθούν στην συνέχεια αναλυτικές οδηγίες. **Πάμε να δούμε ένα-ένα τα APIs μας.**

1. Endpoint: / (Root Page)

- **Λειτουργία:** Ανακατευθύνει τον χρήστη στην κύρια σελίδα του γυμναστηρίου.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:** Δεν υπάρχουν.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Αν υπάρξει σφάλμα, εμφανίζεται μήνυμα σφάλματος μέσω flash.

2. Endpoint: /mainpage (Main Page)

- **Λειτουργία:** Επιστρέφει την κύρια σελίδα του site χωρίς να απαιτείται σύνδεση.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:** Δεν υπάρχουν.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Αν υπάρξει σφάλμα, εμφανίζεται μήνυμα σφάλματος μέσω flash.

3. Endpoint: /userlogin (User Login)

- **Λειτουργία:** Διαχειρίζεται τη σύνδεση του χρήστη.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - **POST:**
 - username: Το όνομα χρήστη του χρήστη που προσπαθεί να συνδεθεί.
 - password: Ο κωδικός πρόσβασης του χρήστη.
- **Σώμα:** Το σώμα της POST κλήσης περιέχει τα στοιχεία σύνδεσης του χρήστη (username, password).

- **Σημειώσεις:**
 - Αν ο χρήστης είναι ήδη συνδεδεμένος, ανακατευθύνεται στην κύρια σελίδα χρήστη.
 - Αν τα στοιχεία σύνδεσης είναι λανθασμένα, εμφανίζεται μήνυμα σφάλματος.

4. Endpoint: /adminlogin (Admin Login)

- **Λειτουργία:** Διαχειρίζεται τη σύνδεση του διαχειριστή.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - **POST:**
 - username: Το όνομα χρήστη του διαχειριστή που προσπαθεί να συνδεθεί.
 - password: Ο κωδικός πρόσβασης του διαχειριστή.
- **Σώμα:** Το σώμα της POST κλήσης περιέχει τα στοιχεία σύνδεσης του διαχειριστή (username, password).
- **Σημειώσεις:**
 - Αν ο διαχειριστής είναι ήδη συνδεδεμένος, ανακατευθύνεται στην κύρια σελίδα διαχειριστή.
 - Αν τα στοιχεία σύνδεσης είναι λανθασμένα, εμφανίζεται μήνυμα σφάλματος.

5. Endpoint: /logout (Logout)

- **Λειτουργία:** Διαχειρίζεται την αποσύνδεση του χρήστη.
- **Μέθοδος Κλήσης:** GET
- **Παράμετροι:** Δεν υπάρχουν.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Αν ο χρήστης είναι συνδεδεμένος, το session διαγράφεται και ο χρήστης ανακατευθύνεται στην κύρια σελίδα.

6. Endpoint: /userlogout (User Logout)

- **Λειτουργία:** Διαχειρίζεται την αποσύνδεση του χρήστη.
- **Μέθοδος Κλήσης:** GET
- **Παράμετροι:** Δεν υπάρχουν.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Αν ο χρήστης είναι συνδεδεμένος, το session διαγράφεται και ο χρήστης ανακατευθύνεται στην κύρια σελίδα.

7. Endpoint: /getRequests (Get Requests)

- **Λειτουργία:** Επιστρέφει τη λίστα των αιτημάτων εγγραφής χρηστών για τον διαχειριστή.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:** Δεν υπάρχουν.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Επιστρέφει τη λίστα των αιτημάτων ή ένα μήνυμα ότι δεν υπάρχουν αιτήματα.

8. Endpoint: /requests/decline/<email> (Decline Request)

- **Λειτουργία:** Διαγράφει το αίτημα χρήστη με συγκεκριμένο email.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - email: Το email του χρήστη του οποίου το αίτημα απορρίπτεται.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Διαγράφει το αίτημα και ανακατευθύνει στη λίστα αιτημάτων.

9. Endpoint: /oneUser/<email> (One User Info)

- **Λειτουργία:** Επιστρέφει τις πληροφορίες ενός αιτήματος χρήστη με συγκεκριμένο email.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - email: Το email του χρήστη του οποίου οι πληροφορίες ζητούνται.
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Επιστρέφει τις πληροφορίες του χρήστη ή ένα μήνυμα ότι ο χρήστης δεν υπάρχει.

10. Endpoint: /requests/accept/<email> (Accept Request)

- **Λειτουργία:** Αποδέχεται το αίτημα εγγραφής χρήστη με συγκεκριμένο email.
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - email: Το email του χρήστη του οποίου το αίτημα αποδέχεται.
 - role: Ο ρόλος που θα ανατεθεί στον χρήστη (από τη φόρμα).
- **Header:** Δεν καθορίζεται ειδικό header.
- **Σημειώσεις:**
 - Δημιουργεί νέο χρήστη στη βάση και διαγράφει το αίτημα εγγραφής.

11. Get the Users List

- **Λειτουργία:** Επιστρέφει τη λίστα των χρηστών.
- **Endpoint:** /userlist/<flag>
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - flag (URL parameter): Χρησιμοποιείται για την εμφάνιση μηνύματος σφάλματος μετά την ενημέρωση.
- **Σημείωση:** Εάν δεν υπάρχουν εγγεγραμμένοι χρήστες, επιστρέφει κενό πίνακα με μήνυμα.

12. Delete User

- **Λειτουργία:** Διαγράφει έναν χρήστη με το συγκεκριμένο user_id.
- **Endpoint:** /deleteUser/<user_id>
- **Μέθοδος Κλήσης:** DELETE, GET
- **Παράμετροι:**
 - user_id (URL parameter): Το ID του χρήστη που θα διαγραφεί.
- **Σημείωση:** Εάν ο χρήστης δεν υπάρχει, επιστρέφει μήνυμα σφάλματος.

13. Edit User

- **Λειτουργία:** Επιστρέφει πληροφορίες για το χρήστη με το συγκεκριμένο user_id.
- **Endpoint:** /edit/<user_id>
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - user_id (URL parameter): Το ID του χρήστη του οποίου οι πληροφορίες θα επεξεργαστούν.
- **Σημείωση:** Αν ο χρήστης δεν υπάρχει, επιστρέφει μήνυμα σφάλματος.

14. Update User

- **Λειτουργία:** Ενημερώνει τις πληροφορίες ενός χρήστη με το συγκεκριμένο user_id.
- **Endpoint:** /updateuser/<user_id>
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - user_id (URL parameter): Το ID του χρήστη που θα ενημερωθεί.
- **Body:**
 - name, surname, country, town, username, password, email, address, role: Οι νέες τιμές για τον χρήστη.
- **Σημείωση:** Ελέγχει για διπλές εγγραφές email και username πριν από την ενημέρωση.

15. Get Countries

- **Λειτουργία:** Επιστρέφει τη λίστα των χωρών από ένα online API.
- **Endpoint:** /api/countries
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:** Δεν υπάρχουν.
- **Σημείωση:** Καλεί ένα εξωτερικό API και επιστρέφει τη λίστα των χωρών.

16. Get Cities

- **Λειτουργία:** Επιστρέφει τη λίστα των πόλεων για μια συγκεκριμένη χώρα από ένα online API.
- **Endpoint:** /api/cities/<country>
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - country (URL parameter): Το όνομα της χώρας για την οποία θα επιστραφούν οι πόλεις.

- **Σημείωση:** Εάν η χώρα δεν βρεθεί, επιστρέφει έναν κενό πίνακα.

17. Drop Cancellations

- **Λειτουργία:** Διαγράφει όλες τις καταχωρίσεις ακυρώσεων από τη βάση δεδομένων.
- **Endpoint:** /dropcancels
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:** Δεν υπάρχουν.
- **Σημείωση:** Απαιτεί σύνδεση ως διαχειριστής.

18. Get Announcements

- **Λειτουργία:** Επιστρέφει τη λίστα των ανακοινώσεων.
- **Endpoint:** /announcementlist
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:** Δεν υπάρχουν.
- **Σημείωση:** Εάν δεν υπάρχουν ανακοινώσεις, επιστρέφει μήνυμα κενής βάσης δεδομένων.

19. Delete Announcement

- **Λειτουργία:** Διαγράφει μια ανακοίνωση με βάση το ann_id.
- **Endpoint:** /deleteAnnouncement/<ann_id>
- **Μέθοδος Κλήσης:** DELETE, GET
- **Παράμετροι:**
 - ann_id: Το ID της ανακοίνωσης που θα διαγραφεί.

20. Edit Announcement

- **Λειτουργία:** Επιστρέφει τις πληροφορίες μιας ανακοίνωσης με βάση το ann_id για επεξεργασία.
- **Endpoint:** /editAnnouncement/<ann_id>
- **Μέθοδος Κλήσης:** GET, POST

- **Παράμετροι:**
 - **ann_id:** Το ID της ανακοίνωσης της οποίας οι πληροφορίες θα εμφανιστούν.

21. Update Announcement

- **Λειτουργία:** Ενημερώνει τα στοιχεία μιας ανακοίνωσης με βάση το ann_id.
- **Endpoint:** /updateAnnouncement/<ann_id>
- **Μέθοδος Κλήσης:** POST
- **Παράμετροι:**
 - **ann_id:** Το ID της ανακοίνωσης της οποίας τα στοιχεία θα ενημερωθούν.
- **Σώμα:** Δεδομένα φόρμας (π.χ. title, text).

22. Create Announcement

- **Λειτουργία:** Δημιουργεί μια νέα ανακοίνωση.
- **Endpoint:** /createannouncement
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:** Κανένα.
- **Σώμα:** Δεδομένα φόρμας (π.χ. title, text).

23. Get Offers

- **Λειτουργία:** Επιστρέφει τη λίστα των προσφορών.
- **Endpoint:** /offerlist
- **Μέθοδος Κλήσης:** POST, GET

24. Delete Offer

- **Λειτουργία:** Διαγράφει μια προσφορά με βάση το offer_id.
- **Endpoint:** /deleteOffer/<offer_id>
- **Μέθοδος Κλήσης:** DELETE, GET
- **Παράμετροι:**

- **offer_id**: Το ID της προσφοράς που θα διαγραφεί.

25. Edit Offer

- **Λειτουργία**: Ανακτά τις πληροφορίες για μια συγκεκριμένη προσφορά και τις επιστρέφει στο frontend.
- **Endpoint**: /editOffer/<offer_id>
- **Μέθοδος**: GET, POST
- **Παράμετροι**:
 - **offer_id** (Παράμετρος URL): Το ID της προσφοράς που θα ανακτηθεί.
- **Σώμα**: Δεν υπάρχει σώμα για τα αιτήματα GET. Για τα αιτήματα POST, το σώμα χρησιμοποιείται εάν ο πελάτης στείλει δεδομένα για ενημέρωση της προσφοράς.
- **Περιγραφή**:
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Επαληθεύει την ύπαρξη της προσφοράς με το συγκεκριμένο offer_id.
 - Εάν βρεθεί, ανακτά τις πληροφορίες της προσφοράς και επιστρέφει τη σελίδα "Offers-Update" με αυτές τις πληροφορίες.

26. Update Offer

- **Λειτουργία**: Ενημερώνει μια συγκεκριμένη προσφορά με τις παρεχόμενες λεπτομέρειες.
- **Endpoint**: /updateOffer/<offer_id>
- **Μέθοδος**: POST
- **Παράμετροι**:
 - **offer_id** (Παράμετρος URL): Το ID της προσφοράς που θα ενημερωθεί.
- **Σώμα**:

- title: Ο νέος τίτλος της προσφοράς.
- text: Το νέο κείμενο της προσφοράς.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Ανακτά τα δεδομένα της φόρμας από το αίτημα για να ενημερώσει την προσφορά.
 - Ενημερώνει την προσφορά στη βάση δεδομένων με τα νέα δεδομένα.

27. New Offer Form

- **Λειτουργία:** Εμφανίζει τη φόρμα για τη δημιουργία μιας νέας προσφοράς.
- **Endpoint:** /newoffer
- **Μέθοδος:** GET, POST
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Επιστρέφει τη σελίδα "Offers-New" για τη δημιουργία μιας νέας προσφοράς.

28. Create Offer

- **Λειτουργία:** Δημιουργεί μια νέα προσφορά και την αποθηκεύει στη βάση δεδομένων.
- **Endpoint:** /createoffer
- **Μέθοδος:** POST, GET
- **Παράμετροι:** Καμία.
- **Σώμα:**
 - title: Ο τίτλος της νέας προσφοράς.
 - text: Το κείμενο της νέας προσφοράς.

- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Δημιουργεί ένα νέο offer_id βάσει των υπαρχουσών προσφορών στη βάση δεδομένων.
 - Εισάγει τη νέα προσφορά στη βάση δεδομένων.

29. Trainer List

- **Λειτουργία:** Ανακτά και εμφανίζει μια λίστα όλων των γυμναστών.
- **Endpoint:** /trainerlist
- **Μέθοδος:** GET, POST
- **Παράμετροι:** Καμία.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Ανακτά όλους τους γυμναστές από τη βάση δεδομένων.
 - Επιστρέφει τη σελίδα "Trainers-List" με τα δεδομένα που ανακτήθηκαν.

30. Delete Trainer

- **Λειτουργία:** Διαγράφει έναν συγκεκριμένο γυμναστή βάσει του παρεχόμενου trainer_id.
- **Endpoint:** /deleteTrainer/<trainer_id>
- **Μέθοδος:** DELETE, GET
- **Παράμετροι:**
 - trainer_id (Παράμετρος URL): Το ID του γυμναστή που θα διαγραφεί.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.

- Επαληθεύει την ύπαρξη του γυμναστή με το συγκεκριμένο trainer_id.
- Διαγράφει τον γυμναστή από τη βάση δεδομένων.

31. Edit Trainer

- **Λειτουργία:** Ανακτά τις πληροφορίες για έναν συγκεκριμένο γυμναστή και τις επιστρέφει στο frontend για επεξεργασία.
- **Endpoint:** /editTrainer/<trainer_id>
- **Μέθοδος:** GET, POST
- **Παράμετροι:**
 - trainer_id (Παράμετρος URL): Το ID του γυμναστή που θα ανακτηθεί.
- **Σώμα:** Δεν υπάρχει σώμα για τα αιτήματα GET. Για τα αιτήματα POST, το σώμα χρησιμοποιείται εάν ο πελάτης στείλει δεδομένα για ενημέρωση του γυμναστή.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Επαληθεύει την ύπαρξη του γυμναστή με το συγκεκριμένο trainer_id.
 - Εάν βρεθεί, ανακτά τις πληροφορίες του γυμναστή και επιστρέφει τη σελίδα "Trainers-Update" με αυτές τις πληροφορίες.

32. Trainer Update

- **Λειτουργία:** Ενημερώνει έναν συγκεκριμένο γυμναστή με τις παρεχόμενες λεπτομέρειες.
- **Endpoint:** /updateTrainer/<trainer_id>
- **Μέθοδος:** POST
- **Παράμετροι:**

- `trainer_id` (Παράμετρος URL): Το ID του γυμναστή που θα ενημερωθεί.
- **Σώμα:**
 - `name`: Το ενημερωμένο όνομα του γυμναστή.
 - `surname`: Το ενημερωμένο επώνυμο του γυμναστή.
 - `programs`: Τα ενημερωμένα προγράμματα που αναλαμβάνει ο γυμναστής.
 - `bio`: Το ενημερωμένο βιογραφικό του γυμναστή.
 - `email`: Το ενημερωμένο email του γυμναστή.
 - `phone`: Το ενημερωμένο τηλέφωνο του γυμναστή.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.
 - Ανακτά τα δεδομένα της φόρμας από το αίτημα για να ενημερώσει τον γυμναστή.
 - Ενημερώνει τον γυμναστή στη βάση δεδομένων με τα νέα δεδομένα.

33. New Trainer

- **Λειτουργία:** Εμφανίζει τη φόρμα για τη δημιουργία ενός νέου γυμναστή.
- **Endpoint:** `/newtrainer`
- **Μέθοδος:** GET, POST
- **Παράμετροι:** Καμία.
- **Header:** Δεν απαιτούνται ειδικά headers.
- **Σώμα:** Δεν εφαρμόζεται.
- **Περιγραφή:**
 - Ελέγχει αν ο χρήστης είναι συνδεδεμένος.

- ο Επιστρέφει τη σελίδα "Trainers-New" για τη δημιουργία ενός νέου γυμναστή.

34. Program List

- **Λειτουργία:** Επιστρέφει τη λίστα των διαθέσιμων προγραμμάτων.
- **Endpoint:** /programlist
- **Μέθοδος:** POST, GET
- **Παράμετροι:** Καμία

35. Delete Program

- **Λειτουργία:** Διαγράφει ένα πρόγραμμα με συγκεκριμένο program_id.
- **Endpoint:** /deleteProgram/<program_id>
- **Μέθοδος:** DELETE, GET
- **Παράμετροι:**
 - ο program_id (στο URL): Το ID του προγράμματος που θα διαγραφεί.

36. Edit program

- **Λειτουργία:** Επιστρέφει τις πληροφορίες ενός προγράμματος με συγκεκριμένο program_id για να επεξεργαστεί.
- **Endpoint:** /editProgram/<program_id>
- **Μέθοδος:** GET, POST
- **Παράμετροι:**
 - ο program_id (στο URL): Το ID του προγράμματος που θα επεξεργαστεί.

37. update program

- **Λειτουργία:** Ενημερώνει τις πληροφορίες ενός προγράμματος με συγκεκριμένο program_id.
- **Endpoint:** /updateProgram/<program_id>
- **Μέθοδος:** POST
- **Παράμετροι:**
 - program_id (στο URL): Το ID του προγράμματος που θα ενημερωθεί.
- **Σώμα:**
 - name: Το νέο όνομα του προγράμματος.
 - text: Το νέο κείμενο περιγραφής του προγράμματος.

38. new program form

- **Λειτουργία:** Εμφανίζει τη φόρμα για τη δημιουργία νέου προγράμματος.
- **Endpoint:** /newprogram
- **Μέθοδος:** POST, GET

39. create new program

- **Λειτουργία:** Δημιουργεί ένα νέο πρόγραμμα και το αποθηκεύει στη βάση δεδομένων.
- **Endpoint:** /createprogram
- **Μέθοδος:** POST, GET
- **Σώμα:**
 - name: Το όνομα του νέου προγράμματος.
 - text: Το κείμενο περιγραφής του νέου προγράμματος.

40. class list

- **Λειτουργία:** Επιστρέφει τη λίστα των διαθέσιμων τάξεων.
- **Endpoint:** /classlist
- **Μέθοδος:** POST, GET

41. Delete class

- **Λειτουργία:** Διαγράφει μια τάξη με συγκεκριμένο class_id.
- **Endpoint:** /deleteClass/<class_id>
- **Μέθοδος:** DELETE, GET
- **Παράμετροι:**
 - class_id (στο URL): Το ID της τάξης που θα διαγραφεί.

42. edit class

- **Λειτουργία:** Επιστρέφει τις πληροφορίες μιας τάξης με συγκεκριμένο class_id για να επεξεργαστεί.
- **Endpoint:** /editClass/<class_id>
- **Μέθοδος:** GET, POST
- **Παράμετροι:**
 - class_id (στο URL): Το ID της τάξης που θα επεξεργαστεί.

43. Update class

- **Λειτουργία:** Ενημερώνει τις πληροφορίες μιας τάξης με συγκεκριμένο class_id.
- **Endpoint:** /updateClass/<class_id>
- **Μέθοδος:** POST
- **Παράμετροι:**
 - class_id (στο URL): Το ID της τάξης που θα ενημερωθεί.

- **Σώμα:**
 - program: Το πρόγραμμα της τάξης.
 - day: Η ημέρα της τάξης.
 - time: Η ώρα της τάξης.
 - type: Ο τύπος της τάξης.
 - max_people: Ο μέγιστος αριθμός συμμετεχόντων.
 - availablespots: Οι διαθέσιμες θέσεις.

44. form for new class

- **Λειτουργία:** Εμφανίζει τη φόρμα για τη δημιουργία νέας τάξης.
- **Endpoint:** /newclass
- **Μέθοδος:** POST, GET

45. Create new class

- **Λειτουργία:** Δημιουργεί μια νέα τάξη και την αποθηκεύει στη βάση δεδομένων.
- **Endpoint:** /createclass
- **Μέθοδος:** POST, GET
- **Σώμα:**
 - program: Το πρόγραμμα της νέας τάξης.
 - day: Η ημέρα της νέας τάξης.
 - time: Η ώρα της νέας τάξης.
 - type: Ο τύπος της νέας τάξης.
 - max_people: Ο μέγιστος αριθμός συμμετεχόντων.
 - availablespots: Οι διαθέσιμες θέσεις.

46. update class

Λειτουργία: Επαναφέρει τις διαθέσιμες θέσεις της τάξης στο μέγιστο αριθμό.

- **Endpoint:** /refreshclass/<class_id>
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - class_id (μέρος της διεύθυνσης URL): Το ID της τάξης που πρέπει να ανανεωθεί.
- **Επιστροφή:** Ανακατεύθυνση στη λίστα των τάξεων με μήνυμα επιτυχίας ή αποτυχίας.

47. Register form

Λειτουργία: Εμφανίζει τη φόρμα εγγραφής χρήστη.

- **Endpoint:** /formregister
- **Μέθοδος Κλήσης:** POST, GET
- **Επιστροφή:** Φόρμα εγγραφής χρήστη.

48. Create User

Λειτουργία: Δημιουργεί ένα νέο χρήστη με βάση τα δεδομένα της φόρμας.

- **Endpoint:** /register
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:**
 - name (μέρος του σώματος της αίτησης): Το όνομα του χρήστη.
 - surname (μέρος του σώματος της αίτησης): Το επώνυμο του χρήστη.

- country (μέρος του σώματος της αίτησης): Η χώρα του χρήστη.
- town (μέρος του σώματος της αίτησης): Η πόλη του χρήστη.
- email (μέρος του σώματος της αίτησης): Το email του χρήστη.
- address (μέρος του σώματος της αίτησης): Η διεύθυνση του χρήστη.
- username (μέρος του σώματος της αίτησης): Το όνομα χρήστη.
- password (μέρος του σώματος της αίτησης): Ο κωδικός πρόσβασης του χρήστη.
- **Σώμα:** Η φόρμα εγγραφής με τα δεδομένα του χρήστη.
- **Επιστροφή:** Ανακατεύθυνση στη κύρια σελίδα με μήνυμα επιτυχίας ή αποτυχίας.

49. Browse program

Λειτουργία: Εμφανίζει τη λίστα των προγραμμάτων χωρίς να απαιτείται σύνδεση.

- **Endpoint:** /browsePrograms
- **Μέθοδος Κλήσης:** GET
- **Επιστροφή:** Σελίδα με τη λίστα των προγραμμάτων.

50. browse program with login

Λειτουργία: Εμφανίζει τη λίστα των προγραμμάτων για συνδεδεμένους χρήστες.

- **Endpoint:** /userbrowsePrograms
- **Μέθοδος Κλήσης:** GET
- **Επιστροφή:** Σελίδα με τη λίστα των προγραμμάτων για συνδεδεμένους χρήστες.

51. Announcements Browse

Λειτουργία: Εμφανίζει τη λίστα των ανακοινώσεων για συνδεδεμένους χρήστες.

- **Endpoint:** /AnnouncementsBrowse
- **Μέθοδος Κλήσης:** GET
- **Επιστροφή:** Σελίδα με τη λίστα των ανακοινώσεων.

52. Offer browse

Λειτουργία: Εμφανίζει τη λίστα των προσφορών για συνδεδεμένους χρήστες.

- **Endpoint:** /OffersBrowse
- **Μέθοδος Κλήσης:** GET
- **Παράμετροι:** Όχι απαραίτητοι.
- **Επιστροφή:** Σελίδα με τη λίστα των προσφορών.

53. book form

Λειτουργία: Εμφανίζει τη φόρμα για την επιλογή του προγράμματος για ραντεβού.

- **Endpoint:** /formofBook
- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:** Όχι απαραίτητοι.
- **Επιστροφή:** Φόρμα επιλογής προγράμματος με διαθέσιμα προγράμματα.

54. appointment hours

Λειτουργία: Εμφανίζει διαθέσιμες ώρες για το επιλεγμένο πρόγραμμα και ημερομηνία.

- **Endpoint:** /showtimes

- **Μέθοδος Κλήσης:** GET, POST
- **Παράμετροι:**
 - `available_classes` (μέρος του σώματος της αίτησης): Το επιλεγμένο πρόγραμμα.
 - `date` (μέρος του σώματος της αίτησης): Η επιλεγμένη ημερομηνία.
- **Σώμα:** Φόρμα με επιλεγμένο πρόγραμμα και ημερομηνία.
- **Επιστροφή:** Σελίδα με τις διαθέσιμες ώρες για το πρόγραμμα και την ημερομηνία που επιλέχθηκε.

55. book appointment

Λειτουργία: Υποβάλλει ένα νέο ραντεβού για τον χρήστη, ελέγχοντας τη διαθεσιμότητα θέσεων και την εγκυρότητα της ημερομηνίας.

- **Endpoint:** `/submitAppointment`
- **Μέθοδος Κλήσης:** POST
- **Παράμετροι:**
 - `class_name` (μέρος του σώματος της αίτησης): Το όνομα της τάξης για την οποία γίνεται το ραντεβού.
 - `date` (μέρος του σώματος της αίτησης): Η ημερομηνία του ραντεβού.
 - `times` (μέρος του σώματος της αίτησης): Η ώρα του ραντεβού.
- **Σώμα:** Περιέχει τα πεδία `class_name`, `date`, και `times`.
- **Επιστροφή:** Ανακατεύθυνση στη σελίδα με τα ραντεβού του χρήστη με μήνυμα επιτυχίας ή αποτυχίας.

56. list of appointments

Λειτουργία: Εμφανίζει τη λίστα των ενεργών ραντεβού του χρήστη. Μεταφέρει τα παλαιά ραντεβού σε άλλη συλλογή εάν η ημερομηνία τους είναι παρελθοντική.

- **Endpoint:** /myAppointments
- **Μέθοδος Κλήσης:** POST, GET
- **Παράμετροι:** Όχι απαραίτητοι.
- **Επιστροφή:** Σελίδα με τα ενεργά ραντεβού του χρήστη.

57. old appointments

Λειτουργία: Εμφανίζει τη λίστα με τα παλιά ραντεβού του χρήστη. Ενδέχεται να γίνει και διαγραφή των ραντεβού από την κύρια συλλογή.

- **Endpoint:** /olderAp
- **Μέθοδος Κλήσης:** GET, DELETE
- **Παράμετροι:** Όχι απαραίτητοι.
- **Επιστροφή:** Σελίδα με τα παλιά ραντεβού του χρήστη.

58. cancel appointments

Λειτουργία: Ακυρώνει ένα συγκεκριμένο ραντεβού, εάν δεν είναι προγραμματισμένο να πραγματοποιηθεί σε λιγότερο από 2 ώρες. Ενημερώνει τη διαθεσιμότητα θέσεων της τάξης.

- **Endpoint:** /cancel/<app_id>
- **Μέθοδος Κλήσης:** DELETE, POST, GET
- **Παράμετροι:**
 - app_id (μέρος της διεύθυνσης URL): Το ID του ραντεβού που πρόκειται να ακυρωθεί.
- **Επιστροφή:** Ανακατεύθυνση στη λίστα ραντεβού με μήνυμα επιτυχίας ή αποτυχίας.

Κεφάλαιο 4: Εγχειρίδιο Διαχειριστή

Για την ανάπτυξη της εφαρμογής, χρησιμοποιήθηκε το **XAMPP**, ένα λογισμικό που περιλαμβάνει Apache server το οποίο επιτρέπει την εύκολη διαχείριση και εκτέλεση της εφαρμογής σε τοπικό διακομιστή (localhost).

Η βάση δεδομένων διαχειρίζεται μέσω του **MongoDB Compass**, ένα εργαλείο γραφικού περιβάλλοντος που επιτρέπει την εύκολη προβολή και διαχείριση των δεδομένων της MongoDB. Στη συγκεκριμένη περίπτωση, η βάση δεδομένων ονομάζεται "Gym" και περιέχει διάφορες συλλογές (collections) που οργανώνουν τα δεδομένα. Η υλοποίηση των API της εφαρμογής έγινε με τη χρήση της γλώσσας προγραμματισμού Python, ενώ το frontend σχεδιάστηκε με τη χρήση HTML, CSS, Bootstrap και JavaScript, παρέχοντας έτσι μια ολοκληρωμένη και λειτουργική εμπειρία χρήστη.

Μπορείτε να κατεβάσετε το XAMPP από την επίσημη ιστοσελίδα εδώ: <https://www.apachefriends.org/download.html>

Μπορείτε να κατεβάσετε το MongoDB Compass από την επίσημη ιστοσελίδα εδώ: <https://www.mongodb.com/products/tools/compass>

Για να τρέξει η εφαρμογή μέσω XAMPP, είναι απαραίτητο να μεταφέρουμε τον φάκελο της εργασίας στον φάκελο "htdocs" του XAMPP.

Ο φάκελος "htdocs" είναι ο προκαθορισμένος κατάλογος όπου ο Apache server αναζητά αρχεία για να τα τρέξει ως ιστοσελίδες όταν ανοίγουμε τη διεύθυνση "localhost" στον περιηγητή μας. Τοποθετώντας τον φάκελο της εργασίας στο "htdocs", διασφαλίζουμε ότι ο Apache μπορεί να βρει και να τρέξει την εφαρμογή μας σωστά.

Για να προετοιμάσουμε το περιβάλλον ανάπτυξης (environment), είναι σημαντικό να ανοίξουμε το Visual Studio Code και να εκτελέσουμε τις ακόλουθες εντολές στο terminal:

`python -m venv myenv`

και

```
PS C:\xampp\htdocs\ergasia> & c:/xampp/htdocs/ergasia/myenv/Scripts/Activate.ps1
(myenv) PS C:\xampp\htdocs\ergasia> python -m venv myenv
(myenv) PS C:\xampp\htdocs\ergasia> myenv\Scripts\activate
(myenv) PS C:\xampp\htdocs\ergasia> |
```

Με την πρώτη εντολή, δημιουργούμε ένα εικονικό περιβάλλον (virtual environment) με το όνομα "myenv". Το εικονικό περιβάλλον είναι ένας απομονωμένος χώρος μέσα στον οποίο μπορούμε να εγκαταστήσουμε τις συναρτήσεις της εφαρμογής μας (δηλαδή βιβλιοθήκες Python) χωρίς να επηρεάσουμε το υπόλοιπο σύστημα ή άλλες εφαρμογές.

Με τη δεύτερη εντολή, ενεργοποιούμε το εικονικό περιβάλλον, ώστε οι επόμενες εντολές που θα εκτελούνται στο terminal να χρησιμοποιούν τις βιβλιοθήκες και ρυθμίσεις που βρίσκονται στο περιβάλλον αυτό. Αυτό διασφαλίζει ότι η εφαρμογή μας θα τρέχει πάντα με τις σωστές εκδόσεις των libraries και με τις ακριβείς ρυθμίσεις που έχουμε ορίσει.

Συνοπτικά, το εικονικό περιβάλλον είναι ένα απομονωμένο σύστημα που μας επιτρέπει να διαχειριζόμαστε τις βιβλιοθήκες και τις ρυθμίσεις της εφαρμογής μας με ακρίβεια, ενώ η τοποθέτηση του φακέλου της εφαρμογής στο "htdocs" επιτρέπει στον Apache server του XAMPP να την τρέξει σωστά μέσω localhost.

Μόλις το κάνουμε αυτό θα εγκαταστήσουμε ότι λείπει από το περιβάλλον μας. Δηλαδή τις βιβλιοθήκες μας.

Στην περίπτωση μας εγκαθιστούμε flask , mongo και requests και τις εντολές:

1. Pip install flask
2. Pip install pymongo
3. Pip install requests

Πλέον , ανοίγουμε το αρχείο app.py στο vs code , πατάμε run και είμαστε έτοιμοι για να χρησιμοποιήσουμε την εφαρμογή μας!

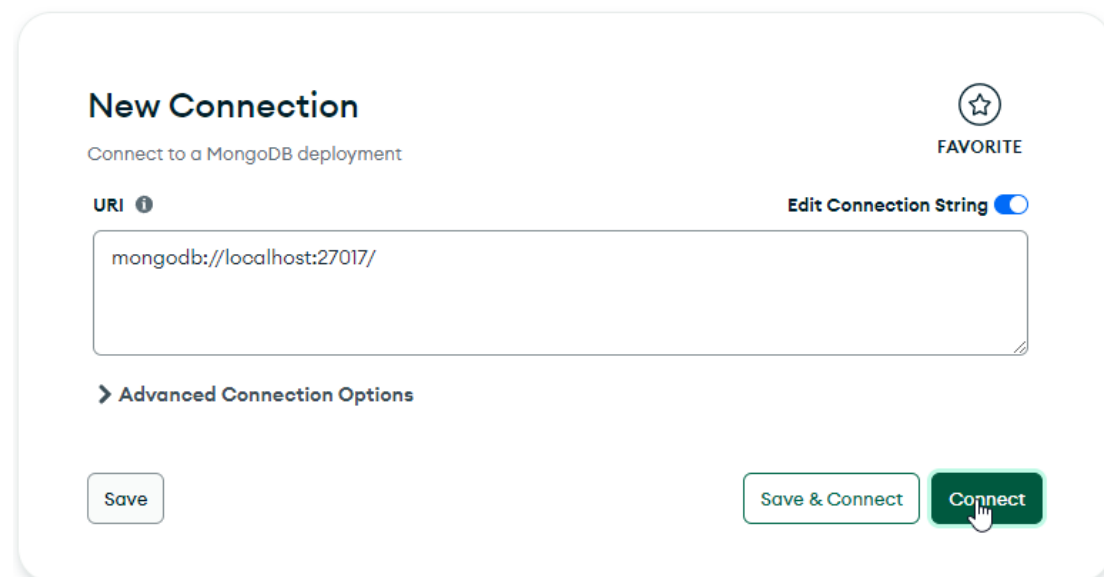
Πατάμε πάνω δεξιά:



Επιπλέον , ανοίγουμε το xampp και το module του Apache πρέπει να είναι σε κατάσταση run:



Στη συνέχεια βεβαιωνόμαστε πως έχουμε συνδέσει τη βάση μας και το MongoDB Compass τρέχει κανονικά:



Αφού τρέξουμε το `app.py` , στο terminal θα εμφανιστεί:

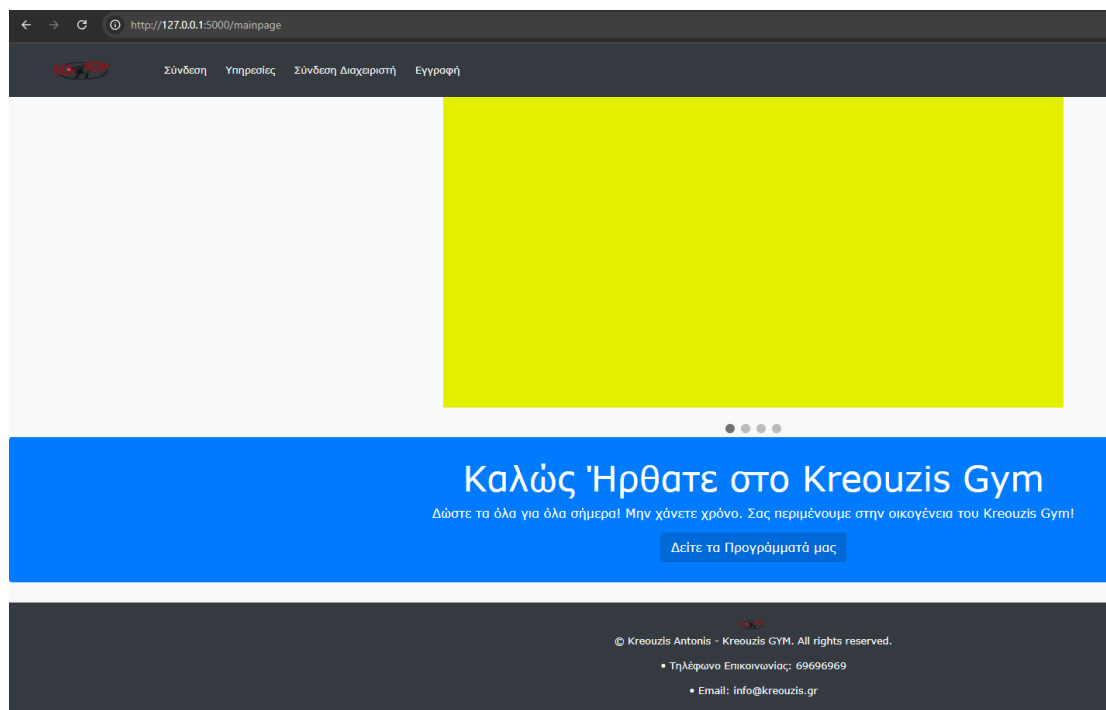
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER  COMMENTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\ergasia> & c:/xampp/htdocs/ergasia/myenv/Scripts/Activate.ps1
(myenv) PS C:\xampp\htdocs\ergasia> & c:/xampp/htdocs/ergasia/myenv/Scripts/python.exe c:/xampp/htdocs/ergasia/Flask/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.1.131:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 101-258-796
```

Παίρνουμε το <http://127.0.0.1:5000> και αφού το κάνουμε επικόλληση στον browser μας η εφαρμογή τρέχει κανονικά!

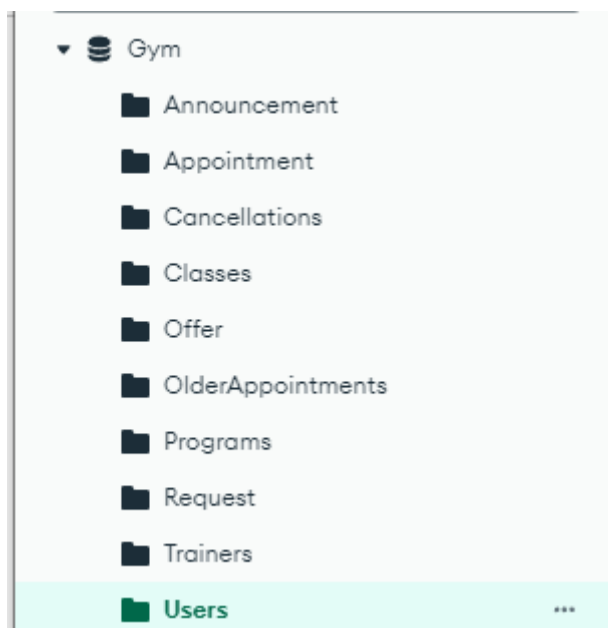


Η εφαρμογή εγκαταστάθηκε με επιτυχία!

Κεφάλαιο 5: Εγχειρίδιο Χρήστη

Πάμε να δούμε αναλυτικά πως δουλεύει η εφαρμογή μας καθώς και κάποια παραδείγματα με κάθε οθόνη.

Παρακάτω ακολουθεί η εικόνα με το πως είναι η βάση δεδομένων μου , η Gym.



Για να ξεκινήσουμε την εκτέλεση της εφαρμογής μας θα πάμε στην συλλογή (collection) Users και θα βάλουμε έναν **admin**:

```
_id: ObjectId('66ca0305f1da10f706bd4092')
name : "a"
surname : "a"
country : "Afghanistan"
town : "Herat"
email : "a@a.com"
address : "asd"
username : "a"
password : "1"
role : "admin"
user_id : "1"
```

Όπως βλέπουμε τα στοιχεία σύνδεσης του είναι:

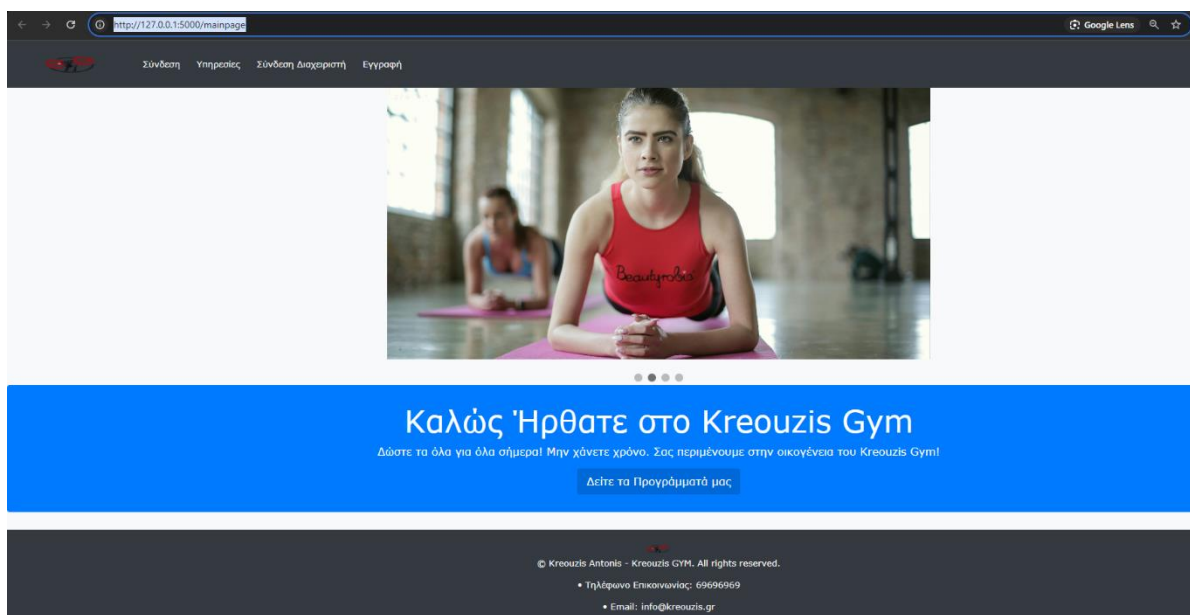
Username: a

Password: 1

Ξεκινάμε τώρα:

Ανοίγουμε τον browser και αφού κάναμε όλα τα βήματα του προηγούμενου κεφαλαίου , πάμε στην διεύθυνση:

<http://127.0.0.1:5000/mainpage> , αυτή είναι:



Όπως βλέπουμε είναι η αρχική σελίδα του γυμναστηρίου μας. Θα ξεκινήσουμε με την διαδικασία εγγραφής νέου χρήστη.

Διαδικασία εγγραφής νέου χρήστη:

Επιλέγουμε από το μενού την επιλογή «Εγγραφή» και μας εμφανίζεται μία φόρμα την οποία πρέπει να συμπληρώσουμε:

The screenshot shows a web browser window with the URL `http://127.0.0.1:5000/formregister`. The page has a dark header with navigation links: [Σύνδεση](#), [Υπηρεσίες](#), [Σύνδεση Διαχειριστή](#), and a blue button [Εγγραφή](#). The main content area is titled "Εγγραφή Χρήστη" and contains the following fields:

- Όνομα Χρήστη:**
- Επώνυμο Χρήστη:**
- Email Χρήστη:**
- Χώρα Κατοικίας:**
- Πόλη Κατοικίας:**
- Διεύθυνση Κατοικίας:**
- Username:**
- Password:**

At the bottom of the form are two buttons: a green "Δημιουργία Χρήστη" button and a red "Ακύρωση" button. The footer of the page contains the following text:

© Kreouzis Antonis - Kreouzis GYM. All rights reserved.
• Τηλέφωνο Επικοινωνίας: 69696969
• Email: info@kreouzis.gr


Αφού την συμπληρώσουμε θα πατήσουμε «Δημιουργία Χρήστη». Αφού πατήσουμε και το κουμπί, γίνεται redirect στην αρχική οθόνη.

Πηγαίνοντας στη βάση μας, θα δούμε ότι στο collection Request υπάρχει ο χρήστης και περιμένει από τον admin έγκριση ή απόρριψη.

The screenshot shows a database management interface. On the left is a sidebar menu with the following items: [Gym](#), [Announcement](#), [Appointment](#), [Cancellations](#), [Classes](#), [Offer](#), [OlderAppointments](#), [Programs](#), [Request](#) (highlighted in green), [Trainers](#), and [Users](#). At the top of the main area are four buttons: [ADD DATA](#), [EXPORT DATA](#), [UPDATE](#), and [DELETE](#). Below these buttons is a text box containing the following JSON data:

```
{
  "_id": ObjectId('66ce40d6d443675474ffaed8'),
  "name": "antonios",
  "surname": "kreouzis",
  "country": "Greece",
  "town": "Galatsi",
  "email": "kreouzis@gmail.com",
  "address": "Olympou 15 - Chalandri",
  "username": "kreouzis",
  "password": "123"
}
```

Όπως είμαστε στην αρχική σελίδα , επιλέγουμε «Σύνδεση Διαχειριστή».



Έχετε αποσυνδεθεί

Ψευδώνυμο

Συνθηματικό

Σύνδεση

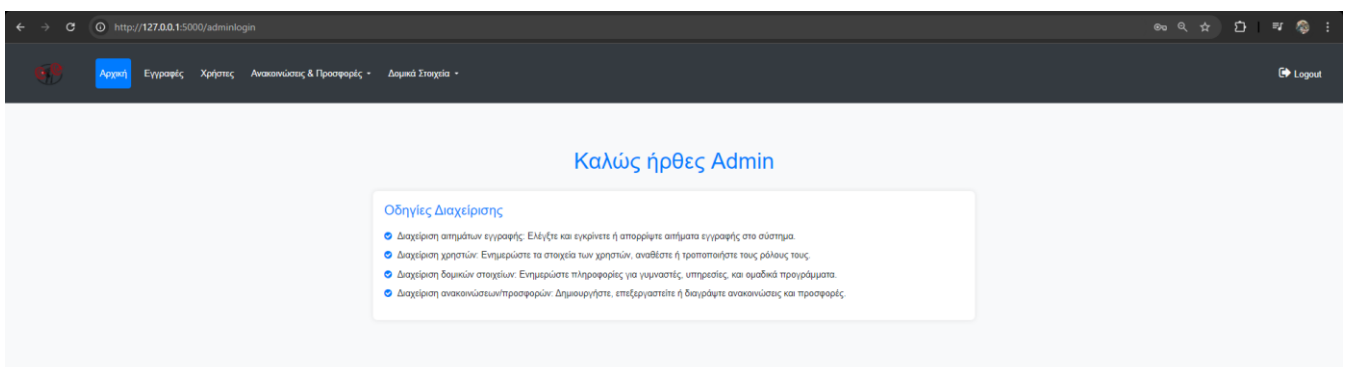
[Επιστροφή](#)

Συνδεόμαστε με τα στοιχεία:

Username: a

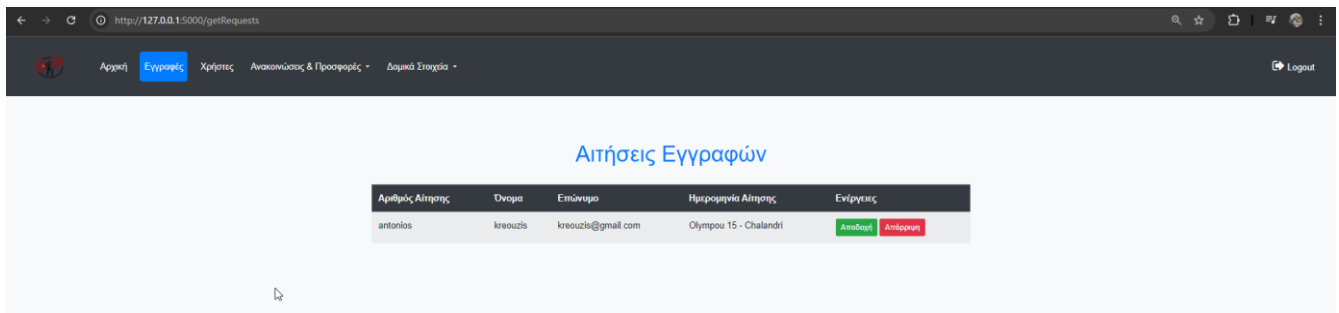
Password: 1

Αφού συνδεθούμε το μενού του διαχειριστή είναι κάπως έτσι:



The screenshot shows a web browser window with the URL `http://127.0.0.1:5000/adminlogin`. The page has a dark header with a navigation menu containing 'Αρχική', 'Εγγραφές', 'Χρήστες', 'Ανακοινώσεις & Προσφορές', and 'Διαμικά Στοιχεία'. A 'Logout' link is in the top right. The main content area has a light blue background with the heading 'Καλώς ήρθες Admin'. Below this is a box titled 'Οδηγίες Διαχείρισης' containing four bullet points: 1. Διαχείριση αιτημάτων εγγραφής: Ελέγχετε και εγκρίνετε ή απορρίπτετε αιτήματα εγγραφής στο σύστημα. 2. Διαχείριση χρηστών: Ενημερώστε τα στοιχεία των χρηστών, αναβάστε ή τροποποιήστε τους ρόλους τους. 3. Διαχείριση δομικών στοιχείων: Ενημερώστε πληροφορίες για γυμναστές, υπηρεσίες και ομάδες προγράμματα. 4. Διαχείριση ανακοινώσεων/προσφορών: Δημιουργήστε, επεξεργαστείτε ή διαγράψτε ανακοινώσεις και προσφορές.

Για να εγκρίνει ή να απορρίψει ένα αίτημα εγγραφής ο διαχειριστής επιλέγει από το μενού την επιλογή «Εγγραφές».

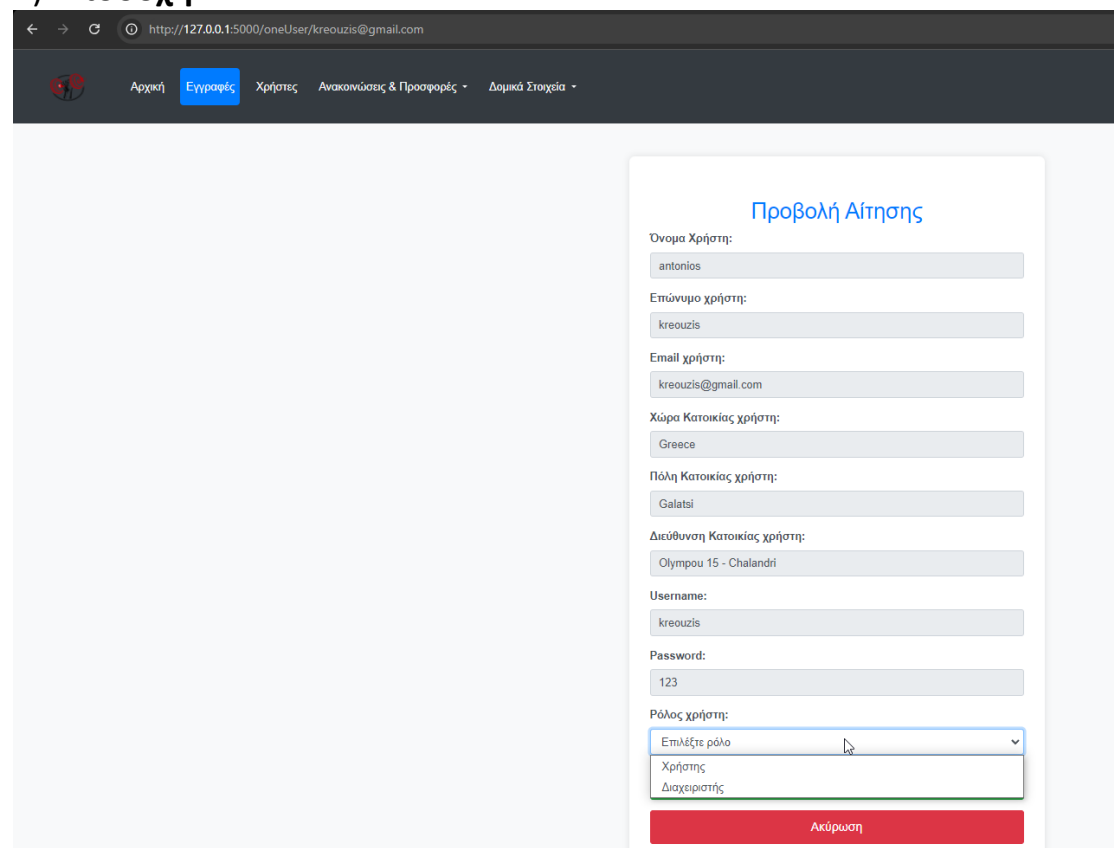


Εδώ ο διαχειριστής βλέπει τις αιτήσεις εγγραφών που έχουν γίνει. Έχει 2 επιλογές:

1) Απόρριψη:

Εάν επιλέξει απόρριψη η αίτηση διαγράφεται εντελώς – αυτό σημαίνει πως διαγράφεται και από το collection Request.

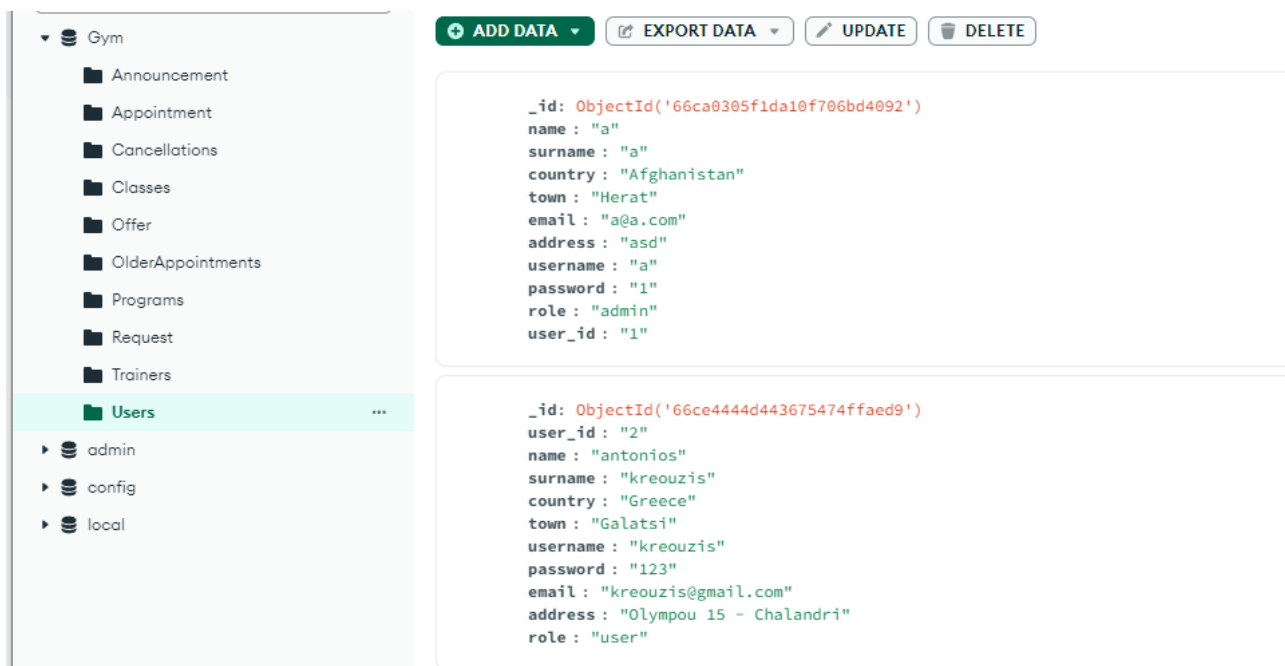
2) Αποδοχή:



Εδώ ο διαχειριστής πρέπει να επιλέξει τον ρόλο που θα έχει ο νέος χρήστης. Οι ρόλοι είναι 2:
Είτε απλός χρήστης , είτε διαχειριστής.

Εμείς διαχειριστή έχουμε, οπότε θα δώσουμε το ρόλο του Χρήστη και πατάμε το κουμπί που λέει «Αποθήκευση Αλλαγών».

Μόλις γίνει αυτό , η αίτηση διαγράφεται από το collection Request και ο χρήστης μεταφέρεται στο Collection Users:



The screenshot shows a web application interface for managing a database. On the left, there is a sidebar with a tree view under the 'Gym' category. The 'Users' collection is selected and highlighted in green. Below it, there are three sub-items: 'admin', 'config', and 'local'. On the right, there are four buttons: 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. Below these buttons, there are two JSON objects representing user records.

```
{
  "_id": "ObjectId('66ca0305f1da10f706bd4092')",
  "name": "a",
  "surname": "a",
  "country": "Afghanistan",
  "town": "Herat",
  "email": "a@a.com",
  "address": "asd",
  "username": "a",
  "password": "1",
  "role": "admin",
  "user_id": "1"
}
```

```
{
  "_id": "ObjectId('66ce444d443675474ffaed9')",
  "user_id": "2",
  "name": "antonios",
  "surname": "kreouzis",
  "country": "Greece",
  "town": "Galatsi",
  "username": "kreouzis",
  "password": "123",
  "email": "kreouzis@gmail.com",
  "address": "Olympou 15 - Chalandri",
  "role": "user"
}
```

Συνεχίζουμε με τις οθόνες του διαχειριστή -εφόσον είμαστε συνδεδεμένοι εδώ.

Η επιλογή «Χρήστες»:

Σε αυτή την επιλογή ο διαχειριστής βλέπει όλους τους χρήστες και τους διαχειριστές του συστήματος. Με άλλα λόγια βλέπει το περιεχόμενο του Collection Users.

Πίνακας Χρηστών									
ID	Name	Surname	Country	Town	Address	Email	Username	Role	Actions
1	a	a	Afghanistan	Herat	asd	a1@a.com	a	admin	Ενημέρωση Διαγραφή
2	antonios	kreouzis	Greece	Galatsi	Olympou 15 - Chalandri	kreouzis@gmail.com	kreouzis	user	Ενημέρωση Διαγραφή

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιου χρήστη επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:

Αρχική	Εγγραφές	Χρήστες	Ανακοινώσεις & Προσφορές	Δομικά Στοιχεία	Logout
--------	----------	---------	--------------------------	-----------------	--------

Ενημέρωση Χρήστη

ID Χρήστη:
2

Όνομα Χρήστη:
antonios

Επώνυμο χρήστη:
kreouzis

Email χρήστη:
kreouzis@gmail.com

Χώρα Κατοικίας χρήστη:
Greece

Πόλη Κατοικίας χρήστη:
Aegina

Διεύθυνση Κατοικίας χρήστη:
Olympou 15 - Chalandri

Username:
kreouzis

Password:
123

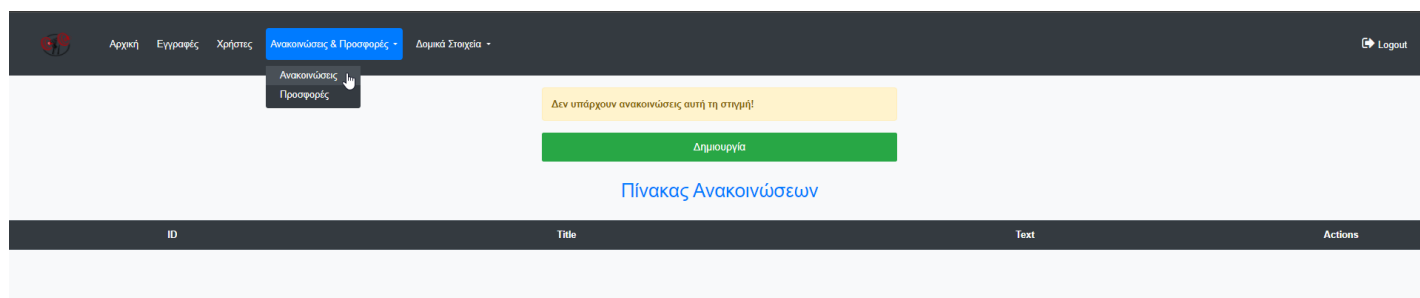
Ρόλος χρήστη:
user

[Αποθήκευση Αλλαγών](#) [Ακύρωση](#)

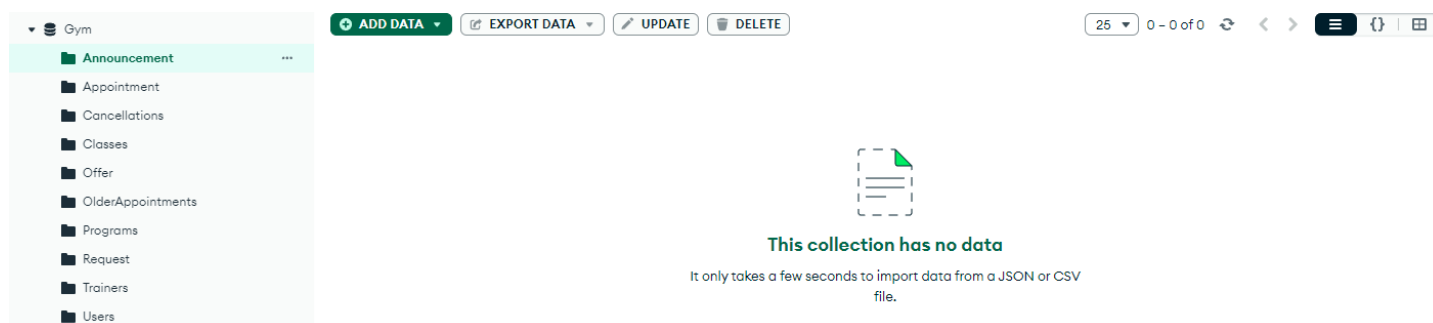
Επιπλέον μπορεί να διαγράψει όποιον χρήστη επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει τον χρήστη και από το Collection Users.

Η επιλογή «Ανακοινώσεις & Προσφορές»:

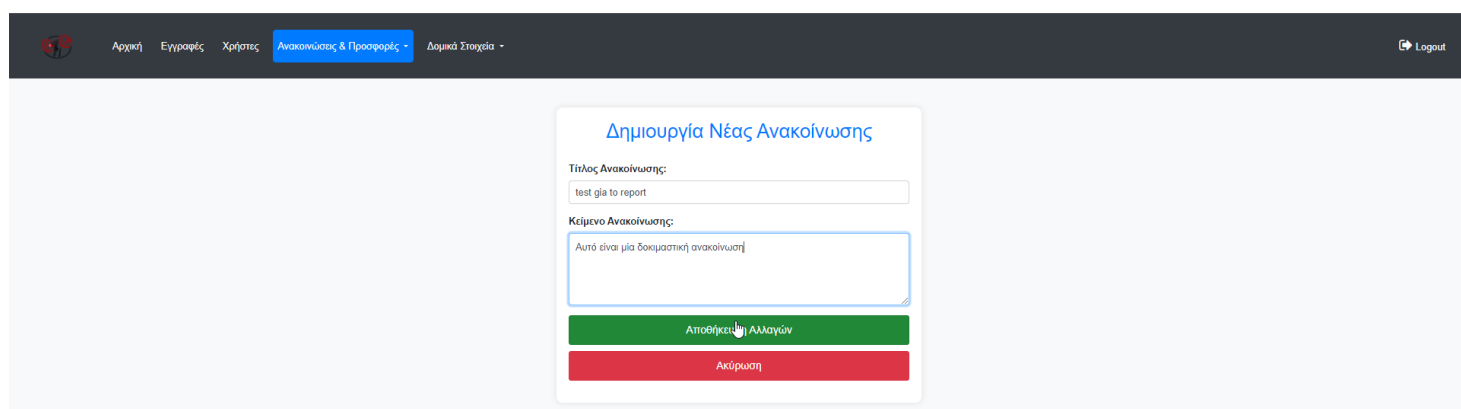
Υπό επιλογή -> Ανακοινώσεις:



Οι ανακοινώσεις αποθηκεύονται στη βάση στο collection Announcement. Δεν υπάρχει καμία ανακοίνωση όπως βλέπουμε στο UI της σελίδας και το επαληθεύουμε κοιτάζοντας το collection το οποίο είναι άδειο:



Πατώντας το κουμπί «Δημιουργία» πάμε να δημιουργήσουμε μία νέα ανακοίνωση:



Πατώντας το κουμπί «Αποθήκευση Αλλαγών» δημιουργείται η ανακοίνωσή μας, βγαίνει μήνυμα επιτυχούς δημιουργίας και γίνεται redirect στην σελίδα με τις ανακοινώσεις:

ID	Title	Text	Actions
1	test gia to report	Αυτό είναι μία δοκιμαστική ανακοίνωση	<button>Ενημέρωση</button> <button>Διαγραφή</button>

Όπως θα δούμε, η ανακοίνωση πλέον είναι και στο αντίστοιχο Collection:

```
{
  "_id": ObjectId('66cede1cc956997ce50b4c3c'),
  "ann_id": "1",
  "title": "test gia to report",
  "text": "Αυτό είναι μία δοκιμαστική ανακοίνωση"
}
```

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιας ανακοίνωσης επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:

Ενημέρωση Ανακοίνωσης

ID Ανακοίνωσης: 1

Τίτλος Ανακοίνωσης: test gia to report

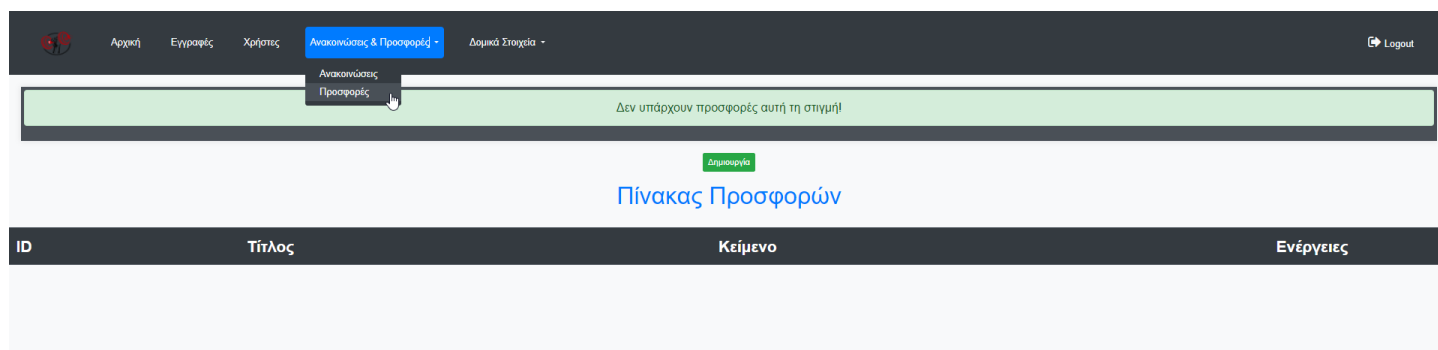
Κείμενο Ανακοίνωσης: Αυτό είναι μία δοκιμαστική ανακοίνωση

Αποθήκευση Αλλαγών Ακύρωση

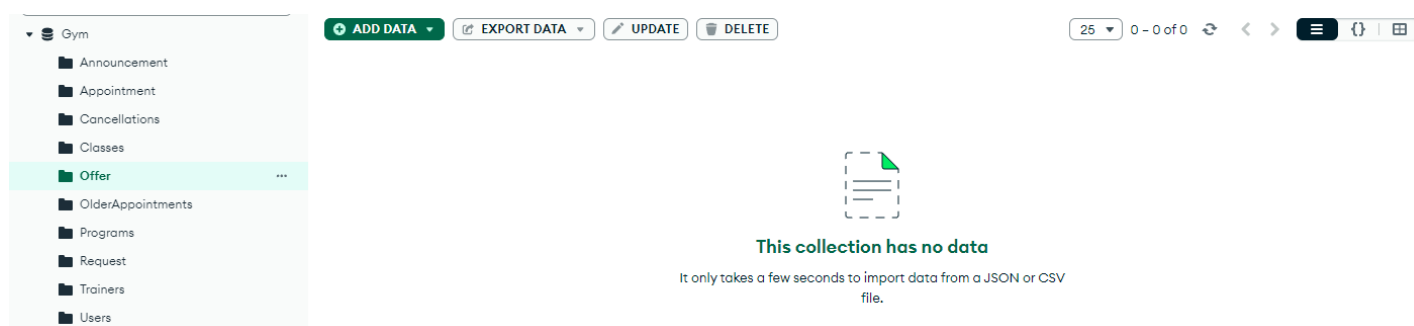
Επιπλέον μπορεί να διαγράψει όποια ανακοίνωση επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει την ανακοίνωση και από το Collection Announcement.

Η επιλογή «Ανακοινώσεις & Προσφορές»:

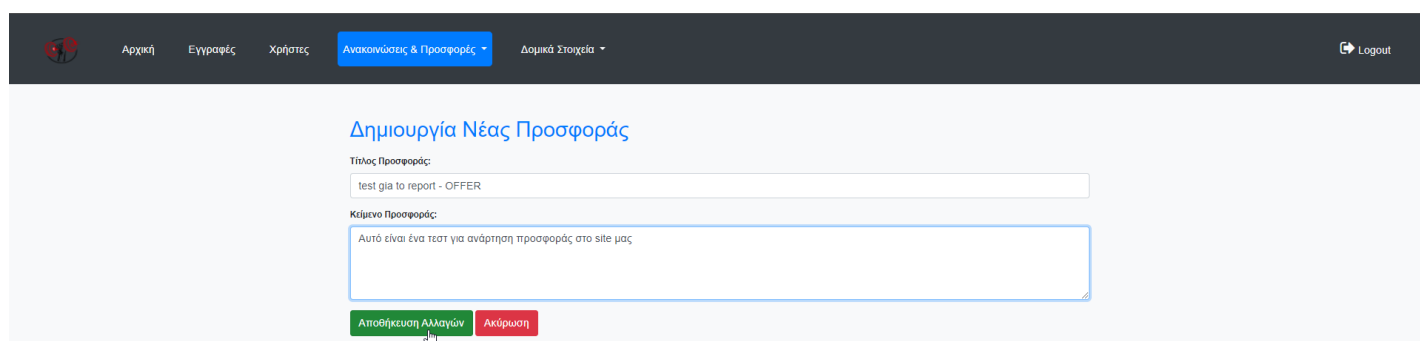
Υπό επιλογή -> Προσφορές:



Οι προσφορές αποθηκεύονται στη βάση στο collection Offer. Δεν υπάρχει καμία ανακοίνωση όπως βλέπουμε στο UI της σελίδας και το επαληθεύουμε κοιτάζοντας το collection το οποίο είναι άδειο:



Πατώντας το κουμπί «Δημιουργία» πάμε να δημιουργήσουμε μία νέα προσφορά:



Πατώντας το κουμπί «Αποθήκευση Αλλαγών» δημιουργείται η προσφορά μας , βγαίνει μήνυμα επιτυχούς δημιουργίας και γίνεται redirect στην σελίδα με τις ανακοινώσεις:

ID	Τίτλος	Κείμενο	Ενέργειες
1	test gia to report - OFFER	Αυτό είναι ένα τεστ για ανάρτηση προσφοράς στο site μας...	Ενημέρωση Διαγραφή

Όπως θα δούμε , η προσφορά πλέον είναι και στο αντίστοιχο Collection:

Gym

Announcement

Appointment

Cancellations

Classes

Offer

OlderAppointments

Programs

Request

Trainers

Users

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 1 of 1

<

>

⋮

⌵

⌶

```
_id: ObjectId('66cee217c956997ce50b4c3d')
offer_id: "1"
title: "test gia to report - OFFER"
text: "Αυτό είναι ένα τεστ για ανάρτηση προσφοράς στο site μας"
```

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιας προσφοράς επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:

ΑρχικήΕγγραφέςΧρήστεςΑνακοινώσεις & ΠροσφορέςΔομικά ΣτοιχείαLogout

Ενημέρωση Προσφοράς

ID Προσφοράς:

1

Τίτλος Προσφοράς:

test gia to report - OFFER

Κείμενο Προσφοράς:

Αυτό είναι ένα τεστ για ανάρτηση προσφοράς στο site μας

Αποθήκευση Αλλαγών

Ακύρωση

Επιπλέον μπορεί να διαγράψει όποια προσφορά επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει την ανακοίνωση και από το Collection Offer.

Η επιλογή «Δομικά Στοιχεία»:

Υπό επιλογή -> Γυμναστές:

The screenshot shows the top navigation bar of the application. It includes a logo on the left, followed by links: Αρχική, Εγγραφές, Χρήστες, Ανακοινώσεις & Προσφορές, and Δομικά Στοιχεία. The 'Δομικά Στοιχεία' link is highlighted in blue. Below it, a dropdown menu is open, showing options: Γυμναστές (highlighted), Προγράμματα Γυμναστηρίου, and Πρόγραμμα Τάξεων. To the right of the dropdown, a green banner displays the text 'γυμναστές αυτή τη στιγμή!'. Further right is a 'Logout' link. Below the navigation bar, a green button labeled 'Δημιουργία' is visible, followed by the heading 'Πίνακας Γυμναστών'.

Οι Γυμναστές αποθηκεύονται στη βάση στο collection Trainers. Δεν υπάρχει κανένας γυμναστής στο γυμναστήριο μας όπως βλέπουμε στο UI της σελίδας και το επαληθεύουμε κοιτάζοντας το collection το οποίο είναι άδειο:

The screenshot shows a database management interface for the 'Trainers' collection. On the left is a sidebar with a tree view containing folders like Announcement, Appointment, Cancellations, Classes, Offer, OlderAppointments, Programs, Request, Trainers (highlighted), and Users. The main area has a header with buttons: ADD DATA, EXPORT DATA, UPDATE, and DELETE. Below the header, there is a message: 'This collection has no data. It only takes a few seconds to import data from a JSON or CSV file.' The message is accompanied by a document icon.

Πατώντας το κουμπί «Δημιουργία» πάμε να δημιουργήσουμε μία νέα καταχώρηση γυμναστή στο γυμναστήριο μας:

The screenshot shows a form titled 'Δημιουργία Νέου Γυμναστή'. The form contains several input fields: Ονομα (filled with 'Στέλιος'), Επώνυμο (filled with 'Κόνος'), Προγράμματα (filled with 'Πιλάτες / Γιόγκα'), Βιογραφικό (filled with 'Πλούσιο βιογραφικό - Απόφοιτος ΤΕΦΑΑ Αθηνών με παραστάσεις σε γυμναστήρια εξωτερικού'), Email (filled with 'konos@gmail.com'), and Τηλέφωνο (filled with '6987786545'). At the bottom of the form are two buttons: 'Αποθήκευση Αλλαγών' (highlighted in green) and 'Ακύρωση' (highlighted in red).

Πατώντας το κουμπί «Αποθήκευση Αλλαγών» δημιουργείται η καταχώρηση μας , βγαίνει μήνυμα επιτυχούς δημιουργίας και γίνεται redirect στην σελίδα με τους γυμναστές:

ID	Όνομα	Επίθετο	Προγράμματα	Βιογραφικό	Email	Τηλέφωνο	Ενέργειες
1	Στέλιος	Κόνος	Πιλάτες / Γιόγκα	Πλούσιο βιογραφικό - Απόφοιτος ΤΕΦΑΑ Αθηνών με παραστάσεις σε γυμναστήρια εξωτερικού	konos@gmail.com	6987786545	Ενημέρωση Διαγραφή

Όπως θα δούμε , ο γυμναστής μας πλέον είναι και στο αντίστοιχο Collection:

Gym

Announcement

Appointment

Cancellations

Classes

Offer

OlderAppointments

Programs

Request

Trainers

Users

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 1 of 1

<

>

≡

{ }

⌂

```
_id: ObjectId('66cee4fbc956997ce50b4c3e')
trainer_id: "1"
name: "Στέλιος"
surname: "Κόνος"
programs: "Πιλάτες / Γιόγκα"
bio: "Πλούσιο βιογραφικό - Απόφοιτος ΤΕΦΑΑ Αθηνών με παραστάσεις σε γυμναστή..."
email: "konos@gmail.com"
phone: "6987786545"
```

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιου γυμναστή επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:

ID Προπονητή:	1
Όνομα:	Στέλιος
Επίθετο:	Κόνος
Προγράμματα:	Πιλάτες / Γιόγκα
Βιογραφικό:	Πλούσιο βιογραφικό - Απόφοιτος ΤΕΦΑΑ Αθηνών με παραστάσεις σε γυμναστήρια εξωτερικού
Email:	konos@gmail.com
Τηλέφωνο:	6987786545
Αποθήκευση Αλλαγών Ακύρωση	

Επιπλέον μπορεί να διαγράψει όποιον γυμναστή επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει τον γυμναστή και από το Collection Trainers.

Η επιλογή «Δομικά Στοιχεία»:

Υπό επιλογή -> Προγράμματα Γυμναστηρίου:

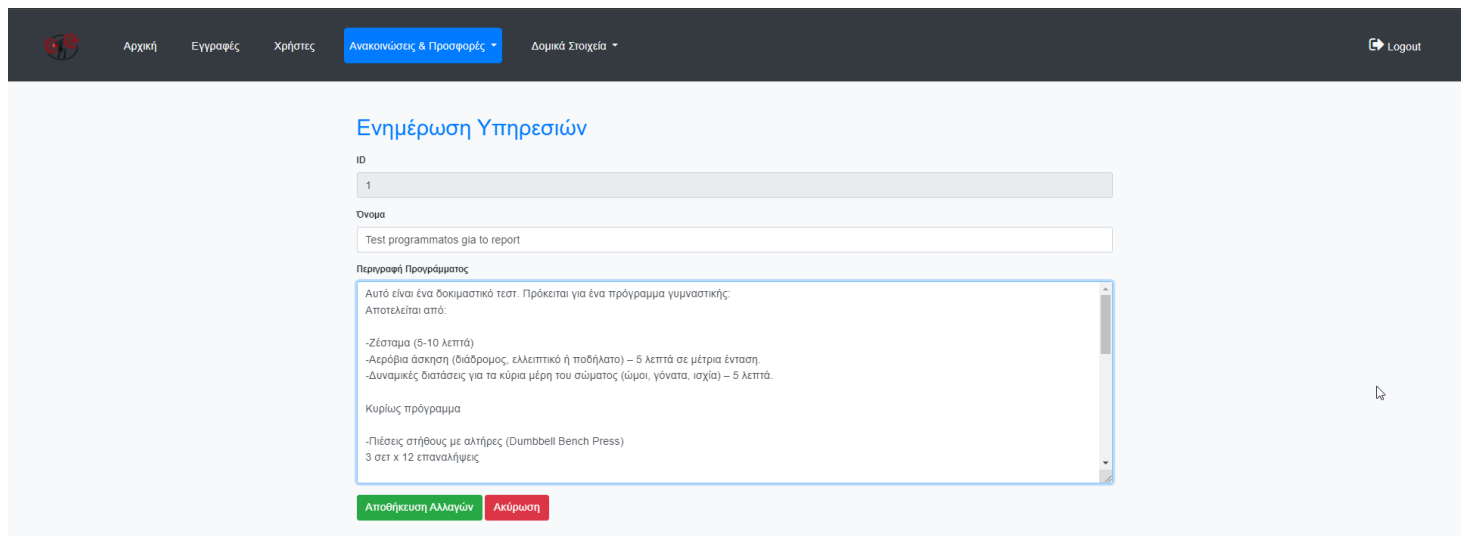
The screenshot shows the top navigation bar of the Gym application. The 'Δομικά Στοιχεία' (Structural Elements) menu is open, showing options: 'Γυμναστές' (Trainers), 'Προγράμματα Γυμναστηρίου' (Gym Programs), and 'Πρόγραμμα Τάξεων' (Classes Program). The 'Προγράμματα Γυμναστηρίου' option is selected. Below the navigation bar, there is a green button labeled 'Δημιουργία' (Create) and a blue header for the 'Πίνακας Υπηρεσιών' (Services Table). The table has columns for 'ID', 'Όνομα' (Name), and 'Ενέργειες' (Actions).

Τα προγράμματα γυμναστηρίου αποθηκεύονται στη βάση στο collection Programs. Δεν υπάρχει κανένα πρόγραμμα στο γυμναστήριο μας όπως βλέπουμε στο UI της σελίδας και το επαληθεύουμε κοιτάζοντας το collection το οποίο είναι άδειο:

The screenshot shows the MongoDB Compass interface. On the left, the 'Gym' database is selected, and the 'Programs' collection is highlighted. The main area shows a message: 'This collection has no data. It only takes a few seconds to import data from a JSON or CSV file.' The interface includes buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'.

Πατώντας το κουμπί «Δημιουργία» πάμε να δημιουργήσουμε ένα νέο πρόγραμμα:

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιου προγράμματος επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:



The screenshot shows a web application interface with a dark header bar. The header contains a logo on the left, navigation links: Αρχική, Εγγραφές, Χρήστες, Ανακοινώσεις & Προσφορές (highlighted in blue), and Δομικά Στοιχεία. On the right of the header is a Logout button. The main content area has a title 'Ενημέρωση Υπηρεσιών' in blue. Below the title is a form with the following elements:

- ID: A text input field containing the number '1'.
- Όνομα: A text input field containing 'Test programmatos gia to report'.
- Περιγραφή Προγράμματος: A large text area containing the following text:

Αυτό είναι ένα δοκιμαστικό τεστ. Πρόκειται για ένα πρόγραμμα γυμναστικής.
Αποτελείται από:

 - Ζέσταμα (5-10 λεπτά)
 - Αερόβια άσκηση (διαδρόμος, ελλειπτικό ή ποδήλατο) – 5 λεπτά σε μέτρια ένταση.
 - Δυναμικές διατάσεις για τα κύρια μέρη του σώματος (ώμοι, γόνατα, ισχία) – 5 λεπτά.

Κυρίως πρόγραμμα

 - Πιέσεις στήθους με αλτήρες (Dumbbell Bench Press)

3 σετ x 12 επαναλήψεις

At the bottom of the form are two buttons: 'Αποθήκευση Αλλαγών' (green) and 'Ακύρωση' (red).

Τέλος, μπορεί να διαγράψει όποιο πρόγραμμα επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει τον γυμναστή και από το Collection Programs.

Η επιλογή «Δομικά Στοιχεία»:

Υπό επιλογή -> Πρόγραμμα τάξεων:

ΑρχικήΕγγραφέςΧρήστεςΑνακοινώσεις & ΠροσφορέςΔομικά ΣτοιχείαLogout

Γυμναστής

Προγράμματα Γυμναστηρίου

Πρόγραμμα Τάξεων

Δημιουργία

Πίνακας Τάξεων

ID	Name	Surname	Programs	Biography	Email	Phone	Actions
----	------	---------	----------	-----------	-------	-------	---------

Τα προγράμματα τάξεων αποθηκεύονται στη βάση στο collection Classes. Δεν υπάρχει κανένα πρόγραμμα τάξης στο γυμναστήριο μας όπως βλέπουμε στο UI της σελίδας και το επαληθεύουμε κοιτάζοντας το collection το οποίο είναι άδειο:

Gym

- Announcement
- Appointment
- Cancellations
- Classes
- Offer
- OlderAppointments
- Programs
- Request
- Trainers
- Users

ADD DATAEXPORT DATAUPDATEDELETE

250 - 0 of 0

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Πατώντας το κουμπί «Δημιουργία» πάμε να δημιουργήσουμε μία νέα τάξη:

ΑρχικήΕγγραφέςΧρήστεςΑνακοινώσεις & ΠροσφορέςΔομικά ΣτοιχείαLogout

Δημιουργία Νέας Τάξης

Πρόγραμμα:

pilates

Ημέρα:

Monday

Ωρα:

12:00

Τύπος Τάξης:

Ομαδικό

Μεγιστος αριθμος απομιν:

100

Ελευθερες θεσεις:

88

Αποθήκευση Αλλαγών

Ακύρωση

Πατώντας το κουμπί «Αποθήκευση Αλλαγών» δημιουργείται τάξης μας , βγαίνει μήνυμα επιτυχούς δημιουργίας και γίνεται redirect στην σελίδα με τις τάξεις:

Δημιουργία							
Πίνακας Τάξεων							
ID	Program	Day	Time	Type	Max People	Available Spots	Actions
1	pilates	Monday	12:00	group	100	88	<div>Ενημέρωση</div> <div>Διαγραφή</div>

Όπως θα δούμε , η τάξη μας πλέον είναι και στο αντίστοιχο Collection:

Gym

Announcement

Appointment

Cancellations

Classes

Offer

OlderAppointments

Programs

Request

Trainers

Users

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 1 of 1

<

>

≡

{ }

⌂

```
_id: ObjectId('66ceee41c956997ce50b4c40')
class_id: "1"
program: "pilates"
day: "Monday"
time: "12:00"
type: "group"
max_people: "100"
availablespots: "88"
```

Ο διαχειριστής έχει επιπλέον τη δυνατότητα να αλλάξει τα στοιχεία όποιας τάξης επιθυμεί με το κουμπί «Ενημέρωση». Η οθόνη της ενημέρωσης είναι η ακόλουθη:

ID Τάξης:	1
Προγραμμα:	pilates
Ημέρα:	Monday
Ωρα:	12:00
Τύπος Τάξης:	group
Μεγιστος αριθμος ατομων:	100
Ελευθερος χωρος:	88
<div>Αποθήκευση Αλλαγών</div> <div>Ακύρωση</div>	

Τέλος, μπορεί να διαγράψει όποια τάξη επιθυμεί με το κουμπί διαγραφή. Αυτό διαγράφει τη τάξη και από το Collection Classes.

Εφόσον ο διαχειριστής έδωσε έγκριση στον χρήστη και αποδέχτηκε το αίτημα του είμαστε έτοιμοι να δείξουμε τις οθόνες του χρήστη.

Στοιχεία σύνδεσης:

Όνομα χρήστη: kreouzis

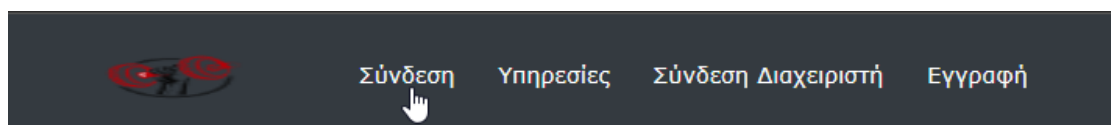
Κωδικός πρόσβασης: 123

Ορίστε και στη βάση – στο collection Users - ο χρήστης:




```
_id: ObjectId('66ce444d443675474ffaed9')
user_id: "2"
name: "antonios"
surname: "kreouzis"
country: "Greece"
town: "Galatsi"
username: "kreouzis"
password: "123"
email: "kreouzis@gmail.com"
address: "Olympou 15 - Chalandri"
role: "user"
```

Κάνουμε αποσύνδεση από το διαχειριστικό σύστημα και τώρα συνδεόμαστε στο σύστημα των χρηστών. Επομένως από το αρχικό μενού επιλέγουμε:



Στην συνέχεια συμπληρώνουμε τα στοιχεία του χρήστη:



Έχετε αποσυνδεθεί

Ψευδώνυμο

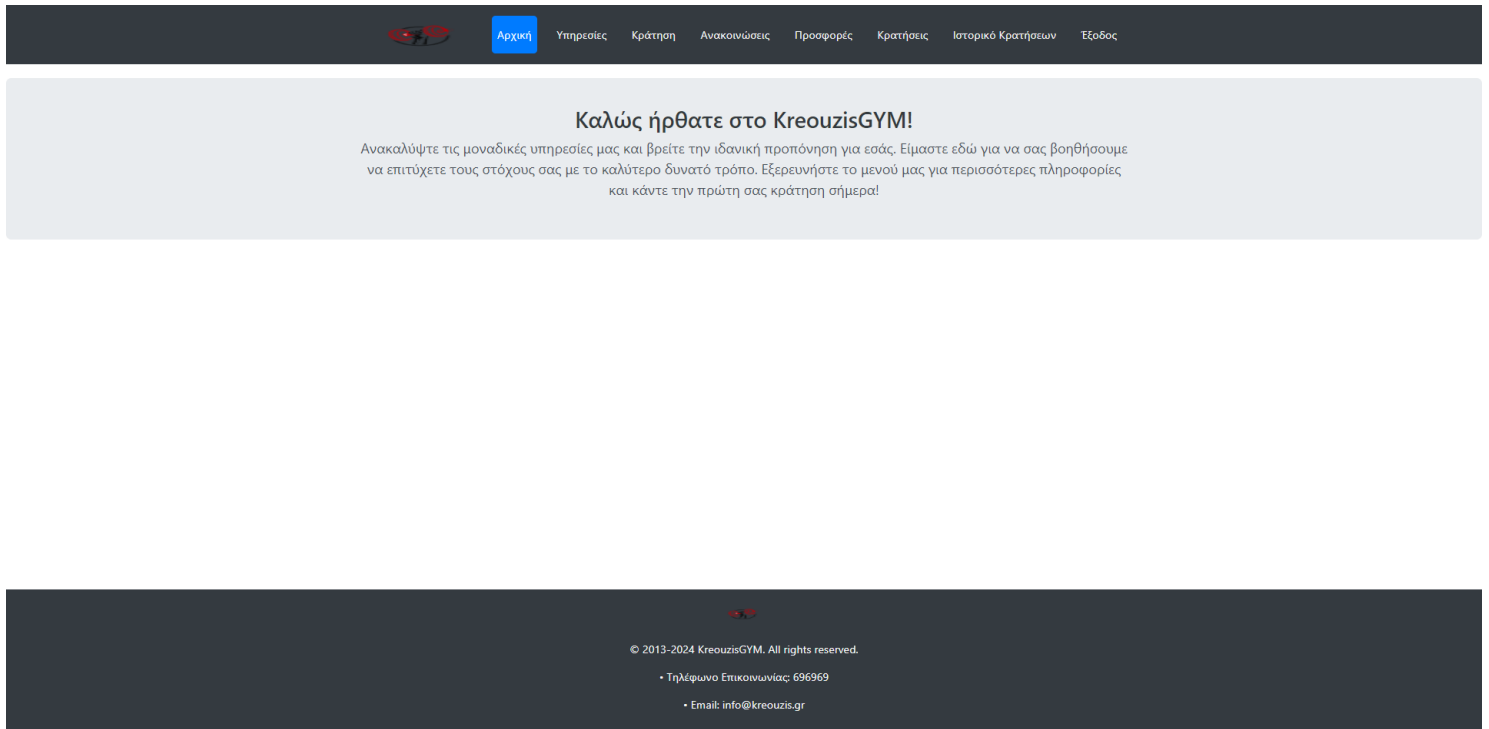
Συνθηματικό

Σύνδεση

[Επιστροφή](#)

Αρχική σελίδα χρήστη:

Όταν ο χρήστης κάνει επιτυχή σύνδεση , η σελίδα που του εμφανίζεται είναι η ακόλουθη:



Σελίδα με τα προγράμματα γυμναστικής:

Επιλέγοντας «Υπηρεσίες» από το μενού , ο χρήστης βλέπει την ακόλουθη οθόνη:

Archiki **Υπηρεσίες** Κράτηση Ανακοινώσεις Προσφορές Κρατήσεις Ιστορικό Κρατήσεων Έξοδος

Προγράμματα του Γυμναστηρίου μας

Test programmatos gia to report

Αυτό είναι ένα δοκιμαστικό τεστ. Πρόκειται για ένα πρόγραμμα γυμναστικής:

Κάθε Τετάρτη στις 12

Αποτελείται από:

- Ζέσταμα (5-10 λεπτά)
- Αερόβια άσκηση (διάδρομος, ελλειπτικό ή ποδήλατο) – 5 λεπτά σε μέτρια ένταση.
- Δυναμικές διατάσεις για τα κύρια μέρη του σώματος (ώμοι, γόνατα, ισχία) – 5 λεπτά.

Κυρίως πρόγραμμα

- Πιέσεις στήθους με αλτήρες (Dumbbell Bench Press)

© Kreouzis Antonis - Kreouzis GYM. All rights reserved.

• Τηλέφωνο Επικοινωνίας: 69696969

• Email: info@kreouzis.gr

Εδώ ο χρήστης βλέπει τα προγράμματα που έχει φτιάξει ο διαχειριστής. Το είδαμε προηγουμένως , τα προγράμματα αποθηκεύονται στη βάση στο Collection Programs:

Gym

- Announcement
- Appointment
- Cancellations
- Classes
- Offer
- OlderAppointments
- Programs**
- Request
- Trainers
- Users

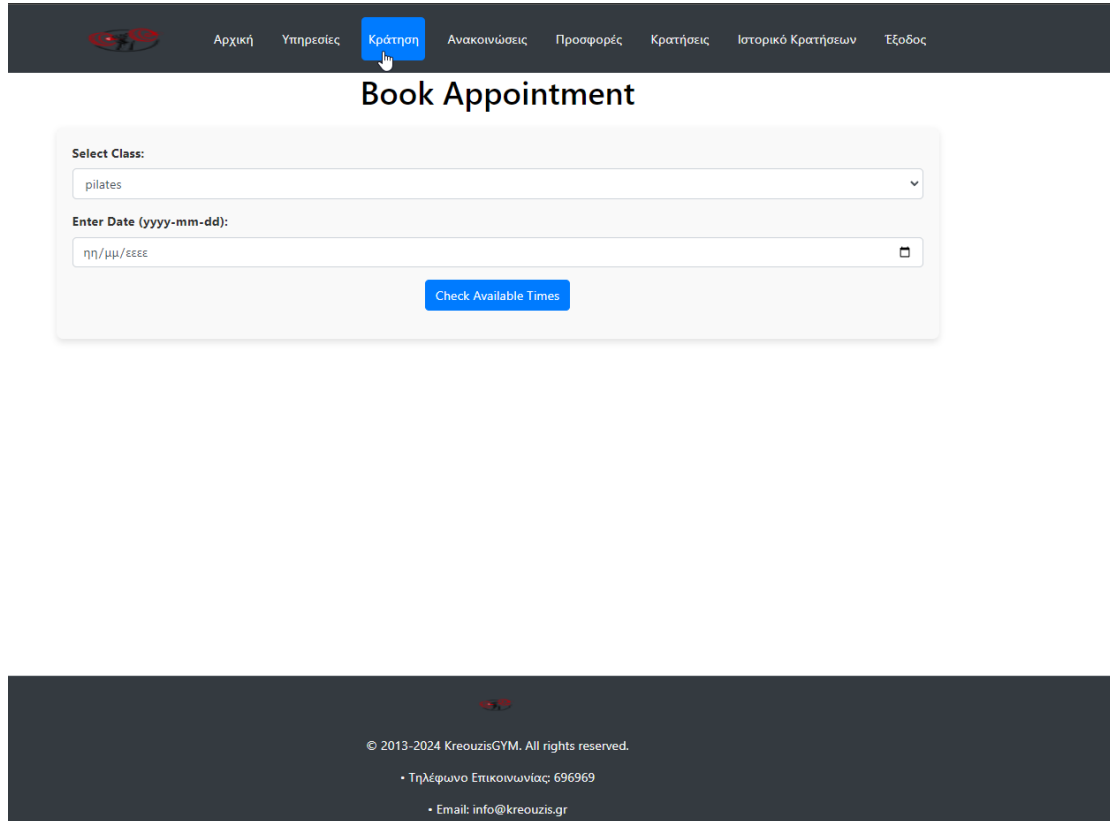
ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 1 of 1

<pre>{ "_id": ObjectId("66ceeaf9c956997ce50b4c3f"), "program_id": "1", "name": "Test programmatos gia to report", "text": "Αυτό είναι ένα δοκιμαστικό τεστ. Πρόκειται για ένα πρόγραμμα γυμναστικής..." }</pre>

Σελίδα με τη διαδικασία κράτησης προγράμματος:

Επιλέγοντας την «Κράτηση» από το μενού , οι χρήστες μπορούν να κλείσουν θέση για τα διαθέσιμα μαθήματα:



The screenshot shows a web application interface for booking appointments. At the top is a dark navigation bar with a logo on the left and several menu items: Αρχική, Υπηρεσίες, Κράτηση (highlighted with a blue button and a mouse cursor), Ανακοινώσεις, Προσφορές, Κρατήσεις, Ιστορικό Κρατήσεων, and Έξοδος. Below the navigation bar, the title 'Book Appointment' is centered. The main content area contains a light gray form with two input fields: 'Select Class:' with a dropdown menu showing 'pilates', and 'Enter Date (yyyy-mm-dd):' with a text input showing 'ηη/μμ/εεεε' and a calendar icon. A blue button labeled 'Check Available Times' is positioned below the date field. At the bottom of the page is a dark footer bar containing the copyright notice '© 2013-2024 KreouzisGYM. All rights reserved.', the phone number '• Τηλέφωνο Επικοινωνίας: 696969', and the email address '• Email: info@kreouzis.gr'.

Στο Select Class επιλέγουν την τάξη και στο Enter Date επιλέγουν την ημέρα.

Στο συγκεκριμένο παράδειγμα , οι χρήστες από το μενού «Υπηρεσίες» βλέπουν πως οι πιλάτες είναι κάθε Τετάρτη στις 12.

Μόλις επιλέξουν και ημέρα , πρέπει να επιλέξουν ώρα:

Book Appointment

Class Selected:

Enter Date (yyyy-mm-dd):

Select Time:

12:00

12:00


Submit Appointment

Μόλις η κράτηση γίνει , ο χρήστης γίνεται redirect στην οθόνη «Κρατήσεις».


ID	Day	Week	Text	Actions
1	pilates	28/8/2024	12:00	Διαγραφή

Μόλις γίνει η κράτηση , στο collection Appointment περνάει η κράτηση και ο χρήστης μπορεί να τη δει και φυσικά να τη διαγράψει. Εάν τη διαγράψει , φεύγει και από τη βάση προφανώς.

Σελίδα με το ιστορικό κρατήσεων:

[Αρχική](#) [Υπηρεσίες](#) [Κράτηση](#) [Ανακοινώσεις](#) [Προσφορές](#) [Κρατήσεις](#) [Ιστορικό Κρατήσεων](#) [Έξοδος](#)

Ημέρα	Ωρα	Πρόγραμμα	Τύπος
Wednesday	12:00	pilates	group



© 2013-2024 KreouzisGYM. All rights reserved.

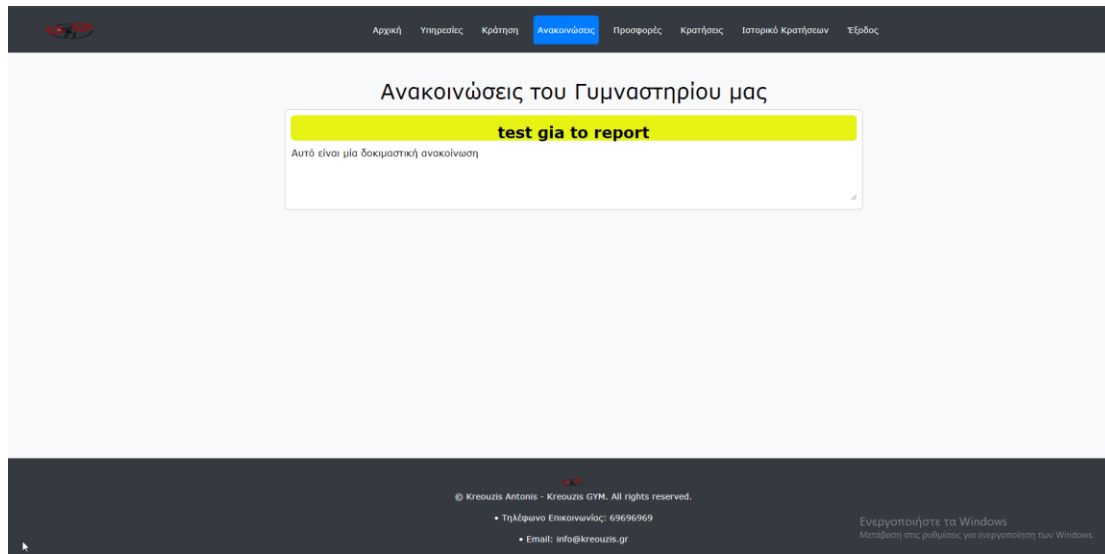
• Τηλέφωνο Επικοινωνίας: 696969

• Email: info@kreouzis.gr

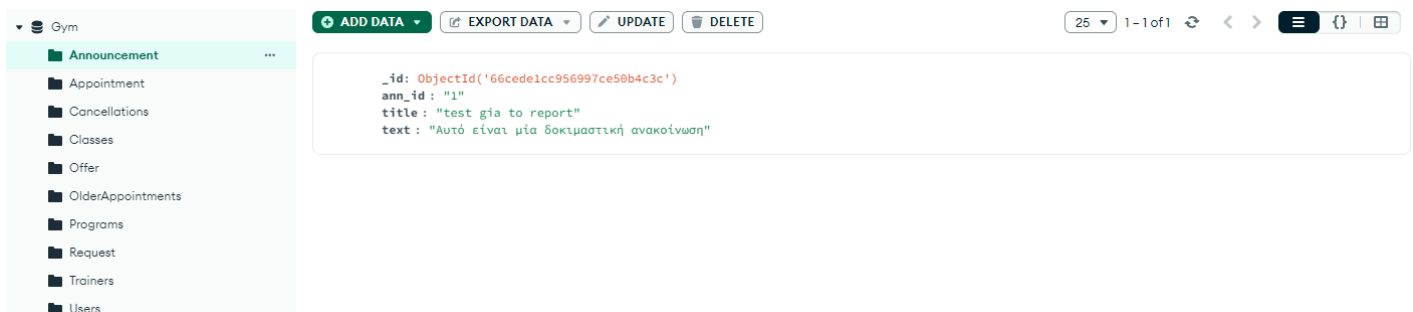
Μόλις ο χρήστης πραγματοποιήσει μία κράτηση , αυτή αποθηκεύεται στο Collection Appointment. Ταυτόχρονα όμως αποθηκεύεται και στην OlderAppointments έτσι ώστε να εμφανίζουμε το ιστορικό κρατήσεων του χρήστη.

Εάν διαγραφεί από την appointment τότε διαγράφεται και από το OlderAppointments.

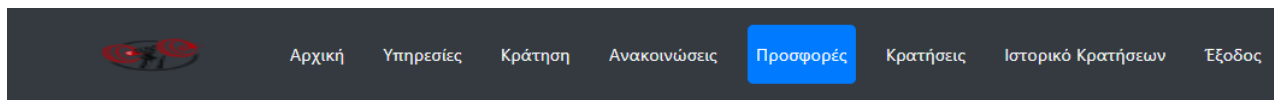
Σελίδα με τις ανακοινώσεις:



Σε αυτή τη σελίδα ο χρήστης βλέπει τις ανακοινώσεις που βγάζει το γυμναστήριο. Στην παραπάνω εικόνα η ανακοίνωση που φτιάξαμε από τη σελίδα του admin στην προηγούμενη ενότητα. Εδώ εμφανίζονται οι ανακοινώσεις που αποθηκεύονται στη βάση στο collection Announcement:



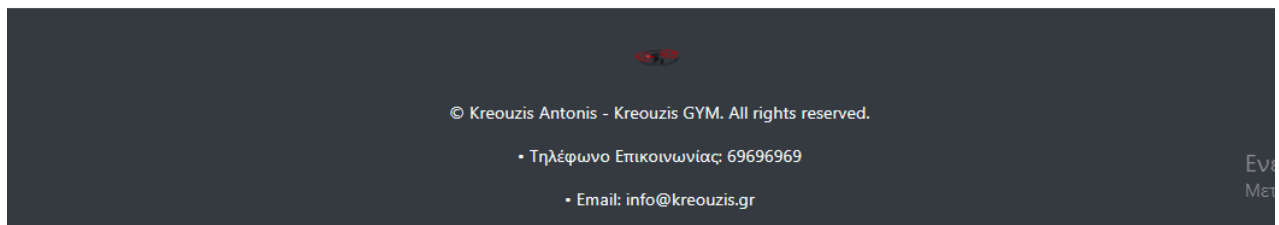
Σελίδα με τις προσφορές:



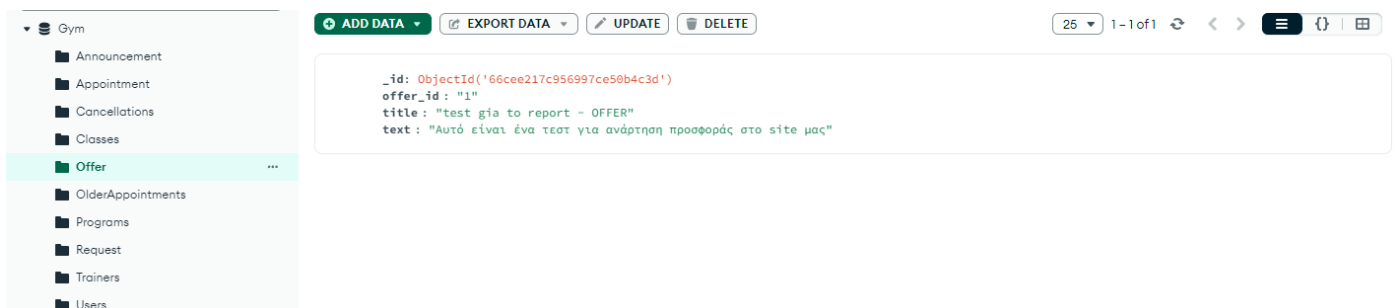
Προσφορές του Γυμναστηρίου μας

test gia to report - OFFER

Αυτό είναι ένα τεστ για ανάρτηση προσφοράς στο site μας



Σε αυτή τη σελίδα ο χρήστης βλέπει τις προσφορές που βγάζει το γυμναστήριο. Στην παραπάνω εικόνα η ανακοίνωση που φτιάξαμε από τη σελίδα του admin στην προηγούμενη ενότητα. Εδώ εμφανίζονται οι ανακοινώσεις που αποθηκεύονται στη βάση στο collection Offer:



Αυτή ήταν η παρουσίαση της δικής μου εργασίας.

Ευχαριστώ πολύ για το χρόνο σας.

Με εκτίμηση,

Αντώνης Παπακωνσταντίνου.